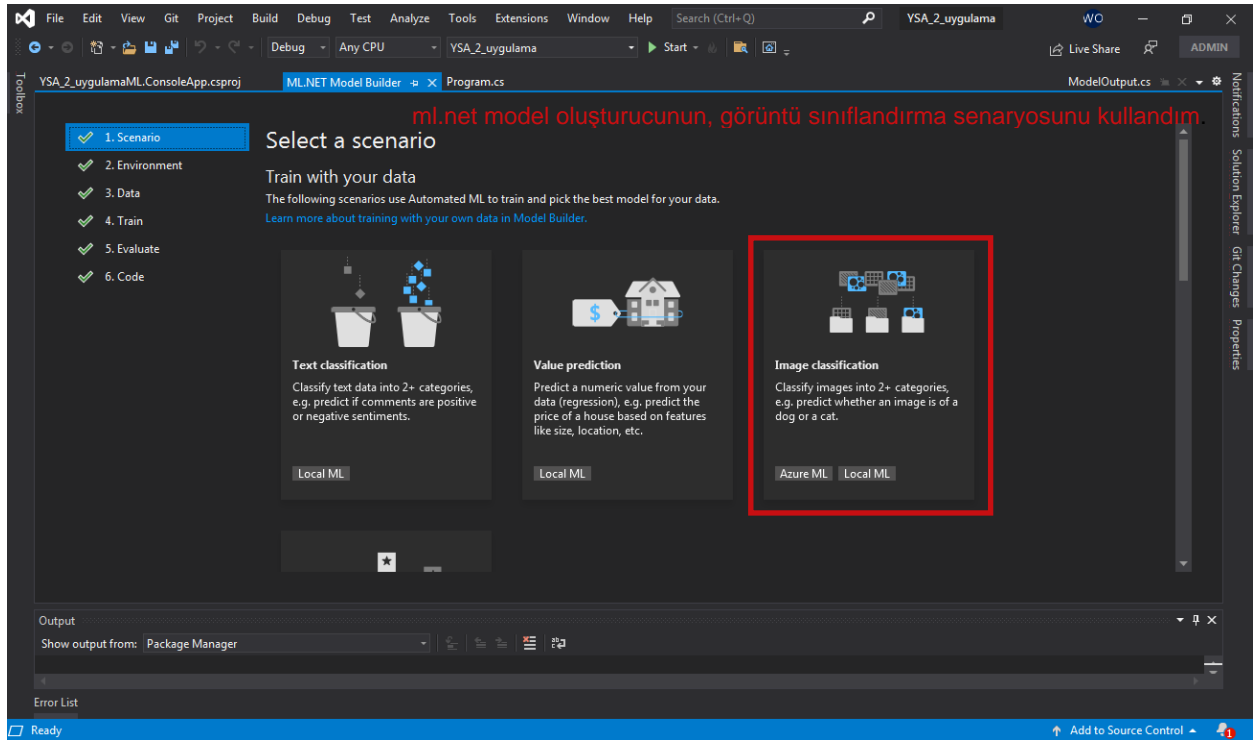


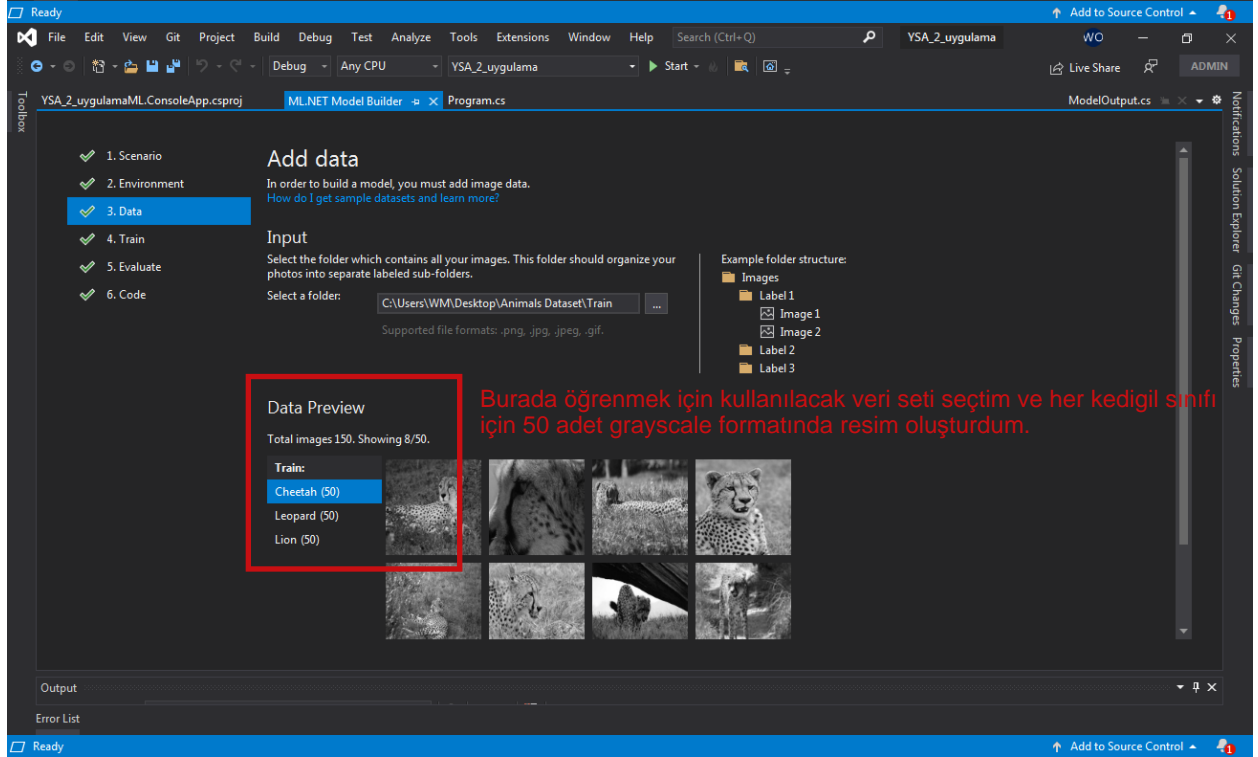
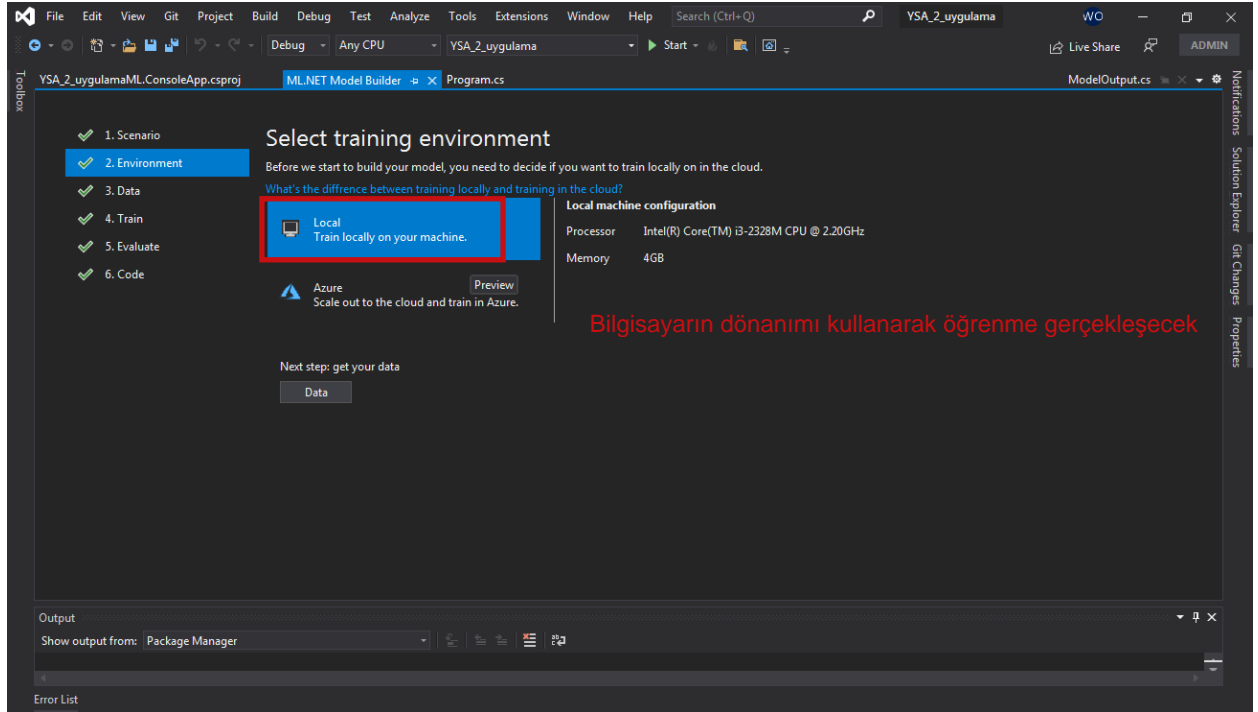
Hüsamettin DÖNÜŞ

2017710225027

1.Öğretim

A)Görüntüden Bilgi Çıkarma Projesi:





YSA_2_uygulama

File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help Search (Ctrl+Q)

YSA_2_uygulama ML.NET Model Builder -i-x Program.cs

1. Scenario 2. Environment 3. Data 4. Train 5. Evaluate 6. Code

Train

Model Builder automatically sets the training time for image classification based on the size of your dataset.
[How long should I train for?](#)

Training setup summary

Train again Training complete

30 dakika öğrenme sürecinden geçtikten sonra elde ettiğimiz en iyi doğruluk oranı 90.91%.

Training results

Best accuracy: 90.91%
Best model: DNN + ResNet50
Training time: 1807.94 seconds
Models explored (total): 1

Next step: evaluate your model

Evaluate

Output

Error List

Ready

YSA_2_uygulama

File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help Search (Ctrl+Q)

YSA_2_uygulama ML.NET Model Builder -i-x Program.cs

1. Scenario 2. Environment 3. Data 4. Train 5. Evaluate 6. Code


Evaluate

Results of training for your model can be found below.
[How do I understand my model performance?](#)

Best model:
Accuracy: 90.91%
Model: DNN + ResNet50

Try your model

sonuç olarak bazı resimlerde 100% tahmin oranı elde edildi.



Try another image

Results

Cheetah	100%
Lion	< 1%
Leopard	< 1%

Output

Error List

YSA_2_uygulama ML.ConsoleApp.csproj ML.NET Model Builder Program.cs ModelOutput.cs


1. Scenario
2. Environment
3. Data
4. Train
5. Evaluate
6. Code

Evaluate

Results of training for your model can be found below.
[How do I understand my model performance?](#)

Best model:
Accuracy: 90.91%
Model: DNN + ResNet50

Try your model



Try another image

Results	
Cheetah	99%
Leopard	< 1%
Lion	< 1%

Output

Error List

Ready Add to Source Control

YSA_2_uygulama ML.ConsoleApp.csproj ML.NET Model Builder Program.cs ModelOutput.cs


1. Scenario
2. Environment
3. Data
4. Train
5. Evaluate
6. Code

Evaluate

Results of training for your model can be found below.
[How do I understand my model performance?](#)

Best model:
Accuracy: 90.91%
Model: DNN + ResNet50

Try your model



Try another image

Results	
Cheetah	97%
Leopard	3%
Lion	< 1%

Output

Error List

Ready Add to Source Control

YSA_2_uygulama ML ConsoleApp.csproj ML.NET Model Builder Program.cs ModelOutput.cs


1. Scenario
2. Environment
3. Data
4. Train
5. Evaluate
6. Code

Evaluate

Results of training for your model can be found below.
[How do I understand my model performance?](#)

Best model:
Accuracy: 90.91%
Model: DNN + ResNet50

Try your model



Try another image

Results	
Leopard	96%
Cheetah	4%
Lion	< 1%

Output

Error List

Ready Add to Source Control

YSA_2_uygulama ML ConsoleApp.csproj ML.NET Model Builder Program.cs ModelOutput.cs


1. Scenario
2. Environment
3. Data
4. Train
5. Evaluate
6. Code

Evaluate

Results of training for your model can be found below.
[How do I understand my model performance?](#)

Best model:
Accuracy: 90.91%
Model: DNN + ResNet50

Try your model



Try another image

Results	
Leopard	86%
Cheetah	14%
Lion	< 1%

Output

Error List

Ready Add to Source Control

YSA_2_uygulama

File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help Search (Ctrl+Q)

Debug Any CPU YSA_2_uygulama Start

YSA_2_uygulamaML.ConsoleApp.csproj ML.NET Model Builder Program.cs ModelOutput.cs


1. Scenario
2. Environment
3. Data
4. Train
5. Evaluate
6. Code

Evaluate

Results of training for your model can be found below.
[How do I understand my model performance?](#)

Best model:
Accuracy: 90.91%
Model: DNN + ResNet50

Try your model



Try another image

Results	
Leopard	88%
Cheetah	12%
Lion	< 1%

Output

Error List

Ready Add to Source Control

File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help Search (Ctrl+Q)

Debug Any CPU YSA_2_uygulama Start

YSA_2_uygulamaML.ConsoleApp.csproj ML.NET Model Builder Program.cs ModelOutput.cs


1. Scenario
2. Environment
3. Data
4. Train
5. Evaluate
6. Code

Evaluate

Results of training for your model can be found below.
[How do I understand my model performance?](#)

Best model:
Accuracy: 90.91%
Model: DNN + ResNet50

Try your model



Try another image

Results	
Lion	98%
Leopard	1%
Cheetah	1%

Output

Error List

Ready Add to Source Control

YSA_2_uygulamaML.ConsoleApp.csproj ML.NET Model Builder Program.cs ModelOutput.cs


1. Scenario
2. Environment
3. Data
4. Train
5. Evaluate
6. Code

Evaluate

Results of training for your model can be found below.
[How do I understand my model performance?](#)

Best model:
Accuracy: 90.91%
Model: DNN + ResNet50

Try your model



[Try another image](#)

Results	
Lion	96%
Cheetah	2%
Leopard	1%

Output

Error List

Ready YSA_2_uygulamaML.ConsoleApp.csproj ML.NET Model Builder Program.cs ModelOutput.cs


1. Scenario
2. Environment
3. Data
4. Train
5. Evaluate
6. Code

Evaluate

Results of training for your model can be found below.
[How do I understand my model performance?](#)

Best model:
Accuracy: 90.91%
Model: DNN + ResNet50

Try your model



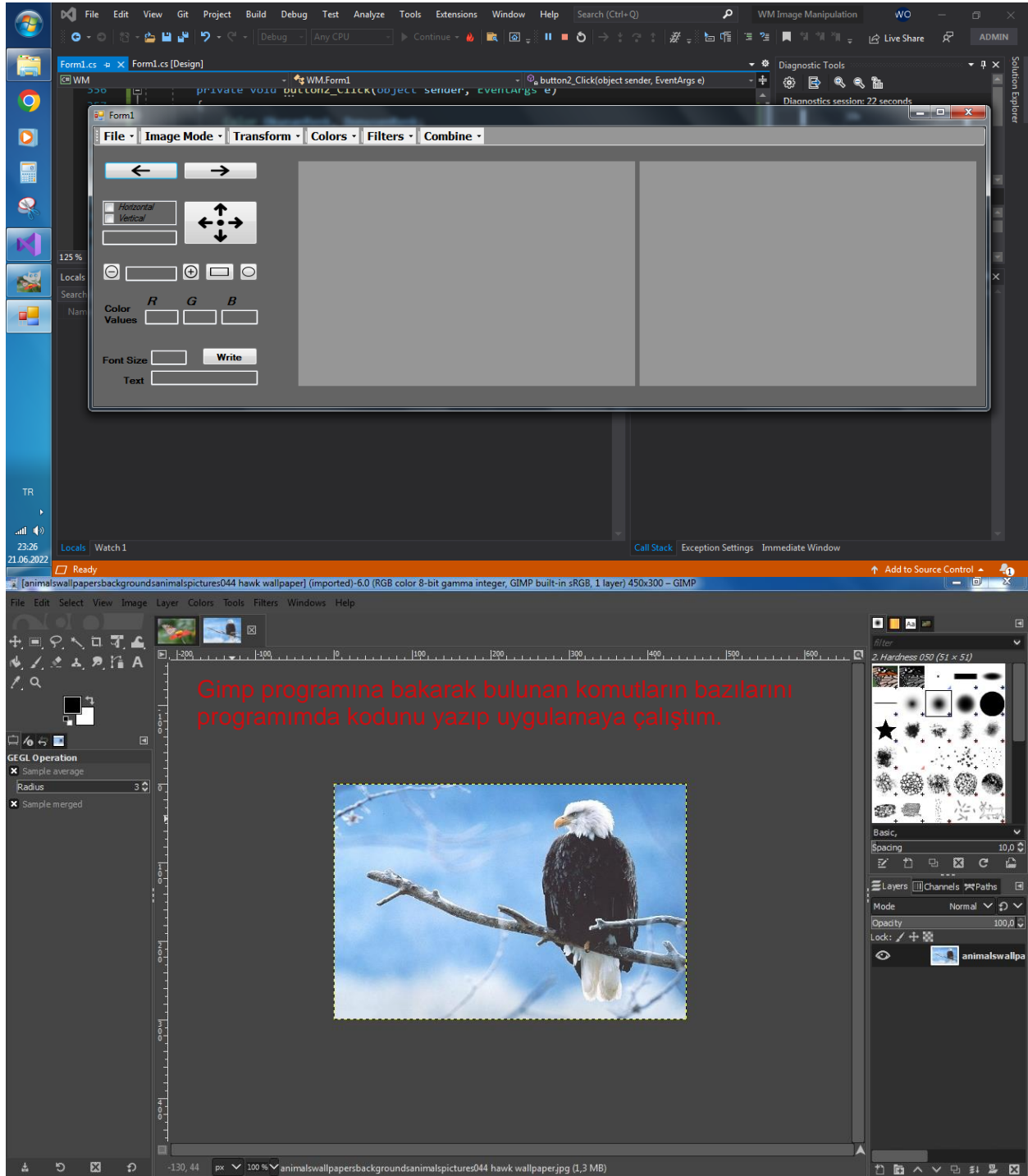
[Try another image](#)

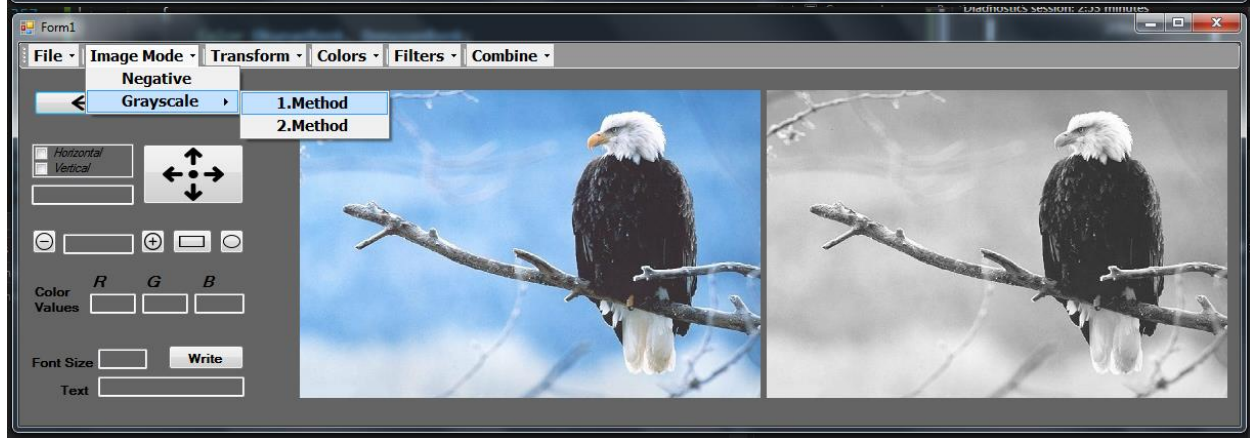
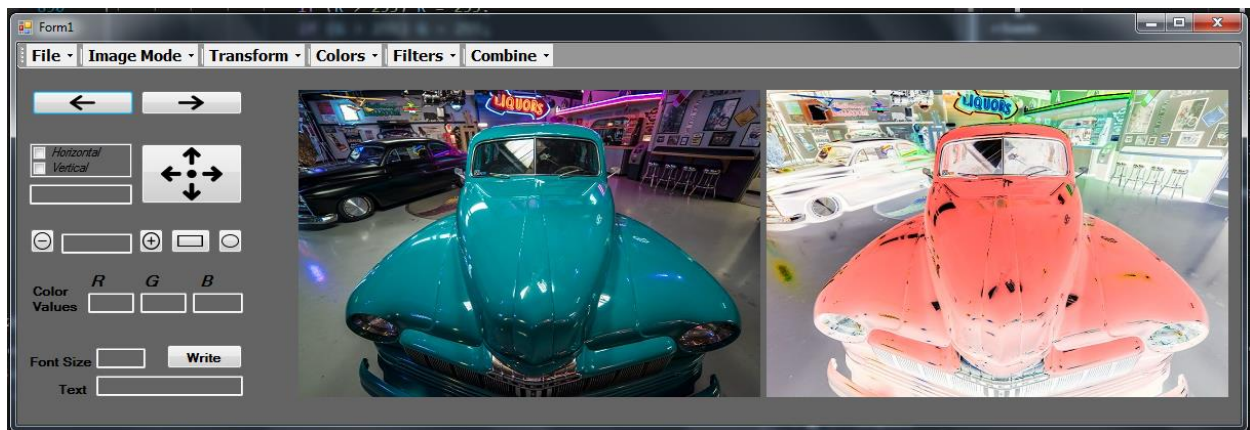
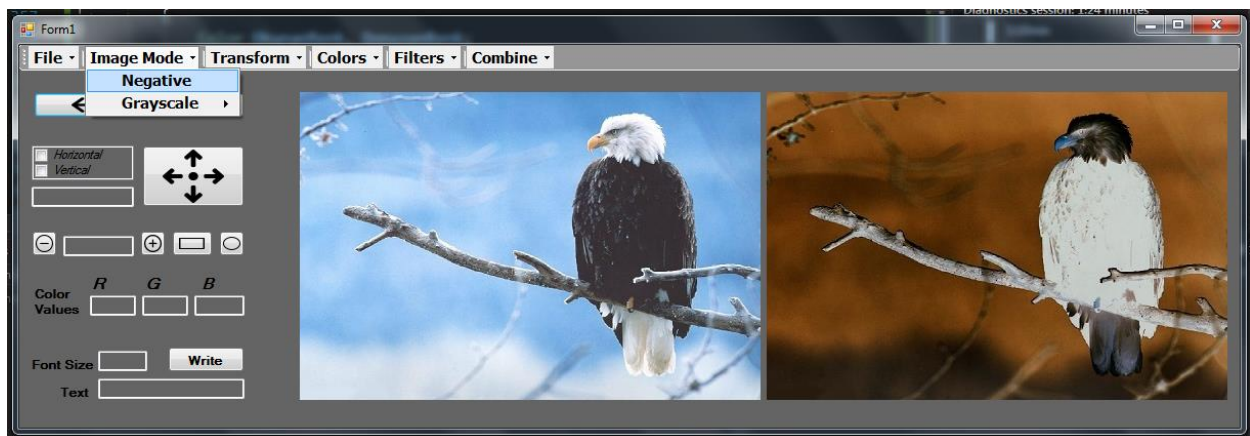
Results	
Lion	97%
Cheetah	3%
Leopard	< 1%

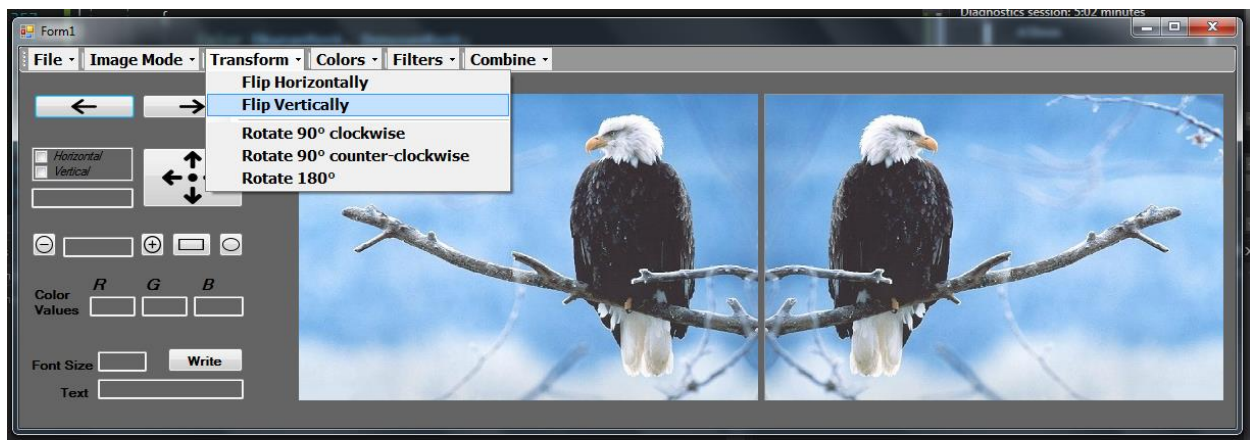
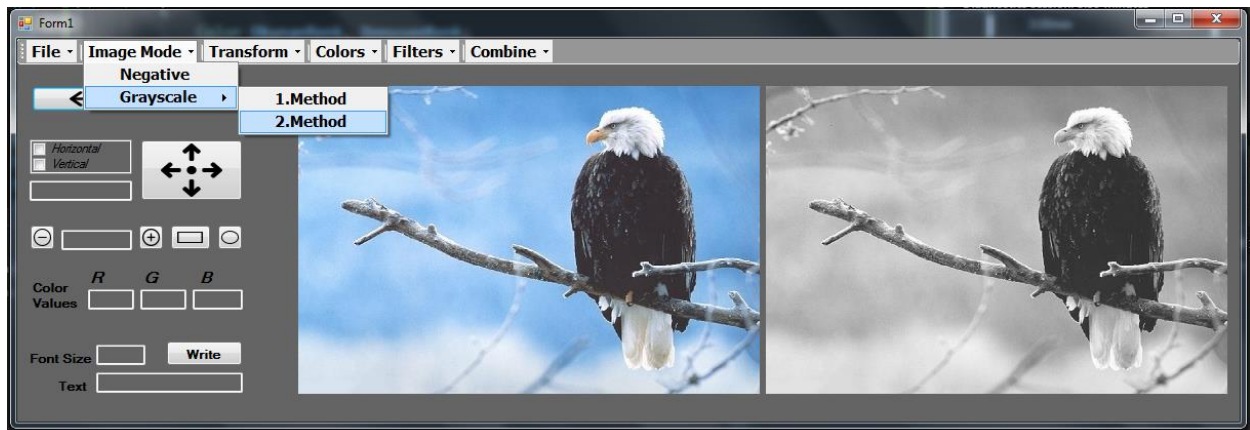
Output

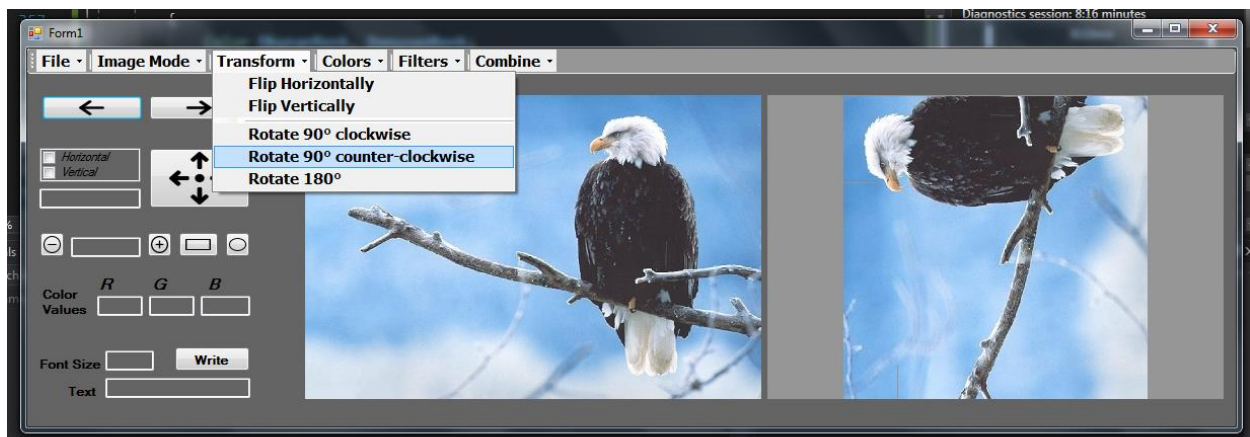
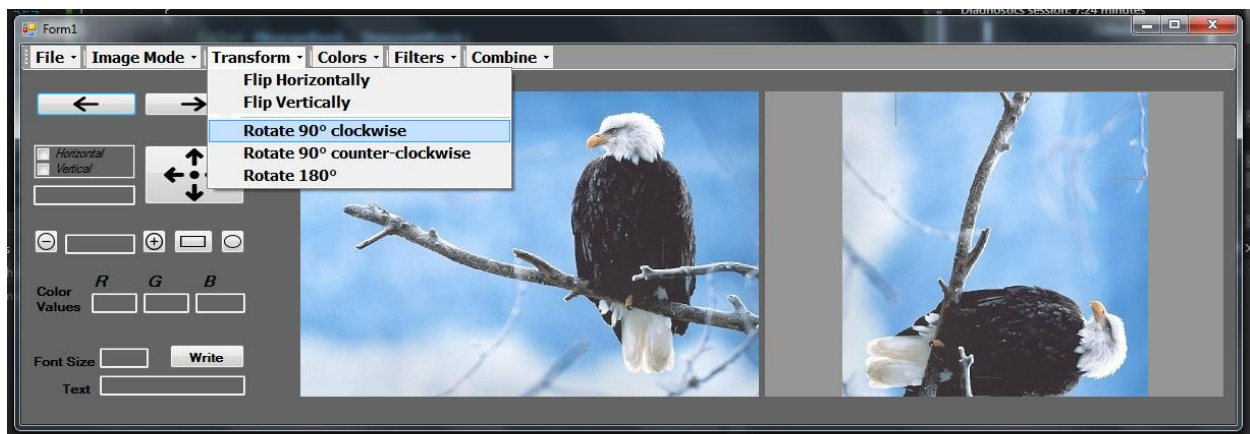
Error List

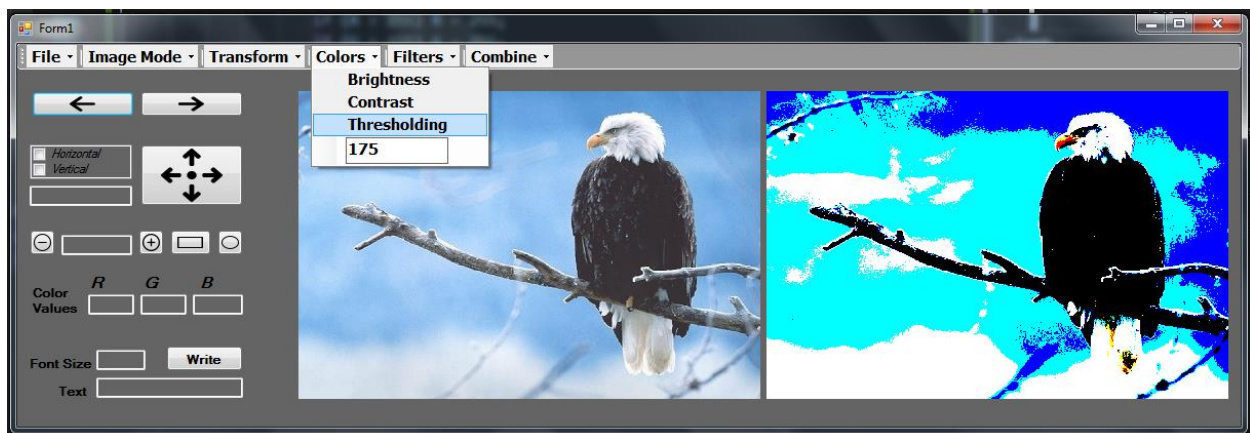
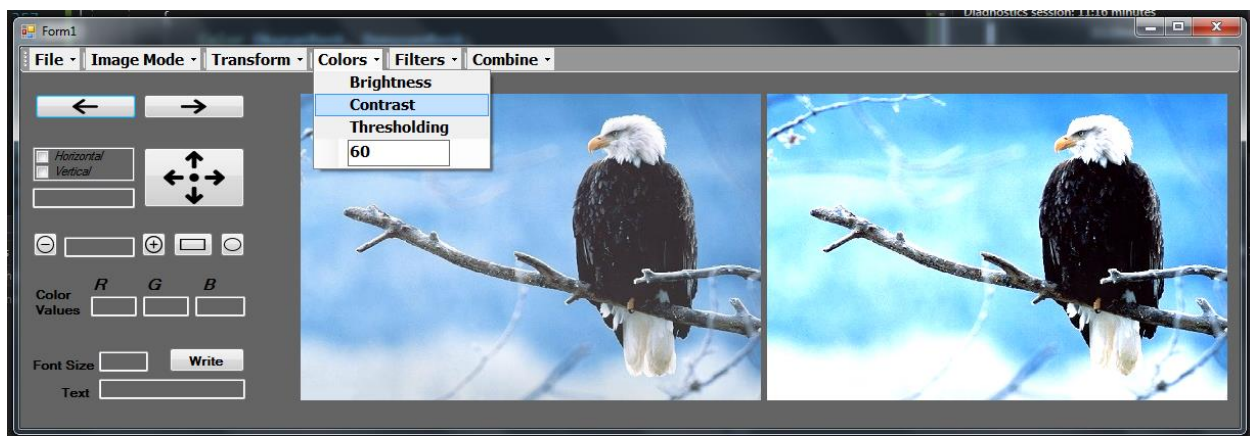
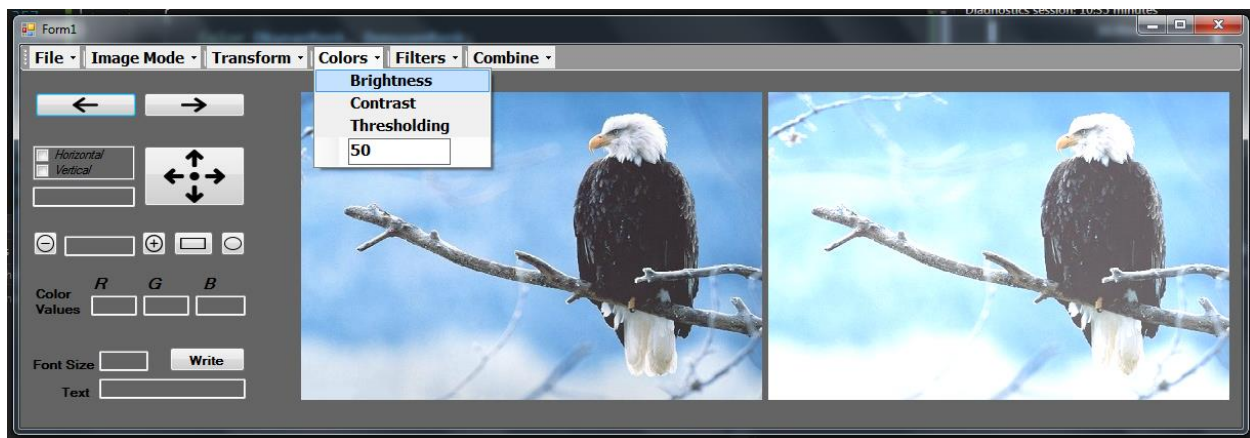
B) Resim İşleme Programı:

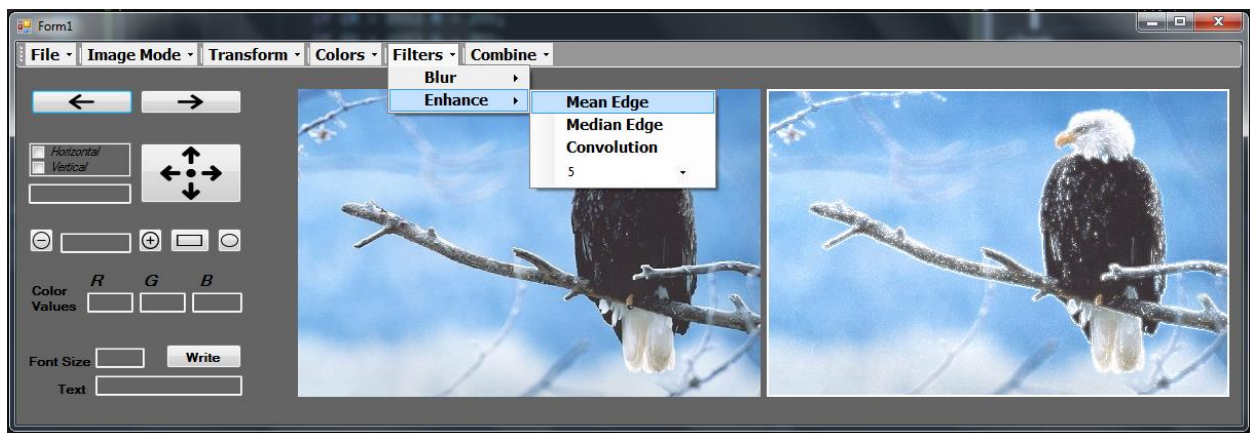


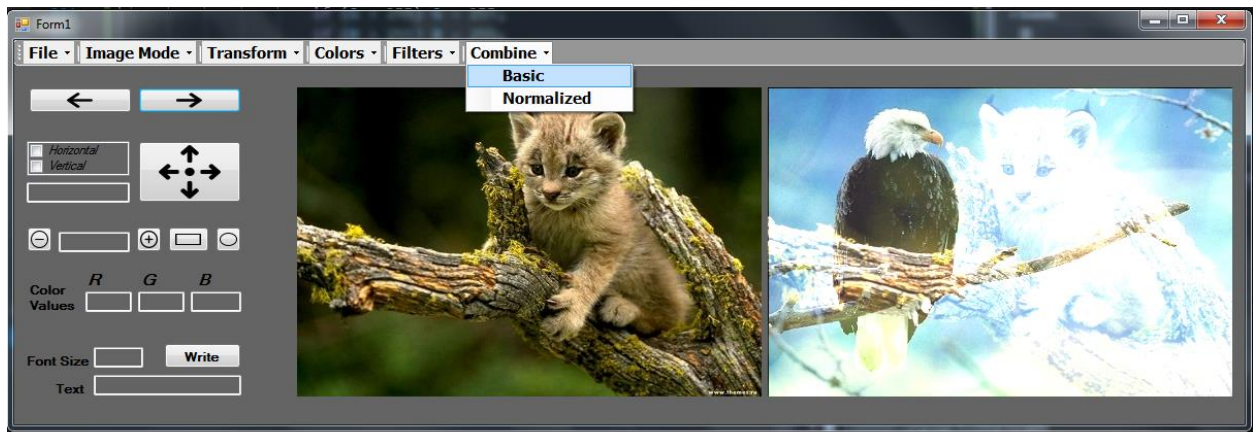
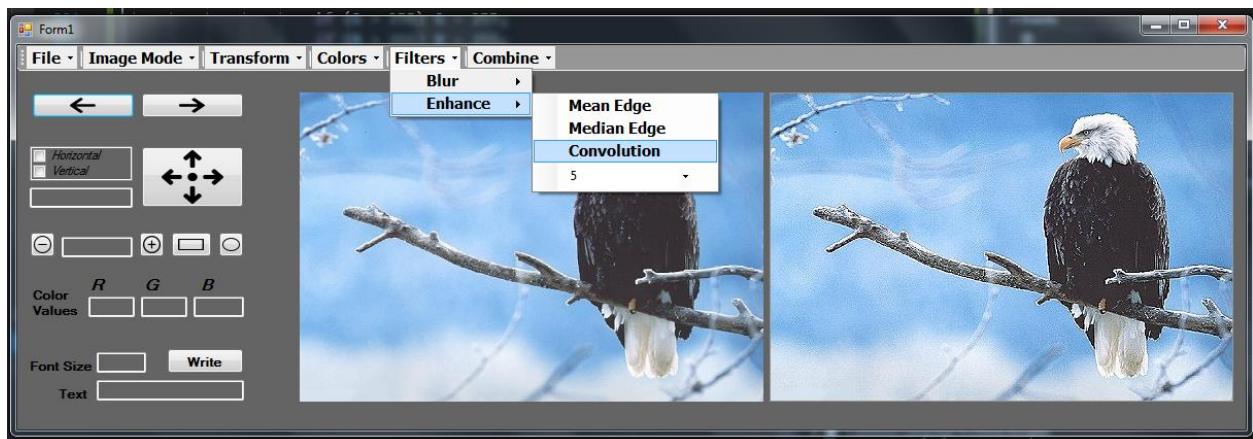
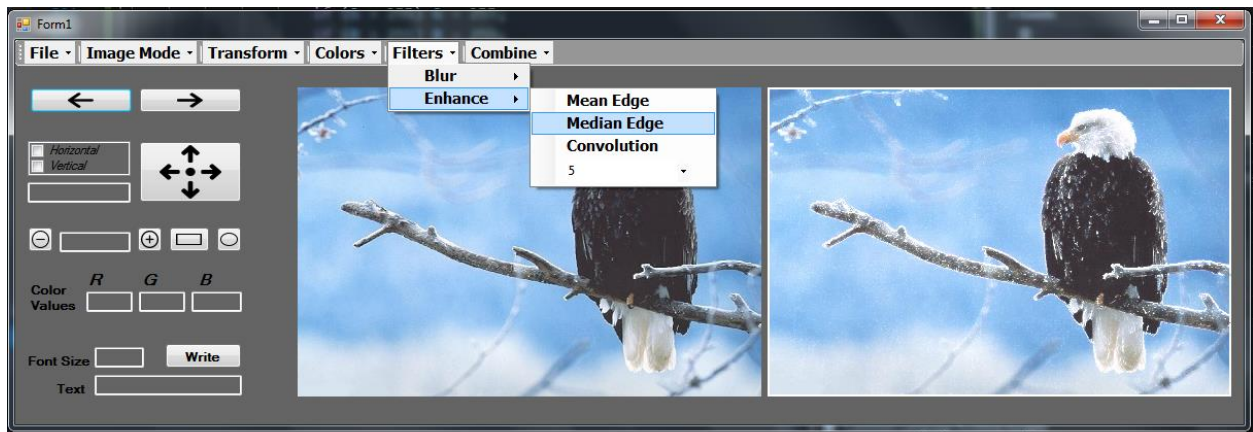


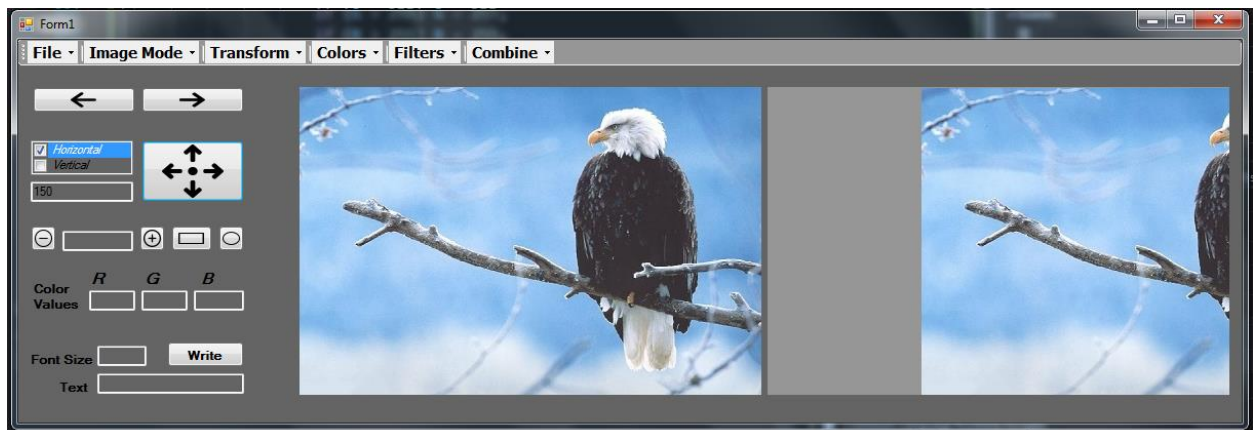
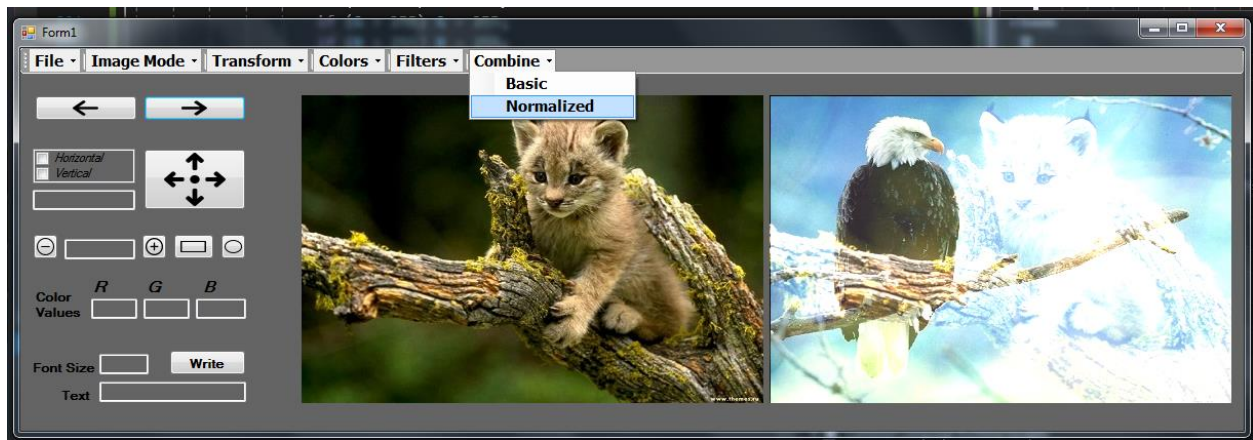




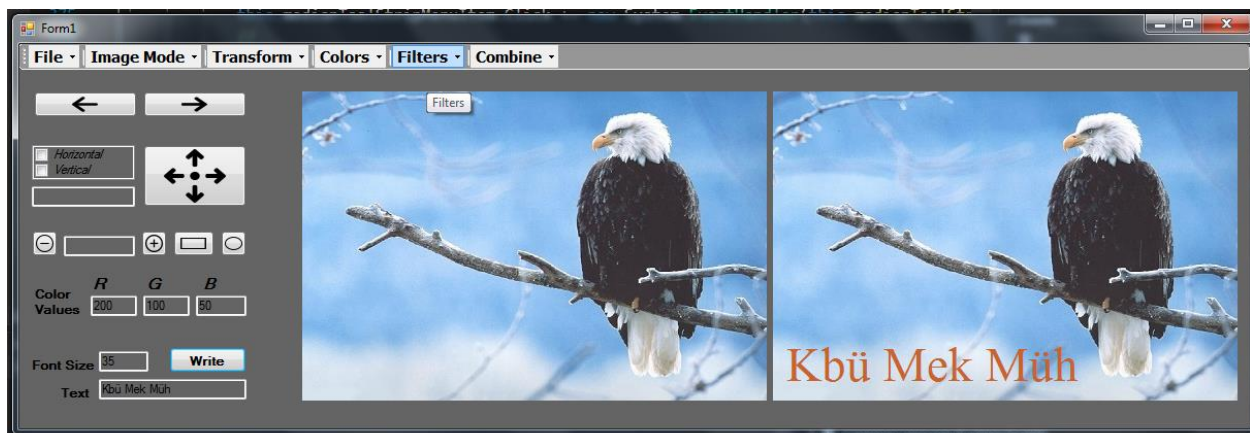
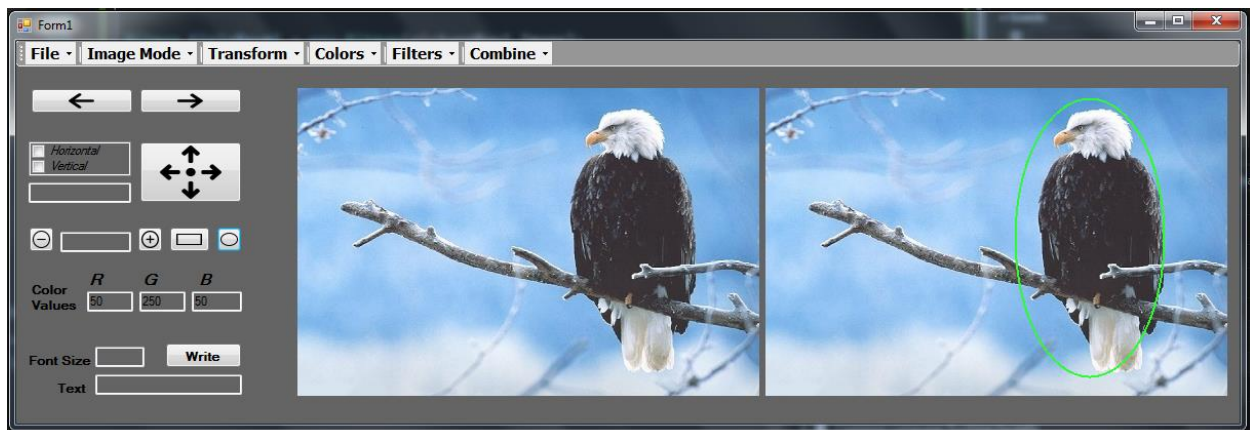
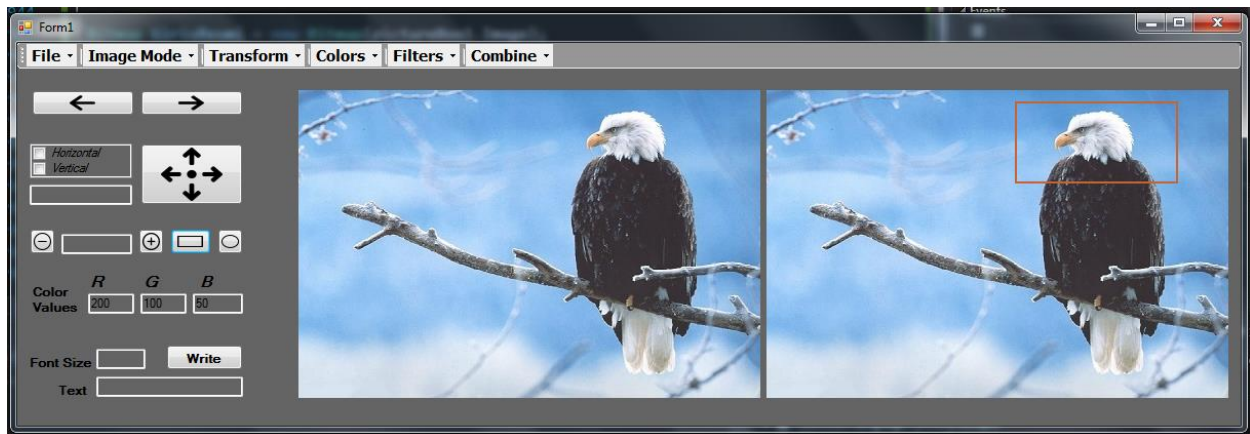












Code:

```
using System;

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

        private void openToolStripMenuItem_Click(object sender, EventArgs e)
        {
            try
            {

```

```
openFileDialog1.DefaultExt = ".jpg";

openFileDialog1.Filter = "Image Files(*.BMP;*.JPG;*.GIF)|*.BMP;*.JPG;*.GIF|All files (*.*)|*.*";

openFileDialog1.ShowDialog();

String ResminYolu = openFileDialog1.FileName;

pictureBox1.Image = Image.FromFile(ResminYolu);

}

catch { }

}
```

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

{

    SaveFileDialog saveFileDialog1 = new SaveFileDialog();

    saveFileDialog1.Filter = "Jpeg Resmi | *.jpg | Bitmap Resmi | *.bmp | Gif Resmi | *.gif";

    saveFileDialog1.Title = "Resmi Kaydet";

    saveFileDialog1.ShowDialog();

    if (saveFileDialog1.FileName != "")

    {

        FileStream DosyaAkisi = (FileStream)saveFileDialog1.OpenFile();

        switch (saveFileDialog1.FilterIndex)

        {

            case 1:

                pictureBox2.Image.Save(DosyaAkisi, System.Drawing.Imaging.ImageFormat.Jpeg);

                break;

            case 2:

                pictureBox2.Image.Save(DosyaAkisi, System.Drawing.Imaging.ImageFormat.Bmp);

            
```

```
        break;

    case 3:

        pictureBox2.Image.Save(DosyaAkisi, System.Drawing.Imaging.ImageFormat.Gif);

        break;

    }

    DosyaAkisi.Close();

}

}
```

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

{

    Color OkunanRenk, DonusenRenk;

    int R = 0, G = 0, B = 0;

    Bitmap GirisResmi, CikisResmi;

    GirisResmi = new Bitmap(pictureBox1.Image);

    int ResimGenisligi = GirisResmi.Width;

    int ResimYuksekligi = GirisResmi.Height;

    CikisResmi = new Bitmap(ResimGenisligi, ResimYuksekligi);

    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 methodToolStripMenuItem_Click(object sender, EventArgs e)
{
    Color OkunanRenk, DonusenRenk;

    Bitmap GirisResmi = new Bitmap(pictureBox1.Image);

    int ResimGenisligi = GirisResmi.Width;

    int ResimYuksekligi = GirisResmi.Height;

    Bitmap CikisResmi = new Bitmap(ResimGenisligi, ResimYuksekligi);

    int GriDeger = 0;

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

            double R = OkunanRenk.R;

            double G = OkunanRenk.G;

            double B = OkunanRenk.B;

            GriDeger = Convert.ToInt16((R + G + B) / 3);

```

```

        DonusenRenk = Color.FromArgb(GriDeger, GriDeger, GriDeger);

        CikisResmi.SetPixel(x, y, DonusenRenk);

    }

}

pictureBox2.Image = CikisResmi;

}

```

```

private void methodToolStripMenuItem1_Click(object sender, EventArgs e)
{
    Color OkunanRenk, DonusenRenk;

    Bitmap GirisResmi = new Bitmap(pictureBox1.Image);

    int ResimGenisligi = GirisResmi.Width;

    int ResimYuksekligi = GirisResmi.Height;

    Bitmap CikisResmi = new Bitmap(ResimGenisligi, ResimYuksekligi);

    int GriDeger = 0;

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

            double R = OkunanRenk.R;

            double G = OkunanRenk.G;

            double B = OkunanRenk.B;

            GriDeger = Convert.ToInt16(R * 0.3 + G * 0.6 + B * 0.1);

            DonusenRenk = Color.FromArgb(GriDeger, GriDeger, GriDeger);

```

```

        CıkisResmi.SetPixel(x, y, DonusenRenk);

    }

}

pictureBox2.Image = CıkisResmi;
}

private void flipHorizontallyToolStripMenuItem_Click(object sender, EventArgs e)
{
    Color OkunanRenk;

    Bitmap GirisResmi, CıkisResmi;

    GirisResmi = new Bitmap(pictureBox1.Image);

    int ResimGenisligi = GirisResmi.Width;

    int ResimYuksekligi = GirisResmi.Height;

    CıkisResmi = new Bitmap(ResimGenisligi, ResimYuksekligi);

    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 = Convert.ToInt16(x1);

```

```

        y2 = Convert.ToInt16(-y1 + 2 * y0);

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

            CikisResmi.SetPixel((int)x2, (int)y2, OkunanRenk);

        pictureBox2.Image = CikisResmi;

    }

}

}

private void flipVerticallyToolStripMenuItem_Click(object sender, EventArgs e)
{
    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;

    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 = Convert.ToInt16(-x1 + 2 * x0);

```



```

        y2 = Convert.ToInt16(y1);

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

            CikisResmi.SetPixel((int)x2, (int)y2, OkunanRenk);

        pictureBox2.Image = CikisResmi;

    }

}

}

private void rotate90ClockwiseToolStripMenuItem_Click(object sender, EventArgs e)
{
    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 = 90;

    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 rotate180ToolStripMenuItem_Click(object sender, EventArgs e)

```

```

{
    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 = -90;

    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 rotate180ToolStripMenuItem1_Click(object sender, EventArgs e)
{
    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 brightnessToolStripMenuItem_Click(object sender, EventArgs e)
{
    int R = 0, G = 0, B = 0;

    Color OkunanRenk, DonusenRenk;

    Bitmap GirisResmi, CikisResmi;

    GirisResmi = new Bitmap(pictureBox1.Image);

    int ResimGenisligi = GirisResmi.Width;

    int ResimYuksekligi = GirisResmi.Height;

    CikisResmi = new Bitmap(ResimGenisligi, ResimYuksekligi);

    int i = 0, j = 0;

```

```

int derece = Convert.ToInt32(toolStripTextBox1.Text);

for (int x = 0; x < ResimGenisligi; x++)
{
    j = 0;

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

        R = OkunanRenk.R + derece;

        G = OkunanRenk.G + derece;

        B = OkunanRenk.B + derece;

        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 contrastToolStripMenuItem_Click(object sender, EventArgs e)
{
    int R = 0, G = 0, B = 0;

```

```
Color OkunanRenk, DonusenRenk;

Bitmap GirisResmi, CikisResmi;

GirisResmi = new Bitmap(pictureBox1.Image);

int ResimGenisligi = GirisResmi.Width;

int ResimYuksekligi = GirisResmi.Height;

CikisResmi = new Bitmap(ResimGenisligi, ResimYuksekligi);

double C_KontrastSeviyesi = Convert.ToInt32(toolStripTextBox1.Text);

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);

        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 button1_Click(object sender, EventArgs e)
{
    Color OkunanRenk, DonusenRenk;

    int R = 0, G = 0, B = 0;

    Bitmap GirisResmi, CikisResmi;

    GirisResmi = new Bitmap(pictureBox2.Image);

    int ResimGenisligi = GirisResmi.Width;

    int ResimYuksekligi = GirisResmi.Height;

    CikisResmi = new Bitmap(ResimGenisligi, ResimYuksekligi);

    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;

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

```

```

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

pictureBox1.Image = CikisResmi;
}

private void button2_Click(object sender, EventArgs e)
{
    Color OkunanRenk, DonusenRenk;

    int R = 0, G = 0, B = 0;

    Bitmap GirisResmi, CikisResmi;

    GirisResmi = new Bitmap(pictureBox1.Image);

    int ResimGenisligi = GirisResmi.Width;

    int ResimYuksekligi = GirisResmi.Height;

    CikisResmi = new Bitmap(ResimGenisligi, ResimYuksekligi);

    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;

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

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

```



```
    }  
}  
  
pictureBox2.Image = CikisResmi;  
}
```

```
private void thresholdingToolStripMenuItem_Click(object sender, EventArgs e)  
{  
    int R = 0, G = 0, B = 0;  
  
    Color OkunanRenk, DonusenRenk;  
  
    Bitmap GirisResmi, CikisResmi;  
  
    GirisResmi = new Bitmap(pictureBox1.Image);  
  
    int ResimGenisligi = GirisResmi.Width;  
  
    int ResimYuksekligi = GirisResmi.Height;  
  
    CikisResmi = new Bitmap(ResimGenisligi, ResimYuksekligi);  
  
    int EsiklemeDegeri = Convert.ToInt32(toolStripTextBox1.Text);  
  
    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 btnMove_Click(object sender, EventArgs e)
{
    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;

    int TasimaDegeri = Convert.ToInt32(textBox6.Text);

    for (int x1 = 0; x1 < (ResimGenisligi); x1++)

```

```

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

        if (checkedListBoxTasima.GetItemCheckState(0) == CheckState.Checked)
        {
            x2 = x1 + TasimaDegeri;

            y2 = y1;
        }

        if (checkedListBoxTasima.GetItemCheckState(1) == CheckState.Checked)
        {
            x2 = x1;

            y2 = y1 + TasimaDegeri;
        }

        if (checkedListBoxTasima.GetItemCheckState(0) == CheckState.Checked &&
checkedListBoxTasima.GetItemCheckState(1) == CheckState.Checked)
        {
            x2 = x1 + TasimaDegeri;

            y2 = y1 + TasimaDegeri;
        }

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

            CikisResmi.SetPixel((int)x2, (int)y2, OkunanRenk);
        }
    }

    pictureBox2.Image = CikisResmi;
}

```

```
private void button3_Click(object sender, EventArgs e)
{
    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;

    int KucultmeKatsayisi = Convert.ToInt32(textBox1.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 button4_Click(object sender, EventArgs e)
{
    Color OkunanRenk;

    Bitmap GirisResmi, CikisResmi;

    GirisResmi = new Bitmap(pictureBox1.Image);

    int ResimGenisligi = GirisResmi.Width;

    int ResimYuksekligi = GirisResmi.Height;

    CikisResmi = new Bitmap(ResimGenisligi, ResimYuksekligi);

    int BuyutmeKatsayisi = Convert.ToInt32(textBox1.Text);

    int x2 = 0, y2 = 0;

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

            for (int i = 0; i < BuyutmeKatsayisi; i++)
            {
                for (int j = 0; j < BuyutmeKatsayisi; j++)
                {
                    x2 = x1 * BuyutmeKatsayisi + i;

                    y2 = y1 * BuyutmeKatsayisi + j;

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

                        CikisResmi.SetPixel(x2, y2, OkunanRenk);
                }
            }
        }
    }
}
```

```

    }

    }

    pictureBox2.Image = CikisResmi;
}

private void meanToolStripMenuItem_Click(object sender, EventArgs e)
{
    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 = Convert.ToInt32(toolStripComboBox1.SelectedItem);

    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));

}

}

pictureBox2.Image = CikisResmi;

}

```

```

private void medianToolStripMenuItem_Click(object sender, EventArgs e)
{
    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 = Convert.ToInt16(toolStripComboBox1.SelectedItem);

```

```

int ElemanSayisi = SablonBoyutu * SablonBoyutu;

int[] R = new int[ElemanSayisi];

int[] G = new int[ElemanSayisi];

int[] B = new int[ElemanSayisi];

int[] Gri = new int[ElemanSayisi];

int x, y, i, j;

for (x = (SablonBoyutu - 1) / 2; x < ResimGenisligi - (SablonBoyutu - 1) / 2; x++)
{
    for (y = (SablonBoyutu - 1) / 2; y < ResimYuksekligi - (SablonBoyutu - 1) / 2; y++)
    {
        int k = 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);

                R[k] = OkunanRenk.R;

                G[k] = OkunanRenk.G;

                B[k] = OkunanRenk.B;

                Gri[k] = Convert.ToInt16(R[k] * 0.299 + G[k] * 0.587 + B[k] * 0.114);

                k++;
            }
        }
    }

    int GeciciSayi = 0;

    for (i = 0; i < ElemanSayisi; i++)

```



```

{
    for (j = i + 1; j < ElemanSayisi; j++)
    {
        if (Gri[j] < Gri[i])
        {
            GeciciSayi = Gri[i];

            Gri[i] = Gri[j];

            Gri[j] = GeciciSayi;

            GeciciSayi = R[i];

            R[i] = R[j];

            R[j] = GeciciSayi;

            GeciciSayi = G[i];

            G[i] = G[j];

            G[j] = GeciciSayi;

            GeciciSayi = B[i];

            B[i] = B[j];

            B[j] = GeciciSayi;
        }
    }
}

CikisResmi.SetPixel(x, y, Color.FromArgb(R[(ElemanSayisi - 1) / 2], G[(ElemanSayisi - 1) /
2], B[(ElemanSayisi - 1) / 2]));

}

}

pictureBox2.Image = CikisResmi;

```

```
}
```

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

```
{
```

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

```
    Bitmap BulanikResim = MeanFiltresi(OrjinalResim);
```

```
    Bitmap KenarGoruntusu = OrjinalResimdenBulanikResmiCikarma(OrjinalResim, BulanikResim);
```

```
    Bitmap NetlesmisResim = KenarGoruntusulleOrjinalResmiBirlestir(OrjinalResim,  
KenarGoruntusu);
```

```
    pictureBox2.Image = NetlesmisResim;
```

```
}
```

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

```
{
```

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

```
    Bitmap BulanikResim = MedianFiltresi(OrjinalResim);
```

```
    Bitmap KenarGoruntusu = OrjinalResimdenBulanikResmiCikarma(OrjinalResim, BulanikResim);
```

```
    Bitmap NetlesmisResim = KenarGoruntusulleOrjinalResmiBirlestir(OrjinalResim, KenarGoruntusu);
```

```
    pictureBox2.Image = NetlesmisResim;
```

```
}
```

```
public Bitmap MeanFiltresi(Bitmap OrjinalResim)
```

```
{
```

```
    Color OkunanRenk;
```

```
    Bitmap GirisResmi, CikisResmi;
```

```
    GirisResmi = OrjinalResim;
```

```
int ResimGenisligi = GirişResmi.Width;

int ResimYuksekligi = GirişResmi.Height;

CikisResmi = new Bitmap(ResimGenisligi, ResimYuksekligi);

int SablonBoyutu = Convert.ToInt16(toolStripComboBox2.SelectedItem);

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 = GirişResmi.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);
```

```

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

```

```

public Bitmap MedianFiltresi(Bitmap OrjinalResim)
{
    Color OkunanRenk;
    Bitmap GirisResmi, CıkisResmi;
    GirisResmi = new Bitmap(pictureBox1.Image);
    int ResimGenisligi = GirisResmi.Width;
    int ResimYuksekligi = GirisResmi.Height;
    CıkisResmi = new Bitmap(ResimGenisligi, ResimYuksekligi);
    int SablonBoyutu = Convert.ToInt16(toolStripComboBox2.SelectedItem);
    int ElemanSayisi = SablonBoyutu * SablonBoyutu;
    int[] R = new int[ElemanSayisi];
    int[] G = new int[ElemanSayisi];
    int[] B = new int[ElemanSayisi];
    int[] Gri = new int[ElemanSayisi];
    int x, y, i, j;
    for (x = (SablonBoyutu - 1) / 2; x < ResimGenisligi - (SablonBoyutu - 1) / 2; x++)
    {
        for (y = (SablonBoyutu - 1) / 2; y < ResimYuksekligi - (SablonBoyutu - 1) / 2; y++)
        {

```

```

int k = 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);

        R[k] = OkunanRenk.R;

        G[k] = OkunanRenk.G;

        B[k] = OkunanRenk.B;

        Gri[k] = Convert.ToInt16(R[k] * 0.299 + G[k] * 0.587 + B[k] * 0.114);

        k++;
    }
}

int GeciciSayi = 0;

for (i = 0; i < ElemanSayisi; i++)
{
    for (j = i + 1; j < ElemanSayisi; j++)
    {
        if (Gri[j] < Gri[i])
        {
            GeciciSayi = Gri[i];

            Gri[i] = Gri[j];

            Gri[j] = GeciciSayi;

            GeciciSayi = R[i];

            R[i] = R[j];

```

```

        R[j] = GeciciSayi;

        GeciciSayi = G[i];

        G[i] = G[j];

        G[j] = GeciciSayi;

        GeciciSayi = B[i];

        B[i] = B[j];

        B[j] = GeciciSayi;

    }

}

}

CikisResmi.SetPixel(x, y, Color.FromArgb(R[(ElemanSayisi - 1) / 2], G[(ElemanSayisi - 1) /
2], B[(ElemanSayisi - 1) / 2]));

}

}

return CikisResmi;

}

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 = 2;

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 * Math.Abs(OkunanRenk1.R - OkunanRenk2.R));

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

        B = Convert.ToInt16(Olcekleme * Math.Abs(OkunanRenk1.B - OkunanRenk2.B));

        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 KenarGoruntusulleOrjinalResmiBirlestir(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;

        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 convolutionToolStripMenuItem_Click(object sender, EventArgs e)
{
    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 = 3;

    int ElemanSayisi = SablonBoyutu * SablonBoyutu;

    int x, y, i, j, toplamR, toplamG, toplamB;

    int R, G, B;

    int[] Matris = { 0, -2, 0, -2, 11, -2, 0, -2, 0 };

    int MatrisToplami = 0 + -2 + 0 + -2 + 11 + -2 + 0 + -2 + 0;

    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;

            int k = 0;

            for (i = -((SablonBoyutu - 1) / 2); i <= (SablonBoyutu - 1) / 2; i++)
```

```

{
    for (j = -((SablonBoyutu - 1) / 2); j <= (SablonBoyutu - 1) / 2; j++)
    {
        OkunanRenk = GirişResmi.GetPixel(x + i, y + j);

        toplamR = toplamR + OkunanRenk.R * Matris[k];

        toplamG = toplamG + OkunanRenk.G * Matris[k];

        toplamB = toplamB + OkunanRenk.B * Matris[k];

        k++;
    }
}

R = toplamR / MatrisToplami;

G = toplamG / MatrisToplami;

B = toplamB / MatrisToplami;

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;

CikisResmi.SetPixel(x, y, Color.FromArgb(R, G, B));

}

}

pictureBox2.Image = CikisResmi;

}

```

```
public int XK, YK, XD, YD, XU, YU;
```

```
private void pictureBox1_MouseClick(object sender, MouseEventArgs e)
```

```
{
```

```
    XK = e.X;
```

```
    YK = e.Y;
```

```
}
```

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

```
{
```

```
    Bitmap Resim1, Resim2, CikisResmi;
```

```
    Resim1 = new Bitmap(pictureBox1.Image);
```

```
    Resim2 = new Bitmap(pictureBox2.Image);
```

```
    int ResimGenisligi = Resim1.Width;
```

```
    int ResimYuksekligi = Resim1.Height;
```

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

```
    Color Renk1, Renk2;
```

```
    int x, y;
```

```
    int R = 0, G = 0, B = 0;
```

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

```
{
```

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

```
{
```

```
            Renk1 = Resim1.GetPixel(x, y);
```

```

        Renk2 = Resim2.GetPixel(x, y);

        R = Renk1.R + Renk2.R;

        G = Renk1.G + Renk2.G;

        B = Renk1.B + Renk2.B;

        if (R > 255) R = 255;

        if (G > 255) G = 255;

        if (B > 255) B = 255;

        CikisResmi.SetPixel(x, y, Color.FromArgb(R, G, B));

    }

}

pictureBox2.Image = CikisResmi;

}

private void normalaizedToolStripMenuItem_Click(object sender, EventArgs e)
{
    Bitmap Resim1, Resim2, CikisResmi;

    Resim1 = new Bitmap(pictureBox1.Image);

    Resim2 = new Bitmap(pictureBox2.Image);

    int ResimGenisligi = Resim1.Width;

    int ResimYuksekligi = Resim1.Height;

    CikisResmi = new Bitmap(ResimGenisligi, ResimYuksekligi);

    Color Renk1, Renk2;

    int x, y;

    int R = 0, G = 0, B = 0;

    int EnBuyukDeger = 0, EnKucukDeger = 0;

```

```
for (x = 0; x < ResimGenisligi; x++)
{
    for (y = 0; y < ResimYuksekligi; y++)
    {
        Renk1 = Resim1.GetPixel(x, y);
        Renk2 = Resim2.GetPixel(x, y);

        R = Renk1.R + Renk2.R;
        G = Renk1.G + Renk2.G;
        B = Renk1.B + Renk2.B;
        int Gri = (R + G + B) / 3;
        if (Gri > EnBuyukDeger)
            EnBuyukDeger = Gri;
        if (Gri < EnKucukDeger)
            EnKucukDeger = Gri;

        if (R > 255) R = 255;
        if (G > 255) G = 255;
        if (B > 255) B = 255;

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

pictureBox2.Image = CikisResmi;
}
```

```
private void btnYaz_Click(object sender, EventArgs e)
{
    Bitmap GirisResmi = new Bitmap(pictureBox1.Image);

    Bitmap CikisResmi = new Bitmap(GirisResmi);

    CikisResmi = GirisResmi;

    int FontBoyutu = Convert.ToInt32(textBox9.Text);

    Font FontBilgisi = new Font("Times New Roman", FontBoyutu, FontStyle.Regular);

    string Metin = textBox10.Text;

    Color Renk = Color.FromArgb(255, Convert.ToInt32(textBox11.Text),
    Convert.ToInt32(textBox12.Text), Convert.ToInt32(textBox13.Text)); //Alfa,Red,Green,Blue ile
    tanımladık

    SolidBrush firs = new SolidBrush(Renk);

    int xKonum = XK;

    int yKonum = YK;

    Graphics Grafik = Graphics.FromImage(CikisResmi);

    Grafik.DrawString(Metin, FontBilgisi, firs, xKonum, yKonum);

    pictureBox2.Image = CikisResmi;
}

private void pictureBox1_MouseUp(object sender, MouseEventArgs e)
{
    XU = e.X;

    YU = e.Y;
}
```

```
private void pictureBox1_MouseDown_1(object sender, MouseEventArgs e)
{
    XD = e.X;
    YD = e.Y;
}
```

```
private void button6_Click(object sender, EventArgs e)
{
    Bitmap GirisResmi = new Bitmap(pictureBox1.Image);
    Bitmap CikisResmi = new Bitmap(GirisResmi);
    CikisResmi = GirisResmi;
    Graphics Grafik = Graphics.FromImage(CikisResmi);
    Color Renk = Color.FromArgb(255, Convert.ToInt32(textBox11.Text),
Convert.ToInt32(textBox12.Text), Convert.ToInt32(textBox13.Text));
    Pen KalemCizgi = new Pen(Renk, 2);
    Grafik.DrawEllipse(KalemCizgi, XD, YD, XU - XD, YU - YD);
    pictureBox2.Image = CikisResmi;
}
```

```
private void button5_Click(object sender, EventArgs e)
{
    Bitmap GirisResmi = new Bitmap(pictureBox1.Image);
    Bitmap CikisResmi = new Bitmap(GirisResmi);
    CikisResmi = GirisResmi;
    Graphics Grafik = Graphics.FromImage(CikisResmi);
```

```
Color Renk = Color.FromArgb(255, Convert.ToInt32(textBox11.Text),  
Convert.ToInt32(textBox12.Text), Convert.ToInt32(textBox13.Text));
```

```
Pen KalemCizgi = new Pen(Renk, 2);
```

```
Grafik.DrawRectangle(KalemCizgi, XD, YD, XU - XD, YU - YD);
```

```
pictureBox2.Image = CikisResmi;
```

```
}
```

```
}
```

```
}
```