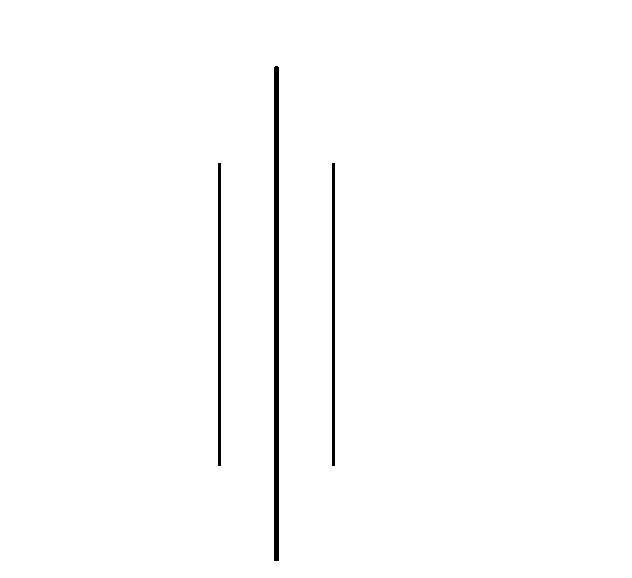
NEPAL ENGINEERING COLLEGE

( Affiliated To Pokhara University )

Changunarayan, Bhaktapur



Report on

Lab 1b: Transformation and Saving Techniques

SUBMITTED BY : SUBMITTED TO:

Name: Suja Baral Department of Computer Roll no: 021 – 387 Science and Engineering Date: 05/07/2025 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 Lab1

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

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)

{

originalImage = new Bitmap(open.FileName);

pictureBoxOriginal.Image = originalImage;

// 90 degree

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

rotate90.RotateFlip(RotateFlipType.Rotate90FlipNone);

pictureBox90.Image = rotate90;

// 180 degree

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

rotate180.RotateFlip(RotateFlipType.Rotate180FlipNone);

pictureBox180.Image = rotate180;

// 270 degree

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

rotate270.RotateFlip(RotateFlipType.Rotate270FlipNone);

pictureBox270.Image = rotate270;

// Blur

Bitmap blurred = ApplyBlur(originalImage);

pictureBoxBlur.Image = blurred;

MessageBox.Show("Image processed!");

}

}

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 r = 0, g = 0, b = 0;

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

{

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

{

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

r += pixel.R;

g += pixel.G;

b += pixel.B;

}

}

r /= 9;

g /= 9;

b /= 9;

blurred.SetPixel(x, y, Color.FromArgb(r, g, b));

}

}

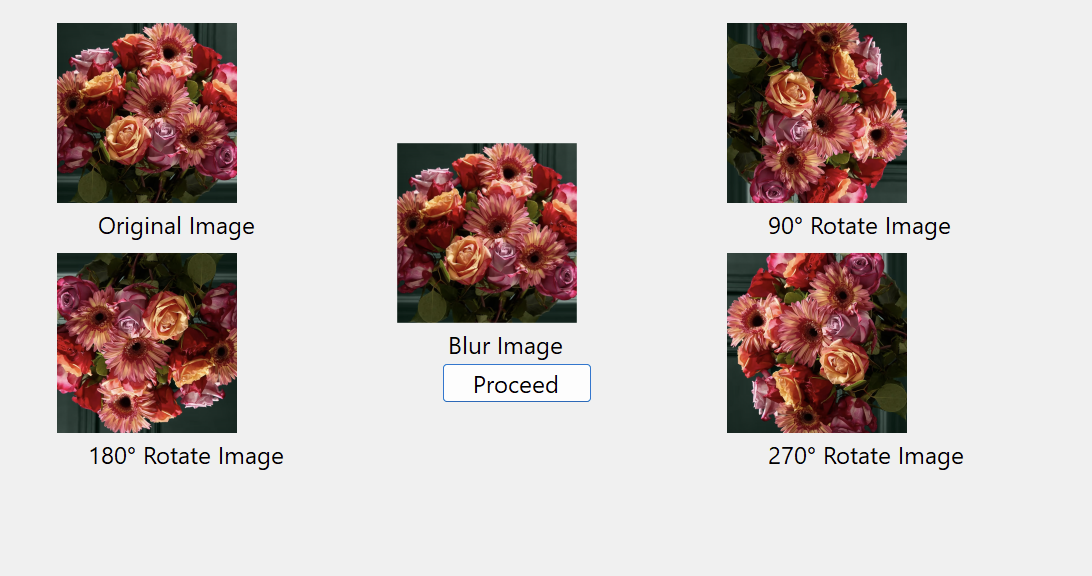
return blurred;

}

}

}

**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.