**REST ile Sınıflandırma Projesi**

**Giriş**

Bu proje, veri setindeki 16 farklı çiçek sınıfını sınıflandırmak amacıyla yapılmıştır. Bu amaç doğrultusunda, önceden eğitilmiş bir ResNet50 modeli kullanılarak transfer öğrenme tekniği uygulanmıştır. Modelin eğitimi ve doğrulaması için TensorFlow ve Keras kütüphaneleri kullanılmıştır. Bu rapor, projenin tüm aşamalarını ve sonuçlarını detaylı olarak açıklamaktadır.

**Veri Seti**

Veri seti: <https://www.kaggle.com/datasets/l3llff/flowers/data>

Proje, flowers adlı bir veri seti kullanılarak gerçekleştirilmiştir. Bu veri seti, 16 farklı çiçek sınıfına ait resimlerden oluşmaktadır. Her sınıf, farklı sayıda görüntü içermekte olup, bu görüntüler eğitim ve doğrulama seti olarak ikiye ayrılmıştır.Toplam görüntü sayısı 15740 dır.

**Sınıflar şunlardır:** astilbe, bellflowe , black\_eyed\_susan , calendula , california\_poppy, carnation , common\_daisy, coreopsis , daffodil , dandelion , iris, magnolia , rose , sunflower, tulip , water\_lily

**Veri Setinin Hazırlanması**

Veri seti, TensorFlow'un tf.keras.preprocessing.image\_dataset\_from\_directory fonksiyonu kullanılarak yüklenmiştir. Veri setinin %80'i eğitim, %20'si ise doğrulama için kullanılmıştır. Görüntüler, modelin gereksinimlerine uygun olacak şekilde yeniden boyutlandırılmıştır.



**One - Hot Kodlama İşlemi**

**One-Hot Kodlama**

Modelimizin eğitimi öncesinde, veri setimizin etiketlerini one-hot kodlama yöntemiyle dönüştürdük. Bu işlem, sınıfların kategorik verilerden sayısal verilere dönüştürülmesini sağlar ve sınıflandırma modellerinin etiketleri daha kolay işlemesine yardımcı olur.

Etiketlerin One-Hot Kodlanması:

One-Hot Kodlama Fonksiyonu: one\_hot\_encode fonksiyonu, görüntü ve etiketleri alır ve etiketleri one-hot kodlamaya dönüştürür.

Map Fonksiyonu ile Dönüşüm: Eğitim ve doğrulama veri setleri, map fonksiyonu kullanılarak one-hot kodlamaya tabi tutulur.



**RESNET50**

ResNet50 (Residual Networks): Derin öğrenme modelleri arasında önemli bir yer tutan ve 50 katmandan oluşan bir yapıdır. Microsoft Research tarafından geliştirilmiş ve ImageNet yarışmasında büyük başarı elde etmiştir. ResNet, özellikle derin sinir ağlarının eğitiminde karşılaşılan "vanishing gradient" (kaybolan gradyan) problemini çözmeyi hedefler.

**ResNet50'nin Özellikleri**

Derinlik: 50 katmanlı derin bir yapıya sahiptir, bu sayede karmaşık özellikleri öğrenme kapasitesi yüksektir.

Residual (Artık) Bağlantılar: Modelin anahtar özelliklerinden biridir. Bu bağlantılar, katmanlar arasında atlamalar yaparak bilgi akışını kolaylaştırır ve kaybolan gradyan problemini azaltır. Bu sayede daha derin ağların eğitimi mümkün hale gelir.

Önceden Eğitilmiş Ağırlıklar: Model, geniş bir veri seti olan ImageNet üzerinde önceden eğitilmiş ağırlıkları içerir. Bu durum, modelin farklı veri setlerinde kullanılmasını ve transfer öğrenme yoluyla hızlıca adapte olmasını sağlar.

Kullanım Alanları

Görüntü Sınıflandırma: Farklı nesne ve kategorileri tanımlamada kullanılır.

Nesne Tespiti: Görüntü içindeki nesneleri belirler ve sınıflandırır.

Segmentasyon: Görüntülerin belirli bölümlerini ayırt eder.

Görüntü Tanıma ve Analizi: Medikal görüntülerde anomali tespiti gibi alanlarda yaygın kullanılır.

**Model Mimarisi**

Model Mimarisi

Modelimiz, Sequential model olarak yapılandırılmış olup, önceden eğitilmiş bir ResNet50 tabanına sahiptir ve sonrasında birkaç ek katman içerir:

ResNet50 (Functional):

Çıkış Şekli: (None, 2048)

Parametre Sayısı: 23,587,712

Aktivasyon Fonksiyonu: ReLU (ResNet50 içinde)

Özelliği: Önceden eğitilmiş ağırlıklar kullanılmış ve katmanları eğitilebilir değil.

Flatten:

Çıkış Şekli: (None, 2048)

Parametre Sayısı: 0

Özelliği: Veriyi tek boyutlu bir vektöre dönüştürür.

Dense (512 nöron):

Çıkış Şekli: (None, 512)

Parametre Sayısı: 1,049,088

Aktivasyon Fonksiyonu: ReLU

Dropout (0.5):

Çıkış Şekli: (None, 512)

Parametre Sayısı: 0

Özelliği: Overfitting'i önlemek için %50 oranında nöronları rastgele kapatır.

Dense (16 nöron):

Çıkış Şekli: (None, 16)

Parametre Sayısı: 8,208

Aktivasyon Fonksiyonu: Softmax

Toplam Parametre Sayısı: 24,645,008

Eğitilebilir Parametreler: 1,057,296

Eğitilemez Parametreler: 23,587,712



**MODELİN DERLENMESİ**

Optimizasyon Algoritması:

Adam optimizasyon algoritması kullanılmıştır. Bu algoritma, öğrenme oranını dinamik olarak ayarlayarak hızlı ve etkili bir eğitim süreci sağlar. Öğrenme oranı 0.001 olarak belirlenmiştir.

Kayıp Fonksiyonu:

Categorical Crossentropy kayıp fonksiyonu kullanılmıştır. Bu fonksiyon, çok sınıflı sınıflandırma problemlerinde yaygın olarak kullanılır ve modelin tahminlerinin doğruluğunu ölçer.

Performans Metrikleri:

Modelin performansını değerlendirmek için accuracy metriği kullanılmıştır. Bu metrik, modelin doğruluğunu hesaplar ve eğitim süreci boyunca modelin başarımını izlemeye olanak tanır.



**MODELİN EĞİTİLMESİ :**

Model, yapılandırıldıktan ve derlendikten sonra eğitim sürecine geçilir. Eğitim sırasında model, eğitim veri seti üzerinde 50 epoch boyunca eğitilmiştir. Her epoch'ta model, veri setindeki tüm örnekleri gözden geçirir ve ağırlıklarını güncelleyerek hata oranını azaltmaya çalışır. Eğitim süreci boyunca, modelin başarımı doğrulama veri seti üzerinde de değerlendirilir.

Eğitim süreci sonunda model, eğitim veri seti üzerinde yüksek doğruluk oranlarına ulaşmıştır ve belirli bir doğrulama doğruluğu elde etmiştir. Bu aşamalar, modelin belirli bir görevi (bu durumda çiçek sınıflandırması) öğrenmesi ve genelleme yapabilme kabiliyetini geliştirmesini sağlamıştır.



**Sonuçlar ve Değerlendirme**

Modelin eğitimi sonucunda elde edilen sonuçlar aşağıdaki gibidir:

Epoch sayısı arttıkça, eğitim ve doğrulama veri setlerindeki doğruluk oranları artmıştır.

İlk epoch'ta eğitim doğruluğu %71.85 iken, son epoch'ta %97.04'e yükselmiştir.

Doğrulama doğruluğu ise ilk epoch'ta %84.50 iken, son epoch'ta %90.63'e yükselmiştir.

Kayıp değerleri de eğitim süreci boyunca azalmıştır.

Bu sonuçlar, modelin çiçek sınıflarını doğru bir şekilde tanımlayabildiğini ve genelleme yeteneğinin yüksek olduğunu göstermektedir. Ancak, modelin bazı epoch'larda aşırı uydurma (overfitting) eğilimi gösterdiği görülmektedir. Bu durum, modelin daha fazla veriyle eğitilerek veya modelin karmaşıklığının azaltılarak düzeltilebilir.





**GRAFİKLER**

eğitim sırasında kaydedilen eğitim ve doğrulama (validation) doğruluk değerlerini kullanarak bir grafik oluşturuldu. Amaç, modelin eğitim sürecindeki doğruluğunun ve doğrulama setindeki doğruluğunun nasıl değiştiğini görselleştirmektir.





Aynı zamanda eğitim sırasında elde edilen eğitim kaybı (loss) ve doğrulama (validation) kaybı değerlerini kullanarak bir grafik oluşturuldu . Amacı, modelin eğitim sürecindeki kayıp (hata) değerlerinin nasıl değiştiğini görselleştirmektir.



**TEST ETME**

Bu bölümde,örnek olarak iris çiçeğinin görüntüsünü işleyip, model tarafından sınıflandırma süreci açıklanmaktadır

İlk olarak, iris çiçeğinin bir görüntüsü, model tarafından sınıflandırılacak şekilde işlenir.

Model, işlenmiş görüntüyü alır ve çıktı olarak sınıf tahmininde bulunur.

Sonuç olarak, modelin başarıyla çalıştığı ve görüntünün iris çiçeği sınıfını doğru bir şekilde belirlediği sonucuna ulaşırız.



