

T.C.
İSTANBUL BEYKENT ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI
BİLGİSAYAR MÜHENDİSLİĞİ BİLİM DALI

**KAMERA GÖRÜNTÜLERİNDEN ÇARPIŞMA
TAHMİNLEMESİ**

Doktora Tezi

Tezi Hazırlayan
Reşat Buğra ERKARTAL

İstanbul, 2025

T.C.
İSTANBUL BEYKENT ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI
BİLGİSAYAR MÜHENDİSLİĞİ BİLİM DALI

**KAMERA GÖRÜNTÜLERİNDEN ÇARPIŞMA
TAHMİNLEMESİ**

Doktora Tezi

Tezi Hazırlayan

Reşat Buğra ERKARTAL

Öğrenci No

1908021003

ORCID ID

0000-0002-0626-3181

Danışman

Doç. Dr. Atınç YILMAZ

İstanbul, 2025

YEMİN METNİ

Doktora Tezi olarak sunduđum “**Kamera Görüntülerinden Çarpışma Tahminlemesi**” başlıklı bu çalışmanın, bilimsel ahlak ve geleneklere uygun şekilde tarafımdan yazıldığını, bu tezdeki bütün bilgileri akademik ve etik kurallar içinde elde ettiđimi, yararlandığım eserlerin tamamının kaynaklarda gösterildiğini ve çalışmamın içinde kullanıldıkları her yerde bunlara atıf yapıldığını, patent ve telif haklarını ihlal edici bir davranışının olmadığını belirtir ve bunu onurumla doğrularım. 06/10/2025

Reşat Buğra ERKARTAL

TEZ ONAYI

İstanbul Beykent Üniversitesi Lisansüstü Eğitim Enstitüsü *Bilgisayar Mühendisliği* *Doktora* öğrencisi **1908021003** numaralı *Reşat Buğra ERKARTAL*'ın hazırladığı "*Kamera Görüntülerinden Çarpışma Tahminlemesi*" konulu DOKTORA TEZİ ile ilgili TEZ SAVUNMA SINAVI Lisansüstü Öğretim ve Sınav Yönetmeliğinin ilgili maddesi uyarınca 06.10.2025 günü saat 15.00'te Zoom programı aracılığıyla on-line olarak yapılmış, sorulan sorulara alınan cevaplar sonucunda adayın tezinin *KABUL*'üne *OYBİRLİĞİ*'yle karar verilmiştir.

JÜRİ ÜYESİ	KANAATI	İMZA
Doç. Dr. At*** YI*** (Danışman) (İstanbul Beykent Üniversitesi)		
Dr. Öğr. Üyesi Ed*** ŞA*** (Üye) (İstanbul Beykent Üniversitesi)		
Dr. Öğr. Üyesi İs*** KA*** (Üye) (Işık Üniversitesi)		
Dr. Öğr. Üyesi Üm*** ÖZ*** (Üye) (İstanbul Beykent Üniversitesi)		
Dr. Öğr. Üyesi Bu*** İŞ*** (Üye) (İstanbul Topkapı Üniversitesi)		

Adı ve Soyadı : Reşat Buğra ERKARTAL
Danışmanı : Doç. Dr. Atınç YILMAZ
Derecesi ve Tarihi : Doktora Tezi, 2025
Alanı : Bilgisayar Mühendisliği
Anahtar Kelimeler : Otonom Araçlar, Görüntü İşleme, Yapay Zeka, Çarpışma Tahminleme

ÖZ

KAMERA GÖRÜNTÜLERİNDEN ÇARPIŞMA TAHMİNLEMESİ

Otonom araç teknolojileri geliştikçe, dinamik trafik ortamlarında güvenliği sağlamak için güvenilir algılama ve yanıt mekanizmalarına duyulan ihtiyaç kritik hale gelmektedir. Bu çalışma, tek kamera girişi kullanarak araç çarpışmalarını gerçek zamanlı olarak tespit ve tahmin etmek için R-CNN, Kalman Filtresi, Gizli Markov Zinciri ve Yapay Sinir Ağı ile geliştirilmiş yeni bir hibrit model sunmaktadır. Sistem, video akışlarında nesne tespiti ve çoklu nesne takibi gerçekleştirmekte, modifiye edilmiş bir Kalman Filtresi aracılığıyla olasılıksal hareket tahmini uygulamakta ve Gizli Markov Zinciri ile Yapay Sinir Ağı'nın hibrit yaklaşımını kullanarak durum geçişlerini iyileştirmektedir. Önerilen mimari, tespit doğruluğunu önemli ölçüde artırmakta ve hızlı hareket altında gecikmeyi azaltmaktadır; bu da geleneksel filtreleme tekniklerinin sınırlamalarını tekrar göz önüne alınması gerekliliğini ortaya koymaktadır. Model CARLA simülasyonu, video görüntüleri ve gerçek zamanlı olarak test edilmiş ve gerçekleştirilen deneyler yüksek etkinlik göstermekte olup video analizinde %94.5 tespit doğruluğu ve gerçek zamanlı uygulamada %91.2 doğruluk sağlamaktadır. ANN entegrasyonu, tespit gecikmesinde 0.5 saniyelik bir azalma sağlamış olup, bu da ortalama insan tepki süreleri göz önüne alındığında kritik bir iyileştirme sağlamaktadır. Bu sonuçlar, derin öğrenmenin olasılıksal modelleme ile birleştirilmesinin zamansal olarak göz ardı edilebilecek bir artışa neden olsa da tespit tutarlılığında daha verimli sonuçlar elde etmektedir.

Name and Surname : Reşat Buğra ERKARTAL
Supervisor : Assoc. Prof. Atınç YILMAZ
Degree and Date : Ph.D, 2025
Major : Computer Engineering
Keywords : Autonomous Vehicles, Image Processing, Artificial Intelligence, Crash Prediction

ABSTRACT

CRASH PREDICTION FROM CAMERA IMAGES

As autonomous vehicle technologies advance, the need for reliable perception and response mechanisms to ensure safety in dynamic traffic environments is becoming critical. This work presents a novel hybrid model, combining R-CNN, a Kalman Filter, a Hidden Markov Model (HMM), and an Artificial Neural Network (ANN), to detect and predict vehicle collisions in real-time using a single camera input. The system performs object detection and multi-object tracking in video streams, applies probabilistic motion prediction via a modified Kalman Filter, and enhances state transitions using a hybrid approach of the HMM and ANN. The proposed architecture significantly increases detection accuracy and reduces latency under rapid motion, addressing the limitations of traditional filtering techniques. The model was tested in the CARLA simulation, on video footage, and in real-time. The conducted experiments demonstrate high efficacy, achieving 94.5% detection accuracy in video analysis and 91.2% accuracy in the real-time application. The ANN integration resulted in a 0.5-second reduction in detection latency, a critical improvement when considering average human reaction times. These results suggest that combining deep learning with probabilistic modeling, despite introducing a negligible temporal overhead, yields more efficient outcomes in terms of detection consistency.

İÇİNDEKİLER

Sayfa No.

ÖZ

ABSTRACT

TABLolar LİSTESİ	iii
ŞEKİLLER LİSTESİ	iv
KISALTMALAR	v
SÖZLÜK.....	vii
GİRİŞ.....	1

1. OTONOM ARAÇLARIN TANIMI VE TARİHÇESİ

1.1. Otonom Araçların Tanımı ve Tarihçesi.....	2
1.2. Otonom Araçların Sağladığı Avantajlar ve Toplumsal Etkiler	4
1.3. Nesne Algılamada Kullanılan Teknikler	6
1.3.1. İki Boyutlu Nesne Tespit Yöntemleri	6
1.3.2. Üç Boyutlu Nesne Algılama	7
1.3.3. Yapay Zeka Yöntemleri	8
1.3.4. Sensor Fusion Tekniği	9
1.3.5. Davranışsal Tahminin Sürüşteki Uygulamaları	11
1.3.6. Yaya Davranışlarının Öngörüsü.....	11
1.3.7. Öngörücü Araç Davranışları	12

2. LİTERATÜR ÇALIŞMASI

3. MODEL

3.1. Yapay Zekânın Temel Özellikleri	17
3.2. R-CNN.....	19
3.2.1. Bölge Önerme (Region Proposal)	19
3.2.2. Özellik Çıkarımı (Feature Extraction)	20
3.3. Kalman Filtresi	22
3.4. Saklı Markov Zinciri	25
3.5. ANN.....	27
3.5.1. Nöronlar	27
3.5.2. Aktivasyon Fonksiyonları	27

3.6. Sinir Ağı Mimarisi	28
3.7. Önerilen Model	29
3.7.1. Veri Seti ve Ön İşleme	32
3.7.2. Kalman Filtresi.....	33
3.7.3. R-CNN	36
3.7.4. Izgara Modeli	37
3.7.5. Saklı Markov Zinciri ve ANN Kararları	39
3.8. Deney Kurulumu.....	40
SONUÇ	42
KAYNAKÇA.....	56



TABLolar LİSTESİ

	Sayfa No.
Tablo 1. Otonom Araçlarda Kullanılan Sensör Türleri.....	3
Tablo 2. Literatürde Yapılan Çalışmalar.....	16
Tablo 3. CNN Performans Değerlendirmesi.....	18
Tablo 4. Sık kullanılan aktivasyon Fonksiyonları.....	27
Tablo 5. Izgara Modelinin Koordinatları	38
Tablo 6. Hangi Aracın Hangi Izgaraya Ait Olduğu Tablosu, Doğrusal Kalman Filtresi	42
Tablo 7. Hangi Aracın Hangi Izgaraya Ait Olduğu Tablosu, Zaman Değişkenli Kalman Filtresi (10*10).....	42
Tablo 8. Hangi Aracın Hangi Izgaraya Ait Olduğu Tablosu, Zaman Değişkenli Kalman Filtresi (20*20).....	43
Tablo 9. Farklı Modellerdeki Tespit Süreleri.....	46
Tablo 10. Video ve Gerçek Zamanlı Modellerin Doğruluk Oranları.....	49
Tablo 11. Uzak ve Yakın Çarpışmalardaki Performanslar	50
Tablo 12. Farklı Modellerin Performans Parametreleri	51

ŞEKİLLER LİSTESİ

	Sayfa No.
Şekil 1. Sensör Verilerinin Toplanması ve Karar Sürecine Geçiş Akışı	4
Şekil 2. CNN Mimarisi	19
Şekil 3. Markov Zinciri Modeli	25
Şekil 4. Modelin Akış Şeması.....	30
Şekil 5. Modelin Sözde Kodu	31
Şekil 6. Modelin Görseli	32
Şekil 7. Veri Seti Örneği	32
Şekil 8. Veri Setindeki Görsellerden Örnekler	33
Şekil 9. Kalman Filtresinde Yapılan Değişiklikler	34
Şekil 10. Izgara Modelinin Koordinatları	37
Şekil 11. Farklı Kalman Filtreleri ile Elde Edilen Doğruluk	45
Şekil 12. Gerçek Zamanlı Deneyin Görseli	49

KISALTMALAR

ANN	: Artificial Neural Network (Yapay Sinir Ağları)
AI	: Artificial Intelligence (Yapay Zeka)
AutoML	: Automatred Machine Learning (Otomatik Makine Öğrenimi)
CARLA	: Car Learning to Act (Carla sürücü Simulasyonu)
CNN	: Convonutional Neural Network (Evrişimsel Sinir Ağları)
GAN	: Generative Adversarial Networks (Çekişmeli Üretici Ağlar)
GPS	: Global Positioning System (Küresel Konumlama Sistemi)
HMM	: Hidden Markov Model (Saklı markov Zinciri)
HOG	: Histogram of Oriented Gradients (Gradyanların Histogramı)
IoT	: Internet of Things (Nesnelerin İnterneti)
ITS	: Intelligent Transportation Systems (Akıllı Taşıma Sistemleri)
LBP	: Local Binary Patterns (Yerel İkili Yollar)
LIDAR	: Light Detection and Ranging (Radar, ışık sensörleri)
LSTM	: Long Shorth Term Memory (Uzun-Kısa Süreli Bellek)
mAP	: mean Average Precision (Ortalama Keskinlik)
NHTSA	: National Highway Traffic Safety Administration (Uluslararası Ototban Güvenliği Merkezi)
PSNR	: Peak Signal to Noise Ratio (Yüksek sinyal Gürültü Oranı)
R-CNN	: Region Based Convonutional Neural Network (Bölge Tabanlı Evrişimli Sinir Ağları)
RGB	: Red, Green, Blue Format (Kırmızı Yeşil Mavi Formatı)
RNN	: Recurrent Neural Network (Evrişimli Sinir Ağları)
RoI	: Region of Interest (İlgilenilen Bölge)

SSD	: Single Shot MultiBox Detector (Tek Zamanlı Çok Kutulu Tespit Edici)
SVM	: Support Vector Machines (Destek Vektör Makineleri)
vb.	: ve benzeri
vd.	: ve diğerleri
ViT	: Vision Transformers (Görüntü Dönüştürücü)
WHO	: World Health Organization (Dünya Sağlık Örgütü)
YOLO	: You Only Look Once (Sadece Bir Kere Bak)



SÖZLÜK

Bölge Önerme: Bir görüntüde olası nesne içeren bölgelerin otomatik önerilmesi işlemidir.

Genellikle nesne tespiti ve görüntü üzerinde çalışılan diğer sınıflandırma görevlerinde ön adım olarak kullanılır. Hesaplama yükünü azaltır ve tespit doğruluğunu artırır.

Özellik Çıkartma: Özellik çıkarma (feature extraction), verilerden anlamlı özelliklerin belirlenmesi ve bu özelliklerin kullanılarak modelleme yapılabilmesi sürecidir. Genellikle makine öğrenimi ve veri madenciliği alanlarında, verilerin daha etkili bir şekilde analiz edilmesi amacıyla kullanılır. Özellik çıkarma, özellikle karmaşık ve yüksek boyutlu veri setlerinde kullanılır.

GİRİŞ

Otonom araçlar, geleceğin ulaşımını şekillendirme potansiyeli taşımakla birlikte, yaygınlaşmalarının önünde aşılması gereken önemli engeller bulunmaktadır. Bu engellerin başında yasal ve düzenleyici belirsizlikler gelmektedir. Olası kaza durumlarında hukuki sorumluluğun nasıl dağıtılacağı ve araçların güvenlik standartları, net bir yasal çerçeveye oturtulmalıdır. Teknolojinin toplumsal kabulü için bir diğer kritik eşik ise kamu güvenidir. Halkın bu sistemlere güven duyması, otonom araçların benimsenmesinde kilit rol oynayacaktır.

Aynı zamanda, veri güvenliği ve gizliliği de hayati önem taşımaktadır. Otonom araçların topladığı ve işlediği devasa miktardaki kişisel ve operasyonel verinin siber saldırılara karşı korunması ve mahremiyetinin güvence altına alınması şarttır. Son olarak, mevcut altyapının bu teknolojiye entegrasyonu da önemli bir zorluktur. Yolların, sinyalizasyon sistemlerinin ve genel trafik altyapısının otonom araçların verimli ve güvenli bir şekilde çalışmasını destekleyecek şekilde modernize edilmesi gerekmektedir.

Sonuç olarak, yapay zekâ destekli otonom araçlar, ulaşımın geleceği için devrim niteliğinde bir vizyon sunmaktadır. Trafik kazalarını azaltma, enerji verimliliğini artırma ve yolcu konforunu en üst düzeye çıkarma gibi vaatleri, bu teknolojiyi son derece cazip kılmaktadır. Bununla birlikte, bu potansiyelin tam olarak hayata geçirilebilmesi, az önce bahsedilen yasal düzenlemeler, veri güvenliği ve altyapı uyumluluğu gibi temel zorlukların başarıyla yönetilmesine bağlıdır. Bu engeller aşıldığında, otonom araçların yaygınlaşmasıyla birlikte ulaşım sektöründe köklü bir dönüşüm yaşanması ve çok daha güvenli, verimli ve konforlu bir gelecek inşa edilmesi beklenmektedir.

1. OTONOM ARAÇLARIN TANIMI VE TARİHÇESİ

Otonom araçlar, insan müdahalesine gerek kalmadan çevrelerini algılayarak hareket edebilen, yapay zekâ destekli, sensör ve veri toplama teknolojileriyle donatılmış sistemlerdir. Günümüz teknolojisinde otonom araçlar, ulaşım sektöründe güvenlik, verimlilik, çevresel etki ve toplumsal kullanım açısından önemli gelişmeler sunmakta ve geleceğin mobilite anlayışını köklü bir biçimde değiştirme potansiyeline sahiptir (Galvao vd., 2021). Bu tezde, otonom araçların teknolojik altyapısı, uygulama örnekleri, sunduğu avantajlar, karşılaştığı zorluklar ve geleceğe yönelik beklentiler detaylı bir biçimde ele alınacaktır.

1.1. Otonom Araçların Tanımı ve Tarihçesi

İlk otonom araç denemeleri, DARPA'nın 2004 yarışması ile gündeme gelmiş ve bu süreç, teknolojinin günümüzde Google, Tesla, Waymo gibi büyük teknoloji ve otomobil firmaları tarafından devrim niteliğinde geliştirilmesiyle devam etmiştir (Beiker, 2012).

Otonom teknolojiler, günümüzde yalnızca araçlar alanında değil, tarım, savunma, endüstriyel otomasyon gibi pek çok alanda da kullanılmaya başlanmıştır. Bu teknolojilerin gelişmesinde yapay zekâ, makine öğrenimi, sensör teknolojileri (LİDAR, radar, ultrasonik sensörler) ve GPS gibi sistemler temel rol oynamaktadır. Otonom sistemlerin evrimi, insan faktörüne bağlı hataları azaltma ve sistematik, analiz edilebilir veri akışı sayesinde güvenliği artırmayı hedeflemektedir.

Otonom araçların performansı ve güvenliği, bünyelerinde kullanılan teknolojik bileşenlere doğrudan bağlıdır. Bu bileşenler arasında sensör sistemleri, yapay zekâ algoritmaları, veri toplama ve işleme teknolojileri, bulut bilişim ve “edge computing” gibi altyapılar yer almaktadır.

Otonom araçların çevre algılaması için kullandığı başlıca sensörler şunlardır:

LİDAR: Işık dalgaları kullanarak nesnelerin mesafesini ve boyutunu tespit eder.

Radar: Hız ve mesafe ölçümü sağlayarak hareket halindeki nesneleri izler.

Kameralar: Trafik işaretleri, şeritler, yol durumu ve çevresel koşulları analiz eder.

Ultrasonik Sensörler: Yakın mesafedeki objeleri algılar ve park yardımı gibi uygulamalarda kullanılır.

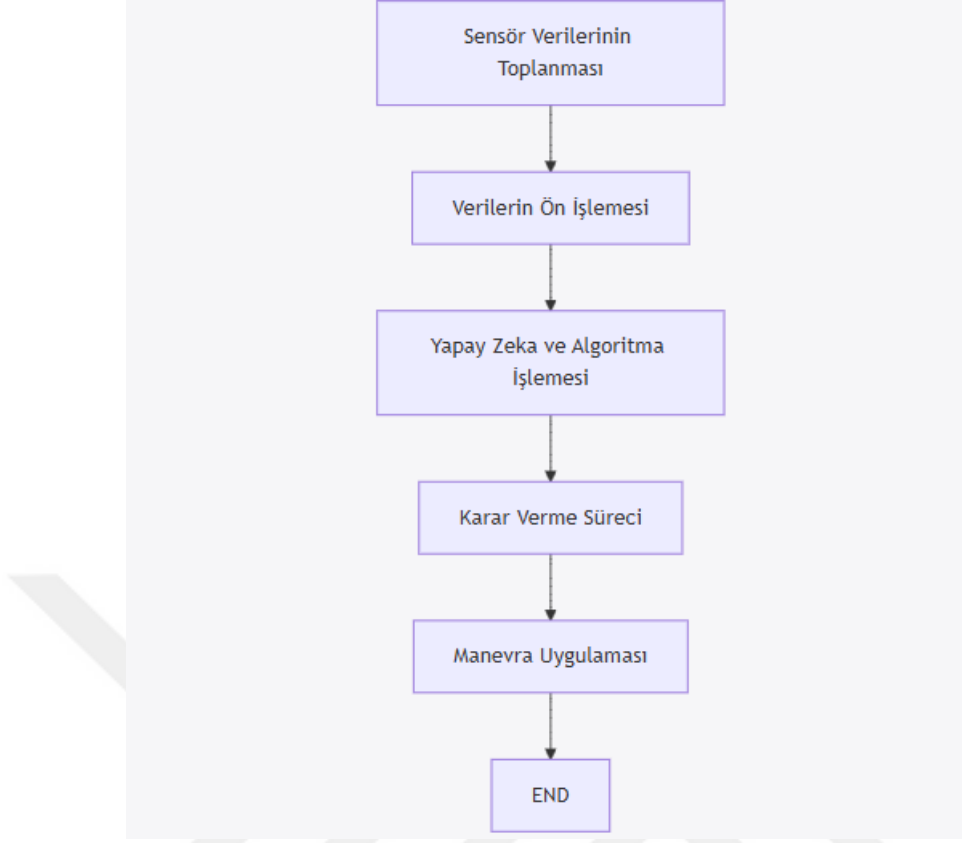
Tablo 1. Otonom Araçlarda Kullanılan Sensör Türleri

Sensör Türü	Kullanım Amacı	Temel Özellikler
LİDAR	Nesne mesafesi ve boyut takibi	Işık dalgaları ile yüksek hassasiyet
Radar	Hız ve mesafe ölçümü	Düşük maliyet, tüm hava koşullarında çalışır
Kameralar	Görsel bilgi ve durum takibi	Yüksek çözünürlük, trafik işaretlerini tanıma
Ultrasonik Sensör	Yakın mesafe tespiti	Park yardımı, engel algılama

Kaynak: Yazar tarafından oluşturulmuştur.

Araçların topladığı çok miktarda veriyi gerçek zamanlı işleyip analiz edebilmesi, yapay zekâ algoritmaları sayesinde mümkündür. Bu algoritmalar, çevreden gelen verileri birleştirir, daha önce yaşanan durumları analiz eder ve gelecekteki davranışları tahmin eder. Örneğin, bir aracın sensörleri, yayaları, diğer araçları veya trafik işaretlerini algılar. Yapay zekâ, bu verileri analiz ederek olası senaryoları değerlendirir (örneğin, yaya geçişi, şerit değişikliği) ve karar verir (Beiker, 2012).

Aşağıdaki akış diyagramı, bir otonom aracın sensör verilerini işleme sürecini ortaya koymaktadır:



Şekil 1. Sensör Verilerinin Toplanmasından Karar Sürecine Geçiş Akışı

Kaynak: Yazar tarafından oluşturulmuştur.

1.2. Otonom Araçların Sağladığı Avantajlar ve Toplumsal Etkiler

Otonom araçlar, sadece teknolojik bir yenilik olmanın ötesinde, toplumsal düzen, çevresel etkiler ve ekonomik faydalar açısından da önemli avantajlar sunmaktadır.

Mevcut veriler, trafikte meydana gelen kazaların yaklaşık %90'ının insan hatalarından kaynaklandığını göstermektedir (Ranney vd., 2001).

. Otonom araçların, algoritmik karar verme ve gerçek zamanlı veri analizleri sayesinde bu oranı önemli ölçüde azaltması beklenmektedir. Yapay zekâ destekli sistemler, sürücü hatalarını minimize ederek daha güvenli bir sürüş deneyimi sağlayabilir (Zhao vd., 2024).

Otonom araçların sunduğu en önemli avantajlardan biri, yaşlılar, engelliler ve ehliyetsiz bireyler için mobilitenin artırılmasıdır. Bu teknoloji, toplumun her kesiminin

ulařım hizmetlerine eřit erişim sağlayabilmesini mümkün kılmaktadır (Zhao vd., 2024).

İlave olarak, toplumda trafik yoğunluğunun azaltılması, toplu taşıma sistemlerinin verimliliğinin artırılması ve şehir içi ulaşım maliyetlerinin düşürülmesi gibi faydalar da gözlemlenmektedir.

Otonom araçların güvenliğı, sensörlerin doğruluğı, yapay zekâ algoritmalarının geliştirilmesi ve gerçek zamanlı veri işleme kapasitesine bağıdır. Geleneksel IT altyapılarının yetersiz kalması, “edge computing” ve bulut bilişim teknolojilerinin entegrasyonunu şart kořmaktadır (Alam vd., 2023).

Ayrıca, tamamen otonom sürüş (seviye 5) için altyapıda ve yasal düzenlemelerde yapılacak köklü değışiklikler gerekmektedir. Altyapının geliştirilmesi, trafik yönetiminin yanı sıra araçlar arası iletişim (V2X) teknolojilerinin de optimize edilmesini zorunlu kılmaktadır.

Otonom araçların veri toplama ve iletim süreçleri, siber saldırılara açığı beraberinde getirmektedir. Araçların “hacklenmesi”, sistemin çökmesi veya verilerin kötüye kullanılması gibi riskler, siber güvenlik önlemlerinin keskin bir şekilde artırılmasını gerektirmektedir (Jiang vd., 2020).

Siber güvenliğin sağlanması, sadece araç üreticileri için değıl, aynı zamanda altyapı ve veri merkezleri için de kritik bir konudur. Otonom araçlar, teknolojik gelişmelerin hızlanmasıyla birlikte gelecekte daha yaygın ve güvenilir hale gelecektir. Mevcut durumda kısmi otonom (seviye 2-3) özelliklere sahip araçlar, tam otonom sürüş (seviye 5) için altyapı, regölasyon ve teknolojik iyileştirmeler beklenmektedir (Anderson vd., 2014).

Bunun yanı sıra, yapay zekâ algoritmalarının sürekli öğrenmesi ve çevresel verilerden elde edilen geri bildirimlerle sistemlerin gelişmesi beklenmektedir. Bu durum hem sürüş güvenliğini hem de sistem verimliliğini artıracaktır.

Otonom araçlar, geleceğın ulaşım teknolojilerinin temel taşlarından biridir. Hem teknolojik altyapı hem de toplumsal etkileri açısından değılendirildiğinde; Teknolojik altyapıda sensör, yapay zekâ, veri işleme ve iletişim sistemlerinin entegrasyonu büyük önem taşımaktadır.

Otonom araçlar, insan hatasından kaynaklanan kazaları azaltarak güvenliğini artırmakta, aynı zamanda yakıt verimliliği ve çevresel faydalar sağlamaktadır. Toplumsal olarak, engelli, yaşlı ve ehliyetsiz bireyler için mobilitayı kolaylaştırmak ve trafik yoğunluğunu azaltmak gibi önemli avantajlar sunmaktadır. Bununla birlikte, teknik zorluklar, siber güvenlik riskleri ve hukuki-etik sorunlar, bu teknolojinin yaygınlaşmasında aşılması gereken önemli engellerdendir. Geleceğe yönelik beklentiler, tam otonom sürüş seviyesine ulaşma, ileri veri işleme teknolojilerinin benimsenmesi ve şehir içi ulaşımın optimize edilmesi üzerine odaklanmaktadır.

Sonuç olarak, otonom araçlar günümüz ulaşım teknolojisinin evriminde belirleyici bir rol oynamakta ve hem bireysel hem de toplumsal düzeyde büyük potansiyel taşımaktadır. Güvenlik, verimlilik, çevresel sürdürülebilirlik ve erişilebilirlik gibi alanlarda sunduğu yenilikler, geleceğin mobilite anlayışını yeniden tanımlayacaktır.

1.3. Nesne Algılamada Kullanılan Teknikler

Nesne algılama, sahne anlayışında yerleştirme (nesnelerin bir görüntüdeki konumunu belirleme) ve sınıflandırma (nesnelerin kimliğini belirleme) süreçlerini kapsayan önemli bir aşamadır. Geleneksel yöntemler, büyük ölçüde önceden belirlenmiş özellikler ve sezgisel yaklaşımlar üzerine inşa edilirken, son gelişmeler derin öğrenmeye dayalı çözümler daha üstün sonuçlar sunmaktadır.

Bu bağlamda nesne tespiti, araçların, yayaların, trafik işaretlerinin ve diğer önemli çevresel unsurların konumlarını ve türlerini belirleme sürecidir. Bu bölümde, 2D ve 3D nesne tespitinde kullanılan temel yöntemler, bu yöntemlerin otonom sürüşe özgü zorlukları nasıl ele aldığı ve performanslarını etkileyen unsurlar incelenecektir.

1.3.1. İki Boyutlu Nesne Tespit Yöntemleri

Otonom araçlarda 2D nesne tespiti için kullanılan tek aşamalı ve iki aşamalı dedektörler, mimari tasarım, verimlilik ve doğruluk bakımından belirgin farklılıklar arz eder.

Tek aşamalı dedektörler, hızları ve basit tasarımlarıyla dikkat çeker. Bu dedektörler, görüntüden nesne tespitini doğrudan ve tek bir adımda gerçekleştirerek gerçek zamanlı uygulamalar için önemli avantajlar sunar (Carranza-Garcia vd., 2021; Soviany ve Tudor Ionescu, 2018). Hesaplama kaynaklarının kısıtlı olduğu ortamlarda hızlı işlem yapabilmeleri nedeniyle otonom araçlarda sıklıkla tercih edilmektedirler (Liang vd., 2022). Ancak, bu hız avantajı, doğruluk açısından bazı tavizler verilmesine yol açabilir ve özellikle küçük veya gizli nesnelerin tespitinde iki aşamalı detektörlerin gerisinde kalabilirler (Lu vd., 2020).

İki aşamalı dedektörler, daha karmaşık bir işlem sürecine sahip olmaları dolayısıyla yüksek doğruluk sağlar. Bu dedektörler, nesne tespitinde öncelikle aday bölgeleri tanımlar, ardından bu bölgeleri kapsamlı bir şekilde analiz ederek daha hassas konumlandırma ve sınıflandırma yapar (Cao vd., 2020; Soviany ve Tudor Ionescu, 2018). Ancak, iki aşamalı dedektörler genellikle daha yavaş performans sergiler ve gerçek zamanlı uygulamalar için ilave optimizasyonlar talep edebilir. Sonuç olarak, tek aşamalı dedektörler hız ve verimlilik, iki aşamalı dedektörler ise doğruluk ve hassasiyet açısından avantaj sunar; seçim, uygulamanın önceliklerine göre gerçekleştirilmelidir (Carranza-Garcia vd., 2021; Zhang ve Berger, 2023).

1.3.2. Üç Boyutlu Nesne Algılama

LİDAR ve kamera tabanlı 3D nesne tespit yöntemleri, otonom araçlarda makine öğrenimi açısından yaklaşım, veri temsili ve ele aldıkları zorluklar bakımından belirgin farklılıklar arz etmektedir. LİDAR tabanlı yöntemler, esasen hassas derinlik verisi elde etmeye odaklanırken, kamera tabanlı yöntemler zengin anlamsal detayları yakalama konusunda daha üstündür.

LiDAR tabanlı 3D nesne tespitinde, sensörlerden elde edilen verilerle yüksek doğrulukta derinlik bilgisi temin edilir. LİDAR, lazer darbeleri gönderip geri dönen sinyalleri ölçerek ayrıntılı 3D nokta bulutları oluşturur ve bu sayede nesnelerin üç boyutlu konumları hassas bir şekilde tespit edilir (Mohapatra vd., 2025). Ancak, LİDAR verisi doğası gereği seyrek ve düzensizdir; bu nedenle veri işleme aşamasında gürültü ve nokta bulutu temsili gibi tekniklere ihtiyaç duyulmaktadır (Wen ve Jo, 2021). Ek olarak, LİDAR tabanlı yöntemler genellikle kamera tabanlı yöntemlerden

farklı 3D koordinat sistemleri kullanır; bu durum, iki farklı sensör verisinin entegrasyonu sırasında ilave karmaşıklık oluşturabilir. Otonom sürüşte gerçek zamanlı işleme kritik bir öneme sahip olduğundan, LİDAR tabanlı sistemlerde çoklu görev öğrenimi ve verimli ağ mimarileri ile düşük gecikme sağlanması hedeflenmektedir (Mohapatra vd., 2025).

Ancak, kameralar doğrudan derinlik bilgisi sunmadığından, derinlik tahmini önemli bir zorluk teşkil etmektedir. Bu amaçla genellikle stereo görüş veya monoküler derinlik tahmin teknikleri kullanılmaktadır (Garnett vd., 2017). Ek olarak, kamera tabanlı yöntemler aydınlatma koşullarından etkilenmektedir; düşük ışık seviyelerinde performans düşüklüğü gözlemlenebilir (Rashed vd., 2019).

1.3.3. Yapay Zeka Yöntemleri

Otonom araçların çevrelerini yalnızca nesne düzeyinde değil, aynı zamanda piksel düzeyinde ayrıntılı bir şekilde analiz edebilmesi, güvenli ve bağlama duyarlı kararlar alabilmeleri açısından kritik öneme sahiptir. Bu bölümde, otonom araçların çevrelerini kapsamlı bir şekilde analiz etmelerini sağlayan segmentasyon yöntemleri, aralarındaki temel farklılıklar ve her birinin sunduğu avantajlar incelenecektir.

Otonom araçlar için makine öğrenimi bağlamında, semantik segmentasyon ve örnek segmentasyonu, farklı yöntemler ve amaçlar taşıyan iki önemli görevdir.

Semantik segmentasyon yöntemlerinin temel amacı, her pikselin yol, araç veya yaya gibi bir semantik kategoriye tahsis edilmesidir. Bu genellikle, her sınıfa özgü desen ve özellikleri tanımak amacıyla eğitilmiş evrişimli sinir ağları (CNN) aracılığıyla gerçekleştirilir (Feng vd., 2020). Bazı yaklaşımlar, otonom sürüşte kritik öneme sahip yaya ve araç gibi nesnelere daha fazla ağırlık vermek için "importance-aware loss" gibi özel kayıp fonksiyonları kullanmaktadır (Y. Zhang vd., 2019). Ayrıca, karmaşık ortamlarda segmentasyonun dayanıklılığını ve doğruluğunu artırmak amacıyla "karşılıklı bilgi maksimize etme" gibi çoklu sensör verilerinin entegrasyonunu sağlayan teknikler de kullanılmaktadır (Yang vd., 2024).

Örnek segmentasyonu, aynı sınıfa mensup farklı nesneleri ayırt etme yeteneğiyle belirginleşir. Bu amaçla, ağırlıkla çıkırtısını kolayca kümelenebilir hale getiren ayırt edici kayıp fonksiyonları kullanılmakta ve bu yöntemler genellikle nesne

önerilerine ihtiyaç duymadan gerçek zamanlı uygulamalarda etkilidir (De Brabandere vd., 2017). Belirli yaklaşımlar, dikkat mekanizmasına sahip tekrarlayan sinir ağları (RNN) kullanarak ilgi bölgelerini sıralı bir şekilde üretir, her bir bölgede baskın nesneleri segmentler ve insan benzeri bir sayma sürecini taklit eder (Ren ve Zemel, 2017). Yöntemler, MaskPLS gibi, örtüşmeyen ikili maskeleri ve semantik sınıfları uçtan uca tahmin ederek, sonradan yapılan kümeleme gibi ek adımlara ihtiyaç duymaz (Marcuzzi vd., 2023).

Son yıllarda geliştirilen Sem2Ins modeli, koşullu üretici adversaryal ağlar (cGAN) ve derin denetim kullanarak semantik segmentasyon çıktılarından örnek sınırları üretmekte ve böylece iki yöntem arasındaki boşluğu kapatmaktadır. Bu model, hızlı semantik segmentasyon teknikleriyle entegre edildiğinde gerçek zamanlı performans sergileyebilmektedir (Yin vd., 2021). QuadroNet gibi çok görevli öğrenme yaklaşımları, örnek ve semantik segmentasyon gibi çeşitli görevleri tek bir ağda entegre ederek, görevler arasında özellik paylaşımı aracılığıyla verimliliği ve performansı artırmaktadır (Goel vd., 2021).

1.3.4. Sensor Fusion Tekniği

Otonom araçlar, çevrelerini kapsamlı ve güvenilir bir biçimde algılamak amacıyla genellikle çeşitli sensör türlerini (kamera, LİDAR, radar vb.) bir arada kullanır. Her sensörün kendine has avantajları ve kısıtlamaları bulunduğundan, bu çeşitli sensörlerden elde edilen verilerin akıllıca entegrasyonu, yani sensör füzyonu, algı sisteminin genel performansını ve dayanıklılığını artırmada hayati bir öneme sahiptir. Makine öğrenimi, çeşitli veri türlerini birleştirerek ve bu verilerden anlamlı bilgiler elde ederek sensör füzyonu süreçlerini optimal etmede etkili bir araçtır. Bu bölümde, otonom araçlarda yaygın olarak kullanılan radar-kamera ve LİDAR-kamera füzyon teknikleri başta olmak üzere, sensör füzyonunun temel motivasyonları, çeşitli füzyon seviyeleri ve bu alandaki güncel yaklaşımlar ele alınacaktır.

Otonom araçların algı yeteneklerinin geliştirilmesinde sensör füzyon teknikleri, özellikle radar-kamera ve LİDAR-kamera füzyonu, hayati bir öneme sahiptir. Bu teknikler, çeşitli sensörlerin tamamlayıcı güçlü yönlerinden faydalanarak nesne tespiti, takibi ve çevresel algının doğruluğunu artırır; bu da güvenli ve verimli

otonom sürüş için kritik öneme sahiptir. Makine öğrenimi, bu füzyon süreçlerinde daha karmaşık veri entegrasyonu ve analiz imkânı sağlayarak mühim bir rol oynamaktadır. Sensör füzyonunun birincil amacı, tek bir sensör tarafından elde edilebilen doğruluk ve güvenilirliği aşmaktır. Füzyon işlemi, veri düzeyinde (ham verilerin entegrasyonu), özellik düzeyinde (farklı sensörlerden elde edilen özelliklerin entegrasyonu) veya karar düzeyinde (her bir sensörün tespit veya sınıflandırma sonuçlarının entegrasyonu) gerçekleştirilebilir.

Radar-kamera füzyonunda, her iki sensörün tamamlayıcı faydaları vurgulanmaktadır. Radar ve kamera verilerinin entegrasyonunda, özellik düzeyinde füzyon ve temsil öğrenimi gibi yöntemler kullanılmakta; radar verisinin görüntü düzlemine yansıtılması veya özellikler arasındaki etkileşimlerin modellenmesiyle veri ilişkisi güçlendirilmektedir (Dong vd., 2021; Kurniawan ve Trilaksono, 2023). Bu sorunların çözümü için derin temsil öğrenimi ve radar nokta bulutlarının kümeleme yöntemleri önerilmiş ve tespit doğruluğunda kayda değer iyileşmeler elde edilmiştir.

LİDAR-kamera füzyonunda, LİDAR'ın sağladığı hassas üç boyutlu mekânsal bilgi, kameranın sunduğu doku ve renk verisiyle bütünleştirilir (Cui vd., 2021). Erken füzyon (ham verilerin birleştirilmesi) ve geç füzyon (işlenmiş verilerin birleştirilmesi) gibi çeşitli stratejiler, LİDAR ve kamera verilerinin entegrasyonunda uygulanmaktadır. Ancak, sürüş koşullarına uyum sağlayabilen esnek bir füzyon yöntemi, sistemin dayanıklılığını ve verimliliğini artırabilir (Ignatious vd., 2023). Derin öğrenme temelli modeller, örneğin Bayesian sinir ağları, tespit doğruluğunu artırmak ve çeşitli sürüş koşullarındaki belirsizlikleri azaltmak amacıyla kullanılmaktadır; bu modeller, LİDAR ve kamera verisinin avantajlarından birlikte faydalanır (Ravindran vd., 2022).

Çoklu sensör füzyonunda, radar, LİDAR ve kamera verilerini entegre eden kapsamlı çerçeveler (örneğin, EZFusion ve hibrit çoklu sensör yapıları) geliştirilmiştir. Bu sistemler, gerçek zamanlı uygulamalar için maliyet etkin ve verimli çözümler sunarak 3B nesne tespiti ve takibinde yüksek başarı elde etmektedir (Jahromi vd., 2019).

Bu sensörlerin entegrasyonunda, gömülü sistemlerin kısıtları dikkate alınarak doğruluk ile işlem maliyeti arasında bir denge sağlanmalıdır. Ek olarak, çevresel

koşullara dinamik bir şekilde uyum sağlayabilen adaptif füzyon stratejilerinin geliştirilmesi, otonom sürüş teknolojisinin ilerlemesi açısından hayati öneme sahiptir. Bu unsurlar, sensör füzyonu metodolojilerinde sürekli araştırma ve yenilik gereksinimini vurgulamaktadır.

1.3.5. Davranışsal Tahminin Sürüşteki Uygulamaları

Otonom araçların trafikte güvenli ve etkili bir şekilde hareket edebilmesi, çevrelerini algılamanın yanı sıra diğer yol kullanıcılarının (yayalar, araçlar vb.) gelecekteki davranışlarını da tahmin edebilme yeteneğine bağlıdır. Davranışsal tahmin, otonom sistemlerin proaktif kararlar almasını, potansiyel tehlikelerden kaçınmasını ve insanlarla daha uyumlu etkileşimde bulunmasını temin eder. Makine öğrenimi, özellikle derin öğrenme ve pekiştirmeli öğrenme yöntemleri, karmaşık trafik senaryolarında yaya ve araç davranışlarını modelleme ve tahmin etme konusunda etkili araçlar sağlamaktadır. Bu bölümde, otonom sürüşte yaya ve araç davranışlarının tahminine yönelik makine öğrenimi temelli yöntemler ve bu yöntemlerin karşılaştığı zorluklar incelenecektir.

1.3.6. Yaya Davranışlarının Öngörüsü

Otonom araçların güvenli bir şekilde işletilebilmesi için yaya davranışlarının öngörülmesi hayati bir unsurdur. Makine öğrenimi, özellikle derin öğrenme, bu zorluğun üstesinden gelmek için çeşitli veri girdilerini analiz ederek yaya niyetlerini ve hareket rotalarını tahmin etmede etkili bir araç olarak öne çıkmaktadır. Bu yetenek, otonom araçların yaya hareketlerini önceden tahmin edip tepki verebilmesi bakımından yol güvenliğini ve trafik verimliliğini artırmak için elzemdir.

Görsel tabanlı sistemler, görüntü verilerini kullanarak yaya davranışını öngörmektedir. Bu alanda öne çıkan yaklaşımlardan biri, yaya konumlarını tespit eden, vücut pozlarını tahmin eden ve karşıdan karşıya geçme niyetlerini belirleyen konvolüsyonel çoklu görevli alt-üst modeldir (Razali vd., 2021). Diğer bir yöntem, video verisi aracılığıyla gelecekteki karelerin tahmin edilmesi yoluyla yaya geçiş davranışını öngörmektir (Chaabane vd., 2020).

Yaya hareket güzergâhı verileri, makine öğrenimi modelleriyle entegre edilerek sinyalsiz yaya geçitlerinde bekle-geç kararlarının öngörülmesinde kullanılmaktadır. Özellikle “gradient boosting” karar ağaçları gibi modeller, tarihsel verileri kullanarak hareket özelliklerini analiz eder ve otonom araçlar için yaya davranışlarının tahminine yönelik bir çerçeve sağlar (Yang vd., 2024). Bağlamsal modeller, araç-yaya etkileşimini dikkate alarak, geleneksel LSTM modellerine kıyasla daha yüksek doğruluk ve zamanlama başarısı sunmaktadır (Neogi vd., 2020).

“Encoder/decoder” ağları ve uzamsal-zamansal ağlar gibi modeller, bireysel özellikler ve çevresel etkileşimler gibi çeşitli unsurları analiz ederek yaya davranışını tahmin edebilmektedir (Chaabane vd., 2020). Özellikle, öz denetimli sürekli öğrenme çerçeveleri, çevrimiçi yaya verileriyle modelleri sürekli güncelleyerek unutma sorununu azaltmakta ve daha önce karşılaşılmamış senaryolarda performansı artırmaktadır (Knoedler vd., 2022).

Çoklu ve bağlamsal modeller de ön plana çıkmaktadır. Örneğin, “Context-CVGN” gibi çok modlu güzergâh tahmin ağları, sahne semantik modellemesi aracılığıyla çevresel ve etkileşimsel unsurları tespit ederek çeşitli olası yaya güzergâhlarını yüksek doğrulukla tahmin edebilmektedir (Yang vd., 2024). Örneğin, modeller genellikle RMSE gibi nicel ölçütlere odaklanmakta, ancak bu ölçütler insan benzeri davranışları tam olarak yansıtamayabilir. Bu zorluklar, otonom sürüş sistemlerinin güvenliğini ve güvenilirliğini artırmak amacıyla sürekli araştırma ve geliştirme yapılmasının önemini vurgulamaktadır.

1.3.7. Öngörücü Araç Davranışları

Otonom araçlarda makine öğrenimi ile araç davranışının tahmin edilmesi, güvenli ve verimli bir seyir için hayati bir unsurdur. Bu süreç, çevredeki araçların mevcut ve geçmiş gözlemlerine dayanarak gelecekteki durumlarının tahmin edilmesini ve otonom aracın potansiyel tehlikelere karşı farkındalığını artırmayı içerir. Özellikle derin öğrenme, makine öğrenimi alanında geleneksel yöntemlere nazaran karmaşık ortamlarda üstün performans sergileyerek etkili bir araç olarak öne çıkmaktadır. Çeşitli tekniklerin entegrasyonu, derin öğrenme, pekiştirmeli öğrenme ve

otomatik makine öğrenimi (AutoML) gibi, araç davranışı tahmininin doğruluğunu ve güvenilirliğini artırmak amacıyla incelenmektedir.

AutoML teknikleri, otonom araçlarda risk analizi ve davranış değerlendirme süreçlerini otomatikleştirmek amacıyla tasarlanmıştır. Bu yöntemler, denetimsiz risk tespiti, özellik öğrenimi ve modelin otomatik ayarlanmasını birleştirerek, yüksek doğrulukla risk seviyeleri ve davranış özellikleri tahmini gerçekleştirebilmektedir. Bir AutoML yaklaşımıyla gerçekleştirilen çalışmada, sensör konfigürasyonları gibi otonom araç tasarımına ilişkin önemli görümler sunulmaktadır (Shi vd., 2020).

Derin öğrenme temelli yöntemler, karmaşık sürüş senaryolarını ve uzun vadeli tahminleri işleme yetenekleri nedeniyle artan bir ilgi toplamaktadır. Bu yöntemlerde genellikle konvolüsyonel sinir ağları (CNN) ve uzun-kısa süreli bellek (LSTM) ağları aracılığıyla araç davranışının uzamsal ve zamansal özellikleri tespit edilmektedir (Mozaffari vd., 2020). Bir modelde CNN ve LSTM'nin hibrit kullanımıyla uzun vadeli araç davranışları tahmin edilmiştir (Ignatious vd., 2023). Diğer bir yaklaşımda, yüksek çözünürlüklü semantik haritalar ve hava görüntüleri kullanılarak gelecekteki güzergâhlar tahmin edilmiş ve uzamsal-zamansal öğrenmenin doğruluk üzerindeki olumlu etkisi kanıtlanmıştır (Rezaei vd., 2023).

Hiyerarşik çerçeveler, trafik ışığı bilgilerini ve araçlar arası etkileşimleri modele entegre ederek, Bayes ağları ve pekiştirmeli öğrenme ile ayrık niyet ve sürekli güzergâh tahmini gerçekleştirmekte ve çeşitli senaryolarda düşük ortalama sapma hataları elde etmektedir (Yang vd., 2023). Olasılıksal modeller, büyük veri setleriyle insan benzeri öngörü yeteneklerini taklit ederek, sürüş manevralarını yüksek doğrulukla sınıflandırmakta ve yanal pozisyonları minimal hata ile tahmin edebilmektedir (Wirthmüller vd., 2020).

Makine öğrenimi, otonom araçların davranış tahmininde kayda değer ilerlemeler elde etmiş olsa da bu modellerin kısıtlamaları ve karşılaşılan zorluklar dikkate alınmalıdır. Ek olarak, Nesnelerin İnterneti (IoT) ve gelişmiş sensör teknolojilerinin entegrasyonu, otonom araçların yeteneklerini artırmak için yeni fırsatlar sunmakta ve akıllı mobilite çözümleri açısından umut verici bir gelecek vaat etmektedir.

2. LİTERATÜR ÇALIŞMASI

Kazalar, tıkanıklık veya düzensiz desenler dahil olmak üzere trafik anomalileri, ulaşım sistemlerinin verimliliği, güvenliği ve güvenilirliği açısından önemli zorluklar yaratır. Bu düzensizliklerin zamanında tespiti, verimli trafik yönetimi ve yol ağlarında kesintisiz bir akışın sürdürülmesi için önemlidir. Bu literatür özeti, trafik anomalisi tespitinde kullanılan çeşitli yaklaşımları ve teknolojileri inceleyerek bu alandaki gelişmeler, sorunlar ve gelecekteki yönler hakkında içgörüler sunmaktadır.

“Haar kaskadları” ve HOG (Yönlendirilmiş Gradyanların Histogramı) gibi geleneksel nesne tespit teknikleri, derin öğrenmenin ortaya çıkmasından önce daha büyük öneme sahipti (Dalal ve Triggs, 2005; Galvao vd., 2021). Bununla birlikte, bu metodolojiler dinamik trafik ortamlarında ölçek, tıkanıklık ve aydınlatmadaki değişiklikleri ele almada kısıtlamalarla karşılaşmıştır. Evrişimli sinir ağlarının (CNN'ler) ortaya çıkışı, nesne tespiti alanını kökten değiştirmiştir. İstatistiksel yöntemler, trafik anomalilerini tespit etmede önemlidir ve netlik ve kolay anlaşılabilirlik sağlamıştır. Ghahremannezhad vd. 2023, trafik akışı verilerindeki anomalileri belirlemek için Gauss süreçlerine dayanan olasılıksal bir model kullanmıştır. Metodolojileri anormal trafik modellerini önemli bir hassasiyetle doğru bir şekilde tespit etmiştir.

Girshick ve arkadaşları (2014), PASCAL VOC veri kümesinde %58,5 doğruluk elde eden bir R-CNN modeli önermektedir. Ancak bu ilerlemeler, sırasıyla %70,0 ve %76,4 ortalama hassasiyet (mAP) ile Hızlı R-CNN ve Daha Hızlı R-CNN tarafından gölgede bırakılmıştır.

Redmon ve arkadaşları (2016), tek bir görüntüdeki birkaç nesne için sınıf olasılıklarını tahmin eden gerçek zamanlı bir nesne algılama sistemi olan yalnızca bir kez bak (YOLO) modelini geliştirmiştir. YOLO'nun çerçevesi, 57,9'luk bir mAP tutarlılığına ulaşmıştır.

Liu ve arkadaşları (2016) tarafından önerilen benzer bir sistem, %74,3'lük bir mAP ile doğrulukla gerçek zamanlı bir algılama ile de vurgu yaparak, hızlı araç tanımlama görevleri için uygulanabilirliğini ortaya koymuştur.

Birçok araştırma, avantajlarından yararlanmak için çeşitli yaklaşımları entegre eden hibrit prosedürleri araştırmıştır. Moayed (2020) tarafından yapılan bir analiz, CNN gibi derin öğrenmeyi Destek Vektör Makineleri (SVM) gibi sınıflandırma yöntemlerinin yardımıyla entegre etmenin kentsel trafik bağlamlarında %90'a kadar doğruluk seviyelerine ulaşabileceğini göstermiştir.

Tan ve arkadaşları (2020), ağ derinliğini, genişliğini ve çözünürlüğünü optimize etmeyi hesaba katan ve hesaplama verimliliğini korurken üstün doğruluk elde eden EfficientDet'i kullanmıştır. EfficientDet, COCO veri setinde %52,2'lik ortalama bir Ortalama Hassasiyet (mAP) elde ederek değerlendirmelerinde mevcut birçok modeli geride bırakmaktadır.

Bochkovski ve arkadaşları (2020), çalışmalarının veri seti olarak CSPDarknet53'ü entegre ederek YOLOv4 çerçevesi üzerinde çalışmış, modelleri COCO veri setinde %43,5'lik bir mAP doğruluk seviyesini korurken 43 FPS'lik bir algılama hızına sahip olmuştur. Wang vd. (2022), önceden eğitilmiş modellerin kullanılmasının gerekliliğini vurgulayan, "Cityscapes" veri setinde %10 oranında önceden eğitilmiş bir "Faster R-CNN" modeli kullanmıştır.

Rubaiyat ve arkadaşları (2018), LIDAR ve kamera verilerinin entegrasyonunu inceleyerek KITTI veri setinde %90,4'lük yüksek bir tespit doğruluğu elde etmiştir. Derinlik bilgilerinin dahil edilmesi, yalnızca belirsiz arabalar için tespit yeteneğini artırmakla kalmamış aynı zamanda genel tespit güvenini de yükseltmiştir.

Khan ve arkadaşları (2023) tarafından hava koşulları dikkate alındığında, zorlu koşullar altında %78,5'lik bir mAP elde etmiştir. Modelin CNN modeli uygulaması, KITTI veri setinde %91,5'lik bir doğruluğa ulaşmıştır. Öğrenme, Chen ve diğerleri tarafından vurgulandığı gibi, model eğitimi için gerekli etiketli veri miktarını kanıtlamıştır.

Gupta ve arkadaşları (2023), gerçek zamanlı bir uygulama ile %92,7 mAP doğruluğu koruyarak EfficientDet veri kümesinde budama ve niceleme içeren modeli kullanmıştır. YOLO çerçevesi ve sonraki yinelemeleri (YOLOv4, YOLOv5), zamanla gerçek zamanlı tespitte daha iyi doğruluk puanlarına ulaşmıştır.

S. Liang ve Zhang, (2024), özellikle otoyollarda gerçek zamanlı araç tanımlama için tasarlanmış geliştirilmiş bir YOLOv4 modeli tanıtmış, araştırma

sonuçları, önceki yinlemeleri önemli ölçüde geliştiren %91,2'lik bir mAP ile sonuçlanmıştır.

Tahir ve arkadaşları (2024) tarafından yapılan bir çalışma, CNN'i entegre eden bir hibrit modelin, %93,8'lik dikkate değer bir mAP ile kötü hava koşullarında önemli bir doğruluk artışı elde ettiğini göstermiştir. Bu, bilgisayarlı görüş uygulamalarında transformatör modellerinin artan yaygınlığını vurgulamaktadır.

Üretici Çelişkili Ağlar (GAN'lar), araç tespit sistemlerinin dayanıklılığını artırmaya giderek daha fazla katkıda bulunmaktadır. Kukreja vd. (2020), eğitim veri kümelerini geliştirmek için GAN'ları kullanmış, böylece yetersiz açıklamalı verilerden kaynaklanan zorlukları hafifletmiştir. Araç algılama teknolojileri, yaygın ölçütlerde %90'ın üzerinde dikkate değer doğruluk oranlarına ulaştırmış ve çeşitli coğrafi durumlara uyum sağlayabilmiştir. Aşağıdaki tabloda (Tablo 2) literatürde bulunan farklı modellerin özeti bulunmaktadır.

Tablo 2. Literatürde Yapılan Çalışmalar

Yazar	Yıl	Model	Performans
Girshick vd.	2014	R-CNN	70
Bochosvkiy vd.	2016	YOLOv3	57.9
Liu vd.	2016	SSD	74.3
Tan vd.	2020	EfficientDet	52.2
Redmon and Farhadi	2020	YOLOv4t	43.5
Moayed vd.	2021	CNN + SVM	90
Wang vd.	2021	CNN	92
Khang vd.	2021	Faster R-CNN	93
Khan vd.	2021	CNN	91.5
Liang vd.	2021	YOLOv4	91.2
Kukreja vd.	2021	GANs	90
Rubiayat vd.	2022	R-CNN	94.5
Xiang vd.	2023	YOLOv5	94.5
Gupta vd.	2023	EfficientDet	92.7
Jocher vd.	2023	YOLOv5, YOLOv4	50
Tahir vd.	2023	CNN+ViT (Vision Transformers)	93.8
Chughtai vd.	2024	YOLO v5	96
Enrico vd.	2025	Multi-Granularity Detection	68

Kaynak: Yazar tarafından oluşturulmuştur.

3. MODEL

Nesne algılama, bilgisayarla görme alanında önemli bir yer tutmaktadır ve farklı algoritmalar, belirli uygulamaların gereksinimlerine göre çeşitli avantajlar ve dezavantajlar sunmaktadır. Bu bölümde, temel CNN'ler, R-CNN (Region-based Convolutional Neural Networks) ve YOLO (You Only Look Once) algoritmalarının özellikleri ve performansları karşılaştırılacaktır.

3.1. Yapay Zekânın Temel Özellikleri

Görüntü işleme, yapay zekâ ve bilgisayarla görme alanlarının önemli bir parçasıdır. Derin öğrenme mimarileri, bu alanda devrim yaratmış ve nesne tanıma, sınıflandırma gibi görevlerde önemli ilerlemeler sağlamıştır. Bu tezde, Convolutional Neural Network (CNN), Regions with CNN features (R-CNN) ve You Only Look Once (YOLO) gibi üç temel mimarinin temel prensipleri, yapılandırmaları ve uygulama alanları karşılaştırılacaktır.

CNN genellikle bir dizi konvolüsyonel, aktivasyon ve pooling katmanından oluşur. Özellikle görüntü sınıflandırma için tasarlanmıştır. Görüntü sınıflandırma, özellik çıkarımı alanlarında kullanılan bu model basit yapısı ile görsel verilerden etkili bir şekilde özellik öğrenebilir. Ancak nesne algılama için doğrudan kullanılmaz; nesne konumlarını ve sınıflarını çıkarma yeteneği yoktur.

Convolutional Neural Network (CNN), yapay sinir ağları arasında en yaygın olanlardan biridir ve görüntü verilerini sınıflandırmak için özel olarak tasarlanmıştır. CNN'ler, giriş görüntülerini işlemekte kullanılan katmanlar arasında konvolüsyon işlemleri uygular. Bu yapı, görüntülerdeki temel özellikleri (kenarlar, dokular vb.) öğrenerek daha karmaşık özellikleri ve sınıfları tanımak için derin katmanlar oluşturur. CNN mimarisi, genellikle konvolüsyon katmanları, aktivasyon fonksiyonları, havuzlama katmanları ve tamamen bağlı katmanlardan oluşur.

R-CNN, nesne tespiti alanında önemli bir yenilik olarak ortaya çıkmıştır. Basit bir sınıflandırmadan ziyade, belirli bölgelerde nesne varlığını tespit etmek için tasarlanmıştır. R-CNN, görüntülerde nesne olabilecek alanları belirlemek için öneri bölgeleri oluşturur ve ardından bu bölgeleri bir CNN ile işleyerek özellikler çıkarır. R-

CNN'nin yapılandırması, önce bölge önerme (region proposal) ile nesne olma olasılığı yüksek alanların tespiti içerir, ardından bu bölgeleri bir CNN ile işleyerek her biri için özellikler elde eder ve en son olarak bu çıkarılan özelliklerle nesnenin türünü tanımlar ve konumlandırma bilgilerini düzeltir.

YOLO, nesne tespiti için başka bir popüler mimardır fakat farklı bir yaklaşım sergiler. YOLO, bütün görüntüyü tek bir geçişte işler ve bu sayede daha hızlı sonuçlar üretir. Görüntüyü bölmelere ayırarak her bölmede nesne varlığını tahmin eder ve bu tahminleri aynı anda gerçekleştirir. YOLO'nun temel bileşenleri, tek bir geçiş yaparak görüntüyü işlemesi, grid yapısı kullanarak görüntüyü hücreler halinde bölmesi ve sınıf olasılıkları ile konum tahminleri döndürmesidir. Aşağıdaki tabloda (Tablo 3) CNN tabanlı yapay zekâ yöntemlerinin hız, doğruluk, kullanım ve uygulama açısından karşılaştırılmıştır.

Tablo 3. CNN Performans Değerlendirmesi

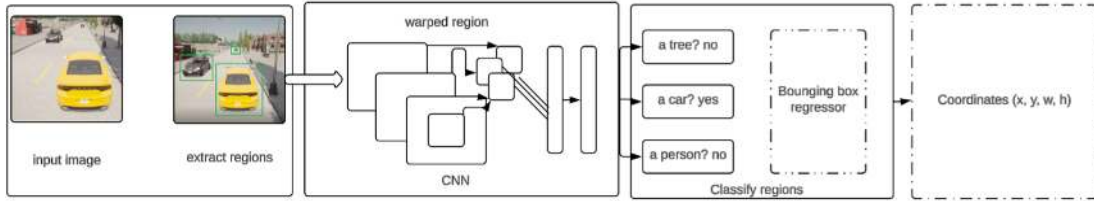
Özellik	CNN	R-CNN	YOLO
Hız	Orta	Yavaş	Hızlı
Doğruluk	Orta	Yüksek	Orta-Yüksek
Kullanım	Görüntü sınıflandırma	Nesne algılama	Gerçek zamanlı nesne algılama
Uygulama	Genel görsel görevler	Güvenlik ve izleme	Otonom sürüş, video analizi

Kaynak: Yazar tarafından oluşturulmuştur.

CNN, R-CNN ve YOLO, nesne algılama alanında farklı yaklaşımlar sunan üç önemli yöntemdir. CNN'ler daha çok görüntü sınıflandırma için kullanılırken, R-CNN nesne algılamasında yüksek doğruluk sağlarken daha yavaş çalışır. YOLO, gerçek zamanlı uygulamalarda hızlı sonuçlar sunmakta, ancak küçük nesnelerde zorluk yaşayabilmektedir. Her bir yöntemin kendine özgü güçlü ve zayıf yanları vardır, bu nedenle kullanım amacına yönelik seçimler yapılmalıdır. Bu çalışmada R-CNN mimarisi kullanılmıştır.

3.2. R-CNN

Otonom araçlar ve akıllı ulaşım sistemleri, çevrelerindeki araçları doğru ve hızlı bir şekilde algılamayı gerektirir. R-CNN (Regions with CNN features), nesne algılama alanında öncü bir yöntem olarak, bu ihtiyaca cevap verebilir. CNN'nin genel görünümü aşağıdaki şekilde gösterilmiştir (Şekil 2).



Şekil 2. CNN Mimarisi

Kaynak: Yazar tarafından oluşturulmuştur.

Bölge önerme (region proposal), özellik çıkarımı (feature extraction), sınıflandırma (classification), olmak üzere nesne algılama sürecini temel olarak üç ana aşamada gerçekleştirir

Bu aşamaların her biri, nesne algılama başarısı için kritik öneme sahiptir ve birlikte çalışarak görüntüdeki nesnelerin konumlarını ve sınıflarını belirlemektedir.

3.2.1. Bölge Önerme (Region Proposal)

Bölgeler önerme aşaması, görüntüde olası nesnelerin bulunabileceği bölgelerin belirlenmesini içerir. R-CNN, bu amaçla Selective Search gibi algoritmaları kullanır. Selective Search, görüntüyü farklı ölçeklerde segmentlere ayırarak ve benzer bölgeleri birleştirerek nesne adaylarını oluşturur (Alam vd., 2023).

Başlangıç Segmentasyonu: Görüntü, renk, doku, boyut ve şekil gibi özelliklere göre küçük bölgelere ayrılır.

Benzer Bölgelerin Birleştirilmesi: Komşu bölgeler arasındaki benzerlikler hesaplanır ve en benzer bölgeler birleştirilir. Bu işlem, hiyerarşik bir şekilde tekrarlanır.

Bölge Önerilerinin Oluşturulması: Birleştirme işlemi sırasında elde edilen tüm bölgeler, olası nesne bölgeleri olarak kabul edilir.

“Selective Search”, farklı ölçeklerde ve şekillerde bölgeler önererek çeşitli nesne boyutlarını kapsar. Bu, özellikle trafikteki araçların farklı uzaklıklarda ve açılarda görünmesi durumunda önemlidir.

3.2.2. Özellik Çıkarımı (Feature Extraction)

Özellik çıkarımı aşaması, her bir bölge adayı için anlamlı ve ayırt edici özelliklerin elde edilmesini içerir. R-CNN, bu amaçla önceden eğitilmiş bir CNN (Convolutional Neural Network) modeli kullanır. CNN, görüntüleri otomatik olarak öğrenen ve hiyerarşik özellikler üreten derin öğrenme modelidir.

CNN Modelinin Yapısı:

Evrişim Katmanları (Convolutional Layers): Görüntü üzerinde evrişim işlemleri yaparak farklı özellik haritaları oluşturur.

Aktivasyon Fonksiyonları (Activation Functions): Evrişim katmanlarının çıktılarına doğrusal olmayan bir aktivasyon uygular (örneğin, ReLU).

Havuzlama Katmanları (Pooling Layers): Özellik haritalarının boyutunu küçültürken hesaplama yükünü azaltır ve ağına daha genel özellikler öğrenmesini sağlar.

Tam Bağlantılı Katmanlar (Fully Connected Layers): Yüksek seviyeli özellikleri birleştirerek sınıflandırma için kullanılır.

CNN, büyük bir veri kümesi üzerinde (örneğin, ImageNet) önceden eğitilir ve transfer öğrenimi (transfer learning) yaklaşımıyla R-CNN'e dahil edilir. Transfer öğrenimi, modelin daha az veriyle daha iyi performans göstermesini sağlar.

R-CNN, görüntüdeki aday nesne bölgelerini tanımlayan ve her bölgeden özellikleri çıkarmak için bir CNN kullanan, son olarak her bölgeyi belirli bir nesne sınıfına sınıflandıran ve nesne konumlandırmasını iyileştirmek için sınırlayıcı kutu regresyonu gerçekleştiren üç ana adımda çalışır, bunlar bölge önerisi, özellik çıkarımı ve sınıflandırma ve regresyondur.

Giriş görüntüsünün şu şekilde tanımlanırsa:

$$[I \in R^{H \times W \times C}] \quad (1)$$

(H, W): Resmin boyu ve eni.

(C): Renk kanalı sayısı.

Seçici Arama, her bir teklifin (r_i) (I)'nin dikdörtgen bir alt bölgesi olduğu bir bölge teklifi kümesi (R) üretir.

$$r_i = [x_{min}, y_{max}, x_{min}, y_{max}] \forall i \in 1, 2 \dots n \quad (2)$$

Burada n alt bölge sayısını ifade etmektedir.

$r_i = [x_{min}, y_{max}, x_{min}, y_{max}]$ alt bölgeyi çerçeveleyen kutunun koordinatlarıdır.

Özellik çıkarmak için öncelikle CNN'nin girdisi olan r_i 'nin yeniden uygun boyuta dönüştürülmesi gerekir. Önceden eğitilmiş CNN'i kullanarak özellik çıkarmak için

$$f_{CNN}(r_i) = \text{Feature Vector} \in D \quad (3)$$

Burada D : Özellik vektörünün boyutudur. Evrişim operasyonu aşağıdaki şekilde formüle edilir:

$$Z_{i,j,k} = \sum_m \sum_n \sum_c W_{m,n,c,k} \cdot I_{(i+m),(j+n),c} + b_k \quad (4)$$

(z_k): k 'nın puanı

$p(y = k|r_i)$: r_i alt bölgesinin k sınıfında bulunma olasılığı.

Sınırlayıcı Kutu Regresyonu: Bir regresyon modeli, sınırlayıcı kutunun konumunu iyileştirir. Her bölge (r_i), için, regresyon modeli şunları öngörür: $[\Delta x, \Delta y, \Delta w, \Delta h]$. Bu değerler, sınırlayıcı kutu koordinatlarını ayarlar:

$$x' = x + w \cdot \Delta x \quad (7)$$

$$y' = y + h \cdot \Delta y \quad (8)$$

$$w' = w \cdot \exp(\Delta w) \quad (9)$$

$$h' = h \cdot \exp(\Delta h) \quad (10)$$

$[x', y', w', h']$: Ayarlı sınırlayıcı kutu.

$[x, y, w, h]$: Orijinal sınırlayıcı kutu.

Burada kayıp fonksiyonu CNN’i, sınıflandırma ve yerelleştirmeyi birleştiren çoklu görevli bir kaybı optimize etmek adına hesaplanmalıdır.

$$L = L_{cls} + \lambda L_{loc} \quad (11)$$

L_{cls} : Cross-entropy kayıp değeri.

$$L_{cls} = - \sum_k t_k \log(p_k) \quad (12)$$

Here, (t_k) (k) ’nın gerçek sınıfı (k) , and (p_k) ise tahmin edilen sınıf olarak tanımlanır.

(L_{loc}) : (L_1) regresyonun kayıp değeridir.

$$L_{loc} = \sum_{i \in x, y, w, h} smooth_{L_1}(\Delta_i - \Delta'_i) \quad (13)$$

$$smooth_{L_1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise} \end{cases} \quad (14)$$

Giriş görüntüsü (I) , (N) bölge teklifi $\{r_1, r_2, \dots, r_N\}$ üretmek için bölge teklifi algoritmasına geçirilir.

Her teklif r_i , bir özellik vektörü çıkarmak için CNN'e beslenir.

Özellik vektörü, “softmax” işlevi kullanılarak nesne kategorilerinden veya arka plandan birine sınıflandırılır. R-CNN’in birincil dezavantajı, her bölge teklifinin CNN üzerinden işlenmesini gerektirmesi ve bunun sonucunda önemli bir hesaplama masrafı oluşması nedeniyle yavaş performansdır. R-CNN’deki seyrek eğitim, sınıflandırıcı ve sınırlayıcı kutunun ayrı ayrı eğitilmesiyle sonuçlanır ve bu da verimliliği azaltır. Hızlı R-CNN, Daha Hızlı R-CNN ve Maske R-CNN gibi gelişmeler, paylaşılan CNN hesaplamalarını dâhil ederek ve bölge tekliflerinin entegrasyonunu geliştirerek bu sınırlamaları hafifletir.

3.3. Kalman Filtresi

Kalman filtresi, zamanla değişen dinamik sistemlerin durumunu tahmin etmek ve bu tahminleri gözlem verileri ile güncellemek için geliştirilmiş bir algoritmadır.

İleri düzey matematiğe dayanan bu yöntem, özellikle mühendislik, robotik, uzay ve otomotiv endüstrilerinde yaygın olarak kullanılmaktadır. Kalman filtresinin temel amacı, gürültü ve belirsizlik içeren sistemler üzerinde en iyi tahmin değerlerini elde etmektir.

Kalman filtresi, belirli bir sistemin durumunu zaman içinde sürekli olarak tahmin eder ve ölçüm verilerinden yararlanarak bu tahminleri güncelleyerek durumun en doğru tahmin değerini bulmaya çalışır. İlk olarak, Rudolf Kalman tarafından 1960'larda geliştirilen bu metod, dinamik sistemlerin durum uzayını modellemek için güçlü bir araçtır.

Bir Kalman filtresi, Durum geçiş modeli, gözlem modeli olmak üzere iki temel model ile çalışır. Durum geçiş modeli;

$$x_k = F_k x_{k-1} + B_k u_{k-1} + w_{k-1} \quad (15)$$

x_k : k zamanındaki durum vektörü

F_k : Durum geçiş matrisidir. Bu matris, sistemin geçmişten geleceğe nasıl geçeceğini tanımlar.

B_k : Kontrol girdi matrisidir; sistemin kontrol girdisini temsil eder.

u_{k-1} : Kontrol vektörü. Kontrol girişi olmadığında bu terim sıfırlanabilir.

w_{k-1} : Durum geçişinde oluşan gürültü veya hata terimidir. Genellikle bağımsız ve normal dağılıma sahip bir gürültü olarak model alınır.

Sistemin gözlemlerinin nasıl elde edildiğini gösteren denklem ise:

$$z_k = H_k x_k + v_k \quad (16)$$

Burada;

z_k : k zamanındaki ölçüm vektörü.

H_k : Ölçüm matrisidir. Bu matris, gerçek durumun gözlemlenen veriye nasıl dönüştüğünü açıklar.

v_k : Ölçüm hatası veya gürültü terimi; genellikle normal dağılımla tanımlanır.

Kalman filtresi, tahmin adımı ve güncelleme adımı iki ana adımda çalışır. Tahmin adımı:

$$x'_{k|k-1} = F_k x'_{k-1|k-1} + B_k u_{k-1} \quad (17)$$

Kovaryans Tahmini:

$$P_{k|k-1} = F_k P_{k-1|k-1} F_k^T + Q \quad (18)$$

Ölçüm gerçekleştiğinde, tahminleri güncellemek için aşağıdaki adımlar izlenir:

Mevcut tahmin ile gerçek ölçüm arasındaki farkı hesaplanır

$$y_k = z_k - H_k x'_{k|k-1} \quad (19)$$

Ölçüm hatasının ne kadar güvenilir olduğunu anlamak için gözlem hatası kovaryansını hesaplanır.

$$S_k = H_k P_{k|k-1} H_k^T + R \quad (20)$$

Güncelleme adımında ne kadar ağırlık verilmesi gerektiğini belirlenir

$$K_k = P_{k|k-1} H_k^T S_k^{-1} \quad (21)$$

Durum tahminini güncelleyerek en güncel durumu elde edilir

$$x'_{k|k} = x'_{k|k-1} + K_k y_k \quad (22)$$

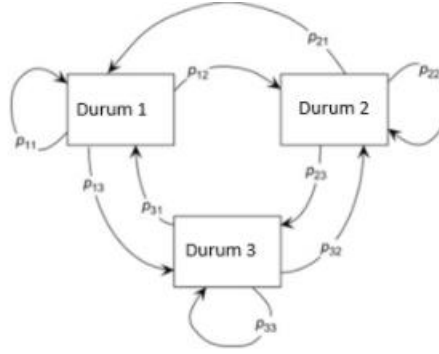
Belirsizliği güncelleyerek yeni taut ya da emin durum tahminini elde edilir

$$P_{k|k} = (I - K_k H_k) P_{k|k-1} \quad (23)$$

Kalman filtresi, durum tahmininin en düşük hata kare ortalamasını sağlamakta, bu da onu en optimal çözüm haline getirir. Çok sayıda uygulamada, ölçüm ve durum gürültülerinin normal dağıldığı varsayımı altında çalışır. Bu filtre, sistemlerin lineer olduğu varsayımına dayanır. Lineer olmayan sistemler için genellikle Genişletilmiş Kalman Filtresi (EKF) veya Unscented Kalman Filtresi (UKF) gibi yöntemler kullanılır. Kalman filtresi, gürültü terimlerini normal dağılıma dayandırarak varsayımlar yapar. Sistemleri modellemek için kullanılan gürültü matrisleri, sistemin dinamiklerini düzeltmeye yardımcı olur. Kalman filtresi, dinamik sistemlerin durumunu tahmin etmek ve ölçüm verilerini işlemek için etkili bir yöntem sunar. Matematiksel modeli, tahmin süreci ve güncelleme mekanizması ile birlikte, sistemlerin performansını artırmak için güçlü bir araçtır.

3.4. Saklı Markov Zinciri

Markov zinciri, her bir durumun yalnızca bir önceki durumdan bağımsız olarak, gelecek durumlarının belirlenmesine dayanan bir stokastik süreçtir. Başka bir deyişle, bir sistemin gelecekteki durumu, mevcut durumuna yalnızca bağlıdır ve geçmiş durumların bilgisi bu tahmin için gereksizdir. Bu özellik, Markov zincirine "Markov Özelliği" olarak adlandırılır. Markov zincirlerinin genel yapısı aşağıdaki şekilde verilmiştir (Şekil 3).



Şekil 3. Markov Zinciri Modeli

Kaynak: Yazar tarafından oluşturulmuştur.

Bir Markov zinciri, belirli bir durumu temsil eden durum uzayı $S = \{s_1, s_2, \dots, s_n\}$ ile tanımlanır. Zamanın kesikli olması durumunda, sistemin gelecekteki durumu, geçiş olasılıkları kullanılarak hesaplanır. P 'nin geçiş olasılıkları matrisini temsil ettiğini varsayalım; burada P_{ij} i durumundan j durumuna geçme olasılığıdır.

$$P = \begin{bmatrix} P_{11} & \dots & P_{1n} \\ \vdots & \ddots & \vdots \\ P_{n1} & \dots & P_{nn} \end{bmatrix} \quad (24)$$

P_{ij} tüm olasılıkları temsil eder ve her satırın tüm olasılıklarının toplamı 1'e eşit olmalıdır. Eğer sistem (t) adımında s_i durumundaysa, $(t+1)$ adımında s_j durumuna geçiş olasılığı şu şekilde verilir:

$$P(X_{t+1} = s_j | X_t = s_i) = P_{ij} \quad (25)$$

Burada X_t sistemin t anındaki durumu ifade eder. Markov zincirleri, stokastik süreçleri modellemek için etkili bir matematiksel araçtır. Özellikle belirsizliklerin bulunduğu durumların analizi için kullanışlıdır. Geçiş olasılıkları ve başlangıç

dağılımı ile tanımlanan bu modeller, çeşitli disiplinlerde uygulama bulmuş ve önemli sonuçlar sağlamıştır. Gizli Markov Modelleri, normal Markov modellerinin bir uzantısıdır. Ana özellik, sistemin evriminin bir Markov sürecini takip etmesine rağmen, durumların doğrudan gözlemlenmemesidir. Bunun yerine, her gizli durum bir olayı olasılıksal bir süreç aracılığıyla gözlemlenebilir hale getirir. Bu, zaman adımı t 'deki gözlemlenmeyen durumu temsil eden ayrık rastgele değişkenlerdir. N gizli duruma sahip bir modelde, şu şekilde gösterilir:

$$s(t) \in \{1, 2, \dots, N\} \quad (26)$$

Bu değişkenler ölçebildiğimiz verileri temsil eder. Ayrık veya sürekli olabilirler. Tüm olası gözlemlerin kümesi Ω ile gösterilir.

$$x(t) \in \Omega \quad (27)$$

Başlangıç durumu dağılımı (π), sürecin i durumunda başlama olasılığıdır ve şu şekilde verilir:

$$\pi(i) = P(S(1) = i) \quad \forall i \quad (28)$$

Geçiş olasılıkları (a_{ij}), i durumundan j durumuna geçiş olasılığıdır ve şu şekilde gösterilir:

$$a_{ij} = P(S(t+1) = j | S(t) = i) \quad (29)$$

Şu özelliği sağlamalıdır:

$$\sum_i a_{ij} = 1 \quad (30)$$

Emisyon Olasılıkları $b_j(x)$, gizli durum j verildiğinde belirli bir çıktı x 'i gözlemlene olasılığıdır:

$$b_j(x) = P(X(t) = x | S(t) = j) \quad (31)$$

Böylece, tam ortak dağılım kısaca şu şekilde ifade edilebilir:

$$P(S(1:T), X(1:T)) = \pi(S(1)) \cdot \prod_{t=2}^T b_{S(t)}(X(t)) \quad (32)$$

S notasyonu, modelin nasıl başladığını (π), gizli durumlar a_{ij} arasında nasıl geçiş yaptığını, ($b_j(x)$) çıktıları göstermektedir.

3.5. ANN

Yapay Sinir Ağları (ANN), insan beyninin çalışma prensiplerinden esinlenerek geliştirilmiş, karmaşık verileri işlemek ve öğrenmek amacıyla kullanılan bir makine öğrenme modelidir. Gelişmiş yapıları ve öğrenme yetenekleri ile günümüzde birçok uygulama alanında (görüntü tanıma, doğal dil işleme, oyun oynama vb.) yaygın olarak kullanılmaktadır.

3.5.1. Nöronlar

Bir yapay sinir ağının temel birimi, biyolojik sinir hücrelerine benzer şekilde işlev gören nörondur. Her nöron, birkaç temel bileşene sahiptir:

Girişler (Inputs): Nöronun aldığı veriler. Bu veri seti genellikle başka nöronlardan gelir.

Ağırlıklar (Weights): Her girişe atanmış ve modelin öğrenme sürecinde güncellenen katsayılar. Ağırlıklar, girişlerin nöron üzerindeki etkisini belirler.

Aktivasyon Fonksiyonu: Nöronun çıktısını belirleyen matematiksel fonksiyondur. Genellikle lineer olmayan fonksiyonlar kullanılır.

3.5.2. Aktivasyon Fonksiyonları

Aktivasyon fonksiyonu, nöronun hangi durumda etkin olacağını belirler ve geçerli bir çıktının üretimini sağlar. Yaygın kullanılan aktivasyon fonksiyonları şunlardır:

Tablo 4. Sık kullanılan aktivasyon Fonksiyonları

Aktivasyon Fonksiyonu	Formül	Kullanım Amacı
Sigmoid	$\sigma(x) = \frac{1}{1 + e^{-x}}$	Sigmoid, çıktıyı (0, 1) aralığında sıkıştırır
ReLU (Recified Linear Unit)	$f(x) = \max(0, x)$	Negatif değerleri sıfıra eşitler ve pozitif değerleri olduğu gibi geçirir.
Tanh (Hiperbolik Tanjant)	$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$	Çıktıyı (-1, 1) aralığında sıkıştırır

Kaynak: Yazar tarafından oluşturulmuştur.

3.6. Sinir Ağı Mimarisi

Yapay sinir ağları genellikle üç ana katmandan oluşur:

Giriş Katmanı: Ağa verilerin girdiği yerdir. Her nöron bir özellik veya giriş verisini temsil eder. Gizli Katmanlar: Giriş ve çıkış katmanı arasında bulunan ve modelin karmaşık ilişkileri öğrenmesine yardımcı olan katmanlardır. Birden fazla gizli katman kullanılabilir. Çıkış Katmanı: Modelin son çıktısını üreten katmandır. Çıktıdaki sınıf sayısı, sınıflandırma problemlerinin doğasına göre değişir.

Bir nöronun çıktısını hesaplamak için kullanılan formül şu şekildedir:

$$y = f\left(\sum_{i=1}^n w_i x_i + b\right) \quad (33)$$

Burada:

y nöronun çıktısıdır.

w_i : ağırlıklardır.

x_i : girişlerdir.

b : bias terimidir, nöronun çıktısını offsetlemek için kullanılır.

f : seçilen aktivasyon fonksiyonunu temsil eder.

Geri Yayılım (Backpropagation)

Yapay sinir ağlarının öğrenme süreci, geri yayılım algoritması ile gerçekleştirilir. Bu süreç, modelin tahmin hatasını her nöron için hesaplamaya ve ağırlıkları güncellemeye dayanır.

Öncelikle, ağ bir tahminde bulunur ve bu tahminle gerçek değer arasındaki hata hesaplanır:

$$E = \frac{1}{2} (y_{gerçek} - y_{tahmin})^2 \quad (34)$$

Sonrasında, hata gradyanı, ağırlıkların güncellenmesi için geri yayılır:

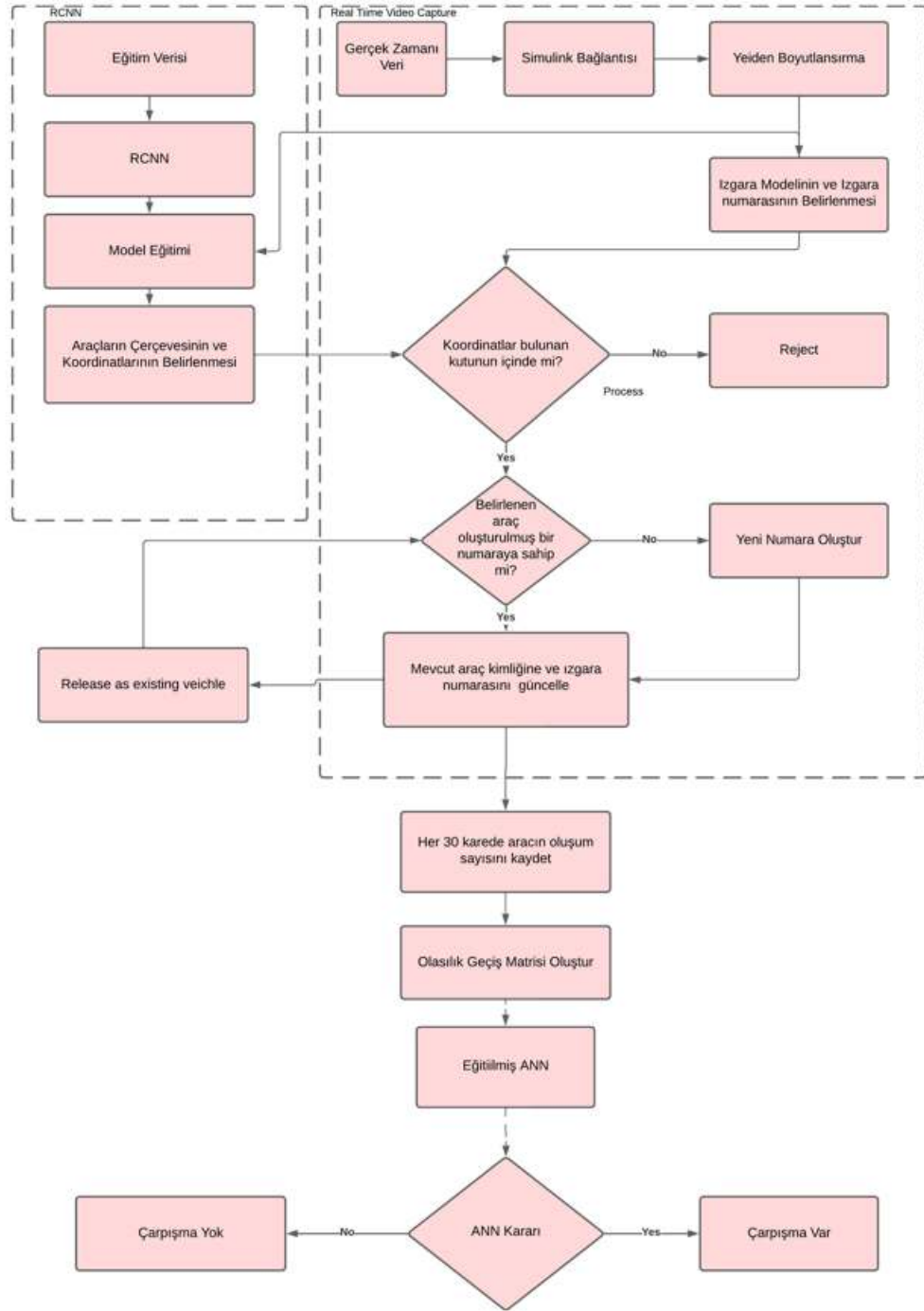
$$w_i^{yeni} - w_i^{eski} - \eta \frac{\partial E}{\partial w_i} \quad (35)$$

Burada η öğrenme oranını temsil eder ve kullanıcı tarafından belirlenir.

Yapay sinir ağları, karmaşık veri ilişkilerini modellemek için güçlü bir araçtır. Matematiksel yapısı, nöronlar, katmanlar ve öğrenme mekanizmaları ile sistemin işleyişini anlamamıza yardımcı olur. Bu yapı, sinir ağlarının çeşitli uygulamalarda başarılı olmasını sağlamaktadır.

3.7. Önerilen Model

Bu çalışma, Hızlı R-CNN, gürültü azaltma için ayarlanmış bir Kalman Filtresi, tahmin için Gizli Markov Zinciri ve yalnızca kamera girişi kullanarak gerçek zamanlı olarak araç kazalarını tespit ve tahmin etmek için ANN'i kullanan hibrit model sunmaktadır. Aşağıdaki Şekiller (Şekil 1 ve Şekil 2), modelin akış şemasını ve sözde kodunu temsil etmektedir.



Şekil 4. Modelin Akış Şeması

Kaynak: Yazar tarafından oluşturulmuştur.

```

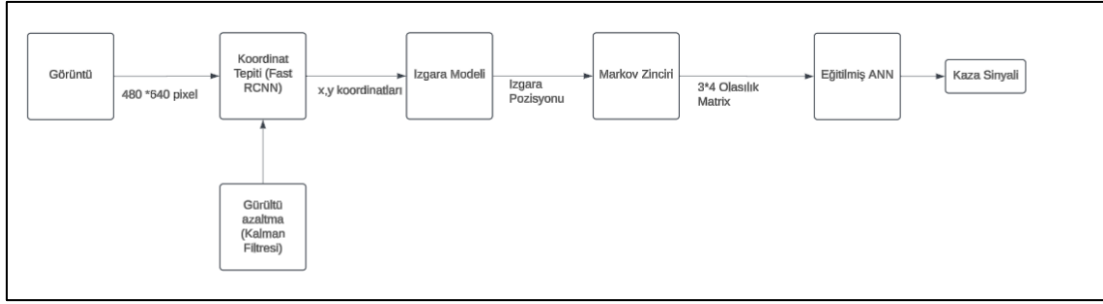
# Step 1: Initialize the System
Initialize Faster RCNN model and Initialize ANN model and Initialize Vehicle Tracking
Database
# Step 2: Process Incoming Video Frames
while True:
    frame = Capture_Real_Time_Video() and resized_frame = Resize_Frame(frame)
# Step 3: Detect Vehicles
    detected_vehicles = Detect_Vehicles(resized_frame, RCNN_Model)
    for vehicle in detected_vehicles:
        bounding_box, coordinates = Get_Bounding_Box(vehicle)
# Step 4: Check if coordinates fall inside bounding box
        if not Inside_Bounding_Box(coordinates, bounding_box): Reject_Vehicle() and continue
# Step 5: Check if vehicle matches an existing ID
        if Vehicle_Exists(vehicle): Update_Vehicle_Features(vehicle)
        else: Register_New_Vehicle(vehicle)
# Step 6: Release vehicle as existing vehicle
        Release_Vehicle(vehicle)
# Step 7: Record vehicle occurrences every 30 frames
        if Frame_Count % 30 == 0: Update_Vehicle_Occurrence() and
Generate_Probability_Transition_Matrix
# Step 8: Train ANN Model
        Train_ANN(Vehicle_IDs, Grid_Labels)
# Step 9: Get ANN Weights
        ANN_Weights = Get_ANN_Weights()
# Step 10: Calculate Probability of Grid Change
        Probability = Calculate_Grid_Change_Probability(ANN_Weights)
# Step 11: Check if Probability Exceeds Threshold
        if Probability > Predetermined_Threshold: Send_Crash_Signal()
    else:
        No_Crash_Signal()

```

Şekil 5. Modelin Sözde Kodu

Kaynak: Yazar tarafından oluşturulmuştur.

İlk test için kullanılan veri seti, YouTube gibi herkese açık çevrimiçi platformlardan belirlenirken, veri setinin kalitesini ve sağlam performans değerlendirmesi için uygunluğunu sağlamak amacıyla sistematik bir derleme ve protokolü kullanılmıştır. Bu veri setini hazırlamanın kritik bir bileşeni, her kaza için dikkatlice belirlenmektedir. Bu, seçilen 67 videonun her birinin detaylı, manuel analizi ile çarpma anını tam olarak tanımlamak ve zaman damgası almak suretiyle gerçekleştirilmiştir. İzgara görüntü başlangıç noktası (0,0) koordinatında başladığı için, “dash cam” ler aracın genişliğinin tam ortasına yerleştirilmelidir. Aynı nedenle, “dash cam” tarafından aracın kaputunun görünmemesi gerekmektedir. Çarpma zamanı, üç farklı gözlemcinin ortalamasını alarak hesaplanmıştır. Veri setini çeşitlendirmek için CARLA simülasyon motoru kullanılarak çeşitli çarpışma senaryoları oluşturulmuştur. Bu çalışmada kullanılan model, aşağıdaki şekilde (Şekil 6) gösterilmiştir.



Şekil 6. Modelin Görseli

Kaynak: Yazar tarafından oluşturulmuştur.

3.7.1. Veri Seti ve Ön İşleme

Nesne algılama modülü, açıklamalı araç görüntülerinden oluşan MATLAB tabanlı bir veri seti kullanılarak eğitilmiştir. Görüntüler önce uint8 biçimine normalize edilir ve [0–255] yoğunluk aralığına ölçeklenir. ResNet-50 üzerinde önceden eğitilmiş bir Fast R-CNN modeli, her karede araçları algılamak için kullanılır. Şekil 5 ve Şekil 6’da örnek veri seti ve verasetinde kullanılan resimler verilmiştir. Model, ufuk çizgisinin altındaki araçlara odaklanacak şekilde yapılandırılmıştır, bu da yanlış pozitifleri ve hesaplama karmaşıklığını azaltır. Tekil bir görüntü içindeki araçları tanımlamak ve bunları ilgili algılama puanlarıyla açıklamak için, araç görüntüleri üzerinde bir Hızlı R-CNN modeli eğitilir. Otomobilleri algılamak için, 295 görüntü içeren Caltech Cars 1999 (Weber ve Perona, 2022) ve 2001 (Philip vd., 2022) veri kümelerinden alınan eğitim için verilerin hazırlanması gerekir.

	1 imageFilename	2 vehicle
1	'vehicleImages/image_00001.jpg'	[220,136,35,28]
2	'vehicleImages/image_00002.jpg'	[175,126,61,45]
3	'vehicleImages/image_00003.jpg'	[108,120,45,33]
4	'vehicleImages/image_00004.jpg'	[124,112,38,36]
5	'vehicleImages/image_00005.jpg'	[257,124,43,31]
6	'vehicleImages/image_00006.jpg'	[364,108,29,33]
7	'vehicleImages/image_00007.jpg'	[180,124,54,47]
8	'vehicleImages/image_00008.jpg'	[178,127,40,35]
9	'vehicleImages/image_00009.jpg'	[171,120,38,31]
10	'vehicleImages/image_00010.jpg'	[169,117,43,38]
11	'vehicleImages/image_00011.jpg'	[164,115,57,49]

Şekil 7. Veri Seti Örneği

Kaynak: Yazar tarafından oluşturulmuştur.



Şekil 8. Veri Setindeki Görsellerden Örnekler

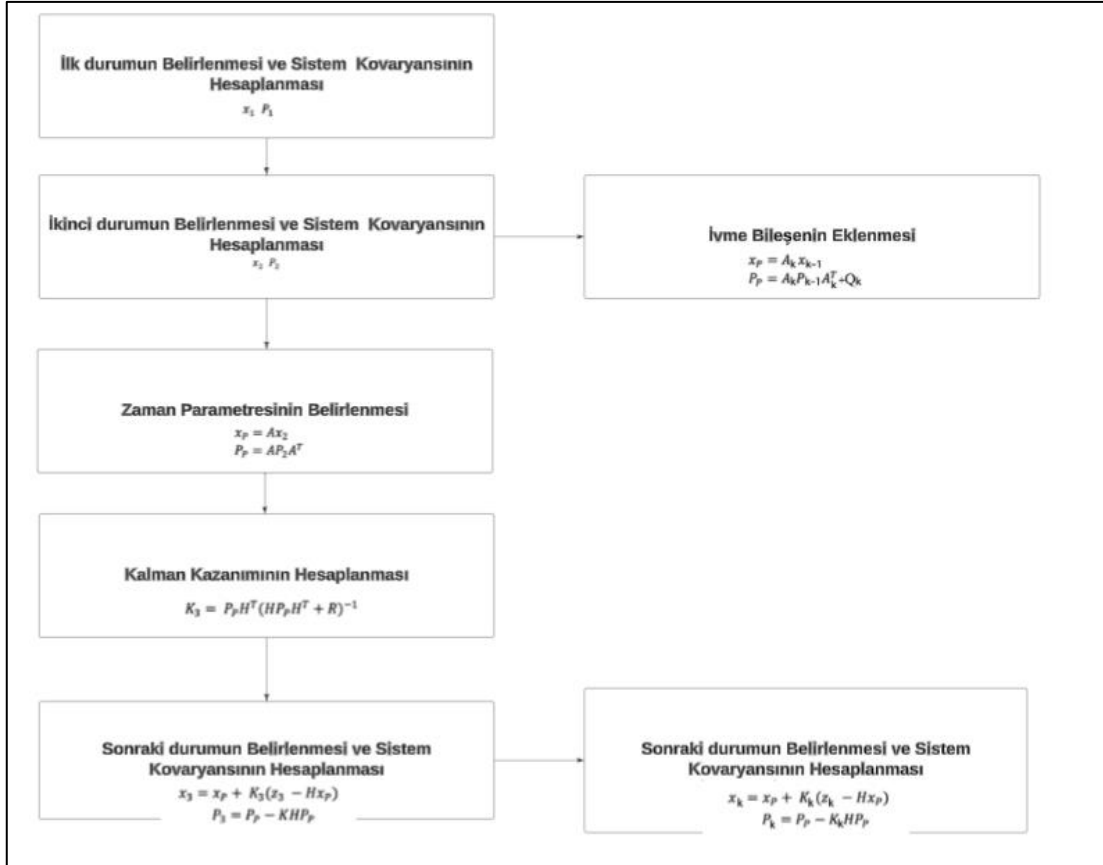
Kaynak: Yazar tarafından oluşturulmuştur.

3.7.2. Kalman Filtresi

Başlangıç durumu $x_t = [x, y, \hat{x}, \hat{y}]$ olan bir nesneyi takip edildiğini varsayıldığında (burada (x, y) pozisyonu, $(\hat{x}, \hat{y})$ hızı) burada oluşan matris aşağıdaki şekilde oluşur.

$$A = \begin{bmatrix} 1 & 0 & dt & 0 \\ 0 & 1 & 0 & dt \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (36)$$

Aşağıdaki şekilde Kalman Filtresinde ivme bileşeni eklenmesi durumunda yapılması gereken değişiklikler bulunmaktadır.



Şekil 9. Kalman Filtresinde Yapılan Değişiklikler

Kaynak: Yazar tarafından oluşturulmuştur.

F_k , B_k , u_k , Q_k , H_k , R_k , Δt_k zaman değişkenleridir. Bu parametrelerin uyarlanması, yanıt verebilirliği artırır, gecikmeyi azaltır ve kırılganlığı yükselen dinamik, örtüşme (occlusion) içeren trafik sahnelerinde özellikle kritik olan tepkiselliği artırır.

Bu yapı sabit hızda hareket eden araçları modellemede iyi çalışırken yüksek hızlı hareketleri tespit etmede yetersiz kalacaktır. Kalman Filtresi, tahminleri çerçeveler ilerledikçe ayarlayarak gürültüyü dikkate alarak çalışır. Gürültülü gözlemler ile tahminlerin birleştirilmesi, Kalman Filtresi'nin sağlam durum tahminleri elde etmesini sağlar ve bu, resim işleme görevlerinde gürültüyü azaltmak ve izlemeyi geliştirmek için hayati bir araç haline getirir.

Mevcut durum vektörü artık şudur:

$$state = [x; vx; y; w; wv; h; hv] \quad (37)$$

Bu duruma hız bileşenini eklemek için durum vektörü şu şekilde güncellenmelidir.

$$state = [x; vx; ax; y; vy; ay; w; wv; aw; h; hv; ah] \quad (38)$$

Burada $ax, ay, aw, ah, ax, ay, aw, ah$ sırasıyla x, y, genişlik ve yükseklik için ivme bileşenleridir. Düzenlemelerden sonra, durumu ivmeyi de içerecek şekilde geçiş matrisini uzatmak gerekir. Yeni matris şu şekilde yapılandırılmıştır:

$$A = \begin{bmatrix} 1 & dt & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & dt & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & dt & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & dt & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & dt \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (39)$$

Ayrıca, ölçüm modelinde ivme bileşenlerini dikkate almak gerekmektedir. Ancak sadece konum ve boyut bilgileri ölçüldüğünden, ölçüm modeline ivme bileşenleri eklemek gerekmez. Ölçüm modeli olduğu gibi kalabilir. Gürültü matrisinin güncellenmesi, ivme bileşenlerinin dikkate alınması gerektiğinden şöyle olmalıdır:

$$G_{ld} = [\frac{dt^2}{2}; dt; 1] \quad (40)$$

$$Q_{ld} = G_{ld} * G'_{ld} \quad (41)$$

$$Q = blkdiag(Q_{ld}; Q_{ld}; Q_{ld}; Q_{ld}) \quad (42)$$

Ayrıca ivme bileşenlerini sıfıra ayarlamak gerekmektedir. Son olarak, güncellenmiş parametreler ile filtre oluşturulmalıdır. Nesne ile kamera arasındaki mesafeyi tespit etmek için bir eşik değeri belirlenmelidir. Uygun koşullara rağmen tespitler izlemelere atanmadığında, bu değer artırılması önerilir. Bu çalışmada, nesnelerin daha büyük mesafelerde etkili bir şekilde tespit edilmesi için eşik değeri 50 olarak belirlenmiştir. Atanmamış her tespit yeni bir süreci başlatır. Bazı tespitler hatalı olabilir; bu nedenle, tüm tespitler başlangıçta geçici olarak sınıflandırılır. Bu örnek,

beş güncellemeden üç tespit varsayılan ayarını kullanmaktadır. Bu çalışmada, araçları daha yüksek bir doğrulukla tespit etmek için eşik değeri 50 olarak genel bir değerlendirme olmasına rağmen 30 olarak belirlenmiştir.

3.7.3. R-CNN

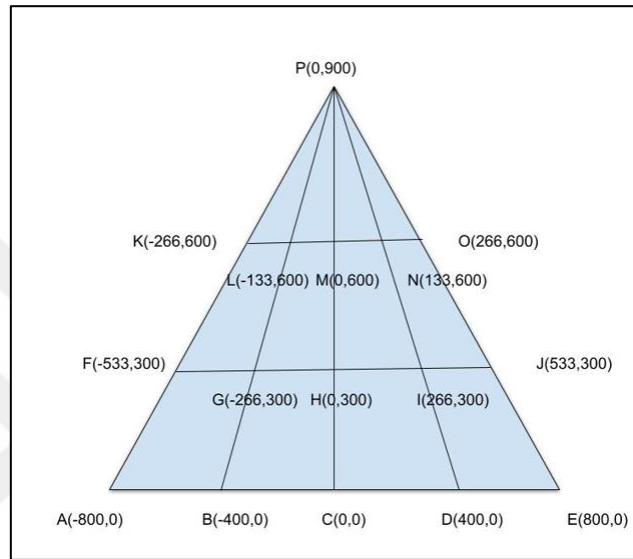
Özellik vektörü, bir nesne kategorisi veya arka plan olarak softmax fonksiyonu kullanılarak sınıflandırılır. RCNN'in en büyük dezavantajı, her bölge önerisinin CNN aracılığıyla işlenmesini gerektirmesi nedeniyle yavaş performansdır ki bu da önemli bir hesaplama maliyeti doğurur. RCNN'deki seyrek eğitim, sınıflandırıcının ve sınırlayıcı kutunun ayrı ayrı eğitilmesine neden olur; bu da verimliliği azaltır. Fast RCNN, Faster RCNN ve Mask RCNN gibi gelişmeler, paylaşılan CNN hesaplamalarını entegre ederek ve bölge önerilerinin entegrasyonunu geliştirerek bu sınırlamaları azaltır. Fast RCNN'de, tek bir CNN geçişi gerçekleştirildikten sonra tüm bölge önerileri ortak bir özellik haritasında (ROI Pooling) işlenir. Özellik haritası, aşağıdaki formül ile görüntüden tek seferde çıkarılır:

$$F = CNN(I) \quad (43)$$

Bölge önerileri bu haritadan çıkarılır. Her bölge, ROI Pooling ile sabit boyutlu bir vektöre dönüştürüldükten sonra, hem sınıflandırma hem de sınırlayıcı kutu regresyonu tek bir tam bağlı katman ile gerçekleştirilir. Her kareyi 60fps hızında işlerken, dedektör, çıkan verileri "multiObjectTracker" için gerekli bir nesneye derler. Bu çalışma, bir tespit yöntemi olarak RCNN kullanmaktadır. Toplayıcı kanal özellikleri (ACF) ve derin öğrenmeye dayalı bir nesne tespit yöntemi olan Faster RCNN, araç tespit sistemlerinin temelini oluşturmaktadır. Dedektörler, araç takibindeki etkilerini değerlendirmek için önceden eğitilmiş ağlarla kolayca değiştirilebilir. Önceden eğitilmiş bir araç dedektörü yüklendikten ve birden fazla nesne için bir izleyici yapılandırıldıktan sonra, dedektör video karesinin her birinde çalışır. İzleme sonuçlarıyla bir videoyu oynamadan önce, tespit sonuçları izleyiciye dâhil edilmelidir. Fast RCNN, tüm görüntüden konvolüsyonel özellikleri bir kez çıkararak ve İlgi Bölgesi (RoI) havuzlaması uygulayarak önerileri paylaşılan özellik haritasına yansıtarak tespit verimliliğini önemli ölçüde artırır. Her öneri, sınır kutusu regresyonu kullanılarak sınıflandırılır ve iyileştirilir.

3.7.4. Izgara Modeli

Her görüntü karesi, araç hızına göre dinamik olarak ayarlanan değişken ızgara Şekil 9'daki gibi bölünmektedir. Bu segmentasyon, konum takibi ve HMM a girdi sağlamak amacı ile kullanılmaktadır. Dikey ızgara, perspektifi yansıtacak şekilde bir yapıya sahiptir, bu şekilde uzak araçlara daha az ağırlık atmaktadır.



Şekil 10. Izgara Modelinin Koordinatları

Kaynak: Yazar tarafından oluşturulmuştur.

Şekil 10’da gösterildiği gibi, ızgara hattı 4x3 olarak seçilmiştir çünkü bir aracın görüntüde gösterilebilecek maksimum alanı yalnızca ekranın $\frac{1}{4}$ ’ünü kapsar. Böylece, daha büyük nesnelerin ızgaradan kaçmaması için ekran 4 dikey ve 3 yatay köşeye bölünmüştür. İkizkenar üçgenin özellikleri ve kesişim noktaları geometri açısından önemli konulardır. Üçgenin tabanı 1600 birim, yüksekliği ise 900 birimdir. Üçgenin A noktası, taban köşeleri B (-400, 0) ve C (0, 0) noktalarında bulunmaktadır. A noktasının koordinatları (-800, 0) olarak belirlenmiştir. Kesişim noktaları geometrik denklemlerle hesaplanmaktadır ve aşağıdaki tabloda verilmiştir.

Tablo 5. Izgara Modelinin Koordinatları

Nokta	x Hesabı	y Hesabı	Sonuç
A	0 - 800	0	(-800, 0)
B	0 - 400	0	(-400, 0)
C	0	0	(0, 0)
D	0 + 400	0	(400, 0)
E	0 + 800	0	(800, 0)
F	0 - 533	300	(-533, 300)
G	0 - 266	300	(-266, 300)
H	0	300	(0, 300)
I	0 + 266	300	(266, 300)
J	0 + 533	300	(533, 300)
K	0 - 266	600	(-266, 600)
L	0 - 133	600	(-133, 600)
M	0	600	(0, 600)
N	0 + 133	600	(133, 600)
O	0 + 266	600	(266, 600)
P	0+800	900	(800, 900)

Kaynak: Yazar tarafından oluşturulmuştur.

Bir nesnenin gerçek boyutunu, örneğin bir aracın uzunluğunu belirlemek için, nesnenin yakalanan görüntülerdeki piksel cinsinden uzunluğunu kaydetmek önemlidir. Örneğin, eğer nesne yüksekliği 300 piksel olarak ölçülüyorsa, bu değeri kaydedin. Eğer nesne 4 metre yüksekliğinde ve piksel boyutu 300 ise, ölçek 4m / 300px olarak hesaplanır ve bu da 0,0133 m/piksele karşılık gelir. A'dan P'ye kadar olan her nokta, hem ölçek faktörü hem de yerle ilgili hız faktörü ile çarpılır. Hız faktörü, yer hızı (1/yer hızı) ile ters oran olarak hesaplanır ve bu, aracın hızında bir artışın ızgaranın buna uygun olarak ayarlanmasını sağlar. Bu yöntem, ızgara değişim hesaplamalarının hassasiyetini artırır.

3.7.5. Saklı Markov Zinciri ve ANN Kararları

Markov Zinciri, zaman içinde bir dizi durum arasında geçiş yapan bir sistemi karakterize eden matematiksel bir modeldir. Gelecek duruma geçiş olasılığı önceki durumlardan bağımsız olmakla beraber yalnızca mevcut duruma bağlıdır. Bu özelliğe Markov Özelliği denir.

Markov zincirlerinin basitliği, dikkate değer bir avantajı temsil eder. Markov özelliği, gelecekteki durumun yalnızca mevcut duruma bağlı olduğunu ifade ettiğinden, karmaşık sistemlerin açık bir şekilde temsil edilmesini kolaylaştırır. Bu özellik, problemi boyutsal olarak azaltarak durum geçişlerinin analizini ve hesaplanmasını kolaylaştırır. Sonuç olarak, sınırlı sayıda parametre ile geliştirilebilen modeller, tahmin ve yorumlama açısından kolaylık sağlar.

Markov zincirlerinin bir diğer avantajı, çeşitli uygulamalar üzerindeki sağladığı esnekliktir. Telekomünikasyondaki kuyruk teorisi ve finans alanındaki hisse senedi piyasa trendi tahmini gibi birçok alanda sistemleri modelleyebilirler. Doğal dil işleme alanında, Markov zincirleri dil üretimi ve konuşma tanıma için kullanılmaktadır; bu model, mevcut duruma dayalı olarak bir sonraki kelimeyi veya fonemi tahmin etmeyi mümkün kılar. Bu uyarlanabilirlik, araştırmacıların ve uygulayıcıların Markov zinciri modellerini belirli gereksinimleri karşılamak ve çeşitli veri türlerini barındırmak için değiştirmelerini sağlar.

Ayrıca, Markov zincirleri, Markov Zincir Monte Carlo (MCMC) gibi etkili örnekleme tekniklerinin oluşturulmasını kolaylaştırır. Bu yöntemler, özellikle Bayesian istatistikleri ve makine öğrenimi uygulamalarında karmaşık olasılık dağılımlarını yaklaşık olarak hesaplamak için önemlidir. MCMC, yüksek boyutlu alanlarda etkili örnekleme yaparak, aksi takdirde hesaplaması zor olan posterior dağılımların tahmininde yardımcı olur.

Son olarak, son çıktı katmanı Markov geçiş matrisini modüle ederek, hızlı hareket altında tahmin doğruluğunu artırır.

3.8. Deney Kurulumu

Deneyler hem simüle edilmiş hem de gerçek dünya ortamlarında gerçekleştirilir. Kontrol testleri için gömülü LIDAR sensörleri ile birlikte yer doğruluğu pozisyonları sağlayan CARLA simülatörü kullanılmaktadır.

Car Learning to Act (CARLA), otonom araç kontrol sistemlerinin çalışmasını ve geliştirilmesini kolaylaştırmak için tasarlanmış açık kaynaklı bir araç simülatörüdür. Farklı makine öğrenme algoritmalarından oluşturulan otonom araçlar değerlendirilebilir, bu da çeşitli algoritmaların hızla uygulanmasını ve birçok bağlamda deney yapılmasını sağlar. Ayrıca, simüle edilmiş ortamda aydınlatma koşulları, yağış, yaya ve diğer araçlar gibi riskler de dâhil edilebilir.

Simülasyon platformu, sensör kümelerinin esnek bir şekilde yapılandırılmasını sağlar ve GPS koordinatları, hız, ivme ve çarpışmalar ile diğer ihlallere ilişkin kapsamlı veriler de dahil olmak üzere sürüş taktiklerini eğitmek için uygulanabilir sinyaller sunar. Hava durumu ve günün saati gibi çeşitli çevresel faktörler de belirtilebilir.

Sonuç olarak, CARLA'nın trafik senaryoları, otonom araç sistemlerinin geliştirilmesi için hayati öneme sahiptir. CARLA, araştırmacılara çeşitli çevreler, araç etkileşimleri ve hava koşullarını detaylı simülasyonlar gerçekleştirmek için araçlar sunarak, daha güvenli ve daha güvenilir otonom sürüş çözümlerinin geliştirilmesinde sorumlu ve verimli yenilik sağlanmasına yardımcı olur. Tamamen ücretsizdir ve açık kaynaklı yapısı sayesinde geliştiriciler tarafından özelleştirme imkânı sunar.

CARLA, kamera, LIDAR, radar, GNSS ve IMU gibi çeşitli sensör türlerini destekler. Sensör verilerini işleyebilme kapasitesine sahiptir ve otonom sürüş sistemlerinin test edilmesi mümkündür. Gerçek dünya testleri ile ilişkili olan riskler göz önünde bulundurulduğunda, değerlendirmeler tamamen sanal ve güvenli bir ortamda gerçekleştirilebilir. Kaza senaryoları gibi kritik ve tehlikeli durumlar etkili bir şekilde modellenebilir.

Modelde kullanılan görüntüler RGB (Kırmızı, Yeşil, Mavi) formatındadır. Kalan her sütun, bir araç veya dur işareti gibi tek bir nesne sınıfını temsil eden M-by-4 matrislerinden oluşan hücre vektörlerini içermelidir. Sütunlar, her biri [x, y, genişlik, yükseklik] formatında M adet sınırlayıcı kutu temsil eden 4 elemanlı double dizilerini

içerir. Format, sol üst köşenin konumunu ve ilgili görüntü içindeki sınırlayıcı kutunun boyutlarını tanımlar

Gerçek zamanlı uygulama, bir Intel i7 CPU üzerinde MATLAB'da Simulink kullanarak gerçekleştirilir ve video görüntüleri İstanbul'un O-7 otoyolunda kaydedilir. Gerçek zamanlı sistem, 60 FPS işlemektedir ve nesne konumları ızgaralara haritalanarak, CARLA'dan elde edilen LIDAR verileri kullanılarak yer doğruluğu ile karşılaştırılır.



SONUÇ

Deneyler, modelin işlevselliğini sağlamak amacıyla gerçek zamanlı olarak gerçekleştirilmiştir. Deneyler İstanbul’da yapılmış ve sonuçlar aşağıda sunulmuştur.

Modelin her çerçevesinin (x, y) koordinatları, ayarlanmış Kalman filtresi güncellemeleri olmadan R-CNN tarafından kaydedilmiştir ve ızgaraların konumları bilindiğinden, her pozisyona ızgaralar atanabilir.

Tablo 6. Hangi Aracın Hangi Izgaraya Ait Olduğu Tablosu, Doğrusal Kalman Filtresi

Frame	Koordinatı (x, y)	Araç Numarası	Izgara Numarası	LIDAR Koordinatı
1	Belirlenemedi			
2	Belirlenemedi			
3-29	Belirlenemedi			
30	[68.2801, -5.9767]	1	1	[66.381, -4.2587]
	[0.2980, -0.2589]	2	2	[31.2580, -0.1589]
31	[64.5202, -3.5626]	1	1	[66.7202, -3.4568]
	[31.6258, -0.07522]	2	2	[33.6398, -0.0258]
32	[62.9445, -4.3945]	1	1	[62.8445, -3.3945]
	[32.2565, -0.0566]	2	2	[30.2105, -0.0562]

Kaynak: Yazar tarafından oluşturulmuştur.

Tablo 7. Hangi Aracın Hangi Izgaraya Ait Olduğu Tablosu, Zaman Değişkenli Kalman Filtresi (10*10)

Frame	Koordinatı (x, y)	Araç Numarası	Izgara Numarası	LIDAR Koordinatı
1	[66.2581, -4.2587]	1	1	[66.381, -4.2587]
	[31.8280, -0.1589]	2	2	[31.2580, -0.1589]
2	[66.9554, -3.4568]	1	1	[66.7202, -3.4568]
	[33.5238, -0.0258]	2	2	[33.6398, -0.0258]
3	[62.254, -3.3945]	1	1	[62.8445, -3.3945]
	[30.2105, -0.8456]	2	2	[30.2105, -0.0562]

Kaynak: Yazar tarafından oluşturulmuştur.

Tablo 8. Hangi Aracın Hangi İzgaraya Ait Olduğu Tablosu, Zaman Değişkenli Kalman Filtresi (20*20).

Frame	Koordinatı (x, y)	Araç Numarası	İzgara Numarası	LIDAR Koordinatı
1	[68.0825, -3.8747]	1	1	[66.3081, -4.2587]
	[29.2580, -0.0589]	2	2	[31.2580, -0.1589]
2	[62.7202, -3.5625]	1	1	[66.7202, -3.4568]
	[29.6608, -0.0282]	2	2	[33.6398, -0.0258]
3	[60.8445, -3.3945]	1	1	[62.8445, -3.3945]
	[30.2105, -0.0562]	2	2	[30.2105, -0.0562]

Kaynak: Yazar tarafından oluşturulmuştur.

Dağıtımlara göre atanmış ızgara konumları tutarlı olsa da, zaman değişken olmayan modeller kullanıldığında yaklaşık 30 karelik (60 fps’da yaklaşık 0,5 s) sistematik bir gecikme gözlemlenmiştir. Bu fark güvenlik bağlamlarında son derece önemli olmasının nedeni sürücü reaksiyon süresi genellikle 0,75sn–1,5sn civarında olmasıdır (WHO, 2023). Algılama gecikmesinde yarım saniyelik bir azalma bu nedenle anlamlı bir güvenlik kazancı olarak değerlendirilmektedir. Ayrıca, zaman değişken çekirdekler kullanılarak hareket hâkimiyetiyle hızla ilerleyen araçlar daha doğru izlenmiş; bu çekirdekler ivmelenme ve frenleme değişikliklerine daha etkili biçimde uyum sağlamıştır. Neticede, dinamik sürüş senaryolarında, özellikle hızlanma, yavaşlama ve dönüş gibi farklı hareket rejimlerinin hızlı geçtiği durumlarda, modelin tepkiselliği ve güvenilirliği belirgin biçimde artmaktadır. Işık ve hava şartları esas olarak sinyal bozunları olarak etkili olup; buna ek olarak yüksek ISO’da görülen atış/okuma gürültüsü, yağmur damlacıklarının oluşturduğu lekeler gibi ölçüm gürültüleri, fotometrik kaymalar (parlaklık/kontrast/gamma değişimleri, gölgeler, parlama), kontrast bozulması (sis/pus) ve sıkıştırma artefaktları da dâhildir.

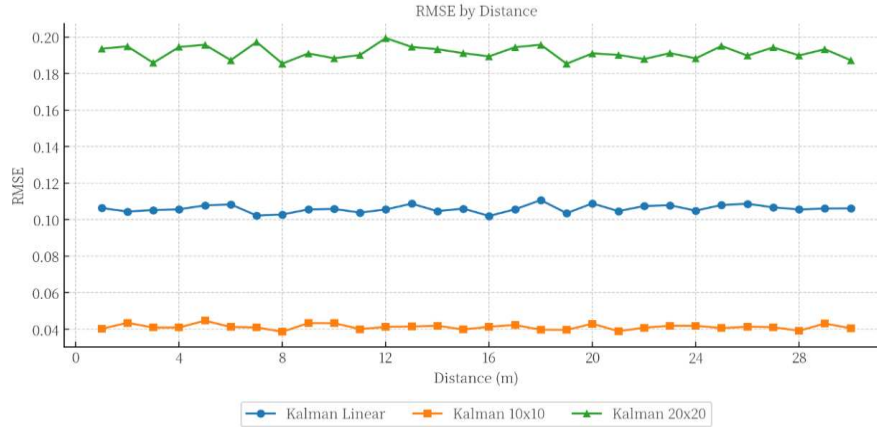
Sonuç olarak, küçük giriş bozulmaları bile çok büyük çıktı değişikliklerine yol açmaz; böylece algılama titreşimi ile hatalı sınırlama kutusu kaymaları azalır. Benzer biçimde, HMM (Gizli Markov Modeli) daha temiz durum geçişleri (ızgara atamaları) alır ve geçiş olasılığı tahminlerini iyileştirmektedir. İkisinin bir arada kullanılması özellikle yağmur, sis, parlama ve düşük ışık koşullarında karar gecikmesini azaltır; böylece güvenilirliği ve kararlılığı artırır.

Doğruluk karşılaştırmaları ayrıca zaman değişken Kalman filtrelerinin, orta-uzun menzillerde (>4 m) geleneksel yaklaşımları daha iyi yaptığına işaret eder; yakın mesafelerde ise piksel yoğunluğu ve ızgara duyarlılığı nedeniyle performans sınırlanabilir veya bazen düşebilir. Önemli olan, test edilen yapıların hiçbiri 30 m'nin ötesinde güvenilir algılama sürdürememiştir; bu durum, ticari dashcam çözümlerinin çözünürlüğünün doğal sınırlamalarıyla uyumludur. Bu nedenle, mesafe arttıkça zaman değişkenlikten elde edilen faydalar daha belirgin hâle gelmektedir.

Carla senaryosundan elde edilen ölçümler, her tespit edilen araç için AFC filtreleri kullanılarak toplanmış ve bu ölçümler, simülasyon motorunda bir metrelik aralıklarla LIDAR sensörü değerleriyle karşılaştırılmıştır. Bu tür karşılaştırmalar, modelin gerçek dünyadaki sensör ekosistemiyle uyumunu ve çok-kaynaklı entegrasyonun güvenilirliğini değerlendirirken kritik rol oynar; çünkü her sensörün hata özellikleri farklıdır ve bunlar birlikte çalıştığında elde edilen güvenilirlik, yalnızca tek bir sensörün başarısına bağlı olmaktan daha sağlamdır.

Atanan ızgaralar birbirine benzer olsa da, aralarında zaman farkı bulunmaktadır. Yakından incelendiğinde, aynı araç kimliği aynı koordinatlara ulaştığında neredeyse 30 karelik bir fark olduğu görülmektedir. Videoların 60 FPS hızında çekildiği göz önüne alındığında, saniyedeki 30 karelik bir fark, aracın tespit edilmesinde yaklaşık 0,5 saniyelik bir fark yaratmaktadır. Çalışmalar, sürücünün stresli bir durumda olmayan ortamlardaki ortalama tepki süresinin yaklaşık 0,75 ile 1,5 saniye olduğunu önermektedir (Broughton ve Perret, 1999), bu da 0.5 saniyelik bir sürenin tespit sırasında çok önemli bir etkiye sahip olduğunu göstermektedir. Bu gecikme, hızlı ivmelenme anında meydana gelir. Hızla hareket eden araçlar, ayarlanmış Kalman filtresi ile daha doğru bir şekilde tespit edilmektedir.

Zaman farkının yanı sıra, doğruluk puanları da birbirinden farklılık göstermektedir. Aşağıdaki tablo, bunlar arasındaki doğruluk puanlarını göstermektedir. Gerçek nesne konumu, kamera görüntülerinden tam olarak bilinemediğinden, CARLA simülatöründeki araçlara LIDAR sensörleri yerleştirilmiş ve gerçek konumu belirlenmiş, doğruluk puanı RMSE ile hesaplanmıştır.



Şekil 11. Farklı Kalman Filtreleri ile Elde Edilen Doğruluk

Kaynak: Yazar tarafından oluşturulmuştur.

Bu testte ölçülen tüm mesafeler boyunca hatayı en aza indirme açısından Kalman 10×10 yapılandırması en uygun seçenek görünmektedir. Hem daha büyük (20×20) hem de zaman-değişmez matris yaklaşımları daha yüksek hatalar üretmektedir; bu da senaryonuz için daha az uygun olduklarını gösterir, çünkü zaman değişkenlerine (ör. ivmelenme) ilişkin bileşenler eklendiğinde Q_k (gürültü değişkeni) artar. Nöral karar katmanının değerini değerlendirmek için, Yapay Sinir Ağı (ANN) tabanlı iyileştirme uygulanmış ve uygulanmamış durumlarda Gizli Markov Modeli (HMM) çıktıları karşılaştırılmıştır. Tablo 9’da gösterildiği gibi, ANN’in eklenmesi genel doğruluğu %92,3’ten %94,5’e yükselterek %2,2’lik bir kazanım sağlamıştır. Sayısal olarak mütevazı görünse de, bu iyileştirme yüksek hızlı senaryolarda küçük artışların bile kazaları önleyebileceği durumlarda anlamlıdır. ANN modülü, doğrusal olmayan uzamsal-zamansal bağımlılıkları modelleyip gürültüyü süzerek Markov çerçevesinin olasılıksal geçişlerini tamamlamıştır. Tüm hattın tespit süresi ve mAP puanları, modüller devredeyken ve devre dışıyken hesaplanmıştır. Testler tek bir GPU (RTX 3060/3070) üzerinde yürütülmüştür. Girdi boyutu 640×640 ’tır. Süreler ortalama $\pm$ standart sapma biçiminde raporlanmıştır. Ayarlanmış Kalman filtresi, geleneksel olanına göre daha yüksek doğruluk puanları elde etmesine rağmen, kameranın araçla arasındaki mesafe yakın olduğunda düşük performans sergilemektedir. Ayarlanmış Kalman filtresinin performansı, mesafe arttıkça daha yüksek olmaktadır. Her iki model de model parametreleri nedeniyle 30 metreden daha uzak araçları tespit edememektedir. Ne kadar aracın model tarafından tespit edildiği aşağıdaki denklemle hesaplanmıştır.

Tablo 9. Farklı Modellerdeki Tespit Süreleri

Model	Kalman	HMM (ms)	ANN (ms)	Tespit Süresi (ms)	mAP
RCNN + KALMAN doğrusal + HMM + ANN	61 ± 8	26 ± 7	8 ± 3	95 ± 19	0.78
RCNN + KALMAN zaman değişkenli+ HMM + ANN	85 ± 9	24 ± 6	9 ± 3	119 ± 21	0.79
RCNN + KALMAN time invariant + ANN	64 ± 8	-	8 ± 2	120 ± 22	0.74
RCNN + KALMAN doğrusal + ANN	84 ± 9	-	9 ± 4	93 ± 20	0.75
RCNN + KALMAN zaman değişkenli+ HMM	86 ± 11	25 ± 8	-	108 ± 18	0.77
RCNN + KALMAN doğrusal + HMM	67 ± 9	24 ± 7	-	107 ± 22	0.76
YOLOv8	-	-	-	70–120	0.73
YOLOv9	-	-	-	70–115	0.74
YOLOv10	-	-	-	90–140	0.75
Real-Time Transformer Detector	-	-	-	90–135	0.75

Kaynak: Yazar tarafından oluşturulmuştur.

Bu tablo, zamansal modüllerle güçlendirilmiş RCNN boru hatları ile YOLO taban çizgileri arasındaki hız-doğruluk karşılaştırmasını AP@0.5 kullanarak değerlendirmektedir. RCNN varyantları arasında, zaman-değişken Kalman + HMM + ANN içeren model, 119±21 ms’de en yüksek doğruluğa AP@0.5=0.79 ile ulaşmaktadır. Güçlü bir alternatif olan doğrusal Kalman + HMM + ANN yapılandırması ise belirgin biçimde daha hızlıdır (95±19 ms) ve yalnızca küçük bir doğruluk düşüşü (0.78) göstererek en iyi genel dengeyi sunar. RCNN seçeneği, orta düzey doğrulukla (0.75) zaman-değişkenli Kalman + ANN’dir (93±20 ms) dir.

Modül bazında bakıldığında, zaman-değişken Kalman, doğrusal Kalman güncellemelere kıyasla tutarlı biçimde yaklaşık +20 ms ekler ve yığınlar eşlendiğinde küçük ama sistematik bir AP artışı sağlamakta (yaklaşık +0.01). HMM, orta düzey bir maliyetle (26 ms) en büyük doğruluk artışını sağlamaktadır. YOLOv8s, YOLOv9s, Real-Time Transformer Detector ve YOLOv10s benzer gecikme aralıklarına

düşmekte (sırasıyla 70–120 ms ve 90–140 ms) ve AP@0.5 olarak 0.73 ile 0.75); RCNN, benzer gecikmelerde bu doğrulukları aşabilmektedir. Önemli bir nokta olarak, uçtan uca toplam algılama süresi (Detection Duration), her zaman listelenen modül sürelerinin toplamına eşit olmamaktadır. Bu durum; dedektör çıkarımı (inference), ön/son işleme, G/Ç (I/O) ve zamanlama mekanizmasından kaynaklanan ek yüklerle (overhead) işaret etmektedir. Değişkenlik modül başına ılımlı olsa da, toplamdaki dalgalanma (yaklaşık 18–22 ms), zamanlama varyasyonunda asıl etkenlerin boru hattı (pipeline) ve sistem genelindeki etkiler olduğunu düşündürmektedir. Çalışma zamanlarının adil bir şekilde karşılaştırılabilmesini sağlamak amacıyla tüm zamanlama ölçümleri, 640×640 giriş çözünürlüğünde, tek bir RTX 3060/3070 GPU üzerinde ve tüm model varyantları için aynı ön işleme ile NMS parametreleri kullanılarak, kare başına milisaniye (ms) cinsinden raporlanmıştır.

Bu sonuçlar; RCNN + Kalman boru hatlarının, özellikle de HMM ile birleştirilmiş zamanla değişen Kalman filtresi (ANN destekli veya desteksiz) varyantlarının, bu senaryoda hız ve doğruluk arasında en iyi dengeyi (trade-off) sunduğunu göstermektedir. Bu modeller, gerçek zamanlı çalışma kısıtlarını rahatlıkla karşılarken hata oranını en aza indirmektedir. Buna karşılık, bu çalışmada yapılandırılan YOLO tabanlı modellerin hem daha yavaş çalıştığı hem de daha düşük doğruluk sunduğu görülmüştür. mAP'teki küçük farkların (örneğin %38'e karşılık %39-40) deneysel değişkenlikten kaynaklanabileceği unutulmamalıdır; saçılım değerlerinin raporlanması, bu noktada istatistiksel anlamlılığı netleştirecektir. Ayrıca, bu sonuçlar modeller arasında kullanılan donanım ve değerlendirme protokollerinin eşdeğer olduğu varsayımına dayanmaktadır ve mAP metriği, bu çalışmadaki spesifik tanımına göre yorumlanmalıdır. Performans darboğazlarını tespit etmek ve gerçek zamanlı uygulanabilirliği daha net değerlendirmek amacıyla, kare başına toplam gecikme süresi şu bileşenlere ayrıştırılmıştır: dedektör çıkarımı, Kalman güncellemesi, HMM çıkarımı, ANN ile iyileştirme ve son işleme. Tüm zamanlama ölçümleri, N adet kare boyunca, aynı donanım, giriş çözünürlüğü ve ön işleme adımları kullanılarak gerçekleştirilmiştir.

Tablo 10, sistemin doğruluğunu çevrimdışı video analizi ve gerçek zamanlı uygulama senaryolarında karşılaştırmaktadır. Önerilen model, önceden etiketlenmiş videoların çevrimdışı analizinde %94,5 doğruluk oranına ulaşırken, İstanbul O-7

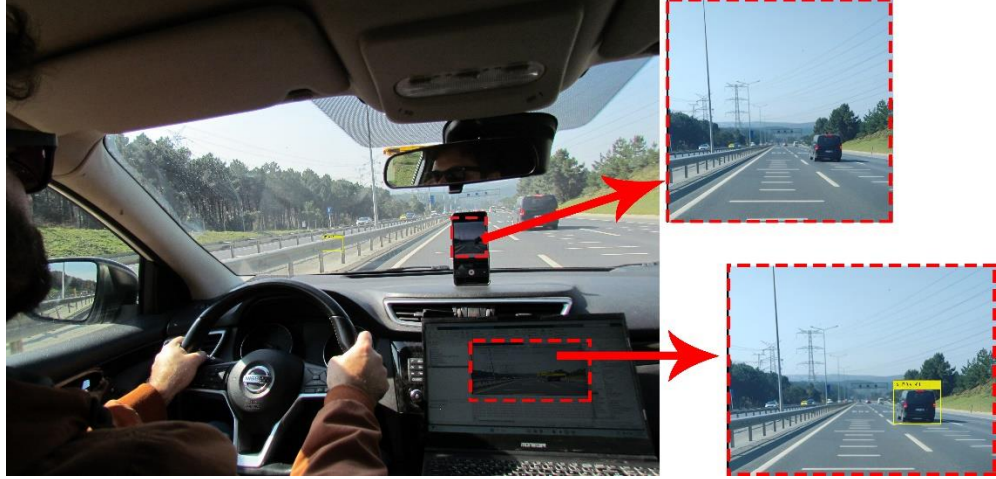
otoyolunda gerçekleştirilen canlı uygulamada bu oran %91,2 olarak ölçülmüştür. Doğrulukta bu hafif düşüşün, canlı sistemin getirdiği ek işlem yükleri (computational overhead) ve sahadaki çevresel değişkenliklerden kaynaklandığı düşünülmektedir.

Dijital görüntüleme ve video teknolojisinin ortaya çıkması, büyük hacimlerde görsel veriyi yorumlamak ve yönetmek için karmaşık analitik yöntemlerin geliştirilmesini zorunlu kılmıştır. Bu yöntemlerden biri, sistemlerin durumlar arasında geçişlerini modellemek için olasılık kriterleri ile yönetilen matematiksel bir çerçeve olan Markov Zinciridir.

Bu tezde, 4x3 ızgaralara bölünmüş video kare dizilerinin analizinde Markov Zincirlerinin kullanımını incelemekte ve bu işlemle ilişkili faydalar ve metodolojileri açıklamaktadır. Geleneksel bir video kare dizisinde, her bir kare bir Markov Zinciri içindeki bir durum olarak görülebilir. 4x3 ızgarayı kullanarak, her kare 12 farklı alana veya duruma bölünmekte, bu da video dizisindeki coğrafi ve zamansal değişikliklerin detaylı bir şekilde incelenmesine olanak tanımaktadır.

Markov zincirinin geçiş olasılıkları matrisinden, aynı kimliğe sahip bir aracın bir sonraki karede bulunma olasılığı hesaplanmaktadır. (Olasılık matrisinin ayrıntılı hesaplaması 3.5.4 bölümde açıklanmaktadır). Bu segmentasyon, karedeki çeşitli bileşenler arasındaki dinamik etkileşimlerin yakalanmasını olanaklı kılmakta ve zaman içinde hareket, nesne varlığı ve sahne ilerlemesini daha karmaşık bir şekilde anlamayı sağlamaktadır. ANN modeli Markov Zincirinden çıkan olasılık matrisi önceden eğitilmiş yapay zeka modeli tarafından çarpışma tahminlemesinde bulunmuştur. HMM zinciri performansı ölçümünde sınır değeri olarak 0.7 seçilmiştir.

Modelin uygulanabilirliğini test etmek amacıyla, gerçek zamanlı analizler de yapılmıştır. Görüntü ile gerçekleştirilen model, MATLAB ortamında Simulink kullanılarak gerçek zamanlı video ile de uygulanmıştır. Analizler, Intel(R) Core(TM) i7-9750H CPU @ 2.60GHz, 2601 MHz, 6 Çekirdek, 12 Mantıksal İşlemci ile gerçekleştirilmiştir.



Şekil 12. Gerçek Zamanlı Deneyin Görseli

Kaynak: Yazar tarafından oluşturulmuştur.

Model gerçek zamanlı olarak İstanbul O-7 karayolunda uygulanmış olup doğrulukları aşağıdaki tabloda gösterilmiştir.

Tablo 10. Video ve Gerçek Zamanlı Modellerin Doğruluk Oranları

	Video	Gerçek Zaman
Doğruluk	94.5	91.2

Kaynak: Yazar tarafından oluşturulmuştur.

Sunulan tablo, video kaydı ve gerçek zamanlı uygulamadan elde edilen veriler temelinde araç tespit sistemlerinin performansını iki farklı senaryoda değerlendirmektedir. Video kaydından yapılan analizde tespit edilen araçların yüzdesi %94,5 olarak belirtilmiştir. Bu, sistemin video görüntülerinin sonraki incelemesinde araçların büyük çoğunluğunu başarıyla tespit ettiğini göstermektedir.

Diğer yandan, gerçek zamanlı analizde tespit edilen araçların yüzdesi %91,2'ye düşmektedir. Bu azalma, gerçek zamanlı sistemlerin, veri işleme ve karar verme süreçlerini anında gerçekleştirme gerekliliğinden dolayı daha fazla zorlukla karşılaştığını ve bu nedenle performansta hafif bir düşüş beklentisinin söz konusu olduğunu göstermektedir. Çevresel faktörler, aydınlatma koşulları ve sistemin işleme gücü, gerçek zamanlı tespit performansını etkileyebilecek faktörlerden bazılarıdır.

Çarpışma doğruluğunu test etmek amacıyla, 67 vakadan oluşan çarpışma video derlemesi, videoların istenen formatta hazırlanmasının ardından test edilmiştir. Sonuçlar aşağıdaki tabloda verilmiştir. Bu videolar, YouTube'dan (URL-1) sağlanmış ve eğitim verilerindeki aynı özellikteki görüntülere dönüştürülmüştür.

TP = Çarpışmalar doğru bir şekilde çökme olarak tanımlandı

FN = Çarpışmalar yanlış bir şekilde çökme olmadığı şeklinde tanımlandı

$$\text{Hassasiyet} = \frac{\text{Doğru pozitiflerin sayısı}}{\text{Doğru pozitiflerin sayısı} + \text{Yanlış negatiflerin sayısı}} \quad (37)$$

Tablo 11. Uzak ve Yakın Çarpışmalardaki Performanslar

Kendi Çarpan Araçlar			Uzaktaki Çarpışmalar		
Doğru Pozitif (%)	Yanlış Negatif (%)	Hassasiyet (%)	Doğru Pozitif (%)	Yanlış Negatif (%)	Hassasiyet (%)
88	7	88	95	14	95

Kaynak: Yazar tarafından oluşturulmuştur.

Genel olarak, uzaktan bir çarpışmayı tespit etmek, Gerçek Pozitif oranı ve Hassasiyet açısından kendi kendine yapılan çarpışmalarla karşılaştırıldığında daha doğru bir şekilde tespit edilebilir. Ancak, ızgaralar arasındaki mesafe daha dar olduğundan ızgarayı değiştirme olasılığı daha yüksektir.

Diğer deney varyantları kombinasyonları ile yapılan sonuçlar aşağıdaki tablo da verilmiştir.

Tablo 12. Farklı Modellerin Performans Parametreleri

Model	Model Tipi	Hassasiyet (%)	Recall (%)	F1 (%)
Offline	Full (R-CNN + Kalman (TV) + HMM + ANN	79.4	88.0	83.47
Offline	HMM olmadan	77.6	86.1	81.62
Offline	ANN olmadan	75.8	85.9	80.53
Real-Time	R-CNN + Kalman (TV) + HMM + ANN	78.9	86.6	82.57
Real-Time	HMM olmadan	76.7	84.5	80.41
Real-Time	ANN olmadan	74.2	82.2	77.99
YOLO	YOLOV8	74.5	83.6	78.79
YOLO	YOLOV9	76.6	85.4	80.76
YOLO	YOLOV10	78.4	86.3	82.16
Real-Time Transformer Detector	Version 2	78.6	86.6	82.41

Kaynak: Yazar tarafından oluşturulmuştur.

Tüm ayarlarda, tam zamansal yığın (R-CNN + Kalman [TV] + HMM + ANN) kesinlik ve duyarlılık arasında en iyi dengeyi sağlar; en yüksek F1 değerlerine ulaşır: çevrimdışı %83,47, gerçek zamanlı %82,57. Zamansal bileşenlerin çıkarılması performansı tutarlı ve yorumlanabilir biçimde düşürür. Tam modele kıyasla HMM'nin çıkarılması, F1'i çevrimdışı 1,85 yüzde puan (pp) (%83,47 → %81,62) ve gerçek zamanlı 2,16 pp (%82,57 → %80,41) azaltır. ANN'nin çıkarılması daha büyük bir etkiye sahiptir: çevrimdışı -2,94 pp (%80,53'e) ve gerçek zamanlı -4,58 pp (%77,99'a). Bu farklar, özellikle gerçek zaman kısıtları altında, zamansal modüller içinde en büyük artımsal kazanımı ANN'nin sağladığını göstermektedir.

Kesinlik-duyarlılık ayrımı da aynı eğilimleri yansıtır. Tam modele kıyasla HMM'nin kaldırılması, her iki ortamda da kesinlik/duyarlılığı yaklaşık 2 pp azaltır (Çevrimdışı: -1,8/-1,9; Gerçek Zaman: -2,2/-2,1). ANN'nin kaldırılması ise daha büyük düşüşe yol açar (Çevrimdışı: -3,6/-2,1; Gerçek Zaman: -4,7/-4,4). Buna göre ANN, başlıca kesinliği artırırken duyarlılığı da iyileştirir; HMM ise her ikisine daha

küçük ama tutarlı katkı sağlar. Çevrimdışıdan Gerçek Zaman'a geçiş, tüm varyantlar için mütevazı bir genel performans kaybı doğurur; tam modelde F1 0,90 pp azalır (kesinlik -0,5, duyarlılık -1,4). En büyük ortama bağlı bozulma, ANN yokken gözlenir (F1 -2,54 pp; duyarlılık -3,7 pp); bu da öğrenilmiş iyileştirmenin, gerçek zamanlı çalışmada duyarlılık cezasını gidermede özellikle önemli olduğunu düşündürür.

Modelin, güncel alternatiflerle kıyaslamak amacıyla, YOLOv8'den v10'a kadar olan modellerle ve gerçek zamanlı bir Transformer dedektörünün (Sürüm 2) resmî uygulamalarıyla bir karşılaştırma yapılmıştır. Tüm değerlendirmeler, standartlaştırılmış ayarlar (IoU eşiği 0.5, varsayılan NMS, yığın boyutu 1) kullanılarak adil bir şekilde gerçekleştirilmiştir.

Bu kıyaslamada elde edilen sonuçlar şöyledir:

- Gerçek zamanlı Transformer (Sürüm 2): %82,41 F1 skoru (%78,6 kesinlik, %86,6 duyarlılık)
- YOLOv10: %82,16 F1 skoru (%78,4 kesinlik, %86,3 duyarlılık)
- YOLOv9: %80,76 F1 skoru
- YOLOv8: %78,79 F1 skoru

Özetle, bu veri kümesinde zamansal hiyerarşi kümülatiftir: marjinal faydada ANN > HMM; ikisinin birlikte kullanımı en güçlü F1'i verir. Gerçek zamanlı çalışma, küçük ama tutarlı kayıplar getirir—bunlar kesinlikten çok duyarlılık kaynaklıdır—ve her iki zamansal modülün korunmasıyla en iyi telafi edilir. Bağlam için, tek-aşamalı taban çizgileri güçlüdür ancak biraz geridedir: YOLOv10 %82,16 F1'e ve gerçek zamanlı transformer dedektörü %82,41 F1'e ulaşırken, her ikisi de tam zamansal yığının gerçek zamanlı değerinin (%82,57) gerisinde kalır; bu da zamansal modellemenin faydasını vurgulamaktadır.

Tablo 12, test edilen tüm modellerin karşılaştırmalı performans analizini sunar; Kalman süzgeci sonuçlarını, ANN ile zenginleştirilmiş çerçeveleri, çevrimdışı video ve gerçek zamanlı koşulları birleştirir. ANN ile güçlendirilmiş ve çevrimdışı video modelleri her ikisi de en yüksek doğruluğa (%94,5) ulaşırken, mesafe-temelli çarpışma tespiti en yüksek duyarlılığı (%95) kaydetmiştir. Buna karşılık, geleneksel Kalman

filtresi en düşük performansı göstermiştir (%86,6); bu da dinamik ve doğrusal olmayan trafik senaryolarındaki sınırlamalarının altını çizmektedir.

Sürüş, karmaşık bir bilişsel ve motor beceri seti, yüksek derecede dikkat ve hızlı karar verme süreçleri gerektiren bir eylemdir. Sürücünün tepki süresi, bu süreçlerin başarısında hayati bir rol oynamaktadır. Tepki süresi, bir uyarıcının algılanmasından uygun bir eylemin başlatılmasına kadar geçen süreyi ifade eder ve doğrudan sürüş güvenliğini etkiler. Önerilen model her 30. kareyi aldığından, tepki süresindeki iyileşme 0.5 saniye olacaktır. Genel olarak kabul edilen görüşe göre, bir sürücünün bir tehlikeyi fark etmesi ve tepki vermeye başlaması için geçen süre yaklaşık 1.5 ile 2 saniye arasında değişmektedir. Bu süre zarfında sürücü görsel veya işitsel bir uyarıcıyı algılar, bu uyarıcıyı işler ve bir yanıt kararı alır (Dalvi, 2004).

Kaza tepki süresi, sürücünün tehlikeyi fark etmesi, karar vermesi ve frene basma süresidir. Bu süre kişiden kişiye değişiklik gösterse de literatürde ortalama 1.5 saniye olarak kabul edilmektedir (Patel, 2022). Bu bağlamda, tehlikeyi tespit ederken 0.5 saniyelik bir artışın kaza olasılığını azaltacağı söylenebilir.

Diğer yandan, LIDAR sensörlerinin maliyetleri yüksek olduğu ve otonom araçların pahalı olduğu dikkate alındığında, bu model yalnızca bir dash cam ve entegre bir program kullanarak yüksek doğruluk oranıyla çarpışmaları tespit edebilmektedir. GPU teknolojisinin gelişimiyle modelin, işlem ve eğitim süresini azaltmak için bir GPU üzerinde çalıştırılması mümkündür.

Bu araştırma, Fast R-CNN, Kalman Filtreleri ve Markov Zincirlerinin güçlü yönlerini bir araya getirerek, yapay sinir ağı ile güçlendirilmiş gerçek zamanlı araç tespit ve takip eden etkili bir hibrit modeli sunmaktadır. Modelin mimarisi, otonom sürüşteki önemli zorluklara: trafik anormalliklerini tespit etme, tepki süresini azaltma ve dinamik koşullar altında tahmin doğruluğunu artırma konularını ele almaktadır. ANN'nin Markov modeli ile entegrasyonu, tespit hassasiyetini %2.2 artırmayı başarmış ve bu da kaza önlemeyi kritik hale getiren 0.5 saniyelik bir tepki süresi iyileştirmesine karşılık gelmektedir. Gerçek dünya ve simüle edilmiş testler, modelin yüksek doğruluğunu ve pratik uygulanabilirliğini, özellikle kentsel ortamlarda doğrulamaktadır. GPU yetenekleri ve gerçek zamanlı işlem gücü geliştikçe, bu tür

sistemlerin verimliliği ve güvenilirliği artması beklenmekte, bu da onları yeni nesil otonom sürüş ve trafik yönetim uygulamaları için değerli kılmaktadır.

Temel olarak bu tez üç ana noktada literatüre katkı sunmaktadır.

Filtreleme Mekanizmalarının Etkisi: Zaman-değişken çekirdeklerin eklenmesi, özellikle hızlı akan trafik senaryolarında algılama gecikmesini anlamlı ölçüde azaltır. Ancak kısa mesafelerde bu iyileşme zayıflar; bu da yoğun kentsel ortamlarda hibrit modellerin veya uyarlanabilir eşiklerin gerekebileceğini düşündürür.

Nöral Karar Katmanlarının Rolü: ANN entegrasyonundan elde edilen mütevazı kazanımlar bile güvenlik bağlamlarında kritiktir; algılamadaki her saniye kesri önem taşır. Nöral modülün gürültüye karşı dayanıklılığı, farklı aydınlatma ve hava koşullarında güvenilirliği artırır.

Operasyonel Sağlamlık: Çevrimdışı ve gerçek zamanlı analiz arasındaki fark, doğruluk ile hesaplama yapılabirliği arasındaki ödünleşimi öne çıkarır. Gerçek zamanlı mimarilerin, olası GPU hızlandırması veya hafif derin öğrenme modelleriyle optimize edilmesi, gelecekteki çalışmalar için kritik bir yöndür.

Önerilen sistem, çevrimdışı video analizinde genel algılama doğruluğu olarak %94,5'e, gerçek zamanlı dağıtımda ise %91,2'ye ulaşmıştır. Düzeltme faktörü olarak ANN'in eklenmesi algılamayı %2,2 iyileştirmiş; zaman-değişken Kalman süzgeci ise geleneksel yöntemlere göre algılama gecikmelerini yaklaşık 0,5 s azaltmıştır. Pahalı LIDAR sensörleri yerine yalnızca kamera girdisine dayanması sayesinde çerçeve, performansla birlikte maliyet etkinliğini dengelerken mevcut araç içi kamera teknolojisiyle uyumlu bir algılama menzilini (~30 m) korur.

Önemli bir diğer sınırlama, ticari araç kameralarının çözünürlüğü ve görüş alanı tarafından belirlenen yaklaşık 30 m'lik etkili algılama menzildir. Bu mesafe, hızların daha düşük ve fren mesafelerinin kısa olduğu kentsel sürüş için yeterli olabilir; ancak araçların çok daha uzun durma mesafelerine ihtiyaç duyduğu yüksek hızlı otoyol koşulları için yetersizdir. Örneğin 120 km/s hızda 30 m menzil bir saniyeden daha az uyarı süresine karşılık gelir; bu da yeterli güvenlik payı sağlamayabilir. Bu kısıtı gidermek, sensör düzeyinde iyileştirmeler (örn. daha yüksek çözünürlük, çoklu kamera kurulumu, stereo görüş) veya süper-çözünürlük modülleri ve öngörücü yörünge dışsaptırımı gibi algoritmik geliştirmeler gerektirecektir. Gelecek çalışmalar,

çerçevenin uygulanabilirliğini kentsel bağlamdan yüksek hızlı otoyol ortamlarına genişletmek için bu yönleri araştırmalıdır.

Bu güçlü yönlerle karşı sınırlamalar sürmektedir: araç/sürücü kaynaklı tek taraflı kazalarda düşen algılama doğruluğu, kısa mesafelerde performans bozulması ve ölçeklenebilirliği kısıtlayabilecek MATLAB Simulink ile Intel i7 sınıfı donanıma bağımlılık. Bu zorlukların ele alınması, özellikle düşük maliyetli gömülü donanımın optimize edilmesinde, devam eden çalışmaların temelini oluşturmaktadır.

Ek Bir Sınırlama ve Olası Çözümler:

Önerilen çerçevenin bir başka önemli sınırlaması, ticari araç kameralarının çözünürlüğü ve görüş alanı tarafından doğal olarak sınırlandırılan yaklaşık 30 m'lik etkili algılama menzildir. Bu menzil, tipik olarak hızların 50 km/s'nin altında kaldığı ve fren mesafelerinin 20–30 m içinde olduğu kentsel sürüş için kabul edilebilir olsa da, otoyol senaryoları için yetersizdir. 120 km/s hızda bir araç saniyede 33 m kat eder; dolayısıyla 30 m algılama menzili bir saniyeden kısa bir uyarı süresi sağlar. Bu hızlarda güvenli durma mesafeleri çoğu zaman 100 m'yi aşar; bu da sistem yeteneği ile gerçek dünya gereksinimleri arasında önemli bir boşluk bırakarak çelişki yaratır. Bu sınırlamanın aşılması, donanım iyileştirmeleri (daha yüksek çözünürlüklü veya geniş açılı kameralar, stereo görüş, çoklu kamera dizileri) ya da algoritmik geliştirmeler (süper-çözünürlük, çok-kare füzyonu, yörünge dışsaptırımı) gerektirir. Gelecek çalışmalar, modelin ölçeklenebilirliğini kentsel trafik izlemenin ötesinde yüksek hızlı otoyol güvenliği uygulamalarına doğru genişletmek için bu yönleri araştırmalıdır.

KAYNAKÇA

- Alam, M. K., Ahmed, A., Salih, R., Al Asmari, A. F. S., Khan, M. A., Mustafa, N., Mursaleen, M., ve Islam, S. (2023). Faster RCNN based robust vehicle detection algorithm for identifying and classifying vehicles. *Journal of Real-Time Image Processing*, 20(5), 93.
- Anderson, J. M., Nidhi, K., Stanley, K. D., Sorensen, P., Samaras, C., ve Oluwatola, O. A. (2014). *Autonomous vehicle technology: A guide for policymakers*. Rand Corporation.
- Beiker, S. A. (2012). Legal aspects of autonomous driving. *Santa Clara L. Rev.*, 52, 1145.
- Bochkovski, A., Wang, C.-Y., ve Liao, H.-Y. M. (2020). Yolov4: Optimal speed and accuracy of object detection. *ArXiv Preprint ArXiv:2004.10934*.
- Cao, J., Cholakkal, H., Anwer, R. M., Khan, F. S., Pang, Y., ve Shao, L. (2020). D2det: Towards high quality object detection and instance segmentation. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11485–11494.
- Carranza-García, M., Torres-Mateo, J., Lara-Benítez, P., & García-Gutiérrez, J. (2021). On the Performance of One-Stage and Two-Stage Object Detectors in Autonomous Vehicles Using Camera Data. *Remote Sensing*, 13(1), 89. <https://doi.org/10.3390/rs13010089>
- Chaabane, M., Trabelsi, A., Blanchard, N., ve Beveridge, R. (2020). Looking ahead: Anticipating pedestrians crossing with future frames prediction. *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2297–2306.
- Cui, Y., Chen, R., Chu, W., Chen, L., Tian, D., Li, Y., ve Cao, D. (2021). Deep learning for image and point cloud fusion in autonomous driving: A review. *IEEE Transactions on Intelligent Transportation Systems*, 23(2), 722–739.

- Dalal, N., ve Triggs, B. (2005). Histograms of oriented gradients for human detection. *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, 1, 886–893.
- De Brabandere, B., Neven, D., ve Van Gool, L. (2017). Semantic instance segmentation for autonomous driving. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 7–9.
- Dong, X., Zhuang, B., Mao, Y., ve Liu, L. (2021). Radar camera fusion via representation learning in autonomous driving. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1672–1681.
- Feng, D., Haase-Schütz, C., Rosenbaum, L., Hertlein, H., Glaeser, C., Timm, F., Wiesbeck, W., ve Dietmayer, K. (2020). Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges. *IEEE Transactions on Intelligent Transportation Systems*, 22(3), 1341–1360.
- Galvao, L. G., Abbod, M., Kalganova, T., Palade, V., ve Huda, M. N. (2021). Pedestrian and Vehicle Detection in Autonomous Vehicle Perception Systems—A Review. *Sensors*, 21(21). <https://doi.org/10.3390/s21217267>
- Garnett, N., Silberstein, S., Oron, S., Fetaya, E., Verner, U., Ayash, A., Goldner, V., Cohen, R., Horn, K., ve Levi, D. (2017). Real-time category-based and general obstacle detection for autonomous driving. *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 198–205.
- Girshick, R., Donahue, J., Darrell, T., ve Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 580–587.
- Goel, K., Srinivasan, P., Tariq, S., ve Philbin, J. (2021). Quadronet: Multi-task learning for real-time semantic depth aware instance segmentation. *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 315–324.
- Gupta, M., Miglani, H., Deo, P., ve Barhatte, A. (2023). Real-time traffic control and monitoring. *E-Prime-Advances in Electrical Engineering, Electronics and Energy*, 5, 100211.

- Ignatious, H. A., El-Sayed, H., ve Kulkarni, P. (2023). Multilevel data and decision fusion using heterogeneous sensory data for autonomous vehicles. *Remote Sensing*, 15(9), 2256.
- Jiang, Z., Zhao, L., Li, S., ve Jia, Y. (2020). Real-time object detection method based on improved YOLOv4-tiny. *ArXiv Preprint ArXiv:2011.04244*.
- Khan, M. A., Park, H., ve Chae, J. (2023). A lightweight convolutional neural network (CNN) architecture for traffic sign recognition in urban road networks. *Electronics*, 12(8), 1802.
- Knoedler, L., Salmi, C., Zhu, H., Brito, B., ve Alonso-Mora, J. (2022). Improving pedestrian prediction models with self-supervised continual learning. *IEEE Robotics and Automation Letters*, 7(2), 4781–4788.
- Kukreja, V., Kumar, D., Kaur, A., ve others. (2020). GAN-based synthetic data augmentation for increased CNN performance in Vehicle Number Plate Recognition. *2020 4th International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, 1190–1195.
- Kurniawan, I. T., ve Trilaksono, B. R. (2023). ClusterFusion: Leveraging radar spatial features for radar-camera 3D object detection in autonomous vehicles. *IEEE Access*, 11, 121511–121528.
- Liang, S., ve Zhang, X. (2024). An Object Detection System for Automatic Driving: MEC-YOLO Based on "Cloud-Edge-End". *2024 6th International Conference on Natural Language Processing (ICNLP)*, 536–541.
- Liang, T., Bao, H., Pan, W., ve Pan, F. (2022). ALODAD: An anchor-free lightweight object detector for autonomous driving. *IEEE Access*, 10, 40701–40714.
- Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., ve Berg, A. C. (2016). Ssd: Single shot multibox detector. *Computer Vision--ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11--14, 2016, Proceedings, Part I 14*, 21–37.
- Lu, X., Li, Q., Li, B., ve Yan, J. (2020). Mimicdet: Bridging the gap between one-stage and two-stage object detection. *Computer Vision--ECCV 2020: 16th European Conference, Glasgow, UK, August 23--28, 2020, Proceedings, Part XIV 16*, 541–557.

- Marcuzzi, R., Nunes, L., Wiesmann, L., Behley, J., ve Stachniss, C. (2023). Mask-based panoptic lidar segmentation for autonomous driving. *IEEE Robotics and Automation Letters*, 8(2), 1141–1148.
- Moayed, Z. (2020). *Automated Multiview Safety Analysis At Complex Road Intersections*. Auckland University Of Technology.
- Mohapatra, S., Yogamani, S., Kumar, V. R., Milz, S., Gotzig, H., ve Mäder, P. (2025). Lidar-bevmtn: Real-time lidar bird's-eye view multi-task perception network for autonomous driving. *IEEE Transactions on Intelligent Transportation Systems*.
- Mozaffari, S., Al-Jarrah, O. Y., Dianati, M., Jennings, P., ve Mouzakitis, A. (2020). Deep learning-based vehicle behavior prediction for autonomous driving applications: A review. *IEEE Transactions on Intelligent Transportation Systems*, 23(1), 33–47.
- Neogi, S., Hoy, M., Dang, K., Yu, H., ve Dauwels, J. (2020). Context model for pedestrian intention prediction using factored latent-dynamic conditional random fields. *IEEE Transactions on Intelligent Transportation Systems*, 22(11), 6821–6832.
- Patel, J. (2022). *Scenario Generation For Vehicles Using Deep Learning*. KTH Royal Institute of Technology.
- Philip, B., Updike, P., ve Perona, P. (2022). *Caltech Cars 2001*. <https://doi.org/https://doi.org/10.22002/D1.20085>
- Ranney, T. A., Garrott, W. R., ve Goodman, M. J. (2001). *NHTSA driver distraction research: Past, present, and future*.
- Rashed, H., Ramzy, M., Vaquero, V., El Sallab, A., Sistu, G., ve Yogamani, S. (2019). Fusemodnet: Real-time camera and lidar based moving object detection for robust low-light autonomous driving. *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, 0.
- Ravindran, R., Santora, M. J., ve Jamali, M. M. (2022). Camera, LiDAR, and radar sensor fusion based on Bayesian neural network (CLR-BNN). *IEEE Sensors Journal*, 22(7), 6964–6974.

- Razali, H., Mordan, T., ve Alahi, A. (2021). Pedestrian intention prediction: A convolutional bottom-up multi-task approach. *Transportation Research Part C: Emerging Technologies*, 130, 103259.
- Redmon, J., Divvala, S., Girshick, R., ve Farhadi, A. (2016). You only look once: Unified, real-time object detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 779–788.
- Ren, M., ve Zemel, R. S. (2017). End-to-end instance segmentation with recurrent attention. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 6656–6664.
- Rezaei, S., Gbadegoye, J., Masoud, N., ve Khojandi, A. (2023). A deep learning-based approach for vehicle motion prediction in autonomous driving. *2023 International Conference on Control, Automation and Diagnosis (ICCAD)*, 1–6.
- Rubaiyat, A. H. M., Fallah, Y., Li, X., Bansal, G., ve Infotechnology, T. (2018). Multi-sensor data fusion for vehicle detection in autonomous vehicle applications. *Electronic Imaging*, 30, 1–6.
- Shahian Jahromi, B., Tulabandhula, T., ve Cetin, S. (2019). Real-time hybrid multi-sensor fusion framework for perception in autonomous vehicles. *Sensors*, 19(20), 4357.
- Shi, X., Wong, Y. D., Chai, C., ve Li, M. Z.-F. (2020). An automated machine learning (AutoML) method of risk prediction for decision-making of autonomous vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 22(11), 7145–7154.
- Soviany, P., ve Tudor Ionescu, R. (2018). Frustratingly easy trade-off optimization between single-stage and two-stage deep object detectors. *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, 0.
- Tahir, N. U. A., Zhang, Z., Asim, M., Chen, J., ve ELAffendi, M. (2024). Object detection in autonomous vehicles under adverse weather: A review of traditional and deep learning approaches. *Algorithms*, 17(3), 103.

- Tan, M., Pang, R., ve Le, Q. V. (2020). Efficientdet: Scalable and efficient object detection. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10781–10790.
- Weber, M., ve Perona, P. (2022). *Caltech Cars 1999*. <https://doi.org/10.22002/D1.20084>
- Wen, L.-H., ve Jo, K.-H. (2021). Fast and accurate 3D object detection for lidar-camera-based autonomous vehicles using one shared voxel-based backbone. *IEEE Access*, 9, 22080–22089.
- Wirthmüller, F., Schlechtriemen, J., Hipp, J., ve Reichert, M. (2020). Teaching vehicles to anticipate: A systematic study on probabilistic behavior prediction using large data sets. *IEEE Transactions on Intelligent Transportation Systems*, 22(11), 7129–7144.
- World Health Organization. *Global Status Report on Road Safety 2023: Time for a New Normal*; World Health Organization: Geneva, Switzerland, 2023. Available online: <https://www.who.int/teams/social-determinants-of-health/safety-and-mobility/global-status-report-on-road-safety-2023> (accessed on 07.05.2025).
- Yang, X., Wang, S., Zhu, Y., Zhou, D., ve Li, T. (2024). Context CVGN: A conditional multimodal trajectory prediction network based on scene semantic modeling. *Information Sciences*, 666, 120433.
- Yin, C., Tang, J., Yuan, T., Xu, Z., ve Wang, Y. (2021). Bridging the gap between semantic segmentation and instance segmentation. *IEEE Transactions on Multimedia*, 24, 4183–4196.
- Zhang, C., ve Berger, C. (2023). Pedestrian behavior prediction using deep learning methods for urban scenarios: A review. *IEEE Transactions on Intelligent Transportation Systems*, 24(10), 10279–10301.
- Zhang, Y., Wang, S., Chen, B., Cao, J., ve Huang, Z. (2019). Trafficgan: Network-scale deep traffic prediction with generative adversarial nets. *IEEE Transactions on Intelligent Transportation Systems*, 22(1), 219–230.

Zhao, R., Tang, S., Supeni, E. E. Bin, Rahim, S. B. A., ve Fan, L. (2024). A Review of Object Detection in Traffic Scenes Based on Deep Learning. *Applied Mathematics and Nonlinear Sciences*, 9(1). <https://doi.org/10.2478/amns-2024-0322>



TOPLUMSAL KATKI

Otonom araç teknolojileri, günümüz ulaşım sistemlerinde devrim niteliğinde değişimlere yol açmaktadır. Bu araçlar, insan müdahalesi olmadan hareket edebilme yeteneğiyle, trafik güvenliğini artırmak ve kaza oranlarını azaltmak gibi önemli faydalar sunmaktadır. Otonom araçlarda çarpışma önleme sistemleri, bu teknolojinin en kritik bileşenlerinden biridir ve topluma birçok farklı açıdan katkıda bulunmaktadır. Çarpışma önleme sistemleri, genellikle çeşitli sensörler, kameralar ve yapay zeka algoritmaları kullanarak, olası çarpışmaları önlemek amacıyla araçların çevresini sürekli izler. Bu sistemler, tehlikeyi algıladıklarında sürücüyü uyarabilir ve gerektiğinde otomatik frenleme veya yön değiştirme gibi önlemler alarak çarpışmaları engelleyebilir. Otonom araçların en önemli katkılarından biri, trafik kazalarının önemli ölçüde azaltılmasıdır. Dünya Sağlık Örgütü'ne göre, her yıl yaklaşık 1.3 milyon insan trafik kazalarında hayatını kaybetmektedir. Otonom araçlar, insan hatası kaynaklı hataları minimize ederek, kaza oranlarını azaltacak potansiyele sahiptir. Örneğin, yapılan araştırmalara göre, otonom araçların yaygınlaşmasıyla birlikte görüş açısı sınırlı durumlarda bile çarpışma riskinin %90 oranında azalabileceği öngörülmektedir. Trafik kazalarının azaltılması, ekonomik açıdan da toplum üzerindeki yükü hafifletmektedir. Kazalardan kaynaklanan sağlık harcamaları, trafik tıkanıklığı ve hasar maliyetleri gibi unsurlar, yıllık milyarlarca dolarlık kayıplara neden olmaktadır. Otonom araçlar sayesinde bu maliyetlerde önemli bir azalma beklenmektedir. Ayrıca, potansiyel olarak artırılan yol güvenliği, sigorta primlerinin düşmesine yol açabilir. Otonom araçlar, sürdürülebilir ulaşım hedeflerine de katkıda bulunmaktadır. Çarpışma önleme sistemleri, araçların daha verimli bir şekilde hareket etmesini sağlayarak, yakıt tüketimini ve karbondioksit salınımını azaltabilir. Bunun yanı sıra, toplu taşımaya yönelik uygulamalar ile birleştirildiğinde, şehir içi ulaşımında genel trafik yoğunluğunu azaltarak çevresel etkileri minimize edebilir. Her ne kadar otonom araçlar birçok fayda sağlasa da, toplumdaki kabul düzeyi de önemli bir faktördür. Çarpışma önleme sistemlerinin etkinliği konusunda yapılan kampanyalar ve eğitimler, kullanıcıların bu teknolojilere olan güvenini artırabilir. Toplumun bu konuda bilinçlenmesi, otonom araçların benimsenmesi için kritik bir aşamadır. Otonom araçlarda çarpışma önleme sistemleri, topluma trafik güvenliği, ekonomik faydalar ve çevresel etkilerin

iyileştirilmesi gibi birçok olumlu katkı sunmaktadır. Ancak, bu sistemlerin etkinliğinin artırılması ve toplumsal kabulünün sağlanması için sürekli bir çaba gerekmektedir. Gelecekte, otonom teknolojilerin daha geniş bir şekilde benimsenmesiyle birlikte, trafik kazalarının büyük ölçüde azalması ve ulaşım sistemlerinin daha güvenli hale gelmesi beklenmektedir.

