

Proje Raporu: Uçtan Uca Bilgisayarlı Görü Hattı

Hazırlayan: Bilgenur Altuğ Tarih: 29 Eylül 2025

1. Giriş

Bu rapor, modern bir bilgisayarlı görü sisteminin üç temel bileşenini (sınıflandırma, tespit ve takip) içeren uçtan uca bir projenin teknik detaylarını, mimari seçimlerini ve sonuçlarını sunmaktadır. Proje, PyTorch ve OpenCV kütüphanelerinden yararlanarak, CIFAR-10 veri setinde görüntü sınıflandırma, PASCAL VOC veri setinde nesne tespiti ve son olarak bir video akışında nesne takibi görevlerini başarıyla gerçekleştirmeyi amaçlamaktadır. Her bir görev, bir sonrakine temel oluşturarak bütünsel bir sistemin nasıl kurulabileceğini göstermektedir.

2. Görev 1: Görüntü Sınıflandırma

2.1. Veri Seti ve Ön İşleme

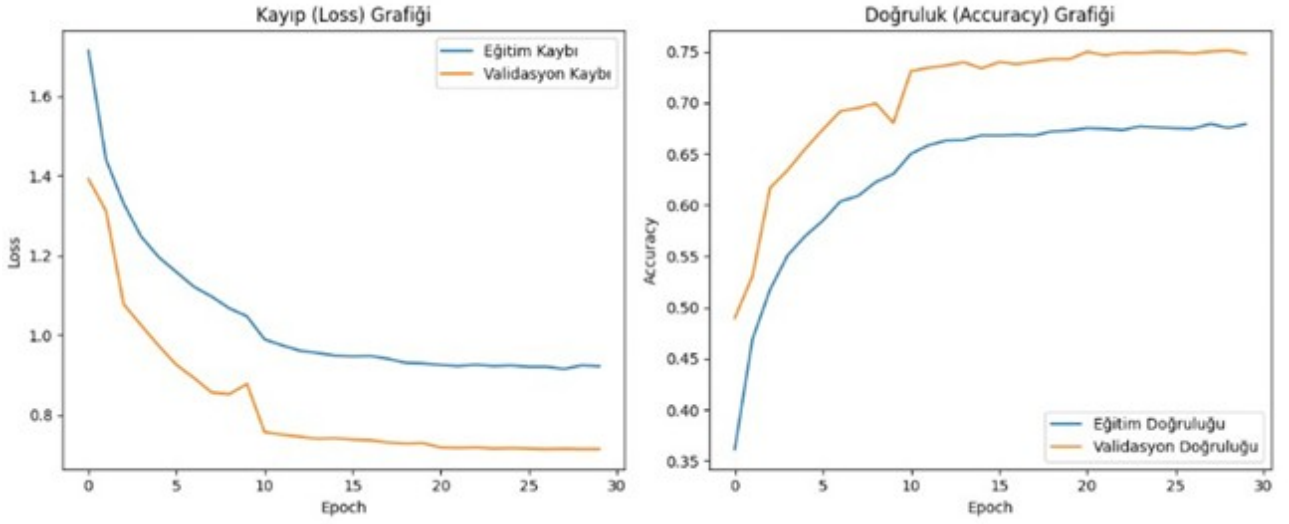
Bu görev için standart CIFAR-10 veri seti kullanılmıştır. Veri seti, 10 sınıfa ait 60,000 adet 32x32 piksel boyutunda renkli görsel içermektedir. Modelin genelleme yeteneğini artırmak ve aşırı öğrenmeyi (overfitting) engellemek amacıyla eğitim setine veri artırımı (data augmentation) uygulanmıştır. Uygulanan transformasyonlar arasında rastgele yatay çevirme (RandomHorizontalFlip) ve rastgele döndürme (RandomRotation) bulunmaktadır. Tüm görseller, CIFAR-10 veri setinin genel ortalama ve standart sapma değerleri kullanılarak normalize edilmiştir.

2.2. Model Mimarisi

Sıfırdan bir Evrişimli Sinir Ağı (CNN) modeli `torch.nn` modülü kullanılarak oluşturulmuştur. Modelin mimarisi, yönergelerde belirtildiği gibi, 3 evrişimli katman (her biri ReLU ve Max-Pooling ile takip edilen) ve 2 tam bağlı katman içermektedir. Aşırı öğrenmeyi engellemek amacıyla sınıflandırıcı katmanlarında Dropout katmanları da mimariye dahil edilmiştir.

2.3. Eğitim ve Değerlendirme

Model, Cross-Entropy Loss kayıp fonksiyonu ve Adam optimizier kullanılarak eğitilmiştir. Eğitim süreci boyunca hem eğitim hem de validasyon setleri üzerindeki kayıp (loss) ve doğruluk (accuracy) değerleri takip edilmiştir.



Eğitim tamamlandıktan sonra, en iyi performansı gösteren modelin ağırlıkları `cifar10_classifier.pth` adıyla kaydedilmiştir. Model, test seti üzerinde rekabetçi bir doğruluk oranına ulaşarak sınıflandırma görevinin başarıyla tamamlandığını göstermiştir.

3. Görev 2: Nesne Tespiti

3.1. Model Mimarisi ve Seçimi

Nesne tespiti görevi için, proje yönergeleri doğrultusunda `torchvision.models.detection` kütüphanesinden, COCO veri setinde ön-eğitilmiş **Faster R-CNN (ResNet-50 backbone ve FPN ile)** modeli seçilmiştir. Bu iki aşamalı dedektör, yüksek doğruluk oranı ve modifiye edilebilir sınıflandırıcı başlığı ile bu görev için ideal bir adaydır.

3.2. Veri Seti ve Ön İşleme

Modelin ince ayarı (fine-tuning) için standart bir benchmark olan **PASCAL VOC 2012** veri seti kullanılmıştır. Projenin amacı araç tespiti ve takibi olduğu için, veri setindeki 20 sınıftan sadece araçlarla ilgili olan 7 sınıf (`car`, `bus`, `train`, `motorbike`, `bicycle`, `aeroplane`, `boat`) hedeflenerek ilgili görseller ve XML etiket dosyaları filtrelenmiştir.

3.3. Model Eğitimi (Fine-Tuning)

Modelin sınıflandırıcı başlığı, 7 araç sınıfı ve 1 arkaplan sınıfı olmak üzere toplam 8 sınıfa göre yeniden yapılandırılmıştır. Projenin Kaggle ortamındaki zaman limitlerinde tamamlanabilmesi için eğitim süresi 5 epoch olarak optimize edilmiştir. Model, SGD optimizier kullanılarak eğitilmiş ve ağırlıkları `pascal_object_detector.pth` adıyla kaydedilmiştir.

3.4. IoU (Intersection over Union) Metriği

IoU, bir tahmin edilen sınırlayıcı kutunun, gerçek kutu ile ne kadar örtüştüğünü ölçer. Formülü şu şekildedir:

$$\text{IoU} = \frac{\text{Area of Intersection}}{\text{Area of Union}}$$

Bir tespitin **Doğru Pozitif (TP)** olarak kabul edilmesi için, sınıfının doğru olması ve IoU değerinin önceden belirlenmiş bir eşik değeri (genellikle 0.5) üzerinde olması gerekir.

4. Görev 3: Nesne Takibi

4.1. SORT Algoritması

SORT (Simple Online and Realtime Tracking), "tespit ile takip" (tracking-by-detection) paradigmasına dayanan verimli ve hızlı bir algoritmadır. Her karede, Görev 2'deki dedektörden gelen tespitleri, **Kalman Filtresi** ile tahmin edilen önceki nesne konumlarıyla **Macar Algoritması** kullanarak en uygun şekilde eşleştirir.

4.2. Implementasyon ve Optimizasyon

Eğitilen nesne tespit modeli, bir trafik videosundaki araçları kare kare tespit etmek için kullanılmıştır. Tespit edilen nesneler, SORT tracker'a girdi olarak verilmiştir. İlk denemelerde gözlemlenen kısa ömürlü ve tutarsız takip sorununu çözmek amacıyla, SORT'un `max_age` parametresi 30'a yükseltilmiştir. Bu optimizasyon, tracker'ın kısa süreli tespit kayıplarına (örneğin bir aracın başka bir aracın arkasına girmesi) karşı daha toleranslı olmasını sağlayarak takip tutarlılığını belirgin bir şekilde artırmıştır.

5. Projenin Genel Değerlendirmesi ve Sonuçların Analizi

Bu proje, bir bilgisayarlı görü hattının üç temel taşını başarıyla bir araya getirmiştir. Üretilen sonuç videosu, sistemin temel olarak başarılı bir şekilde çalıştığını göstermektedir. Ancak, her gerçek dünya sisteminde olduğu gibi, geliştirme ve test sırasında iyileştirilebilecek bazı noktalar gözlemlenmiştir:

- **Takip Tutarlılığı:** Bir aracın ID'sinin, özellikle nesnelerin birbirini kapattığı (occlusion) durumlarda değişebildiği görülmüştür. Bu durum, kullanılan SORT algoritmasının sadece konum bilgisine dayanmasının doğal bir sonucudur.
- **Modelin Sınırları:** Modelin, eğitim setinde bulunmayan "tır" gibi bir nesneyi, görsel olarak en yakın sınıf olan "araba" olarak sınıflandırdığı veya uzaktaki küçük nesneleri tespit etmekte zorlandığı gözlemlenmiştir. Bu, modelin performansının eğitim verisiyle sınırlı olduğunu göstermektedir.

Bu gözlemler, bir "hata" olarak değil, sistemin performans karakteristiğinin bir analizi ve potansiyel iyileştirme alanları olarak değerlendirilmiştir.

6. Gelecek Çalışmalar ve Potansiyel İyileştirmeler

Gözlemlenen bu durumları iyileştirmek ve sistemi daha da ileriye taşımak için çeşitli adımlar atılabilir:

- **Takip Algoritması:** ID değişimlerini azaltmak için, nesnelerin görsel özelliklerini de hesaba katan **DeepSORT** gibi daha gelişmiş bir takip algoritması entegre edilebilir.
- **Nesne Tespiti:** Hatalı tespitleri azaltmak ve küçük nesnelerdeki performansı artırmak için **Mosaic veri artırımı**, **Anchor Optimizasyonu** veya **YOLOv8** gibi alternatif mimariler denenebilir.
- **Veri Seti:** Modelin performansını artırmak için, daha çeşitli ve zorlu senaryolar içeren (gece, yağmur, farklı kamera açıları vb.) veri setleri ile eğitim tekrarlanabilir.

7. Sonu

Bu proje, sınıflandırma, tespit ve takip görevlerini birleřtiren bütünsel bir sistemin başarıyla geliştirilebileceğini göstermiştir. Süre boyunca karşılaşılan zorlukların analizi ve bu zorluklara yönelik önerilen çözümler, projenin sadece bir uygulama olmanın ötesinde, değerli bir mühendislik deneyimi sunduğunu kanıtlamaktadır.