

**GELİŞMİŞ YAYA TESPİT SİSTEMİ İÇİN NESNE ALGILAMA
ALGORİTMALARININ KARŞILAŞTIRMALI ANALİZİ**

YÜKSEK LİSANS TEZİ

Rehnüma TURUNÇ

Danışman

Doç. Dr. Ahmet Haşim YURTTAKAL

İNTERNET VE BİLİŞİM TEKNOLOJİLERİ YÖNETİMİ ANABİLİM DALI

Ocak 2026

AFYON KOCATEPE ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

YÜKSEK LİSANS TEZİ

GELİŞMİŞ YAYA TESPİT SİSTEMİ İÇİN NESNE ALGILAMA
ALGORİTMALARININ KARŞILAŞTIRMALI ANALİZİ

Rehnüma TURUNÇ

Danışman

Doç. Dr. Ahmet Haşim YURTTAKAL

İNTERNET VE BİLİŞİM TEKNOLOJİLERİ YÖNETİMİ
ANABİLİM DALI

Ocak 2026

TEZ ONAY SAYFASI

Rehnüma KÜÇÜKİLHAN TURUNÇ tarafından hazırlanan “GELİŞMİŞ YAYA TESPİT SİSTEMİ İÇİN NESNE ALGILAMA ALGORİTMALARININ KARŞILAŞTIRMALI ANALİZİ” adlı tez çalışması lisansüstü eğitim ve öğretim yönetmeliğinin ilgili maddeleri uyarınca 15 / 01 / 2026 tarihinde aşağıdaki jüri tarafından **oy birliği** ile Afyon Kocatepe Üniversitesi Fen Bilimleri Enstitüsü **İnternet ve Bilişim Teknolojileri Yönetimi Anabilim Dalı’nda YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Danışman : Doç. Dr. Ahmet Haşim YURTTAKAL

Başkan : Doç. Dr. Uçman ERGÜN

Afyon Kocatepe Üni. Mühendislik Fak.

Üye : Doç. Dr. Hakan KÖR

Hitit Üni. Mühendislik Fakültesi

Üye : Doç. Dr. Ahmet Haşim YURTTAKAL

Afyon Kocatepe Üni. Mühendislik Fak.

Afyon Kocatepe Üniversitesi
Fen Bilimleri Enstitüsü Yönetim Kurulu’nun
..... /..... /..... tarih ve
..... sayılı kararıyla onaylanmıştır.

.....

Prof. Dr. Bekir YALÇIN

Enstitü Müdürü

BİLİMSEL ETİK BİLDİRİM SAYFASI

Afyon Kocatepe Üniversitesi

Fen Bilimleri Enstitüsü, tez yazım kurallarına uygun olarak hazırladığım bu tez çalışmada;

- Tez içindeki bütün bilgi ve belgeleri akademik kurallar çerçevesinde elde ettiğimi,
- Görsel, işitsel ve yazılı tüm bilgi ve sonuçları bilimsel ahlak kurallarına uygun olarak sunduğumu,
- Başkalarının eserlerinden yararlanılması durumunda ilgili eserlere bilimsel normlara uygun olarak atıfta bulunduğumu,
- Atıfta bulunduğum eserlerin tümünü kaynak olarak gösterdiğimi,
- Kullanılan verilerde herhangi bir tahrifat yapmadığımı,
- Ve bu tezin herhangi bir bölümünü bu üniversite veya başka bir üniversitede başka bir tez çalışması olarak sunmadığımı

beyan ederim.

23 / 01 / 2026

Rehnüma TURUNÇ

İmza

ÖZET

Yüksek Lisans Tezi

GELİŞMİŞ YAYA TESPİT SİSTEMİ İÇİN NESNE ALGILAMA ALGORİTMALARININ KARŞILAŞTIRMALI ANALİZİ

Rehnüma TURUNÇ

Afyon Kocatepe Üniversitesi

Fen Bilimleri Enstitüsü

İnternet ve Bilişim Teknolojileri Yönetimi Anabilim Dalı

Danışman: Doç. Dr. Ahmet Haşim YURTTAKAL

Bu araştırmada, yaya çarpışmalarını önlemek için derin öğrenme tabanlı bir yaya algılama sistemi geliştirilmiştir. Bu çarpışmalar, kentsel trafik kazalarının önemli bir bölümünü oluşturmaktadır. MakeSense etiketleme aracını kullanarak 620 adet yüksek çözünürlüklü yaya görüntüsünden oluşan özel bir veri seti oluşturuldu ve bu veriler etiketlendi. Bu veri setini kullanarak YOLOv8, YOLOv11 ve YOLOv12 modelleri eğitildi ve bunları kesinlik, duyarlılık, mAP ve F1 puanı temelinde değerlendirdik. Eğitim GPU hızlandırma desteği ile Python kullanılarak Google Colab ortamında gerçekleştirildi. Modeller arasında YOLOv11-S, %94,9'luk F1 puanı ve %95.7 mAP değeri ile en yüksek performansı elde etti. Daha sonra eğitilmiş modeli PyQt5 tabanlı bir masaüstü simülasyon arayüzüne entegre ederek gerçek zamanlı yaya algılama ve otomatik trafik ışığı kontrolü sağlandı. Sonuçlar, derin öğrenme tabanlı yaya algılama sistemlerinin gerçek zamanlı senaryolarda etkili bir şekilde çalışabileceğini ve akıllı şehir altyapıları için sürdürülebilir, ölçeklenebilir bir çözüm sunabileceğini göstermektedir.

2026, ix + 69 sayfa

Anahtar Kelimeler: Yapay zeka, Derin öğrenme, YOLO, Trafik yönetimi, Yaya algılama.

ABSTRACT

M.Sc. Thesis

COMPARATIVE ANALYSIS OF OBJECT DETECTION ALGORITHMS FOR ADVANCED PEDESTRIAN DETECTION SYSTEMS

Rehnüma TURUNÇ

Afyon Kocatepe University

Graduate School of Natural and Applied Sciences

Department of Internet and Information Technologies Management

Supervisor: Assoc. Prof. Ahmet Haşim YURTTAKAL

In this research, a deep learning-based pedestrian detection system was developed to prevent pedestrian collisions. These collisions constitute a significant portion of urban traffic accidents. A custom dataset consisting of 620 high-resolution pedestrian images was created using the MakeSense labelling tool, and these images were labelled. Using this dataset, YOLOv8, YOLOv11, and YOLOv12 models were trained and evaluated based on precision, recall, mAP, and F1 score. Training was performed using Python with GPU acceleration support in the Google Colab environment. Among the models, YOLOv11-S achieved the highest performance with an F1 score of 94.9% and an mAP value of 95.7%. The trained model was then integrated into a PyQt5-based desktop simulation interface to enable real-time pedestrian detection and automatic traffic light control. The results demonstrate that deep learning-based pedestrian detection systems can operate effectively in real-time scenarios and offer a sustainable, scalable solution for smart city infrastructure.

2026, ix + 69 pages

Keywords: Artificial intelligence, Deep learning, YOLO, Traffic management, Pedestrian detection.

TEŞEKKÜR

Bu araştırmanın konusu, deneysel çalışmaların yönlendirilmesi, sonuçların değerlendirilmesi ve yazımı aşamasında göstermiş olduğu özverili destek ve değerli katkılarından dolayı tez danışmanım Sayın Doç. Dr. Ahmet Haşim YURTTAKAL’a en içten teşekkürlerimi sunarım. Araştırma ve yazım sürecinde, akademik bir çalışma yürütmenin gerektirdiği titizlik ve disiplin konusunda bana yol gösteren, fikirleriyle bakış açımı zenginleştiren, öneri ve eleştirileriyle çalışmamın olgunlaşmasına katkı sağlayan tüm hocalarıma ve arkadaşlarıma teşekkürü bir borç bilirim.

Bu zorlu süreç boyunca sabırları, anlayışları ve destekleriyle her zaman yanımda olan aileme ve eşime minnettarım. Yoğun çalışma dönemlerinde motivasyonumu korumamı sağlayan, her aşamada beni cesaretlendiren ve başarıya ulaşma yolunda güç veren ailem ve eşim, bu çalışmanın ortaya çıkmasında en büyük paya sahip olanlardır. Onların varlığı ve güveni benim için daima en değerli destek kaynağı olmuştur. Bu nedenle hem manevi hem de maddi anlamda gösterdikleri sonsuz destek için kendilerine gönülden teşekkür ederim.

Rehnüma TURUNÇ
Afyonkarahisar 2026

İÇİNDEKİLER DİZİNİ

	Sayfa
ÖZET	i
ABSTRACT	ii
TEŞEKKÜR	iii
İÇİNDEKİLER DİZİNİ.....	iv
SİMGELER ve KISALTMALAR DİZİNİ	vi
ŞEKİLLER DİZİNİ	vii
ÇİZELGELER DİZİNİ.....	viii
RESİMLER DİZİNİ	ix
1.GİRİŞ.....	1
1.1 Problem Tanımı	1
1.2 Araştırmanın Önemi ve Amacı	2
1.3 Tezin Yapısı.....	3
2. LİTERATÜR BİLGİLERİ	5
3. KAVRAMSAL ÇALIŞMALAR.....	7
3.1 Yapay Zeka	8
3.2 Makine Öğrenmesi.....	9
3.3 Derin Öğrenme	11
3.3.1 Evrişimsel Sinir Ağları.....	13
3.3.1.1 Evrişim Katmanları	14
3.3.1.2 Havuzlama Katmanları.....	14
3.3.1.3 Aktivasyon Fonksiyonları	15
3.3.1.4 Tam Bağlı Katmanları.....	15
3.3.2 Nesne Tespiti ve YOLO Mimarisi	16
3.3.2.1 YOLOv5.....	20
3.3.2.2 YOLOv7.....	21
3.3.2.3 YOLOv8.....	21
3.3.2.4 YOLOv9.....	21
3.3.2.5 YOLOv10.....	22
3.3.2.6 YOLOv11.....	22

3.3.2.7 YOLOv12.....	22
4. MATERYAL ve METOT	24
4.1 Veri Toplama	24
4.2 Veri Etiketleme	25
4.3 Veri Ön İşleme.....	26
4.4 Model	26
4.4.1 YOLOv8.....	27
4.4.2 YOLOv11.....	29
4.4.3 YOLOv12.....	30
4.4.4 YOLOv8, YOLOv11 ve YOLOv12'nin Karşılaştırılması.....	32
4.5 Değerlendirme Metrikleri	35
4.5.1 Karmaşıklık Matrisi	35
4.5.2 Kesinlik	36
4.5.3 Duyarlılık	37
4.5.4 Ortalama Kesinlik (mAP)	37
4.5.5 F1 Skor	38
5. BULGULAR	39
5.1 Model Performans Değerlendirmeleri	39
5.1.1 YOLOv8 Model Performans Değerlendirmesi	39
5.1.2 YOLOv11 Model Performans Değerlendirmesi	44
5.1.3 YOLOv12 Model Performans Değerlendirmesi	48
5.2 En iyi Modelin Seçilip Çıktı Alınması	54
5.3 Simülasyon.....	55
6. TARTIŞMA ve SONUÇ	58
7. KAYNAKLAR.....	64
ÖZGEÇMİŞ.....	68

SİMGELER ve KISALTMALAR DİZİNİ

Simgeler

α	Öğrenme oranı
λ	Kayıp fonksiyonlarında kullanılan ağırlık katsayısı
σ	Öğrenme Sürecinde Kullanılan Standart Sapma veya Hata Değişimi

Kısaltmalar

API	Uygulama Programlama Arayüzü
CNN	Evrişimsel Sinir Ağı
CPU	Merkezi İşlem Birimi
DCNN	Derin Evrişimli Sinir Ağı
DNN	Derin Sinir Ağı
FPS	Saniyedeki Kare Sayısı
GELAN	Genelleştirilmiş Verimli Katman Birleştirme Ağı
GPU	Grafik İşlem Birimi
GUI	Grafiksel Kullanıcı Arayüzü
IOT	Nesnelerin İnterneti
IoU	Birleşim Üzerine Kesişim
mAP	Ortalama Kesinlik
RELU	Düzeltilmiş Doğrusal Birim
SGD	Stokastik Gradyan İnişi
YOLO	Sadece Bir Kez Bak

ŞEKİLLER DİZİNİ

	Sayfa
Şekil 3.1 Hiyerarşik ilişki.....	8
Şekil. 3.2 YOLO Algoritmasının Çalışma Prensibi	17
Şekil 4.1 Yapılan Çalışmanın Akış Şeması.....	24
Şekil 4.3 YOLOv8 mimarisinin genel yapısı ((Jocher, Chaurasia, ve Qiu 2023))	28
Şekil 4.4. YOLOv11 Mimarisinin Genel Yapısı.....	29
Şekil 4.6 Karmaşıklık Matrisi	36

ÇİZELGELER DİZİNİ

Çizelge 3.1 YOLO Sürümlerinin Karşılaştırılması	19
Çizelge 4.2 YOLOv8, YOLOv11, YOLOv12 Sürümlerinin Detaylı Karşılaştırması ...	33
Çizelge 5.2 YOLO Modellerle Eğitim Denemeleri.....	52
Çizelge 5.3 YOLO Sürümlerinden En İyi Performans Gösteren Modellerin Karşılaştırması	54

RESİMLER DİZİNİ

Resim 4.2. Veri seti içerisindeki örnek görüntüler	25
Resim 5.1 YOLOv8-S En yüksek performansı gösteren modelin Kesinlik–Duyarlılık eğrisi	40
Resim 5.2 YOLOv8-S en yüksek performans gösteren modelin F1–Güven eğrisi	41
Resim 5.3 YOLOv8-S en yüksek performansı gösteren modelin normalize edilmiş karmaşıklık matrisi	43
Resim 5.4 YOLOv11-S en yüksek performansı gösteren modelin normalize edilmiş karmaşıklık matrisi	45
Resim 5.5 YOLOv11-S en yüksek performansı gösteren modelin Kesinlik–Duyarlılık eğrisi	46
Resim 5.6 YOLOv11-S en yüksek performansı gösteren modelinin F1-Güven Eğrisi .	47
Resim 5.7 YOLOv12-N en yüksek performansı gösteren modelin Kesinlik–Duyarlılık eğrisi	49
Resim 5.8 YOLOv12-N en yüksek performansı gösteren modelin Normalleştirilmiş Karmaşıklık Matrisi	50
Resim 5.9 YOLOv12-N en yüksek performansı gösteren modelin F1–Güven Skoru Eğrisi.....	51
Resim 5.10 YOLOv11-S modeline ait örnek tespit çıktıları.....	54
Resim 5.11. Simülasyon Arayüzü	56

1.GİRİŞ

1.1 Problem Tanımı

Teknolojik gelişmeler, günümüzde toplumsal yaşamın her alanına nüfuz ederek bireylerin karşılaştığı birçok soruna yenilikçi çözümler sunmaktadır. Bu gelişmeler hem yaşam kalitesini artırmakta hem de sosyal ve ekonomik açıdan daha sürdürülebilir kent modellerinin ortaya çıkmasına katkı sağlamaktadır. Bununla birlikte, hızlı kentleşme ve artan araç trafiği, özellikle trafik güvenliği alanında yeni sorunların ortaya çıkmasına neden olmaktadır. Dünya Sağlık Örgütü'nün raporlarına göre trafik kazaları, tüm dünyada ölümlerin önde gelen nedenlerinden biri olup, kazaların önemli bir bölümü yaya geçitleri ve yoğun nüfuslu bölgelerde meydana gelmektedir (Sungur vd. 2014).

Birçok insanın günlük yaşamının önemli bir bölümünü trafikte geçiriyor. Bu süreçte yaya veya araç güvenliği ciddi riskler oluşturmaktadır. Artan araç trafiği ve yoğun nüfus nedeniyle modern şehirlerde yaya güvenliği ciddi bir endişe kaynağı haline gelmiştir. Mevcut trafik yönetim sistemleri ve yaya algılama teknolojileri yayaların korunması açısından yeterli güvenliği sağlamamaktadır. Trafik kazalarının çoğu trafiğin yoğun olduğu yerlerde ve yaya geçitlerinde, özellikle şehir merkezlerinde ve okul bölgelerinde meydana geldiğinden, gerçek zamanlı bir yaya algılama sisteminin geliştirilmesi büyük bir ihtiyaç olarak görülmektedir (Karabayir 2016). Bu kazaların büyük bir kısmı sürücülerin yayaları anında fark edememesinden ya da dikkatsizlik, yetersiz trafik sinyalleri, hızlı araç kullanımı gibi nedenlerden kaynaklanmaktadır. Bu sebeplerin önüne geçmek için bu sistemin trafik sinyalizasyon sistemlerine entegre edilmesi ve sistemin yayalara ve sürücülere anlık uyarı vermesi büyük bir ihtiyacı karşılayacaktır. Böylece hem şehir içi trafik akışı iyileştirilecek hem de yaya kazalarının önlenmesi için etkin bir çözüm sağlanmış olacaktır.

Modern şehirlerde araç yoğunluğunun artması ve yayaların trafikte daha fazla yer alması, yaya güvenliği sorununu kritik bir halk sağlığı problemi haline getirmiştir (Vikruthi vd. 2025). Mevcut trafik yönetim sistemleri yoğunlukla araç akışını düzenlemeye odaklanmakta, yayaların korunmasına yönelik çözümler ise sınırlı kalmaktadır. Trafik

kazalarının önemli bir kısmı sürücülerin yayaları zamanında fark edememesi, dikkatsizlik, yetersiz sinyalizasyon altyapısı ve hız ihlalleri gibi nedenlerden kaynaklanmaktadır. Özellikle şehir merkezleri, okul bölgeleri ve yaya yoğunluğu yüksek alanlar, riskin en fazla olduğu noktaları oluşturmaktadır.

Bu nedenle, gerçek zamanlı yaya algılama sistemlerinin geliştirilmesi ve bu sistemlerin trafik sinyalizasyon altyapısına entegre edilmesi kritik bir ihtiyaç haline gelmiştir(Wang vd. 2024). Böyle bir sistem hem sürücülerin hem de yayaların anlık olarak uyarılmasına imkân tanıyarak kazaların önüne geçebilir ve trafik akışını daha güvenli hale getirebilir.

1.2 Araştırmanın Önemi ve Amacı

Bu çalışmanın önemi, yalnızca trafik güvenliği sağlamaya yönelik bir yaklaşım geliştirmesiyle sınırlı değildir; aynı zamanda akıllı şehir uygulamaları bağlamında sürdürülebilir, ölçeklenebilir ve teknolojik açıdan yenilikçi bir çözüm modeli önermektedir. Akıllı şehir vizyonu, kaynakların daha verimli kullanılmasını, yaşam kalitesinin artırılmasını ve güvenlik sorunlarının teknoloji aracılığıyla çözülmesini hedeflemektedir. Geliştirilecek olan yaya algılama sistemi, bu vizyona katkı sağlayarak hem trafik güvenliği hem de trafik akışının optimizasyonu açısından önemli bir rol üstlenecektir.

Ayrıca bu çalışma akıllı şehir konseptine katkıda bulunmayı amaçlamaktadır. Sistem sadece trafik güvenliğini sağlamakla kalmayacak, aynı zamanda trafik akışını optimize edecek ve şehir içi ulaşımı daha verimli hale getirecektir. Uzun vadede yaya algılama sisteminin kent genelinde yaygın olarak kullanılması, gelecekteki akıllı kent uygulamaları için sürdürülebilir bir çözüm modeli sağlayacaktır. Bu entegrasyon sayesinde yaya geçidinde bir yaya tespit edildiğinde sinyalizasyon sistemine anlık bildirim gönderilecek ve sürücülere araç içi uyarılar verilerek çarpışma riskini önceden tespit etmeleri ve kazaları önlemeleri sağlanacak.

Bu sistemin temel amacı, derin öğrenme tabanlı nesne algılama yöntemlerini kullanarak gerçek zamanlı yaya tespiti gerçekleştirmek ve bu tespitleri trafik sinyalizasyon

sistemlerine entegre ederek sürücü ve yayalara anlık uyarılar sunmaktır. Böylelikle, yaya geçitlerinde meydana gelebilecek kazaların önlenmesi, şehir içi trafik akışının daha güvenli ve verimli hale getirilmesi amaçlanmaktadır. Uzun vadede bu yaklaşım, akıllı şehirlerin güvenlik altyapılarına entegre edilebilecek sürdürülebilir ve ölçeklenebilir bir model sunacaktır.

Bu tez çalışması, şu katkıları hedeflemektedir:

- YOLO mimarisine dayalı derin öğrenme modelleri ile gerçek zamanlı yaya algılama sistemi geliştirmek,
- Geliştirilen modeli özgün bir yaya veri seti üzerinde eğiterek performansını değerlendirmek,
- Algılama sonuçlarını trafik sinyalizasyon sistemleri ile entegre ederek sürücülere ve yayalara anlık uyarılar sağlamak,
- Geliştirilen sistemi akıllı şehir altyapılarıyla uyumlu, sürdürülebilir ve ölçeklenebilir bir çözüm modeli olarak önermek.

1.3 Tezin Yapısı

Bu tez toplam yedi bölümden oluşmaktadır. İkinci bölümde, nesne algılama alanında yapılan çalışmalar incelenmiş; özellikle derin öğrenme tabanlı yöntemler ve YOLO mimarisi üzerine gerçekleştirilen yaya algılama çalışmaları kapsamlı bir literatür taramasıyla sunulmuştur. Üçüncü bölümde, çalışmanın kuramsal altyapısını oluşturan kavramsal çalışmalar ele alınmış; derin öğrenme, evrimsel sinir ağları, nesne algılama yaklaşımları ve YOLO modellerinin temel prensipleri açıklanmıştır. Dördüncü bölümde, çalışmada kullanılan materyal ve yöntemler detaylandırılmış; özgün veri setinin oluşturulma süreci, etiketleme işlemleri, veri ön işleme adımları, kullanılan model mimarileri ve eğitim parametreleri açıklanmıştır. Beşinci bölümde, farklı YOLO sürümleri kullanılarak elde edilen deneysel bulgular sunulmuş; modellerin doğruluk, kesinlik, duyarlılık, F1-skor ve mAP gibi performans metrikleri karşılaştırmalı olarak değerlendirilmiştir. Altıncı bölümde, elde edilen bulgular literatürdeki benzer çalışmalarla karşılaştırılarak tartışılmış; çalışmanın güçlü yönleri, sınırlılıkları ve

literatüre katkıları ortaya konulmuş, ayrıca genel sonuçlara yer verilmiştir. Yedinci ve son bölümde ise çalışmada yararlanılan tüm kaynaklar, akademik yazım kurallarına uygun şekilde kaynakça bölümünde sunulmuştur.

2. LİTERATÜR BİLGİLERİ

Son yıllarda derin öğrenme tabanlı yaya tespiti üzerine birçok çalışma yapılmıştır. Razzok ve arkadaşları tarafından yapılan çalışmada R-CNN, SSD ve YOLOv3 gibi son teknoloji CNN tabanlı nesne tespit modelleri karşılaştırılmış ve özellikle YOLOv3 modelinin yaya tespitinde en iyi performansı gösterdiği tespit edilmiştir (Razzok vd. 2022). Bu çalışma, farklı CNN mimarilerinin yaya algılama doğruluğu ve hızı açısından etkinliğini göstermesi açısından önemli bir referans oluşturmaktadır.

Tian vd. (2015) yaya tespitinde sıklıkla karşılaşılan oklüzyon problemini çözmek için DeepParts adında bir model önermiştir (Tian vd. 2015). Bu parça tabanlı yaklaşım, kısmi görünümelerde bile yayaları doğru bir şekilde tanıyabilmekte ve düşük IoU değerlerine sahip pozitif örneklerle başa çıkabilmektedir. Caltech veri kümesi üzerinde yapılan deneylerde %11,89 gibi düşük hata oranıyla dikkat çeken bu model, döneminin en başarılı sonuçlarından birini elde etmiştir.

Tome vd. (2016) genel bir evrişimli sinir ağı yapısını optimize ederek yaya tespiti için düşük hesaplama maliyetli ve yüksek doğrulukta bir sistem geliştirmiştir (Tomè vd. 2016). Sistem 192 çekirdekli NVIDIA Jetson TK1 üzerinde test edilmiş ve otonom araç uygulamaları için uygun bir çözüm olarak değerlendirilmiştir.

Xu vd. (2024) tarafından geliştirilen MM-Net modeli, kamera ve LiDAR verilerini birleştiren çok modlu bir yaya algılama sistemidir (Xu vd. 2024). Bu çalışmada sensör kalibrasyonu, veri senkronizasyonu ve füzyon modüllerini birleştiren MM-Net modeli çift dallı bir mimari ile önerilmiştir. KITTI ve özel veri kümeleri üzerinde yapılan testler, bu çok modlu yaklaşımın yüksek doğruluk sağladığını göstermektedir.

Benzer şekilde Wei (2015) tarafından geliştirilen bir başka yaklaşımda da Toplanmış Kanal Özellikleri ve Derin Evrişimsel Sinir Ağı özellikleri birlikte kullanılmıştır (Chen Xiaogang and Wei 2015). Bu yöntemde ACF aday yaya alanlarını belirlemek için kullanılırken DCNN daha doğru sınıflandırma için kullanılmıştır. INRIA, Caltech ve

ETH veri kümeleri üzerinde yapılan deęerlendirmelerde yöntemin yüksek performans sergiledięi görölmüştür.

Son olarak, Dollár vd. (2009) düşük çözünürlüklü ve kısmen örtüşen yaya görüntülerini içeren Caltech Yaya veri setini tanıtmıştır (Dollar vd. 2009). Bu veri kümesi, yaya tespit sistemlerinin karşılaştırılması ve deęerlendirilmesi için standart bir referans haline gelmiş ve birçok çalışmada temel veri kümesi olarak kullanılmıştır.

Bu bağlamda, literatürdeki çalışmaların çoęu belirli veri kümeleri üzerinde algoritma performansını kıyaslamaya odaklanmıştır. Ancak bu çalışmanın katkısı, yalnızca algoritmaların doğruluk oranlarını incelemek deęil; aynı zamanda YOLO mimarisine dayalı, gerçek zamanlı çalışan ve trafik ışıkları ile entegre edilebilecek bir yaya algılama sistemi geliştirmek olacaktır. Böylece tez, hem akademik anlamda yeni bir metodolojik katkı sunmayı hem de uygulamaya dönük, pratik bir çözüm geliştirmeyi amaçlamaktadır. Günümüzde teknolojinin hızla ilerlemesi birçok sorun için yenilikçi çözümler geliştirilmesine ve uygulanmasına olanak sağlamaktadır. Güncel teknolojilerin kullanımı günlük hayatta karşılaşılan zorlukları azaltmakta, daha yenilikçi ve verimli bir yaşam tarzı sunmaktadır. Bu gelişmeler hem bireylerin hayatını kolaylaştırıyor hem de sorunlara daha modern ve etkili yaklaşımlar getirilmesine katkı sağlıyor. Teknolojinin hayatımızdaki sorunlara getirdięi çözümlerin yanı sıra trafik güvenlięi gibi temel konular da yeni yaklaşımlar gerektiriyor. Özellikle artan araç trafięi ve yaya güvenlięi gibi sorunlar teknoloji entegrasyonu ile ele alınabilir hale gelmiştir. Bu bağlamda, şehir içi trafik kazalarının çoęunun yaya geçitlerinde ve yoğun alanlarda meydana gelmesi, bu konuda hızlı çözümler geliştirilmesi gerektięine işaret etmektedir.

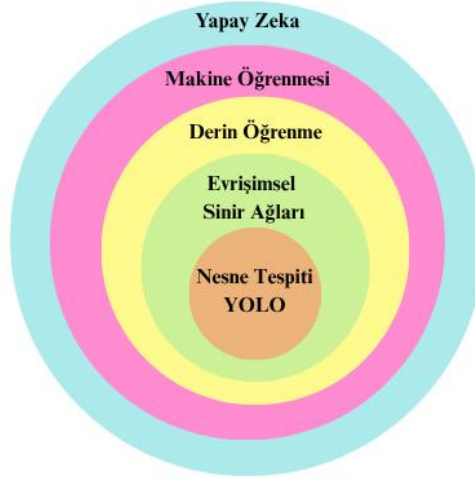
3. KAVRAMSAL ÇALIŞMALAR

Bu bölümde, çalışmanın temelini oluşturan yapay zekâ, makine öğrenmesi ve derin öğrenme kavramları teorik açıdan ele alınmaktadır. Ayrıca bu kavramların alt dallarından biri olan Evrişimsel Sinir Ağları ayrıntılı biçimde açıklanmış, Evrişimsel Sinir Ağları mimarisinde yer alan temel katmanların işlevleri detaylandırılmıştır.

Bu kavramlar arasındaki hiyerarşik ilişki Şekil 3.1’te görsel olarak özetlenmiştir. Şekilde, yapay zekâdan başlayarak YOLO mimarisine kadar uzanan çok katmanlı yapı gösterilmekte; yapay zekânın en genel kavram, YOLO’nun ise onun altında yer alan derin öğrenme temelli bir nesne tespiti yöntemi olduğu vurgulanmaktadır. Bu hiyerarşi, çalışmada ele alınan yöntemlerin birbirine nasıl bağlı olduğunu kavramsal olarak ortaya koymaktadır.

Bölümün ilerleyen kısımlarında, görüntü tabanlı nesne tespiti alanında yaygın olarak kullanılan YOLO mimarisi ve farklı sürümleri (YOLOv5, YOLOv7, YOLOv8, YOLOv9, YOLOv10, YOLOv11 ve YOLOv12) hakkında bilgiler verilmiştir. Bu modellerin performansları, hız ve doğruluk açısından literatürde yer alan çalışmalar ışığında değerlendirilmiştir.

Son olarak, bu bölümde yaya tespiti alanında yapılan önceki çalışmalar incelenmiş, literatürde kullanılan yöntemler, veri kümeleri ve performans ölçütleri detaylandırılmıştır. Böylece, bu tez çalışmasının mevcut literatüre sunduğu katkıların kuramsal zemini oluşturulmuştur.



Şekil 3.1 Hiyerarşik ilişki

3.1 Yapay Zeka

Yapay zekâ bilgisayar sistemlerinin insan zekâsına özgü öğrenme, akıl yürütme, problem çözme, algılama ve karar verme gibi bilişsel işlevleri taklit edebilmesini amaçlayan bir araştırma alanıdır (Öztürk vd. 2018). Başka bir ifadeyle yapay zekâ, makinelerin çevresinden aldığı verileri işleyerek akıllı davranışlar sergilemesini sağlayan algoritmalar ve yöntemler bütünüdür. İlk dönemlerde yalnızca kurallara dayalı uzman sistemler şeklinde geliştirilen yapay zekâ uygulamaları, günümüzde makine öğrenmesi ve derin öğrenme tekniklerinin katkısıyla çok daha güçlü, esnek ve ölçeklenebilir hale gelmiştir.

Yapay zekânın günlük yaşamdaki uygulamaları oldukça geniş bir yelpazeye yayılmaktadır. Sağlık alanında, medikal görüntülerin analizi, tümör tespiti, ilaç keşfi ve kişiselleştirilmiş tedavi planlarının oluşturulmasında etkin rol oynamaktadır. Ulaşım sektöründe, otonom araç teknolojileri, trafik akışının optimizasyonu ve kazaların önlenmesine yönelik akıllı sistemler yapay zekâ desteğiyle geliştirilmektedir. Güvenlik alanında, yüz tanıma, davranış analizi, şüpheli durum tespiti ve siber güvenlik tehditlerinin önlenmesi yapay zekâ tabanlı çözümlerle gerçekleştirilmektedir. Üretim ve endüstri 4.0 uygulamalarında ise yapay zekâ, üretim hatlarının otomasyonu, kalite kontrol, arıza tahmini ve kaynak optimizasyonunda kullanılmaktadır (Çetin vd. 2024). Bunun yanı sıra günlük yaşamda kişisel asistanlar, öneri sistemleri, doğal dil işleme

tabanlı sohbet robotları ve akıllı cihazlar aracılığıyla yapay zekâ ile sürekli etkileşim içinde bulunmaktadır.

Yapay zekânın en önemli etkilerinden biri de akıllı şehirler bağlamında ortaya çıkmaktadır. Akıllı şehir uygulamaları; enerji yönetimi, atık toplama, çevresel izleme, toplu taşıma planlaması ve trafik kontrolü gibi alanlarda yapay zekâ teknolojilerinden yararlanarak kaynakların daha verimli kullanılmasını sağlar. Özellikle trafik yönetimi ve yaya güvenliği konularında yapay zekâ tabanlı çözümler, şehirlerde artan nüfus yoğunluğunun ve araç trafiğinin oluşturduğu sorunlara yenilikçi yaklaşımlar sunmaktadır (Arslan 2020). Gerçek zamanlı görüntü işleme ile yayaların ve araçların algılanması, trafik ışıklarının dinamik olarak yönetilmesi ve sürücülere anlık uyarıların iletilmesi yapay zekânın bu alandaki katkılarına örnek olarak verilebilir.

Dolayısıyla yapay zekâ, yalnızca akademik bir araştırma alanı değil, aynı zamanda yaşamın her alanına nüfuz eden ve gelecekteki toplumsal yapıları şekillendirecek bir teknolojik dönüşüm aracıdır. Bu tez kapsamında da yapay zekâ, özellikle derin öğrenme tabanlı görüntü işleme yöntemleriyle yaya tespiti ve akıllı trafik yönetimi bağlamında ele alınmaktadır.

3.2 Makine Öğrenmesi

Makine öğrenmesi, yapay zekânın en önemli alt dallarından biri olup, bilgisayar sistemlerinin açıkça programlanmadan, verilerden öğrenmesini ve deneyim yoluyla performansını geliştirmesini amaçlayan bir paradigma olarak tanımlanmaktadır (Zilyas vd. 2023). Temel prensip olarak makine öğrenmesi, büyük miktarda veriden istatistiksel örüntüleri ve ilişkileri çıkarmayı, bu örüntülerden modeller üretmeyi ve bu modelleri kullanarak tahmin, sınıflandırma veya karar verme gibi görevleri yerine getirmeyi hedefler. Böylelikle, bilgisayar sistemleri yalnızca önceden tanımlanmış kuralları izlemek yerine, çevresinden gelen veriler doğrultusunda kendi davranışlarını uyarlayabilir.

Makine öğrenmesi süreci, verilerin işlenmesi ve temsil edilmesi ile başlar. Ham veriler, genellikle matematiksel yapılara dönüştürülür ve bu temsiller üzerinden model parametreleri optimize edilir (Pandey vd. 2019). Modelin başarısı, yalnızca eğitim verisine uyum sağlamasıyla değil, aynı zamanda daha önce karşılaşmadığı yeni veriler üzerinde genelleme yapabilmesiyle ölçülür.

Makine öğrenmesi yöntemleri üç temel paradigma altında incelenmektedir:

1. Denetimli Öğrenme: Bu yaklaşımda model, girdi verilerine karşılık gelen doğru etiketler ile eğitilir (Osisanwo vd. 2017). Öğrenme süreci boyunca model, tahmin ettiği çıktılar ile gerçek çıktılar arasındaki farkı ölçen bir kayıp fonksiyonu üzerinden optimize edilir. Denetimli öğrenme, sınıflandırma ve regresyon gibi geniş bir uygulama alanına sahiptir.
2. Denetimsiz Öğrenme: Bu paradigmada veriler etiketlenmemiştir ve amaç, veri içerisindeki gizli yapıları, örüntüleri ve kümelenmeleri keşfetmektir. Kümeleme algoritmaları benzer veri noktalarını gruplarken; boyut indirgeme yöntemleri yüksek boyutlu verilerin daha az boyutta anlamlı temsillerini üretir. Denetimsiz öğrenme, özellikle keşifsel veri analizi, anomalilik tespiti ve özellik çıkarımı alanlarında kritik bir rol oynamaktadır.
3. Pekiştirmeli Öğrenme: Bu öğrenme biçiminde bir ajan, çevresiyle etkileşime girerek eylemlerine karşılık ödül veya ceza alır ve bu geri bildirimlere göre stratejisini zamanla geliştirir. Pekiştirmeli öğrenme, özellikle robotik kontrol, oyun oynama, finansal karar destek sistemleri ve otonom sürüş gibi alanlarda öne çıkmaktadır.

Makine öğrenmesi algoritmalarının başarımı birçok faktöre bağlıdır. Veri kalitesi ve miktarı, öğrenme sürecinin en kritik bileşenlerinden biridir. Yetersiz veya dengesiz veri, modelin genelleme performansını olumsuz etkileyebilir. Model karmaşıklığı, öğrenme sürecinde denge kurulması gereken bir diğer unsurdur: Aşırı uyum durumunda model eğitim verisine fazla bağımlı hale gelerek yeni verilere genelleme yapamaz; eksik uyum durumunda ise veri içerisindeki örüntüleri yeterince öğrenemez (Varoquaux vd. 2023). Bu sorunları gidermek amacıyla düzenleme yöntemleri, çapraz doğrulama, erken durdurma ve uygun veri bölünmesi (eğitim–doğrulama–test) stratejileri kullanılmaktadır.

Geleneksel makine öğrenmesi yöntemleri arasında lojistik regresyon, destek vektör makineleri, karar ağaçları, rassal ormanlar ve k-en yakın komşu algoritmaları yer almaktadır. Bu yöntemler birçok klasik problemde başarılı sonuçlar vermekle birlikte, özellikle yüksek boyutlu veriler ve karmaşık doğrusal olmayan ilişkiler söz konusu olduğunda sınırlı performans gösterebilmektedir. Bu noktada devreye giren derin öğrenme yöntemleri, çok katmanlı yapay sinir ağları sayesinde daha esnek, ölçeklenebilir ve güçlü çözümler sunarak görüntü işleme, doğal dil işleme ve ses tanıma gibi alanlarda önemli başarılar elde etmiştir (Reich vd. 1999).

Sonuç olarak, makine öğrenmesi yalnızca kuramsal bir alan değil, aynı zamanda günümüzde birçok disiplinin dönüşümünü sağlayan temel bir teknolojidir. Bu tez kapsamında ise makine öğrenmesi, özellikle derin öğrenme tabanlı yaya tespiti ve akıllı trafik yönetimi sistemlerinin geliştirilmesinde kritik bir rol oynamaktadır.

3.3 Derin Öğrenme

Derin öğrenme, makine öğrenmesinin bir alt dalı olup, çok katmanlı yapay sinir ağları aracılığıyla veriden hiyerarşik özellik temsilleri öğrenmeyi amaçlayan bir yöntemler bütünü olarak tanımlanmaktadır (Linik vd. 2017). Geleneksel makine öğrenmesi yöntemlerinde özellik çıkarımı genellikle insan uzmanlığına dayanırken, derin öğrenme modelleri bu süreci otomatik olarak gerçekleştirmektedir. Bu durum, özellikle görüntü, ses ve doğal dil gibi yüksek boyutlu ve karmaşık veri türlerinde önemli bir avantaj sağlamaktadır.

Bir derin öğrenme modeli tipik olarak giriş katmanı, birden fazla gizli katman ve çıkış katmanından oluşur. Girdi verisi bu katmanlardan geçerken, her aşamada doğrusal dönüşümler ve doğrusal olmayan aktivasyon fonksiyonları uygulanır (Chen vd. 2015). Böylece model, alt katmanlarda düşük seviyeli özellikleri, orta katmanlarda nesne parçalarını, üst katmanlarda ise daha soyut ve kavramsal temsilleri öğrenir. Bu çok katmanlı yapı, derin öğrenmenin karmaşık örüntüleri yakalayabilmesini ve yüksek doğrulukta tahminler yapabilmesini mümkün kılar.

Derin öğrenme modellerinin eğitimi, geri yayılım algoritması ile gerçekleştirilir. Bu süreçte modelin tahmini ile gerçek çıktı arasındaki hata, bir kayıp fonksiyonu kullanılarak hesaplanır. Ardından zincir kuralı aracılığıyla bu hata geriye doğru iletilerek her katmandaki ağırlıkların gradyanları hesaplanır. Elde edilen gradyanlar, Stokastik Gradyan İnişi (SGD), Adam, AdamW gibi optimizasyon algoritmaları ile kullanılarak model parametreleri güncellenir. Böylece model, her iterasyonda hatayı azaltacak şekilde kendini optimize eder.

Eğitim sürecinin başarısı birçok faktöre bağlıdır:

- Veri miktarı ve kalitesi: Yeterli ve dengeli veri setleri olmadan derin öğrenme modelleri genelleme yapmada başarısız olabilir.
- Model mimarisi: Katman sayısı, bağlantı türleri ve parametre sayısı, modelin öğrenme kapasitesini doğrudan etkiler.
- Optimizasyon stratejileri: Öğrenme oranı planlaması, momentum, ağırlık başlatma yöntemleri eğitimin stabilitesini belirler.
- Düzenleştirme teknikleri: Bırakma, Toplu Normalleştirme, veri artırma gibi yöntemler aşırı uyumu engelleyerek modelin genelleme yeteneğini artırır.
- Hesaplama kaynakları: GPU ve TPU gibi donanımlar, yüksek boyutlu derin öğrenme modellerinin verimli şekilde eğitilmesini sağlar.

Son yıllarda derin öğrenme araştırmalarında transfer öğrenme ve öz-denetimli öğrenme yaklaşımları öne çıkmıştır. Transfer öğrenme, büyük ölçekli veri kümeleri üzerinde önceden eğitilmiş modellerin daha küçük ve özel veri setlerine uyarlanmasını mümkün kılarak eğitim süresini kısaltır ve performansı artırır. Öz-denetimli öğrenme ise etiketlenmiş veriye duyulan ihtiyacı azaltarak, verilerden daha verimli temsiller öğrenmeyi sağlamaktadır.

Derin öğrenme, günümüzde bilgisayarla görme, doğal dil işleme ve ses işleme gibi birçok alanda en ileri düzey başarıları ortaya koymuştur. Bu tez kapsamında ise derin öğrenme, yaya tespiti ve akıllı trafik yönetimi uygulamalarında kullanılacak olan nesne algılama algoritmalarının temelini oluşturmaktadır.

3.3.1 Evrişimsel Sinir Ağları

Evrişimsel Sinir Ağları (CNN), özellikle görüntü işleme ve bilgisayarla görme alanında devrim niteliğinde gelişmelerin önünü açan derin öğrenme mimarileridir (Krizhevsky vd. 2012). CNN'lerin en önemli özelliği, görüntü verilerindeki mekânsal ilişkileri ve yerel bağıntıları koruyarak otomatik özellik çıkarımı yapabilmeleridir. Geleneksel makine öğrenmesi yaklaşımlarında özelliklerin çoğunlukla insan uzmanlığı ile belirlenmesi gerekirken, CNN'ler veriden doğrudan özellik öğrenebilmekte ve bu sayede daha esnek, güçlü ve ölçeklenebilir çözümler sunmaktadır.

CNN'lerin mimarisi, insan beynindeki görsel korteksin çalışma biçiminden esinlenmiştir. Bu mimarilerde yerel bağlantı ve parametre paylaşımı prensipleri uygulanır. Böylece aynı filtre parametreleri, görüntünün farklı bölgelerine uygulanarak özellik çıkarımı yapılır. Bu yaklaşım hem hesaplama maliyetini hem de bellek kullanımını ciddi şekilde azaltır ve yüksek boyutlu görüntü verilerinin işlenmesini mümkün kılar.

Modern CNN mimarileri, geleneksel yapılardan çok daha gelişmiş teknikler içermektedir. VGGNet, ResNet, DenseNet, EfficientNet gibi mimariler, daha derin ve optimize edilmiş ağ yapılarıyla farklı görevlerde yüksek performans sergilemektedir. Özellikle özellik piramit ağları, çok ölçekli temsil öğrenimi sağlayarak farklı boyutlardaki nesnelerin tespit edilmesine olanak tanımaktadır (Ketkar vd. 2021). Ayrıca son yıllarda geliştirilen dikkat mekanizmaları, modelin önemli bölgeleri ön plana çıkarmasına ve daha isabetli tahminler yapmasına yardımcı olmaktadır.

CNN tabanlı nesne algılama mimarilerinde, eğitim süreci genellikle iki tür kayıp fonksiyonunun birlikte optimize edilmesine dayanır: sınıflandırma kaybı ve konum kaybı. Böylece model hem nesneleri doğru sınıflandırmayı hem de doğru konumlandırmayı öğrenir. Çıkarım aşamasında ise Maksimum Olmayan Bastırma tekniği kullanılarak aynı nesneye ait birden fazla tahmin elenir ve yalnızca en güçlü tahmin korunur (Türkoğlu vd. 2020).

Sonuç olarak, CNN günümüzde görüntü sınıflandırma, nesne algılama, yüz tanıma, tıbbi görüntü analizi, video işleme ve otonom sürüş gibi birçok alanda temel bir yapı taşı olarak kullanılmaktadır. Bu tez kapsamında ise CNN tabanlı derin öğrenme mimarileri, özellikle yaya tespiti ve akıllı trafik yönetimi sistemlerinde nesne algılama süreçlerinin başarımını artırmak amacıyla incelenmektedir.

3.3.1.1 Evrişim Katmanları

Evrişim katmanları, CNN temel bileşenlerinden biridir (Özdemir ve Türkoğlu 2022). Bu katmanlar, CNN'in "öğrenme" gücünün ana kaynağı olup modelin görsel örüntüleri otomatik olarak öğrenmesini sağlar ve giriş görüntüsündeki yerel özellikleri algılamak için filtre adı verilen küçük matrisleri kullanır. Her filtre, görüntü üzerinde kaydırılarak belirli bir özellik haritası üretir. Bu sayede kenar, köşe, doku gibi düşük seviyeli özellikler tespit edilir. Bu filtreler öğrenilebilir parametrelerdir ve eğitim sürecinde veriye adapte olur. Evrişim işlemi sırasında ağırlık paylaşımı tekniği kullanılarak parametre sayısı azaltılır ve ağırlık genelleme yeteneği artar. Derin katmanlarda ise bu özellikler daha soyut hale gelir; örneğin insan yüzü, araç veya yaya gibi nesneleri temsil eden yüksek seviyeli özellikler yakalanabilir.

3.3.1.2 Havuzlama Katmanları

Havuzlama katmanları, evrişim katmanlarından sonra gelen ve özellik haritalarının boyutunu azaltmak için kullanılan katmanlardır (Jie ve Wanda 2020). Bu katmanlar, Özellik haritalarının boyutunu küçültmek, parametre sayısını azaltmak ve konumsal değişimlere karşı modelin dayanıklılığını artırarak modelin genel performansını artırmak için kullanılır. En yaygın kullanılan havuzlama türleri arasında Maksimum Havuzlama ve Ortalama Havuzlama bulunur. Maksimum Havuzlama, belirli bir alandaki en yüksek değeri seçerek önemli özellikleri korurken, Ortalama Havuzlama o alandaki ortalama değeri alır. Bu işlem, görüntüdeki küçük konum değişikliklerine karşı modelin daha kararlı hale gelmesini sağlar. Böylece CNN, farklı ölçek ve pozisyonlarda bulunan nesneleri daha doğru bir şekilde tanıyabilir.

- Özellik haritalarının boyutunu küçültmek, parametre sayısını azaltmak ve konumsal değişimlere karşı modelin dayanıklılığını artırmak için kullanılır. Maksimum havuzlama, belirli bir bölgede en baskın özelliği seçerken; ortalama havuzlama, özelliklerin ortalama değerini alır.

3.3.1.3 Aktivasyon Fonksiyonları

Aktivasyon fonksiyonları, sinir ağlarındaki her nöronun çıktısına doğrusal olmayanlık kazandırmak için kullanılır (Apicella vd. 2021). Bu sayede ağ, karmaşık ilişkileri ve örüntüleri öğrenebilir hale gelir. Evrimsel Sinir Ağlarında en sık kullanılan aktivasyon fonksiyonu ReLU (Düzeltilmiş Doğrusal Birim)'dur. ReLU fonksiyonu negatif girişleri sıfıra eşitlerken pozitif girişleri olduğu gibi geçirir. Bu yaklaşım, hesaplama maliyetini azaltır ve gradyan sönmesi problemini hafifletir. Bunun dışında Leaky ReLU, Sigmoid, Tanh gibi alternatif aktivasyon fonksiyonları da kullanılabilir. Uygulamada, ReLU ve türevleri derin ağlarda daha başarılı sonuçlar vermektedir.

3.3.1.4 Tam Bağlı Katmanları

Tam bağlı katmanlar, CNN mimarisinin son aşamalarında yer alır ve evrişim ile havuzlama katmanlarından elde edilen özellikleri kullanarak sınıflandırma işlemini gerçekleştirir (Basha vd. 2020). Bu katmanlarda her nöron, bir önceki katmandaki tüm nöronlarla bağlantılıdır; bu nedenle “tam bağlı” olarak adlandırılır. Bu katman, yüksek seviyeli özelliklerin birleştirilmesiyle görüntüdeki nesnelerin hangi sınıfa ait olduğunu belirler. Çıktı katmanında genellikle Softmax fonksiyonu kullanılarak, her sınıf için olasılık değerleri üretilir. Tam bağlı katmanların yoğun parametre yapısı, modelin güçlü öğrenme kapasitesine sahip olmasını sağlar; ancak bu durum aynı zamanda aşırı öğrenme riskini de artırabilir. Bu nedenle eğitim sürecinde bırakma gibi düzenleme yöntemleri sıklıkla kullanılır.

3.3.2 Nesne Tespiti ve YOLO Mimarisi

Nesne Tespiti, bilgisayarla görme ve yapay zeka alanında görüntüler veya videolar üzerindeki nesneleri tanımlama, konumlandırma ve sınıflandırma sürecidir. Bu teknik, bir sahnede bulunan farklı nesnelerin hem türlerini hem de konumlarını belirlemektedir. Böylece nesnelerin çevreye göre nasıl konumlandığı ve etkileşime girdiği anlaşılmaktadır. Derin öğrenme tabanlı nesne algılama modelleri, özellikle CNN, nesne algılama görevlerinde doğruluk ve hız açısından büyük avantaj sağlamaktadır (Zou vd. 2023). Modelin eğitim sürecinde, büyük miktarda etiketlenmiş veriyle nesnelerin çeşitli perspektif, boyut ve açılardaki görüntülerini öğrenmesi sağlanmaktadır ve bu sayede karmaşık sahnelerde farklı nesneleri tespit edebilme kapasitesine ulaşılmaktadır.

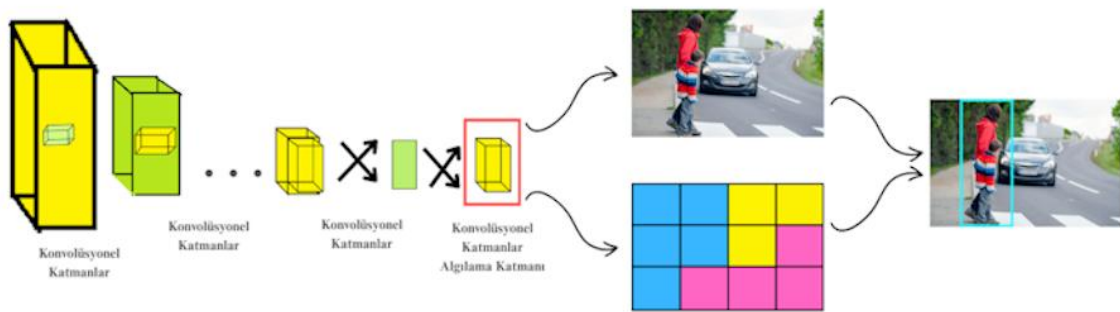
Nesne algılama, iki temel bileşen olan sınıflandırma ve konumlandırmadan oluşur. Öncelikle model, görüntüdeki her bir nesnenin hangi sınıfa ait olduğunu tahmin eder. Ardından, her nesnenin görüntü içerisindeki konumunu belirlemek için sınır kutularını oluşturur. Algoritmaların performansı; doğruluk, hız ve sahne karmaşıklığı gibi metriklerle değerlendirilir.

Tek aşamalı nesne algılama yöntemleri, giriş görüntüsünü yalnızca bir kez işleyerek nesnelerin varlığını ve konumunu tahmin eder. Bu yöntemler, sınıflandırma ve konumlandırma işlemlerini tek bir geçişte gerçekleştirir. Böylece, hesaplama açısından verimlilik sağlar ve kaynak kısıtlı ortamlarda nesneleri gerçek zamanlı olarak tespit etmek için kullanılabilir. Ancak, tek aşamalı algılama genellikle diğer yöntemlere göre daha düşük doğruluk sağlar ve küçük nesneleri tespit etmede daha az etkilidir. Tek aşamalı modellerden olan YOLO algoritması, hız avantajı sayesinde gerçek zamanlı uygulamalarda öne çıkar. YOLO, sınıflandırma ve konumlandırmayı tek bir geçişte yaparak bu süreçleri önemli ölçüde hızlandırır.

Nesne algılamanın temel avantajı, çeşitli endüstrilerde geniş uygulama alanı sunmaktadır. Otonom sürüş alanında, araçların çevresindeki yayaları, diğer araçları ve nesneleri tespit etmesini sağlamaktadır. Güvenlik alanında ise kamera görüntülerinden şüpheli durumları tespit ederek kazaları veya güvenlik ihlallerini önlemeye yardımcı olmaktadır. Sağlık

alanında ise medikal görüntülerde tümör tespiti gibi anormallikleri saptayarak tanı sürecini hızlandırıp destek sağlamaktadır (Kazdal 2013). Nesne algılama, ayrıca üretim tesislerindeki kalite kontrol süreçlerini otomatikleştirmekte ve akıllı şehir uygulamalarında trafiğin düzenlenmesine katkıda bulunmaktadır. Bu nedenle, nesne algılama, gelişen algoritmalar ve derin öğrenme yöntemleriyle bilgisayarla görme alanında önemli bir yere sahiptir.

YOLO, derin öğrenme tabanlı nesne algılama algoritmaları arasında gerçek zamanlı ve yüksek doğruluklu performansı ile öne çıkan bir modeldir (Tan vd. 2021). Geleneksel nesne algılama yöntemlerinde tespit işlemi, çok adımlı bir süreçten geçerken, YOLO tek bir ileri besleme aşamasında tüm nesnelerin konum ve sınıf tahminlerini gerçekleştirebilmektedir. Bu yaklaşım, görüntünün tamamını tek bir analiz adımıyla işleyerek hem işlem süresini kısaltmakta hem de modelin gerçek zamanlı uygulamalara uyum sağlamasına olanak tanımaktadır. Böylece görüntüdeki tüm nesneleri aynı anda tanımlayarak nesne tespitini hızlı ve verimli bir şekilde gerçekleştirmektedir. Nesne tespiti, görüntü veya video akışlarında belirli nesnelerin tespiti, sınıflandırılması ve sınır kutularının belirlenmesini içermektedir. YOLO, diğer nesne tespit algoritmalarına kıyasla çok daha düşük gecikmeyle çalışır ve bu yüzden özellikle otonom sürüş, güvenlik sistemleri ve robotik gibi hızın hayati önem taşıdığı alanlarda kullanılmaktadır. Bu özellikleri, YOLO'yu gerçek zamanlı uygulamalarda ön plana çıkararak nesne algılamada önemli bir yere getirmiştir.



Şekil. 3.2 YOLO Algoritmasının Çalışma Prensibi

Şekil 3.2’de Yolo Algoritmasının temel prensibi verilmiştir. YOLO’nun çalışma prensibi, verilen girdiyi belirli bir ızgara sistemine bölerek nesneleri tahmin etmeye dayanmaktadır. Görüntünün tamamını işlemek için tek bir sinir ağı kullanmakta, ardından görüntüyü ızgaralara bölmektedir. Bu ızgara hücreleri, belirli bir sınıfa ait nesne olup olmadığını analiz etmekte ve sınır kutularını tahmin etmektedir. Her hücre, bir nesne içermesi durumunda sınır kutusunun konum bilgilerini ve sınıf olasılıklarını üretmektedir. YOLO, nesne algılama sürecinde konumlandırma ve sınıflandırmayı aynı anda yaparak, tespit işlemini tek adımda tamamlamaktadır. YOLO’nun görüntüdeki tüm nesneleri aynı anda algılayabilmesi hem işlem süresini kısaltmakta hem de geniş ölçekli veri setlerinde uygulanabilirliğini artırmaktadır. Bu özellikler sayesinde YOLO, yüksek hızda analiz yaparken sınıf tahmininde de başarılı sonuçlar elde edilmektedir.

Bunlara ek olarak YOLO, nesne algılamada hızlı ve hassas sonuçlar sunması bakımından öne çıkan avantajlı bir modeldir (Murat 2021). Diğer birçok algoritmanın aksine, YOLO tek bir analizle tüm görüntüyü işleyerek çok daha hızlı nesne algılaması sağlamaktadır. Bu hızlı işlem yeteneği, nesne algılama sürecindeki yanlış pozitif ve yanlış negatif oranını düşürürken doğruluğu artırmaktadır. YOLO ayrıca, yüksek kare hızında çalışabilme özelliğiyle video analizlerinde oldukça başarılı sonuçlar vermektedir. Bu özellikler, modeli gerçek zamanlı analiz gerektiren uygulamalar için ideal bir seçenek haline getirmektedir. YOLO’nun sunduğu hız, doğruluk ve güvenilirlik geniş bir uygulama alanı sunmaktadır.

YOLO, bilgisayarlı görüş alanında devrim yaratan popüler bir nesne algılama algoritmasıdır. Hızlı ve verimlidir, bu da onu gerçek zamanlı nesne algılama görevleri için mükemmel bir seçim haline getirmektedir. Çeşitli kıyaslamalarda en son teknoloji performansına ulaşmış ve çeşitli gerçek dünya uygulamalarında yaygın olarak benimsenmiştir. YOLO’nun başlıca avantajlarından biri, görüntüleri gerçek zamanlı olarak işlemesine olanak tanıyan hızlı çıkarım hızıdır. Video gözetimi, otonom araçlar ve artırılmış gerçeklik gibi uygulamalar için oldukça uygundur. Ayrıca, YOLO basit bir mimariye sahiptir ve minimum eğitim verisi gerektirir, bu da yeni görevlere uygulanmasını ve uyarlanmasını kolaylaştırır.

Küçük nesnelerle mücadele ve ince taneli nesne sınıflandırması yapamama gibi sınırlamalara rağmen, YOLO nesne tespiti için değerli bir araç olduğunu kanıtladı ve araştırmacılar ve uygulayıcılar için birçok yeni olasılık açmıştır.

YOLO'nun öne çıkan avantajları şunlardır:

- Hızlı çıkarım süresi: Tek adımlı işleme sayesinde yüksek kare hızında (FPS) tahmin yapabilmektedir.
- Yüksek doğruluk: Optimize edilmiş mimarisi, yanlış pozitif ve yanlış negatif oranlarını minimize eder.
- Gerçek zamanlı kullanım imkânı: Video analizi, canlı gözetim ve otonom kontrol sistemlerinde etkili şekilde çalışır.
- Donanım uyumluluğu: GPU tabanlı hızlandırmaya uygun olduğu gibi, optimize edilmiş sürümleri düşük güçlü cihazlarda da çalışabilmektedir.

YOLO algoritmaları, nesne algılama ve bilgisayarla görme alanında yüksek hız ve doğruluk performansı ile önemli bir yere sahiptir. Aşağıda YOLOv5'ten YOLOv12'ye kadar olan sürümlerin temel özelliklerini içeren karşılaştırmalı tablo verilmiştir. Çizelge 3.1 de her sürümün öne çıkan teknik özelliklerini ve kullanım alanlarını özetlemektedir.

Çizelge 3.1 YOLO Sürümlerinin Karşılaştırılması

YOLO Sürümü	Çıkış Tarihi	Model Boyutları	Parametre Sayısı	Katman Sayısı	Doğruluk (mAP)	Ağ Çıkışı
YOLOv5	2020	Küçük, Orta, Büyük, X-Large	~7M - ~76M	225 - 365	İyi	Tek çıkış
YOLOv7	2022	Küçük, Orta, Büyük, X-Large	~7M - ~100M	244 - 490	Çok İyi	Çoklu çıkış

YOLO Sürümü	Çıkış Tarihi	Model Boyutları	Parametre Sayısı	Katman Sayısı	Doğruluk (mAP)	Ağ Çıkışı
YOLOv8	2023	Nano, Küçük, Orta, Büyük	~3M - ~87M	195 - 390	Yüksek doğruluk	Optimize edilmiş çıkış
YOLOv9	2024	Küçük, Orta, Büyük, X-Large	~3M - ~88M**	200 - 420**	Yüksek doğruluk	Optimize edilmiş çoklu çıkış**
YOLOv10	2024	Küçük, Orta, Büyük, X-Large	~3M - ~90M	205 - 400	Yüksek doğruluk	Optimize edilmiş çoklu çıkış
YOLOv11	2025	Nano, Küçük, Orta	~2.5M - ~85M**	190 - 380**	Yüksek doğruluk	Optimize edilmiş çoklu çıkış
YOLOv12	2025	Küçük, Orta, Büyük	~3M - ~92M*	210 - 430*	Yüksek doğruluk	Optimize edilmiş çoklu çıkış

* Bu veriler tahmini olup, literatürde yer alan mimari gelişmelerden yola çıkarak belirlenmiştir.

3.3.2.1 YOLOv5

YOLOv5, 2020 yılında Ultralytics tarafından geliştirilmiş ve PyTorch tabanlı bir yapıya sahip bir nesne algılama modelidir (Jocher 2020). Öne çıkan özellikleri arasında dinamik ankraj kutuları yer alır. Bu özellik, nesne boyutlarına daha iyi uyum sağlamak amacıyla kümeleme algoritmalarını kullanarak geliştirilmiştir. Mekansal Piramit Havuzlama (SPP) özelliği ise küçük nesne algılama performansını artırmaya yöneliktir. Model, EfficientDet tabanlı bir mimariyle donatılmış olup bu sayede daha yüksek doğruluk ve geniş nesne kategorisi desteği sunmaktadır. YOLOv5, hız ve doğruluk arasında dengeli bir yapı sağlayarak farklı donanım gereksinimlerine uyum sağlamaktadır.

3.3.2.2 YOLOv7

2022 yılında piyasaya sürülen YOLOv7, nesne algılama algoritmaları arasında hız ve enerji verimliliği açısından önemli bir gelişme sunmaktadır (Wang vd. 2022). Bağlantı kutuları, modelin daha geniş bir nesne çeşitliliğini algılama kapasitesini artırmaktadır. Ayrıca odak kaybı fonksiyonu sayesinde zor örneklerle daha fazla dikkat verilerek küçük nesnelerin tespiti iyileştirilmiştir. YOLOv7, 608x608 piksel çözünürlükte çalışarak daha küçük nesnelerin algılanmasını kolaylaştırmaktadır. Gerçek zamanlı uygulamalarda ise saniyede 155 karelik hızlı çıkarım süresi ile öne çıkmaktadır.

3.3.2.3 YOLOv8

YOLOv8, Ultralytics tarafından geliştirilen en güncel YOLO sürümüdür (Jocher vd. 2023). Yeni bir API ile birlikte gelen model, eğitim ve çıkarım süreçlerini daha kullanıcı dostu hale getirmiştir. YOLOv8, optimize edilmiş mimarisi sayesinde hem doğruluk hem de hız açısından önceki sürümlere kıyasla üstün performans sunmaktadır. Model, küçük nesne tespiti, çoklu nesne algılama ve sınırlı kaynaklı cihazlar üzerinde daha verimli çalışacak şekilde tasarlanmıştır. Ayrıca, veri artırımı ve optimizasyon teknikleriyle eğitim sürecinde daha iyi genelleme sağlamaktadır. YOLOv8, akıllı şehir uygulamaları, güvenlik sistemleri, sağlık teknolojileri ve otonom araçlar gibi pek çok alanda etkin bir şekilde kullanılabilir.

3.3.2.4 YOLOv9

YOLOv9, önceki sürümlerden farklı olarak ağ katmanlarını geri çevrilebilir yaparak veri kaybını önleyen tersine çevrilebilir fonksiyonlar özelliğini sunmaktadır (Wang ve Liao 2024). Bu yaklaşım, gradyan dengesizliğini azaltarak modelin daha verimli öğrenmesini sağlamaktadır. Ek olarak, programlanabilir gradyan bilgisi ile gradyan hesaplamaları sırasında doğruluk artırılmakta ve çıkarım hızı korunmaktadır. Model, CSPNet ve ELAN mimarilerinin birleşimi olan GELAN sayesinde gerçek zamanlı performansı artırarak daha hassas nesne tespiti sunmaktadır. YOLOv9, daha az parametre kullanmasına rağmen

doğruluk ve işlem hızında önemli iyileştirmeler sağlamaktadır. Özellikle küçük nesne tespitiinde kayda değer bir performans artışı göstermektedir.

3.3.2.5 YOLOv10

YOLOv10, gerçek zamanlı uyarlanabilirlik sağlayan bir model olarak endüstriyel uygulamalara esneklik sunmaktadır (Ao Wang Hui Chen 2024). Bu model, üretimde kusur tespiti veya tarımda hassas gözlem gibi spesifik alanlarda üstün performans göstermektedir. Karmaşık ve yoğun sahnelerde, özellikle şehir trafiği gibi ortamlarda, gelişmiş özellik çıkarım teknikleriyle daha doğru nesne tespiti yapabilmektedir. Modelin yüksek verimliliği hem CPU hem de GPU cihazlarında optimize edilmiş bir şekilde çalışmasına olanak tanımaktadır.

3.3.2.6 YOLOv11

YOLOv11, özellikle düşük gecikmeli ve yüksek doğruluklu nesne algılama gereksinimlerini hedefleyen bir model olarak geliştirilmiştir (Khanam ve Hussain 2024). Bu sürümde, Çok Ölçekli Özellik Harmanlama modülü ile hem küçük hem büyük nesneler üzerinde daha dengeli bir algılama performansı sağlanmıştır. Ek olarak, Dönüşümsel Konvolüsyonlar sayesinde model, ışık değişimi, açı farklılıklarına karşı daha dayanıklı hale getirilmiştir. YOLOv11, ileri seviye "hafif ağ" tasarımıyla mobil ve gömülü cihazlarda da üstün performans sunarak gerçek zamanlı uygulamalar için önemli avantajlar sağlamaktadır. Özellikle yoğun şehir trafiği gibi zorlu ortamlarda daha hızlı ve isabetli nesne tespiti performansı göstermektedir.

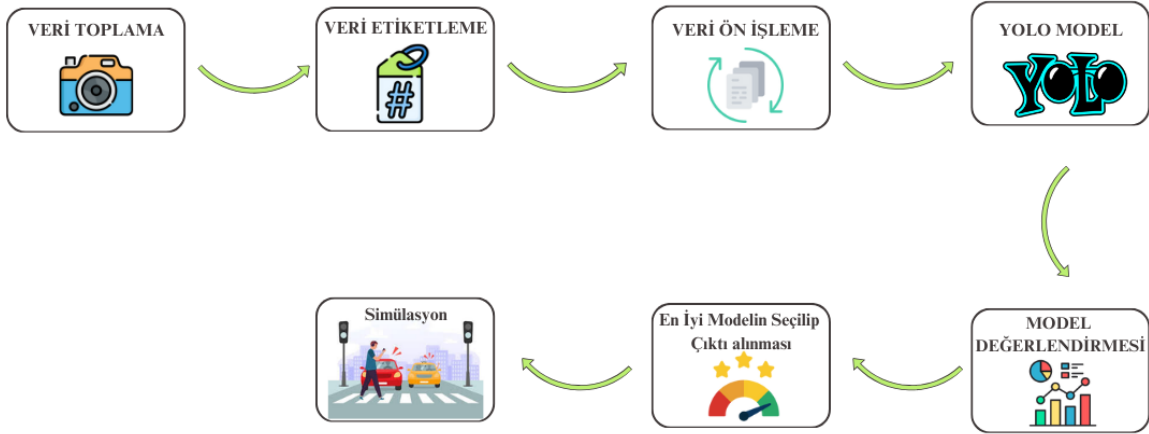
3.3.2.7 YOLOv12

YOLOv12, nesne algılamada devrim niteliğinde yenilikler sunarak adaptif öğrenme yeteneklerini öne çıkarmıştır (Tian vd. 2025). Dinamik Karar Katmanları kullanılarak model, farklı sahne türlerine otomatik uyum sağlayabilmekte ve algılama kesinliğini ortam koşullarına göre optimize edebilmektedir. Ayrıca, Enerji Verimli Çıkarım Motoru entegrasyonu ile pil ömrü sınırlı cihazlarda bile uzun süreli gerçek zamanlı kullanım

olanađı sunmaktadır. YOLOv12, hiperparametre optimizasyonunda otomatik iyileřtirme teknikleri kullanarak hem eđitim sũresini kısaltmakta hem de daha yũksek dođruluk sađlamaktadır. Tıp, tarım ve akıllı řehir sistemlerinde kritik gũrevlerde kullanım iin olduka uygundur.

4. MATERYAL ve METOT

Bu çalışmada, YOLO mimarisi kullanılarak geliştirilen nesne algılama projesinin iş akışı Şekil 4.1'de gösterilmektedir.



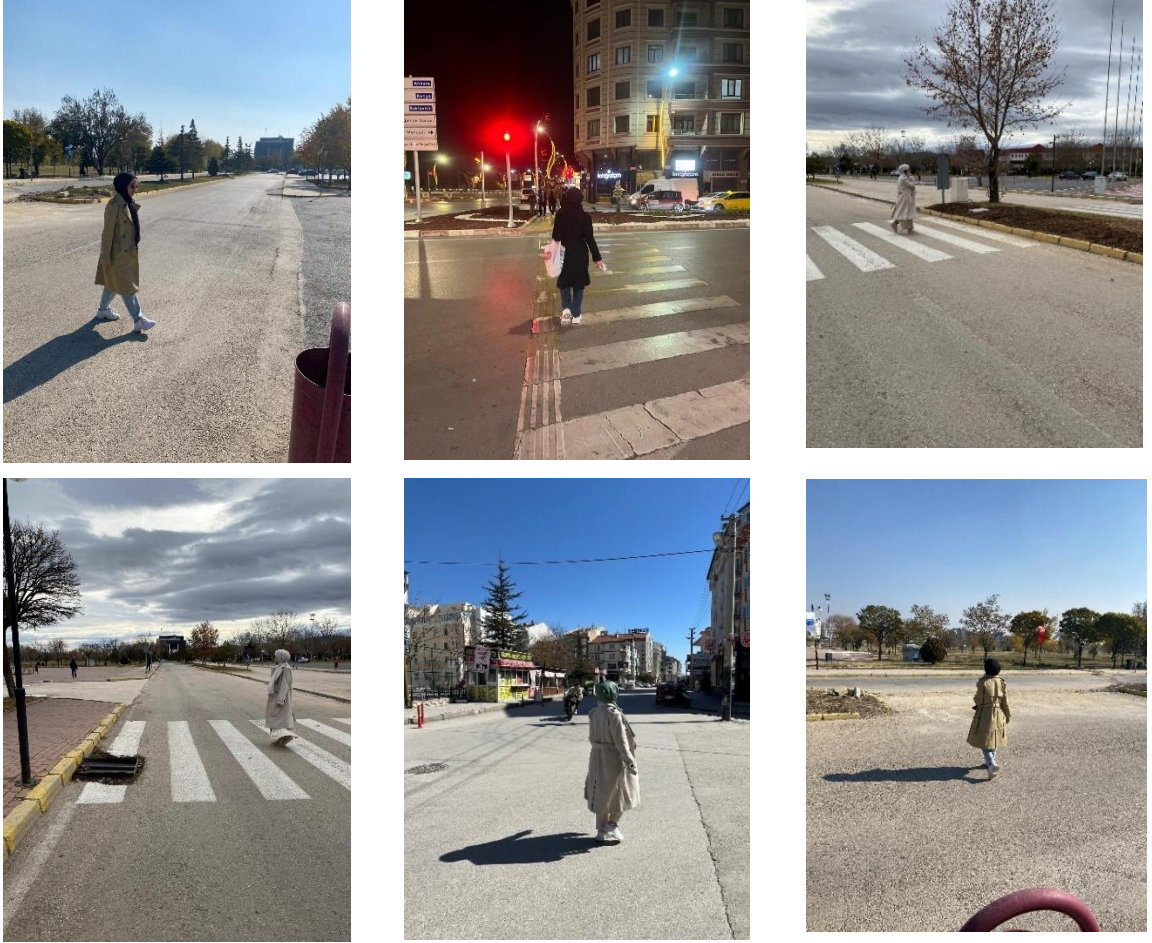
Şekil 4.1 Yapılan Çalışmanın Akış Şeması

4.1 Veri Toplama

İlk olarak, nesne algılama için gerekli veri setini oluşturmak üzere veriler toplandı. Ardından, veri setindeki görüntüler bu araştırma için özel olarak sıfırdan bir yaya veri seti hazırlandı. Veri seti, gerçekçi görsel çeşitlilik sağlamak için doğal dış mekân koşullarında iPhone 16 kamera kullanılarak çekilen 620 adet yüksek çözünürlüklü yaya görüntüsünden oluşmaktadır. Fotoğraflar, gerçek dünyadaki yaya hareket senaryolarını yansıtmak için çeşitli aydınlatma koşulları, açılar ve kalabalık yoğunluklarında çekilmiştir. Her görüntünün çözünürlüğü 4032×3024 pikseldir ve hareket tutarlılığını ve netliğini korumak için saniyede 30 kare (FPS) hızında kaydedilmiştir. Veri seti, tarafımdan sıfırdan hazırlanmış olup hem tekil hem de çoklu yaya durumlarını içeren geniş bir örneklem sunmaktadır.

Toplanan görüntülerde görünen tüm katılımcılar bilgilendirilmiş onam vermiş olup, tüm veri toplama prosedürleri Kişisel Verilerin Korunması Kanunu (KVKK) uyarınca gerçekleştirilmiştir. Kişisel olarak tanımlanabilir hiçbir bilgi saklanmamış ve tüm veriler işlenmeden önce anonimleştirilerek etik ve gizlilik kurallarına uygun veri seti

hazırlanması sağlanmıştır. Resim 4.2’te sunulan örnek görüntüler, veri seti kapsamında yer alan farklı yaya senaryolarını temsil etmekte olup; gündüz ve gece çekimleri, farklı yaya pozisyonları, tekil ve çoklu yaya durumları ile yaya geçidi gibi gerçek trafik ortamlarını içeren sahneleri göstermektedir. Bu çeşitlilik, geliştirilen yaya tespit modelinin gerçek dünya koşullarında genellenebilirliğini ve dayanıklılığını artırmayı amaçlamaktadır.



Resim 4.2. Veri seti içerisindeki örnek görüntüler

4.2 Veri Etiketleme

Bu veri seti, yaya tespiti ve sınıflandırma gibi derin öğrenme algoritmalarının eğitimi ve doğrulaması için tasarlanmıştır. Fotoğraflar, farklı açılar, ışık koşulları ve yoğunluk seviyelerini içerecek şekilde çeşitli senaryoları kapsamaktadır. Böylece, modelin gerçek dünyadaki yaya hareketlerini etkili bir şekilde algılayıp yönetebilmesi hedeflenmiştir.

Oluşturulan veri setindeki 620 adet görüntü “yaya” etiket adı ile make sense uygulaması ile etiketleme işlemi gerçekleştirilmiştir.

4.3 Veri Ön İşleme

Veri ön işleme aşamasında, öncelikle tüm görüntüler, YOLO modelinin gereksinimlerine uygun bir formata dönüştürülmüştür. Bu aşama, her bir görselin doğru şekilde etiketlenmesi ve modelin öğrenmesi için uygun hale getirilmesini içerir. Görsellerin boyutları, YOLO'nun optimal performansı için 640x640 piksele normalize edilmiştir. Bu adım, modelin giriş boyutunu sabitleyerek daha hızlı ve verimli bir şekilde işlem yapmasını sağlar. Ayrıca, görüntülerin kalitesiz olanları (bulanık, düşük çözünürlüklü, ya da bozulmuş görseller) tespit edilerek veri setinden çıkarılmıştır. Bu şekilde, yalnızca net ve anlamlı görsellerin modelin eğitimine dahil edilmesi sağlanmıştır. Bu ön işleme süreci, modelin doğru ve hızlı sonuçlar üretmesine katkıda bulunmuş, aynı zamanda veri setindeki hatalı etiketlerden ve bozulmuş görsellerden kaynaklanabilecek olumsuz etkiler azaltılmıştır.

Veri kümesi üç alt kümeye ayrıldı: %80'i eğitim, %10'u doğrulama ve %10'u test için. Bu bölünme, tarafsız bir değerlendirme sürecini korurken model öğrenimi için yeterli çeşitliliği sağlamaktadır. Her alt küme hem tek yaya hem de çoklu yaya örnekleri içerir, böylece model farklı gerçek dünya senaryolarında etkili bir şekilde genelleme yapılabilmektedir.

4.4 Model

Bu bölümde, yaya tespiti amacıyla kullanılan derin öğrenme tabanlı YOLO mimarisinin farklı sürümleri ayrıntılı olarak ele alınmıştır. Çalışmada, YOLOv8, YOLOv11 ve YOLOv12 modelleri değerlendirilmiş; bu modellerin yapısal farklılıkları, dikkat mekanizmaları, özellik birleştirme stratejileri ve çıkış katmanları açısından karşılaştırılmıştır. Ayrıca her bir modelin mimari bileşenleri olan Omurga, Boyun ve Çıkış başlığı kısımları detaylı biçimde açıklanmıştır. Bölümün sonunda, söz konusu üç modelin temel özellikleri, performansları ve kullanım avantajları tablo halinde sunulmaktadır.

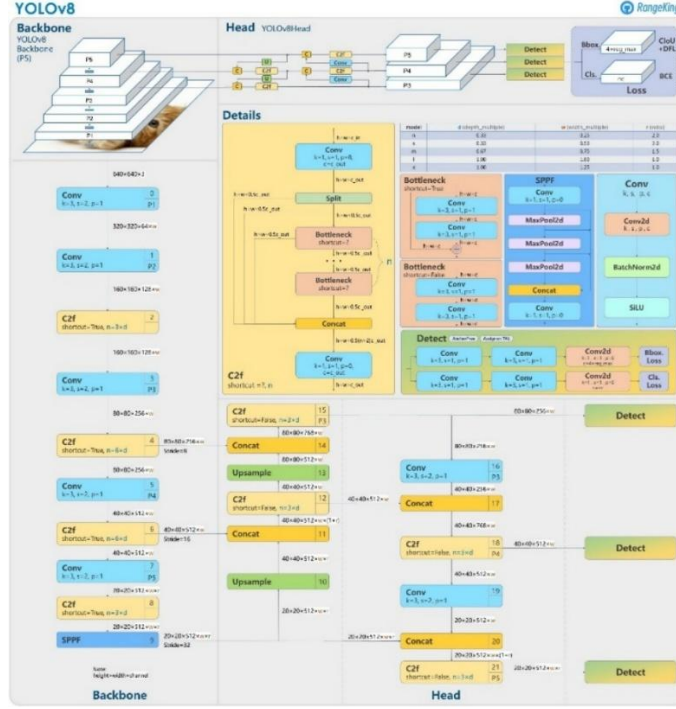
karşılaştırmalı bir değerlendirme yapılmıştır. Bu karşılaştırma, yaya tespiti ve akıllı trafik yönetimi gibi gerçek zamanlı uygulamalarda hangi modelin daha uygun olduğunu belirlemek açısından önemli bir temel oluşturmaktadır.

Bu çalışmanın bir parçası olarak gerçekleştirilen tüm deneyler, Google Colab ortamında Python programlama dili kullanılarak gerçekleştirildi. Sistem, 16 GB belleğe sahip bir NVIDIA T4 GPU ile donatılmıştı ve Colab ek olarak 25 GB RAM tahsis etmiştir. Model eğitimi ve değerlendirmesi, geliştirilen yaya veri seti ve YOLOv8, YOLOv11 ve YOLOv12 dahil olmak üzere YOLO modelinin çeşitli sürümleri kullanılarak gerçekleştirildi.

Eğitim sırasında, adil bir karşılaştırma sağlamak için her model farklı dönem ayarları altında aynı veri seti üzerinde eğitildi. Her yapılandırma için performans ölçütleri kaydedildi. Google Colab'ın kullanımı, GPU tabanlı derin öğrenme deneyleri için esnek ve erişilebilir bir platform sağladı ve özel yerel donanım ihtiyacını ortadan kaldırdı.

4.4.1 YOLOv8

YOLOv8 mimarisinin genel yapısı, çalışmada kullanılan modelin temel bileşenlerini şekil 4.3 de görsel olarak ortaya koymaktadır. Bu mimari üç ana bölümden oluşmaktadır: Omurga, Boyun ve Çıkış başlığı. Omurga katmanında temel özellik çıkarımı gerçekleştirilirken, Boyun bölümü farklı çözünürlük seviyelerindeki öznitelikleri bir araya getirerek çok ölçekli bilgi akışını sağlamaktadır (Ouassine vd. 2025). Çıkış başlığı kısmı ise nesne tespiti için yapıldığı son katman olarak görev almaktadır. Bu çok katmanlı yapı, YOLOv8 modelinin hem hız hem de doğruluk açısından dengeli bir performans ortaya koymasına olanak tanımaktadır.



Şekil 4.3 YOLOv8 mimarisinin genel yapısı ((Jocher vd. 2023))

YOLOv8, Ultralytics tarafından geliştirilen ve YOLO serisinin en yaygın kullanılan sürümlerinden biridir. Modern mimari tasarımı sayesinde, farklı boyutlardaki modelleri (YOLOv8n, YOLOv8s, YOLOv8m, YOLOv8l, YOLOv8x) ile esnek bir kullanım sunmaktadır.

Omurga bölümünde, C2f blokları kullanılmaktadır. Bu yapı, önceki sürümlerdeki CSP bloklarının geliştirilmiş bir versiyonudur ve daha derin ağlarda bilgi kaybını azaltarak özelliklerin etkin bir şekilde taşınmasını sağlar. Ayrıca, odak katmanı yerine doğrudan evrimsel işlemler kullanılarak ağ daha verimli hale getirilmiştir.

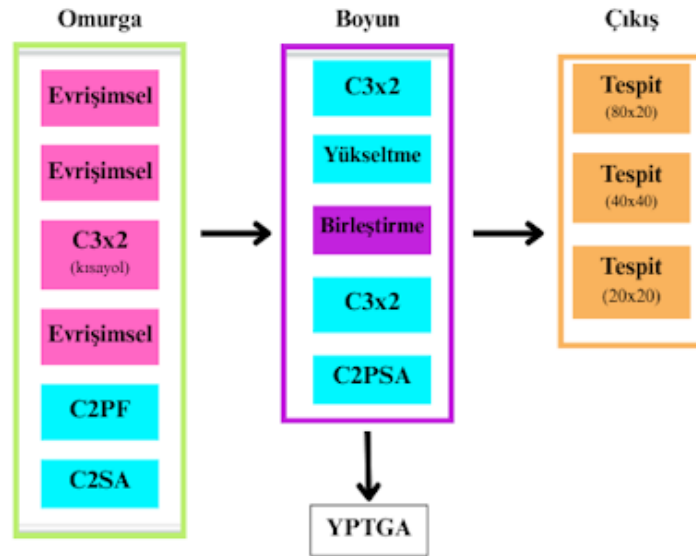
Boyun kısmında, PANet tabanlı bir özellik birleştirme mekanizması kullanılmaktadır. Burada, farklı çözünürlüklerden gelen öznitelik haritaları FPN ile desteklenerek birleştirilir. Bu sayede hem küçük hem de büyük nesnelerin aynı anda doğru şekilde tespit edilmesi mümkün olmaktadır. YOLOv8 ayrıca, daha hafif yükseltme yöntemleri ile hız avantajı sağlamaktadır.

Çıkış başlığı bölümünde, önceki YOLO sürümlerinden farklı olarak çapa gerektirmeyen tespit yaklaşımı benimsenmiştir. Bu yöntem, sabit önceden tanımlanmış çapa kutularına bağlı kalmadan, doğrudan nesne merkezleri ve boyutları üzerinden tahmin yapılmasını sağlar. Bu sayede hem eğitim süreci hızlanmakta hem de özellikle küçük nesnelerin tespitinde daha yüksek doğruluk elde edilmektedir.

YOLOv8'in öne çıkan özelliklerinden biri, sadece nesne tespiti için değil, aynı zamanda segmentasyon ve görüntü sınıflandırma görevleri için de entegre edilebilir bir yapıya sahip olmasıdır (Lu 2023). Bu çok yönlülük, YOLOv8'i modern bilgisayarla görüş uygulamalarında yaygın olarak kullanılan bir mimari haline getirmiştir.

4.4.2 YOLOv11

YOLOv11 mimarisi, geleneksel YOLO yapısının gelişmiş bir versiyonu olarak, nesne tespitinde doğruluk ve hız performansını artırmak amacıyla optimize edilmiştir (Alif 2024). Şekil 4.4 de görülen YOLOv11 mimarisi omurga, boyun ve çıkış başlığı üç temel bileşenden oluşmaktadır.



Şekil 4.4. YOLOv11 Mimarisinin Genel Yapısı

Omurga, giriş görüntüsünden çok ölçekli özelliklerin çıkarılmasını sağlar. Bu aşamada sırasıyla konvolüsyon katmanları ve gelişmiş bloklar yer almaktadır. Evrişimsel katmanlar, görüntüdeki temel özellikleri çıkarmak için kullanılır. C3x2, derin konvolüsyon blokları olup, kısayol bağlantıları sayesinde daha verimli öğrenme sağlar (Kotthapalli vd. 2025). Bu yapı, artık bağlantılarla bilgi kaybını azaltır. C2PF, özelliklerin farklı yollar üzerinden taşınmasını sağlayarak, daha zengin bir temsil üretir. C2SA, kanal ve mekânsal dikkat mekanizmaları kullanarak, modelin önemli bölgelere ve kanallara odaklanmasını sağlar.

Boyun farklı seviyelerdeki öznitelik haritalarını birleştirerek tespit için daha anlamlı hale getirir. C3x2 Blokları, derin öznitelik işleme için kullanılır. Yükseltme Katmanları, düşük çözünürlüklü özellik haritalarını büyütür ve farklı çözünürlüklerle birleştirilmesini sağlar. Birleştirme, farklı seviyelerdeki öznitelikleri birleştirerek, çok ölçekli bilgi akışını güçlendirir. C2PSA hem kanal hem de mekânsal dikkat katmanlarını entegre ederek, daha hassas ve odaklı öznitelik temsilleri üretir.

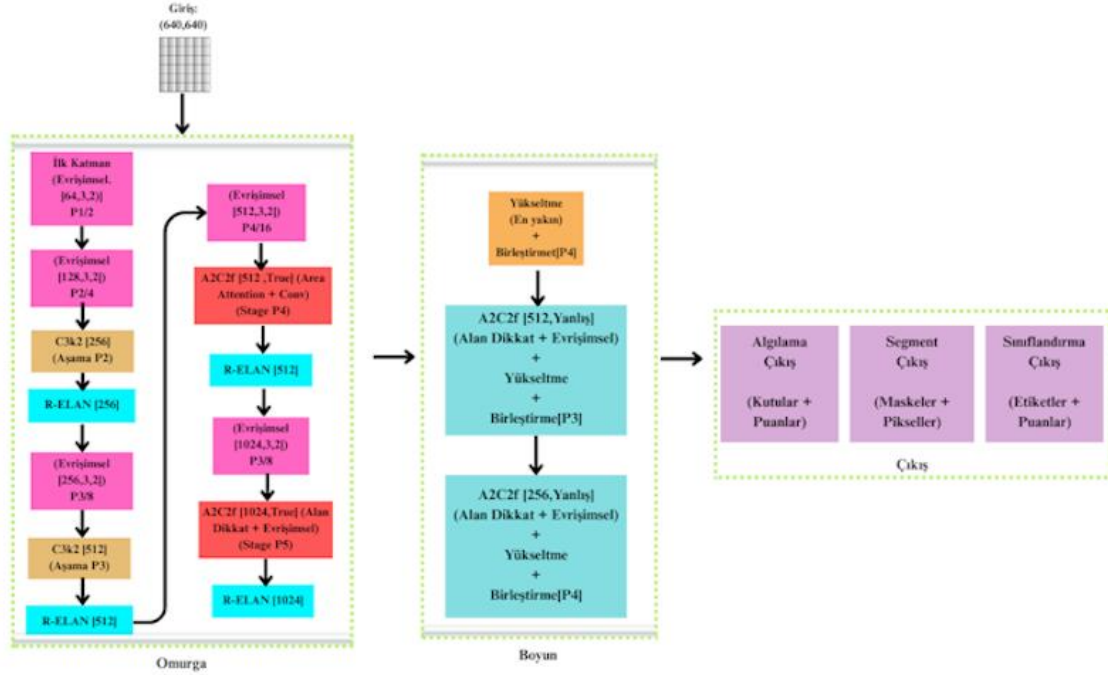
Çıkış başlığı, tespit yapılacak son katmandır. Bu bölümde üç farklı ölçekte tespit yapılır. Detect (80×80, 40×40, 20×20), her çözünürlük düzeyinde, sınırlayıcı kutuların koordinatları, nesne olasılığı ve sınıf tahminleri yapılır. Bu çok ölçekli yaklaşım, farklı boyuttaki nesnelerin tespitinde performansı artırır.

Bu yapı, özellikle kentsel ortamlarda yaya tespiti gibi karmaşık görevlerde yüksek doğrulukla çalışacak şekilde optimize edilmiştir. C2PSA ve C2PF gibi gelişmiş blokların entegrasyonu, YOLOv11'i klasik versiyonlara göre daha hassas hale getirmektedir.

4.4.3 YOLOv12

Bu çalışmada kullanılan YOLOv12 mimarisinin genel yapısı, omurga, boyun ve çıkış başlığı bileşenlerinin birbirleriyle olan etkileşimlerini bütüncül bir şekilde ortaya koymaktadır. Modelde yer alan katmanların veri akışı, çok ölçekli özellik çıkarımı, bağlamsal bilgi entegrasyonu ve tahminleme aşamalarının birbirleriyle uyumlu olarak nasıl çalıştığını göstermektedir. YOLOv12'nin mimarisine ilişkin ayrıntılı şema Şekil

4.5’te sunulmuş olup, bu yapı üzerinden modelin hesaplama verimliliğini artıran bloklar, geliştirilen dikkat mekanizmaları ve çapa gerektirmeyen tahminleme süreci görsel olarak açıklanmaktadır.



Şekil 4.5 YOLOv12 Mimarisinin Genel Yapısı

YOLOv12 mimarisi, önceki YOLO sürümlerine kıyasla daha optimize edilmiş bir yapı sunmakta ve özellikle hesaplama verimliliği ile yüksek doğruluk oranlarını bir arada sağlamaktadır. YOLOv12’nin geliştirilmesindeki temel amaç hem gömülü sistemlerde hem de yüksek donanım kapasiteli cihazlarda ölçeklenebilir bir performans elde etmektir (Sapkota ve Karkee 2025). Omurga kısmında, çok ölçekli özellik çıkarımı için optimize edilmiş konvolüsyon katmanları ve ileri seviye bloklar yer almaktadır. Bu aşamada kullanılan Efficient-C2f blokları, parametre sayısını azaltarak modelin hafif ve hızlı çalışmasına imkân tanır. Ayrıca, Dinamik Konvolüsyon katmanları sayesinde, giriş verisine göre uyarlanabilen filtreler kullanılmakta ve bu durum modelin genelleme kapasitesini artırmaktadır.

Boyun bölümünde, YOLOv11’de kullanılan PANet benzeri yapıların daha gelişmiş bir versiyonu entegre edilmiştir. YOLOv12’nin dikkat çeken özelliği, Küresel Bağlam

Modülü eklenmesidir. Bu yapı, farklı çözünürlüklerdeki öznitelik haritalarını birleştirirken sadece yerel bağlamı değil, aynı zamanda küresel bağlam bilgisini de hesaba katar. Böylece özellikle karmaşık sahnelerde nesne tespit performansı yükselmektedir.

Çıkış başlığı kısmında ise, klasik üç ölçekli tespit katmanları (80×80, 40×40, 20×20) korunmuş, ancak tahminleme sürecinde Çapa içermeyen mekanizma benimsenmiştir. Bu yaklaşımda nesne merkezleri üzerinden tahmin yapılmakta, bu sayede hem küçük nesne tespitinde doğruluk artmakta hem de eğitim süresi kısalmaktadır.

YOLOv12'nin bu özellikleri, özellikle gerçek zamanlı yaya tespiti ve akıllı trafik yönetimi gibi uygulamalarda, hız ve doğruluğun dengeli şekilde elde edilmesini sağlamaktadır. Böylece YOLOv12 hem mobil cihazlarda hem de yüksek kapasiteli sunucularda kullanılabilir hale gelmiştir.

4.4.4 YOLOv8, YOLOv11 ve YOLOv12'nin Karşılaştırılması

YOLO ailesinin gelişim sürecinde her yeni sürüm, nesne tespitinde hız, doğruluk ve verimlilik açısından iyileştirmeler sunmuştur. Aşağıda YOLOv8, YOLOv11 ve YOLOv12'nin temel özellikleri karşılaştırılmıştır.

İlk olarak mimari yapıları karşılaştırıldığında YOLOv8, omurga kısmında C2f blokları kullanılır. Çapa içermeyen tespit yaklaşımı ile nesne merkezleri doğrudan tahmin edilir. PANet tabanlı boyun yapısı ile çok ölçekli öznitelik birleşimi sağlanır. YOLOv11 geleneksel YOLO yapısını geliştirerek C2PF ve C2SA gibi dikkat mekanizmaları eklenmiştir. Bu sayede daha hassas özellik çıkarımı yapılır. Boyun kısmında C2PSA blokları ile hem kanal hem mekânsal dikkat güçlendirilmiştir. YOLOv12 ise Efficient-C2f ve Dinamik Konvolüsyon blokları ile hesaplama maliyeti düşürülmüş, Küresel Bağlam Modülü ile küresel bağlam bilgisi modele eklenmiştir. Çıkış başlığı kısmında çapa içermeyen yapı korunarak daha hızlı ve esnek tespit sağlanmıştır.

Dikkat mekanizmaları karşılaştırıldığında YOLOv8, temel C2f blokları ile bilgi kaybı azaltılmıştır ancak dikkat modülleri sınırlıdır. YOLOv11, kanal ve mekânsal dikkat

mekanizmaları sayesinde modelin önemli bölgelere odaklanması sağlanmıştır. YOLOv12 ise global Bağlam Modülü entegrasyonu ile hem lokal hem de küresel dikkat bilgisi işlenerek karmaşık sahnelerde daha güçlü performans elde edilmiştir.

Boyun Yapısı temel özelliği karşılaştırıldığında YOLOv8, PANet + FPN kombinasyonu ile çok ölçekli öznitelikler birleştirilmiştir. YOLOv11, Yükseltme + Birleştirme + C3x2 blokları ile farklı çözünürlüklü özellikler birleştirilmiştir. YOLOv12 ise geliştirilmiş FPN ve Küresel Bağlam ile çok katmanlı bilgi akışı güçlendirilmiştir.

Çıkış başlığı karşılaştırıldığında YOLOv8; Çapa içermeyen mekanizma ile küçük nesne tespitinde avantaj sağlamaktadır. YOLOv11; klasik çapa tabanlı yaklaşım korunmuş, üç farklı ölçek (80×80, 40×40, 20×20) üzerinden tahmin yapılmaktadır. YOLOv12 ise Çapa içermeyen tahminleme korunmuş, daha hızlı ve esnek bir yapı sağlanmıştır.

Son olarak performansları karşılaştırıldığında YOLOv8; hızlı, hafif ve çok yönlüdür. Segmentasyon ve sınıflandırma gibi ek görevlerde de kullanılabilir. YOLOv11; daha hassas bir modeldir. Özellikle yaya tespiti gibi kritik görevlerde, C2SA ve C2PF bloklarının etkisiyle yüksek doğruluk sağlar. YOLOv12 ise hem hız hem doğruluk açısından dengeli bir yapı sunar. Daha az parametreyle daha yüksek performans elde edilebilmesi, gömülü sistemlerde kullanımını kolaylaştırır.

Aşağıdaki çizelge 4.2 de YOLOv8, YOLOv11, YOLOv12 sürümlerinin karşılaştırması detaylı bir şekilde gösterilmiştir.

Çizelge 4.2 YOLOv8, YOLOv11, YOLOv12 Sürümlerinin Detaylı Karşılaştırması

Özellik / Model	YOLOv8	YOLOv11	YOLOv12
Omurga	C2f blokları, hafif tasarım	Conv + C3x2 + C2PF + C2SA	Efficient-C2f, Dinamik Konv.

Özellik / Model	YOLOv8	YOLOv11	YOLOv12
Dikkat Mekanizması	Sınırlı (C2f ile bilgi kaybı azaltma)	C2SA ve C2PSA (kanal + mekânsal dikkat)	Küresel Bağlam Modülü (küresel + lokal dikkat)
Boyun Yapısı	PANet + FPN, Yükseltme, Birleştirme	C3x2, Yükseltme, Birleştirme, C2PSA	Geliştirilmiş FPN, Küresel Bağlam, Birleştirme
Çıkış başlığı (Çıkış Katmanı)	Çapa içermeyen, doğrudan merkez tabanlı tespit	Anchor tabanlı, 3 ölçekli (80×80, 40×40, 20×20)	Çapa içermeyen, hızlı ve esnek
Çok Ölçekli Tespit	Desteklenir (küçük-büyük nesneler)	Desteklenir (3 farklı ölçek)	Desteklenir (çapa içermeyen + global bağlam)
Ek Özellikler	Segmentasyon ve sınıflandırma görevlerinde kullanılabilir	Yaya tespiti gibi kritik görevlerde yüksek doğruluk	Daha az parametreyle daha yüksek doğruluk, gömülü sistem uyumu
Avantajları	Hafif, hızlı, çok yönlü	Yüksek doğruluk, güçlü dikkat mekanizmaları	Hız-doğruluk dengesi, global bağlam farkındalığı
Dezavantajları	Dikkat mekanizmaları sınırlı	Parametre sayısı ve hesaplama yükü daha fazla	Yeni yapı nedeniyle daha karmaşık optimizasyon gerektirir

4.5 Değerlendirme Metrikleri

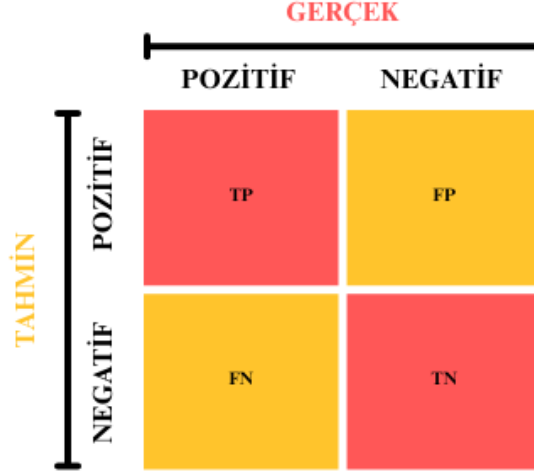
Nesne tespiti yapılırken F1 skoru, duyarlılık ve kesinlik gibi performans ölçütleri oldukça önemli ve yaygın olarak kullanılır. Bu terimler, modelin ne kadar doğru ve etkili çalıştığını değerlendirmenin farklı yollarıdır. Bu metrikler, yaya tespiti gibi uygulamalarda modelin ne kadar doğru ve güvenilir olduğunu anlamak için kritik öneme sahiptir. Karmaşıklık matrisi ise bu metriklerin temelini oluşturan TP, FP, TN ve FN bileşenlerini ayrıntılı biçimde göstererek, modelin hangi durumlarda doğru algılama yaptığı ve hangi hata türlerine eğilimli olduğu hakkında doğrudan bilgi sağlar. Matris üzerinden görülen hata dağılımı, kesinlik değerini etkileyen yanlış pozitifleri ve duyarlılık değerini belirleyen kaçırılan örnekleri açıkça ortaya koyar. Böylece karmaşıklık matrisi, performans metriklerinin sayısal arka planını somutlaştırarak modelin genel davranışının daha kapsamlı şekilde değerlendirilmesine imkân tanır.

4.5.1 Karmaşıklık Matrisi

Karmaşıklık matrisi, modelin hangi örnekleri doğru veya yanlış sınıflandırdığını ayrıntılı biçimde gösteren temel bir performans analiz aracıdır. Bu matris, True Positive (TP), False Positive (FP), True Negative (TN) ve False Negative (FN) değerleri üzerinden modelin davranışını inceleme imkânı sağlar. TP değeri modelin hedef sınıfa ait nesneleri doğru şekilde tespit ettiği durumları, FN değeri ise gerçek bir nesnenin model tarafından algılanamadığı durumları temsil eder. Benzer şekilde FP değeri, modelin var olmayan bir nesneyi yanlışlıkla tespit etmesi; TN ise arka plan veya hedef dışı bölgelerin doğru şekilde negatif olarak sınıflandırılması anlamına gelir.

Bu yapı, özellikle tek sınıflı nesne tespitinde modelin hata türlerini açıkça ortaya koyar. FN değerleri modelin kaçırdığı nesneleri, FP değerleri ise ürettiği yanlış alarmları gösterdiğinden, bu iki hata tipi sistem güvenilirliği açısından kritik öneme sahiptir. Yüksek TP ve TN oranları modelin doğru sınıflandırma kapasitesinin güçlü olduğunu, düşük FP ve FN oranları ise yanlış algılama ve kaçırma eğiliminin sınırlı olduğunu gösterir. Böylece karmaşıklık matrisi, modelin hangi hata türlerine eğilimli olduğunu belirleyerek Kesinlik, Duyarlılık, F1-skor ve mAP gibi metriklerin arka planındaki sayısal

davranışı anlamaya olanak sağlar. Bu çalışmada modelin sınıflandırma performansını değerlendirmek üzere kullanılan hata analiz yapısı, Şekil 4.6’te sunulan karmaşıklık matrisi üzerinden detaylandırılmaktadır.



Şekil 4.6 Karmaşıklık Matrisi

4.5.2 Kesinlik

Kesinlik, modelin doğru algıladığı pozitif örneklerin, modelin algıladığı tüm pozitif örneklerin oranını ifade eder. Başka bir deyişle, modelin yaya olarak sınıflandırdığı nesnelerin kaç tanesinin gerçekten yaya olduğunu gösterir. Kesinlik, modelin doğru algıladığı nesnelerin sayısını gösterir. Yüksek kesinlik, modelin yanlış alarm oranının düşük olduğunu gösterir. Kesinlik değerine ait denklem (1) aşağıda yer almaktadır.

$$Kesinlik = \frac{TP}{TP+FP} \quad (1)$$

- TP (Gerçek Pozitifler): Gerçekten yaya olan ve model tarafından doğru tespit edilen örnekler.
- FP (Yanlış Pozitifler): Gerçekten yaya olmayan, ancak model tarafından yanlışlıkla yaya olarak tespit edilen örnekler.

4.5.3 Duyarlılık

Duyarlılık, modelin doğru tespit ettiği pozitif örneklerin, gerçek pozitif örneklerin toplamına oranıdır. Yani, modelin gerçek yayaların ne kadarını doğru tespit ettiğini gösterir. Duyarlılık, modelin gerçek yayaları tespit etme yeteneğini ölçer. Yüksek Duyarlılık, modelin yayaları kaçırma oranının düşük olduğunu belirtir. Aşağıdaki duyarlılığa ait denklem (2) yer almaktadır.

$$Duyarlılık = \frac{TP}{TP+FN} \quad (2)$$

- TP (Gerçek Pozitifler): Gerçekten yaya olan ve model tarafından doğru tespit edilen örnekler.
- FN (Yanlış Negatifler): Gerçekten yaya olan ancak model tarafından tespit edilemeyen örnekler.

4.5.4 Ortalama Kesinlik (mAP)

Ortalama Kesinlik (AP), bir dizi tahmin için kesinlik-geri çağırma eğrisinin altındaki alan olarak hesaplanır. Sınıflandırma eşiğini değiştirerek, kesinlik ve geri çağırma arasındaki denge bir eğri olarak görselleştirilir. Bu eğrinin altındaki alan bir sınıf için AP'yi verir ve tüm sınıfların ortalaması ortalama kesinlik (mAP) olarak adlandırılır. Nesne algılama görevlerinde, AP ve mAP hesaplamaları, tahmin edilen sınırlayıcı kutu ile gerçek sınırlayıcı kutu arasındaki çakışmayı ölçen Birleşim Üzerine Kesişim (IoU) için bir eşik tanımlanmasını gerektirir. Bu çalışmada, IoU eşiği, yaya algılama karşılaştırmalarında yaygın olarak kullanılan 0,5 olarak ayarlandı. Bir tahmin, yalnızca IoU bu eşiği aştığında Gerçek Pozitif olarak kabul edilir; aksi takdirde, Yanlış Pozitif olarak sayılır. Bu IoU eşiğine göre hesaplanan mAP metriği, Denklem 3'te sunulmuştur.

$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i \quad (3)$$

4.5.5 F1 Skor

Başarı Oranı, kesinlik ve duyarlılığın harmonik ortalamasıdır ve bu değerlerin arasında bir denge sağlar. F1 skoru, modelin genel doğruluğunu ölçmek için kullanılır ve yüksek bir F1 skoru hem yüksek kesinlik hem de yüksek duyarlılık olduğunu gösterir. Bu, modelin hem doğru tespit yapma oranının yüksek olduğunu hem de kaçırma oranının düşük olduğunu belirtir. Bu F1 skor hesaplaması, Denklem 4'te sunulmuştur.

Formül:

$$F1\ Skor = \frac{2 \times Kesinlik \times Duyarlilik}{Kesinlik + Duyarlilik} \quad (4)$$

5. BULGULAR

Bu bölümde, geliştirilen derin öğrenme tabanlı yaya tespit sistemi kapsamında gerçekleştirilen model eğitimleri ve performans değerlendirmelerine ilişkin bulgular sunulmaktadır. Çalışmada, farklı YOLO sürümleri (YOLOv8, YOLOv11 ve YOLOv12), çeşitli model varyantları (N, S, M, L), optimizasyon algoritmaları (Adam, AdamW, SGD) ile Parti boyutu ve epok parametrelerinin performans üzerindeki etkileri ayrıntılı biçimde incelenmiştir. Ayrıca, deneysel sonuçlar, optimizasyon algoritması ve eğitim parametrelerinin model başarımı üzerindeki etkilerini somut biçimde göstermekte, böylece geliştirilen sistemin nihai konfigürasyonuna ilişkin bilimsel temeli oluşturmaktadır.

5.1 Model Performans Değerlendirmeleri

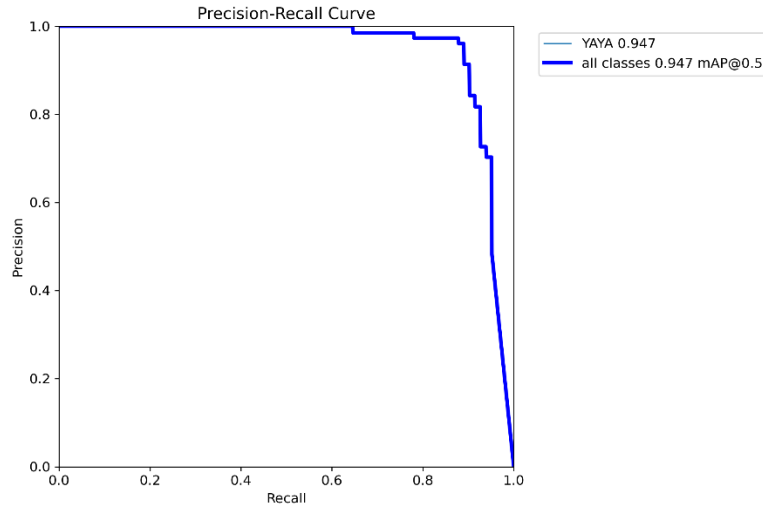
Tüm model varyantlarında varsayılan olarak SiLU (Sigmoid Doğrusal Ünite) aktivasyon fonksiyonu kullanılmıştır. Eğitimlerde kullanılan optimizasyon algoritmaları arasında AdamW, SGD ve Adam yer almakta olup weight decay parametresi 0.0005 olarak ayarlanmıştır. Her bir model için elde edilen Kesinlik, Duyarlılık, mAP ve F1-Skor değerleri karşılaştırılmış, modellerin doğruluk, hız ve genelleme kabiliyeti açısından performans farklılıkları değerlendirilmiştir.

5.1.1 YOLOv8 Model Performans Değerlendirmesi

YOLOv8 modellerinde yapılan farklı parametre denemeleri sonucunda en yüksek F1-Skor değeri 0.944 olarak bulunmuştur. Bu sonuç, Parti boyutu 8, Epok 50 ve AdamW optimizatörü kullanılarak elde edilmiştir. YOLOv8 üzerinde yapılan testlerde özellikle Parti boyutu'nun küçük olması modelin daha iyi genel performans göstermesini sağlamıştır. Bu, küçük batch'lerde modelin her güncellemede daha hassas değişiklikler yapabilmesi sayesinde olmuştur. Optimizasyon tercihlerinde AdamW öne çıkmış, SGD ve Adam optimizatörleri de kullanılmış ancak AdamW ile alınan sonuçlar genellikle daha yüksektir. Epok değerinin artırılması modelin F1-Skor değerine çok büyük katkılar sağlamamış, yani erken durdurma yöntemleriyle eğitim süresi kısaltılabilir. Ayrıca,

Adam optimizatörü ile elde edilen sonuçlar da kötü olmamakla birlikte, AdamW'nin istikrarlı ve yüksek başarı sağladığı görülmüştür. Bu sonuçlar, YOLOv8 modelinin hızlı ve doğru bir şekilde küçük veri grupları üzerinde etkili bir şekilde öğrenebildiğini ve optimizasyon süreçlerine karşı duyarlı olduğunu göstermektedir. Özellikle gerçek zamanlı uygulamalarda ve sınırlı donanım kaynaklarında, küçük Parti boyutu ile AdamW optimizasyonu tercih edilerek daha verimli sonuçlar alınabileceği anlaşılmaktadır.

- YOLOv8-S modeli 8 Parti boyutu, 50 epok ve AdamW optimizasyonu ile en yüksek mAP (%94,7) ve F1-skor (%94,4) değerlerine ulaşmıştır.
- Bu model, özellikle hem yüksek kesinlik hem de duyarlılık gerektiren yaya tespiti gibi güvenlik odaklı uygulamalarda oldukça başarılıdır.
- 100 epok ile yapılan eğitimde modelin Duyarlılık oranı artarken, Kesinlik'da küçük bir düşüş gözlemlenmiştir. Bu da veri seti ve optimizasyon parametrelerine bağlı olarak uzun süreli eğitimin her zaman performansı artırmayabileceğini göstermektedir.



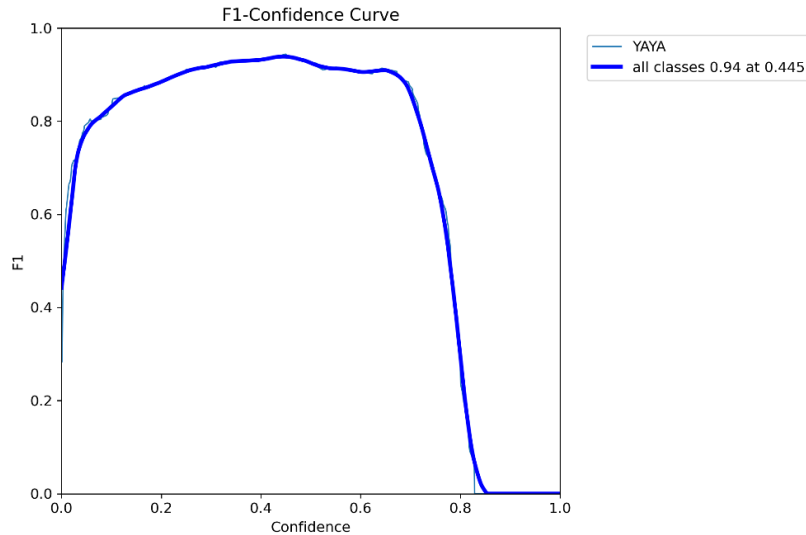
Resim 5.1 YOLOv8-S En yüksek performansı gösteren modelin Kesinlik–Duyarlılık eğrisi

Resim 5.1 de verilen Kesinlik–Duyarlılık eğrisi, modelin farklı eşik değerlerinde doğruluk ve duyarlılık performansını birlikte ortaya koymaktadır. Grafik incelendiğinde, modelin geniş bir Duyarlılık aralığında oldukça yüksek Kesinlik değerlerini koruyabildiği görülmektedir. Özellikle Duyarlılık değerinin 0.80 ile 1.00 aralığı arasında seyrettiği

bölgede Kesinlik'in 1.0 seviyesine yakın kalması, modelin yüksek duyarlılık talep edilen durumlarda bile yanlış pozitif üretme oranının düşük olduğunu göstermektedir.

Kesinlik-Duyarlılık eğrisi altında kalan alan, sınıfa ait Average Kesinlik (AP) değerini temsil eder ve bu çalışmada $AP=0.947$, diğer bir ifadeyle %94.7 olarak hesaplanmıştır. Grafik üzerinde yer alan "all classes 0.947 mAP@0.5" ifadesi, tek sınıflı (yaya) bir problemde modelin $IoU=0.5$ eşik değerinde oldukça yüksek bir ortalama doğruluk sergilediğini doğrulamaktadır. Eğrinin keskin bir düşüşünün yalnızca Duyarlılık'un 1.0 değerine yaklaşıldığı uç noktada gerçekleşmesi, modelin karar eşiği azaltıldığında dahi uzun süre tutarlı bir Kesinlik seviyesi koruduğunu göstermektedir.

Bu davranış, özellikle gerçek zamanlı yaya tespit sistemlerinde kritik öneme sahiptir; zira düşük Duyarlılık değerlerinde yayaların algılanamaması güvenlik problemlerine yol açarken, düşük Kesinlik değeri gereksiz alarm üretimine sebep olur. Kesinlik-Duyarlık eğrisinin büyük bölümünde yüksek Kesinlik ve yüksek Duyarlılık değerlerinin birlikte korunması, modelin her iki hata türünü de minimize ettiğini ve pratik kullanımda güvenilir bir performans sunduğunu göstermektedir.



Resim 5.2 YOLOv8-S en yüksek performans gösteren modelin F1–Güven eğrisi

Resim 5.2 de verilen F1–Güven eğrisi, modelin farklı güven eşiklerinde elde ettiği F1 skorunu göstermektedir. F1 skoru, Kesinlik ve Duyarlılık'ın harmonik ortalaması olduğu

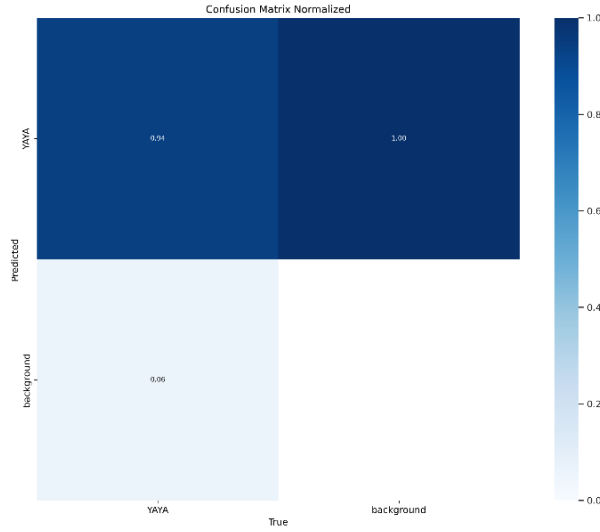
için, bu eğri modelin hem yanlış pozitifleri hem de yanlış negatifleri aynı anda minimize ettiği optimum karar eşiğini belirlemek açısından kritik öneme sahiptir.

Eğrinin genel yapısı incelendiğinde, düşük güven eşiklerinde (0.0–0.1 aralığı) F1 skorunun hızlı bir şekilde yükseldiği görülmektedir. Bu durum, düşük eşiklerde her ne kadar model daha fazla nesne tespit etse de başlangıçta yanlış pozitif oranının yüksek olduğunu göstermektedir. Güven değeri arttıkça modelin tahminleri daha seçici hâle gelir; bu da Kesinlik’ı artırarak F1 skorunda belirgin bir iyileşmeye yol açar.

Grafikte F1 skorunun maksimum yaklaşık 0.94 değerine, güven = 0.445 noktasında ulaştığı görülmektedir. Bu nokta, modelin Kesinlik ve Duyarlılık dengesini en iyi sağladığı eşiği temsil eder ve pratik uygulamalarda kullanılabilecek optimal karar eşiği olarak değerlendirilebilir. Güven değeri 0.5 civarından sonra F1 skorunun kademeli bir şekilde düşmeye başlaması, yüksek eşiklerde Duyarlılık’ın dramatik biçimde azalmasından kaynaklanmaktadır. Model güven değerini fazla yükselttiğinde daha az yaya tespiti yapılmakta, bu da yanlış negatiflerin artmasına neden olarak F1 skorunu düşürmektedir.

Güven değeri 0.8 seviyesine yaklaştığında F1 skorunun keskin bir düşüş göstermesi, bu eşikten sonra modelin aşırı muhafazakâr hâle geldiğini ve yalnızca en yüksek olasılıklı tespitleri kabul ederek pek çok gerçek yaya örneğini kaçırdığını göstermektedir. Bu eğilim, gerçek zamanlı yaya tespit uygulamalarında yüksek güven eşiklerinin tercih edilmemesi gerektiğini ortaya koymaktadır.

Sonuç olarak, F1–Güven eğrisi modelin en dengeli performansı 0.445 güven eşiğinde verdiğini göstermekte ve bu eşik, sistemin hem yanlış uyarıları hem de kaçırılan tespitleri minimize ederek en verimli şekilde çalışmasını sağlamaktadır. Bu değer, gerçek zamanlı yaya güvenliği sisteminin pratik uygulamalarda kullanılabilecek optimal yönetim parametresi olarak önerilebilir.



Resim 5.3 YOLOv8-S en yüksek performansı gösteren modelin normalize edilmiş karmaşıklık matrisi

Resim 5.3 de modelin normalize edilmiş karmaşıklık matris sonuçlarını göstermektedir. Model, gerçek yaya örneklerinin %94'ünü doğru biçimde tespit etmiş, yalnızca %6'sını yanlışlıkla arkaplan olarak sınıflandırmıştır. Bu sonuç, modelin yaya tespit performansının yüksek olduğunu ve duyarlılık değerinin güçlü şekilde korunduğunu göstermektedir.

Arkaplan sınıfı, YOLO mimarilerinde gerçek bir sınıf olarak modellenmediğinden, normalize matrisin ilgili satırında arkaplan'ın %100 oranında YAYA sınıfına karşılık gelmesi beklenen bir sonuçtur ve yanlış pozitif anlamına gelmemektedir. Bu durum, veri setinde yalnızca tek hedef sınıfın (yaya) bulunmasına bağlı olarak normalizasyonun otomatik şekilde bu sonuca ulaşmasından kaynaklanmaktadır. Dolayısıyla bu satır, modelin performansını olumsuz yansıtan bir hata değil, tek sınıflı tespit problemlerine özgü bir normalizasyon etkisidir.

Genel olarak karmaşıklık matris, modelin yüksek doğrulukla yaya tespit ettiğini, kaçırma oranının oldukça düşük olduğunu ve gerçek trafik senaryolarında kullanılabilir bir performans sunduğunu ortaya koymaktadır.

5.1.2 YOLOv11 Model Performans Değerlendirmesi

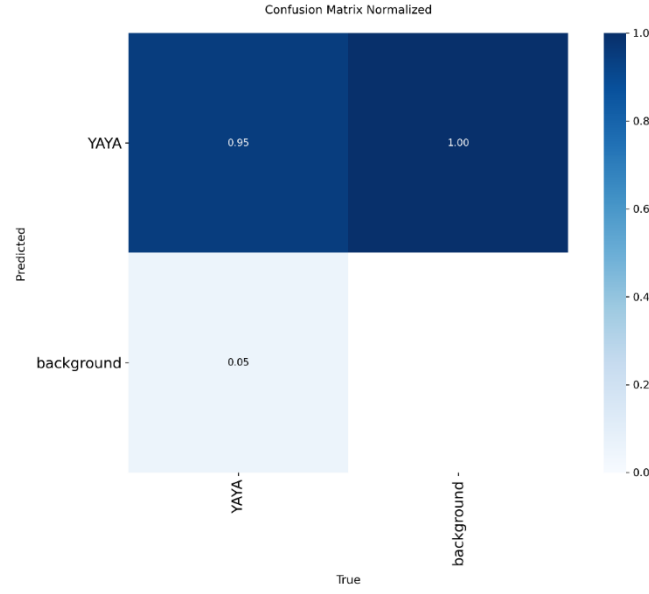
YOLOv11 modelleri üzerinde yapılan testlerde en yüksek F1-Skor değeri 0.9494 olarak kaydedilmiştir. YOLOv11'in farklı varyantları arasında (S, N, M, L) S modeli, diğerlerine kıyasla daha başarılı sonuçlar üretmiştir. Bu başarılı sonuç, YOLOv11-S varyantında Parti boyutu 16, Epok 100 ve AdamW optimizatörü kullanılarak elde edilmiştir. YOLOv11-N modeli, özellikle daha küçük mimari yapısıyla daha az kaynak tüketirken, 8 batch ve 50 epok konfigürasyonunda %92.2 F1-Skor ile dikkat çekmiştir. Bu, gerçek zamanlı sistemlerde yüksek doğrulukla çalışabilme potansiyelini ortaya koymaktadır. YOLOv11-M modeli ise, genel olarak diğer modellerin gerisinde kalmış, yalnızca %93.8 mAP değeri ile orta düzey bir başarı göstermiştir.

Ayrıca yapılan deneylerde Epok sayısının artırılması, özellikle 100 Epok civarında, modelin doğruluk ve genel F1-Skor değerlerini yükseltmiştir. Ancak Epok 150 gibi daha yüksek değerlerde bazı modellerde F1-Skor değerinde küçük bir düşüş gözlemlenmiştir, bu da aşırı öğrenmenin başlamış olabileceğine işaret etmektedir. Optimizatör tercihlerinde AdamW'nin yine ön plana çıktığı ve SGD ile karşılaştırıldığında daha dengeli ve yüksek doğruluk değerleri sunduğu görülmüştür. Adam optimizasyonu, diğerlerine göre daha düşük performans göstermiştir. Bu durum, Adam algoritmasının bu veri seti üzerinde aşırı öğrenmeye yatkın olabileceğini düşündürmektedir. SGD kullanıldığında da bazı modellerde iyi sonuçlar alınsa da AdamW ile alınan skorlar genellikle daha yüksektir. Bunun yanında, YOLOv11 için Parti boyutu hem 16 hem de 8 kullanılmış, ancak genel olarak 16 kullanıldığında daha stabil ve yüksek sonuçlar elde edilmiştir.

YOLOv11'in özellikle yüksek doğruluk, hızlı çıkarım ve büyük veri setleri üzerinde etkin performans sunabilmesi, onu özellikle şehir içi yaya güvenliği, akıllı trafik yönetimi ve otonom sistemlerde kullanım için güçlü bir aday haline getirmiştir.

- YOLOv11-S modeli, 100 epok ve AdamW optimizasyon ile %96 Kesinlik ve %95,7 mAP değerleri ile dikkat çekti.

- SGD optimizasyon algoritması genel olarak daha yüksek mAP değerleri sağladı; örneğin, 50 epok'ta %97,2 mAP değeri elde edildi.
- AdamW ile 150 epok eğitiminde Kesinlik yüksek tutulurken, duyarlılıkta bir düşüş gözlemlendi (%88,6).



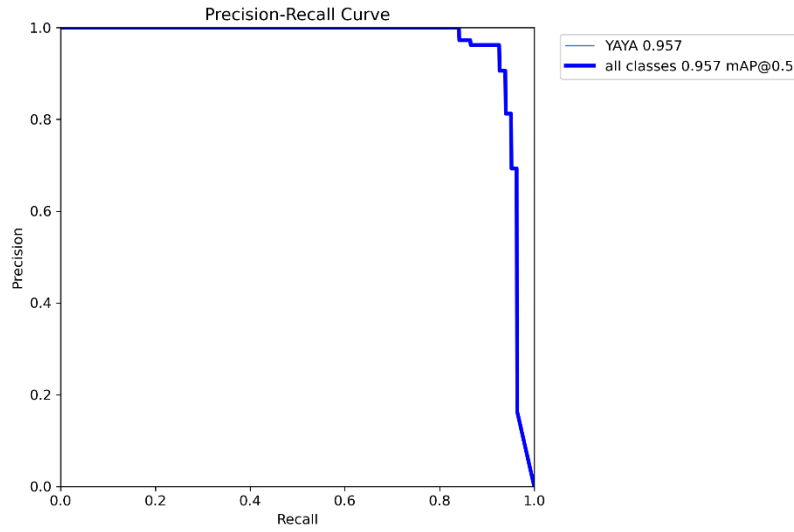
Resim 5.4 YOLOv11-S en yüksek performansı gösteren modelin normalize edilmiş karmaşıklık matrisi

Resim 5.4 de YOLOv11 modelinin normalize edilmiş karmaşıklık matris sonuçlarını göstermektedir. Model, gerçek yaya örneklerinin %94'ünü doğru biçimde tespit etmiş, yalnızca %6'sını yanlışlıkla arkaplan olarak sınıflandırmıştır. Bu sonuç, modelin yaya tespit performansının yüksek olduğunu ve duyarlılık değerinin güçlü şekilde korunduğunu göstermektedir.

Arkaplan sınıfı, YOLO mimarilerinde gerçek bir sınıf olarak modellenmediğinden, normalize matrisin ilgili satırında arkaplan'ın %100 oranında YAYA sınıfına karşılık gelmesi beklenen bir sonuçtur ve yanlış pozitif anlamına gelmemektedir. Bu durum, veri setinde yalnızca yaya sınıfının bulunmasına bağlı olarak normalizasyonun otomatik şekilde bu sonuca ulaşmasından kaynaklanmaktadır. Dolayısıyla bu satır, modelin

performansını olumsuz yansıtan bir hata değil, tek sınıflı tespit problemlerine özgü bir normalizasyon etkisidir.

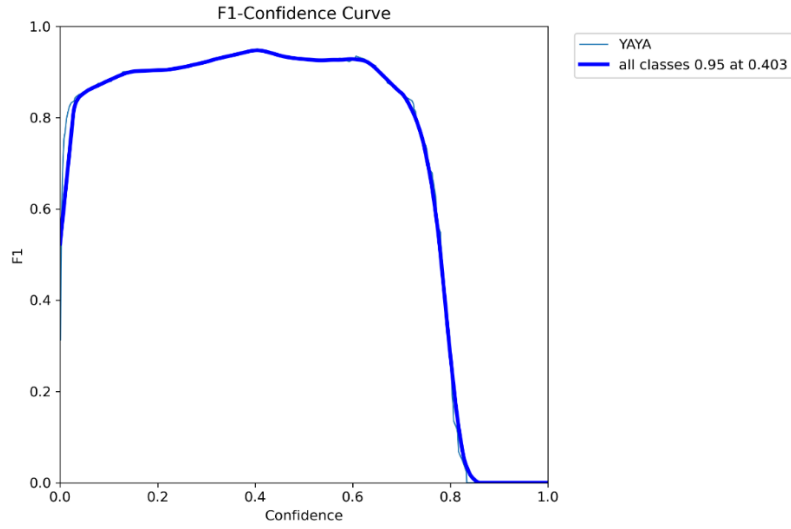
Genel olarak karmaşıklık matris, modelin yüksek doğrulukla yaya tespit ettiğini, kaçırma oranının oldukça düşük olduğunu ve gerçek trafik senaryolarında kullanılabilir bir performans sunduğunu ortaya koymaktadır.



Resim 5.5 YOLOv11-S en yüksek performansı gösteren modelin Kesinlik–Duyarlılık eğrisi

Resim 5.5 de eğitilen YOLOv11 modelinin yaya sınıfı için elde ettiği Kesinlik–Duyarlılık eğrisini göstermektedir. Kesinlik-Duyarlılık eğrisi; modelin farklı güven eşiği değerlerinde kesinlik ve duyarlılık arasındaki dengeyi açığa çıkarmaktadır. Model için mAP@0.5 değeri 0.957 olarak hesaplanmıştır; bu değer, modelin $\text{IoU} \geq 0.5$ koşulu altında yüksek ortalama doğruluk sağladığını göstermektedir.

Eğri incelendiğinde, duyarlılık geniş bir aralıkta (yaklaşık 0.8–0.98) yüksek kesinlik değerlerinde seyretmekte olup, bu durum modelin hem yüksek duyarlılık hem de düşük yanlış pozitif oranı sunduğunu göstermektedir. Ancak duyarlılık değerinin en uç noktasına (duyarlılık ≈ 1.0) yaklaşıldığında kesinlik’de belirgin bir düşüş gözlenmektedir; bu davranış tipik olarak çok düşük güven eşiği seçildiğinde modelin tüm olası adayları tespit ederken yanlış pozitif sayısını artırmasından kaynaklanır. Sonuç olarak, model genel performansı açısından $\text{mAP}@0.5 = 0.957$ güçlü bir başarıyı işaret etmektedir.



Resim 5.6 YOLOv11-S en yüksek performansı gösteren modelinin F1-Güven Eğrisi

Resim 5.6 da YOLOv11 modelinin yaya sınıfı için hesaplanan F1-Güven eğrisini göstermektedir. F1 skoru; kesinlik ve duyarlılık arasındaki dengeyi aynı anda temsil ettiği için, modelin farklı güven eşiği seviyelerindeki genel tespit performansını değerlendirmede kritik bir ölçüttür.

Grafik incelendiğinde F1 skorunun düşük eşik değerlerinde hızlı bir artış gösterdiği, yaklaşık 0.30 – 0.60 aralığında ise yüksek bir plato oluşturarak stabil bir maksimum performans bölgesi sunduğu görülmektedir. Modellerde genellikle düşük eşik seviyelerinde yanlış pozitif oranı yüksek olduğu için F1 düşük başlar; eşik artırıldıkça hem kesinlik iyileşir hem de duyarlılık makul seviyede kalır ve dolayısıyla F1 yükselir.

Bu çalışmada en yüksek F1 değeri 0.95, güven ≈ 0.403 seviyesinde elde edilmiştir. Bu nokta, kesinlik ve duyarlılık'ın en optimal birleşiminin sağlandığı ve modelin en dengeli performansı sunduğu eşik olarak değerlendirilebilir. Güven değerinin 0.40 civarında optimuma ulaşması, modelin yüksek güven isteyen tahminlerde başarıyla genelleştğini, ancak eşik 0.70 ve üzerine çıktığında duyarlılık'ın hızla düşmesi nedeniyle F1 skorunun keskin biçimde azaldığını göstermektedir.

Bu sonuç, gerçek zamanlı uygulamalarda 0.35–0.45 aralığındaki bir güven eşiğinin hem yüksek doğruluk hem de düşük kaçırma oranı sağlayacağını göstermektedir.

5.1.3 YOLOv12 Model Performans Değerlendirmesi

YOLOv12 modellerinde yapılan denemeler sonucunda en yüksek F1-Skor değeri 0.9420 olarak ölçülmüştür. Bu başarı, YOLOv12-N varyantı kullanılarak, Parti boyutu 16, Epok 50 ve AdamW optimizatörü tercih edilerek elde edilmiştir. YOLOv12-S modeli ise bu başarıya ulaşamamış, sadece %94.7 mAP ve %89.7 F1-Skor ile nispeten daha düşük kalmıştır. YOLOv12 modelleri üzerinde yapılan testlerde, AdamW optimizatörü, SGD ve Adam gibi diğer optimizasyon yöntemlerine göre daha iyi sonuçlar üretmiştir. Benzer şekilde, 75 epok kullanıldığında da mAP değeri korunmuş ancak F1-Skor bir miktar düşmüştür. Bu da fazla eğitim ile modelin Duyarlılık üzerinde güçlenirken Kesinlik’den ödün verebileceğini göstermektedir. Bu sonuçlar genel olarak en iyi sonuçların 50 Epok eğitim süresi sonunda görüldüğünü açıklamaktadır. Bu durum, YOLOv12’nin çok uzun eğitim döngülerine ihtiyaç duymadan yüksek performansa ulaşabildiğini göstermektedir. Ayrıca, Parti boyutu açısından 16 kullanımı modellerin daha stabil öğrenmesini sağlamış ve daha yüksek doğruluk skorları elde edilmiştir. YOLOv12-N varyantı, küçük ve orta ölçekli nesnelerin tespitinde oldukça başarılı bulunmuş, buna karşın daha büyük nesnelerin tespitinde YOLOv11-S gibi varyantlara göre biraz daha düşük performans göstermiştir. YOLOv12 genel olarak daha hafif, daha hızlı ve kaynak kullanımında daha verimli bir model olması ile ön plana çıkmaktadır. Özellikle kaynak verimliliği önemli olan projelerde YOLOv12 güçlü bir tercih olabilmektedir.

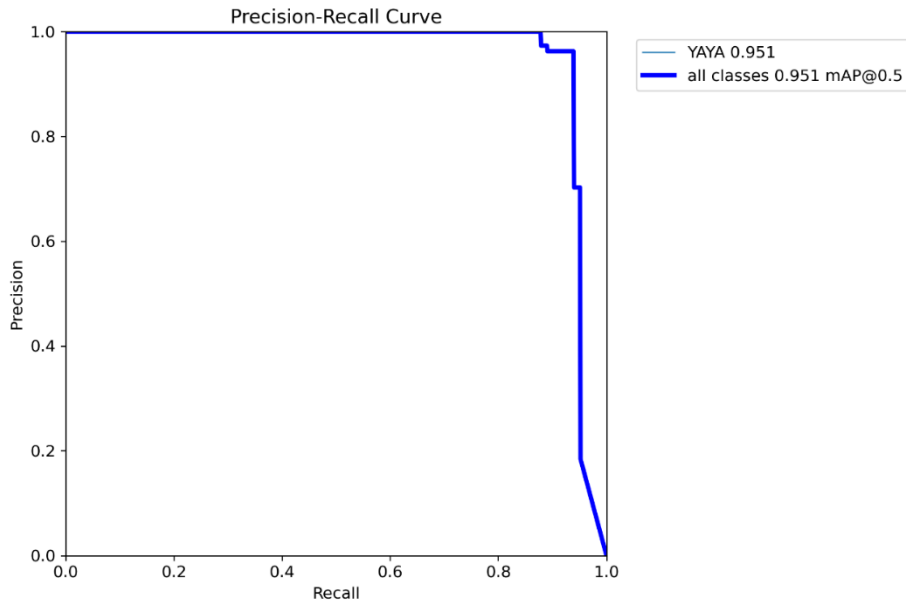
- YOLOv12-N modeli 50 epok AdamW ile %95.1 mAP ve %94.7 Duyarlılık ile başarılı sonuçlar üretmiştir.
- SGD ve Adam gibi diğer optimizasyon algoritmaları daha düşük performans göstermiştir.
- 75 epok'ta AdamW ile eğitimde %95.6 Duyarlılık elde edilmiş, bu da yayaların kaçırılmadan tespitinde önemli bir başarı olarak öne çıkmıştır.

AdamW optimizasyon algoritması, genel olarak SGD ve Adam’a göre daha istikrarlı sonuçlar vermiştir. SGD bazı modellerde yüksek mAP sağlasa da AdamW ile elde edilen

sonular hem Kesinlik hem Duyarlılık aısından daha dengeli olmuştur. Adam algoritması ise genellikle daha düşük F1-Skor üretmiştir.

Parti boyutu’nun 8 olması, bazı modellerde özellikle küçük mimarilerde daha yüksek performans sağlamıştır. Epok sayısı arttıka genellikle mAP deęerlerinde artış gözlemlenmiş, ancak bu artış her zaman F1-Skor’a olumlu yansımamıştır. Bu da dikkatli bir erken durdurma stratejisinin önemini ortaya koymaktadır.

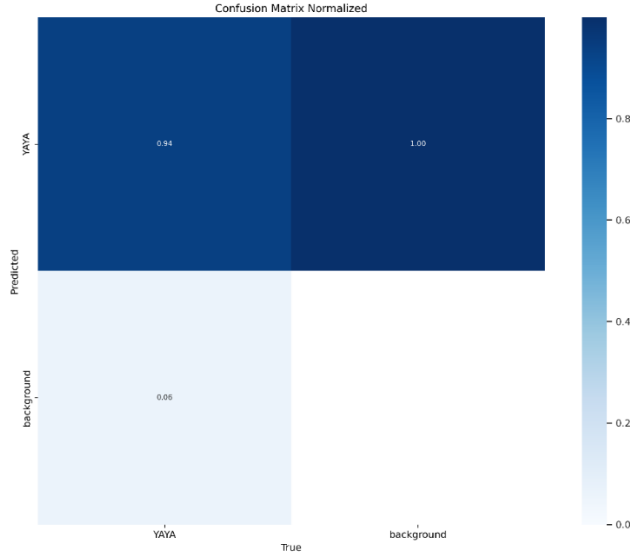
“S” modeller hafif olmalarına rağmen özellikle YOLOv8-S, güçlü bir performans göstermiştir. “M” ve “L” modeller daha az denenmiş olsa da daha fazla kaynak gerektirmektedir.



Resim 5.7 YOLOv12-N en yüksek performansı gösteren modelin Kesinlik–Duyarlılık eğrisi

Resim 5.7 de eğitilen modelin yaya sınıfı için elde ettiği Kesinlik–Duyarlılık eğrisini göstermektedir. Kesinlik-Duyarlılık eğrisi; modelin farklı güven eşięi deęerlerinde kesinlik ve duyarlılık arasındaki dengeyi aığa çıkarmaktadır. Model için mAP@0.5 deęeri 0.951 olarak hesaplanmıştır; bu deęer, modelin $IoU \geq 0.5$ koşulu altında yüksek ortalama doğruluk sağladığını göstermektedir.

Eğri incelendiğinde, duyarlılık geniş bir aralıkta (yaklaşık 0.8–0.98) yüksek kesinlik değerlerinde seyretmekte olup, bu durum modelin hem yüksek duyarlılık hem de düşük yanlış pozitif oranı sunduğunu göstermektedir. Ancak duyarlılık değerinin en uç noktasına (duyarlılık ≈ 1.0) yaklaşıldığında kesinlik’da belirgin bir düşüş gözlenmektedir; bu davranış tipik olarak çok düşük güven eşiği seçildiğinde modelin tüm olası adayları tespit ederken yanlış pozitif sayısını artırmasından kaynaklanır. Sonuç olarak, model genel performansı açısından $mAP@0.5 = 0.951$ güçlü bir başarıyı işaret etmekle birlikte, farklı IoU eşikleri ve farklı uygulama gereksinimlerine göre (ör. güvenlik kritik sistemlerde yüksek kesinlik mı yoksa yüksek duyarlılık mı tercih edildiği) eşik optimizasyonu yapılması önerilir.



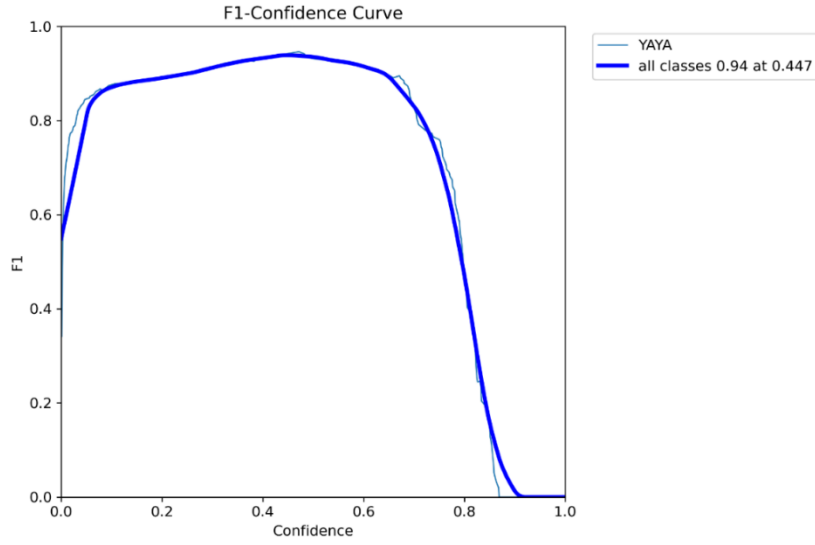
Resim 5.8 YOLOv12-N en yüksek performansı gösteren modelin Normalleştirilmiş Karmaşıklık Matrisi

Resim 5.8 de modele ait normalleştirilmiş karmaşıklık matrisi gösterilmektedir. Bu matris, modelin “Yaya” sınıfını ne ölçüde doğru tespit ettiğini ve arka plan sınıfı ile karışma durumlarını ayrıntılı biçimde yansıtmaktadır. Normalizasyon sayesinde değerler 0–1 aralığında oran olarak ifade edilmiştir.

Model, gerçek yaya örneklerinin %95’ini doğru bir şekilde “Yaya” sınıfı olarak tahmin etmiştir. Bu sonuç, modelin yaya tespit performansının oldukça yüksek olduğunu ve pozitif sınıf üzerinde güçlü bir ayırt etme yeteneğine sahip olduğunu göstermektedir.

Arka plan sınıfına ait piksellerin %5’lik bölümü yanlışlıkla "arkaplan" yerine “Yaya” olarak tahmin edilmiştir. Bu durum, özellikle yoğun yaya trafiği veya karmaşık sahne yapılarında modelin zaman zaman arka planı yaya olarak algıladığını göstermektedir. Bununla birlikte, arka plan sütununda 1.00 değerinin görülmesi, modelin arka planı doğru tanıma konusunda yüksek doğruluk seviyesine sahip olduğunu ortaya koymaktadır.

Genel olarak matristeki yüksek diyagonal değerler, modelin sınıfları doğru ayırt edebildiğini; düşük diyagonal dışı değerler ise hata oranının sınırlı olduğunu göstermektedir. Bu doğruluk yapısı, modelin simülasyon ortamında trafik ışıklarını yönetme ve yayaları gerçek zamanlı olarak tespit etme görevinde güvenilir bir performans sergilediğini desteklemektedir.



Resim 5.9 YOLOv12-N en yüksek performansı gösteren modelin F1–Güven Skoru Eğrisi

Resim 5.9 da modelin farklı güven skoru değerleri altında elde ettiği F1 performansının değişimi gösterilmektedir. Eğri, güven eşiği arttıkça modelin kesinlik ve duyarlılık arasındaki dengenin nasıl etkilendiğini ortaya koymaktadır.

Düşük güven skorlarında model daha fazla pozitif tahminde bulunduğundan, yanlış pozitiflerin artmasıyla F1 skoru görece düşük seyretmektedir. Güven eşiğinin yükselmesiyle birlikte yanlış pozitif sayısı azalmakta ve bu durum F1 skorunun belirgin

biçimde iyileşmesine katkı sağlamaktadır. Model, yaklaşık 0.447 güven eşiğinde 0.94 ile en yüksek F1 değerini elde etmiş olup bu eşik, kesinlik ve duyarlılık arasında en dengeli performansın sağlandığı optimum nokta olarak değerlendirilebilir.

Güven eşiğinin özellikle 0.70 seviyesinin üzerine çıkmasıyla birlikte F1 skorunda hızlı bir düşüş gözlenmektedir. Bu durum, yüksek eşik değerlerinin modelin pozitif tahminlerini aşırı biçimde sınırlandırarak gerçek pozitif örnekleri kaçırmaya neden olduğunu göstermektedir. Dolayısıyla, yüksek eşik seviyelerinin model performansını olumsuz etkilediği ve kullanım senaryosuna uygun bir eşik seçiminin kritik olduğu anlaşılmaktadır.

Genel olarak eğri, modelin farklı karar eşiklerinde sergilediği davranışın ayrıntılı bir analizini sunmakta; optimum eşik değerinin belirlenmesinin özellikle gerçek zamanlı yaya tespiti gibi güvenlik odaklı uygulamalarda model doğruluğunu ve kararlılığını artırmada önemli bir rol oynadığını göstermektedir.

Bu çalışmada, farklı YOLO sürümleri (YOLOv11, YOLOv12, YOLOv8), model boyutları (S, M, L, N), optimizasyon algoritmaları (AdamW, SGD, Adam), epok ve Parti boyutu değişkenleri dikkate alınarak geniş çaplı eğitim denemeleri gerçekleştirilmiştir. Modeller Python 3.10 ve PyTorch 2.1 kullanılarak uygulandı ve eğitim Google Colab'da GPU hızlandırma kullanılarak gerçekleştirildi ve Çizelge 5.2 de gösterilmiştir. Amaç, yaya tespiti görevinde en yüksek doğruluk, genellebilirlik ve kararlılığı sağlayan model konfigürasyonlarını belirlemektir.

Çizelge 5.2 YOLO Modellerle Eğitim Denemeleri

YOLO Sürümü	Parti boyutu	Epok	Optimizasyon	Kesinlik	Duyarlılık	mAP	F1-Skor
YOLOv11-S	16	50	AdamW	0.949	0.913	0.948	0.9307
YOLOv11-S	16	100	AdamW	0.960	0.939	0.957	0.9494
YOLOv11-S	16	50	SGD	0.921	0.955	0.972	0.9377
YOLOv11-S	16	100	SGD	0.938	0.954	0.951	0.9459

YOLO Sürümü	Parti boyutu	Epok	Optimizasyon	Kesinlik	Duyarlılık	mAP	F1-Skor
YOLOv11-S	16	50	Adam	0.880	0.893	0.921	0.8865
YOLOv11-S	16	100	Adam	0.953	0.902	0.943	0.9268
YOLOv11-S	16	150	AdamW	0.960	0.886	0.947	0.9215
YOLOv11-M	16	50	AdamW	0.894	0.878	0.938	0.8859
YOLOv11-S	8	50	AdamW	0.955	0.927	0.957	0.9408
YOLOv11-S	8	100	AdamW	0.929	0.890	0.930	0.9091
YOLOv11-N	8	50	AdamW	0.960	0.887	0.941	0.9221
YOLOv11-N	16	50	AdamW	0.962	0.854	0.936	0.9048
YOLOv11-S	8	50	SGD	0.935	0.902	0.935	0.9182
YOLOv11-L	8	50	AdamW	0.934	0.902	0.944	0.9177
YOLOv12-N	16	50	AdamW	0.937	0.947	0.951	0.9420
YOLOv12-N	16	50	SGD	0.910	0.904	0.942	0.9070
YOLOv12-N	16	50	Adam	0.943	0.873	0.942	0.9067
YOLOv12-N	16	75	AdamW	0.900	0.956	0.970	0.9272
YOLOv12-N	8	50	AdamW	0.928	0.902	0.964	0.9148
YOLOv12-S	16	50	AdamW	0.890	0.904	0.947	0.8969
YOLOv8-S	16	50	AdamW	0.929	0.923	0.956	0.9260
YOLOv8-N	16	50	AdamW	0.929	0.918	0.959	0.9235
YOLOv8-S	8	50	AdamW	0.959	0.930	0.947	0.9443
YOLOv8-S	8	100	AdamW	0.913	0.947	0.965	0.9297
YOLOv8-S	8	50	Adam	0.937	0.910	0.936	0.9233
YOLOv8-S	8	50	SGD	0.938	0.928	0.945	0.9330

5.2 En iyi Modelin Seçilip Çıktı Alınması

Bu üç model hem doğruluk hem kesinlik, hem de genel başarı (F1-skor ve mAP) açısından en güçlü performanslara sahiptir. Çizelge 5.3 de üç modelin en iyi performans gösteren en iyi model çıktıları sunulmuştur.

Çizelge 5.3 YOLO Sürümlerinden En İyi Performans Gösteren Modellerin Karşılaştırması

Model	Optimizasyon	Epok	Parti boyutu	F-1 Skor	mAP
YOLOv8-S	AdamW	50	8	%94.4	%94.7
YOLOv11-S	AdamW	100	16	%94.9	%95.7
YOLOv12-N	AdamW	50	16	%94.2	%95.1



Resim 5.10 YOLOv11-S modeline ait örnek tespit çıktıları

Resim 5.10’da, Çizelge 5.4’te sunulan performans karşılaştırmaları sonucunda en yüksek F1-skoru ve mAP değeri elde eden YOLOv11-S modeline ait örnek tespit çıktıları gösterilmektedir. Görseller, farklı aydınlatma koşulları, farklı yaya yoğunlukları ve çeşitli trafik senaryolarında modelin yaya tespit performansını ortaya koymaktadır.

Modelin, yaya sınıfını yüksek güven skorları (0.3–0.9 aralığında) ile başarılı bir şekilde tespit ettiği, özellikle yaya geçidi ve kavşak gibi kritik bölgelerde konumlanan yayaları doğru sınır kutuları ile belirlediği görülmektedir. Bu durum, YOLOv11-S modelinin hem precision hem de recall açısından dengeli bir performans sergilediğini ve F1-skorunun yüksek olmasının sahaya yansıyan bir sonucu olduğunu göstermektedir.

Ayrıca, farklı kamera açıları ve arka plan karmaşıklığına rağmen modelin yanlış pozitif ve yanlış negatif tespit oranlarının düşük olduğu gözlemlenmiştir. Elde edilen bu sonuçlar, YOLOv11-S modelinin gerçek zamanlı yaya güvenliği uygulamaları ve akıllı trafik yönetim sistemleri için uygun ve güvenilir bir çözüm sunduğunu doğrulamaktadır.

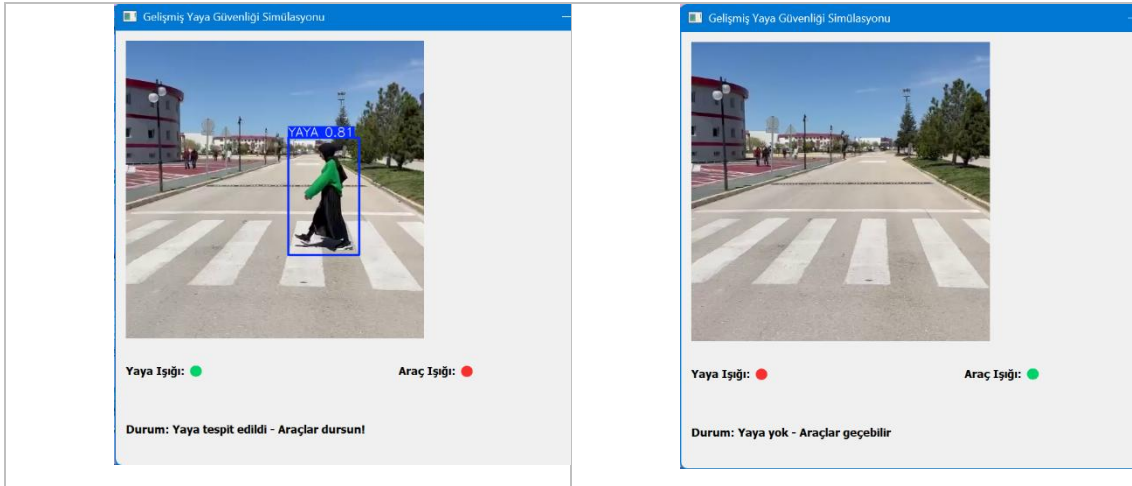
5.3 Simülasyon

Bu çalışma kapsamında yaya güvenliğini artırmak için bir simülasyon geliştirilmiştir. Geliştirilen sistem, bir video akışından gerçek zamanlı olarak yayaları tespit etmekte ve tespit edilen yayaların güvenliğini sağlamak için trafik ışıklarını kontrol etmektedir. Bu simülasyon, eğitilen modeller arasında en yüksek başarıyı sağlayan YOLOv11-S derin öğrenme modelini kullanarak yayaları tespit etmekte ve PyQt5 tabanlı bir arayüz ile görsel olarak kullanıcıya sunulmaktadır. PyQt5, Python tabanlı masaüstü uygulamaları geliştirmek için kullanılan modern bir arayüz kütüphanesidir. Bu çalışmada PyQt5’in yüksek performanslı grafik yapısı ve OpenCV ile uyumlu çalışabilmesi sayesinde, gerçek zamanlı tespit sonuçlarının kullanıcıya anlık olarak iletilmesi sağlanmıştır.

Bu sistemde video kaynağı olarak video.mp4 gibi bir video dosyası kullanılmıştır. Her kare, OpenCV kütüphanesi kullanılarak bu videodan yakalanmıştır. Gerçek zamanlı bir video kaynağı da kullanılabilir, ancak bu çalışmada örnek olarak bir video dosyası seçilmiştir. Yaya tespiti için Derin Öğrenme Modeli olarak eğitilen YOLOv11 modeli

sisteme entegre edilmiştir. Bu model her video karesinden yaya sınıfına ait nesneleri tespit etmektedir. Model bu tespitleri kullanarak trafik ışıklarını kontrol edip kullanıcıya bilgi verirken aynı zamanda aydınlatma ve durum mesajlarını da güncellemektedir. Görsel arayüz kendi kütüphanesi ile geliştirilmiştir. PyQt5 simülasyonu görsel olarak kullanıcıya sunan bir arayüzdür. Kullanıcı gerçek zamanlı video akışını izlerken yaya algılama ve trafik ışığı durumu hakkında bilgilendirilir.

Simülasyonun Çalışma Prensibi İlk olarak, simülasyon bir video kaynağı (video.mp4) kullanılarak başlamaktadır. Video kaynağındaki her kare OpenCV aracılığıyla okunmaktadır. Her video karesi YOLOv11 modeline gönderilir ve model görüntüdeki nesneleri sınıflandırır ve tespit edilen nesnelerin üzerine kutular çizer. Eğer görüntüde "yaya" sınıfına ait bir nesne tespit edilirse bu durum simülasyona yansıtılmaktadır.



Resim 5.11. Simülasyon Arayüzü

Şekil 4'te yayanın tespit edildiği veya edilmediği durumlar için simülasyon gösterilmektedir. Her video karesi üzerinde yapılan tespitler ve trafik ışığı durumu QImage ve QPixmap kullanılarak PyQt5 arayüzüne aktarılmaktadır. Böylece kullanıcı istediği zaman simülasyonun o anki durumunu izleyebilmektedir. Tespit edilen yayalar ve trafik ışıkları görsel olarak ekranda gösterilmektedir.

Simülasyonun önemli bir özelliği de her video karesinde yapılan yaya tespitine göre trafik ışıklarının otomatik olarak güncellenmesidir. Kullanıcı, yayaların varlığına göre ışıkların nasıl değiştiğini ve simülasyonun nasıl çalıştığını anlık olarak izleyebiliyor. Böyle bir

sistem, özellikle akıllı şehirlerde yaya güvenliğini artırmaya yönelik önemli bir adım olabilir. Yaya algılama ve trafik ışıklarının senkronize yönetimi hem yayaların hem de araçların güvenliğini sağlamak için kritik öneme sahiptir. Bu simülasyon, video akışını izleyen kullanıcıya her karedeki yaya algılama ve trafik ışığı durumu hakkında gerçek zamanlı olarak bilgi sağlar. Bu sayede simülasyonun nasıl çalıştığı ve yayaların güvenliği hakkında önemli veriler kullanıcıya sunulur.

Bu çalışmada, yaya güvenliğini artırmak amacıyla geliştirilen bir sistem kapsamında derin öğrenme tabanlı yaya tespiti gerçekleştirilmiş ve bu doğrultuda gerçek zamanlı bir masaüstü simülasyon uygulaması tasarlanmıştır. Model eğitimi için 12 MP kamera ile 4032x3024 piksel çözünürlükte çekilmiş toplam 620 görüntüden oluşan özel bir veri kümesi kullanılmıştır. Bu veri kümesi hem tekli hem de çoklu yaya geçitlerini içeren doğal senaryolarla zenginleştirilmiştir. Etiketleme işlemleri "yaya" sınıfı için MakeSense aracı ile gerçekleştirilmiştir. YOLOv8-S, YOLOv11-S ve YOLOv12-N mimarileri, Python 3 ve T4 GPU çalışma zamanı ile Google Colab kullanılarak eğitildi. F1 skoruna göre en iyi başarı YOLOv11-S modeli ile elde edilmiştir. Bu nedenle bu model simülasyon sistemine entegre edilmiştir.

6. TARTIŞMA ve SONUÇ

Bu tez çalışmasında, şehir içi yaya güvenliğini artırmak amacıyla derin öğrenme tabanlı gerçek zamanlı bir yaya tespit sistemi geliştirilmiştir. Eğitim verisi, farklı açılardan ve ışık koşullarından çekilmiş, 12 MP kamera ile elde edilen 4032×3024 piksel çözünürlüğünde toplam 620 görüntüden oluşan özel bir veri kümesi kullanılarak hazırlanmıştır. Veri kümesi hem tekli hem de çoklu yaya geçitlerini içeren doğal senaryolarla zenginleştirilmiş ve etiketleme işlemleri “yaya” sınıfı için MakeSense aracıyla gerçekleştirilmiştir.

Literatürde son yıllarda geliştirilmiş YOLO algoritmaları referans alınarak YOLOv8-S, YOLOv11-S ve YOLOv12-N modelleri eğitim ve karşılaştırmalı analizlerde kullanılmıştır. Eğitimler sonucunda, en yüksek performans %94,94 F1 skoru ile YOLOv11-S modeli tarafından elde edilmiştir. Eğitim süreci tamamlandıktan sonra, modelin ağırlıkları kullanılarak PyQt5 ve Qt Designer ile masaüstü simülasyon uygulaması geliştirilmiştir. Bu uygulama aracılığıyla kullanıcılar, gerçek zamanlı video akışları üzerinden yayaları tespit edebilmekte ve tespit sonuçları trafik ışığı kontrol sistemi ile entegre edilerek güvenlik senaryoları test edilebilmektedir.

YOLOv11-S modelinin yaya tespiti konusunda en yüksek performansı göstermesi, optimize edilmiş mimarisi ve yenilikçi teknik özellikleriyle doğrudan ilişkilidir. Modelin omurga ve Boyun yapısı, sahnedeki yayaların konum ve hareketlerini ayrıntılı bir şekilde analiz edebilmesini sağlamaktadır. Bu mimari avantaj, özellikle çoklu ve hareketli yayaların tespitinde yüksek doğruluk elde edilmesine imkân tanımaktadır. Ayrıca, modelin parametre sayısının dengeli olması, eğitim sürecinde hızlı öğrenmeyi desteklerken, küçük nesnelerin veya uzak mesafedeki yayaların algılanmasını kolaylaştırmaktadır.

Buna ek olarak, YOLOv11-S modelinde kullanılan geliştirilmiş özellik çıkarım katmanları, doğruluk ile çıkarım hızı arasındaki optimum dengeyi sağlamaktadır. Bu sayede model, farklı açılardan ve farklı hızlarda görülen yayaları yüksek doğrulukla tespit edebilmektedir. Modelin kompakt ve optimize edilmiş yapısı, gerçek zamanlı çıkarım

performansını artırarak sistemin kararlı ve güvenilir bir şekilde çalışmasını mümkün kılmaktadır.

Sonuç olarak, bu tez çalışması kapsamında geliştirilmiş olan derin öğrenme tabanlı yaya tespit sistemi, şehir içi yaya güvenliği simülasyonlarında diğer modellere göre üstün performans göstermiştir. Sistem, tespit ettiği yayaların varlığı doğrultusunda araçlara dur, yayalara ise geç sinyali vererek güvenliği ön planda tutmakta ve gerçek zamanlı trafik yönetimine katkı sağlamaktadır. Bu çalışma, akıllı şehir uygulamaları ve trafik güvenliği sistemleri için önemli bir referans ve temel oluşturabilecek niteliktedir.

Çizelge 6.5 Önerilen modelin diğer yöntemlerle karşılaştırılması

Kriter	Bizim Çalışmamız	Çağla Erdem Öztaş (2024)	Razzok vd. (2022)	Tian vd. (2015)
Yaya Algılama	Evet	Evet	Evet	Evet
Veri Kümesi Türü	Orijinal olarak oluşturulmuş yaya görüntüsünü içeren özel veri kümesi.	FLIR ADAS ve TarDAL	COCO ve Daimler Mono Yaya	Caltech Yaya
Görüntü Sayısı	620	600	4097	350.000
Eğitilen Mimari	YOLOv8, YOLOv11, YOLOv12	YGH+DVM, AlexNet, DarkNet-53, ResNet-101	Önceden eğitilmiş modeller Daha hızlı R-CNN, SSD ve YOLOv3	DeepParts
F1 Puanı / AP (%)	YOLOv11-S 0.9494.	ResNet-101 %95 F1 puanı,	YOLOv3 (AP): %31,6;	Hata Oranı %11,89;

			Daha Hızlı R-CNN: 39,4; SSD: %22,68.	
Gerçek Zamanlı Uygulama	PyQt5 tabanlı masaüstü uygulaması	-	-	-

Çizelge 6.5'teki çalışmalar incelendiğinde, bu çalışmanın literatürdeki diğer çalışmalardan hem özgün veri kullanımı hem de gerçek zamanlı uygulama açısından farklılaştığı görülmektedir. Geliştirilen sistem, özgün olarak oluşturulmuş toplam 620 yaya görüntüsünden oluşan özel bir veri kümesi üzerinde eğitilmiştir. Bu yönüyle hazır veri kümeleri kullanan çalışmalardan farklılık göstermektedir. Çağla Erdem Öztaş'ın (2024) çalışmasında FLIR ADAS ve TarDAL gibi hazır veri setleri kullanılmıştır. Benzer şekilde Razzok vd. (2022) çalışmasında COCO ve Daimler Mono Yaya veri kümeleri; Tian vd. (2015) çalışmasında ise büyük ölçekli Caltech Yaya veri kümesi tercih edilmiştir. Veri boyutu açısından karşılaştırıldığında, bu çalışmadaki 620 görüntüden oluşan özel veri kümesi, Çağla Erdem Öztaş'ın 600 görüntü, Razzok ve arkadaşlarının 4097 görüntü ve Tian vd. 350.000 görüntüden oluşan veri kümelerine kıyasla daha küçük boyuttadır. Ancak elde edilen sonuçlar açısından oldukça başarılıdır. Model mimarisi açısından bu çalışmada YOLOv8, YOLOv11 ve YOLOv12 gibi modern YOLO versiyonları test edilmiş ve en yüksek başarıyı sağlayan YOLOv11-S modeli tercih edilmiştir. Diğer çalışmalarda ise sırasıyla YGH+SVM, AlexNet, DarkNet-53, ResNet-101 (Çağla Erdem Öztaş), YOLOv3, Faster R-CNN, SSD (Razzok vd.) ve DeepParts (Tian vd.) gibi farklı mimariler kullanılmıştır. Performans karşılaştırmalarına göre bu çalışmada elde edilen 0,9494 F1 skoru Tablo 1'de yer alan diğer çalışmalardaki bazı modellerin üzerinde bir başarı göstermektedir. Örneğin Çağla Erdem Öztaş'ın ResNet-101 modeli ile elde ettiği %95 F1 skoru ile benzer seviyededir. Razzok ve arkadaşlarının çalışmasında YOLOv3 için %31,6, Faster R-CNN için %39,4 ve SSD için %22,68 AP skorları elde edilmiştir. Tian ve arkadaşlarının çalışmasında ise hata oranı %11,89 olarak raporlanmıştır. Ayrıca çizelge 6.5'teki diğer çalışmalardan farklı olarak bu çalışmada sadece adresinde gerçek zamanlı bir masaüstü uygulaması geliştirilmiştir. PyQt5 tabanlı bu uygulama kullanıcıya yaya algılama ve trafik ışığı kontrolünü eş zamanlı olarak

sağlamaktadır. Bu yönüyle geliştirilen sistem yaya güvenliği açısından sadece teorik olarak değil pratik olarak da uygulanabilir bir çözüm sunmaktadır.

Bu çalışmanın en önemli avantajlarından biri, tamamen özgün biçimde oluşturulmuş ve farklı açılardan çekilmiş 620 görüntüden oluşan özel bir veri kümesinin kullanılmasıdır. Hazır veri setleri yerine özgün görüntülerden oluşturulan bu veri kümesi, sistemin gerçek şehir içi koşullarına daha yakın bir senaryoda eğitilmesini sağlamıştır. Bunun yanı sıra YOLOv8, YOLOv11 ve YOLOv12 gibi güncel ve modern nesne algılama mimarilerinin karşılaştırmalı analizinin gerçekleştirilmesi, literatürdeki pek çok çalışmada yer almayan geniş kapsamlı bir model değerlendirmesi sunmaktadır. Elde edilen yüksek F1 skoru ve PyQt5 tabanlı gerçek zamanlı bir masaüstü uygulamasının geliştirilmesi, teorik bir model sunmanın ötesine geçerek çalışmanın pratik anlamda uygulanabilirliğini ve işlevselliğini göstermektedir. Sistem, tespit edilen yayaların varlığına göre trafik ışıklarını yönlendirebilmekte ve bu yönüyle akıllı şehir uygulamalarında doğrudan kullanılacak bir çerçeve sunmaktadır.

Bununla birlikte, çalışmanın belirli sınırlılıkları bulunmaktadır. Öncelikle kullanılan veri kümesinin 620 görüntü ile sınırlı olması, özellikle derin öğrenme modellerinin yüksek veri gereksinimleri göz önünde bulundurulduğunda kapsamın genişletilebilir olduğu bir noktaya işaret etmektedir. Veri kümesinin yalnızca bir şehirde ve tek bir kamera türü ile oluşturulmuş olması, farklı çevresel koşulların, farklı kamera sensörlerinin veya olumsuz hava şartlarının sistem performansı üzerindeki etkilerinin değerlendirilememesine yol açmıştır. Ayrıca çalışmada yalnızca “yaya” sınıfının tespit edilmesi, sistemin karmaşık trafik ortamlarında karşılaşılabilecek araç çeşitliliği, bisikletli kullanıcılar veya elektrikli scooter gibi farklı nesneleri kapsamadığını göstermektedir. Geliştirilen uygulamanın masaüstü ortamında test edilmiş olması da sistemin gömülü donanım veya mobil cihazlar üzerindeki performansının değerlendirilmesine engel olmuştur. Bu sınırlılıklar, çalışmanın mevcut kapsamını belirlemekle birlikte gelecekte yapılacak geliştirmeler için yol gösterici niteliktedir.

Gelecek çalışmalar kapsamında sistemin çok daha geniş ve çeşitli koşulları içeren bir veri kümesi ile yeniden eğitilmesi planlanmaktadır. Özellikle gece, yağmur, sis ve değişken

ışık koşulları gibi zorlu senaryoların veri setine eklenmesi, modelin gerçek çevresel değişimlere karşı daha dayanıklı hâle getirilmesini sağlayacaktır. Ayrıca RGB görüntülerin yanı sıra termal kameralar veya derinlik sensörlerinin entegre edilmesiyle çoklu sensör füzyonu sağlanarak tespit performansının artırılması öngörülmektedir. Model yapısı açısından Transformer tabanlı DETR, DETA veya RT-DETR gibi modern mimarilerin sisteme dahil edilmesi ve bu mimarilerin YOLO modelleri ile karşılaştırılması, performans analizine yeni bir boyut kazandıracaktır. Uygulama düzeyinde ise geliştirilen sistemin NVIDIA Jetson, Raspberry Pi 5 gibi gömülü sistemlerde optimize edilerek test edilmesi, saha uygulamalarında gerçek zamanlı olarak kullanılabilecek prototiplerin geliştirilmesine katkı sağlayacaktır. Ayrıca, akıllı şehir platformlarına IoT tabanlı bağlantı ile entegre edilmesi ve trafik yoğunluğu, araç hızları ve yaya hareketliliği gibi ek değişkenlere göre adaptif trafik yönetim mekanizmaları oluşturulması mümkündür. Bu geliştirmeler sayesinde yaya güvenliği odaklı bu çalışmanın daha geniş ölçekte uygulanabilir bir yapıya kavuşması hedeflenmektedir.

Planlanan sistemin yalnızca tek bir sabit kameradan elde edilen görüntülerle değil, farklı açılardan ve farklı konumlardan elde edilen çoklu kamera görüntüleri ile çalışabilecek şekilde genişletilmesi planlanmaktadır. Özellikle 360 derece dönebilen kamera sistemlerinin entegrasyonu ile kavşak, yaya geçidi ve yoğun trafik bölgelerinde kör noktaların minimize edilmesi ve daha kapsamlı bir çevresel algı sağlanması hedeflenmektedir. Çoklu kamera verilerinin eşzamanlı olarak işlenmesiyle, farklı perspektiflerden elde edilen bilgilerin birleştirilmesi sayesinde yaya tespit doğruluğunun ve sistemin genel güvenilirliğinin artırılması öngörülmektedir.

Ayrıca geliştirilen sistemin yalnızca elektrikli veya belirli araç grupları ile sınırlı kalmayıp, içten yanmalı, hibrit ve otonom araçlar dâhil olmak üzere tüm araç türleri ile uyumlu çalışabilecek şekilde tasarlanması planlanmaktadır. Araçtan bağımsız bir yapı benimsenerek, sistemin V2X haberleşme altyapıları ile entegre edilebilmesi ve farklı üreticilere ait araç platformlarında kullanılabilmesi amaçlanmaktadır. Bu sayede yaya güvenliğine yönelik çözümün daha geniş bir ulaşım ekosisteminde uygulanabilirliği artırılabilecektir.

Bununla birlikte, çoklu kamera ve sensörlerden elde edilen büyük hacimli verilerin daha etkin bir biçimde işlenebilmesi amacıyla gelişmiş bir veri işleme altyapısının sisteme entegre edilmesi planlanmaktadır. Edge ve bulut tabanlı hibrit mimariler kullanılarak gerçek zamanlı veri akışının sağlanması, uzun dönemli trafik verilerinin analiz edilmesi ve yaya-araç etkileşimlerine yönelik örüntülerin çıkarılması hedeflenmektedir. Bu yaklaşım, yalnızca anlık yaya tespiti değil, aynı zamanda trafik yoğunluğu tahmini, riskli bölgelerin belirlenmesi ve adaptif trafik planlaması gibi ileri seviye akıllı şehir uygulamalarına da zemin hazırlayacaktır.

Gerçek zamanlı uygulamanın başarısı ve trafik sinyalleri ile entegre edilebilmesi, bu çözümün akıllı şehir altyapılarına kolaylıkla adapte edilebileceğini göstermektedir. Literatürdeki çalışmalarla karşılaştırıldığında bu çalışma, özgün bir veri seti kullanımı ve uygulanabilir masaüstü simülasyon sistemi ile teorik olduğu kadar pratik faydalar da sunmaktadır. Gelecekte sistemin farklı sensör verileri ile genişletilmesi ve düşük ışık ve olumsuz hava koşullarında güvenilir çalışması için geliştirilmesi planlanmaktadır. Ayrıca mobil ve gömülü sistemlerde çalışabilecek daha optimize versiyonların geliştirilmesi ve saha uygulamalarına entegre edilmesi hedeflenmektedir.

7. KAYNAKLAR

- Alif M A R, 2024, YOLOv11 for vehicle detection: Advancements, performance, and applications in intelligent transportation systems, arXiv preprint, arXiv:2410.22898.
- Apicella A, Scognamiglio R, Tramontana P, Scala A, 2021, A survey on modern trainable activation functions, *Neural Networks*, 138, 14-32.
- Arslan K, 2020, Eğitimde yapay zekâ ve uygulamaları, *Batı Anadolu Eğitim Bilimleri Dergisi*, 11(1), 71-88.
- Basha S S, Dubey S R, Pulabaigari V, Mukherjee S, 2020, Impact of fully connected layers on performance of convolutional neural networks for image classification, *Neurocomputing*, 378, 112-119.
- Chen X, Yang J, Zhang J, 2014, Pedestrian detection with deep convolutional neural network, *Asian Conference on Computer Vision Proceedings*, 354-365.
- Çetin M, Karakuş A, Geçgel Ş, 2024, Gelişen bir paradigma: Yapay zekâ okuryazarlığı, *International Journal of Active Learning*, 8(1), 50-63.
- Dollar P, Wojek C, Schiele B, Perona P, 2009, Pedestrian detection: A benchmark, *IEEE Conference on Computer Vision and Pattern Recognition*, 304-311.
- İnik Ö, Ülker E, 2017, Derin öğrenme ve görüntü analizinde kullanılan derin öğrenme modelleri, *Gaziosmanpaşa Bilimsel Araştırma Dergisi*, 6(3), 85-104.
- Jie H J, Wanda P, 2020, RunPool: A dynamic pooling layer for convolution neural network, *International Journal of Computational Intelligence Systems*, 13(1), 66-76.
- Jocher G, 2020, Ultralytics YOLOv5, Software Repository, Zenodo, doi:10.5281/zenodo.3908559.
- Karabayır A, 2016, Yaya-Taşıt trafiği çatışmasının incelenmesi, *Dokuz Eylül Üniversitesi, Yüksek Lisans Tezi*.
- Kazdal S, 2013, Beyin tümörlerinin ileri görüntü işleme ve örüntü tanıma teknikleri kullanılarak bilgisayar destekli tespiti, *Marmara Üniversitesi, Yüksek Lisans Tezi*.

- Ketkar N, Moolayil J, 2021, Convolutional neural networks, Deep Learning with Python: Learn Best Practices of Deep Learning Models with PyTorch, 197-242.
- Khanam R, Hussain M, 2024, YOLOv11: An overview of the key architectural enhancements, arXiv preprint, arXiv:2410.17725.
- Kotthapalli M, Ravipati D, Bhatia R, 2025, YOLOv1 to YOLOv11: A comprehensive survey of real-time object detection innovations and challenges, arXiv preprint, arXiv:2508.02067.
- Krizhevsky A, Sutskever I, Hinton G E, 2017, ImageNet classification with deep convolutional neural networks, Communications of the ACM, 60(6), 84-90.
- Lu L, 2023, Improved YOLOv8 detection algorithm in security inspection image, arXiv preprint, arXiv:2308.06452.
- Murat S, 2021, İnsansız hava aracı görüntülerinden derin öğrenme yöntemleriyle nesne tanıma, Maltepe Üniversitesi, Yüksek Lisans Tezi.
- Osisanwo F Y, Akinsola J E T, Awodele O, Hinmikaiye J O, Olakanmi O, Akinjobi J, 2017, Supervised machine learning algorithms: Classification and comparison, International Journal of Computer Trends and Technology, 48(3), 128-138.
- Ouassine Y, El Hassouni M, Bouikhalene B, 2025, Deep learning for automated coral reef monitoring: A novel system based on YOLOv8 detection and DeepSORT tracking, Ecological Informatics, 89, 103170.
- Özdemir E, Türkoğlu İ, 2022, Yazılım güvenlik açıklarının evrimsel sinir ağları ile sınıflandırılması, Fırat Üniversitesi Mühendislik Bilimleri Dergisi, 34(2), 517-529.
- Öztürk K, Şahin M E, 2018, Yapay sinir ağları ve yapay zekâya genel bir bakış, Takvim-i Vekayi, 6(2), 25-36.
- Pandey D, Niwaria K, Chourasia B, 2019, Machine learning algorithms: A review, Machine Learning, 6(2), 1-10.
- Razzok M, Al-Qudah Y, Al-Zoubi A, 2022, Pedestrian detection system based on deep learning, International Journal of Advanced Applied Sciences, 9(4), 195-202.

- Reich Y, Barai S V, 1999, Evaluating machine learning models for engineering problems, *Artificial Intelligence in Engineering*, 13(3), 257-272.
- Sapkota R, Karkee M, 2025, Improved YOLOv12 with LLM-generated synthetic data for enhanced apple detection and benchmarking against YOLOv11 and YOLOv10, *arXiv preprint*, arXiv:2503.00057.
- Sungur İ, Akdur R, Piyal B, 2014, Analysis of traffic accidents in Turkey, *Ankara Medical Journal*, 14(3), 1-10.
- Tan F G, Yıldız K, Demirci A, 2021, Derin öğrenme teknikleri ile nesne tespiti ve takibi üzerine bir inceleme, *Avrupa Bilim ve Teknoloji Dergisi*, 25, 159-171.
- Tian Y, Yan J, Zhang Y, Zha H, 2015, Deep learning strong parts for pedestrian detection, *IEEE International Conference on Computer Vision*, 1904-1912.
- Tian Y, Ye Q, Doermann D, 2025, YOLOv12: Attention-centric real-time object detectors, *arXiv preprint*, arXiv:2502.12524.
- Tomè D, Monti F, Baroffio L, Bondi L, Tagliasacchi M, Tubaro S, 2016, Deep convolutional neural networks for pedestrian detection, *Signal Processing: Image Communication*, 47, 482-489.
- Türkoğlu M, Aydın O, Aksoy B, 2020, Derin evrişimsel sinir ağı kullanılarak kayısı hastalıklarının sınıflandırılması, *Bitlis Eren Üniversitesi Fen Bilimleri Dergisi*, 9(1), 334-345.
- Varoquaux G, Colliot O, 2023, Evaluating machine learning models and their diagnostic value, *Machine Learning for Brain Disorders*, 601-630.
- Vikruthi S, Prakash N, Kumar R, 2025, Detection of emergency vehicles in traffic and assign traffic free path using deep learning, *International Conference on Sentiment Analysis and Deep Learning*, 1252-1261.
- Wang A, Bochkovskiy A, Liao H-Y M, 2024, YOLOv10: Real-time end-to-end object detection, *Advances in Neural Information Processing Systems*, 37, 107984-108011.

- Wang C-Y, Bochkovskiy A, Liao H-Y M, 2023, YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors, IEEE Conference on Computer Vision and Pattern Recognition, 7464-7475.
- Wang C-Y, Yeh I-H, Liao H-Y M, 2024, YOLOv9: Learning what you want to learn using programmable gradient information, European Conference on Computer Vision, 1-21.
- Wang J-M, Li Y, Zhang H, Liu X, 2024, Parametric analysis of craniocerebral injury mechanism in pedestrian traffic accidents based on finite element methods, Chinese Journal of Traumatology, 27(4), 187-199.
- Xu H, Zhang L, Wu Y, Wang Z, 2024, Deep learning-based pedestrian detection using RGB images and sparse LiDAR point clouds, IEEE Transactions on Industrial Informatics, 20(5), 7149-7161.
- Zilyas D, Yılmaz A, 2023, Makine öğrenmesi yöntemleri ile eğitim başarısının tahmini modeli, Dicle Üniversitesi Mühendislik Fakültesi Mühendislik Dergisi, 14(3), 437-447.
- Zou Z, Shi Z, Guo Y, Ye J, 2023, Object detection in 20 years: A survey, Proceedings of the IEEE, 111(3), 257-276.

ÖZGEÇMİŞ

Adı Soyadı : Rehnüma Turunç
Doğum Yeri ve Tarihi : Afyonkarahisar / 22.08.2000
Yabancı Dili : İngilizce
İletişim (Telefon / e-posta) : 0 545 376 16 76/ kucukilhanrehnuma @gmail.com

Eğitim Durumu (Kurum ve Yıl)

Lise : Afyon Kocatepe Anadolu Lisesi (2014 – 2018)
Lisans : Necmettin Erbakan Üniversitesi, Bilgisayar Mühendisliği
Böl., (2018– 2023)

Çalıştığı Kurum/Kurumlar ve Yıl

: Smart Software Sağlık Ticaret Sanayi Anonim Şirketi
(2023 –2023)
: Hobala E Ticaret Yazılım Bilişim Reklam Ve Aracılık
Limited (2023-2025)
: EntegreTürk Teknoloji Limited Şirketi (2023 – Devam
Ediyor)

Yayınları (SCI ve diğer) :

Candan E N, Küçükilhan R, Eroğlu A, 2022, Spam Mail Detection in Turkish and English Languages: A Holistic Study of AI-based Techniques Including Individual, Ensemble and

Hybrid Approaches, Necmettin Erbakan Üniversitesi Fen ve Mühendislik Bilimleri Dergisi, Cilt 7, 189–205.

Küçükilhan R, Yurttakal A H, 2025, Advanced Pedestrian Detection and Intelligent Traffic Management via Deep Learning, International Journal of 3D Printing Technologies and Digital Industry, Cilt 9, 707–720.