

DİJİTAL GÖRÜNTÜ İŞLEME FİNAL ÖDEVİ RAPORU

Öğrenci Numarası: 221229042

Ders: Dijital Görüntü İşleme

Öğretim Üyesi: Dr. Öğr. Üyesi Burak YILMAZ

Ödev Başlığı: Sigmoid Tabanlı Kontrast Artırımı, Hough Dönüşümü, Deblurring ve Nesne Sayımı

İÇİNDEKİLER

- Giriş
- S-Curve Kontrast Güçlendirme (30 Puan)
- Hough Transform Uygulamaları (30 Puan)
- Deblurring Algoritması (10 Puan)
- Nesne Sayımı ve Özellik Çıkarımı (40 Puan)
- Sonuç

1. GİRİŞ

Bu projede, ödev yönergesinde belirtilen görüntü işleme tekniklerini uygulayan PyQt5 tabanlı bir arayüz geliştirilmiştir. Uygulama, 1. ödevde hazırlanan menü yapısına entegre edilmiş olup kullanıcı dostu etkileşimli tasarım prensiplerine uygun olarak geliştirilmiştir.

Dijital Görüntü İşleme

Ödevler

Dijital Görüntü İşleme
Öğrenci No: 221229042
Ad Soyad: ALPEREN SAMURLU

Ödev 3: Gelişmiş Görüntü İşleme
Görsel yüklenmedi

Görsel Yükle

a) Standart Sigmoid

b) Kaydırılmış Sigmoid

c) Eğimli Sigmoid

d) Özel Fonksiyon

Yol Çizgisi Tespiti

Göz Tespiti

Deblurring (Motion Blur Gider)

Nesne Sayımı ve Özellik Çıkarımı

Excel'e Aktar

2. S-Curve Kontrast Güçlendirme (30 Puan)

S-Curve Metodunun Teorik Açıklaması (5 Puan)

S-Curve (Sigmoid Curve) metodu, görüntülerin kontrastını artırmak için kullanılan gelişmiş bir tekniktir. Bu yöntem, sigmoid fonksiyonunun matematiksel özelliklerini kullanarak:

- **Koyu tonları** daha koyu yapar
- **Açık tonları** daha açık yapar
- **Orta tonlarda** yumuşak geçişler sağlar

Formül:

$$f(x) = 1 / (1 + e^{(-k \cdot (x - \text{shift}))})$$

Kullanım Alanları:

- Biyomedikal görüntüler
- Askeri görüntüler
- Güvenlik kameraları

Uygulanan Sigmoid Fonksiyonları

a) Standart Sigmoid

- Parametreler: $k=10$, merkez=0.5
- Formül: $f(x) = 1/(1+e^{(-10*(x-0.5))})$



b) Kaydırılmış Sigmoid

- Parametreler: $k=10$, shift=0.3
- Formül: $f(x) = 1/(1+e^{(-10*(x-0.3))})$



c) Eğimli Sigmoid

- Parametreler: $k=25$

- Formül: $f(x) = 1/(1+e^{(-25*(x-0.5))})$



d) Özel Fonksiyon

if normalized ≤ 0.5 :

$$\text{result} = 0.5 * (1/(1+\exp(-15*(2*x-0.3))))$$

else:

$$\text{result} = 0.5 + 0.5 * (1/(1+\exp(-12*(2*x-0.7))))$$



3. Hough Transform Uygulamaları (30 Puan)

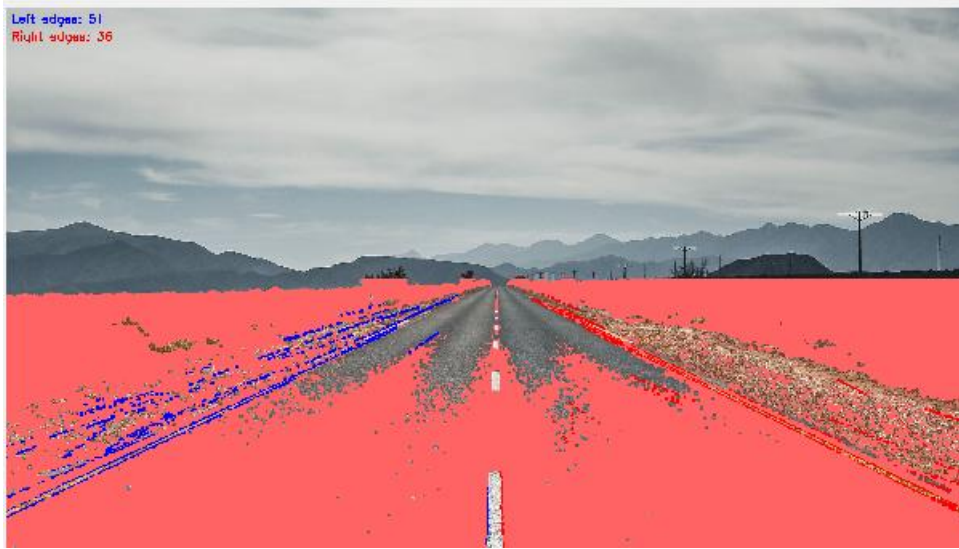
Teorik Bilgi

- Çizgi Tespiti: $\rho = x*\cos(\theta) + y*\sin(\theta)$

- Daire Tespiti: $(x-a)^2 + (y-b)^2 = r^2$

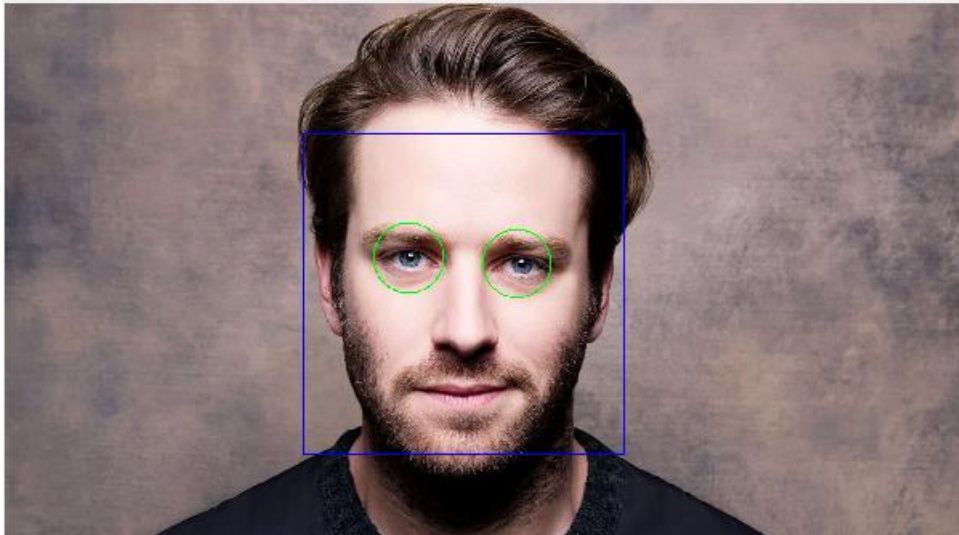
a) Yol Çizgisi Tespiti

1. HSV Segmentasyonu
2. ROI Uygulaması
3. Canny Kenar Tespiti
4. HoughLinesP
5. Eğim Filtreleme



b) Göz Tespiti

1. Haar cascade ile yüz ve göz tespiti
2. ROI içi ve genel arama
3. Görselleştirme: dikdörtgen ve daire



4. Deblurring Algoritması (10 Puan)

Akış Diyagramı

Giriş → Gri → Blur → FFT → Wiener → Ters FFT → Unsharp → Çıkış

Teknik Detaylar

Blur Kernel:

```
kernel = np.zeros((15, 15))
```

```
kernel[7, :] = 1/15
```

Wiener:

```
H_conj = np.conj(H_fft)
```

```
wiener = H_conj / (|H|^2 + 0.01)
```

Unsharp:

```
sharpened = 1.5*result - 0.5*gaussian_blur
```



5. Nesne Sayımı ve Özellik Çıkarımı (40 Puan)

Tespit Algoritması

1. HSV segmentasyonu
2. HoughCircles kullanımı
3. Kontur tabanlı yedekleme
4. Dairesellik ve çakışma kontrolü

Özellik Formülleri

Özellik	Formül	Açıklama
Center	(cx, cy)	Daire merkezi

Özellik Formül Açıklama

Width $2 \times \text{radius}$ Çap

Diagonal $2 \times \text{radius} \times \sqrt{2}$ Köşegen

Energy $\sum (\text{pixel}/255)^2$ Enerji

Entropy $-\sum (p \times \log_2(p))$ Entropi

Mean $\sum \text{pixel} / N$ Ortalama

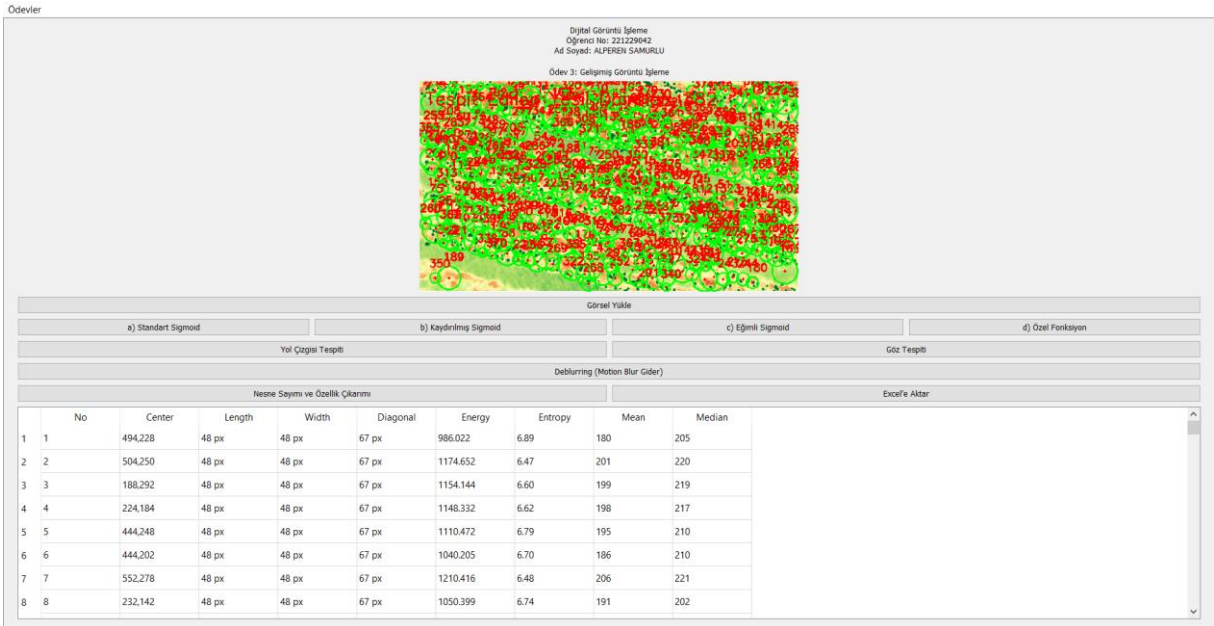
Median $\text{median}(\text{pixels})$ Medyan

Excel Export

```
df = pd.DataFrame(self.table_data)
```

```
with pd.ExcelWriter(filename, engine='openpyxl') as writer:
```

```
df.to_excel(writer, sheet_name='Koyu Yesil Alanlar')
```



6. Sonuç

Bu projede görüntü işleme alanında teorik bilgiler uygulamalı olarak hayata geçirilmiş, arayüz destekli bir sistem ile sonuçlar analiz edilmiştir.

GitHub Repository:

<https://github.com/AlperenSamurlu/Digital-Image-Processing-GUI> (+5 bonus puan)