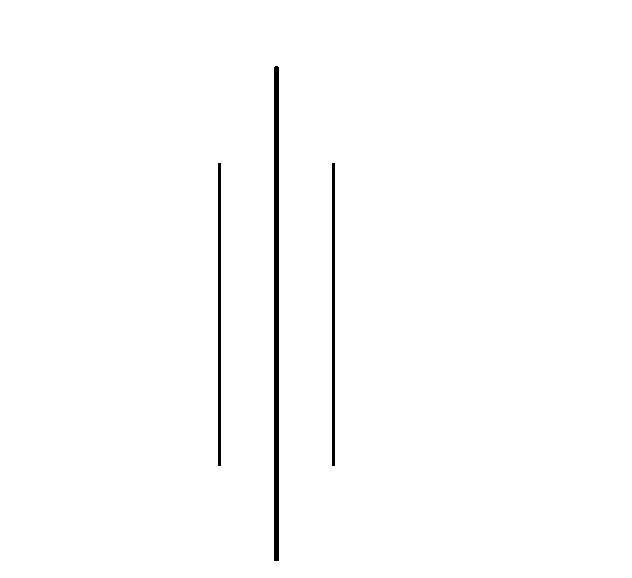
NEPAL ENGINEERING COLLEGE

( Affiliated To Pokhara University )

Changunarayan, Bhaktapur



Report on

# Lab 5: CLAHE

SUBMITTED BY : SUBMITTED TO:

Name: Rishi Raj Pandey Department of Computer Roll no: 021 – 366 Science and Engineering Date: 08 / 03 / 2082 Department of Computer Science and Engineering

**Objectives**: To implements CLAHE: Contrast Limited Adaptive Histogram Equalization on given image.

**Convert the Given RGB image into**

* Load Image
* CLAHE: Contrast Limited Adaptive Histogram Equalization
* Save image in different formats

**Code**

namespace lab5

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

Bitmap originalBitmap;

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

{

OpenFileDialog ofd = new OpenFileDialog();

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

{

originalBitmap = new Bitmap(ofd.FileName);

pictureBox1.Image = originalBitmap;

ShowImage(originalBitmap, "Original Image");

}

}

static void ShowImage(Bitmap bmp, string title)

{

Form form = new Form();

form.Text = title;

form.ClientSize = new Size(bmp.Width, bmp.Height);

PictureBox pb = new PictureBox();

pb.Dock = DockStyle.Fill;

pb.Image = bmp;

pb.SizeMode = PictureBoxSizeMode.StretchImage;

form.Controls.Add(pb);

form.Show();

}

public static Bitmap ApplyCLAHE(Bitmap image, int tileSize, int clipLimit)

{

int width = image.Width;

int height = image.Height;

Bitmap output = new Bitmap(width, height);

int tilesX = width / tileSize;

int tilesY = height / tileSize;

byte[,] gray = new byte[width, height];

for (int y = 0; y < height; y++)

for (int x = 0; x < width; x++)

{

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

byte value = (byte)(0.299 \* pixel.R + 0.587 \* pixel.G + 0.114 \* pixel.B);

gray[x, y] = value;

}

for (int ty = 0; ty < tilesY; ty++)

{

for (int tx = 0; tx < tilesX; tx++)

{

int startX = tx \* tileSize;

int startY = ty \* tileSize;

int[] hist = new int[256];

for (int y = 0; y < tileSize; y++)

for (int x = 0; x < tileSize; x++)

{

int px = startX + x;

int py = startY + y;

if (px < width && py < height)

hist[gray[px, py]]++;

}

int totalExcess = 0;

for (int i = 0; i < 256; i++)

{

if (hist[i] > clipLimit)

{

totalExcess += hist[i] - clipLimit;

hist[i] = clipLimit;

}

}

int redist = totalExcess / 256;

for (int i = 0; i < 256; i++) hist[i] += redist;

int[] cdf = new int[256];

cdf[0] = hist[0];

for (int i = 1; i < 256; i++) cdf[i] = cdf[i - 1] + hist[i];

int cdfMin = cdf[0];

int pixelsInTile = tileSize \* tileSize;

byte[] map = new byte[256];

for (int i = 0; i < 256; i++)

{

map[i] = (byte)(((cdf[i] - cdfMin) \* 255) / (pixelsInTile - cdfMin + 1));

}

for (int y = 0; y < tileSize; y++)

for (int x = 0; x < tileSize; x++)

{

int px = startX + x;

int py = startY + y;

if (px < width && py < height)

{

byte newVal = map[gray[px, py]];

output.SetPixel(px, py, Color.FromArgb(newVal, newVal, newVal));

}

}

}

}

return output;

}

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

{

if (originalBitmap != null)

{

pictureBox2.Image = ApplyCLAHE(originalBitmap, Int32.Parse(txtTileSize.Text), Int32.Parse(txtClipLimit.Text));

ShowImage((Bitmap)pictureBox2.Image, "CLAHE Image");

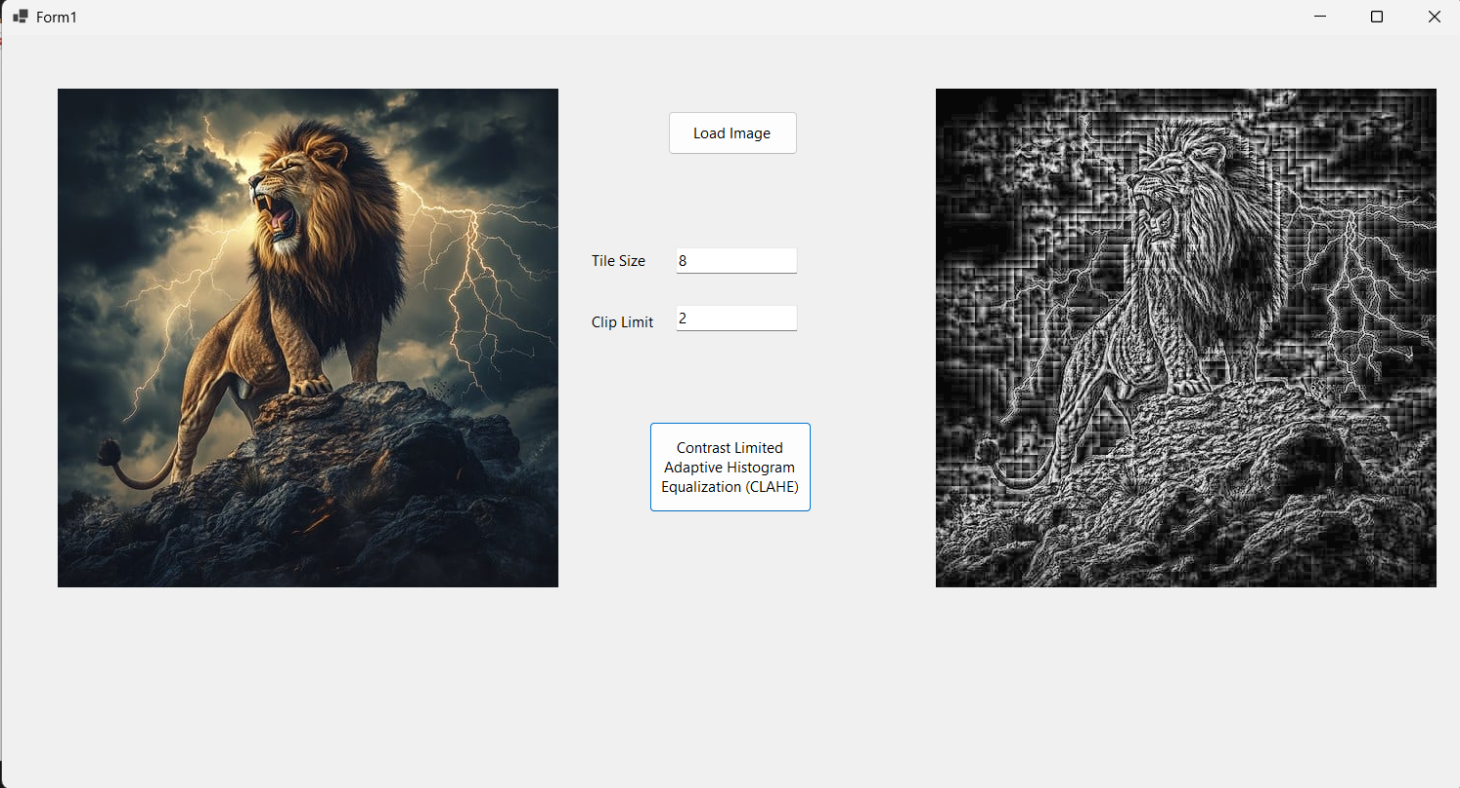
}

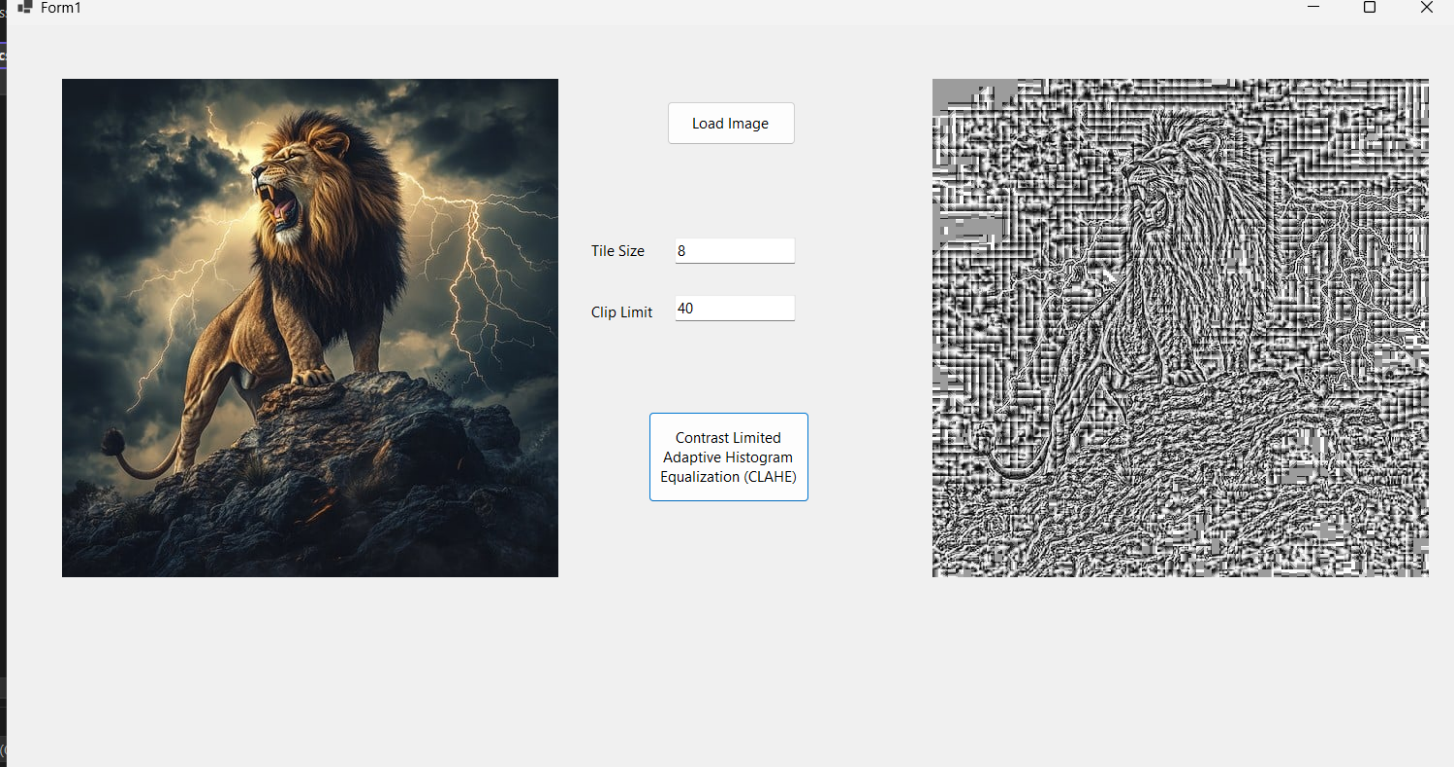
}

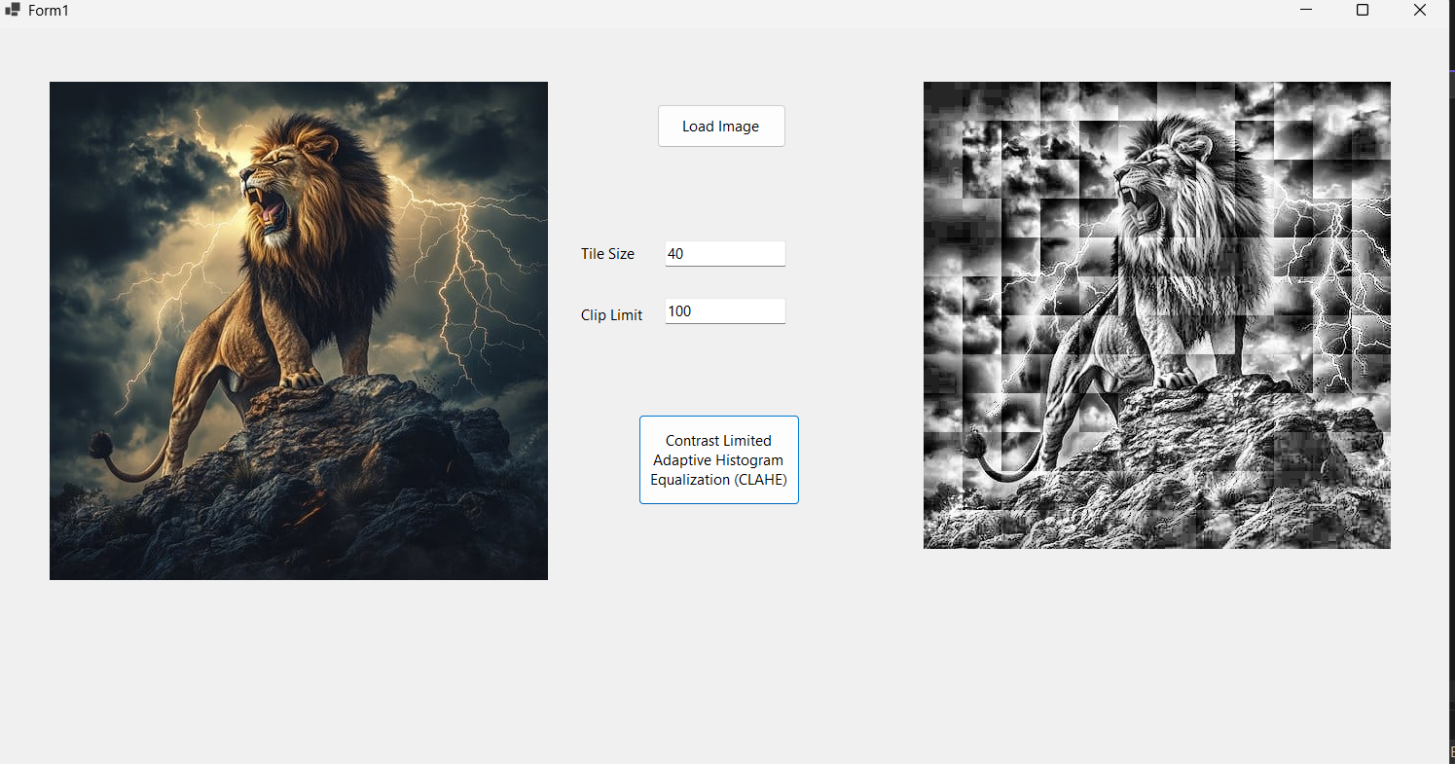
}

}

**OUTPUT**

****





**OBSERVATION**

The RGB image was successfully loaded using an image processing library such as OpenCV, retaining all its color channels for further enhancement.

Applying CLAHE (Contrast Limited Adaptive Histogram Equalization) significantly improved the local contrast of the image, especially in areas with poor lighting. Unlike global histogram equalization, CLAHE enhanced the contrast in small regions (tiles) while preventing over-amplification of noise by applying a contrast limit.

The result was a more visually balanced image, with clearer details in both dark and bright areas. The enhanced image was saved in multiple formats such as PNG, JPEG, and BMP, with slight differences in file size and image quality depending on the format used.

**CONCLUSION**

This experiment demonstrated the effectiveness of CLAHE in enhancing image contrast adaptively. By adjusting contrast locally and limiting noise amplification, CLAHE produced a more natural-looking and detailed image compared to traditional histogram equalization.

Saving the output in various formats helped observe the impact of compression on image quality. Overall, the lab provided practical understanding of CLAHE and its usefulness in improving visual quality in challenging lighting conditions.