

AD SOYAD CENGİZHAN TOPÇU
NUMARA 2017010225048

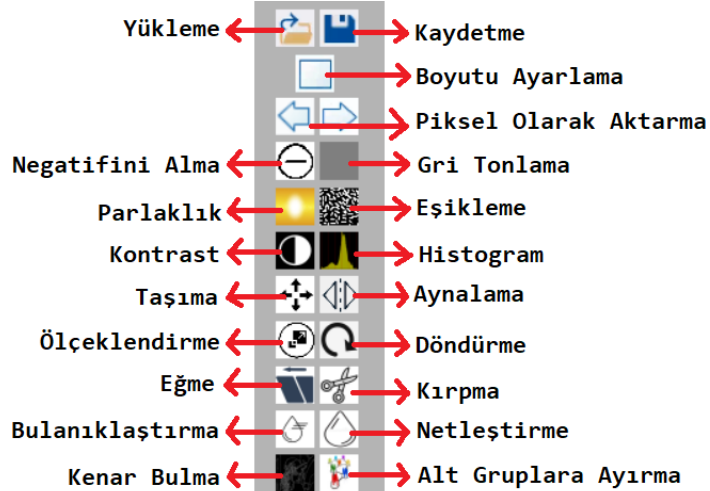
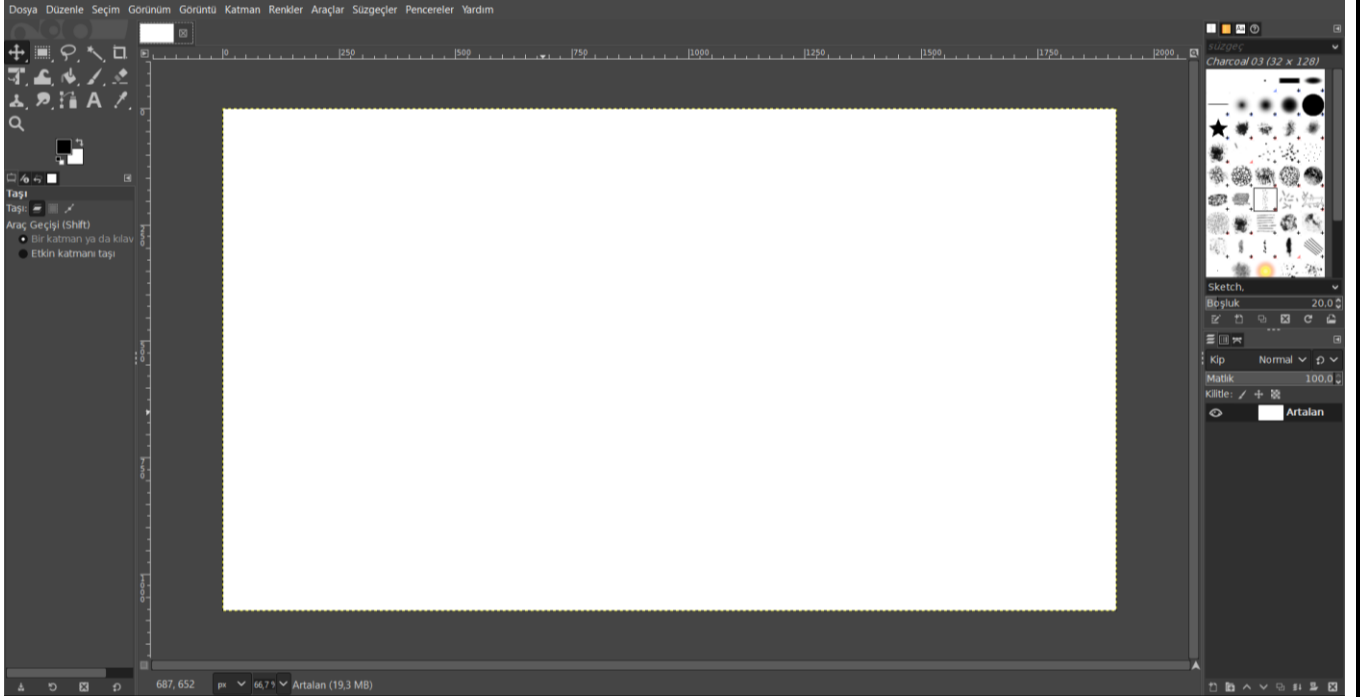
GÖRÜNTÜ İŞLEME ÖDEV-7

KARABÜK ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
MEKATRONİK MÜHENDİSLİĞİ

Görüntü İşleme Ödev-7



[Bağıksız]-2.0 (RGB rengi 8-bit gama tam sayı değeri, GIMP built-in sRGB, 1 katman) 1920x1080 - GIMP



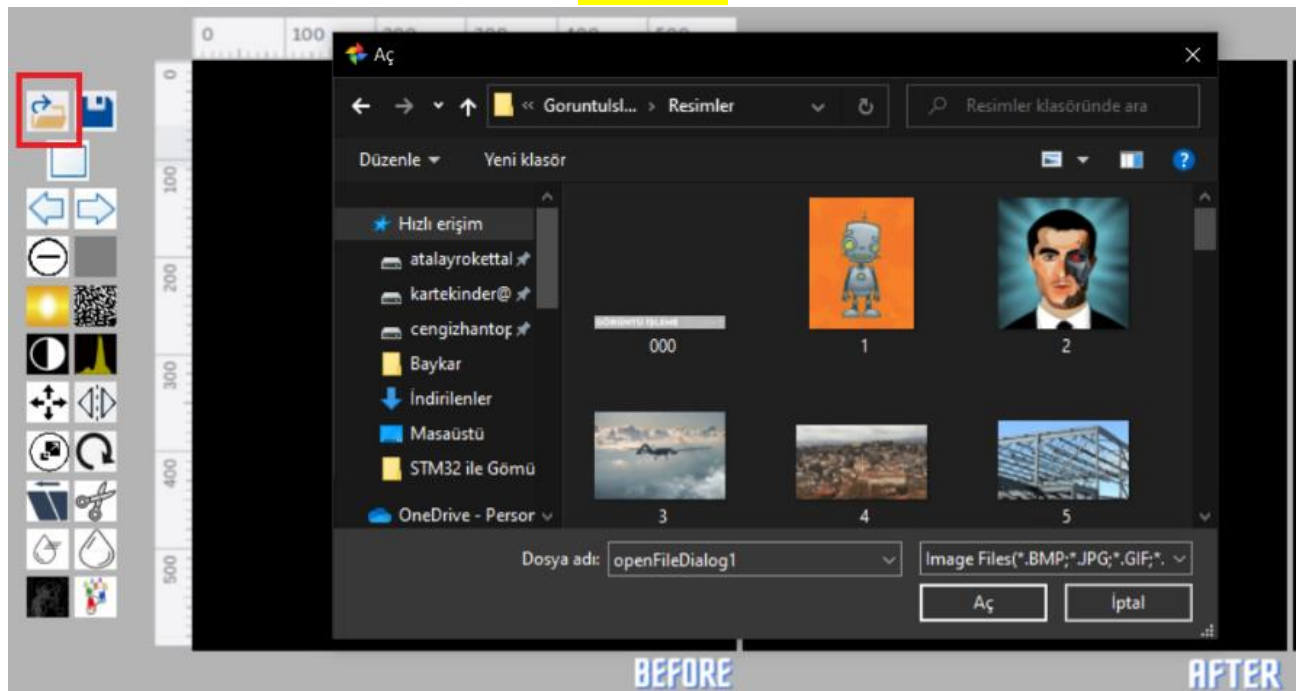
```

using System;
using System.Collections;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.IO;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace GoruntuIsleme_DersNotlari_Ödev7
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
    }
}

```

1.YÜKLEME



```

public void DosyaAc()
{
    try
    {
        openFileDialog1.DefaultExt = ".jpg";
        openFileDialog1.Filter = "Image
Files (*.BMP;*.JPG;*.GIF;*.PNG)|*.BMP;*.JPG;*.GIF;*.PNG|All files (*.*)|*.*";
        openFileDialog1.ShowDialog();
        String ResminYolu = openFileDialog1.FileName;
        pictureBox1.Image = Image.FromFile(ResminYolu);
    }
    catch
    {
        MessageBox.Show("HATA! Resim yüklenmedi");
    }
}

private void pbDosyaAc_Click(object sender, EventArgs e)
{
    DosyaAc();
}

private void DosyaAc_menu_Click(object sender, EventArgs e)

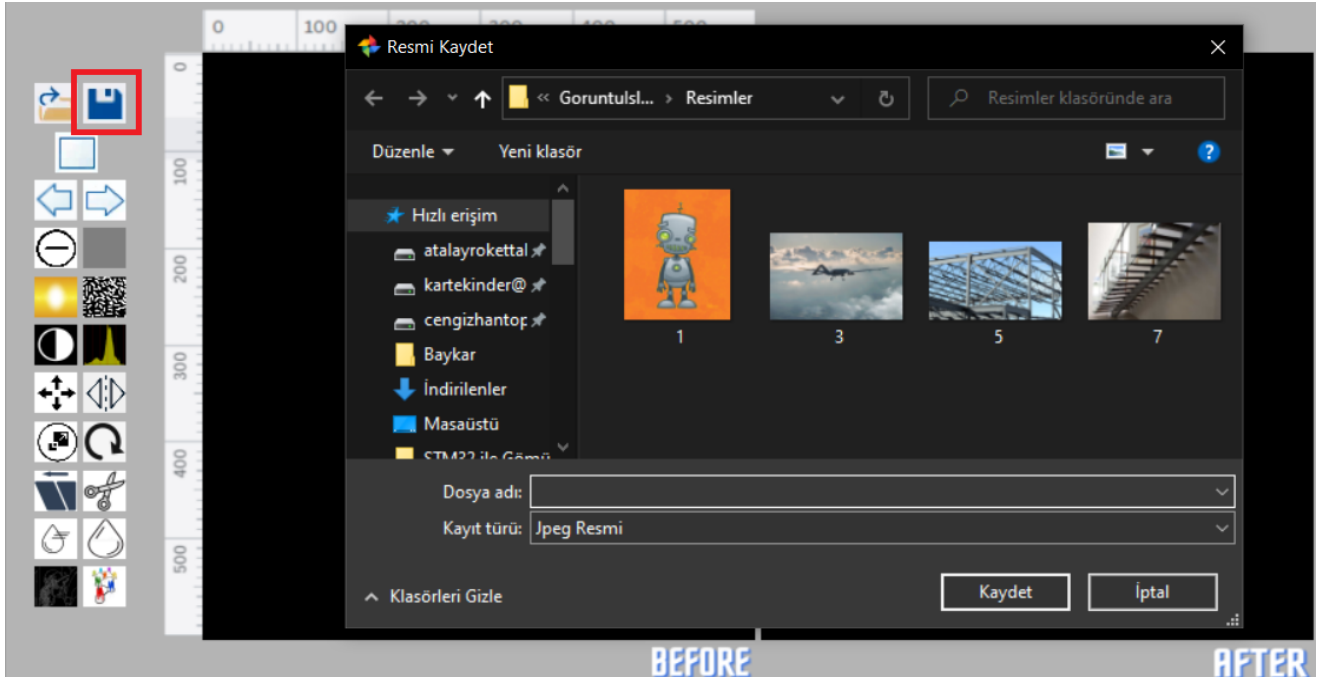
```

```

{
    DosyaAc ();
}

```

2.KAYDETME



```

public void ResmiKaydet ()
{
    SaveFileDialog saveFileDialog1 = new SaveFileDialog ();
    saveFileDialog1.Filter = "Jpeg Resmi|*.jpg|Bitmap Resmi|*.bmp|Gif
Resmi|*.gif";
    saveFileDialog1.Title = "Resmi Kaydet";
    saveFileDialog1.ShowDialog ();

    if (saveFileDialog1.FileName != "") //Dosya adı boş değilse kaydedecek.
    {
        // FileStream nesnesi ile kayıttı gerçekleştirecek.
        FileStream DosyaAkisi = (FileStream)saveFileDialog1.OpenFile ();

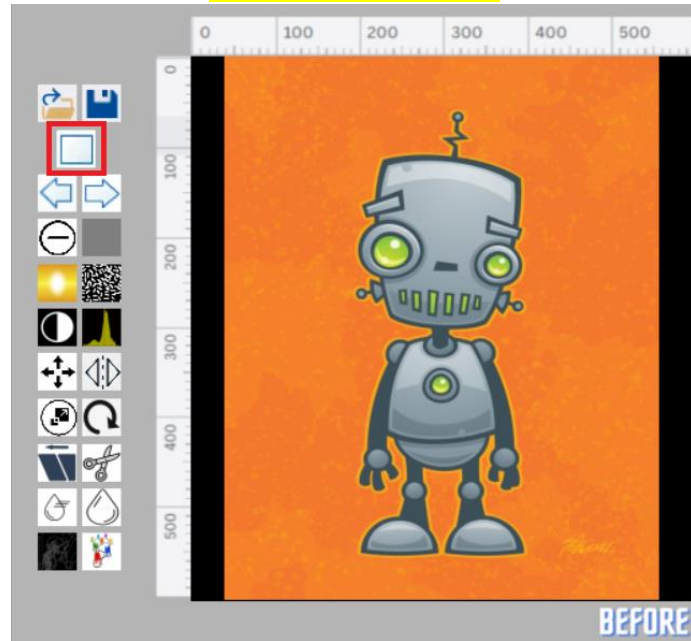
        switch (saveFileDialog1.FilterIndex)
        {
            case 1:
                pictureBox1.Image.Save (DosyaAkisi,
System.Drawing.Imaging.ImageFormat.Jpeg);
                break;
            case 2:
                pictureBox1.Image.Save (DosyaAkisi,
System.Drawing.Imaging.ImageFormat.Bmp);
                break;
            case 3:
                pictureBox1.Image.Save (DosyaAkisi,
System.Drawing.Imaging.ImageFormat.Gif);
                break;
        }
        DosyaAkisi.Close ();
    }
}

private void pcKaydet_Click(object sender, EventArgs e)
{
    ResmiKaydet ();
}

private void Kaydet_menu_Click(object sender, EventArgs e)
{
    ResmiKaydet ();
}

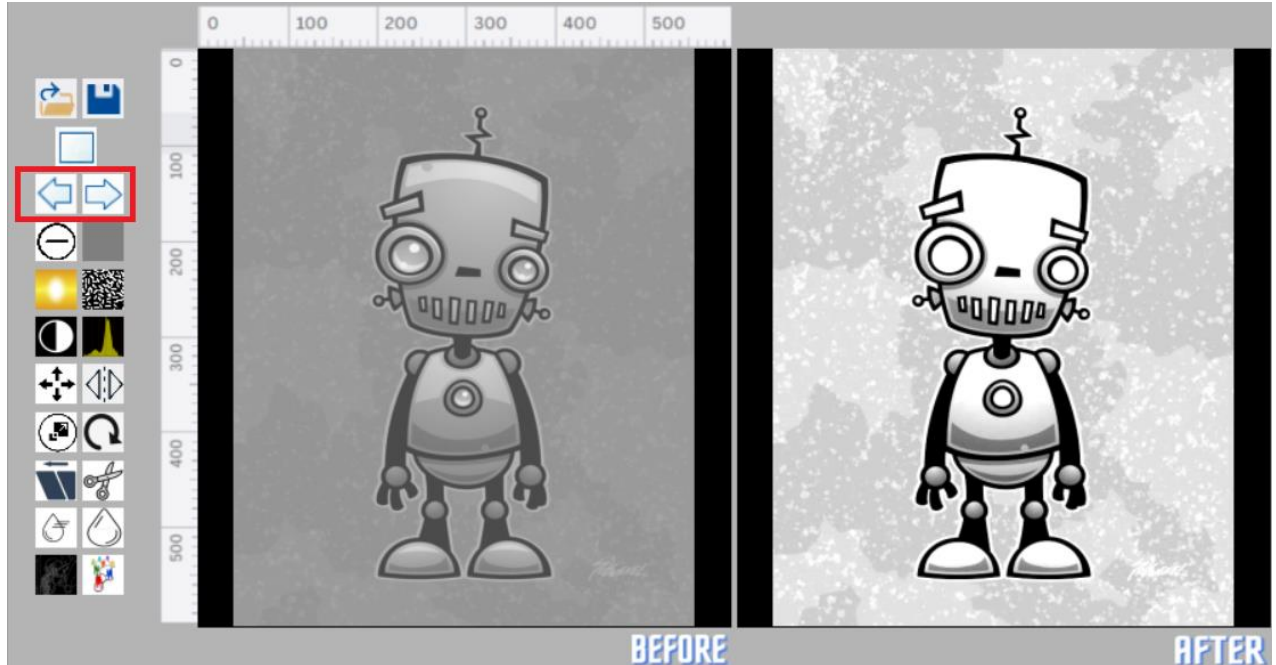
```

3.BOYUTU AYARLAMA



```
public void BoyutAyarla()  
{  
    pictureBox1.SizeMode = PictureBoxSizeMode.Zoom;  
    pictureBox2.SizeMode = PictureBoxSizeMode.Zoom;  
}  
  
private void pbBoyutAyarla_Click(object sender, EventArgs e)  
{  
    BoyutAyarla();  
}
```

4.PİKSEL OLARAK AKTARMA



```
public void SagaAktarma()  
{  
    //pictureBox2.Image = pictureBox1.Image;  
  
    Color OkunanRenk, DonusenRenk;  
    int R = 0, G = 0, B = 0;  
  
    Bitmap GirisResmi, CikisResmi;  
    GirisResmi = new Bitmap(pictureBox1.Image);  
  
    int ResimGenisligi = GirisResmi.Width; //GirisResmi global tanımlandı.
```

İçerisine görüntü yüklendi.

```
int ResimYuksekligi = GirişResmi.Height;
CikisResmi = new Bitmap(ResimGenisligi, ResimYuksekligi); //Cikis
resmini oluşturuyor. Boyutları giriş resmi ile aynı olur. Tanımlaması globalde
yapıldı.

int i = 0, j = 0; //Çıkış resminin x ve y si olacak.
for (int x = 0; x < ResimGenisligi; x++)
{
    for (int y = 0; y < ResimYuksekligi; y++)
    {
        OkunanRenk = GirişResmi.GetPixel(x, y);

        R = OkunanRenk.R;
        G = OkunanRenk.G;
        B = OkunanRenk.B;

        DonusenRenk = Color.FromArgb(R, G, B);
        CikisResmi.SetPixel(x, y, DonusenRenk);
    }
}
pictureBox2.Image = CikisResmi;
}

private void pbSagaAktarma_Click(object sender, EventArgs e)
{
    SagaAktarma();
}

public void SolaAktarma()
{
    Color OkunanRenk, DonusenRenk;
    int R = 0, G = 0, B = 0;

    Bitmap GirişResmi, CikisResmi;
    GirişResmi = new Bitmap(pictureBox2.Image);

    int ResimGenisligi = GirişResmi.Width; //GirişResmi global tanımlandı.
İçerisine görüntü yüklendi.
    int ResimYuksekligi = GirişResmi.Height;
    CikisResmi = new Bitmap(ResimGenisligi, ResimYuksekligi); //Cikis
resmini oluşturuyor. Boyutları giriş resmi ile aynı olur. Tanımlaması globalde
yapıldı.

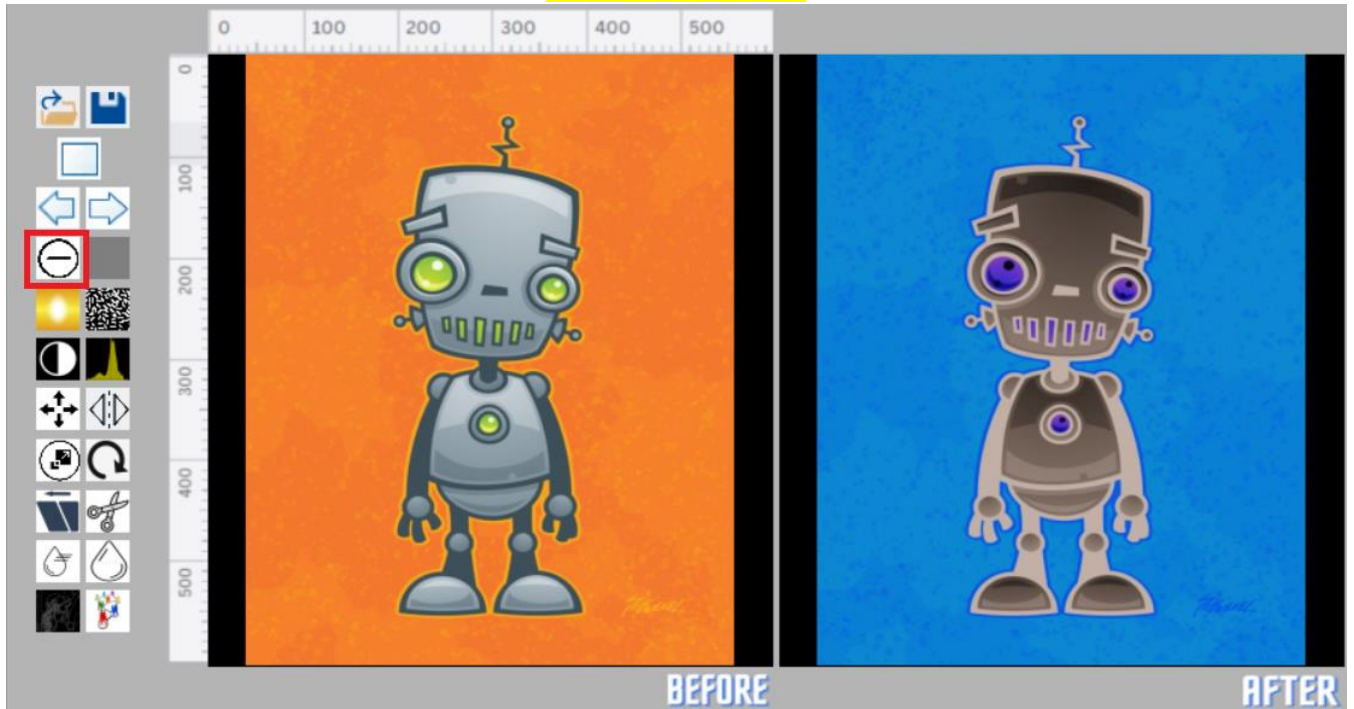
    int i = 0, j = 0; //Çıkış resminin x ve y si olacak.
    for (int x = 0; x < ResimGenisligi; x++)
    {
        for (int y = 0; y < ResimYuksekligi; y++)
        {
            OkunanRenk = GirişResmi.GetPixel(x, y);

            R = OkunanRenk.R;
            G = OkunanRenk.G;
            B = OkunanRenk.B;

            DonusenRenk = Color.FromArgb(R, G, B);
            CikisResmi.SetPixel(x, y, DonusenRenk);
        }
    }
    pictureBox1.Image = CikisResmi;
}

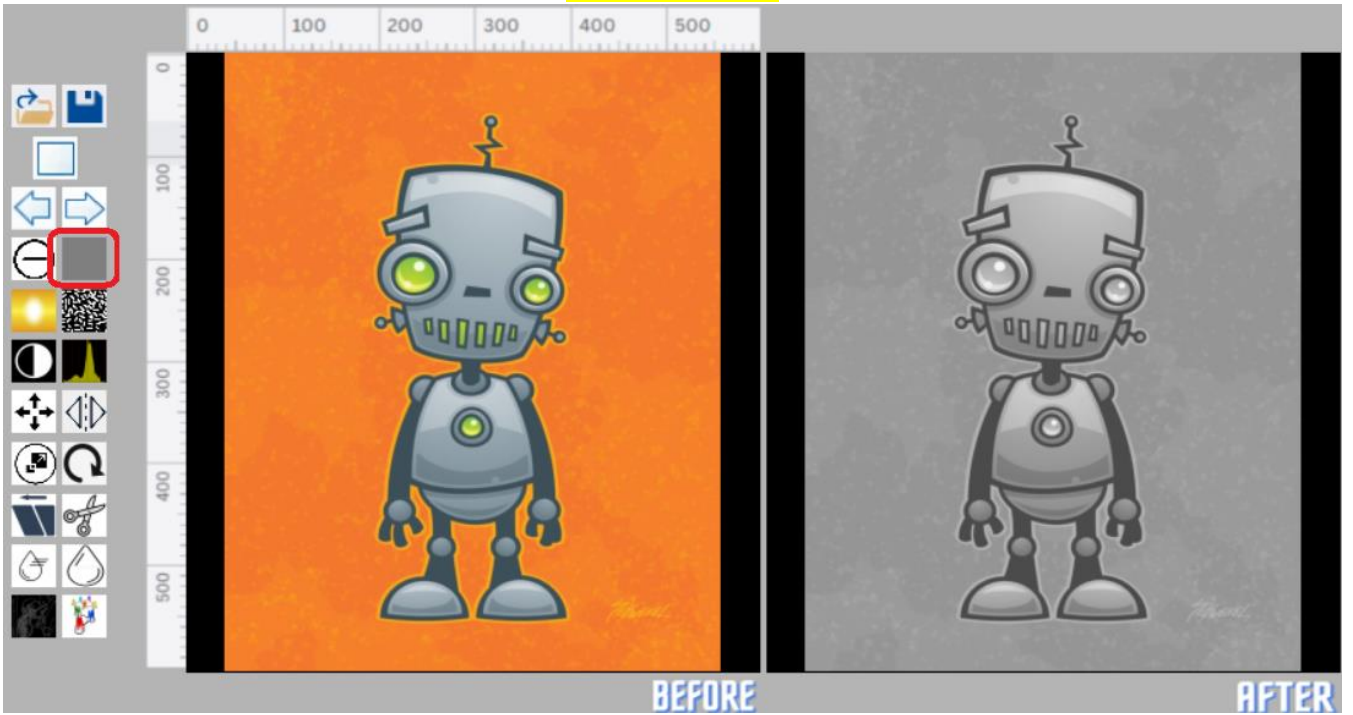
private void pictureBox11_Click(object sender, EventArgs e)
{
    SolaAktarma();
}
```

5.NEGATİFİNİ ALMA



```
public void Negatif()  
{  
    Color OkunanRenk, DonusenRenk;  
    int R = 0, G = 0, B = 0;  
  
    Bitmap GirisResmi, CikisResmi;  
    GirisResmi = new Bitmap(pictureBox1.Image);  
  
    int ResimGenisligi = GirisResmi.Width; //GirisResmi global tanımlandı.  
    İçerisine görüntü yüklendi.  
    int ResimYuksekligi = GirisResmi.Height;  
    CikisResmi = new Bitmap(ResimGenisligi, ResimYuksekligi); //Cikis  
    resmini oluşturuyor. Boyutları giriş resmi ile aynı olur. Tanımlaması globalde  
    yapıldı.  
  
    int i = 0, j = 0; //Çıkış resminin x ve y si olacak.  
    for (int x = 0; x < ResimGenisligi; x++)  
    {  
        for (int y = 0; y < ResimYuksekligi; y++)  
        {  
            OkunanRenk = GirisResmi.GetPixel(x, y);  
            R = 255 - OkunanRenk.R;  
            G = 255 - OkunanRenk.G;  
            B = 255 - OkunanRenk.B;  
            DonusenRenk = Color.FromArgb(R, G, B);  
            CikisResmi.SetPixel(x, y, DonusenRenk);  
        }  
    }  
    pictureBox2.Image = CikisResmi;  
}  
  
private void pbNegatif_Click(object sender, EventArgs e)  
{  
    Negatif();  
}
```


6.GRİ TONLAMA



```
public void Gri()
{
    Color OkunanRenk, DonusenRenk;
    int R = 0, G = 0, B = 0;

    Bitmap GirisResmi, CikisResmi;
    GirisResmi = new Bitmap(pictureBox1.Image);

    int ResimGenisligi = GirisResmi.Width; //GirisResmi global tanımlandı.
    İçerisine görüntü yüklendi.
    int ResimYuksekligi = GirisResmi.Height;
    CikisResmi = new Bitmap(ResimGenisligi, ResimYuksekligi); //Cikis
    resmini oluşturuyor. Boyutları giriş resmi ile aynı olur. Tanımlaması globalde
    yapıldı.

    int i = 0, j = 0; //Çıkış resminin x ve y si olacak.
    for (int x = 0; x < ResimGenisligi; x++)
    {
        for (int y = 0; y < ResimYuksekligi; y++)
        {
            OkunanRenk = GirisResmi.GetPixel(x, y);

            R = OkunanRenk.R;
            G = OkunanRenk.G;
            B = OkunanRenk.B;

            int Gri = Convert.ToInt16(R * 0.3 + G * 0.6 + B * 0.1); //Gri
Ton

            DonusenRenk = Color.FromArgb(Gri, Gri, Gri);
            CikisResmi.SetPixel(x, y, DonusenRenk);

        }
    }
    pictureBox2.Image = CikisResmi;
}

private void pcGri_Click(object sender, EventArgs e)
{
    Gri();
}
```

7.PARLAKLIK



```
public void Parlaklik()
{
    int R = 0, G = 0, B = 0;
    Color OkunanRenk, DonusenRenk;
    Bitmap GirisResmi, CikisResmi;
    GirisResmi = new Bitmap(pictureBox1.Image);

    int ResimGenisligi = GirisResmi.Width; //GirisResmi global tanımlandı.
    int ResimYuksekligi = GirisResmi.Height;
    CikisResmi = new Bitmap(ResimGenisligi, ResimYuksekligi); //Cikis
    resmi oluşturun. Boyutları giriş resmi ile aynı olur.

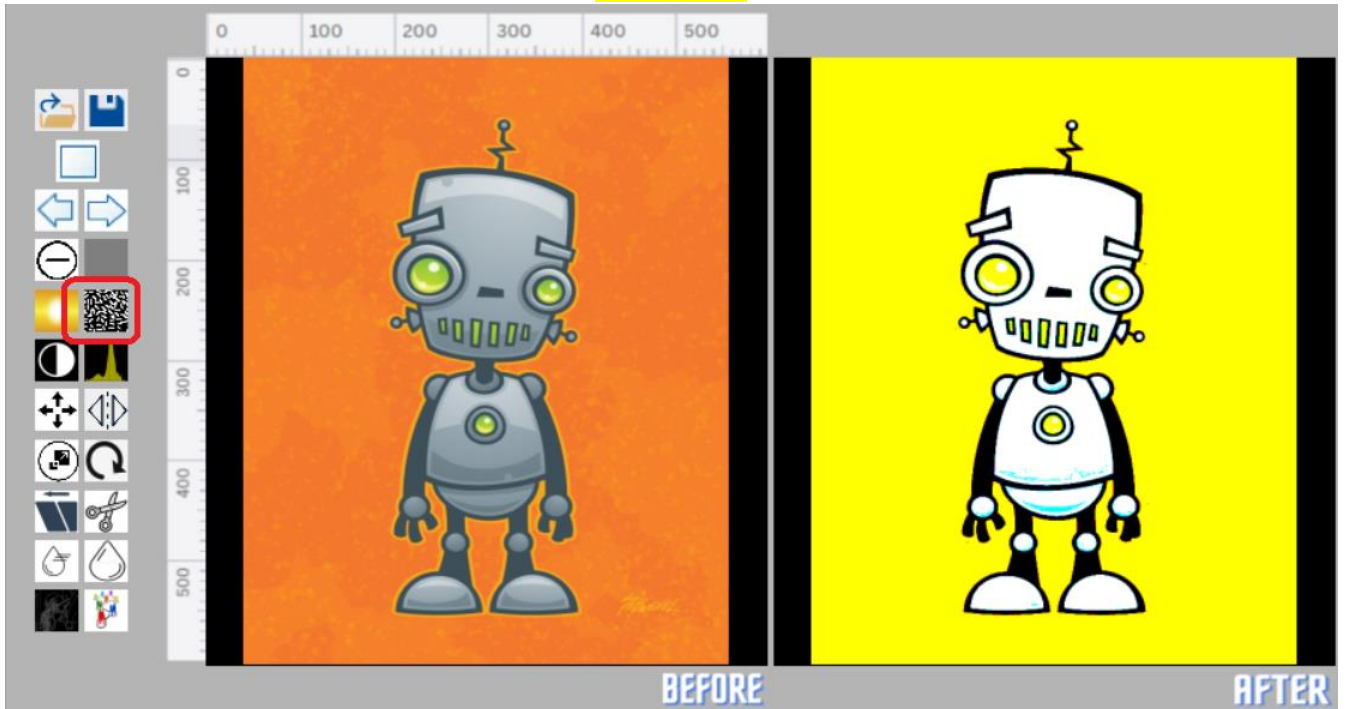
    int i = 0, j = 0;
    //Çıkış resminin x ve y si olacak.
    for (int x = 0; x < ResimGenisligi; x++)
    {
        j = 0;
        for (int y = 0; y < ResimYuksekligi; y++)
        {
            OkunanRenk = GirisResmi.GetPixel(x, y);
            //Rengini 50 değeri ile açacak.
            R = OkunanRenk.R + 50; G = OkunanRenk.G + 50;
            B = OkunanRenk.B + 50;

            //Renkler 255 geçtiyse son sınır olan 255 alınacak.
            if (R > 255) R = 255;
            if (G > 255) G = 255;
            if (B > 255) B = 255;

            DonusenRenk = Color.FromArgb(R, G, B);
            CikisResmi.SetPixel(i, j, DonusenRenk);
            j++;
        }
        i++;
    }
    pictureBox2.Image = CikisResmi;
}

private void pcParlaklik_Click(object sender, EventArgs e)
{
    Parlaklik();
}
```


8.EŞİKLEME



```
public void Esikleme()
{
    int R = 0, G = 0, B = 0;
    Color OkunanRenk, DonusenRenk;
    Bitmap GirisResmi, CikisResmi;
    GirisResmi = new Bitmap(pictureBox1.Image);

    int ResimGenisligi = GirisResmi.Width; //GirisResmi global tanımlandı.
    int ResimYuksekligi = GirisResmi.Height;
    CikisResmi = new Bitmap(ResimGenisligi, ResimYuksekligi); //Cikis
    resmi oluşturun. Boyutları giriş resmi ile aynı olur.

    int EsiklemeDegeri = 110;

    for (int x = 0; x < ResimGenisligi; x++)
    {
        for (int y = 0; y < ResimYuksekligi; y++)
        {
            OkunanRenk = GirisResmi.GetPixel(x, y);

            if (OkunanRenk.R >= EsiklemeDegeri)
                R = 255;
            else R = 0;

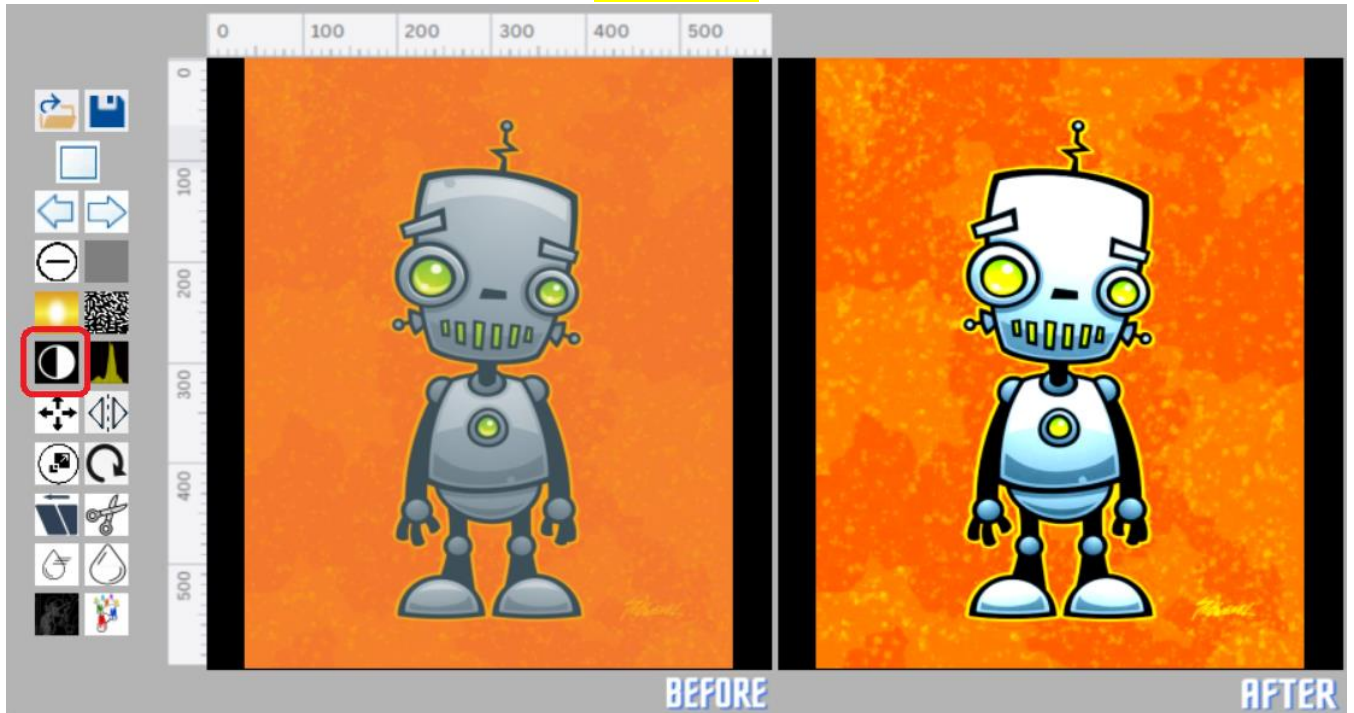
            if (OkunanRenk.G >= EsiklemeDegeri)
                G = 255;
            else G = 0;

            if (OkunanRenk.B >= EsiklemeDegeri)
                B = 255;
            else B = 0;

            DonusenRenk = Color.FromArgb(R, G, B);
            CikisResmi.SetPixel(x, y, DonusenRenk);
        }
    }
    pictureBox2.Image = CikisResmi;
}

private void pbEsikleme_Click(object sender, EventArgs e)
{
    Esikleme();
}
```

9.KONTRAST



```
public void Kontrast()
{
    int R = 0, G = 0, B = 0;
    Color OkunanRenk, DonusenRenk;
    Bitmap GirisResmi, CikisResmi;
    GirisResmi = new Bitmap(pictureBox1.Image);

    int ResimGenisligi = GirisResmi.Width; //GirisResmi global tanımlandı.
    int ResimYuksekligi = GirisResmi.Height;
    CikisResmi = new Bitmap(ResimGenisligi, ResimYuksekligi); //Cikis
    resmi oluşturuyor. Boyutları giriş resmi ile aynı olur.

    double C_KontrastSeviyesi = 150;

    double F_KontrastFaktoru = (259 * (C_KontrastSeviyesi + 255)) / (255 *
    (259 - C_KontrastSeviyesi));

    for (int x = 0; x < ResimGenisligi; x++)
    {
        for (int y = 0; y < ResimYuksekligi; y++)
        {
            OkunanRenk = GirisResmi.GetPixel(x, y);
            R = OkunanRenk.R;
            G = OkunanRenk.G;
            B = OkunanRenk.B;

            R = (int)((F_KontrastFaktoru * (R - 128)) + 128);
            G = (int)((F_KontrastFaktoru * (G - 128)) + 128);
            B = (int)((F_KontrastFaktoru * (B - 128)) + 128);

            //Renkler sınırların dışına çıktıysa, sınır değeri alınacak.
            if (R > 255) R = 255;
            if (G > 255) G = 255;
            if (B > 255) B = 255;

            if (R < 0) R = 0;
            if (G < 0) G = 0;
            if (B < 0) B = 0;

            DonusenRenk = Color.FromArgb(R, G, B);
            CikisResmi.SetPixel(x, y, DonusenRenk);
        }
    }
}
```

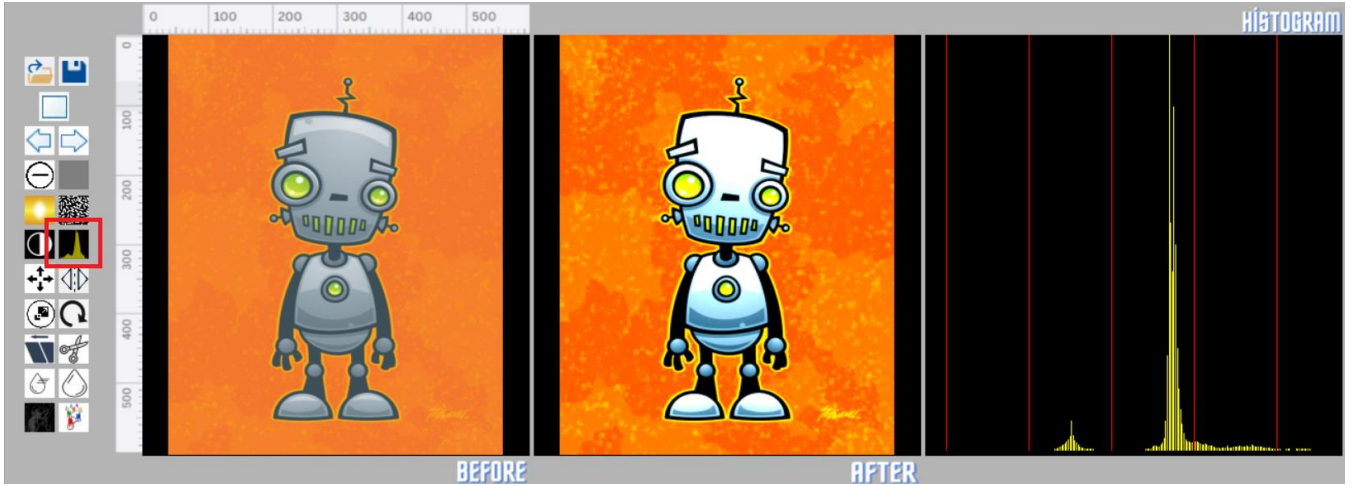
```

    }
    pictureBox2.Image = CikisResmi;
}

private void pbKontrast_Click(object sender, EventArgs e)
{
    Kontrast();
}

```

10.HİSTOGRAM



```

public void Histogram()
{
    ArrayList DiziPiksel = new ArrayList();

    int OrtalamaRenk = 0;
    Color OkunanRenk;
    int R = 0, G = 0, B = 0;
    Bitmap GirisResmi; //Histogram için giriş resmi gri-ton olmalıdır.
    GirisResmi = new Bitmap(pictureBox1.Image);

    int ResimGenisligi = GirisResmi.Width; //GirisResmi global tanımlandı.
    int ResimYuksekligi = GirisResmi.Height;

    int i = 0; //piksel sayısı tutulacak.
    for (int x = 0; x < GirisResmi.Width; x++)
    {
        for (int y = 0; y < GirisResmi.Height; y++)
        {
            OkunanRenk = GirisResmi.GetPixel(x, y);
            OrtalamaRenk = (int) (OkunanRenk.R + OkunanRenk.G +
OkunanRenk.B) / 3; //Griton resimde üç kanal rengi aynı değere sahiptir.
            DiziPiksel.Add(OrtalamaRenk); //Resimdeki tüm noktaları diziyeye
            atıyor.
        }
    }

    int[] DiziPikselSayilari = new int[256];
    for (int r = 0; r < 255; r++) //256 tane renk tonu için dönecek.
    {
        int PikselSayisi = 0;
        for (int s = 0; s < DiziPiksel.Count; s++) //resimdeki piksel
sayısınca dönecek.
        {
            if (r == Convert.ToInt16(DiziPiksel[s]))
                PikselSayisi++;
        }
        DiziPikselSayilari[r] = PikselSayisi;
    }

    //Değerleri listbox'a ekliyor.
    int RenkMaksPikselSayisi = 0; //Grafikte y eksenini ölçeklerken
    kullanılacak.
}

```

```

for (int k = 0; k <= 255; k++)
{
    //listBox1.Items.Add("Renk:" + k + "=" + DiziPikselSayilari[k]);
    if (DiziPikselSayilari[k] > RenkMaksPikselSayisi)
    {
        RenkMaksPikselSayisi = DiziPikselSayilari[k];
    }
}

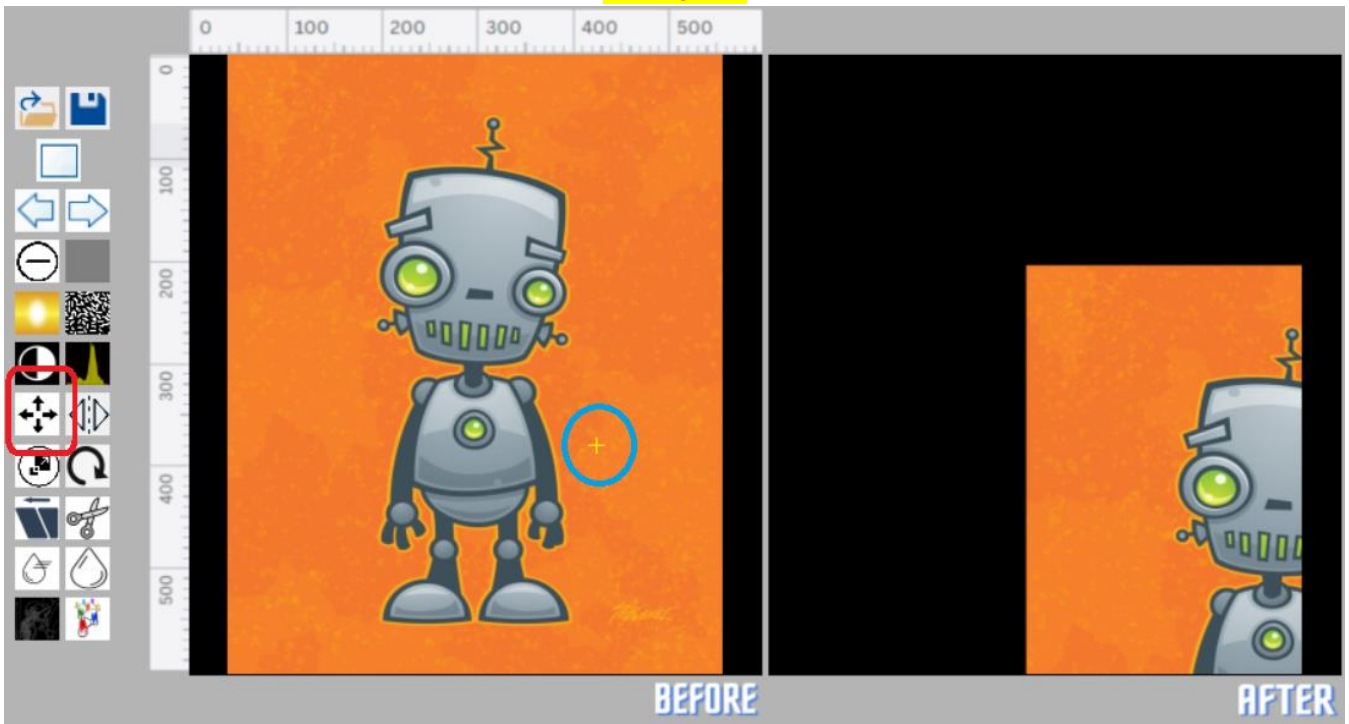
//Grafiği çiziyor.
Graphics CizimAlani;
Pen Kalem1 = new Pen(System.Drawing.Color.Yellow, 1);
Pen Kalem2 = new Pen(System.Drawing.Color.Red, 1);
CizimAlani = pictureBox12.CreateGraphics();

pictureBox12.Refresh();
int GrafikYuksekligi = 400;
double OlcekY = RenkMaksPikselSayisi / GrafikYuksekligi, OlcekX = 1.6;
for (int x = 0; x <= 255; x++)
{
    CizimAlani.DrawLine(Kalem1, (int)(20 + x * OlcekX),
GrafikYuksekligi, (int)(20 + x * OlcekX), (GrafikYuksekligi -
(int)(DiziPikselSayilari[x] / OlcekY))); if (x % 50 == 0)
CizimAlani.DrawLine(Kalem2, (int)(20 + x * OlcekX), GrafikYuksekligi, (int)(20 + x
* OlcekX), 0);
}
//textBox1.Text = "Maks.Piks=" + RenkMaksPikselSayisi.ToString();
}

private void pbHistogram_Click(object sender, EventArgs e)
{
    Histogram();
}

```

11.TAŞIMA



```

Graphics CizimAlani;
Pen Kalem1 = new Pen(System.Drawing.Color.Yellow, 1);
Pen Kalem2 = new Pen(System.Drawing.Color.Red, 2);

private void Form1_Load_1(object sender, EventArgs e)
{
    CizimAlani = pictureBox1.CreateGraphics();
}

```

```

int tiklanma_sayisi;
int a;
int x_1, y_1, x_2, y_2;
int x1, y1, x2, y2;
int basildi;

private void pictureBox1_MouseUp(object sender, MouseEventArgs e)
{
    tiklanma_sayisi++;

    //Kırpma
    if (tiklanma_sayisi % 2 == 1)
    {
        x_1 = e.X;
        y_1 = e.Y;
    }

    else if (tiklanma_sayisi % 2 == 0)
    {
        x_2 = e.X;
        y_2 = e.Y;
    }

    //Çizdirme
    CizimAlani.DrawLine(Kalem1, e.X - 5, e.Y, e.X + 5, e.Y);
    CizimAlani.DrawLine(Kalem1, e.X, e.Y - 5, e.X, e.Y + 5);

    if (tiklanma_sayisi == 1)
    {
        x1 = e.X;
        y1 = e.Y;
    }
    else if (tiklanma_sayisi >= 2)
    {
        x2 = e.X;
        y2 = e.Y;
        CizimAlani.DrawLine(Kalem2, x1, y1, x2, y2);
        x1 = x2;
        y1 = y2;
    }
}

public void Tasima()
{
    Color OkunanRenk;
    Bitmap GirisResmi, CikisResmi;
    GirisResmi = new Bitmap(pictureBox1.Image);
    int ResimGenisligi = GirisResmi.Width;
    int ResimYuksekligi = GirisResmi.Height;
    CikisResmi = new Bitmap(ResimGenisligi, ResimYuksekligi);

    double x2 = 0, y2 = 0;

    //Taşıma mesafelerini atıyor.
    int Tx = x_1;
    int Ty = y_1;

    for (int x1 = 0; x1 < (ResimGenisligi); x1++)
    {
        for (int y1 = 0; y1 < (ResimYuksekligi); y1++)
        {
            OkunanRenk = GirisResmi.GetPixel(x1, y1);

            x2 = x1 + Tx;
            y2 = y1 + Ty;

            if (x2 > 0 && x2 < ResimGenisligi && y2 > 0 && y2 <

```



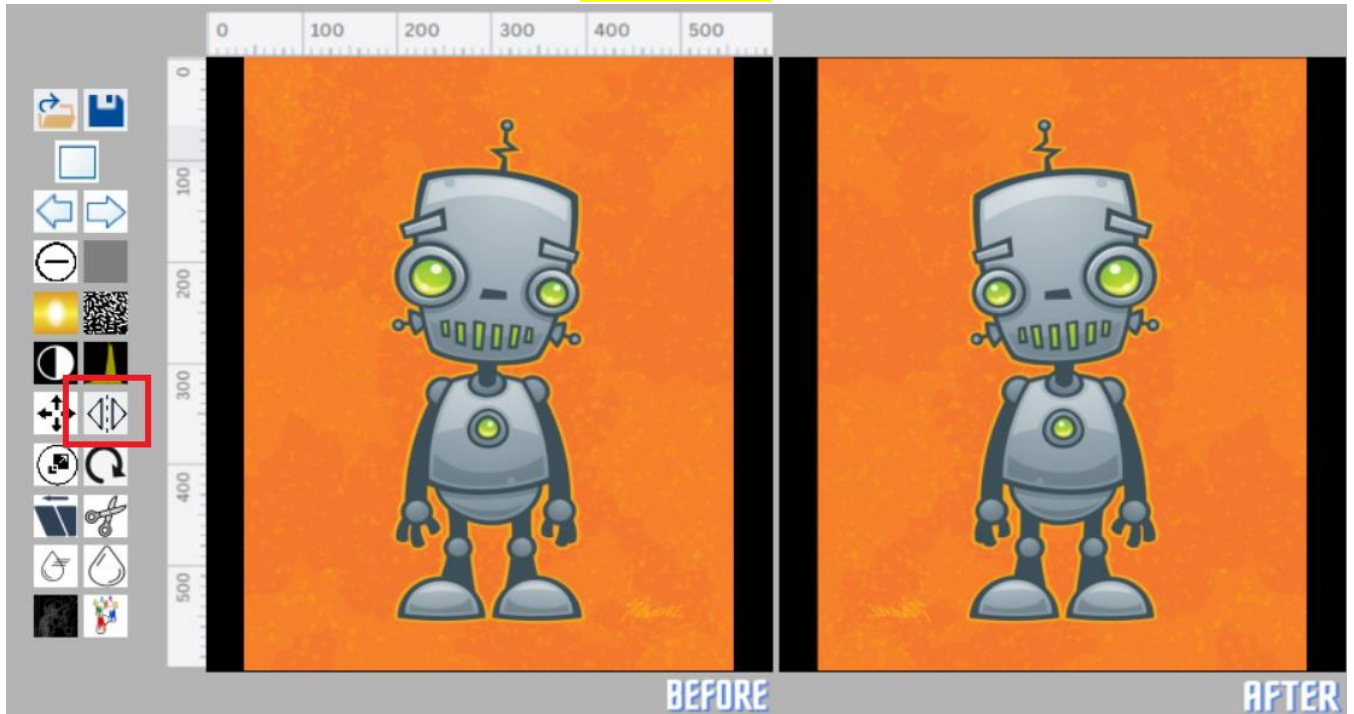
```

ResimYuksekligi)
        CikisResmi.SetPixel((int)x2, (int)y2, OkunanRenk);
    }
}
pictureBox2.Image = CikisResmi;
}

private void pbTasima_Click(object sender, EventArgs e)
{
    Tasima();
}

```

12.AYNALAMA



```

public void Aynalama()
{
    Color OkunanRenk;
    Bitmap GirisResmi, CikisResmi;

    GirisResmi = new Bitmap(pictureBox1.Image);
    int ResimGenisligi = GirisResmi.Width;
    int ResimYuksekligi = GirisResmi.Height;

    CikisResmi = new Bitmap(ResimGenisligi, ResimYuksekligi);

    double x2 = 0, y2 = 0;

    //Taşıma mesafelerini atıyor.
    int x0 = ResimGenisligi / 2;
    int y0 = ResimYuksekligi / 2;

    for (int x1 = 0; x1 < (ResimGenisligi); x1++)
    {
        for (int y1 = 0; y1 < (ResimYuksekligi); y1++)
        {
            OkunanRenk = GirisResmi.GetPixel(x1, y1);

            x2 = x1 + 2 * (x0 - x1);
            y2 = y1;

            if (x2 > 0 && x2 < ResimGenisligi && y2 > 0 && y2 <
ResimYuksekligi)
                CikisResmi.SetPixel((int)x2, (int)y2, OkunanRenk);
        }
    }
    pictureBox2.Image = CikisResmi;
}

```



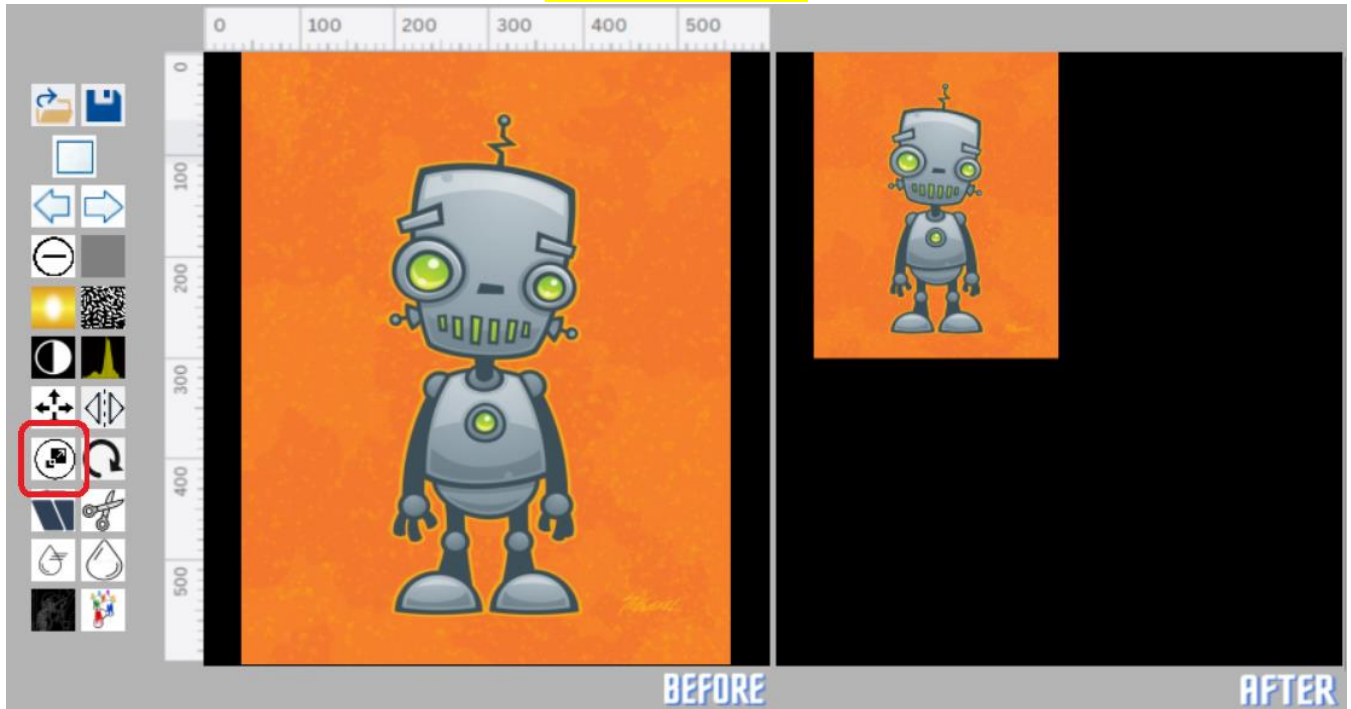
```

}

private void pbAynalama_Click(object sender, EventArgs e)
{
    Aynalama();
}

```

13.ÖLÇEKLENDİRME



```

public void Olceklendirme()
{
    Color OkunanRenk;
    Bitmap GirisResmi, CikisResmi;

    GirisResmi = new Bitmap(pictureBox1.Image);

    int ResimGenisligi = GirisResmi.Width;
    int ResimYuksekligi = GirisResmi.Height;

    CikisResmi = new Bitmap(ResimGenisligi, ResimYuksekligi);

    int x2 = 0, y2 = 0; //Çıkış resminin x ve y si olacak.
    int KucultmeKatsayisi = 2;
    //int KucultmeKatsayisi = Convert.ToInt32(textBox3.Text);

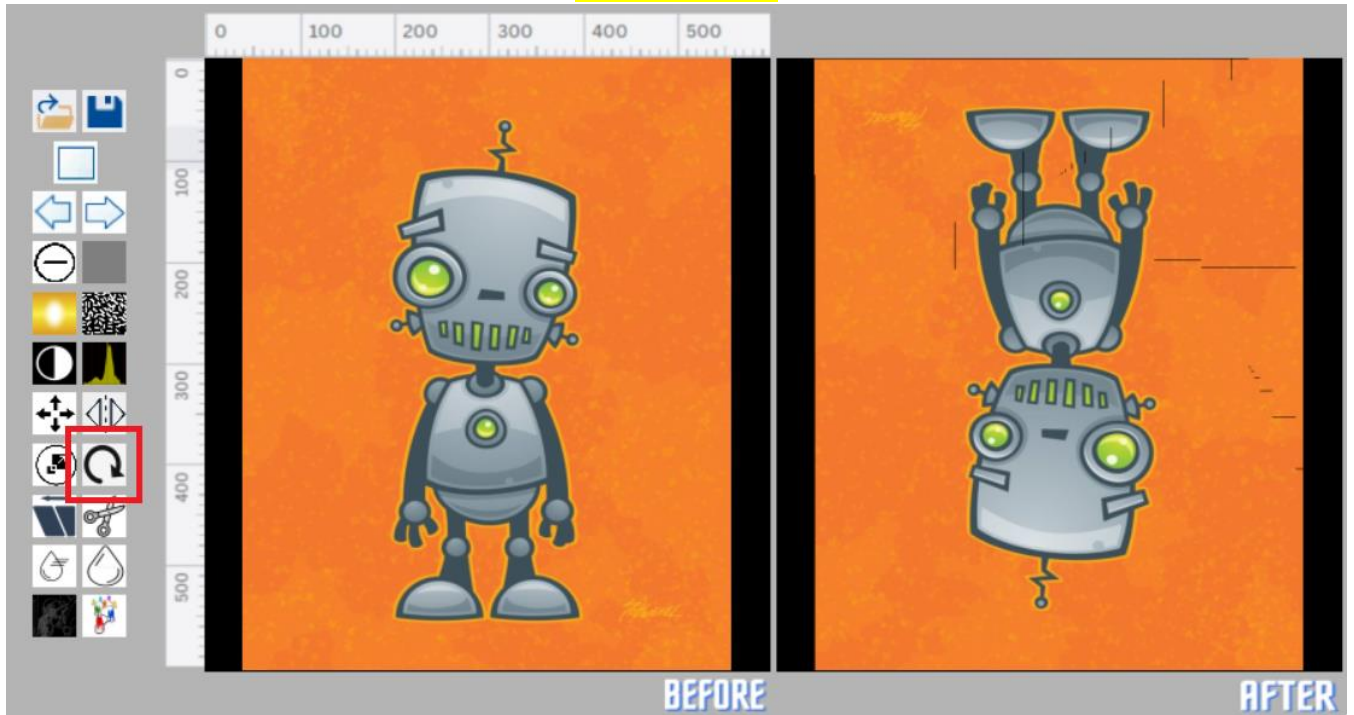
    for (int x1 = 0; x1 < ResimGenisligi; x1 = x1 + KucultmeKatsayisi)
    {
        y2 = 0;
        for (int y1 = 0; y1 < ResimYuksekligi; y1 = y1 + KucultmeKatsayisi)
        {
            OkunanRenk = GirisResmi.GetPixel(x1, y1);

            CikisResmi.SetPixel(x2, y2, OkunanRenk);
            y2++;
        }
        x2++;
    }
    pictureBox2.Image = CikisResmi;
}

private void pbOlceklendirme_Click(object sender, EventArgs e)
{
    Olceklendirme();
}

```

14.DONDURME



```

public void Dondurme()
{
    Color OkunanRenk;
    Bitmap GirisResmi, CikisResmi;
    GirisResmi = new Bitmap(pictureBox1.Image);

    int ResimGenisligi = GirisResmi.Width;
    int ResimYuksekligi = GirisResmi.Height;

    CikisResmi = new Bitmap(ResimGenisligi, ResimYuksekligi);

    int Aci = 180;
    double RadyanAci = Aci * 2 * Math.PI / 360;

    double x2 = 0, y2 = 0;

    int x0 = ResimGenisligi / 2;
    int y0 = ResimYuksekligi / 2;

    for (int x1 = 0; x1 < (ResimGenisligi); x1++)
    {
        for (int y1 = 0; y1 < (ResimYuksekligi); y1++)
        {
            OkunanRenk = GirisResmi.GetPixel(x1, y1);

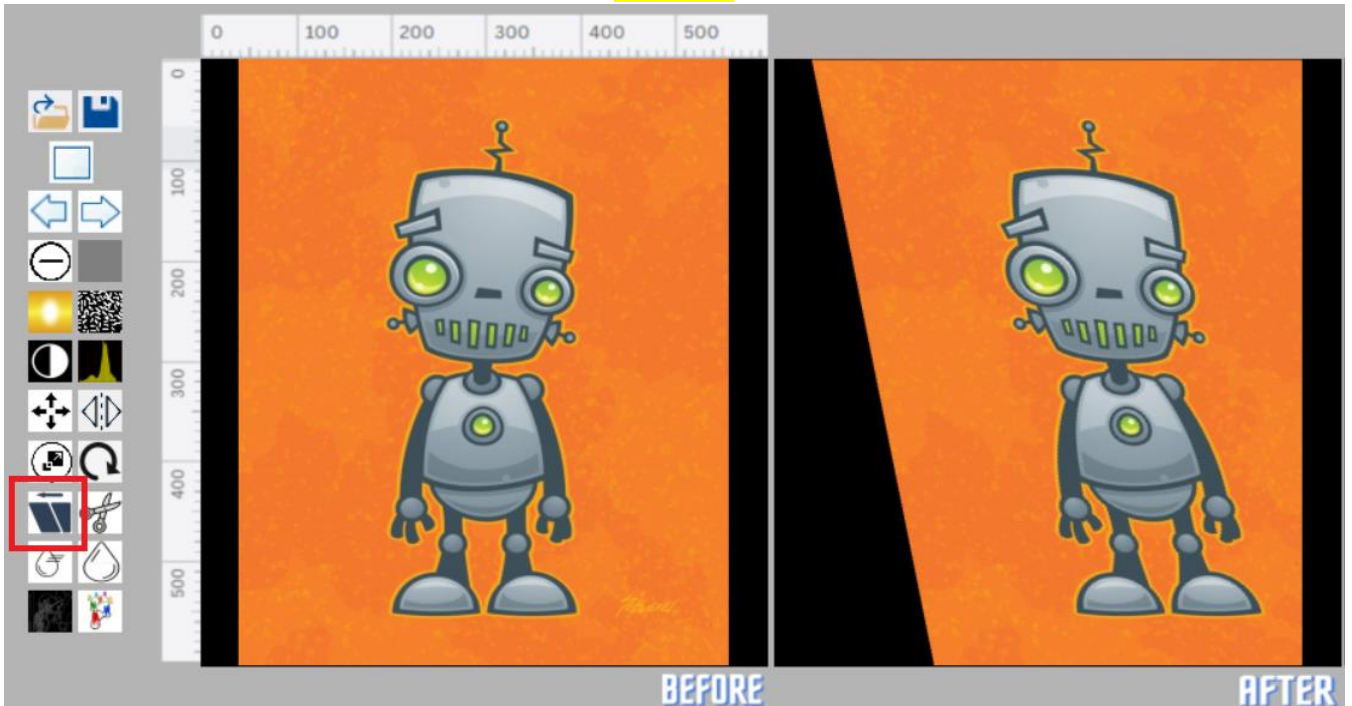
            x2 = Math.Cos(RadyanAci) * (x1 - x0) - Math.Sin(RadyanAci) *
(y1 - y0) + x0;
            y2 = Math.Sin(RadyanAci) * (x1 - x0) + Math.Cos(RadyanAci) *
(y1 - y0) + y0;

            if (x2 > 0 && x2 < ResimGenisligi && y2 > 0 && y2 <
ResimYuksekligi)
                CikisResmi.SetPixel((int)x2, (int)y2, OkunanRenk);
        }
    }
    pictureBox2.Image = CikisResmi;
}

private void pbDondurme_Click(object sender, EventArgs e)
{
    Dondurme();
}

```

15.EĞME



```

public void Egme ()
{
    Color OkunanRenk;
    Bitmap GirisResmi, CikisResmi;
    GirisResmi = new Bitmap (pictureBox1.Image);

    int ResimGenisligi = GirisResmi.Width;
    int ResimYuksekligi = GirisResmi.Height;

    CikisResmi = new Bitmap (ResimGenisligi, ResimYuksekligi);

    double EgmeKatsayisi = 0.2;
    double x2 = 0, y2 = 0;

    for (int x1 = 0; x1 < (ResimGenisligi); x1++)
    {
        for (int y1 = 0; y1 < (ResimYuksekligi); y1++)
        {
            OkunanRenk = GirisResmi.GetPixel (x1, y1);

            // +X eksenini yönünde
            x2 = x1 + EgmeKatsayisi * y1;
            y2 = y1;

            // -X eksenini yönünde
            //x2 = x1 - EgmeKatsayisi * y1;
            //y2 = y1;

            // +Y eksenini yönünde
            //x2 = x1;
            //y2 = EgmeKatsayisi * x1 + y1;

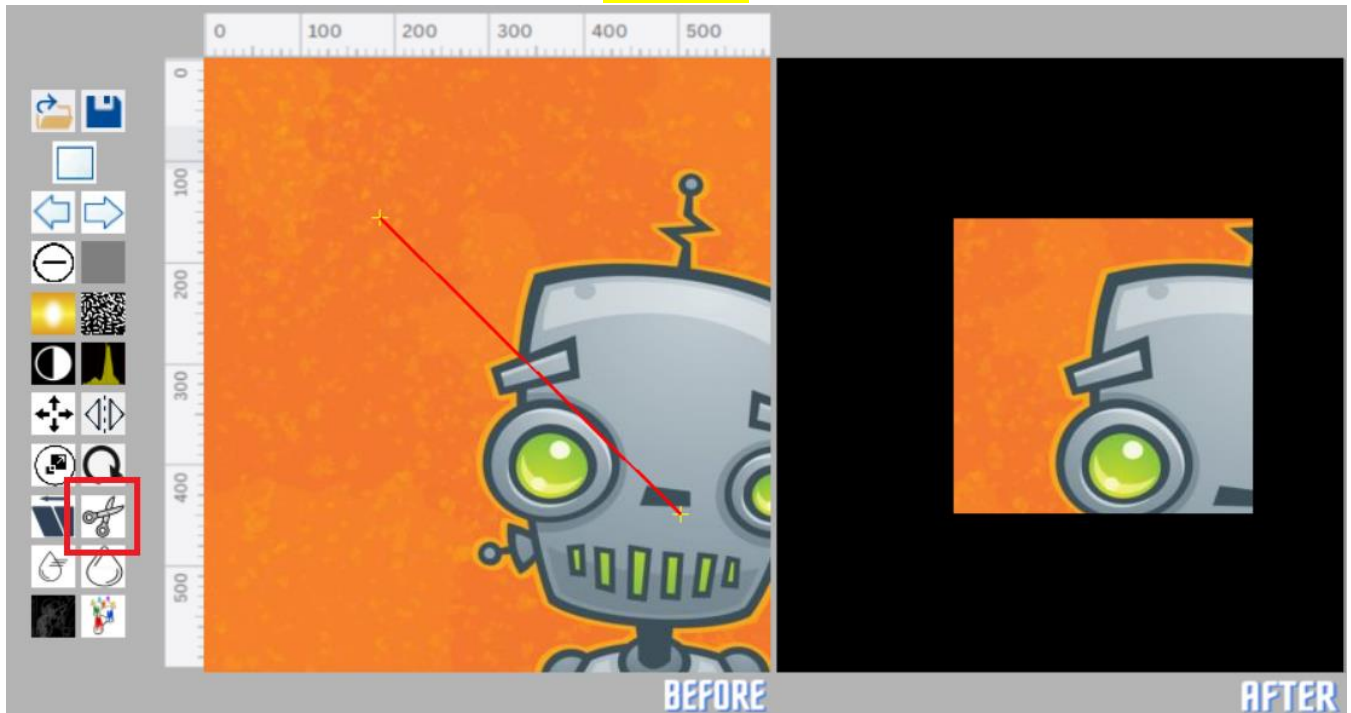
            // -Y eksenini yönünde
            //x2 = x1;
            //y2 = -EgmeKatsayisi * x1 + y1;

            if (x2 > 0 && x2 < ResimGenisligi && y2 > 0 && y2 <
ResimYuksekligi)
                CikisResmi.SetPixel ((int)x2, (int)y2, OkunanRenk);
        }
    }
    pictureBox2.Image = CikisResmi;
}

```

```
private void pbEgme_Click(object sender, EventArgs e)
{
    Egme();
}
```

16.KIRPMA



```
public void Kirpma()
{
    Color OkunanRenk;
    Bitmap GirisResmi, CikisResmi;
    GirisResmi = new Bitmap(pictureBox1.Image);

    int ResimGenisligi = GirisResmi.Width;
    int ResimYuksekligi = GirisResmi.Height;

    CikisResmi = new Bitmap(ResimGenisligi, ResimYuksekligi);

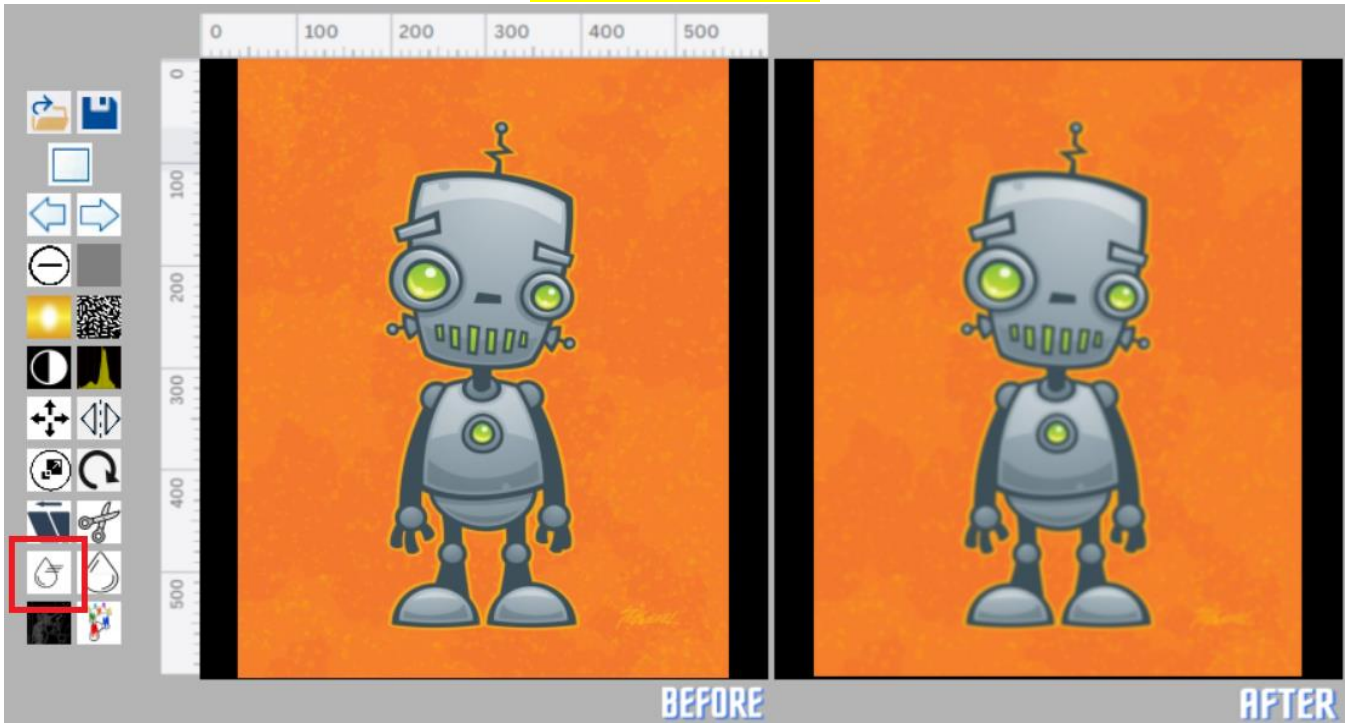
    int a = x_1;
    int b = y_1;
    int c = x_2;
    int d = y_2;

    for (int x1 = 0; x1 < (ResimGenisligi); x1++)
    {
        for (int y1 = 0; y1 < (ResimYuksekligi); y1++)
        {
            OkunanRenk = GirisResmi.GetPixel(x1, y1);

            if (x1 > a && x1 < c && y1 > b && y1 < d)
                CikisResmi.SetPixel((int)x1, (int)y1, OkunanRenk);
        }
    }
    pictureBox2.Image = CikisResmi;
}

private void pbKirpma_Click(object sender, EventArgs e)
{
    Kirpma();
}
```

17.BULANIKLAŞTIRMA



```
public Bitmap MeanFiltresi()  
{  
    Color OkunanRenk;  
    Bitmap GirisResmi, CikisResmi;  
    GirisResmi = new Bitmap(pictureBox1.Image);  
  
    int ResimGenisligi = GirisResmi.Width;  
    int ResimYuksekligi = GirisResmi.Height;  
  
    CikisResmi = new Bitmap(ResimGenisligi, ResimYuksekligi);  
  
    int SablonBoyutu = 5; //şablon boyutu 3 den büyük tek rakam olmalıdır  
    (3,5,7 gibi).  
    int x, y, i, j, toplamR, toplamG, toplamB, ortalamaR, ortalamaG,  
    ortalamaB;  
  
    for (x = (SablonBoyutu - 1) / 2; x < ResimGenisligi - (SablonBoyutu -  
1) / 2; x++)  
    {  
        for (y = (SablonBoyutu - 1) / 2; y < ResimYuksekligi -  
(SablonBoyutu - 1) / 2; y++)  
        {  
            toplamR = 0;  
            toplamG = 0;  
            toplamB = 0;  
  
            for (i = -((SablonBoyutu - 1) / 2); i <= (SablonBoyutu - 1) /  
2; i++)  
            {  
                for (j = -((SablonBoyutu - 1) / 2); j <= (SablonBoyutu - 1)  
/ 2; j++)  
                {  
                    OkunanRenk = GirisResmi.GetPixel(x + i, y + j);  
  
                    toplamR = toplamR + OkunanRenk.R;  
                    toplamG = toplamG + OkunanRenk.G;  
                    toplamB = toplamB + OkunanRenk.B;  
  
                }  
            }  
        }  
    }  
}
```



```

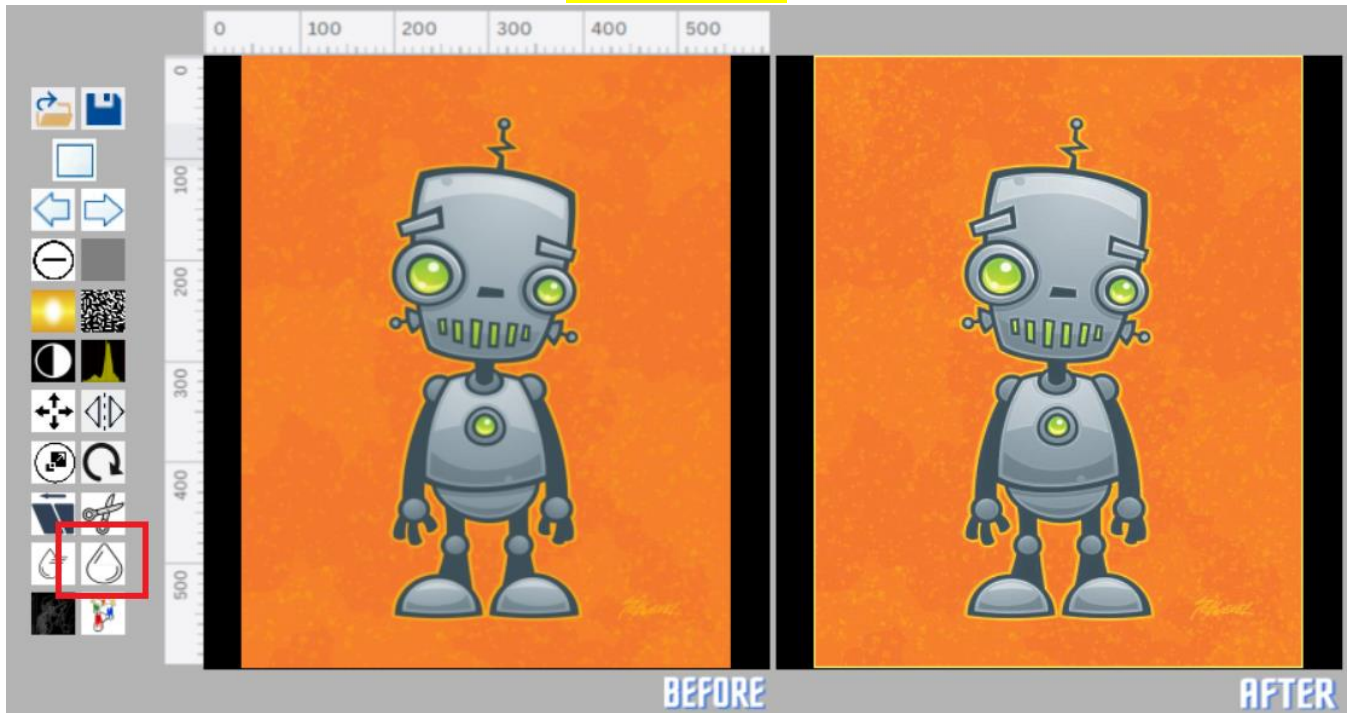
        ortalamaR = toplamR / (SablonBoyutu * SablonBoyutu);
        ortalamaG = toplamG / (SablonBoyutu * SablonBoyutu);
        ortalamaB = toplamB / (SablonBoyutu * SablonBoyutu);

        CikisResmi.SetPixel(x, y, Color.FromArgb(ortalamaR, ortalamaG,
ortalamaB));
    }
}
return CikisResmi;
}

private void pbBulaniklastirma_Click(object sender, EventArgs e)
{
    pictureBox2.Image = MeanFiltresi();
}

```

18.NETLEŞTİRME



```

public Bitmap OrjinalResimdenBulanikResmiCikarma(Bitmap OrjinalResim,
Bitmap BulanikResim)
{

```

```

    Color OkunanRenk1, OkunanRenk2, DonusenRenk;
    Bitmap CikisResmi;

```

```

    int ResimGenisligi = OrjinalResim.Width;
    int ResimYuksekligi = OrjinalResim.Height;

```

```

    CikisResmi = new Bitmap(ResimGenisligi, ResimYuksekligi);

```

```

    int R, G, B;
    double Olcekleme = 1.7; //0.2-0.7 arası uygundur.
    for (int x = 0; x < ResimGenisligi; x++)
    {

```

```

        for (int y = 0; y < ResimYuksekligi; y++)
        {

```

```

            OkunanRenk1 = OrjinalResim.GetPixel(x, y);
            OkunanRenk2 = BulanikResim.GetPixel(x, y);

```

```

            R = Convert.ToInt16(Olcekleme * (OkunanRenk1.R -
OkunanRenk2.R));

```

```

            G = Convert.ToInt16(Olcekleme * (OkunanRenk1.G -
OkunanRenk2.G));

```

```

            B = Convert.ToInt16(Olcekleme * (OkunanRenk1.B -

```



```
OkunanRenk2.B));
```

```
//Renkler sınırların dışına çıktıysa, sınır değeri alınacak.
```

```
if (R > 255) R = 255;
```

```
if (G > 255) G = 255;
```

```
if (B > 255) B = 255;
```

```
if (R < 0) R = 0;
```

```
if (G < 0) G = 0;
```

```
if (B < 0) B = 0;
```

```
DonusenRenk = Color.FromArgb(R, G, B);
```

```
CikisResmi.SetPixel(x, y, DonusenRenk);
```

```
}
```

```
}
```

```
return CikisResmi;
```

```
}
```

```
public Bitmap KenarGoruntusuIleOrjinalResmiBirlestir(Bitmap OrjinalResim,  
Bitmap KenarGoruntusu)
```

```
{
```

```
Color OkunanRenk1, OkunanRenk2, DonusenRenk;
```

```
Bitmap CikisResmi;
```

```
int ResimGenisligi = OrjinalResim.Width;
```

```
int ResimYuksekligi = OrjinalResim.Height;
```

```
CikisResmi = new Bitmap(ResimGenisligi, ResimYuksekligi);
```

```
int R, G, B;
```

```
for (int x = 0; x < ResimGenisligi; x++)
```

```
{
```

```
for (int y = 0; y < ResimYuksekligi; y++)
```

```
{
```

```
OkunanRenk1 = OrjinalResim.GetPixel(x, y);
```

```
OkunanRenk2 = KenarGoruntusu.GetPixel(x, y);
```

```
R = OkunanRenk1.R + OkunanRenk2.R;
```

```
G = OkunanRenk1.G + OkunanRenk2.G;
```

```
B = OkunanRenk1.B + OkunanRenk2.B;
```

```
//Renkler sınırların dışına çıktıysa, sınır değeri alınacak.
```

```
if (R > 255) R = 255;
```

```
if (G > 255) G = 255;
```

```
if (B > 255) B = 255;
```

```
if (R < 0) R = 0;
```

```
if (G < 0) G = 0;
```

```
if (B < 0) B = 0;
```

```
DonusenRenk = Color.FromArgb(R, G, B);
```

```
CikisResmi.SetPixel(x, y, DonusenRenk);
```

```
}
```

```
}
```

```
return CikisResmi;
```

```
}
```

```
private void pbNetlestirme_Click(object sender, EventArgs e)
```

```
{
```

```
Bitmap OrjinalResim = new Bitmap(pictureBox1.Image);
```

```
Bitmap BulanikResim;
```

```
BulanikResim = MeanFiltresi();
```

```
Bitmap KenarGoruntusu =
```

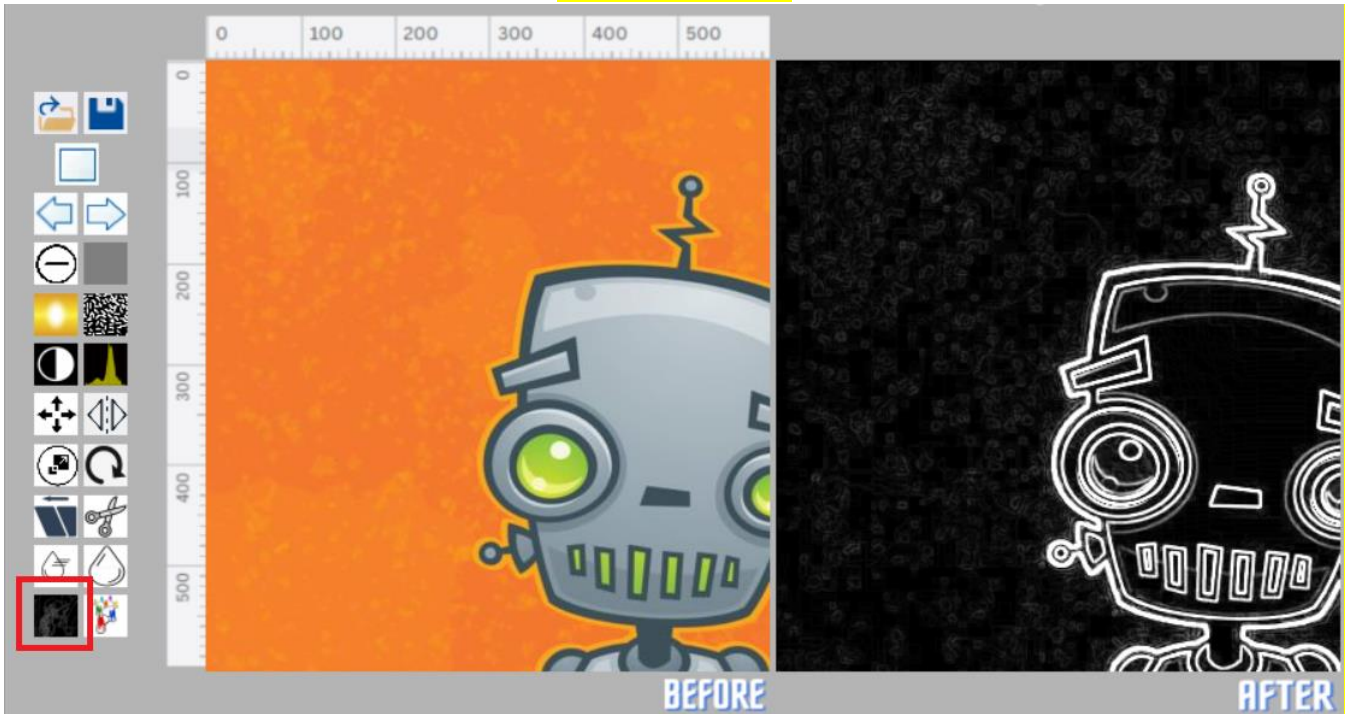
```
OrjinalResimdenBulanikResmiCikarma(OrjinalResim, BulanikResim);
```

```

        Bitmap NetlesmisResim =
KenarGoruntusuIleOrjinalResmiBirlestir(OrjinalResim, KenarGoruntusu);
        pictureBox2.Image = NetlesmisResim;
    }

```

19.KENAR BULMA



```

public void KenarBulma()
{
    Bitmap GirisResmi, CikisResmi;
    GirisResmi = new Bitmap(pictureBox1.Image);

    int ResimGenisligi = GirisResmi.Width;
    int ResimYuksekligi = GirisResmi.Height;

    CikisResmi = new Bitmap(ResimGenisligi, ResimYuksekligi);

    int SablonBoyutu = 3;
    int ElemanSayisi = SablonBoyutu * SablonBoyutu;

    int x, y; //, i, j, toplamR, toplamG, toplamB, ortalamaR, ortalamaG,
ortalamaB;

    for (x = (SablonBoyutu - 1) / 2; x < ResimGenisligi - (SablonBoyutu -
1) / 2; x++) //Resmi taramaya şablonun yarısı kadar dış kenarlardan içeride
başlayacak ve bitirecek.
    {
        for (y = (SablonBoyutu - 1) / 2; y < ResimYuksekligi -
(SablonBoyutu - 1) / 2; y++)
        {
            Color Renk;
            int P1, P2, P3, P4, P5, P6, P7, P8, P9;

            Renk = GirisResmi.GetPixel(x - 1, y - 1);
            P1 = Renk.R;

            Renk = GirisResmi.GetPixel(x, y - 1);
            P2 = Renk.R;

            Renk = GirisResmi.GetPixel(x + 1, y - 1);
            P3 = Renk.R;

            Renk = GirisResmi.GetPixel(x - 1, y);
            P4 = Renk.R;

```

```

Renk = GirisResmi.GetPixel(x, y);
P5 = Renk.R;

Renk = GirisResmi.GetPixel(x + 1, y);
P6 = Renk.R;

Renk = GirisResmi.GetPixel(x - 1, y + 1);
P7 = Renk.R;

Renk = GirisResmi.GetPixel(x, y + 1);
P8 = Renk.R;

Renk = GirisResmi.GetPixel(x + 1, y + 1);
P9 = Renk.R;

int RenkDegeri = Math.Abs((P1 + 2 * P2 + P3) - (P7 + 2 * P8 +
P9)) + Math.Abs((P3 + 2 * P6 + P9) - (P1 + 2 * P4 + P7));

//Renkler sınırların dışına çıktıysa, sınır değeri alınacak.
if (RenkDegeri > 255) RenkDegeri = 255;
CikisResmi.SetPixel(x, y, Color.FromArgb(RenkDegeri,
RenkDegeri, RenkDegeri));
}
}
pictureBox2.Image = CikisResmi;
}

private void pbKenarBulma_Click(object sender, EventArgs e)
{
KenarBulma();
}

```

20.ALT GRUPLARA AYIRMA



```

//Resmi önce gri tona dönüştürüyor
public Bitmap ResmiGriTonaDonustur(Bitmap GirisResmi)
{
int R = 0, G = 0, B = 0;
Color OkunanRenk, DonusenRenk;
//Bitmap GirisResmi = new Bitmap(pictureBox1.Image);

int ResimGenisligi = GirisResmi.Width; //GirisResmi global tanımlandı.
Fonksiyonla gelmedi.
int ResimYuksekligi = GirisResmi.Height;

```

```
//CikisResmi = new Bitmap(ResimGenisligi, ResimYuksekligi,  
System.Drawing.Imaging.PixelFormat.Format8bppIndexed); //8 bir formatında gri-ton  
resim oluşturmak için.  
Bitmap CikisResmi = new Bitmap(ResimGenisligi, ResimYuksekligi);  
//Cikis resmini oluşturuyor. Boyutları giriş resmi ile aynı olur.  
  
int i = 0, j = 0; //Çıkış resminin x ve y si olacak.  
for (int x = 0; x < ResimGenisligi; x++)  
{  
    j = 0;  
    for (int y = 0; y < ResimYuksekligi; y++)  
    {  
        OkunanRenk = GirişResmi.GetPixel(x, y);  
  
        //int GriDegeri = (int)(OkunanRenk.R + OkunanRenk.G +  
OkunanRenk.B)/3; //Ortalama Gri-ton formülü  
        //int GriDegeri = Convert.ToInt16(OkunanRenk.R * 0.299 +  
OkunanRenk.G * 0.587 + OkunanRenk.B * 0.114); //Gri-ton formülü  
        int GriDegeri = Convert.ToInt16(OkunanRenk.R * 0.21 +  
OkunanRenk.G * 0.71 + OkunanRenk.B * 0.071); //Gri-ton formülü  
  
        R = GriDegeri;  
        G = GriDegeri;  
        B = GriDegeri;  
        DonusenRenk = Color.FromArgb(R, G, B);  
  
        CikisResmi.SetPixel(i, j, DonusenRenk);  
        j++;  
    }  
    i++;  
}  
return CikisResmi;  
}  
  
//Resmi 128 ile eşikleme siyah beyaz yapıyor  
public Bitmap ResmiEsiklemeYap(Bitmap GirişResmi)  
{  
    int R = 0, G = 0, B = 0;  
    Color OkunanRenk;  
    //Bitmap GirişResmi = new Bitmap(pictureBox1.Image);  
  
    int ResimGenisligi = GirişResmi.Width; //GirişResmi global tanımlandı.  
    int ResimYuksekligi = GirişResmi.Height;  
    Bitmap CikisResmi = new Bitmap(ResimGenisligi, ResimYuksekligi);  
    //Cikis resmini oluşturuyor. Boyutları giriş resmi ile aynı olur.  
  
    //Eşik değerini trackBar'dan alacaktır.  
    int EsikDegeri = 15;  
  
    int i = 0, j = 0; //Çıkış resminin x ve y si olacak.  
    for (int x = 0; x < ResimGenisligi; x++)  
    {  
        j = 0;  
        for (int y = 0; y < ResimYuksekligi; y++)  
        {  
            OkunanRenk = GirişResmi.GetPixel(x, y);  
            if (OkunanRenk.R >= EsikDegeri)  
                R = 255;  
            else  
                R = 0;  
  
            if (OkunanRenk.G >= EsikDegeri)  
                G = 255;  
            else  
                G = 0;
```

```

        if (OkunanRenk.B >= EsikDegeri)
            B = 255;
        else
            B = 0;

        Color DonusenRenk = Color.FromArgb(R, G, B);
        CikisResmi.SetPixel(i, j, DonusenRenk);
        j++;
    }
    i++;
}

return CikisResmi;
}

private void pbAltGrupAyirma_Click(object sender, EventArgs e)
{
    Bitmap GirisResmi, CikisResmi;

    int KomsularinEnKucukEtiketDegeri = 0;

    GirisResmi = new Bitmap(pictureBox1.Image);
    int ResimGenisligi = GirisResmi.Width;
    int ResimYuksekligi = GirisResmi.Height;
    int PikselSayisi = ResimGenisligi * ResimYuksekligi;

    GirisResmi = ResmiGriTonaDonustur(GirisResmi); //Resmi önce gri tona
dönüştürüyor.
    GirisResmi = ResmiEsiklemeYap(GirisResmi); //Resmi 128 ile eşikleme
siyah beyaz yapıyor.
    //pictureBox2.Image = GirisResmi; //Resmin son halini gösteriyor.

    CikisResmi = new Bitmap(ResimGenisligi, ResimYuksekligi);

    int x, y, i, j, EtiketNo = 0;

    int[,] EtiketNumarasi = new int[ResimGenisligi, ResimYuksekligi];
//Resmin her pikselinin etiket numarası tutulacak.

    //Tüm piksellerin Etiket numarasını başlangıçta 0 olarak atayacak.
Siyah ve beyaz farketmez. Zaten ileride beyaz olanlara numara verilecek.
    for (x = 0; x < ResimGenisligi; x++)
    {
        for (y = 0; y < ResimYuksekligi; y++)
        {
            EtiketNumarasi[x, y] = 0;
        }
    }

    int IlkDeger = 0, SonDeger = 0;
    bool DegisimVar = false; //Etiket numaralarında değişim olmayana kadar
dönmesi için sonsuz döngüyü kontrol edecek.
    do //etiket numaralarında değişim kalmayana kadar dönecek.
    {
        DegisimVar = false;
        //----- Resmi tarıyor -----
        ---
        for (y = 1; y < ResimYuksekligi - 1; y++) //Resmin 1 piksel
içerisinden başlayıp, bitirecek. Çünkü çekirdek şablon en dış kenardan başlamalı.
        {
            for (x = 1; x < ResimGenisligi - 1; x++)
            {
                //Resim siyah beyaz olduğu için tek kanala bakmak yeterli
olacak. Sıradaki piksel beyaz ise işlem yap. Beyaz olduğu 255 yerine 128 kullanarak
yapıldı.
                if (GirisResmi.GetPixel(x, y).R > 128)
                {

```

```

        //işlem öncesi ele alınan pikselin etiket değerini
okuyacak. İşlemler bittikten sonra bu değer değişirse, sonsuz döngü için işlem
yapılmış demektir.

        İlkDeger = EtiketNumarasi[x, y];

        //Komşular arasında en küçük etiket numarasını bulacak.
        KomsularinEnKucukEtiketDegeri = 0;
        for (j = -1; j <= 1; j++) //Çekirdek şablon 3x3 lük bir
matris. Dolayısı ile x,y nin -1 den başlayıp +1 ne kadar yer kaplar.
        {
            for (i = -1; i <= 1; i++)
            {
                if (EtiketNumarasi[x + i, y + j] != 0 &&
KomsularinEnKucukEtiketDegeri == 0) //hücresinin etiketi varsa ve daha hiç en küçük
atanmadı ise ilk okuduğu bu değeri en küçük olarak atayacak.
                {
                    KomsularinEnKucukEtiketDegeri =
EtiketNumarasi[x + i, y + j];
                }
                else if (EtiketNumarasi[x + i, y + j] <
KomsularinEnKucukEtiketDegeri && EtiketNumarasi[x + i, y + j] != 0 &&
KomsularinEnKucukEtiketDegeri != 0) //En küçük değer ve okunan hücreye etiket
atanmışsa, içindeki değer en küçük değerden küçük ise o zaman en küçük o hücrenin
değeri olmalıdır.
                {
                    KomsularinEnKucukEtiketDegeri =
EtiketNumarasi[x + i, y + j];
                }
            }
        }

        if (KomsularinEnKucukEtiketDegeri != 0) //Beyaz komşu
buldu ve içlerinde en küçük etiket değerine sahip numara da var. O zaman orta
piksele o numarayı ata.
        {
            EtiketNumarasi[x, y] =
KomsularinEnKucukEtiketDegeri;
        }
        else if (KomsularinEnKucukEtiketDegeri == 0)
//Komşuların hiç birinde etiket numarası yoksa o zaman yeni bir numara ata
        {
            EtiketNo = EtiketNo + 1;
            EtiketNumarasi[x, y] = EtiketNo;
        }

        SonDeger = EtiketNumarasi[x, y]; //İşlem öncesi ve
işlem sonrası değerler aynı ise ve bütün piksellerde hep aynı olursa artık değişim
yok demektir.

        if (IlkDeger != SonDeger)
            DegisimVar = true;
    }
}
} while (DegisimVar == true);

// Etiket değerine bağlı resmi renklendirecek-----
// Önce etiket numaralarını diziyeye çekecek.
int[] DiziEtiket = new int[PikselSayisi];
i = 0;
for (x = 1; x < ResimGenisligi - 1; x++)
{
    for (y = 1; y < ResimYuksekligi - 1; y++)
    {
        i++;
    }
}

```



```

        DiziEtiket[i] = EtiketNumarasi[x, y];
    }
}

//Dizideki etiket numaralarını sıralıyor. Hazır fonksiyon kullanıyor.
Array.Sort(DiziEtiket);

//Tekrar eden etiket numaraarını çıkarıyor. Hazır fonksiyon kullanıyor.
int[] TekrarsizEtiketNumaralari = DiziEtiket.Distinct().ToArray();

int[] RenkDizisi = new int[TekrarsizEtiketNumaralari.Length];

for (j = 0; j < TekrarsizEtiketNumaralari.Length; j++)
{
    RenkDizisi[j] = TekrarsizEtiketNumaralari[j]; //sıradaki ilk renge,
ait olacağı etiketin kaç numara olacağını atıyor.
}

int RenkSayisi = RenkDizisi.Length;

Color[] Renkler = new Color[RenkSayisi];
Random Rastgele = new Random();
int Kirmizi, Yesil, Mavi;

for (int r = 0; r < RenkSayisi; r++) //sonraki renkler.
{
    Kirmizi = Rastgele.Next(5, 25) * 10; //Açık renkler elde etmek ve
10 katları şeklinde olmasını sağlıyor. yani 150-250 arasındaki sayıları atıyor.
    Yesil = Rastgele.Next(5, 25) * 10;
    Mavi = Rastgele.Next(5, 25) * 10;

    Renkler[r] = Color.FromArgb(Kirmizi, Yesil, Mavi);
}

//Color[] Renkler= { Color.Black, Color.Blue, Color.Red, Color.Orange,
Color.LightPink, Color.LightYellow, Color.LimeGreen, Color.MediumPurple,
Color.Olive, Color.Magenta, Color.Maroon, Color.AliceBlue, Color.AntiqueWhite,
Color.Aqua, Color.LightBlue, Color.Azure, Color.White };

for (x = 1; x < ResimGenisligi - 1; x++) //Resmin 1 piksel içerisinden
başlayıp, bitirecek. Çünkü çekirdek şablon en dış kenardan başlamalı.
{
    for (y = 1; y < ResimYuksekligi - 1; y++)
    {
        int RenkSiraNo = Array.IndexOf(RenkDizisi, EtiketNumarasi[x,
y]); //Dikkat: önemli bir komut. Dizinin değerinden sıra numarasını alıyor. int[]
array = { 2, 3, 5, 7, 11, 13 }; int index = Array.IndexOf(array, 11); // returns 4

        if (GirisResmi.GetPixel(x, y).R < 128) //Eğer bu pikselin rengi
siyah ise aynı pikselin CikisResmi resimde siyah yapılacaktır.
        {
            CikisResmi.SetPixel(x, y, Color.Black);
        }
        else
        {
            CikisResmi.SetPixel(x, y, Renkler[RenkSiraNo]);
        }
    }
}
pictureBox2.Image = CikisResmi;
}
}
}

```