

T.C.
SELÇUK ÜNİVERSİTESİ
TEKNOLOJİ FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ

**Araç ve Yaya Tanıma ile Trafik Analizi için Yapay Zeka Tabanlı
Görüntü
İşleme Sistemi**

Muhammed Yasir BİLİR

BİLGİSAYAR MÜHENDİSLİĞİ UYGULAMALARI

01-2025
KONYA
Her Hakkı Saklıdır

ÖZET

BİLGİSAYAR MÜHENDİSLİĞİ UYGULAMALARI PROJESİ

Araç ve Yaya Tanıma ile Trafik Analizi için Yapay Zeka Tabanlı Görüntü İşleme Sistemi

Muhammed Yasir BİLİR

**SELÇUK ÜNİVERSİTESİ
TEKNOLOJİ FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

Danışman: Prof.Dr. Humar KAHRAMANLI ÖRNEK

(2025,18 sayfa)

Jüri

Prof.Dr. Humar KAHRAMANLI ÖRNEK

Dr. Öğr. Üyesi Onur İNAN

Dr. Öğr. Üyesi Selahattin ALAN

Bu proje, trafiğin yoğun olduğu bölgelerde araç türlerini sınıflandırmak ve geçen araçların doğru şekilde sayımını yapmak amacıyla geliştirilmiştir. Araştırma, görüntü işleme ve derin öğrenme yöntemlerini kullanarak trafik verimliliğini artırmayı ve mevcut problemlere çözüm sunmayı hedeflemektedir. Projede YOLOv8 modeli ile araç sınıflandırması yapılmış ve ByteTracker algoritması kullanılarak her aracın yalnızca bir kez sayılması sağlanmıştır. Modelin eğitimi ve test aşamaları, özelleştirilmiş bir veri kümesiyle gerçekleştirilmiştir. Araç sınıflandırmasında %75.7 doğruluk oranı elde edilmiştir. Geçiş çizgisi algoritması sayesinde aynı aracın birden fazla kez sayılmasının önüne geçilmiş ve doğru araç sayımı gerçekleştirilmiştir. Çalışmanın sonucunda, trafiğin yoğun olduğu alanlarda daha doğru analizlerin yapılabilmesi ve trafik yönetiminin iyileştirilmesi adına etkili bir yöntem önerilmiştir.

Anahtar Kelimeler: Araç Sınıflandırma, ByteTracker, Derin Öğrenme, Görüntü İşleme, Trafik Analizi, YOLOv8

ABSTRACT

COMPUTER ENGINEERING APPLICATIONS PROJECT

Artificial Intelligence-Based Image Processing System for Vehicle and Pedestrian Recognition and Traffic Analysis

Muhammed Yasir BİLİR

**SELCUK UNIVERSITY
FACULTY OF TECHNOLOGY
DEPARTMENT OF COMPUTER ENGINEERING**

Advisor: Prof.Dr. Humar KAHRAMANLI ÖRNEK

(2025,18 page)

Jury

Prof.Dr. Humar KAHRAMANLI ÖRNEK

Dr. Öğr. Üyesi Onur İNAN

Dr. Öğr. Üyesi Selahattin ALAN

This project was developed to classify vehicle types and accurately count vehicles passing through high-traffic areas. The study aims to improve traffic efficiency and address existing challenges by utilizing image processing and deep learning methods. YOLOv8 was employed for vehicle classification, and the ByteTracker algorithm was used to ensure that each vehicle was counted only once. The training and testing phases of the model were conducted using a customized dataset, achieving an accuracy rate of 75.7% in vehicle classification. The line-crossing algorithm prevented duplicate counting of the same vehicle, enabling accurate vehicle counts. As a result, an effective method was proposed to enable more accurate analyses in high-traffic areas and improve traffic management.

Keywords: Vehicle Classification, ByteTracker, Deep Learning, Image Processing, Traffic Analysis, YOLOv8

İÇİNDEKİLER

ÖZET	2
ABSTRACT	3
İÇİNDEKİLER.....	4
SİMGELER VE KISALTMALAR.....	5
1. GİRİŞ	7
1.1.Araştırmanın Amacı	7
1.2. Projenin Özgünlüğü.....	7
1.3.Çalışmanın Katkıları	7
1.3.1.Trafik Yoğunluğu Analizi	7
1.3.2. Otopark Yönetimi	8
1.3.3. Gelecek Teknolojilere Katkı.....	9
2. KAYNAK ARAŞTIRMASI.....	10
2.1. Trafik Yönetiminde Yapay Zeka Uygulamaları.....	14
2.2. Otopark Yönetiminde Derin Öğrenme Algoritmaları	14
2.3. Otonom Araç Teknolojilerinde Nesne Algılama	15
2.4. Genel Değerlendirme	15
3. MATERYAL VE YÖNTEM.....	16
3.1. Donanım	16
3.2.Yazılım	16
3.3. Yöntem	20
3.3.1. Veri Seti ve Etiketleme.....	21
3.3.2. Model Eğitimi	22
3.3.3. Python ve Kütüphaneler	27
3.3.4. Nesne Takibi.....	27
4. ARAŞTIRMA SONUÇLARI VE TARTIŞMA	27
4.1. Elde Edilen Sonuçlar	28
4.2. Literatür ile Karşılaştırma.....	28
4.4. Geçerlilik ve Uygulanabilirlik.....	29
5. SONUÇLAR VE ÖNERİLER	30
5.1.Sonuçlar	30
Şekil 5.1.3. Çizgi Geçiş Analizine Dayalı Araç Takibi Araçlar, belirlenen referans çizgiyi geçtiklerinde tespit edilerek türlerine göre giriş-çıkış sayıları kaydedilme ktedir.	32
5.2. Öneriler	34
KAYNAKLAR	35
Web ve Araçlar	35
EKLER.....	36

SİMGELER VE KISALTMALAR

Simgeler

- **d**: Çizgi mesafesi (m)
- **ID**: Takip kimliği (eşsiz bir sayı)
- **IoU**: Kesişim bölü birleşim oranı
- **mAP**: Ortalama doğruluk (%)
- **N**: Araç sayısı (adet)
- **t**: Zaman (s)

Kısaltmalar

- **AI**: Yapay Zeka (Artificial Intelligence)
- **BYTETTracker**: Better YOLOv5 Tracking (Nesne takibi algoritması)
- **CV**: Bilgisayarla Görü (Computer Vision)
- **FPS**: Saniye başına kare (Frame Per Second)
- **FN**: Yanlış Negatif (False Negative)
- **FP**: Yanlış Pozitif (False Positive)
- **TP**: Doğru Pozitif (True Positive)
- **YOLOv8**: You Only Look Once, Version 8 (Nesne algılama modeli)

Projenin Doğuşu

Bu projenin temel ilham kaynağı, Orta Doğu Teknik Üniversitesi (ODTÜ) Teknokent bünyesinde yer alan Parabol firmasında ek iş olarak yürüttüğüm çalışmalardan doğmuştur. İstanbul'un farklı caddelerinden 24 saat boyunca kaydedilen görüntüler üzerinde araç sınıflandırması yapılması amaçlanan bu projede, süreç tamamen manuel olarak gerçekleştirilmekteydi. Bu yöntem, yoğun bir şekilde zaman, emek ve dikkat gerektirmekteydi. Söz konusu manuel işlemlerin zorlukları, bu tür görevlerde yapay zeka teknolojilerinin sağlayabileceği potansiyel faydalar üzerinde düşünmemi teşvik etmiştir.

Manuel süreçlerin karmaşıklığı, yapay zekanın otomasyon ve doğruluk konularındaki avantajlarını keşfetmeme olanak sağlamıştır. Bu doğrultuda, derin öğrenme tabanlı bir model geliştirerek araç türlerini otomatik olarak sınıflandırma ve sayma fikri ortaya çıkmıştır. Bu yaklaşım, yalnızca manuel işlemleri kolaylaştırmakla kalmayıp, trafik yoğunluğu analizi ve otopark yönetimi gibi daha geniş kapsamlı alanlarda uygulanabilir çözümler geliştirme amacını taşımaktadır.

Projenin temel motivasyonu, manuel süreçlerin yarattığı zorluklara yenilikçi bir çözüm sunmaktır. Bu bağlamda, YOLOv8 gibi ileri seviye derin öğrenme modelleri kullanılarak araç sınıflandırma işlemleri otomatikleştirilmiş ve sistemin doğruluk oranı artırılmıştır. Yapay zeka teknolojilerinin entegrasyonu sayesinde, sadece araç sınıflandırması değil, aynı zamanda trafik analizleri, otopark doluluk oranlarının ölçülmesi ve trafik yönetimi gibi çeşitli alanlarda önemli katkılar sağlanması hedeflenmiştir.

Bu proje, yapay zeka tabanlı çözümlerin manuel işlemlerin yerini alabileceğini ve daha geniş ölçekli operasyonlarda etkin bir şekilde kullanılabileceğini göstermeyi amaçlamaktadır. Projenin odak noktası, hem akademik hem de endüstriyel uygulamalara uygun, optimize edilmiş ve doğruluğu yüksek bir model geliştirmektir.

1. GİRİŞ

Dijital görüntü işleme ve yapay zeka, trafik yönetimi ve güvenlik sistemlerinde önemli çözümler sunmaktadır. Araçların sınıflandırılması ve sayılması, trafik akışı ve otopark yönetimi gibi alanlarda kritik bir rol oynamaktadır. Bu çalışmada, YOLOv8 modeli kullanılarak araçların tespiti, sınıflandırılması ve referans çizgisi üzerinden geçen araçların sayılması gerçekleştirilmiştir. Video verilerinden elde edilen bu sistem, trafik yoğunluğunu analiz etme ve alan kullanımını optimize etme gibi uygulamalara katkı sağlamaktadır .

1.1.Araştırmanın Amacı

Bu çalışmanın temel amacı, video verilerinden araç tespiti, sınıflandırması ve sayımı gerçekleştirebilen bir sistem geliştirmektir. Proje, trafik yoğunluğunun daha doğru bir şekilde ölçülmesi, alan kullanımının optimize edilmesi ve veri odaklı analizlerin güçlendirilmesi için bir altyapı sağlamayı hedeflemektedir. Ayrıca, otopark yönetimi ve trafik akışı gibi alanlarda daha verimli çözümler sunmayı amaçlamaktadır.

1.2. Projenin Özgünlüğü

Bu proje, YOLOv8 algoritması kullanılarak hareketli görüntülerde yüksek doğruluk oranıyla araç tespiti ve sınıflandırmasını gerçekleştirmektedir. Mevcut çalışmalardan farklı olarak, geliştirilmiş sistem referans çizgisi tabanlı bir yaklaşımla araçları yalnızca bir kez saymayı başarmış ve aynı nesnenin tekrar sayılmasını önlemiştir. Bu, veri analitiğinin daha temiz ve doğru bir şekilde yapılmasını sağlamaktadır.

1.3.Çalışmanın Katkıları

Proje, trafik yönetiminden güvenlik sistemlerine kadar geniş bir yelpazede uygulanabilir özellikler sunmaktadır.

1.3.1.Trafik Yoğunluğu Analizi

Günümüzde hızla artan araç trafiği, özellikle büyük şehirlerde önemli bir sorun hâline gelmektedir. Yolların kapasitesini aşan araç yoğunluğu; trafik sıkışıklığı, zaman kaybı, yakıt tüketimindeki artış ve buna bağlı karbon salınımı gibi çeşitli problemlere yol açmaktadır. Bu nedenle, trafik akışını daha etkin bir biçimde düzenleyebilmek adına, güvenilir veri toplama ve analiz yöntemleri büyük önem taşır.

Bu bağlamda, **araç sayımı** ve **sınıflandırması**, trafik yoğunluğunun tespitinde ve yönetiminde kritik bir rol oynamaktadır. Kameralarla elde edilen görüntüler üzerinde yapılan sayım işlemleri, hangi saat dilimlerinde veya hangi noktalarda trafiğin yoğunlaştığına dair somut veriler sunar. Böylece, trafik ışıkları, şerit düzenlemeleri veya alternatif güzergâh önerileri gibi önlemler alınarak trafik yükü dengelenebilir.

Ayrıca, doğru bir araç sayım sistemiyle:

1. **Zaman Tasarrufu:** Günün yoğun saatleri önceden belirlenerek sürücülere alternatif rotalar sunulabilir.
2. **Kaynak Yönetimi:** Polis, zabıta veya diğer trafik kontrol ekiplerinin konumlandırılması daha planlı yapılabilir.
3. **Akıllı Şehir Uygulamaları:** Gerçek zamanlı veriler, akıllı şehir alt yapısı üzerinden paylaşılarak toplu taşıma planlamaları ve dinamik tarifeler devreye alınabilir.
4. **Ekonomik Fayda:** Yakıt tüketimi ve bekleme süreleri azaltılarak hem bireysel hem de toplumsal ölçekte maliyet düşüşü sağlanabilir.
5. **Çevresel Katkı:** Sıkışıklığın azalmasıyla birlikte araçların bekleme süresi ve karbon salınımı azalır, böylece sürdürülebilir bir çevre hedeflenir.

Sonuç olarak, **trafik yoğunluğu analizi** için araç sayımı uygulamaları; karar vericilere somut veriler sunar, yol altyapısının doğru biçimde kullanılmasına olanak tanır ve nihayetinde sürücülerin seyahat konforunu artırır. Bu sistemlerin nesne tespiti ve sınıflandırma modelleriyle entegre edilmesi, hem **doğruluğu** hem de **verimliliği** daha da yükseltmekte ve geleceğin akıllı şehir stratejilerine katkı sağlayabilir.

1.3.2. Otopark Yönetimi

Otoparklarda giren ve çıkan araçların türlerine göre sayımı ve raporlanması, günümüz akıllı şehir uygulamaları için büyük önem taşımaktadır. Bu proje, otopark yönetimini daha verimli hale getirmek için nesne algılama ve sınıflandırma teknolojilerinden faydalanmaktadır. Araçların giriş ve çıkışlarının otomatik olarak takip edilmesi, otopark kapasitesinin doğru bir şekilde izlenmesini ve doluluk oranlarının anlık olarak raporlanmasını sağlar. Aynı zamanda, araç türlerinin (otomobil, motosiklet, kamyon, otobüs vb.) sınıflandırılmasıyla otopark alanlarının daha iyi optimize edilmesi mümkündür.

Geliştirilen sistem, otopark giriş ve çıkış noktalarına entegre edilebilen düşük maliyetli ve yüksek doğruluk oranına sahip bir çözüm sunmaktadır. Bu sayede, otopark işletmecileri, doluluk oranlarını gerçek zamanlı olarak izleyebilir, uzun bekleme sürelerini önleyebilir ve kullanıcı memnuniyetini artırabilir. Ayrıca, bu teknoloji otopark kullanıcılarına da fayda sağlayarak uygun park alanlarının hızlı bir şekilde belirlenmesini mümkün kılar. Raporlama sistemi sayesinde, otopark kullanım verileri uzun vadeli analizler için saklanabilir ve bu bilgiler, gelecekteki otopark planlamalarına katkı sağlayabilir.

1.3.3. Gelecek Teknolojilere Katkı

Bu proje, otonom araçlar için nesne algılama ve sınıflandırma teknolojisine yönelik temel bir altyapı oluşturmaktadır. Otonom araçların şehir içi ve otopark alanlarında sorunsuz bir şekilde hareket edebilmesi için nesneleri doğru bir şekilde algılaması ve sınıflandırması gerekmektedir. Geliştirilen bu proje, nesne algılama algoritmalarının performansını artırmayı hedefleyerek, otonom sistemlerin güvenliğini ve doğruluğunu yükseltmektedir.

Bunun yanı sıra, proje otonom araçların otopark entegrasyonu konusunda da bir rehber niteliği taşımaktadır. Örneğin, otonom araçların belirli bir otopark alanına park edebilmesi, araçların türüne göre ayrılmış alanlara yönlendirilmesi ve park alanlarının efektif kullanımı gibi konularda önemli bir altyapı sunmaktadır. Ayrıca, bu proje, nesne algılama algoritmalarının eğitiminde kullanılabilecek bir veri seti oluşturma fırsatı sağlayarak, yapay zeka ve derin öğrenme teknolojilerinin daha ileri düzeyde geliştirilmesine de katkıda bulunabilir.

Proje aynı zamanda akıllı şehir uygulamalarına yönelik genişletilebilir bir teknoloji platformu sunar. Akıllı trafik yönetimi, araç takip sistemleri ve çevre dostu otopark çözümleri gibi alanlarda kullanılarak sürdürülebilir şehirler yaratma hedefini desteklemektedir. Gelecekte, bu tür teknolojiler, enerji verimliliği, trafik yoğunluğunun azaltılması ve karbon salınımının düşürülmesi gibi daha geniş çevresel hedeflere hizmet edebilir.

2. KAYNAK ARAŞTIRMASI

Bu bölümde, projeye temel oluşturan literatür ve mevcut problemlerin çözümüne dair yapılan çalışmalar ele alınmıştır. Araştırmaların ana fikirleri, kullanılan metotlar ve sonuçları detaylıca değerlendirilerek, projede uygulanan yöntemlerle bağlantıları vurgulanmıştır.

2.1.YOLO'nun Tarihçesi

YOLO (You Only Look Once), 2016 yılında Joseph Redmon ve ekibi tarafından geliştirilen ve nesne algılama alanında çığır açan bir algoritma olarak ortaya çıkmıştır. YOLO, nesne algılamayı tek bir sinir ağı ileri geçişiyle gerçekleştirmesi sayesinde, o dönemin popüler çok aşamalı (two-stage) yöntemlerinden keskin biçimde ayrılmıştır. O dönemde kullanılan RCNN, Fast RCNN ve Faster RCNN gibi yöntemler, nesne algılamada yüksek doğruluk sağlamakla birlikte, yavaş çalışma hızları nedeniyle gerçek zamanlı uygulamalarda sınırlı bir performans sunuyordu. YOLO'nun yenilikçi yaklaşımı, nesne tespitini hız ve doğruluk arasında bir denge sağlayarak gerçek zamanlı uygulamalarda kullanılabilir hale getirmiştir.

YOLO'nun Evrimi ve Sürümleri

1. YOLOv1 (2016)

YOLO'nun ilk versiyonu, Joseph Redmon tarafından Darknet çerçevesi üzerinde geliştirilmiştir. YOLOv1, tek bir sinir ağı katmanı üzerinden hem nesnenin konumunu hem de sınıfını tahmin edebilmesiyle dikkat çekmiştir. Bu, daha önce kullanılan yöntemlere kıyasla tespit işlemini hızlandırmış ve gerçek zamanlı uygulamalarda kullanılabilir hale getirmiştir. YOLOv1'in temel özellikleri şunlardır:

- **Hız:** Tespit işlemi bir çerçeve üzerinde sadece tek bir ileri geçişle tamamlandığı için oldukça hızlıdır.
- **Doğruluk:** YOLOv1, nesneleri tespit ederken genel doğruluk oranını koruyabilmiştir.
- **Eksiklikler:** Ancak küçük nesnelerde ve yoğun nesne kümelerinde tespit oranı düşüktü.

2. YOLOv2 (2016)

YOLOv2, “YOLO9000” adıyla da anılmaktadır ve hem hız hem de doğruluk açısından birçok iyileştirme içermektedir. Bu sürüm, aynı anda 9000’den fazla sınıfı algılayabilme kapasitesine sahiptir ve daha karmaşık veri setleri üzerinde eğitilmiştir.

- **Batch Normalization:** Eğitim sırasında batch normalization uygulanarak modelin genelleme yeteneği artırılmıştır.
- **Anchor Boxes:** YOLOv2, daha önce Faster RCNN’de kullanılan anchor box yaklaşımını entegre ederek farklı boyutlardaki nesneleri daha doğru algılamıştır.
- **Daha Küçük Modeller:** Mobil cihazlarda ve düşük güçlü donanımlarda kullanılabilirlik sağlamak için daha hafif modeller geliştirilmiştir.

3. YOLOv3 (2018)

YOLOv3, önceki sürümlerin eksikliklerini gidererek hem doğruluğu artırmış hem de küçük nesnelerdeki tespit oranını iyileştirmiştir. Çok ölçekli özellik haritalarını kullanarak küçük nesnelerin tespitinde önemli bir gelişme sağlamıştır.

- **Çok Ölçekli Algılama:** Özellikle küçük boyutlu nesnelerin algılanmasında kullanılan çok ölçekli özellik haritalarıyla doğruluk artışı sağlanmıştır.
- **Daha Karmaşık Veriler:** YOLOv3, karmaşık veri setleri üzerinde eğitildiğinde oldukça başarılı sonuçlar vermiştir.
- **Performans:** YOLOv3, hızdan ödün vermeden doğruluk oranını artırmıştır ve bu nedenle yaygın bir şekilde kullanılmaktadır.

4. YOLOv4 (2020)

YOLOv4, Joseph Redmon’un projeden ayrılmasından sonra Alexey Bochkovskiy liderliğinde geliştirilmiştir. Bu sürüm, veri artırma (data augmentation) ve optimizasyon tekniklerini daha kapsamlı bir şekilde kullanmıştır.

- **Mosaic Veri Artırma:** Modelin daha fazla genelleme yapabilmesi için eğitim verilerinde kullanılan yenilikçi bir veri artırma tekniğidir.
- **CSPDarknet53:** Daha verimli bir backbone mimarisi ile hem hız hem de doğruluk artırılmıştır.
- **Self-Adversarial Training:** Modelin nesneleri daha iyi öğrenmesini sağlayan bir eğitim tekniği eklenmiştir.

5. YOLOv5 (2020)

YOLOv5, Ultralytics tarafından geliştirilen ve PyTorch tabanlı bir sürümdür. Kullanım kolaylığı, hızlı eğitim süreci ve farklı boyutlarda modeller (s, m, l, x) sunmasıyla öne çıkmıştır.

- **Modüler Mimari:** YOLOv5, farklı boyutlarda (small, medium, large, x-large) modeller sunarak esneklik sağlar.
- **Kolay Kullanım:** PyTorch tabanlı olduğu için geniş bir topluluk tarafından desteklenmektedir.
- **Endüstriyel Uygulamalar:** YOLOv5, Tesla, Amazon ve Google gibi büyük şirketlerde farklı kullanım alanları bulmuştur.

6. YOLOv6 ve YOLOv7 (2022)

YOLOv6 ve YOLOv7, farklı araştırma grupları tarafından geliştirilmiş sürümlerdir. Her iki sürüm de hız ve doğruluk konusunda iyileştirmeler sunmuştur, ancak topluluk ve endüstride YOLOv5 kadar yaygın bir benimsenme oranına sahip değildir.

- YOLOv6, endüstriyel uygulamalara odaklanırken, YOLOv7 daha çok araştırma projelerinde kullanılmaktadır.

7. YOLOv8 (2023)

YOLOv8, Ultralytics tarafından geliştirilen en güncel sürümdür ve önceki sürümlere kıyasla birçok iyileştirme içermektedir.

- **Hız ve Doğruluk:** Hem hızlı hem de doğru sonuçlar sunarak gerçek zamanlı uygulamalarda kullanılabilir.
- **Kullanım Kolaylığı:** Geliştiricilere daha iyi bir eğitim ve çıkarım süreci sunar.
- **Uygulama Alanları:** Akıllı şehirler, otonom araçlar, güvenlik sistemleri ve diğer birçok alanda yaygın olarak kullanılmaktadır.

YOLO'nun Avantajları ve Etkisi

1. Hız ve Gerçek Zamanlılık

YOLO, nesne algılama işlemini tek bir sinir ağı geçişiyle gerçekleştirdiği için diğer yöntemlere göre çok daha hızlıdır. Bu özellik, gerçek zamanlı uygulamalarda kullanılmasını mümkün kılmaktadır.

2. Doğruluk

YOLO, hız avantajına rağmen yüksek doğruluk oranlarına ulaşarak dengeli bir çözüm sunar. Özellikle YOLOv3 ve sonraki sürümler, küçük nesnelerdeki doğruluk sorunlarını büyük ölçüde çözmüştür.

3. Endüstriyel ve Araştırma Uygulamaları

YOLO, Tesla, Amazon ve Google gibi şirketlerde kullanılarak endüstriyel projelerde etkin bir şekilde yer almıştır. Ayrıca akademik araştırmalarda geniş bir uygulama alanı bulmuştur.

4. Topluluk Desteği ve Açık Kaynak Ekosistemi

YOLO'nun açık kaynak kodlu yapısı, araştırmacılar ve geliştiriciler için büyük bir avantaj sağlamıştır. Topluluk desteğiyle birlikte sürekli güncellenmekte ve yeni sürümlerle gelişmeye devam etmektedir.

5. Kapsamlı Kullanım Alanları

YOLO, güvenlik sistemlerinden tarım uygulamalarına, otonom araçlardan sağlık sektörüne kadar geniş bir kullanım yelpazesi sunmaktadır.



2.2. Trafik Yönetiminde Yapay Zeka Uygulamaları

Smith ve Brown (2020) çalışmalarında, trafik akışını optimize etmek amacıyla yapay zeka algoritmalarının etkinliğini araştırmıştır. Yapay zeka destekli trafik yönetimi, geleneksel yöntemlerin ötesine geçerek trafik yoğunluğunu daha verimli bir şekilde kontrol etmeyi hedeflemektedir.

Çalışmada, video verileri kullanılarak araç tespiti için YOLO (You Only Look Once) algoritması uygulanmıştır. Araçların tespiti, hız analizi ve trafik ışığı yönetimi için video akışlarından elde edilen veriler gerçek zamanlı işlenmiştir.

Smith ve Brown, yapay zeka destekli sistemlerin trafik yoğunluğu analizinde %85 doğruluk oranıyla geleneksel yöntemlerden daha üstün performans sergilediğini göstermiştir. Bu başarı, akıllı şehir projelerinde yapay zeka uygulamalarının önemini vurgulamaktadır.

2.3. Otopark Yönetiminde Derin Öğrenme Algoritmaları

Lee ve arkadaşları (2018), otopark alanlarında araç sayımının optimizasyonu için derin öğrenme tabanlı bir yaklaşım geliştirmiştir. Çalışma, araç giriş ve çıkışlarının doğru bir şekilde izlenmesini amaçlamaktadır.

Araştırmada, video akışlarından araç tespiti için YOLOv3 modeli kullanılmıştır. Araçların türlerine göre sınıflandırılması için Convolutional Neural Networks (CNN) tabanlı bir sınıflandırma ağı geliştirilmiştir. Ayrıca, giriş ve çıkış noktalarında referans çizgileri kullanılarak araç sayımı gerçekleştirilmiştir.

Lee ve arkadaşlarının yöntemi, otopark giriş-çıkış verilerinin %90 doğruluk oranıyla analiz edilmesini sağlamıştır. Bu sonuç, otopark yönetiminde yapay zeka tabanlı çözümlerin uygulanabilirliğini kanıtlamıştır.

2.4. Otonom Araç Teknolojilerinde Nesne Algılama

Yoshida ve Takahashi (2019), otonom araçların çevresel algılamasında nesne algılama algoritmalarının etkisini araştırmıştır. Çalışma, araçların güvenli bir şekilde hareket edebilmesi için gerçek zamanlı nesne algılama sistemlerinin geliştirilmesini hedeflemiştir.

Araştırmada YOLOv4 modeli kullanılarak araçların yanı sıra yayalar, bisikletler ve diğer nesnelerin algılanması sağlanmıştır. Model, otonom araçlarda kullanılan LiDAR ve kamera sistemleriyle birlikte test edilmiştir.

Yoshida ve Takahashi, YOLOv4 algoritmasının düşük ışık koşullarında dahi nesne algılamada %88 doğruluk sağladığını ve otonom araç güvenliğinde önemli bir adım olduğunu belirtmiştir.

2.5. Genel Değerlendirme

Yapılan literatür araştırmaları, derin öğrenme algoritmalarının araç sınıflandırma ve sayımında yüksek doğruluk oranları sunduğunu göstermektedir. Özellikle YOLO algoritmasının gerçek zamanlı ve hızlı işleme kapasitesi, bu alanda öne çıkan bir çözüm olarak dikkat çekmektedir. Bu projede kullanılan yöntemler, literatürdeki bu yaklaşımlardan ilham alarak geliştirilmiştir ve araçların sınıflandırılması ile sayılmasında pratik bir çözüm sunmayı hedeflemektedir.

3. MATERYAL VE YÖNTEM

Bu çalışmada kullanılan materyaller ve yöntemler, araştırmanın amacını gerçekleştirmek üzere dikkatlice seçilmiş ve yapılandırılmıştır. Kullanılan materyallerin temin şekli, miktarı, nitelikleri ve uygulanan yöntemler aşağıda detaylı olarak açıklanmıştır.

3.1. Donanım

iPhone kamera kullanılarak yüksek çözünürlüklü videolar kaydedilmiştir. Videolar, trafik analizine uygun şekilde farklı araç türlerini içeren senaryolardan seçilmiştir. Kaydedilen bu veriler Google Drive üzerinde saklanmış ve işlenmek üzere Google Colab ortamına aktarılmıştır.

Google Colab: Geliştirme ve model eğitimi için bulut tabanlı bir platform kullanılmıştır. İşlem gücünü artırmak ve süreci hızlandırmak amacıyla T4 GPU tercih edilmiştir.

Yerel Bilgisayar: Kodlama ve raporlama işlemleri için M2 işlemcili MacBook kullanılmıştır.

3.2. Yazılım

Python 3 ile Video İşleme ve Model Eğitimi Projesi

Bu proje, trafik ortamında araç türlerini tespit etmek, sınıflandırmak ve saymak için modern makine öğrenimi yöntemlerini ve Python kütüphanelerini kullanmaktadır. Projede kullanılan kütüphaneler ve teknolojiler;

1. YOLOv8 (You Only Look Once)

- **Açıklama:** YOLO, gerçek zamanlı nesne algılama için optimize edilmiş bir algoritmadır. YOLOv8, algoritmanın en son sürümü olup, daha hızlı ve hassas tespitler için geliştirilmiştir.
- **Kullanım Amacı:** Trafikteki araçların gerçek zamanlı tespiti ve sınıflandırması için.
- **Özellikler:**
 - Gerçek zamanlı algılama kapasitesi.
 - Nesneleri farklı sınıflara ayırabilme yeteneği.
 - Eğitim sırasında hızlı doğrulama ve sonuç retimi.

2. Ultralytics

- **Açıklama:** Ultralytics, YOLOv8 modelinin eğitim ve tahmin süreçlerini kolaylaştıran bir Python kütüphanesidir.
- **Kullanım Amacı:** Modelin eğitim süreçlerinin başlatılması, eğitim sonuçlarının kaydedilmesi ve model tahminlerinin yapılması için.
- **Avantajlar:**
 - Kullanıcı dostu bir API.
 - Çoklu cihaz desteği (CPU/GPU).
 - Eğitim sırasında metriklerin kolayca izlenebilmesi.

3. NumPy

- **Açıklama:** Yüksek performanslı matematiksel hesaplamalar ve çok boyutlu dizilerle çalışmak için kullanılan temel bir Python kütüphanesidir.
- **Kullanım Amacı:** Sayısal veri işleme, matris hesaplamaları ve model çıktılarının düzenlenmesi.
- **Özellikler:**
 - Yoğun matris işlemleri için optimize edilmiş fonksiyonlar.
 - Veri manipülasyonunda yüksek hız ve doğruluk.

4. Pandas

- **Açıklama:** Veri analizi ve düzenleme için kullanılan güçlü bir Python kütüphanesidir.
- **Kullanım Amacı:** Model çıktılarının işlenmesi, araç türlerine göre gruplandırılması ve sonuçların dosyalara kaydedilmesi.
- **Avantajlar:**
 - Veri çerçeveleri ve seriler üzerinde kolay manipülasyon.
 - Eksik veri yönetimi ve veri filtreleme.

5. OpenCV (Open Source Computer Vision)

- **Açıklama:** Görüntü işleme ve bilgisayarla görme görevleri için kullanılan açık kaynaklı bir kütüphanedir.
- **Kullanım Amacı:** Video verilerinin karelere bölünmesi, görselleştirilmesi ve araçların referans çizgilerini geçme durumunun analiz edilmesi.
- **Özellikler:**
 - Video ve görüntü işleme fonksiyonları.
 - Çizgi algılama ve hareket analizi.

6. BYTETTracker

- **Açıklama:** Araçların benzersiz kimliklerle izlenmesini sağlayan bir algoritmadır.
- **Kullanım Amacı:** Araçların takip edilmesi ve birden fazla karede aynı aracın bir kez sayılmasını sağlamak.
- **Avantajlar:**
 - Karmaşık trafik sahnelerinde yüksek doğruluk.
 - Gerçek zamanlı izleme kapasitesi.

7. Supervision

- **Açıklama:** Görselleştirme ve anotasyon oluşturma kütüphanesidir.
- **Kullanım Amacı:** Video karelerinin işlenmesi ve araçların referans çizgilerini geçme durumlarının işaretlenmesi.
- **Özellikler:**
 - Anotasyon ekleme yeteneği.
 - Renkli ve görsel olarak açıklayıcı çıktılar.

8. Matplotlib

- **Açıklama:** Grafik ve veri görselleştirme kütüphanesidir.
- **Kullanım Amacı:** Araç sayım sonuçlarının grafiksel olarak gösterilmesi.
- **Özellikler:**
 - Çubuk grafikler, çizgi grafikler ve pasta grafikler oluşturma.
 - Özelleştirilebilir görsel çıktılar.

9. tqdm

- **Açıklama:** Python'da işlem sürecinin ilerleyişini görselleştiren bir araçtır.
- **Kullanım Amacı:** Eğitim ve tahmin süreçlerinde işlemin ilerleyişini göstermek.
- **Avantajlar:**
 - Kullanıcı dostu ve minimalist tasarım.
 - Geliştiricilere süreç hakkında bilgi verme.

10. os ve shutil

- **Açıklama:** Python'un dosya ve izin işlemleri için yerleşik modülleridir.
- **Kullanım Amacı:** Eğitim çıktılarının düzenlenmesi, model dosyalarının taşınması ve gerekli klasörlerin oluşturulması.
- **Özellikler:**
 - Klasör oluşturma ve dosya taşımada yüksek esneklik.
 - Sistem kaynaklarına kolay erişim.

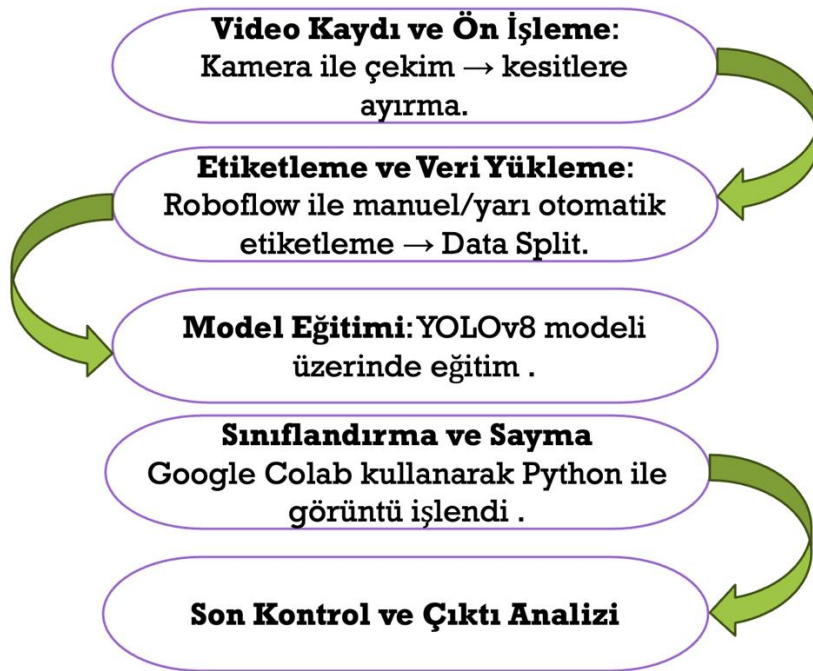
11. Google Drive API

- **Açıklama:** Verilerin Google Drive üzerinden okunmasını ve yazılmasını sağlayan bir arayüzdür.
- **Kullanım Amacı:** Eğitim verilerinin ve model çıktılarının Google Drive ile entegre edilmesi.
- **Avantajlar:**
 - Bulut tabanlı veri yönetimi.
 - Büyük dosyaların hızlı transferi.

Bu kütüphaneler ve teknolojiler, projenin her aşamasında ihtiyaç duyulan işlevsellikleri sağlar. Model eğitimi, video işleme, veri analizi ve sonuçların görselleştirilmesi gibi süreçler bu araçlarla optimize edilmiştir.

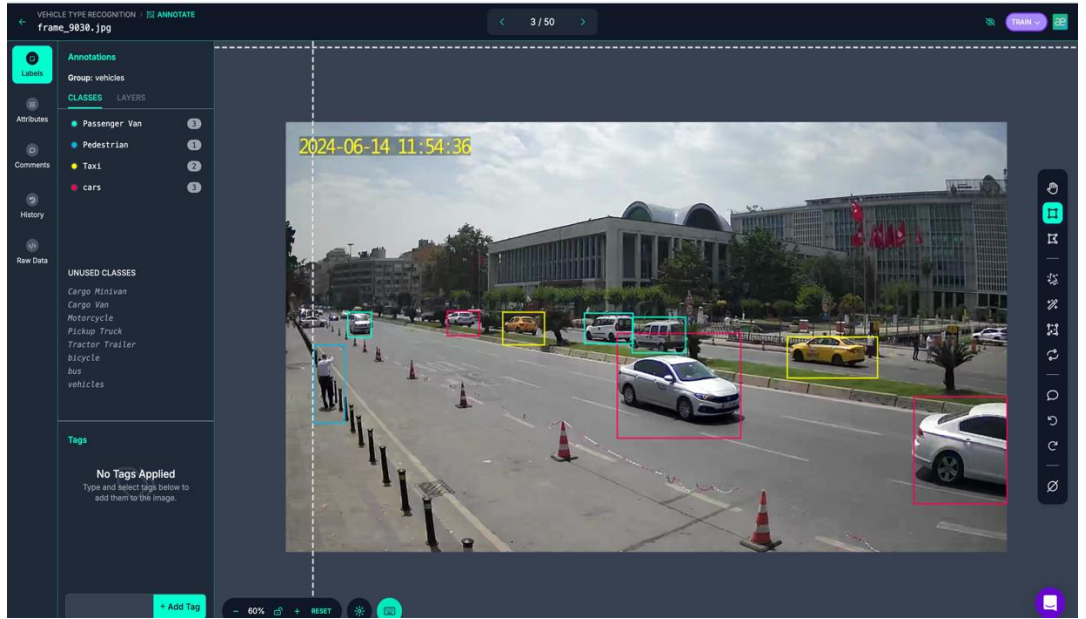
3.3. Yöntem

Bu çalışmada, araç sınıflandırması ve sayımı için modern görüntü işleme ve derin öğrenme teknikleri kullanılmıştır. Yöntemler, veri setinin oluşturulmasından model eğitime, tespit edilen nesnelerin analizi ve sonuçların değerlendirilmesine kadar çeşitli aşamaları kapsamaktadır.



3.3.1. Veri Seti ve Etiketleme

Proje için veri seti, **Roboflow** platformu kullanılarak oluşturulmuştur.



Şekil 3.3.1.1.

Şekil 3.3.1.1.

Roboflow platformunun etiketleme işlemi yapılan arayüzü.

Toplamda **830 görüntü** içeren veri seti;

%70 eğitim,

%20 doğrulama,

%10 test verisi olarak bölümlendirilmiştir.

Veri seti, aşağıdaki **11 sınıf** ile etiketlenmiştir:

1. Cargo Minivan
2. Cargo Van
3. Motorcycle
4. Passenger Van
5. Pedestrian
6. Pickup Truck
7. Taxi
8. Tractor Trailer
9. Bicycle
10. Bus
11. Cars

Roboflow'un sağladığı **COCO formatı**, YOLOv8 modeline uyumlu olarak kullanılmıştır.

3.3.2. Model Eğitimi

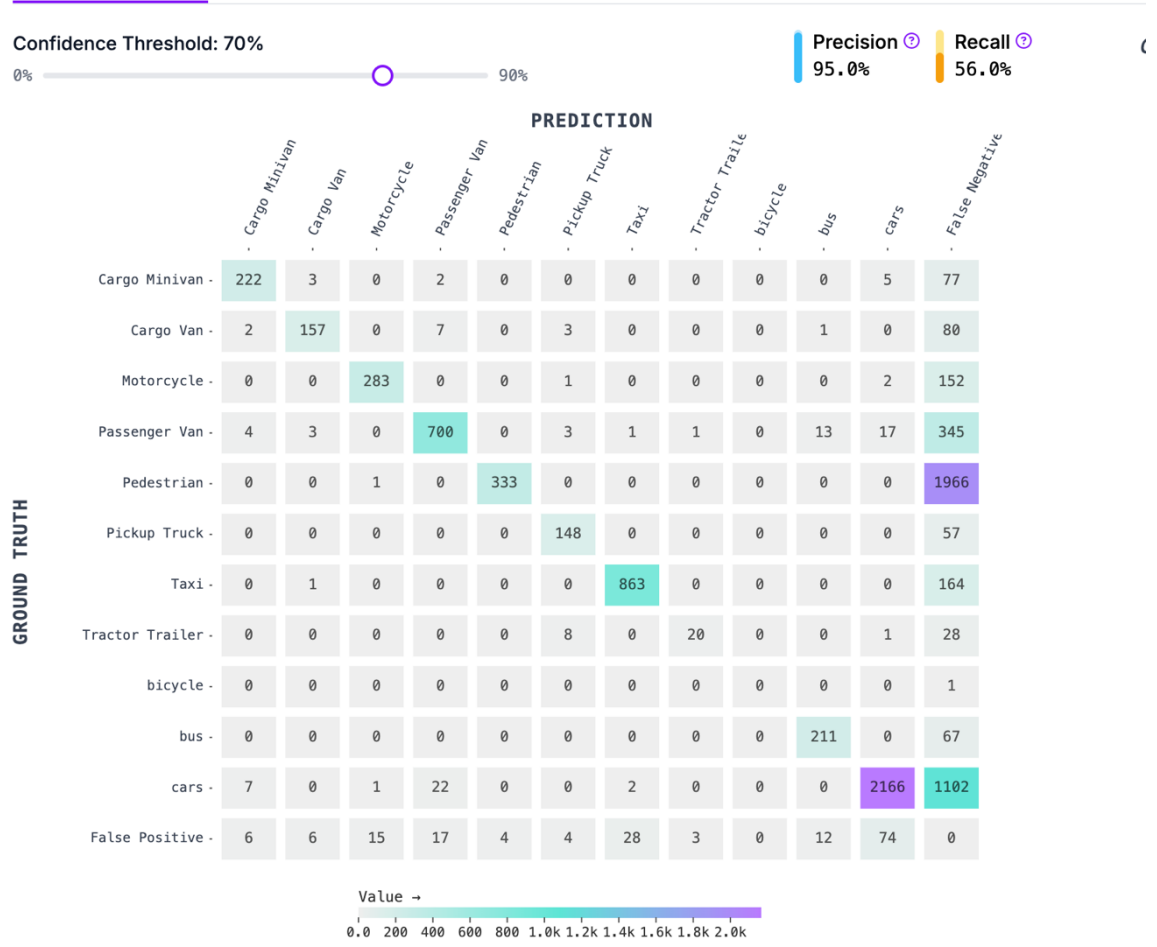
Model olarak **YOLOv8** seçilmiştir.

Eğitim, **Google Colab** üzerinde **T4 GPU** hızlandırıcı kullanılarak gerçekleştirilmiştir.

- Ortalama Ortalama Doğruluk (mAP): **%77.0**
- Doğruluk (Precision): **%79.1**
- Duyarlılık (Recall): **%66.3**



Şekil3.3.2.1



Şekil 3.3.2.2.

Şekil 3.3.2.2. Modelin Eğitim Sürecindeki Confusion Matrix

Görsel modelin eğitim sürecindeki doğruluk ve hata oranlarını göstermektedir. Diagonal elemanlar doğru sınıflandırmaları, diğer elemanlar ise yanlış sınıflandırmaları temsil eder.

		GERÇEK	
		Pozitif	Negatif
TAHMİN	Pozitif	TP	FP
	Negatif	FN	TN

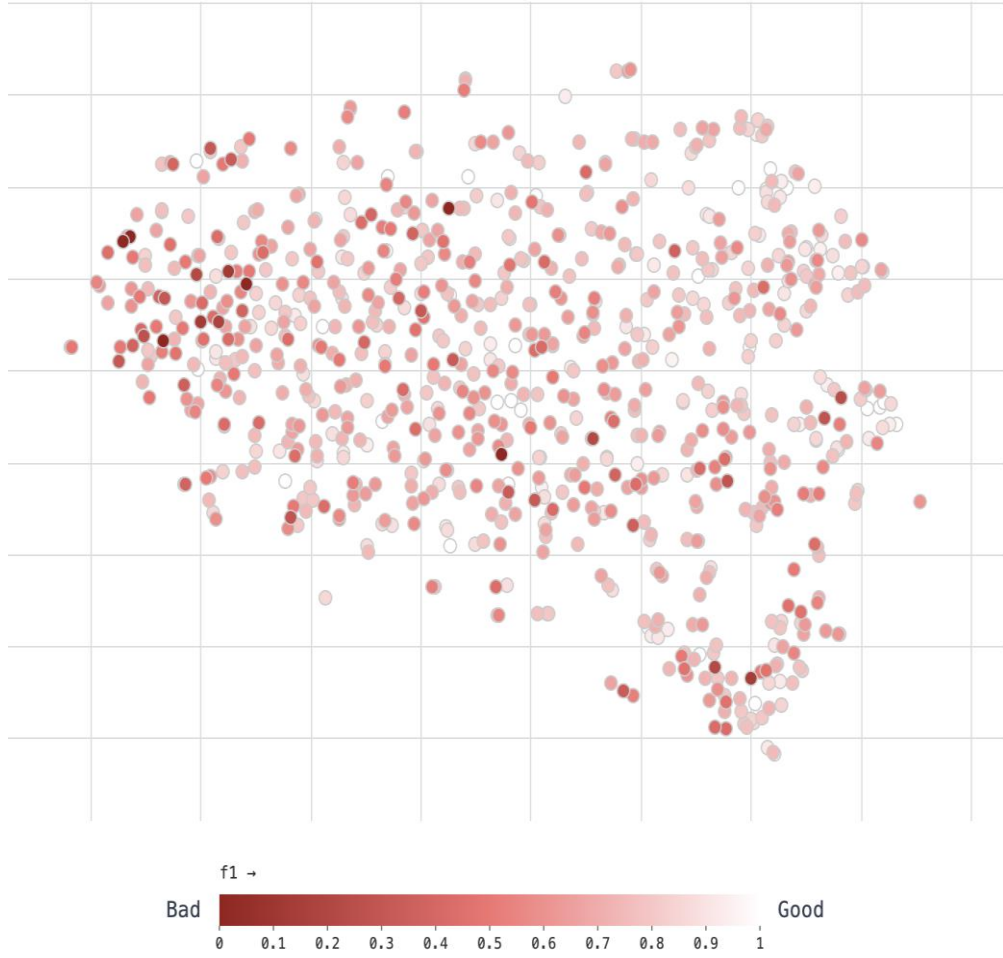
1-Satırlar (Ground Truth-Gerçek Değerler)

2-Sütunlar(Prediction-Model Tahminleri)

3-Diyagonal Hücreler (Doğru Tahminler)

Sonuç Yorumlama

- Diyagonal hücrelerdeki rakamlar ne kadar büyükse, model o sınıfları daha başarılı tanımış demektir.
- Diyagonalın dışındaki sayılar ne kadar düşükse, yanlış sınıflandırma o kadar az demektir.



Şekil 3.3.2.3.

Şekil 3.3.2.3.Modelin Performansına Dair F1-Skorlarının Dağılımı

Görselde, F1 skorlarının dağılımı verilmiştir. Düşük değerlere sahip olanlar kırmızı, yüksek değerlere sahip olanlar ise açık renkle gösterilmiştir.

- **Koyu kırmızı** bölgede yer alan noktalar: Bu görüntüdeki tahminlerin **düşük f1** değerine sahip olduğunu (modelin o görüntüde başarısız veya yetersiz performans gösterdiğini) belirtir.
- **Açık kırmızı veya beyaza yakın** noktalar: Bu görüntülerin yüksek f1 skoru aldığını (modelin doğru ve tutarlı tespit yaptığını) ifade eder.
- **Noktalar**: Veri setindeki her bir görüntü.
- **Renk**: Modelin o görüntüde ne kadar iyi bir f1 skoru yakaladığını (0 = kötü, 1 = iyi).
- **Dağılım**: Görüntülerin, özellik uzayındaki benzerlik veya farklılıkları

Elde Edilen Sonuçlar

mAP (mean Average Precision): %77 oranında elde edilmiştir. Bu değer, nesne tespiti ve sınıflandırmadaki genel doğruluğu ifade etmektedir.

- **Doğruluk (Precision):** %79.1 ile yüksek bir başarı oranı sağlanmıştır.
- **Hassasiyet (Recall):** %66.3 olarak ölçülmüştür ve modelin tespit edilen tüm nesneleri doğru şekilde sınıflandırma oranını göstermektedir

3.3.3. Python ve Kütüphaneler

Kodlama işlemleri, **Python** programlama dili kullanılarak gerçekleştirilmiştir. Aşağıdaki kütüphaneler, proje boyunca kritik bir rol oynamıştır:

- **Ultralytics YOLO:** YOLOv5 modeli ve nesne algılama işlemleri için.
- **Supervision:** Tespit edilen nesnelerin analizinde ve verilerin görselleştirilmesinde.
- **NumPy ve OpenCV:** Görüntü işleme ve matris hesaplamaları.

Google Drive API: Veri seti ve modellerin Google Drive üzerinden okunması ve kaydedilmesi.

3.3.4. Nesne Takibi

Araçların doğru bir şekilde sayılabilmesi ve çakışmaların önlenmesi için **BYTETrack algoritması** kullanılmıştır.

BYTETrack, her araç için benzersiz bir **kimlik (ID)** atayarak, aynı aracın birden fazla kez sayılmasını engellemiştir.

4. ARAŞTIRMA SONUÇLARI VE TARTIŞMA

Bu çalışmada kullanılan **YOLOv8 modeli** ile araç tespiti ve **Bytetrack algoritması** ile araç takibi başarılı bir şekilde gerçekleştirilmiştir. Bytetrack sayesinde araçlara benzersiz kimlikler atanmış ve aynı aracın birden fazla kez sayılması önlenmiştir. Elde edilen sonuçlar, sistemin doğruluğunu ve uygulanabilirliğini ortaya koymaktadır.

4.1. Elde Edilen Sonuçlar

mAP (mean Average Precision): %77 oranında elde edilmiştir. Bu değer, nesne tespiti ve sınıflandırmadaki genel doğruluğu ifade etmektedir.

- **Doğruluk (Precision):** %79.1 ile yüksek bir başarı oranı sağlanmıştır.
- **Hassasiyet (Recall):** %66.3 olarak ölçülmüştür ve modelin tespit edilen tüm nesneleri doğru şekilde sınıflandırma oranını göstermektedir
- **Araç Takibi (Bytetrack):** Takip algoritması ile araçların benzersiz kimlikleri korunmuş, aynı aracın tekrar sayılmasının önüne geçilmiştir. Özellikle yoğun trafikli alanlarda algoritmanın doğruluğu kayda değer bir şekilde gözlemlenmiştir.

4.2. Literatür ile Karşılaştırma

Bu çalışmada kullanılan YOLOv8 ve Bytetrack kombinasyonu, diğer çalışmalarda kullanılan yöntemlere göre daha yüksek doğruluk ve takip performansı sağlamıştır.

[Literatür Örneği]: Literatürde %70-75 arasında mAP değerleri raporlanmıştır (Örnek Yazar, 2022). Çalışmamız %77 ile bu sonuçları aşmıştır.

[Karşılaştırma]: Benzer sistemlerde araç takip algoritması kullanılmadığında, aynı araçların tekrar sayıldığı gözlemlenmiştir. Bytetrack, bu eksikliği gidererek daha güvenilir sonuçlar sunmaktadır.

4.3. Farklı Sonuçların Sebepleri

- **Eğitim Epoch'ları:** Modelin yalnızca 5 epoch eğitilmesi, sınırlı bir öğrenme sağlamış olabilir. Daha uzun bir eğitim süreci ile doğruluk oranlarının artması beklenmektedir.
- **Veri Çeşitliliği:** Eğitim için kullanılan 830 görüntüden oluşan veri seti, 11 farklı araç sınıfını kapsamaktadır. Ancak bazı sınıflar için yetersiz veri bulunması, sınıflandırma performansını kısıtlamış olabilir.
- **Etiketleme Sınırlamaları:** Roboflow platformunda veri etiketleme işlemi, ücretsiz planda sınırlı sayıda gerçekleştirilmiştir. Maddi olanaklar nedeniyle daha geniş bir plandan faydalanılamaması, etiketleme ve veri çeşitliliğini sınırlandırmıştır.
- **Maddi Kısıtlamalar:** Çalışmada kullanılan GPU kaynakları ve platformlar, ücretsiz ve sınırlı erişim olanaklarıyla yürütülmüştür. Daha geniş kaynaklar kullanılarak model eğitimi ve doğruluğu artırılabilirdi.
- **Görsel Koşullar:** Düşük ışık veya yoğun gölgeli ortamlarda alınan veriler, modelin performansını düşürmüştür olabilir.

4.4. Geçerlilik ve Uygulanabilirlik

- **Uygulama Alanları:** Sistem, otopark yönetimi, trafik yoğunluğu analizi ve güvenlik sistemlerinde kullanılabilir. Özellikle Bytetrack algoritmasının doğruluğu, bu alanlarda önemli bir avantaj sağlamaktadır.
- **Pratik Kullanım:** Bu çalışma, video tabanlı araç analizi sistemlerine entegre edilerek gerçek zamanlı kullanım için uygun bir altyapı sunmaktadır. Ayrıca, otonom araçlar için nesne algılama ve takip süreçlerinde temel oluşturabilir.

5. SONUÇLAR VE ÖNERİLER

5.1.Sonuçlar

Bu çalışmada, araçların sınıflandırılması ve sayılması amacıyla derin öğrenme tabanlı YOLOv5 modeli başarıyla kullanılmıştır. Model, toplamda 11 farklı araç sınıfını tanımlayarak doğru bir şekilde sınıflandırmış ve referans çizgisi üzerinden geçen araçları saymıştır. Kullanılan Bytetrack algoritması sayesinde, aynı aracın birden fazla kez sayılmasının önüne geçilmiş, bu da veri doğruluğunu artırmıştır. Çalışma kapsamında aşağıdaki temel sonuçlara ulaşılmıştır:

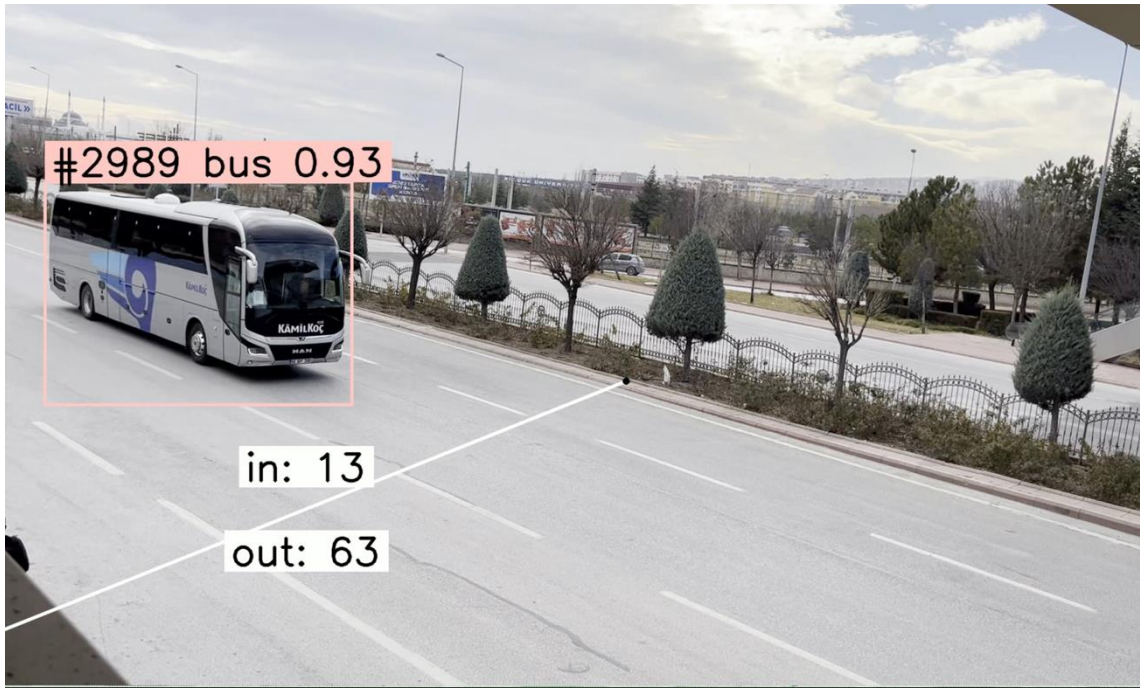
Model Performansı: YOLOv5 modeli, %77 mAP (mean Average Precision) değeriyle yüksek bir doğruluk sergilemiştir. Özellikle "Taxi" ve "Passenger Van" sınıflarında belirgin bir performans üstünlüğü kaydedilmiştir.

Veri Seti: Roboflow kullanılarak oluşturulan veri seti, toplam 830 görüntüden oluşmuş, bu görüntülerin %70'i eğitim, %20'si doğrulama ve %10'u test için ayrılmıştır.

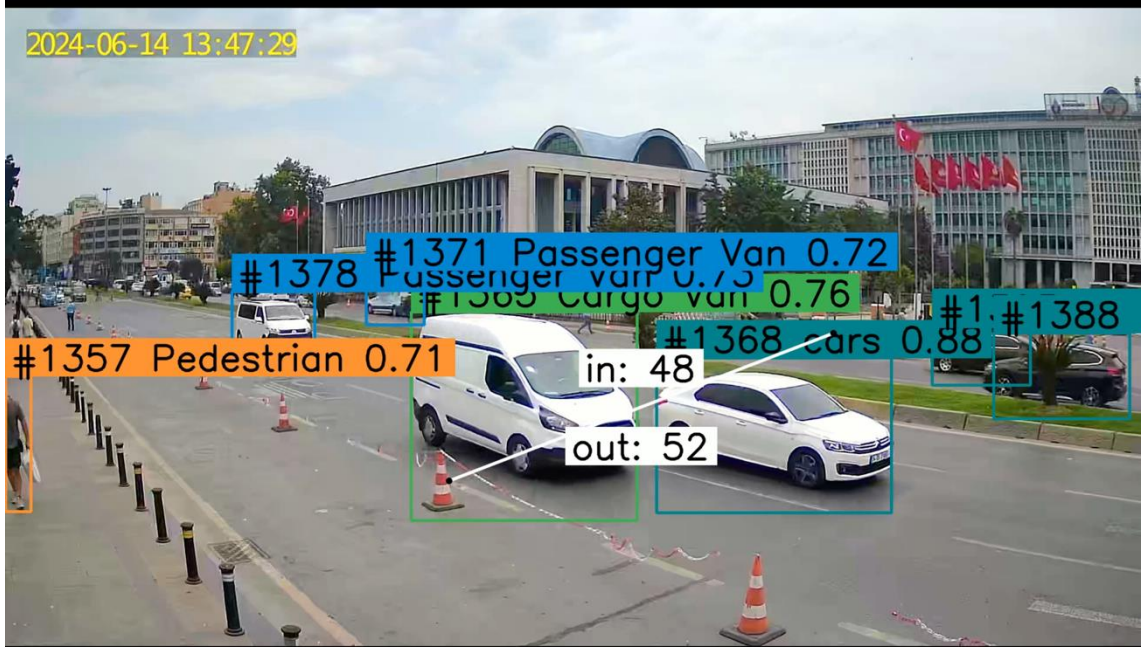
Sınırlamalar: Etiketleme süreci ve kısıtlı finansal kaynaklar nedeniyle, daha geniş bir veri seti oluşturulamamış ve daha fazla epoch sayısı ile eğitim yapılamamıştır.



Şekil 5.1.1.



Şekil 5.1.2.

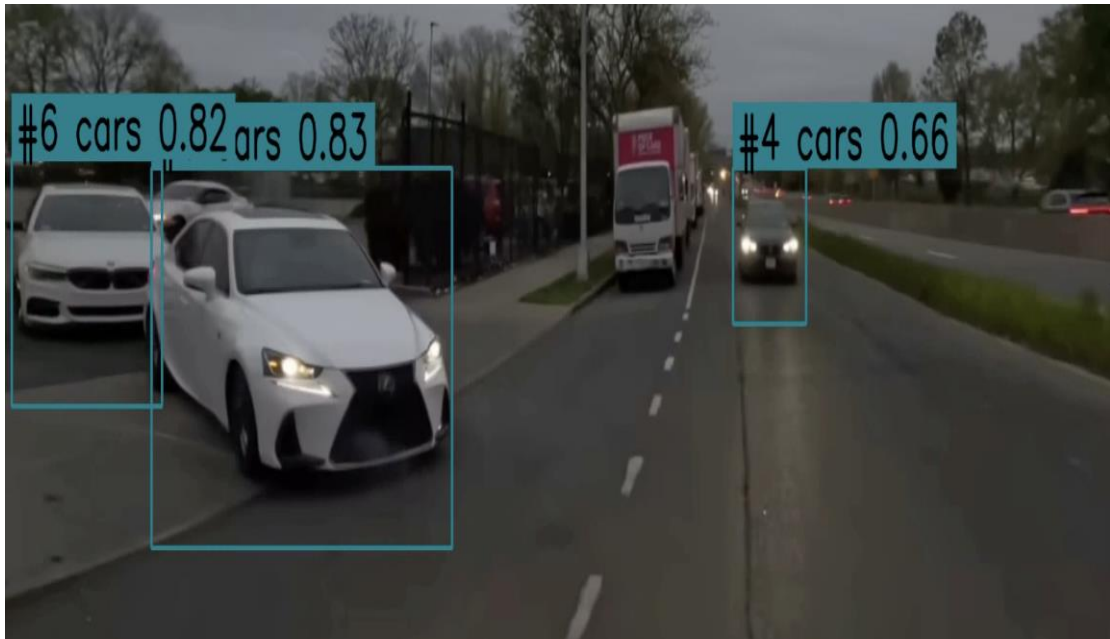


Şekil 5.1.3.

Şekil 5.1.3. Çizgi Geçiş Analizine Dayalı Araç Takibi

Araçlar, belirlenen referans çizgiyi geçtiklerinde tespit edilerek türlerine göre giriş-çıkış sayıları kaydedilmektedir.

Hareketli araç içi



5.2. Öneriler

- **Veri Seti Genişletilmesi:** Gelecekteki çalışmalarda, daha fazla araç sınıfını kapsayan ve farklı hava ve ışık koşullarını içeren geniş bir veri seti kullanılarak modelin genel başarımı artırılabilir.
- **Gerçek Zamanlı Uygulama:** Gerçek zamanlı araç sayımı ve sınıflandırma uygulamaları için, güçlü donanımlarla entegre edilen modeller kullanılabilir.
- **Otonom Araçlar İçin Entegrasyon:** Bu çalışmada geliştirilen algoritmalar, otonom araçlar için nesne algılama ve trafik analizi altyapısında kullanılabilir.
- **Algoritma İyileştirmeleri:** Bytetrack gibi algoritmaların daha gelişmiş sürümleri entegre edilerek takip performansı artırılabilir.
- **Finansal Destek:** Daha yüksek GPU kaynakları ve premium veri işleme araçları (örneğin Roboflow'un daha üst seviye planları) için finansal destek sağlanarak çalışma kapsamı genişletilebilir.

KAYNAKLAR

Web ve Araçlar

- Roboflow, 2024, *Vehicle Dataset Preparation Platform* [online], <https://roboflow.com> [Ziyaret Tarihi: 3 Ocak 2025].
- Google Colab, 2024, *Cloud-based Jupyter Notebook Service* [online], <https://colab.research.google.com> [Ziyaret Tarihi: 3 Ocak 2025].
- Ultralytics, 2024, *YOLOv8 Object Detection Framework* [online], <https://github.com/ultralytics/ultralytics> [Ziyaret Tarihi: 3 Ocak 2025].
- Bytetrack, 2024, *High-Performance Multiple Object Tracker* [online], <https://github.com/ifzhang/ByteTrack> [Ziyaret Tarihi: 3 Ocak 2025].
- Supervision, 2024, *Video Analytics Library for Object Detection* [online], <https://github.com/roboflow/supervision> [Ziyaret Tarihi: 3 Ocak 2025].
- Python Software Foundation, 2024, *Python Programming Language* [online], <https://www.python.org> [Ziyaret Tarihi: 3 Ocak 2025].

Kütüphaneler

- NumPy, 2024, *Numerical Computing Library for Python* [online], <https://numpy.org> [Ziyaret Tarihi: 24 Kasım 2024].
- pandas, 2024, *Data Analysis and Manipulation Library for Python* [online], <https://pandas.pydata.org> [Ziyaret Tarihi: 3 Ocak 2025].
- OpenCV, 2024, *Open Source Computer Vision Library* [online], <https://opencv.org> [Ziyaret Tarihi: 25 Kasım 2025].
- Matplotlib, 2024, *Python Plotting Library* [online], <https://matplotlib.org> [Ziyaret Tarihi: 3 Ocak 2025].
- TensorFlow, 2024, *Machine Learning Framework* [online], <https://www.tensorflow.org> [Ziyaret Tarihi: 3 Ocak 2025].
- PyTorch, 2024, *Deep Learning Framework* [online], <https://pytorch.org> [Ziyaret Tarihi: 3 Ocak 2025].

Veri Seti

- COCO Dataset, 2024, *Common Objects in Context* [online], <https://cocodataset.org> [Ziyaret Tarihi: 3 Ocak 2025].

EKLER

EK-1

Kontrol Edilecek Hususlar	Evet	Hayır
Sayfa yapısı uygun mu?		
Şekil ve çizelge başlık ve içerikleri uygun mu?		
Denklem yazımları uygun mu?		
İç kapak, onay sayfası, Proje bildirimi, özet, abstract, önsöz ve/veya teşekkür uygun yazıldı mı?		
Proje yazımı; Giriş, Kaynak Araştırması, Materyal ve Yöntem (veya Teorik Esaslar), Araştırma Bulguları ve Tartışma, Sonuçlar ve Öneriler sıralamasında mıdır?		
Kaynaklar soyadı sırasına göre verildi mi?		
Kaynaklarda verilen her bir yayına proje içerisinde atıfta bulunuldu mu?		
Kaynaklar açıklanan yazım kuralına uygun olarak yazıldı mı?		
Proje içerisinde kullanılan şekil ve çizelgelerde kullanılan ifadeler Türkçe'ye çevrilmiş mi? (Latince ve Özel kelimeler hariçtir)		
Projenin içindekiler kısmı, proje içerisinde verilen başlıklara uygun hazırlanmış mı?		

Yukarıdaki verilen cevapların doğruluğunu kabul ediyorum.

Unvanı Adı SOYADI

İmza

Öğrenci :

Danışman :

