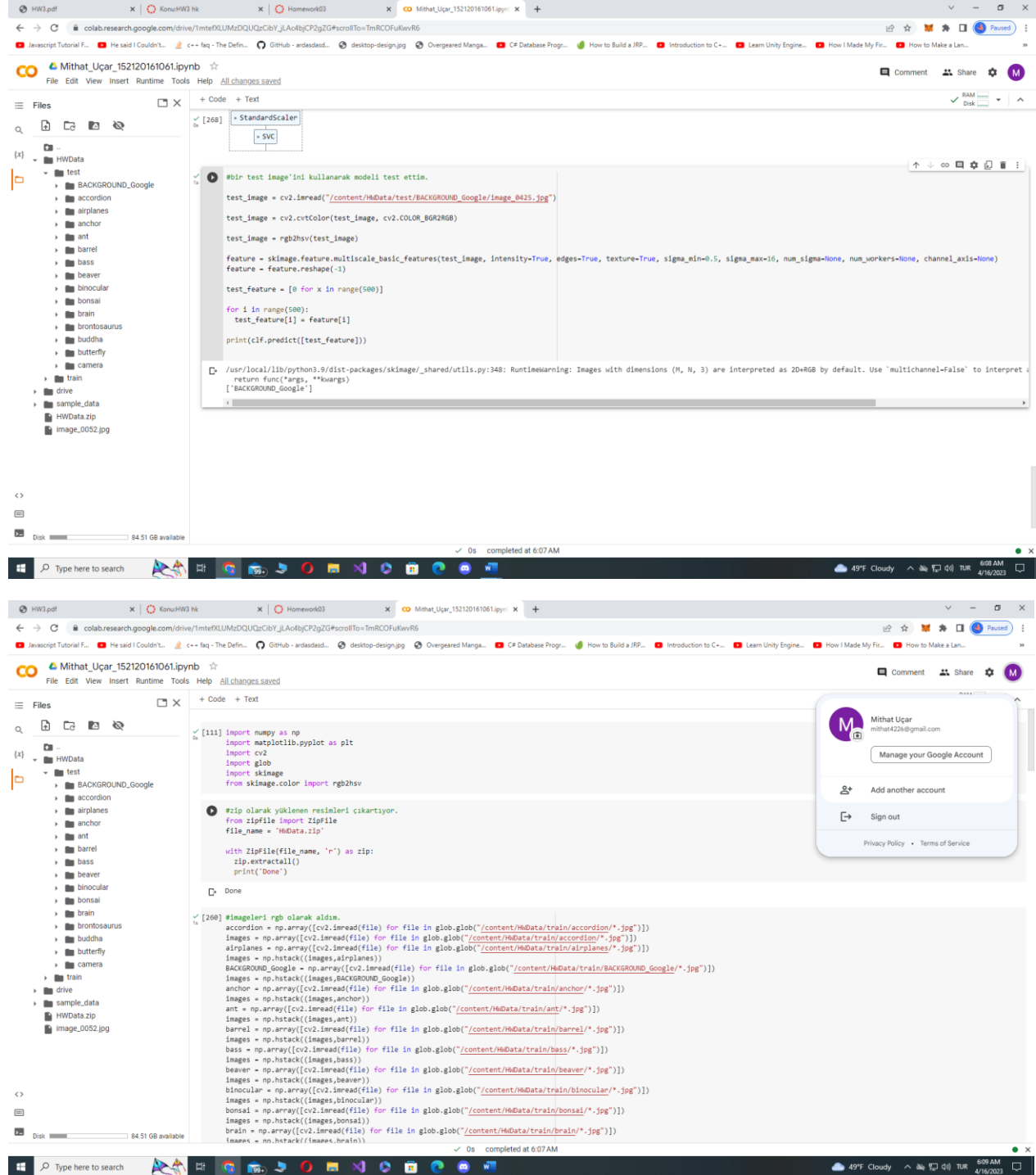


Pattern Recognition HW3

Programın çıktısı test dosyasındaki resimlerden birinin öz niteliklerinin olduğu 1x500 vektörü svc modeline göndererek resmin ait olduğu sınıfı print etmektedir.



```
#bir test image'ini kullanarak modeli test ettim.

test_image = cv2.imread("/content/HwData/test/BACKGROUND_Google/image_0425.jpg")

test_image = cv2.cvtColor(test_image, cv2.COLOR_BGR2RGB)

test_image = rgb2hsv(test_image)

feature = skimage.feature.multiscale_basic_features(test_image, intensity=True, edges=True, texture=True, sigma_min=0.5, sigma_max=16, num_sigma=None, num_workers=None, channel_axis=None)
feature = feature.reshape(-1)

test_feature = [0 for x in range(500)]

for i in range(500):
    test_feature[i] = feature[i]

print(clf.predict([test_feature]))
```

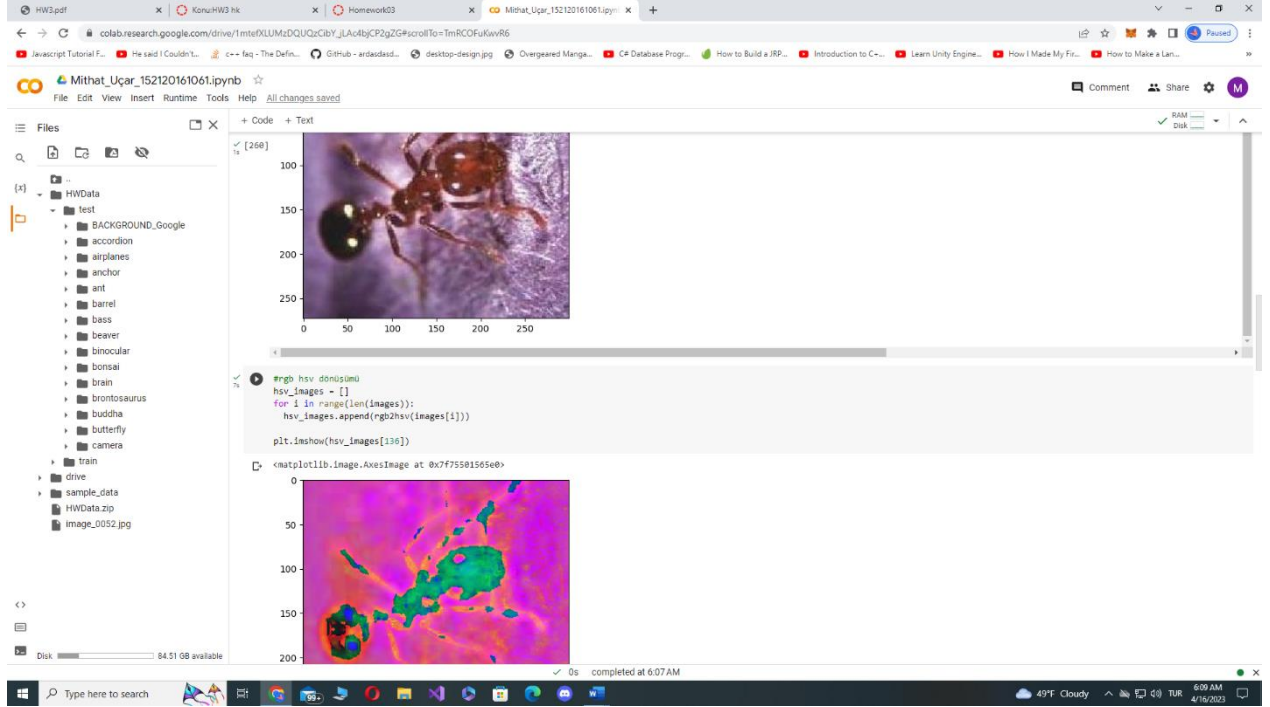
```
import numpy as np
import matplotlib.pyplot as plt
import cv2
import glob
import skimage
from skimage.color import rgb2hsv

#zip olarak yuklenen resimleri cikartiyor.
from zipfile import ZipFile
file_name = 'HwData.zip'

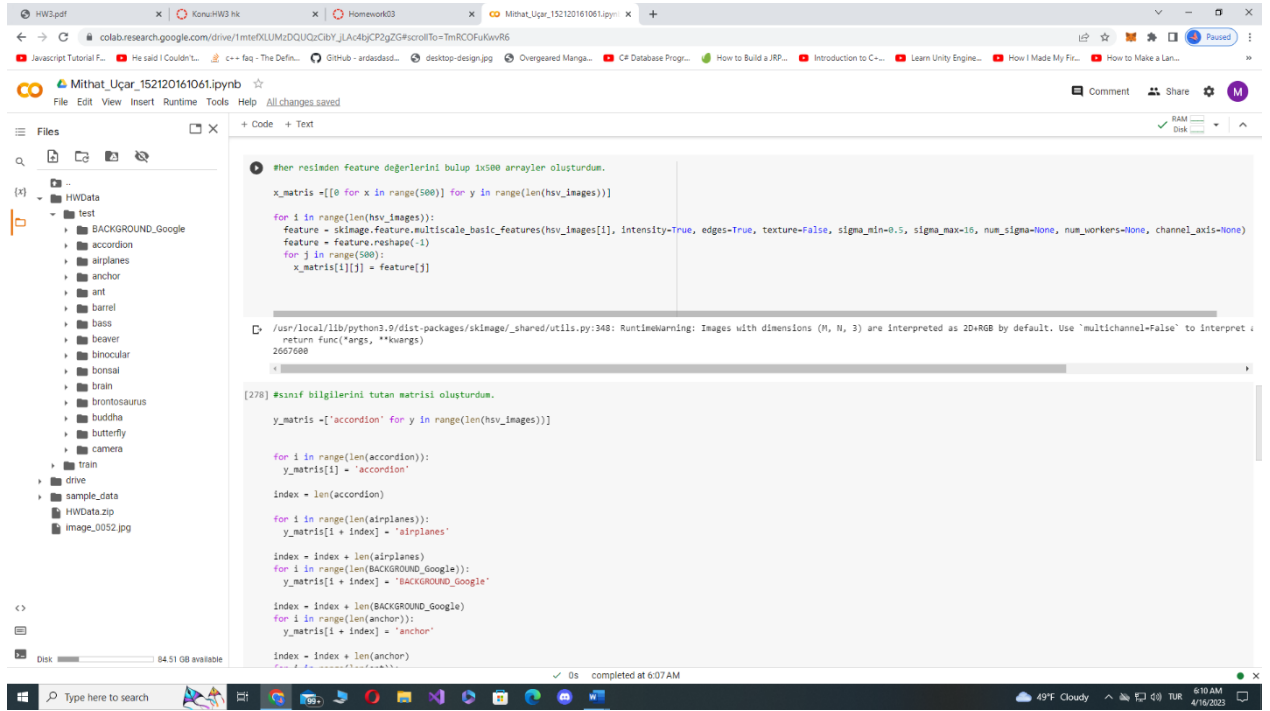
with ZipFile(file_name, 'r') as zip:
    zip.extractall()
    print('Done')
```

```
#imgeleri rgb olarak aldim.
accordion = np.array([cv2.imread(file) for file in glob.glob("/content/HwData/train/accordion/*.jpg")])
images = np.array([cv2.imread(file) for file in glob.glob("/content/HwData/train/accordion/*.jpg")])
images = np.hstack((images,airplanes))
BACKGROUND_Google = np.array([cv2.imread(file) for file in glob.glob("/content/HwData/train/BACKGROUND_Google/*.jpg")])
images = np.hstack((images,BACKGROUND_Google))
anchor = np.array([cv2.imread(file) for file in glob.glob("/content/HwData/train/anchor/*.jpg")])
images = np.hstack((images,anchor))
ant = np.array([cv2.imread(file) for file in glob.glob("/content/HwData/train/ant/*.jpg")])
images = np.hstack((images,ant))
barrel = np.array([cv2.imread(file) for file in glob.glob("/content/HwData/train/barrel/*.jpg")])
images = np.hstack((images,barrel))
bass = np.array([cv2.imread(file) for file in glob.glob("/content/HwData/train/bass/*.jpg")])
images = np.hstack((images,bass))
beaver = np.array([cv2.imread(file) for file in glob.glob("/content/HwData/train/beaver/*.jpg")])
images = np.hstack((images,beaver))
binocular = np.array([cv2.imread(file) for file in glob.glob("/content/HwData/train/binocular/*.jpg")])
images = np.hstack((images,binocular))
bonsai = np.array([cv2.imread(file) for file in glob.glob("/content/HwData/train/bonsai/*.jpg")])
images = np.hstack((images,bonsai))
brain = np.array([cv2.imread(file) for file in glob.glob("/content/HwData/train/brain/*.jpg")])
images = np.hstack((images,brain))
```

Resimleri colab'a zip olarak yükledim, zip dosyasını kod ile çıkartarak colab'da kullanabildim. Resimleri rgb olarak images arrayine atadım daha sonra sınıf bilgilerini kullanmak için değişkenlerde sakladım.



Resimleri rgbden hsvye dönüştürdüm.



```
#her resaden feature degerlerini bulup 1x500 arrayler olusturdum.
x_matris = [[0 for x in range(500)] for y in range(len(hsv_images))]

for i in range(len(hsv_images)):
    feature = skimage.feature.multiscale_basic_features(hsv_images[i], intensity=True, edges=True, texture=False, sigma_min=0.5, sigma_max=16, num_sigma=None, num_workers=None, channel_axis=None)
    feature = feature.reshape(-1)
    for j in range(500):
        x_matris[i][j] = feature[j]

[278] #sınıf bilgilerini tutan matrisi olusturdum.
y_matris = ['accordion' for y in range(len(hsv_images))]

for i in range(len(accordion)):
    y_matris[i] = 'accordion'

index = len(accordion)

for i in range(len(airplanes)):
    y_matris[i + index] = 'airplanes'

index = index + len(airplanes)

for i in range(len(BACKGROUND_Google)):
    y_matris[i + index] = 'BACKGROUND_Google'

index = index + len(BACKGROUND_Google)

for i in range(len(anchor)):
    y_matris[i + index] = 'anchor'

index = index + len(anchor)
```

Resimlerin featurelarını `skimage.feature.multiscale_basic_features` ile bulup feature matrisini 1 boyutlu arraye reshape ile dönüştürdüm. İlk 500 elemanını alarak `x_matris`ini oluşturdum. Bu matris ile sınıf bilgilerini tutan `y_matris` kullanarak `svc` modelini oluşturdum.