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 WindowsFormsApp1

{

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);

}

}

image = resultImage;

pictureBox1.Image = resultImage;

pictureBox1.Refresh();

label1.Text = "Обработка завершена";

}

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

{

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

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

int threshold = 128;

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

{

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

{

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

int result;

if (sourceColor.R < threshold)

{

result = 0;

} else

{

result = 255;

}

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

resultImage.SetPixel(i, j, resultColor);

}

}

pictureBox1.Image = resultImage;

pictureBox1.Refresh();

image = resultImage;

label1.Text = "Обработка завершена";

}

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

{

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

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

int w\_size = 3;

int radiusX = w\_size / 2;

int radiusY = w\_size / 2;

double res2= 0;

double sum = 0;

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

{

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

{

double resColor1 = 0;

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);

resColor1 += image.GetPixel(idX, idY).R;

}

}

double res = resColor1 / (w\_size \* w\_size);

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);

sum += (image.GetPixel(idX, idY).R - res) \* (image.GetPixel(idX, idY).R - res);

}

}

res2 = Math.Sqrt(sum / (w\_size \* w\_size));

int T = (int)(res + (-0.2) \* res2);

sum = 0;

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

int result;

if (sourceColor.R < T)

{

result = 0;

}

else

{

result = 255;

}

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

resultImage.SetPixel(i, j, resultColor);

}

}

image = resultImage;

pictureBox1.Image = resultImage;

pictureBox1.Refresh();

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;

}

}

}