University of Technology
الجامعة التكنولوجية

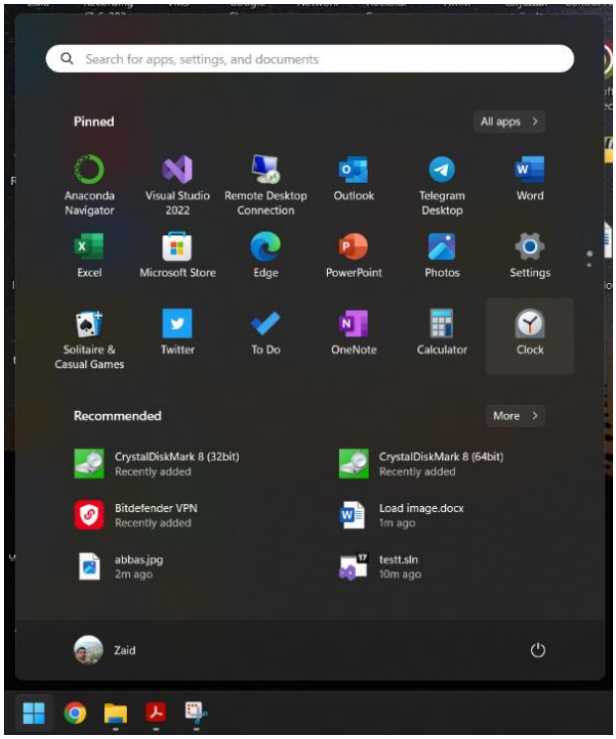
Computer Science Department
قسم علوم الحاسوب

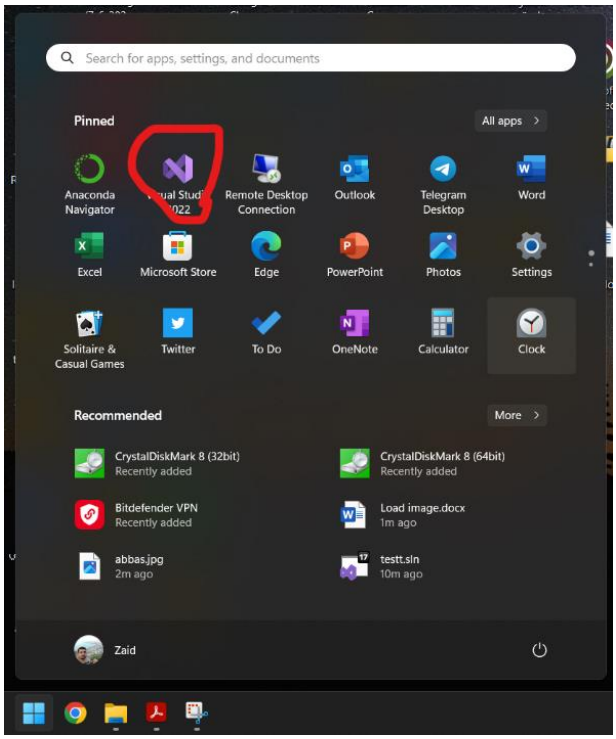
Machine Vision
الرؤية بالماكنة

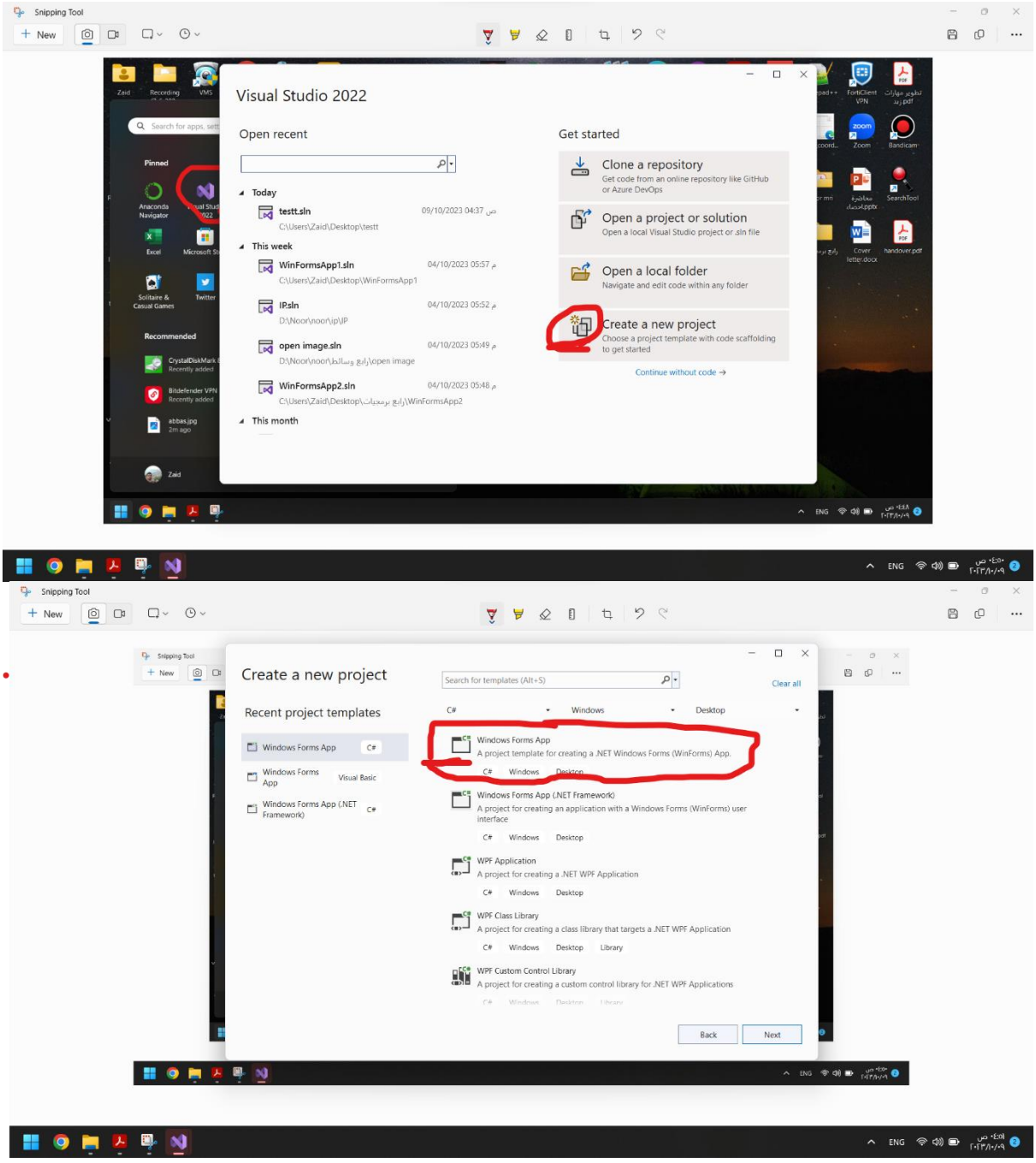
M.Sc. Zena Mohammad Saadi
م.م. زينة محمد سعدي

cs.uotechnology.edu.iq

البرنامج المستخدم visual studio 2022 :





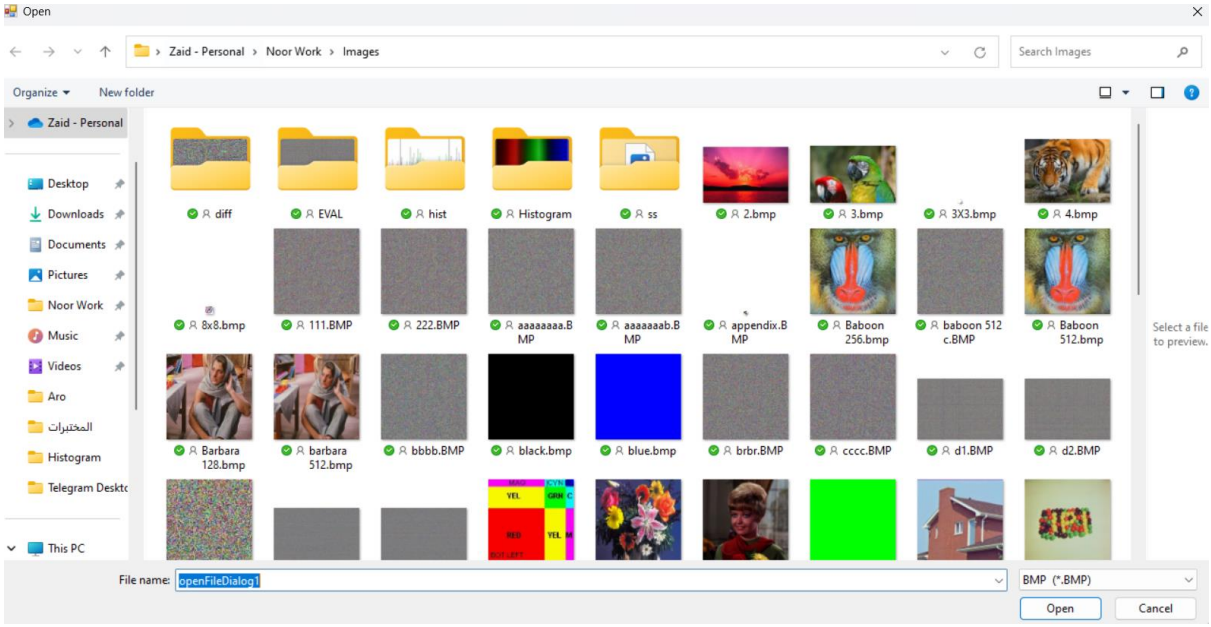

Snipping Tool
New
Visual Studio 2022
Open recent
Today
testt.sln
C:\Users\Zaid\Desktop\testt
This week
WinFormsApp1.sln
C:\Users\Zaid\Desktop\WinFormsApp1
IP.sln
D:\Noor\noor\ip\IP
open image.sln
WinFormsApp2.sln
This month
Get started
Clone a repository
Get code from an online repository like GitHub or Azure DevOps
Open a project or solution
Open a local Visual Studio project or .sln file
Open a local folder
Navigate and edit code within any folder
Create a new project
Choose a project template with code scaffolding to get started
Continue without code →

Snipping Tool
New
Create a new project
Search for templates (Alt+S)
Clear all
Recent project templates
Windows Forms App
Windows Forms App (.NET Framework)
Windows Forms App
A project template for creating a .NET Windows Forms (WinForms) App.
Windows Forms App (.NET Framework)
A project for creating an application with a Windows Forms (WinForms) user interface
WPF Application
A project for creating a .NET WPF Application
WPF Class Library
A project for creating a class library that targets a .NET WPF Application
WPF Custom Control Library
A project for creating a custom control library for .NET WPF Applications
Back
Next

Lecture1:

Load Image

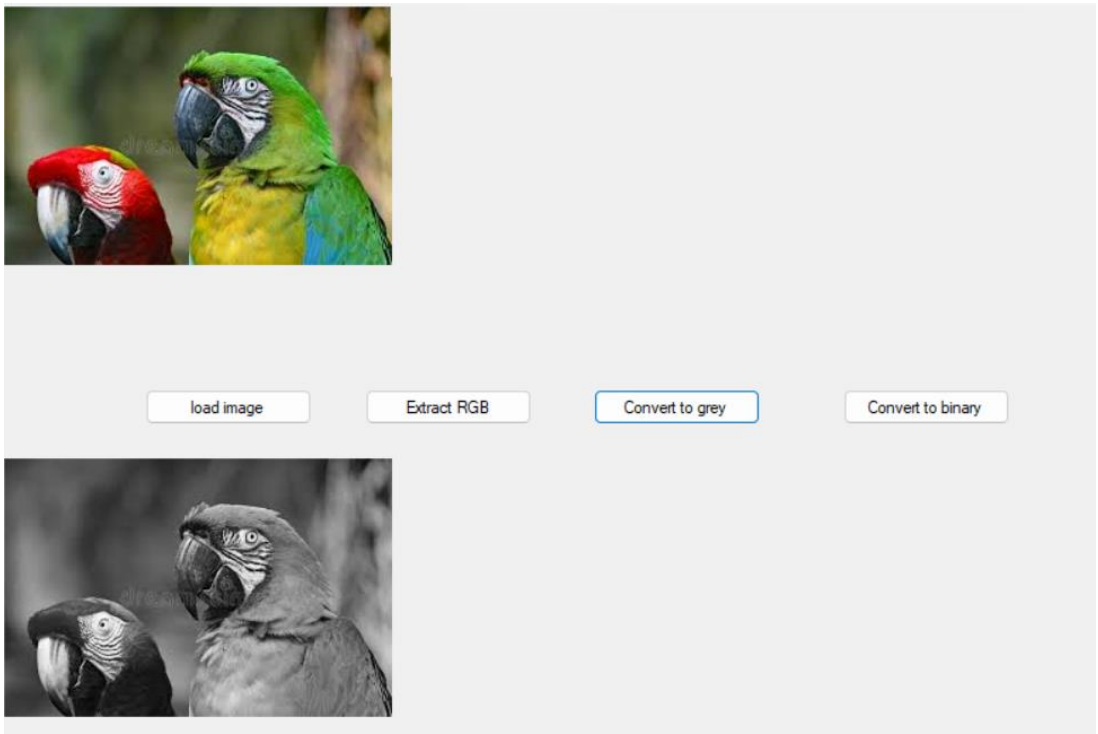
```
openFileDialog1.Filter = "BMP |*.BMP";
        if (openFileDialog1.ShowDialog() == DialogResult.OK)
            pictureBox1.ImageLocation = openFileDialog1.FileName;
        Bitmap orignal_image = Bitmap.FromFile(openFileDialog1.FileName)
as Bitmap;
```



And then choose a picture to open it.

Convert color image to gray

Input: color image (R,G,B) *Output: gray image(one band)*
Begin *grey_image_size = color_image_size* *For each pixel in color_image* *grey_pixel = round (red * 0.3 + green * 0.59 + blue * 0.11)* *set grey_pixel in grey_image* *next* *End*




```
Bitmap input = (Bitmap)pictureBox1.Image;
        Bitmap grey = new Bitmap(input);
        Color cl;
        byte red, green, blue;
```

```
        byte gr;
        for (int i = 0; i < input.Width; i++)
            for (int j = 0; j < input.Height; j++)
            {
                cl = input.GetPixel(i, j);
                red = cl.R;
                green = cl.G;
                blue = cl.B;
                gr = (byte)(red * 0.3 + green * 0.59 + blue * 0.11);
                cl = Color.FromArgb(gr, gr, gr);
                grey.SetPixel(i, j, cl);
            }
pictureBox2.Image = grey;
```

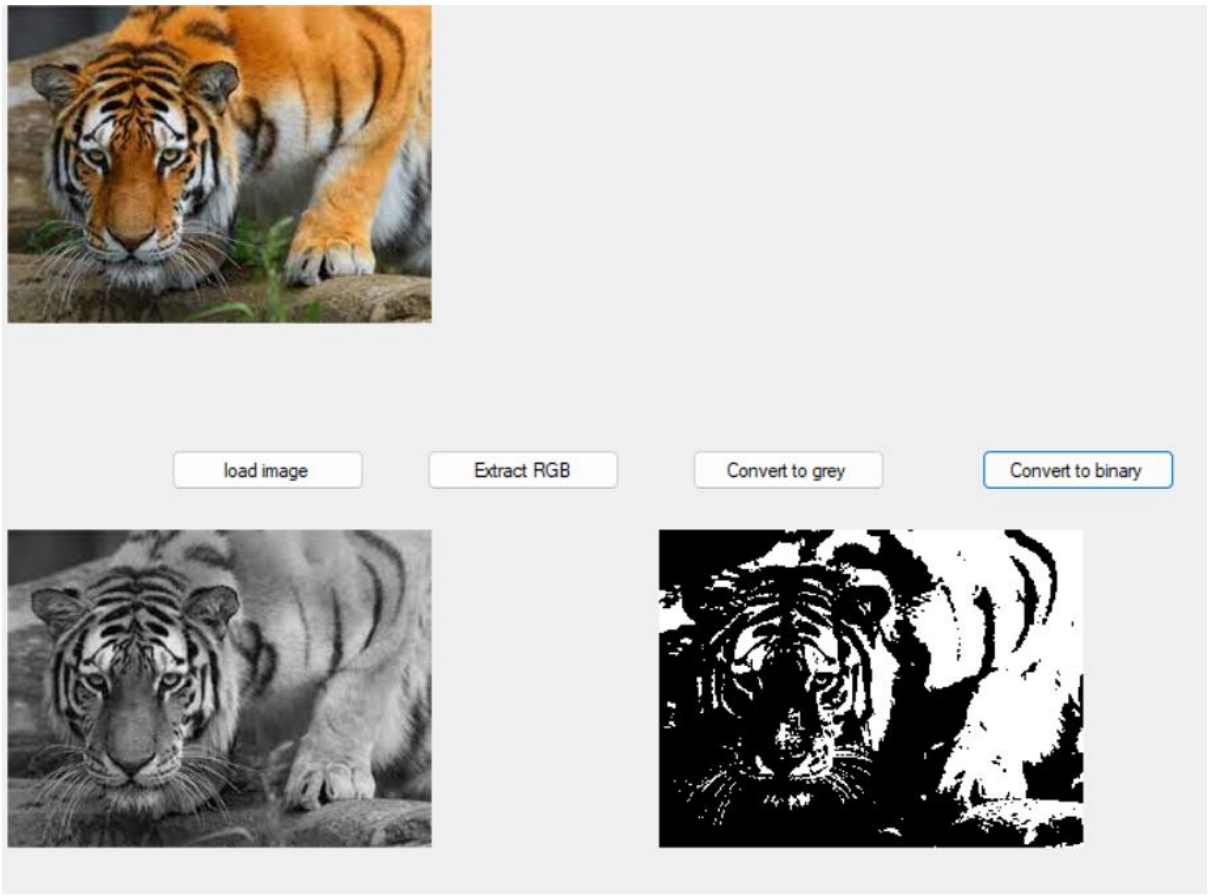
Lecture2:

Convert Image to Binary

Input: color image (R, G, B) *Output: Binary image*

Begin:
binary_image_size = color_image_size
For each pixel in color_image
*grey_pixel = round (red * 0.3 + green * 0.59 + blue * 0.11)*
if grey_pixel > 128 then
binary_pixel = 1
else

```
                    binary_pixel = 0
                set binary_pixel in binary_image
        next
End
```




```
Bitmap input = (Bitmap)pictureBox5.Image;
Bitmap grey = new Bitmap(input);
Color cl;
byte pix;
byte gr;
for (int i = 0; i < input.Width; i++)
    for (int j = 0; j < input.Height; j++)
    { cl = input.GetPixel(i, j);
        pix = cl.R;
```

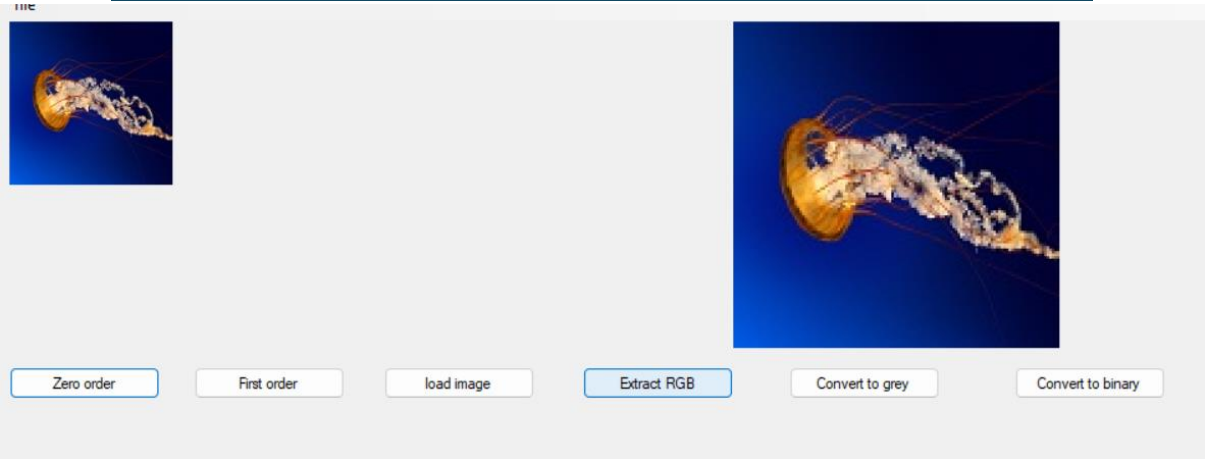
```
        if (pix > 128)
            gr = 255;
        else
            gr = 0;
        cl = Color.FromArgb(gr, gr, gr);
        grey.SetPixel(i, j, cl);
    }
pictureBox6.Image = grey;
```

Lecture3:

Enlarge images

Zero order

Input: Original image. *Output: Return enlarged image.*
Begin: *New_image_width = image_width * 2* *New_image_height = image_height * 2* *For I = 0 to image_width* *For J=0 to image_height* *{* *New_image(2*I,2*J) = image(I,J)* *New_image(2*I,2*J+1) = image(I,J)* *New_image(2*I+1,2*J) = image(I,J)* *New_image(2*I+1,2*J+1) = image(I,J)* *}* *End.*



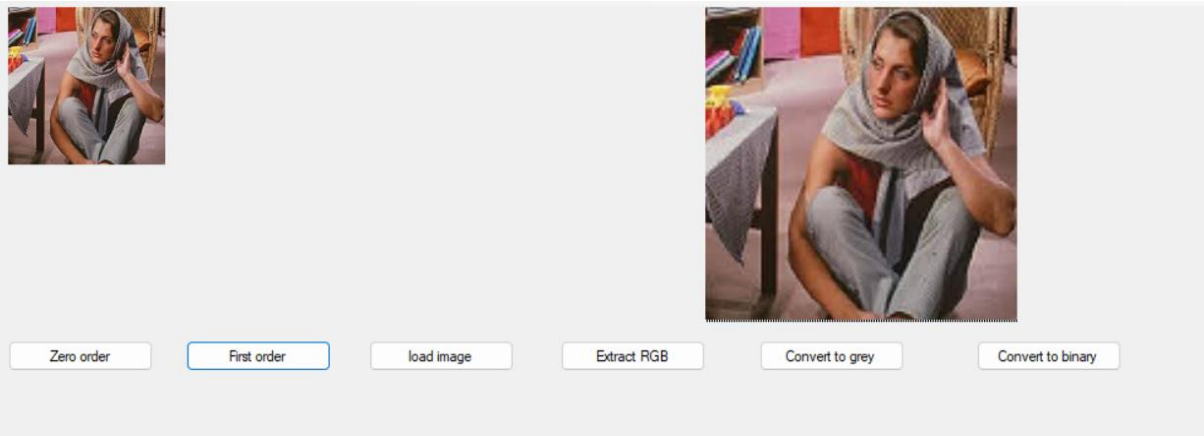

```
Bitmap input = (Bitmap)pictureBox1.Image;
Bitmap zero_order = new Bitmap(input.Width * 2, input.Height * 2);
pictureBox3.Width= input.Width*2;
pictureBox3.Height=input.Height*2;
Color cl;
for (int i = 0; i < input.Width; i++)
   for (int j = 0; j < input.Height; j++)
   {
      cl = input.GetPixel(i, j);
      zero_order.SetPixel(i*2, j*2, cl);
      zero_order.SetPixel(i*2,(j * 2) + 1, cl);
      zero_order.SetPixel((i * 2) + 1, j * 2, cl);
      zero_order.SetPixel((i * 2) + 1, (j * 2) + 1, cl);
   }
pictureBox3.Image = zero_order;
```

Lecture4:

First Order

Input: Original image.
Output: Return enlarged image.

```
Begin:
   New_image_width = (image_width - 1) * 2 + 1
   New_image_height = (image_height - 1) * 2 + 1
   For I = 0 to image_width - 2
      For J = 0 to image_height - 2
      {
         New_image(2*I, 2*J) = image(I, J)
         New_image(2*I + 1, 2*J) = (image(I, J) + image(I + 1, J)) / 2
         New_image(2*I, 2*J + 1) = (image(I, J) + image(I, J + 1)) / 2
         New_image(2*I + 1, 2*J + 1) = (image(I, J) + image(I + 1, J) +
             image(I, J + 1) + image(I + 1, J + 1)) / 4
      }
End.
```




```
Bitmap input = (Bitmap)pictureBox1.Image;
Bitmap temp = new Bitmap(input.Width, input.Height * 2);
Bitmap firstorder = new Bitmap(input.Width * 2, input.Height * 2);
pictureBox3.Width = input.Width * 2;
pictureBox3.Height = input.Height * 2;
Color cl,cl2,cl3;
```

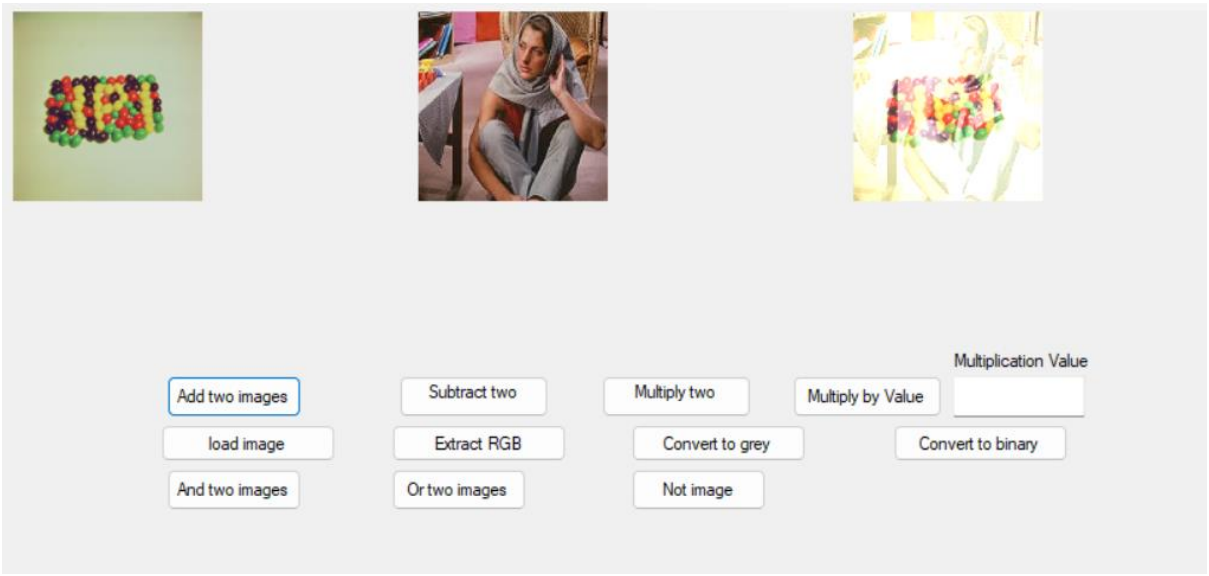
```
byte red, green, blue;
int t;
for (int i = 0; i < input.Width; i++)
   for (int j = 0; j < input.Height-1; j++)
   {
      cl = input.GetPixel(i, j);
      cl2 = input.GetPixel(i, j+1);
      t = cl.R + cl2.R;
      red = (byte)(t/2);
      t= cl.G + cl2.G;
      green = (byte)(t / 2);
      t= cl.B + cl2.B;
      blue = (byte)(t / 2);
      cl3=Color.FromArgb(red, green, blue);
      temp.SetPixel(i, j * 2, cl);
      temp.SetPixel(i, (j * 2) + 1, cl3);
   }
for (int j = 0; j < temp.Height; j++)
   for (int i = 0; i < temp.Width-1; i++)
   {
      cl = temp.GetPixel(i, j);
      cl2 = temp.GetPixel(i + 1, j);
      t = cl.R + cl2.R;
      red = (byte)(t / 2);
      t = cl.G + cl2.G;
      green = (byte)(t / 2);
      t = cl.B + cl2.B;
      blue = (byte)(t / 2);
      cl3 =Color.FromArgb(red, green, blue);
```

```
            firstorder.SetPixel(i*2, j, cl);
            firstorder.SetPixel((i * 2) + 1, j, cl3);
        }
    pictureBox3.Image = firstorder;
```

Lecture5:

Arithmetic Operation

Add two images

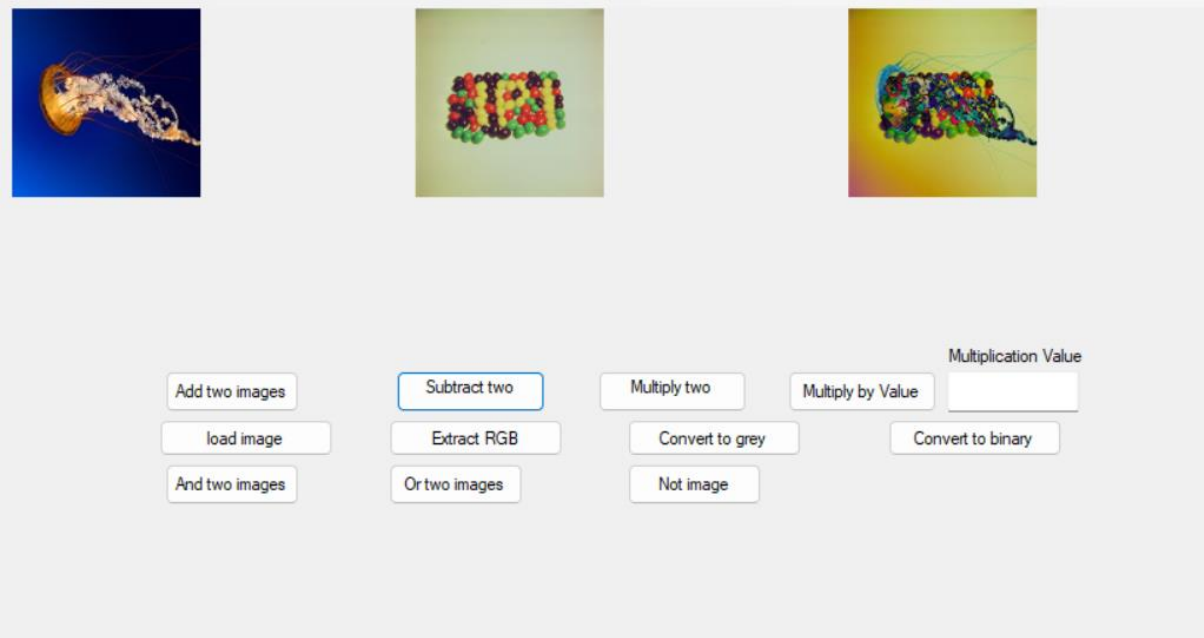



```
private void button1_Click_1(object sender, EventArgs e)
{
   openFileDialog1.Filter = "BMP |*.BMP";
   if (openFileDialog1.ShowDialog() == DialogResult.OK)
      pictureBox1.ImageLocation = openFileDialog1.FileName;
   Bitmap input = Bitmap.FromFile(openFileDialog1.FileName) as Bitmap;
   openFileDialog1.Filter = "BMP |*.BMP";
   if (openFileDialog1.ShowDialog() == DialogResult.OK)
      pictureBox2.ImageLocation = openFileDialog1.FileName;
```

```
    Bitmap input_2 = Bitmap.FromFile(openFileDialog1.FileName) as Bitmap;
    Bitmap output = new Bitmap(input.Width,input.Height);
    Color cl;
    int Red,Green,Blue;
    for (int i = 0; i < input.Width; i++)
        for (int j = 0; j < input.Height; j++)
        {
            Red = input.GetPixel(i, j).R + input_2.GetPixel(i, j).R;
            if (Red > 255)
                Red = 255;
            Green = input.GetPixel(i, j).G + input_2.GetPixel(i, j).G;
            if (Green > 255)
                Green = 255;
            Blue = input.GetPixel(i, j).B+ input_2.GetPixel(i, j).B;
            if (Blue > 255)
                Blue = 255;
            cl = Color.FromArgb(Red, Green, Blue);
            output.SetPixel(i, j, cl);
        }
    pictureBox3.Image = output;
}
```

Lecture6:

Subtract Two Images



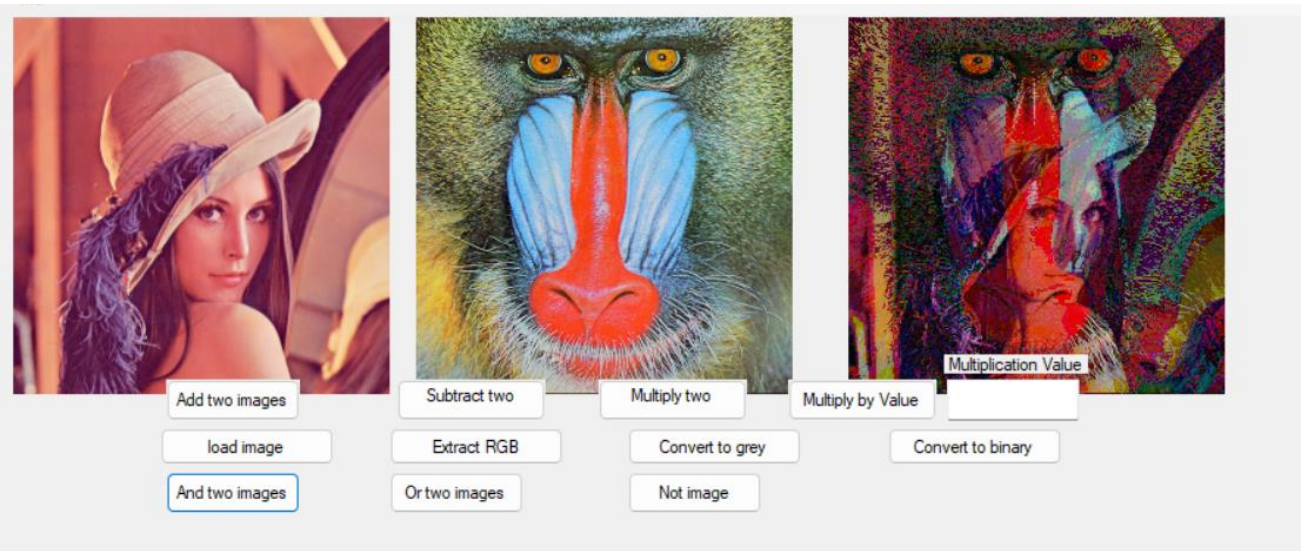

```
private void button5_Click(object sender, EventArgs e)
{
    openFileDialog1.Filter = "BMP |*.BMP";
    if (openFileDialog1.ShowDialog() == DialogResult.OK)
        pictureBox1.ImageLocation = openFileDialog1.FileName;
    Bitmap input = Bitmap.FromFile(openFileDialog1.FileName) as Bitmap;
    openFileDialog1.Filter = "BMP |*.BMP";
    if (openFileDialog1.ShowDialog() == DialogResult.OK)
        pictureBox2.ImageLocation = openFileDialog1.FileName;
    Bitmap input_2 = Bitmap.FromFile(openFileDialog1.FileName) as Bitmap;
    Bitmap output = new Bitmap(input.Width, input.Height);
    Color cl;
    int Red, Green, Blue;
    for (int i = 0; i < input.Width; i++)
```

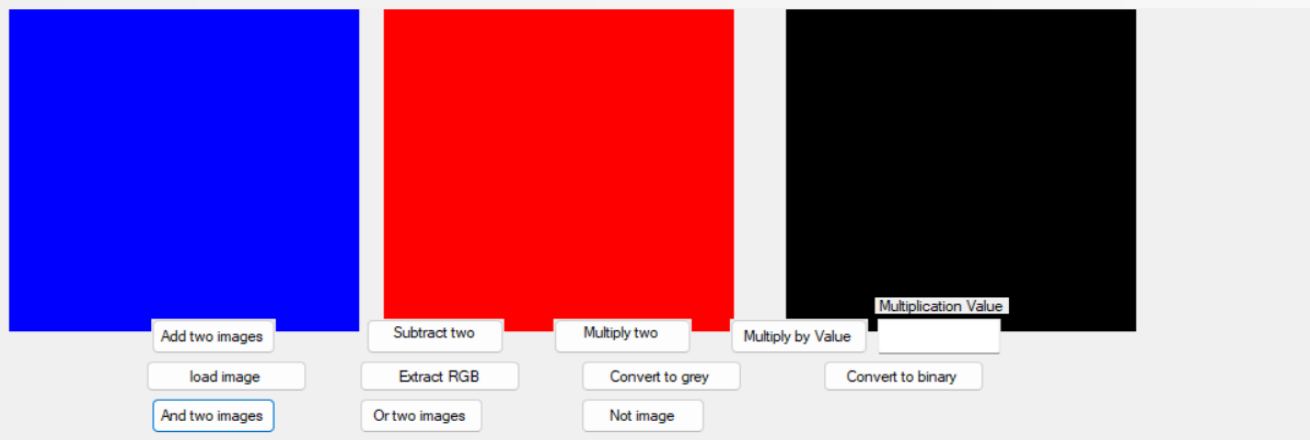
```
        for (int j = 0; j < input.Height; j++)
        {
            Red = Math.Abs(input.GetPixel(i, j).R - input_2.GetPixel(i, j).R);
            Green = Math.Abs(input.GetPixel(i, j).G - input_2.GetPixel(i, j).G);
            Blue = Math.Abs(input.GetPixel(i, j).B - input_2.GetPixel(i, j).B);
            cl = Color.FromArgb(Red, Green, Blue);
            output.SetPixel(i, j, cl);
        }
    pictureBox3.Image = output;

}
```

Lecture7:

Logic Operation

AND Operation






```
openFileDialog1.Filter = "BMP |*.BMP";
 if (openFileDialog1.ShowDialog() == DialogResult.OK)
    pictureBox1.ImageLocation = openFileDialog1.FileName;
 Bitmap input = Bitmap.FromFile(openFileDialog1.FileName) as Bitmap;
 openFileDialog1.Filter = "BMP |*.BMP";
 if (openFileDialog1.ShowDialog() == DialogResult.OK)
    pictureBox2.ImageLocation = openFileDialog1.FileName;
 Bitmap input_2 = Bitmap.FromFile(openFileDialog1.FileName) as Bitmap;
 Bitmap output = new Bitmap(input.Width, input.Height);
 Color cl;
 int Red, Green, Blue;
 for (int i = 0; i < input.Width; i++)
    for (int j = 0; j < input.Height; j++)
    {
       Red = input.GetPixel(i, j).R & input_2.GetPixel(i, j).R;
       Green = input.GetPixel(i, j).G & input_2.GetPixel(i, j).G;
       Blue = input.GetPixel(i, j).B & input_2.GetPixel(i, j).B;
       cl = Color.FromArgb(Red, Green, Blue);
       output.SetPixel(i, j, cl);
    }
 pictureBox3.Image = output;
```

Lecture8:

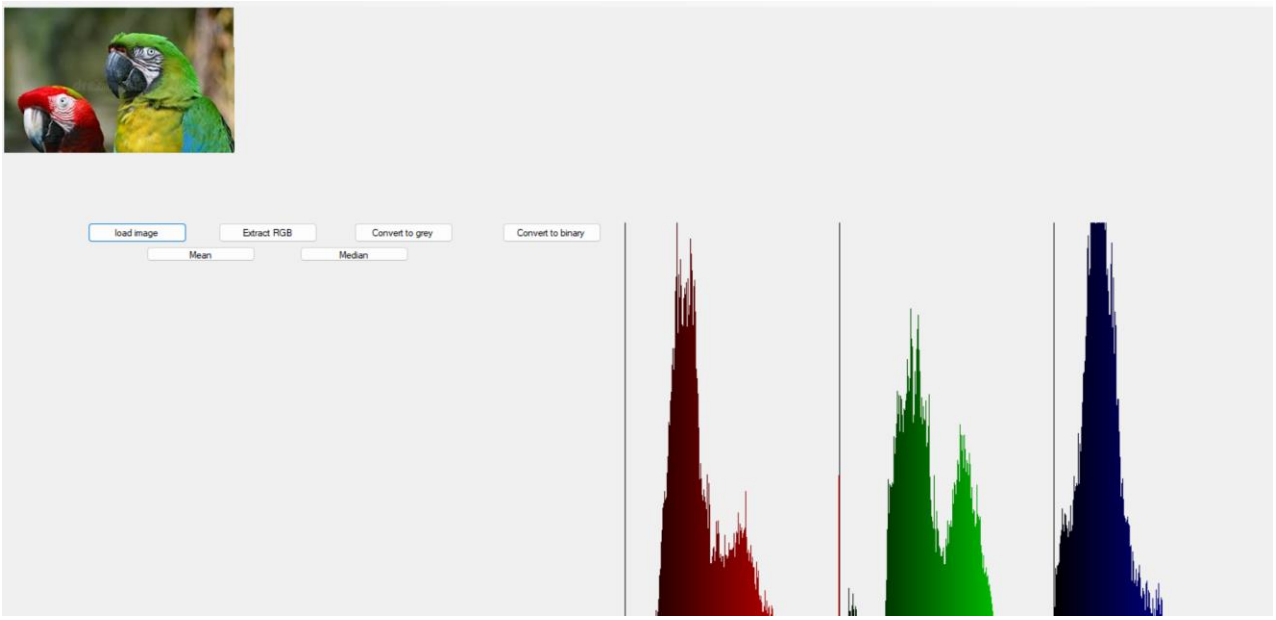
OR operation of two images


```
openFileDialog1.Filter = "BMP |*.BMP";
if (openFileDialog1.ShowDialog() == DialogResult.OK)
    pictureBox1.ImageLocation = openFileDialog1.FileName;
Bitmap input = Bitmap.FromFile(openFileDialog1.FileName) as Bitmap;
openFileDialog1.Filter = "BMP |*.BMP";
if (openFileDialog1.ShowDialog() == DialogResult.OK)
    pictureBox2.ImageLocation = openFileDialog1.FileName;
Bitmap input_2 = Bitmap.FromFile(openFileDialog1.FileName) as Bitmap;
Bitmap output = new Bitmap(input.Width, input.Height);
Color cl;
int Red, Green, Blue;

for (int i = 0; i < input.Width; i++)
    for (int j = 0; j < input.Height; j++)
    {
        Red = input.GetPixel(i, j).R | input_2.GetPixel(i, j).R;
        Green = input.GetPixel(i, j).G | input_2.GetPixel(i, j).G;
```

```
        Blue = input.GetPixel(i, j).B | input_2.GetPixel(i, j).B;
        cl = Color.FromArgb(Red, Green, Blue);
        output.SetPixel(i, j, cl);
    }
pictureBox3.Image = output;
```

Lecture9:

Histogram




```
Bitmap Histogram(Bitmap image)
{
    int max = 0;
    int[,] Hist;
    Hist = new int[256, 3];
    for (int i = 0; i < image.Width; i++)
        for (int j = 0; j < image.Height; j++)
        {
            Hist[image.GetPixel(i, j).R, 0]++;
            if (Hist[image.GetPixel(i, j).R, 0] > max)
                max = Hist[image.GetPixel(i, j).R, 0];
```

```
            Hist[image.GetPixel(i, j).G, 1]++;
            if (Hist[image.GetPixel(i, j).G, 0] > max)
                max = Hist[image.GetPixel(i, j).G, 0];
            Hist[image.GetPixel(i, j).B, 2]++;
            if (Hist[image.GetPixel(i, j).B, 0] > max)
                max = Hist[image.GetPixel(i, j).B, 0];


        }
    return Histogram2Bitmap(Hist,max);

}
```

Lecture10:

Histogram2Bitmap

```
Bitmap Histogram2Bitmap(int[,] h,int max)
{
    if (max == 0)
        max = 256;
    Bitmap hist = new Bitmap(768, max);
    Graphics g = Graphics.FromImage(hist);
    for (int i = 1; i < 256; i++)
    {


        g.DrawLine(new Pen(Color.FromArgb(i, 0, 0)), i, max, i, max - h[i, 0]);
        g.DrawLine(new Pen(Color.FromArgb(0, i, 0)), i + 256, max, i + 256, max
- h[i, 1]);
        g.DrawLine(new Pen(Color.FromArgb(0, 0, i)), i + 512, max, i + 512, max
- h[i, 2]);
    }
```

```
    g.DrawLine(new Pen(Color.FromArgb(0, 0, 0)), 0, 0, 0, max);
    g.DrawLine(new Pen(Color.FromArgb(0, 0, 0)), 256, 0, 256, max);
    g.DrawLine(new Pen(Color.FromArgb(0, 0, 0)), 512, 0, 512, max);
 return hist;
}
```