using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.IO;

using System.Text;

using System.Windows.Forms;

using System.Drawing.Imaging;

namespace photoeditor

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

Image file;

Boolean opened = false;

OpenFileDialog openFileDialog1 = new OpenFileDialog();

void openImage()

{

DialogResult dr = openFileDialog1.ShowDialog();

if (dr == DialogResult.OK)

{

file = Image.FromFile(openFileDialog1.FileName);

pictureBox1.Image = file;

opened = true;

}

}

void hue()

{

float changered = red.Value \* 0.1f;

float changegreen = green.Value \* 0.1f;

float changeblue = blue.Value \* 0.1f;

red.Text = changered.ToString();

green.Text = changeblue.ToString();

blue.Text = changegreen.ToString();

reload();

if (!opened)

{

}

else

{

Image img = pictureBox1.Image; // storing image into img variable of image type from picturebox1

Bitmap bmpInverted = new Bitmap(img.Width, img.Height); /\* creating a bitmap of the height of imported picture in picturebox which consists of the pixel data for a graphics image

and its attributes. A Bitmap is an object used to work with images defined by pixel data.\*/

ImageAttributes ia = new ImageAttributes(); //creating an object of imageattribute ia to change the attribute of images

ColorMatrix cmPicture = new ColorMatrix(new float[][] // now creating the color matrix object to change the colors or apply image filter on image

{

new float[]{1+changered, 0, 0, 0, 0},

new float[]{0, 1+changegreen, 0, 0, 0},

new float[]{0, 0, 1+changeblue, 0, 0},

new float[]{0, 0, 0, 1, 0},

new float[]{0, 0, 0, 0, 1}

});

ia.SetColorMatrix(cmPicture); //pass the color matrix to imageattribute object ia

Graphics g = Graphics.FromImage(bmpInverted); /\*create a new object of graphics named g, ; Create graphics object for alteration.

Graphics newGraphics = Graphics.FromImage(imageFile); is the format of loading image into graphics for alteration\*/

g.DrawImage(img, new Rectangle(0, 0, img.Width, img.Height), 0, 0, img.Width, img.Height, GraphicsUnit.Pixel, ia);

/\* g.drawimage(image, new rectangle(location of rectangle axix-x, location axis-y, width of rectangle, height of rectangle),

location of image in rectangle x-axis, location of image in rectangle y-axis, width of image, height of image,

format of graphics unit,provide the image attributes \*/

g.Dispose(); //Releases all resources used by this Graphics.

pictureBox1.Image = bmpInverted;

}

}

void saveImage()

{

if (opened)

{

SaveFileDialog sfd = new SaveFileDialog(); // create a new save file dialog object

sfd.Filter = "Images|\*.png;\*.bmp;\*.jpg";

ImageFormat format = ImageFormat.Png;// you want to store it in by default format

if (sfd.ShowDialog() == System.Windows.Forms.DialogResult.OK)

{

string ext = Path.GetExtension(sfd.FileName);

switch (ext)

{

case ".jpg":

format = ImageFormat.Jpeg;

break;

case ".bmp":

format = ImageFormat.Bmp;

break;

}

pictureBox1.Image.Save(sfd.FileName, format);

}

}

else { MessageBox.Show("No image loaded, first upload image "); }

}

/\*

-----------------------------------------------------------Color Matrix Combinations----------------------------------------------------

R G B A W R G B A W R G B A W

R [1 0 0 0 0] R [c 0 0 0 0] R [sr+s sr sr 0 0]

G [0 1 0 0 0] G [0 c 0 0 0] G [ sg sg+s sg 0 0]

B [0 0 1 0 0] X B [0 0 c 0 0] X B [ sb sb sb+s 0 0]

A [0 0 0 1 0] A [0 0 0 1 0] A [ 0 0 0 1 0]

W [b b b 0 1] W [t t t 0 1] W [ 0 0 0 0 1]

Brightness Matrix Contrast Matrix Saturation Matrix

R G B A W

R [c(sr+s) c(sr) c(sr) 0 0 ]

G [ c(sg) c(sg+s) c(sg) 0 0 ]

===> B [ c(sb) c(sb) c(sb+s) 0 0 ]

A [ 0 0 0 1 0 ]

W [ t+b t+b t+b 0 1 ]

Transformation Matrix

\*/

//-----------------------------------------------------------------------Gray Scale Filter ------------------------------------------------------------------------------------

void grayscale()

{

if (!opened)

{

MessageBox.Show("Open an Image then apply changes");

}

else

{

Image img = pictureBox1.Image; // storing image into img variable of image type from picturebox1

Bitmap bmpInverted = new Bitmap(img.Width, img.Height); /\* creating a bitmap of the height of imported picture in picturebox which consists of the pixel data for a graphics image

and its attributes. A Bitmap is an object used to work with images defined by pixel data.\*/

ImageAttributes ia = new ImageAttributes(); //creating an object of imageattribute ia to change the attribute of images

ColorMatrix cmPicture = new ColorMatrix(new float[][] // now creating the color matrix object to change the colors or apply image filter on image

{

new float[]{0.299f, 0.299f, 0.299f, 0, 0},

new float[]{0.587f, 0.587f, 0.587f, 0, 0},

new float[]{0.114f, 0.114f, 0.114f, 0, 0},

new float[]{0, 0, 0, 1, 0},

new float[]{0, 0, 0, 0, 0}

});

ia.SetColorMatrix(cmPicture); //pass the color matrix to imageattribute object ia

Graphics g = Graphics.FromImage(bmpInverted); /\*create a new object of graphics named g, ; Create graphics object for alteration.

Graphics newGraphics = Graphics.FromImage(imageFile); is the format of loading image into graphics for alteration\*/

g.DrawImage(img, new Rectangle(0, 0, img.Width, img.Height), 0, 0, img.Width, img.Height, GraphicsUnit.Pixel, ia);

/\* g.drawimage(image, new rectangle(location of rectangle axix-x, location axis-y, width of rectangle, height of rectangle),

location of image in rectangle x-axis, location of image in rectangle y-axis, width of image, height of image,

format of graphics unit,provide the image attributes \*/

g.Dispose(); //Releases all resources used by this Graphics.

pictureBox1.Image = bmpInverted;

}

}

void fog()

{

if (!opened)

{

MessageBox.Show("Open an Image then apply changes");

}

else

{

Image img = pictureBox1.Image; // storing image into img variable of image type from picturebox1

Bitmap bmpInverted = new Bitmap(img.Width, img.Height); /\* creating a bitmap of the height of imported picture in picturebox which consists of the pixel data for a graphics image

and its attributes. A Bitmap is an object used to work with images defined by pixel data.\*/

ImageAttributes ia = new ImageAttributes(); //creating an object of imageattribute ia to change the attribute of images

ColorMatrix cmPicture = new ColorMatrix(new float[][] // now creating the color matrix object to change the colors or apply image filter on image

{

new float[]{1+0.3f, 0, 0, 0, 0},

new float[]{0, 1+0.7f, 0, 0, 0},

new float[]{0, 0, 1+1.3f, 0, 0},

new float[]{0, 0, 0, 1, 0},

new float[]{0, 0, 0, 0, 1}

});

ia.SetColorMatrix(cmPicture); //pass the color matrix to imageattribute object ia

Graphics g = Graphics.FromImage(bmpInverted); /\*create a new object of graphics named g, ; Create graphics object for alteration.

Graphics newGraphics = Graphics.FromImage(imageFile); is the format of loading image into graphics for alteration\*/

g.DrawImage(img, new Rectangle(0, 0, img.Width, img.Height), 0, 0, img.Width, img.Height, GraphicsUnit.Pixel, ia);

/\* g.drawimage(image, new rectangle(location of rectangle axix-x, location axis-y, width of rectangle, height of rectangle),

location of image in rectangle x-axis, location of image in rectangle y-axis, width of image, height of image,

format of graphics unit,provide the image attributes \*/

g.Dispose(); //Releases all resources used by this Graphics.

pictureBox1.Image = bmpInverted;

}

}

void flash()

{

if (!opened)

{

MessageBox.Show("Open an Image then apply changes");

}

else

{

Image img = pictureBox1.Image; // storing image into img variable of image type from picturebox1

Bitmap bmpInverted = new Bitmap(img.Width, img.Height); /\* creating a bitmap of the height of imported picture in picturebox which consists of the pixel data for a graphics image

and its attributes. A Bitmap is an object used to work with images defined by pixel data.\*/

ImageAttributes ia = new ImageAttributes(); //creating an object of imageattribute ia to change the attribute of images

ColorMatrix cmPicture = new ColorMatrix(new float[][] // now creating the color matrix object to change the colors or apply image filter on image

{

new float[]{1+0.9f, 0, 0, 0, 0},

new float[]{0, 1+1.5f, 0, 0, 0},

new float[]{0, 0, 1+1.3f, 0, 0},

new float[]{0, 0, 0, 1, 0},

new float[]{0, 0, 0, 0, 1}

});

ia.SetColorMatrix(cmPicture); //pass the color matrix to imageattribute object ia

Graphics g = Graphics.FromImage(bmpInverted); /\*create a new object of graphics named g, ; Create graphics object for alteration.

Graphics newGraphics = Graphics.FromImage(imageFile); is the format of loading image into graphics for alteration\*/

g.DrawImage(img, new Rectangle(0, 0, img.Width, img.Height), 0, 0, img.Width, img.Height, GraphicsUnit.Pixel, ia);

/\* g.drawimage(image, new rectangle(location of rectangle axix-x, location axis-y, width of rectangle, height of rectangle),

location of image in rectangle x-axis, location of image in rectangle y-axis, width of image, height of image,

format of graphics unit,provide the image attributes \*/

g.Dispose(); //Releases all resources used by this Graphics.

pictureBox1.Image = bmpInverted;

}

}

void Frozen()

{

if (!opened)

{

MessageBox.Show("Open an Image then apply changes");

}

else

{

Image img = pictureBox1.Image; // storing image into img variable of image type from picturebox1

Bitmap bmpInverted = new Bitmap(img.Width, img.Height); /\* creating a bitmap of the height of imported picture in picturebox which consists of the pixel data for a graphics image

and its attributes. A Bitmap is an object used to work with images defined by pixel data.\*/

ImageAttributes ia = new ImageAttributes(); //creating an object of imageattribute ia to change the attribute of images

ColorMatrix cmPicture = new ColorMatrix(new float[][] // now creating the color matrix object to change the colors or apply image filter on image

{

new float[]{1+0.3f, 0, 0, 0, 0},

new float[]{0, 1+0f, 0, 0, 0},

new float[]{0, 0, 1+5f, 0, 0},

new float[]{0, 0, 0, 1, 0},

new float[]{0, 0, 0, 0, 1}

});

ia.SetColorMatrix(cmPicture); //pass the color matrix to imageattribute object ia

Graphics g = Graphics.FromImage(bmpInverted); /\*create a new object of graphics named g, ; Create graphics object for alteration.

Graphics newGraphics = Graphics.FromImage(imageFile); is the format of loading image into graphics for alteration\*/

g.DrawImage(img, new Rectangle(0, 0, img.Width, img.Height), 0, 0, img.Width, img.Height, GraphicsUnit.Pixel, ia);

/\* g.drawimage(image, new rectangle(location of rectangle axix-x, location axis-y, width of rectangle, height of rectangle),

location of image in rectangle x-axis, location of image in rectangle y-axis, width of image, height of image,

format of graphics unit,provide the image attributes \*/

g.Dispose(); //Releases all resources used by this Graphics.

pictureBox1.Image = bmpInverted;

}

}

//-----------------------------------------------------------------------RED Filter ------------------------------------------------------------------------------------

void redscale()

{

if (!opened)

{

MessageBox.Show("Open an Image then apply changes");

}

else

{

Image img = pictureBox1.Image; // storing image into img variable of image type from picturebox1

Bitmap bmpInverted = new Bitmap(img.Width, img.Height); /\* creating a bitmap of the height of imported picture in picturebox which consists of the pixel data for a graphics image

and its attributes. A Bitmap is an object used to work with images defined by pixel data.\*/

ImageAttributes ia = new ImageAttributes(); //creating an object of imageattribute ia to change the attribute of images

ColorMatrix cmPicture = new ColorMatrix(new float[][] // now creating the color matrix object to change the colors or apply image filter on image

{

new float[]{.393f, .349f, .272f, 0, 0},

new float[]{.769f, .686f, .534f, 0, 0},

new float[]{.189f, .168f, .131f, 0, 0},

new float[]{0, 0, 0, 1, 0},

new float[]{0, 0, 0, 0, 1}

});

ia.SetColorMatrix(cmPicture); //pass the color matrix to imageattribute object ia

Graphics g = Graphics.FromImage(bmpInverted); /\*create a new object of graphics named g, ; Create graphics object for alteration.

Graphics newGraphics = Graphics.FromImage(imageFile); is the format of loading image into graphics for alteration\*/

g.DrawImage(img, new Rectangle(0, 0, img.Width, img.Height), 0, 0, img.Width, img.Height, GraphicsUnit.Pixel, ia);

/\* g.drawimage(image, new rectangle(location of rectangle axix-x, location axis-y, width of rectangle, height of rectangle),

location of image in rectangle x-axis, location of image in rectangle y-axis, width of image, height of image,

format of graphics unit,provide the image attributes \*/

g.Dispose(); //Releases all resources used by this Graphics.

pictureBox1.Image = bmpInverted;

}

}

void filter1()

{

if (!opened)

{

MessageBox.Show("Open an Image then apply changes");

}

else

{

Image img = pictureBox1.Image; // storing image into img variable of image type from picturebox1

Bitmap bmpInverted = new Bitmap(img.Width, img.Height); /\* creating a bitmap of the height of imported picture in picturebox which consists of the pixel data for a graphics image

and its attributes. A Bitmap is an object used to work with images defined by pixel data.\*/

ImageAttributes ia = new ImageAttributes(); //creating an object of imageattribute ia to change the attribute of images

ColorMatrix cmPicture = new ColorMatrix(new float[][] // now creating the color matrix object to change the colors or apply image filter on image

{

new float[]{.393f, .349f, .272f+1.3f, 0, 0},

new float[]{.769f, .686f+0.5f, .534f, 0, 0},

new float[]{.189f+2.3f, .168f, .131f, 0, 0},

new float[]{0, 0, 0, 1, 0},

new float[]{0, 0, 0, 0, 1}

});

ia.SetColorMatrix(cmPicture); //pass the color matrix to imageattribute object ia

Graphics g = Graphics.FromImage(bmpInverted); /\*create a new object of graphics named g, ; Create graphics object for alteration.

Graphics newGraphics = Graphics.FromImage(imageFile); is the format of loading image into graphics for alteration\*/

g.DrawImage(img, new Rectangle(0, 0, img.Width, img.Height), 0, 0, img.Width, img.Height, GraphicsUnit.Pixel, ia);

/\* g.drawimage(image, new rectangle(location of rectangle axix-x, location axis-y, width of rectangle, height of rectangle),

location of image in rectangle x-axis, location of image in rectangle y-axis, width of image, height of image,

format of graphics unit,provide the image attributes \*/

g.Dispose(); //Releases all resources used by this Graphics.

pictureBox1.Image = bmpInverted;

}

}

void filter2()

{

if (!opened)

{

MessageBox.Show("Open an Image then apply changes");

}

else

{

Image img = pictureBox1.Image; // storing image into img variable of image type from picturebox1

Bitmap bmpInverted = new Bitmap(img.Width, img.Height); /\* creating a bitmap of the height of imported picture in picturebox which consists of the pixel data for a graphics image

and its attributes. A Bitmap is an object used to work with images defined by pixel data.\*/

ImageAttributes ia = new ImageAttributes(); //creating an object of imageattribute ia to change the attribute of images

ColorMatrix cmPicture = new ColorMatrix(new float[][] // now creating the color matrix object to change the colors or apply image filter on image

{

new float[]{.393f, .349f+0.5f, .272f, 0, 0},

new float[]{.769f+0.3f, .686f, .534f, 0, 0},

new float[]{.189f, .168f, .131f+0.5f, 0, 0},

new float[]{0, 0, 0, 1, 0},

new float[]{0, 0, 0, 0, 1}

});

ia.SetColorMatrix(cmPicture); //pass the color matrix to imageattribute object ia

Graphics g = Graphics.FromImage(bmpInverted); /\*create a new object of graphics named g, ; Create graphics object for alteration.

Graphics newGraphics = Graphics.FromImage(imageFile); is the format of loading image into graphics for alteration\*/

g.DrawImage(img, new Rectangle(0, 0, img.Width, img.Height), 0, 0, img.Width, img.Height, GraphicsUnit.Pixel, ia);

/\* g.drawimage(image, new rectangle(location of rectangle axix-x, location axis-y, width of rectangle, height of rectangle),

location of image in rectangle x-axis, location of image in rectangle y-axis, width of image, height of image,

format of graphics unit,provide the image attributes \*/

g.Dispose(); //Releases all resources used by this Graphics.

pictureBox1.Image = bmpInverted;

}

}

void filter3()

{

if (!opened)

{

MessageBox.Show("Open an Image then apply changes");

}

else

{

Image img = pictureBox1.Image; // storing image into img variable of image type from picturebox1

Bitmap bmpInverted = new Bitmap(img.Width, img.Height); /\* creating a bitmap of the height of imported picture in picturebox which consists of the pixel data for a graphics image

and its attributes. A Bitmap is an object used to work with images defined by pixel data.\*/

ImageAttributes ia = new ImageAttributes(); //creating an object of imageattribute ia to change the attribute of images

ColorMatrix cmPicture = new ColorMatrix(new float[][] // now creating the color matrix object to change the colors or apply image filter on image

{

new float[]{.393f+0.3f, .349f, .272f, 0, 0},

new float[]{.769f, .686f+0.2f, .534f, 0, 0},

new float[]{.189f, .168f, .131f+0.9f, 0, 0},

new float[]{0, 0, 0, 1, 0},

new float[]{0, 0, 0, 0, 1}

});

ia.SetColorMatrix(cmPicture); //pass the color matrix to imageattribute object ia

Graphics g = Graphics.FromImage(bmpInverted); /\*create a new object of graphics named g, ; Create graphics object for alteration.

Graphics newGraphics = Graphics.FromImage(imageFile); is the format of loading image into graphics for alteration\*/

g.DrawImage(img, new Rectangle(0, 0, img.Width, img.Height), 0, 0, img.Width, img.Height, GraphicsUnit.Pixel, ia);

/\* g.drawimage(image, new rectangle(location of rectangle axix-x, location axis-y, width of rectangle, height of rectangle),

location of image in rectangle x-axis, location of image in rectangle y-axis, width of image, height of image,

format of graphics unit,provide the image attributes \*/

g.Dispose(); //Releases all resources used by this Graphics.

pictureBox1.Image = bmpInverted;

}

}

//-----------------------------------------------------------------------RED Filter ------------------------------------------------------------------------------------

void Winter()

{

if (!opened)

{

MessageBox.Show("Open an Image then apply changes");

}

else

{

Image img = pictureBox1.Image; // storing image into img variable of image type from picturebox1

Bitmap bmpInverted = new Bitmap(img.Width, img.Height); /\* creating a bitmap of the height of imported picture in picturebox which consists of the pixel data for a graphics image

and its attributes. A Bitmap is an object used to work with images defined by pixel data.\*/

ImageAttributes ia = new ImageAttributes(); //creating an object of imageattribute ia to change the attribute of images

ColorMatrix cmPicture = new ColorMatrix(new float[][] // now creating the color matrix object to change the colors or apply image filter on image

{

new float[]{1,0,0,0,0},

new float[]{0,1,0,0,0},

new float[]{0,0,1,0,0},

new float[]{0, 0, 0, 1, 0},

new float[]{0, 0, 1, 0, 1}

});

ia.SetColorMatrix(cmPicture); //pass the color matrix to imageattribute object ia

Graphics g = Graphics.FromImage(bmpInverted); /\*create a new object of graphics named g, ; Create graphics object for alteration.

Graphics newGraphics = Graphics.FromImage(imageFile); is the format of loading image into graphics for alteration\*/

g.DrawImage(img, new Rectangle(0, 0, img.Width, img.Height), 0, 0, img.Width, img.Height, GraphicsUnit.Pixel, ia);

/\* g.drawimage(image, new rectangle(location of rectangle axix-x, location axis-y, width of rectangle, height of rectangle),

location of image in rectangle x-axis, location of image in rectangle y-axis, width of image, height of image,

format of graphics unit,provide the image attributes \*/

g.Dispose(); //Releases all resources used by this Graphics.

pictureBox1.Image = bmpInverted;

}

}

//it simply reload the image so all previous effects removed..

void reload()

{

if (!opened)

{

// MessageBox.Show("Open an Image then apply changes");

}

else

{

if (opened)

{

file = Image.FromFile(openFileDialog1.FileName);

pictureBox1.Image = file;

opened = true;

}

}

}

void brightness()

{

float changeb = trackBar1.Value \* 0.1f;

float changec = trackBar2.Value \* 0.1f;

float changes = trackBar3.Value \* 0.1f;

// float changealpha = trackBar3.Value \* 0.1f;

// float changep = trackBar3.Value \* 0.1f;

trackBar1.Text = changeb.ToString();

trackBar2.Text = changec.ToString();

trackBar3.Text = changes.ToString();

reload();

if (!opened)

{

}

else

{

System.Drawing.Image img = pictureBox1.Image; // storing image into img variable of image type from picturebox1

Bitmap bmpInverted = new Bitmap(img.Width, img.Height); /\* creating a bitmap of the height of imported picture in picturebox which consists of the pixel data for a graphics image

and its attributes. A Bitmap is an object used to work with images defined by pixel data.\*/

ImageAttributes ia = new ImageAttributes(); //creating an object of imageattribute ia to change the attribute of images

ColorMatrix cmPicture = new ColorMatrix(new float[][] // now creating the color matrix object to change the colors or apply image filter on image

{

new float[]{1, 0, 0, 0, 0},

new float[]{0, 1, 0, 0, 0},

new float[]{0, 0, 1, 0, 0},

new float[]{0, 0, 0, 1, 0},

new float[]{0+changeb, 0+changeb, 0+changeb, 0, 1}

});

ia.SetColorMatrix(cmPicture); //pass the color matrix to imageattribute object ia

Graphics g = Graphics.FromImage(bmpInverted); /\*create a new object of graphics named g, ; Create graphics object for alteration.

Graphics newGraphics = Graphics.FromImage(imageFile); is the format of loading image into graphics for alteration\*/

g.DrawImage(img, new Rectangle(0, 0, img.Width, img.Height), 0, 0, img.Width, img.Height, GraphicsUnit.Pixel, ia);

/\* g.drawimage(image, new rectangle(location of rectangle axix-x, location axis-y, width of rectangle, height of rectangle),

location of image in rectangle x-axis, location of image in rectangle y-axis, width of image, height of image,

format of graphics unit,provide the image attributes \*/

g.Dispose(); //Releases all resources used by this Graphics.

pictureBox1.Image = bmpInverted;

}

}

void contrast()

{

float changeb = trackBar1.Value \* 0.1f;

float changec = trackBar2.Value \* 0.1f;

float changes = trackBar3.Value \* 0.1f;

float t = 0;

// float changealpha = trackBar3.Value \* 0.1f;

// float changep = trackBar3.Value \* 0.1f;

trackBar1.Text = changeb.ToString();

trackBar2.Text = changec.ToString();

trackBar3.Text = changes.ToString();

reload();

if (!opened)

{

}

else

{

System.Drawing.Image img = pictureBox1.Image; // storing image into img variable of image type from picturebox1

Bitmap bmpInverted = new Bitmap(img.Width, img.Height); /\* creating a bitmap of the height of imported picture in picturebox which consists of the pixel data for a graphics image

and its attributes. A Bitmap is an object used to work with images defined by pixel data.\*/

ImageAttributes ia = new ImageAttributes(); //creating an object of imageattribute ia to change the attribute of images

ColorMatrix cmPicture = new ColorMatrix(new float[][] // now creating the color matrix object to change the colors or apply image filter on image

{

new float[]{1+changec, 0, 0, 0, 0},

new float[]{0, 1+changec, 0, 0, 0},

new float[]{0, 0, 1+changec, 0, 0},

new float[]{0, 0, 0, 1, 0},

new float[]{t, t, t, 0, 1}

});

if (changec == 0)

t = (1f - changec) / 2f;

ia.SetColorMatrix(cmPicture); //pass the color matrix to imageattribute object ia

Graphics g = Graphics.FromImage(bmpInverted); /\*create a new object of graphics named g, ; Create graphics object for alteration.

Graphics newGraphics = Graphics.FromImage(imageFile); is the format of loading image into graphics for alteration\*/

g.DrawImage(img, new Rectangle(0, 0, img.Width, img.Height), 0, 0, img.Width, img.Height, GraphicsUnit.Pixel, ia);

/\* g.drawimage(image, new rectangle(location of rectangle axix-x, location axis-y, width of rectangle, height of rectangle),

location of image in rectangle x-axis, location of image in rectangle y-axis, width of image, height of image,

format of graphics unit,provide the image attributes \*/

g.Dispose(); //Releases all resources used by this Graphics.

pictureBox1.Image = bmpInverted;

}

}

void saturation()

{

float changeb = trackBar1.Value \* 0.1f;

float changec = trackBar2.Value \* 0.1f;

float changes = trackBar3.Value \* 0.1f;

// float changealpha = trackBar3.Value \* 0.1f;

// float changep = trackBar3.Value \* 0.1f;

float lumR = 0.3086f;

float lumG = 0.6094f;

float lumB = 0.0820f;

float sr = (1 - changes) \* lumR;

float sg = (1 - changes) \* lumG;

float sb = (1 - changes) \* lumB;

trackBar1.Text = changeb.ToString();

trackBar2.Text = changec.ToString();

trackBar3.Text = changes.ToString();

reload();

if (!opened)

{

}

else

{

System.Drawing.Image img = pictureBox1.Image; // storing image into img variable of image type from picturebox1

Bitmap bmpInverted = new Bitmap(img.Width, img.Height); /\* creating a bitmap of the height of imported picture in picturebox which consists of the pixel data for a graphics image

and its attributes. A Bitmap is an object used to work with images defined by pixel data.\*/

ImageAttributes ia = new ImageAttributes(); //creating an object of imageattribute ia to change the attribute of images

ColorMatrix cmPicture = new ColorMatrix(new float[][] // now creating the color matrix object to change the colors or apply image filter on image

{

new float[]{sr+changes, sr, sr, 0, 0},

new float[]{sg, sg+changes, sg, 0, 0},

new float[]{sb, sb, sb+changes, 0, 0},

new float[]{0, 0, 0, 1, 0},

new float[]{0, 0, 0, 0, 1}

});

ia.SetColorMatrix(cmPicture); //pass the color matrix to imageattribute object ia

Graphics g = Graphics.FromImage(bmpInverted); /\*create a new object of graphics named g, ; Create graphics object for alteration.

Graphics newGraphics = Graphics.FromImage(imageFile); is the format of loading image into graphics for alteration\*/

g.DrawImage(img, new Rectangle(0, 0, img.Width, img.Height), 0, 0, img.Width, img.Height, GraphicsUnit.Pixel, ia);

/\* g.drawimage(image, new rectangle(location of rectangle axix-x, location axis-y, width of rectangle, height of rectangle),

location of image in rectangle x-axis, location of image in rectangle y-axis, width of image, height of image,

format of graphics unit,provide the image attributes \*/

g.Dispose(); //Releases all resources used by this Graphics.

pictureBox1.Image = bmpInverted;

}

}

//--------------------------------------------------------------

private void button7\_Click(object sender, EventArgs e)//flash

{

reload();

flash();

}

private void button3\_Click(object sender, EventArgs e)//sepia

{

reload();

redscale();

}

private void button4\_Click(object sender, EventArgs e)//artistic

{

reload();

Winter();

}

private void button5\_Click(object sender, EventArgs e)//gray

{

reload();

grayscale();

}

private void button1\_Click(object sender, EventArgs e)//openimage

{

openImage();

}

private void button2\_Click(object sender, EventArgs e)//saveimage

{

saveImage();

}

private void button12\_Click(object sender, EventArgs e)//none

{

green.Value = 0;

red.Value = 0;

blue.Value = 0;

green.Text = "0";

reload();

}

private void green\_Scroll(object sender, EventArgs e)

{

hue();

}

private void red\_Scroll(object sender, EventArgs e)

{

hue();

}

private void blue\_Scroll(object sender, EventArgs e)

{

hue();

}

private void button6\_Click(object sender, EventArgs e)//spike

{

reload();

fog();

}

private void button8\_Click(object sender, EventArgs e)//frozen

{

reload();

Frozen();

}

private void button9\_Click(object sender, EventArgs e)//suji

{

reload();

// redscale();

// Winter();

filter2();

}

private void button10\_Click(object sender, EventArgs e)//dramatic

{

reload();

filter3();

}

private void button11\_Click(object sender, EventArgs e)//kakao

{

reload();

filter1();

}

private void trackBar1\_Scroll(object sender, EventArgs e)

{

brightness();

}

private void trackBar2\_Scroll(object sender, EventArgs e)

{

contrast();

}

private void trackBar3\_Scroll(object sender, EventArgs e)

{

saturation();

}

}

}