Doctoral Dissertation
Doctoral Program in Computer and Control Engineering (34$^{th}$cycle)

# Image Processing for Machine Vision Applications

By

**Sorath Asnani**

******

**Supervisor**
Prof. Bartolomeo Montrucchio

**Doctoral Examination Committee:**
Prof. Claudio Fornaro, Referee, Universita Telematica Internazionale Uninettuno
Prof. Fiaz Gul Khan, Referee, COMSATS Institute of Information Technology
Prof. Dr. Mukhtiar Ali Unar, Mehran University of Engineering and Technology
Prof. Maurizio Rebaudengo, Politecnico di Torino
Prof. Sanam Narejo, Mehran University of Engineering and Technology

Politecnico di Torino
July 2022

# Declaration

I hereby declare that, the contents and organization of this dissertation constitute my own original work and does not compromise in any way the rights of third parties, including those relating to the security of personal data.

Sorath Asnani
July 2022

* This dissertation is presented in partial fulfillment of the requirements for **Ph.D. degree** in the Graduate School of Politecnico di Torino (ScuDo).

*I would like to dedicate this thesis to my loving parents*

# Acknowledgements

I would take this opportunity to express my sincere and heartfelt gratitude to everyone who supported me throughout my Ph.D. journey and contributed towards turning my dream into reality.

After the Almighty, the Creator of all; my first and foremost gratitude goes to my Ph.D. supervisor, Prof. Bartolomeo Montrucchio. His utmost support, guidance, mentor-ship, knowledge, encouragement and most importantly his believe in me, has played a major role in molding me into a valuable research scientist. I admire him both as a skillful researcher and as a wonderful human being. It was due to him that my Ph.D. journey was smooth and interesting. Thank you very much dear professor.

I am equally thankful to Renato Ferrero for his constant guidance and support during my Ph.D. He is indeed a smart researcher, a kind and a caring human being. I extend my gratitude to Prof. Maurizio Rebaudengo for his valuable collaboration in a major part of my research project.

I would like to extend my gratitude to my dissertation referees, Prof. Claudio Fornaro and Prof. Fiaz Khan, whose wise comments and suggestions helped me to improve my dissertation.

My whole-hearted thanks goes to the fellow members of Lab-2, who were always there for me whenever I needed to talk. Time spent with them will always be in my good memories. Thanks to Edoardo and Muhammad for their constant support and guidance.

I can never forget the sacrifice of my family for sending me abroad for pursuing Masters and Ph.D. from Italy. I have no words to express my feelings of love and gratitude for them. It was due to their unconditional love, endless support, encouragement, and trust that I was able to face the toughest situations in my life. Thank you Maa, Papa, Munesh, Sadhna and Natasha for always being there for me.

The words “thank you” are not enough to express my deepest gratitude to my husband, Rohit Hansrajani. Although miles apart, he was always with me through every thick and thin. He pushed me hard to complete my tasks in time. Had he not encouraged me or believed in me, I am afraid I would not be writing this today, I am afraid I would be far away from achieving my goals. His constant words, “I am always with you in every situation”, are my life time strength.

Thanks to my in-laws family members for their constant prayers and best wishes. I am truly grateful for your love and support you have provided towards the fulfilment of my dreams.

I would like to pay my sincere gratitude to Prof. Sanam Narejo, my teacher, my mentor, for all the emotional and moral support she provided at the most crucial time when I was stressed out and clueless on how to proceed in certain situations. My whole-hearted gratitude goes to Prof. Dr. Mukhtiar Unar, my mentor. He has always been an inspiration and a source of motivation for me.

I would greatly acknowledge all my friends, both from Italy and Pakistan, for providing me with unfailing support, even at distance. Zunera, Saira, Aisha, Pirah, Anam, and Sofia, all of you are my real gems. Thank you for everything you did for me. May all of you achieve the best of everything in your lives.

# Abstract

Images are a vital part of our everyday life and Image Processing is the heart of all the modern technologies, including machine vision, artificial intelligence, robotics, deep learning. It would not be wrong to say that image processing is one of the many reasons for achieving success in any industrial domain, whether it be medical, food, textile, or any other automation industry. It is next to impossible to work in these domains without having sufficient knowledge and skills about image processing techniques.

In this thesis document you will find the significance of image processing used in three diverse projects. Each one of the projects is described as a separate chapter in this document.

The first project is focused on reducing the power consumption in OLED-based devices. Actually there are two main goals of this project, first one, as the name suggests, is to minimize the power consumed by an OLED device to display images, and the second goal is to simultaneously enhance the color contrasts in images. OLED display panels have become increasingly popular in recent years, thanks to their numerous advantages over the traditional LCD displays. Power consumption in OLED displays depends on the contents where as the backlight is responsible for power consumption in LCD displays. This image-dependent or content-dependent power consumption model of OLED displays have encouraged numerous researchers to create possibilities for reducing the power consumption in OLED-based devices. One such possibility has been explored in this Ph.D. research work.

Another industrial application has been presented in the second part of the thesis document. It is a part of the “Food Digital Monitoring” project, funded by Regione Piemonte. The major aim of this project is to identify the healthy and contaminated hazelnuts by using fluorescence and spectral imaging techniques. Two types of contamination are discussed in this work, one, caused by bacterial and fungal

infections, called “rotten hazelnuts” and the other caused by insect bites, known as “pest-infected hazelnuts”. The idea is to illuminate the hazelnut samples under UV (ultraviolet) light using an axial illuminator setup. When the hazelnuts are illuminated with the UV light, its molecules get excited and as a result, they emit fluorescent light. The amount of fluorescence depends on the quality of hazelnuts. The emitted light is captured in the form of images and then by applying some simple yet powerful image processing techniques, the healthy, rotten and pest-infected hazelnuts are classified into their respective classes.

The last part of the thesis is concerned with measuring the quality of built-in smartphone cameras. With the advancement in smartphone technology, the use of social networking applications have also raised significantly, and the increase in the social media has contributed towards the production of graphical contents in the form of images and videos. The core component for generating this type of data is camera. The latest phones are introducing more advanced features for their built-in cameras. The manufacturers are making efforts to enhance the quality of the smartphone cameras. If we see the technical specifications manual provided by the manufacturers, we will find a great deal of technical parameters under the camera section, for example, image resolution, sensor size, aperture size, focal length, zooming factor, pixel size, etc. Although these are the perfect quantities to define the quality of camera but they are irrelevant for a non-technical person who wants to compare the quality of different mobile phone cameras but is unaware of the technical aspects of photography. It is proposed that along with the technical specifications, the manufacturers should include one additional parameter to represent the quality of their phone camera with a numerical value. As a part of this project, we have developed a mobile application to compute the quality of smartphone cameras numerically by means of Modulation Transfer Function (MTF).

Overall, this research work has provided a great opportunity to me to enhance my knowledge and skills in the area of image processing by working in different application domains. Working on different projects have enabled me to explore different dimensions to further research topics and areas.

# Contents

# List of Figures

# List of Tables

# Nomenclature

**Roman Symbols**

| | |
|---|---|
| AMOLED | Active Matrix Organic Light Emitting Diode |
| CCD | Charge Coupled Device |
| ESF | Edge Spread Function |
| FDM | Food Digital Monitoring |
| FHD | Full High Definition |
| FPS | Frames Per Second |

**Greek Symbols**

| | |
|---|---|
| GLGI | Good Looking Green Images |
| GNSS | Global Navigation Satellite System |
| GPRS | General Packet Radio Service |
| GUI | Graphical User Interface |
| HE | Histogram Equalization |
| ISO | International Organization for Standardization |
| IT | Information Technology |
| LAN | Local Area Network |
| LCD | Liquid Crystal Display |

| | |
|---|---|
| LSF | Line Spread Function |
| mAh | milli Ampere per hour |
| mm | Millimeters |
| MP | Mega Pixels |
| MTF | Modulation Transfer Function |
| OLED | Organic Light Emitting Diode |
| OTF | Optical Transfer Function |
| PCB | Printed Circuit Board |
| PCs | Personal Computers |
| PGM | Portable Gray Map |
| PSF | Point Spread Function |
| QVGA | Quarter Video Graphics Array |

**Superscripts**

| | |
|---|---|
| RGB | Red Green Blue |
| ROI | Region of Interest |

**Subscripts**

| | |
|---|---|
| SIFT | Scale Invariant Feature Transform |
| UV | Ultra Violet |

# Chapter 1

# Introduction

An image is defined as a two-dimensional signal, represented as $f(x,y)$, where $x$ and $y$ are the spatial (horizontal and vertical) co-ordinates and the amplitude of $f$ at any point $(x,y)$ is called the *intensity* of the image at that coordinate location. A digital image is composed of a finite number of picture elements, known as "pixels". Every pixel or the picture element has a particular value and location within the image.

The range of pixel intensity values depends on the type of image representation. For example, in 8-bit representation, the pixel intensity values lie between 0 and $2^8 - 1$, i.e., between 0 (black) and 255 (white). In 16-bit representation, the range of pixel intensity values are from 0 to $2^{16} - 1$, i.e., from 0 (black) to 65535 (white), and so on.

*Digital image processing* refers to the processing of digital images through a digital computer. An image processing system takes an image as input, performs some operations using different algorithms, in order to enhance the image or to extract some useful information, and provides as output either the modified image or extracts some useful attributes from the image.

Image processing can be categorized into three sub-levels: low, mid and high level processing. In low level processing both the inputs and outputs are images. It involves basic image pre-processing operations such as noise reduction in images, contrast enhancement and image sharpening. Inputs to a mid level process are generally images while its outputs are the attributes extracted from those images. Image Segmentation, i.e. partitioning of an image into various segments or regions, and object classification or recognition are the examples of mid level image processing.

Some examples of outputs provided by mid level processes are detected edges, contours, and the recognized objects. Finally the high level image processing is associated with computer vision. It involves *making sense* from a group of recognized objects and take relevant actions to imitate human behaviour.

Images are a vital and integral part of our everyday life. In this modern era of technology, almost all of the technical fields are impacted by image processing in some way. There is a very wide variety of application areas of image processing. Medical diagnostic imaging, X-ray imaging, satellite imaging, microscopic imaging, remote sensing, industrial inspection, image sharpening and restoration, machine vision, color processing, and pattern recognition, etc. are just to name a few applications.

Machine Vision can be defined as the ability of a computer to see. It is the technology that enables various imaging based applications in industrial domain such as automatic inspection, process control and robot guidance. Machine vision is used in a wide variety of industrial applications. Electronic component analysis, currency inspection, materials inspection, medical image analysis, pattern recognition, object recognition, handwriting recognition, optical character recognition, quality inspection, and signature identification are some of the examples of machine vision applications.

## 1.1 Structure of the thesis

The following subsections are aimed at briefly introducing the projects discussed in this thesis and also provide an outlook of the following thesis chapters. The next chapters of this document are based on exhaustive description of these projects.

### 1.1.1 Optimization of Power Consumption in OLEDs

The first part of this thesis is focused on the optimization of power consumption in OLED-based devices. Energy efficiency is one of the major design concerns in this modern era of technology. One of the many aspects to survive and make progress in the manufacturing world of electronic devices is to design *energy-efficient* appliances. Among numerous other electronic devices, tight power constraints are imposed on portable and handheld devices such as smartphones, tablets, notebook computers, laptops, and GNSS (Global Navigation Satellite System) etc. because of their battery-size and life time. Power consumption is critical for these devices as they typically support power-hungry peripherals such as displays for image/video streaming, wireless cards for LAN connectivity, GPRS interface for mobile internet and global communication etc. [1].

Among other energy-draining components, the “Displays”, concerned with the visual representation of data, are the most power-hungry sub-systems, often consuming more than half the laptop or handheld system’s total energy [2]. Specifically, the displays on personal computers (PCs) consume up to 38% of the total power while displays on mobile or handheld devices consume up to 50% of the total energy [3] [4]. The more the presentation of graphical contents on the Display, the more is the power consumption of the device, and vice versa. Consequently, the battery life is decreased due to extensive usage of graphical data.

LCD and OLED are two popular display types. An OLED (Organic Light Emitting Diode) is the modern display technology in smartphones and televisions. OLED is different from the LCD as the energy consumption in the former depends on the displayed contents, i.e. the color and brightness of the image being displayed contribute towards the power consumption in OLEDs, where as the power consumption in LCDs is driven by the backlight.

The increased use of handheld technology has given rise to various social networking platforms such as Facebook, Instagram. Snapchat, and Tiktok etc. Most of the social networking platforms encourage rich graphical contents, such as images and videos. Millions of images and videos are posted on social media on daily basis which causes extended screen time by the users. The extended screen time results in the enormous amount of power drainage in mobile devices and these devices need to get recharged quite often.

An image is composed of various picture elements, also known as *pixels*. A pixel consists of red, green and blue color components, most commonly represented as RGB color components. Research has shown that the blue color component requires more energy as compared to the red and green color components. If the intensity of blue color is reduced in some way, the total energy can be reduced in energy-aware devices, such as OLED displays.

In this work, a novel idea has been proposed for producing energy-efficient images, which consume less energy when displayed on an OLED device. In addition to that, some image enhancement algorithms have also been used to enhance color contrasts in images to improve the visual quality of images. One simple way to reduce the power consumption in energy-aware devices, such as OLEDs, would be to reduce the overall intensity of the red, green and blue pixel components. This approach, in turn, would adversely affect the visual quality of images. To improve the perceptual quality, some image-enhancement techniques would be required to compensate for the deteriorated visual experience. This two-fold strategy, i.e. reducing the RGB intensity level, and enhancing the color contrast of images, is the major idea of the proposed work.

The detailed explanation of this work is provided in Chapter 2.

### 1.1.2 Automatic Inspection of Hazelnuts

The second part of the thesis is focused on the quality assessment of hazelnuts. It is a part of the Food Digital Monitoring (FDM) project, funded by Regione Piemonte, Italy.

Quality inspection of food items is one of the major application areas of Machine Vision. This area of research has contributed significantly to the food industry. Food processing is undergoing vast changes to yield products of ever higher added value, while at the same time there is the possibility of harm through bacteria and micro-toxins, and a great many additives and contaminants can be regarded as undesirable and/or unattractive for human consumption. As a result, careful control of the manufacturing process is required and high standards of quality control are in increasing demand. Digital images are key to the possibilities for attaining such control, as a single compact sensor can provide huge amounts of crucial information at extremely high rates [5].

The aim of this study was to use the fluorescence and spectral imaging techniques to categorize hazelnuts into contaminated and healthy classes. The contaminated hazelnuts were further categorized into pest-infected and rotten categories. Rotten hazelnuts are usually contaminated by bacterial and fungal infections, where as the pest-infected hazelnuts are caused by pests/insects bites. To produce fluorescence in hazelnut samples, the ultraviolet (UV) light has been used as the source of illumination. This project also involves the 3D model designing and 3D printing to create a case for an axial illumination system for analyzing the hazelnut samples.

Traditionally, the quality assessment, separation and classification of hazelnuts (or most of the food items) is done by human labor in various food processing industries. This method is highly time consuming, labor intensive, tedious and expensive. For this reason, an automatic, machine vision based, cost-effective, and faster method is required for examining the quality of hazelnuts. Our proposed system allows for the automatic inspection of the quality of hazelnuts by applying simple image processing techniques.

This work is discussed in detail in Chapter 3.

### 1.1.3 Measuring the Smartphone Camera Quality

The last part of this thesis is concerned with measuring the quality of built-in cameras in smartphones.

Life nowadays cannot even be imagined without having a smartphone. It has become a necessity for human life rather than a luxury for various reasons, unless you decide to lead a very different life than rest of the humanity.

In this era of modern technology, and competitive world, almost everyone wants to have the best of everything in his/her life. Best house, best appliances, best smartphones, best cameras etc. But how do we know which camera is the best?

Traditionally the quality of a camera is determined by capturing images and recording video clips of various subjects and objects around us, and then looking at those images and videos to understand the camera quality. However, this method of quality assessment is vague and highly depends on the observer.

If we look at a smartphone's technical specifications provided by the manufacturers, we would notice that in the section of camera specifications, there is plenty of details available regarding various parameters, such as number of cameras, image resolution, aperture size, lens type, focal length, sensor size, pixel size, and magnification capacity etc. All these terms are meaningful and makes sense for an expert of photography. But unfortunately, not everyone is an expert in this field. Therefore, such detailed information is often useless for a common person.

What is missing in those specifications is a single metric, we call it as the *camera quality*. The aim of this project was to provide a single numerical value to quantify the quality of a built-in smartphone camera. This was done by means of Modulation Transfer Function (MTF). A mobile application was also developed under this project to directly measure the quality of a phone camera.

This work is presented in Chapter 4 with detailed explanation.

# Chapter 2

# Optimization of Power Consumption in OLED-based Devices

As discussed in Chapter 1, power consumption is one of the most concerned metrics in the design of electronic devices. With the advancement in technology and battery-powered devices, such as smartphones, tablets, notebook computers, smartwatches etc, much attention is directed towards extending battery life of electronic gadgets, so that devices can be used for longer periods of time without the need of charging them frequently.

In this chapter we will discuss in detail an application area of image processing related to the optimization of power consumption in OLED-based devices. In particular, we will see that how images can be modified in such a way that they consume less power when displayed on an OLED device and their color contrast is enhanced as well, to provide a better visual experience to end-users.

Most of the work presented in this chapter has been originally published in [6] and [7].

## Roadmap

The chapter is organized as follows: Section 2.1 describes the differences between LCDs and OLEDs. Section 2.2 presents the OLED power model and the motivation for the proposed work. Literature review is presented in Section 2.3. The problem statement is presented in Section 2.4. Section 2.5 describes the steps of the proposed method in detail. The experimental results are provided and discussed in Section 2.6 followed by the conclusion in Section 2.7.

## 2.1 LCD vs OLED Display

Modern display panels can be categorized as emissive and non-emissive displays. The two display types have distinct working principles and vary considerably in power consumption characteristics. Emissive displays such as Organic Light-Emitting Diodes (OLEDs) do not require an external light source to display images because its pixels emit their own light, whereas non-emissive displays, such as Liquid Crystal Displays (LCDs) need a constant "backlight" to display contents on the screen.

The energy consumption of a pixel in OLEDs depends on its color and brightness. OLED displays are built from small organic molecules that efficiently emit light when stimulated by an electric field [8]. OLED pixels are only turned on when needed whereas LCD backlights must be fully on during use [9].

Emissive displays have a number of advantages over non-emissive ones, including high contrast and low power consumption [10, 11]. An emissive display can turn off individual pixels to express complete darkness and achieve a high contrast ratio. In addition, each pixel in an emissive display can be driven independently and the power consumption of a pixel is proportional to its intensity level, therefore emissive displays exhibit low power consumption as compared to the non-emissive displays. The key benefit of OLEDs is their ability to vary energy consumption based on the number of pixels turned on as well as the individual pixels' brightness and color [2]. In general OLEDs have better image quality compared to conventional LCDs, better horizontal and vertical viewing angles, higher brightness and faster response times [8].

Backlight accounts for a major proportion of the total power consumption in non-emissive displays i.e. LCDs. Almost 80% of the total energy is consumed by the backlight in LCDs at 200nits brightness [12]. Therefore, several power minimization techniques have been proposed for LCDs based on backlight scaling [1, 12–14]. The backlight scaling is a technique which dynamically dims the backlight to reduce the power dissipation in LCD systems. Although the backlight dimming can significantly reduce power consumption, it often results in distorted displays. As a result of display distortion, image enhancement techniques are usually applied for preserving the image fidelity. These techniques, however, are only limited to LCDs and cannot be applied to emissive displays because of the distinct working principles of the two display types.

Although an OLED system consumes nearly zero energy when displaying a black screen (due to zero pixel values), it may consume more than twice the power required by an LCD system to display a bright image with a white background [15]. Due to the emissive nature of OLEDs, the power consumption is directly proportional to the intensity of image pixels. In other words, the power consumption in OLED depends highly on the displayed contents. This feature motivates researchers to study and develop image processing techniques to reduce power consumption in energy-aware devices.

## 2.2 OLED Power Model and Motivation

In this section we will see the power model i.e. power consumption characteristics of an OLED display and the main motivation behind this work.

Dong et al. [16] presented a pixel-level power model that estimates the power consumption of OLED devices based on red, green and blue (RGB) pixel components. The power consumption of an image displayed on an OLED system is the sum of each individual pixel's power consumption, which in turn is the sum of the power consumed by its three sub-pixels [17]. The power consumed by a single pixel, represented in R, G, B form, can be modeled as shown below [16]:

$$P_{pixel} = P(R) + P(G) + P(B) \tag{2.1}$$

where P(R), P(G) and P(B) are the functions to compute power consumed by red, green and blue components of a pixel. The total power consumption of an OLED display with *n* pixels can be modeled as follows [16]:

$$P = C + \sum_{i=1}^{n} \{P(R_i) + P(G_i) + P(B_i)\} \tag{2.2}$$

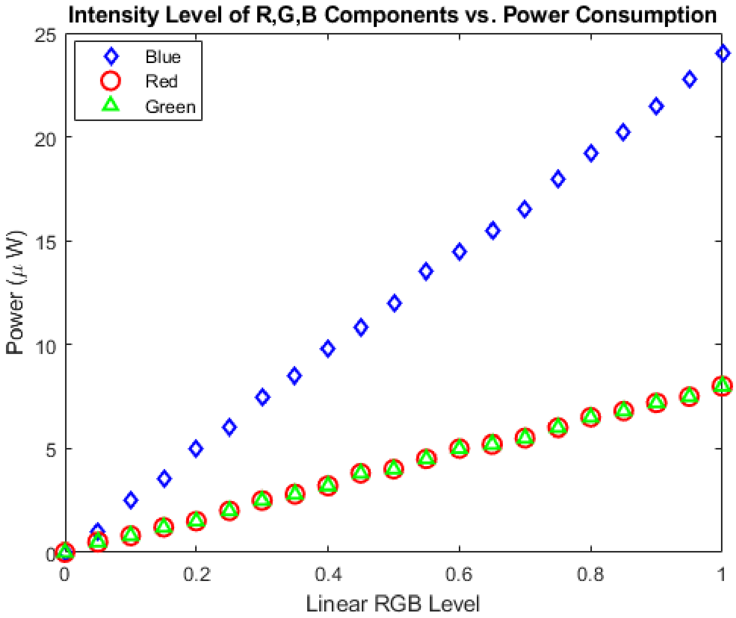


Fig. 2.1 Intensity level of RGB pixel components of an OLED device vs. power consumption [16]

where $C$ is a constant, representing the constant power consumption in OLED devices regardless of the displayed contents. Fig. 2.1 shows the linear relation between (non-gamma corrected) RGB values and power consumption measured on QVGA OLED display module [16]. It illustrates that the energy consumed by the OLED display is directly proportional to the intensity of red, green and blue pixel components. In other words, the higher the luminance of the pixels, the greater will be the power consumption and vice versa. From Fig. 2.1 it is important to note that the blue color is the most power-hungry of the three-pixel components i.e. it consumed almost three times the power consumed by red and green color components.

After understanding the OLED power model, we want to verify this behavior on a real OLED-based device. We have used Samsung Galaxy A50 smartphone, having AMOLED display technology for performing our experiments. The power consumption was measured for five monochromatic images of size 768 x 512 px including black (R=G=B=0), red (R=255, G=B=0), green (G=255, R=B=0), blue (B=255, R=G=0) and white (R=G=B=255) colors, when displayed on the mobile device under consideration. The experimental details about measuring power consumption can be found in Section 2.6. Here the main focus is on comparing the power consumption values.

Fig. 2.2 shows the actual power consumption values for the five monochromatic images. The power consumption value is represented in terms of average current in mA (milliAmperes). As can be seen, the black image consumed minimum power (86.52mA) where as the white image consumed the maximum power (437.81mA). It was discussed earlier that an OLED device consumes nearly zero energy when displaying a black image but as can be seen from Fig. 2.2 the black image also consumed some battery capacity, although much less than the other images. Since all RGB pixels are turned off in the black image, the power consumption while displaying the black image is the constant $C$ which was shown in Equation (2.2). It is the constant power required by an OLED device to display an image and it does not depend on the image contents. On the other hand, the power consumed while displaying the white image is the greatest of all the images because all the pixels are turned on for displaying the white color.

Another interesting fact can be seen from Fig. 2.2, that the power consumed by the blue image (311.71mA) is more than that of red (221.76mA) and green (219.19mA) images. This fact was also observed in Fig. 2.1. However, the power consumed by the blue color was almost 3 times of the red and green colors in Fig. 2.1 and it is almost 1.5 times in Fig. 2.2. This difference is due to the fact that different displays have different power consumption characteristics and that the relative power consumed by the red and green colors varies significantly from device to device [15].

Since the power consumption in OLED displays highly depends on the contents, reducing the RGB intensity levels, especially blue color pixels, in images would result in power savings. Hence, the main goal of the present work is to decrease the intensities of the red, green and blue color components without adversely affecting the visual quality of an image. We found that the Land-Effect [18] is the most effective way to achieve this target because it does not take into account blue separation records while forming the full-color images. Hence the proposed idea is inspired by the Land-Effect due to its effectiveness in reducing the intensity of the blue color component.

## 2.3 Related Work

The invention of OLED displays has attracted numerous researchers to work in this domain and contribute towards some major findings. Organic Light-Emitting

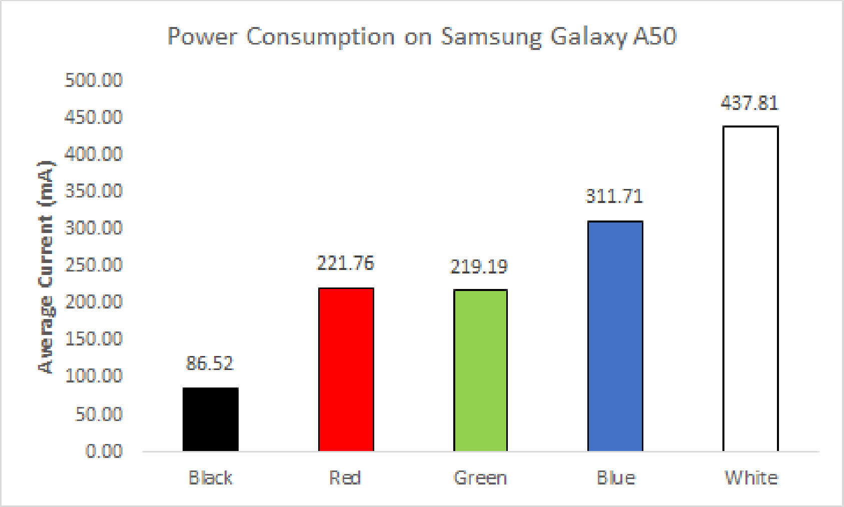


Fig. 2.2 Black, red, green, blue and white images of size 768x512 and their corresponding consumption level in mA on Samsung Galaxy A50 AMOLED-display mobile device.

Diode (OLED) [9, 19] is a modern display technology that offers enhanced image quality, wider horizontal and vertical view angles, and brighter colors, as compared to conventional LCDs. Unlike an LCD, which relies on an external light source, the backlight, the pixels of an OLED display are self-emissive and emit light individually corresponding to the electric current provided to a given pixel [20]. The image-dependent power consumption model is the core of power optimization techniques for OLED-based displays. Several power saving techniques have been proposed in the literature for OLED displays, such as partial display dimming, color re-mapping, dynamic voltage scaling, image pixel scaling and contrast enhancement, are just to name a few. These techniques are discussed in the following paragraphs.

Energy reduction for OLED display based mobile devices was pioneered by HP Labs [2, 8, 21]. According to these studies, a typical user uses only 60% of the available screen for most of the applications [8]. Therefore, the main idea was to develop energy-aware user interfaces by automatically changing the luminescence and colors of non-active areas of the screen to reduce the power consumption. A *partial display dimming* technique was patented by J.Betts-LaCroix [22], in which some selective parts of the screen, not important to the user, were darkened to reduce power consumption. Tan et al. [23] exploited this partial dimming technique to dim the display's top and bottom portions because it was a common observation that a user generally focuses on half of the screen for most applications. However, these techniques were mostly targeted to Graphical User Interfaces (GUIs) and web-pages, and therefore cannot be applied to images or videos because they tend to change the users' visual experiences.

Another GUI transformation technique called *color remapping* was proposed by Dong et al. [16], in which the original colors of the display were interchanged with the colors that consume less power. In their extended work [15, 20], the authors proposed a system for transforming web pages and GUIs, in which they focused on preserving the *usability* rather than *fidelity*. The system drastically altered the colors while preserving the distinguishability of the text from the background. Such a technique is suitable for GUIs but it may not be applied to images or videos where image fidelity is preferred over usability. Li et al. [24] proposed an energy optimization technique for web-pages which was aimed at automatically rewriting web applications using different color schemes which consume less energy. This work was extended by Wan et al. [25] to detect user interfaces of android apps whose energy consumption was greater than optimal (called energy-hotspots). Vasquez et al. [26] proposed a similar technique to optimize energy consumption of GUIs particularly in Android apps. All these techniques were quite effective to serve the purpose of reducing power consumption in GUIs, but these cannot be beneficial for graphical contents.

In later years, the attention was shifted from GUIs and webpages towards still images and videos to retain the users' visual experience. OLED *Dynamic Voltage Scaling (DVS)* technique was introduced by Shin et al. [27] which retained the original visual quality of images by reducing the supply voltage of each pixel's circuit. The maximum brightness produced by the emissive devices was limited by DVS, hence image compensation was performed on each pixel to retain image quality. Another hardware-assisted technique, similar to [27] but targeted for video streaming, was proposed by Chen et al. [28]. It was based on dividing an OLED display into multiple rectangular regions and optimizing the voltage of each region to maintain the required quality. Both of these techniques required modifications in the analog hardware, thus could not be used by off-the-shelf displays that do not support DVS.

To overcome the problem of hardware modifications, image-level optimizations were proposed in the literature to retain the perceptual quality of images. The objective of these techniques was to minimize the power required to display an image without adversely affecting the user's visual experience. This required transforming an input image to an output image by using some transformation functions such that the modified image retains the perceptual quality and requires less power when displayed on an OLED device. One such technique is *Image pixel scaling*, introduced by Lin et al. [17], aimed at scaling down the pixel values in differently-shaped regions in an image. It was studied that different regions in an image require varying degrees of attention, hence image distortion can be accepted in the less-interesting regions. The major concern in this work was to find the less important regions in an image.

An alternative way of generating energy efficient images was proposed by Hadizadeh et el. [29]. In their study, an image was first transformed from RGB to CIELAB color space and then the image colors were exchanged with alternative colors that consume less power within a sphere described by them. This study was further extended [30], in which the author used a Just-Noticeable-Difference (JND) threshold to reduce the luminance of the pixels in an image, thereby reducing the overall energy consumption. These techniques resulted in reducing power consumption along with maintaining the visual quality of images. The major task in these techniques was to define the JND threshold to replace the image colors with alternative colors.

Mobile users often lack photographic expertise and lighting conditions are not always ideal at the time of image acquisition [31], therefore attention was shifted to enhance image quality along with power-saving mechanism in OLEDs. Contrast enhancement is a process that brings out the image features more clearly by optimally using the colors available on the display device. The Power Constrained Contrast Enhancement (PCCE) algorithm was first proposed by Lee et al. [32]. The algorithm was based on Histogram Equalization (HE), which is a widely used technique to enhance low-contrast images. The authors developed a power model for OLED-based displays and formulated an objective function comprising a contrast term and a power term. They tried to simultaneously achieve contrast enhancement and power savings by minimizing the objective function based on convex optimization theory. Although the PCCE algorithm was successful in enhancing the contrast of images, the power ratio was dependent on the power control parameter, which must be regulated for different kinds of images. A Histogram Shrinking (HS) with contrast enhancement algorithm was proposed by Peng et al. [33] to automatically reduce the power consumption of OLED panels with entropy preservation. The authors aimed at reducing the empty bins of the image histogram and maintaining a constant value of image entropy. This technique reduced the power consumption before enhancing the image quality. Since the image contrast was not improved simultaneously, the system required Histogram-equalization based algorithm that balanced power consumption and enhanced color contrast automatically.

Among other contrast enhancement techniques reported in [34], Retinex is a well-known non-linear image enhancement method. The Retinex theory was proposed by Land and McCann [18], and it was adopted by Jobson et al. to Single Scale Retinex (SSR) [35] and Multi-Scale Retinex (MSR) [36]. The MSR is a generalized version of SSR. The more advanced version of Retinex, called Multi-Scale Retinex with Color Restoration (MSRCR), is a versatile automatic image enhancement algorithm that simultaneously provides dynamic range compression, color constancy, and color rendition [34]. A variant of PCCE based on Multi-Scale Retinex (MSR) was proposed by Nam et al. [37]. The authors were successful in achieving the target power consumption by enhancing images with different scales and generating the output image by using the weighted sum of the enhanced images. However, the sub-band decomposed Multi-Scale Retinex algorithm has high computational complexity due to extensive iterations to achieve the required power consumption. Jang et al. [38] demonstrated that similar results can be obtained by a non-iterative optimization procedure. These techniques were effective but some of them suffer from a major drawback of enhanced complexity. For example, a non-linear optimization problem had to be solved in [32] to identify the optimal transformation for a given image, which was a very complex computational task. Furthermore, an iterative approach was used in [37], which had high complexity. Although this problem was avoided in [38], it still required several loops involving divisions and fractional powers. Moreover, the implementation of these techniques requires a high-end CPU, which itself might consume a large amount of energy.

Apart from these techniques, there is yet another solution, proposed by Kang et al. [39], that tackle the problem of computation complexity. In that solution, the majority of the computational effort was moved to the *image acquisition* phase that used a custom camera application in a mobile device. The advantage of that approach is the low computational complexity as the image is transformed directly during the acquisition phase, while the obvious disadvantage is that the images need to be captured through their proposed camera application. Another solution that lowered the computational complexity and yielded enhanced images with power-saving was proposed by Pagliari et al. [40]. The authors proposed an adaptive power-saving and contrast enhancement technique called LAPSE, that simultaneously constrained the power consumption and maximized the image contrast while limiting the amount of image modifications. Although LAPSE produced significant results, it was at the cost of *hardware-friendly* transformation function and the *computationally intensive* offline training phase.

From the presented literature review we felt that there is a gap that needs to be filled in terms of image manipulation. A simple, faster, and effective technique is required to modify the images in such a way that they achieve the desired goal of power savings and image enhancement. The present study was proposed to fill this gap. Table 2.1 summarizes different state-of-the-art techniques discussed above and presents the limitations associated with each of these techniques.

## 2.4 Problem Statement

There are various hardware and software components in a portable or handheld device and each of the components contribute differently to power consumption characteristics of the device. Fig. 2.3, taken from [41], shows the power consumption breakdown in a smartphone in the idle state when the backlight is turned off. It can be seen that the Graphics module consumed the largest proportion of energy. The graphics module is responsible for displaying graphical contents, such as images and videos.

The rapid rise in smartphone technology has tremendously increased the use of social networks such as Facebook and Instagram [42]. According to the 2021 Social Media Demographics Guide [43], 42% of the world population ($\sim 3.2$ billion people), use social media, including Facebook, Instagram, Twitter, LinkedIn, YouTube, and Snapchat. People generally scroll through these applications for hours to see graphic contents - images and videos, which is the major cause of battery drainage in smartphones. The average daily time spent on these social networking sites, as reported in [43], is represented graphically in Fig. 2.4.

One general trend can be observed from this report i.e. the more the platform is rich in graphical contents, the more is the average daily time spent on that platform by a user. If a person has an account on all the above mentioned platforms, his daily usage of social networking sites would be approximately equal to 2.5 hours

Table 2.1 Limitations of the state-of-the-art techniques

| State-of-the-art Techniques | Limitations |
|---|---|
| Partial display dimming technique [22, 23] | Mostly targeted to GUIs and web-pages, and therefore cannot be applied to images or videos because they tend to change the users' visual experiences. |
| Color remapping technique [16, 15, 20, 24–26] | Designed to interchange the original colors of the display with the colors that consume less power, therefore it is suitable for GUIs and webpages but not for graphical contents. |
| Dynamic Voltage Scaling (DVS) technique [27, 28] | Require modifications in the analog hardware, thus could not be used by off-the-shelf displays that do not support DVS. |
| Image Pixel Scaling [17, 29, 30] | Aimed at scaling down the pixel values in differently-shaped regions in an image. The major concern is to find the less important regions in an image where distortions might be accepted. |
| Power Constrained Contrast Enhancement (PCCE) technique [32] | Successful in enhancing the contrast of images but the power ratio was dependent on the power control parameter, which must be regulated for different kinds of images. Hence this technique cannot be generalized for all kinds of images. |
| PCCE based on Multi-Scale Retinex (MSR) technique [37] | The algorithm has high computational complexity due to extensive iterations to achieve the required power consumption. |
| Power optimization at image acquisition phase [39] | Reduced the computational complexity as the image is transformed directly during the acquisition phase, while the obvious disadvantage is that the images need to be captured through their proposed camera application. |
| Adaptive power-saving and contrast enhancement technique (LAPSE) [40] | This technique requires hardware-friendly transformation function and the computationally intensive offline training phase. |

on average. Moreover, majority of users access these platforms via mobile devices. Another study [44], performed on young college students reported that, a typical teenager spends on average about 506 minutes ($\sim$ 8.4 hours) per day on Facebook, Twitter, Instagram and Snapchat, scrolling through millions of images and videos.

Such enormous usage of these applications is due to the presence of extensive and exciting graphical content that has the potential to engage users for endless hours. On one hand, images and videos are extremely essential part of visual communication, on the other hand, they are the main reason for power drainage in energy-aware devices.

Hence, the rich graphical content has contributed towards the extensive use of social media and the extended hours of social media usage has given rise to enormous amount of battery drainage in smartphones.

The research studies described in the previous section aim to reduce the power consumption using several different techniques, but some of them are only suitable for GUIs and webpages, some of them drastically changes the colors of display contents, some require modifications in the hardware, some cannot be generalized to all kinds of images, while others have very high computational complexities. Our proposed method is designed to overcome these limitations and drawbacks due to its simple yet powerful approach as will be shown in the rest of the chapter.

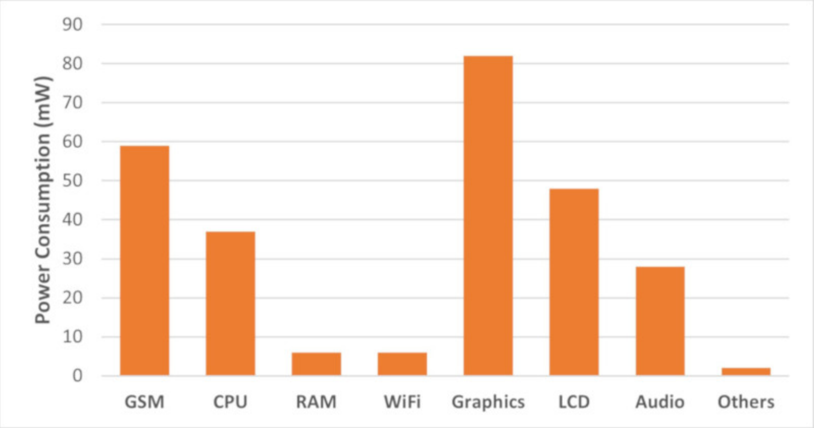


Fig. 2.3 Breakdown of power consumption in a smartphone in the idle state with backlight off [41]

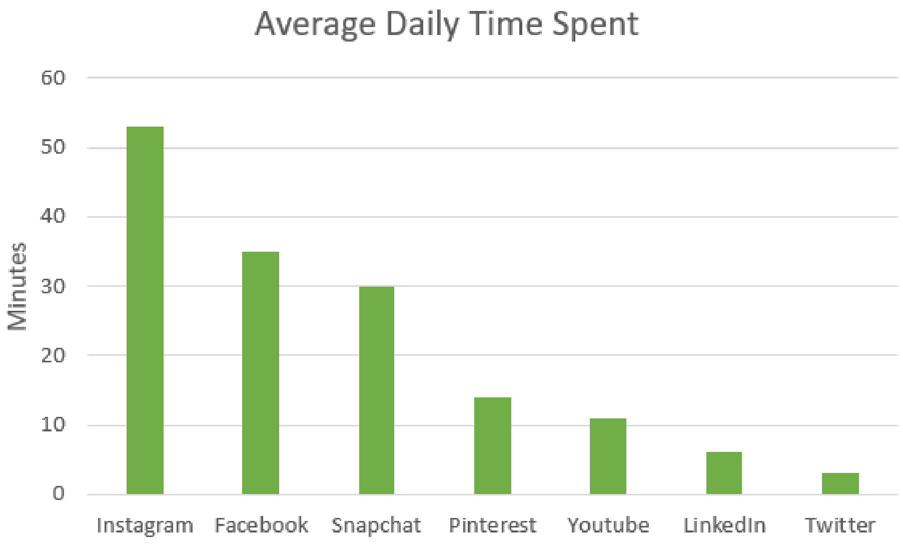


Fig. 2.4 Average daily time spent on various social media platforms

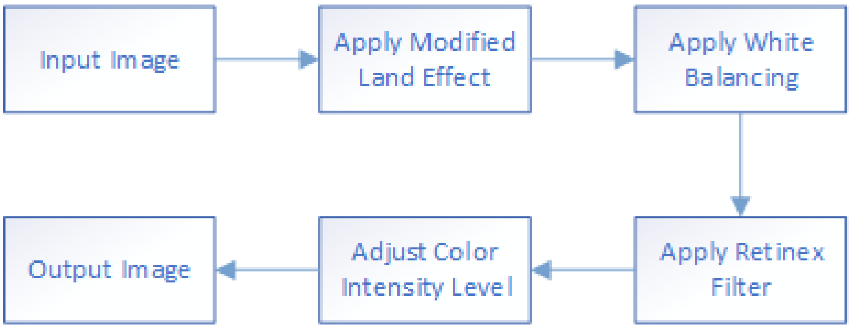


Fig. 2.5 Block diagram of the proposed system

## 2.5 The Proposed Method

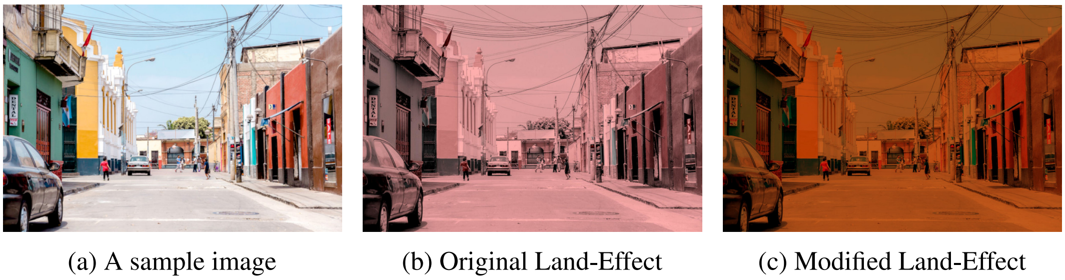
The block diagram of the proposed system is shown in Fig. 2.5. An input image is provided to the system, which then passes through four operations to obtain the final output image. The four operations namely, modified Land-Effect, white-balance, retinex filter, and adjustment of color intensity levels, are described in the following subsections.

### 2.5.1 The Land Effect

The idea proposed in this paper is inspired by the “Land-Effect”, named after its inventor Edward H. Land [45]. While performing some experiments in Color Vision, Land found that it was possible to obtain a full-color image with only two color components instead of three. In the experiments performed by Land [45], one image was photographed through a red filter, referred to as “red separation record” and the other through a green filter, known as “green separation record”. The red separation record was projected through a monochromatic red filter, and the green separation record was projected through white light without any filter. These two images were then superimposed, with the expectation that the superimposed image would depict red, white and various shades of pink, since green separation record was projected through white light. But to their utter surprise, one could see a picture remarkably similar to the original full-color image [45, 46]. The superimposed image contained nearly all the colors depicted in the original image instead of only showing red, white and different shades of pink, but of-course at the cost of lower image quality. The obtained image was then passed through the Retinex filter to enhance visual perception. This phenomenon was named the “Land-Effect”.

Fig. 2.6 shows the result of applying the Land-Effect, in its original form, on a sample image taken from the experimental test set. It provides the significance of the need for the modified Land-Effect. The original sample image is shown in Fig. 2.6a and the modified image after applying the Land-Effect is shown in Fig 2.6b. As can be seen from Fig. 2.6b, the colors in the image are distinguishable to some extent. Particularly, one is able to recognize the shades of blue and green colors even though they are not actually present in the image. But the modified image does not represent all the colors that are found in the original image. For example, the yellow color is not visible in Fig. 2.6b. Moreover, the visual quality of the original image has also been deteriorated. We were not truly satisfied by the results of the Land-Effect and decided to modify it slightly by introducing the green filter for the “green separation record”.

Fig. 2.7 shows the steps involved in our proposed modified version of Land-Effect. As illustrated in Fig. 2.7, the first step was to decompose the original RGB image into it’s Red, Green and Blue color components or separation records. This is achieved by passing an image through red, green and blue filters that allow only a specific wavelength to pass and block all other colors. The next step was to pass the



(a) A sample image (b) Original Land-Effect (c) Modified Land-Effect

Fig. 2.6 Comparison of the actual and the proposed version of Land-Effect. (a) shows a sample image from the experimental test set, (b) is the resulting image after applying the actual Land-Effect and (c) is the resulting image after applying the proposed modified Land-Effect.

red separation record through the monochromatic red filter and the green separation record through the monochromatic green filter. Note that the modified version of Land-Effect involves the addition of this green filter. The red and green filtered images are then superimposed on each other.

From the physics of color, it is a known fact that the wavelength of green color is less than that of the red color and its frequency and energy are higher than the red color's frequency and energy [47]. Due to this reason, when the red and green filtered images are superimposed, the green color becomes more prominent than the red, giving the entire image a greenish look. In order to compensate for the prominence of green color, we reduced the opacity of the green filter image to 24.5% and the opacity of the red filtered image to 84.5%. These percentage values were chosen by the "trial and error" method. We experimented with several combinations of these values and found the selected values to be optimal for most of the images in the experimental test set. Finally, after setting the opacity levels, the red and green filtered images were superimposed and the resulting image was a true-color image, as can be seen from Fig. 2.6c. It is evident from the comparison of Fig. 2.6b and Fig. 2.6c that the modified version of Land-Effect yielded visually better results than the actual Land-Effect. Although the image was darker in Fig. 2.6c, but it successfully displayed almost all the colors that were present in the original image as opposed to Fig. 2.6b, i.e., one is able to perceive the presence of different shades of yellow, red, green and blue colors from the modified land-effect image.

It is important to note that the original as well as the modified Land-effect does not take into account the blue color component. It only combines red and green filtered images by passing them through monochromatic red and green filters, respectively.

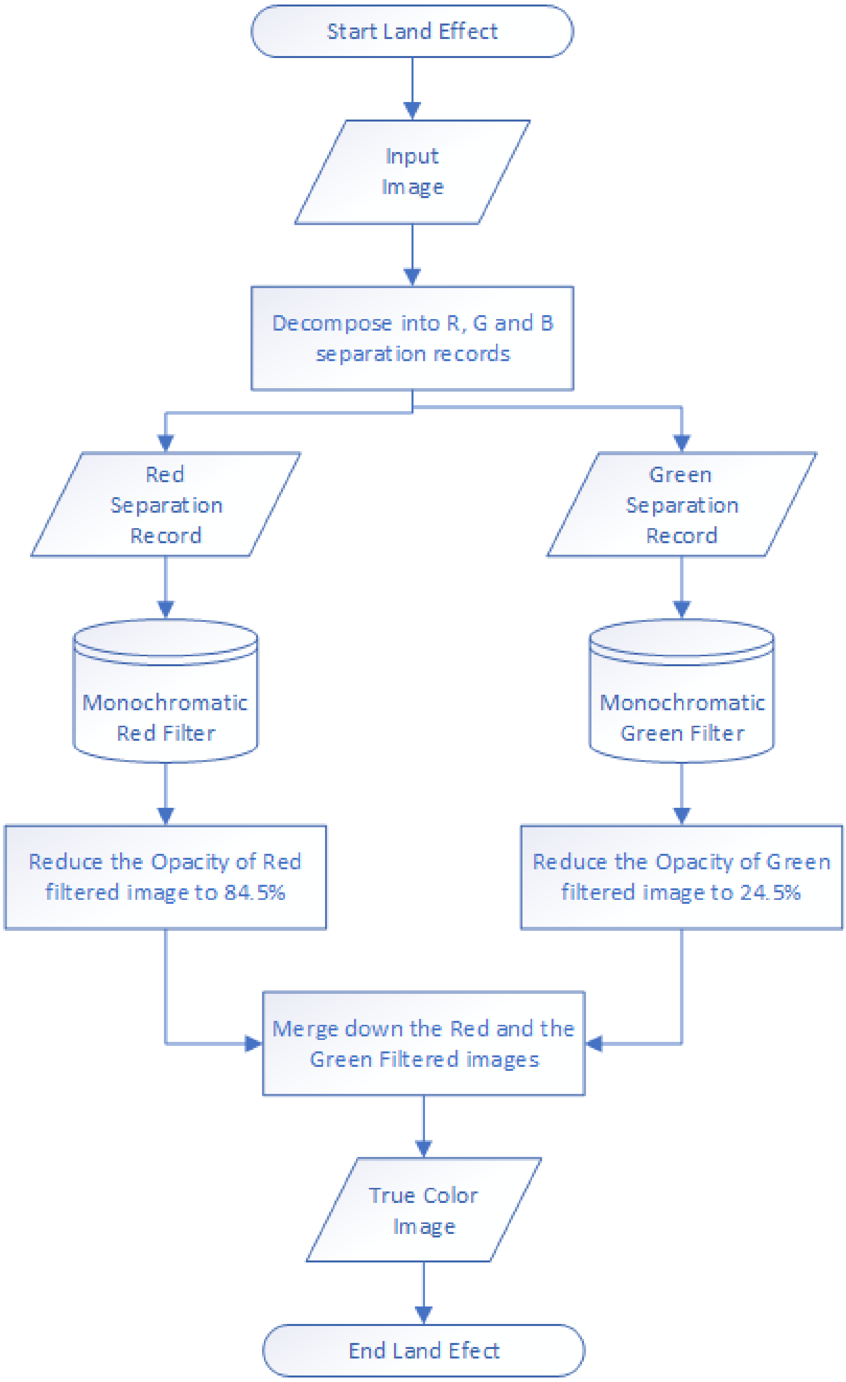

Start Land Effect
Input Image
Decompose into R, G and B separation records
Red Separation Record
Green Separation Record
Monochromatic Red Filter
Monochromatic Green Filter
Reduce the Opacity of Red filtered image to 84.5%
Reduce the Opacity of Green filtered image to 24.5%
Merge down the Red and the Green Filtered images
True Color Image
End Land Efect

Fig. 2.7 Steps for producing the modified version of Land-Effect

From the earlier discussion we know that the blue is the most power consuming color component and by applying the Land-Effect we got rid of the most energy-draining component. We also know that the power consumption in OLEDs depends on the intensity of RGB colors. Therefore, it can be stated that the application of Land-Effect would greatly contribute towards reducing the power consumption in OLED-based displays. But at the same time, some quality enhancement mechanism is required to improve the visual quality which has been deteriorated by applying the Land-Effect. In the proposed work, white balancing and the retinex filter were used as image-enhancement techniques.

### 2.5.2 White Balance Operation

Now let's understand the RGB histograms of the sample image under consideration depicted in Fig. 2.8. The RGB histograms of the original image are shown in Figs. 2.8b, 2.8c and 2.8d, respectively. As can be seen, these histograms are concentrated towards the higher intensity values at the right side of the plot. After applying the proposed Land-effect, the RGB histograms were also modified as shown in Figs. 2.8f, 2.8g and 2.8h. Two interesting facts can be observed from these histograms: a) the proposed Land-effect operation contracted the histogram values towards the lower intensity region at the left side of the plot, b) it significantly reduced the intensity of blue color and therefore the blue histogram is almost negligible as compared to the red and green histograms. Both the facts confirm the effectiveness of the Land-effect in reducing the RGB intensity values and hence the power consumption in OLED-based devices.

The reduced RGB intensity values, obtained after applying the proposed Land-effect, resulted in decreasing the overall visual quality of the image as can be seen in Fig. 2.8e. For enhancing the image quality, it is therefore necessary to balance the RGB intensity values by stretching the histogram plots. To do so, the white-balance operation was performed as its main purpose is to adjust the colors in an image by stretching the red, green and blue channel histograms separately.

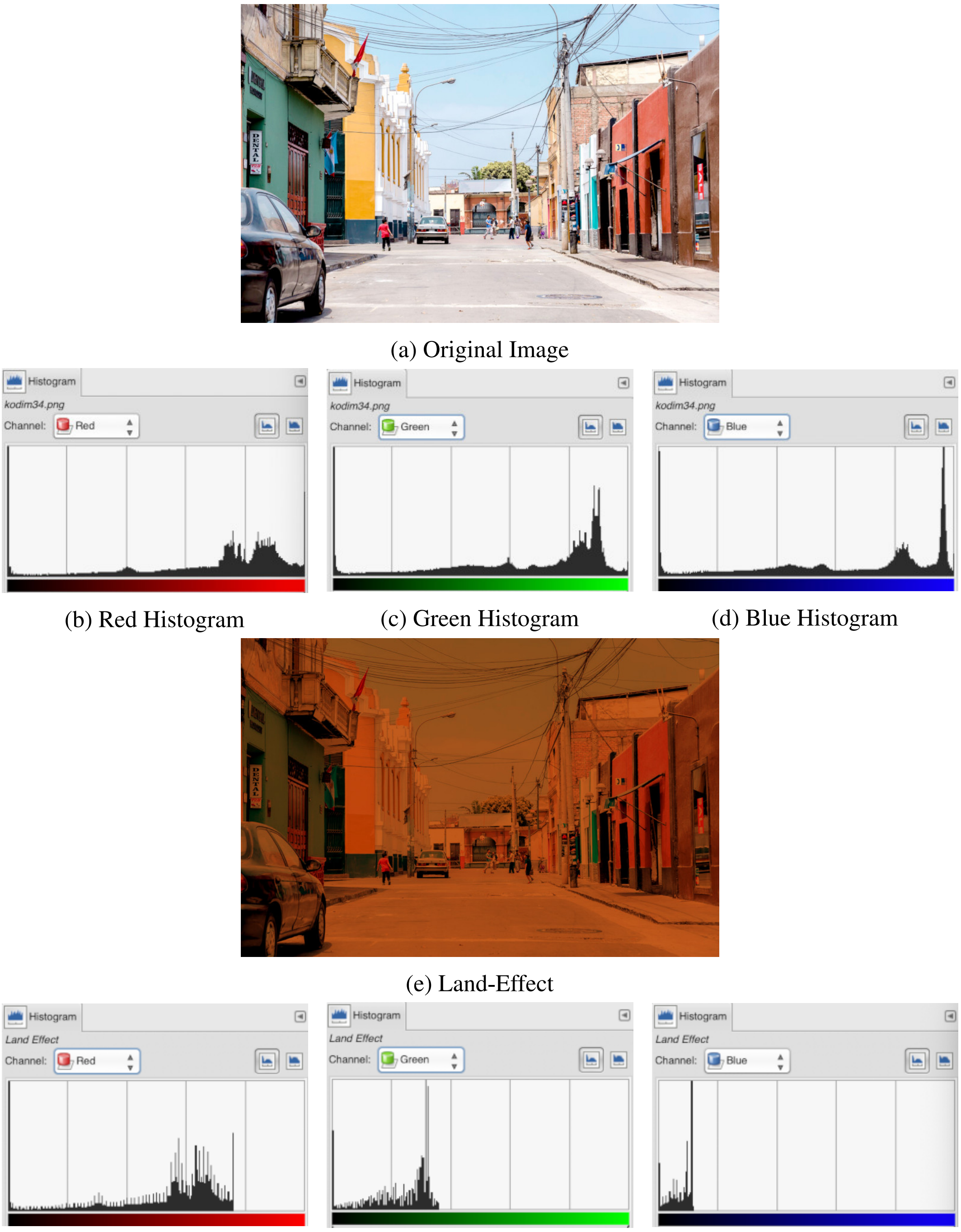
Lighting conditions under which an object is viewed affect the colors of the object in an image. As can be seen in Fig. 2.8e, the image quality is not good and due to some unwanted color casts, the white color objects do not appear white. White Balance is the process of removing those unrealistic color casts from images so that white objects appear white in an image instead of any other color. The white balance mechanism adjusts the colors in an image by stretching the red, green and blue intensity values separately. To do so, it discards the pixel colors at each end of the red, green and blue histograms that are used by only 0.05% of the pixels in an image [48]. It then stretches the remaining range of pixels as much as possible. White balance operates on RGB layers in images. Figs. 2.8j, 2.8k and 2.8l show the RGB histograms after applying the white-balance operation. As can be seen, the histograms have been stretched over the entire range of the intensity values which results in an enhanced visual quality of the image as can be seen in Fig. 2.8i. The histogram stretching creates gaps between the pixel columns, giving it a striped look [48].

### 2.5.3 Retinex Filter

The white-balance operation, improved the visual quality of image but created gaps in the RGB histograms. In other words, these gaps indicate the discarded or missing information for some RGB intensity values. In order to reconstruct the RGB intensity values in a more balanced way, the Retinex Filter was used. The RGB histograms after applying the Retinex filter are depicted in Figs. 2.8n, 2.8o and 2.8p. As can be seen, the Retinex filter operation further enhanced the visual quality of the image by removing the gaps. Now, there for each intensity there is some value for the red, green and blue color, which means there is no missing information about the RGB intensity values.

It was realized by Land that even when there were no green or blue wavelengths present in the superimposed image, the human visual system is so powerful to perceive them as green or blue by discounting the red illumination. It is due to the fact that both the human eye (through retina) and the brain (through cortex) are involved in the processing of color identification. This fact was later formulated as “Retinex Theory” [46], where the term “Retinex” is the contraction of the terms “Retina” and “Cortex”. The retina is the part of the eye that detects color and the cortex is the part of the brain that processes the visual information it receives from the retina. Soon after Land’s theory of color vision, the Retinex algorithm became the most popular image enhancement algorithm based on illumination compensation. The Retinex algorithm modifies the color contrasts in an image in such a way that the image appears very similar to the one perceived by the human visual system.

The human eye can see colors even when the light is low, but this is not true for a camera. The goal of Retinex processing is to provide a model for human



(a) Original Image

kodim34.png
Channel: Red

(b) Red Histogram

kodim34.png
Channel: Green

(c) Green Histogram

kodim34.png
Channel: Blue

(d) Blue Histogram

(e) Land-Effect

Land Effect
Channel: Red

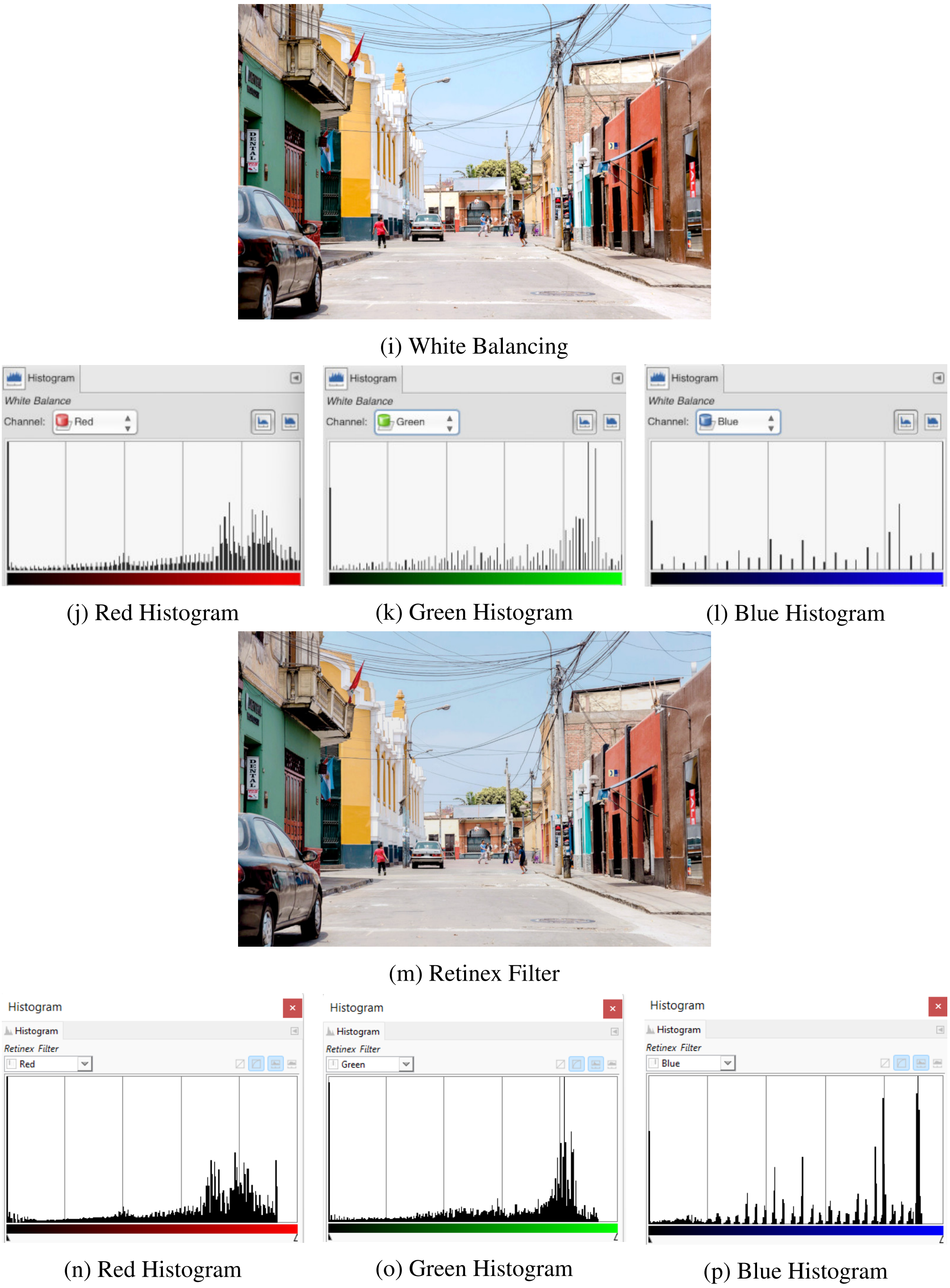
(f) Red Histogram

Land Effect
Channel: Green

(g) Green Histogram

Land Effect
Channel: Blue

(h) Blue Histogram



(i) White Balancing

(j) Red Histogram (k) Green Histogram (l) Blue Histogram

(m) Retinex Filter

(n) Red Histogram (o) Green Histogram (p) Blue Histogram

Fig. 2.8 A sample image in its original, land-effected and white balanced form along with RGB histograms before and after applying the white balancing. (a) the original sample image, (b-d) RGB histograms of (a); (e) Result of applying proposed Land-Effect, (f-h) RGB histograms of (e); (i) Result of applying white balance on (e), (j-l) RGB histograms of (i).

vision color constancy. It is used in wide dynamic scenes to bring out details in the under-exposed regions in an image. In our proposed work, the Retinex filter is used for compensating the illumination introduced by the white-balance operation. The applied Retinex filter is based on the Multi Scale Retinex with Color Restoration (MSRCR) algorithm, which is an automatic image enhancement algorithm that simultaneously provides dynamic range compression, color constancy, and color rendition [34]. When the Retinex filter was applied, the color constancy and contrast were enhanced and the images were rendered similarly to how human vision is believed to perceive them.

### 2.5.4 Adjusting Color Intensity

After carrying out the three operations described above, the visual quality of images was enhanced significantly. The next step was to adjust the color intensity level to reduce the luminance which would further lower down the power consumption. The purpose of this step was to reduce the power consumption to the best possible level without compromising the visual quality of images. After performing several experiments on different images, we found that by reducing the intensity value of RGB pixels from 1.00 to 0.72, the luminance was reduced and the color contrast was enhanced. This step also contributed to reducing power consumption in OLED-based displays.

It was noted in some of the related works that the images were converted from RGB to some other color spaces for processing. For example, in the work proposed by Lee et al. [32] and Nam et al. [37], the RGB images were converted into YUV format. Similarly in the work proposed by Jang et al. [38] and Pagliari et al. [40], the RGB images were converted to YCbCr color space. But this is not the case with our proposed work as all the processing requires images in the RGB format. All the operations in the proposed work, including Land-Effect, white balancing, retinex filter, and intensity adjustment, are directly modifying the RGB pixel values in some way.

## 2.6 Experimental Results and Discussion

To evaluate the performance of the proposed system, extensive experiments were conducted on 48 test images which contain a wide variety of subjects, brightness, and contrast. 24 of these images were taken from the Kodak lossless true-color image database [49], which is an extensively used data-set in the research field of image processing [29, 31, 38, 39]. In the Kodak database each image is either 768 x 512 px or 512 x 768 px in size. Sixteen images were taken from two different image databases, Pixabay (Beautiful Free Images) [50] and Unsplash (Photos for everyone) [51], and the remaining 8 images were the screenshots of the GUIs captured from Microsoft, Android and iOS-based smartphones. All of these images are of varying sizes ranging from the smallest size (282 x 501 px) to the largest size (4096 x 2304 px). The images in the test set were not resized to examine the applicability of the proposed system on several differently sized images.

There are two main goals of the proposed work. The first one is to produce energy-efficient images (which consume less power) by reducing the RGB intensity values and the second goal is to enhance the color contrast of images to improve the visual quality. The following subsections discuss the experimental results in detail and provide many different performance evaluation metrics with respect to these goals. The experimental results are also compared with some of the related works and will be described in the following subsections.

### 2.6.1 RGB Intensity Values and Power Consumption

As mentioned earlier, the power consumption in OLED-based displays is directly proportional to the RGB intensity values of the display contents, which means the higher the intensity levels of RGB pixels, the greater will be the power consumption and vice versa. One of the main goals of this research work was to reduce the intensity levels of the pixels in RGB colored images in order to reduce power consumption. In fact, this was the primary reason for applying Land-Effect as it did not take into account the most power consuming color component i.e. Blue color.

The RGB intensity values were computed for all the test images before and after applying the proposed system. The red, green and blue intensity values of the original and the modified images were computed as the sum of red, green and blue pixels, respectively. The MATLAB code for computing the intensity values of red, green and blue pixels is provided in Appendix B.1.

Figs. 2.9, 2.10 and 2.11 show the mean intensity values of the red, green and blue pixels, respectively, in the original and the modified images. The proposed system was able to successfully reduce the intensity values for the red and green pixels for all the test images, as can be seen in Figs. 2.9 and 2.10 respectively. In other words, the red and green intensity values of the modified images are less than that of the original images. This is also true for the blue intensity levels except for 8 out of 48 images where the intensity level of the modified image is slightly greater than the original image as can be seen in Fig. 2.11. The explanation for this behavior is as follows; as already discussed, the Land-Effect does not consider the blue separation records and it greatly deteriorates the image quality, therefore white balancing operation was performed. The white balancing image operation stretches the red, green and blue histograms to enhance the image quality. However, due to extremely lower levels of blue intensities (resulting from the applied Land-Effect), the white balance operation needed to forcefully stretch the blue histograms, resulting in over-stretching of intensity levels for few images. But this fact does not adversely affect the overall RGB intensity because the red and green intensity levels are showing the decreasing trends. Moreover, on average, the red, green and blue intensities are reduced by 22.52%, 21.13%, and 15.90%, respectively, for the modified images which can be regarded as a great achievement of the proposed system.

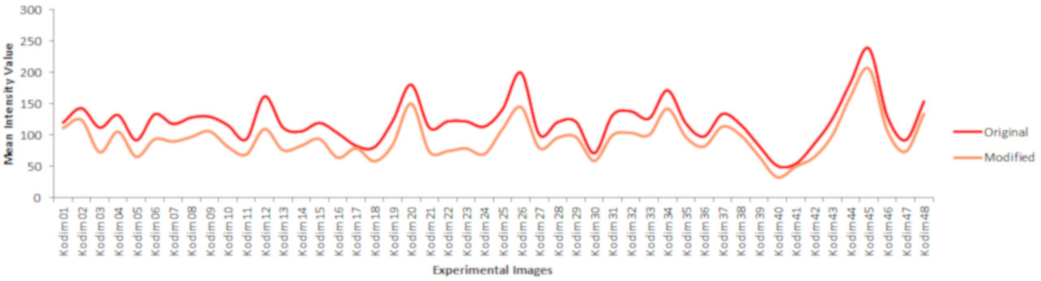


Fig. 2.9 Experimental images vs. mean intensity values of red pixels in original and modified images.

The total power consumed in an energy-adaptive display is proportional to the number of “on” pixels and the brightness of their red (R), green (G) and blue (B) components [4]. Chuang et al. [4] proposed that the sum of linear (non-gamma-

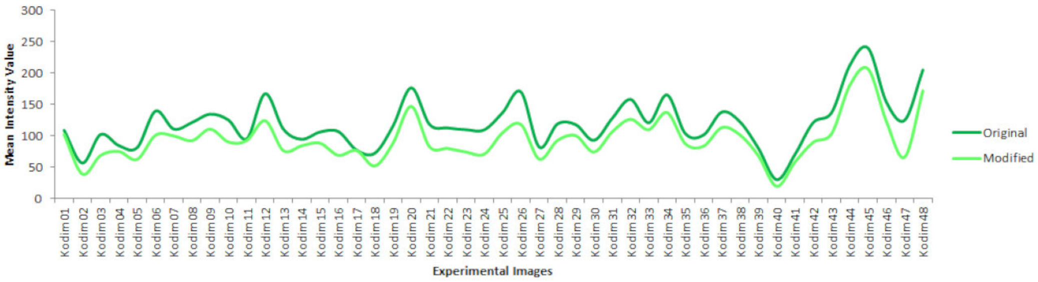


Fig. 2.10 Experimental images vs. mean intensity values of green pixels in original and modified images.

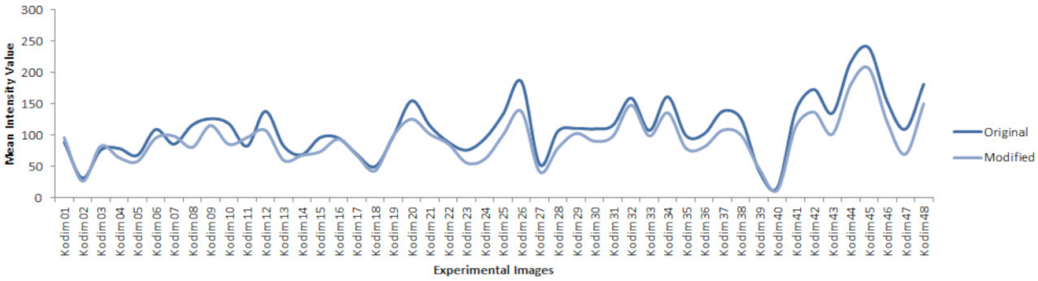


Fig. 2.11 Experimental images vs. mean intensity values of blue pixels in original and modified images.

corrected) RGB pixel components can be used as a simple measure of the power consumption in an OLED display. The power consumption can be estimated by computing the RGB intensity levels of image pixels. Hence, if C = (R, G, B) is the color of a particular pixel, the corresponding RGB intensity of that pixel or the power consumed by the pixel is [29]:

$$P(C) = R + G + B \tag{2.3}$$

If there are n number of pixels in an image, then the total RGB intensity or the power consumption of all the pixels in an image is given as:

$$P = \sum_{i=1}^{n} \{P(R_i) + P(G_i) + P(B_i)\} \tag{2.4}$$

Before measuring the actual power consumption on the OLED-based smartphone, let us first observe the values of power consumption based on the RGB intensity levels. The RGB intensity reduction percentage of the modified images is computed

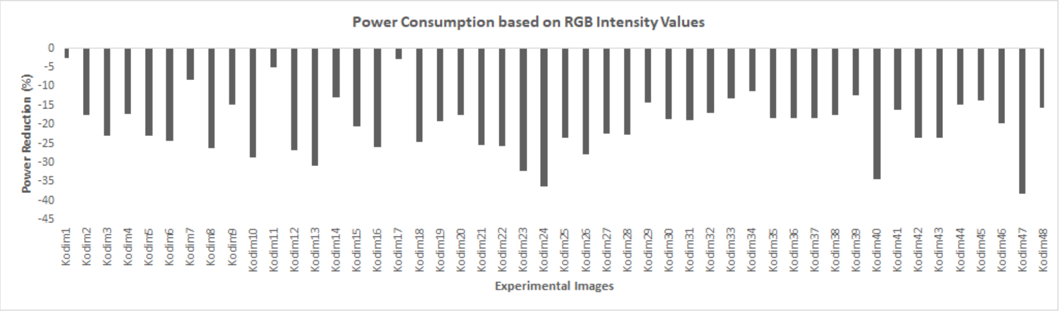


Fig. 2.12 RGB intensity reduction percentages of the modified images in the experimental test set.

as follows:

$$\Delta P = (P_2 - P_1)/P_1 * 100 \tag{2.5}$$

where P1 and P2 is the total RGB intensity of the original and the modified image respectively. Fig. 2.12 shows the RGB intensity reduction percentages of the modified images in the experimental test set. As can be seen, the proposed system successfully reduced the RGB intensity levels (first goal achieved!) of all the images of the experimental test set with values ranging from 2.5% to 38.2%, with an average being 20.09%. This also implies that our experimental test set covers a variety of images having diverse intensity range of RGB values and allows a comprehensive study of the proposed method.

One possible limitation of the proposed method could be the white balance operation, which stretches the RGB histogram values. It could happen that for some very rare cases that the white balancing results in over-stretching for all the red, green and blue color histograms, resulting in an over all increase in power consumption. This scenario has not occurred in our experimental test set but it may occur with some rare images.

### 2.6.2 Measure of Enhancement (EME)

From the results presented above we know that the proposed system was successful in reducing the RGB intensity values and therefore the power consumption in the experimental images. Our next point of concern is to see whether the proposed system enhances the color contrast in images.

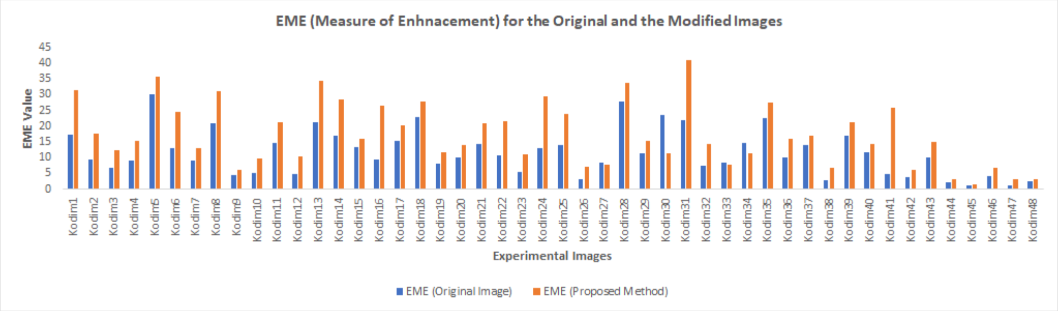


Fig. 2.13 EME Values for the original and modified images in the experimental test set.

Agaian et al. [52] proposed a quantitative metric, EME (Measure of Enhancement) to measure the visual quality or contrast of images. EME is excessively used metric in research to evaluate the visual quality of images [31, 33, 37, 39, 53]. The higher the value of EME, the greater will be the color contrast and vice versa. Let an image I be divided into $k_1, k_2$ non-overlapping sub-blocks of size $w_1 * w_2$. EME is computed as follows:

$$EME = \frac{1}{k_1 k_2} \sum_{i=1}^{k_1} \sum_{j=1}^{k_2} 20 \ln \frac{max(I_{i,j})}{min(I_{i,j}) + \varepsilon} \tag{2.6}$$

where $max(I_{i,j})$ and $min(I_{i,j})$ are the maximum and minimum intensity values respectively in a given block, and $\varepsilon$ is a small constant equal to 0.0001 to avoid dividing by zero. The block size of $w_1 x w_2 = 8x8$ is used in our experiments. Fig. 2.13 shows the EME values of the original and modified images in the experimental test set. It is worth noting that along with reducing the power consumption, the proposed algorithm performs significantly well in improving the color contrasts of nearly all the images.

## 2.6.3 Significance of the Proposed Method

As discussed, the proposed system comprises of four steps, which collectively result in enormous amount of power savings and enhanced color contrasts when applied on the experimental images. Here, an important question arises that how each step of the proposed method is contributing towards the overall power savings. In other words, what is an individual gain of each of the steps involved in the proposed method. In order to understand the contribution of each image operation towards reducing the power consumption, the power consumption percentages were computed individually after applying each step of the proposed methodology.

The results are presented in Fig. 2.14. As can be seen, the proposed Land-effect resulted in the maximum levels of power reduction as compared to all the other steps. From the explanation of the modified Land-Effect it was expected that this step would drastically reduce the power consumption percentages as it does not take into account the most power hungry, i.e. the blue color component. By applying the modified Land-Effect, the overall power consumption of the modified images was reduced by 58% on average. The second most significant step in reducing the overall power consumption is the last step of the proposed methodology, i.e. adjusting the RGB intensity levels of the images. This step was performed to mitigate the power consumption effects of the contrast-enhancing algorithms without adversely affecting the user visual experience. The overall power consumption was reduced by 20.09% on average for the modified images after decreasing the RGB intensity values.

The other two steps of the proposed method, namely, the white-balance operation and the retinex filter, are concerned with enhancing the images by improving color contrasts. The visual quality of the images was deteriorated after applying the modified Land-Effect. Therefore, the white-balance operation was performed to balance the RGB intensity values. The white-balance operation stretched the RGB histograms to balance the color contrasts, and created gaps in the intensity levels as demonstrated in Fig. 2.8. Then the Retinex filter was applied to enhance the color contrasts even more to represent the images close to the human perceptual system, i.e. as perceived by the human eye. The Retinex filter operation filled the gaps to the best possible levels in the RGB intensity values that were created by white balancing operation. Both of these image operations contribute very little towards reducing the power consumption, i.e., 1.9% on average by white-balancing and 0.16% on average by Retinex filter, as they are related to enhancing the visual quality of images by increasing the color contrasts.

As already mentioned, there were two main goals of this project: 1) reducing the power consumption in OLED-based devices, and 2) enhancing the color-contrasts and thereby the visual quality of images. From the above discussion, it can be deduced that the proposed Land-Effect and the adjustment of RGB intensity levels

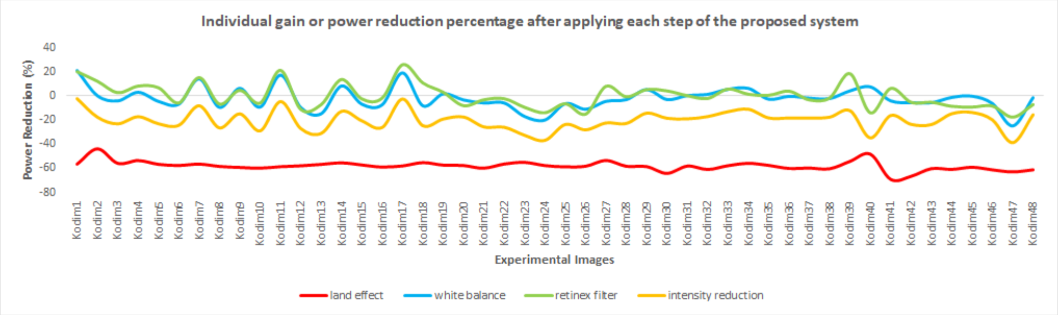
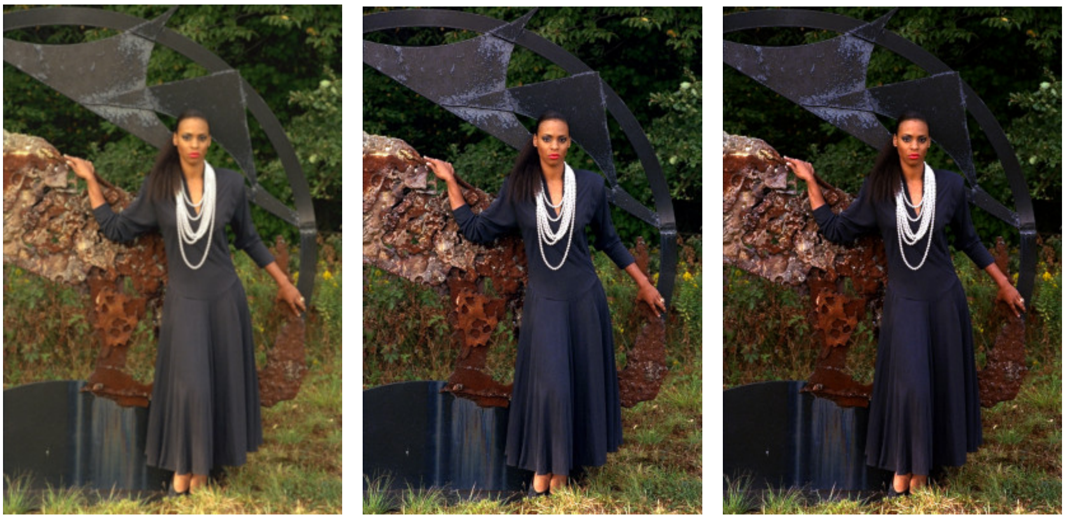


Fig. 2.14 Individual gain or power reduction percentage after applying each step of the proposed system.

are related to the first goals, whereas, the white-balancing and Retinex filter are related to the second goal of this research work.

The modified Land-Effect is the most crucial step of the proposed method, as it results in significant amounts of power savings, of-course at the cost of degrading visual experience. The application of white-balancing is of pivotal importance in the proposed system, as it greatly enhances the visual quality of an image (which was deteriorated by the modified Land-Effect). The question arises about the significance of using the retinex filter. It's role is depicted in Fig. 2.15. Given a sample image from the experimental test set, we produced two versions of the modified image, one produced by using all the steps of the proposed system and the other produced by skipping the retinex filter from the proposed system. The output image in Fig: 2.15b reduced power consumption by approximately 25% where as the image in Fig. 2.15c reduced the power consumption by approximately 40% as compared to the original image. However, the image produced without using the retinex filter is slightly darker and hides some of the details present in the image. But considering the huge power savings, the retinex filter can even be ignored where the power saving is immensely preferred over the visual quality of images. In other words, the application of the Retinex filter can even be ignored where small image details are not extremely important. Finally, the last step of the proposed system contributes towards overall power savings and it is equally an important step.



(a) Original Image (b) With Retinex filter (c) Without Retinex filter

Fig. 2.15 Comparison of a sample image modified with and without using retinex filter. (a) Original Image, (b) modified image using the proposed method, and (c) modified image without using the retinex filter

### 2.6.4 Power Consumption on Samsung Galaxy A50

Till now we have discussed the power consumption in terms of RGB intensity values as we know that the power consumption in OLED-based devices is directly proportional to the RGB intensity of the display contents. Theoretically, this assumption holds true. In this subsection we will see the actual power consumption measured on a real OLED-based device.

Samsung Galaxy A50, an OLED-based smartphone, with display resolution of 2340 x 1080 (FHD+, more than Full HD display), has been used for performing the experiments. The power consumption was measured by using a mobile application, named, AccuBattery [54], available on Google Play Store. AccuBattery protects battery health, displays battery usage information, and measures battery capacity in milli-Ampere per hour (mAh). The application is based on actual measurements of a device's performance and backed by scientific research on how to protect battery health and performance over time [55].

The power consumption was monitored for all the original and their corresponding modified images in the smartphone. In order to measure the actual power consumption, the smartphone was kept under the following conditions:

- the brightness level of the display was set to maximum
- the device was kept in the airplane mode
- all radio and network communications including Bluetooth, Wifi, and 4G were disabled
- all other background services and applications were disabled to stop the battery consumption due to unnecessary system services.

It is important to note that some system services, necessary to run the operating system, could not be disabled, and hence contributed towards the power consumption. Moreover, the Accubattery app itself would have consumed some energy since it was actively running.

Under the above-mentioned conditions, each image was displayed in the full-screen mode at the highest brightness level of the mobile phone. The app was turned on and the battery consumption values in milliAmpere per hour (mAh) were being displayed on the screen. However, due to the absence of *ideal* conditions, the value of the battery consumption was found to be fluctuating between a certain range while displaying the images. To accurately measure the power consumption to the best possible level, each image was displayed on the mobile device for 10 minutes, and the battery consumption value (mAh) was recorded after every 30 seconds. At the end of 10 minutes, there were 21 values in total i.e. the battery consumption value (mAh) measured at (t, t + 30, t + 60, t + 90, ... , t + 600) seconds or (t, t + 0.5, t + 1, t + 1.5, ... , t + 10) minutes. At the end of 10 minutes, the mean value was computed for all the recorded values which represent the power consumed by the displayed image.

The actual power consumption of the original and the modified images is illustrated in Fig. 2.16. The graph shows an overall decreasing trend in power consumption for modified images except for a few images. For 88% of the total images, i.e., 42 out of 48 images, the power consumed by the modified images was less than that of the original images. These results signify the achievement of the proposed system which successfully decreased the power consumption by 13.16% on average for all the experimental test set. An interesting information is worth-noting at this point, that the average power saving measured using RGB intensity values was 20.09%, as mentioned in Section 2.6.1, where as the average power saving

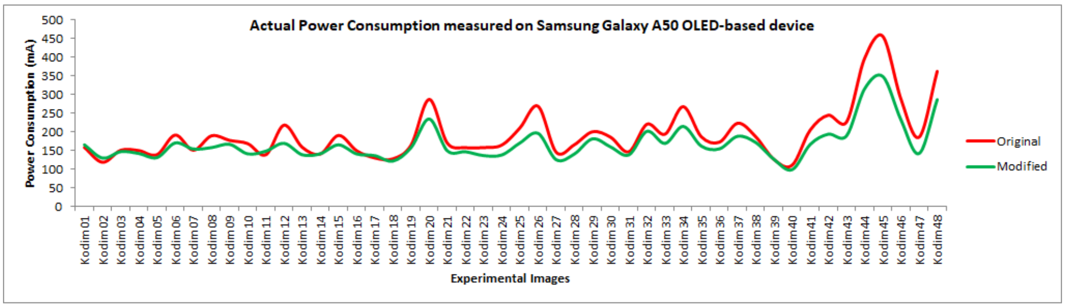


Fig. 2.16 Experimental images vs. actual consumption in (mA) measured on Samsung Galaxy A50 mobile device.

measured on the real mobile device is 13.16%. This difference in the values of power savings is obvious because of the use of real OLED-based device, as there are many components in the smartphone which are responsible in the overall power consumption.

In order to get a deeper insight into the test set, the experimental images were categorized into five groups, namely, Animals, Buildings, GUIs, Nature, and People, based on the image contents. For example, the images in the Animals group contain different types of animals, the images in the Buildings group have some kind of buildings or architecture in them, and so on. The number of images in each group was 6, 10, 8, 13 and 11 respectively. Table 2.2 shows the power reduction percentage ($\Delta P$) for each image in each category. By creating these groups we can analyze in a better way that what type of images contribute in power savings. $\Delta P$ is computed by using Equation (2.4), where P1 and P2 is the actual power consumed by the original and the modified image on Samsung Galaxy A50. The maximum average power reduction was recorded to be 20.49% for the GUI images and the minimum was 7.88% for the Building images. While the power reduction percentages for Nature, Animal and People groups were found to be 11.11%, 8.97%, and 8.51% respectively. The GUI image group contains the screenshots of the GUIs captured from Microsoft, Android and IO devices, as already described in Section 1. The images in the GUI group resulted in the maximum power savings.

## 2.6.5 Application Aspect of the Proposed System

Considering a social networking site, Instagram, for example, has reached over 2 billion monthly active users by December 2021, according to the latest statistics

Table 2.2 Percentage of the power reduction for all the image groups of the experimental test set

| **Animals** | $\Delta P(\%)$ | **Buildings** | $\Delta P(\%)$ | **GUIs** | $\Delta P(\%)$ | **Nature** | $\Delta P(\%)$ | **People** | $\Delta P(\%)$ |
|---|---|---|---|---|---|---|---|---|---|
| A01 | -5.79 | B01 | 4.65 | G01 | -20.80 | N01 | -14.15 | P01 | -9.50 |
| A02 | -15.08 | B02 | 2.24 | G02 | -20.85 | N02 | -13.96 | P02 | -5.22 |
| A03 | -13.60 | B03 | -16.76 | G03 | -19.29 | N03 | -2.59 | P03 | -5.67 |
| A04 | -0.03 | B04 | -12.20 | G04 | -20.50 | N04 | -11.15 | P04 | -22.39 |
| A05 | -11.07 | B05 | -16.27 | G05 | -16.37 | N05 | 6.88 | P05 | 0.50 |
| A06 | -8.23 | B06 | -19.17 | G06 | -23.70 | N06 | -18.28 | P06 | -4.71 |
|  |  | B07 | -19.75 | G07 | -23.52 | N07 | -27.04 | P07 | -15.67 |
|  |  | B08 | -4.03 | G08 | -18.86 | N08 | -10.46 | P08 | -13.29 |
|  |  | B09 | 9.49 |  |  | N09 | -13.47 | P09 | 4.73 |
|  |  | B10 | -7.02 |  |  | N10 | -12.88 | P10 | -16.39 |
|  |  |  |  |  |  | N11 | -8.60 | P11 | -6.03 |
|  |  |  |  |  |  | N12 | -12.83 |  |  |
|  |  |  |  |  |  | N13 | -5.87 |  |  |
| **Average** | **-8.97%** |  | **-7.88%** |  | **-20.49%** |  | **-11.11%** |  | **-8.51%** |

[56]. Instagram has around 1.9 billion daily active users, over 50 billion photos have been uploaded on Instagram so far [57]. These are the statistics for just one social networking platform. Let's talk about Facebook now. According to [58], there are 2.91 billion monthly active users on Facebook, of which, 1.93 billion are daily active users, and the most engaging form of Facebook content is in the form of images and videos. Apart from Instagram and Facebook, there are other popular social networks which contain huge amount of graphical data, such as TikTok, Youtube, Snapchat, etc, with millions of active users per day.

When billions of users scroll through these platforms daily to view graphical contents, one can imagine the extensive power consumption rates in mobile devices. If the proposed system is integrated to these social networks, it can results in huge amount of power savings. According to the experimental results of our proposed system, if on average 13.16% of the energy is saved on each image, then the total amount of energy reduction would be enormous for millions of images accessed by billions of users each day. The CPU power will be consumed only once while uploading an image but that modified image would be used indefinitely by billions of users, saving extensive amounts of energy. As a consequence, the battery life of the mobile devices would be improved significantly.

### 2.6.6 Subjective Evaluation

We have seen the usefulness of the proposed system in terms of power savings by computing power consumption values based on RGB intensity levels and also by measuring the actual power consumption in Samsung Galaxy A50, an OLED-based smartphone. We have also shown that the modified images were better in color contrasts as compared to the original images, by computing the EME (Measure of Enhancement) values. We have also discussed that the proposed method can be applied to social networking platforms that could result in enormous amounts of power savings. All the results presented above are in favor of our proposed system, it is now important to know the end-user acceptability of the modified images produced using the proposed method, as they would be the one viewing the modified images when applied to the social media platforms.

To evaluate the user acceptance, a subjective test was conducted on the images of the experimental test set. In total 99 participants were involved in the subjective test, both males and females of age between 20 and 24 years. It is interesting to note that according to "The 2020 social media demographics guide" [59], the highest percentage of Facebook and Instagram users are between 18 and 29 years of age. This fact justifies our subjective participants to come under the category of most active social media users. All participants had a normal or corrected-to-normal vision.

The test was created using Google forms as follows: for each image, three options were provided and the participants were asked to select one of them. The three options included the original image, corresponding modified image, and a text option "indifferent". The images were placed side by side (one image on the left side of the screen and the other on the right side) and the "indifferent" option was placed as a plain text below the images as shown in Fig. 2.17. The participants did not know which image was in its original form and which was the modified image. In order to counteract side bias, for randomly chosen half of the cases, the original images were placed on the left side and the modified versions on the right side and vice versa. In this way, the participants would not have any idea that which side of the screen contained which image version. Each participant was provided with the test link and was asked to perform the test on his/her laptop or desktop. The brightness level of the screen was not suggested, neither the time limit was provided, to imitate the real-world scenarios. The participants were asked to select one of the images (either

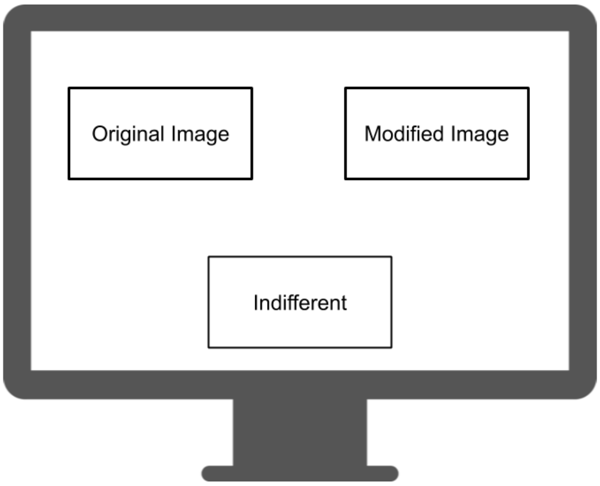


Fig. 2.17 An example of the subjective test performed on the experimental images.

left one or the right one) based on the visual quality of images i.e. whichever image looks better to them. They were advised to select the third option, “indifferent”, only when the two images seemed to be indistinguishable from each other.

The purpose of performing subjective evaluation was to know the extent to which the modified images (which consumed less power than the original images) were preferred over the original images. The results of the subjective test are represented graphically in Fig. 2.18. According to these results, 58% of the total test images were preferred in their modified version, while only 35% images were chosen in their original form. Moreover, 7% of the images were indistinguishable in their original and modified versions. In other words, 7% of the modified images were visually same as that of the original images, which describes another positive aspect of the proposed system. It can be concluded that only 35% of the experimental images were preferred in their original version, and all the other images were preferred by the users in the modified form. These results further confirm that the proposed system is successful in reducing power consumption as well as enhancing the visual quality of images and that the end-users mostly prefer the modified images over original images due to better color contrasts.

To further understand the subjective evaluation, these results were separated for each of the five image groups described earlier. The categorical subjective evaluation is depicted in Fig. 2.19. The chart shows the number of votes for each category of images. It can be clearly seen that there is a higher number of votes for the modified images as compared to the original images for all image groups except for the GUI

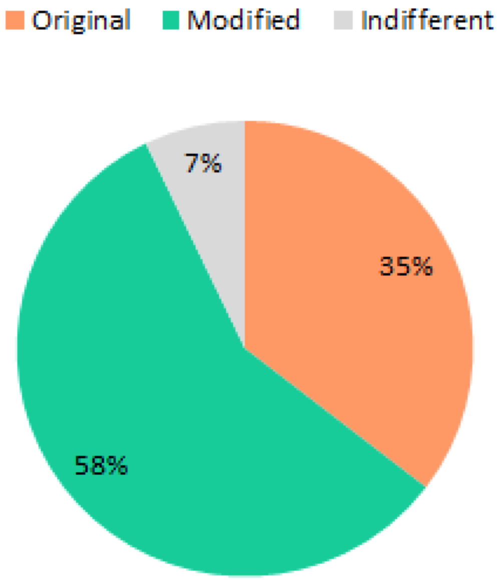


Fig. 2.18 Subjective evaluation results for the original and the modified images.

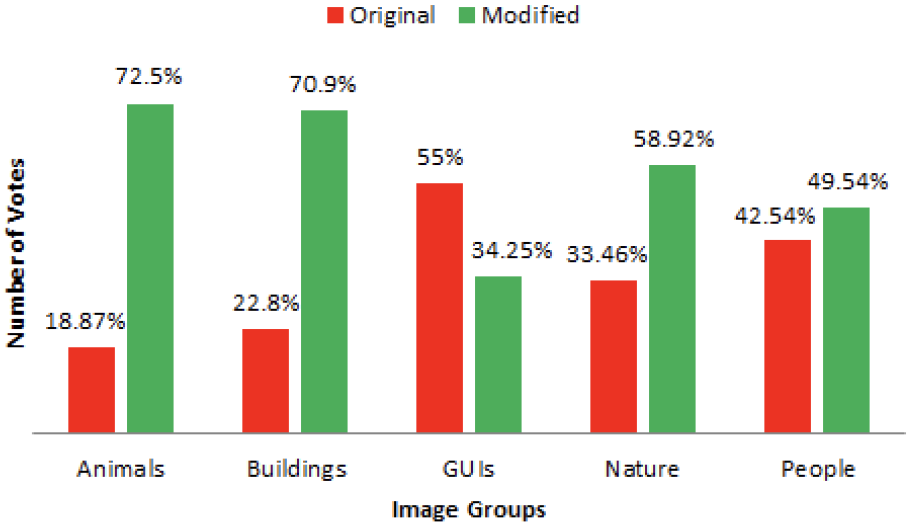


Fig. 2.19 Subjective evaluation for each category of images.

images. 55% of participants preferred GUI images in their original form whereas 34.25% of participants opted them in their modified versions. There is an interesting point to note, although Table 2.2 showed that the maximum value of power reduction was recorded for the GUI group, it is the least preferred group for modified images according to the results of subjective evaluation, as shown in Fig. 2.19. The reason for this contradiction is very simple, the proposed system aims at darkening the images by enhancing color contrasts and it does the same for the GUI images. But this image darkening or the contrast enhancement is not desirable for GUI images. This is not surprising as the proposed system is not targeted for modifying the GUIs, instead the system is proposed for general images regularly posted on social media platforms to reduce the power consumption in mobile devices.

Moreover, the modified images from Animal and Building groups showed a higher acceptance ratio (72.5% and 70.9% respectively) as compared to the modified images from Nature and People groups whose acceptance rate is 58.92% and 49.54% respectively. It can be concluded from these results that the proposed system is widely acceptable in general for a variety of images except for the GUIs.

### 2.6.7 GIMP Plugin

The experiments were conducted using an open-source image manipulation tool, known as GIMP (GNU Image Manipulation Program) [60]. A GIMP Plugin was created in Python programming language to automatically generate modified images using the proposed method. Given an input image, the plugin applies the proposed Land-Effect followed by the white-balance operation and the retinex filter. Finally, it reduces the RGB intensity levels of the image and provides the modified image as an output of the program. The python code for the plugin is provided in Appendix B.2.

### 2.6.8 Implementation Issues

The average processing time for producing energy-efficient images of the experimental test set, when executed on an Intel Core i7 having a 3.6 GHz processor with 8 GB RAM, with the unoptimized developed plugin, was recorded to be 3.6 seconds. The CPU power usage was also observed while running the plugin on the images. On an average, an image of size (768 x 512 px) consumed only about 5% of the CPU power during the image modification process, whereas a larger image of size (4096 x 2304 px) used about 13% of the total CPU power. Hence, the proposed system in its current form is more suitable for server-side implementation. The future work would be to optimize the plugin for real-time image modifications.

It is important to note that, it was claimed earlier that the experimental test set contains different size images to check the applicability of the proposed work. From the results it is clear that the proposed system was able to process all the provided images, although the CPU power usage was low for smaller images and high for larger images.

### 2.6.9 Comparison of Results with Related Work

The proposed method is compared with two existing works found in the literature, as discussed in the following subsections.

#### Good Looking Green Images (GLGI)

The proposed algorithm is compared with another method of producing energy-efficient, or Good Looking Green Images [29], as they named it. We will refer to this method as GLGI for short. Hadizadeh et al. [29] proposed a method for generating good looking and less energy-consuming images when presented on an energy-adaptive display. They have used 24 images of the Kodak image database [49] for their experiments. At first they performed color quantization on the images. Starting from a color quantized image, colors were first converted to the CIELAB color space. They have used just-noticeable-difference (JND) model to find the radius of spheres containing perceptually indistinguishable colors. The strategy was to find some alternative colors for each sphere, which consume less power, to reduce the overall power consumption [29]. The final energy consumption of the modified images was computed by using Equations (2.3) and (2.4).

In order to compare the experimental results obtained by the proposed method with the GLGI method, we extracted the results of the same 24 images from the Kodak image database for fair comparisons. Fig. 2.20 shows the comparison of the power reduction percentages of the modified images by GLGI [29] and the proposed method. As can be seen, the proposed method significantly outperforms the GLGI method in terms of power reduction of the images.

A detailed comparison between GLGI and the proposed system is provided in Table 2.3, based on different comparison parameters. Experimental test images have been taken from the Kodak image database (24 images in total) for both the systems under consideration. The GLGI method reduces the power consumption by 4.25% on average as compared to 20.46% by the proposed method when applied on the same set of test images. In other words, the proposed method reduces the power consumption by almost 5 times, on average, as compared to the GLGI method. The next comparison parameter is the average processing time for modifying original images into their energy-efficient version. The GLGI method required 23 seconds on average, where as the proposed method required only 1 second on average for the images of same size. Hence, despite of saving enormous amount of power, the proposed method is also faster in terms of processing time. Furthermore, the results of subjective evaluation of the GLGI and the proposed method can be considered as very close to each other. It should be noted that the number of participants involved in the subjective evaluation of the GLGI and the proposed method is 24 and 99, respectively. Based on the subjective results, the original images in the GLGI and the proposed method were preferred by almost 30% and 36% of the participants, respectively. Similarly, the modified images in the GLGI and the proposed method were preferred by almost 69% and 58% of the participants, respectively. Thus, considering the higher number of participants, the proposed method is highly acceptable in its current form.

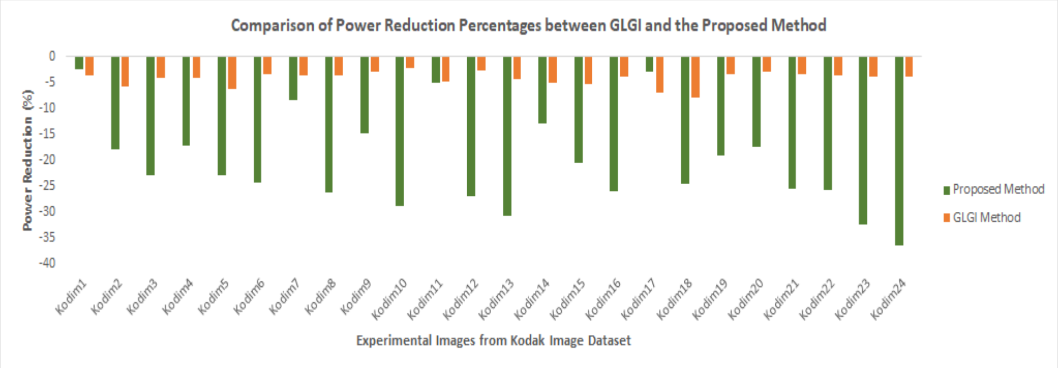


Fig. 2.20 Comparison of power reduction results between the proposed and the GLGI Method

Table 2.3 Comparison of the GLGI and the proposed method based on different parameters

| **Comparison Parameters** | **GLGI Method** | **Proposed Method** |
|---|---:|---:|
| **Number of images** | 24 | 24 |
| **Power savings** | 4.25% | 20.46% |
| **Average processing time** | 23 seconds | 1 second |
| **Number of subjective test participants** | 24 | 99 |
| **Percentage of votes for original images** | 30.55% | 36.54% |
| **Percentage of votes for modified images** | 69.44% | 57.95% |
| **Percentage of Votes for indifferent** | - | 4.5% |

### Histogram Equalization (HE)

Histogram Equalization (HE) is a very effective technique for contrast enhancement, widely used in literature [31, 32, 53, 61]. Histogram equalization is a method to process images in order to adjust the contrast of an image by modifying the intensity distribution of the histogram [62]. It spreads out the most frequent pixel intensity values or stretches out the intensity range of an image, to enhance the image contrast. Although, being a powerful technique in enhancing the color contrast, HE sometimes results in an overly-enhanced image which gives unnatural look and creates visual artifacts [31]. As discussed earlier, EME (Measure of Enhancement) is widely used metric to measure color contrasts in images. Fig. 2.21 shows the comparison of EME values of the modified images by using HE and the proposed method. HE outperforms the proposed method in terms of EME values for almost 70% of the cases (because the higher the EME, the better is the contrast). Interestingly, a higher value of EME does not necessarily mean better visual results. This can be verified by looking at Fig. 2.22, which provides comparison of the EME values and the visual results of the HE and the proposed method. Undoubtedly, the results produced by HE are not very pleasing in spite of the higher value of EME. It can be concluded

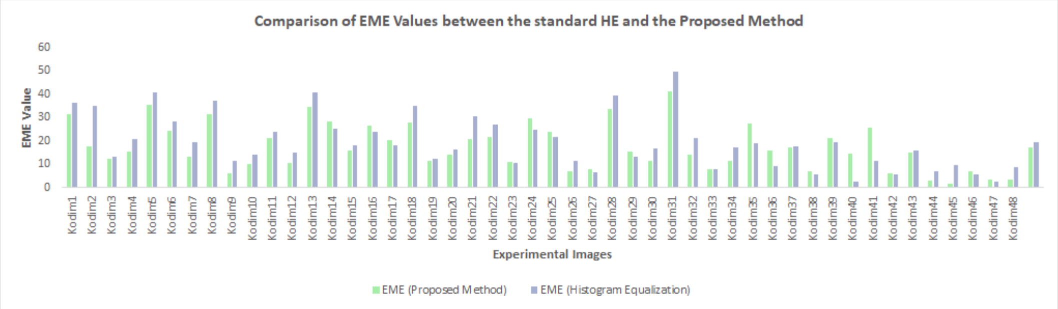
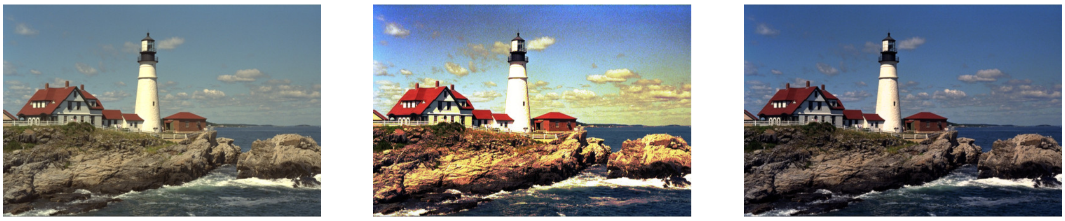


Fig. 2.21 Comparison of EME Values between the standard HE and the proposed method



(a) Original Image
EME=14.30

(b) Modified with HE
EME=30.45

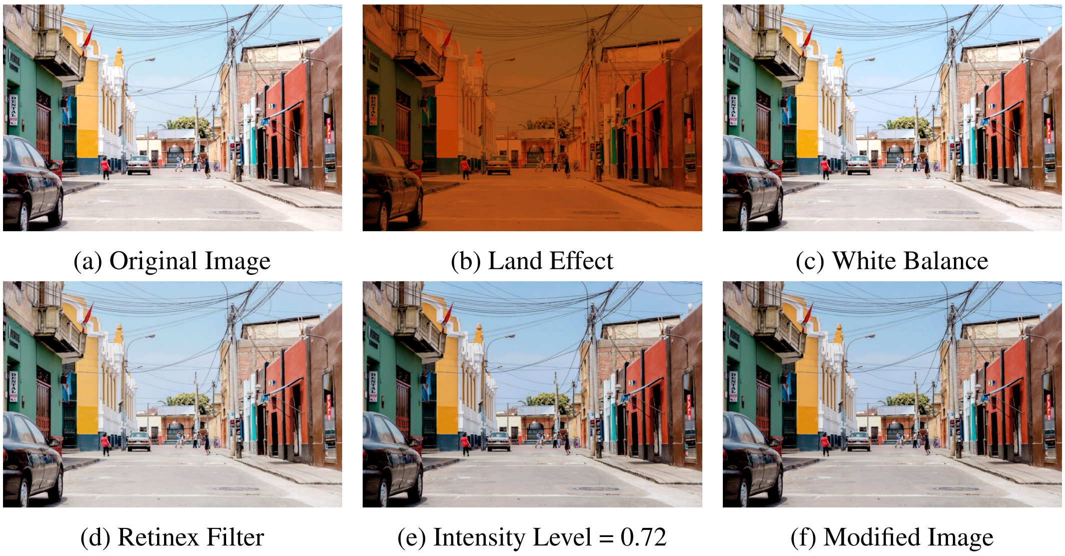
(c) Proposed method
EME=20.74

Fig. 2.22 Comparison of experimental visual results of a sample image. (a) Original Image, (b) modified image using HE, and (c) modified image using the proposed method

that although the EME values of the proposed method are less than that of HE but the visual results of the proposed method are better as compared to that of the HE.

## 2.6.10 Visual Results

This section presents some sample images to show the visual differences between their original and modified versions. Fig. 2.23 illustrates step by step results of applying the proposed method on a sample image named B07.png. (Note - This image is taken from the Buildings group as it contains different buildings). Fig. 2.23a is the original image, taken from the experimental test set. After applying the proposed Land-Effect on the original image, the resulting image is shown in Fig. 2.23b. It can be seen that although the blue color component is not taken into account during this step, different shades of blue can be perceived from the image. Apart from that, the visual quality of the image is far from acceptance. To enhance the perceptual quality, white balancing was performed and the resulting image is depicted in Fig. 2.23c. The effect of applying the white-balance operation is clearly

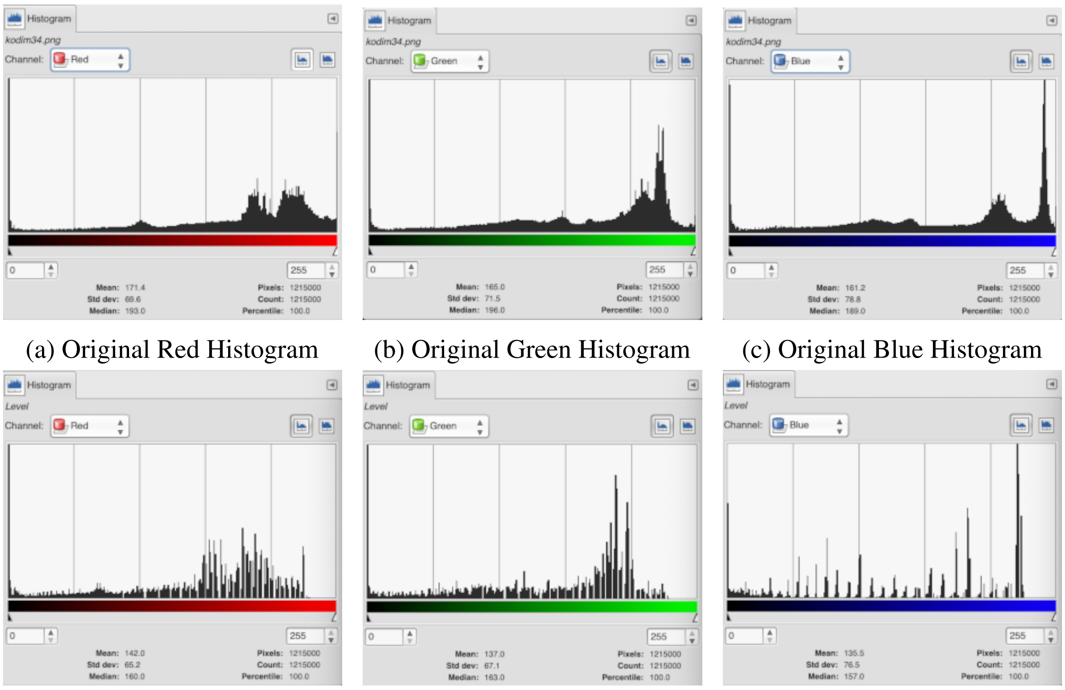

(a) Original Image (b) Land Effect (c) White Balance

(d) Retinex Filter (e) Intensity Level = 0.72 (f) Modified Image

Fig. 2.23 Experimental results for a sample image named B07.png. The modified image (f) consumes 19.75% less power than the original image (a) and is preferred by 71% of the participants of the subjective test.

visible in the image. Although the visual quality was greatly improved, the image appeared to be too bright for human eyes. To fix this issue, and to represent the image similar to as it would be perceived by the human eye, the Retinex filter was applied and Fig. 2.23d illustrates the obtained result. Finally, the last step of lowering the overall RGB intensity value was applied, as shown in Fig. 2.23e, to further enhance the color contrasts and to reduce power consumption. The final modified image is shown in Fig. 2.23f. The power saving percentage for the image was recorded to be 19.75% when displayed on Samsung Galaxy A50, and it was preferred by 71% of the participants in the subjective test.

Now we will discuss the power savings in terms of RGB intensity levels. The RGB intensity level histograms of the original and modified versions of image B07.png are shown in Fig. 2.24. Figs. 2.24a, 2.24b and 2.24c show the red, green and blue intensity level histograms respectively for the original image. The red, green and blue histograms for the modified image are illustrated in Figs. 2.24d, 2.24e and 2.24f respectively. It is interesting to note that the modified histograms are in a stripped form which was introduced during the white balance operation and was improved during the Retinex filter operation, as discussed earlier. The proposed system successfully reduced the red, green and blue intensity levels by 17.15%, 16.96%, and 15.94% respectively in the modified image.



kodim34.png
Channel: Red
0 255
Mean: 171.4 Pixels: 1215000
Std dev: 69.6 Count: 1215000
Median: 193.0 Percentile: 100.0

(a) Original Red Histogram

kodim34.png
Channel: Green
0 255
Mean: 165.0 Pixels: 1215000
Std dev: 71.5 Count: 1215000
Median: 196.0 Percentile: 100.0

(b) Original Green Histogram

kodim34.png
Channel: Blue
0 255
Mean: 161.2 Pixels: 1215000
Std dev: 78.8 Count: 1215000
Median: 189.0 Percentile: 100.0

(c) Original Blue Histogram

Level
Channel: Red
0 255
Mean: 142.0 Pixels: 1215000
Std dev: 65.2 Count: 1215000
Median: 160.0 Percentile: 100.0

(d) Modified Red Histogram

Level
Channel: Green
0 255
Mean: 137.0 Pixels: 1215000
Std dev: 67.1 Count: 1215000
Median: 163.0 Percentile: 100.0

(e) Modified Green Histogram

Level
Channel: Blue
0 255
Mean: 135.5 Pixels: 1215000
Std dev: 76.5 Count: 1215000
Median: 157.0 Percentile: 100.0

(f) Modified Blue Histogram

Fig. 2.24 RGB Histograms for the sample image B07.png. (a), (b) and (c) are the original red, green and blue intensity level histograms respectively and (d), (e) and (f) are the corresponding histograms for the modified image. Intensity of the red, green and blue colors is reduced by 17.15%, 16.96% and 15.94% respectively in the modified image.

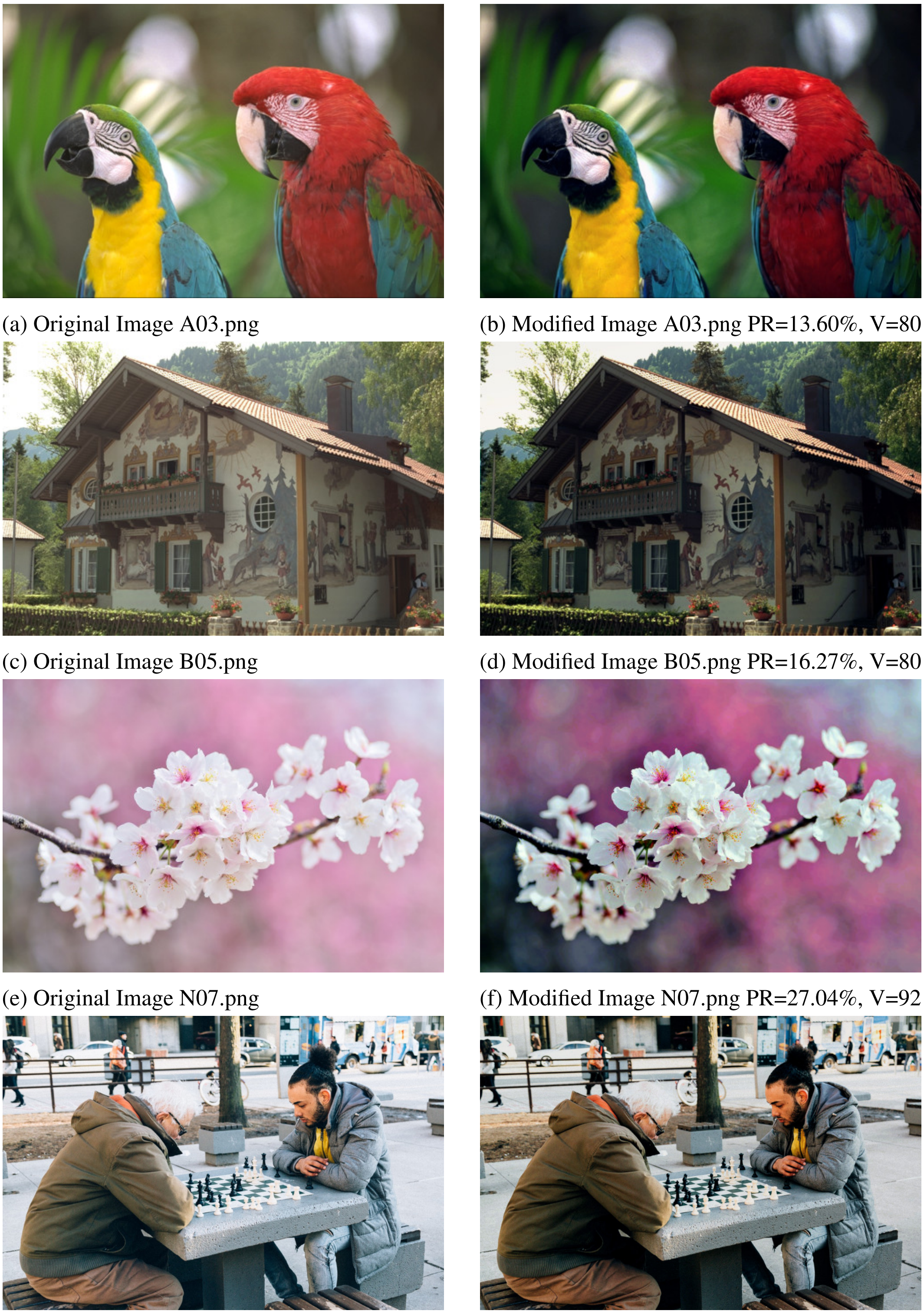
Fig. 2.25 shows some randomly chosen sample images, one from each group (except the GUI group), taken from the experimental test set. The images in the left column are the original images, while those in the right column are the corresponding modified images obtained by applying the proposed system. The visual differences in the original and modified versions of images are clearly visible. The modified images have better color contrasts and visually better appearance as compared to the original images. Moreover, the modified image, shown in Fig. 2.25b, saved 13.60% energy and received 80 out of 99 votes by the subjective test participants. Similarly, the images presented in Figs. 2.25d, 2.25f and 2.25h reduced the power consumption by 16.27%, 27.04% and 15.67% respectively with 80, 92 and 62 votes out of 99 participants.

In addition to the above results, we would like to show why the proposed system is not suitable for GUIs even though saving enormous amount of energy in those images. Two sample images from the GUI group are presented in Fig. 2.26. As can be seen, Fig. 2.26b is the modified version of Fig. 2.26a with power-saving value of 20.80% and 50 out of 99 votes. Similarly, Fig. 2.26d shows the modified version of the original image shown in Fig. 2.26c with power reduction percentage 20.50% and only 29 votes out of 99 participants. It is quite apparent that although the modified images contributed towards huge power savings, they are much darker as compared to the original images and therefore were less preferred by the participants. Generally, the end-users did not like such dark GUIs.

It can be concluded from all the experimental results that the proposed system is extremely successful in producing energy-efficient and contrast-enhanced versions of a wide variety of images. The proposed system is able to save up to 13.16% of power on an average.

## 2.7 Conclusion

Technological advancement in recent years have opened numerous research domains in academia. One such domain is the invention of OLED displays. Traditionally there were LCD displays whose working principle was very different from the OLED displays, such that the LCD displays require a constant backlight to display contents whereas the backlight was not present in OLED displays. The power consumption in OLED directly depends on the display contents, as the RGB pixels are emissive



(a) Original Image A03.png

(b) Modified Image A03.png PR=13.60%, V=80

(c) Original Image B05.png

(d) Modified Image B05.png PR=16.27%, V=80

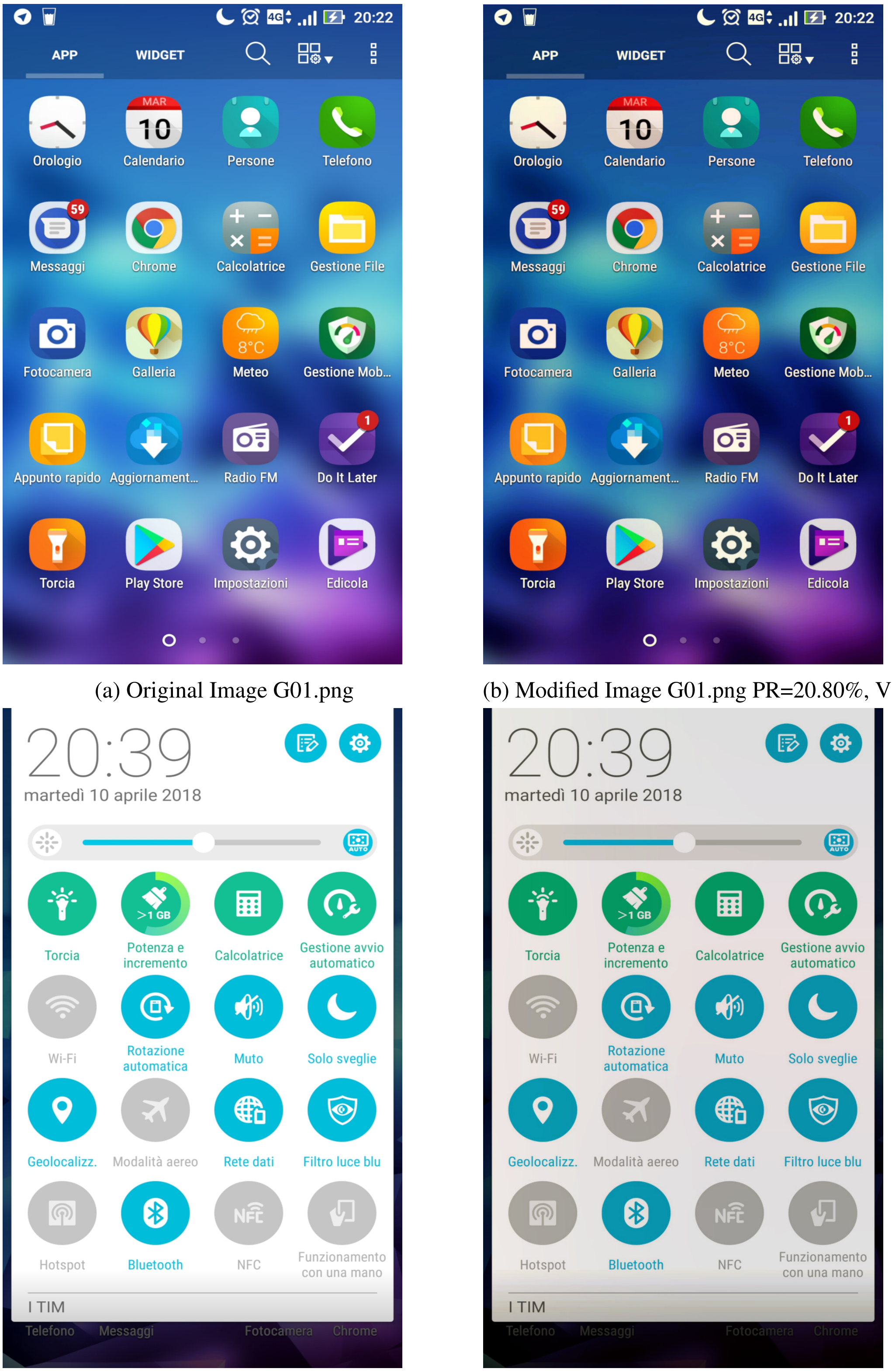
(e) Original Image N07.png

(f) Modified Image N07.png PR=27.04%, V=92

(g) Original Image P07.png

(h) Modified Image P07.png PR=15.67%, V=62

Fig. 2.25 Sample images taken from the experimental test set, one from each image group. Images in the first row are the original images, while the second row shows the corresponding modified versions along with the power reduction (PR) percentages and the total number of votes (V) for that image.



(a) Original Image G01.png

(b) Modified Image G01.png PR=20.80%, V=50

(c) Original Image G04.png

(d) Modified Image G04.png PR=20.50%, V=29

Fig. 2.26 Sample images taken from the GUI group. Image (b) is the modified version of image (a) with power reduction value of 20.80% and subjective votes equal to 50. Image shown in (d) is the modified form of image (c) with 20.50% power reduction and 29 subjective votes.

in nature. This property of OLED displays attracted many researchers to work on reducing power consumption in OLED devices by image manipulation techniques. As the power consumption in OLED devices is directly proportional to the RGB intensity levels, which means the lower the RGB intensity the lower will be the power consumption and vice versa. This fact has been utilized in this work to achieve power savings in OLED-based displays by manipulating image pixels. In summary, the proposed system comprised of four steps, namely, the modified Land-Effect, white-balancing, retinex filter and adjustment of RGB intensity values. The modified images resulted in huge power savings and were preferred the most during subjective test analysis.

# Chapter 3

# Identification of Healthy and Contaminated Hazelnuts

A machine vision application is presented in this chapter. It is related to the digital monitoring of a food item by applying digital image processing techniques. Each and every food item must be checked before its consumption. Either the food is consumed directly or it is used as an ingredient in some other edible items, its quality must be assured before using it. In this chapter we will study about hazelnuts, one of the most consumed nuts in the food industry, particularly in chocolate and baking industry. The traditional way of examining the quality of hazelnuts is by means of visual inspection, and there are various limitations of the manual assessment of the hazelnuts. In this chapter we will see how some very simple yet effective image processing techniques are applied to examine the quality of hazelnuts.

Some of the work presented in this chapter has been originally published in [63] and a part of the work is currently being reviewed in a scientific research journal.

## Roadmap

The rest of the chapter is organized as follows: Section 3.1 introduces the topic and presents the problem statement. Section 3.2 presents the literature review conducted for the presented work. The proposed method is described in detail in Section 3.3.

The experimental results are presented and discussed in Section 3.4, followed by the conclusion in Section 3.5.

## 3.1 Introduction

Hazelnuts *(Corylus avellana)* are among the most consumed nuts in the world because of their great organoleptic characteristics and their rich nutritional properties. Hazelnuts are one of the main ingredients used in pastry, confectionery, bakery and chocolate industries, as they add intense flavor and texture to food items [64, 65]. Furthermore, hazelnuts play an important role in human health as they contain numerous essential nutrients, including protein, oil, vitamins, and minerals [66, 67].

According to the Food and Agriculture Organization Statistical Database site [68], approximately 1,096,571 tonnes of in-shell hazelnuts were produced worldwide during the year 2020, making hazelnuts one of the most popular crops in the world. The top annual producers of hazelnuts are Turkey, Italy, the USA, and Azerbaijan, with crop production in 2020 of 665000, 140560, 64410, and 49465 tonnes, respectively. Other minor producers are Chile, Georgia, and China.

Like almost every other kind of food, hazelnuts are not an exception from potential problems in terms of quality degradation, which may occur directly in the field during the growing phase, or later during harvesting and storage phases. Temperature, humidity, soil, and storage conditions are among the most critical factors for the contamination of various food crops, including hazelnuts [69]. The quality of hazelnuts is usually assessed in terms of freshness of the product, presence/absence of contaminants, insects, mites, and other visual defects such as the presence of damaged, infested, and discolored kernels. The quality of hazelnuts directly affects their export revenues [70].

Hazelnut kernel defects can be classified into the following types: blanks, doubles, poorly filled nuts, moldy kernels, brown-stain disorder, kernels with black tips, and shriveled kernels [71]. Some other common defects are insect-damaged nuts, rotten and decayed kernels. In the present study, we will be focusing on two types of defected hazelnuts, i.e. pest-infected hazelnuts and rotten hazelnuts.

Pest-infected hazelnuts, as the name suggests, are caused by different species of bugs, insects, and pests which feed on nut crops. Severe climatic condition is one of the major causes of insect-damaged crops. Two main types of damage occur depending on the bug attack in various stages of fruit maturation:

1. “blank”, early attacks during nut development lead to seed abortion and nut drop [72];
2. “cimiciato”, results from the bug attacks during the advanced stage of kernel growth. These bug attacks result in whitish or brownish spots, spongy tissues, and surface depression [72, 73]. In general, the pest bites usually result in localized brown spots which further cause off-flavors in hazelnuts.

While the primary reason for pest-infected hazelnuts is insect bites, there are several causes for damaged or rotten hazelnuts, such as micro-fungal diseases, high moisture content, improper drying, post-harvest storage conditions, etc. The initial insect infestation may also cause damage to the kernel at later stages [74]. High moisture content can accelerate the development of micro-fungi, producing specific enzymes able to break down carbohydrates into their monomers. It can also hydrolyze lipids into free fatty acids [65, 69, 75, 76]. These chemical reactions cause undesirable odors, decreased nutritional value, and bitterness in taste perceived by consumers, and hence they reduce the consumer acceptability [73, 77, 78]. This loss in the quality of hazelnuts negatively impacts the food industry [79].

Due to these reasons, it is extremely important for industries to carefully examine the incoming batches of hazelnuts before processing them for food items. The current quality control procedures involve visual inspection and manual sorting, where expert assessors analyze representative samples collected from each incoming batch of hazelnuts. The decision regarding the quality of hazelnuts is made by looking at the samples, and the hazelnuts are separated into their respective groups depending on their quality. Traditional inspection systems have several drawbacks. Examinations and control made by human personnel are time-consuming, tedious, labor-intensive, and costly. Moreover, the visual process is expert-dependent and cannot be delegated and the results might be inconsistent over time [70, 80]. Considering all the above-mentioned problems, there is a dire need for the development of an autonomous inspection system that would be more reliable and consistent. Furthermore, the automated system would be cost-effective, labor-saving, and time-saving.

Multi-spectral and fluorescence imaging techniques have massively contributed to measuring the quality of different food items over the past few decades. Fluorescence imaging is a type of imaging technique that helps in visualizing the internal processes taking place in an organism or identifying the internal structure of an object. It is widely used in the fields of molecular biology and biochemical laboratories as well as in quality control applications. When a certain molecule absorbs electromagnetic radiations, the electrons get excited by the absorption of light and eventually move towards higher energy states. After a short duration of time (typically of the order of nanoseconds), the electrons come back to their ground state by emitting fluorescent light. The intensity of the emitted fluorescent light can be measured by means of a CCD-Camera by capturing a digital image. This technique is known as Fluorescence Imaging [81]. Most of the spectral imaging systems found in the literature use some kind of light to illuminate samples and then measure the light reflected by the sample to measure the internal structure or the quality of the sample under consideration.

Despite the effectiveness of spectral imaging technology in detecting defects and contamination in various kinds of fruits, this approach has been poorly exploited with hazelnuts, as only a couple of studies are available in scientific literature [82–84]. The current work fills this gap, by presenting and evaluating a fluorescence imaging technique able to classify the hazelnut kernels into healthy, pest-infected, and rotten classes. Samples of hazelnut kernels were illuminated by Ultra Violet (UV) light in the specially developed axial illumination system. The images of the illuminated hazelnut kernels were then captured at 450nm and 750nm by using emission filters. These images were further processed to classify the hazelnuts in one of the three possible classes.

## 3.2 Related work

A number of studies have been performed to classify hazelnuts into different categories based on several features. Menesatti et al. [85] demonstrated the use of shape-based techniques to discriminate among four traditional Italian cultivars of hazelnuts (Tonda di Giffoni, San Giovanni, Mortarella, and Tonda Romana). In a study proposed by Bayrakdar et al. [86], the type and quality of shelled hazelnuts were determined with image processing by using the size and shape characteristics of hazelnuts. The hazelnuts were classified into class I, class II or class extra according to the size of the diameter of the shelled hazelnuts. Guvenc et al. [87] used the computer vision approach and chose various image features such as color contrast, homogeneity, energy, gray level average, and standard deviation to classify hazelnuts into fully peeled, partially peeled, and rotten classes. Moscetti et al. [79] proposed High Dynamic Range (HDR) short-wave infrared hyper-spectral imaging to sort hazelnuts into one of the following four classes namely, class extra, class I, class II and damaged hazelnuts class. The authors in [88] used image processing and mean-based, k-mean clustering technique to classify hazelnut varieties according to the size of the fruit, i.e., big, medium, and small. Caner et al. [89] classified three different hazelnut cultivars (‘Sivri’, ‘Kara’, ‘Tombul’) using algorithms of deep learning and ensemble learning. A deep learning Convolutional Neural Network (CNN) model was used in [70] to classify 17 widely grown varieties of hazelnuts.

Automatic detection of defected hazelnuts is of utmost importance for industries to separate contaminated hazelnuts from uncontaminated ones. In this regard, several different techniques have been found in the literature. Non-invasive and non-destructive technology, such as Near Infra-Red (NIR) spectroscopy, has gained popularity over the years to detect hidden defects in hazelnuts by means of measuring the moisture content, acidity, lipid oxidation, fatty acid content, and internal disorders. Bellincontro et al. [90] measured free acidity and water content in hazelnuts using NIR technique. In their extended study [91], the authors used NIR-AOTF along with Magnetic Resonance Imaging (MRI) to predict oxidation status on the basis of water and oil presence in whole hazelnuts. Moscetti et al. [92] used NIR spectroscopy on hazelnut kernels to detect internal flaws such as abnormal coloration of the skin and/or kernel, insect damage, detrimental disorders, and fungal growth. The authors in [93], by means of the NIR technique, estimated the degree of hazelnut kernel defect and the level of lipid oxidation in hazelnuts. Khosa and Pasero [94] proposed an X-ray imaging-based system for the classification of hazelnuts into healthy, damaged, and infected classes. Nuclear Magnetic Resonance (NMR) technique was proposed by Di Caro et al. [95] to detect hidden defects in whole hazelnuts by examining the oil extracted from healthy and unhealthy hazelnuts. All the above-mentioned techniques are expensive and require specially trained personnel to handle sophisticated equipment. Moreover, the NIR spectroscopy depends on less-precise reference methods, which is the major limitation of this technique in food analysis [96]. In a recent study [80], an RGB colorgram based automated system was proposed for the detection of defective hazelnuts. The authors used multivariate analysis methods to create classification models on 2000 training images, to classify hazelnuts into “sound”, “rotten”, and “pest-affected” categories. RGB-based classification techniques usually require an extensive image dataset for correctly training the models.

Despite their high accuracy and effectiveness, the above mentioned methods are expensive, time consuming, and difficult to use for industrial purposes. Our proposed system, on the other hand, does not require any sophisticated equipment, other than camera, filters, lens and mirror, which are commonly available and easy to use. It is cost-effective and easy to install in industries. In fact, our proposed solution simplifies the multi-spectral imaging by capturing the images only on two frequencies. Surely, multi-spectral imaging techniques have contributed significantly in food quality assessment due to numerous advantages. Spectral imaging technology has been utilized to detect defects and contamination in various dried fruits such as pecans [97], peanuts [98], figs [99], pistachio nuts [100], different fruits such as cherries [101], apples [102, 103], tomato [104], nectarines [105], and other food items like, corn kernels [106], and bell peppers [107]. However, only a few studies have been reported for the use of multi-spectral imaging to identify defects in hazelnuts [82–84]. The authors, in those studies, used the Local Discriminant Bases (LDB) algorithm to extract the most discriminative features along the spectral and spatial frequency axis of the multi-spectral images of shelled hazelnuts for the classification of uninfested and fungal-infested hazelnuts. In their proposed system, the samples were screened using 12 different filters, and then three feature selection algorithms were used followed by a linear classifier. In order to obtain the fungal-infested nuts, the authors artificially contaminated the hazelnuts by altering the moisture content and parasiticus inoculation. Although the authors in [84] initiated the study of hazelnuts using multi-spectral imaging, the use of lots of filters and the feature extraction and selection process can be regarded as complex in terms of computation. The lack of the number of studies using spectral imaging to identify defects in hazelnuts is the main motivation behind this work.

## 3.3 Proposed Method

### 3.3.1 Hazelnut Samples

Three different categories of hazelnut samples, pest-infected, rotten, and healthy hazelnuts, have been used in this study. The sample set consists of 63 hazelnuts in

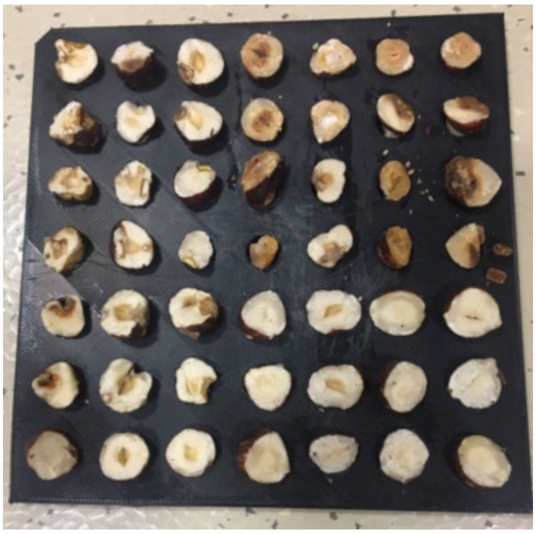


Fig. 3.1 A tray containing hazelnut samples from the pest-infected, rotten and healthy categories

total, thanks to Ferrero SpA. for providing us with the hazelnut samples required for conducting the experiments. The provided samples were half-cut from the center such that the hazelnut kernel is clearly visible. We conducted the experiments on 21 half-cut hazelnuts from each of the three categories.

The hazelnut samples were placed in a 7x7 plastic tray with small holders for placing the hazelnuts as shown in Fig. 3.1. Hazelnut samples from all the three categories can be seen in the figure.

### 3.3.2 Image Acquisition Setup

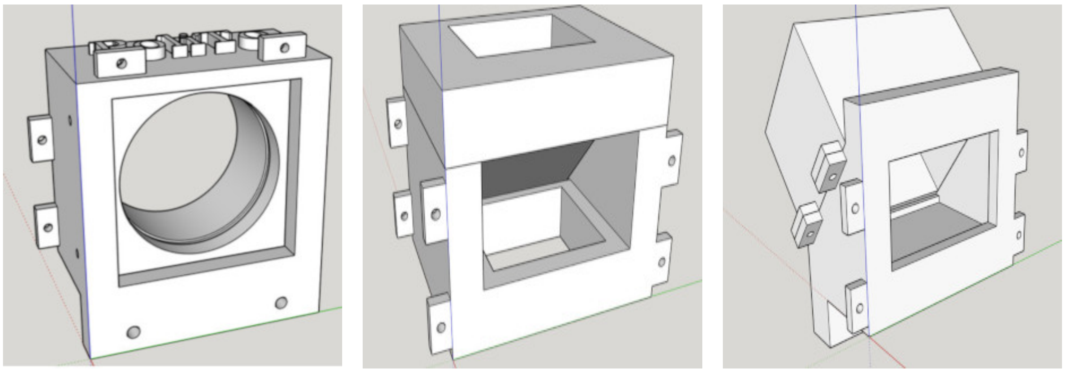
The images of the hazelnut samples were captured under Ultraviolet (UV) light, which requires a special setup to properly illuminate the hazelnut samples, so that better images can be captured for performing the experimental tests. Following components were required to setup the image acquisition system:

- Mirror: A 3mm thick semi-reflective mirror was used. A semi-reflective mirror is also known as a one-way mirror as it is reflective on one side and transparent on the other side. They are actually half-silvered i.e. only a very thin layer of reflective coating is applied to them to make them semi-reflective. When light hits a one-way mirror, only half of the light is reflected back, while the other half passes straight through it.

- Lens: A condenser lens was used with the following specifications: 75mm diameter, 0.75 numerical aperture, 50mm focal length, 3mm edge thickness, and 30mm center thickness. The lens (EO46244) was bought from Edmund Optics [108]. Condenser lenses are specifically designed for illumination applications. They typically have large apertures and short focal lengths. The purpose of a condenser lens is to render a divergent beam of light from source into a parallel or converging beam of light to illuminate a sample under observation.
- Camera: A camera is a core component in any imaging application. The Point Gray Flea2 FL2G-13S2M camera was used in this work to capture images of hazelnut samples, The technical specifications of the camera are as follows: pixel resolution of 1288 x 964, frame rate of 30 FPS (Frames Per Second), 1.3 MP (Megapixels), CCD sensor, and pixel size of 3.75 $\mu$m.
- Interference Filters: Two band-pass filters, with variable bandwidth (around 20 nm), were used to capture images at 450nm and 750nm.

The hazelnut samples were illuminated using axial illumination technique, which guarantees uniform illumination of the samples under consideration. The mirror, the lens and the camera must all be setup properly for capturing the sample images. The Ultraviolet light was chosen as the source light to illuminate the samples. The working principle of an axial illumination system is as follows: at first there is a source light, in our case the UV light, then there is the condenser lens which takes the beam from the source light and converges it to the mirror placed ahead of the lens. The hazelnut samples must be placed below the mirror. Half of the light coming to the mirror is reflected and goes down to illuminate the hazelnuts while the other half passes straight through it. This other half of the light must be trapped in a black box to protect access of light to pass to the camera. The camera must be placed directly above the mirror to capture the illuminated hazelnut samples.

In order to place and align the hardware components, we need a lens holder, a mirror holder and a light trapping box. All these components were designed and modeled using a 3D design software Sketchup (by Trimble Inc., Online Web version) [109]. Fig. 3.2 shows the modeled components of the illuminator. It consists of a



(a) Lens holder

(b) Mirror holder

(c) Light trap

Fig. 3.2 3D modeled components of axial illuminator

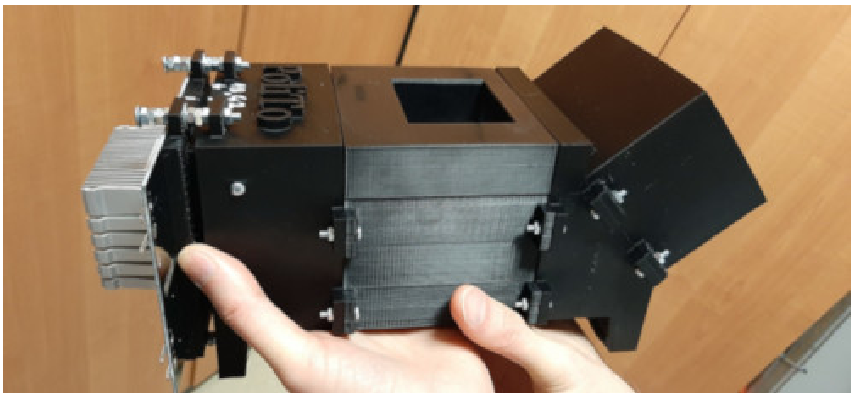


Fig. 3.3 3D printed components of axial illuminator

lens holder, a mirror holder and a light-trap box as shown in Figs. 3.2a, 3.2b and 3.2c. The 3D designed components were then printed using MakerBot Replicator2, a 3D printer. The 3D printed model of the illuminator is presented in Fig. 3.3.

A PCB (Printed Circuit Board), with a heat sink, can also be seen attached to the illuminator components in Fig. 3.3. The PCB was designed to work as a light source selector. Different LEDs, including, white, green, UV, and high power LED light were attached to the PCB as shown in Fig. 3.4. Any light source can be selected to illuminate the hazelnut samples. As we are working with fluorescence imaging, the UV light was used as the source light.

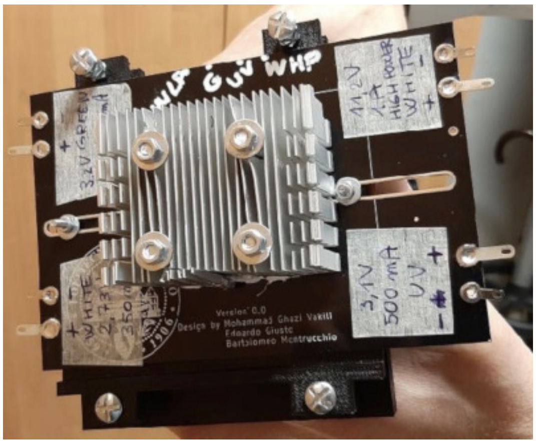


Fig. 3.4 PCB with light source selector and a heat sink

### 3.3.3 Acquiring Sample Images

Fig. 3.5 shows the setup created for acquiring sample images of hazelnuts for performing the experiments. Fig. 3.6 illustrates the close-up top view of the illuminator when the hazelnuts are placed underneath and are illuminated by light. The mirror is set up at approximately 45 degrees angle inside the mirror holder. The camera is placed at 90 degrees angle to the mirror holder. The UV light source is placed directly in front of the mirror with the lens in between the light source and the mirror. The beam of light generated by the LED and made parallel by the lens proceeds directly to the semi-reflective mirror. The mirror reflects half of the light down to the hazelnuts and the remaining half of the light proceeds straight towards the light-trap box, which is lined internally with black velvet to absorb all the excess light so that the light does not move back and forth and is not reflected back to the camera. The 50% reflected light travels downward and illuminates the hazelnut samples placed at the bottom of the illuminator.

Images were captured for the pest-infected, rotten and healthy hazelnuts at 450nm and 750nm by means of the band-pass filters. Hazelnuts reflect light in different ways depending on whether they are healthy or contaminated. Healthy hazelnuts produce uniform fluorescence at 450nm, while the rotten hazelnuts appear almost black or nearly darker and do not produce any fluorescence. Pest-infected hazelnuts appear black only at the infected regions. Sample images of pest-infected, rotten and healthy hazelnuts captured at 450nm and 750nm are shown in Figs. 3.7, 3.8 and 3.9 respectively.

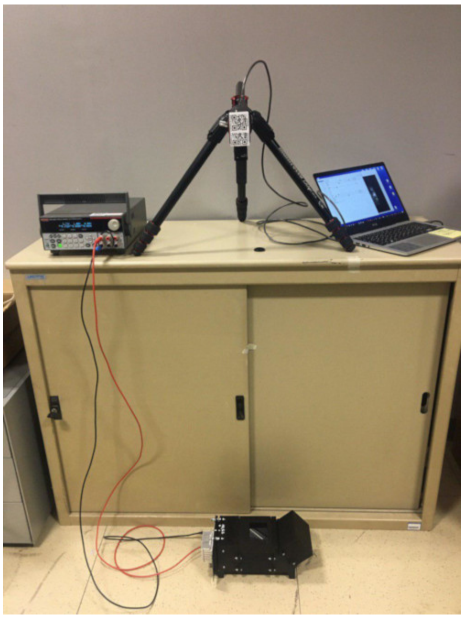


Fig. 3.5 Image Acquisition Setup

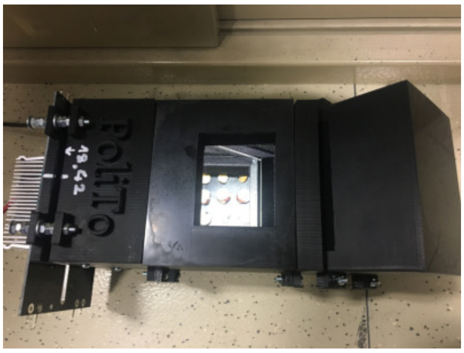
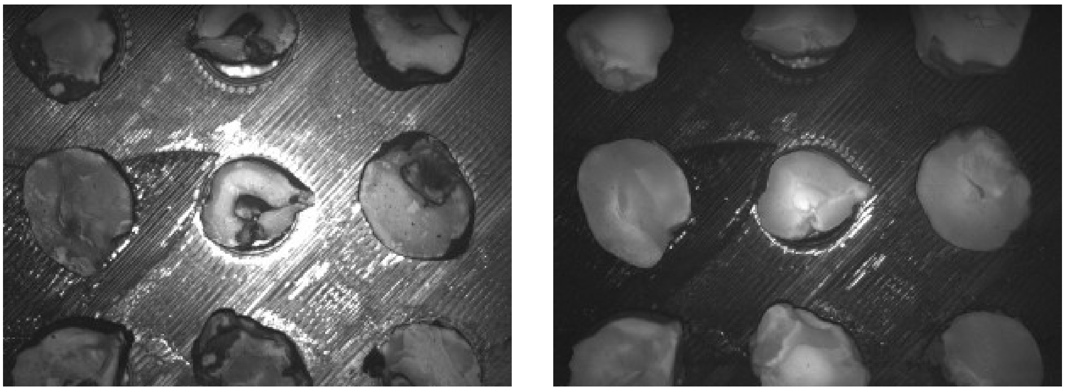


Fig. 3.6 Illuminated Hazelnuts

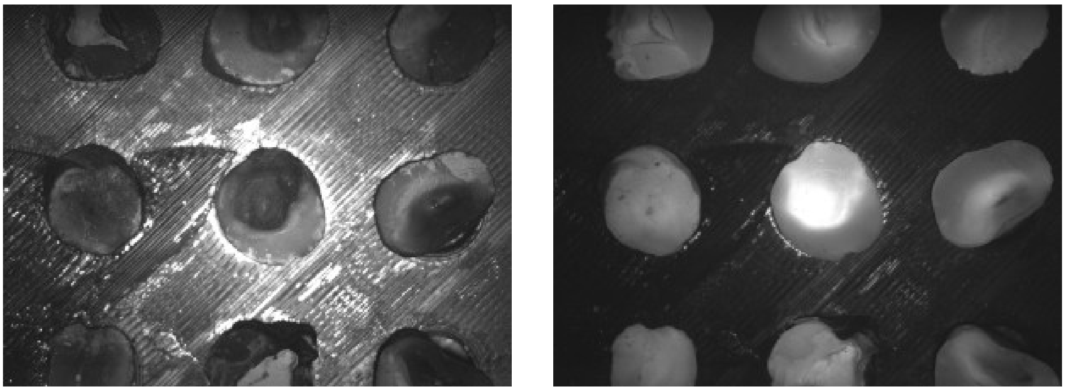


(a) Pest-infected hazelnuts at 450nm

(b) Pest-infected hazelnuts at 750nm

Fig. 3.7 Sample images of pest-infected hazelnuts captured at 450nm and 750nm.

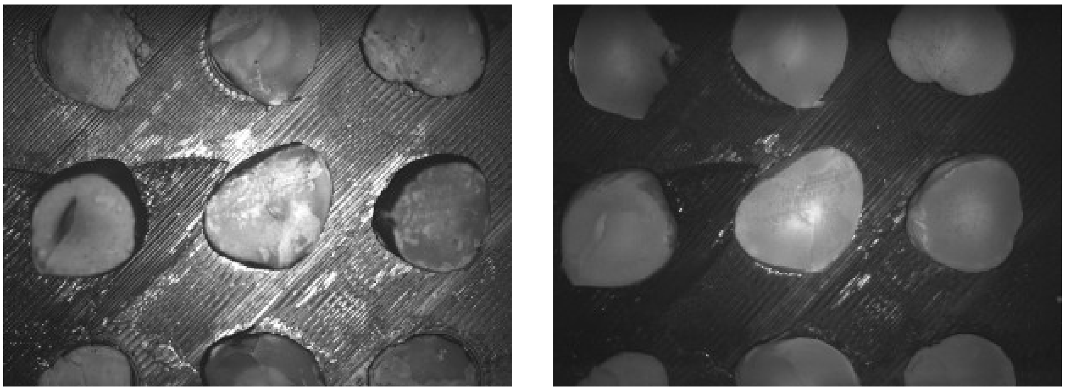
For each of the 21 hazelnut samples in all the three categories, a pair of images was acquired, one at 450nm and the other at 750nm, this makes total 42 images for the pest-infected hazelnuts, 42 images for the rotten hazelnuts and 42 images for the healthy hazelnuts, making 126 images in total. About 9 samples can be captured at a time and this size is dependent on the mirror holder's opening for placing the hazelnut sample tray. Only the central sample is considered for processing for each of the captured image. After image acquisition, the next step is to perform several operations on the acquired images. The image processing was carried out using Fiji



(a) Rotten hazelnuts at 450nm

(b) Rotten hazelnuts at 750nm

Fig. 3.8 Sample images of rotten hazelnuts captured at 450nm and 750nm.



(a) Healthy hazelnuts at 450nm

(b) Healthy hazelnuts at 750nm

Fig. 3.9 Sample images of healthy hazelnuts captured at 450nm and 750nm.

ImageJ (version: 2.0.0) [110] which is an open source image processing software containing many built-in plugins.

### 3.3.4 Image Alignment

The pair of images captured at 450nm and 750nm were not aligned with one another due to the fact that the band-pass filters needed to be changed every time to capture images and this caused some movement in the sample tray. Therefore, both the images need to be aligned to each other so that the hazelnut samples placed at the center should be on top of one another, exactly at the same position, in both the images. In order to align both images, a built-in plugin of ImageJ, named "Linear

Stack Alignment with SIFT" [111], has been used with default parameter values. This step of image alignment was repeated for all the sample images.

### 3.3.5 Combining Images

After aligning all the images, the next step is to combine the pair of images by dividing them with one another to get a new image. The division operation is represented mathematically as shown in Equation (3.1). The division operation was performed in ImageJ [110] by using a plugin named Calculator Plus [112].

$$i = (i1/i2) * k1 + k2 \tag{3.1}$$

where $i1$ is the image captured at 450nm, $i2$ is the image captured at 750nm, $k1$ and $k2$ are the scaling factors to align brightness and contrast in order to highlight the differences between healthy and unhealthy parts of hazelnuts. After performing several experiments by trial and error method, the optimal values of $k1$ and $k2$ were found to be 65535 and 0, respectively. These values are optimal because they result in darkening the pixels of bad parts of hazelnuts and brightening the pixels of good parts of hazelnuts.

### 3.3.6 Image Thresholding

After performing the division operation, the next and final step is to perform image thresholding. The resulting images are already in gray-scale mode, they are converted into binary images in order to clearly represent the differences among the pest-infected, healthy and rotten hazelnuts.

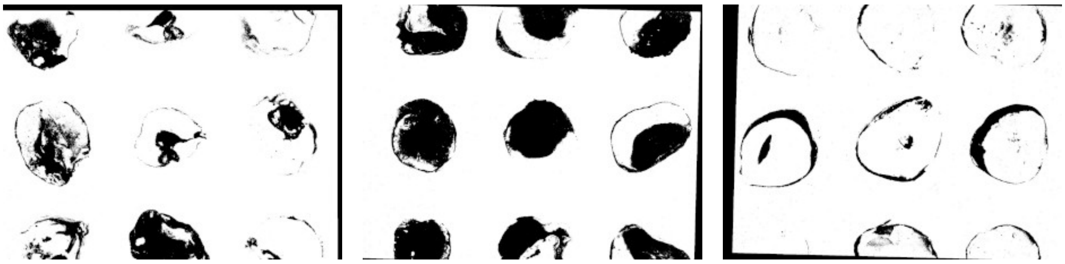
## 3.4 Experimental Results and Discussion

Experiments were conducted on 21 half-cut hazelnut samples from each of the three hazelnut categories under consideration. This section presents and discusses the results after applying the proposed system to the acquired sample hazelnut images.

The images captured under UV illumination were stored in PGM (Portable Gray Map) format. The .PGM extension stores an image in gray-scale format. So the stored images are already in gray-scale form rather than RGB color images. The pre-requisite of applying the proposed operations on the image dataset is that the pair of images must be aligned to one another. Hence, the first operation was the image alignment as described in Section 3.3. After carefully aligning the images, image division was performed on the two images, one captured at 450nm and the other at 750nm. Division of images is very basic yet powerful operation to highlight the differences between contaminated (either rotten or pest-infected) and uncontaminated hazelnuts. The further image thresholding operation converted the gray-scale image obtained after image division operation into binary form. Again, image thresholding is very simple yet effective operation in highlighting the differences among pest-infected, rotten and healthy hazelnuts, where the black pixels represent the contaminated hazelnuts and white pixels represent the healthy hazelnuts.

Fig. 3.10 presents the experimental results for the pest-infected, rotten and healthy hazelnuts in Figs. 3.10a, 3.10b, and 3.10c, respectively. By looking at the experimental results, it is quite clear that the image of rotten hazelnuts contain more black than white pixels, whereas the image of healthy hazelnuts contain more white than black pixels. The pest-infected hazelnuts lie in between the rotten and healthy hazelnuts in terms of number of black and white pixels.

A pest-infected nut is different from a rotten nut in such a way that the former is only contaminated in parts where insects or other micro-organisms have attacked it or eaten it and the rest of the part appears to be good, whereas a rotten nut is almost homogeneously contaminated. Therefore if we look at Fig. 3.10b, the black spots are almost continuous and are larger, and by looking at Fig. 3.10a, the black pixels appear at discontinuous spots and are relatively smaller as compared to the rotten ones, representing the damage caused by pests and insects. The healthy hazelnuts also contain some black pixels, usually around the corner and/or at the center, as can be seen in Fig. 3.10c. This is due to the fact that some of the hazelnuts samples contain a thin brown skin at the outer side which is represented by black pixels after applying the proposed system. The central part of healthy hazelnuts is sometimes represented by black pixels because of the seed itself. Nevertheless, the number of black pixels are much less in healthy hazelnuts as compared to the other two categories of hazelnuts.



(a) Pest-infected hazelnuts (b) Rotten hazelnuts (c) Healthy hazelnuts

Fig. 3.10 Experimental results of the pest-infected, rotten and healthy hazelnuts.

Since all the resulting images contain black as well as white pixels, it is important to calculate the total number of these pixels in order to distinguish the pest-infected, rotten and healthy hazelnuts. For that purpose, we cropped the central hazelnut from each of the resulting images, and then calculated the total number of black and white pixels in the cropped image of that particular hazelnut. The average percentages of black and white pixels have been computed for all the sample images of pest-infected, rotten and healthy hazelnuts. The results are presented in Table 3.1. As can be seen, the rotten hazelnuts contain the maximum number of black pixels i.e. approximately 57.13% on average, whereas the healthy hazelnuts contain the least number of black pixels, i.e. only 12.82% on average. The number of black pixels for the pest-infected hazelnuts lie in between the rotten and the healthy hazelnuts i.e. 30.82% on average. The opposite is true for white pixels. The number of white pixels is maximum for healthy hazelnuts, 87.17% on average. Pest-infected hazelnuts contain 69.16% of white pixels, whereas the rotten hazelnuts have only 42.86% of white pixels.

By looking at these percentages it can be concluded that the rotten hazelnuts contain the maximum number of black pixels whereas the healthy hazelnuts contain the maximum number of white pixels. The pest-infected hazelnuts remain in between these two categories, with more white pixels than black pixels,

The proposed method successfully distinguishes the contaminated hazelnuts from the uncontaminated ones by applying simple yet effective image processing techniques. The contaminated hazelnuts can either belong to the rotten or pest-infected category depending on the cause of their damage. The major part of the success of this work is contributed to the image acquisition phase and to the specialized axial illuminator due to which it was possible to uniformly illuminate the hazelnut samples under UV light. In view of an effective application on an industrial scale of the proposed method, it would be necessary to optimize the model in order

Table 3.1 Average percentages of black and white pixels in rotten, pest-infected, and healthy hazelnuts

| Hazelnut Class | Average % of black pixels | Average % of white pixels |
|---|---|---|
| Rotten | 57.13% | 42.86% |
| Pest-Infected | 30.82% | 69.16% |
| Healthy | 12.82% | 87.17% |

to further improve its robustness and versatility. In this regard, the calibration dataset should include an extended dataset, taking into account different kinds of hazelnuts to generalize the method.

Unfortunately, it is very difficult to compare the results with the state of the art due to differently sized datasets used in every research study and different variety of hazelnuts used in several works. Moreover, the target of each study is not to classify the hazelnuts in the three categories we have considered.

## 3.5 Conclusion

In this chapter we presented the usefulness of the fluorescence imaging technique to identify the contaminated hazelnuts. In particular, the hazelnut samples were successfully classified into pest-infected, rotten, and healthy categories. We also showed the effectiveness of simple image processing techniques to highlight the defects in hazelnuts.

The axial illuminator played an important role in the success of this work. It is due to this illumination technique that we were able to capture the images of hazelnut samples by using ultraviolet radiations. The produced fluorescence actually gave the possibility to identify the contaminated and uncontaminated parts of hazelnuts.

The proposed method can be applied to various food items to automatically identify the defects in them instead of visually inspecting them. However, the proposed system is in initial stage at the moment. It must be calibrated and tested thoroughly to check the effectiveness, first on many different categories of hazelnuts, and then on different food items. We propose to incorporate these features in the future work.

# Chapter 4

# Measuring the Quality of Built-in Smartphone Cameras

Work described in this chapter was originally presented in [113].

## 4.1 Introduction

Almost every other person carries a smartphone nowadays. Some people often carry multiple phones for different purposes. The most important step in buying a smartphone is to check its technical specifications thoroughly. Moreover, the smartphone industry has become too diverse to easily decide about the phone to buy. Many companies are launching newer and upgraded versions of the smartphones by introducing advanced features every now and then.

As we discussed in Chapter 2 that our young generation is largely active on social media where majority of the data is in the form of images and videos. Hence, the smartphone camera quality become one of the most important features to consider by youth as well as adults before buying a smartphone. Camera specifications are often provided as the total number of front and back cameras, total megapixels, aperture size, sensor size, focal length, pixel size, etc. Probably a non-technical person (one who is not an expert in photography or optical devices) would not be aware of these properties or camera specifications.

Moreover, the smartphone manufacturers fail to provide a numerical value to quantify the performance or quality of a phone camera. They do provide a lot of technical details, but unfortunately those technical terms are of nearly no use for a common man having no background in such measures, which makes it very difficult for non-experts to decide about the camera quality and people often end up in making wrong choices about smartphone selection.

Millions of cameras have been sold by the camera manufacturers by treating the megapixel number as the only measure to define the camera quality. The buyers usually consider the total number of cameras and their megapixels to decide about which smartphone/camera to buy. But essentially the quality of a camera does not necessarily depend on these measures.

Traditionally, the quality of a camera is determined by capturing some images and recording video clips of subjects and surroundings and then looking at those images and videos to understand the camera quality. This method is vague and does not provide a sound basis to actually measure the performance of cameras, as it does not guarantee a concrete value. Therefore it would be very useful to have a numerical quantity that alone expresses the optical performance of a camera.

One such quantity to measure the performance of optical systems is known as the Modulation Transfer Function (MTF). MTF is one of the most important parameters to measure the quality of an image. Modulation Transfer Function is a commonly and widely used measure to compare the performance of optical systems or imaging devices. It has been studied by various researchers and developers, and its calculation involves different mathematical operations, as we will see in next sections of this chapter.

Different desktop applications have been developed to calculate the value of the MTF to measure the performance of an image. One such application is provided by Imatest LLC [114], which is a company that produces image quality testing software, equipment and test charts. It is one of the most famous tools to measure the performance of optical devices.

Desktop applications have the limitation that they cannot be directly used on a mobile phone. Hence there is a gap in the existing industrial solutions to measure the performance of phone camera directly on the phone itself. This study was performed to develop a mobile application which can be directly installed on smartphones and their camera quality can easily be measured. Moreover, through this study we want to reach out to the smartphone manufacturers in order to propose to them to include the MTF value in the technical specifications of the smartphones. This will allow the end-users to compare the quality of various phone-cameras using a single quantity.

## Roadmap

The rest of the chapter is organized as follows: Section 4.2 defines the common terms used by smartphone manufacturers to describe the smartphone camera features, provided in the technical specifications of smartphones. Section 4.3 presents the basic concepts of the Modulation Transfer Function and also presents the related studies. Section 4.4 describes the experimental setup and discuss in detail the steps required to compute the MTF. The experimental results are presented and discussed in Section 4.5 along with the comparison with Imatest software. The final words are provided in Section 4.6.

## 4.2 Smartphone Camera Specifications

In this sub-section of the chapter we will define the terms described in the technical specifications for smartphone cameras and will try to understand the purpose of each of these terms.

Table 4.1 lists the camera specifications for six different smartphones, namely, iPhone 11 Pro, iPhone 11, iPhone XS Max, Huawei P20 Pro, Huawei Mate 10 Pro, and JiaYu S3, taken from [115] and [116]. For each of the six smartphones, the table mentions the camera type which is actually the number of rear cameras, the number of pixels in terms of Megapixels (MP), the aperture size of camera, the focal length (represented in millimeters - mm), the sensor size (measured in inches - in), and the pixel size of the camera (measured in micrometers - $\mu$m). As can be seen, all these values vary for different smartphones. (Note: “N/A” in the table refers to Not Applicable and “-” specifies the missing values.) Let’s understand the significance and purpose of each of the technical specifications provided for smartphone cameras.

Table 4.1 Specifications of the cameras of smartphones used in Experiments

| | **iPhone 11 Pro** | **Huawei P20 Pro** | **Huawei Mate 10 Pro** | **iPhone XS Max** | **iPhone 11** | **JiaYu S3** |
|---|---|---|---|---|---|---|
| Camera Type | Triple Camera | Triple Camera | Dual Camera | Dual Camera | Dual Camera | Single Camera |
| **Main Camera** | | | | | | |
| Megapixels (MP) | 12 | 40 | 12 | 12 | 12 | 13 |
| Aperture size | f/1.8 | f/1.8 | f/1.6 | f/1.8 | f/1.8 | f/2 |
| Focal Length (mm) | 26 | 27 | 27 | 26 | 26 | - |
| Sensor Size (in) | 1/2.55" | 1/1.78" | 1/2.9" | 1/2.55" | 1/2.55" | 1/3.06" |
| Pixel Size ($\mu$m) | 1.4 | 1.55 | 1.25 | 1.4 | 1.4 | 1.1 |
| **Second Camera** | | | | | | |
| Megapixels (MP) | 12 | 8 | 20 | 12 | 12 | N/A |
| Aperture size | f/2.0 | f/2.4 | f/1.6 | f/2.4 | f/2.4 | N/A |
| Focal Length (mm) | 52 | 80 | 27 | 52 | 13 | N/A |
| Sensor Size (in) | 1/3.4" | 1/4" | 1/2.7" | 1/3.4" | 1/3.6" | N/A |
| Pixel Size ($\mu$m) | 1 | - | 1 | 1 | 1.4 | N/A |
| **Third Camera** | | | | | | |
| Megapixels (MP) | 12 | 20 | N/A | N/A | N/A | N/A |
| Aperture size | f/2.4 | f/1.6 | N/A | N/A | N/A | N/A |
| Focal Length (mm) | 13 | 27 | N/A | N/A | N/A | N/A |
| Sensor Size (in) | 1/3.6" | 1/2.7" | N/A | N/A | N/A | N/A |
| Pixel Size ($\mu$m) | - | 1 | N/A | N/A | N/A | N/A |

### 4.2.1 Multi-Camera Smartphones

Camera performance is one of the most critical aspects for smartphone users and manufacturers. As the users demand for thin and slim smartphones, it becomes difficult for the manufacturers to provide best quality camera using a single camera in such a compact space. Therefore, the smartphone companies have decided to increase the width and height of smartphones to incorporate multiple cameras for better performance. Most of the smartphones of modern generation have dual and triple cameras. Some smartphones even have more than 3 cameras, for example Xiaomi Redmi 10 has 4 cameras.

### 4.2.2 Megapixels

Camera resolution or the size of an image is often measured in Megapixels (MP). One megapixel is equals to 1 million pixels. This quantity specifies how much detail a camera sensor can capture. The higher the number of megapixels, more detail can be captured within an image. Non technical users often consider the number of megapixels as the quality of smartphone camera. But in reality, this is not the only measure to quantify the performance of a smartphone camera.

### 4.2.3 Aperture size

The aperture of a camera is defined as the opening through which light travels to reach to the camera sensor. In photography, an aperture resembles to the pupil of human eye. It can expand or shrink to allow more or less light to pass to a camera's sensor. The larger the aperture or opening, the more light comes in, resulting in a brighter photo. On the other hand, the smaller aperture size results in a darker photo because less light can enter due to smaller aperture.

Aperture size is represented as "f/number" format. Smaller value of number corresponds to the larger aperture size hence more light and brighter photo, where as a higher number corresponds to the smaller aperture, hence less light and darker photo. Usually the aperture sizes range from f/1.4 (largest) to f/32 (smallest) [117].

### 4.2.4 Focal Length

The focal length of a lens is defined as the measure of distance between the optical center of the lens and the camera's sensor. The optical center of the lens is the point where all the light rays converge to form a sharp image of an object. The focal length is represented in millimeters (mm) and is determined when the lens is focused at infinity.

Lens focal length describes the *angle of view* i.e. how much of a scene before us would be captured by the lens, and the *magnification* i.e. how large the individual objects appear within a frame. The angle of view and the magnification are inversely related to each other which means that the wider the angle of view the lower will be the magnification and vice versa. The shorter focal length results in wider angle of view which in turn results in lower magnification. In other words, a larger portion of the scene can be captured with short focal length. Conversely, the longer focal length results in narrower angle of view and hence captures smaller portion of the scene but with greater magnification [118].

### 4.2.5 Sensor size

The size of a camera sensor, measured along the diagonal, determines the amount of light used by the camera to create an image. The bigger the size of a camera sensor, the better will be the captured photos. Cameras with larger sensors allow to produce high resolution images, i.e. images with fine details.

While large megapixel counts are an increasingly popular trend, it's the size of the camera image sensor that's actually far more important. A sensor is just like a solar panel that gathers light in a digital camera to create an image. Larger camera sensors can gather more light as compared to the smaller sensors, and more light produces better looking pictures. Most smartphone sensors are typically 1/2.55 inches, although some high-end smartphones are having 1/1.7-inch and larger sensors [119].

### 4.2.6 Pixel size

A pixel, also known as a photosite, is the one responsible for capturing or holding the light on a sensor that enters a camera. The bigger the size of a pixel, more light it can capture, resulting in brighter and better images. In other words, the larger pixels results in well-exposed images even in poor lighting conditions. The size of a pixel is measured in micrometers ($\mu$m) [120].

Phone cameras are very compact and much smaller as compared to the digital cameras. It is quite obvious that the overall size of a digital camera is greater than a phone camera. Hence, a digital camera can accommodate larger aperture, bigger sensors and pixels. Therefore the photo quality of a digital camera is much higher than that of a phone camera. Pixel size plays a very important role in smartphone cameras. It is quite difficult to incorporate larger sensors in phone cameras, therefore by increasing the pixel size, image quality can be enhanced greatly in compact cameras of smartphones. Smaller pixels often result in noisy images along with under-exposed photos.

## 4.3 Background and Related Work

In this section we will study the basic concepts of the Modulation Transfer Function (MTF) and will describe different ways to measure it. We will also present literature review on the techniques to measure the quality of optical devices.

### 4.3.1 Modulation Transfer Function (MTF)

The MTF is a commonly used measure to compare the performance of optical systems. Image resolution and contrast are the components of MTF. Resolution is the ability of an imaging system to distinguish object details. High-resolution images exhibit a large amount of detail and minimal blurring. Low-resolution images, on the other hand, are devoid of fine details. The degree to which the minimum and maximum intensity values are faithfully transferred from the object plane to the image plane is referred to as contrast or modulation of an imaging system. Contrast ratio of an image is defined as the ratio of the luminance or brightness of the brightest pixel to the darkest pixel in an image. Mathematically it is represented as follows:

$$CR = \frac{I_{max} - I_{min}}{I_{max} + I_{min}} \tag{4.1}$$

where $CR$ is the Contrast Ratio, $I_{max}$ and $I_{min}$ is the luminance of the maximum and the minimum intensity level respectively.

The Modulation Transfer Function (MTF) of a lens is a measurement of its ability to transfer contrast at a particular resolution from the object to the image. In other words, MTF is a way to combine resolution and contrast in a single entity. MTF is one of the best tools available to quantify a system's imaging performance in terms of resolution and contrast [121].

The impulse response of a system is called Point Spread Function (PSF), i.e. the response of an optical system to a point source. The Fourier Transform of PSF is known as the Optical Transfer Function (OTF). The modulus or magnitude of the OTF is referred to as the Modulation Transfer Function (MTF). It defines the ability of an optical system to resolve a contrast at a given resolution (or spatial frequency) [122]. The MTF is a mathematical expression that describes the spatial resolution of an imaging device, and it is crucial to quantify its performance.

The measurement of MTF requires specially designed targets to measure the performance of optical devices. The imaging device under test captures the images of the chosen special target, also known as test charts. Usually an optical system results in blurred edges in the test charts and the amount of blurriness is represented as a transfer function in frequency domain [123].

### 4.3.2 Target Patterns / Test Charts

Several different types of target patterns or test charts have been used in the literature for measuring the Modulation Transfer Function. Some of them are: Siemens Star target [124], Dead Leaves target [125], Random target [126, 127], Three and Four-Bar targets [128], Sine-Wave target [122, 129], Grill (bar / square) pattern [129], and Slanted-Edge target [130]. Among all the other target patterns, the slanted-edge target is widely used due to its simplicity and reliability. In fact, it is the least computational method for measuring MTF as compared to the other target patterns [129]. A relatively higher number of studies have used the slanted-edge target method or a slight modification of the slanted-edge method for computing the MTF, as can be found in Refs. [122, 129, 131–141].

### 4.3.3 Slanted-Edge Target

The slanted-edge target consists of roughly equal amounts of dark and bright portions separated by a sharp edge tilted by a small angle as shown in Fig. 4.1. The two important characteristics of the slanted-edge target are i) contrast ratio and ii) edge angle. A study performed by J. Roland [131] measured the impact of different contrast ratios and edge angles of the slanted-edge target on MTF50 values. The MTF50 values remained highly stable for several contrast ratios (with the lowest being 1.10:1 and the highest being 21.0:1) in the absence of noise. However, for noisy images, the MTF values dropped significantly for the lowest contrast image. The remaining contrasts showed similar values of MTF50 as that in the absence of noise [131]. From that experiment, it can be deduced that the contrast has no effect on the MTF unless significant noise is present. In other words, the MTF values would generally be stable for high contrast images regardless of the noise. The study also showed that the slanted-edge angle greatly affects the MTF values. The greater the edge angle, the lower will be the MTF value as shown in Fig. 4.2 [131]. It is interesting to note that the angles from 2° to 10° represent stable results for MTF. In other words, the value of MTF decreases significantly for edge angles greater than 10°.

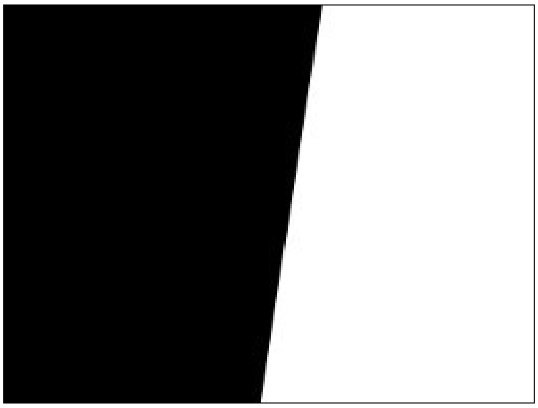


Fig. 4.1 Slanted-Edge Target

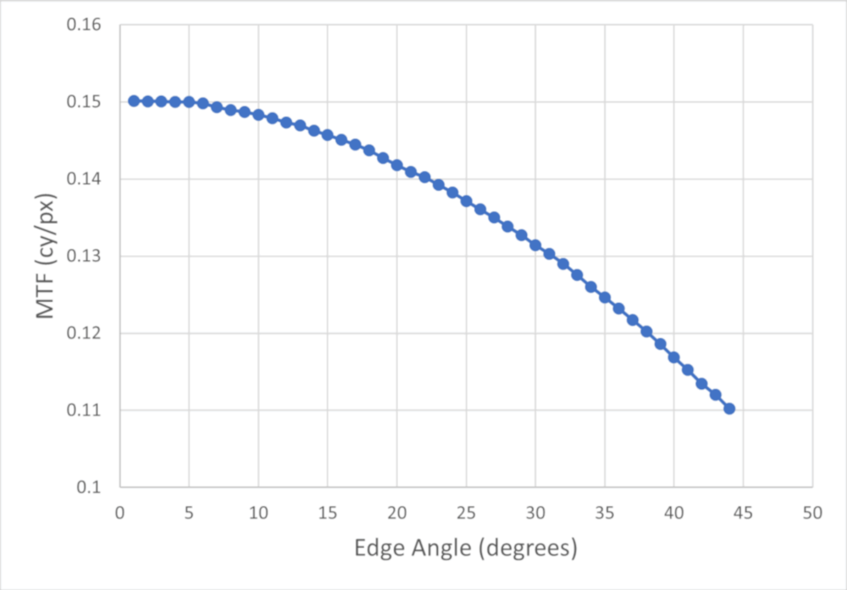


Fig. 4.2 Dependence of MTF values on various edge angles [131]

### 4.3.4 Slanted-Edge Technique

The slanted-edge technique is an ISO 12233 standard to measure the Modulation Transfer Function. This is also known as "tilted knife edge technique", because the target consists of a sharp edge tilted by a small angle.

The computation of the MTF using the slanted-edge target can be described in simple words as follows: first, select the region of interest (ROI) in the target image, then determine the exact angle of the edge, then compute the distance of individual pixels to the edge. This will construct a super-sampled Edge Spread Function (ESF), i.e the response of the imaging device to an edge. Differentiation of the ESF generates a Line Spread Function (LSF) and it is mathematically expressed as shown in Equation (4.2). The Fourier Transform of LSF produces the Modulation Transfer Function (MTF) [133, 135]. The mathematical expression for the MTF is expressed in Equation (4.3). This method of computing MTF is adapted in the ISO standard for digital camera resolution and is known as ISO 12233 standard [130]. The major advantage of this technique is that it offers a quick and efficient measurement of the MTF.

$$LSF(x) = \frac{d}{dx} ESF(x) \tag{4.2}$$

$$MTF(v) = \mid \mathscr{F}\{LSF(x)\} \mid \tag{4.3}$$

### 4.3.5 MTF50 and MTF50P

While MTF provides the spatial frequency response of a system, some studies, for example [142], show that the better indicators to measure image sharpness are the spatial frequencies where MTF is 50% of its low-frequency value, known as MTF50, or 50% of its peak value, called MTF50P. Both MTF50 and MTF50P are recommended for comparing the sharpness of different cameras and lenses [142]. The MTF of an optical system is visually represented as a continuous signal. Therefore, it is quite difficult to compare the camera quality of different smartphones by looking at the MTF curves. However, the MTF50 is a numerical value, and therefore, it is easier to compare the performance of imaging devices by simple looking at the MTF50 values for different devices.

The work presented in the chapter discusses the development of a mobile application, MTFCalculator [143], as we named it, to compute the MTF50 values of built-in cameras in smartphones. The app computes the MTF by using the slanted-edge target, which is an ISO 12233 standard to assess the quality of image capturing devices. One of the most interesting features of our app is its ranking ability, i.e. it keeps a record of the top 5 smartphones with the highest camera quality. This list is updated whenever a mobile device is tested for its camera performance. The results of the MTFCalculator are compared to the results provided by Imatest software (version 2020.2) [114], one of the most important tools for testing the quality of an image capture technology.

## 4.4 Methodology

This section provides the details of the experimental setup and describes the algorithm used to compute the Modulation Transfer Function.

### 4.4.1 Experimental Setup

The ISO standard 12233 slanted-edge target has been used in our experiments for the computation of MTF. As discussed in Section 3.1, the MTF value depends on the target's edge angle. The greater the edge angle, the lower will be the MTF value and vice versa. Smaller MTF values are associated with low quality optical devices. In other words, the quality of an optical device decreases with decreasing MTF values. It was noticed in the study performed by J. Roland [131] that the MTF values show stability for the edge angles from 2° to 10°. Angles greater than 10° correspond to decreasing MTF values. In other words, the MTF values decrease rapidly for the edge angles greater than 10°. It is advisable to have an edge tilted by minimum 2° and maximum 10° to get reliable results for computing the MTF [144]. We have used the slanted-edge target with an edge tilt angle of 9° in our experiments, which can be considered as an acceptable choice for computing the MTF value, as the selected angle lies between the stable range for computing the MTF value. The choice of edge angle of 9° was made to avoid computing the MTF values in ideal case of getting the maximum values with lower angles such as 1° or 2° etc. The contrast ratio, i.e. the luminance of the brightest pixel to the darkest pixel of the target image is 21:1. The size of the target image is 2560x1920 pixels.

The target image needs to be present physically so that its image can be captured by the camera of the device to be observed. For that purpose, the target image was printed on an A4 paper using a good-quality printer. It is important to note that the quality of printer and paper greatly affects the results of the MTF values. A low-quality printer or a low-quality paper would add noise to the printed target image and the added noise may decrease the value of MTF, resulting in declaring the phone camera as of lower quality. The better the print quality of the target, the higher will be the MTF and vice versa. The MTF value is negatively affected by the presence of noise. Therefore it is recommended to use a high-quality printer as well as a good quality paper to avoid any noise to the best possible level. Of course, if the quality of the printed target image itself was not good, the captured image would not be optimal and hence the image capturing device would be regarded as of low quality due to lower values of MTF.

The slanted-edge target must be uniformly illuminated to capture good quality images. This was done by using a lamp, which was placed above the target so that the light falls directly on the A4 paper containing the slanted-edge target image. By uniforming illuminating the target, we make sure to have well-exposed and bright image of the slanted edge target. By doing this we avoid the possibility of getting dark image consisting of noise which may decrease the MTF values, resulting in declaring the smartphone camera as of low quality. The camera must be placed perpendicular to the target. It is best to fix the smartphone using some kind of stand to prevent the image from motion blur and shaking. The distance between the target and the camera depends on the size of the target image and the sensor and pixel size of the smartphone camera because our goal is to place the camera in such a way that the target covers the entire screen of the smartphone without any background. For example, Huawei Mate 10 Pro, having a display size of 6 inches, was placed approximately 18cm above the target image.

We have used six different smartphones to assess the quality of their built-in cameras by means of their MTF values, including Huawei Mate 10 Pro, Huawei P20 Pro, iPhone XS Max, iPhone 11, iPhone 11 Pro, and JiaYu S3.

### 4.4.2 The MTFCalculator Application

We have developed a mobile application, which we named as MTFCalculator, to measure the MTF value in order to assess the quality of the phone camera. The application is currently available for android phones. A user needs to install the application to measure the quality of his/her phone camera. The slanted edge target image is provided in the application itself. The user needs to print the image on an A4 paper and then capture the photo using the phone's camera.

As the user starts the application, a welcome screen appears with a message stating, "Welcome to MTF Calculator" as shown in Fig: 4.3a. There is a button at the bottom of the screen, "Go to Menu". When the user clicks at it, the main menu screen appears as shown in Fig. 4.3b. Five buttons are presented on the screen, whose purpose is described as follows:

- **My Scores:** Keeps the history of user measurements with the relative MTF50 results obtained. A sample My Scores window is shown in Fig: 4.3d.
- **Shoot Guide:** Provides a link for downloading the slanted edge target image and also the instructions for using the application, guidelines on lighting and

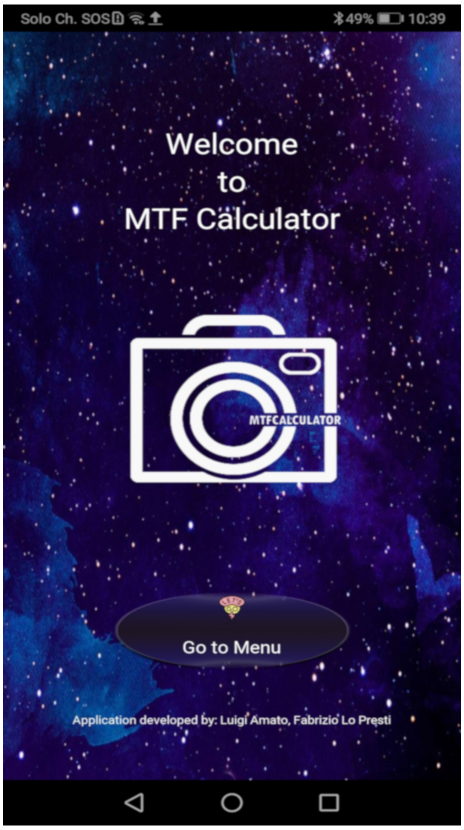
positioning with respect to the target. A sample screenshot of the Shoot Guide window is shown in Fig: 4.3e.

- **Top 5:** Keeps an up-to-date ranking of five devices that have achieved the best results with this application. The ranking will not be directly modified by the score of a user who manages to fall within the top 5 ( since the database measurements are carried out in ideal boundary conditions by the application development team), but if the user manages to get into the Top 5, his best score will be displayed in this window.

- **FAQ:** This section contains the Frequently Asked Question about the technical details on the concepts on which the application is based (definition of MTF, MTF50 etc.), and some explanations on the variability of results and boundary conditions.

- **New Shoot:** When a user clicks it, the camera automatically opens in landscape mode. The user is asked to adjust the target to capture an image. There is an overlaid image to guide the user to place the target according to the guideline so that the slanted-edge is aligned exactly in the center of the screen and the target covers the entire screen of the smartphone, as depicted in Fig. 4.3c. Extra care must be taken to align the target in the camera so that there is only the target and nothing else in the background is visible.

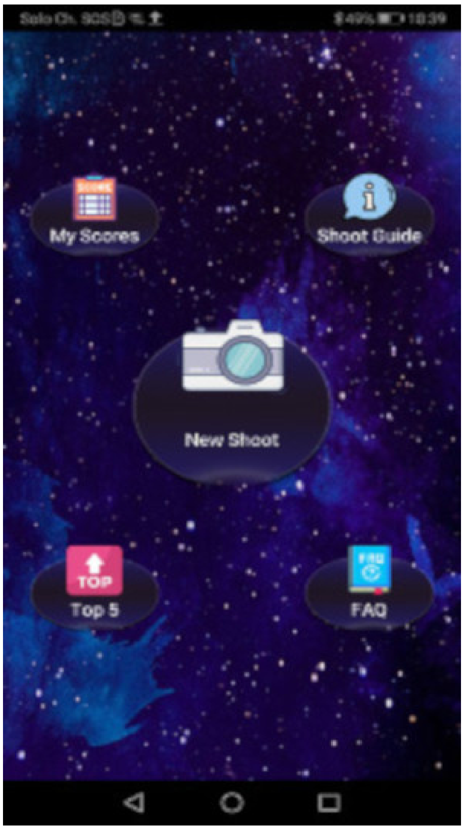
However, it is not always possible to work under the ideal conditions in the real world. Some possible limitations might occur such as unavailability of a high-quality printer, a poor-quality paper, printing low-quality target image, non-uniform illumination or poor lighting conditions (which result in darker image), some possible human error while aligning the target and capturing the photo, etc. In order to mitigate these non-ideal environmental conditions, the MTFCalculator application asks the user to capture three shots of the target image and computes the MTF50 value for each image, and finally calculates the average of the three values as the resulting MTF50 value. A counter is also implemented in the application to count and display the number of captured images. The application requires minimal user interaction i.e. the user is only required at the beginning to set the target and capture the image. After that, the application performs all the calculations on its own without requiring user intervention and provides the results at the end, which helps the user to determine his/her camera quality.



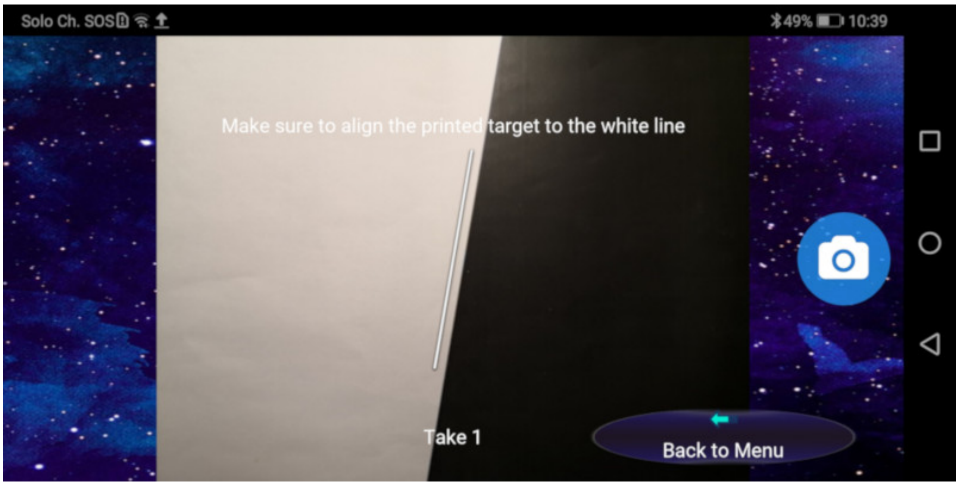
Welcome to MTF Calculator

Application developed by: Luigi Amato, Fabrizio Lo Presti

(a) Welcome to MTF Calculator - User Interface

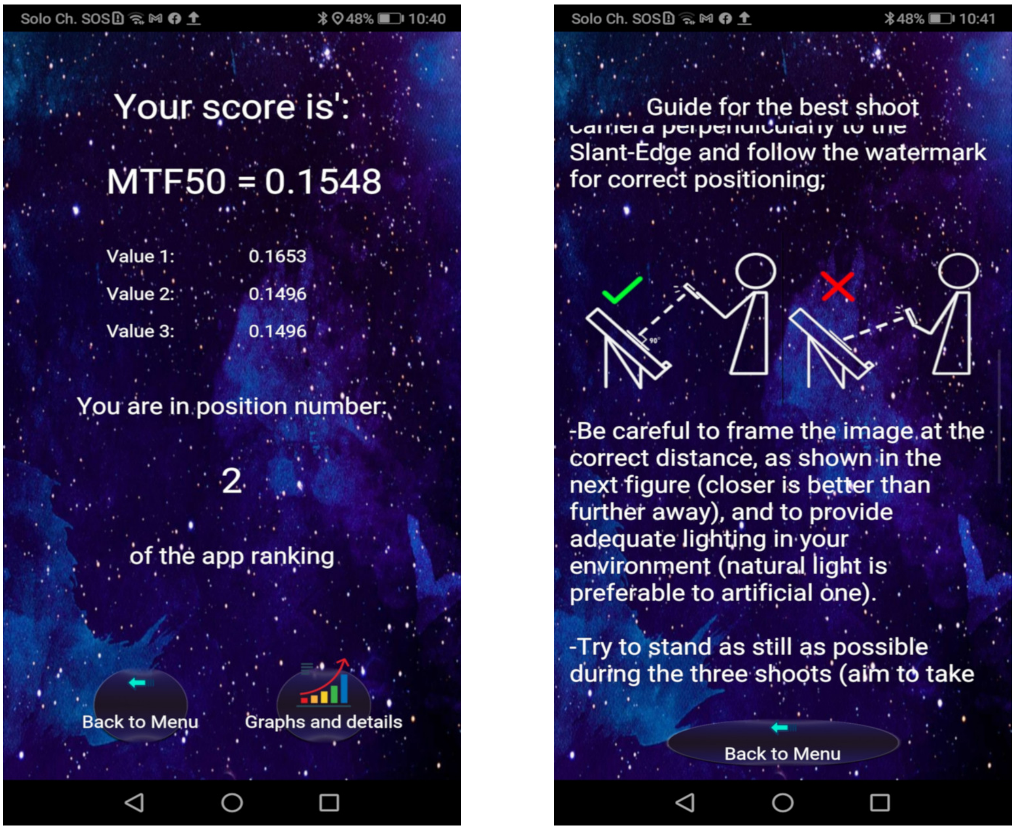


(b) MTFCalculator Main Menu - User Interface



Make sure to align the printed target to the white line

(c) MTFCalculator - Target alignment guideline



Your score is':

MTF50 = 0.1548

| | |
|---|---|
| Value 1: | 0.1653 |
| Value 2: | 0.1496 |
| Value 3: | 0.1496 |

You are in position number:

2

of the app ranking

Guide for the best shoot

camera perpendicularly to the Slant-Edge and follow the watermark for correct positioning;

-Be careful to frame the image at the correct distance, as shown in the next figure (closer is better than further away), and to provide adequate lighting in your environment (natural light is preferable to artificial one).

-Try to stand as still as possible during the three shoots (aim to take

(d) My scores window

(e) Shoot Guide Window

Fig. 4.3 MTFCalculator - Some screenshots of different windows of the application

### 4.4.3 Computation of the MTF value

We will now discuss the algorithm for computing the MTF of built-in cameras in smartphones. Fig. 4.4 shows the step-by-step procedure for the MTF computation using slanted-edge target. The details of the flow diagram are provided in the following sub-sections.

#### ROI Selection

The first step in the computation of MTF, after capturing the target image, is the selection of Region of Interest (ROI). A static ROI, of size 1920x1080 px, is selected starting from the center of the image, as shown in Fig. 4.5. The size of ROI is chosen arbitrarily. It is possible to select the ROI of different sizes. The only important thing to consider while selecting the ROI is to make sure that it contains nearly equal

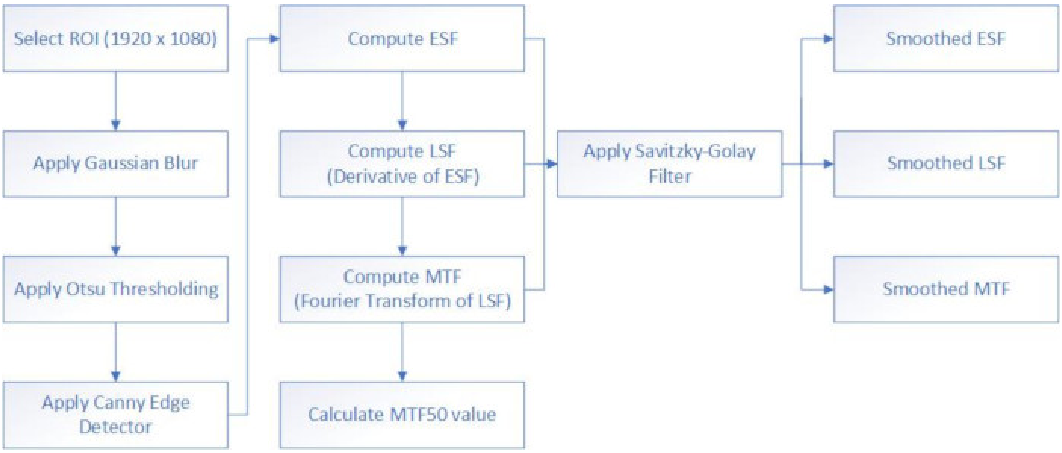


Fig. 4.4 Flow diagram for computing the Modulation Transfer Function using an ISO 12233 standard Slanted-Edge Target.

portions of black and white pixels and that the slanted-edge should be in the center of the selected ROI.

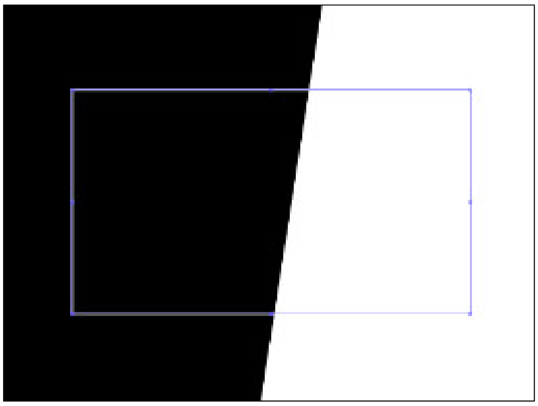


Fig. 4.5 ROI Selection

### Gaussian Blur

It is possible that the captured image of the target may contain some noise due to various reasons, as already discussed. It is essential to reduce the noise for determining correct MTF values. After selecting the region of interest, it was smoothed with the Gaussian blur operator to reduce the amount of noise and speckles within the captured image. It is important to remove high-frequency components, otherwise, these can cause false edge detection [145]. This was done by using

Gaussian Blur, also known as Gaussian Smoothing. Gaussian blur is a type of low pass filter that smoothes uneven pixel values in an image by cutting out the extreme outliers. If a photo is taken in low light, and it appears as under-exposed and has lot of noise, the Gaussian blur can reduce the noise and produces more uniform image. Images may contain noise due to random variation of brightness or hue among pixels. These image pixels have a high standard deviation, i.e. the groups of pixels exhibit a lot of variation. Gaussian blur is obtained by convolving an image with Gaussian kernel. The 2D form of the Gaussian kernel is expressed as [146]:

$$G_{2D} = \frac{1}{2\pi\sigma^2} \exp^{-\frac{x^2+y^2}{2\sigma^2}} \tag{4.4}$$

where $x$ and $y$ are the pixel location indices within an image, and $\sigma$ is the standard deviation of the distribution. To apply the kernel to the current pixel, an average of the the color values of the pixels surrounding it is calculated, weighted by the values in the kernel. In a Gaussian blur, the pixels nearest the center of the kernel are given more weight than those far away from the center. This averaging is done on a channel-by-channel basis, and the average channel values become the new value for the pixel in the filtered image. Larger kernels have more values factored into the average, and this implies that a larger kernel will blur the image more than a smaller kernel [147].

### Otsu Thresholding

Image thresholding is a type of image segmentation where a gray scale image is converted to a binary image, consisting of only black and white pixels, for better analysis of the image.

The next step is to threshold the image with a certain threshold value so that only the major edge, separating the black and white portions of the target image, is detected. Instead of manually choosing a threshold value, an automatic algorithm of thresholding, called Otsu's method [148], was used. Otsu Binarization is an automatic image thresholding method that returns a single intensity threshold that separates pixels into foreground and background classes. This threshold is calculated by minimizing intra-class intensity variance (or within-class variance), or alternatively, by maximizing inter-class variance (or between-class variance) [149]. The within-class variance means the intensity variance or the standard deviation of pixels inside a class. The within-class variance must be minimum, so that we can be sure that the pixels appearing within one class are similar to one another. Where as the between-class variance, i.e. the pixel intensity variance between two different classes, much be maximum, in order to make sure that the two classes have different group of pixels.

Once the thresholding was performed, the minimum and maximum intensity values were saved, along with the information about the area above the threshold and the area below the threshold, in order to pass this data to the next computations.

### Canny Edge Detector

Finally, after performing all the pre-processing steps, the Canny Edge Detection operator was applied to clearly identify the slanted-edge in the target image. Canny Edge Detector is a multi-step edge detection operator to find various edges present in an image [150].

An edge in an image is the sudden change of pixel intensity values. The two operations described above, namely, the Gaussian blur and the Otsu thresholding, are actually the pre-requisites of applying the Canny Edge Detection operation. The output of applying the Canny Edge Detector on the target image is that the edge separating the black and white portions of the target image gets highlighted, which is required by the next steps of the methodology.

### ESF, LSF, MTF

The three major steps in the implementation of the ISO 12233 slanted-edge target method to find MTF values, are the computation of ESF (Edge Spread Function), LSF (Line Spread Function), and MTF (Modulation Transfer Function).

After detecting the slanted edge in the target image, the next step is to compute ESF. The Edge Spread Function is the response of a system to a high contrast edge. The method of constructing the Edge Spread Function (ESF) from the slanted edge image is explained in detail in [151] and [152]. The edge is detected in the target image as described above. A line perpendicular to the edge is constructed. For each row of the image, each point around the edge transition is projected onto the perpendicular line. The process is then repeated for each row of the image. The distance of each point to the edge is calculated.

ESF is useful for extracting essential shape characteristics: each row of the sensor will give a different ESF, and a value called Spatial Frequency Response (SFR) will be computed by checking the pixel values in one column that crosses the slanted-edge. The oversampled ESF is obtained by interleaving the intensity values of each row of the ROI. 2-dimensional image intensity values $i(x,y)$ are projected onto a one-dimensional representation $i(z)$ for constructing the ESF. The magnitude of $z$ represents the shortest distance from a pixel at coordinates $(x,y)$ to the slanted edge. For each pixel the value z is calculated as follows [152]:

$$z = [x - e(y)] * cos(\theta) \tag{4.5}$$

where $e(y)$ denotes the location of the edge in row $y$ and $\theta$ denotes the relative edge angle. The cosine factor transforms a distance measured along a row into a distance measured perpendicularly to the edge [152].

The visual demonstration of the ESF sampling is shown in Fig. 4.6. For each row and each column of the ROI, the pixel values are plotted according to their intensity values. At first, the pixels are black therefore they are plotted as the lowest points, as it reaches the edge, there is a transition from black to white and the pixels are plotted from lower to higher positions. The sample points of the oversampled ESF are then uniformly spaced to make it suitable for the next step.

Once the ESF has been resampled, the next step is to compute the Line Spread Function, which is the system's response to a high contrast line. It is computed by taking the derivative of the ESF, by using (4.2). Finally, the MTF is calculated by finding the Fourier Transform of the LSF, using (4.3).

### Savitzky-Golay Filter

The obtained ESF, LSF, and MTF signals were smoothed by using the Savitzky-Golay filter [153], which is a digital filter for smoothing the data to increase the precision of the data without distorting the signal tendency.

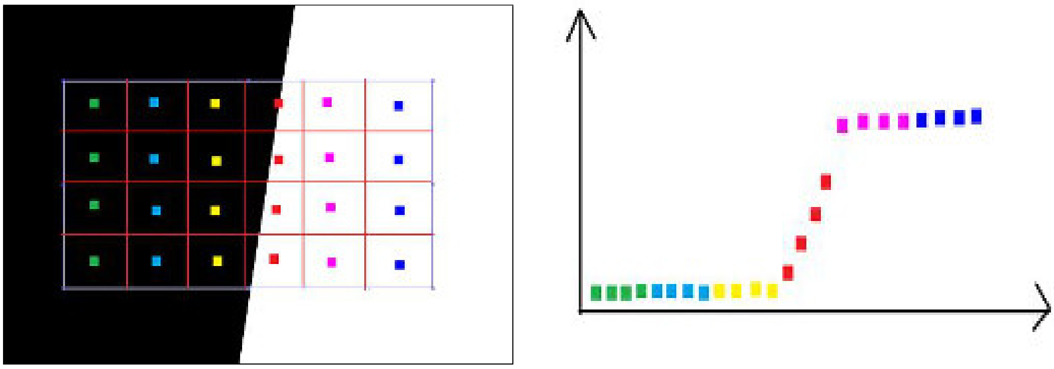


Fig. 4.6 Concept of the ESF sampling from the slanted-edge target’s ROI.

### MTF50 value

As mentioned earlier, the better indicator of image sharpness is the spatial frequency where MTF is 50% of its low-frequency value, known as MTF50. Moreover, the results of ESF, LSF, and MTF are visually represented in the form of curves, which are difficult to use for making comparisons. We need one numerical value to compare the quality of different image capturing devices. Therefore, the final step is to find the MTF50 value, which will be the final metric to compare the performances of built-in cameras of several smartphones. To do so, a specific neighborhood around the closest 50% value of contrast has been extracted, and interpolation on the x values has been performed, obtaining the MTF50 value.

## 4.4.4 Development of MTFCalculator App

An open-source python program for the computation of MTF is available on GitHub [154], which has been used as a base for the development of the MTFCalculator [143] mobile application. The major contributions in the development of our app are as follows: At first, the desktop application was modified to allow its porting to mobile devices. The ROI selection function was modified from a user-defined ROI to a fixed-size ROI for our purpose. The Gaussian blur function was implemented before applying the thresholding. Finally, the computation of the MTF50 value was implemented, which was not present in [154]. Some code snippets for the calculation of ESF, LSF, MTF and MTF50 are available in Appendix B.3, B.4, and B.5, respectively.

## 4.5 Experimental Results and Discussion

In this section we will present the experimental results and will also discuss these results thoroughly.

In the presented study we have used six different smartphones, namely, Huawei Mate 10 Pro, Huawei P20 Pro, iPhone XS Max, iPhone 11, iPhone 11 Pro, and JiaYu S3 to test and compare the performance of their built-in cameras. Both the android and iOS based mobile phones have been used in this study to test the applicability and performance of the developed software. For the android phones, the mobile application was directly installed on the smartphones, while for the iOS-based phones, the image was captured by iPhones and then the calculation was performed on the desktop version of the application.

The experimental results were compared with those obtained by the Imatest software, version 2020.2 [114], which is one of the most popular tools in the optical and imaging industry. Imatest software also provides test charts to measure the characteristics and quality of cameras and imaging systems. Imatest assesses the quality of image capturing devices in terms of MTF50 values. According to the Imatest documentation [155], the best indicators of image sharpness are the spatial frequencies where MTF is 50% of its low frequency value (MTF50) or 50% of its peak value (MTF50P).

As an example, we will present the results obtained by Huawei Mate 10 Pro. If we refer to the Table 4.1, we know that Huawei Mate 10 Pro is a dual camera phone, having the main camera of 12 MP, aperture size of f/1.6, focal length of 27mm, sensor size of 1/2.9" and pixel size of 1.25 $\mu$m. With all these technical details and features, a non-technical person would not be able to clearly understand the camera quality. It is therefore necessary to represent the quality of imaging device as a single numerical value. That is the significance of calculating the MTF50 values.

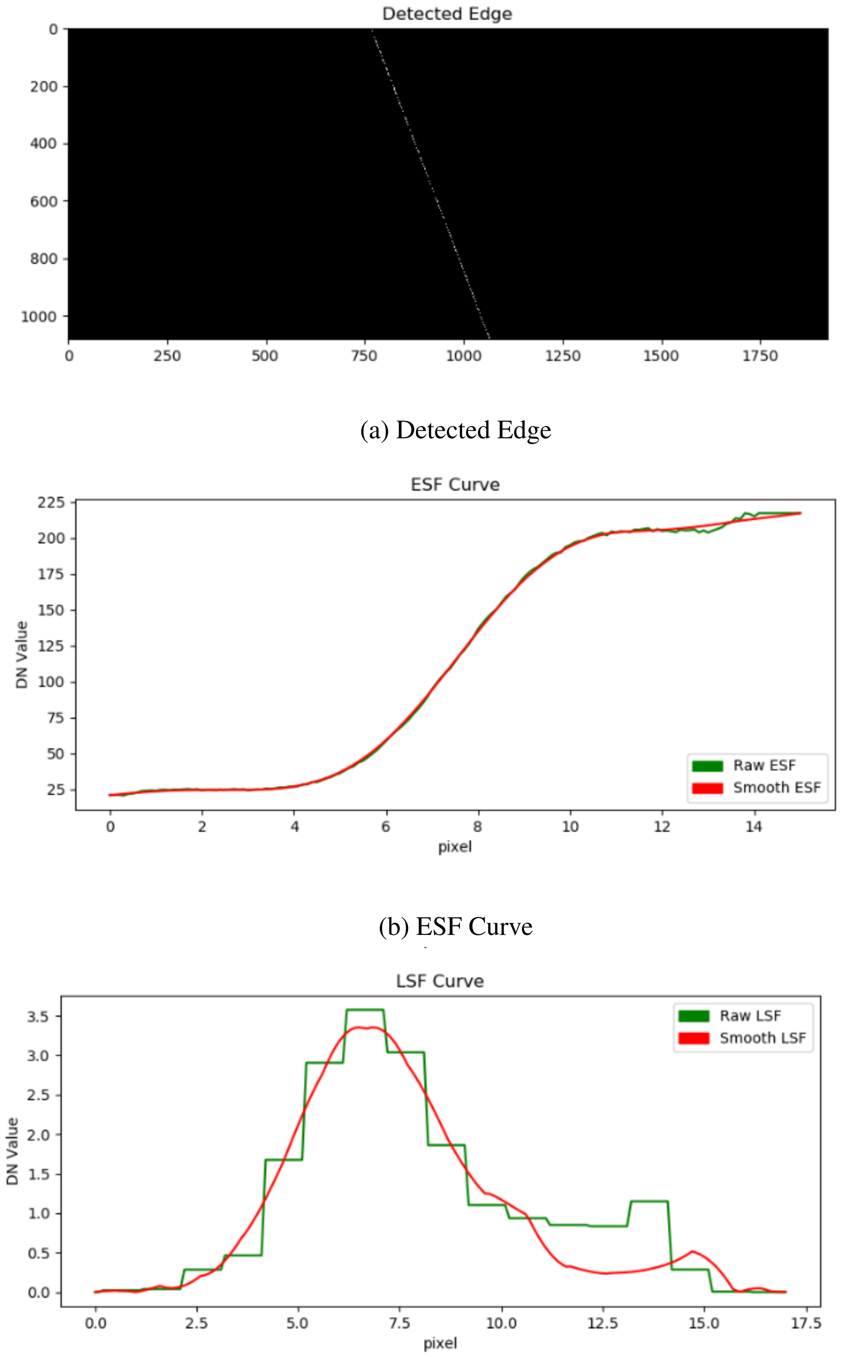
Fig. 4.7 shows the ESF, LSF, and MTF curves produced for Huawei Mate 10 Pro using our developed application, MTFCalculator. Fig. 4.7a shows the detected edge after applying all the pre-processing steps, including ROI selection, Gaussian blur, and Otsu thresholding, along with the Canny Edge Detector. As can be seen, the slanted-edge is clearly visible in the middle and every other pixel is 0, i.e. black in color, where as the detected edge is represented as a thin white line. The ESF curve is depicted in Fig. 4.7b, the oversampled ESF was obtained by interleaving the intensity values of each row of the selected region of interest. Once the ESF was obtained, it was differentiated to get the LSF curve, as illustrated in Fig. 4.7c. Finally, Fig. 4.7d shows the MTF curve, which was obtained by taking the Fourier Transform of the LSF. For the ESF, LSF, and MTF curves, both the raw and smooth curves are plotted. As already mentioned, the raw curves were smoothed by using the Savitzky-Golay filter. The MTF50 value (i.e. the spatial frequency where MTF is 50% of its low-frequency value) for the smartphone under test was found to be 0.1338. Fig.4.8 shows a sample result page of the MTFCalculator application after computing the average MTF50 value and also provides the ranking of the smartphone in terms of its camera quality.

To compare the results obtained by MTFCalculator with the Imatest, we performed the experiments using both the applications under identical conditions. We used the same slanted edge target image as input to both applications. The tests were conducted on all the six smartphones under discussion using both the applications. We needed to make a little modification in our application to make the calculations identical to the Imatest software i.e. Imatest uses the ROI size of 1432x1080 pixels, where as the ROI size in MTFCalculator was defined as 1920x1080 pixels. We modified the size of our ROI to match with that of the Imatest.

Imatest is actually a desktop application, so we installed it on our personal computer, and also installed the desktop version of the MTFCalculator. We then captured the images of the slanted-edge target using each of the above-mentioned smartphones, provided those images as input to both the applications, and computed the MTF50 values for each of the captured images. Table 4.2 shows the MTF50 values values computed for the six smartphones using both the applications.

Fig. 4.9 compares the resulting MTF50 values for all the smartphones computed using MTFCalculator and Imatest [114] using graphical illustration. As can be seen, there is only a marginal difference between the results of the two applications. For all the smartphones, except iPhone 11 Pro, the MTF50 values obtained through MTFCalculator are slightly greater than those achieved by Imatest. For iPhone 11 Pro, the MTF50 value by MTFCalculator is slightly lower than that of the Imatest.

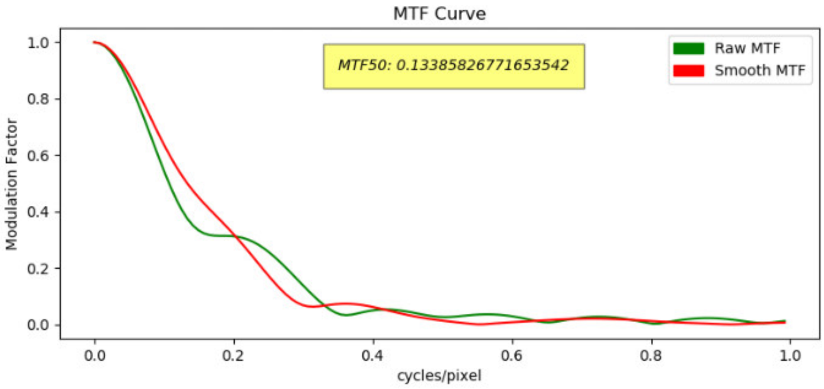
It is interesting to note that the ranking results of both the applications are exactly the same, i.e. arranging, in descending order, the smartphones according to their MTF50 values, results in the same order for both the MTFCalculator and the Imatest. In the ranking order, the top phone in the list is the iPhone 11 Pro having values



(a) Detected Edge

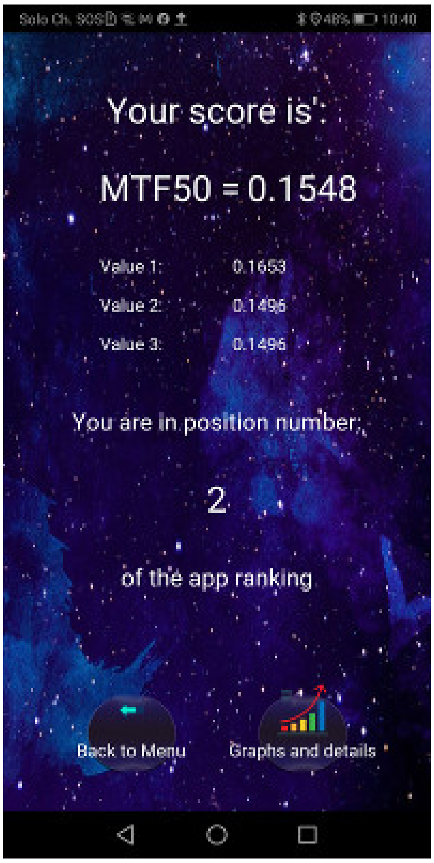
(b) ESF Curve

(c) LSF Curve



(d) MTF Curve

Fig. 4.7 Experimental results obtained by MTFCalculator for Huawei Mate 10 Pro.



Your score is:

MTF50 = 0.1548

Value 1: 0.1653
Value 2: 0.1496
Value 3: 0.1496

You are in position number:

2

of the app ranking

Fig. 4.8 MTFCalculator - MTF50 value and ranking result

Table 4.2 MTF50 values using MTFCalculator and Imatest for the smartphones under test

| **Smartphone** | **MTFCalculator** | **Imatest** |
|---|---|---|
| iPhone 11 Pro | 0.1717 | 0.1749 |
| JiaYu S3 | 0.1460 | 0.1399 |
| iPhone 11 | 0.1417 | 0.1312 |
| Huawei Mate 10 Pro | 0.1338 | 0.1306 |
| Huawei P20 Pro | 0.1259 | 0.1123 |
| iPhone XS Max | 0.1140 | 0.1088 |

of 0.1717 and 0.1749, obtained by MTFCalculator and Imatest, respectively. After iPhone 11 Pro, the next smartphones according to the ranking results are JiaYu S3, iPhone 11, Huawei Mate 10 Pro, Huawei P20 Pro, iPhone XS Max.

It is a fact that the experiments using MTFCalculator were performed in non-ideal conditions, for example, the paper and the printer quality used for our tests were obviously not the best ones. Besides, the lighting conditions may vary from place to place. We conducted the experiments at home, the results might have different if the images were captured directly under sunlight in the brightness. Considering all these facts, and looking at the comparison of results with one of the most advanced image testing tools, like Imatest, it can be regarded as a great achievement of our application.

## 4.6 Conclusion

The work in this chapter presented the development of a mobile application, named, MTFCalculator, which measures the performances of built-in smartphone cameras in terms of MTF50 values. The slanted-edge target, an ISO 12233 standard, has been used for the computation of the Modulation Transfer Function. The application is easy to use and requires minimal user interaction. The user only needs to set the target and capture an image, after that the application computes everything on its

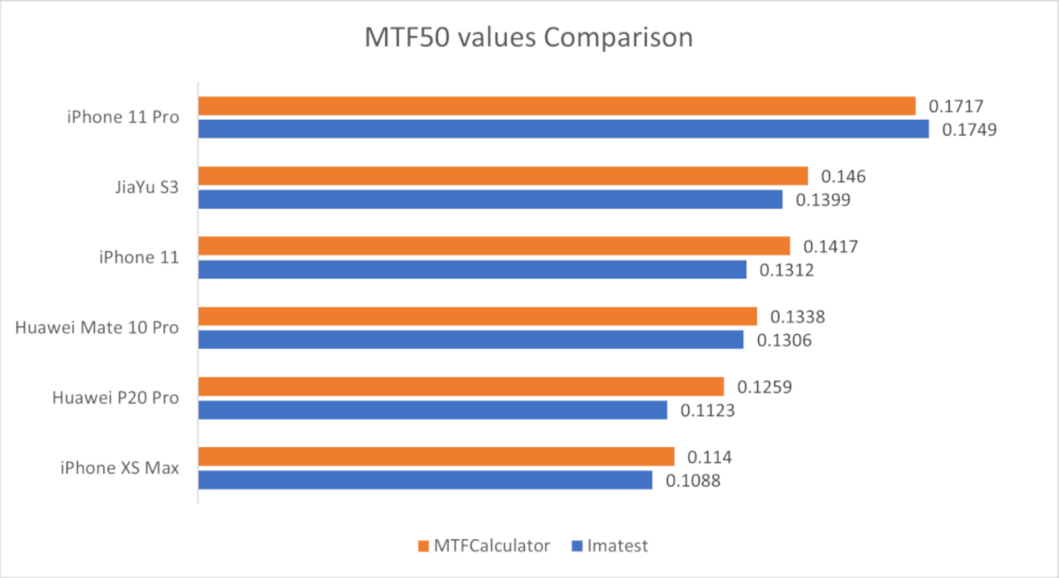


Fig. 4.9 Comparison of MTF50 values between Imatest and MTFCalculator

own and provides the results at the end in terms of MTF50 values, as well as the ESF, LSF and MTF charts are also available for better understanding these quantities.

The experiments were performed on six different smartphones, including, Huawei Mate 10 Pro, Huawei P20 Pro, iPhone XS Max, iPhone 11, iPhone 11 Pro, and JiaYu S3. The experimental results were compared with the Imatest software, which is one of the most famous tools for measuring the performances of imaging devices. The Imatest provides a desktop application whereas the MTFCalculator is a mobile application. The results obtained through MTFCalculator were quite similar as compared to the results achieved by Imatest. In particular, the ranking of the smartphones from the highest MTF50 value to the lowest one results in exactly the same order using both the applications. This can be considered as an achievement of the computational power of our app.

The future work may include to transform the android-based applications into an iOS-based application so that it can be used directly on smartphones having different platforms. Another future work could be to optimize the application to make it more robust and user-friendly.

# Chapter 5

# Conclusions and Future Perspectives

Images are a vital part of our every day life. It would not be wrong to say that no any modern field of technology can survive without using images in one way or the other. Whether it be medical industry, food industry, smartphone industry, or name any automotive industry, applications of image processing can definitely be found everywhere.

The research work carried out during this Ph.D. thesis focused primarily on multiple interdisciplinary applications with image processing being the center of attention. This chapter presents the major contributions of the thesis and provides an outlook for the possible future research work.

## 5.1 Optimization of Power Consumption (Chapter 2)

### 5.1.1 Contributions

The aim of the work, described in Chapter 2, was to reduce the power consumption in OLED-based devices without adversely affecting the displayed contents. The power consumption in OLED displays depends on the color intensity of image pixels. Moreover, the blue color component consumes an enormous amount of power as compared to the red and green color components. The major idea of the proposed solution was to first reduce the intensity of blue color and then to color balance the image so that the resulting image would consume less power and visually would be very similar to the original image. The proposed system was tested on a real OLED-based smartphone, Samsung Galaxy A50. The experimental results demonstrated the success of the proposed system, as the modified images resulted in reducing the power consumption in the smartphone. Furthermore, the subjective test was also conducted to check the acceptability of the modified images by end users, which also yielded the satisfactory results, as most of the votes were obtained for the modified images instead of the original images.

### 5.1.2 Future Work

One way to extend this work is to increase the experimental test set of images to include even more variety of images having many different subjects, and lighting conditions to check the stability of the proposed system. Another approach could be to test the system on multiple smartphones and see how it works from one device to the other. Does the performance remains the same or it varies for different devices? One possible idea is to try to integrate the system to the suggested social networking platforms to see the applicability of the system. Finally, different modern contrast enhancement techniques can be applied and the obtained results can be compared with each other to know the most suitable transformation algorithms.

## 5.2 Automatic Inspection of Hazelnuts (Chapter 3)

### 5.2.1 Contributions

The work described in Chapter 3 is related to the “Food Digital Monitoring” project, in which the quality of hazelnuts was examined using image processing techniques. The axial illumination technique was applied to illuminate the hazelnuts under UV (Ultraviolet) light. When the molecules of hazelnut samples interacted with the UV light, they produced fluorescence and the amount of emitted light was captured using a CCD camera. An axial illuminator case was modeled in a 3D graphics design software and printed using a 3D printer. The hazelnuts samples were classified into contaminated and healthy hazelnut categories. The contaminated hazelnuts were further classified into rotten and pest-infected categories. The proposed system is an automatic inspection system. Once designed, it can be re-used in industries to separate the healthy hazelnuts from the unhealthy ones. The system is cost-effective and faster as compared to the manual inspection techniques.

### 5.2.2 Future Work

One of the many possible future works is to extend the data-set of hazelnut samples to include a wide variety of hazelnuts from many different categories. Another possibility could be to test the proposed system on whole hazelnuts instead of half-cut ones. The proposed system could also be applied on different kinds of food such as other nuts, fruits and vegetables to identify the contaminated parts in them.

## 5.3 Measure of Smartphone Camera Quality (Chapter 4)

### 5.3.1 Contributions

The final project of this thesis was presented in Chapter 4. The aim of the project was to perform qualitative analysis of the built-in cameras in smartphones. Usually the technical specifications provided by the smartphone manufacturers are devoid of a metric related to camera quality. They provide a lot of other features and technical measures which are difficult to understand by a non-technical person. Moreover, it is very difficult to make fair comparisons based on the provided specifications. A mobile application was developed as part of the project to directly measure the camera quality of smartphones. The obtained results were compared with a well-known program, named Imatest, and some satisfactory conclusions were made based on the experimental results.

### 5.3.2 Future Work

A possible future work would be to improve the user interface of the mobile application. Development of the corresponding iOS application is also suggested as a further work for future researchers. The impact of noise could also be measured on the Slanted-edge target by varying the degree of noise in the target image.

# References

[1] Franco Gatti, Andrea Acquaviva, Luca Benini, and Bruno Ricco. Low power control techniques for tft lcd displays. In *Proceedings of the 2002 international conference on Compilers, architecture, and synthesis for embedded systems*, pages 218–224. ACM, 2002.

[2] Parthasarathy Ranganathan, Erik Geelhoed, Meera Manahan, and Ken Nicholas. Energy-aware user interfaces and energy-adaptive displays. *Computer*, 39(3):31–38, 2006.

[3] Vasily G Moshnyaga and Eiji Morikawa. Lcd display energy reduction by user monitoring. In *Computer Design: VLSI in Computers and Processors, 2005. ICCD 2005. Proceedings. 2005 IEEE International Conference on*, pages 94–97. IEEE, 2005.

[4] Johnson Chuang, Daniel Weiskopf, and Torsten Möller. Energy aware color sets. In *Computer Graphics Forum*, volume 28, pages 203–211. Wiley Online Library, 2009.

[5] E Roy Davies. *Image processing for the food industry*, volume 37. World Scientific, 2000.

[6] Sorath Asnani, Maria Giulia Canu, and Bartolomeo Montrucchio. Producing green computing images to optimize power consumption in oled-based displays. In *2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC)*, volume 1, pages 529–534. IEEE, 2019.

[7] Sorath Asnani, Maria Giulia Canu, Laura Farinetti, and Bartolomeo Montrucchio. On producing energy-efficient and contrast-enhanced images for oled-based mobile devices. *Pervasive and Mobile Computing*, page 101384, 2021.

[8] Subu Iyer, Lu Luo, Robert N Mayo, and Parthasarathy Ranganathan. Energy-adaptive display system designs for future mobile environments. In *MobiSys*, volume 3, pages 245–258, 2003.

[9] Stephen R Forrest. The road to high efficiency organic light emitting devices. *Organic Electronics*, 4(2-3):45–48, 2003.

[10] Jin Jang, Sungkyoo Lim, and Myunghwan Oh. Technology development and production of flat panel displays in korea. *Proceedings of the IEEE*, 90(4):501–513, 2002.

[11] Kouji Suzuki. Past and future technologies of information displays. In *IEEE InternationalElectron Devices Meeting, 2005. IEDM Technical Digest.*, pages 16–21. IEEE, 2005.

[12] Achintya K Bhowmik and Robert J Brennan. System-level display power reduction technologies for portable computing and communications devices. In *Portable Information Devices, 2007. PORTABLE07. IEEE International Conference on*, pages 1–5. IEEE, 2007.

[13] Inseok Choi, Hojun Shim, and Naehyuck Chang. Low-power color tft lcd display for hand-held embedded systems. In *Proceedings of the 2002 international symposium on Low power electronics and design*, pages 112–117. ACM, 2002.

[14] Wei-Chung Cheng and Massoud Pedram. Power minimization in a backlit tft-lcd display by concurrent brightness and contrast scaling. *IEEE Transactions on Consumer Electronics*, 50(1):25–32, 2004.

[15] Mian Dong and Lin Zhong. Chameleon: a color-adaptive web browser for mobile oled displays. In *Proceedings of the 9th international conference on Mobile systems, applications, and services*, pages 85–98. ACM, 2011.

[16] Mian Dong, Yung-Seok Kevin Choi, and Lin Zhong. Power modeling of graphical user interfaces on oled displays. In *Proceedings of the 46th Annual Design Automation Conference*, pages 652–657. ACM, 2009.

[17] Chun-Han Lin, Chih-Kai Kang, and Pi-Cheng Hsiu. Catch your attention: Quality-retaining power saving on mobile oled displays. In *Design Automation Conference (DAC), 2014 51st ACM/EDAC/IEEE*, pages 1–6. IEEE, 2014.

[18] Edwin H Land and John J McCann. Lightness and retinex theory. *Josa*, 61(1):1–11, 1971.

[19] Joseph Shinar and Vadim Savvateev. Introduction to organic light-emitting devices. In *Organic Light-Emitting Devices*, pages 1–41. Springer, 2004.

[20] Mian Dong and Lin Zhong. Power modeling and optimization for oled displays. *IEEE Transactions on Mobile Computing*, 11(9):1587–1599, 2011.

[21] Tim Harter, Sander Vroegindeweij, Erik Geelhoed, Meera Manahan, and Parthasarathy Ranganathan. Energy-aware user interfaces: an evaluation of user acceptance. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 199–206. ACM, 2004.

[22] Jonathan Betts-LaCroix. Selective dimming of oled displays, 2010. US Patent App. 12/538,846.

[23] Kiat Wee Tan, Tadashi Okoshi, Archan Misra, and Rajesh Krishna Balan. Focus: a usable & effective approach to oled display power management. In *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*, pages 573–582. ACM, 2013.

[24] Ding Li, Angelica Huyen Tran, and William GJ Halfond. Making web applications more energy efficient for oled smartphones. In *Proceedings of the 36th International Conference on Software Engineering*, pages 527–538, 2014.

[25] Mian Wan, Yuchen Jin, Ding Li, and William GJ Halfond. Detecting display energy hotspots in android apps. In *2015 IEEE 8th International Conference on Software Testing, Verification and Validation (ICST)*, pages 1–10. IEEE, 2015.

[26] Mario Linares-Vásquez, Gabriele Bavota, Carlos Eduardo Bernal Cárdenas, Rocco Oliveto, Massimiliano Di Penta, and Denys Poshyvanyk. Optimizing energy consumption of guis in android apps: a multi-objective approach. In *Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering*, pages 143–154, 2015.

[27] Donghwa Shin, Younghyun Kim, Naehyuck Chang, and Massoud Pedram. Dynamic voltage scaling of oled displays. In *Proceedings of the 48th Design Automation Conference*, pages 53–58. ACM, 2011.

[28] Xiang Chen, Jian Zheng, Yiran Chen, Mengying Zhao, and Chun Jason Xue. Quality-retaining oled dynamic voltage scaling for video streaming applications on mobile devices. In *Proceedings of the 49th Annual Design Automation Conference*, pages 1000–1005. ACM, 2012.

[29] Hadi Hadizadeh, Ivan V Bajic, Parvaneh Saeedi, and Scott Daly. Good-looking green images. In *Image Processing (ICIP), 2011 18th IEEE International Conference on*, pages 3177–3180. IEEE, 2011.

[30] Hadi Hadizadeh. Energy-efficient images. *IEEE Trans. Image Processing*, 26(6):2882–2891, 2017.

[31] Tarik Arici, Salih Dikbas, and Yucel Altunbasak. A histogram modification framework and its application for image contrast enhancement. *IEEE Transactions on image processing*, 18(9):1921–1935, 2009.

[32] Chulwoo Lee, Chul Lee, Young-Yoon Lee, and Chang-Su Kim. Power-constrained contrast enhancement for emissive displays based on histogram equalization. *IEEE Trans. Image Processing*, 21(1):80–93, 2012.

[33] Yan-Tsung Peng, Fan-Chieh Cheng, Li-Ming Jan, and Shanq-Jang Ruan. Histogram shrinking for power-saving contrast enhancement. In *Image Processing (ICIP), 2013 20th IEEE International Conference on*, pages 891–894. IEEE, 2013.

[34] Zia-ur Rahman, Glenn A Woodell, and Daniel J Jobson. A comparison of the multiscale retinex with other image enhancement techniques. *NASA Langley Research Center*, 1997.

[35] Daniel J Jobson, Zia-ur Rahman, and Glenn A Woodell. Properties and performance of a center/surround retinex. *IEEE transactions on image processing*, 6(3):451–462, 1997.

[36] Daniel J Jobson, Zia-ur Rahman, and Glenn A Woodell. A multiscale retinex for bridging the gap between color images and the human observation of scenes. *IEEE Transactions on Image processing*, 6(7):965–976, 1997.

[37] Yeon-Oh Nam, Dong-Yoon Choi, and Byung Cheol Song. Power-constrained contrast enhancement algorithm using multiscale retinex for oled display. *IEEE Transactions on Image Processing*, 23(8):3308–3320, 2014.

[38] Chan Young Jang, Suk-Ju Kang, and Young Hwan Kim. Noniterative power-constrained contrast enhancement algorithm for oled display. *Journal of Display Technology*, 12(11):1257–1267, 2016.

[39] Chih-Kai Kang, Chun-Han Lin, and Pi-Cheng Hsiu. A win-win camera: Quality-enhanced power-saving images on mobile oled displays. In *Low Power Electronics and Design (ISLPED), 2015 IEEE/ACM International Symposium on*, pages 267–272. IEEE, 2015.

[40] Daniele Jahier Pagliari, Enrico Macii, and Massimo Poncino. Lapse: Low-overhead adaptive power saving and contrast enhancement for oleds. *IEEE Transactions on Image Processing*, 27(9):4623–4637, 2018.

[41] Aaron Carroll, Gernot Heiser, et al. An analysis of power consumption in a smartphone. In *USENIX annual technical conference*, volume 14, pages 21–21. Boston, MA, 2010.

[42] Cao Gao, Anthony Gutierrez, Madhav Rajan, Ronald G Dreslinski, Trevor Mudge, and Carole-Jean Wu. A study of mobile device utilization. In *Performance Analysis of Systems and Software (ISPASS), 2015 IEEE International Symposium on*, pages 225–234. IEEE, 2015.

[43] Khoros LLC. The 2021 social media users demographics guide. https://khoros.com/resources/social-media-demographics-guide. Last accessed: March 31, 2022.

[44] Saleem Alhabash and Mengyan Ma. A tale of four platforms: Motivations and uses of facebook, twitter, instagram, and snapchat among college students? *Social Media+ Society*, 3(1):2056305117691544, 2017.

[45] Edwin H Land. Experiments in color vision. *Scientific American*, 200(5):84–99, 1959.

[46] Edwin H. Land. The retinex theory of color vision. *Scientific American*, 237(6):108–128, 1977.

[47] Physics of colour. https://www.cs.mcgill.ca/~rwest/wikispeedia/wpcd/wp/c/Color.htm, 2007. Last Accessed: March 31, 2022.

[48] 8.17. white balance. https://docs.gimp.org/2.10/en/gimp-layer-white-balance.html, 2020. Last accessed: March 31, 2022.

[49] Richard W. Franzen. True color kodak images. http://r0k.us/graphics/kodak/, 2013. Last accessed: March 31, 2022.

[50] Pixabay splendide immagini gratuite. https://pixabay.com/it/, 2020. Last Accessed: March 31, 2022.

[51] Unsplash beautiful free images & pictures. https://unsplash.com/, 2013. Last Accessed: March 31, 2022.

[52] Sos S Agaian, Karen Panetta, and Artyom M Grigoryan. Transform-based image enhancement algorithms with performance measure. *IEEE Transactions on image processing*, 10(3):367–382, 2001.

[53] Turgay Celik and Tardi Tjahjadi. Contextual and variational contrast enhancement. *IEEE Transactions on Image Processing*, 20(12):3431–3441, 2011.

[54] Accubattery - apps on google play. https://play.google.com/store/apps/details?id=com.digibites.accubattery&hl=en, 2020. Last Accessed: March 31, 2022.

[55] Track your battery’s health & performance. https://accubatteryapp.com/, 2022. Last Accessed: March 31, 2022.

[56] Instagram: active users 2018. https://www.statista.com/statistics/253577/number-of-monthly-active-instagram-users/, 2019. Last Accessed: March 31, 2022.

[57] Omnicore, instagram by the numbers: Stats, demographics and fun facts. https://www.omnicoreagency.com/instagram-statistics/, 2020. Last Accessed: March 31, 2022.

[58] 63 facebook statistics you need to know in 2022. https://www.omnicoreagency.com/facebook-statistics/, 2022. Last Accessed: March 31, 2022.

[59] The 2020 social media demographics guide. https://khoros.com/resources/social-media-demographics-guide, Jun 2020. Last Accessed: March 31, 2022.

[60] Gimp - gnu image manipulation program. https://www.gimp.org/. Last Accessed: March 31, 2022.

[61] Li-Ming Jan, Fan-Chieh Cheng, Chia-Hua Chang, Shanq-Jang Ruan, and Chung-An Shen. A power-saving histogram adjustment algorithm for oled-oriented contrast enhancement. *Journal of Display Technology*, 12(4):368–375, 2015.

[62] Arthur COSTE. Project 1 : Histograms. http://www.sci.utah.edu/~acoste/uou/Image/project1/Arthur_COSTE_Project_1_report.html#:~:text=Histogram%20equalization%20is%20a%20method,function%20associated%20to%20the%20image, Sep 2012. Last Accessed: March 31, 2022.

[63] Sorath Asnani, Maurizio Rebaudengo, and Bartolomeo Montrucchio. Detection of contaminated hazelnuts under UV illumination. In *4th International Conference on Image and Graphics Processing*, pages 46–51, 2021.

[64] B Fallico, E Arena, and M Zappala. Roasting of hazelnuts. role of oil in colour development and hydroxymethylfurfural formation. *Food Chemistry*, 81(4):569–573, 2003.

[65] Hakan Kibar and Turgut Öztürk. The effect of moisture content on the physico-mechanical properties of some hazelnut varieties. *Journal of Stored Products Research*, 45(1):14–18, 2009.

[66] Feramuz Ozdemir and Ibrahim Akinci. Physical and nutritional properties of four major commercial turkish hazelnut varieties. *Journal of Food Engineering*, 63(3):341–347, 2004.

[67] A İlhami Köksal, Nevzat Artik, Atilla Şimşek, and Nurdan Güneş. Nutrient composition of hazelnut (corylus avellana l.) varieties cultivated in turkey. *Food Chemistry*, 99(3):509–515, 2006.

[68] FAOSTAT. Food and agriculture organization of the united nations. http://www.fao.org/faostat/en/#data/QCL, 2020. Last accessed: March 31, 2022.

[69] Guner Ozay, Ferda Seyhan, Ceyda Pembeci, Sena Saklar, and Aysun Yilmaz. Factors influencing fungal and aflatoxin levels in turkish hazelnuts (corylus avellana l.) during growth, harvest, drying and storage: A 3-year study. *Food Additives and Contaminants*, 25(2):209–218, 2008.

[70] Alper Taner, Yeşim Benal Öztekin, and Hüseyin Duran. Performance analysis of deep learning cnn models for variety classification in hazelnut. *Sustainability*, 13(12):6527, 2021.

[71] SA Mehlenbacher, DC Smith, and Loretta K Brenner. Variance components and heritability of nut and kernel defects in hazelnut. *Plant Breeding*, 110(2):144–152, 1993.

[72] Lara Bosco, Silvia T Moraglio, and Luciana Tavella. Halyomorpha halys, a serious threat for hazelnut in newly invaded areas. *Journal of Pest Science*, 91(2):661–670, 2018.

[73] Antonia Memoli, Donatella Albanese, Marco Esti, Claudio Lombardelli, Alessio Crescitelli, Marisa Di Matteo, and Ilaria Benucci. Effect of bug damage and mold contamination on fatty acids and sterols of hazelnut oil. *European Food Research and Technology*, 243(4):651–658, 2017.

[74] Roberto Moscetti, Ron P Haff, Sirinnapa Saranwong, Danilo Monarca, Massimo Cecchini, and Riccardo Massantini. Nondestructive detection of insect infested chestnuts based on nir spectroscopy. *Postharvest Biology and Technology*, 87:88–94, 2014.

[75] A Lopez, MT Pique, J Boatella, J Parcerisa, A Romero, A Ferrá, and J Garcí. Influence of drying conditions on the hazelnut quality. i. lipid oxidation. *Drying technology*, 15(3-4):965–977, 1997.

[76] Cristian Silvestri, Loretta Bacchetta, Andrea Bellincontro, and Valerio Cristofori. Advances in cultivar choice, hazelnut orchard management, and nut storage to enhance product quality and safety: an overview. *Journal of the Science of Food and Agriculture*, 101(1):27–43, 2021.

[77] Daniela Ghirardello, Cecilia Contessa, Nadia Valentini, Giuseppe Zeppa, L Rolle, Vincenzo Gerbi, and Roberto Botta. Effect of storage conditions on chemical and physical characteristics of hazelnut (corylus avellana l.). *Postharvest Biology and Technology*, 81:37–43, 2013.

[78] A Turan. Effect of drying on the chemical composition of çakıldak (cv) hazelnuts during storage. *Grasas y Aceites*, 70(1):e296–e296, 2019.

[79] Roberto Moscetti, Wouter Saeys, Janos C Keresztes, Mohammad Goodarzi, Massimo Cecchini, Monarca Danilo, and Riccardo Massantini. Hazelnut quality sorting using high dynamic range short-wave infrared hyperspectral imaging. *Food and bioprocess technology*, 8(7):1593–1604, 2015.

[80] A Giraudo, R Calvini, G Orlandi, A Ulrici, F Geobaldo, and F Savorani. Development of an automated method for the identification of defective hazelnuts based on rgb image analysis and colourgrams. *Food Control*, 94:233–240, 2018.

[81] Amersham Pharmacia Biotech. Fluorescence imaging: Principles and methods. http://www.bu.edu/picf/files/2010/10/Fluor-ImagingPrinciples.pdf, 2000. Last accessed: March 31, 2022.

[82] Habil Kalkan and Yasemin Yardimci. Extraction of discriminative features from hyperspectral data. In *International Conference on Data Mining Workshops*, pages 414–419. IEEE, 2008.

[83] Habil Kalkan and Yasemin Yardimci. Detection of contaminated hazelnuts by multispectral imaging. In *16th Signal Processing, Communication and Applications Conference*, pages 1–4, Turkey, 2008. IEEE.

[84] H Kalkan, P Beriat, Y Yardimci, and TC Pearson. Detection of contaminated hazelnuts and ground red chili pepper flakes by multispectral imaging. *Computers and Electronics in Agriculture*, 77(1):28–34, 2011.

[85] Paolo Menesatti, Corrado Costa, Graziella Paglia, Federico Pallottino, Stefano D'Andrea, Valentina Rimatori, and Jacopo Aguzzi. Shape-based methodology for multivariate discrimination among italian hazelnut cultivars. *Biosystems Engineering*, 101(4):417–424, 2008.

[86] Sümeyye Bayrakdar, Bekir Çomak, Derya Başol, and İbrahim Yücedag. Determination of type and quality of hazelnut using image processing techniques. In *23nd Signal Processing and Communications Applications Conference (SIU)*, pages 616–619. IEEE, 2015.

[87] Seyit Ahmet Guvenc, Fatih Ahmet Senel, and Bayram Cetisli. Classification of processed hazelnuts with computer vision. In *23rd Signal Processing and Communications Applications Conference (SIU)*, pages 1362–1365, Turkey, 2015. IEEE.

[88] Serdar Solak and Umut Altinişik. Detection and classification of hazelnut fruit by using image processing techniques and clustering methods. *Sakarya Üniversitesi Fen Bilimleri Enstitüsü Dergisi*, 22(1):56–65, 2018.

[89] KOC Caner, Dilara Gerdan, Maksut B Eminoğlu, UğUr YegüL, KOC Bulent, and Mustafa Vatandaş. Classification of hazelnut cultivars: comparison of dl4j and ensemble learning algorithms. *Notulae Botanicae Horti Agrobotanici Cluj-Napoca*, 48(4):2316–2327, 2020.

[90] A Bellincontro, A Fracas, G Anelli, F Mencarelli, C DiNatale, and G Esposito. Use of nir technique to measure the acidity and water content of hazelnuts. *Acta horticulturae*, 2005.

[91] A Bellincontro, F Mencarelli, R Forniti, and M Valentini. Use of nir-aotf spectroscopy and mri for quality detection of whole hazelnuts. In *7th International Congress on Hazelnut 845*, pages 593–598, 2008.

[92] Roberto Moscetti, Ron P Haff, Ben Aernouts, Wouter Saeys, Danilo Monarca, Massimo Cecchini, and Riccardo Massantini. Feasibility of vis/nir spectroscopy for detection of flaws in hazelnut kernels. *Journal of Food Engineering*, 118(1):1–7, 2013.

[93] A Pannico, RE Schouten, B Basile, R Romano, EJ Woltering, and C Cirillo. Non-destructive detection of flawed hazelnut kernels and lipid oxidation assessment using nir spectroscopy. *Journal of Food Engineering*, 160:42–48, 2015.

[94] Ikramullah Khosa and Eros Pasero. Feature extraction in x-ray images for hazelnuts classification. In *International Joint Conference on Neural Networks (IJCNN)*, pages 2354–2360, China, 2014. IEEE.

[95] Domenico Di Caro, Consolatina Liguori, Antonio Pietrosanto, and Paolo Sommella. Investigation of the nmr techniques to detect hidden defects in hazelnuts. In *International Instrumentation and Measurement Technology Conference Proceedings*, pages 1–5, Taiwan, 2016. IEEE.

[96] Brian G Osborne. Near-infrared spectroscopy in food analysis. *Encyclopedia of analytical chemistry: applications, theory and instrumentation*, 2006.

[97] Ted W Tyson and Rex L Clark. An investigation of the fluorescent properties of aflatoxin infected pecans. *Transactions of the ASAE*, 17(5):942–0944, 1974.

[98] Susumu Hirano, Noriyuki Okawara, and Shuichirou Narazaki. Near infra red detection of internally moldy nuts. *Bioscience, biotechnology, and biochemistry*, 62(1):102–107, 1998.

[99] Ali Güneş, Habil Kalkan, Efkan Durmuş, and Mehmet Burak Bütükcan. Detection of aflatoxin contaminated figs using near-infrared (nir) reflectance spectroscopy. In *International Conference on Electronics, Computer and Computation (ICECCO)*, pages 123–126, Turkey, 2013. IEEE.

[100] Giuseppe Bonifazi, Giuseppe Capobianco, Riccardo Gasbarrone, and Silvia Serranti. Contaminant detection in pistachio nuts by different classification methods applied to short-wave infrared hyperspectral images. *Food Control*, page 108202, 2021.

[101] Daniel Guyer and Xiukun Yang. Use of genetic artificial neural networks and spectral imaging for defect detection on cherries. *Computers and electronics in agriculture*, 29(3):179–194, 2000.

[102] Patrick M Mehl, Yud-Ren Chen, Moon S Kim, and Diane E Chan. Development of hyperspectral imaging technique for the detection of apple surface defects and contaminations. *Journal of food engineering*, 61(1):67–81, 2004.

[103] Piotr Baranowski, Wojciech Mazurek, and Joanna Pastuszka-Woźniak. Supervised classification of bruised apples with respect to the time after bruising on the basis of hyperspectral imaging data. *Postharvest Biology and Technology*, 86:249–258, 2013.

[104] Changhong Liu, Wei Liu, Wei Chen, Jianbo Yang, and Lei Zheng. Feasibility in multispectral imaging for predicting the content of bioactive compounds in intact tomato fruit. *Food Chemistry*, 173:482–488, 2015.

[105] Sandra Munera, José Blasco, Jose M Amigo, Sergio Cubero, Pau Talens, and Nuria Aleixos. Use of hyperspectral transmittance imaging to evaluate the internal quality of nectarines. *Biosystems Engineering*, 182:54–64, 2019.

[106] Tom C Pearson, Donald T Wicklow, Elizabeth B Maghirang, Feng Xie, and Floyd E Dowell. Detecting aflatoxin in single corn kernels by transmittance and reflectance spectroscopy. *Transactions of the ASAE*, 44(5):1247, 2001.

[107] Ze'ev Schmilovitch, Timea Ignat, Victor Alchanatis, Janna Gatker, Viacheslav Ostrovsky, and József Felföldi. Hyperspectral imaging of intact bell peppers. *Biosystems engineering*, 117:83–93, 2014.

[108] Edmund optics: 75mm diameter x 50mm fl , aspheric condenser lens. https://www.edmundoptics.eu/p/75mm-diameter-x-50mm-flaspheric-condenser-lens/6712/. Last accessed: March 31, 2022.

[109] Trimble Inc. Sketchup for web. https://www.sketchup.com/products/sketchup-for-web, Aug 2000. Last accessed: March 31, 2022.

[110] Wayne Rasband. Imagej - national institutes of health. https://imagej.nih.gov/ij/, 2020. Last accessed: March 31, 2022.

[111] Stephan Saalfeld. Linear stack alignment with sift. https://imagej.net/Linear_Stack_Alignment_with_SIFT, Jun 2008. Last accessed: March 31, 2022.

[112] Wayne Rasband and Gabriel Landini. Calculator plus - imagej. https://imagej.nih.gov/ij/plugins/calculator-plus.html, Jun 2009. Last accessed: March 31, 2022.

[113] Sorath Asnani, Fabrizio Lo Presti, Luigi Amato, and Bartolomeo Montrucchio. Mtfcalculator–a mobile application for measuring the modulation transfer function of built-in cameras of smartphones using iso 12233 slanted-edge method. In *2021 IEEE 6th International Conference on Signal and Image Processing (ICSIP)*, pages 1273–1278. IEEE, 2021.

[114] Imatest master version 2020.2. https://www.imatest.com/support/download/, 2020. Last Accessed: March 31, 2022.

[115] Phonearena - phone news, reviews and specs. https://www.phonearena.com/. Last Accessed: March 31, 2022.

[116] Jiayu s3 specs, review, release date. https://phonesdata.com/en/smartphones/jiayu/jiayu-s3-5460147/. Last Accessed: March 31, 2022.

[117] Pete. Photography 101: What is camera aperture?: Motif blog. https://www.blog.motifphotos.com/photography-101-what-is-camera-aperture/, Jul 2021. Last Accessed: March 31, 2022.

[118] Dave Black, Diane Berkenfeld, Lindsay Silverman, and Mike Corrado. Focal length: Understanding camera zoom and lens focal length: Nikon. https://www.nikonusa.com/en/learn-and-explore/a/tips-and-techniques/understanding-focal-length.html#. Last Accessed: March 31, 2022.

[119] Why camera sensor size is more important than more megapixels. https://www.androidauthority.com/camera-sensor-size-1095299/, Apr 2021. Last Accessed: March 31, 2022.

[120] Good phone camera vs. bad phone camera: 7 major differences. https://smartphonedomain.com/good-phone-camera-vs-bad-phone-camera/#:~:text=One%20thousand%20micrometres%20equals%201,counts%2C%20it%20uses%20pixel%20binning. Last Accessed: March 31, 2022.

[121] Edmund optics: Introduction to modulation transfer function. https://www.edmundoptics.eu/knowledge-center/application-notes/optics/introduction-to-modulation-transfer-function/, 2011. Last Accessed: March 31, 2022.

[122] Magali Estribeau and Pierre Magnan. Fast mtf measurement of cmos imagers using iso 12333 slanted-edge methodology. In *Detectors and Associated Signal Processing*, volume 5251, pages 243–252. International Society for Optics and Photonics, 2004.

[123] Patrick Mueller, Matthias Lehmann, and Alexander Braun. Optical quality metrics for image restoration. In *Digital Optical Technologies 2019*, volume 11062, page 1106214. International Society for Optics and Photonics, 2019.

[124] Christian Loebich, Dietmar Wueller, Bruno Klingen, and Anke Jaeger. Digital camera resolution measurements using sinusoidal siemens stars. In *Digital Photography III*, volume 6502, pages 214–224. SPIE, 2007.

[125] Uwe Artmann. Image quality assessment using the dead leaves target: experience with the latest approach and further investigations. In *Digital Photography XI*, volume 9404, pages 130–144. SPIE, 2015.

[126] Arnold Daniels, Glenn D Boreman, Alfred Dale Ducharme, and Eyal Sapir. Random transparency targets for modulation transfer function measurement in the visible and infrared regions. *Optical Engineering*, 34(3):860–868, 1995.

[127] Xusheng Zhang and Dingguo Sha. Modulation transfer function evaluation of charge-coupled-device camera system based on liquid-crystal display random targets. In *Optical Design and Testing II*, volume 5638, pages 1014–1021. International Society for Optics and Photonics, 2005.

[128] Glenn D Boreman and Sidney Yang. Modulation transfer function measurement using three-and four-bar targets. *Applied optics*, 34(34):8050–8052, 1995.

[129] Xujie Zhang, Tamar Kashti, Dror Kella, Tal Frank, Doron Shaked, Robert Ulichney, Mani Fischer, and Jan P Allebach. Measuring the modulation transfer function of image capture devices: what do the numbers really mean? In *Image Quality and System Performance IX*, volume 8293, page 829307. International Society for Optics and Photonics, 2012.

[130] Iso 12233:2017 photography - electronic still picture imaging - resolution and spatial frequency responses. https://www.iso.org/standard/71696.html, Jan 2017. Last Accessed: March 31, 2022.

[131] Jackson KM Roland. A study of slanted-edge mtf stability and repeatability. In *Image Quality and System Performance XII*, volume 9396, page 93960L. International Society for Optics and Photonics, 2015.

[132] Alexis P Tzannes and Jonathan Martin Mooney. Measurement of the modulation transfer function of infrared cameras. *Optical Engineering*, 34(6):1808–1817, 1995.

[133] Yen-Hong Kao, Michael Albert, Ann-Katherine Carton, Hilde Bosmans, and Andrew DA Maidment. A software tool for measurement of the modulation transfer function. In *Medical Imaging 2005: Physics of Medical Imaging*, volume 5745, pages 1199–1208. International Society for Optics and Photonics, 2005.

[134] Hyundeok Hwang, Young-Wan Choi, Sunghee Kwak, Moongyu Kim, and WonKyu Park. Mtf assessment of high resolution satellite images using iso 12233 slanted-edge method. In *Image and Signal Processing for Remote Sensing XIV*, volume 7109, page 710905. International Society for Optics and Photonics, 2008.

[135] Don Williams and Peter D Burns. Evolution of slanted edge gradient sfr measurement. In *Image quality and system performance XI*, volume 9016, page 901605. International Society for Optics and Photonics, 2014.

[136] Kenichiro Masaoka, Takayuki Yamashita, Yukihiro Nishida, and Masayuki Sugawara. Modified slanted-edge method and multidirectional modulation transfer function estimation. *Optics express*, 22(5):6040–6046, 2014.

[137] Hang Li, Changxiang Yan, and Jianbing Shao. Measurement of the modulation transfer function of infrared imaging system by modified slant edge method. *Journal of the Optical Society of Korea*, 20(3):381–388, 2016.

[138] Oleg A Perezyabov, Aleksandr N Baranov, Nadezhda K Maltseva, and Aleksandr V Ilinski. Application for determining the modulation transfer function of the smartphone built-in camera. In *2017 Progress In Electromagnetics Research Symposium-Spring (PIERS)*, pages 3804–3808. IEEE, 2017.

[139] Haosu Zhang, Chuang Li, and Yaxuan Duan. Modified slanted-edge method to measure the modulation transfer function of camera. *Optik*, 157:635–643, 2018.

[140] Xufen Xie, Hongda Fan, Anding Wang, Nianyu Zou, and Yuncui Zhang. Regularized slanted-edge method for measuring the modulation transfer function of imaging systems. *Applied optics*, 57(22):6552–6558, 2018.

[141] Etienne Homassel, Catherine Barrat, Frédéric Alves, Gilles Aubry, and Guillaume Arquetoux. Improved modulation transfer function evaluation method of a camera at any field point with a scanning tilted knife edge. In *Electro-Optical and Infrared Systems: Technology and Applications XVII*, volume 11537, page 115370B. International Society for Optics and Photonics, 2020.

[142] Imatest. Sharpness: What is it and how it is measured. https://www.imatest.com/docs/sharpness/, 2021. Last Accessed: March 31, 2022.

[143] Fabrizio Lo Presti and Luigi Amato. Mtf calculator apk. https://github.com/FabrizioDys/MTFCalculatorAPK, Jun 2021. Last Accessed: March 31, 2022.

[144] Albert. How to measure modulation transfer function. https://harvestimaging.com/blog/?p=1328, Jun 2014. Last Accessed: March 31, 2022.

[145] Jonathan M. Blackledge. Chapter 16 - segmentation and edge detection. In Jonathan M. Blackledge, editor, *Digital Image Processing*, Woodhead Publishing Series in Electronic and Optical Materials, pages 487–511. Woodhead Publishing, 2005.

[146] Siddharth Misra and Yaokun Wu. Chapter 10 - machine learning assisted segmentation of scanning electron microscopy images of organic-rich shales with feature extraction and feature ranking. In Siddharth Misra, Hao Li, and Jiabo He, editors, *Machine Learning for Subsurface Characterization*, pages 289–314. Gulf Professional Publishing, 2020.

[147] Blurring images – image processing with python. https://datacarpentry.org/image-processing/06-blurring/. Last Accessed: March 31, 2022.

[148] Nobuyuki Otsu. A threshold selection method from gray level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9:62–66, 1979.

[149] Otsu's method. https://en.wikipedia.org/wiki/Otsu%27s_method, Apr 2021. Last accessed: March 31, 2022.

[150] Opencv - canny edge detection. https://docs.opencv.org/master/da/d22/tutorial_py_canny.html, 2021. Last Accessed: March 31, 2022.

[151] Kevin Kohm. Modulation transfer function measurement method and results for the orbview-3 high resolution imaging satellite. In *Proceedings of ISPRS*, volume 35, pages 12–23, 2004.

[152] Françoise Viallefont-Robinet, Dennis Helder, Renaud Fraisse, Amy Newbury, Frans van den Bergh, DongHan Lee, and Sébastien Saunier. Comparison of mtf measurements using edge method: towards reference data set. *Optics express*, 26(26):33625–33648, 2018.

[153] Savitzky–golay filter. https://en.wikipedia.org/wiki/Savitzky%E2%80%93Golayfilter, Apr 2021. Last Accessed: March 31, 2022.

[154] Bhavin Nayak. Pds compute mtf. https://github.com/bvnayak/PDS_Compute_MTF, Feb 2016. Last Accessed: March 31, 2022.

[155] Norman Koren. https://pcv.oss-cn-shanghai.aliyuncs.com/wp-content/upload/2019/11/Imatest-Documentation.pdf, 2009. Last Accessed: March 31, 2022.

# Appendix A

# Publications

## Journal Papers

- **Asnani, S.**, Canu, M. G., Farinetti, L., and Montrucchio, B. (2021). On producing energy-efficient and contrast-enhanced images for OLED-based mobile devices. Pervasive and Mobile Computing, 101384 (Elsevier).
- **Asnani, S.**, Rebaudengo, M., Ferrero, R, and Montrucchio, B. Classification of healthy, pest-infected, and rotten hazelnuts by means of fluorescence and spectral imaging techniques (Submitted).

## Conference Papers

- **Asnani, S.**, Presti, F. L., Amato, L., and Montrucchio, B. (2021, October). MTFCalculator–A Mobile Application for Measuring the Modulation Transfer Function of Built-in Cameras of Smartphones Using ISO 12233 Slanted-Edge Method. In 2021 IEEE 6th International Conference on Signal and Image Processing (ICSIP) (pp. 1273-1278). IEEE. **(Best Presentation Award)**
- **Asnani, S.**, Rebaudengo, M., and Montrucchio, B. (2021, January). Detection of Contaminated Hazelnuts under UV Illumination. In 2021 The 4th International Conference on Image and Graphics Processing (pp. 46-51).

- **Asnani, S.**, Canu, M. G., and Montrucchio, B. (2019, July). Producing Green Computing Images to Optimize Power Consumption in OLED-Based Displays. In 2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC) (Vol. 1, pp. 529-534). IEEE.

# Appendix B

# Code Snippets

## B.1 Computation of RGB Intensity Values

```
close all
clearvars
clc

inPath = 'D:\PhD\OLED\OLED\Images\InputImages\';
outPath = 'D:\PhD\OLED\OLED\Images\OutputImagesPlugin\';

N = 48;
R_total1= zeros(1,N);
G_total1= zeros(1,N);
B_total1= zeros(1,N);
I_total1= zeros(1,N);
R_percentage1= zeros(1,N);
G_percentage1= zeros(1,N);
B_percentage1= zeros(1,N);
R_total2= zeros(1,N);
G_total2= zeros(1,N);
B_total2= zeros(1,N);
I_total2= zeros(1,N);
R_percentage2= zeros(1,N);
G_percentage2= zeros(1,N);
B_percentage2= zeros(1,N);
intensity_Difference= zeros(1,N);
```

```

for i = 1 : 48
    inFileName = strcat(inPath, 'kodim', num2str(i), '...
   .png');
    if (exist(inFileName, 'file') == 0)
        continue
    end

    I1 = imread(inFileName); % read image
    Ir1=I1(:,:,1); % read red channel
    Ig1=I1(:,:,2); % read green channel
    Ib1=I1(:,:,3); % read blue channel

    R_total1(i) = sum(Ir1(:)); % sum of total red pixels
    G_total1(i) = sum(Ig1(:)); % sum of total green ...
   pixels
    B_total1(i) = sum(Ib1(:)); % sum of total blue pixels

    %% Calculate the image total intensity
    I_total1(i) = R_total1(i) + G_total1(i) + B_total1(i)...
   ;

    %% Calculate the percentage of each Channel
    R_percentage1(i) =  R_total1(i) / I_total1(i) * 100; ...
        % Red Channel Percentage
    G_percentage1(i) =  G_total1(i) / I_total1(i) * 100; ...
        % Green Channel Percentage
    B_percentage1(i) =  B_total1(i) / I_total1(i) * 100; ...
    % Blue Channel Percentage; it is multiplied by 3 ...
   because the intensity of blue channels is almost 3 ...
   times that of Red and Green channels

    outFileName = strcat(outPath, 'kodim', num2str(i), '...
   .png');
    if (exist(outFileName, 'file') == 0)
        continue
    end

    % Read output file
    I2 = imread(outFileName); % read image

    Ir2=I2(:,:,1); % read red channel
    Ig2=I2(:,:,2); % read green channel
    Ib2=I2(:,:,3); % read blue channel

    R_total2(i) = sum(Ir2(:)); % sum of total red pixels ...
   in output image
    G_total2(i) = sum(Ig2(:)); % sum of total green ...
   pixels in output image
    B_total2(i) = sum(Ib2(:)); % sum of total blue pixels...
    in output image

    %% Calculate the image total intensity in output ...
   image
    I_total2(i) = R_total2(i) + G_total2(i) + B_total2(i)...
   ;

    %% Calculate the percentage of each Channel
     R_percentage2(i) =  R_total2(i) / I_total2(i) * 100;...
            % Red Channel Percentage
     G_percentage2(i) =  G_total2(i) / I_total2(i) * 100;...
            % Green Channel Percentage
     B_percentage2(i) =  B_total2(i) / I_total2(i) * 100;...
        % Blue Channel Percentage

    %power = ((I_total1 - I_total2)/I_total1) * 100
    intensity_Difference(i) = (1 - (I_total2(i) / ...
   I_total1(i))) * 100;

end
```

## B.2 Python Script for GIMP Plugin

```
from gimpfu import *

def reduce_power_consumption(image, drawable):

    # To start undo group
    pdb.gimp_image_undo_group_start(image)

    #####################################################
    ############# Modified Land Effect ##################

    # Decomposing an image into RGB layers

    decompose_type = "RGB"
    layers_mode = 1
    new_image_decomposed, new_image, new_image, new_image...
     = pdb.plug_in_decompose(image, drawable, ...
   decompose_type, layers_mode)

    # Get the Red layer of the image
    layer_copy_red = pdb.gimp_layer_new_from_drawable(...
   new_image_decomposed.layers[0], image)
    pdb.gimp_item_set_name(layer_copy_red, "RGB - Red")

    # Insert the Red layer to the existing image
    parent = None
    position = -1
    pdb.gimp_image_insert_layer(image, layer_copy_red, ...
   parent, position)

    # Get the Green layer of the image
    layer_copy_green = pdb.gimp_layer_new_from_drawable(...
   new_image_decomposed.layers[1], image)
    pdb.gimp_item_set_name(layer_copy_green, "RGB - Green...
   ")

    # Insert the Green layer to the existing image
    parent = None
    position = -1
    pdb.gimp_image_insert_layer(image, layer_copy_green, ...
   parent, position)

    # creating a new layer with specified options
    width = pdb.gimp_image_width(image)
    height = pdb.gimp_image_height(image)
    type = 0  # RGB-IMAGE
    name = "Red"
    opacity = 100
    mode = 3 # MULTIPLY-MODE
    layer = pdb.gimp_layer_new(image, width, height, type...
   , name, opacity, mode)

    # set the foreground color to Red
    foreground = (255, 0, 0)
    pdb.gimp_context_set_foreground(foreground)

    # inserting newly created layer to the existing image
    parent = None
    position = 1 # Because it should be above the Red ...
   channel layer and below the green channel layer
    pdb.gimp_image_insert_layer(image, layer, parent, ...
   position)

    # to get the active layer
    active_layer = pdb.gimp_image_get_active_layer(image)

    # to fill the layer with red color
    fill_mode = 0 #FG-BUCKET-FILL
    paint_mode = 24 #REPLACE-MODE
    opacity = 100
    threshold = 0
    sample_merged = 0 #False
    x = 0
    y = 0
    pdb.gimp_edit_bucket_fill(active_layer, fill_mode, ...
   paint_mode, opacity, threshold, sample_merged, x, y)

    # Merge down red layers
    merge_layer = active_layer
    merge_type = 0 #EXPAND-AS-NECESSARY (0)
    layer = pdb.gimp_image_merge_down(image, merge_layer,...
    merge_type)
    # Set the opacity of the merged Red layer to 84.5
    opacity = 84.5
    pdb.gimp_layer_set_opacity(layer, opacity)

    # Repeating above steps for the Green layer
    # creating a new layer with specified options
    width = pdb.gimp_image_width(image)
    height = pdb.gimp_image_height(image)
    type = 0  #RGB-IMAGE
    name = "Green"
    opacity = 100
    mode = 3 #MULTIPLY-MODE
    layer = pdb.gimp_layer_new(image, width, height, type...
   , name, opacity, mode)

    # set the foreground color to Green
    foreground = (0, 255, 0)
    pdb.gimp_context_set_foreground(foreground)

    # inserting newly created layer to the existing image
    parent = None
    position = 0 # The layer will be on top of all other ...
   layers
    pdb.gimp_image_insert_layer(image, layer, parent, ...
   position)

    # to get the active layer
    active_layer = pdb.gimp_image_get_active_layer(image)

    # to fill the layer with green color
    fill_mode = 0 #FG-BUCKET-FILL
    paint_mode = 24 #REPLACE-MODE
    opacity = 100
    threshold = 0
    sample_merged = 0 #False
    x = 0
    y = 0
    pdb.gimp_edit_bucket_fill(active_layer, fill_mode, ...
   paint_mode, opacity, threshold, sample_merged, x, y)

    # Merge down green layers
    merge_layer = active_layer
    merge_type = 0 #EXPAND-AS-NECESSARY (0)
    layer = pdb.gimp_image_merge_down(image, merge_layer,...
    merge_type)

    # Set the opacity of the merged Green layer to 24.5
    opacity = 24.5
    pdb.gimp_layer_set_opacity(layer, opacity)

    ###################################################
    ####### White balancing and Retinex Filter ##########

    # Create a new layer from the visible layer for White...
    Balance
    name = "White Balance"
    layer = pdb.gimp_layer_new_from_visible(image, image,...
    name)

    # Insert the new layer to the existing image for ...
   White Balance
    parent = None
    position = 0
    pdb.gimp_image_insert_layer(image, layer, parent, ...
   position)

    # Create a new layer from the visible layer for ...
   Retinex Filter
    name = "Retinex Filter"
    layer = pdb.gimp_layer_new_from_visible(image, image,...
    name)

    # Insert the new layer to the existing image for ...
   Retinex Filter
    parent = None
    position = 0
    pdb.gimp_image_insert_layer(image, layer, parent, ...
   position)

    # Set the opacity of the Retinex Filter layer to 20
    opacity = 20
    pdb.gimp_layer_set_opacity(layer, opacity)

    # Apply White Balance
    pdb.gimp_levels_stretch(image.layers[1])


    retinex_layer = image.layers[0]
    scale = 240 # Scale
    nscales = 6 # Scale division
    scales_mode = 2 # Level = High
    cvar = 2.4 # Dynamic
    pdb.plug_in_retinex(image, retinex_layer, scale, ...
   nscales, scales_mode, cvar)

    ####################################################
    ########### Reducing Intensity Level ################

    # Create a new layer from the visible layer for ...
   Intensity Level
    name = "Level"
    layer = pdb.gimp_layer_new_from_visible(image, image,...
    name)

    # Insert the new layer to the existing image for ...
   Intensity Level
    parent = None
    position = 0
    pdb.gimp_image_insert_layer(image, layer, parent, ...
   position)

    # Set Intensity Level to 0.72
    channel = 0 # HISTOGRAM-VALUE (0)
    low_input = 0
    high_input = 255
    gamma = 0.72
    low_output = 0
    high_output = 255
    pdb.gimp_levels(layer, channel, low_input, high_input...
   , gamma, low_output, high_output)

    # To end undo group
    pdb.gimp_image_undo_group_end(image)




register(
    "python-fu-reduce_power_consumption",
    "Modifies an image to reduce its power consumption",
    "Performs four main operations on an image to reduce ...
   the power consumed by an image. The four operations ...
   are: The Land Effect, White Balance, Retinex Filter ...
   and Adjusting the Intensity Levels.",
    "Sorath Asnani", "Sorath Asnani", "2019",
    "Reduce Power Consumption",
    "*", # type of image it works on (*, RGB, RGB*, RGBA,...
    GRAY etc...)
    [
        # basic parameters are: (UI_ELEMENT, "variable", ...
   "label", Default)
        (PF_IMAGE, "image", "takes current image", None),
        (PF_DRAWABLE, "drawable", "Input layer", None)
    ],
    [],
    reduce_power_consumption, menu="<Image>/Filters")  # ...
   second item is menu location

main()
```

## B.3 Compuation of Edge Spread Function (ESF)

```
 def ESF_compute(self):
     kernel = np.ones((3, 3), np.float32)/9
     smooth_img = cv2.filter2D(self.data, -1, kernel)
     row = self.data.shape[0]
     column = self.data.shape[1]
     array_values_near_edge = np.empty([row, 13])
     array_positions = np.empty([row, 13])
     edge_pos = np.empty(row)
     smooth_img = smooth_img.astype(float)
     for i in range(0, row):     #analizzo riga per ...
riga l immagine
         diff_img = smooth_img[i, 1:] - smooth_img[i, ...
0:(column-1)]
         abs_diff_img = np.absolute(diff_img)
         abs_diff_max = np.amax(abs_diff_img)  #serve ...
per trovare l edge all interno della riga
         if abs_diff_max == 1:
             raise IOError('No Edge Found')
         app_edge = np.where(abs_diff_img == ...
abs_diff_max)
         bound_edge_left = app_edge[0][0] - 2
         bound_edge_right = app_edge[0][0] + 3
         strip_cropped = self.data[i, bound_edge_left:...
bound_edge_right]
         temp_y = np.arange(1, 6)
         edge_pos_temp = np.interp(self.threshold, ...
strip_cropped, temp_y)
         edge_pos[i] = edge_pos_temp + bound_edge_left...
 - 1
         bound_edge_left_expand = app_edge[0][0] - 6
         bound_edge_right_expand = app_edge[0][0] + 7
         array_values_near_edge[i, :] = self.data[i, ...
bound_edge_left_expand:bound_edge_right_expand]
         array_values_near_edge[i, :] = self.data[i, ...
bound_edge_left_expand:bound_edge_right_expand]
         array_positions[i, :] = np.arange(...
bound_edge_left_expand, bound_edge_right_expand)
     y = np.arange(0, row)
     nans, x = nan_helper(edge_pos)
```

```
        edge_pos[nans] = np.interp(x(nans), x(¬nans), ...
edge_pos[¬nans])

        array_positions_by_edge = array_positions - np.....
transpose(edge_pos * np.ones((13, 1)))
        num_row = array_positions_by_edge.shape[0]
        num_col = array_positions_by_edge.shape[1]
        array_values_by_edge = np.reshape(...
array_values_near_edge, num_row*num_col, order='F')
        array_positions_by_edge = np.reshape(...
array_positions_by_edge, num_row*num_col, order='F')

        bin_pad = 0.0001
        pixel_subdiv = 0.10
        topedge = np.amax(array_positions_by_edge) + ...
bin_pad + pixel_subdiv
        botedge = np.amin(array_positions_by_edge) - ...
bin_pad
        binedges = np.arange(botedge, topedge+1, ...
pixel_subdiv)
        numbins = np.shape(binedges)[0] - 1

        binpositions = binedges[0:numbins] + (0.5) * ...
pixel_subdiv

        h, whichbin = np.histogram(...
array_positions_by_edge, binedges)
        whichbin = np.digitize(array_positions_by_edge, ...
binedges)
        binmean = np.empty(numbins)

        for i in range(0, numbins):
            flagbinmembers = (whichbin == i)
            binmembers = array_values_by_edge[...
flagbinmembers]
            binmean[i] = np.mean(binmembers)
        nans, x = nan_helper(binmean)
        binmean[nans] = np.interp(x(nans), x(¬nans), ...
binmean[¬nans])
        esf = binmean
        xesf = binpositions
        xesf = xesf - np.amin(xesf)
        self.xesf = xesf
        esf_smooth = self.savitzky_golay(esf, 51, 3)
        self.esf = esf
        self.esf_smooth = esf_smooth
        plt.subplot(4, 1, 2)
        plt.title("ESF Curve")
        plt.xlabel("pixel")
        plt.ylabel("DN Value")
        plt.plot(xesf, esf, 'g-', xesf, esf_smooth, 'r-')
        green_patch = mpatches.Patch(color='green', label...
='Raw ESF')
        red_patch = mpatches.Patch(color='red', label='...
Smooth ESF')
        plt.legend(handles=[green_patch, red_patch], loc...
=4)
        print("----- LISCIO 1")
        self.LSF_compute()
```

## B.4 Computation of Line Spread Function (LSF)

```
 def LSF_compute(self):
     diff_esf = abs(self.esf[1:] - self.esf[0:(self....
esf.shape[0] - 1)])
     diff_esf = np.append(0, diff_esf)
     lsf = diff_esf
     diff_esf_smooth = abs(self.esf_smooth[0:(self.esf...
.shape[0] - 1)] - self.esf_smooth[1:])
     diff_esf_smooth = np.append(0, diff_esf_smooth)
     lsf_smooth = diff_esf_smooth
     self.lsf = lsf
     self.lsf_smooth = lsf_smooth
     plt.subplot(4, 1, 3)
     plt.title("LSF Curve")
     plt.xlabel("pixel")
     plt.ylabel("DN Value")
     plt.plot(self.xesf, lsf, 'g-', self.xesf, ...
lsf_smooth, 'r-')
     green_patch = mpatches.Patch(color='green', label...
='Raw LSF')
     red_patch = mpatches.Patch(color='red', label='...
Smooth LSF')
     plt.legend(handles=[green_patch, red_patch])
     print("----- LISCIO 2")
     self.MTF_compute()
```

## B.5 Computation of Modulation Transfer Function (MTF) and MTF50

```
 def MTF_compute(self):
     mtf = np.absolute(np.fft.fft(self.lsf, 2048))
     mtf_smooth = np.absolute(np.fft.fft(self....
lsf_smooth, 2048))
     mtf_final = np.fft.fftshift(mtf)
     mtf_final_smooth = np.fft.fftshift(mtf_smooth)
     ax = plt.subplot(4, 1, 4)
     x_mtf_final = np.arange(0,1,1./127)
     mtf_final = mtf_final[1024:1151]/np.amax(...
mtf_final[1024:1151])
     mtf_final_smooth = mtf_final_smooth[1024:1151]/np...
.amax(mtf_final_smooth[1024:1151])

     i = 0
     for yval in mtf_final_smooth:
         i+= 1
         if yval > 0.49 and yval< 0.59:
             break
     xinterp = [x_mtf_final[i+1], x_mtf_final[i], ...
x_mtf_final[i-1]]
     yinterp = [mtf_final_smooth[i+1], ...
mtf_final_smooth[i], mtf_final_smooth[i-1]]
     mtf_value = np.interp(0.5, yinterp, xinterp)
     mtf50text = "MTF50: " + str(mtf_value)
     ax.text(0.35, 0.9, mtf50text, style='italic', ...
bbox={'facecolor': 'yellow', 'alpha': 0.5, 'pad': 10})
     plt.plot(x_mtf_final, mtf_final, 'g-', ...
x_mtf_final, mtf_final_smooth, 'r-')
     plt.xlabel("cycles/pixel")
     plt.ylabel("Modulation Factor")
     plt.title("MTF Curve")
     green_patch = mpatches.Patch(color='green', label...
='Raw MTF')
     red_patch = mpatches.Patch(color='red', label='...
Smooth MTF')
     plt.legend(handles=[green_patch, red_patch])
     print("Show result")
```

```
      global counter
      counter = counter + 1
      self.graphPath = 'MTFCalculator/tmp/graph'
      self.counterStr = str(counter)
      self.numPath = self.graphPath + self.counterStr
      self.finalPath = self.numPath + '.png'
      self.path = join(primary_external_storage_path(),...
  self.finalPath)
      print(self.path)
      fig = plt.gcf()
      fig.set_figheight(16)
      fig.set_figwidth(9)
      fig.savefig(self.path)
      self.mtf50 = mtf_value
      plt.clf()

  def getMtf50(self):
    return self.mtf50
```