

KOD RAPORU

Bir uygulama 4 kısımdan meydana gelmektedir:

- 1) Verilerin tanımlanması
- 2) İşlemlerin tanımlanması
- 3) Sürecin tanımlanması
- 4) Sonuç

1- Verilerin Tanımlanması

Uygulamada kullanılacak olan veriler; ön işlemlerden geçirilmiş resimler (.jpg formatında), bu resimleri sınıflandırmak için gerekli label (etiket) değerleri, CNN modeli için kullanılacak olan hiperparametreler(öğrenme oranı, iterasyon sayısı, train veri sayısı...) ve sonuçları göstermek için tutulan değişkenler ve dizilerdir.

2- İşlemlerin Tanımlanması

Uygulamada kullanılacak olan işlemler; parametre olarak verilen bir dizinin elemanlarının sıralarını rasgele bir şekilde değiştirmek, resimlerin ve etiket değerlerinin okunması ve kullanılacak model için hazır hale getirilmesi, CNN modelinin tasarlanması ve oluşturulması, Modelin eğitilmesidir.

3- Sürecin Tanımlanması

Süreç Tensorflow un default graf yapısının resetlenmesiyle başlar. Bu noktadan sonra dosya dizin işlemlerinde kullanılan kütüphanelerin yardımıyla, resimlerin bulunduğu klasör (ROOT_PATH) açılır. Daha sonra o klasörde bulunan tüm resim dosyaları bir diziye aktarılır. Bu aktarma işleminden sonra, tanımlanan randomize_array fonksiyonu sayesinde, bu dizi içerisindeki resimlerin sıraları random hale getirilir.

Elde edilen dizideki tüm resimler ve bu resimlerin etiket değerleri, load_data fonksiyonu sayesinde sürece dahil edilir.(x_ax – resimlerin tutulduğu dizi, y_ax – etiketlerin tutulduğu dizi)

***** fonksiyonunda RGB değerlerinin 0-1 aralığına çekilmesi ve bu değerlerin float32 veri türünde olması sağlanmıştır.

Elde edilen image ve label değerleri, aynı şekilde shuffle edileceklerdir. Daha sonra keras kütüphanesinde kullanılan bir fonksiyonunun modifiye edilmiş hali, bizim label değerlerimizdeki en yüksek değeri hesaplamak için ve label değerlerinin tutulduğu diziyi, tasarlanan CNN modeline hazır hale getirmek için kullanılacaktır.

Tüm bu işlemlerden sonra, veriler algoritma için hazır hale getirilmiş olacaktır. Süreç; çekirdekte bir CNN ağ modelinden, bu modelin oluşturulması (get_create_model fonksiyonu), eğitilmesi, test edilmesi, True Positive, True Negative, False Positive, False Negative gibi değerlerin hesaplanması (train_and_evaluate_model fonksiyonu) ve tüm bu işlemlerin KFold algoritması sayesinde çarpaz doğrulama yaparak gerçekleştirilmesini içerecektir.

4- Sonuç

KFold algoritması sayesinde değişen her train ve test verisine karşılık, elde edilen doğruluk değerleri, True Positive, True Negative, False Positive, False Negative değerleri ve her KFold iterasyonunda elde edilen bu doğruluk değerlerinin ortalamaları hesaplanarak, kullanıcıya sonuç raporu mahiyetinde gösterildiği kısımdır.