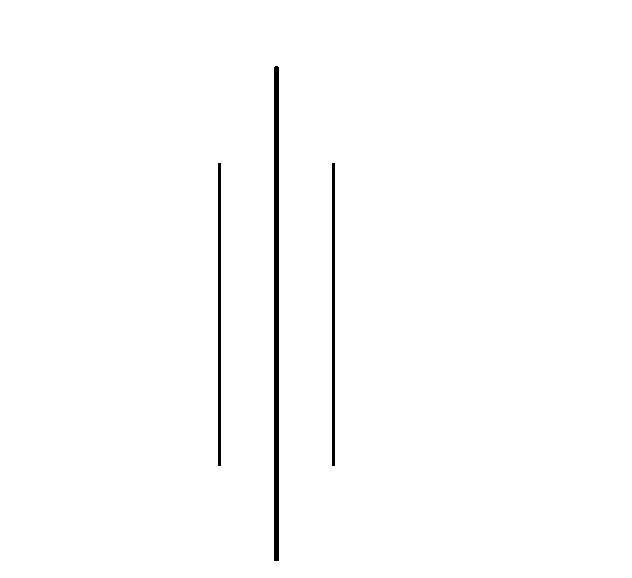
NEPAL ENGINEERING COLLEGE

( Affiliated To Pokhara University )

Changunarayan, Bhaktapur



Report on

# Lab 1b: Image Transformation and Saving Techniques

SUBMITTED BY : SUBMITTED TO:

Name: Rishi Raj Pandey Department of Computer Roll no: 021 – 366 Science and Engineering Date: 15 / 01 / 2082 Department of Computer Science and Engineering

**Objectives**: To perform rotation, apply blur, and save an RGB image in multiple formats.

**Convert the Given RGB image into**

* Rotate 90,180,270 degrees
* Blur the image
* Save image in different formats

**Code**

namespace Lab2

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

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

{

}

Bitmap originalImage;

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

{

OpenFileDialog open = new OpenFileDialog();

open.Filter = "Image Files|\*.jpg;\*.jpeg;\*.png;\*.bmp";

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

{

// Load Original Image

originalImage = new Bitmap(open.FileName);

pictureBox1.Image = originalImage;

// 90 degree Rotate

Bitmap rotate90 = (Bitmap)originalImage.Clone();

rotate90.RotateFlip(RotateFlipType.Rotate90FlipNone);

pictureBox2.Image = rotate90;

rotate90.Save("rotated\_90.png");

// 180 degree Rotate

Bitmap rotate180 = (Bitmap)originalImage.Clone();

rotate180.RotateFlip(RotateFlipType.Rotate180FlipNone);

pictureBox3.Image = rotate180;

rotate180.Save("rotated\_180.png");

// 270 degree Rotate

Bitmap rotate270 = (Bitmap)originalImage.Clone();

rotate270.RotateFlip(RotateFlipType.Rotate270FlipNone);

pictureBox4.Image = rotate270;

rotate270.Save("rotated\_270.png");

// Blur Image

Bitmap blurredImage = ApplyBlur(originalImage);

pictureBox5.Image = blurredImage;

blurredImage.Save("blurred\_image.png");

MessageBox.Show("Image processing completed successfully.");

}

}

private Bitmap ApplyBlur(Bitmap image)

{

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

for (int x = 1; x < image.Width - 1; x++)

{

for (int y = 1; y < image.Height - 1; y++)

{

int avgR =0, avgG =0, avgB =0;

for (int kx = -1; kx <= 1; kx++)

{

for (int ky = -1; ky <= 1; ky++)

{

Color pixel = image.GetPixel(x + kx, y + ky);

avgR += pixel.R;

avgG += pixel.G;

avgB += pixel.B;

}

}

avgR /=9;

avgG /= 9;

avgB /= 9;

blurred.SetPixel(x, y, Color.FromArgb(avgR,avgG,avgB));

}

}

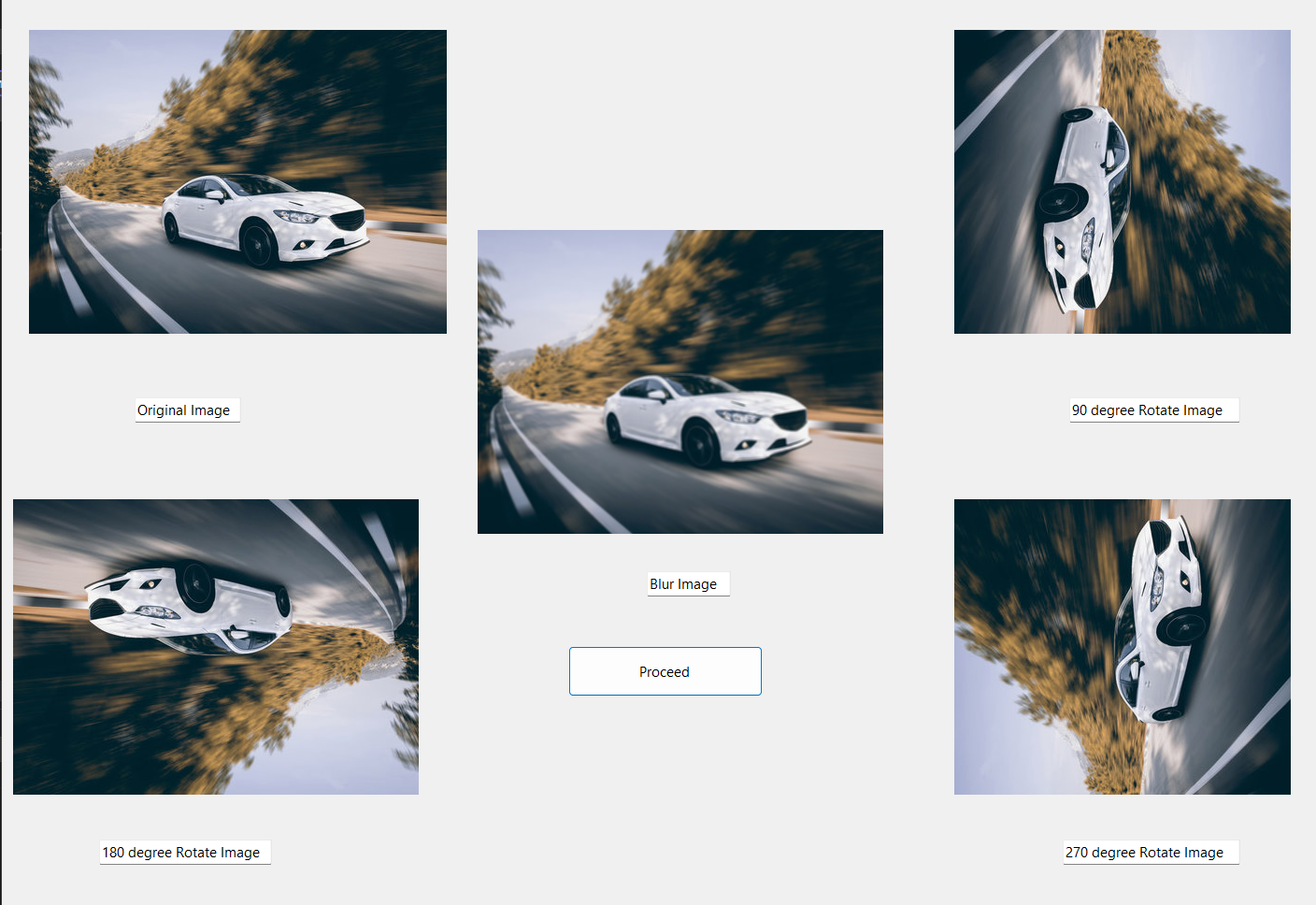
return blurred;

}

}

}

**OUTPUT**



**OBSERVATION**

The RGB image was successfully rotated by 90, 180, and 270 degrees, showing the correct orientation changes. A blur effect was applied using a Gaussian kernel, resulting in a softer and smoother image. The image was then saved in JPEG, BMP, and GIF formats without any loss of essential content

**CONCLUSION**

The experiment demonstrated the effective use of image processing techniques such as rotation, blurring, and format conversion. These basic operations are vital for enhancing, analyzing, and preparing images for various applications in multimedia and computer vision.