

**EGE ÜNİVERSİTESİ**

**MÜHENDİSLİK FAKÜLTESİ**

**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

**YAPAY ZEKÂ YÖNTEMLERİ (3+0)**

**2022-2023 BAHAR YARIYILI**

**PROJE-2 RAPORU**

**TESLİM TARİHİ**

17/06/2023

**HAZIRLAYANLAR**

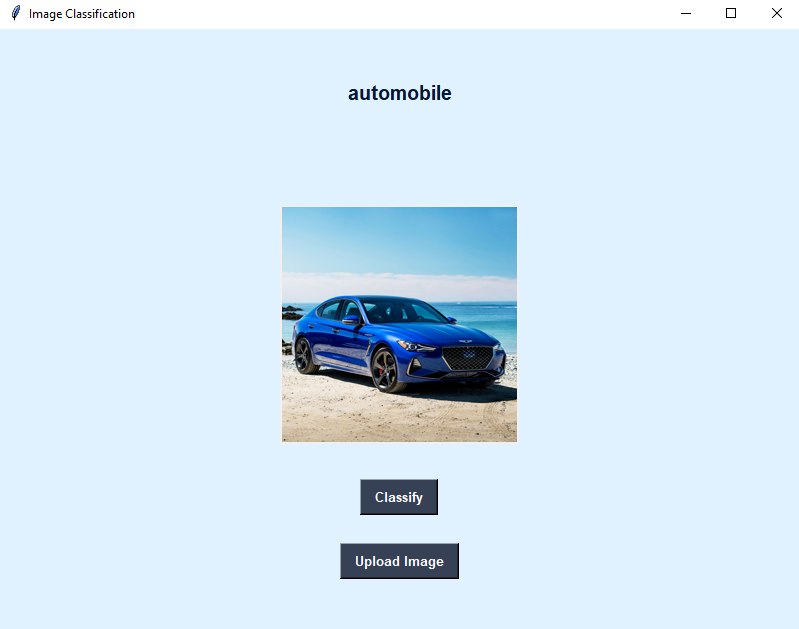
05180000063, Burak KIZILAY

05190000080, Bahri Bilgekağan TAŞ

05210000915, Abdullah Alper BAŞ

# 1. Problemin Tanımı

Bu proje, Keras kütüphanesi yardımı ile yapılan bir resim sınıflandırma projesidir. Bir verisetiyle eğitilen sinir ağı, bu verisetindeki sınıflardan herhangi birini içeren bir resmin hangi sınıfa ait olduğunu tahmin eder. Bu projeye bir GUI ekledik ve aynı zamanda bir web uygulaması haline getirdik. Kullanıcı “Upload an Image” butonuna bastıktan sonra tahmin ettirmek istediği resmi “Upload an Image” butonuna basarak yükledikten sonra “Classify Image” butonuna basarak, tahmin sonucunu ekranda görebilir.





# 2. Araştırma

Araştırmalarımız sonucunda bu projeyi gerçekleştirmek için bir CNN modeli oluşturmanın daha uygun ve verimli olduğunu öğrendik. CNN mimarisini, bir CNN modeli oluştururken nelere dikat etmemiz gerektiğini, katmanların ne anlama geldiğini, katmanlardaki parametrelerin ne işe yaradığı, aktivasyon, optimizasyon ve kayıp fonksiyonlarının ne işe yaradığını öğrendik. Bir modeli oluşturduktan sonra bu modelin başarısını artırmak için hangi yollara başvurmamız gerektiğini öğrendik.

# 3. Kullanılan Ortam, Yöntem ve Kütüphaneler

Projemizde resim sınıflandırma için CNN modelleri kullandık. 4 model oluşturduk daha sonra aralarında en yüksek başarıya sahip modelin başarısını K-Fold yöntemi kullanarak yükselttik.

Projemizde kullandığımız ortamlar ve kütüphaneler şunlardır:

Tensorflow – 2.5.0

Numpy – 1.18.5

Scikit-learn – 0.24.2

Pillow – 8.2.0

Tk – 8.6.10

Flask – 1.1.2

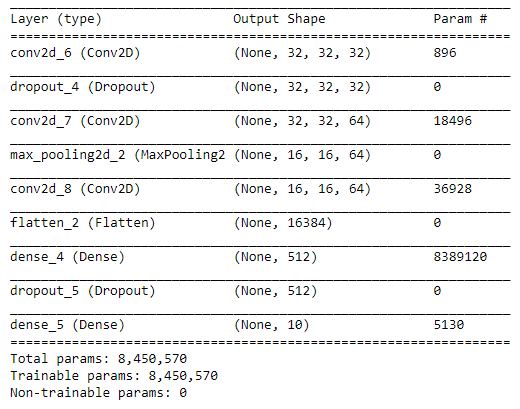
Jupyter Notebook

Pycharm

# 4. Önerilen Yöntem

Projemizde toplam 4 adet model oluşturduk. Bu modeller arasında başarı oranı en yüksek olan modeli GUI ve web uygulamasında kullandık.

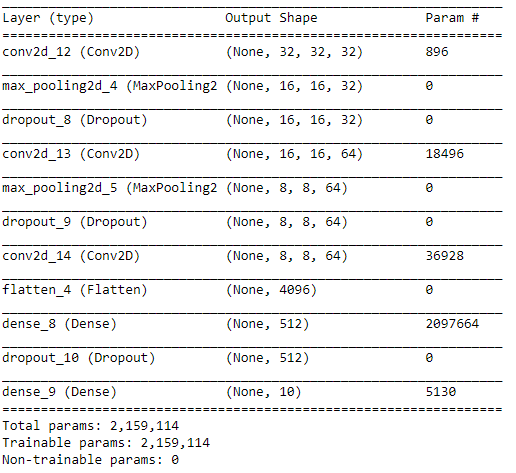
Projemizde kullandığımız 1. modelin yapısı şu şekildedir:



Bu modelin başarısı % 70.21 çıkmıştır. Elimizdeki 50 adet resimle test ettiğimizde ise başarı oranı %52 çıkmıştır.



2. modelin yapısı şu şekildedir:

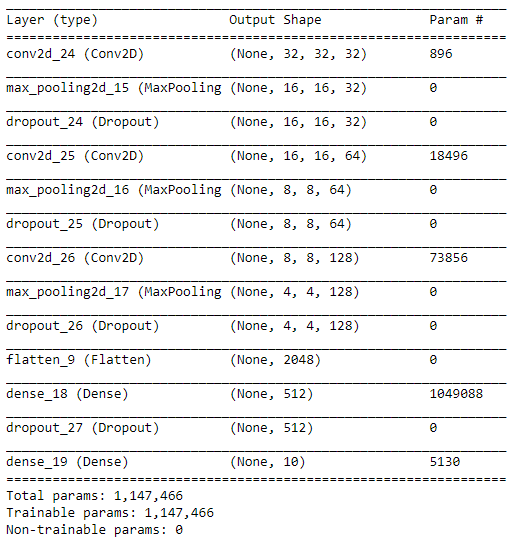


Bu modelin başarısı % 73.58 çıkmıştır. Elimizdeki 50 adet resimle test ettiğimizde ise başarı oranı %52 çıkmıştır.



Bu modeli bir önceki modelin üstüne başarısını artırmak için 1 adet maxpooling katmanı, 1 adet dropout katmanı ekledik ve epoch sayısını 10’dan 20’ye yükselttik.

3. modelin yapısı şu şekildedir:

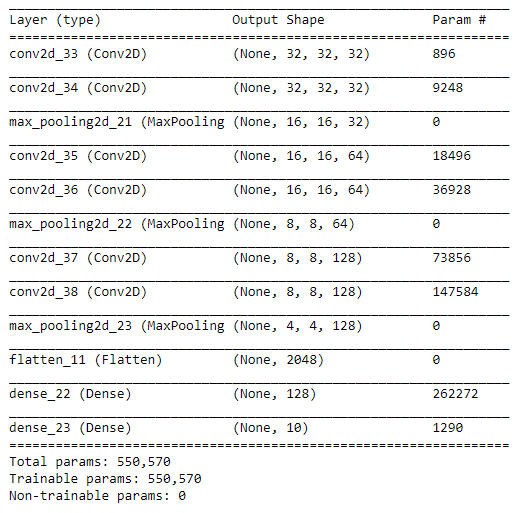


Bu modelin başarı oranı % 75.7 çıkmıştır. Elimizdeki 50 adet resimle test ettiğimizde ise başarı oranı %52 çıkmıştır.



Bu modelde önceki modelde optimizasyon fonksiyonu olarak kullandığımız ‘Adam’ fonksiyonunu ‘SGD’ ile değiştirdik ve 3. Convolutional katmanının çıktı boyutunu 64’ten 128’e yükselttik.

4. modelin yapısı şu şekildedir:

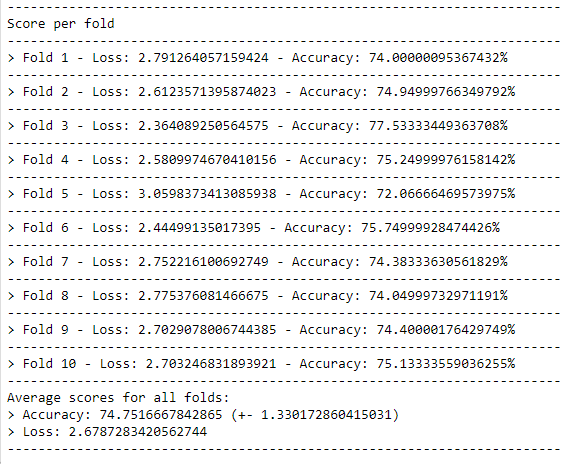


Bu modelin başarısı % 72.98 çıkmıştır. Elimizdeki 50 adet resimle test ettiğimizde ise başarı oranı %72 çıkmıştır.



Bu modelde de önceki modellerden daha fazla convolutional katman kullandık ve dropout katmanlarını çıkardık. Batch sayısını 32’den 64’e, epoch sayısını 100’e yükselttik.

Şu ana kadar en başarılı model 4. Model olduğu için bu modelin başarısını K-Fold kullanarak artırmaya çalıştık. 10 fold kullandık. K-Fold sonuçları şu şekildedir:



K-Fold sonucunda başarı oranı en yüksek (% 77.53) olan 3. Fold’u kullanmaya karar verdik. Bu fold sonucunda oluşan modeli elimizdeki 50 resimle test ettiğimizde %80’lik bir başarı oranı elde ettik. Başarı oranı en yüksek olan model bu olduğu için GUI uygulamasında ve web uygulamasında bu modeli kullandık.

# 5. Deneysel Çalışmalar – Veri Setinin Açıklanması

Bu projede CIFAR-10 isimli veri setini kullandık.

<https://www.cs.toronto.edu/~kriz/cifar.html>

Bu verisetinde 10 sınıf, her sınıfta 6000 adet olmak üzere toplam 60000 adet 32x32 resim bulunmaktadır. Bu sınıflar şunlardır:

Airplane

Car

Bird

Cat

Deer

Dog

Frog

Horse

Ship

Truck

Bu veri setindeki resimlerin 50000 adedi eğitim için, 10000 adedi test için kullanılmıştır.

Bu veri seti halihazırda Tensorflow’un Keras kütüphanesinde bulunmaktadır yani import edilebilir. İndirmeye gerek yoktur.

# 6. Sonuç

Projemizin başarısı ilk modellerde çok düşükken son modelde tatmin edici bir düzeye ulaşmıştır.

Gerçek hayatta, bu projedeki veri seti olmasa da başka veri setleri kullanılarak otonom araçlarda çevredeki nesnelerin, trafik işaretlerinin sınıflandırılmasında kullanılabilir. Veya sağlık alanında görsel verilerle hastalık tanısında kullanılabilir.

Projeyi geliştirirken resim sınıflandırmanın ne olduğunu, hangi alanlarda kullanıldığını, CNN mimarisini, katmanların ne olduğunu ve ne işe yaradığını, katman parametrelerinin, epoch ve batch sayısının ne olduğunu ve başarı oranını nasıl değiştireceğini, aktivasyon ve optimizasyon fonksiyonlarını, K-Fold CV yapmayı, başarı oranını artırmak için neler yapılabileceğini öğrendik.

# 7. Kaynakça

<https://keras.io/api>

<https://medium.com/turkce/keras-ile-derin-öğrenmeye-giriş-40e13c249ea8>

<https://medium.com/@tuncerergin/keras-ile-derin-ogrenme-modeli-olusturma-4b4ffdc35323>

<https://blog.keras.io/building-powerful-image-classification-models-using-very-little-data.html>

<https://www.analyticsvidhya.com/blog/2020/10/create-image-classification-model-python-keras/>

<https://www.tensorflow.org/tutorials/images/classification>

<https://www.youtube.com/watch?v=tPYj3fFJGjk&t=13347s> (TensorFlow 2.0 Complete Course - Python Neural Networks for Beginners Tutorial)

<https://www.machinecurve.com/index.php/2020/02/18/how-to-use-k-fold-cross-validation-with-keras/>

<https://www.machinecurve.com/index.php/2020/02/09/how-to-build-a-convnet-for-cifar-10-and-cifar-100-classification-with-keras/>

# Ek 1.

Başarımı artırmak için katman ekleyip çıkardık, epoch ve batch sayısını artırdık, katman parametrelerinde değişiklikler yaptık, optimizasyon fonksiyonunda değişiklikler yaptık. Bunlar sonucunda başarı oranında kağıt üzerinde % 2-3 civarlarında artış gözlemledik. Fakat bizim test ettiğimiz 50 resim için, başarı oranında % 20’lik bir artış gözlemledik. Bu başarıyı artırmak için de K-Fold yöntemi kullandıktan sonra kağıt üzerinde % 4.5 artış, elimizdeki resimlerle yaptığımız testte % 8 artış gözlemledik.

# Ek 2.

Oluşturduğumuz modelin başarı oranı, bulduğumuz çalışmaların başarı oranından yaklaşık % 10 daha fazladır. Ayrıca projemizin web uygulaması halinde olmasının diğer çalışmalara göre farklılık kazandırdığını düşünüyoruz. Araştırdığımızda herhangi bir online sınıflandırma projesi bulamadık.

# Ek 3.

Yararlandığımız bağlantılar şunlardır:

<https://www.machinecurve.com/index.php/2020/02/18/how-to-use-k-fold-cross-validation-with-keras/>

<https://colab.research.google.com/drive/1ZZXnCjFEOkp_KdNcNabd14yok0BAIuwS#forceEdit=true&sandboxMode=true>

<https://medium.com/@tuncerergin/keras-ile-derin-ogrenme-modeli-olusturma-4b4ffdc35323>

Bu çalışmalarla bizim çalışmamızın arasında model farklılıkları bulunmaktadır. Yeni katmanlar eklendi, bazı katmanlar çıkarıldı, optimizasyon fonksiyonu değiştirildi, bazı katman parametreleri (çıktı boyutları vs.) değiştirildi, epoch ve batch sayılarında farklı değerler denendi.

# Ek 4.

a.1-) Calinski-Harabasz Index (Varyans Oran Kriteri)

Yoğun ve iyi ayrılmış kümeler kavramına dayanan yöntemlerden biri Calinski-Harabasz indeksidir. Bu

yöntemi kullanabilmek için önce kümeler arası dağılımı tanımlamamız gerekir. Göreceli ağırlık

noktaları ve küresel ağırlık merkezine sahip bir k kümemiz varsa, kümeler arası dağılım (BCD) şu

şekilde tanımlanır:

yazı tipi, metin, beyaz, hat sanatı, kaligrafi içeren bir resim

Açıklama otomatik olarak oluşturuldu

a.2-) Rand Index İki veri kümelenmesi arasındaki benzerliğin bir ölçüsüdür. Aynı veya farklı kümelemedeki tüm veri çiftlerini dikkate alarak bu hesaplanır.

yazı tipi, beyaz, logo, grafik içeren bir resim

Açıklama otomatik olarak oluşturuldu

b-) 2017’de Google araştırmacıları ve geliştiricileri, Transformer modelinin yükselişini vurgulayan "Attention is All You Need" başlıklı makaleyi yayınladı. Makalelerinde, önceki doğal dil işleme (NLP) model mimarilerine kıyasla çeviri görevleri için yeni bir seviyeye ulaşıldığından bahsediyor. Transformer modelinin mimarisi, diziden diziye (seq2seq) görevleri yerine getirmek için RNN'lerdeki kodlayıcı-kod çözücü mimarisinde kullanılan dikkat mekanizmasından ilham alır, ancak sıralılık faktörünü ortadan kaldırır; yani, RNN'lerin aksine transformatör verileri sırayla işlemez, bu da daha fazla paralelleştirmeye izin verir ve eğitim süresini azaltır.

# 8. Özdeğerlendirme Tablosu

ÇALIŞMA SÜRELERİ:

Burak KIZILAY – 7-8 saat

Bilgekağan TAŞ – 5-6 saat

Alper BAŞ – 5-6 saat

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **İstenen Madde** | **Var** | **Açıklama** | **Tahmini Not** |
| **1** | **Kapak Sayfası, Problemin Tanımı, Kullanılan Ortam, Yöntem ve Kütüphaneler, Araştırma(10)** |  | Kapak sayfası ve problemin tanımı yapıldı, kullanılan ortam, yöntem ve kütüphaneler açıklandı. Yaptığımız araştırmalar sonucu neler öğrendiğimiz açıklandı. İncelediğimiz linkler Kaynakça başlığı altına eklendi. | 10 |
| **2** | **Önerilen Yöntem (10)** |  | Oluşturduğumuz modellerin katman şemaları ve başarı yüzdeleri açıklandı. Deneysel çalışmalar burada incelendi. | 10 |
| **3** | **Deneysel Çalışmalar (10)** |  | Yaptığımız deneysel çalışmalar Önerilen Yöntem başlığı altında incelendi. | 9 |
| **4** | **Proje Rapor Biçimi, Organizasyonu, Boyutu, Kalitesi, Kaynakça ve atıflar (10)** |  | Yararlandığımız çalışmalar, incelediğimiz yazılar ve izlediğimiz videoların linkleri Kaynakça başlığının altına eklendi. | 10 |
| **5** | **Sonuç (10)** |  | Projenin başarısı yorumlandı. Hangi alanlarda kullanılabileceği ve nasıl yararlı olabileceği açıklandı. Bu projenin bize neler kattığı, öğrettiği, kazandırdığı açıklandı. | 9 |
| **6** | **Ek 1: Başarım İyileştirme (10)** |  | Başarı oranını yükseltmek için neler yaptığımız açıklandı. | 10 |
| **7** | **Ek 2 (10)** |  | Projenin diğer çalışmalardan farkı açıklandı. | 10 |
| **8** | **Ek 3 (10)** |  | Projenin incelediğimiz çalışmalardan farkı açıklandı. | 9 |
| **9** | **Ek 4 (10): Her madde 5’er puan** |  | Araştırılıp belirtildi. | 9 |
| **10** | **Özdeğerlendirme Tablosu (10)** |  | Özdeğerlendirme Tablosu dolduruldu, açıklamalar yapıldı. | 10 |
| **100 üzerinden Toplam Not:** | | | | 97 |