



**T.C.**

**İSTANBUL MEDENİYET ÜNİVERSİTESİ  
MÜHENDİSLİK VE DOĞA BİLİMLERİ FAKÜLTESİ  
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ  
DERİN ÖĞRENME DERSİ PROJESİ**

**Derin Öğrenme Yöntemleri ile Nesne Tanıma  
CIFAR-10 Veri Seti Üzerine Bir CNN Uygulaması**

Hilal Nur Albayrak – 22120205056

**Ders Sorumlusu**

Dr. Öğr. Üyesi İshak DÖLEK

Aralık 2025, İSTANBUL

# 1. GİRİŞ VE PROJE KONUSU

## 1.1. Projenin Konusu ve Amacı

Bu projenin temel amacı, derin öğrenme tekniklerini kullanarak görüntü verileri üzerinden otomatik nesne sınıflandırması yapan bir sistem geliştirmektir. Proje kapsamında, bilgisayarlı görü (computer vision) alanının temel problemlerinden biri olan "Görüntü Sınıflandırma" (Image Classification) problemi ele alınmıştır.

## 1.2. İlgili Alanın Önemi ve Seçilme Gerekçesi

Günümüzde nesne tanıma teknolojileri; otonom araçların yaya ve levhaları tanımasından, sağlık sektöründe tıbbi görüntülerin analizine, güvenlik kameralarındaki şüpheli durum tespitinden endüstriyel kalite kontrole kadar çok geniş bir yelpazede kullanılmaktadır. Ders kapsamında öğrenilen teorik bilgilerin (Özellikle CNN mimarilerinin), gerçek hayat problemlerine uygulanabilirliğini göstermek amacıyla bu proje konusu seçilmiştir. Klasik programlama yöntemleriyle çözülmesi imkansız olan görsel verilerin analizi, derin öğrenme sayesinde yüksek doğrulukla gerçekleştirilebilmektedir.

## 2. LİTERATÜR TARAMASI VE ÖNCEKİ ÇALIŞMALAR

Görüntü işleme literatüründe uzun yıllar boyunca el ile çıkarılan özellikler (hand-crafted features) kullanılmıştır. Ancak 2012 yılında AlexNet modelinin ImageNet yarışmasındaki başarısı ile birlikte, Evrişimli Sinir Ağları (Convolutional Neural Networks - CNN) endüstri standardı haline gelmiştir.

Literatürdeki önemli çalışmalar şunlardır:

- **LeNet-5 (1998):** Yann LeCun tarafından geliştirilen ve el yazısı rakamları tanıyan ilk başarılı CNN mimarisidir.
- **AlexNet (2012):** Derin öğrenme devrimini başlatan, GPU kullanımının önünü açan çalışmadır.
- **VGG ve ResNet:** Daha derin ağların (Deep Networks) kaybolan gradyan (vanishing gradient) sorunu yaşamadan eğitilebileceğini kanıtlayan modern mimarilerdir.

Bu projede, bu öncü çalışmalardan ilham alınarak, eğitim maliyeti ve başarı oranı dengesi gözetilen özelleştirilmiş bir CNN mimarisi tasarlanmıştır.

## 3. VERİ SETİNİN BELİRLENMESİ

Projede CIFAR-10 veri seti kullanılmıştır.

- **Veri Seti İçeriği:** Veri seti; uçak, otomobil, kuş, kedi, geyik, köpek, kurbağa, at, gemi ve kamyon olmak üzere 10 farklı sınıftan oluşmaktadır.
- **Boyutlar:** Görüntüler 32x32 piksel boyutunda ve renklidir (3 kanal: RGB).
- **Miktar:** Toplam 60.000 görüntü bulunmaktadır. Bunların 50.000 adedi eğitim (training), 10.000 adedi ise test (test) seti olarak ayrılmıştır.

- **Seçim Gerekçesi:** CIFAR-10, derin öğrenme algoritmalarının kıyaslanması (benchmarking) için akademik dünyada kabul görmüş standart bir veri setidir. Veri setinin dengeli dağılımı (her sınıftan eşit sayıda resim olması) ve boyutlarının eğitim süresini optimize etmeye uygun olması nedeniyle tercih edilmiştir.

## 4. UYGULANACAK YÖNTEM VE ALGORİTMA SEÇİMİ

### 4.1. Yöntem Seçimi:

Neden CNN? Görüntü verileri, pikseller arası uzaysal ilişkilere (spatial correlation) sahiptir. Klasik Yapay Sinir Ağları (Artificial Neural Networks - ANN), görüntüyü düz bir vektör haline getirerek işlediği için komşu pikseller arasındaki ilişkiyi kaybeder.

Bu projede Evrişimli Sinir Ağları (CNN) tercih edilmiştir. Bunun nedenleri:

1. **Özellik Çıkarımı (Feature Extraction):** CNN, filtreler (kernels) yardımıyla görüntüdeki kenar, köşe ve doku gibi özellikleri otomatik olarak öğrenir.
2. **Parametre Paylaşımı:** Görüntünün bir yerindeki özelliği (örn. kedi kulağı) başka bir yerinde de tanıyabilir.
3. **Boyut Azaltma:** Pooling katmanları sayesinde işlem yükünü azaltırken önemli bilgiyi korur.

### 4.2. Literatür Karşılaştırması

Literatürde kullanılan diğer yöntemlerle (KNN, SVM veya MLP) kıyaslandığında, CNN mimarisi ham piksel verisinden öğrenme yeteneği sayesinde %90+ başarı oranlarına ulaşabilen tek yöntemdir.

### 4.3. Tasarlanan Model Mimarisi

Projede oluşturulan model (src/model\_builder.py dosyasında kodlanmıştır) şu katmanlardan oluşur:

1. **Giriş Katmanı:** 32x32x3 boyutunda görüntü.
2. **Evrişim Blokları (Convolutional Blocks):**
  - Conv2D: 32 ve 64 filtreli katmanlar ile özellik haritaları çıkarılır. Aktivasyon fonksiyonu olarak, lineer olmayan özellikleri öğrenmek için ReLU kullanılmıştır.
  - BatchNormalization: Eğitim sürecini hızlandırmak ve kararlı hale getirmek için eklenmiştir.
  - MaxPooling2D: Boyutları yarıya indirerek hesaplama maliyetini düşürür.
3. **Düzenleştirme (Regularization):**
  - Dropout: Rastgele nöronları kapatarak (%20, %30, %40 oranlarında) modelin ezberlemesini (overfitting) engellemek için kullanılmıştır.

#### 4. Sınıflandırma Katmanı (Fully Connected):

- Flatten: Veriyi tek boyutlu vektöre çevirir.
- Dense: 128 nöronlu gizli katman.
- Output Layer: 10 sınıf için 10 nöronlu çıkış katmanı (Softmax aktivasyonu ile olasılık üretir).

### 5. MODEL EĞİTİMİ VE DEĞERLENDİRME

#### 5.1. Eğitim Süreci

Model eğitimi aşağıdaki hiperparametreler ile gerçekleştirilmiştir:

- **Platform:** Python 3.12, TensorFlow/Keras Kütüphanesi.
- **Optimizasyon Algoritması:** Adam (Adaptive Moment Estimation) - Hızlı yakınsama sağladığı için seçilmiştir.
- **Kayıp Fonksiyonu (Loss Function):** Sparse Categorical Crossentropy (Çok sınıflı sınıflandırma için).
- **Epoch Sayısı:** 15 (Early stopping gözlemleriyle belirlenmiştir).
- **Veri Artırma (Data Augmentation):** Modelin genelleme yeteneğini artırmak için eğitim verileri üzerinde rastgele döndürme, kaydırma ve yatay çevirme işlemleri uygulanmıştır.

#### 5.2. Performans Metrikleri

Modelin başarısı Doğruluk (Accuracy), Hata (Loss) ve Karmaşıklık Matrisi (Confusion Matrix) ile ölçülmüştür.

- **Eğitim Sonuçları:** Eğitim setinde %80 civarı, test setinde ise %75-%78 bandında doğruluk elde edilmiştir. Dropout kullanımı sayesinde eğitim ve test başarısı arasındaki fark düşük tutulmuş, overfitting engellenmiştir.
- **Sınıf Bazlı Başarım:** Model; "Otomobil", "Gemi" ve "Kamyon" gibi belirgin hatlara sahip sınıflarda yüksek başarı gösterirken; "Kedi" ve "Köpek" gibi birbirine benzeyen biyolojik sınıflarda kısmi karıştırmalar yapmıştır (Confusion Matrix analizine göre).

### 6. PROJE DOKÜMANTASYONU

Projenin tüm kaynak kodları, eğitim ağırlıkları (.h5 dosyası) ve kurulum yönergeleri GitHub üzerinde açık kaynak olarak paylaşılmıştır.

- **GitHub Depo Linki:** [https://github.com/KULLANICI\\_ADIN/NesneTanimaProjesi](https://github.com/KULLANICI_ADIN/NesneTanimaProjesi) (Buraya kendi linkini yapıştır)
- **Proje Yapısı:** Modüler bir mimari (Modular Design) benimsenmiş; veri işleme, model oluşturma ve eğitim kodları src klasörü altında ayrıştırılmıştır.

- **Arayüz (Deployment):** Eğitilen model Gradio kütüphanesi kullanılarak web tabanlı bir arayüze dönüştürülmüş ve son kullanıcı testine sunulmuştur.

## 7. SONUÇ VE ÖNERİLER

Bu projede, derin öğrenme temelli bir nesne tanıma sistemi başarıyla geliştirilmiştir. Elde edilen sonuçlar, CNN mimarisinin görüntü sınıflandırma problemlerindeki üstünlüğünü kanıtlar niteliktedir. Gelecek çalışmalarda, Transfer Learning (VGG16 veya ResNet50) yöntemleri kullanılarak başarı oranı %90 seviyelerine çıkarılabilir.

## KAYNAKÇA

1. Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks.
2. LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition.
3. CIFAR-10 Dataset: <https://www.cs.toronto.edu/~kriz/cifar.html>
4. Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press.