

AD SOYAD	CENGİZHAN TOPÇU	KARABÜK ÜNİVERSİTESİ MÜHENDİSLİK FAKÜLTESİ MEKATRONİK MÜHENDİSLİĞİ
NUMARA	2017010225048	
GÖRÜNTÜ İŞLEME ÖDEV-3		

ÖDEV-3

Filtreleri Karşılaştırma

☐ Mean

☐ Medyan

☐ Gauss

İşlemi Gerçekleştir

Seçilen Alanı Bulanıklaştırma

İşlemi Gerçekleştir

Şablon Boyutu

X1

Y1

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

        //Resim Yükle
        private void BtnAc_Click(object sender, EventArgs e)
        {
            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");
            }
        }
    }
}
```

```

    }
}

//Resim Kaydet
private void BtnKaydet_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 != "") //Dosya adı boş değilse kaydedecek.
    {
        // FileStream nesnesi ile kayıtlı gerçekleştirecek.
        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();
    }
}

//Resim Boyutu Ayarlama
private void btnBoyutAyarla_Click(object sender, EventArgs e)
{
    pictureBox1.SizeMode = PictureBoxSizeMode.Zoom;
    pictureBox2.SizeMode = PictureBoxSizeMode.Zoom;
}

//Resmi Sağa Aktarma
private void btnSagaAktar_Click(object sender, EventArgs e)
{
    //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 = GirisResmi.Height;
    CikisResmi = new Bitmap(ResimGenisligi, ResimYuksekligi); //Cikis
resmini oluşturuyor. Boyutları giriş resmi ile aynı olur. Tanımlaması globalde
yapıldı.

    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;

        //if (x>100 && x<200)
        DonusenRenk = Color.FromArgb(R, G, B);
        CikisResmi.SetPixel(x, y, DonusenRenk);

    }
}
pictureBox2.Image = CikisResmi;
}

//Resmi Sola Aktarma
private void btnSolaAktar_Click(object sender, EventArgs e)
{
    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.

    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;
}

//Mean Filtresi
public Bitmap MeanFiltresi()
{
    Color OkunanRenk;
    Bitmap GirişResmi, CikisResmi;
    GirişResmi = new Bitmap(pictureBox1.Image);

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

    CikisResmi = new Bitmap(ResimGenisligi, ResimYuksekligi);

    int SablonBoyutu = Convert.ToInt32(textBox1.Text); //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;
    }

    //Medyan Filtresi
    public void MedyanFiltresi()
    {
        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(textBox1.Text); //şablon boyutu 3
den büyük tek rakam olmalıdır (3,5,7 gibi).
        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++)
            {
                //Şablon bölgesi (çekirdek matris) içindeki pikselleri tarıyor.
                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); //Gri ton formülü-1

                        k++;
                    }
                }

                //Gri tona göre sıralama yapıyor. Aynı anda üç rengide
değiştiriyor.
                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;
                        }
                    }
                }

                //Sıralama sonrası ortadaki değeri çıkış resminin piksel değeri
olarak atıyor.
                CikisResmi.SetPixel(x, y, Color.FromArgb(R[(ElemanSayisi - 1) /
2], G[(ElemanSayisi - 1) / 2], B[(ElemanSayisi - 1) / 2]));
            }
        }
        pictureBox2.Image = CikisResmi;
    }

    //Gauss Filtresi
    public Bitmap GaussFiltresi()
    {
        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(textBox1.Text);
//Convert.ToInt16(textBox1.Text); //şablon boyutu 3 den büyük tek rakam olmalıdır
(3,5,7 gibi).
        int ElemanSayisi = SablonBoyutu * SablonBoyutu;

        int x, y, i, j, toplamR, toplamG, toplamB, ortalamaR, ortalamaG,
ortalamaB;
        //int R, G, B, Gri;
        int[] Matris = { 1, 4, 7, 4, 1, 4, 20, 33, 20, 4, 7, 33, 55, 33, 7, 4,
20, 33, 20, 4, 1, 4, 7, 4, 1 };
        int MatrisToplami = 1 + 4 + 7 + 4 + 1 + 4 + 20 + 33 + 20 + 4 + 7 + 33 +
55 + 33 + 7 + 4 + 20 + 33 + 20 + 4 + 1 + 4 + 7 + 4 + 1;

        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++)
            {
                toplamR = 0;
                toplamG = 0;
                toplamB = 0;

                //Şablon bölgesi (çekirdek matris) içindeki pikselleri tarıyor.
                int k = 0; //matris içindeki elemanları sırayla okurken
kullanılacak.
                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 * Matris[k];
                        toplamG = toplamG + OkunanRenk.G * Matris[k];
                        toplamB = toplamB + OkunanRenk.B * Matris[k];

                        ortalamaR = toplamR / MatrisToplami;
                        ortalamaG = toplamG / MatrisToplami;
                        ortalamaB = toplamB / MatrisToplami;

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

                        k++;
                    }
                }
            }
        }
        return CikisResmi;
    }

    private void BtnBulaniklastirma_Click(object sender, EventArgs e)
    {
        //Mean Filtresi
        if (radioButtonMean.Checked == true)
        {
            pictureBox2.Image = MeanFiltresi();
        }

        //Medyan Filtresi
        if (radioButtonMedyan.Checked == true)
        {
            MedyanFiltresi();
        }
    }

```

```

    }

    //Gauss Filtresi
    if (radioButtonGauss.Checked == true)
    {
        pictureBox2.Image = GaussFiltresi();
    }
}

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

private void Form1_Load(object sender, EventArgs e)
{
    CizimAlani = pictureBox1.CreateGraphics();
}

int tiklanma_sayisi = 0;
int x1, y1, x2, y2;

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

    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;
        textBox_x1.Text = e.X.ToString();
        textBox_y1.Text = e.Y.ToString();
    }
    else if (tiklanma_sayisi >= 2)
    {
        x2 = e.X;
        y2 = e.Y;
        CizimAlani.DrawLine(Kalem2, x1, y1, x2, y2);
        x1 = x2;
        y1 = y2;
        textBox_x2.Text = e.X.ToString();
        textBox_y2.Text = e.Y.ToString();
    }
}

//Seçilen Alanı Bulanıklaştırma
private void btnResmiBulaniklastir_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(textBox1.Text); //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));
        }
    }
    pictureBox2.Image = CikisResmi;
}
}
}

```


Şablon Boyutu

X1

Y1

Filtreleri Karşılaştırma

☐ Mean

☐ Medyan

☐ Gauss

İşlemi Gerçekleştir



Şablon Boyutu

X1

Y1

Filtreleri Karşılaştırma

☒ Mean

☐ Medyan

☐ Gauss

İşlemi Gerçekleştir



Şablon Boyutu

X1

Y1

Filtreleri Karşılaştırma

☐ Mean

☒ Medyan

☐ Gauss

İşlemi Gerçekleştir



Şablon Boyutu

X1

Y1

Filtreleri Karşılaştırma

☐ Mean

☐ Medyan

☒ Gauss

İşlemi Gerçekleştir

MEAN

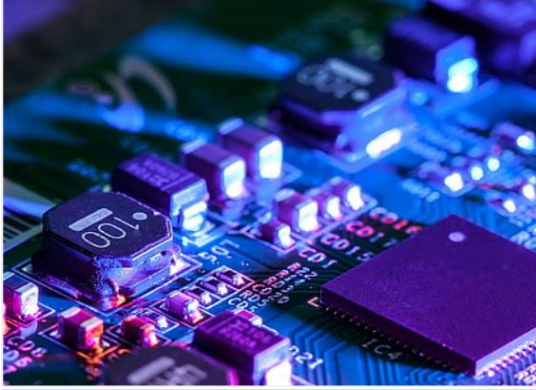


MEDYAN



GAUSS





Şablon Boyutu 5

X1

Y1

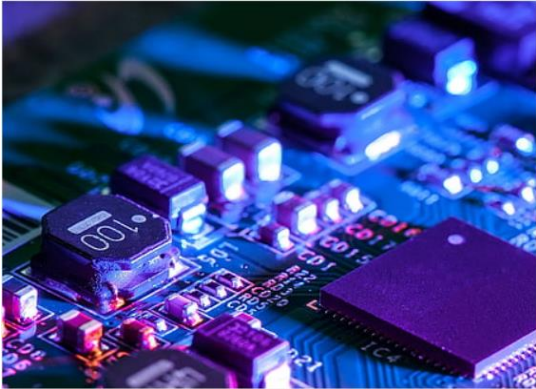
Filtreleri Karşılaştırma

☒ Mean

☐ Medyan

☐ Gauss

İşlemi Gerçekleştir



Şablon Boyutu 5

X1

Y1

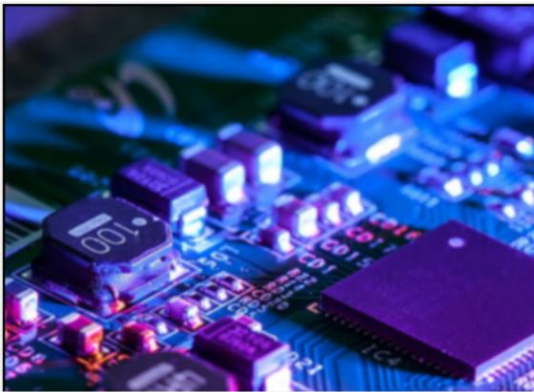
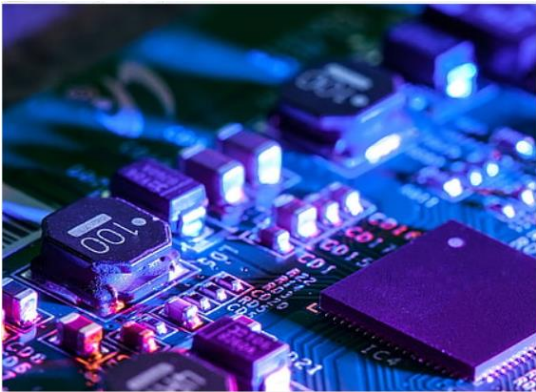
Filtreleri Karşılaştırma

☐ Mean

☒ Medyan

☐ Gauss

İşlemi Gerçekleştir



Şablon Boyutu 5

X1

Y1

Filtreleri Karşılaştırma

☐ Mean

☐ Medyan

☒ Gauss

İşlemi Gerçekleştir



MEAN



MEDYAN



GAUSS

Şablon Boyutu

X1

Y1

Seçilen Alanı Bulanıklaştırma

İşlemi Gerçekleştir

