

NameSurname=SelimCanYazar

StudentID= 152120191023

## Pattern Recognition HW4

```
import zipfile
import os
import numpy as np
import cv2
from skimage.color import rgb2lab
from skimage.feature import hog
from sklearn import svm
from sklearn.metrics import confusion_matrix, accuracy_score
import matplotlib.pyplot as plt

# HWDData.zip dosyasını çıkartma
zip_path = 'HWDData.zip'
extract_path = 'HWDData'
with zipfile.ZipFile(zip_path, 'r') as zip_ref:
    zip_ref.extractall(extract_path)
```

Yukarıdaki kütüphaneler, destek vektör makineleri (SVM) ile dosya işlemleri, görüntü işleme, özellik çıkarma, sınıflandırma ve görselleştirme gibi işlemler için kullanılır. Daha sonra HWDData.zip adlı sıkıştırılmış veri kümesi dosyasının yolunu tutan zip\_path adlı bir değişken oluşturulur. Extract\_path değişkeni kullanılarak bu dosyanın içeriğinin çıkarılacağı klasörün adı HWDData olarak korunur. Zip dosyası, HWDData.zip dosyasını okuma modunda açmak için kalan kodda kullanılır.

ZipFile bileşeni zip\_ref.extractall(extract\_path) komutu ile ZIP dosyasındaki tüm dosyalar HWDData isimli klasöre çıkartılır. Bu işlemler sonucunda HWDData.zip dosyasındaki tüm veriler HWDData klasörüne çıkarılacak ve bu klasör sonraki adımlarda veri işleme ve analiz için kullanılacaktır.

```

# Veri yollarını belirleme
train_dir = os.path.join(extract_path, 'HWData/train')
test_dir = os.path.join(extract_path, 'HWData/test')

# Resimleri ve etiketleri yükleme ve özellik çıkarma
def load_data_and_extract_features(data_dir):
    features = []
    labels = []
    for class_dir in os.listdir(data_dir):
        class_path = os.path.join(data_dir, class_dir)
        if os.path.isdir(class_path):
            for img_name in os.listdir(class_path):
                img_path = os.path.join(class_path, img_name)
                img = cv2.imread(img_path)
                img = cv2.resize(img, (224, 224))
                img_lab = rgb2lab(img)
                img_hog, _ = hog(img_lab, pixels_per_cell=(8, 8),
                                cells_per_block=(2, 2), visualize=True, multichannel=True)
                features.append(img_hog)
                labels.append(class_dir)
    return np.array(features), np.array(labels)

# Eğitim ve test verilerini yükleyip özellik çıkarma
X_train, y_train = load_data_and_extract_features(train_dir)
X_test, y_test = load_data_and_extract_features(test_dir)

# SVM ile eğitim ve test
model = svm.SVC(kernel='linear', decision_function_shape='ovr')
model.fit(X_train, y_train)

```

Yukarıdaki kodda veri yolları belirlenir. Eğitim verilerinin yolu train\_dir olarak, test verilerinin yolu ise test\_dir olarak ayarlanmıştır. Daha sonra, görüntüleri yükleyen ve özellikleri çıkaran bir işlev olan load\_data\_and\_extract\_features tanımlanır. Bu işlev, verilen veri dizinindeki her sınıfa ait görüntü dosyalarını okur ve her görüntünün HOG özelliklerini çıkarır. Daha sonra görüntüler okunup 224x224 piksele yeniden boyutlandırıldıktan sonra LAB renk uzayına dönüştürülerek HOG özellikleri çıkartılır. Bu özellikler özellikler listesine eklenirken görüntünün sınıf etiketi de etiketler listesine eklenir. Bu fonksiyon aracılığıyla eğitim ve test verileri yüklenip özellikleri çıkarıldıktan sonra, X\_train ve y\_train değişkenleri eğitim verilerini ve etiketlerini, X\_test ve y\_test değişkenleri ise test verilerini ve etiketlerini tutar. Son olarak X\_train ve y\_train verileri kullanılarak bir SVM modeli oluşturulur ve eğitilir. Model, birden fazla sınıfın olduğu durumlarda doğrusal bir çekirdek ve "bire karşı dinlenme" karar fonksiyonunu kullanır.

```
# Tahminler
y_pred = model.predict(X_test)

# Karışıklık matrisi ve doğruluk hesaplama
conf_matrix = confusion_matrix(y_test, y_pred)
accuracy = accuracy_score(y_test, y_pred)

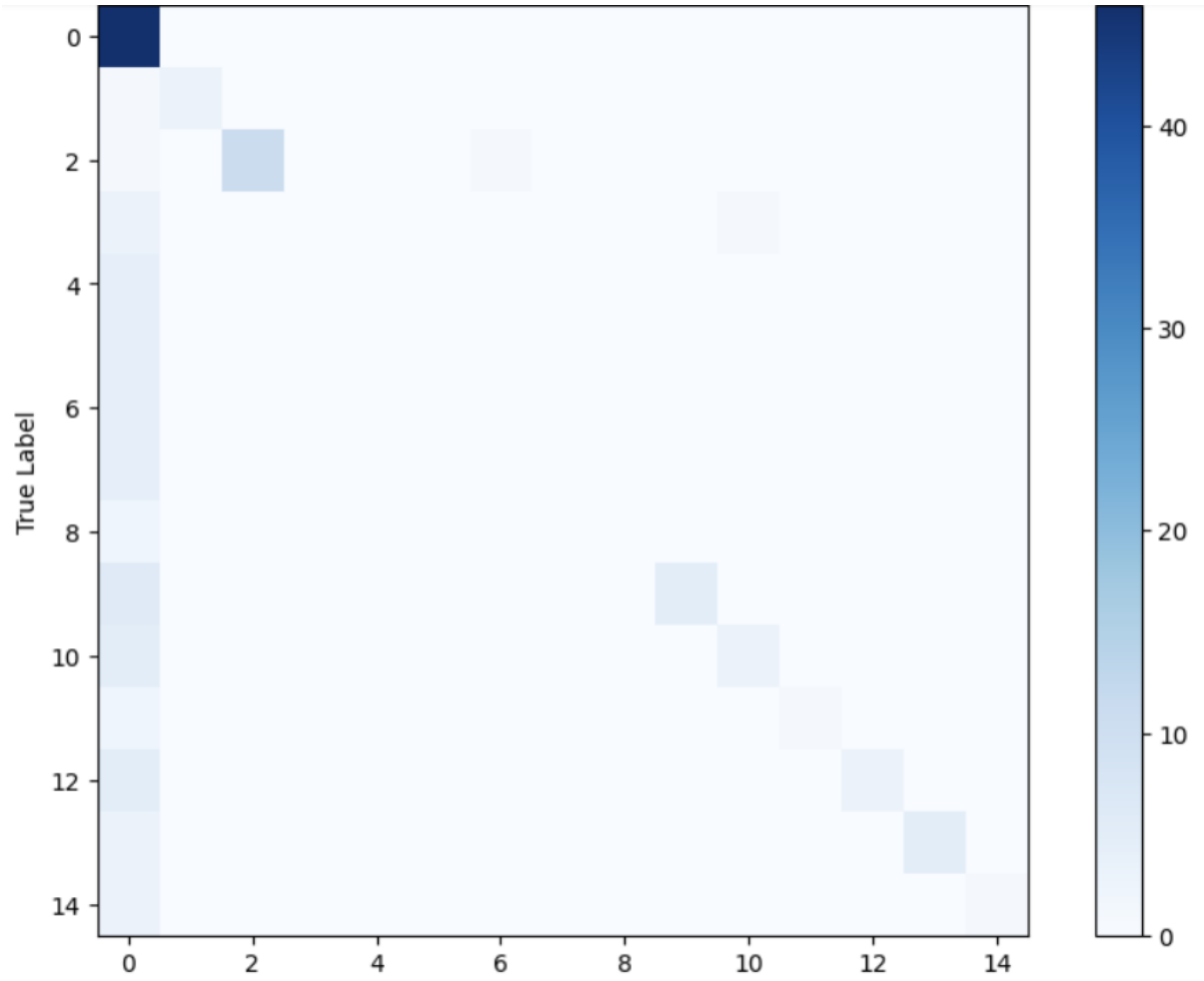
print('Confusion Matrix:\n', conf_matrix)
print('Accuracy:', accuracy)

# Karışıklık matrisini görselleştirme
plt.figure(figsize=(10, 7))
plt.imshow(conf_matrix, interpolation='nearest', cmap=plt.cm.Blues)
plt.title('Confusion Matrix')
plt.colorbar()
plt.xlabel('Predicted Label')
plt.ylabel('True Label')
plt.show()
```

Test verileri üzerinden tahminler yaparak ve performansı değerlendirerek devam ediyoruz. Öncelikle model.predict fonksiyonu aracılığıyla X\_test test verisi üzerinde y\_pred tahminleri yaptık. Ek olarak, kafa karışıklığı\_matrisi ve doğruluk\_skoru, tahmin edilen y\_pred etiketlerinin gerçek y\_test etiketleriyle karşılaştırılması yoluyla hesaplanır. Karışıklık matrisi, her sınıf için doğru ve yanlış sayıların sayısını gösterir. Bu matris, kafa karışıklığı\_matrix işlevi kullanılarak elde edilir. Doğruluk, tüm doğru sonuçların toplam özelliklere oranı olarak hesaplanır.

Hesaplanan karışıklık matrisi ve doğruluk değeri ekrana yazdırılır. Daha sonra matplotlib kurulur ve karmaşık matris görselleştirilir. Bu görselleştirme, karışıklık matrisini bir ısı haritası biçiminde göstermektedir. plt.imshow işlevi, konfigürasyon matrisini görselleştirmek için kullanılır ve plt.colorbar işleviyle bir renk çubuğuna eklenir. Grafiğin güvenilirliğini ve eksenel etiketlerini belirlemek için plt.title, plt.xlabel ve plt.ylabel kapsamı kullanılır. Son olarak karışık matris plt.show fonksiyonu ile görselleştirilmiş olarak gösterilmektedir.

```
Confusion Matrix:
[[46  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 1  3  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 1  0 11  0  0  0  1  0  0  0  0  0  0  0  0]
 [ 3  0  0  0  0  0  0  0  0  0  1  0  0  0  0]
 [ 4  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 4  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 4  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 4  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 2  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 6  0  0  0  0  0  0  0  0  5  0  0  0  0  0]
 [ 5  0  0  0  0  0  0  0  0  0  3  0  0  0  0]
 [ 2  0  0  0  0  0  0  0  0  0  0  1  0  0  0]
 [ 5  0  0  0  0  0  0  0  0  0  0  0  3  0  0]
 [ 3  0  0  0  0  0  0  0  0  0  0  0  0  5  0]
 [ 3  0  0  0  0  0  0  0  0  0  0  0  0  0  1]]
Accuracy: 0.6141732283464567
```



Son olarak accuracy değeri , confusion matrix ve confusion matrix grafiği yukarıdaki gibi olacak şekilde sonuç bulunmuştur.