NITTE EDUCATION TRUST | NMAM INSTITUTE OF TECHNOLOGY

PROJECT REPORT

on

Color Conversion using Image Processing

By

Sudarshan P- 4NM20MC109

Vinayak Naik - 4NM20MC117

Vybhav Hs - 4NM20MC119

Department of Master of Computer Applications NMAM INSTITUTE OF TECHNOLOGY, NITTE-574 110

2021-22

NMAM INSTITUTE OF TECHNOLOGY, NITTE-574110

Department of MCA

BONAFIDE CERTIFICATE

Certified that this project report **"Color Conversion Using Image Processing"** the bonafide work of **"Sudarshan P(4NM20MC109), Vinayak Naik(4NM20MC117), Vybhav hs(4NM20MC119)"** who presented the project under my supervision**.**

SIGNATURE

Head of the department

Dr. Surendra Shetty

Professor & Head

Department of MCA

NMAM.I.T., Nitte

SIGNATURE

Project Guide

Puneeth B R

Assistant Professor

Department of MCA

NMAM.I.T., Nitte

ABSTRACT

The use of color in image processing is motivated by two principal factors. First, color is a powerful descriptor that often simplifies object identification and extraction from a scene. Conversion of a color image into a grayscale image inclusive of salient features is a complicated process. The converted grayscale image may lose contrasts, sharpness, shadow, and structure of the color image. To preserve contrasts, sharpness, shadow, and structure of the color image a new algorithm has proposed. To convert the color image into grayscale image the new algorithm performs RGB approximation, reduction, and addition of chrominance and luminance. The grayscale images generated using the algorithm in the experiment confirms that the algorithm has preserved the salient features of the color image such as contrasts, sharpness, shadow, and image structure.

TABLE OF CONTENTS

1.INTRODUCTION

Digital image processing is an interesting field for its pictorial information and human interpretation. It has improved significantly in recent times and extended to various fields of science and technology. The purpose of a color model (also called color space is to facilitate the specification of colors in some standard, generally accepted way. In essence, a color model is a specification of a coordinate system a subspace within that system where each color model is a specification of a coordinate system and a subspace within that system where each color is represented by a single point. In terms of digital image processing, the hardware-oriented models most commonly used in practice are the RBG (red, green, blue) model for color monitors. The CMYK (cyan, magenta Yellow, Black) models for color printings and this (hue, saturation, intensity) model, which corresponds closely with the way humans describe and interpret color.

1.1 PROBLEM DEFINITION

In this project, the system is built in such a way that we should select an image first. The system does a lot of processes on the image and gives the user the accurate result. After when we provide the image as input to the system, it applies all the calculations on the input image. After it gives filtered image as result.

1.2 OBJECTIVES

The objective of our project is to create a system that needs only a algorithm to convert image into different types of images in UI. To help the user to perform some level of image processing. One application to perform all the process mentioned below.

- Adding Filters to Image.

1.3 CHALLENGES

- Compression-A modern trend in image storage and transmission is to use digital techniques.

- Enhancement-In enhancement, one aims to process images to improve their quality. An image may be of poor quality because its contrast is low, or it is noisy, or it is blurred, etc. Many algorithms have been devised to remove these degradations.
- Visualization-Commonly, visualization is considered as a part of computer graphics. The main task is to generate images.

1.3 LITERATURE SURVEY

[1] in his paper proposed a technique to pick out a threshold mechanically from a grey level bar graph has been derived from the point of view of discriminant analysis. This directly deals with the matter of evaluating the goodness of thresholds. associate optimum threshold (or set of thresholds) is chosen by the discriminant criterion; specifically, by maximising the discriminant measure alphabetic character G.Wyszecki., et al(1982).

[2] proposed a research work that describes regarding the color science ideas and strategies. The RGB show is used here to acknowledge the shading within the image. The RGB show could be a shading model that joins red, inexperienced associated blue lights in numerous approaches to create an assortment of hues. Brownrigg, D. R. K(1984).

[3] in his paper proposed a research work that provides the efficient answer for the matter of minimum error thresholding that comes below the idea of object and picture element level grey values being ordinarily distributed. This is applicable for mutithreshold choice. R. C. Gonzalez.,et al (1992) .

[4] in his paper Digital Image process discusses Image segmentation subdivides a picture into its constituent regions or objects. The amount of segmentation depends on the matter to be solved. Non-trivial image segmentation is one among the foremost tough tasks in image process. The accuracy of the segmentation determines the final word success or failure of a computerised analysis program. N. R. Pal., et al (1992).

[5]. B Thamotharan Ed al, proposed that the digital image processing concepts are done by different algorithms and highlighted noise and edge detection algorithm. He discussed another two concepts like mean and median filtering for radiographic images and compared them.

2. SOFTWARE REQUIREMENT SPECIFICATION

2.1 Functional Requirements:

It describes the functions a software must perform. A function is nothing but inputs these inputs which are the images collected from file system then its behavior which is including several of codes for the specific buttons like view, select a image and so on. After that the outputs will be saved in respective path in local storage. It can be a user interaction, or any other specific functionality which defines what function a system is likely to perform.

2.2 Non-functional Requirements:

These are basically the quality constraints that the system must satisfy according to the project contract. The priority or extent to which these factors are implemented varies from one project to other. They basically deal with issues like:

- Portability
- Security
- Maintainability
- Reliability
- Scalability
- Performance
- Reusability

2.3 HARDWARE REQUIREMENTS

• **Processor**: i5

• **Hard disk**: 5GB

• **Memory**: 1GB RAM

2.4 SOFTWARE REQUIREMENTS

• **Operating System**: Windows7 or Higher.

Python:

Python is a computer programming language often used to build websites and software, automate tasks, and conduct data analysis.

Tkinter:

The tkinter package (“Tk interface”) is the standard Python interface to the Tcl/Tk GUI toolkit. Both Tk and tkinter are available on most Unix platforms, including macOS, as well as on Windows systems.

Opencv:

OpenCV is a great tool for image processing and performing computer vision tasks.

RGB:

The RGB color model is an additive color model[1] in which the red, green, and blue primary colors of light are added together in various ways to reproduce a broad array of colors. The name of the model comes from the initials of the three additive primary colors, red, green, and blue. The main purpose of the RGB color model is for the sensing, representation, and display of images in electronic systems, such as televisions and computers, though it has also been used in conventional photography. Before the electronic age, the RGB color model already had a solid theory behind it, based in human perception of colors.

CMYK:

The CMYK model works by partially or entirely masking colors on a lighter, usually white, background. The ink reduces the light that would otherwise be reflected. Such a model is called subtractive because inks "subtract" the colors red, green and blue from white light. White light minus red leaves cyan, white light minus green leaves magenta, and white light minus blue leaves yellow.

HSI:

The HSI representation models the way different paints mix together to create color in the real world, with the lightness dimension resembling the varying amounts of black or white paint in the mixture (e.g. to create "light red", a red pigment can be mixed with white paint; this white

paint corresponds to a high "intensity" value in the HSI representation). Fully saturated colors are placed around a circle at a lightness value of ½, with a lightness value of 0 or 1 corresponding to fully black or white, respectively.

HSV:

HSV (for hue, saturation, value) also known as HSB, for hue, saturation, brightness) Meanwhile, the HSV representation models how colors appear under light. The difference between HSV is that a color with maximum lightness in HSL is pure white, but a color with maximum value/brightness in HSV is analogous to shining a white light on a colored object (e.g. shining a bright white light on a red object causes the object to still appear red, just brighter and more intense, while shining a dim light on a red object causes the object to appear darker and less bright).

Ycbcr:

Y′ luma component and C_B C_R are the blue-difference and red-difference chroma components. Y′ (with prime) is distinguished from Y, which is luminance, meaning that light intensity is nonlinearly encoded based on gamma corrected RGB primaries.

Binary:

A binary image is one that consists of pixels that can have one of exactly two colors, usually black and white. Binary images are also called bi-level or two-level, Pixel art made of two colors is often referred to as 1-Bit or 1bit.[2] This means that each pixel is stored as a single bit—i.e., a 0 or 1. The names black-and-white, B&W, monochrome or monochromatic are often used for this concept, but may also designate any images that have only one sample per pixel, such as grayscale images. In Photoshop parlance, a binary image is the same as an image in "Bitmap" mode.

YUV:

The Y′UV model defines a color space in terms of one luma component (Y′) and two chrominance components, called U (blue projection) and V (red projection) respectively. The Y′UV color model is used in the PAL composite color video (excluding PAL-N) standard. Previous black-and-white systems used only luma (Y′) information. Color information (U and V)

was added separately via a subcarrier so that a black-and-white receiver would still be able to receive and display a color picture transmission in the receiver's native black-and-white format.

Grayscale:

Grayscale images can be the result of measuring the intensity of light at each pixel according to a particular weighted combination of frequencies (or wavelengths), and in such cases they are monochromatic proper when only a single frequency (in practice, a narrow band of frequencies) is captured. The frequencies can in principle be from anywhere in the electromagnetic spectrum (e.g. infrared, visible light, ultraviolet, etc.

Edge:

In graph theory, an edge coloring of a graph is an assignment of "colors" to the edges of the graph so that no two incident edges have the same color. For example, the figure to the right shows an edge coloring of a graph by the colors red, blue, and green. Edge colorings are one of several different types of graph coloring. The edge-coloring problem asks whether it is possible to color the edges of a given graph using at most k different colors, for a given value of k, or with the fewest possible colors. The minimum required number of colors for the edges of a given graph is called the chromatic index of the graph. For example, the edges of the graph in the illustration can be colored by three colors but cannot be colored by two colors, so the graph shown has chromatic index three.

BGR:

BGR stands for Blue green red. Most often, an BGR color is stored in a structure or unsigned integer with Blue occupying the least significant "area" (a byte in 32-bit and 24-bit formats), Green the second least, and Red the third least. BGR is the same, except the order of areas is reversed. Red occupies the least significant area, Green the second (still), and Blue the third.

3. SYSTEM DESIGN

3.1 Detailed design:

Image collection is the first step in building the project. The image can be anything. It can be image etc. In this project Image is used as our dataset. A huge number of images needs to be collected to get the results accurately.

3.2 Image collection:

Image collection or data collection is the first step in building the project. The data can be anything. It can be text, image, audio etc. In this project Image is used as our dataset. A huge number of images needs to be collected to get the results accurately.

3.3 Image preprocessing:

Image pre-processing is the process in which some algorithms and operations are performed on the images to enhance the quality of the images. This is the one of the most essential process. Because the accuracy of the results depends on the how well the images are pre-processed. Pre-processing also helps in the further operations like training and testing.

3.4 Data Flow Diagram:

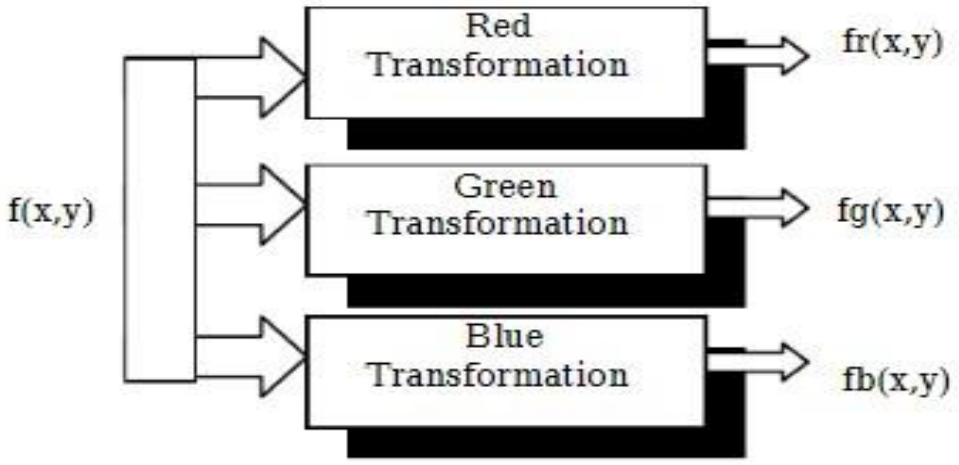
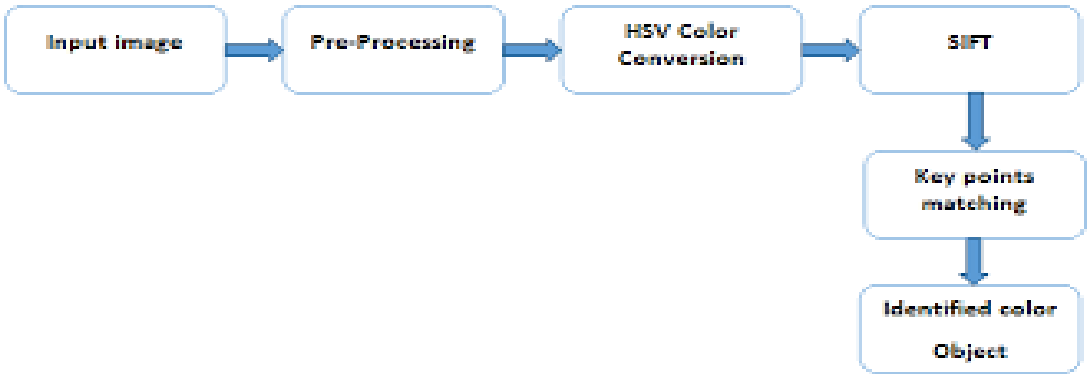


Figure (a). Data flow diagram

4. SYSTEM IMPLEMENTATION

4.1 MODULE DESCRIPTION:



Numpy: Images are stored and processed as numbers. These are taken as arrays. We use NumPy to deal with arrays.

CV2: Imported to use OpenCV for image processing.

Image Pre-processing:

After the image is collected, the image needs to pre-processed to enhance the quality of the data. The pre-processed data will ease the model to precise more precisely.

Displaying the processed image:

A detection applied image, filled with white colour image and Original images are also Displayed. If not the result is displayed as message “First select the image”.

5.2 Tools, Language Used:

To develop this project, Python has been used. Python has been used because it is very flexible language and has a lot of library support, which eases the work of the programmer.

5.TESTING

- Module Testing
- Integration testing

5.1 Module Testing:

Module testing is defined as a software testing type, which checks individual subprograms, subroutines, classes, or procedures in a program. Instead of testing whole software program at once, module testing recommends testing the smaller building blocks of the program. Module testing is largely a white box oriented. The objective of doing Module, testing is not to demonstrate proper functioning of the module but to demonstrate the presence of an error in the module. Instead of testing whole software program at once, module testing recommends testing the smaller building blocks of the program**.**

5.2 Integration testing:

Integration testing is defined as a type of testing where software modules are integrated logically and tested as a group. A typical software project consists of multiple software modules, coded by different programmers. The purpose of this level of testing is to expose defects in the interaction between these software modules when they are integrated. Integration Testing focuses on checking data communication amongst these modules. Integration testing works on the whole component, we can conclude as many unit tests make integration testing. And also a software is complete if whole integration testing is complete.

6.RESULTS

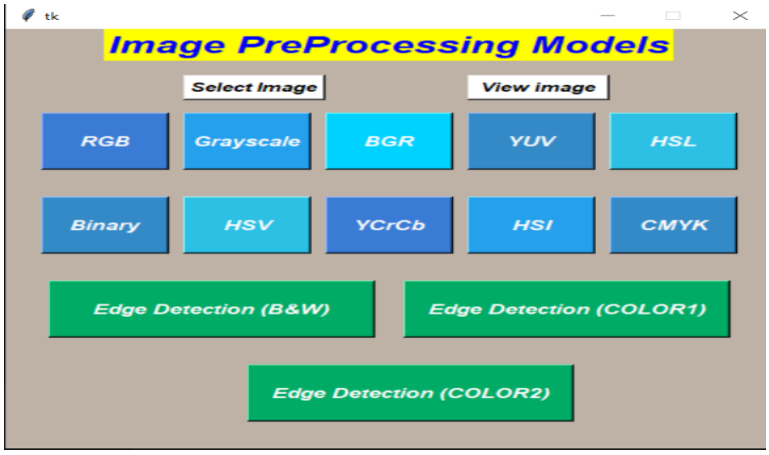


Fig1.Model interface

In this we'll be building a simple user interface using the Tkinter Python library. This user interface will allow us to click a button, triggering a file chooser dialog to select a file from disk. We'll then load the selected image using OpenCV, perform edge detection, and finally display both the original image and edge map in our GUI.

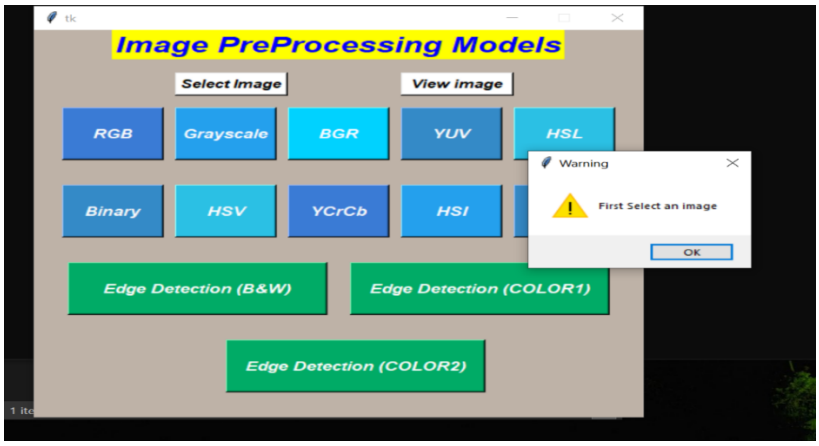


Fig2.Selecting an image

When first loaded, our application when contain only a button that allows us to load an image from disk. After clicking this button, we'll be allowed to navigate our file system and select an image to be processed. We'll then display the original image and converted image in the GUI.

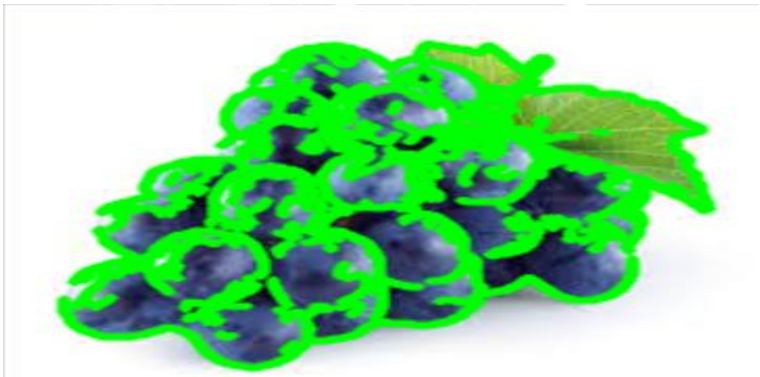


Fig3.Output of Edge detection (color1) Image

After our image has been selected, the edge map is computed using OpenCV, and both the original image and the edge image are added to our Tkinter GUI. Of course, we can repeat this process for different images as well.

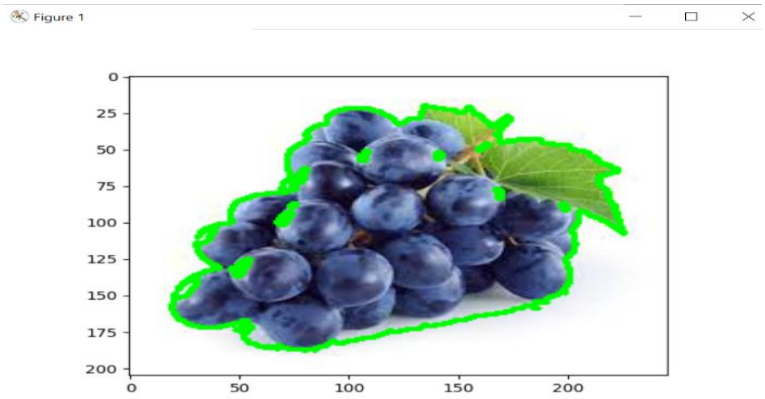


Fig4.Output of Edge detection(color2) Image

Loading an image, computing edges using the edge algorithm, and then displaying the result using Tkinter and OpenCV.

Fig5. Output of RGB Image

When the image file is read with the OpenCV function imread(), the order of colors is BGR (blue, green, red). On the other hand, in Pillow, the order of colors is assumed to be RGB (red, green, blue).Therefore, if you want to use both the Pillow function and the OpenCV function, you need to convert BGR and RGB.

Fig6.Output of HSV Image

The first thing we are going to do is importing the cv2 module. This module will make available the functions we need to both read the image and convert it to another color space. we simply need to call the imshow function. It receives as first input a string with the name to be assigned to the window that will show the image, and as second input the image.

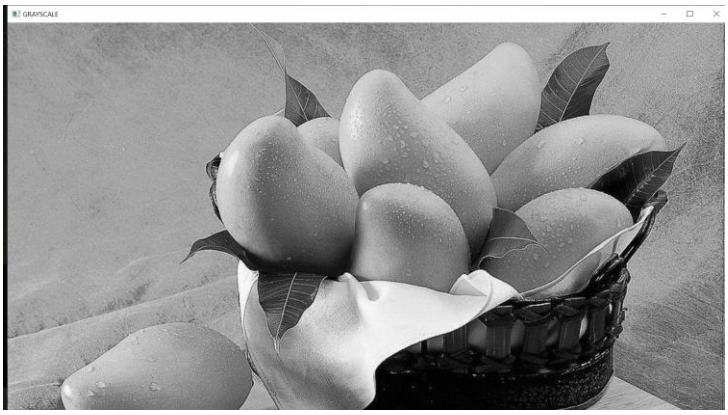


Fig7.Output of Grayscale image

Grayscaling is the process of converting an image from other color spaces e.g. RGB, CMYK, HSV, etc. to shades of gray. It varies between complete black and complete white. For example, In RGB images there are three color channels and has three dimensions while grayscale images are single-dimensional.

Fig8.Output of BGR image

BGR stands for Blue green red. Most often, an BGR color is stored in a structure or unsigned integer with Blue occupying the least significant "area" (a byte in 32-bit and 24-bit formats), Green the second least, and Red the third least. BGR is the same, except the order of areas is reversed. Red occupies the least significant area, Green the second (still), and Blue the third.

Fig9.Output of YUV image

The Y′UV model defines a color space in terms of one luma component (Y′) and two chrominance components, called U (blue projection) and V (red projection) respectively. The Y′UV color model is used in the PAL composite color video (excluding PAL-N) standard. Previous black-and-white systems used only luma (Y′) information.

Fig10.Output of HSL image

The HSL representation models the way different paints mix together to create colour in the real world, with the lightness dimension resembling the varying amounts of black or white paint in the mixture (e.g. to create "light red", a red pigment can be mixed with white paint; this white paint corresponds to a high "lightness" value in the HSL representation). Fully saturated colors are placed around a circle at a lightness value of ½, with a lightness value of 0 or 1 corresponding to fully black or white, respectively.

Fig11.Output of Binary image

Converting an image to black and white involves two steps.

1. Read the source image as grey scale image.
2. Convert the grey scale image to binary with a threshold of your choice.

In the following example, we will read a grey scale image using cv2.imread() and then apply cv2.threshold() function on the image array. There is no difference in converting a color image to black and white and grey scale image to black and white.

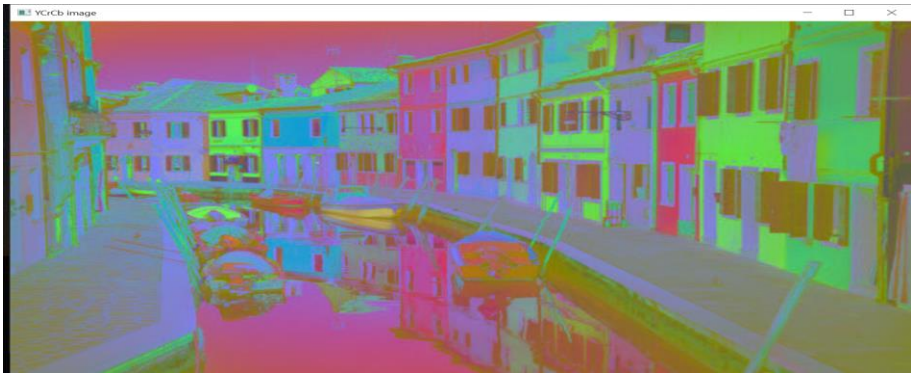


Fig12.Output of Ycbcr image

Important point about YCbCr is that Y component in this mode is non-linear.

Yet another color mode YCbCr uses a different numerical method to represent colors.

- Y stands for Luma component which is the light,
- Cb represents blue in relation to chroma (green) component,
- Cr represents red in relation to chroma (green) component.

Fig13.Output of HSI image

The HSI representation models the way different paints mix together to create color in the real world, with the lightness dimension resembling the varying amounts of black or white paint in the mixture.

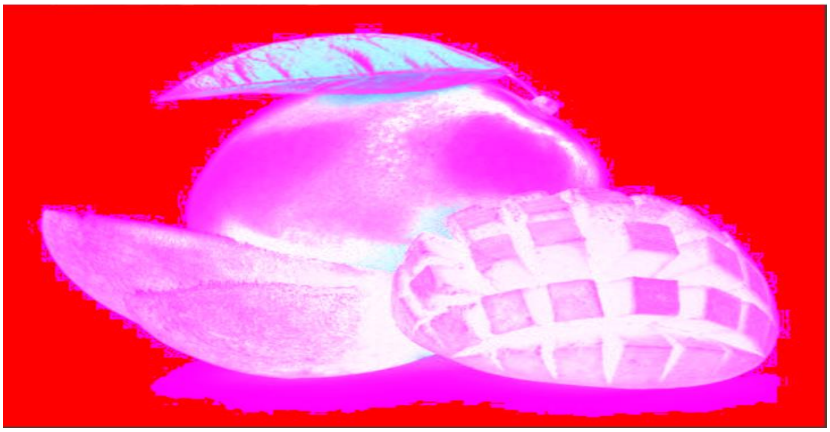


Fig14.Output of CMYK image

Cyan, magenta and yellow are the secondary colors of light and the primary colors of pigments. This means, if white light is shined on a surface coated with cyan pigment, no red light is reflected from it. Cyan subtracts red light from white light. Unlike the RGB color model, CMY is subtractive, meaning higher values are associated with darker colors rather than lighter ones.

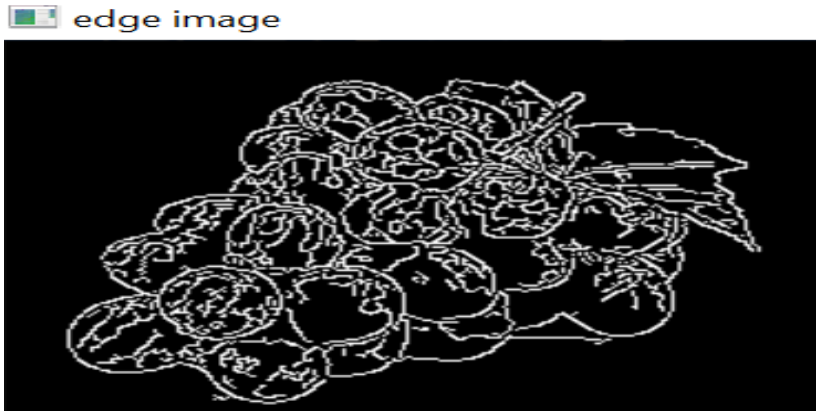


Fig15.Output of Edge detection(B&W) Image

Edge detection is an image-processing technique, which is used to identify the boundaries (edges) of objects, or regions within an image. Edges are among the most important features associated with images. We come to know of the underlying structure of an image through its edges. Computer vision processing pipelines therefore extensively use edge detection in application. Edges are characterized by sudden changes in pixel intensity. To detect edges, we need to go looking for such changes in the neighboring pixels. Come, let's explore the use of two important edge-detection algorithms available in OpenCV. Edge detection is used to find various boundaries/edges of various objects within a single image. There are multiple edge detection algorithms. The first step is to import all the modules needed namely OpenCV, numpy, and matplotlib. We would also be setting the style according to our preference. Before we detect an image we have to read the image in our program using the imread method which will take the path of the image as a parameter.To get the original colors we need to convert the colors to RGB format using the cvtColor function and apply it to the loaded image.

7.CONCLUSION AND FUTURE ENHANCEMENTS

The conversion of color image to grayscale image using the proposed algorithm uses approximation of RGB values using luminance RGB components approximated RGB reduced by three, added with chrominance value and average of these four value results perceptually and structurally good quality of grayscale images. First, the luminance and chrominance values are calculated. Further, the RGB values of the source color image reduced. Finally, the reduced RGB values have added chrominance values. The resulted grayscale images confirm that the luminance and chrominance properties and structure of the color images retained well in the grayscale image. The results confirm that the isoluminant images have handled as handled by other recent techniques. The proposed algorithm is helpful for different applications where good quality of grayscale image is highly required. The proposed algorithm results best quality of grayscale images using RGB reduction and chrominance addition in a short amount of time.

REFERENCES

1. Gonzalez RC, Woods RE (1992) Digital Image Processing, Reading. AddisonWesley, Massachusetts.

2. Van der HF (1994) Image Based Measurement Systems. Wiley Online.

3. Castleman KR (1996) Digital Image Processing. Englewood Cliffs, PrenticeHall, New Jersey.

4. Susstrunk, Sabine, Robert Buckley, Steve Swen (1999) Standard RGB color spaces. Color and Imaging Conference, Society for Imaging Science and Technology.

5. Li, Jian-Feng, Kaun-Quan Wang, David Zhang (2002) A new equation of saturation in RGB-to-HSI conversion for more rapidity of computing. Machine Learning and Cybernetics 3: 1493-1497.

6. Granier, Xavier, Wolfgang Heidrich (2003) A simple layered RGB BRDF model. Graphical Models 65: 171-184.

7. Wen, Che-Yen, Chun-Ming Chou (2004) Color Image Models and Its Applications to Document Examination. Forensic Science Journal 3: 23-32.

8. Harmeet Kaur kelda (2014) A Review: Color Models in Image Processing. Int J Computer Technology and Applications 5: 319-322.