T.C.
İSTANBUL SABAHATTİN ZAİM ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI
BİLGİSAYAR BİLİMLERİ VE MÜHENDİSLİĞİ BİLİM DALI

DERİN ÖĞRENME ALGORİTMALARI İLE OTONOM ARAÇLARIN GÖRME ALGILARININ GELİŞTİRİLMESİNDE YENİ BİR YAKLAŞIM

YÜKSEK LİSANS TEZİ

Mustafa TOKAT

İstanbul
Haziran, 2025

T.C.
İSTANBUL SABAHATTİN ZAİM ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI
BİLGİSAYAR BİLİMLERİ VE MÜHENDİSLİĞİ BİLİM DALI

DERİN ÖĞRENME ALGORİTMALARI İLE OTONOM ARAÇLARIN GÖRME ALGILARININ GELİŞTİRİLMESİNDE YENİ BİR YAKLAŞIM

YÜKSEK LİSANS TEZİ

Mustafa TOKAT

TEZ DANIŞMANI
Dr. Öğrt. Üyesi Zahra ELMI

İstanbul
Haziran, 2025

Lisansüstü Eğitim Enstitüsü Müdürlüğüne,

Bu çalışma, jürimiz tarafından Bilgisayar Mühendisliği Anabilim Dalı, Bilgisayar Bilimleri ve Mühendisliği Bilim Dalında YÜKSEK LİSANS TEZİ olarak kabul edilmiştir.

Danışman	Dr. Öğrt. Üyesi Zahra ELMI	……
Üye	Doç. Dr. Öğrt. Üyesi Emir SEYYEDABBASİ	……
Üye	Dr. Öğrt. Üyesi Şengül BAYRAK HAYTA	……

Onay

Yukarıdaki imzaların, adı geçen öğretim üyelerine ait olduğunu onaylarım.

Prof. Dr. Erhan İÇENER

Enstitü Müdürü

BİLİMSEL ETİK BİLDİRİMİ

Yüksek lisans tezi olarak hazırladığım "Derin Öğrenme Algoritmaları ile Otonom Araçların Görme Algilarının Geliştirilmesinde Yeni Bir Yaklaşım" adlı çalışmanın öneri aşamasından sonuçlandığı aşamaya kadar geçen süreçte bilimsel etiğe ve akademik kurallara özenle uyduğumu, tez içindeki tüm bilgileri bilimsel ahlak ve gelenek çerçevesinde elde ettiğimi, tez yazım kurallarına uygun olarak hazırladığımı, bu çalışmamda doğrudan veya dolaylı olarak yaptığım her alıntıya kaynak gösterdiğimi ve yararlandığım eserlerin kaynakçada gösterilenlerden oluştuğunu beyan ederim.

Mustafa TOKAT

ÖNSÖZ

Araştırmamdaki her aşamada bana yardımcı olan değerli tez danışmanım Dr. Öğr. Üyesi Zahra ELMI'ye, eğitim alanında dersleriyle ve dostluğuyla bana vizyon katan çok değerli hocam Prof. Dr. Çetin Kaya KOÇ'a, yüksek lisans eğitimim boyunca benden desteklerini esirgemeyen Dr. Abuzer DİŞKAYA'ya, araştırmamı yapabilmem için bana gerekli ortamı sağlayan Seyfullah SUBAŞI, Ramazan KALKAN ve Cüneyt YANIK'a teşekkürlerimi sunarım.

Mustafa TOKAT

İstanbul - 2025

ÖZET

DERİN ÖĞRENME ALGORİTMALARI İLE OTONOM ARAÇLARIN GÖRME ALGILARININ GELİŞTİRİLMESİNDE YENİ BİR YAKLAŞIM

Mustafa TOKAT

Yüksek Lisans, Bilgisayar Bilimleri ve Mühendisliği

Tez Danışmanı: Dr. Öğr. Üyesi Zahra ELMI

Haziran, 2025 - 83 Sayfa

Bu çalışma, otonom araçların yol yüzeyi algısının geliştirilmesi ve özellikle çukur tespitinde karşılaşılan zorlukların aşılmasına yönelik yenilikçi bir derin öğrenme yaklaşımını ele almaktadır. Mevcut derin öğrenme modellerinin (CNN, vs.) sınırlılıkları literatürde tartışılırken, çalışmada Vision Transformer (ViT) ve Uzun Kısa Süreli Bellek (LSTM) modellerinin hibrit bir yapı altında entegrasyonu önerilmiştir. Geliştirilen model, farklı aydınlatma, perspektif ve yol yüzeyi koşullarını içeren çeşitli veri setleri üzerinde eğitilmiş ve veri artırma teknikleri ile desteklenmiştir. Deneysel sonuçlar, hibrit modelin çukur tespitinde yüksek doğruluk (%94.2) ve yüksek kesinlik (%99.17) ve yüksek bir duyarlılık (%90.61) sağladığını, bu sayede otonom sürüş sistemlerinin çevresel algısını ve yol güvenliğini artırmada etkili bir çözüm sunduğunu ortaya koymaktadır.

Anahtar Kelimeler: Otonom araçlar, görme algısı, çukur tespiti, derin öğrenme, Vision Transformer, LSTM, CNN, veri artırma, sensör füzyonu.

ABSTRACT

A NEW APPROACH TO DEVELOP VISUAL PERCEPTION OF AUTONOMOUS VEHICLES WITH DEEP LEARNİNG ALGORITHMS

Mustafa TOKAT

Master of Science, Computer Science and Engineering

Thesis Advisor: Asst. Prof. Dr. Zahra ELMI

June, 2025 – 83 Pages

This study addresses the enhancement of visual perception in autonomous vehicles, focusing specifically on the detection of road potholes. While the limitations of conventional deep learning models (e.g., CNNs) are well documented in the literature, this research proposes a novel hybrid approach that integrates Vision Transformer (ViT) and Long Short-Term Memory (LSTM) networks. The developed model is trained and validated on diverse datasets comprising images captured under various lighting conditions, perspectives, and road surface types, supported by advanced data augmentation techniques. Experimental results demonstrate that the hybrid model achieves high accuracy (94.92%), high precision (99.17%) and high recall (90.61%) in potholes detection, offering an effective solution to improve the environmental perception and road safety of autonomous driving systems.

Keywords: Autonomous vehicles, visual perception, pothole detection, deep learning, Vision Transformer, LSTM, CNN, data augmentation, sensor fusion.

İÇİNDEKİLER

TEZ ONAYI ..... **i**

BİLİMSEL ETİK BİLDİRİMİ ..... **ii**

ÖNSÖZ ..... **iii**

ÖZET ..... **iv**

ABSTRACT ..... **v**

İÇİNDEKİLER ..... **vi**

TABLO LİSTESİ ..... **viii**

ŞEKİL LİSTESİ ..... **ix**

KISALTMALAR ..... **x**

BİRİNCİ BÖLÜM **1**

GİRİŞ **1**

1.1. Otonom Sürüşe Genel Bakış 1

1.2. Konu 3

1.3. Amaç 3

1.4. Çalışmanın Önemi ve Hipotezler 3

1.4.1. Mevcut Derin Öğrenme Modelleri Hangi Limitasyonlara Sahiptir? 5

1.4.2. Hangi Derin Öğrenme Teknikleri En İyi Sonucu Verebilir? 8

1.4.3. Çalışma Özelinde Yeni Yaklaşımlar Nasıl Geliştirilebilir ve Test Edilebilir? 9

İKİNCİ BÖLÜM **12**

LİTERATÜR TARAMASI **12**

2.1. Yapılan Çalışmalar 12

ÜÇÜNCÜ BÖLÜM **27**

YÖNTEM **27**

3.1. Veri Seti ve Veri Ön İşleme 27

3.2. Yöntem ve Teknikler 31

3.2.1. Vision Transformer 32
3.2.2. Konvolüsyonel Sinir Ağları (CNN) 34
3.2.3 Long-Short Term Memory (LSTM) 36
3.2.4 You Only Look Once (YOLO) 38
DÖRDÜNCÜ BÖLÜM 41
UYGULAMA VE DENEYLER 41
4.1. Çalışma Prensibi 41
4.2. ViT-LSTM Model Birleşimi 54
4.3. İnce Ayar 56
BEŞİNCİ BÖLÜM 58
BULGULAR VE SONUÇLAR 58
5.1. YOLOv8 Uygulaması 58
5.2. CNN Model Uygulaması 58
5.3. Önerdiğimiz ViT-LSTM Modeli Uygulaması 59
5.4. Model Test Görüntüleri 68
ALTINCI BÖLÜM 74
TARTIŞMA 74
6.1. Sonuçlar 74
6.2. Öneriler 74
KAYNAKÇA 75
ÖZGEÇMİŞ 83

TABLO LİSTESİ

Tablo 2.1: CNN tabanlı karşılaştırmalı çukur tespiti performansları. 16
Tablo 2.2: YOLO versiyonları arasında performans karşılaştırması. 17
Tablo 3.1: Çukur hakkında konum bilgilerini içeren .json dosyası örneği 31
Tablo 5.1:Model metriklerinin karşılaştırmalo tablosu. 63

ŞEKİL LİSTESİ

Şekil 2.1: Semantik segmentasyon görseli. (Kaynak: cityscape.com) 15
Şekil 2.2: Nesne tespiti görseli (Kaynak:www.visionplatform.ai) 15
Şekil 3.1: Pozitif etiketli fotoğraflar örneği. 28
Şekil 3.2: Negatif etiketli fotoğraflar örneği. 28
Şekil 3.3: Verilerin genişlik, yükseklik ve konum ısı haritası 29
Şekil 3.4: Veri etiketlerinin dağılımı. 30
Şekil 3.5: Vision Transformer modeli. (Kaynak:Dosovitskiy vd. 2020) 34
Şekil 3.6: CNN model mimarisi. 36
Şekil 3.7: LSTM model mimarisi (Saheed, Omole, ve Sabit 2025). 38
Şekil 3.8: YOLO modeli. (Kaynak: Redmon vd. 2016) 40
Şekil 4.1: Resimlerin bölünmesi 42
Şekil 4.2: Lineer Projekesiyon 42
Şekil 4.3: Öz dikkat skoru hesaplaması (Kaynak: nature.com) 46
Şekil 4.4: Transformer model mimarisi (Vaswani vd. 2017) 49
Şekil 4.5: Multi-Head Attention (Çok Başlı Dikkat) mekanizması 50
Şekil 4.6: Modelimizin LSTM katmanları. 53
Şekil 4.7: Önerdiğimiz ViT-LSTM modeli veri akışı. 55
Şekil 5.1: Önerdiğimiz modelin karmaşıklık matrisi değerleri. 61
Şekil 5.2: CNN modeli karmaşıklık matrisi değerleri 62
Şekil 5.3: YOLOv8 modeli karmaşıklık matrisi değerleri. 62
Şekil 5.4: Önerdiğimiz modelin ROC Eğrisi. 64
Şekil 5.5: CNN modeli ROC eğrisi. 65
Şekil 5.6: Tahmin - Olasılık Dağılımı Analizi 66
Şekil 5.7: Modelimize ait Precision-Recall Eğrisi. 67
Şekil 5.8: YOLOv8 modelinin PR Eğirisi. 68
Şekil 5.9: Gerçek zemin fotoğrafları 69
Şekil 5.10: Modelimizin tahminleri. 70
Şekil 5.11: CNN modeli tahminleri. 71
Şekil 5.12: YOLOv8 Gerçek Zemin Görüntüleri 72
Şekil 5.13: YOLOv8 Tahmin Görüntüleri 73

KISALTMALAR

LIDAR	:Light Detection and Ranging
VIT	:Vision Transformer
TUİK	:Türkiye İstatistik Kurumu
CNN	:Convolutional Neural Network
LSTM	:Long-Short Term Memory
R-CNN	:Region based Convolutional Neural Network
YOLO	:You Only Look Once
GRU	:Gated Recurrent Unit
MAE	:Mean Average Error
RMS	:Root Mean Squarred
ReLU	:Rectified Linear Unit
CLS	:Classification
AP	:Average Precision
ROC	:Receiver Operating Characteristic
AUC	:Area Under the Curve

BİRİNCİ BÖLÜM

GİRİŞ

1.1. Otonom Sürüşe Genel Bakış

Otonom araç teknolojileri genel olarak ulaşım sistemleri ekosistemini bütünüyle değiştirme potansiyeline sahip bir alandır. Bütünüyle uygulandığında bir yerden bir yere varmanın dışında lojistiğe de büyük faydalar sağlayacağı varsayılmaktadır. Geliştirilmiş pil teknolojileri ile birleştirildiğinde iş-performans anlamında büyük değişiklikler yaratması beklenmektedir. Bu araçların temelinde çevresel farkındalıklarını sağlayan kamera, radar ve LIDAR gibi gelişmiş donanımlar kullanılmaktadır. Otonom araçların başarısı çevrelerini doğru bir şekilde algılayabilme ve çevreye uyumlu karar alabilme yeteneklerine bağlıdır. Karar alma sürecinin en önemli bileşenlerinden biri de görme algısıdır.

Otonom sürüş teknolojilerini hızla gelişmesi nedeniyle bu alandaki araştırmalar ve geliştirmeler dünya çapında büyük ilgi görmektedir. 1905 yılından bu yana dünyada otomotiv sektöründeki standartları belirleyen bir kuruluş olan Otomotiv Mühendisleri Topluluğu (SAE) otonom sürüşü 6 seviye tanımlamıştır. Seviye 0, tamamen insan kontrolüne dayalıdır. Seviye 1, araçta pasif uyarılardan ziyade frenleme veya hızlanmada bazı destek özelliklerin sunulduğu bir seviyedir. Örneğin öndeki araçla mesafeyi korumak amacıyla kendi kendine hızlanan veya fren yapan adaptif hız sabitleme sistemi, araç şeritten çıktığında otomatik olarak şeritte tutma, park yardımı gibi özellikler bu seviyede tanımlanan özelliklerdir. Seviye 2, gelişmiş sürücü destek sistemini sunar. Bu seviye, 1. seviyedeki özelliklerin belirli yol koşulları altında bir kombinasyonunu kullanır. Örneğin araç belirli şartlar altında hızlanabilir, frenleyebilir veya aracın direksiyonunu çevirebilir. Sürücü her an müdahale etmeye hazır olmalıdır. Seviye 3, seviye 2'ye göre daha teknolojik donanımlarla ve yazılımlarla desteklenmiştir. Sürücüye el ve göz serbestliği tanınır ancak yine de talep halinde her an kontrol edilmeye hazır olunmalıdır. Tesla Autopilot 3. Seviyede değerlendirilmektedir. 4. Seviye esasen tamamen otonom bir arabadır, bu aşamada çoğu zaman insan bir sürücüye ihtiyaç duyulmaz. Ancak yine de tam otonom sürüş sınırlandırılmış coğrafi alanlarda mümkün olabilmektedir. Burada detaylı HD haritalar kullanılır ve araç hızı sınırlandırılmış olabilir. Sürücüsüz robotaksiler bu

seviyede kullanılabilir. Paylaşımlı yolculukların mümkün olduğu bu seviyede dünyada araç sahipliğinde azalma olabilir. Google'ın çatı şirketi Alphabet Inc.'in sahip olduğu Waymo taksiler 4. seviye gereksinimlerini karşılamaktadır. Seviye 5 ise herhangi bir sınırlama olmadan tamamen otonom bir sürüşü temsil etmektedir. Her türlü şartta insan sürücüye ihtiyaç duymadan hareket edebilen bir yapıdır (SAE, 2021)

Genel olarak derin öğrenme geçmişine baktıktan sonra tezimizin konusu olan otonom sürüşte derin öğrenme teknolojilerini inceleyebiliriz. Otonom sürüş teknolojilerinde 5 temel bileşen vardır;

1. Algılama,
2. Lokalizasyon ve haritalama,
3. Yol planlama,
4. Karar verme,
5. Araç kontrolü.

Algılama bileşeni sürekli olarak donanımlardan gelen verileri tarar, bilgi toplar. Lokalizasyon ve haritalama bileşeni; lokal ve küresel lokasyonu, sensör datası ve algılama bileşeninin çıktılarına göre hesaplar. Yol planlama, kendisine gelen verilere göre mümkün olan yolları tespit eder. Karar verme süreci mümkün yolları, aracın şu anki durumunu ve yol şartlarını (yağmurlu, karlı, güneşli vb. hava koşulları) dikkate alarak belirler. Araç kontrolü ise hızlanma, frenleme, direksiyon açısı vb. parametreleri hesaplayarak araca anlık komutlar verir ve aracı yönlendirir. Otonom sürüş algılama algoritmaları kendi içinde 3 bölümde değerlendirilebilir: aracılı algılama, davranış reflex algılama, direk algılama. Aracılı algılamada algoritma, otonom aracın yol çizgilerine, yayalara, ışıklara, araçlara vb. olan mesafesini analiz ederek detaylı bir harita geliştirir. Bugün en yaygın kullanılan otonom sürüş algılama tekniği budur. Davranış reflex algılama algoritmaları sensör verilerini direk olarak direksiyona iletmek için yapay zekâ tekniklerini kullanır. Direk algılama algoritmaları ise aracılı algılama algoritmaları ve davranış reflex algılama tekniğinde kullanılan yapay zekâ algoritmalarını kombine eder. Bu yeni bir yaklaşım olduğu için halen ilgili çevrelerce tartışılmaktadır(Van Brummelen, O'Brien, Gruyer, & Najjaran, 2018).

1.2. Konu

Tezimiz, otonom araçların yol yüzeyi algısını geliştirerek çukur tespitini daha güvenilir ve hassas hale getirmek amacıyla yeni bir derin öğrenme tekniğini araştırmaktadır. Mevcut otonom araçların yol üzerindeki çukur tespiti konusunda karşılaştığı zorlukları ele alan bu çalışma, Konvolüsyonel Sinir Ağları (CNN), Vision Transformer (ViT) ve Uzun Kısa Süreli Bellek Ağları (LSTM) gibi ileri düzey algoritmaları bir araya getirerek daha etkili bir tespit modeli geliştirmeyi hedeflemektedir. Bu entegrasyon sayesinde model, çukurların boyut, şekil ve konum gibi değişken özelliklerini daha doğru sınıflandırabilecek ve tespit edebilecektir. Önerilen yöntemin geliştirilmesi ve optimizasyonu sürecinde elde edilen bulgular, hem literatüre katkı sağlayarak çukur tespiti üzerine yapılan çalışmaları ileri taşımayı hem de otonom araçların yol güvenliğini artırmayı amaçlamaktadır.

1.3. Amaç

Ana amacımız otonom araçların yol yüzeylerindeki çukurları algılamaları için kritik öneme sahip olan görme yeteneklerinin geliştirilmesine yönelik yenilikçi ve etkili yöntemler geliştirmektir. Mevcut derin öğrenme modelleri ve algoritmaları üzerine kapsamlı bir literatür taraması yaparak, otonom araçların görme algıları için hangi yöntemlerin daha etkili olduğunu karşılaştırmalı olarak belirlemek ve bu yöntemlerden yola çıkarak otonom araçların görme yeteneklerini iyileştirmeye yönelik mevcut yöntemlerin sınırlılıklarını aşacak yeni bir algoritmanın geliştirilmesini amaçlamaktayız.

1.4. Çalışmanın Önemi ve Hipotezler

Otonom sürüş teknolojisi, beş seviye ile sınıflandırılmaktadır. Seviye 0 tamamen insan kontrolüne dayalı sürüşü ifade ederken, Seviye 5 tamamen otonom sürüşe işaret eder; 5. seviyedeki bir araç, hiçbir insan müdahalesine ihtiyaç duymadan tamamen kendi başına karar alabilen bir sistemle donatılmıştır. Günümüzde Seviye 2 ve Seviye 3 otonom sistemler yaygın olarak kullanılmaktadır. Tesla'nın Autopilot sistemi, Seviye 2 otonom sürüşe örnek olarak gösterilebilir. Bununla birlikte, Waymo gibi şirketler tamamen sürücüsüz Seviye 4 araçları kullanmakta ve Seviye 5 araçları test etmektedir. Yapılan bir araştırmaya göre Seviye 0 ve 1 araçlar, otonom olmayan araçlara kıyasla kaza oranını yaklaşık olarak %29 oranında azaltmaktadır (Placek, 2021). 2023 itibarıyla dünya

genelinde otonom araçlar yaklaşık 1 milyar kilometre test sürüşü yapmıştır. Bu testler, otonom sürüş teknolojilerinin güvenilirliği konusunda önemli veriler sunmaktadır. İstatistiklere göre, Türkiye'de trafik kazalarının %88.9'u insan hatasından kaynaklanmakta ve her yıl yaklaşık 5000 kişi bu kazalarda hayatını kaybetmektedir(TUİK, 2023). Ayrıca maddi hasarlı kazaların yıllık maliyeti ülkemiz için 4 milyar doları bulmaktadır.

Otonom araçlar, insan hatasını ortadan kaldırarak trafik kazalarını ve aynı zamanda kazadan oluşan maliyetleri önemli ölçüde azaltma potansiyeline sahiptir. Otonom araçlar, modern ulaşımın geleceği olarak görülmekte ve bu alandaki ilerlemeler hızla devam etmektedir. Bu araçların temel amacı, insan müdahalesine gerek kalmadan güvenli ve verimli bir şekilde yol alabilmelerini sağlamaktır. Bunun başarılabilmesi için ise çevresel algı ve karar alma süreçleri büyük bir önem taşımaktadır. Araçların, etraflarındaki nesneleri algılayarak uygun tepkiyi vermesi, güvenli bir sürüş deneyiminin temel taşını oluşturur. Bu noktada en kritik unsurlardan biri görme algısıdır. Otonom araçlar, görme algıları sayesinde yolda karşılaştıkları engelleri, trafik işaretlerini, yayaları ve diğer araçları algılayarak karar verir. Bu algı, aracın içindeki sensörler aracılığıyla sağlanır ve bu verilerin işlenmesi sonucunda araç, çevresine uygun tepkiler geliştirir.

Görme algısının bu denli önemli olduğu bir sistemde, araçların kullandığı sensörler ve bunların işlediği verilerin doğruluğu kritik bir hale gelir. Günümüzde otonom araçlar genellikle LIDAR, radar ve kamera sensörleri kullanarak çevresini algılar. Ancak her bir sensörün kendine özgü sınırlamaları bulunmaktadır. Örneğin, LIDAR, yüksek çözünürlükte üç boyutlu görüntüler sunabilir, ancak yüksek maliyetli ve enerji tüketimi açısından yoğun bir sistemdir. Kamera ise görsel veriler sağlarken, düşük ışık koşullarında veya kötü hava şartlarında performansı düşebilir. Bu tür kısıtlamalar, otonom araçların yaygın olarak kullanılmasının önündeki önemli engellerden biridir. Araştırmacılar, bu zorlukların üstesinden gelebilmek için yeni teknolojiler ve algoritmalar geliştirme üzerine çalışmalarını sürdürmektedir. Bu tezde, mevcut sistemlerin zayıf yönlerini giderecek yeni bir yaklaşım geliştirilmesi amaçlanmaktadır.

Otonom sürüş sistemleri, çevre algısı konusunda derin öğrenme algoritmalarını kullanarak araçların etrafındaki nesneleri daha hızlı ve doğru bir şekilde tanımasına yardımcı olur.

Bu da araçların daha güvenli kararlar almasını sağlar. Bununla birlikte, otonom araç teknolojisi sadece algoritmalarla sınırlı değildir; sensörlerden gelen büyük miktardaki verinin işlenmesi ve bu verilerin doğru yorumlanması gerekmektedir. Farklı sensörlerin entegrasyonu ile elde edilen verilerin işlenmesi, araçların daha geniş bir algılama kabiliyetine sahip olmasını sağlar. Bu süreçte sensör füzyonu adı verilen tekniklerle LIDAR, radar ve kamera gibi farklı veri kaynakları birleştirilir ve derin öğrenme algoritmaları bu verileri analiz eder. Ancak bu tür verilerin işlenmesi yüksek hesaplama gücü gerektirmektedir, bu da araçlarda gerçek zamanlı performans için bir zorluk oluşturmaktadır. Otonom sürüş, gelişen teknolojisiyle büyük bir etki yaratırken, halen karşılaşılan zorluklar bulunmaktadır.

1.4.1. Mevcut Derin Öğrenme Modelleri Hangi Limitasyonlara Sahiptir?

• Otonom araçlar için kullanılan başlıca mevcut görme algılama modelleri nelerdir ve bu modellerin sınırlamaları nelerdir?

Otonom sürüşte kullanılan modeller Katmanlı[1] ve Uçtan uca[2] mimariler olmak üzere iki başlık altında incelenebilir. Katmanlı mimariler, otonom sürüşte tüm süreçleri modüler olarak tasarlayıp insan tarafından daha kontrol edilebilir bir tasarım sunar. Sistemin bileşenleri birbirinden bağımsızdır ve hata kontrolüne daha çok müsade eder. Katmanlı mimariler algılama, planlama/karar verme ve kontrol süreçlerinden oluşur. Algılama kısmında derin öğrenme modelleri veya klasik bilgisayarlı görme modelleri kullanılmaktadır (J. Zhao vd., 2024: 4). Çalışmamızda kullanacağımız derin öğrenme mimarilerinden olan uçtan uca mimariler ise doğrudan ham sensör verilerini (örneğin kamera görüntüleri) kullanarak girdileri doğrudan çıktı eylemlerine çeviren derin öğrenme modellerini içerir. Görüntü tabanlı nesne algılama ve sınıflandırmada yaygın olarak kullanılan CNN modelleri, küresel dikkat mekanizması sayesinde daha büyük ölçekli görsel ilişkileri yakalayabilen ViT mimarileri, zaman serisi verileri ile çalışarak hareket eden nesnelerin takip edilmesine olanak tanıyan LSTM modelleri sıkça tercih edilen uçtan

[1] Layered Architectures

[2] End-to-End Architectures

uca mimari modelleridir. Balasubramaniam ve Pasricha yaptığı bir çalışmada, nesneler için önce aday bölgelerin çıkarılıp sonra sınıflandırmaların ya da sınırlayıcı kutuların çizildiği iki aşamalı ve bir seferde hem bölgenin hem de nesnenin konumunun tespit edildiği tek aşamalı nesne tespit yöntemlerinden bahsetmiştir. İki aşamalı nesne tespit yöntemlerinden R-CNN, Mask R-CNN, Fast R-CNN, Faster R-CNN gibi modeller tarafından gerçekleştirildiği, bu modellerin yüksek doğruluğa karşı uzun işlem süresi gerektirdiği belirtilmiştir. Tek aşamalı modellere ise YOLO (You Only Look Once), SSD (Single Shot Multibox Detector) ve RetinaNet'i örnek verilmiştir. Bu modellerin doğruluk oranları iki aşamalı nesne tespit yöntemlerine göre düşük olsa da hızlı işlem gerçekleştirdiğini bildirilmiş, ayrıca iki aşamalı modellerin küçük ve hassas nesneleri tespit etmede tek aşamalı modellere göre daha başarılı olduğu belirtilmiştir (Balasubramaniam & Pasricha, 2022: 4).

• Hangi senaryolarda yukarıda bahsettiğimiz derin öğrenme modelleri yetersiz kalmaktadır?

Öncelikle ortam koşullarına bağlı sınırlamalar mevcuttur. Örneğin olumsuz hava koşullarında (karlı, yağmurlu) ve düşük görüş mesafesinde (örneğin sisli havalarda) radar, kamera veya LIDAR cihazlarının kaliteli veri toplama kabiliyetleri olumsuz etkilenmektedir. Örnek olarak Faster R-CNN gibi birçok algoritma çeşitli hava koşulları altında sensörlerden gelen verileri işledikten sonra yüksek doğruluk oranlarını korumakta yetersiz kalmaktadırlar. Yola aniden çıkan yayalar ya da bir nesnenin arkasında kalarak bir kısmı görünen yayalar, yola düşen cisimler vb. dinamik durumlarda da YOLO, CNN, VGG16 gibi birçok model zayıf kalmaktadır. Gerçek zamanlı algılama ve hesaplama kaynaklarının maliyetinin yüksekliği nedeniyle PointNet, SegNet, DLNet, Dense LightNet, RFNet, ERFNet gibi algoritmalar yetersiz kalmaktadır (Brummelen, O'Brien, Gruyer, & Najjaran, 2018: 388). Aynı zamanda yollarda çizgilerin bulunmaması, trafik işaretlerinin bulunmaması, yoldaki yapısal bozukluklar da olumsuz senaryolara örnek olarak verilebilir. Örneğin Narayan ve arkadaşları yol çizgilerinin bulunmadığı yollar için veya yarı silinmiş yol çizgilerinin otonom araç tarafından tanınmasını konu edinen bir çalışma yapmışlardır. Pioneer 3-AT robotu üzerinde uygulanan sistemde, kamera

görüntüsü 5×5'lik bir ızgaraya[3] bölünerek renk kanallarının ağırlıklandırılmasıyla oluşturulan giriş verileri, sürekli zamanlı geri beslemeli sinir ağı (CTRNN) tarafından işlenmiştir. Simülasyon ve açık hava testlerinde sistem, eğitim verisi dışındaki ortamlarda yüksek başarı göstermiş; gerçek dünya testlerinde ortalama başarı elde etmiştir (Narayan, Tuci, Labrosse, & Alkilabi, 2018). Görme algısı otonom araçların yol durumu, hava durumu, trafik yoğunluğu, yayalar, trafik işaretleri, trafik ışıkları, çevresel unsurlar gibi kritik bileşenleri doğru bir şekilde tanıyabilmesini sağlar. Görme algısının doğruluğu ve güvenilirliği, sisteme veri sağlayan donanım cihazlarına bağlıdır. Örneğin kamera ucuzluk açısından uygun olabilmekte fakat düşük/yüksek ışıklarda performans kaybı yaratabilmektedir. Ya da LIDAR sensörü 3 boyutlu tarama ve mesafe algısı açısından stabil çalışmasına rağmen sürüşte kullanılan diğer donanımlara işlem maliyeti açısından ağır olabilmektedir. Ya da radar sensörlerinin gereksiz aktiviteleri sistemin verimini düşürdüğü gibi diğer elektromanyetik parazitler (diğer radar sinyalleri, kablosuz cihazlar vs.) performansı da olumsuz yönde etkilemektedir (Javadi & Farina, 2020). Kim ve arkadaşları tarafından yapılan bir çalışma, LiDAR sensörlerinin yoğun yağmur (30–40 mm/saat) ve kalın sis (50 m görüş mesafesi) altında tespit performansının anlamlı şekilde azaldığını; özellikle yansıtıcı film kaplı trafik levhalarının %74 oranında algılanabilirliğini korurken, alüminyum ve çelik yüzeylerin 20–30m mesafelerde tamamen algılanamadığını ampirik olarak ortaya koymaktadır (J. Kim, Park, & Kim, 2023: 14). Ayrıca Weihmayr ve arkadaşlarının yaptığı bir çalışmada yağmurlu ve sisli havalarda hem radar sinyallerinin zayıfladığı hem de LIDAR sensörlerinin oluşturduğu 3D nokta bulutlarının[4] düşük çözünürlüklü işlendiği belirtilmiştir (Weihmayr, Sezgin, Tolksdorf, Birkner, & Jazar, 2024). Yapılan başka bir çalışmada radar sensörlerinin araç üzerindeki konumlandırmalarının büyük bir hassasiyetle yapılması gerektiği vurgulanmaktadır. Bu sağlanmadığı takdirde önemli durumlarda kör noktaların oluşabileceği veya coğrafi engellerin sinyal engellemelerine neden olabileceği belirtilmiştir. Aynı çalışmada farklı radarların birden fazla hedefi izlemede zorlandığı, hedeflerin birbirine yakın olması

[3] Grid

[4] Point Cloud

onların tanımlanmasında karışıklıklara neden olabileceği yine Javadi ve arkadaşlarının yaptığı çalışmada belirtilmiştir (Javadi & Farina, 2020).

1.4.2. Hangi Derin Öğrenme Teknikleri En İyi Sonucu Verebilir?

• Farklı derin öğrenme mimarilerinden (CNN, RNN, GAN'lar vb.) otonom araçların görme algısı için uygulanan örnekleri nelerdir ve performansları nasıldır?

Prabhakar ve arkadaşları Hindistan'da normal ve yağışlı hava koşullarında KITTI ve bazı diğer veri setleriyle çalışmalar yapmıştır. Önerdikleri yaklaşımda Faster R-CNN kullanarak otobüs, araba, motorsiklet ve insan etiketlerini kullanmışlardır. Geliştirdikleri uygulamanın sırasıyla ortalama kesinlik[5] oranları, 0.98, 0.92, 1 ve 0.98 olarak tespit etmişlerdir (Prabhakar, Kailath, Natarajan, & Kumar, 2017). Al-Qizwini ve arkadaşları, otonom sürüşte şerit takibi konusunda bir çalışma sunmuşlardır. Söz konusu araştırmada, araçların yol çizgilerine göre konumsal sapmalarını tespit eden ve düzelten bir yaklaşım geliştirilmiştir. Yöntemde, aracın şerit çizgilerine göre sağa, sola ve merkeze olan açısal sapmalarını hesaplayan bir algoritma önerilmiştir. Araştırmacılar, bu karmaşık görsel tanıma problemi için derin öğrenme mimarilerinden GoogleNet'i tercih etmişlerdir. Clarifai, VGGNet ve GoogleNet modelleri arasında yapılan karşılaştırmada en düşük RMSE[6] değerini GoogleNet elde etmiştir (Al-Qizwini, Barjasteh, Al-Qassab, & Radha, 2017). Faster R-CNN gibi klasik CNN temelli algoritmaların dışında nokta bulutu yaklaşımları da kullanılmaktadır. Örneğin nokta temelli algoritmalar mekânsal ve geometrik verileri koruyarak direk olarak ham veriyi işler. Bu sayede karmaşık nokta bulutu verilerini sadeleştirerek tüm bilgileri korur. Volumetrik Piksel olarak ifade de edilen VOXEL temelli yaklaşımlar da mevcuttur. Nokta bulutlarını 3 boyutlu VOXEL ızgaralarına dönüştürerek klasik CNN modellerinin uygulanmasını sağlar. Bunlarla birlikte 2 boyutlu projeksiyon tabanlı yaklaşımlar da kullanılmaktadır. Prensip olarak 3 boyutlu nokta bulutlarını 2 boyutlu bir düzlemin üzerine yansıtarak hem nesne ayrıntılarını korur hem de basitleştirilmiş görüntüler elde eder (J. Zhao vd., 2024).

[5] Average Precision

[6] Root Mean Squarred Error

• Özel zorluklar (bozuk yollar, gece görüşü, kötü hava koşulları, ani engeller vb.) için hangi teknikler daha uygun olabilir?

Khan ve arkadaşları tarafından yoldaki çukurları tespite dönük bir çalışma yapmışlardır. Bu çalışmada önerdikleri YOLOv8 modelini YOLOv5small ve YOLOv5large modelleriyle karşılaştırmışlardır. Sonuçlar incelendiğinde YOLOv8 modelinin diğerlerinden açık ara önde olduğunu tespit etmişlerdir (Khan vd., 2023). Kuang ve arkadaşları gece çekilen görüntülerde düşük kontrast ve parazitler nedeniyle birçok modelin farklı araç tiplerini tanıyamadığını belirtmişlerdir. Bu soruna çözüm olarak gece çekilen görüntülerde bir görüntü iyileştirme tekniği kullanarak resimleri tensörler vasıtasıyla ayrıştırmışlar ve özellik seçimi yapmışlardır. Ardından EdgeBox modeli ile araç içeren pencereler üretilmiş ve sınıflandırma için destek vektör makineleri eğitilip 4 farklı tür aracın sınıflandırmasını gerçekleştirmişlerdir (Kuang, Chen, Chan, Cheung, & Yan, 2019). Mehra ve arkadaşları sisli ve puslu havalarda otonom araçların güvenli bir şekilde sürüş gerçekleştirebilmeleri için ReViewNet adlı modeli tanıtmışlardır. Modelin hızlı ve hafif olmasının yanı sıra kendisine has özgün bir mimariye sahiptir. Model**,** çift bakış açılı[7] yapı, dört renk uzayı (RGB, HSV, YCrCb, LAB) kullanımı, mekânsal özellik havuzlama[8] ve ağırlıklı kayıp fonksiyonu gibi yenilikçi bileşenlerle geliştirilmiştir (Mehra, Mandal, Narang, & Chamola, 2021).

1.4.3. Çalışma Özelinde Yeni Yaklaşımlar Nasıl Geliştirilebilir ve Test Edilebilir?

• Otonom araçlar için hangi soruna yönelik yeni bir derin öğrenme modeli tasarlanabilir ve bu modelin yenilikçi yönleri nelerdir?

Daha önce mevcut durumda otonom sürüşü riskli duruma sokabilecek birçok durumdan bahsetmiştik. Kar-yağmur yağışı, sisli-puslu havalar, asfalt yolun bozuk olması, yolun toprak yol olması, yol üzerinde şeritlerin silinmiş olması, hiç şerit olmaması gibi nedenlerden söz etmiştik. Bizim çalışmamızda spesifik bir sorun olan yol üzerindeki çukurlar konu olarak el alınmıştır. Her çukurun kendine has özellikleri vardır. Herhangi

[7] Multi-Look

[8] Spatial Pooling

bir modelin trafik işaretleri veya yol çizgilerini tanıması ile çukurları tanıması bir olmamaktadır. Çünkü trafik işaret ve levhalarının belirli renkleri, geometrik şekilleri bulunmaktadır. Bu durum çukurlar için geçerli değildir. Her çukurun kendine has bir geometrik şekli olup, içbükey ya da dışbükey olabilmektedir. Ayrıca derinliği de farklılık göstermektedir. Dolayısıyla bu derinliğe göre çukurun içine düşen gölge miktarı değişmektedir. Çukurun içine düşen ışık miktarı, çukurun geometrik şekli ve derinliğine göre oluşan gölgelerin renk tonları da homojen özellikler gösterememektedir. Aynı zamanda hava koşullarına göre çukurlar suyla ve toprakla dolabilmektedir. Dolayısıyla bütün bu sorunları aşabilecek bir model geliştirmeyi düşündük. Modelin yenilikçi tarafı çalışmamızın konusu olan çukur tespitinde, dil modellerinde kullanılan Transformer algoritmasının resim işlemede kullanılan versiyonu Vision Transformer algoritması ile zaman sıralı görüntülerde üstün performans gösteren LSTM algoritmasının birleştirilerek kullanılması olmuştur. Gerçekleştirdiğimiz literatür taramasına göre sunacağımız ViT-LSTM modelinin çukur tespiti için ilk defa yapıldığı kanaatine varılmıştır.

• Bu yeni modelin performansını değerlendirmek için hangi ölçütler ve test yöntemleri kullanılmalıdır?

Doğruluk[9] metriği, modelin genel tahmin başarısını gösterir ancak tek başına yeterli değildir. Çukur tespitinde yol yüzeylerinin büyük kısmı çukur içermediğinden, bu metrik yanıltıcı olabilir. Örneğin, hiç çukur tespit etmeyen bir model bile %95'den fazla doğruluk gösterebilir. Bu yüzden diğer metriklerle birlikte değerlendirilmelidir.

Kesinlik[10], yanlış alarmlara karşı hassasiyet gerektiğinde önemlidir. Gereksiz yol onarımlarını önlemek için çukur olarak işaretlenen bölgelerin gerçekten çukur olması kritiktir.

Duyarlılık[11], çukurların gözden kaçmaması hayati önem taşır. Tespit edilmeyen çukurlar araç hasarlarına, kazalara ve yaralanmalara yol açabilir.

[9] Accurary metric

[10] Precision

[11] Recall

Kesinlik-Duyarlılık eğrisi[12], çukur tespitinde uygulamanın gerektirdiği kesinlik/duyarlılık dengesine göre eşik değeri ayarlanabilir bir metriktir.

Ortalama kesinlik[13], farklı eşik değerlerinde modelin performansını gösterir.

F1 skoru, kesinlik ve duyarlılık arasındaki dengeyi tek bir değerde özetler. Çukur tespitinde hem yanlış alarmları azaltmak hem de çukurları kaçırmamak önemli olduğundan bu denge çalışmamız için gereklidir.

ROC-AUC eğrisi[14], modelin çukur olan ve olmayan bölgeleri ayırt etme yeteneğini ölçer. Bu metrik, farklı çalışma koşullarında (çeşitli hava koşulları, aydınlatma koşulları vb.) modelin sağlamlığını değerlendirmek için seçilmiştir. Performans ölçümlerine dair daha detaylı olarak 4. Bölümde yer verilecektir.

[12] Precision-Recall Curve

[13] Average Precision

[14] Receiver Operating Characteristic - Area Under Curve

İKİNCİ BÖLÜM

LİTERATÜR TARAMASI

2.1. Yapılan Çalışmalar

Derin öğrenme, makine öğrenmesi ve yapay zekanın bir dalı olarak, dördüncü sanayi devriminin (Endüstri 4.0) önemli bir teknolojisi olarak kabul edilmektedir. Çok katmanlı sinir ağları kullanarak büyük veriler üzerinde karmaşık modeller öğrenme yeteneğine sahiptirler. Geleneksel makine öğrenmesi sınırlılıklarını aşarak verilerden temsilleri öğrenir. Temsili öğrenme ham verilerden, örneğin bir fotoğrafın piksel değerlerinden, otomatik olarak özellikler çıkarmasıdır (Lecun, Bengio, & Hinton, 2015: 436). Derin öğrenme bu verileri çok katmanlı bir yapı üzerinden öğrenir. Örneğin ilk katmanlar fotoğrafın kenarlarını çıkarırken sonraki katmanlar bu kenarları birleştirir ve tanır. Veri üzerinden öğrenme yeteneği sayesinde derin öğrenme, sağlık, görsel tanıma, siber güvenlik ve metin analitiği gibi çeşitli alanlarda yaygın olarak kullanılmaktadır.

Tezimizde sunacağımız bu yeni yaklaşımın temeli, derin öğrenme algoritmalarına dayanmaktadır. Yapay zekanın kökenleri Aristoteles'e kadar uzanmaktadır. Aristo insan beyninin nasıl işlediğini anlamaya çalışarak, daha sonra Çağrışımcılık olarak anılacak fikirsel yapısını kendisinin şu cümlelerine dayandırmaktadır:

> *Bu nedenle, bir anımsama eylemini gerçekleştirdiğimizde, arayış içinde olduğumuz hareketin alışkanlıkla sonuçlandığı bir harekete ulaşıncaya kadar, belirli bir dizi öncül hareketten geçeriz. Dolayısıyla, zihinsel trende de, şimdiki zamandan ya da başka bir zamandan, benzer ya da karşıt ya da eşzamanlı olanlardan yola çıkarak ararız. Bu süreç sayesinde anımsama gerçekleşir.*
>
> (Wang & Raj, 2017 :5).

Alexander Bain, 1873 yılında yayımlanan *Mind and Body* adlı eserinde, öğrenme, hafıza ve bilincin sinirsel gruplamalar ve hücresel bağlantılardaki gelişmelerle açıklanabileceğini öne sürdü. Bain, özellikle sinir devrelerinin rolünü vurgulayarak, tekrar eden eylem ve uyarıların sinirsel bağlantıları güçlendirebileceğini savundu (Wilkes & Wade, 1997: 301). Warren S. McCulloch ve Walter Pitts tarafından yayımlanan "A Logical Calculus of the Ideas Immanent in Nervous Activity" başlıklı makale, yapay sinir ağları

teorisinin temellerini atan çalışmalardan biridir. Bu makale, biyolojik sinir hücrelerinin işlevini anlamak amacıyla, mantıksal hesaplama kurallarını kullanarak bir sinir ağı modelini tanımlamıştır. McCulloch ve Pitts, nöronların "hepsi ya da hiçbiri" şeklinde davranan sinirsel olaylar olduğunu ve bu nöronların belirli mantıksal kurallara göre çalıştığını öne sürmüşlerdir (McCulloch & Pitts, 1943: 113). 1950 yılında Donald Hebb tarafından yazılan *The Organization of Behaviour* adlı kitabında Hebb öğrenme kuralından bahsetmiştir. Bu kural "Birlikte ateşleyen nöronlar birlikte bağlantı kurar" şeklinde özetlenmektedir. Hebb, bir nöronun diğerini ateşlemesine yardımcı olduğunda bu iki nöron arasındaki bağlantının güçleneceğini öne sürmüştür. Hebb'in bu çalışması yapay sinir ağlarının gelişmesinde önemli bir aşama olarak görülmüştür (Attneave, B, & Hebb, 1950: 62). 1958'de Frank Rosenblatt tarafından geliştirilen ve Perceptron modelini konu aldığı makalesinde Perceptron'un bilgi depolama ve organizasyon süreçlerini açıklamak için geliştirilmiş olasılıksal bir model olarak tanımlanmıştır. Perceptron günümüzde özellikle gözetimli öğrenme[15] süreçlerinde kullanılır. Bu çalışma özellikle iki sınıflı sınıflandırma problemlerinde kullanılabilecek bir sınıflandırıcı geliştirme amacı taşımaktadır (Rosenblatt, 1958: 387). Ancak, sınırlı veri ve hesaplama gücü nedeniyle uzun yıllar boyunca bu teknoloji tam anlamıyla gelişim gösterememiştir. 1974'te Paul Werbos tarafından doktora tezi olarak sunulan backpropagation, sinir ağlarının eğitimi için büyük bir ilerleme sağlamıştır. Bu tez, yapay sinir ağlarını eğitmek için hata yayılımını ilk kez ayrıntılı bir şekilde açıklamıştır. Werbos bu çalışmayla nöronların hatalarını geri yayarak ağırlıkların düzeltildiği ve öğrenme sürecinin geliştirildiği bir çalışma ortaya koymuştur (Werbos, 1974). 1980'de Fukushima K., sonraları evrişimli sinir ağlarına ilham olacak Neocognitron makalesini yayınladı. Bu mimaride farklı işlem türlerini gerçekleştirme amacıyla farklı katmanlar tasarlanmıştır. Bunlar ham piksel verilerini toplayan bir giriş katmanı, kenarlar ve dokular gibi özellikleri çıkaran özellik çıkartma katmanı, daha karmaşık desenleri ve yapıları tanımak için önceki katmanlardan çıkan özellikleri birleştiren yüksek bir katmandan oluşur (Fukushima, 1980). 1986 yılında M. Jordan tarafından Yinelemeli Sinir Ağları, kendisine ait olan "Serial Order: A Parallel

[15] Supervised Learning

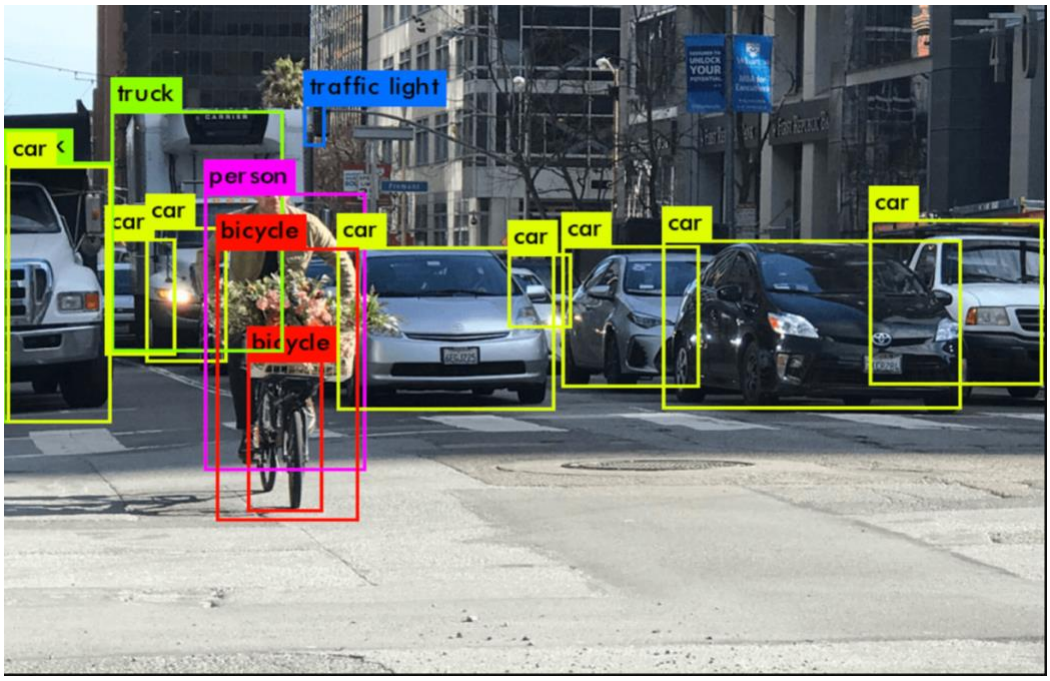
Distributed Processing Approach" adlı makalesi ile tanıtıldı. Bu model öğelerin bir bağlam çerçevesinde temsil edildiğini savunur ve her öğe işlendiğinde sistemdeki bağlam değişir ve bu bir sonraki öğenin hatırlanmasını kolaylaştırır (Jordan, 1986). Y. LeCun tarafından 1989 yılında görüntü tanıma alanında devrim yaratan evrişimli sinir ağları mimarisi olarak LeNet kazandırılmıştır. LeNet el yazısı rakamların tanınması için geliştirilmiş bir evrişimsel sinir ağı modelidir. Model temel olarak özelliklerin çıkartıldığı evrişim katmanı, veri boyutunun düşürülerek hesaplama maliyetlerinin azaltıldığı havuzlama katmanı ve son olarak verilerin sınıflandırıldığı tam bağlantılı katmandan oluşmaktadır (LeCun vd., 1989). 1997 yılında Hochreiter ve Schmidhuber yinelemeli sinir ağlarında sıkça karşılaşılan kaybolan gradyan problemini LSTM mimarisi ile çözmüştür. Diğer bir ifadeyle LSTM, RNN'lerin uzun dönem bağımlılıkları öğrenmede yaşanılan zorlukları çözmek için geliştirilmiştir. Dolayısıyla LSTM, geçmiş bilgileri uzun süreli olarak hatırlayabilme yeteneğiyle donatılmıştır (Hochreiter & Schmidhuber, 1997). 2012 yılında ise G.Hinton ve arkadaşları Dropout yöntemi ile rastgele nöronları devredışı bırakarak modelin aşırı öğrenmesini engellemeyi amaçlamıştır. Bu şekilde daha etkili bir model ortaya çıkarmayı hedeflemiştir (Hinton, Srivastava, Krizhevsky, Sutskever, & Salakhutdinov, 2012). Derin öğrenmenin geçmişindeki bu gelişmeler, onun otonom araçlar gibi görüntü işleme gerektiren teknolojilerde kullanımını da hızlandırmıştır.

Otonom sürüşte algı, aracın etrafında olup biten, diğer bir deyişle hareketli ve hareketsiz nesnelerin donanımlar ve algoritmalar vasıtasıyla tespit edebilme yeteneğidir. Otonom aracın ne yapacağına karar vermesinde birincil etkiye sahiptir. Aracın algılama sisteminde yapılacak herhangi bir hata ölümlü kazalara neden olabilmektedir. O yüzden bu alanda yapılan çalışmalar, algoritmaların, günlük hayatta karşılaşılabilecek tüm koşullara (hava şartları, yol durumu, trafik işaretleri, hareketli hareketsiz yayalar, ağaçlar, kaldırımlar, yol çizgileri vb.) uyum sağlayabileceği şekilde yapılmaya çalışılmaktadır. Bu alanda, bizim çalışmamızın da konusu olan algoritmalar tarafında, anlamsal segmentasyon ve nesne tespiti çalışmaları ağırlık kazanmış durumdadır. Semantik segmentasyon ve nesne tespiti aracın hedefine varıncaya kadar yönlendirilmesinde kullanılır. Semantik segmentasyon belirli bir resimde her pikselin bir sınıfa atanması yoluyla gerçekleşir ve her bir sınıf aynı renge boyanır. Şekil 2.1'de gösterildiği gibi insanlar kırmızı, kaldırım pembe, yol mor, trafik işaretleri sarı vb. renklere boyanmıştır. Nesne tespiti ise bir resimde aranan nesnenin

yeri gösterilir ve nesne tanımlanır. Otonom sürüş sistemlerinde nesnenin etrafı çizilerek gösterilir. Şekil 2.2'de görüldüğü gibi nesnenin etrafı çizilerek üzerine sınıfı yazılmıştır.

Şekil 2.1: Semantik segmentasyon görseli.

Kaynak: Cordts vd., 2016.



Şekil 2.2: Nesne tespiti görseli

Kaynak: Hui, 2018.

Jakubec ve arkadaşları çukur tespiti için iki aşamalı mimarilerden CNN temelli algoritmaları ve tek aşamalı yaklaşımlardan olan YOLO tabanlı yaklaşımları test etmişlerdir. Veriseti olarak farklı hava koşullarında çekilmiş (açık, günbatımı, yağmurlu, akşam ve gece) fotoğrafları kullanmışlardır (Jakubec, Lieskovská, Bučko, & Zábovská, 2023: 12). Test sonuçları Tablo 2.1 ve Tablo 2.2'de karşılaştırmalı olarak sunulmuştur.

Tablo 2.1: CNN tabanlı karşılaştırmalı çukur tespiti performansları.

Model	Weather	Precision	Recall	mAP@.5	mAP@[0.5:0.95]
R-CNN	Clear	64.2%	45.5%	67.2%	26.9%
	Rain	49.4%	42.4%	41.1%	12.7%
	Sunset	51.2%	45.9%	46.4%	15.8%
	Evening	47.5%	41.6%	38.8%	10.4%
	Night	31.6%	24.3%	29.7%	8.1%
Fast R-CNN	Clear	67.9%	48.4%	71.9% (+4.7)	29.2% (+2.3)
	Rain	51.3%	44.0%	44.0% (+2.9)	14.8% (+2.1)
	Sunset	53.2%	46.7%	48.1% (+1.7)	17.5% (+1.7)
	Evening	48.8%	43.3%	39.4% (+0.6)	12.8% (+2.4)
	Night	32.7%	31.6%	31.0% (+1.3)	9.2% (+1.1)
Faster R-CNN	Sunset	55.7%	47.4%	50.2% (+2.1)	18.7% (+1.2)
	Evening	49.6%	44.5%	41.3% (+1.9)	13.6% (+0.8)
	Night	33.9%	32.7%	31.8% (+0.8)	10.2% (+1.0)
	Clear	64.1%	53.4%	70.3% (−3.3)	28.3% (−3.8)
	Rain	55.2%	41.7%	48.3% (+2.8)	16.2% (+0.3)
Mask R-CNN	Sunset	51.6%	44.6%	43.7% (−6.5)	18.6% (−0.1)
	Evening	57.5%	47.2%	53.8% (+12.5)	22.3% (+8.7)
	Night	32.4%	32.7%	28.6% (−3.2)	8.9% (−1.3)
	Clear	71.1%	55.2%	73.7% (+0.1)	33.8% (+1.7)
	Rain	56.4%	46.6%	46.1% (−2.2)	16.6% (+0.4)
Cascade R-CNN	Sunset	57.0%	48.1%	52.4% (+2.2)	19.5% (+0.8)
	Evening	51.5%	45.8%	47.5% (−6.3)	14.4% (−7.9)
	Night	34.8%	33.5%	33.2% (+1.4)	10.6% (+0.4)
	Clear	69.2%	54.2%	74.6% (+0.9)	32.1% (−1.7)
	Rain	58.7%	47.7%	48.9% (+0.6)	16.6% (+0.0)

Kaynak: Jakubec vd., 2023: 14

Tablo 2.2: YOLO versiyonları arasında performans karşılaştırması.

Model	Weather	Precision	Recall	mAP@.5	mAP@[0.5:0.95]
YOLOv5m	Clear	84.0%	81.2%	83.3% (+2.9)	35.7% (+1.3)
	Rain	66.3%	56.1%	54.6% (+1.9)	21.5% (+0.7)
	Sunset	75.0%	53.6%	57.2% (+2.0)	21.0% (+0.8)
	Evening	80.2%	52.2%	51.2% (+1.8)	19.7% (+0.7)
	Night	38.9%	17.0%	18.9% (+0.6)	6.7% (+0.2)
YOLOv5l	Clear	86.1%	85.4%	85.4% (+2.1)	36.6% (+0.9)
	Rain	67.9%	57.5%	55.9% (+1.2)	22.0% (+0.5)
	Sunset	76.9%	54.9%	58.6% (+1.4)	21.5% (+0.5)
	Evening	82.2%	53.5%	52.5% (+1.3)	20.2% (+0.5)
	Night	39.9%	17.4%	19.4% (+0.5)	6.9% (+0.2)
YOLOv6m	Clear	87.9%	84.6%	87.3% (+1.9)	37.4% (+0.8)
	Rain	69.4%	58.7%	57.2% (+1.3)	22.5% (+0.5)
	Sunset	78.6%	56.1%	59.6% (+1.0)	22.0% (+0.5)
	Evening	84.0%	54.7%	53.7% (+1.2)	20.6% (+0.4)
	Night	40.7%	17.8%	19.8% (+0.4)	7.0% (+0.1)
YOLOv6l	Clear	88.2%	86.1%	87.8% (+0.5)	37.6% (+0.2)
	Rain	69.5%	59.1%	57.4% (+0.2)	22.7% (+0.2)
	Sunset	78.8%	56.4%	59.9% (+0.3)	21.0% (-1.0)
	Evening	84.4%	55.0%	54.1% (+0.4)	20.8% (+0.2)
	Night	41.1%	18.6%	19.9% (+0.1)	7.0% (+0.0)
YOLOv7	Clear	88.6%	88.0%	88.4% (+0.6)	40.3% (+2.7)
	Rain	70.6%	60.2%	58.1% (+0.7)	24.9% (+2.2)
	Sunset	79.1%	57.7%	61.5% (+1.6)	22.8% (+0.8)
	Evening	84.9%	56.1%	54.8% (+0.7)	21.5% (+0.7)
	Night	42.8%	19.5%	20.4% (+0.5)	7.5% (+0.5)

Kaynak: Jakubec vd., 2023: 17

Bir grup araştırmacı çeşitli çalışma koşulları altında (yağmur, aydınlatma vb.) semantik segmentasyon algoritmalarının performansını doğrulama problemini ele almışlardır. Bu koşullardaki en ufak bir değişiklik ölümcül kazalara neden olabilir. Bu sorunun üstesinden gelebilmek için araştırma grubu 6 ay boyunca haftalık veriler topladı. Gerçek hayat senaryolarını içeren bu veriler, önerdikleri ENet ve BonNet algoritmaları kullanılarak LIDAR sensörü ile test edildi. Çalışma sonucunda en iyi model seçiminin çalışma koşullarına bağlı olduğuna ve modellerin doğruluğunun verisetine göre değişkenlik gösterdiği sonucuna ulaştılar (Zhou, Berrio, Worrall, & Nebot, 2020: 11). Makine görmesi alanında sıkça karşılan sorunlardan biri de tespit edilen kenarların belirgin olmamasıdır. Lee ve arkadaşları bu sorunun üstesinden gelebilmek amacıyla EfficientNet algoritmasını kullanarak pozisyon bilgisi eklemek için düşük özelliklere koordinasyon konvolüsyonu uyguladılar. Bu metodun uygulanması varolan semantik segmentasyon modellerinden daha iyi bir performans sergilemiştir (Lee & Park, 2020: 523). Yoshioka ve arkadaşları otonom araçlar için LIDAR kullanarak gerçek zamanlı bir nesne sınıflandırıcısı önermişlerdir. Real ADABOOST algoritması ile LIDAR'daki 3 boyutlu nokta bulutları yol nesnelerinin çeşitli özellikleri çıkarılmıştır ve 50 metrelik bir mesafede %90'ın üzerinde doğruluk yakaladığı görülmüştür. Bu algoritmanın bir nesnenin sınıflandırmasını 0.07×10^{-3} saniyede gerçekleştirdiği için otonom sürüş sistemlerinde kullanılabileceği söylenmiştir (Yoshioka, Suganuma, Yoneda, & Aldibaja, 2017: 211). Gluhakovic ve arkadaşları çevredeki araçları tespit ederek sürücüleri potansiyel kazalara karşı uyarmak için bir sistem geliştirdiler. Sistem 2 aşamalı olarak kurgulanmış ve robot işletim sistemine[16] uygulanmıştır. Sistemin ilk parçasında YOLOv2 algoritması kullanılmıştır. Algoritmada araba, otobüs, kamyon ve kamyonet sınıfları eğitilmiştir. İkinci parçası ROS bağlantı noktalarıdır. Mesafeyi değerlendirmek için 2 ROS bağlantı noktası oluşturulmuş, ilki Carla simülatöründe, ikincisi gerçek hayat senaryosunda kullanılmıştır. Algoritma sisli, yağmurlu, gece ve gündüz olmak üzere 4 durumda eğitilmiştir. Nesne tespitinde önerilen yöntem nesnelerin mesafelerini yakın mesafelerde daha doğru tahmin ederken, uzaklaştıkça ve mesafe 50 metreyi geçtikten sonra sınırlayıcı kutuların çizgilerinin

[16] Robot Operating System

belirsizleştiğini, bununla birlikte mesafe tahmininin hatalı sonuçlar verdiği neticesine ulaşmışlardır (Gluhakovic, Herceg, Popovic, & Kovacevic, 2020: 180). NVIDIA araştırma grubu tarafından yapılan bir araştırmada sürücüsüz arabalarda 9 katmanlı bir CNN mimarisi kullanılmıştır. NVIDIA'nın geliştirdiği DRIVE PX donanımına sol-sağ ve merkez olmak üzere 3 adet kamera yerleştirilmiş ve gerekli olan veriler bu donanım ile toplanmıştır. Verileri çoğaltmak amacıyla toplanan data rastgele döndürülmüş ve CNN'de veri artırılmıştır. Backpropagation algoritması ile de optimize edilmiştir. Sonuç olarak uçtan uca tasarlanan CNN mimarisinin yol veya şerit işareti algılama, yol planlama ve yol takibi görevlerinin tamamını öğrenebildiği gösterilmiştir (Bojarski vd., 2016: 9). Ciberlin ve arkadaşları nesne tespiti ve takibine odaklanmışlardır. Nesne tespiti ve takibi süreci, bazı modüller için (örneğin çarpışma önleyici sistem, acil durum frenlemesi, yol planlama gibi) gerekli olan bilgileri üretir. Nesne tespitinde iki algoritma kullanılmıştır: Viola Jones ve YOLOv3. Aracın ön/arka dedektörü, aracın yan kısımları dedektörü, yaya dedektörü, trafik ışığı dedektörü, dur işareti dedektörü, yol kenarı işaretleri dedektörü, uyarı işaretleri dedektörü, mavi arkaplanlı levhalar dedektörü, kırmızı kenarlı levhalar dedektörü olmak üzere toplamda 9 sınıf Viola Jones algoritması tarafından eğitilmiştir. Sonuçlara göre YOLOV3 algoritmasının Viola Jones algoritmasına göre daha iyi sonuçlar verdiği gözlemlenmiştir. Bunun yanında nesne takibinde ise yine 2 algoritma kullanılmıştır: Median Flow metodu ve korelasyon izleme metodu. İşlem hızına göre Median Flow daha etkili sonuçlar verirken MOTA[17] metriğine göre iki algoritmada benzer sonuçlar vermiştir (Ciberlin, Grbić, Teslić, & Pilipović, 2019: 31). Chen ve arkadaşları ölümcül kazalardan kaçınmak için sahnedeki önemli parçaların önemsiz parçalardan ayrılması gerektiğini düşündüler. Örneğin bir sahnedeki araba ve yayalar, gökyüzü ve binalardan daha önemlidir. Başka bir deyişle önem derecesine göre değerlendirilmesi gerektiğini savundular. Bunun için araştırmacılar Önem Duyarlı Kayıp Fonksiyonu[18] yöntemini sunmaktadırlar. Nesnelerin önem derecelerine göre farklı sınıflara atandığı ve farklı ağırlıklarla hesaplandığı bu yöntemde 4 farklı algoritma kullanılmıştır: FCN, SegNet,

[17] Multiple Object Tracking Accuracy

[18] Importance Aware Loss

Enet, ERFNet. Araştırma, adil bir karşılaştırma yapmak için FCN, SegNet, Enet, ERFNet, FCN+Uni, SegNet+Uni, Enet+Uni, ERFNet+Uni ("Uni", çapraz entropi kaybında[19] tüm sınıflar için tek tip ağırlıkları tanımlar.), FCN+IAL, SegNet+IAL, Enet+IAL, ERFNet+IAL testlerini 3 grupta yapmıştır. Değerlendirmelere göre CNN (FCN, SegNet, Enet,ERFNet) + IAL diğer gruplara göre daha iyi sonuçlar verdiği görülmüştür (B. Chen, Gong, & Yang, 2019: 144). Sun ve arkadaşları beklenmedik engellerle anlık olarak başa çıkabilmek için görüntülerde derinlik bilgisiyle birlikte gerçek zamanlı birleşim semantik segmentasyon[20] çalışmalarına dikkat çekmiş ve bu konuda RFNet yaklaşımını önermişlerdir. Bu algoritmanın kodlayıcı kısmı, girdi verileri açısından RGB ve Derinlik görüntülerinin özelliklerini ayrı ayrı çıkarmak için iki bağımsız bileşenden oluşur. Bunun için uygun mimarisi nedeniyle ResNet-18'i seçmişlerdir. RFNet'e göre ResNet-18'in her katmanından sonra derinlik bileşeninin özellik çıktıları RGB bileşenini besler. Sonrasında bu girdiler Attention Feature Complementary modülünde birleştirilir. Ardından Spatial Pyramid Pooling bloğu tarafından iki bileşenle birleştirilmiş RGB-D özellik haritaları üretir. Son olarak RGB bileşeninden direk bir bağlantıyla üretilen özellik haritalarını çözünürlüğünü restore etmek ve derinlik bileşenini atlamak için up-sampling modünü kullanmışlardır. Performans testi için Cityspaces dataseti üzerinde SwiftNet ile karşılaştırma yapılmış, test sonuçları RFNet'in SwiftNet'e oranla daha yüksek doğruluk oranı verdiği görülmüştür (Sun, Yang, Hu, Hu, & Wang, 2020: 5). Naresh ve arkadaşları otonom araç senaryoların için semantik segmentasyonda kullanılmak üzere CNN temelli bir encoder-decoder yapısı önerdiler. Bu yapının encoder tarafı 13 konvolüsyon katmanlı VGG-16 mimarisine dayanmaktadır. Decoder yapısı da encoder yapısıyla benzerlik göstermektedir. Artık öğrenmenin de kullanıldığı bu model, SegNet ve ENet ile performans değerlendirmesi bakımından karşılaştırmaya tabi tutulmuş, CamVid ve Cityspace verisetlerinin kullanıldığı bu araştırmada sonuçlar önerilen modelin diğer iki yaklaşıma göre daha iyi performans gösterdiği sonucuna ulaşılmıştır (Naresh, Little, & O'Connor, 2018: 1054). Chen ve arkadaşları otonom sürüş sistemlerinde doğru yol işareti

[19] Cross Entropy Loss

[20] Real-Time Fusion Sematic Segmentation

çıkarma problemine çözüm üretmeye çalışmışlardır. Yol işaretlerinin karmaşıklığını ve hassasiyetini göz önünde bulundurarak derin öğrenme temelli DFPN[21] modelini önerdiler. DFPN, sığ özellik kanallarını derin özellik kanallarıyla birleştirerek yüksek çözünürlüklü sığ özellik haritalarının ve bol detaylı resimlerin derin özellikleri kullanabilmesini sağladı. Model Mask R-CNN ve Mask Scoring R-CNN mimarilerinden ilham alınarak geliştirildi. Önerilen model yol işaretlerini çıkarmak için uçtan uca mobil lazer tarayıcı[22] ile eğitildi. Optimize etmek için focal loss fonksiyonu kullanıldı. Sonuçlar bakımından mevcut diğer modellerden daha iyi performans gösterdiği ifade edilmiştir (S. Chen vd., 2021: 792). Lowphansirikul ve arkadaşları derin öğrenme kullanılarak nokta bulutlarından 3 boyutlu bir semantik segmentasyon çalışması yapmışlardır. 3 boyutlu tarayıcılarla topladıkları verileri 3 farklı algoritma önererek karşılaştırmışlardır: PointCNN, PointNet, SPGraph. Bu algoritmalar oluşturulmuş veriseti üzerinde eğitilmiş ve doğrulukları karşılaştırılmıştır. Yapılan değerlendirmelere göre sırasıyla: %72.7, %83 ve %83.4 doğruluk oranlarına ulaşılmıştır (Lowphansirikul, Kim, Vinayaraj, & Tuarob, 2019: 243). Yan ve ekibi tarafından bir trafik sahnesinde bağlamı ilişkilendirmek için GRU ve self-attention mekanizması kullanarak semantik segmentasyon çalışması yapılmıştır. Makale, özellik-uzay korelasyonu, resim düzleminde uzun mesafede yayılmış bilgi ve uzun mesafe sekans bilgisi olmak 3 noktayı birleştirmektedir. Daha iyi bir performans için self-atttention mekanizmasını ve bi-directional GRU algoritmalarını kullanmışlardır. Cityspaces, KITTY, Mapillary, CamVid veritabanlarında yapılan eğitimlerin ardından önerilen metodun sonuçları oldukça güçlü çıkmıştır (Yan, Wang, Li, Zhang, & Yang, 2020: 300). Lo Bianco ve arkadaşları tarafından yapılan başka bir çalışmada RGB bilgilerini kullanarak hem yol çizgilerini hem de yola katılan diğer nesneleri tespit edebilen bir semantik segmentasyon yapısı oluşturulmuştur. ERFNet mimarisinin temel olarak kullanıldığı bu mimaride hem hesaplama maliyeti tasarrufu hem de performans artışı hedeflenmiştir. Geniş ölçekli BDD100K veriseti üzerinde yapılan eğitim sonucuna göre önerilen metodun hem yol çizgilerini hem de yola katılan diğer nesneleri başarılı bir

[21] Dense Feature Pyramid Network

[22] Mobil Laser Scanning

şekilde segmente ettiği sonucuna varılmıştır (Bianco, Beltrán, López, García, & Al-Kaff, 2020: 8). Choi ve arkadaşları tarafından yapılan bir çalışmada encoder ve decoder arasındaki bağlantıları atlamadan sadece decoder özelliklerini kullanarak ADFNet (ERFNet ve ENet tabanlı ağların biriktirilmiş kod çözücü özelliklerine sahip bir sinir ağı) ağını kullanmışlardır. Yazarların amacı modelin performansını yükselterek hesaplama maliyetlerinin düşürülmesidir. Yapılan değerlendirmelerde sonuçların başarılı olduğunu belirtmişlerdir (Choi, Ahn, Kim, & Jeon, 2020: 562). Başka bir araştırma grubu tarafından yardımcı donanım olarak kameranın kullanıldığı, ana donanım olarak da milimetrik dalga radar cihazının kullanıldığı ve bu donanımlardan gelen verilerin birleştirildiği bir metod önermişlerdir. Bu yöntemde nesne tespit algoritması olarak Faster R-CNN mimarisi kullanılmıştır. Hedef sekanstaki gözlemlenen değerleri eşleştirmek için Mahalanobis mesafesini kullanmışlardır ve veri füzyonu Joint Probabilistic Data-Association metoduna dayanmaktadır. Sonuç olarak çeşitli hava koşullarında radar ve kameranın, tek sensörlü algılama sistemlerinden daha iyi performans gerçekleştirdiği görülmüştür (Z. Liu vd., 2022: 6649). Dinamik nesnelerin tespiti ve takibi otonom sürüş sistemlerinde sürüş güvenliği açısından oldukça önemli bir konudur. Şu anki otonom sürüş sistemlerinin, bir nesnenin görüntüden çıktıktan sonra nesneyi yeniden yakalayıp onu takip edememesi gibi bazı sınırlılıkları mevcuttur. Bu sorunla başa çıkabilmek için Zhao ve arkadaşları 2 bileşenli bir sistem önermişlerdir: ilki LIDAR ve kamera tabanlı bir 3 boyutlu görüntü takip algoritması, diğeri kameranın görüş açısından çıkan nesnelerin kamera açısına yeniden girdiğinde takibe devam edebilen yeniden takip mekanizması. Önerilen algoritma takip gerektiren birçok nesnenin bulunduğu senaryolarda eşzamanlı olarak bu nesnelerin takibinde sınırlılıklar yaşasa da mevcut yöntemlerle karşılaştırıldığında önerilen metodun sistemin güvenilirlik ihtiyacını karşıladığı yapılan nicel ve nitel analizler sonucunda görülmüştür (L. Zhao, Wang, Su, Liu, & Yang, 2020: 10870). Kaldırımlarda bisiklet kazalarının artması sebebiyle birçok araştırmacı dikkatini bu alana yoğunlaştırmış durumdadır. Burada kullanılan algoritmalar bisikleti ve üstündeki sürücüyü şekil bütünlüğünden dolayı ayırt edememektedir. Bu sorunun üstesinden gelmek için bir grup araştırmacı yayaları ve bisiklet sürücülerini ayırt etmek için VGG16 tabanlı bir nesne tespiti algoritması geliştirmişlerdir. Yapılan çalışmada bisikleti sürücüleri, yayalar ve bisikletler ayrı sınıflar halinde değerlendirilmiştir. Bu algoritmada 15 bin resim

eğitilmiştir. Bu şekilde yapılan değerlendirme sonuçlarına göre bisikletler %80.7 oranında, bisiklet sürücüleri %69.2, yayalar %85.1 oranında tanımlanırken genel değerlendirme sonucunda göre de algoritmanın başarısı %78.3 olarak gözlemlenmiştir (Ishii, Tsuichihara, Takemura, & Mizoguchi, 2020: 721). Bununla birlikte sahnedeki küçük ölçekli yayalar ve gizli kalmış yayaların tespiti konusunda da zorluklar yaşanmaktadır. Bu sorunun çözebilmek için bir grup araştırmacı bir çözüm yolu önermiştir. İlk alt ağda yayaların ayırt edici özelliklerini tespit edebilmesi için çok katmanlı bir özellik çıkartıcı kullanmışlardır. İkinci alt ağda gizli kalmış yayaları tespit edebilmek için deforme edilebilir ilgi alanı havuzunu[23] kullanmışlardır. CityPerson verisetinde yapılan deneysel çalışmalar neticesinde küçük ölçekli ve gizli kalmış yaya tespitinde sırasıyla %40.78 ve %34.60 kayıp oranlarına ulaşırken ikinci en iyi performans gösteren modelde sırasıyla %6 ve %5.87 kayıp oranlarına ulaşmışlardır (T. Liu vd., 2021: 764). Zhao ve arkadaşları 3D LIDAR ve kamera görüntülerinden elde edilen bilgileri birleştirerek yeni bir nesne tespiti metodu önermişlerdir. Öncelikle 3D LIDAR tarafından nesne-bölgeleri adayları önerilir. Sonra bu öneriler ilgi alanlarının seçildiği (region of interest) görüntü alanına eşlenir. Ardından bu veriler CNN'e girdi olarak verilir. KITTI veri kümesi üzerinde elde edilen değerlendirme sonuçlarının şunları gösterdiği söylenmiştir: ilk olarak binlerce aday nesne-bölge önerisi üreten kayan pencerelerin aksine, 3D LIDAR kare başına ortalama 86 gerçek aday sağladığı ve minimum duyarlılık oranı %95'ten daha iyi olduğu, bunun da özellik çıkarma süresini büyük ölçüde azalttığı; ikinci olarak önerilen yöntemin her karesi için ortalama işlem süresinin 66,79 ms olduğu, bunun da otonom araçların gerçek zamanlı talebini karşıladığı; son olarak önerilen yöntemin orta zorluk seviyesindeki otomobiller ve yayalar için ortalama tanımlama doğrulukları sırasıyla %89,04 ve %78,18 olduğu, bunun da önceki yöntemlerin çoğundan daha iyi olduğu sonuçlarına ulaşılmıştır (X. Zhao, Sun, Xu, Min, & Yu, 2020: 4911). Başka bir grup araştırmacı tarafından otonom araçları için CNN kullanarak gerçek zamanlı bir yaya tespit modeli tasarlanmıştır. Standart CNN katmanlarının kullanıldığı metod INRIA Person veriseti, PETA–CUHK veriseti ve kameradan toplanan gerçek zamanlı veriler

[23] Deformable Regional Region of İnterest Pooling

üzerinde test edilmiştir. Veriseti özelliklerine göre değişken olmakla birlikte test sonuçları %96 ile %100 arasında bir başarı gösterdiği söylenmiştir (Pranav & Manikandan, 2020: 159). Sınırlı hafıza ve hesaplama gücü derin evrişimsel sinir ağlarının kullanıldığı otonom araç sistemlerinde henüz aşılmamış bir sorun olarak durmaktadır. Bu problemin üstesinden gelmek için hesaplanması kolay ve güçlü dedektörler tasarlamak gereklidir. Bunun için daha hafif ve hızlı çalışan "Group Convolution" algoritması önerilmiştir. Fakat algoritmanın varolan kuralları Group Convolution'ın tespit hızını en yüksek seviyeye çıkarmak için grupların optimal sayısını göstermemektedir. Bu makalede optimal sayıyı bulmak için DenseLightNet adında bir nesne tespit algoritması sunulmaktadır. Önerilen algoritmanın YOLO v3'den 3 kat daha hızlı olduğu görülmüştür (L. Chen, Ding, Zou, Chen, & Li, 2020: 10605). Bir grup araştırmacı tarafından Single Shot Detection (SSD) olarak adlandırılan bir derin sinir ağı tasarlanmıştır. SSD bir ana ağ, bir de yardımcı ağdan oluşmaktadır. İyi bir sınıflandırma için temel ağ VGGNet kullanırken yardımcı ağ çoklu özellik haritalarını kullanmıştır. Algoritma KITTI verisetinde eğitilmiş ve şu an kullanılan orjinal algoritmadan %6 daha iyi bir performans gerçekleştirmiştir (H. Kim, Lee, Yim, Park, & Kim, 2017). Yang ve arkadaşları tarafından yapılan bir çalışmada eşzamanlı olarak insanların, araçların, yol çizgilerinin ve motorsuz araçların tespitine yönelik RGB-D resimlerinin kullanarak bir algoritma tasarımı yapılmıştır. Doğruluk ve hız değerlerinin geliştirilmesi için 2 network kullanılmıştır. Birisi yol çizgilerinin segment etmesi için LaneNet, insanlar, araçlar ve motorsuz araçların tespiti için de Faster R-CNN mimarileri kullanılmıştır. Yüksek doğruluk elde etmek amacıyla gündüz ve gece çekilen fotoğraflardan oluşan veriseti araştırma grubu tarafından toplanmıştır. Faster R-CNN mimarisinde gündüz çekilen eğitim verilerinin mAP %91 iken gece fotoğrafları için %86'dır. LaneNet algoritması için precision ve recall değerlerine bakılmıştır. Gündüz verileri için precision değeri %95, gece için %86 olurken, recall değerleri gündüz ve gece için sıralı bir şekilde %95, %87 şeklinde ölçülmüştür (Yang, Wang, Wang, & Li, 2020: 11965). Prabhakar ve arkadaşları otonom bir aracın etrafındaki engelleri tespit ve sınıflandırma konusunda bir çalışma yapmışlardır. Çalışma, yoldaki araçlar, yayalar, hayvanlar gibi otonom araca engel olarak tanımlanabilecek nesneleri tanıma ve sınıflalandırma üzerine yoğunlaşmıştır. Bölge temelli bir derin öğrenme algoritması ile PASCAL VOC veritabanında çalışılmıştır. Yapılan değerlendirme sonuçlarına göre nesne

tespit hızının 100 ms altında olduğu görülmüş, verisetindeki farklı sınıflara göre kesinlik değerlerinin %70 üzerinde olduğu belirlenmiştir (Prabhakar vd., 2017). Başka bir çalışmada nesnelerin devam eden güzergahı tahmin edilmeye çalışılmıştır. Diğer bir ifadeyle bu çalışma, içinde bisiklet, araç, yaya vb. bulunan karışık bir sahneyi anlamaya çalışır. Önerilen metod gözlemlenmiş olan 3 farklı güzergahı girdi olarak alır ve çıktı olarak sonraki güzergahı verir. Modelin, GRU ve LSTM ile karşılaştırıldığında daha düşük MAE[24] değerleri aldığı tespit edilmiştir (Hsu, Huang, & Chiang, 2020: 265). Başka bir çalışmada fren lambalarını tanıma üzerine yapılmıştır. Bunun için araştırmacılar tarafından farklı hava şartlarını ve günün farklı zaman dilimlerini içeren gerçek dünya verileri toplanmıştır. Bu metoda göre önce veritabanında fren lambaları desenleri öğrenilir. Araçlar derin öğrenme mimarilerinden bir sınıflandırıcı kullanarak "fren" ya da "normal" olarak sınıflandırılır. Bu esnada doğruluğu artırmak için yol segmentasyonu yapılmış ve ROI[25] uygulanmıştır. ROI, araç boyutlarından ve kaybolan nokta[26] ile belirlenmiştir. Bu bilgiler ışığında yapılan testler sonucunda %99'luk bir doğruluk oranı elde edildiği belirtilmiştir (J. G. Wang vd., 2016: 820). Başka bir ekip tarafından, 4 farklı evrişimsel sinir ağının kullanıldığı karşılaştırmalı bir çalışma yapılmıştır. Bu çalışmada araçlar, yayalar, trafik işaretleri ve bisikletler değerlendirilmiştir. Kullanılan ResNet50, ResNet101, Faster R-CNN Inception v2 ve SSD Inception v2 algoritmaları COCO verisetinde ön eğitimden geçmiştir. Ardından bu algoritmalar transfer öğrenme kullanılarak bir kısmı GRAZ 01 ve GRAZ 02 veriseti üzerinde bir kısmı da araştırmacılar tarafından toplanan veriler üzerinde yeniden eğitilmiştir. Yapılan değerlendirmeler sonucunda %85 doğruluk oranı ile en iyi sonucu veren algoritmanın ResNet101 olduğu görülmüştür (Ozturk, Koker, Eldogan, & Karayel, 2020). Shi ve arkadaşları PointRCNN adında bir nesne tespit algoritması önermişlerdir. Bu metod 2 aşamadan oluşmakta olup ilk aşamada baştan sona ham 3 boyutlu görseller önerilir. İkinci aşamada sonuçları elde etmek için kanonik koordinatlarda yeniden değerlendirilir. İlk aşamadadaki alt ağ,

[24] Mean Absolute Error

[25] Region of Interest

[26] Vanishing Point

sahnenin önündeki ve arkasındaki görüntüleri bölerek 3 boyutlu öneriler üretir. İkinci aşamadaki alt ağ, ağın daha iyi öğrenebilmesi için her önerinin noktalarını ağın daha iyi öğrenebilmesi için kanonik koordinatlara dönüştürür. Nokta bulutunun girdi olarak verildiği önerilen metod KITTI dataseti üzerinde test edilmiş, dikkate değer sonuçlar üretilmiştir (Shi, Wang, & Li, 2019: 7).

ÜÇÜNCÜ BÖLÜM

YÖNTEM

Bu bölüm, yol yüzeyindeki çukurların tespiti için geliştirilen derin öğrenme modelinde kullanılan veri setlerini ve ön işleme süreçlerini içermektedir. Özellikle farklı çevresel koşullarda elde edilmiş görüntülerle çalışılarak, gerçek dünya uygulamalarında karşılaşılabilecek çeşitli durumlar için modelin dayanıklılığı sağlanmaya çalışılmıştır. Ayrıca çalışmamızda kullanılan derin öğrenme modelleri de bu bölümde incelenmektedir.

3.1. Veri Seti ve Veri Ön İşleme

Modelimiz, toplamda 5825 eğitim, 1948 doğrulama ve 1576 test verisi kullanılarak eğitilmiştir. Eğitim, doğrulama ve test veri setleri, yol üzerindeki çukur içeren (pozitif) ve içermeyen (negatif) görüntülerin dengeli bir dağılımına sahiptir. Böylece modelin her iki sınıfı da adil bir şekilde öğrenmesi sağlanmıştır.

Bu çalışma kapsamında Pothole Dataset (Nienaber, Booysen, & Kroon, 2015; Nienaber, Kroon, & Booysen, 2015) ve Cracks and Potholes in Road Dataset (Ninja, 2025) olmak üzere iki farklı veri seti birleştirilerek kullanılmıştır. Şekil 3.1 ve Şekil 3.2'de çalışmada kullanılan verisetlerinden pozitif ve negatif sınıflı veri örnekleri yer almaktadır. Veri setleri, farklı aydınlatma koşulları, perspektif açıları ve yol yüzey tiplerini içerecek şekilde çeşitlendirilmiştir. Bu sayede modelin genelleme yeteneği artırılmıştır.

Kullanılan görüntülerin tamamı JPG formatında olup, 3680×2760 ve 1024×630 çözünürlüklere sahiptir. Görüntüleri ViT-LSTM, CNN ve YOLOv8 algoritmalarında kullanmak amacıyla mevcut JSON dosyaları ile etiketleme süreci gerçekleştirilmiştir. Önerdiğimiz ViT-LSTM modelinde 224x224 ve 384x384 boyutları çalışılmış, en yüksek doğruluk 384x384 boyutlarında görülmüştür. Bu nedenle resimler 384x384 boyutlarına ölçeklendirilmişir. Çalışmamızda modelimizle karşılaştırma amacıyla kullanıdığımız CNN modeli için ResNet50 omurgası kullanılmıştır. ResNet50 omurgası için, ImageNet yarışmalarında hesaplama verimliliği açısından 224x224 tercih edilmektedir. Bu nedenle CNN modelinde resimler 224x224 boyutlarına ölçeklendirilmiştir. YOLOv8 modelinde

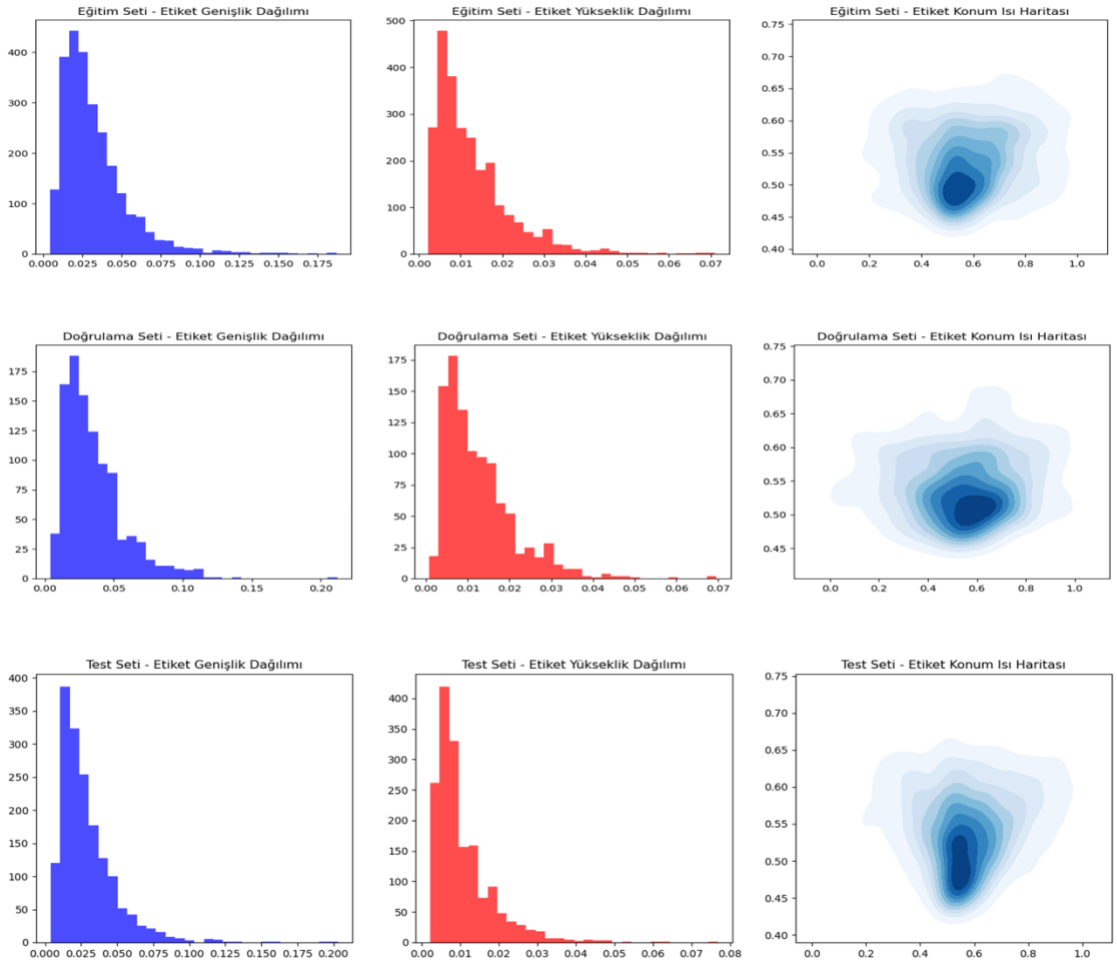
Şekil 3.1: Pozitif etiketli fotoğraflar örneği.

Kaynak: Nienaber, Booysen, vd., 2015

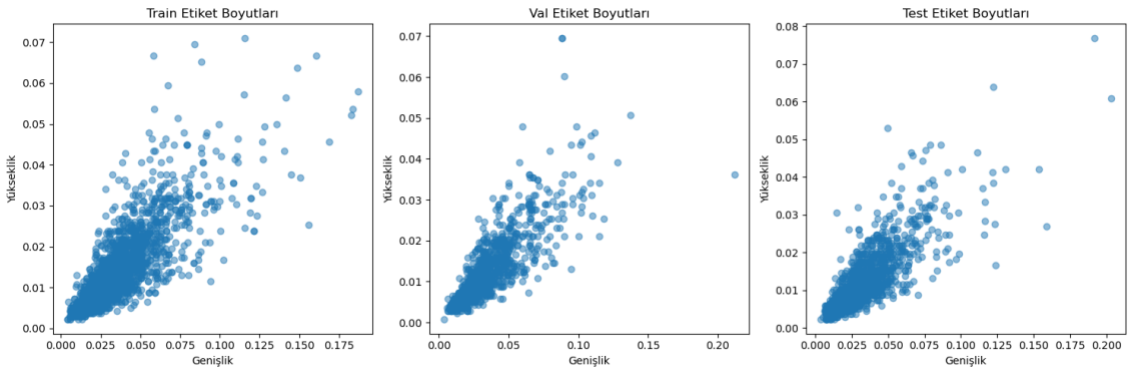
Şekil 3.2: Negatif etiketli fotoğraflar örneği.

Kaynak: Nienaber, Booysen, vd., 2015

640×640 boyutları, hem büyük nesneleri hem de küçük nesneleri tespit etmek için yeterli bir çözünürlüktür. Bu sebeple resimler YOLOv8 modeli için 640x640 piksel boyutlarına yeniden ölçeklendirilmiştir. YOLOv8 modeli için JSON formatındaki etiketler, sınıf ve nesnenin konum bilgisini içeren YOLO formatındaki TXT dosyalarına dönüştürülmüştür. Tablo 3.1'de, pozitif sınıfa ait bir JSON dosyası örneği gösterilmekte olup, çukurun konumu ve ek açıklamaları içermektedir. Şekil 3.3'de veri eğitim, doğrulama ve test veri setlerinin yüseklik, genişlik ve konum ısı haritası verileri yer almaktadır. Çukur görüntüleri çalıştığımız için yükseklik ve genişlik bilgilerinin tutarlı olduğunu görüyoruz. Bununla birlikte konum ısı haritasın baktığımızda çukurların genelde görüntünün merkezinde yer aldığınız söyleyebiliriz. Ayrıca Şekil 3.4'de veri etiketlerinin dağılımına bakıldığında verisetinin dengeli bir yapıya sahip olduğu da görülmektedir.



Şekil 3.3: Verilerin genişlik, yükseklik ve konum ısı haritası



Şekil 3.4: Veri etiketlerinin dağılımı.

Veri setinin genelleme yeteneğini artırmak ve modelin farklı koşullara adapte olmasını sağlamak amacıyla geometrik dönüşümler (döndürme, çevirme, kırpma), ışık ve kontrast düzenlemeleri, bulanıklaştırma ve gürültü ekleme gibi çeşitli veri artırma teknikleri uygulanmıştır. Böylece modelin farklı ışıklandırma, kamera açısı, yüzey dokusu ve çevresel faktörlerden etkilenmeden doğru tahmin yapması amaçlanmıştır.

Çalışma kapsamında ele alınan çukur tespiti problemi, kendine özgü bazı zorluklar içermektedir. Çukurların şekil, boyut ve derinlik açısından geniş bir çeşitliliğe sahip olması, modelin bu yapıları yüksek doğrulukla tespit etmesini zorlaştırmaktadır. Çukurun fiziksel özelliklerine bağlı olarak içine düşen gölgelerin yoğunluğu ve renk tonlarında oluşan farklılıklar, tespit sürecini karmaşık hale getirebilmektedir. Ayrıca, çukurların su veya farklı sıvılarla dolu olması, yüzeyin yansıtıcılığını ve kontrastını değiştirerek modelin genelleme kabiliyetini zorlayabilmektedir. Bunun yanı sıra, yol yüzeyine düşen ağaç, tabela veya diğer nesnelerin gölgeleri, çukurların görünürlüğünü azaltarak tespit sürecini zorlaştırabilmektedir. Farklı zemin türleri (asfalt, beton, taş döşeme vb.) çukurların görsel özelliklerinde değişikliklere yol açmakta olup, tüm bu zorluklara rağmen otonom sürüş esnasında yol ve sürüş güvenliği açısından modelin, farklı yüzey koşullarında tutarlı bir performans sergilemesi gerekmektedir. Mezkur güçlüklerle başa çıkabilmek amacıyla modelin genelleme yeteneği artırılarak ve farklı aydınlatma koşulları, yüzey tipleri, çevresel faktörler göz önünde bulundurularak veri çeşitliliği sağlanmış, böylece tespit performansının yüksek doğruluk seviyesinde korunması hedeflenmiştir.

Tablo 3.1: Çukur hakkında konum bilgilerini içeren .json dosyası örneği


```
{
   "objects": [
      {
         "id": 17448212,
         "classId": 37358,
         "description": "",
         "geometryType": "rectangle",
         "labelerLogin": "gr@datasetninja.com",
         "createdAt": "2023-08-22T16:25:49.631Z",
         "updatedAt": "2023-08-22T16:25:49.631Z",
         "tags": [],
         "classTitle": "çukur",
         "points": {
            "exterior": [ ……..   #Konum bilgileri
               [
                  1368,
                  1732
               ],
               [
                  1482,
                  1770
               ]
            ],
            "interior": []
         }
      },
```


Kaynak: Ninja, 2025

3.2. Yöntem ve Teknikler

Çukur tespiti, temelde bir görüntü sınıflandırma ve nesne tanıma problemidir. Bu alanda uygulanan yöntemler, geleneksel görüntü işleme tekniklerinden modern derin öğrenme yaklaşımlarına kadar geniş bir yelpazede değişmektedir.

Geleneksel görüntü işleme yaklaşımlarında, kenar algılama, doku analizi ve morfolojik işlemler gibi teknikler kullanılarak çukur tespiti yapılabilmektedir. Ancak bu yöntemler, değişken ışık koşulları, gölgeler ve farklı yol dokuları gibi zorlu çevre koşullarında sınırlı başarı göstermektedir.

Derin öğrenme modelleri ise, bu zorlukların üstesinden gelebilecek daha güçlü özellik çıkarma ve sınıflandırma yetenekleri sunmaktadır. Bu modeller, büyük veri setleri üzerinde eğitilerek, farklı koşullar altında bile çukur yapılarını tanıyabilme kabiliyeti kazanmaktadır. Çalışmamızda, modern derin öğrenme mimarilerini kullanarak yüksek doğrulukta bir çukur tespit sistemi geliştirilmiştir.

Bu çalışmada, çukur tespiti için derin öğrenme modeli tabanlı bir çözüm sunulmaktadır. Özellikle, yol yüzeyindeki çukurları tespit etmek amacıyla ViT (Vision Transformer) modelinin global bağlamı yorumlamadaki güçlü yönleri ile küçük nesneleri daha iyi yakalamadaki yetenekleri ve LSTM (Long Short-Term Memory) mimarisinin zamansal bağıntıları analiz edip yüksek doğruluk sağlaması gibi özelliklerinin birleştirildiği yenilikçi bir hibrit model geliştirilmiştir.

3.2.1. Vision Transformer

Vision Transformer (ViT), doğal dil işleme alanında devrim yaratan Transformer mimarisinin görüntü işleme alanına uyarlanmış halidir. Google Brain ekibinden Dosovitsky ve arkadaşları tarafından geliştirilen bu mimari, dikkat mekanizması[27] sayesinde görüntülerin küresel bağlamını anlayabilme yeteneği sunmaktadır (Dosovitskiy vd., 2020).

Geleneksel CNN mimarilerinden farklı olarak ViT, bir görüntüyü öncelikle sabit boyutlu parçalara böler. Örneğin, 384x384 piksel boyutundaki bir görüntü, 16x16 boyutundaki parçalara bölündüğünde, toplam 576 (24x24) parça elde edilir. Bu parçalar, bir pozisyon kodlaması[28] ile zenginleştirilerek Transformer mimarisine beslenir.

ViT'nin işleyişi aşağıdaki adımlardan oluşmaktadır:

[27] Attention Mechanism

[28] Positional Encoding

1. **Görüntü Bölümleme[29]:** Görüntü, düzenli sabit boyutlu parçalara bölünür.

2. **Doğrusal Projeksiyonlar:** Her bir parça, doğrusal projeksiyon uygulanarak gömme vektörlerine[30] dönüştürülür.

3. **Pozisyon Kodlaması:** Parçaların konumsal bilgisini korumak için pozisyon kodlaması eklenir.

4. **Transformer Kodlayıcı[31]:** Parçalar, çok başlı dikkat mekanizması[32] ve ileri besleme ağları[33] içeren Transformer katmanlarından geçirilir.

5. **Sınıflandırma Başlığı[34]:** Son olarak, elde edilen özellikler üzerinde sınıflandırma yapılır.

ViT, görüntü işleme görevlerinde CNN'lere göre bazı avantajları sunar. Küresel bağlam anlama özelliği ile dikkat mekanizması sayesinde, görüntünün tamamı arasındaki ilişkileri öğrenebilir. Uzun menzilli bağımlılıklar özelliği ile görüntüdeki uzak pikseller arasındaki ilişkileri kavrayabilir. Ölçeklenebilirliği sayesinde de model boyutu ve miktarı arttıkça performansı da artma eğilimindedir. Ancak, ViT'nin bazı kısıtlılıkları da bulunmaktadır. Küçük veri setlerinde CNN'lere göre daha düşük performans gösterir. CNN'lere kıyasla daha fazla hesaplama maliyeti gerektirir. Şekil 3.5'de ViT modelinin genel görünümü yer almaktadır.

[29] Image Patching

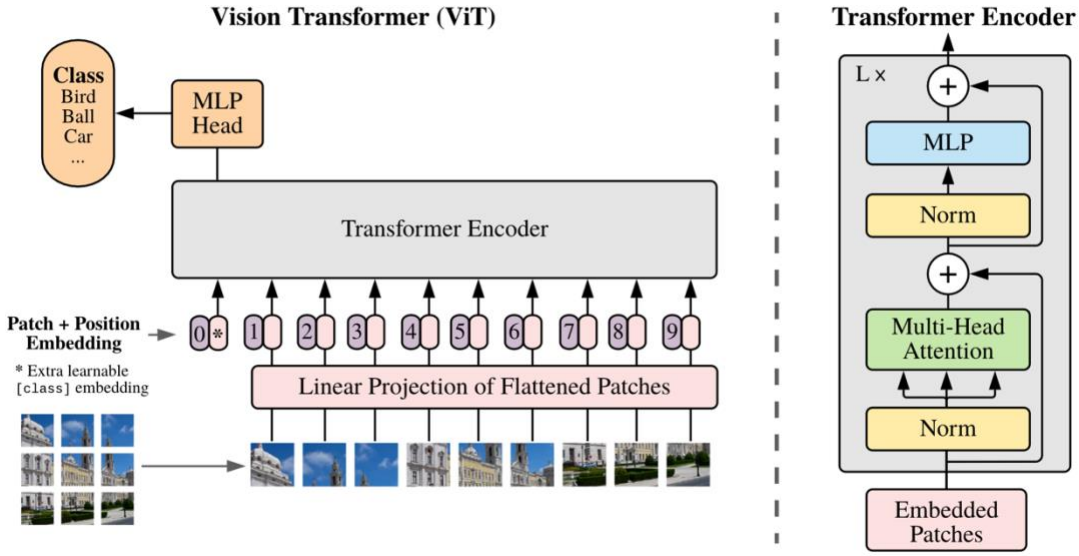
[30] Embedding Vectors

[31] Encoders

[32] Multi-Head Attention

[33] Feed Forward Networks

[34] Classification Head



Şekil 3.5: Vision Transformer modeli.

Kaynak: Dosovitskiy vd., 2020: 3

3.2.2. Konvolüsyonel Sinir Ağları (CNN)

Konvolüsyonel Sinir Ağları (CNN), görüntü işleme alanında yaygın olarak kullanılan, biyolojik görme sisteminden esinlenilerek tasarlanmış derin öğrenme modellerindendir. CNN mimarisi, temel olarak aşağıdaki katmanlardan oluşur:

1. Konvolüsyon Katmanları[35]**:** Bu katmanlar, görüntü üzerinde filtreler (kerneller) kaydırarak özellik haritaları[36] oluşturur. Her filtre, belirli bir görsel deseni (kenarlar, köşeler, dokular vb.) tanımak için eğitilir. Örneğin, 3x3 boyutundaki bir filtre, görüntü üzerinde kaydırılarak her pozisyonda bir konvolüsyon işlemi gerçekleştirir. Örneğin kenar algılama için [[1, 0, -1], [1, 0, -1], [1, 0, -1]] matrisi kullanılır.

[35] Convolutional Layers

[36] Feature Maps

2. Aktivasyon Fonksiyonları: Konvolüsyon işlemi sonrasında, doğrusal olmayan bir aktivasyon fonksiyonu uygulanır. En yaygın kullanılan aktivasyon fonksiyonu ReLU'dur. Rectified Linear Unit formülü:

$$f(x) = max(0, x) \tag{3.1}$$

ReLU, negatif değerleri sıfırlarken, pozitif değerleri olduğu gibi bırakır. Bu, modelin doğrusal olmayan ilişkileri öğrenmesini sağlar.

3. Havuzlama Katmanları[37]: Bu katmanlar, özellik haritalarının boyutunu küçülterek hesaplama maliyetini azaltır ve modelin uzamsal değişimlere karşı dayanıklılığını artırır. En yaygın havuzlama işlemleri:

- **Maksimum Havuzlama[38]**: Belirli bir bölgedeki en yüksek değeri alır.
- **Ortalama Havuzlama[39]**: Belirli bir bölgedeki değerlerin ortalamasını alır.
- **Tam Bağlantılı Katmanlar[40]**: Son konvolüsyon katmanları sonrasında, elde edilen özellikler düzleştirilir ve tam bağlantılı katmanlara beslenir. Bu katmanlar, yüksek seviyeli özellikleri birleştirerek sınıflandırma yapar.

Çukur tespiti için kullanılabilecek çeşitli modern CNN temelli mimariler de vardır:

- **ResNet[41]:** Artık bağlantılar sayesinde derin ağların eğitimini kolaylaştırır.
- **MobileNet:** Mobil ve gömülü cihazlar için optimize edilmiş, hafif bir CNN mimarisidir.
- **EfficientNet:** Ağ genişliği, derinliği ve çözünürlüğü arasında optimal bir denge kurarak yüksek performans sağlar.

[37] Pooling Layers

[38] Max Pooling

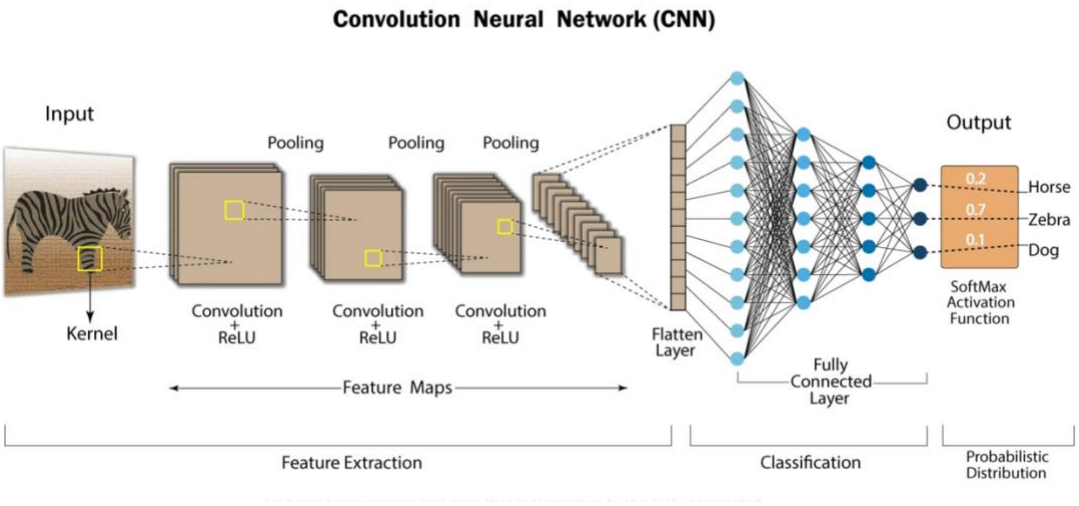
[39] Average Pooling

[40] Fully Connected Layers

[41] Residual Networks

- **UNet:** Özellikle semantik segmentasyon görevleri için etkilidir, çukurların piksel seviyesinde segmentasyonu için kullanılabilir.

CNN'lerin birçok avantajı olduğu gibi dezavatajları da vardır. Örneğin; CNN'ler, filtrelerinin kısıtlı alıcı alanı[42] nedeniyle, görüntünün sadece yerel bölgelerini analiz edebilir ve bu da global bağlamın anlaşılmasını zorlaştırır. CNN'ler, nesnelerin konum, ölçek ve rotasyondaki değişimlere karşı tam olarak değişmez değildir. Standart CNN'ler, nesnelerin geometrik dönüşümlerini açıkça modelleyemez, bu nedenle aynı nesnenin farklı görünümlerini tanımak için fazla veri gerekebilir. Şekil 3.6'da CNN model mimarisi görülmektedir.



Şekil 3.6: CNN model mimarisi.

Kaynak: Machine Learning With Talha, t.y.

3.2.3 Long-Short Term Memory (LSTM)

Long Short-Term Memory (LSTM), Hochreiter ve Schmidhuber tarafından geliştirilen özel bir tekrarlayan sinir ağı (Recurrent Neural Network - RNN) türüdür. Standard

[42] Receptive Field

RNN'lerin uzun vadeli bağımlılıkları öğrenmede yaşadığı "kaybolan gradyan[43]" problemini çözmek için tasarlanmıştır (Hochreiter & Schmidhuber, 1997). LSTM mimarisinin genel görünümü Şekil 3.7'de gösterilmiştir.

LSTM'in temel yapı taşı, bir LSTM hücresidir. Bu hücre, içinde üç kapı mekanizması barındırır:

- Unutma Kapısı[44] : Hücre durumundan hangi bilgilerin atılacağını belirler.
- Giriş Kapısı[45] : Yeni gelen bilgilerden hangilerin hücre durumuna ekleneceğini kontrol eder.
- Çıkış Kapısı[46] : Hücre durumunun hangi kısımlarının çıktı olarak verileceğini belirler.

LSTM'ler, zaman serisi verileri üzerinde çalışırken önemli avantajları sunar. Örneğin, hücre durumu sayesinde uzun süreler boyunca bilgiyi koruyabilir ve eski bilgilerle yeni gelen bilgiler arasında ilişkiler kurabilir. Bir dizi içindeki her elemanı, önceki elemanların bağlamı içinde değerlendirebilir. Unutma ve giriş kapıları sayesinde, hangi bilgilerin saklanacağını ve hangi bilgilerin unutulacağını öğrenebilir. Aynı zamanda hücre durumu ve kapı mekanizmaları, gradyanların kaybolması veya patlaması sorununu azaltır. Çukur tespiti genellikle statik görüntüler üzerinde yapılsa da, gerçek dünya uygulamalarında (örneğin, araç kameralarından alınan video akışı), çukur tespiti zamansal bir boyut kazanır. Bu bağlamda LSTM'ler aşağıdaki avantajları sağlar:

1. Ardışık Kareler Arasındaki İlişkileri Modelleme: Bir çukur, birden fazla video karesinde görünebilir. LSTM, bu ardışık karelerdeki çukur görünümlerini ilişkilendirerek daha güvenilir tespitler yapabilir.

2. Gürültü Filtreleme: Geçici olarak ortaya çıkan yanlış pozitifleri filtreleyebilir. Örneğin, bir kare içinde çukura benzeyen bir gölge, ardışık kareler incelendiğinde elenir.

[43] Vanishing Gradient

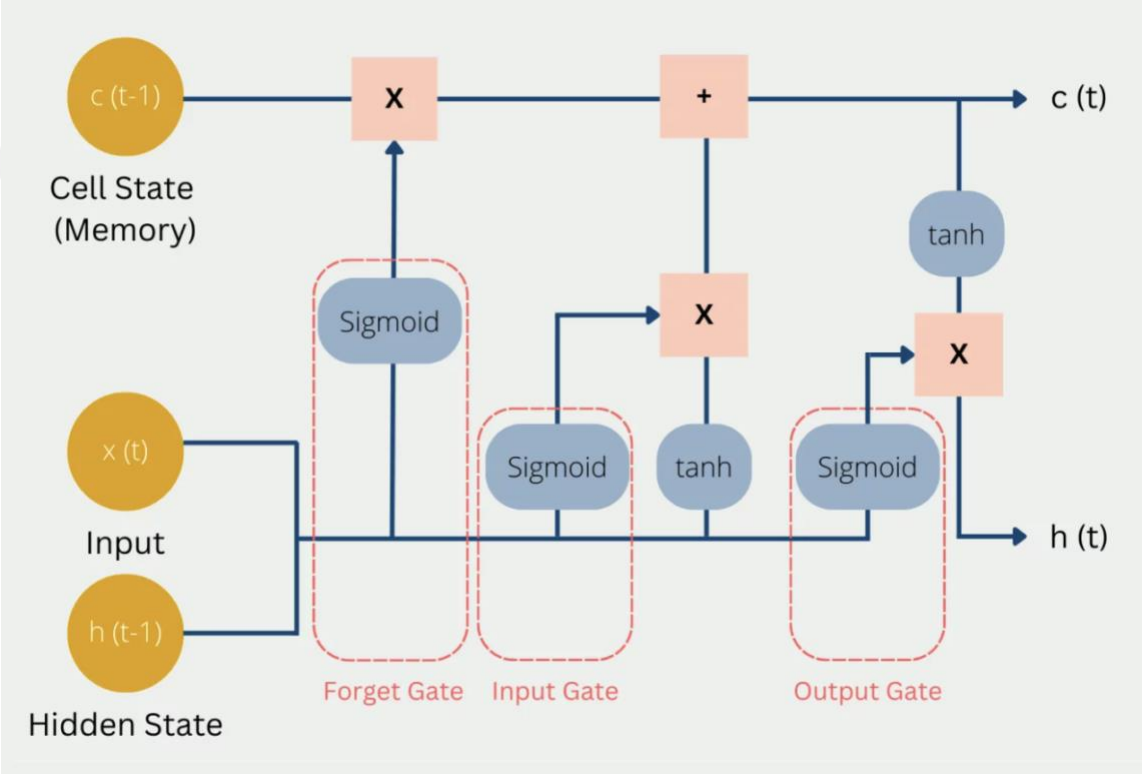
[44] Forget Gate

[45] Input Gate

[46] Output Gate

3. Hareket Bilgisini Kullanma: Aracın hareketi sırasında çukurun görünümündeki değişimleri modelleyerek, 2D yapısını daha iyi anlayabilir.

4. Tahmin Tutarlılığını Arttırma: Tek bir kare üzerinde yapılan tahminler yerine, bir dizi kare üzerinde yapılan tahminlerin ortalaması alınarak daha kararlı sonuçlar elde edilebilir.



Şekil 3.7: LSTM model mimarisi

Kaynak: Saheed, Omole, & Sabit, 2025: 9

3.2.4 You Only Look Once (YOLO)

YOLO (You Only Look Once), gerçek zamanlı nesne tespiti için geliştirilmiş, tek aşamalı bir derin öğrenme algoritmasıdır. Redmon ve arkadaşları (Redmon, Divvala, Girshick, & Farhadi, 2016) tarafından sunulan bu algoritma, nesne tespiti problemini tek bir regresyon problemi olarak ele alarak, görüntüyü tek bir geçişte işleyip hem sınıflandırma (nesne türü) hem de lokalizasyon (konum ve boyut) tahminlerini eşzamanlı gerçekleştirmektedir.

24 konvolüsyon katmanı ve 2 tam bağlantılı katmana sahiptir. Şekil 3.8'de modelin genel gönümü yer almaktadır.

YOLO algoritmasının temel çalışma prensibi şu aşamaları içerir:

1. Konvolüsyon katmanlarını kullanarak özellikleri çıkarır ve sonra bu özellikleri tam bağlantılı katmanlardan geçirerek sınıf olasıklarını tahmin eder.

2. Görüntüsü S×S'lik bir ızgara yapısına bölünür.

3. Her ızgara hücresi, bir dizi sınırlayıcı kutu[47] ve bunların güven skorlarını tahmin eder. Nesnenin merkezi hangi hücreye düşüyorsa o hücre o nesneyi tespit etmekle sorumludur.

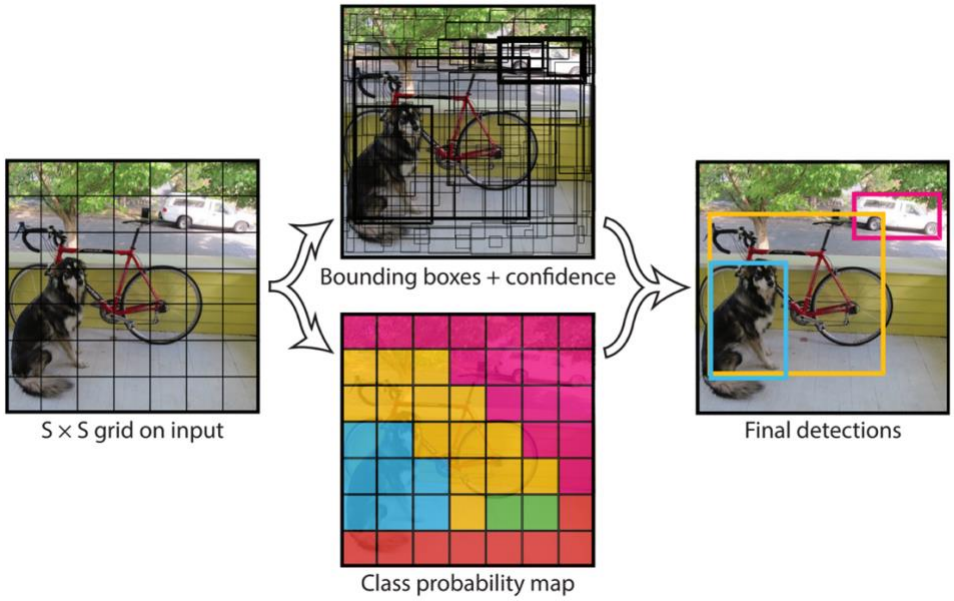
4. Her ızgara hücresi, sınırlayıcı kutuların sayısına bakmaksızın sınıf olasılıklarını tahmin eder.

5. Sınırlayıcı kutular ve sınıf olasılıkları birleştirilerek nihai tespit sonuçları elde edilir.

6. Maximum olmayan bastırma[48] tekniğiyle, çakışan ve gereksiz kutular elenerek sonuçlar optimize edilir.

[47] Bounding Box

[48] Non-Maximum Suppresion



Şekil 3.8: YOLO modeli.

Kaynak: Redmon vd., 2016: 2

DÖRDÜNCÜ BÖLÜM

UYGULAMA VE DENEYLER

4.1. Çalışma Prensibi

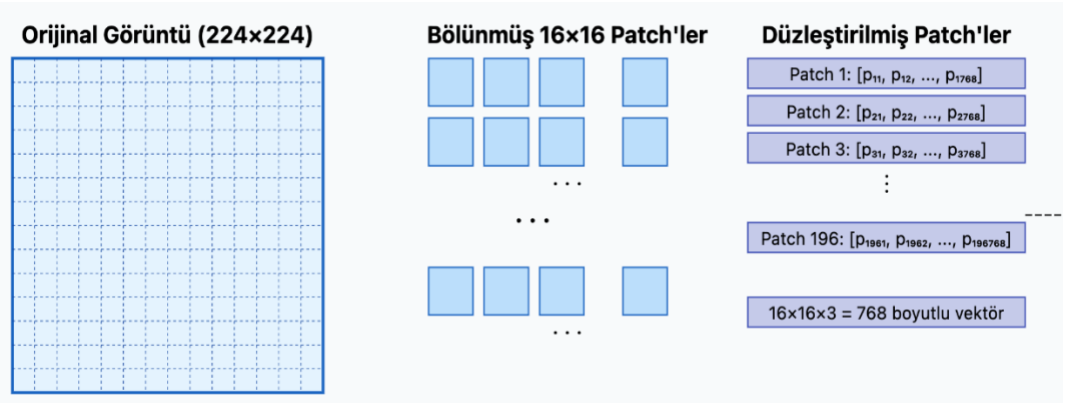
Çalışmamızda kullandığımız ViT-LSTM modelini üç alt başlık halinde alarak detaylı bir analizle açıklayacağız.

- **Vision Transformer Bileşeni:**

Vision Transformer, Doğal Dil İşleme alanında kullanılan Transformer modelinin makine görmesi alanında kullanılmak üzere resim işleme amacıyla türetilmiş bir algoritmadır. Süreç 4 adımda gerçekleşir:

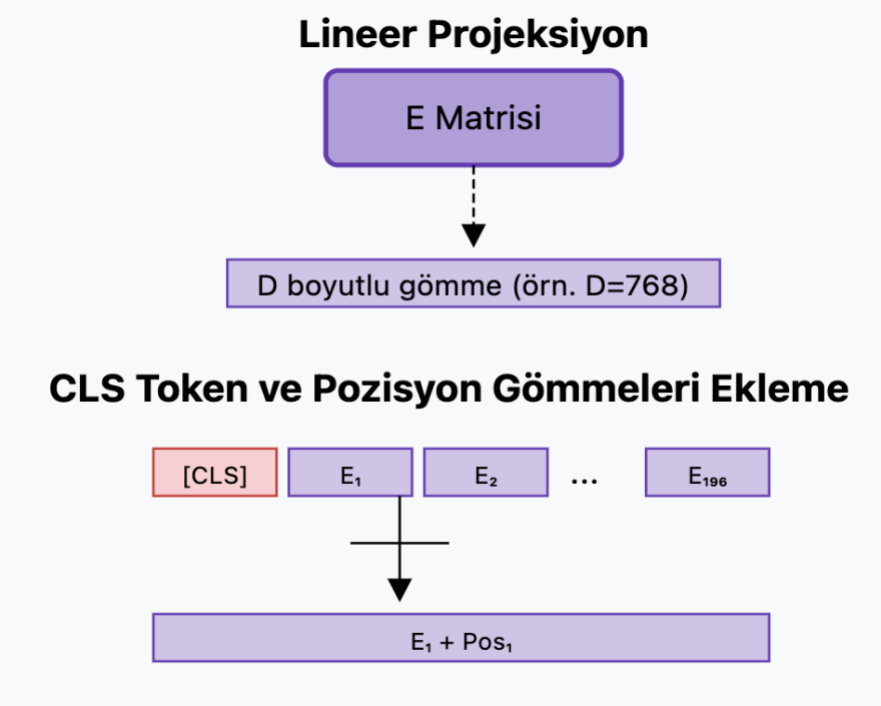
1. Görüntü Parçalama: Çalışmamızda ViT modellerinden "vit_large_patch16_384" omurgası kullanılmıştır. Bu omurga "large" versiyonunun 16x16 patch'lere bölünmüş, 384x384 resim girdisi isteyen modelidir. Bu backbone veri setimizdeki resimleri öncelikle 384x384 olacak şekilde yeniden boyutlandırmaktadır. Ardından her resmi 16x16 patch'lere bölerek bir vektör dizisi haline getirmektedir. Şekil 4.1'de görüldüğü gibi görüntü patch'lere ayrıldıktan sonra düzleştirilmekte, diğer bir ifadeyle yanyana getirilmektedir.

Görüntü Boyutu: 384 x 384
Patch boyutu: 16x16
Renk Kanalı: 3 (RGB)
Patch sayısı: $(384 / 16)^2 = 24^2 = 576$
Vektör Boyutu: 16 x 16 x 3 = 768



Şekil 4.1: Resimlerin bölünmesi

2. Doğrusal Projeksiyon: Her parça, bir doğrusal projeksiyon katmanı ile sabit boyutlu bir gömme vektörüne dönüştürülür. Her vektöre global bağlamı yakalayabilmesi için konum bilgisi de eklenir. Şekil 4.2 temsili bir görsel sunmaktadır.



Şekil 4.2: Lineer Projekesiyon

Burada bir önceki işlemden gelen 768 boyutlu düzleştirilmiş patch vektörü bir E projeksiyon matrisi ile çarpılır. E matrisi sabit olmayan, model eğitilirken öğrenilen

parametrelerden oluşur, fakat başlangıçta rastgele seçilir. Bu işlem sonucunda D boyutlu bir gömme vektörü elde edilir. İki aşamadan oluşan sürecin işlem formülü:

$$D = E \times x_p \tag{4.1}$$

D = Elde edilen yama gömmeleri[49]

E = Eğitim sırasında öğrenilen parametrelerden oluşan matris.

x_p = Düzleştirilmiş yamalar.

3. Pozisyon Kodlaması: Görüntü parçalarının konum bilgisini korumak için her parçaya öğrenilebilir pozisyon kodlaması eklenir. Gömmelerin formülü şu şekildedir:

$$z_0 = [x\_class;\ D_{1p};\ D_{2p};\ \ldots;\ D_{np}] + E_{pos} \tag{4.2}$$

z_0 = Transformer modelinin ilk katmanına giriş olarak verilen tam token dizisidir.

x_class = ViT'nin sınıflandırma tokeni.

D = Elde edilen patch gömülü vektörleri

E_{pos} = Pozisyon gömmeleri (positional embeddings) matrisidir. Bu, Transformer modeline token'ların sırasını/konumunu bildirmek için kullanılır.

ViT tarafından eklenen CLS tokeni, sınıflandırma kararları için global görüntü temsili oluşturmada kullanılır. Transformer işlemi sonrasında, bu tokenin son durumu bir sınıflandırıcıya beslenir.

4. Transformer Kodlayıcı: Google Brain ekibi tarafından geliştirilen Transformer mimarisi kodlayıcı (encoder) ve kod çözücü (decoder) yapılarından oluşur (Vaswani vd., 2017). Şekil 4.3'te, sol taraf Encoder'i, sağ taraf ise Decoder'i temsil etmektedir.

[49] Patch Embedding

Konunun daha net anlaşılması bakımından öncelikle "Öz Dikkat[50]" kavramı açıklanacak, sonra Encoder-Decoder başlıklarına geçilecektir.

⇒ ***Öz Dikkat:***

Öz dikkat kavramı bilgisayar bilimleri dünyasına Google Brain tarafından 2017 yılında yapılan "Attention is All You Need" makalesiyle girmiştir (Vaswani vd., 2017). Bu mekanizma aslında ilk olarak büyük dil modellerinde kullanılmak üzere geliştirildi. Fakat daha sonra yine Google Brain ve Google Research birimleri tarafından makine görmesi alanında kullanılmak üzere dönüştürüldü (Dosovitskiy vd., 2020). Mekanizmanın asıl amacı bir resimdeki bölünmüş herhangi bir nesnenin diğer herhangi bir nesneye ne düzeyde önem arz ettiğini bulmaktır. Birbirleri arasındaki önem düzeyi belirleyici etken olmaktadır. Bunun için 3 kavram geliştirilmiştir:

x: Girdi olarak lineer projeksiyon sonrası elde edilen gömülü vektörleri kullanır.

$W_{q\text{-}k\text{-}v}$: Ağırlık matrisidir. Eğitim sırasında öğrenilerek elde edilir. Başlangıçta küçük değerlerle başlatılır ve eğitim sırasında geri yayılım[51] algoritması ile sürekli güncellenir. Bu sayede model farklılıkları öğrenmeye açık hale gelir. Her bir q, k ve v vektörleri için ayrı birer ağırlık matrisi vardır.

- **Sorgu**[52] : Bloklar arasında hangileriyle ilişkili olduğunu sorgular.

$$Q = x \times Wq \tag{4.3}$$

- **Anahtar**[53] : Bloklara kendisi hakkında bilgi verir.

$$K = x \times Wk \tag{4.4}$$

[50] Self-Attention

[51] Backpropagation

[52] Query

[53] Key

- **Değer**[54] : Bloklara hangi bilgiyi verebileceğine karar veren birimdir.

$$V = x \times Wv \tag{4.5}$$

Bu kavramlar şu şekilde örneklendirilebilir: sınıfa yeni gelen bir öğrenci öncelikle etrafındaki diğer öğrencilere bakar (sorgu), burada amaç kendisine uygun bir arkadaşın olup olmadığıdır. Öğrencinin görünümü o öğrencinin etiketidir (anahtar), kendisi hakkına diğerlerine bilgi verir. Her birine tek tek baktıktan sonra kendisine en iyi katkıda bulunacak (değer) kişiyle arkadaşlığa başlar.

Dikkat puanlarını hesaplamak için her bir query ile her bir key için matris çarpımı yapılır. Bu uyumluluk hakkında bilgi verir.

$$Skor = Q \times KT \tag{4.6}$$

T : Transpoz

Puanlar, key vektörlerinin boyutunun (d_k) karekökü ile ölçeklenir. Bu sayede gradyanların patlamaması sağlanır.

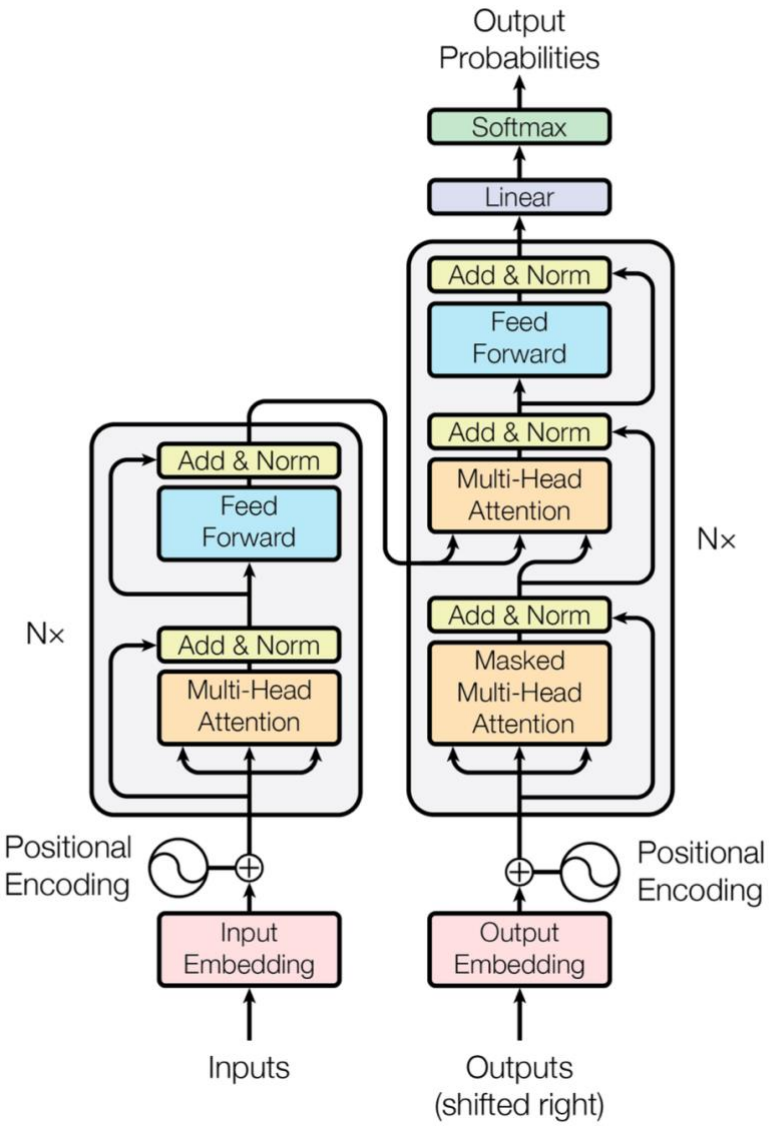
$$Skor = Skor \,/\, \sqrt{(d\_k)} \tag{4.7}$$

Elde edilen skora Softmax fonksiyonu uygulanarak 0-1 arasında değerler elde edilir ve V ile çarpılarak Ağırlıklı Toplam bulunur.

$$Ağırlıklı\ Toplam = softmax\ (Skor) \times V \tag{4.8}$$

[54] Value

Self Attention sürecini anlatan görsel Şekil 4.4'de sunulmuştur.



Şekil 4.3: Transformer model mimarisi

Kaynak: Vaswani vd., 2017: 3

⇒ ***Encoder kısmı,*** Çok Başlı Dikkat[55] ve İleri Besleme[56] katmanlarından oluşur. Her katmandan sonra artık bağlantı eklenir ve normalizasyon işlemi uygulanır. Bu döngü 6 defa yapılır. Artık bağlantı ile en temelde eski bilgilerin kaybolmaması amaçlanırken, normalizasyon işlemi ile de herhangi bir bilgi kaybı oluşturmadan eğitim süreci stabilize edilir.

Multi-Head Attention katmanı, Transformer mimarisindeki en önemli katmandır. Bu katmanda her baş, görüntünün farklı özelliklerine odaklanmaktadır. Şekil 4.5'te 4 başlı bir dikkat mekanizmasına örnek verilmiştir. Bu örnekte her başın farklı özelliklere yoğunlaştığı görülmektedir. Baş 1 kenarları algılarken, Baş 2 nesneler arasındaki ilişkiye odaklanmış, Baş 3 dokuları tanımaya çalışırken, Baş 4 ise uzak bağlantılar arasındaki ilişkiyi öğrenmeye çalışmıştır. Çok başlı dikkat mekanizmasının kaç başlı olması gerektiğine görüntünün içeriği, bu görüntülerle ne yapılmak istendiği ve aynı zamanda donanım ihtiyaçlarına göre karar verilmektedir. Orijinal makalede (Dosovitskiy vd., 2020) ViT modelinin geniş varyantı için 16 başlı bir mekanizma kullanılmaktadır. Bu çalışmada deneysel yolla 4 başlı, 8 başlı ve 16 başlı dikkat mekanizması denenmiş, en optimum sonuç 8 başlı dikkat mekanizmasından elde edilmiştir.

Feed Forward Network (İleri Beslemeli Ağ) ise konum bilgisinden bağımsız çalışarak, her konumdaki bilginin temsil yeteneğini artırmakla sorumludur. Konumdaki bilgileri daha derine indirmeye çalışan bir mekanizmaya sahiptir. Bu durumu bir sınıftaki öğrencilere benzetilebilir: Her öğrencinin bir öğrenci numarası vardır (position embedding), öğretmen bu sınıftaki öğrencileri gruplara bölmüştür ve her gruba aynı matematik kitabından farklı konu başlıkları vermiştir (multi-head attention) ve her öğrenci kendi bilgi ve deneyimlerine bağlı olarak -bütün sınıfta aynı olan matematik- kitabından kendisine verilen kısımda derinlemesine öğrenmeye çalışmaktadır (feed forward). Burada öğrencinin kendi ilgi, bilgi ve deneyimleri konuda ne kadar derinlemesine gideceğine karar vermektedir. Yani bildiği konularda daha derine inerken az bildiği konuları yüzeysel geçmektedir.

[55] Multi Head Attention

[56] Feed Forward

İleri beslemeli ağ, Transformer mimarisinde aşağıdaki formülle ifade edilmektedir:

$$FFN\ (x) = max\ (0, xW1 + b1)\, W2 + b2 \tag{4.9}$$

x : Transformer'da öz-dikkat katmanından gelen çıktıdır.

W_1 : İlk lineer dönüşüm için ağırlık matrisidir.

b_1 : İlk katmanın bias değeridir.

max (0,): ReLU[57] aktivasyon fonksiyonudur. Negatif değerleri sıfırlar, pozitif değerleri olduğu gibi bırakır. Bu sayede negatif değerleri elemiş olur ve önemli özelliklerin ön plana çıkmasına katkıda bulunur.

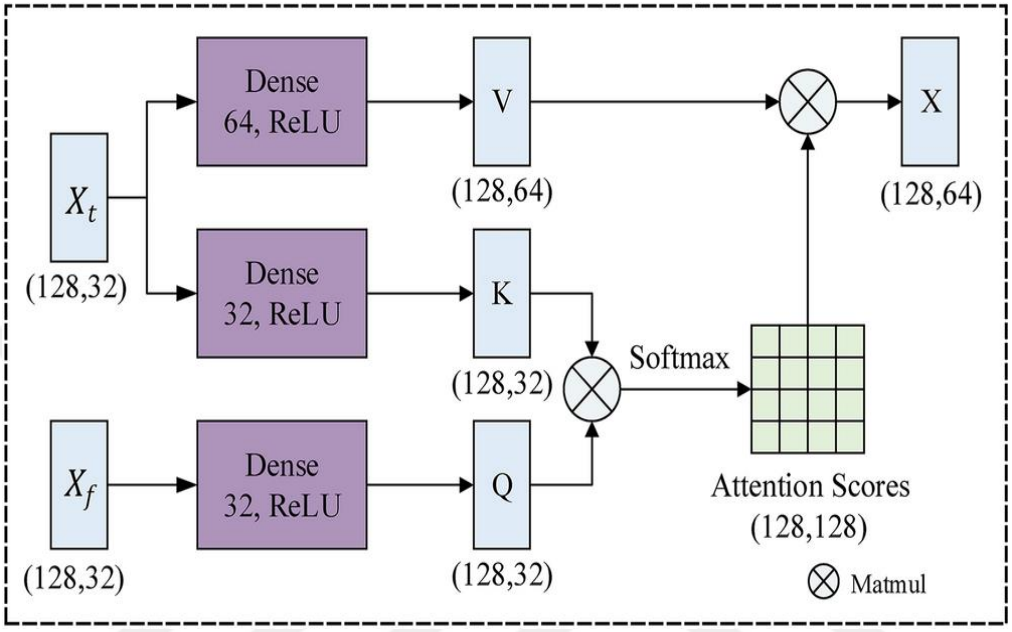
W_2 : İkinci lineer dönüşüm için ağırlık matrisi.

b_2 : İkinci katmanın bias değeridir.

⇒ ***Decoder kısmı,*** Encoder'den farklı olarak ek bir katman daha içermektedir. Bu kısım Maskelenmiş Çok Başlı Dikkat[58], Çok Başlı Dikkat ve İleri Besleme katmanlarından oluşmaktadır. Yine Encoder'daki gibi her katmandan sonra Artık Bağlantı ve normalizasyon işlemleri gerçekleştirilir. Yine orijinal makalede belirtildiği gibi bu işlemler 6 defa gerçekleştirilir, fakat ihtiyaca göre düzenlenebilir. Kod çözücünün son katmanından sonra bir lineer dönüşüm ve ardından softmax fonksiyonu uygulanır. Bu, çıktı dizisindeki her konum için bir olasılık dağılımı oluşturur. Decoder kısımındaki en önemli katman Masked Multi Head Attention katmanıdır. Bu katman ile sonraki bilgilerin görülmesi engellenir. Örneğin; yazı yazarken hep önceki kelime veya kelimeleri düşünerek sonraki kelimeler seçilir, "Ali bugün derse girmek istemedi" cümlesinde "derse" kelimesi sadece "Ali bugün" kelimelerini görebilir. Maskelenmiş çok başlı dikkat mekanizması ile sonraki kelimelerin görülmesi engellenmiş ve böylelikle daha doğal, gerçekçi ve tutarlı bir model ortaya çıkarılmış olur.

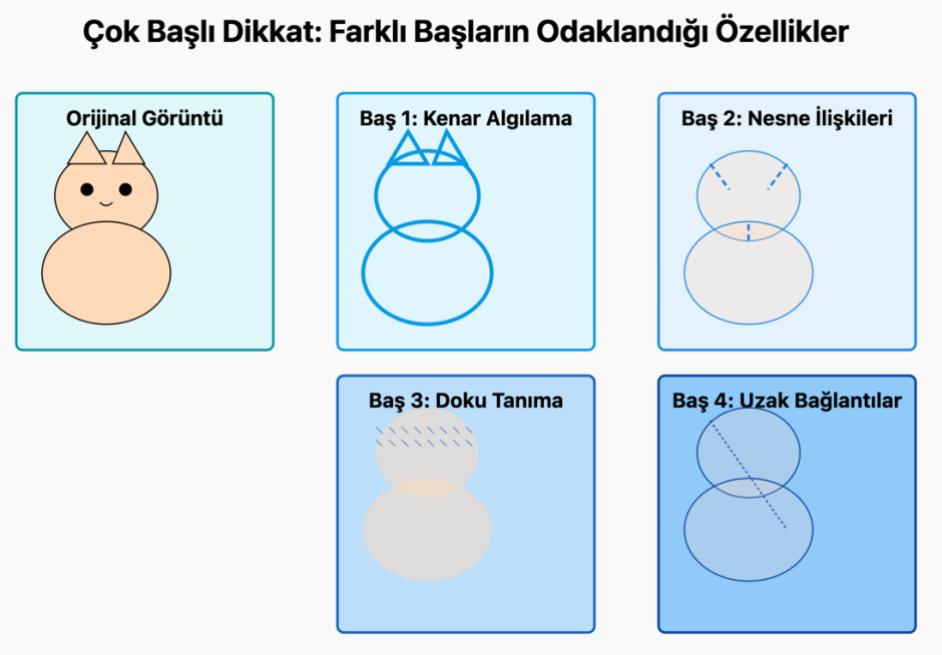
[57] Rectified Linear Unit

[58] Masked Multi Head Attention



Şekil 4.4: Öz dikkat skoru hesaplaması

Kaynak: Xi, Guo, Yang, Yuan, & Ma, 2024: 6



Şekil 4.5: Multi-Head Attention (Çok Başlı Dikkat) mekanizması

- **LSTM Bileşeni**

Makine öğreniminde kaybolan gradyan problemine çözüm olarak geliştirilen LSTM, zaman sıralı verileri işleme üzerine dayanan bir tekrarlayan sinir ağı modelidir.

Tıpkı Self-Attention'da olduğu gibi burada da bias ve W ağırlık matrisleri sabit değildir. Başlangıçta küçük değerlerle başlayıp sonrasında eğitim sürecinde sürekli güncellenen ve eğitim sonunda sabit kalan değerlerdir.

⇒ **Unutma Kapısı[59]:** İlk olarak, LSTM hücresi, hücre durumundan hangi bilgilerin çıkarılacağına karar verir. Bu, "unutma kapısı" tarafından kontrol edilir. Bu kapı, h_{t-1} (önceki çıktı) ve x_t (mevcut giriş) değerlerini alır ve sigmoid fonksiyonundan geçirerek her bir hücre durumu bileşeni için bir "unutma faktörü" üretir. Sigmoid fonksiyonu çıktıyı 0-1 aralığına sıkıştırır. 0 değeri "bu bilgiyi tamamen unut", 1 değeri "bu bilgiyi tamamen koru" anlamına gelir.

[59] Forget Gate

$$Ft = \sigma\ (Wf \cdot [ht-1, xt] + bf) \quad (4.10)$$

- f_t : Unutma kapısının çıktısı (0-1 arası)
- σ : Sigmoid aktivasyon fonksiyonu
- W_f : Unutma kapısı ağırlık matrisi
- h_{t-1} : Önceki gizli durum
- x_t : Mevcut giriş
- b_f : Bias terimi

⇒ **Giriş Kapısı[60]:** Ardından LSTM, hafızasına hangi yeni bilgilerin ekleneceğine karar verir. Bu iki aşamada gerçekleşir.
Giriş kapısı (input gate), hangi değerlerin güncelleneceğine karar verir (sigmoid fonksiyonu kullanılarak).

$$it = \sigma\ (Wi \cdot [ht-1, xt] + bi) \quad (4.11)$$

Unutma kapısı ile aynı işlemler gerçekleştirilir, fakat ek olarak tanh fonksiyonu vasıtasıyla bir aday hücre durumu (candidate cell state) oluşturulur:

$$g_t = tanh\ (W_g \cdot [h_{t-1}, x_t] + b_g) \quad (4.12)$$

- i_t : Giriş kapısının çıktısı (0-1 arası)
- g_t : Aday hücre durumu (-1 ile 1 arası)
- tanh : Hiperbolik tanjant aktivasyon fonksiyonu
- W_i , W_g : İlgili ağırlık matrisleri
- b_i , b_g : Bias terimleri

[60] Input Gate

⇒ **Hücre Durum Güncellemesi[61] :** Unutma ve giriş kapılarından gelen bilgiler kullanılarak hücre durumu güncellenir. Hücrenin ne kadar bilgiyi unutacağını ve ne kadar yeni bilgiyi ekleyeceğini belirler.:

$$Ct = ft \odot Ct - 1 + it \odot gt \quad (4.13)$$

- C_t : Mevcut hücre durumu
- C_{t-1} : Önceki hücre durumu
- $\odot$: Hadamard çarpımı (eleman-eleman çarpım)

⇒ **Çıkış Kapısı[62]:** Çıkış kapısı, hangi bilgilerin dışarı aktarılacağını belirler:

$$ot = \sigma\ (Wo \cdot [ht - 1, xt] + bo) \quad (4.14)$$

Ve son olarak, gizli durum şu şekilde hesaplanır:

$$ht = ot \odot tanh\ (Ct) \quad (4.15)$$

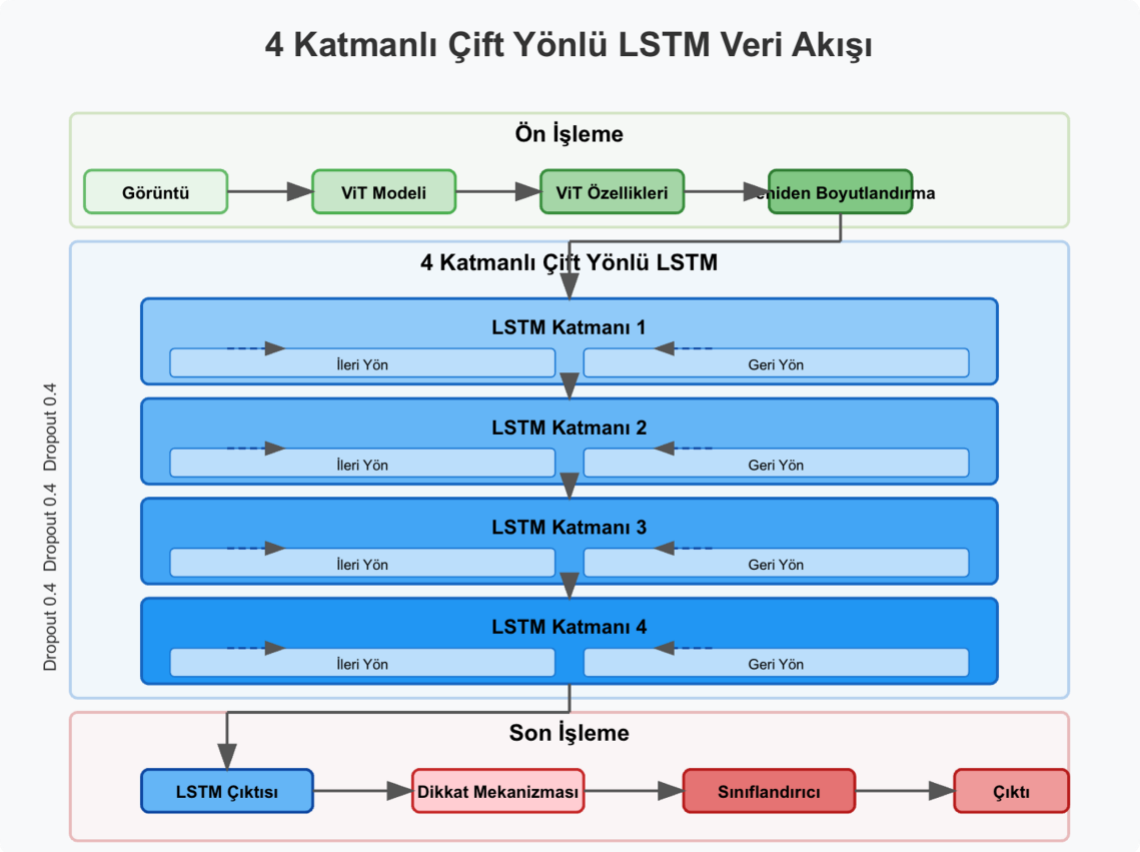
- o_t: Çıkış kapısının çıktısı (0-1 arası)
- h_t: Mevcut gizli durum

Bu çalışmada her katmanın bir önceki katmanı işleyebilmesi ve daha karmaşık bilgileri öğrenebilmesini desteklemek amacıyla 4 katmanlı bir yapı kullanılmıştır. Ayrıca her zaman adımının bir önceki ve bir sonraki bilgileri kullanabilmesi için de çift yönlü LSTM mimarisi kullanılmıştır. Çok katmanlı ve çift yönlü LSTM modelinin ilk katmanları basit ve düşük seviye bilgiler edinir. Orta katmanlar bu basit özellikleri birleştirir ve karmaşık

[61] Cell State Update

[62] Output Gate

bir yapı çıkarır. Son katmanlar ise yüksek seviyeli bilgiler edinir. Modelimizde kullandığımız LSTM modeli Şekil 4.6'da görülmektedir.



Şekil 4.6: Modelimizin LSTM katmanları.

- **ViT-LSTM Modeli**

Çukur tespitinde önerdiğimiz algoritmada iki bileşen şu şekilde çalışmaktadır.

1. Özellik Çıkarma: Vision Transformer bileşeni görüntüden uzamsal bilgiler çıkararak içeriğe dair bir özellik vektörü oluşturur.

2. Sıralı İşleme: ViT'den çıkarılan özellikler LSTM modeline girdi olarak verilir. Özellikle zaman sıralı veri setlerinde yüksek performans gösterir.

3. Bağlam Modelleme: LSTM, ViT'den gelen özellikleri sıralı bir şekilde işleyerek bağlamsal veya zamansal ilişkileri modeller. Örneğin sıralı bir yol veri setinde ağaç gölgesinin oluşturduğu çukur görünümü, sonraki karede yer almayacağı için model buradaki durumu çukur olarak algılamayacaktır. Dolayısıyla yanlış-pozitif sayısını düşürmeye yardımcı olmaktadır.

4. Son Katman: Model çıktısındaki sınıflandırma katmanıdır.

4.2. ViT-LSTM Model Birleşimi

Veri setimizin kısmen zaman sıralı özellikler göstermesi nedeniyle, Vision Transformer (ViT) ve Uzun-Kısa Süreli Bellek (LSTM) ağlarının güçlü yönlerini birleştirerek yüksek performanslı bir model geliştirdik. Modelimizde, önceden eğitilmiş "vit_large_patch16_384" ağı omurga olarak kullanılmıştır. ViT tarafından çıkarılan özellikler, çift yönlü, dört katmanlı ve 1024 gizli birime sahip bir LSTM ağına giriş olarak verilmiştir. Aşırı öğrenmeyi önlemek amacıyla ağın eğitiminde %40 oranında dropout uygulanmıştır. LSTM'nin çıktısı, 8 başlı bir dikkat (attention) mekanizmasına aktarılmış ve böylece modelin görüntüdeki farklı bölgeleri öğrenmesi ve bu bölgeler arasındaki ilişkileri yakalaması sağlanmıştır. Son olarak, sınıflandırıcı katmanına 2048 boyutlu bir vektör giriş olarak verilmiş; bu vektör önce 512 boyutuna indirgenmiş, ardından ReLU aktivasyon fonksiyonu ve dropout uygulanmıştır. Daha sonra 512 boyutundaki vektör 256'ya, son olarak da 2'ye düşürülerek sınıflandırma işlemi gerçekleştirilmiştir. Her Lineer modülünden sonra ReLU aktivasyon fonksiyonu ve dropout işlemleri gerçekleştirilmiştir. Modelimizin genel görünümü Şekil 4.7'de görüldüğü gibidir.

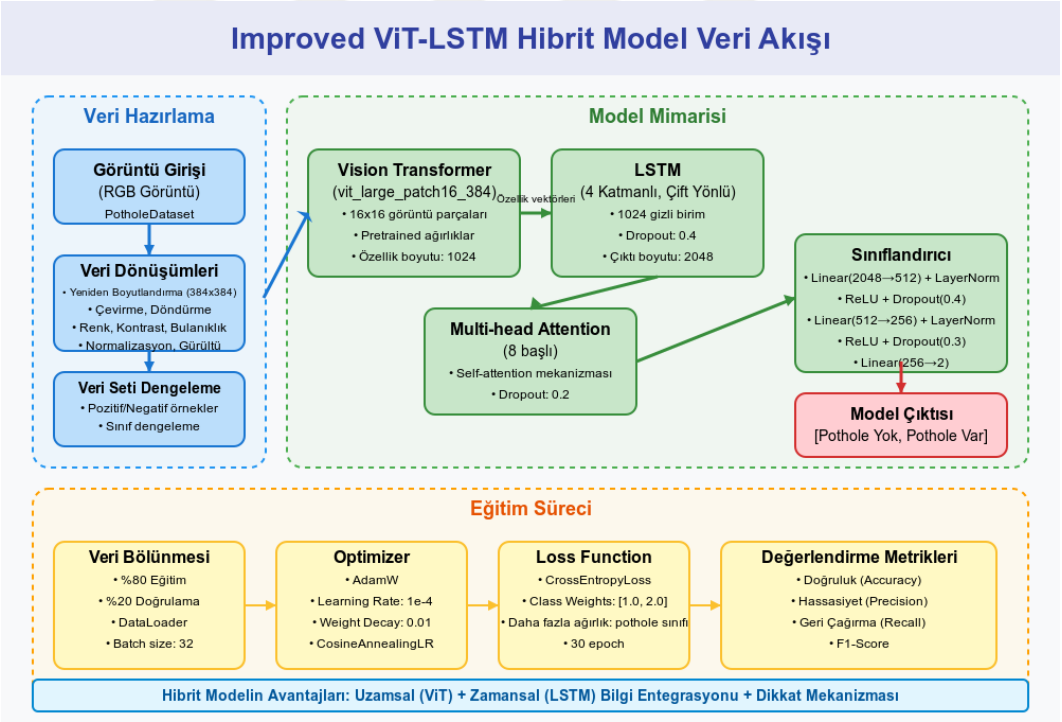
Tüm bunlarla birlikte ViT-LSTM modeline ait zaman karmaşıklığı[63] analizi yapıldığında $O(n \times d^2)$ olduğu görülmüştür. Burada *n*, görüntüdeki yama[64] sayısını (384x384 boyutundaki bir görüntüde 16x16 yama boyutu için 576 yama olur) ifade ederken, *d* ise modelin gömme[65] boyutunu (ViT-Large modelimiz için 1024'tür) temsil etmektedir. Modelin bu karmaşıklığa sahip olmasının temel nedeni self-attention ve feed forward katmanlarıdır. Self-Attention katmanı, elemanların dizideki önemini belirlemek için herbir elemanın diğer elemanlarla etkileşime girdiği bir yapıya sahiptir. Bundan dolayı bu mekanizma, zaman karmaşıklığına büyük oranda etki etmektedir. Feed forward katmanı ise verinin tek yönde (baştan sona) aktığı bir yapıya sahiptir. Bu yapı iki lineer dönüşüm

[63] Time Complexity

[64] Patch

[65] Embedding

ve bir de aktivasyon fonksiyonuna sahiptir. Verinin temsil kapasitesini artırmak amacıyla ilk olarak giriş boyutunun k katına (genellikle 4 katına genişletildiği için bizim modelimizde de 4 katına genişletilmektedir) genişletilir. Ardından aktivasyon fonksiyonu ile doğrusal olmayan ilişkileri öğrenir ve ikinci dönüşümde ise veri tekrar d boyutuna daraltılır. Bu nedenle bu katmandaki giriş boyutunun büyüklüğü feed forward katmanının zaman karmaşıklığına olan etkisini belirlemektedir. LSTM bileşeninde ise sınırlı sekans sayısı nedeniyle zaman karmaşıklığına etkisi ViT bileşenine görece düşüktür. Her ne kadar da çift yönlü LSTM yapısı kullanılsa da sabit boyutlu ViT çıktısı üzerinde işlem yaptığı için toplam hesaplama yükünün çok cüzi bir kısmını oluşturmaktadır. ViT bileşenindeki self-atteniton ve feed forward mekanizmaları nedeniyle bu bileşen daha baskın çıkmaktadır.



Şekil 4.7: Önerdiğimiz ViT-LSTM modelin genel görünümü.

4.3. İnce Ayar[66]

Bu çalışmada modelin eğitim sürecinde optimizasyon algoritması olarak AdamW tercih edilmiştir. AdamW, klasik Adam optimizasyon algoritmasının geliştirilmiş bir versiyonu olup, özellikle Weight Decay mekanizmasını daha etkili bir şekilde ele alarak aşırı öğrenmeyi[67] azaltmakta ve modelin genelleme kapasitesini artırmaktadır. Loshchilov ve Hutter tarafından yayınlanan orijinal makalede, AdamW'nin geleneksel Adam algoritmasına kıyasla daha iyi bir genelleme sağladığı ve özellikle Cosine Annealing gibi öğrenme oranı planlayıcıları ile birlikte kullanıldığında daha üstün performans gösterdiği belirtilmiştir (Loshchilov & Hutter, 2017).

Bu bağlamda, önerilen modelde öğrenme oranının dinamik olarak ayarlanmasını sağlamak amacıyla Cosine Annealing yöntemi entegre edilmiştir. İlk öğrenme oranı belirlendikten sonra, her 5 epoch'ta bir Warm Restarts uygulanarak öğrenme oranı sıfırlanmış ve böylece modelin farklı yerel minimumlara ulaşarak öğrenme kapasitesini artırması sağlanmıştır.

Bunun yanı sıra, Kumar ve arkadaşları tarafından yapılan çalışmada, AdamW optimizasyon algoritmasının veri dağılımındaki değişimlere karşı daha dirençli olduğu belirtilmektedir. Bu çalışmada veri dağılımındaki değişiklikler olarak ifade edilecebilecek olan özellikle farklı yol yüzeyleri, değişken ışık koşulları ve çevresel faktörlerin çukur tespiti üzerindeki etkileri göz önüne alındığında, AdamW'nin Stochastic Gradient Descent (SGD) algoritmasına kıyasla daha kararlı sonuçlar verdiği ifade edilmektedir (Kumar, Shen, Bubeck, & Gunasekar, 2022). Ayrıca, geniş ölçekli modeller için AdamW'nin kullanımı önerilmektedir.

Bu çalışmada, loss fonksiyonu olarak Focal Loss tercih edilmiştir. Focal Loss, Cross Entropy Loss'un bir türevi olup, özellikle sınıf dağılımının dengesiz olduğu veri setlerinde kullanımı yaygındır. Bu çalışma özelinde modelin küçük ve belirgin olmayan çukurları öğrenmesini sağlamak amacıyla zor öğrenilen örneklere daha fazla ağırlık verilmiştir.

[66] Fine Tuning

[67] Overfitting

Böylece, modelin nadiren karşılaşılan veya daha az belirgin çukurları da doğru bir şekilde tespit edebilmesi hedeflenmiştir.

Ek olarak, modelin genelleme yeteneğini artırmak ve farklı çevresel koşullara karşı dayanıklılığını güçlendirmek amacıyla veri artırma[68] teknikleri uygulanmıştır. Bu kapsamda, parlaklık, kontrast, doygunluk değişimleri, dönme transformasyonları ve gürültü ekleme gibi çeşitli tekniklerden faydalanılmıştır.

Sonuç olarak, AdamW optimizasyon algoritmasının, Cosine Annealing öğrenme oranı planlayıcısı ve Warm Restarts stratejisiyle birlikte kullanımı, modelin genelleme yeteneğini artırmış ve öğrenme sürecinin daha etkin bir şekilde gerçekleştirilmesini sağlamıştır. Buna ek olarak, Focal Loss kullanımı ve veri artırma teknikleri, modelin özellikle küçük ve belirgin olmayan çukurlar üzerinde daha başarılı sonuçlar elde etmesine katkıda bulunmuştur.

[68] Data Augmentation

BEŞİNCİ BÖLÜM

BULGULAR VE SONUÇLAR

Bu bölümde çalışmada kullanılan 3 farklı algoritmanın modelleme bilgileri detaylı bir şekilde anlatılacaktır.

5.1. YOLOv8[69] Uygulaması

Nesne tespiti (Object Detection) için optimize edilmiş YOLOv8 modeli kullanılmıştır. Eğitimde Adam optimizer kullanılmıştır. YOLO formatına uygun anotasyonlar otomatik olarak oluşturulmuştur. MPS (Metal Performance Shaders) desteklenmiştir, yani Mac cihazda eğitildiği için cihaza uygun şekilde optimize edilmiştir. Model, 100 epoch boyunca eğitilmiş ve erken durdurma (patience=20) uygulanmıştır. Eğitim sırasında görüntü boyutu 512x512 olarak belirlenmiştir. Batch size 16, Pretrained YOLOv8 modeli kullanılarak transfer öğrenme uygulanmıştır.

5.2. CNN Model Uygulaması

Çok katmanlı, gelişmiş bir CNN modeli kullanılmıştır. 3 konvolüsyon bloğundan oluşur: 64, 128 ve 256 filtre boyutlarına sahip konvolüsyon katmanları içerir. Her konvolüsyon bloğunda Batch Normalization ve Dropout eklenmiştir. MaxPooling ile boyut küçültme ve özellik çıkarımı optimize edilmiştir. Cross-Validation (K=5) ile eğitilmiştir: Farklı veri bölümlerinde modelin test edilmesi sağlanarak genelleme başarısı artırılmıştır. Veri artırma teknikleri daha kapsamlı hale getirilmiştir: Dönme, parlaklık değişimi, yatay kaydırma, zoom ve bulanıklık gibi çeşitli dönüşümler uygulanmıştır. AdamW optimizasyon algoritması kullanılmıştır. Öğrenme oranı[70] = 1e-4, weight decay = 1e-5 Eğitim sırasında erken durdurma (Early Stopping) ve öğrenme oranı azaltma (ReduceLROnPlateau) kullanılmıştır. Bu kullanım oranları farklı değerler kullanılarak denenmiş, optimum değerler olduğu görülmüştür.

[69] You Only Look Once version 8

[70] Learning Rate

5.3. Önerdiğimiz ViT-LSTM Modeli Uygulaması

Vision Transformer (ViT) ve LSTM kombinasyonu kullanılmıştır. Daha büyük ViT modeli olan vit_large_patch16_384 kullanılmıştır. Çift yönlü[71] LSTM katmanı eklenmiştir. Self-Attention mekanizması ile bilgiyi daha iyi yakalama yeteneği geliştirilmiştir. Gürültü ekleme[72] kullanılarak modelin dayanıklılığı artırılmıştır. Veri artırma teknikleri olarak renk dönüşümleri, yatay-dikey çevirme, bulanıklık, affine dönüşümler uygulanmıştır. Çukur sınıfına daha fazla ağırlık verilerek modelin çukurları daha iyi öğrenmesi amacıyla özel class weights eklenmiştir. AdamW optimizasyon algoritması kullanılmıştır: LR = 1e-4, weight decay = 0.01, Cosine Annealing Scheduler ile öğrenme oranı ayarlanmıştır. Focal Loss kayıp fonksiyonu kullanılmıştır, böylece küçük çukurlar daha iyi tespit edilebilmiştir.

Görüntü tanıma görevleri için geliştirilen Vision Transformer (ViT) ve Long Short-Term Memory (LSTM) tabanlı hibrit modelimizin deneysel sonuçları detaylı olarak incelenmektedir. Toplam 1576 örnek üzerinde gerçekleştirilen kapsamlı testler, önerilen modelin sınıflandırma performansını çeşitli metrikler açısından değerlendirmektedir.

Geliştirilen ViT-LSTM modelinin sınıflandırma performansı, öncelikle karmaşıklık matrisi parametreleri ile incelenmiştir. Modelimize ait karmaşıklık matrisi Şekil 5.1.'de gösterilmiştir. CNN modeline ait karmaşıklık matrisi Şekil 5.2'de, YOLOv8 modeline ait karmaşıklık matrisi ise Şekil 5.3'te sunulmuştur. Karmaşıklık matrisi, modelin tahminlerinin gerçek sınıf etiketleriyle karşılaştırılmasını sağlayan temel bir değerlendirme aracıdır.

Karmaşıklık matrisinde görüldüğü üzere, model 714 pozitif örneği doğru bir şekilde pozitif olarak, 782 negatif örneği ise doğru bir şekilde negatif olarak sınıflandırmıştır. Buna karşın, yalnızca 6 örnek yanlış pozitif ve 74 örnek yanlış negatif olarak sınıflandırılmıştır. Bu sonuçlar, modelin sınıf tahminlerinde oldukça başarılı olduğunu göstermektedir.

[71] Bidirectional

[72] Gaussian Noise

ViT-LSTM modelinin performansını kapsamlı bir şekilde değerlendirmek için çeşitli metrikler hesaplanmıştır. Bu metrikler, modelin farklı açılardan değerlendirilmesine olanak tanımaktadır.

Doğruluk : Modelin genel doğruluk oranı %94.92 olarak hesaplanmıştır. Bu değer, tüm örnekler içerisinde doğru sınıflandırılan örneklerin oranını ifade etmektedir ve şu formülle hesaplanmaktadır:

$$Doğruluk = \frac{TP+TN}{TP+TN+FP+FN} \times 100\% = \frac{714+782}{1576} \times 100\% = 94.92\%. \quad (5.1)$$

Kesinlik : Modelin pozitif olarak tahmin ettiği örnekler içerisinde gerçekten pozitif olanların oranını ifade eden kesinlik değeri %99.17 olarak hesaplanmıştır:

$$Kesinlik = \frac{TP}{TP+FP} \times 100\% = \frac{714}{714+6} \times 100\% = 99.17\% \quad (5.2)$$

Bu yüksek kesinlik değeri, modelin pozitif sınıf tahminlerinde neredeyse mükemmel bir performans sergilediğini göstermektedir.

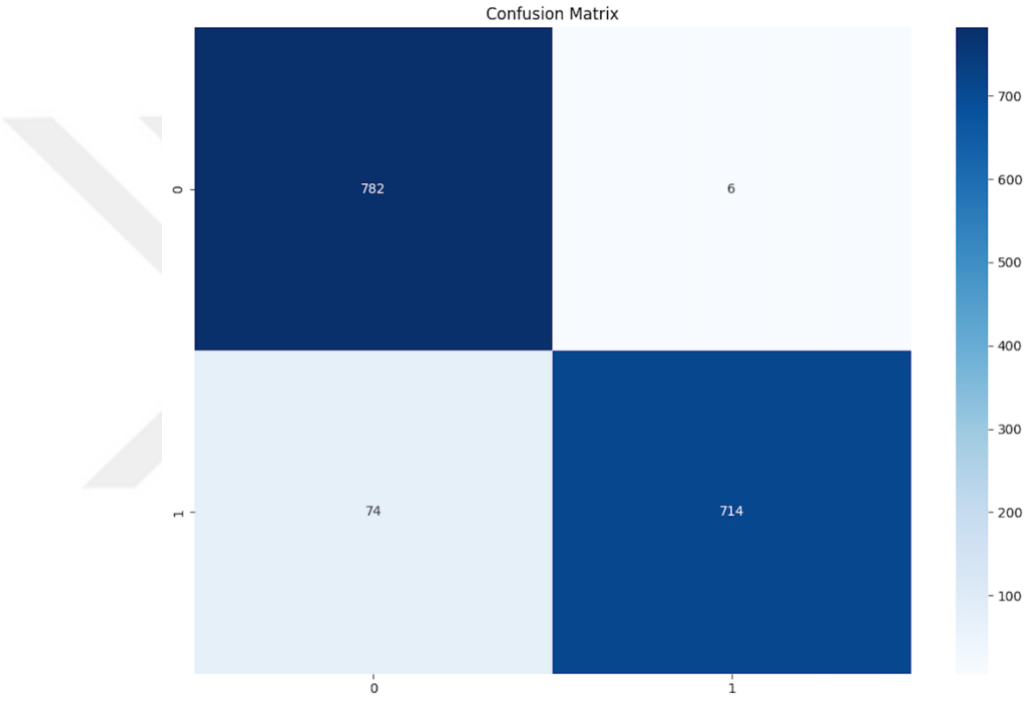
Duyarlılık : Gerçekte pozitif olan örnekler içerisinde model tarafından pozitif olarak tahmin edilenlerin oranını ifade eden duyarlılık değeri %90.61 olarak hesaplanmıştır:,

$$Recall = \frac{TP}{TP+FN} \times 100\% = \frac{714}{714+74} \times 100\% = 90.61\% \quad (5.3)$$

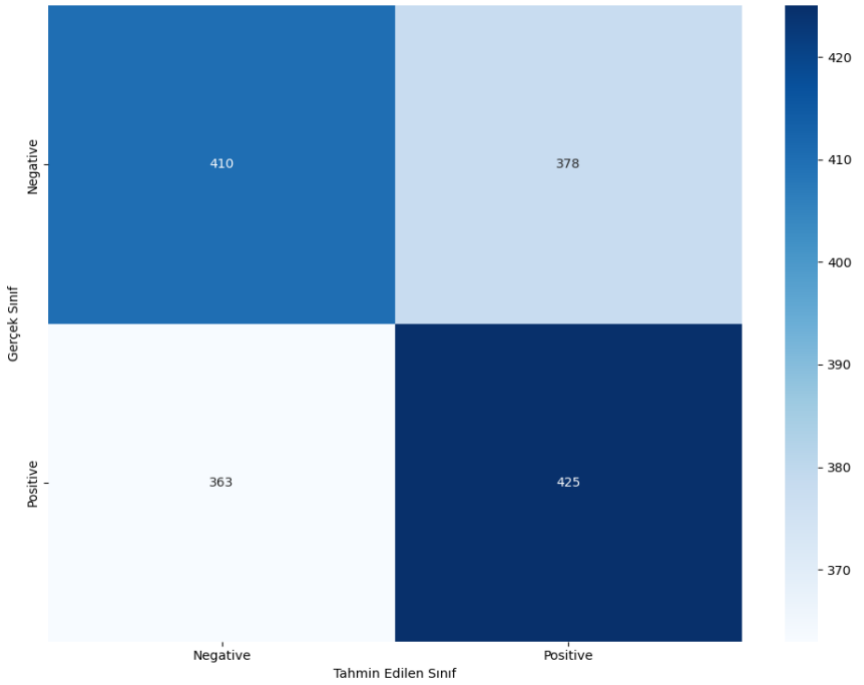
F1 Skoru: Kesinlik ve duyarlılık metriklerinin harmonik ortalaması olan F1 skoru, modelin dengeli bir performans gösterip göstermediğini değerlendirmek için kullanılmaktadır. Önerilen model için F1 skoru %94.69 olarak hesaplanmıştır:

$$F1\ Skoru = 2 \times \frac{Precision \times Recall}{Precision + Recall} \times 100\% = 2 \times \frac{99.17\% \times 90.61\%}{99.17\% + 90.61\%} \times 10 = 94.69\% \quad (5.4)$$

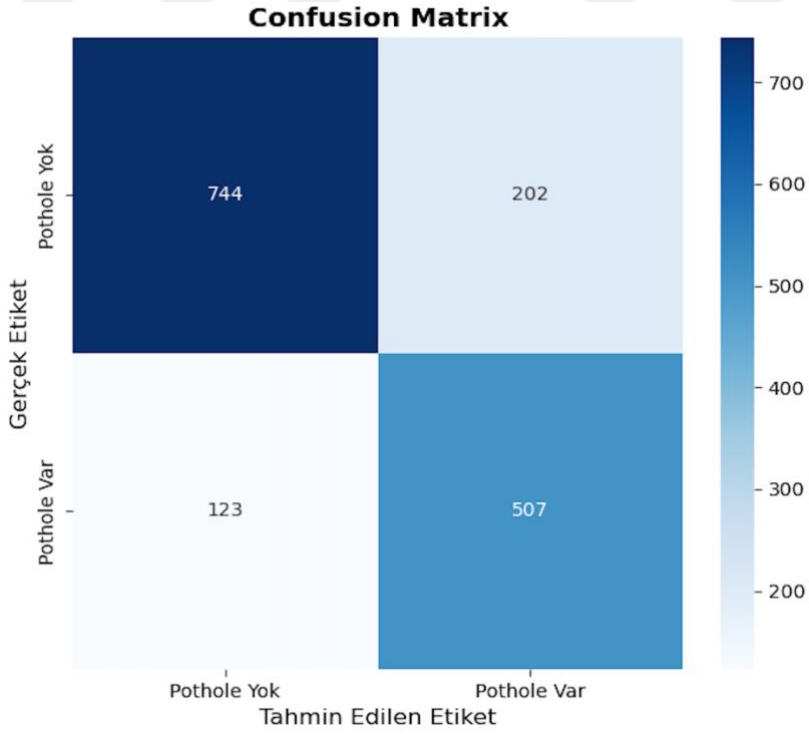
Bu yüksek F1 skoru, modelin hem kesinlik hem de duyarlılık açısından dengeli bir performans sergilediğini göstermektedir.



Şekil 5.1: Önerdiğimiz modelin karmaşıklık matrisi değerleri.



Şekil 5.2: CNN modeli karmaşıklık matrisi değerleri



Şekil 5.3: YOLOv8 modeli karmaşıklık matrisi değerleri.

Model metriklerinin karşılaştırmalı sonuçları Tablo 5.1'deki gibidir.

Tablo 5.1:Model metriklerinin karşılaştırmalo tablosu.

Metrikler	ViT-LSTM (%)	YOLO v8 (%)	CNN (%)
Doğruluk	**94.92**	76.30	52.97
Kesinlik	**99.17**	85.50	52.91
Duyarlılık	**90.61**	63.50	53.93
F1 Skoru	**94.69**	72.80	53.42

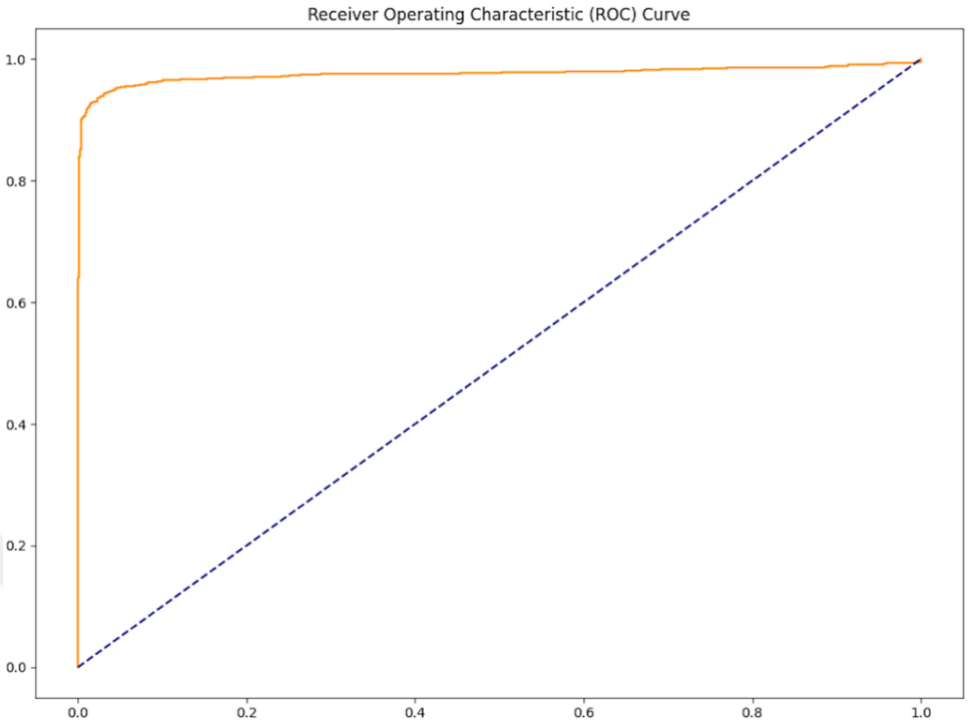
Modelin sınıflandırma performansını daha derinlemesine analiz etmek için ROC[73] eğrisi altında kalan alan[74] ve ortalama kesinlik değerleri hesaplanmıştır.

ROC AUC: ROC eğrisi, farklı eşik değerleri için doğru pozitif oranı (TPR) ve yanlış pozitif oranı (FPR) arasındaki ilişkiyi gösteren bir eğridir. Bu eğri altında kalan alan AUC ise modelin ayrım yapabilme kapasitesini ölçmektedir. Önerilen ViT-LSTM modeli için ROC-AUC değeri 0.9756 olarak hesaplanmıştır. Bu değer, modelin pozitif ve negatif örnekleri yüksek bir başarıyla ayırt edebildiğini göstermektedir. Şekil 5.4'te modelimize ait ROC-AUC eğrisi, Şekil 5.5'te ise CNN modelinin ROC-AUC eğrisi sunulmuştur.

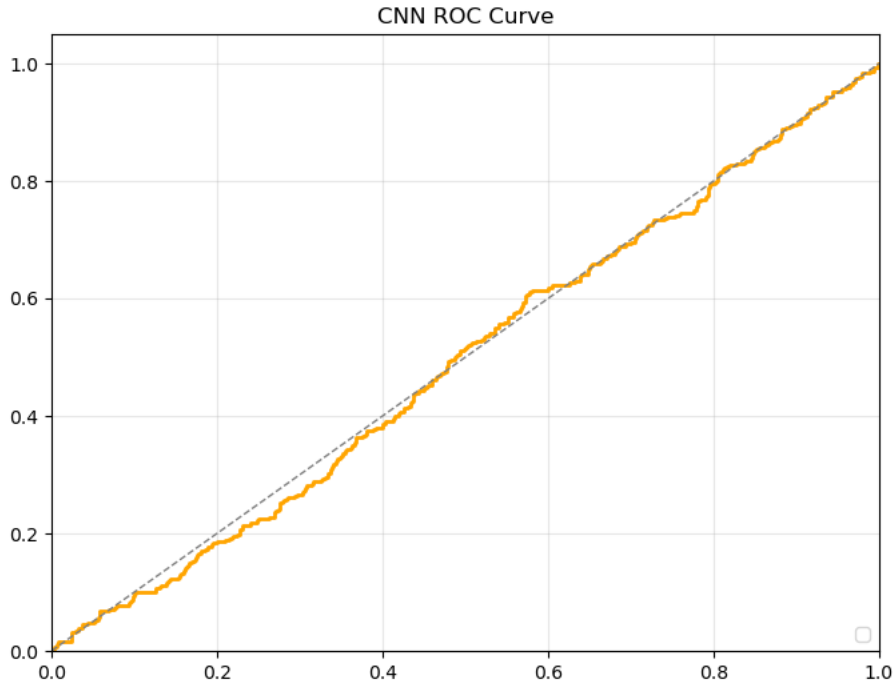
Ortalama Kesinlik: Kesinlik-duyarlılık eğrisi altında kalan alan olarak da bilinen ortalama kesinlik değeri, modelin farklı duyarlılık seviyelerindeki kesinlik performansını değerlendirmektedir. Önerilen model için ortalama kesinlik değeri 0.9848 olarak hesaplanmıştır. Bu yüksek değer, modelin farklı eşik değerleri için tutarlı bir şekilde yüksek kesinlik ve duyarlılık değerleri sergilediğini göstermektedir.

[73] Receiver Operating Characteristic

[74] Area Under the Curve



Şekil 5.4: Önerdiğimiz modelin ROC Eğrisi.



Şekil 5.5: CNN modeli ROC eğrisi.

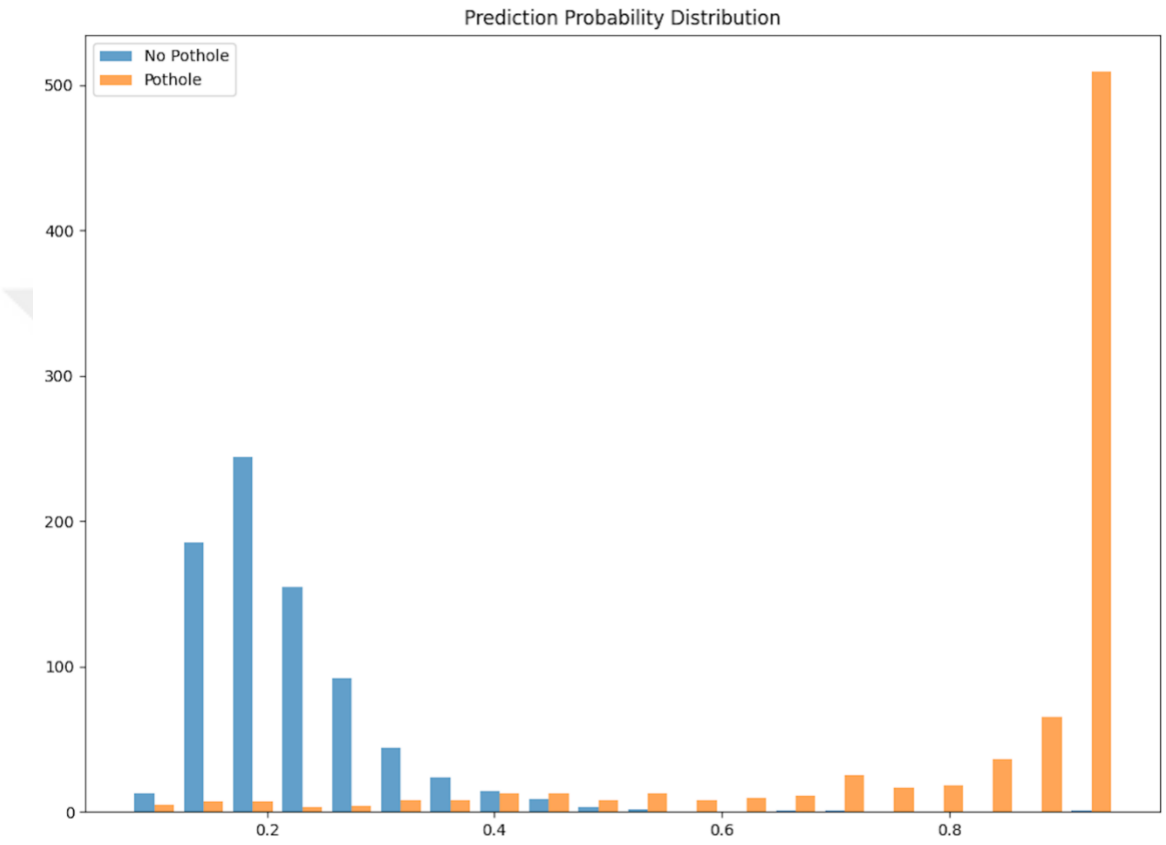
Şekil 5.6'da çukur sınıflandırma modelinin tahmin olasılık dağılımını[75] göstermektedir. Bu dağılım, modelin her bir sınıfa atadığı olasılık değerlerinin frekansını temsil eder.

Grafikte dikkat çeken noktalar şunlardır:

- "No Pothole" sınıfına ait örnekler (mavi sütunlar) ağırlıklı olarak düşük olasılık değerlerinde yoğunlaşmıştır (0.1-0.3 aralığı). Bu, modelin arka plan sınıfına ait örnekleri yüksek güvenle tahmin ettiğini gösterir.
- "Pothole" sınıfına ait örnekler (turuncu sütunlar) çoğunlukla yüksek olasılık değerlerinde (özellikle 0.9+ aralığında) toplanmıştır. Bu durum, modelin çukur olarak sınıflandırdığı örneklerde yüksek güven düzeyine sahip olduğunu gösterir.

[75] Prediction Probability Distribution

- İki dağılım arasında belirgin bir ayrım vardır, ancak 0.4-0.8 aralığında az miktarda örtüşme gözlemlenmektedir. Bu örtüşme bölgesi, modelin kararsız kaldığı sınır durumlarını temsil eder.



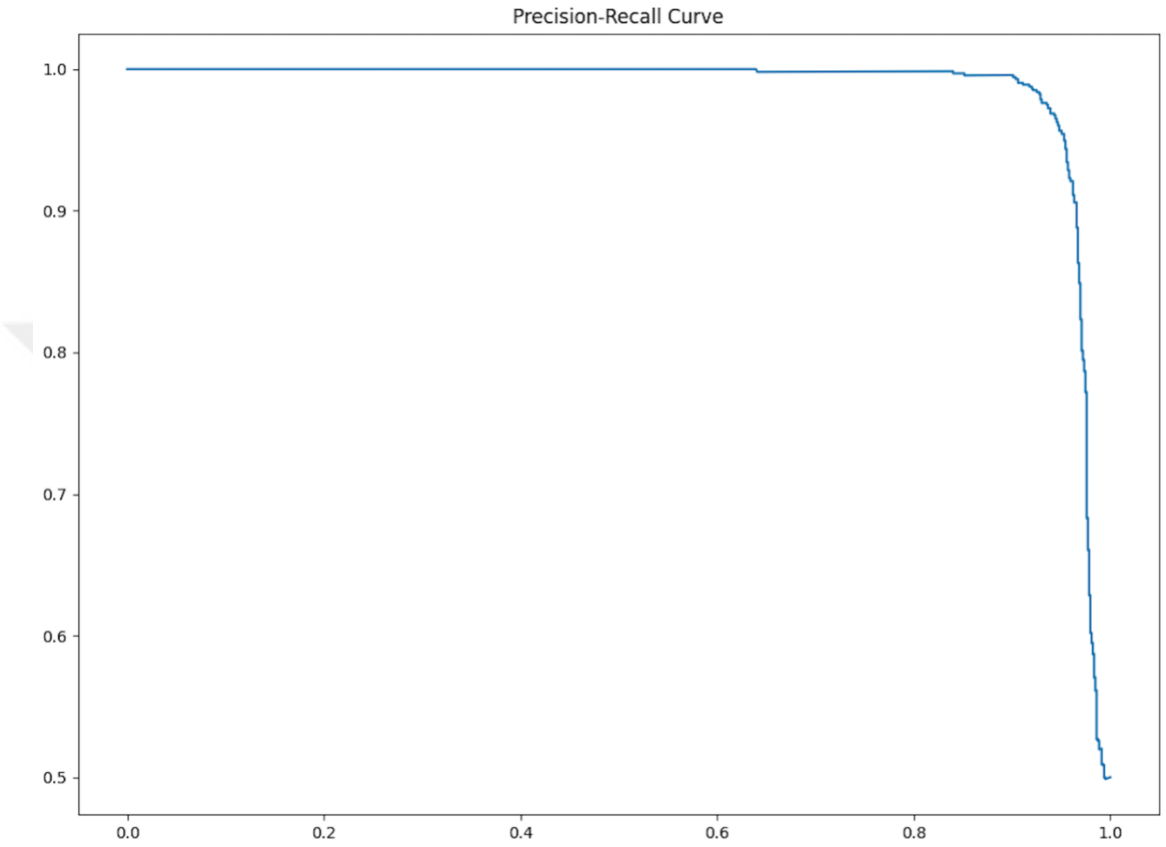
Şekil 5.6: Tahmin - Olasılık Dağılımı Analizi

Precision-Recall eğrisi, sınıflandırma modellerinin performansını değerlendirmek için kullanılan önemli bir grafik gösterimdir. Bu eğri, modelin doğruluk (precision) ve duyarlılık (recall) arasındaki ilişkiyi farklı eşik değerlerinde gösterir.

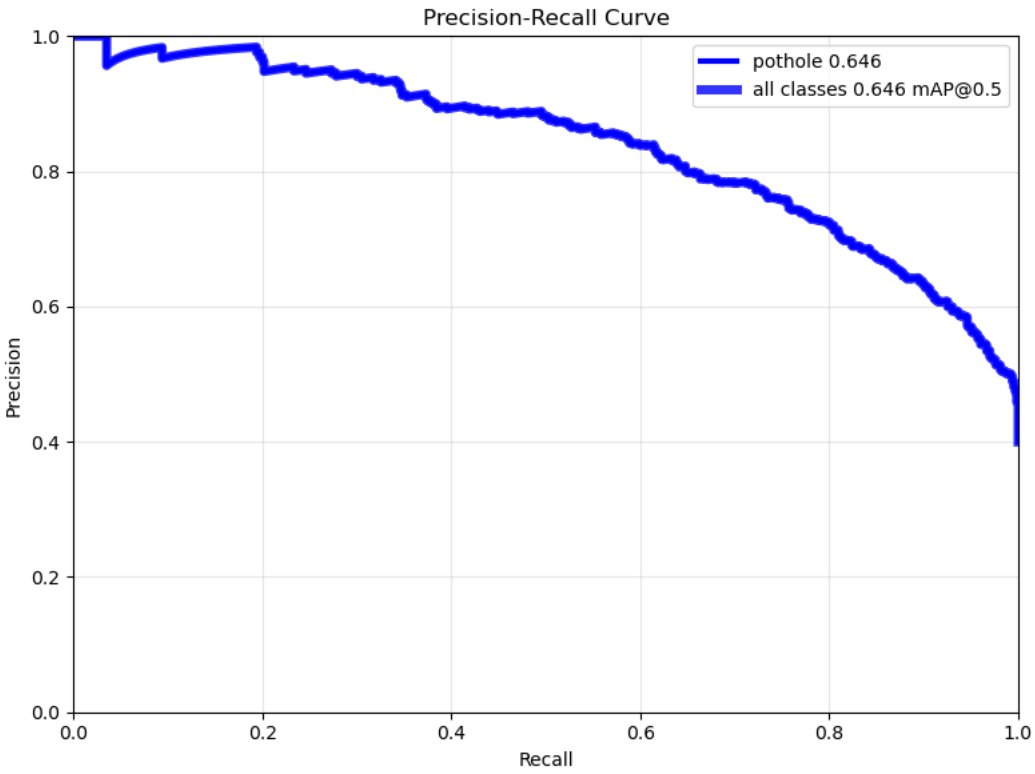
Şekil 5.7'de modelmize ait kesinlik ve duyarlılık metrikleri arasındaki ilişkiyi gösteren bir Precision-Recall eğrisidir. Bu eğriden çıkarılabilecek sonuçlar :

- Eğri, geniş bir duyarlılık (recall) aralığında (0-0.9) neredeyse mükemmel bir kesinlik (precision) değeri (1.0) sürdürmektedir. Bu, modelin çukur sınıfını tanımlamada oldukça kesin olduğunu gösterir.

- Eğrinin altındaki alan oldukça geniştir, bu da modelin genel performansının yüksek olduğunu gösterir.



Şekil 5.7: Modelimize ait Precision-Recall Eğrisi.

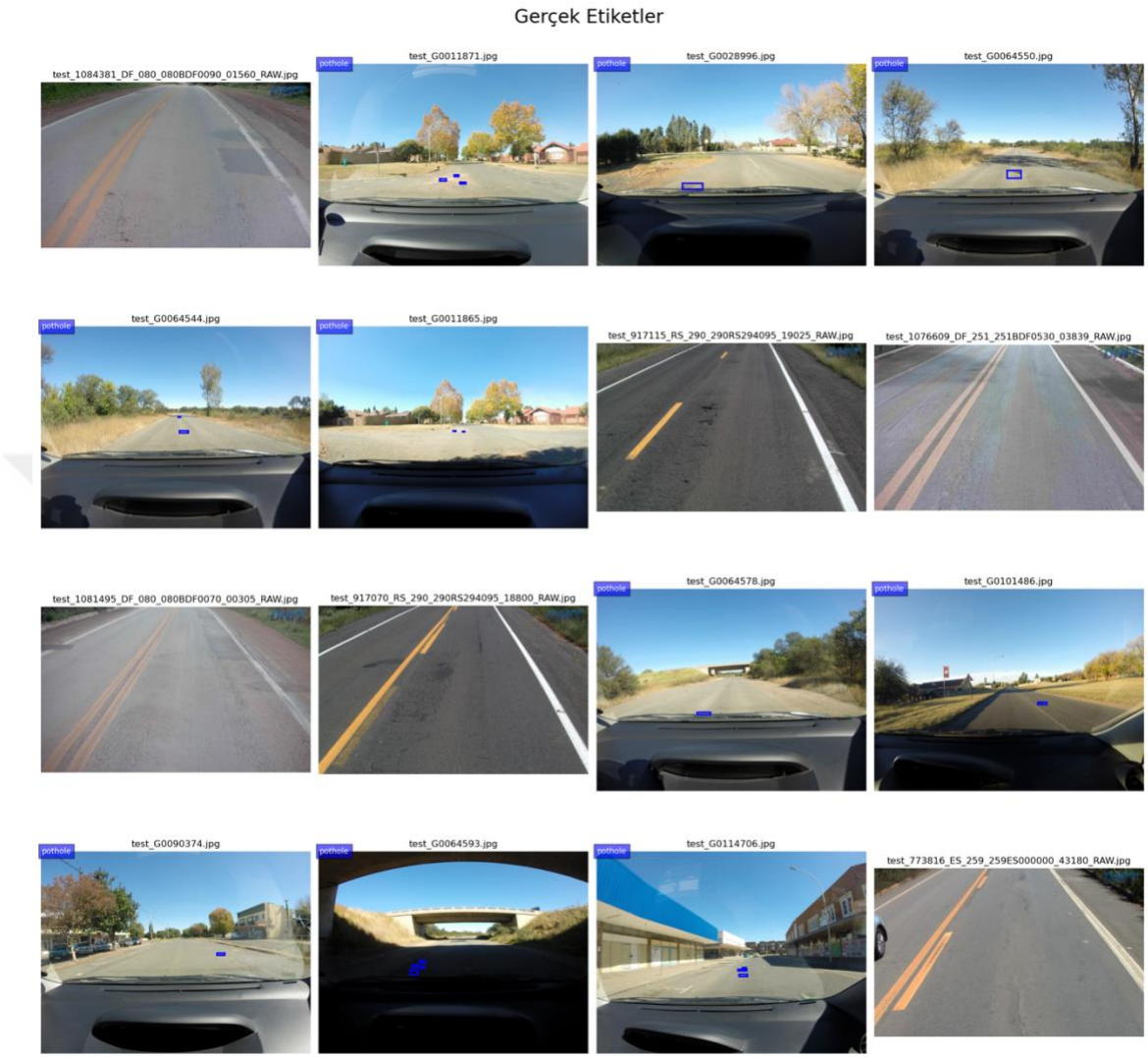


Şekil 5.8: YOLOv8 modelinin PR Eğirisi.

Modelimize en yakın sonucu veren YOLOv8 modeline ait performansı gösteren Precision-Recall Eğrisi Şekil 5.8'de görülmektedir. Tüm sınıflar için ortalama kesinlik değerinin 0.5 IoU (Intersection over Union) eşiğinde 0.646 olduğunu gösteriyor.

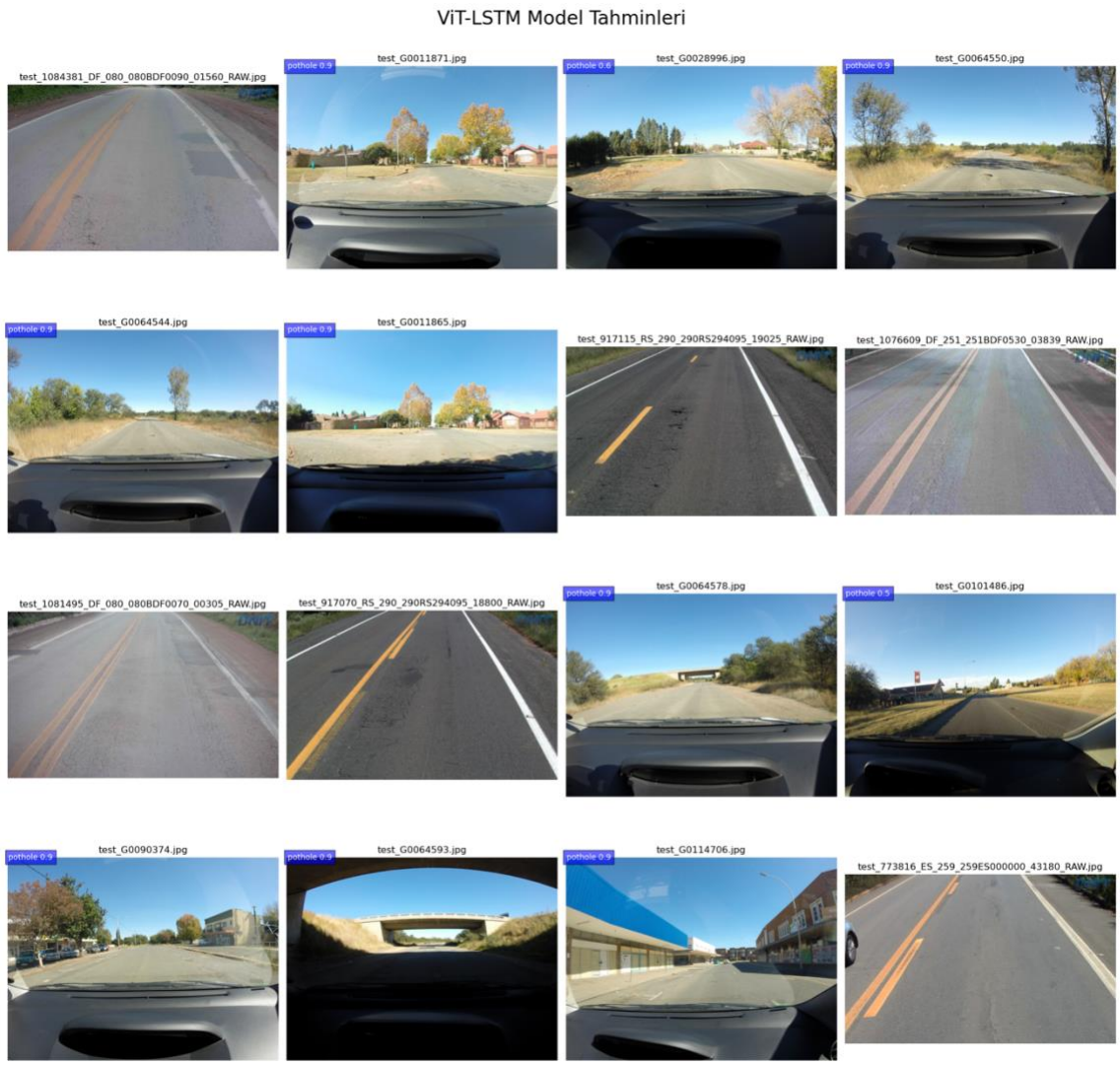
5.4. Model Test Görüntüleri

Şekil 5.9, çalışmada kullanılan veri setinden seçilmiş gerçek etiketli örnek görüntüleri sunmaktadır. Önerdiğimiz ViT-LSTM hibrit modelin test sonuçları Şekil 5.10'da, karşılaştırma amacıyla kullanılan CNN mimarisine ait test çıktıları ise Şekil 5.11'de görülebilir. Değerlendirmenin doğruluğunu artırmak amacıyla veri setinden farklı gerçek zemin etiketleri içeren örnekler Şekil 5.12'de sunulmuştur. Son olarak, karşılaştırmalı analizi tamamlamak üzere YOLOv8 modelinin aynı test verileri üzerindeki sonuçları Şekil 5.13'de incelenebilir.



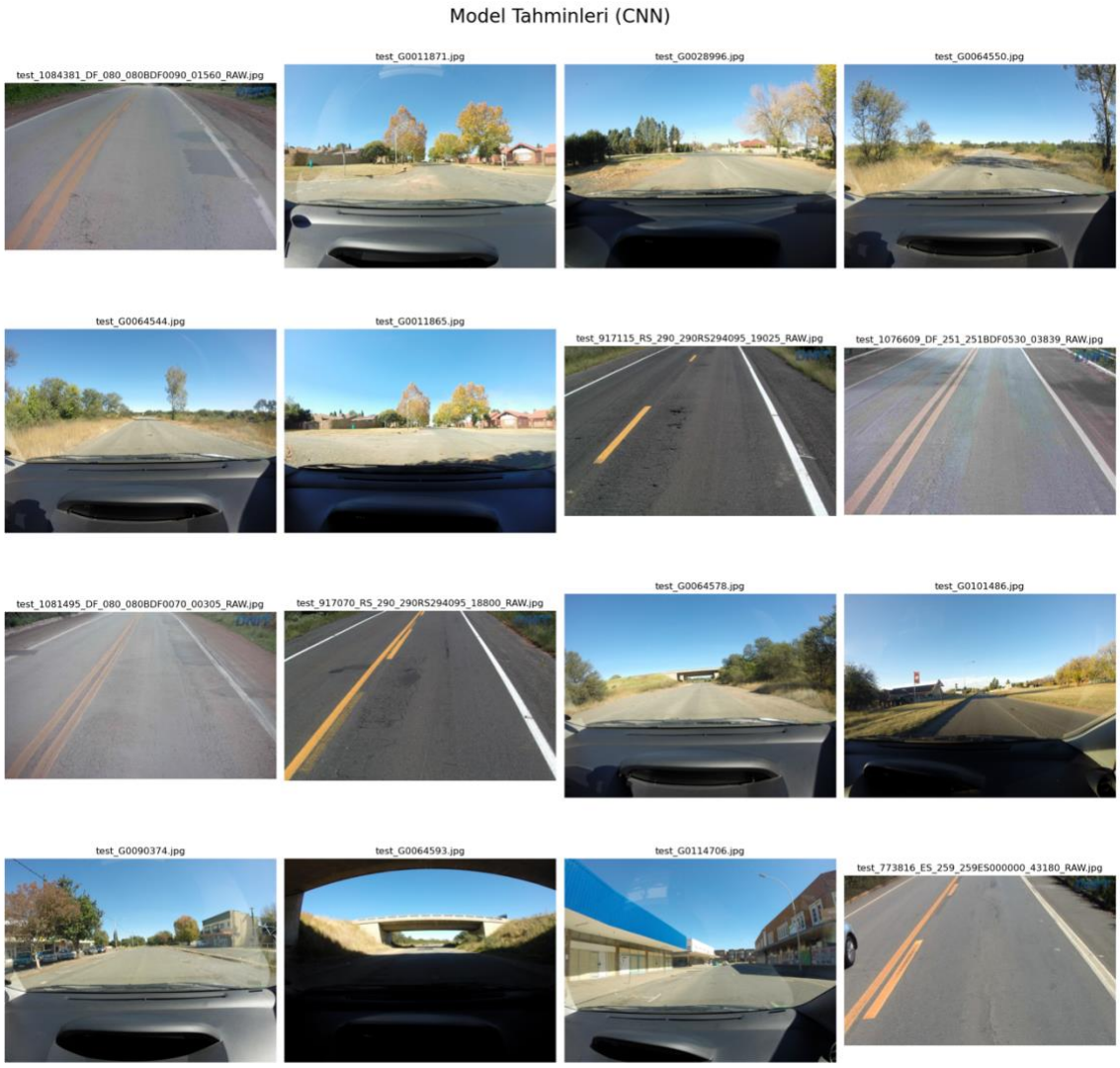
Şekil 5.9: Gerçek zemin fotoğrafları

Kaynak: Nienaber, Booysen, vd., 2015; Ninja, 2025



Şekil 5.10: Modelimizin tahminleri.

Kaynak: Nienaber, Booysen, vd., 2015; Ninja, 2025



Şekil 5.11: CNN modeli tahminleri.

Kaynak: Nienaber, Booysen, vd., 2015; Ninja, 2025

Şekil 5.12: YOLOv8 Gerçek Zemin Görüntüleri

Kaynak: Nienaber, Booysen, vd., 2015

Şekil 5.13: YOLOv8 Tahmin Görüntüleri

Kaynak: Nienaber, Booysen, vd., 2015

ALTINCI BÖLÜM

TARTIŞMA

6.1. Sonuçlar

ViT-LSTM hibrit modelinin performans metrikleri toplu olarak değerlendirildiğinde, modelin yüksek doğruluk (%94.92), yüksek kesinlik (%99.17) ve iyi bir duyarlılık (%90.61) sergilediği görülmektedir. Bu sonuçlar, önerilen hibrit yaklaşımın, Vision Transformer'ın global bağlamdaki özellikleri çıkarma kapasitesi ile LSTM'nin zamansal bağlamları modelleme yeteneğini başarılı bir şekilde birleştirdiğini göstermektedir.

Özellikle dikkat çeken nokta, modelin neredeyse mükemmel olan kesinlik değeridir (%99.17). Bu, modelin pozitif sınıf tahminlerinde son derece güvenilir olduğunu ve neredeyse hiç yanlış pozitif sınıflandırma yapmadığını göstermektedir. Bununla birlikte, göreceli olarak daha düşük olan duyarlılık değeri (%90.61), modelin bazı pozitif örnekleri kaçırdığını işaret etmektedir. Bu durum, modelin daha muhafazakâr bir tahmin stratejisi benimsediğini ve şüpheli durumlarda negatif sınıf tahmininde bulunma eğiliminde olduğunu göstermektedir.

F1 skorunun yüksek olması (%94.69), modelin kesinlik ve duyarlılık arasında iyi bir denge kurduğunu, ROC-AUC eğrisi (0.9756) ve ortalama kesinlik (0.9848) değerlerinin yüksek olması ise modelin farklı eşik değerleri için tutarlı bir performans sergilediğini göstermektedir.

Bu bulgular, önerilen ViT-LSTM hibrit modelinin, görüntü sınıflandırma görevleri için güçlü ve güvenilir bir çözüm sunduğunu doğrulamaktadır. Modelin otonom sürüş uygulamalarında kullanılması durumunda önemli avantajlar sağlayacağı öngörülmektedir.

6.2. Öneriler

Bu çalışma kamera verileri üzerine odaklanmaktadır. Ancak, LIDAR, radar veya diğer sensör verileriyle yapılan veri füzyonu, modelin çevresel koşullarda daha sağlam ve güvenilir sonuçlar vermesini sağlayabilir. Simülasyon ve laboratuvar ortamında elde edilen sonuçların yanı sıra, gerçek dünya koşullarında pilot testlerin yapılması, modelin pratikteki başarısını ölçmek açısından önemli bir adım olabilir. Bu noktada, çeşitli trafik, yol koşullarının test edilmesi önerilebilir.

KAYNAKÇA

Al-Qizwini, M., Barjasteh, I., Al-Qassab, H., & Radha, H. (2017). Deep learning algorithm for autonomous driving using GoogLeNet. *2017 IEEE Intelligent Vehicles Symposium (IV)*, 89-96. https://doi.org/10.1109/IVS.2017.7995703

Attneave, F., B, M., & Hebb, D. O. (1950). The Organization of Behavior; A Neuropsychological Theory. *The American Journal of Psychology*, *63*(4). https://doi.org/10.2307/1418888

Balasubramaniam, A., & Pasricha, S. (2022). Object Detection in Autonomous Vehicles: Status and Open Challenges. *CoRR*, *abs/2201.07706*. Geliş tarihi gönderen https://arxiv.org/abs/2201.07706

Bianco, L. C. L., Beltrán, J., López, G. F., García, F., & Al-Kaff, A. (2020). Joint semantic segmentation of road objects and lanes using Convolutional Neural Networks. *Robotics and Autonomous Systems*, *133*. https://doi.org/10.1016/j.robot.2020.103623

Bojarski, M., Testa, D. D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., … Zieba, K. (2016). *End to End Learning for Self-Driving Cars*. Geliş tarihi gönderen http://arxiv.org/abs/1604.07316

Brummelen, J. V., O'Brien, M., Gruyer, D., & Najjaran, H. (2018). Autonomous vehicle perception: The technology of today and tomorrow. Elsevier Ltd. https://doi.org/10.1016/j.trc.2018.02.012

Chen, B., Gong, C., & Yang, J. (2019). Importance-Aware Semantic Segmentation for Autonomous Vehicles. *IEEE Transactions on Intelligent Transportation Systems*, *20*(1). https://doi.org/10.1109/TITS.2018.2801309

Chen, L., Ding, Q., Zou, Q., Chen, Z., & Li, L. (2020). DenseLightNet: A Light-Weight Vehicle Detection Network for Autonomous Driving. *IEEE Transactions on Industrial Electronics*, *67*(12). https://doi.org/10.1109/TIE.2019.2962413

Chen, S., Zhang, Z., Zhong, R., Zhang, L., Ma, H., & Liu, L. (2021). A Dense Feature Pyramid Network-Based Deep Learning Model for Road Marking Instance Segmentation Using MLS Point Clouds. *IEEE Transactions on Geoscience and Remote Sensing*, *59*(1). https://doi.org/10.1109/TGRS.2020.2996617

Choi, H., Ahn, H., Kim, J., & Jeon, M. (2020). ADFNet: Accumulated decoder features for real-time semantic segmentation. *IET Computer Vision*, *14*(8). https://doi.org/10.1049/iet-cvi.2019.0289

Ciberlin, J., Grbić, R., Teslić, N., & Pilipović, M. (2019). Object detection and object tracking in front of the vehicle using front view camera. *2019 Zooming Innovation in Consumer Technologies Conference, ZINC 2019*. https://doi.org/10.1109/ZINC.2019.8769367

Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., … Schiele, B. (2016). The Cityscapes Dataset for Semantic Urban Scene Understanding. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, *2016-December*, 8. https://doi.org/10.1109/CVPR.2016.350

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., … Houlsby, N. (2020). *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*. Geliş tarihi gönderen http://arxiv.org/abs/2010.11929

Fukushima, K. (1980). Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, *36*(4). https://doi.org/10.1007/BF00344251

Gluhakovic, M., Herceg, M., Popovic, M., & Kovacevic, J. (2020). Vehicle Detection in the Autonomous Vehicle Environment for Potential Collision Warning. *2020 Zooming Innovation in Consumer Technologies Conference, ZINC 2020*, 180. https://doi.org/10.1109/ZINC50678.2020.9161791

Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. R. (2012). *Improving neural networks by preventing co-adaptation of feature detectors*.

Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, *9*(8), 1735-1780. https://doi.org/10.1162/neco.1997.9.8.1735

Hsu, P. Y., Huang, M. L., & Chiang, H. H. (2020). Trajectory of Prediction of Immediate Surroundings for Autonomous Vehicles Using Hierarchical Deep Learning Model. *2nd IEEE Eurasia Conference on IOT, Communication and Engineering 2020, ECICE 2020*. https://doi.org/10.1109/ECICE50847.2020.9301976

Hui, J. (2018). Real-time Object Detection with YOLO, YOLOv2 and now YOLOv3. Geliş tarihi gönderen https://jonathan-hui.medium.com/real-time-object-detection-with-yolo-yolov2-28b1b93e2088

Ishii, K., Tsuichihara, S., Takemura, H., & Mizoguchi, H. (2020). CNN-based System to Identify Bicycle Riders and Pedestrians: Toward Minor Collision Prevention on Sidewalks. *Proceedings of the 2020 IEEE/SICE International Symposium on System Integration, SII 2020*. https://doi.org/10.1109/SII46433.2020.9025905

Jakubec, M., Lieskovská, E., Bučko, B., & Zábovská, K. (2023). Comparison of CNN-Based Models for Pothole Detection in Real-World Adverse Conditions: Overview and Evaluation. MDPI. https://doi.org/10.3390/app13095810

Javadi, S. H., & Farina, A. (2020). Radar networks: A review of features and challenges. *Information Fusion*, *61*. https://doi.org/10.1016/j.inffus.2020.03.005

Jordan, M. I. (1986). Serial order: A parallel distributed processing approach. İçinde *ICS Report* (C. 8604).

Khan, M., Raza, M. A., Abbas, G., Othmen, S., Yousef, A., & Jumani, T. A. (2023). Pothole detection for autonomous vehicles using deep learning: A robust and efficient solution. *Frontiers in Built Environment*, *9*. https://doi.org/10.3389/fbuil.2023.1323792

Kim, H., Lee, Y., Yim, B., Park, E., & Kim, H. (2017). On-road object detection using deep neural network. *2016 IEEE International Conference on Consumer Electronics-Asia, ICCE-Asia 2016*. https://doi.org/10.1109/ICCE-Asia.2016.7804765

Kim, J., Park, B. J., & Kim, J. (2023). Empirical Analysis of Autonomous Vehicle's LiDAR Detection Performance Degradation for Actual Road Driving in Rain and Fog. *Sensors*, *23*(6). https://doi.org/10.3390/s23062972

Kuang, H., Chen, L., Chan, L. L. H., Cheung, R. C. C., & Yan, H. (2019). Feature selection based on tensor decomposition and object proposal for night-time multiclass vehicle detection. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *49*(1), 71-80. https://doi.org/10.1109/TSMC.2018.2872891

Kumar, A., Shen, R., Bubeck, S., & Gunasekar, S. (2022). *How to Fine-Tune Vision Models with SGD*. Geliş tarihi gönderen http://arxiv.org/abs/2211.09359

Lecun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature Publishing Group. https://doi.org/10.1038/nature14539

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., & Jackel, L. D. (1989). Backpropagation Applied to Handwritten Zip Code Recognition. *Neural Computation*, *1*(4). https://doi.org/10.1162/neco.1989.1.4.541

Lee, J. S., & Park, T. H. (2020). Semantic segmentation with Improved Edge Detail for Autonomous Vehicles. *IEEE International Conference on Automation Science and Engineering*, *2020-August*, 523. https://doi.org/10.1109/CASE48305.2020.9217016

Liu, T., Luo, W., Ma, L., Huang, J. J., Stathaki, T., & Dai, T. (2021). Coupled network for robust pedestrian detection with gated multi-layer feature extraction and deformable occlusion handling. *IEEE Transactions on Image Processing*, *30*. https://doi.org/10.1109/TIP.2020.3038371

Liu, Z., Cai, Y., Wang, H., Chen, L., Gao, H., Jia, Y., & Li, Y. (2022). Robust Target Recognition and Tracking of Self-Driving Cars With Radar and Camera Information Fusion Under Severe Weather Conditions. *IEEE Transactions on Intelligent Transportation Systems*, *23*(7). https://doi.org/10.1109/TITS.2021.3059674

Loshchilov, I., & Hutter, F. (2017). *Decoupled Weight Decay Regularization*. Geliş tarihi gönderen http://arxiv.org/abs/1711.05101

Lowphansirikul, C., Kim, K. S., Vinayaraj, P., & Tuarob, S. (2019). 3D Semantic Segmentation of Large-Scale Point-Clouds in Urban Areas Using Deep Learning. *2019 11th International Conference on Knowledge and Smart Technology, KST 2019*. https://doi.org/10.1109/KST.2019.8687813

McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, *5*(4), 113. https://doi.org/10.1007/BF02478259

Mehra, A., Mandal, M., Narang, P., & Chamola, V. (2021). ReViewNet: A Fast and Resource Optimized Network for Enabling Safe Autonomous Driving in Hazy Weather Conditions. *IEEE Transactions on Intelligent Transportation Systems*, *22*(7). https://doi.org/10.1109/TITS.2020.3013099

Narayan, A., Tuci, E., Labrosse, F., & Alkilabi, M. H. M. (2018). A dynamic colour perception system for autonomous robot navigation on unmarked roads. *Neurocomputing*, *275*, 2251-2263. https://doi.org/10.1016/j.neucom.2017.11.008

Naresh, Y. G., Little, S., & O'Connor, N. E. (2018). A residual encoder-decoder network for semantic segmentation in autonomous driving scenarios. *European Signal Processing Conference*, *2018-September*. https://doi.org/10.23919/EUSIPCO.2018.8553161

Nienaber, S., Booysen, M. J. (Thinus), & Kroon, R. (2015). Detecting Potholes Using Simple Image Processing Techniques and Real-World Footage. *Proceedings of the 34th Southern African Transport Conference (SATC 2015)*, (Erişim tarihi: 24 Kasım 2024)

Nienaber, S., Kroon, R. S., & Booysen, M. J. (2015). A comparison of low-cost monocular vision techniques for pothole distance estimation. *Proceedings - 2015 IEEE Symposium Series on Computational Intelligence, SSCI 2015*. https://doi.org/10.1109/SSCI.2015.69, (Erişim tarihi: 24 Kasım 2024)

Ninja, D. (2025). *Visualization Tools for Cracks and Potholes in Road Dataset*. Dataset Ninja. [Online] https://datasetninja.com/cracks-and-potholes-in-road, (Erişim tarihi: 11 Kasım 2024)

Ozturk, G., Koker, R., Eldogan, O., & Karayel, D. (2020). Recognition of Vehicles, Pedestrians and Traffic Signs Using Convolutional Neural Networks. *4th International Symposium on Multidisciplinary Studies and Innovative Technologies, ISMSIT 2020 - Proceedings*. https://doi.org/10.1109/ISMSIT50672.2020.9255148

Placek, M. (2021). *Estimated impact of vehicle automation on collision rates in 2030*. [Online]: https://www.statista.com/statistics/1238242/impact-of-vehicle-automation-on-collision-rates/#statisticContainer, (Erişim tarihi: 24 Eylül 2024)

Prabhakar, G., Kailath, B., Natarajan, S., & Kumar, R. (2017). Obstacle detection and classification using deep learning for tracking in high-speed autonomous driving. *2017 IEEE Region 10 Symposium (TENSYMP)*, 1-6. https://doi.org/10.1109/TENCONSpring.2017.8069972

Pranav, K. B., & Manikandan, J. (2020). Design and Evaluation of a Real-time Pedestrian Detection System for Autonomous Vehicles. *2020 Zooming Innovation in Consumer Technologies Conference, ZINC 2020*. https://doi.org/10.1109/ZINC50678.2020.9161768

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, *2016-December*. https://doi.org/10.1109/CVPR.2016.91

Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological review*, *65 6*, 387.

SAE. (2021). *SAE Levels of Driving Automation™ Refined for Clarity and International Audience*. [Online]: https://www.sae.org/blog/sae-j3016-update, (Erişim tarihi: 14 Ekim 2024)

Saheed, Y. K., Omole, A. I., & Sabit, M. O. (2025). GA-mADAM-IIoT: A new lightweight threats detection in the industrial IoT via genetic algorithm with attention mechanism and LSTM on multivariate time series sensor data. *Sensors International*, *6*, 100297. https://doi.org/10.1016/j.sintl.2024.100297

Shi, S., Wang, X., & Li, H. (2019). PointRCNN: 3D object proposal generation and detection from point cloud. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, *2019-June*. https://doi.org/10.1109/CVPR.2019.00086

Sun, L., Yang, K., Hu, X., Hu, W., & Wang, K. (2020). Real-Time fusion network for rgb-d sementic segmentation incorporating unexpected obstacle detection. *IEEE Robotics and Automation Letters*, *5*(4), 5558-5565. https://doi.org/10.1109/LRA.2020.3007457

TUİK. (2023). Road Traffic Accident Statistics, 2023 (Corrected). [Online]: https://data.tuik.gov.tr/Bulten/Index?p=Road-Traffic-Accident-Statistics-2023-53479&dil=2, (Erişim tarihi: 01 Ekim 2024)

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., … Polosukhin, I. (2017). *Attention Is All You Need*. Geliş tarihi gönderen http://arxiv.org/abs/1706.03762

Wang, H., & Raj, B. (2017). *On the Origin of Deep Learning*. Geliş tarihi gönderen http://arxiv.org/abs/1702.07800

Wang, J. G., Zhou, L., Pan, Y., Lee, S., Song, Z., Han, B. S., & Saputra, V. B. (2016). Appearance-based Brake-Lights recognition using deep learning and vehicle detection. *IEEE Intelligent Vehicles Symposium, Proceedings*, *2016-August*. https://doi.org/10.1109/IVS.2016.7535481

Weihmayr, D., Sezgin, F., Tolksdorf, L., Birkner, C., & Jazar, R. N. (2024). Predicting the Influence of Adverse Weather on Pedestrian Detection with Automotive Radar and Lidar Sensors. *2024 IEEE Intelligent Vehicles Symposium (IV)*, 2591-2597. IEEE. https://doi.org/10.1109/IV55156.2024.10588472

Werbos, P. J. (1974). The roots of backpropagation: From ordered derivatives to neural networks.

Wilkes, A. L., & Wade, N. J. (1997). Bain on neural networks. *Brain and Cognition*, *33*(3), 301. https://doi.org/10.1006/brcg.1997.0869

Xi, H., Guo, W., Yang, Y., Yuan, R., & Ma, H. (2024). Cross-attention mechanism-based spectrum sensing in generalized Gaussian noise. *Scientific Reports*, *14*(1), 23261. https://doi.org/10.1038/s41598-024-74341-4

Yan, M., Wang, J., Li, J., Zhang, K., & Yang, Z. (2020). Traffic scene semantic segmentation using self-attention mechanism and bi-directional GRU to correlate context. *Neurocomputing*, *386*, 293-304. https://doi.org/10.1016/J.NEUCOM.2019.12.007

Yang, J., Wang, C., Wang, H., & Li, Q. (2020). A RGB-D Based Real-Time Multiple Object Detection and Ranging System for Autonomous Driving. *IEEE Sensors Journal*, *20*(20). https://doi.org/10.1109/JSEN.2020.2965086

Yoshioka, M., Suganuma, N., Yoneda, K., & Aldibaja, M. (2017). Real-time object classification for autonomous vehicle using LIDAR. *ICIIBMS 2017 - 2nd International Conference on Intelligent Informatics and Biomedical Sciences*, *2018-January*, 211. https://doi.org/10.1109/ICIIBMS.2017.8279696

Zhao, J., Zhao, W., Deng, B., Wang, Z., Zhang, F., Zheng, W., … Burke, A. F. (2024). Autonomous driving system: A comprehensive survey. Elsevier Ltd. https://doi.org/10.1016/j.eswa.2023.122836

Zhao, L., Wang, M., Su, S., Liu, T., & Yang, Y. (2020). Dynamic object tracking for self-driving cars using monocular camera and LIDAR. *IEEE International Conference on Intelligent Robots and Systems*. https://doi.org/10.1109/IROS45743.2020.9341179

Zhao, X., Sun, P., Xu, Z., Min, H., & Yu, H. (2020). Fusion of 3D LIDAR and Camera Data for Object Detection in Autonomous Vehicle Applications. *IEEE Sensors Journal*, *20*(9). https://doi.org/10.1109/JSEN.2020.2966034

Zhou, W., Berrio, J. S., Worrall, S., & Nebot, E. (2020). Automated Evaluation of Semantic Segmentation Robustness for Autonomous Driving. *IEEE Transactions on Intelligent Transportation Systems*, *21*(5). https://doi.org/10.1109/TITS.2019.2909066

ÖZGEÇMİŞ

Ad Soyad	Mustafa TOKAT
İletişim Bilgileri	• ORCID: 0009-0003-0724-8930
Eğitim	• İstanbul Sabahattin Zaim Üniversitesi Bilgisayar Bilimleri ve Mühendisliği (Yüksek Lisans) 2022 – Devam ediyor. • Niğde Üniversitesi Sınıf Öğretmenliği (Lisans) — 2003 – 2007
Profesyonel Deneyim	• Milli Eğitim Bakanlığı (MEB) - Sınıf Öğretmeni – 3 yıl - Okul Müdürü – 14 yıl • Toromedya (2015–2018) - C# ve Unity ile artırılmış gerçeklik tabanlı mobil proje geliştirme
Yayınlar ve Konferanslar	• "Finding the Flies in the Sky Using Advanced Deep Learning Algorithms" – FoNeS-AIoT 2024 Konferansı • "Colorectal Cancer Diagnosis with Deep Learning Models" – The 28th WMSCI 2024 Konferansı
Teknik Beceriler	Python, Tensorflow ve diğer makine görmesi kütüphaneleri, Unity, C# temelli geliştirme
Yabancı Dil	İngilizce: B2 düzeyi