

Aygaz Görüntü İşleme Bootcamp Projesi
Aralık 2024
Diclehan ve Oğuzhan Ulucan

- GitHub reponuz içerisinde, projenize dair özet anlatımları içeren bir **README.md** dosyanız bulunmalı.
- Veri seti özelliklerinize, hangi kütüphanelerle ne tip analizler yaptığınıza kısaca burada **değirmelisiniz**.
- Notebook dosyanızın içerisinde **markdown** formatında hücreleriniz **bulunmalı**. Bu hücrelerde, o an yapıyor olduğunuz işlemten detaylı bahsetmelisiniz. Komutu çalıştırdığınızda hücrenizin verdiği çıktıyı yorumlamanızı, analizlerinizi ve grafiklerinizi anlatmanızı bekliyoruz.
- **Bu kriter, bundan sonraki tüm adımları kapsayan, proje inceleme sürecinde aradığımız ve dikkat ettiğimiz bir kriterdir. Projeniz, yalnızca kod hücreleri ve çıktılarından oluşmamalıdır.**
- Sadece notebook oluşturmanız yeterli değildir. Notebookunuzu çalıştırmış ve **hücrelerin çıktılarını da elde etmiş olmanız gerekiyor. Notebookunuzun uçtan uca hatasız bir biçimde çalışmış olması notebook için ilk önemli kriterimiz.**

Bu projedeki amaç, CNN modellerinin temelini pekiştirmektir. Bunun için bir CNN modeli kurup bu model ile çeşitli hayvan türlerini sınıflandıracacağız. Örneğin, "panda" ve "köpek" gibi iki sınıfı CNN ile ayırt edeceğiz. Bizim tek bir bakışta yapabildiğimiz bu ayrımı, CNN modellerinin de "basitçe" yapabilmesi için hangi adımları atmamız gerektiğini kavrayacağız. Modeli ne kadar derinleştirmemiz gerektiğine ise sizler karar vereceksiniz. Bir modeli "iyi" yapan onun derinliği midir yoksa az parametre kullanarak verilen görevi yerine getirmesi midir?

Proje, dört adımdan oluşmaktadır. İlk aşamada bir CNN modeli oluşturulacak, bu model eğitilecek ve test edilecektir.

Ardından, öğrenmeye dayalı modellerin ne kadar hassas olduğunu kavramanız için basit bir adım uygulayacağız. Test resimlerini farklı ışıklar ile manipüle edecek ve geliştirdiğimiz model bu resimler üzerinde test edilecektir. Test skorları düştüyse, bu düşüşün neden kaynaklandığını konuşacağız.

Eğer skorlar düştüyse, bu düşüş nasıl minimize edilebilir? Düşmediyse, renk sabitliği algoritması uygulanması halinde başarı artar mı? Derste de kısaca konuştuğumuz, resim manipülasyon tekniklerinden biri sayılabilecek ve kendi başına bir "derya deniz" olan renk sabitliği alanından bir algoritma kullanarak bozuk resmin renklerini düzeltecek ve modeli tekrar test edeceğiz. Skorlarda artış var mı, yok mu, bunu tartışacağız.

Son olarak bu üç farklı senaryo için test başarıları karşılaştırılıp raporlanacak ve çözüm önerileri sunulacaktır. En başarılı model ve çözüm önerisini geliştiren katılımcı, projeyi sunmaya hak kazanacaktır.

Projenin detaylı adımları aşağıda verilmiştir. Herkese başarılar dileriz!

Proje Adımları:

1- Veri Setini Kullanma:

Veri Seti :

<https://drive.google.com/drive/folders/1FYvR3c99nxf0qH235gpyJW0UGPWpmr5U?usp=sharing>

- **Seenek 1:** Verisetini doğrudan indirin ve bilgisayarınızda kullanın.
- **Seenek 2:** Verisetini kendi Google Drive'ınıza kopyalayarak Colab üzerinden alıřtırın.

- JPEGImages klasörüne gidiniz. Bu dosya içerisinde 50 farklı hayvan sınıfı bulunmaktadır. Her sınıfta farklı sayıda görüntü yer almaktadır.

2- Veri Setini Hazırlama:

- Python'da veri setini istediğiniz bir kütüphane ile okuyunuz.
- Sınıflandırma işlemi yalnızca 10 sınıf için yapılacaktır. Bu sınıflar: *collie, dolphin, elephant, fox, moose, rabbit, sheep, squirrel, giant panda* ve *polar bear*. Kalan dosyaları silebilir veya seçilen dosyaları projede kullanılmak üzere yeni bir klasöre kopyalayabilirsiniz.
- Verilerin dengeli olması adına her sınıftan yalnızca ilk 650 resmi kullanınız. (10 sınıfın her biri için ilk 650 resmi tutup gerisini silen bir kod yazabilirsiniz.)
- Tüm resimleri aynı boyuta getirmeli ve normalize etmelisiniz. Modelinizin giriş katmanına göre resimleri uygun boyuta getiriniz. Eğer resim boyutu çok yüksek olursa, modelinizin parametre sayısı ve eğitim süresi artacaktır. Örnek olarak aşağıdaki kod verilebilir:

```
import cv2
img_resized = cv2.resize(input_image, image_size) # image_size yerine istediğiniz resim boyutunu yazabilirsiniz
img_normalized = img_resized / 255.0
```

- Daha sonra veriyi train (eğitim) ve test olmak üzere ikiye ayırınız. Verilerin rastgele eğitim ve test için bölünmesini sağlayınız (örneğin, %70 eğitim, %30 test). Modelinizin her alıřtırıldığında farklı resimleri eğitim ve test için kullanmaması adına, kodunuzu her zaman aynı verileri seçmesi için modifiye ediniz (bunun neden önemli olduğunu burada biraz düşünelim). Örnek olarak aşağıdaki kod kullanılabilir:

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
# X: images, y: labels anlamına gelmektedir
# random_state=42 her alıřtırmada aynı sonuçları almanızı sağlar
```

- Son olarak, modelin farklı durumlarla başa çıkma yeteneğini artırmak ve fitting'i azaltmak adına eğitim setine veri artırma (augmentation) uygulayınız. Bunun için çeşitli manipölasyon tekniklerini (örneğin, bulanıklaştırma, kenar bulma, yeniden boyutlandırma, döndürme ve gürültü ekleme) kullanabilirsiniz. Hepsini kullanmak zorunda değilsiniz.

3- CNN Modelinin Tasarlanması:

- 10 adet hayvan sınıfını birbirinden ayıracak bir CNN modeli tasarlayınız. Bunun için çeşitli katmanları (örneğin, konvolüsyon, aktivasyon, havuzlama, fully connected, dropout) göz önünde bulundurunuz.
- Uygun bir kayıp fonksiyonu (loss function) seçiniz (örneğin, categorical_crossentropy).

4- Modelin Test Edilmesi:

- Modelinizi eğittikten sonra test verisiyle test ediniz. Modeliniz sınıfları yeterli bir başarıyla ayırabiliyorsa bir sonraki adıma geçiniz. Eğer model başarısı (accuracy) çok düşükse, örneğin %50'nin altındaysa, modelinizde parametre değişikliği yapıp veya katman ekleyip/çıkararak tekrar deneyiniz. (Unutmayın, değişikliklerinizi adım adım yapın, bir anda birden fazla şeyi değiştirmeyin.)

5- Resimlerin Farklı Işıklar ile Manipüle Edilmesi:

- Test resimlerinizi “get_manipulated_images” kodu ile manipüle ediniz. Burada elde ettiğiniz resimleri bir dosyaya kaydetmek için bu fonksiyon içerisinde bulunan kaydetme işlemini dilediğiniz gibi değiştirebilirsiniz.

6- Modelin Manipüle Edilmiş Test Seti ile Denenmesi:

- Modelinize farklı ışıklar ile manipüle edilmiş test setini veriniz.
- Başarı oranını kaydediniz.

7- Manipüle Edilmiş Test Setine Renk Sabitliği Algoritması Uygulanması:

- Manipüle edilmiş test setine renk sabitliği algoritmasını uygulayınız. Bu kısım için “get_wb_images” kodunu kullanabilirsiniz (bu kod Gray World algoritmasının sonucunu vermektedir). Oradan isterseniz gerekli düzenlemeleri yapabilirsiniz.

8- Modelin Renk Sabitliği Uygulanmış Test Seti ile Denenmesi:

- Modelinize renk sabitliği uygulanmış test setini veriniz.
- Başarı oranını kaydediniz.

9- Farklı Test Setlerinin Başarılarını Karşılaştırma ve Raporlama:

- Üç farklı test setinin başarılarını karşılaştırarak raporlayınız. Eğer 8. adımdan sonra skorlar hala düşükse, nasıl bir çözüm yolu izlenebileceğini tartışınız.