

**BỘ THÔNG TIN TRUYỀN THÔNG
HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN
THÔNG**



**BÁO CÁO
BÀI TẬP LỚN MÔN XỬ LÝ VÀ TRUYỀN THÔNG
ĐA PHƯƠNG TIỆN**

Đề tài: “Xây dựng ứng dụng xử lý ảnh cơ bản”

Giảng viên: Vũ Hữu Tiến
Nhóm: 01
Tổ bài tập: 21

Hà Nội – 2022

Lời nói đầu

Đề tài: “Xây dựng Ứng dụng xử lý ảnh cơ bản”

Sinh Viên: Nguyễn Quốc Đạt

MSV: B20DCPT053

Xử lý ảnh là một phân ngành trong xử lý số tín hiệu với tín hiệu xử lý là ảnh. Đây là một phân ngành khoa học mới rất phát triển trong những năm gần đây. Xử lý ảnh gồm 4 lĩnh vực chính: xử lý nâng cao chất lượng ảnh, nhận dạng ảnh, nén ảnh và truy vấn ảnh. Sự phát triển của xử lý ảnh đem lại rất nhiều lợi ích cho cuộc sống của con người.

Ngày nay xử lý ảnh đã được áp dụng rất rộng rãi trong đời sống như: photoshop, nén ảnh, nén video, nhận dạng biển số xe, nhận dạng khuôn mặt, nhận dạng chữ viết, xử lý ảnh thiên văn, ảnh y tế,

Chính vì thế, ngày hôm nay em mang đến đây với đề tài “Xây dựng ứng dụng xử lý ảnh cơ bản” để vận dụng được những kiến thức đã học trong xử lý hình ảnh thực tế bằng phần mềm này.

Sinh viên thực hiện

Mục lục

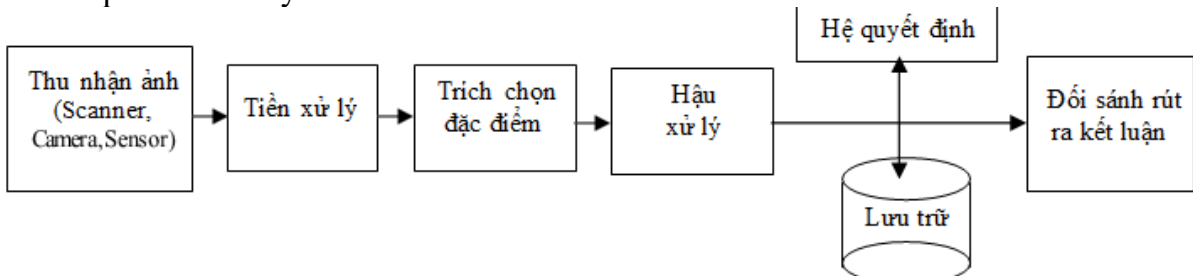
Phần I, Giới thiệu.....	
1.1, Giới thiệu xử lý ảnh	4
1.2, Giới thiệu Visual Studio.....	4
1.3, Giới thiệu Class Bitmap	5
Phần II: Phân tích, thiết kế bố cục và các nhóm chức năng phần mềm	
2.1, Phân tích, xây dựng chức năng bố cục phần mềm.....	6
2.2, Chức năng mở hình	6
2.3, Nhóm chức năng chuyển ảnh (Convert)	7
2.3.1, Chuyển thành ảnh RGB sang R/G/B	7
2.3.2, Chuyển thành ảnh xám	7
2.3.3, Chuyển từ ảnh xám thành ảnh nhị phân	8
2.3.4, Chuyển thành âm bản	8
2.4, Nhóm chức năng hiệu chỉnh (Edit)	9
2.4.1, Thay đổi độ sáng.....	9
2.4.2, Thay đổi độ tương phản.....	10
2.5, Nhóm chức năng phát hiện biên (Filter)	11
2.5.1, Phát hiện biên bằng Sobel.....	11
2.6, Biểu đồ Histogram và Cân bằng tự động	15
2.7, Nhóm chức năng Zoom ảnh.....	16
2.8, Nhóm chức năng Crop (Cắt).....	16
2.9, Chức năng lưu hình	15
2.10, Về phần mềm	16
2.11, Hiệu chỉnh phần mềm	16
Phần III, Kết quả và đánh giá	

Phần I, Giới thiệu chung

1.1, Giới thiệu xử lý ảnh

Quá trình xử lý ảnh là quá trình thao tác ảnh đầu vào nhằm cho ra kết quả mong muốn. Kết quả đầu ra của một quá trình xử lý ảnh có thể là một ảnh “tốt hơn” hoặc một kết luận.

Sơ đồ quá trình xử lý ảnh:

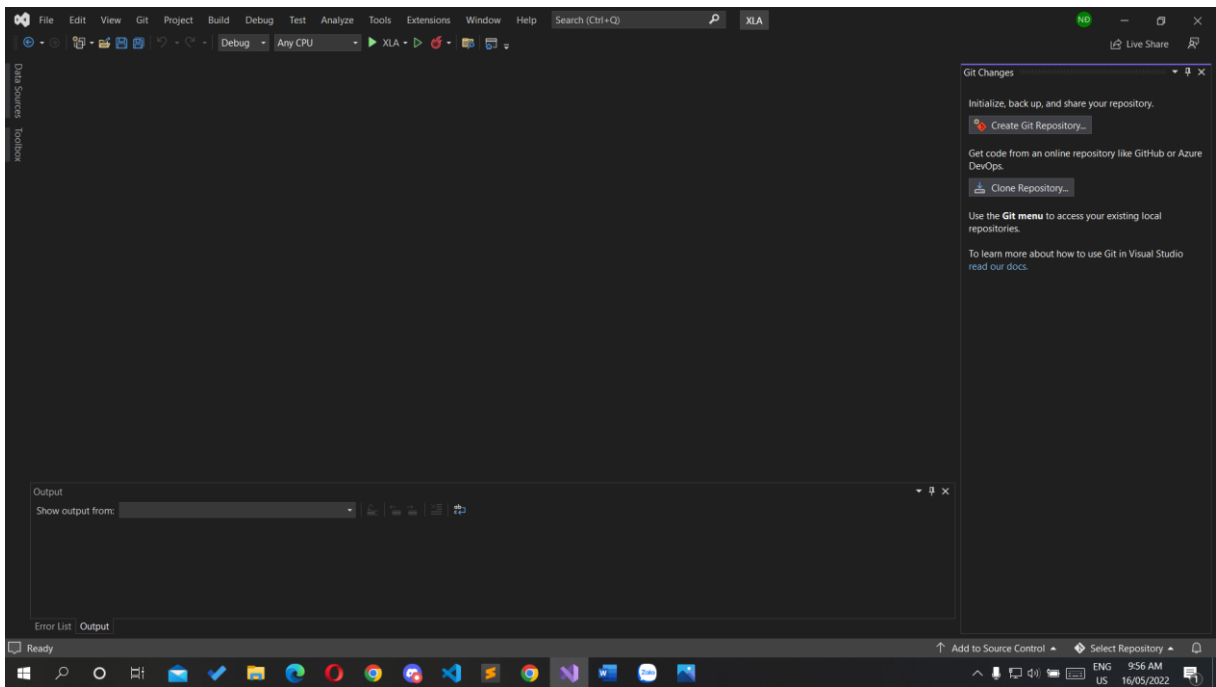


Hình 1.1: Quá trình xử lý ảnh

1.2, Giới thiệu Visual Studio

Microsoft Visual Studio là một môi trường phát triển tích hợp (IDE) từ Microsoft. Nó được sử dụng để phát triển chương trình máy tính cho Microsoft Windows, cũng như các trang web, các ứng dụng web và các dịch vụ web. Visual Studio sử dụng nền tảng phát triển phần mềm của Microsoft như Windows API, Windows Forms, Windows Presentation Foundation, Windows Store và Microsoft Silverlight. Nó có thể sản xuất cả hai ngôn ngữ máy và mã số quản lý.

Visual Studio hỗ trợ nhiều ngôn ngữ lập trình khác nhau và cho phép trình biên tập mã và gỡ lỗi để hỗ trợ (mức độ khác nhau) hầu như mọi ngôn ngữ lập trình. Các ngôn ngữ tích hợp gồm có C/C++ và C++/CLI (thông qua Visual C++), VB.NET (thông qua Visual Basic.NET), C# (thông qua Visual C#) và F# (như của Visual Studio 2010[5]). Hỗ trợ cho các ngôn ngữ khác như J++/J#, Python và Ruby thông qua dịch vụ cài đặt riêng rẽ. Nó cũng hỗ trợ XML/XSLT, HTML/XHTML, JavaScript và CSS.



Hình 1.2: Giao diện chính Microsoft Visual Studio 2022

1.3, Giới thiệu Class Bitmap

Gói gọn trong một GDI + Bitmap, trong đó bao gồm các dữ liệu pixel cho hình ảnh đồ họa và các thuộc tính của nó. Một Bitmap là một đối tượng được sử dụng để làm việc với hình ảnh được xác định bởi dữ liệu từ các pixel của hình ảnh.

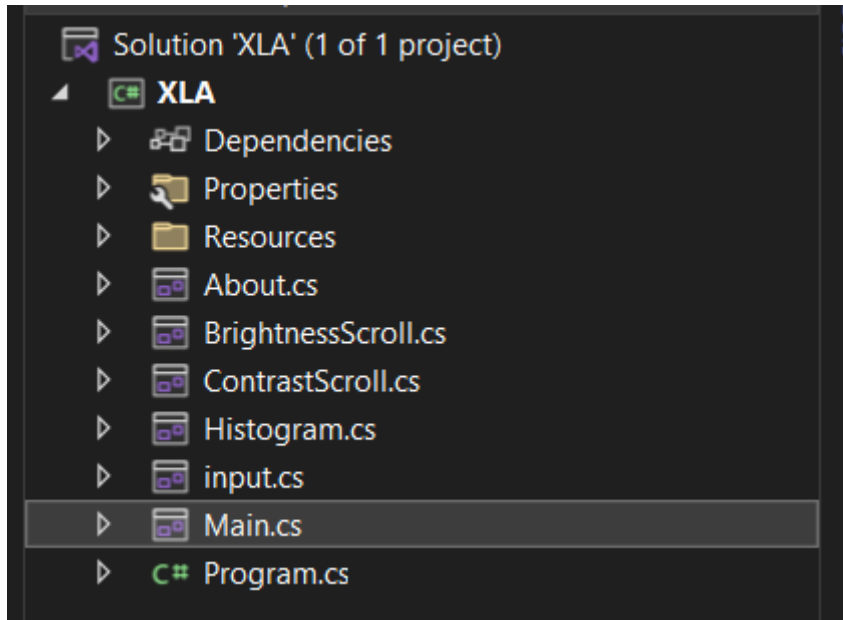
Bao gồm một số thuộc tính xử lý chính:

- GetPixel: Lấy ra một điểm ảnh.
- Color: Lấy ra thuộc tính màu của một pixel.
- SetPixel: Đặt giá trị cho một điểm ảnh.

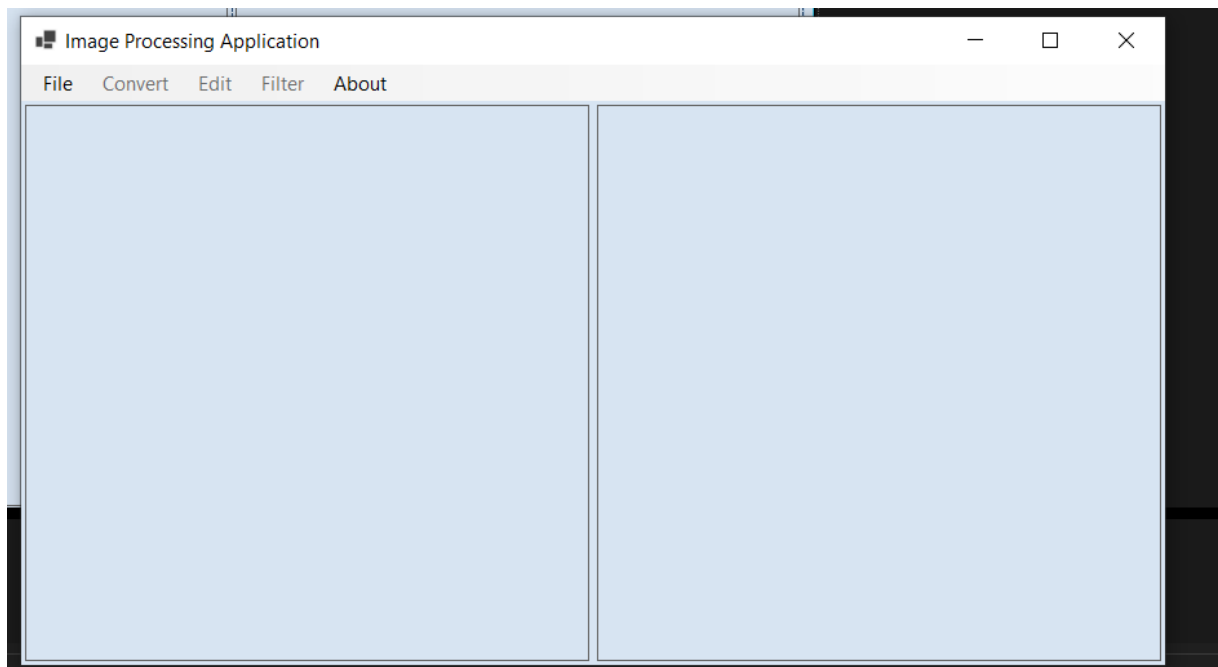
Phần II: Phân tích, thiết kế bố cục và các nhóm chức năng phần mềm

2.1, Phân tích, xây dựng chức năng bố cục phần mềm

Phần mềm được viết bằng ngôn ngữ C#. Bao gồm 5 Form: Main là Form hiển thị chính của phần mềm, các Form còn lại gồm: BrightnessScroll, ContrastScrool, Input, About.



Hình 2.1: Cấu trúc các Form



Hình 2.2: Bố cục của phần mềm

2.2, Chức năng mở hình

Chức năng này sẽ mở ra một cửa sổ cho phép chọn một hình ảnh và dùng nó làm ảnh nguồn để chỉnh sửa.

Vị trí: Thuộc vào Form Main (Form chính).

Hành động: Click vào nút Open.

```
public void openToolStripMenuItem_Click(object sender, EventArgs e)
{
    OpenFileDialog.Title = "Open picture";
    OpenFileDialog.InitialDirectory = @"C:\\";
    if (OpenFileDialog.ShowDialog() == DialogResult.OK)
    {
        RunApp(true);
        Old.Image = Image.FromFile(OpenFileDialog.FileName);
    }
}
```

2.3, Nhóm chức năng chuyển ảnh (Convert)

Bao gồm chuyển thành ảnh từ màu RGB sang cách kênh màu Red, Green, Blue, Gray, ảnh màu xám sang ảnh nhị phân (gray to binary) và màu âm bản.

Chuyển ảnh

2.3.1, Chuyển thành ảnh Từ RGB qua R/G/B

Chức năng này cho phép chuyển ảnh nguồn thành ảnh đỏ.

Ý tưởng: Sử dụng vòng lặp để đọc từng điểm ảnh. Sau đó sử dụng công thức chuyển ảnh màu thành ảnh đỏ để tính giá trị mới. Cuối cùng đặt lại giá trị mới cho điểm ảnh vừa đọc.

Vị trí: Thuộc nhóm chức năng Convert trong thanh Menubar của Form ImagesProcessing

Hành động: Click vào nút RBG to Red trong Convert.

Hàm chuyển từ màu RGB sang màu đỏ:

```
private void rGBToRedToolStripMenuItem_Click(object sender, EventArgs e)
{
    Bitmap img = (Bitmap)Old.Image.Clone();
    Bitmap red = new Bitmap(img.Width, img.Height);
    for (int i = 0; i < img.Width; i++)
    {
        for (int j = 0; j < img.Height; j++)
        {
            Color c = img.GetPixel(i, j);

            byte R = c.R;
            byte A = c.A;

            red.SetPixel(i, j, Color.FromArgb(A, R, 0, 0));
        }
    }
    New.Image = red;
}
```

Tương tự, ta cũng có các kênh màu như Green và Blue.

2.3.2, Chuyển thành ảnh xám

Chức năng này cho phép chuyển ảnh nguồn thành ảnh xám.

Ý tưởng: Sử dụng vòng lặp để đọc từng điểm ảnh. Sau đó sử dụng công thức chuyển ảnh màu thành ảnh xám để tính giá trị mới. Cuối cùng đặt lại giá trị mới cho điểm ảnh vừa đọc.

Vị trí: Thuộc nhóm chức năng Convert trong thanh Menubar của Form Main

Hành động: Click vào nút RBG to Gray trong Convert.

```
private void rGBToGrayToolStripMenuItem1_Click(object sender, EventArgs e)
{
    Bitmap img = (Bitmap)Old.Image.Clone();
    Bitmap gray = new Bitmap(img.Width, img.Height);
    for (int i = 0; i < img.Width; i++)
    {
```

```

        for (int j = 0; j < img.Height; j++)
        {
            Color c = img.GetPixel(i, j);

            byte R = c.R;
            byte G = c.G;
            byte B = c.B;
            byte max = Math.Max(R, Math.Max(G, B));
            byte min = Math.Min(R, Math.Min(G, B));

            //()
            // int x = (int)(c.R * .299 + c.B * .114 + c.G * .578);
            // byte x = (byte)((R + G + B) / 3);
            // using Average

            byte x = (byte)((min + max) / 2);
            gray.SetPixel(i, j, Color.FromArgb(x, x, x));
            // Using Lightness
        }
    }
    New.Image = gray;
}

```

2.3.3, Chuyển từ ảnh xám thành ảnh nhị phân

Chức năng này cho phép chuyển ảnh nguồn có màu xám thành ảnh nhị phân.

Ý tưởng: Sử dụng vòng lặp để đọc từng điểm ảnh. Sau đó lấy trung bình cộng của điểm ảnh có giá trị lớn nhất và nhỏ nhất, so sánh với 128, nếu lớn hơn thì gán về 0 còn nếu nhỏ hơn thì gán về 255. Cuối cùng đặt lại giá trị mới cho điểm ảnh vừa đọc.

Vị trí: Thuộc nhóm chức năng Convert trong thanh Menubar của Form Main

Hành động: Click vào nút Gray to Bin trong Convert.

```

private void grayToBinToolStripMenuItem1_Click(object sender, EventArgs e)
{
    // Gray -> Bin
    Bitmap img = (Bitmap)Old.Image.Clone();
    Bitmap bin = new Bitmap(img.Width, img.Height);
    for (int i = 0; i < img.Width; i++)
    {
        for (int j = 0; j < img.Height; j++)
        {
            Color c = img.GetPixel(i, j);
            byte R = c.R;
            byte G = c.G;
            byte B = c.B;
            byte max = Math.Max(R, Math.Max(G, B));
            byte min = Math.Min(R, Math.Min(G, B));
            byte Bina = (byte)((min + max) / 2);
            if (Bina < 128)
                Bina = 0;
            else
                Bina = 255;
            bin.SetPixel(i, j, Color.FromArgb(Bina, Bina, Bina));
        }
    }
    New.Image = bin;
}

```

2.3.4, Chuyển thành âm bản

Chức năng này cho phép chuyển ảnh nguồn thành âm bản.

Ý tưởng: Sử dụng vòng lặp để đọc từng điểm ảnh trong hình. Với mỗi điểm ảnh, mỗi giá trị màu bằng 255 trừ đi giá trị hiện tại. Cuối cùng đặt lại giá trị mới cho điểm ảnh vừa đọc.

Vị trí: Thuộc nhóm chức năng Convert trong thanh Menubar của Form Main.

Hành động: Click vào nút Invert trong Convert.

```
private void invertToolStripMenuItem1_Click(object sender, EventArgs e)
{
    Bitmap img = (Bitmap)Old.Image.Clone();
    Bitmap res = new Bitmap(img.Width, img.Height);
    for (int i = 0; i < img.Width; i++)
    {
        for (int j = 0; j < img.Height; j++)
        {
            Color c = img.GetPixel(i, j);
            res.SetPixel(i, j, Color.FromArgb(255 - c.R, 255 - c.G, 255 -
c.B));
        }
    }
    New.Image = res;
}
```

2.4, Nhóm chức năng hiệu chỉnh (Edit)

Bao gồm thay đổi độ sáng và thay đổi độ tương phản.

2.4.1, Thay đổi độ sáng

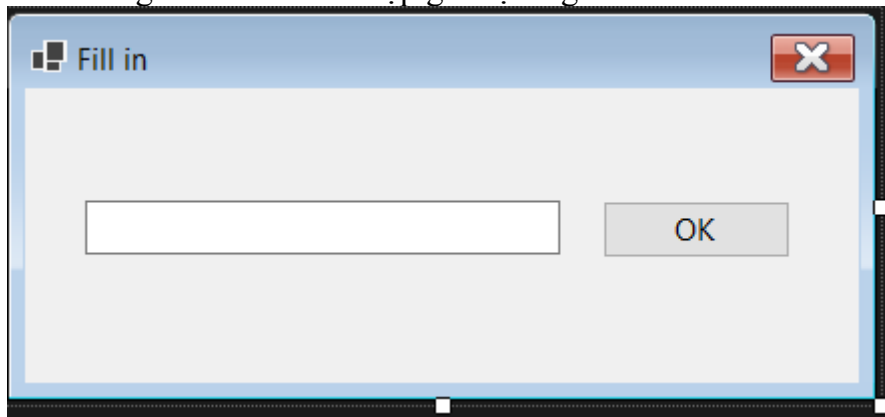
Chức năng này cho phép thay đổi độ sáng theo giá trị đã nhập vào ô input.

Vị trí: Thuộc nhóm chức năng Adjustments trong Edit của thanh Menubar của Form Main.

Ý tưởng: Hiện ra một Form Fill in cho phép nhập vào giá trị mức sáng cần tăng. Xử lý lệnh này bằng các cộng giá trị vừa nhập vào từng điểm ảnh. Nếu giá trị sáng quá 255 thì giá trị sáng max đó bằng 255.

Hành động: Click vào nút Brightness trong Adjustment, nhập giá trị vào ô Brightness trong Form mới và nhấn OK.

Form Brightness chứa ô nhập giá trị Brightness và nút OK.



Hành động: Click vào OK trong Form Fill in.

Hàm xử lý:

```
private void brightnessToolStripMenuItem_Click(object sender, EventArgs e)
{
    input ip = new input();
    ip.ShowDialog();
    if (ip.textBox1.Text == "") return;
    int val = int.Parse(ip.textBox1.Text);
    Bitmap t = (Bitmap)Old.Image.Clone();
    Bitmap res = new Bitmap(t.Width, t.Height);
    for (int x = 0; x < t.Width; x++)
    {
        for (int y = 0; y < t.Height; y++)
        {
            Color c = t.GetPixel(x, y);
```

```

        int x1 = Math.Max(Math.Min(c.R + val, 255), 0);
        int x2 = Math.Max(Math.Min(c.G + val, 255), 0);
        int x3 = Math.Max(Math.Min(c.B + val, 255), 0);
        res.SetPixel(x, y, Color.FromArgb(x1, x2, x3));
    }
}
New.Image = res;
}

```

2.4.2, Thay đổi độ tương phản

Chức năng này cho phép thay đổi độ tương phản theo giá trị nhập vào.

Vị trí: Thuộc nhóm chức năng Adjustments của Mục Edit trong thanh Menubar của Form Main.

Ý tưởng: Hiện ra một Form input cho phép nhập vào giá trị tương phản mới.

Thực hiện tương tự đối với phần Brightness.

Hàm xử lý:

```

private void contrastToolStripMenuItem_Click(object sender, EventArgs e)
{
    input ip = new input();
    ip.ShowDialog();

    Bitmap t = (Bitmap)Old.Image.Clone();
    Bitmap res = new Bitmap(t.Width, t.Height);
    if (ip.textBox1.Text == "") return;
    int v = int.Parse(ip.textBox1.Text);
    double val = ((100.0 + (v - 100)) / 100) + ((100.0 + (v - 100)) /
100);
    for (int i = 0; i < t.Width; i++)
    {
        for (int j = 0; j < t.Height; j++)
        {
            Color c = t.GetPixel(i, j);
            double tmp = (((c.R / 255.0) - 0.5) * val) + 0.5) * 255.0;
            int nr = (int)tmp;

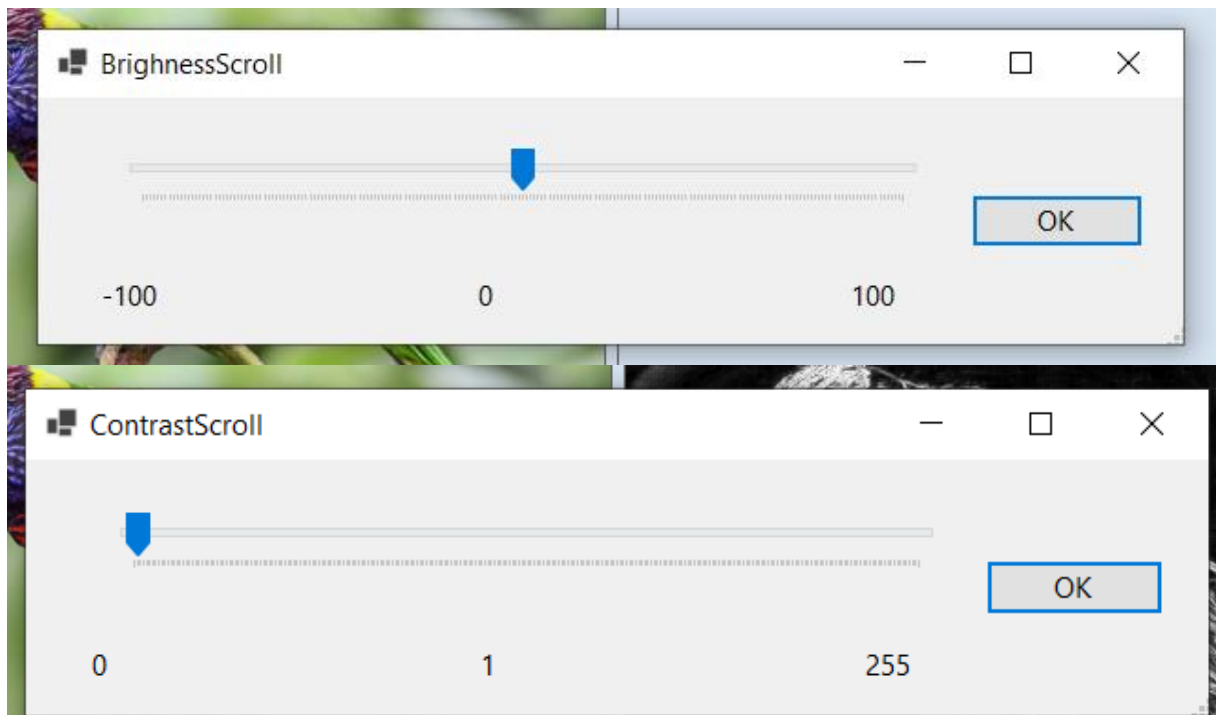
            tmp = (((c.G / 255.0) - 0.5) * val) + 0.5) * 255.0;
            int ng = (int)tmp;

            tmp = (((c.B / 255.0) - 0.5) * val) + 0.5) * 255.0;
            int nb = (int)tmp;

            if (nr < 0) nr = 0;
            if (nr > 255) nr = 255;
            if (ng < 0) ng = 0;
            if (ng > 255) ng = 255;
            if (nb < 0) nb = 0;
            if (nb > 255) nb = 255;
            res.SetPixel(i, j, Color.FromArgb(c.A, nr, ng, nb));
        }
    }
    New.Image = res;
}

```

Ngoài ra, còn có thêm chức năng thay đổi độ tương phản và độ sáng thông qua thanh cuộn Scroll bar.



2.5, Nhóm chức năng phát hiện biên (Filter)

2.5.1, Phát hiện biên bằng Sobel

Thuật toán phát hiện biên Sobel là làm nổi bật đường biên. Được xác định bằng tổng

tích chập của 2 ma trận chập

$$\begin{matrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{matrix} \text{ và } \begin{matrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{matrix}$$

Vị trí: Thuộc nhóm chức năng Filter trong thanh Menubar của Form Main.

Hành động: click vào Filter.

```
private void filterToolStripMenuItem_Click(Object sender, EventArgs e)
{
    Bitmap picture = (Bitmap) Old.Image.Clone();
    int[] matrix = new int[9];
    int[,] pixelPicture = new int[picture.Width - 2, picture.Height - 2];
    int[,] pixelPicture1 = new int[picture.Width - 2, picture.Height - 2];
    int[,] pixelPicture2 = new int[picture.Width - 2, picture.Height - 2];

    int pixel;
    int cnt = 0;
    for (int i = 0; i < picture.Width - 2; i++)
    {
        for (int j = 0; j < picture.Height - 2; j++)
        {
            cnt = 0;
            for (int n = i; n <= i + 2; n++)
            {
                for (int m = j; m <= j + 2; m++)
                {
                    Color c = picture.GetPixel(n, m);
                    matrix[cnt] = c.R;
                    cnt++;
                }
            }
            pixel = matrix[0] * -1 + matrix[1] * -2 + matrix[3] * -1 +
matrix[6] * 1 + matrix[7] * 2 + matrix[8] * 1;
            pixelPicture1[i, j] = pixel;
        }
    }
}
```

```

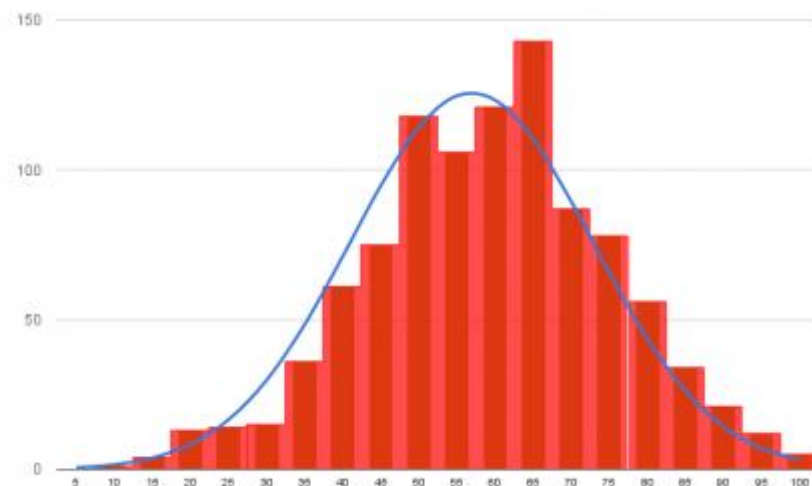
for (int i = 0; i < picture.Width - 2; i++)
{
    for (int j = 0; j < picture.Height - 2; j++)
    {
        cnt = 0;
        for (int n = i; n <= i + 2; n++)
            for (int m = j; m <= j + 2; m++)
            {
                Color c = picture.GetPixel(n, m);
                matrix[cnt] = c.R;
                cnt++;
            }
        pixel = matrix[0] * -1 + matrix[2] * 1 + matrix[3] * -2 +
matrix[5] * 2 + matrix[6] * -1 + matrix[8] * 1;
        pixelPicture2[i, j] = pixel;
    }
}
for (int i = 0; i < picture.Width - 2; i++)
{
    for (int j = 0; j < picture.Height - 2; j++)
    {
        pixelPicture[i, j] = Math.Abs(pixelPicture1[i, j]) +
Math.Abs(pixelPicture2[i, j]);
    }
}
for (int i = 0; i < picture.Width - 2; i++)
{
    for (int j = 0; j < picture.Height - 2; j++)
    {
        Color c = picture.GetPixel(i, j);
        if (pixelPicture[i, j] > 255)
            picture.SetPixel(i, j, Color.FromArgb(255, 255, 255));
        else
            picture.SetPixel(i, j, Color.FromArgb(pixelPicture[i, j],
pixelPicture[i, j], pixelPicture[i, j]));
    }
}
New.Image = picture;
}

```

2.6, Biểu đồ Histogram và Cân bằng tự động

2.6.1, Biểu đồ Histogram

Biểu đồ Histogram là biểu đồ tần suất mức xám g của ảnh I là số điểm ảnh có giá trị g của ảnh I.



Chức năng này sẽ hiển thị ra biểu đồ Histogram.

Vị trí: Thuộc nhóm chức năng Histogram trong thanh Menubar của Form Main.

Ý tưởng: Để vẽ được biểu đồ Histogram, đầu tiên cần phải tìm ra được tất cả các mức xám có trong ảnh và đếm số lượng các mức xám đó. Sau đó sử dụng Bitmap để tạo một ảnh mới chứa Histogram. Hiện thị ra ngoài phần mềm.

Cách vẽ có thể tham khảo thêm.

2.6.2, Cân bằng tự động:

Vị trí: Thuộc nhóm chức năng Histogram trong thanh Menubar của Form ImagesProcessing

Chức năng này sẽ cân bằng Histogram tự động.

Ý tưởng: Dựa vào thuật toán cân bằng Histogram.

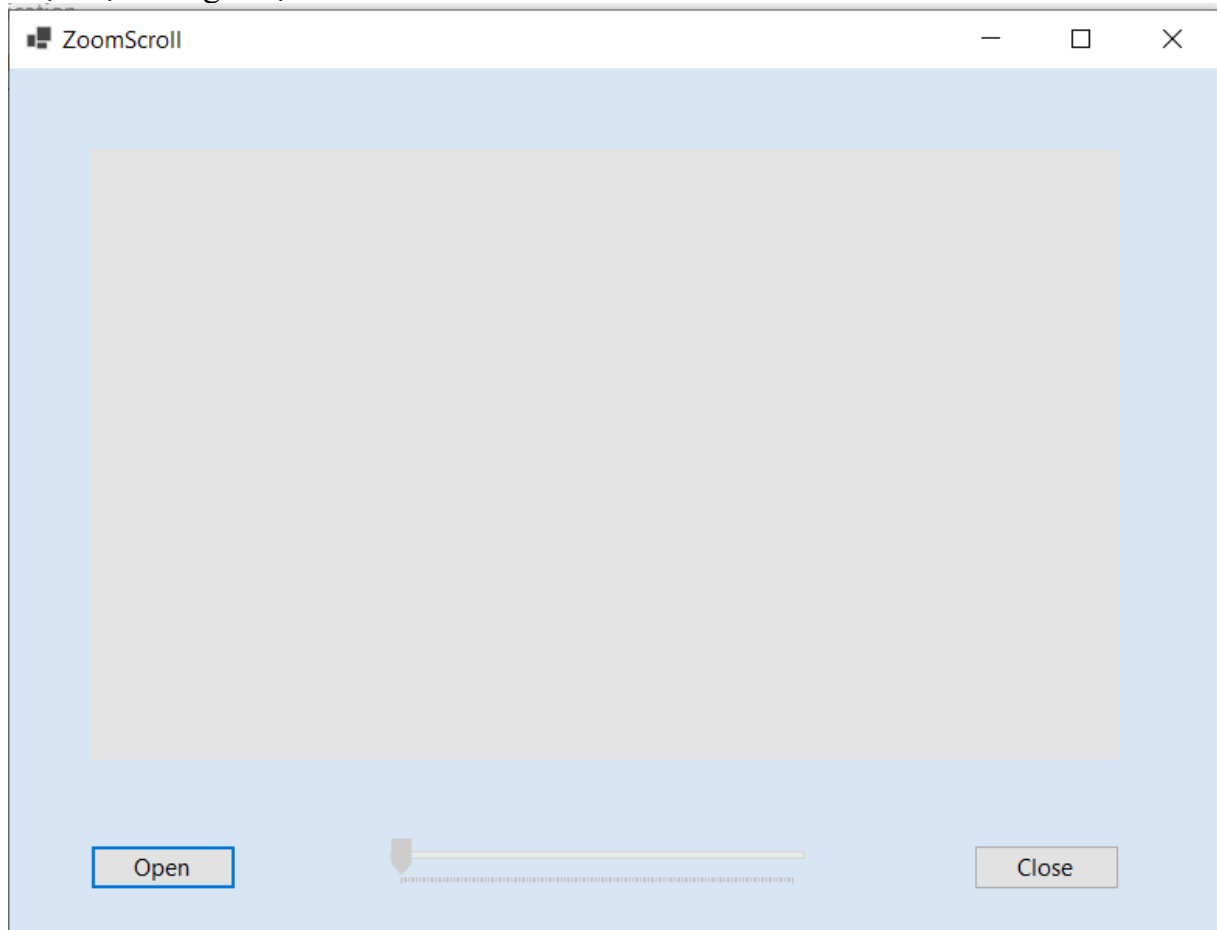
<https://github.com/ngdat2002/BTL-XLTT-DPT-PTIT/blob/main/XLA/Histogram.cs>
links tham khảo code do code của em hơi dài.

2.7, Nhóm chức năng Zoom

Chức năng này sẽ hiển thị một vùng ảnh với kích thước lớn hơn.

Vị trí: thuộc nhóm Zoom trong thanh Menubar của Form Main.

Ý tưởng: Sử dụng lớp Graphics (System.Drawing) trong C#, vẽ lại vùng ảnh được chọn dựa vào giá trị trên thanh scroll bar.



Hành động: Click vào Zoom trên thanh Menubar của Form Main sẽ hiển thị ra Form ZoomScroll như trên. Sau đó tiếp tục mở ảnh, kéo thanh Scroll bar để zoom bức ảnh lên.

Hàm xử lý:

```
Image ZoomPicture(Image img, Size sz)
{
```

```

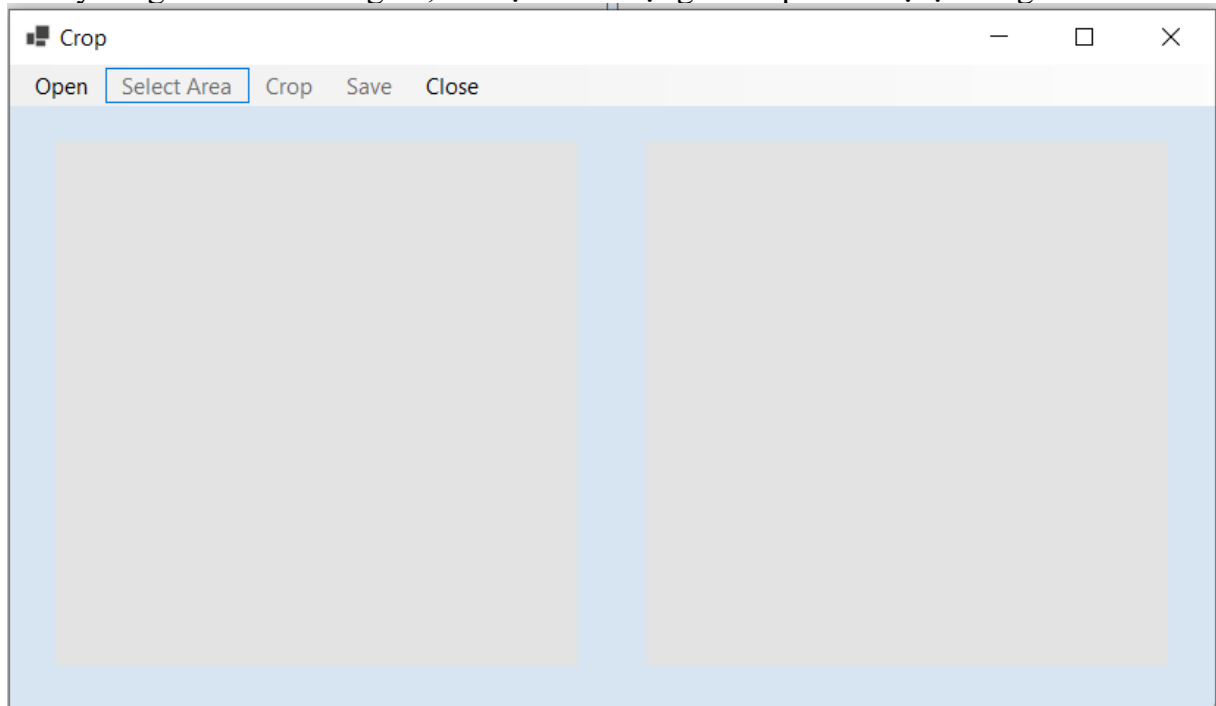
        Bitmap m = new(img, Convert.ToInt32(img.Width) +
        Convert.ToInt32(img.Width * sz.Width / ss), Convert.ToInt32(img.Height) +
        Convert.ToInt32(img.Height * sz.Height / ss));
        Graphics gpu = Graphics.FromImage(m);
        gpu.InterpolationMode =
        System.Drawing.Drawing2D.InterpolationMode.HighQualityBicubic;
        return m;
    }
    private void trackBar1_Scroll(object sender, EventArgs e)
    {
        if(trackBar1.Value != 0)
        {
            pictureBox1.Image = null;
            pictureBox1.Image = ZoomPicture(t.Image, new
            Size(trackBar1.Value, trackBar1.Value));
        }
    }
}

```

2.8, Nhóm chức năng Crop(cắt)

Chức năng này cho phép ta cắt một vùng ảnh của một bức ảnh ban đầu.

Ý tưởng: Dùng tính năng `MouseEventHandler` để lưu lại tọa độ của bức ảnh cần lấy, sau đó lấy từng điểm ảnh trong đó, lưu lại và sử dụng bitmap hiển thị lại vùng ảnh đó.



Hành động: click vào Crop trên thanh Menu bar của Form Main. Sau đó mở ảnh lên. Chọn chức năng Select Area để bắt đầu chọn vùng ảnh. Bấm Crop để cắt ảnh, Ảnh sau khi cắt có thể lưu lại.

Các Hàm xử lý:

// khởi tạo các hàm liên quan đến `MouseEventHandler`.

```

private void selectAreaToolStripMenuItem_Click(object sender, EventArgs e)
{
    pictureBox1.MouseDown += new
    MouseEventHandler(pictureBox1_MouseDown);

    pictureBox1.MouseMove += new
    MouseEventHandler(pictureBox1_MouseMove);

    pictureBox1.MouseEnter += new EventHandler(pictureBox1_MouseEnter);
    Controls.Add(pictureBox1);
}

```

```

int crpX, crpY, rectW, rectH;
public Pen crpPen = new Pen(Color.White);
private void cropToolStripMenuItem_Click(object sender, EventArgs e)
{
    Cursor = Cursors.Default;
    Bitmap bmp2 = new Bitmap(pictureBox1.Width, pictureBox1.Height);
    pictureBox1.DrawToBitmap(bmp2, pictureBox1.ClientRectangle);

    Bitmap crpImg = new Bitmap(rectW, rectH);

    for (int i = 0; i < rectW; i++)
    {
        for (int y = 0; y < rectH; y++)
        {
            Color pxlclr = bmp2.GetPixel(crpX + i, crpY + y);
            crpImg.SetPixel(i, y, pxlclr);
        }
    }
    // đưa ra vị trí ảnh

    pictureBox2.Image = (Image)crpImg;
    pictureBox2.SizeMode = PictureBoxSizeMode.CenterImage;
}
private void pictureBox1_MouseDown(object sender, MouseEventArgs e)
{
    // hàm lấy tọa độ
    base.OnMouseDown(e);

    if (e.Button == System.Windows.Forms.MouseButtons.Left)
    {
        Cursor = Cursors.Cross;
        crpPen.DashStyle = System.Drawing.Drawing2D.DashStyle.Dot;
        crpX = e.X;
        crpY = e.Y;
    }
}
private void pictureBox1_MouseEnter(object sender, EventArgs e)
{
    base.OnMouseEnter(e);
    Cursor = Cursors.Cross;
}

private void pictureBox1_MouseMove(object sender, MouseEventArgs e)
{
    // di chuyển chuột -> thu thập từng bit ảnh
    base.OnMouseMove(e);
    if (e.Button == System.Windows.Forms.MouseButtons.Left)
    {
        pictureBox1.Refresh();
        rectW = e.X - crpX;
        rectH = e.Y - crpY;
        Graphics g = pictureBox1.CreateGraphics();
        g.DrawRectangle(crpPen, crpX, crpY, rectW, rectH);
        g.Dispose();
    }
}
protected override void OnMouseEnter(EventArgs e)
{
    base.OnMouseEnter(e);
    Cursor = Cursors.Default;
}

```

2.9, Chức năng lưu hình

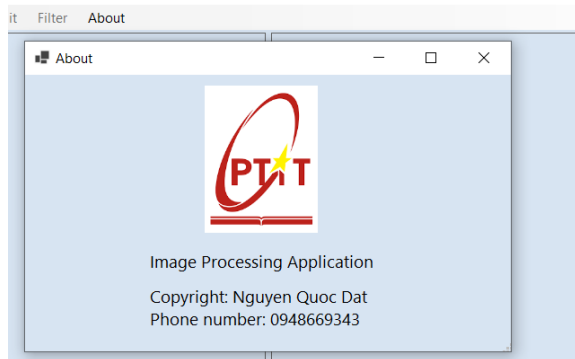
Cho phép lưu lại bức hình đã chỉnh sửa.

Hành động: Click vào nút Save.

2.10, Về phần mềm

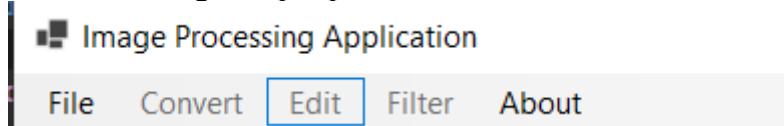
Hiện thị sinh viên thực hiện và phát triển phần mềm.

Hành động: Click vào nút About trên thanh Menu Bar.



2.11, Hiệu chỉnh phần mềm

Khi chưa chọn một hình ảnh nào. Các chức năng trong các nhóm chức năng của phần mềm sẽ không cho phép chọn.



Vô hiệu hóa các nút bấm đó ngay sau khi Form Main_Load() xảy ra. Đây là sự kiện đầu tiên khi chạy phần mềm.

```
private void RunApp(bool x)
{
    adjustmentsToolStripMenuItem.Enabled = x;
    convertToolStripMenuItem.Enabled = x;
    saveToolStripMenuItem.Enabled = x;
    filterToolStripMenuItem.Enabled = x;
    editToolStripMenuItem.Enabled = x;
    convertToolStripMenuItem.Enabled = x;
}
private void Form1_Load(object sender, EventArgs e)
{
    RunApp(false);
}
```

Sau đó bật lại bằng cách nếu đã chọn hình:

```
public void openToolStripMenuItem_Click(object sender, EventArgs e)
{
    OpenFileDialog.Title = "Open picture";
    OpenFileDialog.InitialDirectory = @"C:\Users";
    if (OpenFileDialog.ShowDialog() == DialogResult.OK)
    {
        RunApp(true);
        Old.Image = Image.FromFile(OpenFileDialog.FileName);
    }
}
```

Tính năng này được áp dụng tương tự với các nhóm chức năng Histogram/Zoom/Crop.

Phần III: Kết quả và đánh giá

Hiện tại em đang còn gặp vấn đề trong khâu đóng gói.

Source code: <https://github.com/nqdat2002/BTL-XLTT-PT-PTIT>

Tự đánh giá:

- *Hiện tại em thấy App vẫn còn hơi chậm do thuật toán chưa tối ưu.*
- *Giao diện chưa đồng nhất được các chức năng.*

Lời Kết

Do lần đầu thực hiện nên em còn thiếu kinh nghiệm và có nhiều sai sót, em mong thầy nhận xét và góp ý giúp em, để những lần sau em cải thiện những đề tài khác tốt hơn. Em xin chân thành cảm ơn.