



MÜHENDİSLİK VE DOĞA BİLİMLERİ FAKÜLTESİ
ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ BÖLÜMÜ

EEM-0412

2020-2021 Bahar Dönemi

GÖRÜNTÜ İŞLEME

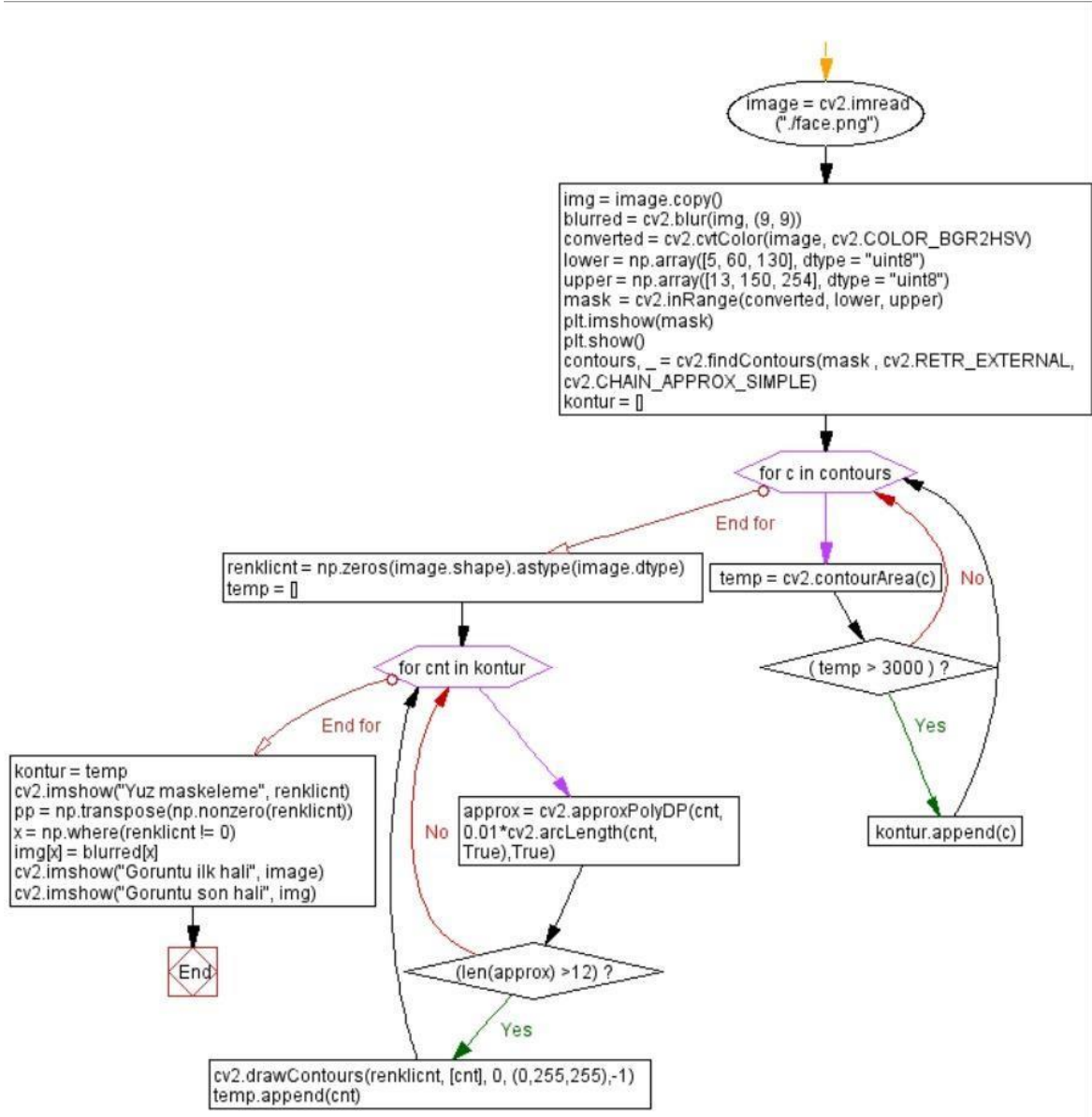
Dersin Sorumlusu

Dr. Öğr. Üyesi Mustafa ÖZDEN

Adı - Soyadı: Hüseyin EREN	ÖDEV NOTU /100
Numarası:	Değerlendiren: Dr.Öğr.Üyesi Mustafa ÖZDEN
İmza:	Rapor Teslim Tarihi: 12.06.21

SORU-2 :

PROBLEMİ ÇÖZMEK İÇİN GELİŞTİRİLEN ALGORİTMA



PROBLEM ÇÖZÜMÜ İÇİN YAZILAN KOD

```
import numpy as np
```

```
import cv2
```

```
import matplotlib.pyplot as plt
```

```
image = cv2.imread("./face.png")
```

```
#görselin bulanık halinin kopyası
```

```
img = image.copy()
```

```
blurred = cv2.blur(img, (9, 9))
```

```
#cv2.imshow("bulanık".format(9, 9), blurred)
```

```
converted = cv2.cvtColor(image, cv2.COLOR_BGR2HSV)
```

```
# Sınırlardan NumPy dizileri oluşturma
```

```
lower = np.array([5, 60, 130], dtype = "uint8")
```

```
upper = np.array([13, 150, 254], dtype = "uint8")
```

```
# Belirlenen sınırlar içindeki renklerin(ten rengi) bulunması ve bölütlemenin uygulanması
```

```
mask = cv2.inRange(converted, lower, upper)
```

```
plt.imshow(mask)
```

```
plt.show()
```

```
contours, _ = cv2.findContours(mask , cv2.RETR_EXTERNAL,  
cv2.CHAIN_APPROX_SIMPLE)
```

```
kontur = []
```

```
for c in contours:
```

```
temp = cv2.contourArea(c)
```

```
if( temp> 3000 ):
```

```
kontur.append(c)
```

```
# -----yüzlerin ayrı görsele maske olarak çıkarılması
```

```
renklicnt = np.zeros(image.shape).astype(image.dtype)
```

```
# ten rengi alanların şekline göre ayıklanması(kol bölgelerinin atılıp sadece  
yüzlerin bırakılması)
```

```
temp = []
```

```
forcnt in kontur:
```

```
approx = cv2.approxPolyDP(cnt, 0.01*cv2.arcLength(cnt, True),True)
```

```
if (len(approx) >12):
```

```
cv2.drawContours(renklicnt, [cnt], 0, (0,255,255),-1)
```

```
temp.append(cnt)
```

```
kontur = temp
```

```
cv2.imshow("Yuz maskeleme", renklicnt)
```

```
#yüze ait piksellerin, bulanık fotoğraftan orjinal fotoğrafa aktarılarak son işlem  
yapılmış olur
```

```
pp = np.transpose(np.nonzero(renklicnt))
```

```
x = np.where(renklicnt != 0)
```

```
img[x] = blurred[x]
```

```
#cv2.drawContours (image, kontur , -1, (0,255,0), 1)
```

```
cv2.imshow("Goruntu ilk hali", image)
```

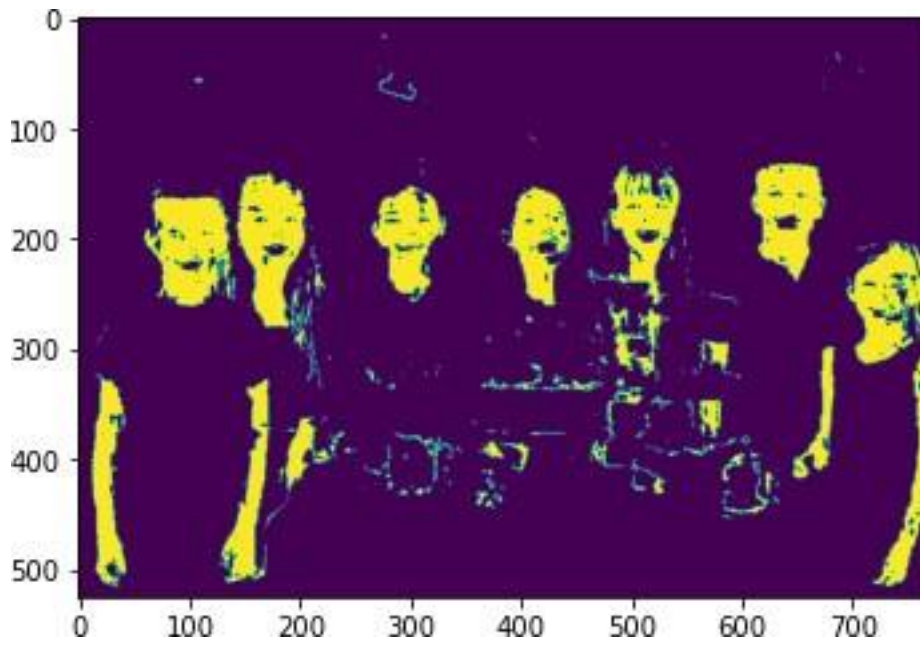
```
cv2.imshow("Goruntu son hali", img)
```

```
cv2.waitKey(0)
```

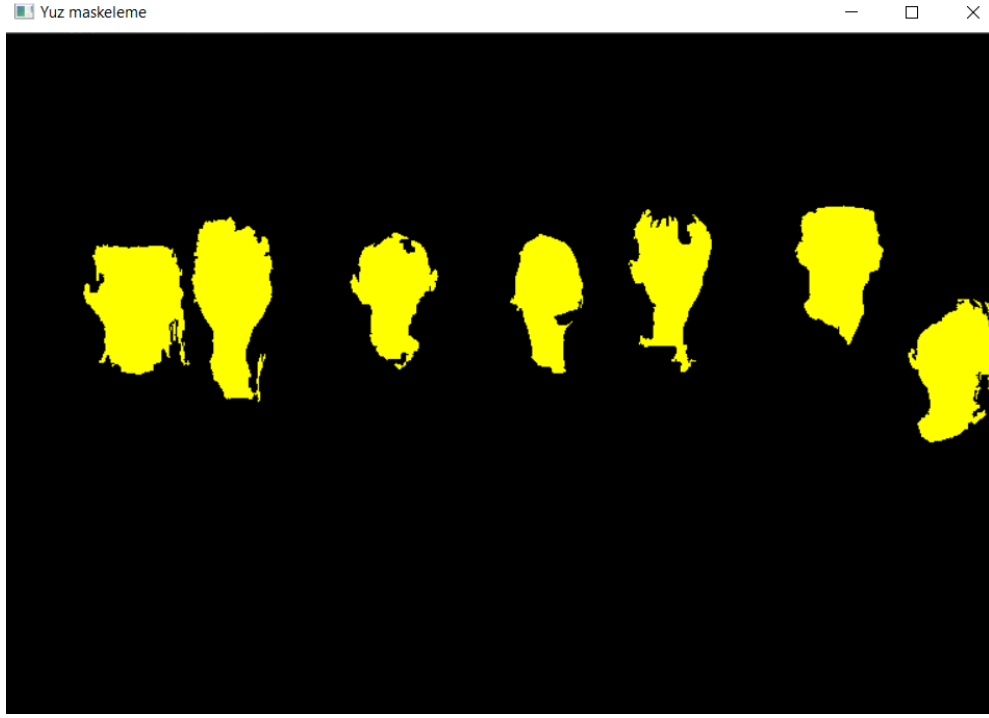
KOD İŞLEM VE ÇIKTILARI



Şekil 1 face.png uzantılı fotoğraf



Şekil 2 Renk Tabanlı Ten Rengi Bölütlemesi



Şekil 3 Kol ve Yüzün Ayrıştırılması

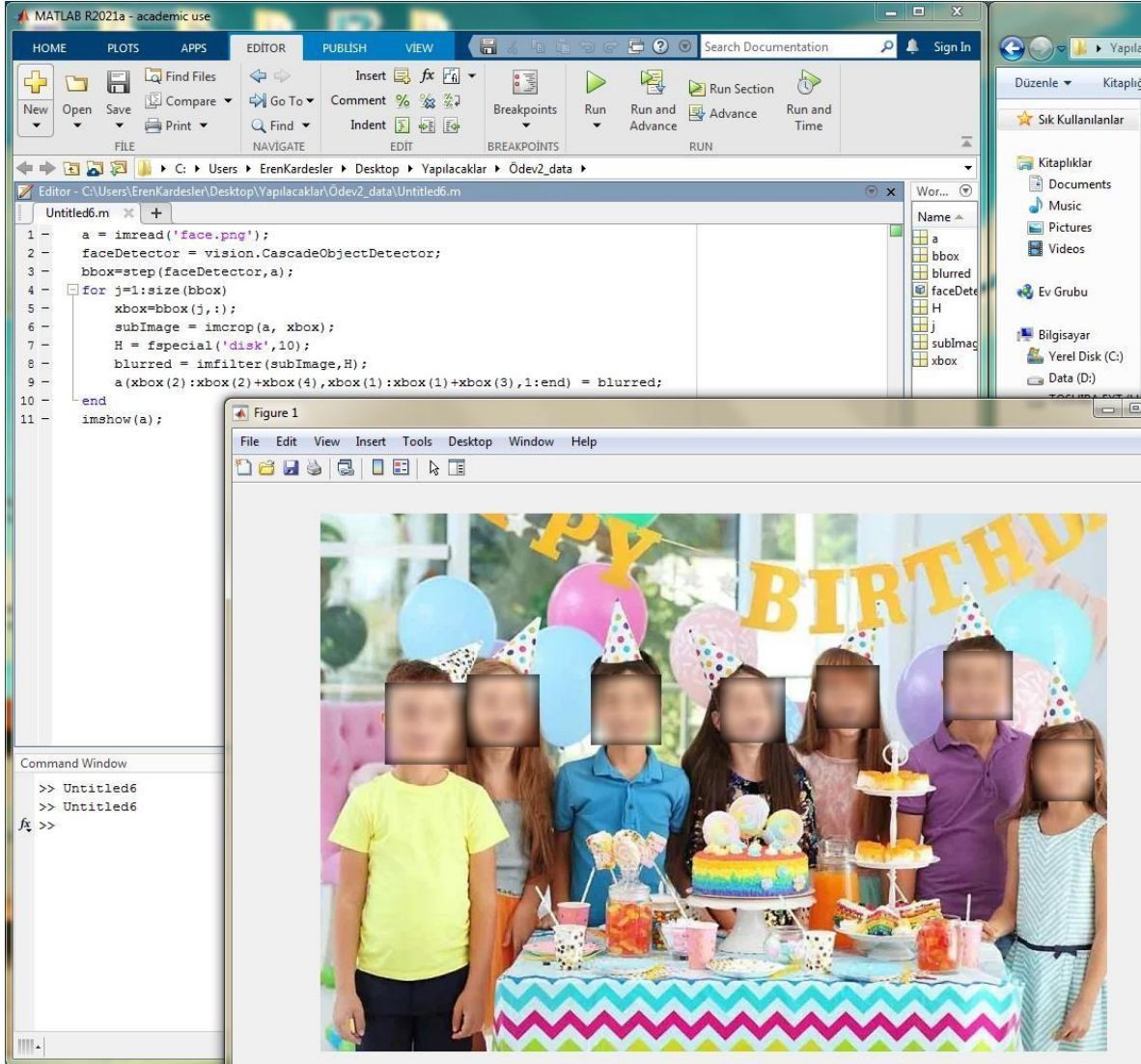
KOD ÇIKTISI :

(GÖRÜNTÜ SON HALİ):



Şekil 4 PHYTON KOD ÇIKTISI(SON HALİ)

MATLAB İLE FARKLI BİR YÖNTEM DENENMESİ VE MATLAB KODU



```
a = imread('face.png');
faceDetector = vision.CascadeObjectDetector;
bbox=step(faceDetector,a);
for j=1:size(bbox)
    xbox=bbox(j,:);
    subImage = imcrop(a, xbox);
    H = fspecial('disk',10);
    blurred = imfilter(subImage,H);
    a(xbox(2):xbox(2)+xbox(4),xbox(1):xbox(1)+xbox(3),1:end) = blurred;
end
imshow(a);
```

Soru-1 :

```
% Soru-1 Gürültü Temizleme
ResimOrjinal = imread('s1.jpg') ;
hBlurringFiltresi = fspecial('disk',10);
blurred = imfilter(ResimOrjinal,hBlurringFiltresi);
figure;
subplot(1,2,1);
imshow(ResimOrjinal);
title('Resmin Orjinal Hali');
subplot(1,2,2);
imshow(blurred);
title('Gürültü Temizlenmiş Hali');
```

