

Sample HW Report

a-) Download or crop the output of your program. Give comments on output of your homework, i.e., explain the graph or explain numerical values in terms of performance score.

Uygulamada resim sınıflandırma için KNN algoritması kullandım. Uygulamanın doğru çalışması için Train ve Test klasörleri derleyiciye eklenmelidir. Koddaki folders listesi, train verilerinin bulunduğu klasörlerin yollarını içerir.

Test edilmek istenilen resmin yolu “test_image_path” değişkenine kopyalanmalıdır.

process_image(img_path) fonksiyonu, resim dosyasının yolu parametre olarak gönderildiğinde, resmi okuma- yeniden boyutlandırma- grayscale’e çevirme ve türevini alma-normalize etme-vektörize etme işlemlerini gerçekleştirir. Resim işleme sırasında herhangi bir aşamada bir hata oluşursa, hata mesajı yazdırılması için try-except blokları ekledim.

load_data(data_folders) fonksiyonu, train verilerini yükler. Her sınıf için bir klasör yolu listesi alır, her resmi işlemek için process_image fonksiyonunu çağırır ve train veri setini (X_data) ve label’ları (Y_data) döndürür.

KNN algoritmasının fonksiyonları:

calculate_distance(a, b): İki vektör arasındaki Öklidyen mesafesini hesaplar.

k_nearest_neighbors(x_data, y_data, test_sample, k=5): Verilen test resmi için en yakın k komşuyu bulur ve en yaygın label’ı tahmin eder.

Uygulama, eğitim verilerini yükledikten sonra (load_data fonksiyonu), test resmini işler(process) ve KNN algoritması kullanarak test resminin etiketini tahmin eder(1-2-3). Tahmin edilen label ise sonucu-kategoriye yazdırmak için kullanılır.

b-) UYGULAMANIN (KOD VE OUTPUT) EKRAN GÖRÜNTÜLERİ:

The screenshot displays a Google Colab notebook interface. The left sidebar shows a file explorer with a directory structure: 'sample_data' containing 'test' and 'train' folders. The 'test' folder contains 'Motorbikes', 'cellphone', and 'flamingo' subfolders. The 'train' folder also contains 'Motorbikes', 'cellphone', and 'flamingo' subfolders. The main area shows a Python script for image processing. The script imports 'cv2', 'numpy', 'os', 'scipy', and 'collections'. It defines a 'process_image' function that takes an image path, reads the image, resizes it to 128x128, and calculates its Sobel derivatives. It then flattens the image and normalizes it. The script also defines a 'load_data' function that loads data from the 'test' and 'train' folders. The script is executed, and the output is displayed in the right sidebar. The output shows the results of the 'load_data' function, including the number of images loaded for each class and the total number of images.

```
import cv2
import numpy as np
import os
from scipy import ndimage
from collections import Counter

def process_image(img_path):
    try:
        # Resmi cv2 ile okur
        img = cv2.imread(img_path)
        if img is None:
            raise Exception(f"!Hata: Resim okunamadı - {img_path}!")

        # Resmi 128x128 olarak boyutlandırır ve grayscale'e çevirir
        gray_img = cv2.resize(cv2.cvtColor(img, cv2.COLOR_BGR2GRAY), (128, 128))

        # Resmin türevini hesaplar
        img_derivative = np.abs(ndimage.sobel(gray_img, axis=0, mode='constant')) + np.abs(ndimage.sobel(gray_img, axis=1, mode='constant'))

        # Flatten, normalize et ve float16'ya çevir
        img_vector = (img_derivative - np.min(img_derivative)) / (np.max(img_derivative) - np.min(img_derivative))
        img_vector = img_vector.flatten().astype(np.float16)
    except Exception as processing_error:
        print(f"!Resmi işlerken hata oluştu: {processing_error}!")
        return None

    return img_vector

def load_data(data_folders):
    x_data = []
    y_data = []

    for label, folder_path in enumerate(data_folders, start=1):
        try:
            if os.path.exists(folder_path):
                for filename in os.listdir(folder_path):
                    image_path = os.path.join(folder_path, filename)

                    vector = process_image(image_path)
                    if vector is not None:
```

Google Hesabı
Ayşe Ayhan
ayse.ayhan6777@gmail.com

Merhaba Ayşe Ayhan!

[Google Hesabınızı yönetin](#)

[Hesap ekle](#) [Oturumu kapat](#)

[Gizlilik Politikası](#) [Hizmet Şartları](#)

80.64 GB kullanılabilir

22:19 9.11.2023

colab.research.google.com/drive/1u1qgEh7EBME6jBZewYmftWUul8q5Ur28#scrollTo=_j22eBXDW8lf

Uygulamalar Google Google Google Yeni Sekme Reso Coder - Flutter... Unlimited UI Kits, Ic... Güvenli Yazılım Geli... Gruplanm Canvas'ta Oturum Aç Online Kurslar - Her... Flutter documentati... Learn Tüm Yer İşaretleri

AyşeAyhan_152120201058_hw2_DeepLearning.ipynb

Dosya Düzenle Göster Ekle Çalışma zamanı Araçlar Yardım Tüm değişiklikler kaydedildi

Yorum Paylaş

Google Hesabı Ayşe Ayhan ayse.ayhan6777@gmail.com

Dosyalar

- sample_data
- test
 - Motorbikes
 - cellphone
 - flamingo
 - image_0022.jpg
 - image_0024.jpg
 - image_0041.jpg
 - image_0043.jpg
 - image_0047.jpg
 - image_0050.jpg
 - image_0051.jpg
 - image_0059.jpg
- train
 - Motorbikes
 - cellphone
 - flamingo

0 sn.

```
vector = process_image(image_path)
if vector is not None:
    x_data.append(vector)
    y_data.append(label)
except Exception as loading_error:
    print(f"! {folder_path} klasöründen veri alınırken hata oluştu: {loading_error} !")

return np.array(x_data), np.array(y_data)

# Train datası içeren üç klasörün pathleri
folders = [
    "/content/train/cellphone",
    "/content/train/flamingo",
    "/content/train/Motorbikes"
]

try:
    x_data, y_data = load_data(folders)

    # KNN Fonksiyonu
    def calculate_distance(a, b):
        return np.linalg.norm(a - b)

    def k_nearest_neighbors(x_data, y_data, test_sample, k=5):
        distances = [np.linalg.norm(test_sample - x) for x in x_data] # Öklidyen mesafeleri hesaplar
        nearest_indices = np.argsort(distances)[:k]

        nearest_labels = [y_data[indicate] for indicate in nearest_indices]

        label_counts = Counter(nearest_labels)
        most_common_label = label_counts.most_common(1)[0][0]

        # print(f"Classification results (numeric): {label_counts}") #cıkıtları kontrol etmek için
        # print(f"Distances: {distances}") #distanceları kontrol etmek için

        return most_common_label

    # Test part
    test_image_path = "/content/test/flamingo/image_0043.jpg" # Test resminin pathi, test edilmek istenen resim yolu kopyalanmalı
    test_sample_vector = process_image(test_image_path)
    test_sample_vector = (test_sample_vector - np.min(test_sample_vector)) / (np.max(test_sample_vector) - np.min(test_sample_vector))
```

0 sn. tamamlanma zamanı: 22:19

https://accounts.google.com/SignInOptions?hl=tr&continue=https://colab.research.google.com/drive/1u1qgEh7EBM...

Ara

9°C 22:19 9.11.2023

colab.research.google.com/drive/1u1qgEh7EBME6jBZewYmftWUul8q5Ur28#scrollTo=_j22eBXDW8lf

AyşeAyhan_152120201058_hw2_DeepLearning.ipynb

Dosya Düzenle Göster Ekle Çalışma zamanı Araçlar Yardım Tüm değişiklikler kaydedildi

Dosyalar

- sample_data
- test
 - Motorbikes
 - cellphone
 - flamingo
 - image_0022.jpg
 - image_0024.jpg
 - image_0041.jpg
 - image_0043.jpg
 - image_0047.jpg
 - image_0050.jpg
 - image_0051.jpg
 - image_0059.jpg
- train
 - Motorbikes
 - cellphone
 - flamingo

```
return most_common_label

# Test part
test_image_path = "/content/test/flamingo/image_0043.jpg" # Test resminin pathi, test edilmek istenen resim yolu kopyalanmalı
test_sample_vector = process_image(test_image_path)
test_sample_vector = (test_sample_vector - np.min(test_sample_vector)) / (np.max(test_sample_vector) - np.min(test_sample_vector))
test_sample_vector = test_sample_vector.flatten().astype(np.float16)

try:
    predicted_label = k_nearest_neighbors(x_data, y_data, test_sample_vector)

    # Predict edilen etiketi yazdırır
    if predicted_label == 1:
        print("Tahmin edilen kategori: cellphone")
    elif predicted_label == 2:
        print("Tahmin edilen kategori: flamingo")
    elif predicted_label == 3:
        print("Tahmin edilen kategori: Motorbike")
    else:
        print("Hata")
except Exception as knn_error:
    print(f"KNN tahmininde hata oluştu: {knn_error}")

except Exception as data_loading_error:
    print(f"!Veri yüklenirken hata oluştu: {data_loading_error}!")
```

Tahmin edilen kategori: flamingo

0 sn. tamamlanma zamanı: 22:19

9°C 22:19 9.11.2023

Browser tabs: AyşeAyhan_152120201058_hw2, python - Minimize distance usi...

Address bar: colab.research.google.com/drive/1u1qgEh7EBME6jBZewYmftWUul8q5Ur28#scrollTo=_j22eBXDW8lf

Navigation bar: Uygulamalar, Google, Google, Google, Yeni Sekme, Reso Coder - Flutter..., Unlimited UI Kits, Ic..., Güvenli Yazılım Geli..., Gruplanm, Canvas'ta Oturum Aç, Online Kurslar - Her..., Flutter documentati..., Learn

File name: AyşeAyhan_152120201058_hw2_DeepLearning.ipynb

Actions: Dosya, Düzenle, Göster, Ekle, Çalışma zamanı, Araçlar, Yardım, Tüm değişiklikler kaydedildi

Left sidebar: Dosyalar

- image_uo000.jpg
- image_0692.jpg
- image_0696.jpg
- image_0708.jpg
- image_0714.jpg
- image_0716.jpg
- image_0727.jpg
- image_0734.jpg
- image_0735.jpg
- image_0750.jpg
- image_0756.jpg
- image_0758.jpg
- image_0770.jpg
- image_0777.jpg
- image_0780.jpg
- image_0788.jpg
- cellphone
 - image_0024.jpg
 - image_0035.jpg
 - image_0037.jpg
 - image_0038.jpg
 - image_0040.jpg
 - image_0046.jpg
 - image_0047.jpg
- flamingo
- train
 - Motorbikes
 - cellphone
 - flamingo

Disk: 80.64 GB kullanılabilir

Code editor:

```
return most_common_label

# Test part
test_image_path = "/content/test/cellphone/image_0046.jpg" # Test resminin pathi, test edilmek istenen resim yolu kopyalanmalı
test_sample_vector = process_image(test_image_path)
test_sample_vector = (test_sample_vector - np.min(test_sample_vector)) / (np.max(test_sample_vector) - np.min(test_sample_vector))
test_sample_vector = test_sample_vector.flatten().astype(np.float16)

try:
    predicted_label = k_nearest_neighbors(x_data, y_data, test_sample_vector)

    # Predict edilen etiketi yazdırır
    if predicted_label == 1:
        print("Tahmin edilen kategori: cellphone")
    elif predicted_label == 2:
        print("Tahmin edilen kategori: flamingo")
    elif predicted_label == 3:
        print("Tahmin edilen kategori: Motorbike")
    else:
        print("Hata")
except Exception as knn_error:
    print(f"KNN tahmininde hata oluştu: {knn_error}")
except Exception as data_loading_error:
    print(f"!Veri yüklenirken hata oluştu: {data_loading_error}!")
```

Output: Tahmin edilen kategori: cellphone

Bottom bar: https://accounts.google.com/SignInOptions?hl=tr&continue=https://colab.research.google.com/drive/1u1qgEh7EBME6jBZewYmftWUul8q5Ur28#ec=GBRAqQM tamamlanma zamanı: 22:21

System tray: 9°C, TUR, 9.11.2023

colab.research.google.com/drive/1u1qgEh7EBME6jBZewYmtfWUul8q5Ur28#scrollTo=_j22eBXDW8If

Ayşe Ayhan_152120201058_hw2_DeepLearning.ipynb

Dosya Düzenle Göster Ekle Çalışma zamanı Araçlar Yardım

Yorum Paylaş

Google Hesabı
Ayşe Ayhan
ayse.ayhan6777@gmail.com

Dosyalar

- image_0569.jpg
- image_0583.jpg
- image_0585.jpg
- image_0586.jpg
- image_0592.jpg
- image_0600.jpg
- image_0609.jpg
- image_0618.jpg
- image_0620.jpg
- image_0626.jpg
- image_0633.jpg
- image_0634.jpg
- image_0635.jpg
- image_0651.jpg
- image_0654.jpg
- image_0662.jpg
- image_0669.jpg
- image_0673.jpg
- image_0680.jpg
- image_0685.jpg
- image_0692.jpg
- image_0696.jpg
- image_0708.jpg
- image_0714.jpg
- image_0716.jpg
- image_0727.jpg
- image_0734.jpg
- image_0735.jpg

```
return most_common_label

# Test part
test_image_path = "/content/test/Motorbikes/image_0696.jpg" # Test resminin pathi, test edilmek istenen resim yolu kopyalanmalı
test_sample_vector = process_image(test_image_path)
test_sample_vector = (test_sample_vector - np.min(test_sample_vector)) / (np.max(test_sample_vector) - np.min(test_sample_vector))
test_sample_vector = test_sample_vector.flatten().astype(np.float16)

try:
    predicted_label = k_nearest_neighbors(X_data, y_data, test_sample_vector)

    # Predict edilen etiketi yazdırır
    if predicted_label == 1:
        print("Tahmin edilen kategori: cellphone")
    elif predicted_label == 2:
        print("Tahmin edilen kategori: flamingo")
    elif predicted_label == 3:
        print("Tahmin edilen kategori: Motorbike")
    else:
        print("Hata")
except Exception as knn_error:
    print(f"KNN tahmininde hata oluştu: {knn_error}")

except Exception as data_loading_error:
    print(f"Veri yüklenirken hata oluştu: {data_loading_error}!")
```

Tahmin edilen kategori: Motorbike

0 sn. tamamlama zamanı: 22:21

9°C 22:21 9.11.2023