using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Image\_processing

{

public partial class Form1 : Form

{

Bitmap image;

public Form1()

{

InitializeComponent();

}

private void Form1\_Load(object sender, EventArgs e)

{

}

private void открытьToolStripMenuItem\_Click(object sender, EventArgs e)

{

OpenFileDialog dialog = new OpenFileDialog();

dialog.Filter = "Image files | \*.png; \*.jpg; \*.bmp; | All Files (\*.\*) | \*.\*";

if (dialog.ShowDialog() == DialogResult.OK)

{

image = new Bitmap(dialog.FileName);

pictureBox1.Image = image;

pictureBox1.Refresh();

}

}

private void сохранитьToolStripMenuItem\_Click(object sender, EventArgs e)

{

SaveFileDialog saveDialog = new SaveFileDialog();

saveDialog.Filter = "JPeg Image|\*.jpg|Bitmap Image|\*.bmp|Gif Image|\*.gif";

if (saveDialog.ShowDialog() == DialogResult.OK)

{

System.IO.FileStream fs = (System.IO.FileStream)saveDialog.OpenFile();

switch (saveDialog.FilterIndex)

{

case 1:

image.Save(fs, System.Drawing.Imaging.ImageFormat.Png);

break;

case 2:

image.Save(fs, System.Drawing.Imaging.ImageFormat.Jpeg);

break;

case 3:

image.Save(fs, System.Drawing.Imaging.ImageFormat.Bmp);

break;

}

}

}

private void grayscaleToolStripMenuItem\_Click(object sender, EventArgs e)

{

label1.Text = "Обработка изображения...";

Bitmap resultImage = new Bitmap(image.Width, image.Height);

for (int i = 0; i < image.Width; i++)

{

for (int j = 0; j < image.Height; j++)

{

Color sourceColor = image.GetPixel(i, j);

int Intensity = (int)(0.299 \* sourceColor.R + 0.587 \* sourceColor.G + 0.114 \* sourceColor.B);

Color resultColor = Color.FromArgb(Intensity, Intensity, Intensity);

resultImage.SetPixel(i, j, resultColor);

}

}

pictureBox1.Image = resultImage;

pictureBox1.Refresh();

image = resultImage;

label1.Text = "Обработка завершена";

}

private void averageToolStripMenuItem\_Click(object sender, EventArgs e)

{

label1.Text = "Обработка изображения...";

Bitmap resultImage = new Bitmap(image.Width, image.Height);

int radiusX = 1;

int radiusY = 1;

int red = 0, green = 0, blue = 0;

for (int i = 0; i < image.Width; i++)

{

for (int j = 0; j < image.Height; j++)

{

for (int l = -radiusY; l <= radiusY; l++)

{

for (int k = -radiusX; k <= radiusX; k++)

{

int idX = Clamp(i + k, 0, image.Width - 1);

int idY = Clamp(j + l, 0, image.Height - 1);

red += image.GetPixel(idX, idY).R;

green += image.GetPixel(idX, idY).G;

blue += image.GetPixel(idX, idY).B;

}

}

Color resultColor = Color.FromArgb(Clamp(red / 9, 0, 255), Clamp(green / 9, 0, 255), Clamp(blue / 9, 0, 255));

red = green = blue = 0;

resultImage.SetPixel(i, j, resultColor);

}

}

pictureBox1.Image = resultImage;

pictureBox1.Refresh();

image = resultImage;

label1.Text = "Обработка завершена";

}

private int Clamp(int value, int min, int max)

{

if (value < min)

return min;

if (value > max)

return max;

return value;

}

private void autocontrastToolStripMenuItem\_Click(object sender, EventArgs e)

{

label1.Text = "Обработка изображения...";

Bitmap resultImage = new Bitmap(image.Width, image.Height);

double R2, G2, B2, R3, G3, B3;

R2 = maxR(image);

G2 = maxG(image);

B2 = maxB(image);

R3 = minR(image);

G3 = minG(image);

B3 = minB(image);

for (int i = 0; i < image.Width; i++)

{

for (int j = 0; j < image.Height; j++)

{

Color sourceColor = image.GetPixel(i, j);

int R1 = sourceColor.R;

int G1 = sourceColor.G;

int B1 = sourceColor.B;

int newR = (int)((R1 - R3) \* 255 / (R2 - R3));

int newG = (int)((G1 - G3) \* 255 / (G2 - G3));

int newB = (int)((B1 - B3) \* 255 / (B2 - B3));

Color resultColor = Color.FromArgb(Clamp(newR, 0, 255), Clamp(newG, 0, 255), Clamp(newB, 0, 255));

resultImage.SetPixel(i, j, resultColor);

}

}

pictureBox1.Image = resultImage;

pictureBox1.Refresh();

image = resultImage;

label1.Text = "Обработка завершена";

}

public static double maxR(Bitmap image)

{

double R = -1;

Color imageColor;

for (int i = 0; i < image.Width; i++)

{

for (int j = 0; j < image.Height; j++)

{

imageColor = image.GetPixel(i, j);

if (R < imageColor.R)

R = imageColor.R;

}

}

return R;

}

public static double maxG(Bitmap image)

{

double G = -1;

Color imageColor;

for (int i = 0; i < image.Width; i++)

{

for (int j = 0; j < image.Height; j++)

{

imageColor = image.GetPixel(i, j);

if (G < imageColor.G)

G = imageColor.G;

}

}

return G;

}

public static double maxB(Bitmap image)

{

double B = -1;

Color imageColor;

for (int i = 0; i < image.Width; i++)

{

for (int j = 0; j < image.Height; j++)

{

imageColor = image.GetPixel(i, j);

if (B < imageColor.B)

B = imageColor.B;

}

}

return B;

}

public static double minR(Bitmap image)

{

double R = 256;

Color imageColor;

for (int i = 0; i < image.Width; i++)

{

for (int j = 0; j < image.Height; j++)

{

imageColor = image.GetPixel(i, j);

if (R > imageColor.R)

R = imageColor.R;

}

}

return R;

}

public static double minG(Bitmap image)

{

double G = 256;

Color imageColor;

for (int i = 0; i < image.Width; i++)

{

for (int j = 0; j < image.Height; j++)

{

imageColor = image.GetPixel(i, j);

if (G > imageColor.G)

G = imageColor.G;

}

}

return G;

}

public static double minB(Bitmap image)

{

double B = 256;

Color imageColor;

for (int i = 0; i < image.Width; i++)

{

for (int j = 0; j < image.Height; j++)

{

imageColor = image.GetPixel(i, j);

if (B > imageColor.B)

B = imageColor.B;

}

}

return B;

}

}

}