

T.C.
BANDIRMA ONYEDİ EYLÜL ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

YÜKSEK LİSANS TEZİ

OTONOM ARAÇLAR İÇİN ŞERİT VE NESNE TESPİTİ

Md Nasrullah

Akıllı Ulaşım Sistemleri ve Teknolojileri Anabilim Dalı

Akıllı Ulaşım Sistemleri ve Teknolojileri Programı

AĞUSTOS 2022

T.C.
BANDIRMA ONYEDİ EYLÜL ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

YÜKSEK LİSANS TEZİ

OTONOM ARAÇLAR İÇİN ŞERİT VE NESNE TESPİTİ

Md Nasrullah

DANIŞMAN
Dr. Öğr. Üyesi Aykut DİKER

Akıllı Ulaşım Sistemleri ve Teknolojileri Anabilim Dalı

Akıllı Ulaşım Sistemleri ve Teknolojileri Programı

AĞUSTOS 2022

ONAY

Md Nasrullah tarafından hazırlanan “**Otonom Araçlar İçin Şerit ve Nesne Tespiti**” adlı tez çalışması .../.../..... tarihinde yapılan sınavla aşağıdaki jüri tarafından oybirliği/oyçokluğu ile Bandırma Onyedİ Eylöl Üniversitesi Fen Bilimleri Enstitüsü Akıllı Ulaşım Sistemleri ve Teknolojileri Anabilim Dalı’nda YÜKSEK LİSANS TEZİ olarak kabul edilmiştir.

Jüri Üyeleri

İmza

Dr. Öğr. Üyesi Aykut DİKER

(Danışman)

Prof. Dr. Mehmet TEKTAŞ

(Üye)

Doç. Dr. Fatih ÖZYURT

(Üye)

Doç. Dr. Gültekin BASMACI

(Kurum Dışı Yedek Üye)

Doç. Dr. Üyesi İlyas ÖZER

(Kurum İçi Yedek Üye)

Bu tezin kabulü, Fen Bilimleri Enstitüsü Yönetim Kurulu’nun .../.../..... gün ve .../... sayılı kararı ile onaylanmıştır.

Doç. Dr. Abdullah YEŞİL

Enstitü Müdürü

BANDIRMA ONYEDİ EYLÜL ÜNİVERSİTESİ FEN BİLİMLERİ ENSTİTÜSÜ

YÜKSEK LİSANS ÇALIŞMASI

ETİK BEYANI

Bandırma Onyedi Eylül Üniversitesi Fen Bilimleri Enstitüsü tez yazım kılavuzuna göre hazırlamış olduğum “**Otonom Araçlar İçin Şerit ve Nesne Tespiti**” adlı tezimin özgün bir çalışma olduğunu, tez hazırlanırken tüm aşamalarda bilimsel etik ilkelerine uygun davrandığımı, tez kapsamında sunulan tüm verileri bilimsel etik ilkelerine uygun elde ettiğimi, tezde faydalandığım tüm eserlere atıf yaptığımı ve kaynaklar kısmında bu eserleri gösterdiğimi beyan ederim.

...../...../20...

Md Nasrullah

İmza

ÖNSÖZ

Otonom Araçlarda Python ve Derin Öğrenme ile Şerit ve Nesne Tespiti adlı çalışmada otonom araçlar ile ilgili incelemeler, araştırmalar yapılmış olup verimliliği artırıcı ve günümüzde önemli bir sorun haline gelen ulaşımı kolaylaştırıcı yöntemler ortaya çıkarılmıştır.

Bu çalışmanın gerçekleştirilmesinde desteğini esirgemeyen ve önemli katkılarda bulunan danışmanım Dr. Öğr. Üyesi Aykut DİKER'e teşekkürlerimi sunarım.

Yüksek lisans süresince çalışma motivasyonumu arttıran, başından sonuna kadar her konuda beni cesaretlendiren ve destekleyen Bölüm Başkanı Prof. Dr. Mehmet TEKTAŞ'a teşekkür ederim.

Hayatımda en kıymetli yardımcım ve destekleriyle hep yanımda olan sevgili babama, aileme ve arkadaşlarıma teşekkür ederim.

ÖZET

YÜKSEK LİSANS TEZİ

OTONOM ARAÇLAR İÇİN ŞERİT VE NESNE TESPİTİ

Md Nasrullah

Bandırma Onyedli Eylül Üniversitesi

Fen Bilimleri Enstitüsü

Akıllı Ulaşım Sistemleri ve Teknolojileri Anabilim Dalı

Danışman: Dr. Öğr. Üyesi Aykut DİKER

Ağustos 2022, 54 sayfa

Otonom araçlar ulaşımın geleceği açısından önemli bir rol oynamaktadır. Yapay zekâdaki gelişmeler sayesinde, gelecekte akıllı otomobillerin ve kamyonların sürücüsüz olması, kazaların azalması ve daha verimli hale gelecekleri öngörülmektedir. Aynı zamanda tepki süresi gecikmesini azaltarak, güvenli şerit değiştirme ihtimalini artırarak ve insan sürüş hatalarını ortadan kaldırarak güvenli seyahat sağlamaktadırlar. Otomobil üreticileri halihazırdaki zorluklarla ilgili araştırma alanlarına yatırım yapmışlardır. Makine öğrenme algoritmalarındaki son gelişmelerin ardından, otonom araçların önümüzdeki yirmi yıl içinde piyasaya çıkması beklenmektedir. Derin öğrenmenin ve derin pekiştirici öğrenme algoritmalarının kullanımının artmasıyla birlikte otonom araçların artışı doğru orantılı olmuştur. Derin öğrenme algoritmaları otonom araç uygulamalarında, Evrişimli Sinir Ağları (ESA) ve yakın zamanlarda ise YOLO algoritması ile daha hızlı ve daha doğru sonuçlar elde edilmiştir. Bu çalışmada, otonom araçlar için nesneler tespiti ve şerit belirleme üzerine bir uygulama yapılmıştır. Derin Sinir Ağları (DSA) ile 1176 tane resim verisi üzerinde yapılan uygulamada nesne takibinde kullanılan YOLO algoritması, Hızlı R-ESA, SimpleNet ve ESA'nın VGG16 yöntemleri algılama süresi açısından karşılaştırılmıştır. Aynı zamanda, şerit tespiti ve şerit değiştirme için Python OpenCV programı kullanılmış, bu amaçla yeni kod yazılmış ve metotlar geliştirilmiştir.

Anahtar kelimeler: Makine Öğrenme, Derin öğrenme, YOLO Algoritması, Python OpenCV

ABSTRACT

M.Sc. THESIS

OBJECT AND LANE DETECTION FOR AUTONOMOUS VEHICLE

Md Nasrullah

Bandırma Onyedi Eylül University

Graduate School of Natural and Applied Sciences

Department of Intelligent Transportation Systems and Technologies

Supervisor: Asst. Prof. Dr. Aykut DİKER

August 2022, 54 pages

Autonomous vehicles play an important role in the future of transportation. It is blessed to the developments in artificial intelligence that it is predicted that smart cars and trucks will become driverless, accidents will decrease and they will become more efficient in the future. They also provide safe travel by reducing reaction time delay, increasing the likelihood of safe lane changes, and eliminating human driving errors. Automakers have invested in research areas relevant to current challenges. Following the latest developments in machine learning algorithms, autonomous vehicles are expected to hit the market in the next two decades. Among the increase in the use of deep learning and deep reinforcement learning algorithms, the increase in autonomous vehicles has been directly proportional. Faster and more accurate results have been obtained with deep learning algorithms, Convolutional Neural Networks (CNN), and recently YOLO algorithm in autonomous vehicle applications. In this study, an application has been made on object detection and lane determination for autonomous vehicles. YOLO algorithm used in object tracking, Fast R-CNN, SimpleNet, and CNN's VGG16 methods were compared in terms of detection time in the application made with Deep Neural Networks (DNN) on 1176 image data. At the same time, the Python-OpenCV program was used for lane detection and lane changing. In this study, we have done some important parts of autonomous vehicles, and their several code was written and methods were developed for this purpose.

Keywords: Machine Learning, Deep Learning, YOLO algorithm, Python OpenCV

İÇİNDEKİLER

	<u>Sayfa</u>
ÖZET	I
ABSTRACT	II
İÇİNDEKİLER.....	III
ŞEKİL LİSTESİ	VI
TABLO LİSTESİ.....	VII
SİMGE ve KISALTMA LİSTESİ.....	VIII
1. GİRİŞ.....	1
1.1 LİTERATÜR ÇALIŞMASI.....	1
1.2 OTONOM ARABALAR.....	2
1.3 OTONOM, OTOMATİK VE OTONOM SÜRÜŞ	4
1.4 OTONOM ARABALARDA DİKKAT EDİLMESİ GEREKEN PARAMETRELER	4
1.4.1 Hava Koşulları.....	4
1.4.2 Trafik Kuralları.....	5
1.4.3 Kaza Sorumluluğu.....	5
1.4.4 Yapay Zekâ ve Duygusal Zekâ	5
1.5 OTONOM ARABALARIN FAYDALARI	5
2. OTONOM ARAÇLARDA KULLANILAN SENSÖRLER	7
2.1 LİDAR	8
2.2 RADAR.....	8
2.3 DİJİTAL KAMERALAR	9
2.4 ULTRASONİK ALGILAYICILAR	9
2.5 KÜRESEL KONUMLAMA SİSTEMİ	10
2.6 ATALET ÖLÇÜM BİRİMİ	10

2.7	ENKODERLER.....	11
3.	ŞERİT ALGILAMA.....	12
3.1	ŞERİT ALGILAMA METODLARI.....	12
3.1.1	İlgili Bölge Metodu	13
3.1.2	Düz ve Eğri Bölgenin Ayrılması.....	13
3.1.3	Düz Model Çözümleme	13
3.1.4	Eğri modeli çözümleme	14
3.1.5	Düz Eğri Modelinin Değiştirilmesi.....	14
3.1.6	Viraj Büküm Yönünün Belirlenmesi	14
3.2	ÖNERİLEN YÖNTEM.....	15
3.2.1	Görüntü Yakalama.....	16
3.2.2	Kırpılmış Resim	16
3.2.3	Gri Ölçeğe Dönüştürme	17
3.2.4	Gürültü Azaltma.....	18
3.2.5	Kenar Algılama.....	19
3.2.6	Hat Algılama	20
3.2.7	Şerit Sınır Taraması	21
3.2.8	Hiperbol Uydurma	23
3.2.9	Yolda Şerit Çizgisi Bulma.....	23
3.3	ŞERİT DEĞİŞTİRME KARARI.....	24
3.3.1	Şerit Değiştirme Stratejisi Tasarımı	25
3.3.2	Şerit Değiştirme Yolu Planlaması	27
3.3.3	Polinom Fonksiyonuna Dayalı Yol Planlama Yöntemi.....	27
3.3.4	Kısıtlama Analizi ve Optimizasyonu.....	29
4.	OTONOM ARAÇLARDA NESNE ALGILAMA İÇİN KULLANILAN ALGORİTMALAR	32
4.1	AŞIRI BAŞARI ALGORİTMASI.....	32

4.2	TAHMİN OLUŞTURMA	35
4.3	EVİRİŞİMSEL SİNİR AĞI	36
4.4	HIZLI R-ESA.....	37
4.5	YOLO ALGORİTMASI	38
5.	YAKLAŞIM	41
5.1	GİZLİ BİRİMLER.....	41
5.2	DOĞRUSAL OLMAYAN AKTİVASYON FONKSİYONU	41
5.3	ALGILAMA TAHMİNİ	43
5.4	UFUK ÇİZGİSİ BASTIRMA.....	43
5.5	KAYIP FONKSİYONU	43
5.6	EĞİTİM ALGORİTMASI	44
6.	DENEYSEL ÇALIŞMALAR	45
6.1	VERİ SETİ.....	45
6.2	BİLGİ İŞLEM ORTAMI.....	46
6.3	ALGORİTMALARIN PERFORMANS ANALİZİ	46
6.4	EĞİTİM.....	47
7.	SONUÇLAR.....	49
8.	KAYNAKLAR.....	51
	ÖZGEÇMİŞ	54

ŞEKİL LİSTESİ

<u>Sekil</u>	<u>Sayfa</u>
Şekil 2-1: Otonom Araçlarda kullanılan sensörler [20]	7
Şekil 3-1: Düz eğri modeli kullanarak şerit algılama	12
Şekil 3-2: Bilgisayarla görme algoritmasına dayalı önerilen şerit tespit tekniği.	15
Şekil 3-3: 3 boyutlu renkli resimden 1 boyutlu Gri resme çevrilmesi	17
Şekil 3-4: 3 boyutlu resim normalizasyonu	18
Şekil 3-5: Kenar algılama.....	20
Şekil 3-6: Hough Dönüşümü.....	21
Şekil 3-7: Görüntü koordinatları	22
Şekil 3-8: Şerit çizgi belirtme.....	22
Şekil 3-9: Görüntü üzerinde uygulanması	24
Şekil 3-10: Zorunlu şerit değişikliği	25
Şekil 3-11: Karar verme stratejisi.....	26
Şekil 3-12: Şerit değiştirme şeması.....	28
Şekil 3-13: Kısıt analizi	30
Şekil 4-1: Algılama için ConvNets'in verimliliği.....	36
Şekil 4-2: ESA Yapısı	37
Şekil 4-3: YOLO algoritması mimarisi.....	39
Şekil 4-4: YOLO algoritması çıktısı.....	39
Şekil 5-1: SimpleNet Mimarisi.....	42
Şekil 6-1: Örnek Veri görüntüsü.....	45
Şekil 6-2: Hata Matris	46
Şekil 7-1: Bir eğitim görüntüsünde pozitif tahmin sonuçları	49

TABLO LİSTESİ

<u>Tablo</u>	<u>Sayfa</u>
Tablo 4-1: Aşırı başarı için mimari özellikler	34
Tablo 4-2: Çok ölçekli yaklaşımının mekânsal boyutları	35
Tablo 4-3: Örnek bir Hızlı R-ESA mimarisi.....	37
Tablo 6-1: Veri kümesi dağılımı.....	45
Tablo 6-2: Derin nesne dedektörleri için tahmin doğruluğu	47
Tablo 6-3: Derin nesne dedektörleri için tahmin hızları.....	48

SİMGE ve KISALTMA LİSTESİ

Simgeler	Açıklama
σ	: Standart sapma
σ^2	: Varyans

Kısaltmalar	Açıklama
AI	: Yapay Zekâ
ML	: Makina Öğrenme
DL	: Derin Öğrenme
NN	: Sınır ağı
CNN	: Evrişimsel Sınır Ağı (ESA)
RL	: Pekiştirmeli Öğrenme
LP	: Doğrusal programlama
DP	: Dinamik program
YOLO	: You Only Look Once
ROI	: İlgili bölge
SLAM	: Eşzamanlı Yerelleştirme ve Haritalama (EYH)

1. GİRİŞ

Trafik kazaları dünyadaki en karmaşık sorunların başında gelmektedir. Türkiye'de her dört dakikada bir kişi trafik kazasında ölüyor. 2020 yılı boyunca yaklaşık 1.37 bin kişi yalnızca trafik kazalarından öldü. Ayrıca Türkiye'de her gün 16 çocuk trafik kazasında vefat ediyor. Bu kazalar, kusurlu yol zemini, insan ve hayvanların beklenmedik şekilde yola çıkmaları ve kaza yapmaları, kaygan ve virajlı yol gibi sebeplerden kaynaklanmaktadır [1]. Bu nedenle virajlı yol koşullarında araçların dönüşlerde aşırı hız yapmaması, şerit değişikliği ve şerit doluluklarındaki düzensizliklerin azaltılması için yoldaki şerit işaretlerinin tespit edilmesi gerekmektedir. Bilgisayarlı görme algoritmaları kullanılarak şerit tespiti, şerit değiştirme kararı ve nesne tespiti gibi uyarı sistemleri tasarlanmıştır [2]. Bilgisayarla görü tabanlı teknolojinin sınırsız gelişmesiyle birlikte bir ortamı çeşitli yönlerde yakalayabildiği için otonom araçlara ihtiyaç artmaktadır. Bu çalışmada otonom araçların en çok ihtiyacı olan nesne tespiti, derin öğrenme, Python ve makine öğrenme metotlarla sağlanmaktadır.

1.1 LİTERATÜR ÇALIŞMASI

Teknolojinin ilerlemesi ve otonom araçların hayatımıza her geçen gün daha fazla girmesiyle birlikte otonom araçların performansı ile ilgili akademik çalışmaların sayısı artmaktadır. Anderson ve arkadaşları [3] tarafından yapılan çalışma Otonom Araçlardaki sensorlar, nesneler algılamakta hangi tekniği gerekli onlara açıklanmaktadır.

Arıkan [4] tarafından Mersin Üniversitesinde Bilgisayar Mühendisliği Bilimleri-Bilgisayar ve Kontrol bölümünde çalışma olarak bazı araçların önünde bulunan mikro dalga radarı ile mesafe ölçümü yerine iki adet kamera kullanılarak derin öğrenme tabanlı gömülü sistem bir mesafe tahmin prototipi hazırlanmıştır. Bu prototip öncelikle kenar bulma algoritması ile test edilmiş, ardından oyuncak arabalar ile farklı uzaklıklarda cm bazında ölçüm tahminleri alınmıştır. Ayrıca kamera gürültüsü ve kalibrasyon hataları işlenmiş en optimum değerler saptanarak üzerine YOLO ile görüntü tanımlama kısmı eklenmiştir. En son alınan değerler YOLO'nun çerçeve hesabıyla gerçek arabalarla yapılmıştır. Alınan sonuçlara göre Jetson Nano platformu gerçek zamanlı nesne tanımlama ve uzaklık tahmini konusunda çok yetersiz olduğu yapılan çalışmalarca ortaya konulmuştur. Gerçek hayatta saniyede 5 kare işlem hızıyla seyir halinde hiçbir araca kullanılamayacağı verilerle sabitlenmiştir. Ayrıca yapılan 150~1500 cm arası ölçüm çalışmasında,

hata oranlarının 500 cm de ~ 0.15 iken 1100 cm de ~ 90.5 olarak hesaplandığından ölçeklendirilebilir olmadığı kanaatine varılmıştır. Bu sonuçlar ışığında öneriler bölümünde başka yöntemler tavsiye edilmiştir.

Bochkovskiy [5] tarafından otonom araçlardaki nesne tespitinde doğrulu ve nesneleri daha verimli halde algılamak için farklı farklı Kayıp fonksiyonu, farklı farklı derin öğrenmeler ve son olarak kesinlik değeri 91% ve Hassasiyet değeri 90.2 % elde ederek çok başarılı bir algoritma yapılmıştır.

Demir [6] tarafından otonom araçlar üzerinde çalışma da lazer tabanlı bir lokalizasyon ve haritalama tekniğinin üzerine odaklanılmıştır. Gerçekleştirilen sistemde sanal bir test ortamı kurulmuş ve bir otonom araç üzerinde denemeler yapılmıştır. Çalışma kapsamında sanal makineler oluşturularak üzerlerine Linux işletim sistemi kurulmuştur. Sonra bu sanal makinelere ROS ortamında TurtleBot3 kurulmuş ve iç mekân lokalizasyonu yapılarak bir harita elde edilmiştir. Bu harita genetik algoritma ile en kısa mesafelerin bulunmasını sağlamak için kullanılmaktadır. Gözlemler neticesinde simülasyon ortamındaki robot yüksek başarımla istenilen konuma gidebildiği sonucuna ulaşılmıştır.

1.2 OTONOM ARABALAR

Otonom bir araç, çevresini algılayabilen ve insan müdahalesi olmadan çalışabilen bir araçtır. Otonom araçlar, sürücüsüz arabalar veya robotik arabalar olarak da bilinir. Bu teknolojiler gelişmeye devam ettikçe sürücüsüz araba terimi bir standart haline gelmektedir [7]. Bir yolcunun herhangi bir zamanda aracın kontrolünü ele alması veya araçta bir insanın bulunması gerekli değildir. Otonom bir araba, geleneksel bir arabanın gittiği her yere gidebilir ve deneyimli bir sürücünün yaptığı her şeyi yapabilir. Otomasyon seviyelerine bağlı olarak farklı türlerde otonom araçlar vardır. Otomotiv Mühendisleri Topluluğu (OMT) şu anda Seviye 0'dan (tamamen manuel) Seviye 5'e (tamamen otonom) kadar değişen 6 sürüş otomasyonu seviyesi tanımlamaktadır. Bu seviyeler ABD Ulaştırma Bakanlığı tarafından kabul edilmiştir. Otonom arabalar akıllı ulaşım sektörünün en temel bileşenidir. Akıllı ulaşım sektörüne dağılımı akıllı robotların gelişmesiyle de hızlı bir şekilde artmaktadır [8].

İnsan hatasının toplam kazaların %94'ünü oluşturduğu tahmin edilen yol güvenliğinin iyileştirilmesi sağlanmıştır [9].

- İŖe gidiŖ ve geliŖ deneyiminin iyileŖtirilmesi, iŖe gidiŖ geliŖ zamanının bir kısmının ara kendi kendine park etme fonksiyonları gibi iŖlemleri gerekleŖtirdiėinde srŖ dıŖındaki grevlere yeniden tahsis edilmesine izin vererek iŖe gidiŖ geliŖ sresini kısaltma potansiyeli saėlanmıŖtır [3].
- Herkes iin uzun sredir aranan hareketliliėin iyileŖtirilmesi, farklı yeteneklere sahip kiŖilerin ulaŖıma eriŖmesini saėlamak ve baėımsızlıėı geliŖtirme olanaėı saėlanmıŖtır [10].
- Yakıt tasarrufu potansiyeli ve diėer Ŗeylerin yanı sıra bu tr teknolojiyi daha geleneksel ulaŖım aralarına yeŖil ve evre dostu bir alternatif olarak sınıflandırmaya yardımcı olan daha ynetilebilir park dzenlemeleri saėlanmıŖtır [3].

Otonom aralar, yazılımı yrtmek iin sensrler, aktatrler, karmaŖık algoritmalar, makine ėrenme sistemleri ve gl iŖlemciler gibi sistemlerle alıŖır. Otonom aralar, aracın farklı yerlerinde bulunan eŖitli sensrlere dayalı olarak evrelerinin bir haritasını oluŖturur ve srdrr. Radar sensrleri yakındaki araların konumunu izler. Video kameralar trafik ıŖıklarını algılar, yol iŖaretlerini okur, diėer araları takip eder ve yayaları arar. Lidar (ıŖık algılama ve mesafe) sensrleri, mesafeleri mek, yol kenarlarını algılamak ve Ŗerit iŖaretlerini belirlemek iin aracın evresinden ıŖık atımlarını yansıtır. Tekerleklerdeki ultrasonik sensrler, park ederken kaldırımları ve diėer araları algılar [11].

Otonom aralar, evrelerini anlamak ve komutlara tepki vermek iin geliŖmiŖ yapay zek ve makine ėrenimi sistemleri ile alıŖır. GeliŖmiŖ bilgisayar grme iŖlevleriyle birlikte karmaŖık sensrler ve aktatrler, evrelerinin srekli deėiŖen haritasını oluŖturmak, yakındaki araların ve yayaların varlıėını tespit etmek, mesafeleri mek ve yollarda ve kaldırımlardaki engebeli yzeyleri tespit etmek iin kullanılır. Otonom aralar, akıllı trafik ıŖıkları ve diėer harici cihazlarla da birbirine baėlanabilir, ancak bu henz gerekleŖtirilmemiŖtir.

GeliŖmiŖ yazılım programı daha sonra tm bu duyusal girdileri iŖler, bir yol izer ve aracın hızlanma, frenleme ve direksiyonu kontrol eden aktatrlerine talimatlar gnderir. Sabit kodlanmış kurallar, engellerden kaınma algoritmaları, tahmine dayalı modelleme ve nesne tanıma, yazılımın trafik kurallarını izlemesine ve engellerde gezinmesine yardımcı olur.

1.3 OTONOM, OTOMATİK VE OTONOM SÜRÜŞ

Otomotiv Mühendisleri Topluluğu, otonom yerine otomatik terimini kullanır. Tam otonom bir araba, kendinin farkında olacak ve kendi seçimlerini yapabilecek kapasitede olacaktır. Örneğin, sahile gitmek gerekirken arabaya komut veren kişinin “beni işe götür” demesi üzerine, araba sizi bir plan dâhilinde sahile götürmeye karar veriyor. Bununla birlikte, tam otomatik bir araba, emirleri takip edecek ve sonra kendi kendini sürecektir [12]. Diğer bir örnekte akıllı depolarda çalışan otonom robotlar yükü insan müdahalesi bulunmadan hedef rafa iletir [13].

Kendi kendine sürüş terimi genellikle otonom kelimesi ile birbirinin yerine kullanılır. Ancak, biraz farklılık var. Kendi kendini süren bir araba, bazı durumlarda veya tüm durumlarda kendi kendine gidebilir. Ancak bir yolcunun her zaman hazır ve kontrolü ele almaya hazır olması gerekir. Kendi kendine giden arabalar, Seviye 3'ün (koşullu sürüş otomasyonu) veya Seviye 4'ün (yüksek sürüş otomasyonu) altına düşecektir. Her yere gidebilen tamamen otonom Seviye 5 arabanın aksine, coğrafi sınırlamaya tabidirler.

1.4 OTONOM ARABALARDA DİKKAT EDİLMESİ GEREKEN PARAMETRELER

Tam otonom (Seviye 5) otomobiller dünyanın çeşitli yerlerinde test ediliyor, ancak hiçbiri henüz genel kullanıma açık değil. Bundan hala çok uzaktayız [14]. Aşağıda otonom arabaların dikkat etmesi gereken parametreler açıklanmıştır.

1.4.1 Hava Koşulları

Otonom bir araba yoğun yağışta sürüldüğünde arabada entegre olan sensorlar algılamada zorluk yaşarlar. Yolda bir kar tabakası varsa şerit ayırıcıların otonom araba tarafında görülmesi zorlaşır. İşaretler su, yağ, buz veya enkaz tarafından kapatılırsa kameralar ve sensörler şerit işaretlerini nasıl takip etmekte zorlanır. Otonom araba bu olumsuz hava şartlarına uygun olarak üretilmesi gerekir [15].

1.4.2 Trafik Kuralları

Otonom araba kullanıldığı bölgenin trafik kurallarına uyması gerekmektedir. Bu kurallardan biri azami hız sınırıdır. Otonom arabanın hareket ettiği yolda azami hız sınırına uyması zorunludur. Bundan dolayı otonom arabanın bölgelere göre ayar parametreleri olması gerekmektedir. Ayrıca, örneğin 3 şeritli yollarda otonom araba belirli bir hızın altında hareket ediyorsa, gideceği şeridi ona göre belirlemesi gereklidir.

1.4.3 Kaza Sorumluluğu

Otonom arabanın neden olduğu kaza durumunda sorumlu olarak tanımlanacak kişi hakkında düzenlemeler getirilmelidir. Otonom arabanın üretimini yapan üretici veya arabanın içinde bulunan yolcunun kazaya neden olma olasılıkları değerlendirilmelidir. En son çalışmalar, tamamen otonom bir Seviye 5 otomobilin bir gösterge panosuna veya direksiyon simidine sahip olmayacağını, bu nedenle bir insan yolcunun acil bir durumda aracın kontrolünü ele geçirme seçeneğine bile sahip olmayacağını gösteriyor [16] .

1.4.4 Yapay Zekâ ve Duygusal Zekâ

Otonom olmayan arabalar kullanılırken sürücüler, anlık karar uyarılarını yapmak ve çevredeki davranışları tahmin etmek için yayalarla göz teması kurmak veya diğer sürücülerin yüz ifadelerini ve vücut dilini okumak gibi ince ipuçlarını kullanmaktadırlar. Otonom araçların da birbirleriyle bu bağlantıyı oluşturabilmeleri gerekmektedir [14].

1.5 OTONOM ARABALARIN FAYDALARI

Kolaylık ve yaşam kalitesi iyileştirmeleri için senaryolar sınırsızdır. Örneğin, yaşlılar ve bedensel engelliler bağımsız bir hayat yaşayabilmeli, çocuklarınız yaz kampındayken dış fırçalarını unuttuysa, araba onlara eksik eşyaları getirebilir. Köpeğinizi bir veteriner randevusuna bile gönderebilirsiniz.

Ancak otonom otomobillerin en önemli avantajı, CO₂ emisyonlarını önemli ölçüde düşürme potansiyelidir. Yakın tarihli bir çalışmada uzmanlar, aynı anda benimsendiği takdirde otonom otomobillerin tüm potansiyelini ortaya çıkaracak üç trend belirlemiştir. Bunlar araç otomasyonu, araç elektrifikasyonu ve araç paylaşımıdır. 2050 yılına kadar, bu devrimler kentsel ulaşımında aşağıdaki faydaları sağlayacağı öngörülmektedir [12].

- US DOT 2017 raporlarına göre trafik ölümlerinde % 90 azalma ve yılda kaza sebebiyle 30.000 ölüm sayısını azaltılır [9].
- Zararlı emisyonlarda %60 düşüş, daha az kaza, daha az trafik sıkışıklığı anlamına gelir. Bu da emisyonlarda düşüş anlamına gelir [9].
- Yakıt ekonomisinde %10 iyileşme sürücüsüz araçların yakıt ekonomisinde %4 ile %10 arasında azalma sağlayacağını belirtiyor.
- Dur-kalk durumlarını %100 ortadan kaldırılması hedeflenmiştir. Illinois Üniversitesi'nde yapılan araştırmaya göre, otonom arabalar dur-kalk davranışının oluşturduğu trafik durumlarını ortadan kaldırabilir. Bu da sadece insanlara zaman kazandırmakla kalmayacak, aynı zamanda arabalarının yolda kalma süresini de azaltacak ve dolayısıyla emisyonları azaltacaktır.
- Şerit kapasitesinde %500 artış beklenmektedir. Aynı raporu göre, şerit kapasitesinin şaşırtıcı bir şekilde %500 artabileceğini de tahmin ediyor [9].
- %40 seyahat zamanının azaltılması ve ABD'de otonom arabaların işçilere işe gidiş gelişlerde kaybedilen 80 milyar saati kurtarması bekleniyor. Bu da ekonomiye 1,3 trilyon dolar tasarruf sağlayacaktır [9].
- Ayrıca 5 milyar dolar tüketici tasarrufu beklenmektedir. Trafikteki azalmadan ve otonom arabanın bir park yeri bulma ve kendi başına park etme kabiliyetinden yararlanarak toplu taşımayı desteklenecektir. Ayrıca, otonom taksiler kullanıldığında taksi bekleme süresinde %88'lik bir düşüş, park için %15 daha az alan ihtiyacı ve yaşlanan bir nüfusta yaşlılar için gelişmiş hareketlilik öngörülmektedir [17].

2. OTONOM ARAÇLARDA KULLANILAN SENSÖRLER

Son yıllarda otonom araçlar alanındaki yapılan araştırmalar, sadece akademik çevrede değil, aynı zamanda endüstri genelinde de büyük ilerlemeler kaydetti [18]. Bunun başlıca nedeni, duyuşal bileşenlerin ve işlemcilerin maliyetlerinin düşürölmesinin yanı sıra mevcut artan hesaplama gücünün faydalarıdır [19].

Herhangi bir otonom arabanın temel gereksinimi, çevresini algılama ve gerçek zamanlı olarak akıllı kararlar verme yeteneğine sahip olmasıdır. Bu, belirli bir uygulama için kullanılabilecek sensör tipini, bu sensörün optimum ve güvenilir bir şekilde performans gösterecek bir sisteme uygulanma yöntemini ve diğer sensörlerle beraber çalışma yöntemlerini belirlemek gereklidir.

Otonom bir arabanın çevresi hakkında bilgi edinme yolu sensörler aracılığıylaıdır. Şekil 2.1'de otonom arabalarda kullanılan sensörler gösterilmiştir. Sensörler, aracın gerçek zamanlı olarak kararlar alabilmesi için çevrenin hem algısal hem de konumsal bir görünümünü oluşturabilmelidir. Bu, daha sonra bilgiyi kontrol algoritmalarına iletebilmesi için aracın nerede olduğına dair yüksek kaliteli bir anlayışı hızlı ve doğru bir şekilde üretmesi gerektiğı anlamına gelir.



Şekil 2-1: Otonom Araçlarda kullanılan sensörler

Otonom bir araçta kullanılan iki ana sensör tipi vardır. Çevreyi algılamak ve nesnelere olan mesafeyi hesaplamak için dış algılayıcı sensörler kullanılırken, sistem içinden değeri ölçmek için iç algılayıcı sensörler kullanılır [15].

2.1 LİDAR

Lidar (Light Detection and Ranging; veya Laser Imaging Detection and Ranging), mesafeleri ölçmek için kullanılan bir uzaktan algılama teknolojisidir. Aydınlatma için kendi enerji kaynaklarını yayan aktif sensörler kullanır. Darbeli bir ışık lazeri gönderip ve darbenin geri yansıması için geçen süreyi ölçerek uçuş süresi ilkesine göre çalışır. Bu ölçümler daha sonra çevredeki ortamın üç boyutlu temsillerini oluşturmak için kullanılabilir. Günümüzün Lidar sensörleri, 150 kilohertz'den (saniyede 150.000 darbe) daha yüksek hızlarda mesafeleri ölçebilir ve 250 metrenin üzerinde bir menzile sahip uzun menzilli bir sensör olarak sınıflandırılabilir [21].

Lidar sensörleri, yüksek hassasiyet ve doğrulukları ile otonom araç teknolojisi için birçok avantaj sunar. Nokta bulutları adı verilen yüksek çözünürlüklü yoğun aralıklı yükseklik noktaları ağı üretebilirler. Bu nokta bulutu verileri, Eşzamanlı Yerelleştirme ve Haritalama (SLAM Simultaneous Localization & Mapping) gibi doğru yerelleştirme ve haritalama teknikleri için gereklidir [21].

Bu sensörlerin tam potansiyeli, yüksek maliyetleri ve düşük kullanılabilirlikleri nedeniyle henüz tam olarak keşfedilmemiştir. Tam 360° görüş sunan ve çok pahalı maliyetlere mal olabilen karmaşık mekanik ayna sistemleridir. Bugün, büyük ölçekli dağıtım için daha uygun olan daha düşük maliyetli Lidar sensörlerinin geliştirilmesine doğru bir yönelim bulunmaktadır. Bu gelişmeler özellikle, katı hal Lidar ve kızılötesi Lidar teknolojilerinden oluşmaktadır.

2.2 RADAR

Radar (Radio Detection And Ranging) radyo ile tespit etme ve menzil tayini olarak tanımlanmaktadır. Nesnelerin mesafesini, açısını ve hızını ölçmek için radyo dalgalarını kullanan bir teknolojidir. Çoklu frekans bantlarında kullanılabilen elektromanyetik radyasyon prensibi ile çalışır. Daha yüksek frekans, radar sensör sisteminin gerçek zamanlı olarak birden fazla nesneyi ayırt etmesine izin veren daha yüksek çözünürlük sağlar. Radar sensörleri kısa ila orta menzilli (50 – 100 m) olarak sınıflandırılır. Ancak bazı radar sensörleri 150 m'den daha uzak bir mesafedeki bir nesneyi tespit edebilir [22].

Radar sensörleri, her türlü çevre koşulunda çalışmaları nedeniyle otonom araçlara birçok fayda sağlar. Lidar sensörlerinin aksine, radar önemli ölçüde daha ucuzdur ve çok daha kolay

temin edilebilir. Radar sensörleri, gelişmiş sürücü destek sistemleri anlamına gelen ADAS sistemleri içeren modern karayolu araçlarında yaygın olarak kullanılmaktadır. Bu sistemler, hız sabitleme fonksiyonlarının yanı sıra çarpışma tespiti sağlamak için tasarlanmıştır. Bu sensörlerin bir diğer önemli özelliği de algılanan nesnelerin bağlı hareketini belirleyebilmeleridir [23].

2.3 DİJİTAL KAMERALAR

Bir kamera, kapalı bir bölgenin dijital görüntüsünü üretmek için pasif ışık sensörleri ilkesiyle çalışır. Kameralar, çevrelerindeki hem hareketli hem de statik nesneleri algılayabilir. Kameraların diğer sensör türlerine göre en büyük avantajı, renkleri ve dokuları görme yeteneklerine sahip olmalarıdır. Bu, teknolojinin aracın yol işaretlerini, trafik ışıklarını, şerit işaretlerini vb. tanımlamasına izin verdiği için, otonom bir aracın algılama sistemini artırmak için büyük bir faydadır. Kameralar ayrıca belirli bir nesneye olan mesafeyi belirleyebilir, ancak bu oldukça karmaşık işleme algoritmaları gerektirir [24].

Bu teknolojinin bir diğer önemli faydası, düşük maliyetli olmaları ve yüksek kullanılabilirlikleridir. Verileri analiz etmek için gereken işlem gücü pahalı olabilir. Ancak yine de Lidar sistemlerinden daha ucuz bir seçenektir. Kameraların ana dezavantajları, düşük yoğunluklu ışığa duyarlı olmaları ve olumsuz hava koşullarından da büyük ölçüde etkilenebilmeleridir. Çoğu otonom araç üreticisi, kameraların tamamen otonom navigasyon için temel bir teknoloji olduğunu, ancak yalnızca sağladıkları verileri Lidar veya Radar sistemlerinin benzerleriyle birleştirerek maksimum potansiyellerinde kullanılabileceği önerme görüşündedirler [25].

2.4 ULTRASONİK ALGILAYICILAR

Ultrasonik sensör, bir nesneye olan mesafeyi ölçmek için ses dalgalarını kullanan bir cihazdır. Belirli bir frekansta bir cisme doğru bir ses dalgası yayılır ve dalga'nın geri dönmesi için geçen süre, mesafeyi hesaplamak için kullanılır. Ultrasonik sensörlerin ana avantajları, genel olarak tartışılan tüm sensör tiplerinin en ucuzu olmalarıdır. Olumsuz hava koşullarında oldukça sağlamdırlar ve ayrıca çoğu otomobil üreticisi tarafından yıllardır park sensörü olarak kullanıldığı kanıtlanmış bir güvenilirlik geçmişine sahiptirler [26].

Ultrasonik sensörler ayrıca yakınlık uygulamaları için en doğru sensör olarak kabul edilir. Ancak bu sensörlerin dezavantajları, ses dalgalarındaki bozulmalardan büyük ölçüde etkilenebilmeleridir. Ses sadece bir ortamda yayılabileceğinden, sıcaklık ve nem gibi çevresel koşullardaki değişiklikler sensörün özelliklerini büyük ölçüde etkileyebilir. Buna uyum sağlamak için çoğu sensör, mevcut sıcaklığa bağlı olarak okumaları ayarlamak için bir algoritma kullanır.

2.5 KÜRESEL KONUMLAMA SİSTEMİ

Küresel konumlama sistemi (KKS-Global position system-GPS) anlamına gelen, dört veya daha fazla GPS uydusuna engelsiz bir görüş hattı olduğu sürece, dünyanın herhangi bir yerindeki bir GPS alıcısına coğrafi konum ve zaman bilgisi sağlayan uydu tabanlı bir radyo navigasyon sistemidir. Bir KKS alıcısı, yörüngedeki uydulardan gönderilen sinyalleri bir 'trilaterasyon' yöntemi kullanarak zamanlayarak konumunu hesaplayabilir. Trilaterasyon, dairelerin, üçgenlerin veya kürelerin geometrisini kullanarak mesafeleri ölçerek noktaların mutlak veya göreceli konumlarını belirleme işlemidir. KKS, kritik konum yetenekleri sağlayabildiği için otonom araçlarda navigasyon ve konum belirleme için yaygın olarak kullanılan bir sensördür. Çoğu el tipi KKS alıcısı, yaklaşık 3 m hassasiyete ulaşabilen tek frekanslı KKS sinyallerini kullanırken, ticari sınıf alıcılar yaklaşık bir metre hassasiyete ulaşabilir. Tipik olarak çift frekanslı olan anket dereceli KKS üniteleri, birkaç santimetre içinde doğruluklara sahiptir, ancak bu üniteler yüksek maliyete mal olabilir. Otonom navigasyon için KKS teknolojisini kullanmanın ana dezavantajı, konumlandırma doğruluğunu azaltabilecek birçok faktörün olmasıdır. KKS alıcıları, uydularla doğrudan bir görüş hattı gerektirir, ancak bu sinyaller genellikle yüksek binalar, ağaçlar, tüneller vb. nedeniyle engellenebilir. Kapalı veya yeraltı ortamlarında navigasyon, KKS sistemlerine güvenemeyecek diğer önemli faktörlerdir [27].

2.6 ATALET ÖLÇÜM BİRİMİ

Atalet Ölçüm Birimi (AÖB-IMU-Inertia Measurment Unit), bir nesnenin kuvvetini, açısal hızını ve manyetik alanını ölçebilen elektronik bir cihazdır. AÖB cihazları genellikle ortogonal X, Y ve Z eksenlerinin her biri için bir tane olmak üzere üç ivmeölçer, jiroskop ve manyetometreden oluşur. AÖB otonom araçların kontrolü ve yönlendirilmesi için yaygın olarak kullanılan bir

cihazdır. Genellikle küresel bir referans çerçevesine göre doğrusal hız, konum ve açısal konumları hesaplamak için ham AÖB verilerini işleyen Ataletsel Navigasyon Sistemlerine uygulanırlar. Bu cihazların ana dezavantajları, aracın gerçek konumu hakkında bilgi sağlayamamaları, bu nedenle bunun KKS gibi başka bir kaynak tarafından belirlenmesi gerekir. AÖB'lerin navigasyonla ilgili diğer bir dezavantajı, sürüklenmeye yol açabilecek birikmiş hatalardan muzdarip olmalarıdır. Bu soruna tipik bir çözüm, KKS gibi sapma hatalarını sürekli olarak düzeltebilen bir konumsal izleme sistemi uygulamaktır [28].

2.7 ENKODERLER

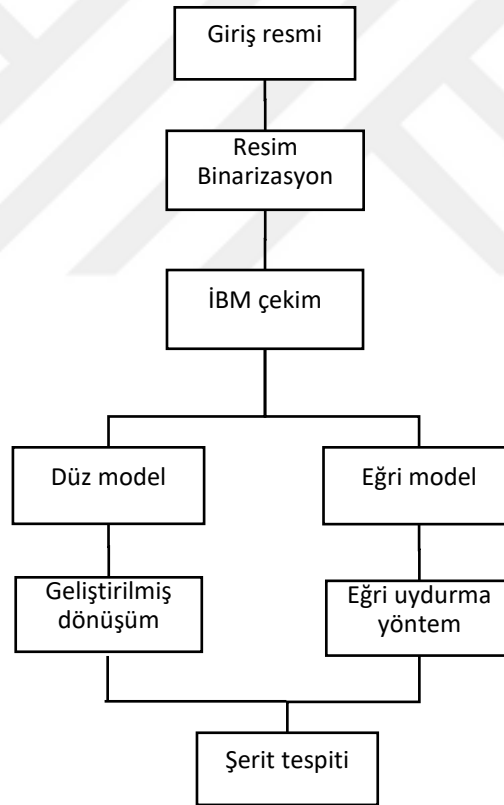
Enkoderler, bir şaftın lineer veya açısal pozisyonunu analog veya dijital sinyale çevirerek onu lineer/açısal trans düşer yapan elektromekanik cihazlardır. Bir kontrol sistemi içinde konum, yön ve hız hakkında bilgi sağlayabilir. İki ana tip döner kodlayıcı vardır. Artımlı kollayıcılar, devir başına belirli sayıda darbe üretirken, bir mutlak kodlayıcı, ayrık konuma karşılık gelen bir bit akışı üretir. Araçta kilometre sayacı verilerini sağlamak için kodlayıcılar kullanılır. Genellikle aracın tekerleklerine bir kodlayıcı takarak ve bilinen bir başlangıç noktasından kat edilen mesafeyi ölçerek araçların göreceli konumunu belirlemek için kullanılırlar. Bu teknik, araçların konumunu gerçek zamanlı olarak tahmin etmek için uygun maliyetli ve uygulanması kolaydır. Ancak tekerlekler ve zemin arasındaki kayma nedeniyle sıklıkla ölçüm hataları meydana gelebilir. Bu kilometre sayacı verilerini (GPS-Global Positioning System) ve Ataletsel Navigasyon Sistemi (INS-Inertial Navigation System) gibi diğer sistemlerle birleştirerek daha doğru bir konum tahmini sağlayabilen birçok yerelleştirme sistemi örneği vardır [29].

3. ŞERİT ALGILAMA

Bu çalışmada şerit tespiti, düz bir eğri modeli algoritması kullanılarak kurulur. İlk başta, ilgili bölge metodu (İBM-Region of interest-ROI), giriş görüntüsünden çıkarılır. Düz bölge, Hough dönüşümü kullanılarak elde edilir ve eğri bölge, en küçük kare eğri uydurma yöntemi ile elde edilir. Eğri modeli, düz ve eğri model arasındaki ilişki ile oluşturulur.

3.1 ŞERİT ALGILAMA METODLARI

Şerit algılama metodları Şekil 3-1’de detaylı olarak verilmiştir. Bu kısımlar aşağıda açıklanmıştır.



Şekil 3-1: Düz eğri modeli kullanarak şerit algılama

3.1.1 İlgili Bölge Metodu

İlgili bölge metodu (İBM-Region of interest-ROI), diğerlerinden daha az önemli olduğu düşünülen (ör. arka plan, İBM olmayan) belirli görüntü alanlarının (ör. ön plan, İBM) kodlama bit hızını azaltmak için yaygın bir çözüm olarak kullanılmaktadır. Giriş görüntüsü grileştirildi ve ikili hale getirilir. Giriş yolu görüntüsünün her satırı toplanır. Piksel birikiminin fark değeri hesaplanır. Bitişik 2 sıra ve bitişik 2 satır, mevcut satır ile önceki satırın hesaplaması arasında çıkarma yapılarak elde edilir. Son olarak, fark değerlerine harici arama uygulanır ve yatay çizgi konumu belirlenir. Daha sonra İBM çıkarılmaktadır [2].

3.1.2 Düz ve Eğri Bölgenin Ayrılması

Düz ve eğri bölgenin ayrılması için 2 yaklaşım yapılır. Düz çizgi, şerit tespiti için yakın alanla eşleşir. Algılama sisteminin uzak alanı olarak kabul edilen eğri çizgidir. Yani yakın görüş alanındaki şerit düz çizgi olarak kabul edilip, uzak bölgedeki şerit eğri bölge olarak kabul edilmiştir. Düz ve eğri modelinin lineer yapısı, düz ve eğri yüzeylerin karasal hat sistemi tarafından başlatılır [30].

3.1.3 Düz Model Çözümleme

Bölgenin yakın alanındaki şeritler düz çizgi olarak kabul edilir ve düz çizginin matematiksel denklemi denklem 1’de ifade edilmiştir [31].

$$Y = kx + 1 \quad (1)$$

Burada, Y= Tahmini hedef değer, K= eğitim değer, x= verilen değer. Düz model çözümünde denklem 1’in çözülmesi gerekir. Denklem 2’de formülü verilen Hough dönüşümü, düz model çözümünde daha büyük öneme sahiptir. Hough dönüşümünün genel temsili parametrik denklem 2’de ifade edilir.

$$P = x \cos(a) + y \sin(a) \quad (2)$$

Şerit, aracın her iki tarafında bulunur. Hough dönüşümü, doğrusal model denklemleriyle bağlantılı olan sağ şerit ve sol şeridin eğim değerlerini üretmektedir.

$$K = -1/\tan(h) \quad (3)$$

$$b = q/\sin(h) \quad (4)$$

Denklem 3 ve denklem 4'deki formüller elde edilip, lineer model denkleminin parametreleri belirlenebilir [31].

3.1.4 Eğri modeli çözümleme

Eğri modelinin çözümünde en küçük kareler için eğri uydurma yöntemi kullanılır. Bu yöntemin ana amacı, veri noktasını karşılamak için gerçek ve uydurma değer hatasını en aza indirmektir. Buna bağlı olarak, eğri modelinin denklem parametreleri çözülebilir [30].

3.1.5 Düz Eğri Modelinin Değiştirilmesi

Araç viraja ulaşmadan önce sürücü uyarılır. Algılanan düz çizgi ile ikinci şerit çizgisi arasındaki farktan hesaplanan artık hata değeri çözülerek eşik belirlenir. Hata değeri eşikten büyük olduğunda, sürüş modu otomatik olarak düz çizgiye geçer. Düz ve eğri noktanın kaybolması, bükülme yönünü ve açısını bulmak için kullanılır [31].

3.1.6 Viraj Büküm Yönünün Belirlenmesi

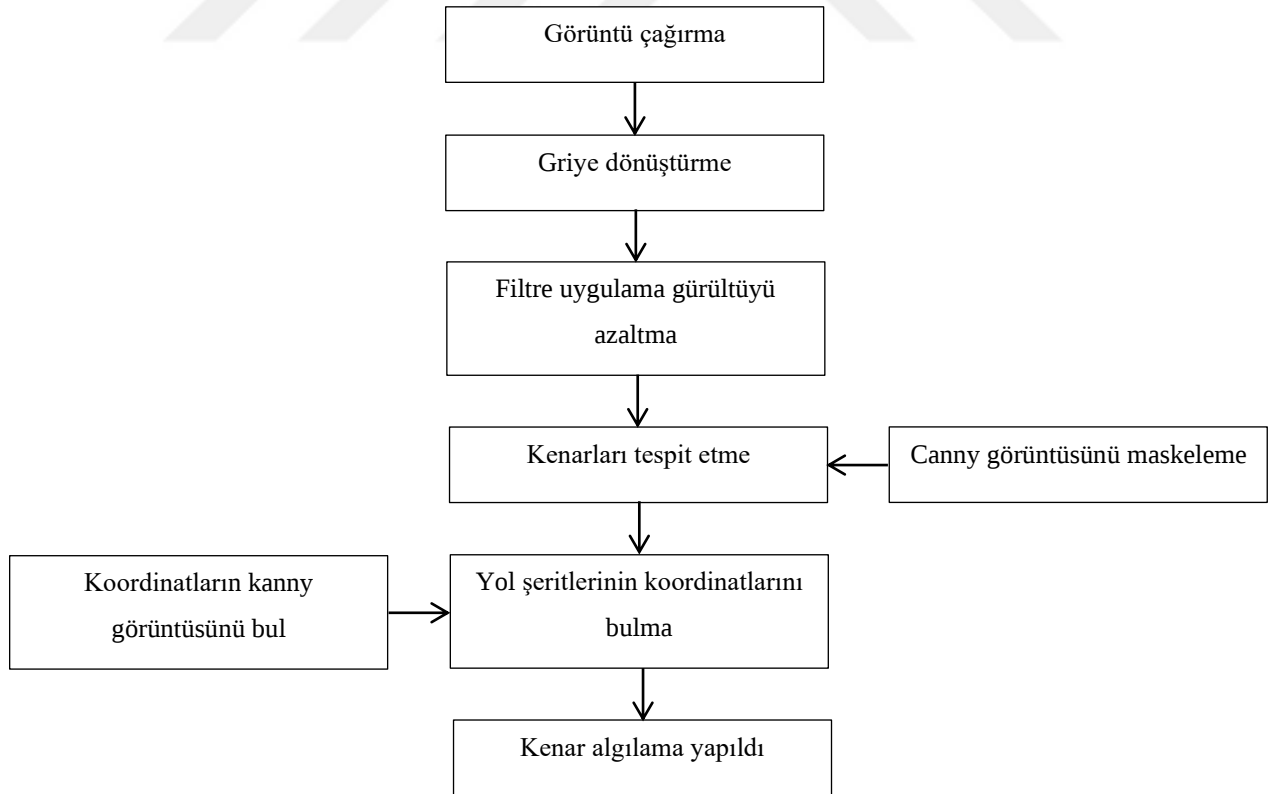
Eğrinin ufuk noktasını ve düz kesişimin göreceli konumunu belirleyerek eğrinin bükülme yönünü tahmin edebiliriz. 4 adımda gerçekleştirilebilir [31].

- 1) Hough dönüşümü lineer denklem ve parametreleri elde eder.
- 2) Hesaplanan sol ve sağ düz çizgi şeritlerinin ortalama konumu belirlenir.

- 3) Kaybolan eğri noktası, düz çizginin buluşma noktası ve eğrinin yönü tahmin edilir. Sola viraj bükümü ve sağa viraj bükümü, düz çizginin ikiye bölünmüş noktası kullanılarak hesaplanır.
- 4) Bükülme yönü, bir görüntünün kesişim ve bükülme noktasının piksel değeri numaralandırılarak tanımlanır.

3.2 ÖNERİLEN YÖNTEM

Burada önerilen şerit tespit tekniği, bir arabanın önüne monte edilmiş bir kamera kullanılarak yapısal bir yol üzerindeki şerit çizgilerini tespit etmek için bilgisayarlı görme tekniklerini kullanılmıştır. Bu teknik, algılama süreci boyunca Açık Kaynak Bilgisayar Görme Kütüphanesi'ni (OpenCV) kullanır. Önerilen yöntemin genel prosedürü, blok diyagram Şekil 3-2'de gösterilmektedir. Sobel filtreleme ve HLS filtreleme yöntemleri kullanılmaktadır. Yoldaki şeritleri ve virajları göstermek için sürgülü pencere araması kullanılmıştır [21].



Şekil 3-2: Bilgisayarla görme algoritmasına dayalı önerilen şerit tespit tekniği.

3.2.1 Görüntü Yakalama

Giriş verileri hareket halindeki bir araçtan alınan renkli bir görüntü dizisidir. Aracın içine orta hat boyunca ön dikiz aynasında renkli bir kamera monte edilmiştir. Yol şeritleri, yoldaki araçlar, yol kenarı ve bazen de yolda bulunmakta olan nesneleri de dâhil olmak üzere aracın önündeki ortamın görüntülerini alır. Görüntü yakalama kartına sahip yerleşik bilgisayar, görüntüleri gerçek zamanlı olarak (30 kare/saniyeye kadar) yakalayıp ve bunları bilgisayar belleğine kaydetmektedir. Şerit tespit sistemi hafızadan görüntü dizisini okur ve işlemeye başlar. Şeritlerin iyi tahminlerini elde etmek ve algoritmanın hızını iyileştirmek için orijinal görüntü boyutu Gauss piramidi ile 620x480 piksele düşürülmüştür.

Dijital kameradan çekilen giriş videosu bazıları bozulmalara maruz kalabilir. Bu bozulmalara lenslerin optik tasarımı ve ayrıca nesneye göre kamera konumu veya nesnenin görüntü çerçevesi içindeki konumu neden olur. Bu bozulmalar, şerit algılama işleminin doğruluğunu etkileyebilir. Görüntüdeki şerit çizgisini doğru bir şekilde tespit etmek için bu bozulmaların üstesinden gelmemiz gerekiyordu. Kamera, kontrol panosu deseninin görüntülerini yakalamak için kalibre edilmiştir. Resim içindeki iç köşeler ve beklenen ölçümden bir sapma hesaplanmıştır. Bu bilgiler kullanılarak bozulma matrisi ve bozulma katsayıları hesaplanarak bozulmamış kalibrasyon görüntüsü verilmiştir.

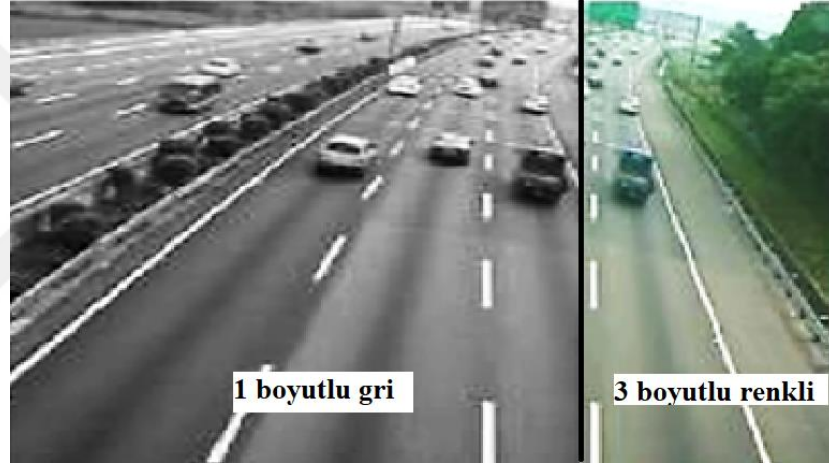
3.2.2 Kırpılmış Resim

Kırpma, bir görüntünün istenmeyen veya ilgisiz kısımlarını kaldırma işlemi olarak tanımlanmaktadır. Ayrıca kırpma işlemi temel görüntüye görsel olarak odaklanmayı sağlar. Şerit çizgilerinin belirlenmesinde sadece şerit çizgilerini görmemiz beklenen alanlara odaklanmalıyız. Bu, yalnızca belirli bölgelerde daha fazla görüntü işleme gerçekleştirerek genel işlem hattını daha hızlı hale getirir. Şerit çizgileri bölgesine odaklanan kırpılmış görüntüyü göstermektedir [32].

3.2.3 Gri Ölçeğe Dönüştürme

Bu işlem, bitişik piksellerde yoğunluktaki şekil değişikliklerini belirleme işlemidir. Matrisle ifade edilecek olursa, Şekil 3-3’de belirtilen matrise göre herhangi bir resmi ele alıp o resmin çözümüne göre filtreler (Sobel filtre, Schass filtreler) konvülasyon (*) işlemleri yapılır ve kenar algılama yapılır.

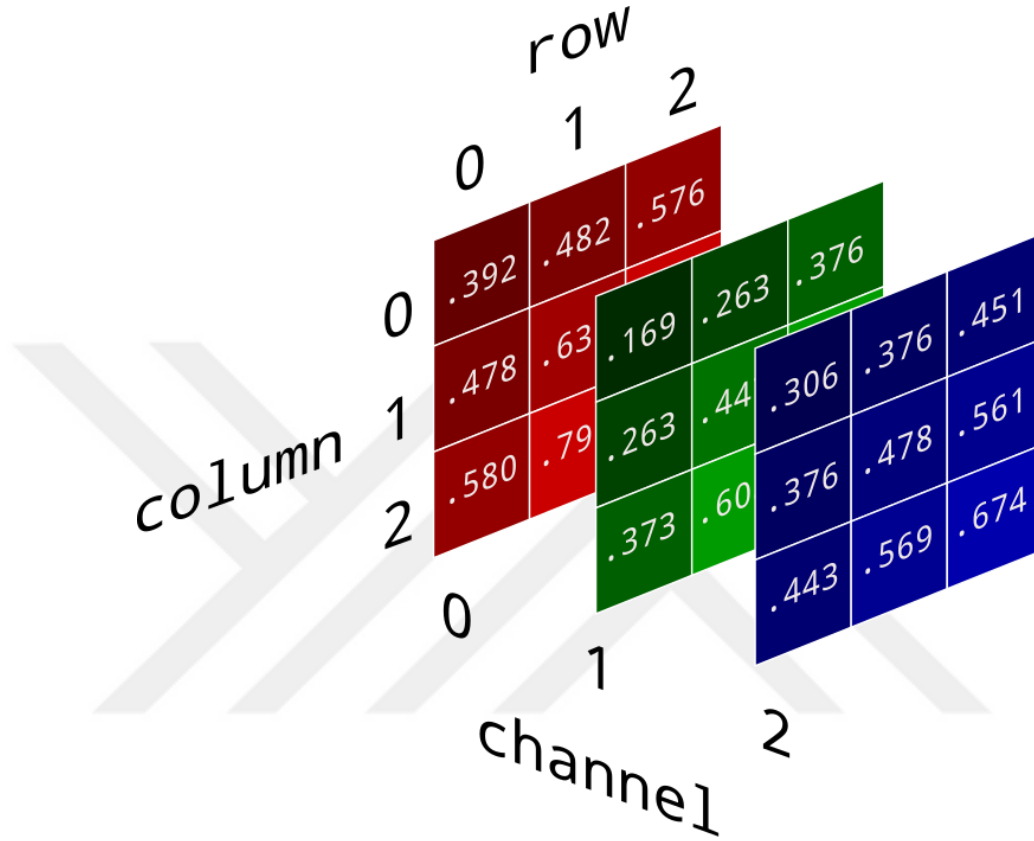
$$A = \begin{bmatrix} 0 & 0 & 255 & 255 \\ 0 & 0 & 255 & 255 \\ 0 & 0 & 255 & 255 \end{bmatrix} \quad (5)$$



Şekil 3-3: 3 boyutlu renkli resimden 1 boyutlu Gri resme çevrilmesi

Renk bilgisini korumak ve renk bilgisini kullanarak yolu şerit sınırlarından ayırmak zor olmaktadır. Bu işlem, kenar algılamada zorluklar oluşturup, işlem süresini etkileyecektir. Uygulamada yol yüzeyi; gölgeler, farklı kaplama stili veya yağmur nedeniyle birçok farklı renkten oluşabilir. Bu da yol yüzeyinin renginin ve şerit işaretlerinin bir görüntü bölgesinden diğerine değişmesine neden olur. Bu nedenle, Şekil 3-3’de görüldüğü gibi renkli görüntü gri tonlamaya dönüştürülür. Bununla birlikte, gri tonlamalı görüntülerin işlenmesi, renkli bir görüntüye kıyasla daha doğru sonuçlar vermektedir. Bu işlev, piksel değerinin kırmızı bileşeninin ağırlıklı bir toplamını oluşturarak, 24 bit, üç kanallı, renkli bir görüntüyü 8 bit, tek kanallı gri tonlamalı bir görüntüye dönüştürür.

Şekil 3-4’de (Piksel değerinin Yeşil bileşeni * 0.3) + (Piksel değerinin Yeşil bileşeni * 0,59) + (Piksel değeri için mavi bileşen * 0.11) formülü ile, karşılık gelen piksel için gri skala değeri oluşturulur.



Şekil 3-4: 3 boyutlu resim normalizasyonu

3.2.4 Gürültü Azaltma

Görüntü ve sinyal işlemenin temel sorunu, dijital bir renkli görüntüden gelen paraziti etkili bir şekilde azaltırken özelliklerini (kenarlar, renk bileşeni mesafeleri, vb.) korumaktır. Üç ana gürültü türü vardır. Bunlar dürtü gürültüsü, toplamsal gürültü ve çarpımsal gürültüdür. Dürtü gürültüsü, genellikle görüntü piksellerinin bir kısmının bozularak kalan pikselleri değiştirmeden bırakmasıyla karakterize edilir. Darbe gürültüsü örnekleri, sabit değerli darbe gürültüsü ve rastgele değerli darbe gürültüsüdür. Her görüntü pikseline belirli bir dağılımdan bir değer eklendiğinde, örneğin bir Gauss dağılımı gibi, toplamsal gürültüden bahsedilir. Gürültünün yoğunluğu sinyal

yoğunluğuna göre değiştiğinden (örneğin, benek gürültüsü) görüntülerden çarpımsal gürültünün çıkarılması genellikle ek gürültüden daha zordur [31].

Gürültü azaltma için birkaç bulanık filtre hâlihazırda geliştirilmiştir. Örneğin yinelemeli bulanık kontrol tabanlı filtreler [33], GOA filtresi [34] gibi çözümler geliştirilmiştir. Bu son teknoloji yöntemlerin çoğu, esas olarak, darbe gürültüsü gibi gürültünün azaltılması için geliştirilmiştir. Bu bulanık filtreler, sıralı sıralama tekniklerinden (medyan tabanlı filtreler gibi) daha iyi performans gösterebilir. Bununla birlikte, mevcut bulanık tekniklerin çoğu, toplamsal gürültü için ikna edici sonuçlar vermemektedir [34]. Mevcut yöntemlerin bir diğer dezavantajı, bu filtrelerin çoğunun özellikle gri tonlamalı görüntüler için geliştirilmiş olmasıdır. Elbette bu filtreleri diğer bileşenlerden bağımsız olarak her bir renk bileşenine ayrı ayrı uygulayarak renkli görüntülere genişletmek mümkündür. Ancak bu, özellikle kenar veya doku öğelerinde birçok yapaylık ortaya çıkarır.

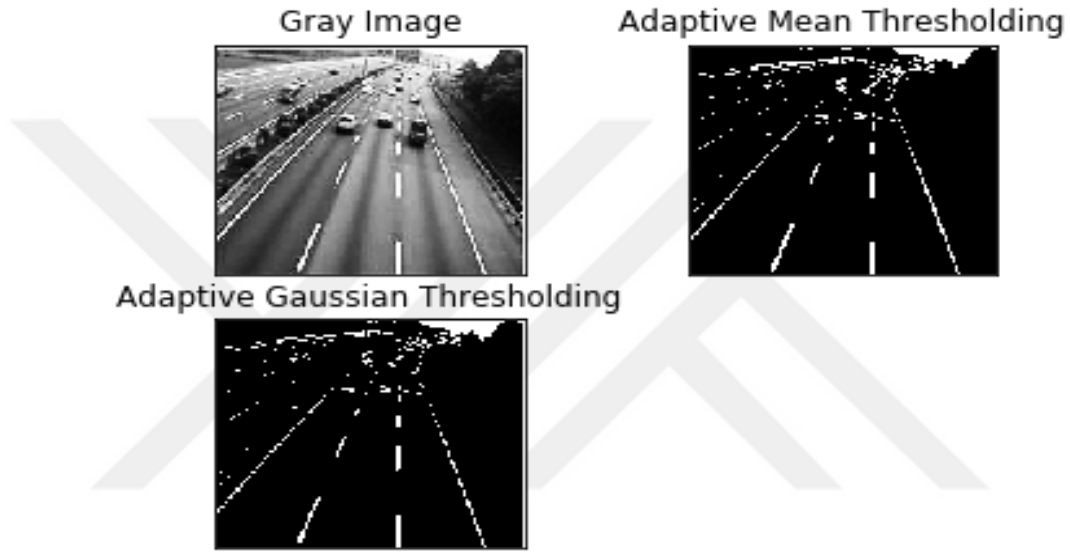
3.2.5 Kenar Algılama

Şerit sınırları, yol yüzeyi ile boyalı çizgiler veya bir tür kaldırım dışı yüzey arasındaki keskin kontrast ile tanımlanır. Bu keskin kontrastlar görüntüdeki kenarlardır. Bu nedenle kenar dedektörleri şerit sınırlarının yerini belirlemede çok önemlidir. Ayrıca, bir yolun ana hatları görüntüden çıkarılabiliyorsa, görüntüyü önemli ölçüde basitleştirerek gereken öğrenme verisi miktarını da azaltır. Bu algoritma için uygulanan ve değerlendirilen tüm kenar algılayıcılardan en iyi kenar görüntülerini üreten kenar algılayıcı, 'Canny' kenar algılayıcıdır [14]. Kenar yönüne dik doğrultuda görüntü fonksiyonunun kısmi türevinin maksimumunu bularak ve sinyali kenar yönü boyunca azaltarak Canny'nin operatörü maksimuma ulaşır.

$$G_{\sigma} = \frac{1}{\sqrt{2\pi\sigma^2}} \exp \left[-\frac{x^2+y^2}{2\sigma^2} \right] \quad (6)$$

$$\mathbf{n} = \frac{\nabla G * I}{|\nabla G * I|} \quad (7)$$

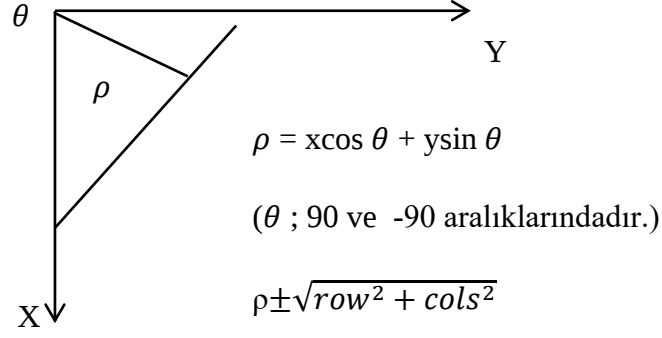
Burada denklem 6 ve denklem 7'deki eşitlikleri otomatik olarak seçebilen Şekil 3-5'de görüldüğü gibi kenar algılama algoritmasına sahip olmak önemlidir. Ancak varsayılan Canny'de kullanılan otomatik eşik çok fazla kenar bilgisi üretmiştir. Canny tarafından tespit edilen kenarda hafif bir değişiklik yapılması daha arzu edilen sonuçlar vermiştir. Gerekli olan tek değişiklik, kenar olmayan piksellerin miktarını, farklı görüntü yakalama ortamı koşullarında daha doğru kenarlar veren en iyi değere ayarlamaktır. Keskin kenar detektörü, diğer yaklaşımlar gibi gürültü üretmemesi açısından da çok arzu edilen bir özelliğe sahiptir [31].



Şekil 3-5: Kenar algılama

3.2.6 Hat Algılama

Kullanılan hat detektörü, kısıtlı bir arama alanına sahip standart bir Hough dönüşümüdür. Standart Hough dönüşümleri, Şekil 3-6'da gösterilen denklemi kullanarak doğruları aramaktadır.

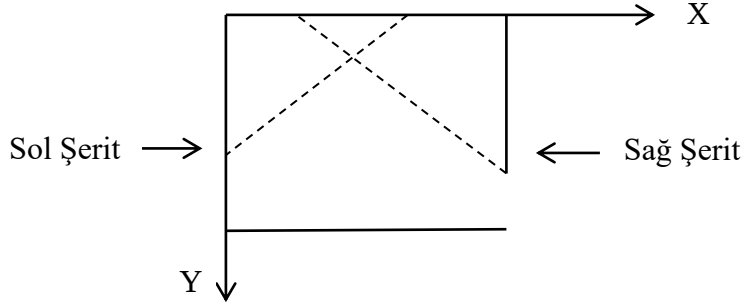


Şekil 3-6: Hough Dönüşümü

Gerçekte, belirli bir bölgenin dışında kalan herhangi bir çizgiyi yok sayabiliriz [35]. Örneğin, yatay bir çizgi muhtemelen şerit sınırı değildir ve yok sayılabilir. Kısıtlanmış Hough dönüşümü, arama alanını her bir taraf için 45° ile sınırlamak üzere değiştirilmiştir. Ayrıca giriş görüntüsü, görüntünün sağ ve sol tarafını verecek şekilde ikiye bölünür. Sağ ve sol tarafların her biri ayrı ayrı aranır ve 45° pencereye denk gelen yarım görüntüdeki en baskın çizgi döndürülür. Ufuk, sol ve sağ Hough çizgileri kullanılarak ve bunların kesişimlerine yansıtılarak basitçe hesaplanır. Bu kesişme noktasındaki yatay çizgiye ufuk çizgisi denir.

3.2.7 Şerit Sınır Taraması

Şerit sınırı tarama aşaması, giriş olarak Hough çizgilerini ve ufuk çizgisini kenar görüntüsünü kullanır. Kenar görüntüsü, taranan şeydir ve kenarlar, topladığı veri noktalarıdır. Tarama, yansıtılan Hough çizgilerinin görüntünün altındaki görüntü sınırıyla kesiştiği yerde başlar. Bu kavşak bulunduğunda, hangi kavşağın bulunduğuyla ilgili olarak sol veya sağ arama için başlangıç noktası olarak kabul edilir. Başlangıç noktasından, arama, şeridin merkezine doğru belirli sayıda pikselle başlar ve ardından maksimum aralıktan sonra belirli bir piksel sayısına ulaşana kadar ilk kenar pikselini aramaya devam eder. Aralık, önceden yerleştirilmiş kenar pikseli artı bir arabellek konumuna ayarlanacaktır [36].

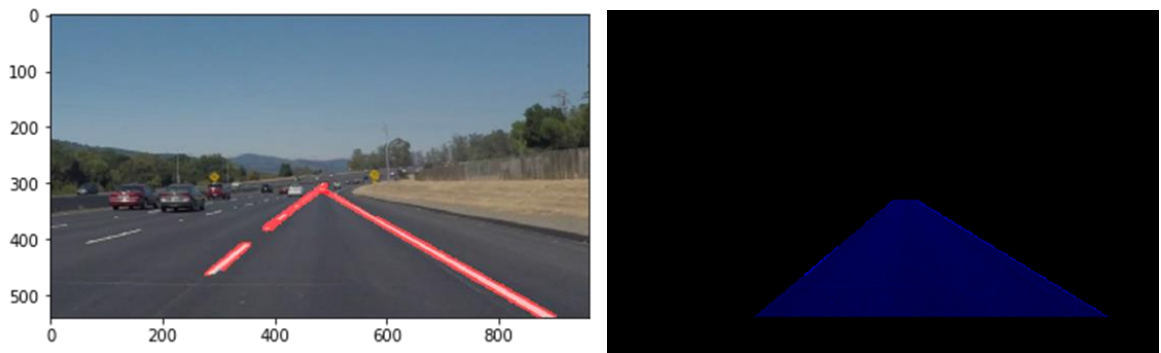


Şekil 3-7: Görüntü koordinatları

$$L^{(l)} = \{(u_1^{(l)}, v_1^{(l)}), (u_2^{(l)}, v_2^{(l)}), \dots, (u_m^{(l)}, v_m^{(l)})\} \quad (8)$$

$$L^{(r)} = \{(u_1^{(r)}, v_1^{(r)}), (u_2^{(r)}, v_2^{(r)}), \dots, (u_m^{(r)}, v_m^{(r)})\} \quad (9)$$

Başlangıç koşulu için, arama, Şekil 3-7’de gösterildiği gibi, zaten en sol veya en sağ sınırdadır. Bu nedenle, maksimum aralık sınırın kendisi olacaktır. Aramanın ekstra tampon bölgesi, şerit sınırlarının dışı doğru eğrilerinin izlenmesini kolaylaştırmaya yardımcı olacaktır. Şerit noktaları, $L(l)$ ve $L(r)$ olarak ifade edilen iki liste halinde düzenlenmiştir. Şerit belirleme işleminin örnek görüntü üzerinde uygulanması Şekil 3-8’de verilmiştir.



Şekil 3-8: Şerit çizgi belirtme

3.2.8 Hiperbol Uydurma

Hiperbol çifti uydurma aşaması, girdi olarak şerit taramasından alınan iki veri noktası vektörünü kullanır. Verilere bir hiperbol sığdırmak için en küçük kareler tekniği kullanılır. Veri noktalarının vektörlerinin her birine bir hiperbol uygundur. Ancak çift model oldukları için aynı anda çözülürler. İki hiperbolün parametreleri ilişkilidir. Çünkü Şekil 3-8'de gösterildiği gibi yolun geometrisi nedeniyle aynı noktada birleşmeleri gerekir [25]. Görüntü düzleminde yol sınır noktası (u, v) verildiğinde şerit sınırını bir hiperbol olarak ifade etme formülü denklem 10'da verilmiştir.

$$U = \frac{k}{u-h} + b(v-h) + c \quad (10)$$

u ve v, görüntü referans çerçevesindeki x ve y koordinatlarıdır. h; görüntü referans çerçevesindeki ufkun Y koordinatıdır. k, b ve c eğrinin parametreleridir.

3.2.9 Yolda Şerit Çizgisi Bulma

Şekil 3-9'da görüldüğü gibi bir şerit hattı için birden fazla hat tespit edilmiştir. Bunun için ortalama bir çizgi bulunması gerekmektedir. Ayrıca, bazı şerit çizgileri yalnızca tanınmaktadır. Tüm şerit çizgisi uzunluğunu kapsayacak şekilde çizginin tahmin edilmesi gerekmektedir.

İki şerit çizgisi biri sol, diğeri sağ için bulunur. Sol şerit pozitif bir eğime, sağ şerit ise negatif bir eğime sahip olmalıdır. Bu nedenle, pozitif eğim çizgilerini ve negatif eğim çizgilerini ayrı ayrı toplayıp ortalamaları alınarak ve bir üçgen çizilir [37].



Şekil 3-9: Görüntü üzerinde uygulanması

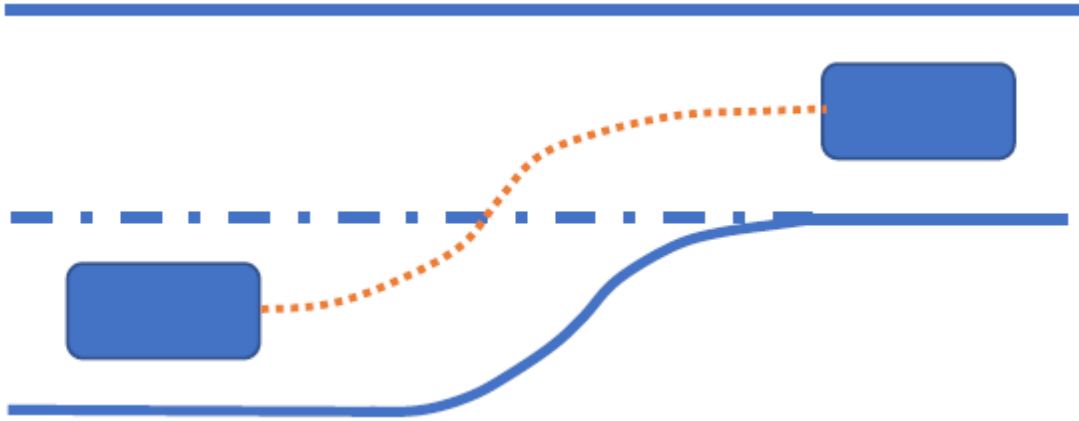
Yapılan çalışma, video görüntülerinin şerit çizgilerinin doğru bir şekilde algılandığını ve çizgilerin çok düzgün bir şekilde işlendiğini açıkça göstermesi açısından başarılı olmuştur. Yalnızca düz şerit çizgileri algılanır. Eğri şeritleri ele almak gelişmiş bir konudur. Düz çizgilere sığdırmak yerine perspektif dönüşümü ve ayrıca poli uydurma şerit çizgilerini kullanılması gerekmektedir. Bunu söyledikten sonra, arabanın yakınındaki şeritlerin çoğu görüntülerde düzdür. Eğrilik, dik bir eğri olmadığı sürece daha uzakta görünür. Dolayısıyla, bu temel şerit bulma tekniği hala çok kullanışlıdır [37].

3.3 ŞERİT DEĞİŞTİRME KARARI

Otonom araçların başarısı için yasal, güvenlik ve teknik açıdan, otomasyon sistemlerine kullanıcı kabulü ve güveninin önemli bir etkisi olduğu düşünülmektedir. Sürüş tarzı bu faktörleri kesin olarak etkileyebilir. Sürüş dinamiklerinin yanı sıra, hız ve ivmeler, taktik kararlar (örneğin şerit için) gibi değişiklikler ayrıca otoyollardaki sürüş stilini de tanımlar. Katılımcıların otomatik sürüş sırasında yolcular tarafından beklenen şerit değiştirme davranışının daha iyi anlaşılması ve şerit değişikliklerini gerçekleştirmek için zaman içinde istenen nokta başlatılmıştır. Daha sonra, bir lojistik regresyon analizi, bir şeridin olasılığını belirleyebilir [25].

3.3.1 Şerit Değiştirme Stratejisi Tasarımı

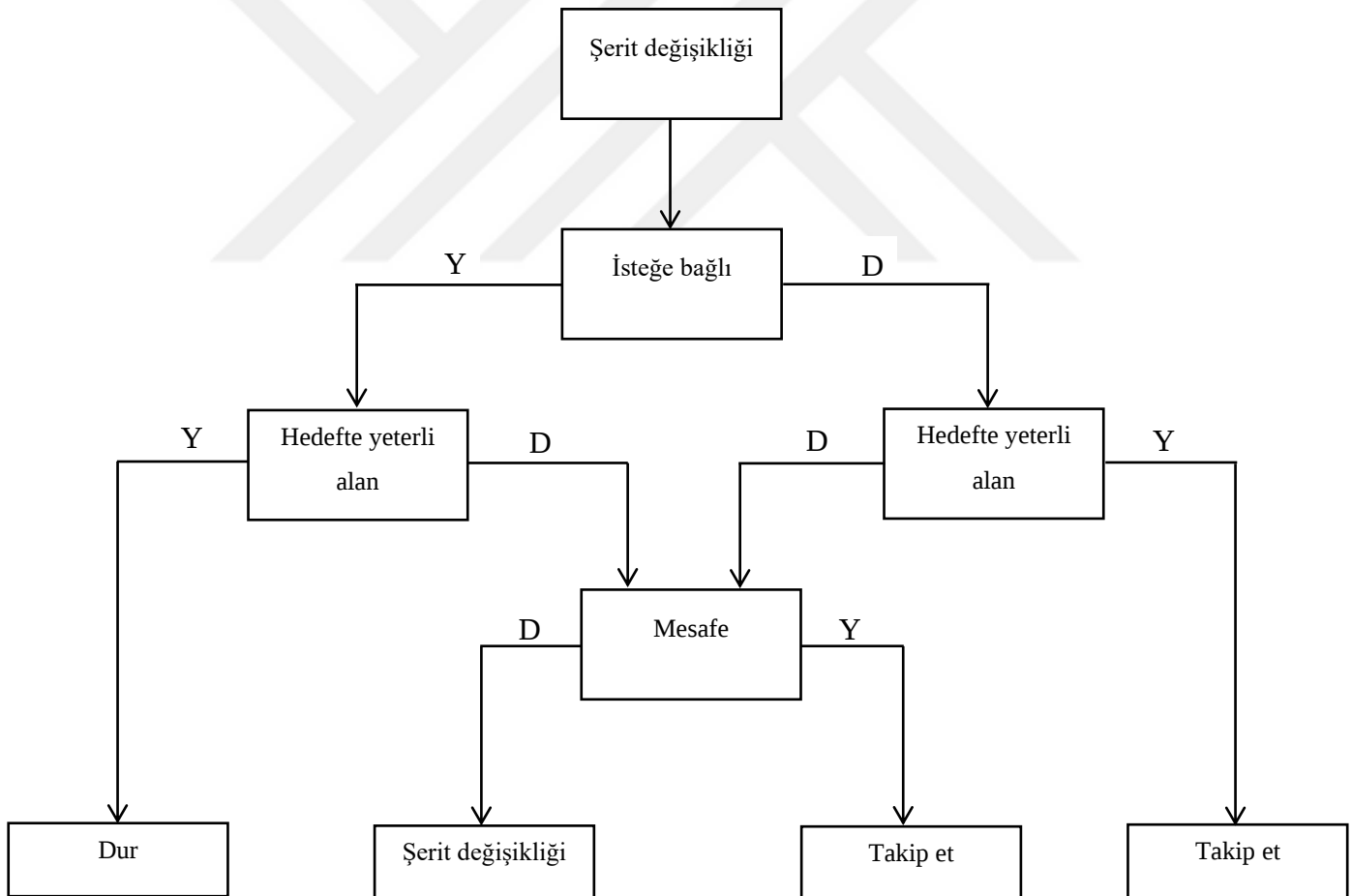
Farklı şerit değiştirme ortamına göre şerit değiştirme davranışı, zorunlu şerit değiştirme ve isteğe bağlı şerit değiştirme olmak üzere iki duruma ayrılabilir. Zorunlu şerit değiştirme, aracın şerit değiştirmek zorunda olduğu normal sürüş sürecidir. Bu tür davranışın kilit noktası, Şekil 3-10'da gösterildiği gibi genellikle kavşaklarda, şerit birleşmelerinde, saptırmalarda ve engellerde meydana gelen, insansız aracın o noktaya gitmeden önce şerit değişikliğini tamamlaması gereken en son şerit değiştirme noktasının olmasıdır. İsteğe bağlı şerit değiştirme ise, amacı sürüş hedefine ulaşmak ve öndeki aracı geçmek olan insansız araçların şerit değiştirme davranışıdır. Bu davranış, genellikle arkadaki aracın öndeki araçtan daha hızlı gittiği ve bitişik şeritteki sürüş koşullarının daha üstün olduğu durumlarda ortaya çıkan sürüş süresini azaltmak için sürüş amacını tamamlamak içindir.



Şekil 3-10: Zorunlu şerit değişikliği

Araçlar şerit değiştirme kararı verirken üç tür durum olabilir: dur, takip et ve şerit değiştir. Durdurma, aracı yavaş yavaş sabit bir duruma getirmek için fren sistemi tarafından gerçekleştirilir [27]. İnsansız araçlar için zorunlu şerit değişikliğinin olduğu ancak fiili durumun şerit değişikliği için gerekli koşulları sağlayamadığı durumlarda durma güvenli ve etkili bir yöntemdir. Öndeki arabayı takip etmek, arkadaki arabanın orijinal şeritte sürmeye devam etmek için öndeki arabayı takip etmesi anlamına gelir. Otonom araçlar isteğe bağlı şerit değiştirme durumundayken, ancak şerit değişikliği için gerekli koşulları gerçekleştiremezse, arabayı takip etmek uygulanabilir bir

yöntemdir. Sürüş devam ederken, kontrol sistemi orijinal durumunu koruyacak ve şerit değiştirme fırsatını bekleyecektir. Hedef şeritte yeterli şerit değiştirme alanı ve öndeki araca olan mesafe varsa, otonom araç şerit değiştirebilecektir. Bu tezde, trafik verimliliğini artıran ve güvenliği sağlayan bir şerit değiştirme stratejisi tasarlanmıştır. Karar vermenin akış şeması Şekil 3-11’de gösterilmektedir. Sürücüsüz bir araç zorunlu şerit değişikliği ile karşılaştığında hedef şerit değiştirme alanı yeterliyse araç şerit değiştirecek, aksi takdirde araç sadece duracak ve bir sonraki şerit değiştirme fırsatını bekleyecektir. Sürücüsüz araç, öndeki araçla çarpışmayı önlemek için veya isteğe bağlı şerit değişikliği olan diğer sürüş amaçları için şerit değiştirdiğinde, hedef şerit yeterli şerit alanına sahipse ve öndeki araca olan mesafe güvenli mesafeden fazlaysa, araç şerit değiştir. Aksi takdirde aracın yeni şerit değiştirme fırsatları oluşana kadar orijinal şeridi takip etmeye devam etmesi gerekir [38].



Şekil 3-11: Karar verme stratejisi

Otonom araçlar, şerit değiştirme kararına göre yol planlaması yapmaktadırlar. Esas olarak şerit değiştirme kararına dayalı olarak öndeki aracı durdurmak ve takip etme koşulları uygulanır. Şerit değişikliği uygun olduğunda, uygun bir yol planlanacaktır.

3.3.2 Şerit Değiştirme Yolu Planlaması

Araçlar için şerit değiştirme çok yaygın bir davranıştır. Bitişik bir şeride yumuşak geçiş, otonom sürüşün temel gereksinimidir. Özellikle araçlar orijinal rotayı engelleyen engellerle karşılaştığında şerit değiştirme hareketlerinin önemi oldukça belirgindir. Sağduyu, şerit değiştirme hareketlerinin otonom sürüşün sürekliliği ile ilgilidir. Uygun şerit değiştirme yolunun seçilmesi geçişin düzgün olması açısından önemlidir [39].

Sinüs şeridi değiştirme yörüngesi tasarımı basit ve ayarlanması kolaydır, ancak ilk yanal hızlanma sıfır değildir, bu da yolculara hemen yanal bir yük uygulanacağı anlamına gelir. Bu tür bir darbe seyahat deneyimlerine çok zarar verir. Ark yörüngesi, şerit değiştirme sürecinin zaman tüketimini en aza indirir de yanal hızlanmada yanal kontrolün zorluğunu artıracak adım değişikliği vardır.

Trapezoidal hızlanma şerit değiştirme yörüngesi sürekli olarak değişebilir, ancak planlama parametrelerinin ayarlanması gereklidir, bu nedenle esneklik zayıftır. Şu anda, şerit değiştirme araştırmasında kullanılan polinom yörünge yukarıdaki kusurlara sahip değildir. Bu planlanan polinom fonksiyonunun yer değiştirmesi, hızı ve ivmesi düzgündür. Yol planlaması, gerçek sürüş koşullarına göre gerçekleştirilir ve şerit değişimi sabittir ve kontrol edilmesi kolaydır. Bu bölüm polinom temelinde yolu tasarlayacak ve engel kısıtlama koşulları altında yolu oluşturmak için uygun yöntemleri analiz edecektir [40].

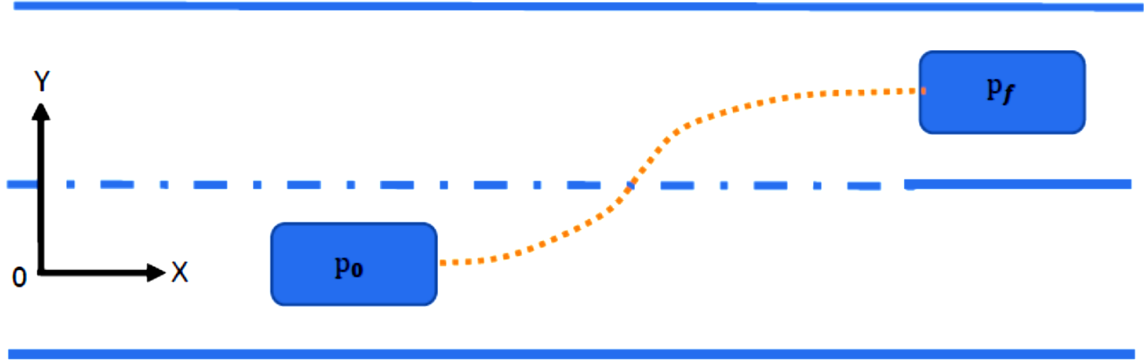
3.3.3 Polinom Fonksiyonuna Dayalı Yol Planlama Yöntemi

Şekil 3-12’de aracın ilk durumu varsayılarak şerit değiştirme diyagramı gösterilmektedir. t_0 için denklem 11’de polinom değişim değerleri verilmiştir.

$$p_0 = [x_0, x'_0, x''_0, y_0, y'_0, y''_0] \quad (11)$$

Altı durum için değişken, boyuna ve yanal durumu gösterir. Sırasıyla yer değiştirme, hız ve ivmedir. t_f ise bitiş zamanındaki durumdur.

$$p_f = [x_f, x'_f, x''_f, y_f, y'_f, y''_f] \quad (12)$$



Şekil 3-12: Şerit değiştirme şeması

Başlangıç durumu çeşitli sensörler tarafından ölçülebilir ve tahmin edilebilir. Son durum ise durum, sürüş talebine göre tahmin edilebilir. Vektörden, bilmek kolaydır yanal ve boylamsal durumlar sırasıyla 6 kısıtlamaya sahiptir. Bu nedenle, bir 6 parametrelilik nicel polinom yolu, boyuna ve yanal yol planlaması ve ilişkisi aşağıdaki gibi verilmiştir.

$$X(t) = \sum_{i=0}^{i=5} a_i \times t^i \quad (13)$$

$$Y(t) = \sum_{i=0}^{i=5} b_i \times t^i \quad (14)$$

Polinom katsayıları vektörlerde ifade edilebilir:

$$a = [a_0, a_1, a_2, a_3, a_4, a_5]^T, \quad b = [b_0, b_1, b_2, b_3, b_4, b_5]^T$$

Vektörler aşağıdaki gibi hesaplanır:

$$a = T^{-1}q_x, \quad b = T^{-1}q_y \quad (15)$$

$$q_x = [x_0, x'_0, x''_0, x_f, x'_f, x''_f]^T \quad (16)$$

$$q_y = [y_0, y'_0, y''_0, y_f, y'_f, y''_f]^T \quad (17)$$

$$T = \begin{bmatrix} 1 & t_0 & t_0^2 & t_0^3 & t_0^4 & t_0^5 \\ 0 & 1 & 2t_0 & 3t_0^2 & 4t_0^3 & 5t_0^4 \\ 0 & 0 & 2 & 6t_0 & 12t_0^2 & 20t_0^3 \\ 1 & t_f & t_f^2 & t_f^3 & t_f^4 & t_f^5 \\ 0 & 1 & 2t_f & 3t_f^2 & 4t_f^3 & 5t_f^4 \\ 0 & 0 & 2 & 6t_f & 12t_f^2 & 20t_f^3 \end{bmatrix} \quad (18)$$

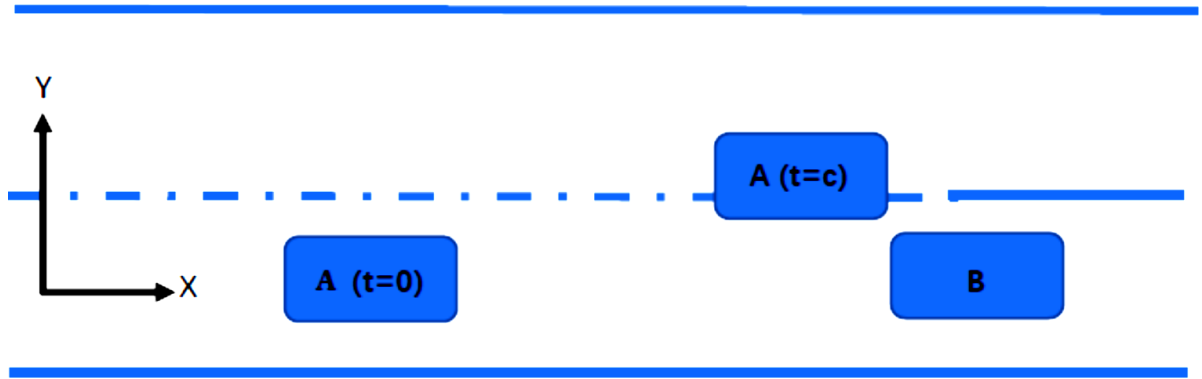
Zaman aralığı, başlangıç durumu ve son durum bilinebilirse polinom katsayısı hesaplanabilir ve ardından ideal yol elde edilebilir.

Otonom araçlar, kontrolün uygulanmasının zorluğunu azaltmak için ileri doğru hareket ederken normalde sabit bir hız seçerler. Arabanın t_0 'da 10m/s sabit hızla (0,0) konumuna hareket ettiğini ve önünde 50m statik bir engel olduğunu varsayalım. Ardından şerit değişikliği yapmak gerekiyor ve ilk olarak son konumu tahmin edilmesi gerekir. Yanal son konum genellikle 3,5 m olan yolun genişliğidir. Boyuna son konum o zaman 50m'dir. Sabit hız otonom aracın kontrolünün zorluğunu azaltabileceğinden boylamsal harekette sabit hız planlaması yapılması gerekir. Son boyuna hızın ilk hız ile aynı olduğunu varsayalım. Her ikisini de 10m/s'dir, o zaman şerit değiştirme işlemini tamamlamak, $50/10=5s$ sürer. O zaman $q_x = [0,10,0,50,10,0]$ ve $q_y = [0,0,0,3,5,0,0]$ elde edebiliriz. (18) denklemini kullanarak, Şekil 3-12 de gösterildiği gibi boyuna ve yanal durumu elde edebiliriz. Planlanan yolun istenen konuma doğru bir şekilde ulaşabileceği, hız ve ivme değişikliklerinin nispeten düzgün olduğu planlama sonuçlarından görülebilmektedir, Bu durumda yörünge takibi için kontrol sistemi ve şerit değişimi çok kolay bir şekilde sağlamaktadır [40].

3.3.4 Kısıtlama Analizi ve Optimizasyonu

Tezin bu bölümündeki araştırma, temel olarak engellerin olduğu duruma odaklanmaktadır. İsteğe bağlı şerit değiştirme durumu nispeten basittir. Şerit değiştirmenin amacı genellikle önünüzdeki engellerden kaçınmaktır. Ancak şerit değiştirme süreci aynı zamanda çarpışmaya da yol açabilir. Bu nedenle şerit değiştirme sürecini sınırlamak için makul kısıtlamalar koymak

gerekir. Bu arada, planlanan şerit değiştirme yörüngesinin uygulanabilir olmasını sağlayan aracın dinamik özellikleri de dikkate alınmalıdır. Engellerden kaçınma için uygun koşulların belirlenmesi, şerit değiştirme planlamasının tamamlanmasında önemli bir adımdır. Aşağıdaki bölümler dış ve iç kısıtlamaları tanımlar [21].



Şekil 3-13: Kısıt analizi

Şerit değiştirme sırasında meydana gelen çarpışma, esas olarak, aracın engele yaklaştığı ve bitişik şeride tam olarak girmediği, bunun sonucunda engelle temas ve çarpışma ile sonuçlanan durumu ifade eder. Çarpışmayı önlemek için ilk kısıtlama şu şekilde tanımlanabilir: A aracının ön ucu B engelinin sonuna ulaştığında, ikisi arasındaki yanıl mesafe engelin genişliğinden daha büyük olmalıdır, aksi takdirde çarpışma meydana gelir [14].

Genel pruvla açısı nispeten küçük olduğundan, arabanın yanıl yer değıştirmesi boyuna yer değıştirmeden çok daha küçük olduğu için, araba ile yolun boyuna eksenini arasındaki açının yaklaşık olarak sıfır olduğu varsayılabilir. Kritik çarpışma diyagramı Şekil 3-13'te gösterilmiştir. t_c zamanından sonra, A arabasının boylamasına yer değıştirmesi L ve yanıl yer değıştirmesi y_c 'dir ve araba A'nın ön ucu ve engelin ucu, doğrusal duruma yakındır. Çoğu durumda başka bir araç olan engelin genişliğinin w olduğunu varsayın. O zaman ilk kısıtlama $y_c > w$ ile temsil edilebilir. Mevcut hız V ise, kritik zaman $t_c = L/V$ ile kararlaştırılabilir. Mantıksız şerit değıştirme rota planlaması durumunda, şerit değıştirme işlemi sırasında bir çarpışma meydana gelebilir. Polinom fonksiyonu üzerine yapılan araştırmaya dayanarak, aynı boylamsal yer değıştirmede yanıl yer değıştirme, boyuna nihai yer değıştirme azaltılarak artırılabilir. Bu şekilde çarpışma önlenmiş olacaktır. Şekil 3-13 te planlanan rotalar temelinde, uzunlamasına yer değıştirmeyi 40m'ye ve şerit değıştirme süresini 4s'ye düşürün. Şekil 3-13, ayarlama öncesi ve ayarlama sonrası yer

deęiřtirmelerin karřılařtırmasının sonuçlarını göstermektedir. Açıkır ki, aynı boylamsal yer deęiřtirmede yanal yer deęiřtirmede önemli bir artış vardır.



4. OTONOM ARAÇLARDA NESNE ALGILAMA İÇİN KULLANILAN ALGORİTMALAR

Son yıllarda, insan müdahalesi olmadan yollarda gezinmek amacıyla yakın çevresini algılayabilen ve tepki verebilen bir otomobil platformu olan otonom aracın gelişimini destekleyen araştırmalarda önemli bir artış olmuştur ve pek çok otonom sürüş sisteminde, nesne algılama alt görevi, otonom navigasyonun en önemli ön koşullarından biridir, çünkü bu görev, araba kontrolörünün gelecekteki olası yörüngeleri değerlendirirken engelleri hesaba katmasına izin verir; bu nedenle, mümkün olduğunca doğru olan nesne algılama algoritmalarını arzuladığımız sonucu çıkar. Son yıllarda yaşanan teknolojik gelişimler ışığında birçok yüksek kaliteli nesne algılayıcı geliştirilmiştir. Bununla birlikte, otonom sürüş ayarına aktarılan nesne algılama sorunu, önde gelen birçok teknik tarafından ele alınmayan benzersiz zorluklar sunar. Bu zorluklar aşağıda verilmiştir [41].

- Araba kontrolörleri, uygulanabilir kontrolü elde etmek için genellikle optimizasyon problemlerini en az saniyede bir çözmelidir; Bu, araba kontrolörünün benzer bir zaman diliminde algılama sonuçlarını alması gerektiği anlamına gelmektedir.
- Algılama ardışık düzeninin tamamının hızlı olması gereksinimi, genellikle SIFT tanımlayıcılarının, bölge önerilerinin veya kayan pencerelerin çıkarılması gibi algılama performansını artıran görüntü ön işlemenin kullanımını engeller. Bu nedenle, otonom bir araba için bir dedektör, kesinlikle giriş görüntüsü üzerinde çalışırken makul derecede doğru kalmalıdır.
- Araba bilgi işlem sistemleri genellikle büyük ölçüde bellek kısıtlıdır ve bu nedenle, özellikle büyük girdi görüntü hacimleri ile büyük miktarda parametreye sahip dedektörleri depolamak ve çalıştırmak çoğu zaman mümkün değildir. Özellikle, bu sinir ağı yaklaşımlarının derinliğini sınırlardır [41].

4.1 AŞIRI BAŞARI ALGORİTMASI

Aşırı başarı algoritması (Overfeat) [42], nesne tanıma, algılama ve yerelleştirmeyi birlikte gerçekleştiren 2013 yılında piyasaya sürülen bir Evrişimli Sinir Ağı modelidir. Aşırı uyuma, ImageNet Büyük Ölçekli Görsel Tanıma Yarışması 2013'te yerelleştirme görevini kazanan, bugüne kadar ki en başarılı tespit modellerinden biridir. Aşırı başarı, sekiz katman derinliğindedir

ve büyük ölçüde, birden çok ölçekte algılama kutuları üreten ve bunları yinelemeli olarak toplayan örtüşen bir şemaya bağlıdır. Birlikte yüksek güvenilirlikli tahminlere dönüştürülür [32]. Görüntülerdeki nesneleri aynı anda sınıflandırmak, konumlandırmak ve algılamak için bir evrişimli ağ eğitiminin, tüm görevlerin sınıflandırma doğruluğunu ve algılama ve yerelleştirme doğruluğunu artırabileceğini göstermektedir. Makale, tek bir ConvNet ile nesne algılama, tanıma ve yerelleştirme için yeni bir entegre yaklaşım önermektedir. Ayrıca, öngörülen sınırlayıcı kutuları toplayarak yerelleştirme ve algılama için yeni bir yöntem sunuyoruz. Artan zorluk derecesinde üç bilgisayarlı görü görevi bulunmaktadır. Bunlar sınıflandırma, lokalizasyon ve algılamadır.

Bu zorluğun sınıflandırma görevinde her görüntüye, görüntüdeki ana nesneye karşılık gelen tek bir etiket atanır. Doğru cevabı bulmak için beş tahmine izin verilir. Bunun nedeni, görüntülerin birden fazla etiketlenmemiş nesne içerebilmesidir. Yerelleştirme görevi, görüntü başına 5 tahmine izin verildiği için benzerdir, ancak ek olarak, her tahminle birlikte tahmin edilen nesne için bir sınırlayıcı kutu döndürülmelidir. Tahmin edilen kutunun doğru kabul edilmesi için en az %50 temel gerçeğe uyması (pascal kesişme üzerinde birleşme kriteri kullanılarak) ve ayrıca doğru sınıfla etiketlenmesi (yani her tahmin bir etiket ve sınırlayıcı kutudur) olmalıdır. Birlikte ilişkilidir). Algılama görevi, her görüntüde (sıfır dâhil) herhangi bir sayıda nesne olabileceği ve yanlış pozitiflerin ortalama keskinlik 3 (mAP) ölçüsü tarafından cezalandırılması bakımından yerelleştirmeden farklıdır [43].

Özellik haritalarının uzamsal boyutu çıkarım adımı sırasında değişen giriş görüntüsünün boyutuna bağlıdır. Tablo 4-1'e bakıp uzaysal boyutları eğitmek ve katman 5'in en üstteki evrişim katmanı olduğuna dikkat edilmesi gerekir. Sonraki katmanlar tamamen bağlı, sürgülü pencere tarzında katman 5'e uygulanan bir sınıflandırıcı kullanılmaktadır. Bu tamamen bağlantılı katmanlar, uzaysal bir ortamda 1x1 kıvrımlar olarak görülebilir.

Tablo 4-1: Aşırı başarı için mimari özellikler

Layer (katman)	1	2	3	4	5	6	7	Output
Stage	conv+max	Conv+max	conv	conv	Conv+max	full	full	full
Channels	96	256	512	1024	1024	3072	4096	1000
Filter size	11 x 11	5 x 5	3 x 3	3 x 3	3 x 3	-	-	-
Conv. Sride	4 x 4	1 x 1	1 x 1	1 x 1	1 x 1	-	-	-
Pooling size	2 x 2	2 x 2	-	-	2 x 2	-	-	-
Pooling stride	2 x 2	2 x 2	-	-	2 x 2	-	-	-
Zero- padding size	-	-	1x1x1x1	1x1x1x1	1x1x1x1	-	-	-
Spatial input	231x231	24x24	12x12	12x12	12x12	6 x 6	1 x 1	1 x 1

Bu işlemler, sınıflandırıcının görüntüleme penceresinin havuzlama yoluyla 1 piksel kaydırılması olarak görülebilir. Bu prosedür, her görüntünün yatay olarak çevrilmiş versiyonu için tekrarlanır. Daha sonra dikkate alınması gereken kısımlar aşağıda açıklanmıştır.

- (i) Her sınıf için, her ölçekte ve çevirmede uzamsal maksimumu almak gerekir.
- (ii) Ortalama alma farklı ölçeklerden ve çevirmelerden elde edilen C-boyutlu vektörler belirlenmelidir.
- (iii) İlk 1 veya ilk 5'i alarak ortalama sınıf vektöründen öğeler (değerlendirme kriterine bağlı olarak) belirlenmelidir.

Yukarıda görülen Tablo 4-1 birkaç dikkate değer özelliğe sahiptir [44]. İlk olarak, ağıın iki yarısı, yani, öznitelik çıkarma katmanları (1-5) ve sınıflandırıcı katmanları (6-çıkı) zıt şekillerde kullanılır. Özellik çıkarma bölümünde, filtreler tek geçişte tüm görüntü boyunca kıvrılır.

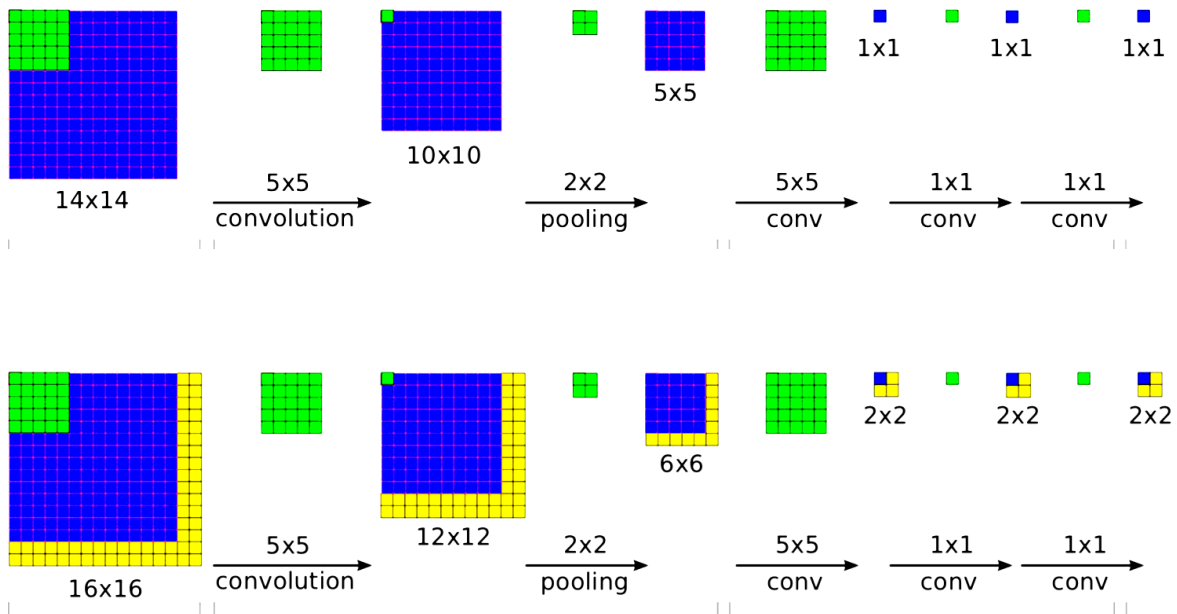
Tablo 4-2: Çok ölçekli yaklaşımının mekânsal boyutları

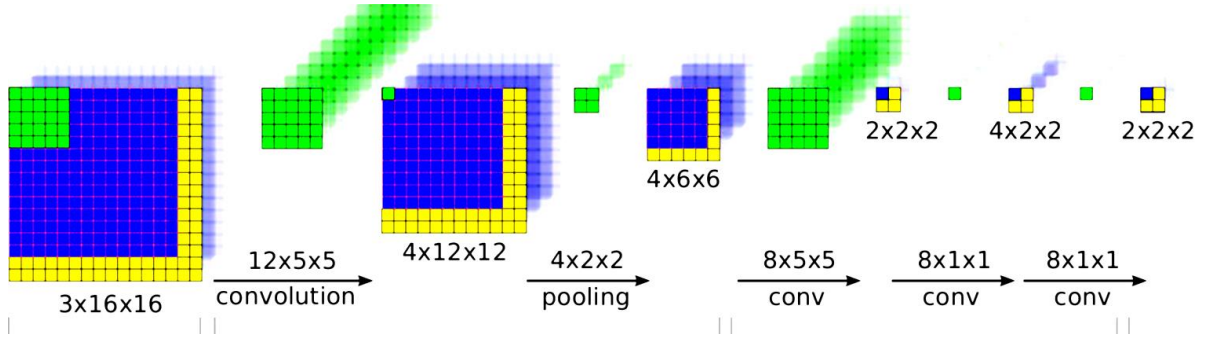
Scale	Input size	Layer 5 Per-pool	Layer 5 Post - pool	Classifier map(pre-reshape)	ClassifierMap size
1	245x245	17x17	(5x5)x(3x3)	(1x1)x(3x3)xC	3x3xC
2	281x317	20x23	(6x7)x(3x3)	(2x3)x(3x3)xC	6x9xC
3	317x389	23x29	(7x9)x(3x3)	(3x5)x(3x3)xC	9x15xC
4	389x461	29x35	(9x11)x(3x3)	(5x7)x(3x3)xC	15x21xC
5	425x497	32x35	(10x11)x(3x3)	(6x7)x(3x3)xC	18x24xC
6	461x569	35x44	(11x14)x(3x3)	(7x10)x(3x3)xC	21x30xC

6 farklı boyutta giriş görüntüsü farklı uzamsal çözünürlükte katman 5 havuzlanmamış özellik haritaları ile belirtilmemiş olsa da sonuçlanır Yukarıda verilen Tablo 4-2'deki parametrelerin hepsinde 256 özellik kanalı vardır.

4.2 TAHMİN OLUŞTURMA

Nesne sınırlayıcı kutu tahminleri oluşturmak için, sınıflandırıcı ve regresör ağlarını tüm konumlarda ve ölçeklerde aynı anda çalıştırılmaktadır. Bunlar aynı özellik çıkarma katmanlarını paylaştığından, sınıflandırma ağı hesaplandıktan sonra yalnızca son regresyon katmanlarının yeniden hesaplanması gerekir [9].



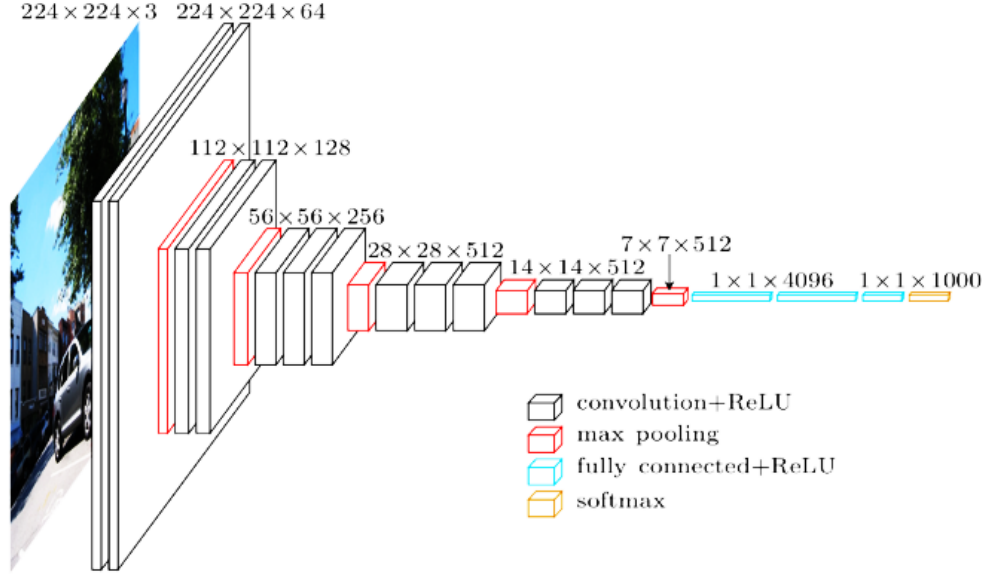


Şekil 4-1: Algılama için ConvNets'in verimliliği

Şekil 4-1'de eğitim sırasında, bir ConvNet yalnızca 1 uzamsal çıkmıştır. Her konumdaki bir c sınıfı için son softmax katmanının çıktısı, karşılık gelen görüş alanında c sınıfı bir nesnenin (mutlaka tam olarak içermese de) bulunduğu dair bir güven puanı sağlar. Böylece her sınırlayıcı kutuya bir güven atayabiliriz. Bir görünüm içinde yerleştirme, nesnenin sınırlayıcı kutusunu tahmin etmek için sınıflandırma ağı özelliklerinin üstünde bir regresör eğiterek gerçekleştirilir.

4.3 EVRİŞİMSEL SİNİR AĞI

VGG16, 2014 yılında ILSVR(Imagenet) yarışmasını kazanmak için kullanılan bir evrişimsel sinir ağı (ESA) mimarisidir. Bugüne kadar mükemmel görüntü modeli mimarisinden biri olarak kabul edilmektedir. VGG16 ile ilgili en benzersiz şey, çok sayıda hiper parametreye sahip olmak yerine, adım 1 ile 3x3 filtrenin evrişim katmanlarına odaklanmaları ve her zaman aynı dolguyu ve adım 2'nin 2x2 filtresinin maxpool katmanını kullanmalarıdır. Tüm mimari boyunca tutarlı bir şekilde evrişim ve maksimum havuz katmanları ve sonunda 2 FC'ye (tamamen bağlı katmanlar) ardından çıktı için bir softmax'a aktivasyonu sahiptir. VGG16'daki 16, ağırlıkları olan 16 katmana sahip olduğunu ifade eder. Bu ağ oldukça büyük bir ağıdır ve yaklaşık 138 milyon (yaklaşık) parametreye sahiptir [43].



Şekil 4-2: ESA Yapısı

4.4 HIZLI R-ESA

Hızlı R-ESA, hızlarını geliştirirken daha derin modellerin doğruluğunu yakalamaya çalışan bir modeldir. Hızlı R-ESA, bölge teklifleri hakkında tahminde bulunur ve modelin eğitim ve tahmin süresini hızlandırmak için bölge teklifi ve kesilmiş SVD çarpanlarına ayırma başına paylaşılan hesaplamayı kullanır [45].

Tablo 4-3: Örnek bir Hızlı R-ESA mimarisi

Katman	Giriş	Filtre boyutu	Adım	Sayı filtreleri	Aktivasyon	Çıkış
Conv1	80×80×3	8×8	4	32	ReLU	20×20×32
Conv2	20×20×32	4×4	2	64	ReLU	9×9×64
Conv3	9×9×64	3×3	1	64	SoftMax	7×7×64
fc4	7×7×64				ReLU	512
Fc5	512				Linear	3

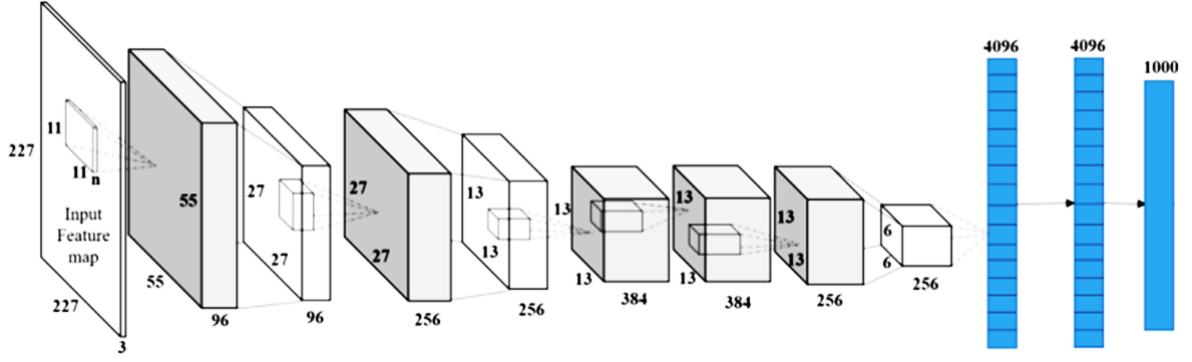
R-ESA'da her bölge önerisi, diğer bölge önerilerinden bağımsız olarak modele beslenir. Bu, tek bir bölgenin işlenmesi S saniye sürüyorsa, N bölgelerin $S*N$ saniye alacağı anlamına gelir. Fast R-ESA, birden fazla teklif arasında hesaplamaları paylaştığı için R-ESA'den daha hızlıdır.

R-ESA aynı görüntüden birden fazla IBM'yi örnekleyen Fast R-ESA'ye kıyasla her görüntüden tek bir IBM 'yi örnekler. Örneğin, R-ESA 128 farklı görüntüden 128 bölgeden oluşan bir toplu iş seçer. Böylece toplam işlem süresi $128*S$ saniyedir.

Daha Hızlı R-ESA için, yalnızca 2 görüntüden (görüntü başına 64 bölge) 128 bölgecik grup seçilebilir. Bölgeler aynı görüntüden örneklandığında, evrişimli katman hesaplamaları paylaşılır ve bu da süreyi azaltır. Böylece işlem süresi $2*S$ 'ye düşer. Bununla birlikte, aynı görüntüden birden fazla bölgenin örneklenmesi, tüm bölgeler birbiriyle ilişkili olduğundan performansı düşürür. Fast R-ESA modelinin avantajlarına rağmen, bölge önerileri oluşturmak için zaman alıcı Seçici Arama algoritmasına bağlı olduğundan kritik bir dezavantajı vardır. Seçmeli Arama yöntemi, belirli bir nesne algılama görevinde özelleştirilemez. Bu nedenle, veri kümesindeki tüm hedef nesneleri algılamak için yeterince doğru olmayabilir [45].

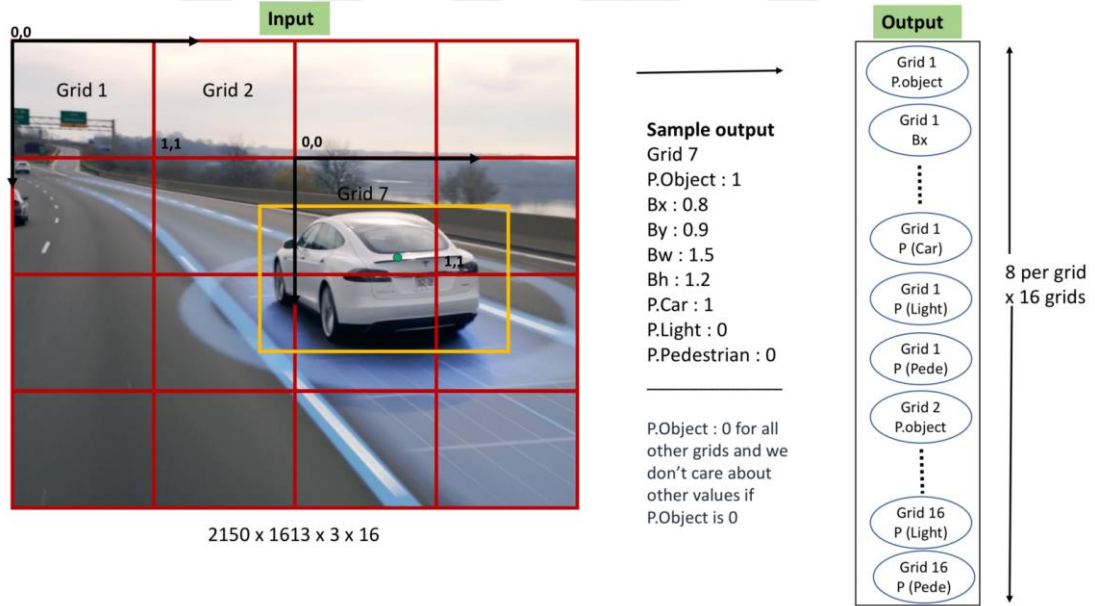
4.5 YOLO ALGORİTMASI

YOLO algoritması, tamamen ESA ve sinir ağından elde edilen son işlem çıktıları algoritmalarından oluşur. ESA'lar, ızgara benzeri veri topolojisini işlemek için uygun sinir ağlarının özel mimarileridir. Nesne tespitinde önem taşıyan ESA'ların ayırt edici özelliği parametre paylaşımıdır [46]. Her ağırlık parametresinin bir kez kullanıldığı ileri beslemeli sinir ağlarından farklı olarak, ESA mimarisinde çekirdeğin her bir üyesi girdinin her konumunda kullanılır, bu da ayrı bir parametre kümesi yerine her konum için bir parametre kümesinin öğrenilmesi anlamına gelir. Bu özellik, yoldaki tüm görüntüyü yakalamada önemli bir rol oynar. Şekil 4-3'de YOLO algoritmasına genel bakış sunulmaktadır [47].



Şekil 4-3: YOLO algoritması mimarisi

Bu algoritma, video akışından tek bir görüntünün çıkarılmasıyla başlar, bir sonraki adımda çıkarılan görüntü yeniden boyutlandırılır. Bu, YOLO ağına girişi temsil eder. Şekil 4-4'de gösterildiği gibi YOLO sınır ağı 4096 katmandan oluşmaktadır. Evrişimsel katmanları kullanmanın yanı sıra, mimarisi ayrıca artık katmanlar [48]. Ayrıca üst örnekleme katmanları ve atlama (kısayol) bağlantıları içerir [40].



Şekil 4-4: YOLO algoritması çıktısı

$$\begin{aligned}
& \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{I}_{ij}^{obj} [(x_i - x_j)^2 + (y_i - y_j)^2] + \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{I}_{ij}^{obj} [(\sqrt{w_i} - \sqrt{w_j})^2 + \\
& (\sqrt{h_i} - \sqrt{h_j})^2] + \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{I}_{ij}^{obj} [(C_i - C_j)^2] + \lambda_{noobl} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{I}_{ij}^{noobj} [(C_i - C_j)^2] + \\
& \sum_{i=0}^{S^2} \mathbb{I}_i^{obj} \sum_{c \in classes} (p_i(c) - p_j(c))^2
\end{aligned} \quad (19)$$

Bx: Objenin orjininin x koordinatı, By: Objenin orjininin y koordinatı, Bw: Objenin genişliđi, Bh: Objenin yüksekliđini göstermektedir. Denklem 19’da ızgara ierisinde obje bulunma durumunun kesinleřtirilmesi belirlenir [43].



5. YAKLAŞIM

Tez çalışmamızda, ESA'ların son zamanlarda bilgisayarlı görme görevlerinde çok önemli sonuçlar elde ettiğinden, temel yapımız olarak SimpleNet ESA mimarisini kullanmaktayız [49]. Bir ESA modeli, en düşük seviyedeki girdinin bir miktar girdi görüntüsü olduğu ve çıktının bir sınırlayıcı kutu tahminleri matrisi olduğu, özyinelemeli bir model bileşimi olarak düşünülebilir. Özyinelemeli olarak oluşturulacak işlevlerin seçimi büyük ölçüde değişir, ancak genellikle bir evrişimli işlemi, ardından doğrusal olmayan bir etkinleştirmeyi ve ardından bir maksimum havuzlama işlemini içerir; bu üç işlevin bileşimini birlikte “katman” olarak adlandırıyoruz. SimpleNet mimarimizde, n katmana sahip olmayı seçiyoruz, ardından son bir evrişim/doğrusal olmama ve bir tahmin matrisi veren yoğun, tam bağlantılı bir katman geliyor. Ağ derinliğinin tahminler üzerindeki etkilerini incelemek için 3 katmanlı ve 5 katmanlı bir model geliştirilmiştir.

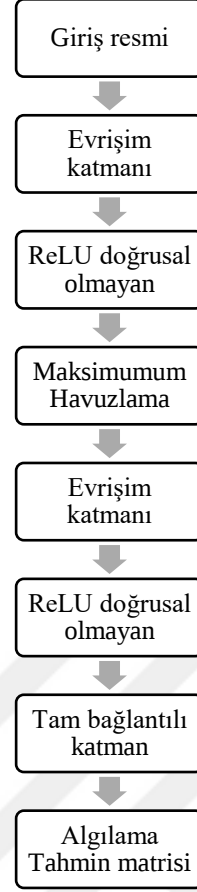
5.1 GİZLİ BİRİMLER

Her bir Evrişim Katmanı için her nöronu daha yerel bir alana yanıt verecek ve maksimum aktivasyon sayısını elde edecek şekilde eğitmek için 3x3'lük bir filtre boyutu kullanılmaktadır. Her Evrişim Katmanı ayrıca 3 nörona sahiptir; bu seçim, birim boyutunu giriş hacmine benzer tutmak ve parametre sayısını oldukça düşük tutmak için yapılmıştır [50].

5.2 DOĞRUSAL OLMAYAN AKTİVASYON FONKSİYONU

Doğrusal olmayan aktivasyonumuz için aşağıdaki formu alan Rektifiye Doğrusal Birim (ReLU) aktivasyonlarını kullanmaktadır.

$$\text{ReLU}(x) = \max\{x, 0\} \quad (20)$$



Şekil 5-1: SimpleNet Mimarisı

SimpleNet mimarisi negatif değerleri kaldırmaktadır. Şekil 5-1’de verilen mimaride negatif değerlerin engelleme işlemini yerine getirmektedir.

Bir sinir ağı modelinde tek bir düğüm düşünelim. Basitlik için, A ve B olarak adlandırılan iki girişi olduğunu varsayalım. A ve B’den düğümümüze gelen ağırlıklar sırasıyla 2 ve 3’tür. Böylece düğüm çıktısı $f(2A+3B)$ olur. f için ReLU fonksiyonunu kullanacağız. Yani $2A+3B$ pozitifse, düğümümüzün çıkış değeri de $2A+3B$ olur. $2A+3B$ negatif ise düğümümüzün çıkış değeri 0’dır.

Somutluk için, $A=1$ ve $B=1$ olduğu bir durumu ele alalım. Çıktı $2A+3B$ ’dir ve A artarsa çıktı da artar. Öte yandan, $B=-100$ ise çıktı 0’dır ve A orta derecede artarsa çıktı 0 kalır. Yani A bizim çıktımızı artırabilir veya artırmayabilir. Sadece B’nin değerinin ne olduğuna bağlı. Bu, düğümün bir etkileşimi yakaladığı basit bir durumdur. Daha fazla düğüm ve katman ekledikçe, etkileşimlerin potansiyel karmaşıklığı yalnızca artar. Ancak şimdi etkinleştirme işlevinin bir etkileşimi yakalamaya nasıl yardımcı olduğunu görmelisiniz [50].

5.3 ALGILAMA TAHMİNİ

Bir sınırlayıcı kutuyu b_2 , R_4 vektörü olarak temsil etmektedir, Burada ilk iki eleman, nesnenin x ve y koordinatlarını kodlar. Sınırlayıcı kutu algılamanın sol üst köşesi ve son iki eleman sağ alt köşenin x ve y koordinatlarını kodlar sınırlayıcı kutu algılama köşesi. Bizim algılama matrisin sabit sayıda satırı vardır ve bu nedenle bir sabit sayıda sınırlayıcı kutu uygulamada tahmin sayısı 21 ile sınırlandırılmaktadır, Buna karşılık gelen eğitim ve testte maksimum algılama sayısı kümelerdir [50].

5.4 UFUK ÇİZGİSİ BASTIRMA

Sınırlayıcı kutuları tahmin ettikten sonra, bir buluşsal yöntem kullanılmaktadır. Ufuk bastırma olarak tanımlanan tabir, yanlış pozitifleri bastırmaya yardımcı olma işlemidir. Çünkü otonom araçtaki kameralar sabit, çoğu yol yüzeyi aynı açıda görünür ve böylece aynı ufuk çizgisine sahiptir. Bir buluşsal yöntem olarak, herhangi bir sağ alt köşesi üstte görünen algılama kutusu tahmini ufuk çizgisi $y = 200$ piksel; bunun arkasındaki sezgi prosedür, tespit edilen nesne ufkun üzerindeyse çizgi, ya çok ileridedir ya da gökyüzündedir. Gökyüzündeysen, tespit edilen nesne olamaz. Bir araba veya yaya ve bu nedenle yanlış bir pozitifdir; aksi takdirde, biz nesneyi tekrar tespit etme şansımız yüksek olacaktır. Bu prosedür, yanlış pozitifleri önemli ölçüde iyileştirir, çünkü ufkun üzerinde birçok önemsiz tahmin yapılır ve eğitim prosedürünün bir eseri olarak belirtilir [50].

5.5 KAYIP FONKSİYONU

Bu tez çalışmasında, tahmin edilen tespit matrisi ile gerçek doğruluk matrisi arasındaki eleman bazında fark üzerinde iki farklı kayıp fonksiyonu denenmiştir. Vektör L_1 kaybı ve vektör L_2 kaybı. $\text{vec}(X)$, $X \in \mathbb{R}^{n \times m}$ matrisini $x \in \mathbb{R}^{nm}$ vektörüne yeniden şekillendirme işlemini gösterebilir, A tahmin matrisimiz olsun ve B temel doğruluk saptama matrisimiz olsun; o zaman iki kaybımız

$$L1(A-B) = \sum_i |vec(A - B)_i| \quad (21)$$

$$L2(A-B) = \sum_i vec(A - B)_i^2 \quad (22)$$

L2 kaybının, tahmin edilen ve kesin doğruluk matrisleri arasındaki büyük farklılıkları L1 kaybından daha sert bir şekilde cezalandırdığını unutmayın; bu, deneylerde çok daha iyi eğitime yol açan çok arzu edilen bir özellik olarak ortaya çıkıyor. Kaybımızda, aşırı uyumla mücadele etmek için ağırlık matrislerimize L2 düzenlemesini de dâhil edilmiştir. 10'dan fazla deneme çalışmasında, 1e – 5'lik bir düzenleme değeri en iyi sonucu verdiğini bulmaktadır [38].

5.6 EĞİTİM ALGORİTMASI

Optimizasyon prosedürü için, ağı her parametresini, önceki gradyan güncellemelerinin bir tür ortalaması olarak hizmet eden ve eğitim ve yakınsamayı hızlandırmaya yardımcı olan ayrı bir momentum parametresiyle güncelleyen adaptif momentumu (Adam) kullandık. Öğrenme hızımız için (1e–3) kullanmaktadır. 10 denemede hiç bozulma olmamasından daha iyi performansla sonuçlanan bir dizi bozulma hızı ve bozulma adımı bulamadığımız için öğrenme hızı kullanmayı ihmal etmektedir. Veri boyutumuz için, her eğitim örneğinin faydasını en üst düzeye çıkarmak için her seferinde bir görüntü üzerinde eğitim vermeyi seçtik. Toplam 30 döngüde eğitimi tamamlanmıştır.

Her görüntüye, her algılamanın sınıflandırmasını, görüntü sınırlayıcı kutu koordinatlarını, algılamanın metre cinsinden genişliğini, uzunluğunu ve yüksekliğini ve algılamanın dünyadaki 3D konumunu ve yönünü verdiği bir algılama listesi eşlik eder. Bu çalışmada, koordinatların yalnızca sınırlayıcı kutu temel gerçeği tahminlerinden yararlanıyoruz. Ancak yaklaşımımızın, saptama tahmin matrisindeki sütun sayısını basitçe genişleterek diğer gerçek değerli temel gerçekleri tahmin etmeye kolayca genişletilebilir olması gerektiğine inanıyoruz.

Veri normalleştirme ve ön işleme için, aydınlatma normalleştirme, yönlendirme ayarı ve rastgele kırpmaya gibi çeşitli ön işleme tekniklerini göz ardı ediyoruz. Çünkü bu tekniklerin birçoğunun otonom bir sistemde gerçek zamanlı olarak uygulanması mümkün değildir ve gerçek dünya algılama senaryolarını mutlaka yansıtmaz.

6. DENEYSEL ÇALIŞMALAR

6.1 VERİ SETİ

Bu tez çalışmasında kullanılan verileri Kaggle tarafından sağlanan halka açık bir veri tabanından elde edilmiştir. Bu veri seti derin öğrenme algoritmalar da kullanacak şekilde elde etmek için Lebelİmg programı kullanmıştır. Bu veri setinde toplam 1176 veri bulunmaktadır. Deneysel çalışmada veri setinin %15'i test ve geriye kalanı eğitim için kullanılmaktadır. Veriler; 380, 676, 3 boyutundadırlar.

Tablo 6-1: Veri kümesi dağılımı

Veri Seti Türü	YOLO V5, V7	ESA	Hızlı R-ESA	SimpleNet
Eğitim	1001	1001	1001	1001
Test	175	175	175	175

Bu çalışmada Kaggle tarafından sağlanan halka açık bir veri örneği aşağıda Şekil 6-1'de gösterilmektedir.



Şekil 6-1: Örnek Veri görüntüsü

6.2 BİLGİ İŞLEM ORTAMI

Algoritmamızı CPU: Intel Core i5 2.9GHz ve GPU ile donatılmış bir bilgisayarda çalıştırıyoruz. GeForce GTX 1080 Ti.grafik kartı ile eğitilen model, ağırlıkları ile birlikte ideal olanı seçme fırsatı veren her 1001 eğitim adımı için kaydedilmiştir ve 1001 karelik eğitimden sonra, yaklaşık 11 saatin sonunda sistemin doğruluğu tespit edilmektedir.

6.3 ALGORİTMALARIN PERFORMANS ANALİZİ

Bu tez çalışmasında, algoritmaların doğruluğu ölçmede kullanan karmaşıklık matrisi kullanılmıştır. Bu matrise ait Denklem (23-(25) 'de verilen Doğruluk, Hassasiyet ve Kesinlik metrikleri kullanılmıştır. Aşağıda Şekil 6-2 de karmaşıklık matrisi verilmiştir.

		GERÇEK	
		Pozitif	Negatif
TAHMİN	Pozitif	TP	FP
	Negatif	FN	TN

Şekil 6-2: Hata Matris

İyileştirilmiş YOLOv5 ve YOLOv7 ağının etkinliğini kontrol etmek için orijinal YOLOv5 modeli ile geliştirilmiş YOLOv5 modeli arasında karşılaştırmalı bir deney yapıldı. Genel olarak konuşursak, test sonuçları dört kategoriye ayrılabilir: TP (Gerçek Pozitif), doğru tahminin pozitif örneğidir; FP (Yanlış Pozitif), yanlış tahminin pozitif örneğidir; TN (Gerçek Negatif), doğru tahminin negatif örneğidir; FN (Yanlış Negatif), yanlış tahminin negatif örneğidir [41].

$$\text{Doğruluk} = \frac{TP+TN}{TP+FP+TN+FN} \quad (23)$$

$$\text{Kesinlik} = \frac{TP}{TP+FP} \quad (24)$$

$$\text{Hassasiyet} = \frac{TP}{TP+FN} \quad (25)$$

Denklem (23) pozitif durumların ne kadar başarılı tahmin edildiğini gösterir ve denklem (24) pozitif olarak tahmin edilen bir durumdaki başarıyı gösteren durumdur. Yukarıdaki karmaşıklık matrisini uygulayarak tablo 6-2 de nesne tespiti için kullanılan bütün algoritmaların yüzdelik doğruluğu verilmiştir.

Tablo 6-2: Derin nesne dedektörleri için tahmin doğruluğu

Model	Algoritmaların Doğrulu	Kesinlik	Hassasiyet
VGG16	94.01%	92%	95.20%
Hızlı R-ESA	80.07%	79.90%	80.70%
SimpleNet	78.30%	75.10%	76%
YOLO V5	76.42%	70%	77.27%
YOLO V7	83.90%	81.50%	84.49%

6.4 EĞİTİM

Bu çalışmada yapılan eğitimde, her biri dört deneyden oluşan on deneme yapılmıştır. İki farklı kayıpla (L1 ve L2) iki farklı mimari (3 katmanlı ve 5 katmanlı) uygulanmıştır. Her bir deney için eğitim ve doğrulama kayıplarının çizimleri Şekil 4-2 ve 4-3’de görülebilir. Her iki kayıp için de 5 katmanlı model, doğrulama kaybının eğitim kaybından önemli ölçüde daha yüksek olmasıyla verilere ciddi şekilde uyum sağlamaktadır. L1 kaybı ile her iki modelin de verilere fazla uyduğu not edilmiştir. Bunun nedeni, L1 kaybının sınırlayıcı kutu tahminlerinin temel gerçekleriyle hizalanması için ağırlıkları yeterince hızlı bir şekilde güncellemeye teşvik edecek kadar ciddi bir ceza olmamasıdır.

Kantitatif sonuçlarımız için, Precision Recall eğrisinin altındaki alan olarak tanımlanan Ortalama Hassasiyet puanını incelemektedir. Tahminimizin gerçek pozitif ve yanlış pozitif

oranını, tahmin edilen kutu ile temel doğruluk kutusu arasındaki örtüşme alanını, tahmin kutumuz ve zemin tarafından kapsanan toplam alana bölen Birlik Üzerinden Kesişme (IoU) metriği aracılığıyla belirlemektedir. Doğruluk kutusu; bu örtüşme bazı eşiklerden büyükse, algılamamızı gerçek bir pozitif olarak işaretlemektedir ve burada 0,5'lik bir eşik kullanmaktadır. Gerçek zamanlı olarak kabul edilmesi için, bir algılama algoritmasının saniyede en az 30 kare veya görüntü başına 0.033 saniye hızında tahmin yapması gerekir. 10 denemede modelimiz, gerçek zamanlı olarak düşünülen minimum hızdan 3 kat daha yavaştır. Görüntü başına ortalama 0.092 saniyelik bir hızda tahmin yapıyor. Diğer yöntemlerle karşılaştırma Tablo 6-2'de bulunmaktadır. SimpleNet'in saygın tahmin süresi elde etmesine rağmen, YOLO mimarisi tarafından hala önemli

Tablo 6-3: Derin nesne dedektörleri için tahmin hızları

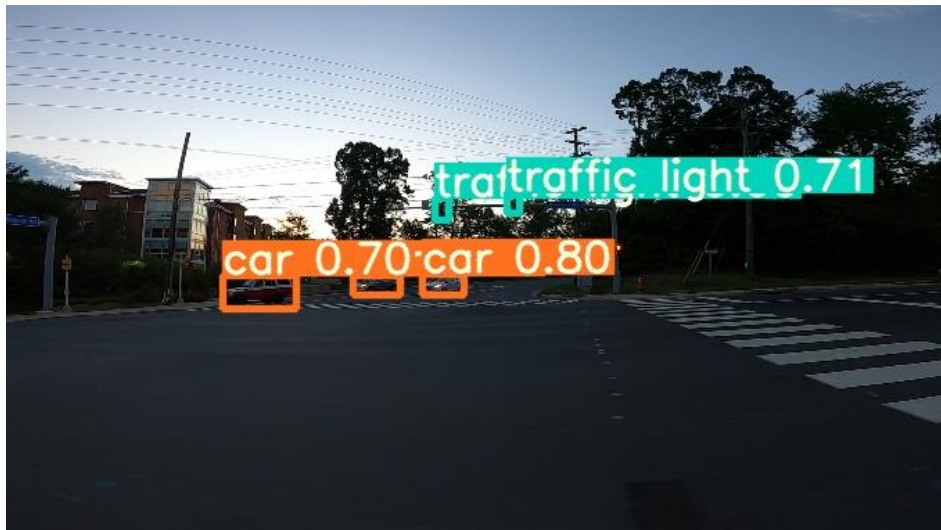
Model	Tek görüntüyü tahmin etme zamanı
VGG16	47s + Bölge zamanı
Hızlı R-ESA	.3s + Bölge zamanı
SimpleNet	.09s
YOLO V5	.0222s
YOLO V7	.018s

Ölçüde geride kaldığını görüyoruz. Aşağıdaki tabloda bölge tahmin süresinin 0.018 saniyeye kadar sürebileceği görülmektedir.

7. SONUÇLAR

Bu çalışmada, deneysel çalışmalar için giderek artan sayıda görüntü ve video karesini içeren bir veri tabanı kullanılmaktadır. Tüm bu görüntüler, farklı çevre koşullarında (güneşli, bulutlu, gece, gölgeli, yağmurlu) otoyollarda ve normal yollarda, kesikli işaretlerde, düz ve kavisli yollarda çekilmiştir. Şekil 3-1, 2. sistemin farklı yol koşulları altındaki performansını göstermektedir. Ek olarak, Şekil 3-5 'de algoritmanın sağlamlığını ve gerçek zamanlı performansını gösteren şerit sınırlarının başarıyla çıkarıldığını gözlemlenebilmektedir. Ancak, görüntünün ön planında keskin eğriler ve şiddetli yağmur altında şeritlerin doğru tespiti gibi bazı sorunlar hala çözülmedi ve yakalanan kareler araç hareketinden dolayı o kadar stabil değil ve bu nedenle, iyileştirmemiz gerekiyor. Bu problemlerin üstesinden gelmek için algoritma geliştirmeye açıktır.

Nitel sonuçlar için, modelimizin performansını daha iyi anlamak için başarılı ve başarısız tespit örneklerini incelemektedir. Birden çok gerçek tahmini olmasına ve detektörümüz yalnızca bir tane yaymasına rağmen, kutu nispeten doğru bir şekilde lokalize edildiğinden ve bir araba kontrolörüne iletilen bilgi, arabanın bir sonraki zaman adımında belirleyici bir eylemde bulunması için yeterli olacağından, bunu hala bir başarı olarak öne çıkmaktadır. Şekil -1'de, iki gerçek tahmininin olduğu ve detektörün iki algılama çıktısı aldığı bir durumu vardır. Bu senaryoda, detektör kutuları temel gerçeklerle uyumlu değildir. Bu durum iyileştirilebilse de (aslında, IoU kullanarak), iki nesneden daha yakın olanın nispeten iyi algılandığı göz önüne alınmaktadır.



Şekil 7-1: Bir eğitim görüntüsünde pozitif tahmin sonuçları

Bu tez çalışmasında, yüksek ön işleme veya yüksek derin değerlendirmelere ihtiyaç duymadan bir görüntü için sınırlayıcı kutu tahminleri yapan derin nesne algılama yaklaşımı incelenmektedir. Ortaya çıkan SimpleNet, makul tahmin doğruluğu sağlar ve neredeyse gerçek zamanlı olarak çalışır. SimpleNet, hiçbir şekilde son teknoloji olmazsa da kayıp fonksiyonlarının gelişmesi için bir çözüm yoludur. Çoğu ağ gücü katmanların derinliğinden alırken, bu çalışmada gücü sınırlı sayıda parametre ve işlevsel temsil ile uygun kayıp işlevlerinden türetmeye çalışılmıştır.

Tez çalışmasında otonom araçlar için nesne tespiti ve şerit belirleme uygulamasını gerçekleştirilmiştir. Bu amaçla YOLO V5, YOLO V7, ESA VGG16, Python OpenCV algoritmaları kullanılmıştır. Otoyoldaki imge görüntülerinin tespit test doğrulukları sırasıyla, VGG-16 ESA modeli 94.01%, YOLO V5 algoritması ile 76.42% elde edilmiştir. Nesne tespiti işleminde, VGG-16 ESA modeli imgenin tümündeki nesneler üzerinde işlem yaparken, YOLO V5 algoritması sürekli veri ya da imge içindeki nesneler algılamaktadır. YOLO V5 algoritmasının imge başına tespit etme hızı 0.018s iken, VGG-16 ESA modelinin GPU'daki bir görüntüdeki nesnelerin algılanması 0.47s'de gerçekleşmektedir. Sonuç olarak YOLO V5 algoritması, VGG-16 ESA model ve diğer modelleri göre daha hızlı bir şekilde nesneleri tespit edebilmekte buna karşın ise VGG-16 ESA modeli daha yüksek tespit doğruluk oranına sahiptir. Ayrıca Tablo 6-2' de görüleceği üzere, şerit belirleme işlemi de OpenCV algoritmaları kullanılarak başarılı bir şekilde gerçekleştirilmiştir.

8. KAYNAKLAR

- [1] “<https://www.statista.com/statistics/438016/number-of-road-deaths-in-turkey/> (Eriřim Tarihi: 20.05.2022)”.
- [2] “Bertozzi, M., & Broggi, A. (1998). GOLD: A parallel real-time stereo vision system for generic obstacle and lane detection. *IEEE transactions on image processing*, 7(1), 62-81.”.
- [3] “Anderson, J. M., Nidhi, K., Stanley, K. D., Sorensen, P., Samaras, C., & Oluwatola, O. A. (2014). *Autonomous vehicle technology: A guide for policymakers*. Rand Corporation.”.
- [4] “Arıkan, A., Kayaduman, A., Polat, S., řimřek, Y., Dikmen, İ. C., Bakır, H. G., ... & Abbasov, T. (2018, September). Control method simulation and application for autonomous vehicles. In *2018 international conference on artificial intelligence and data processing (IDAP)* (pp. 1-4). IEEE.”
- [5] “DEMİR, M. N., & ALTUN, Y. (2020). Otonom Araçla Genetik Algoritma Kullanılarak Haritalama ve Lokasyon. *Düzce Üniversitesi Bilim ve Teknoloji Dergisi*, 8(1), 654-666.”
- [6] “Jo, K., Kim, J., Kim, D., Jang, C., & Sunwoo, M. (2014). Development of autonomous car—Part I: Distributed system architecture and development process. *IEEE Transactions on Industrial Electronics*, 61(12), 7131-7140.”
- [7] “Özbay, H., Közkurt, C., DALCALI, A., & Tektaş, M. (2017). Geleceğın ulaşım tercihi: Elektrikli araçlar. *Akıllı Ulaşım Sistemleri ve Uygulamaları Dergisi*, 3(1), 34-50.”.
- [8] “Tektaş, M., & Tektaş, N. (2019). Akıllı ulaşım sistemleri (AUS) uygulamalarının sektörlere göre dağılımı. *Akıllı Ulaşım Sistemleri ve Uygulamaları Dergisi*, 2(1), 32-41.”.
- [9] “Singh, S. (2015). Critical reasons for crashes investigated in the national motor vehicle crash causation survey (No. DOT HS 812 115).”.
- [10] “Policy, F. A. V. (2016). *Accelerating the Next Revolution in Roadway Safety*. (2016, Sep) Washington. DC: National Highway Traffic Safety Administration.”.
- [11] “Chen, M., Jochem, T., & Pomerleau, D. (1995, August). AURORA: A vision-based roadway departure warning system. In *Proceedings 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems. Human Robot Interaction and Cooperative Robots* (Vol. 1, pp. 243-248). IEEE.”.
- [12] “<https://www.synopsys.com/automotive/what-is-autonomous-car.html> (Eriřim Tarihi: 04.07.2022)”.
- [13] “Özyurt, R., Elen, A., & Közkurt, C. Otomatik Depolama ve Bořaltma Sistemlerinde Kullanılan Robot Türleri. *TAM METİN KİTABI FULL TEXT BOOK*, 205.”.
- [14] “Campbell, M., Egerstedt, M., How, J. P., & Murray, R. M. (2010). Autonomous driving in urban environments: approaches, lessons and challenges. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 368(1928), 4649-4672.”.
- [15] “Suganuma, N., & Uozumi, T. (2012, August). Development of an autonomous vehicle—System overview of test ride vehicle in the Tokyo motor show 2011. In *2012 Proceedings of SICE Annual Conference (SICE)* (pp. 215-218). IEEE.”.
- [16] “Favarò, F. M., Nader, N., Eurich, S. O., Tripp, M., & Varadaraju, N. (2017). Examining accident reports involving autonomous vehicles in California. *PLoS one*, 12(9), e0184952.”.
- [17] “Chase, N., Maples, J., & Schipper, M. (2018, June). Autonomous vehicles: Uncertainties and energy implications. In *2018 EIA Energy Conference*.”.
- [18] “Bimbrow, K. (2015, July). Autonomous cars: Past, present and future a review of the developments in the last century, the present scenario and the expected future of autonomous vehicle technology. In *2015 12th international conference on informatics in control, automation and robotics (ICINCO)* (Vol. 1, pp. 191-198). IEEE.”.

- [19] “Thrun, S., Montemerlo, M., Dahlkamp, H., Stavens, D., Aron, A., Diebel, J., ... & Mahoney, P. (2006). Stanley: The robot that won the DARPA Grand Challenge. *Journal of field Robotics*, 23(9), 661-692.”
- [20] “Motors, G. (2018). Self-driving safety report. General Motors, Detroit.”
- [21] “Chen, J., Zhao, P., Mei, T., & Liang, H. (2013, July). Lane change path planning based on piecewise bezier curve for autonomous vehicle. In *Proceedings of 2013 IEEE International Conference on Vehicular Electronics and Safety* (pp. 17-22). IEEE.”
- [22] “Lee, J., Li, Y. A., Hung, M. H., & Huang, S. J. (2010). A fully-integrated 77-GHz FMCW radar transceiver in 65-nm CMOS technology. *IEEE Journal of Solid-State Circuits*, 45(12), 2746-2756.”
- [23] “Rohling, H., & Moller, C. (2008, May). Radar waveform for automotive radar systems and applications. In *2008 IEEE Radar Conference* (pp. 1-4). IEEE.”
- [24] “Xiaoming, L., Tian, Q., Wanchun, C., & Xingliang, Y. (2010, February). Real-time distance measurement using a modified camera. In *2010 IEEE Sensors Applications Symposium (SAS)* (pp. 54-58). IEEE.”
- [25] “X. W. a. M. Q. Minming Yang, «Examining lane change gap acceptance, duration and impact using naturalistic driving data,» no. <https://doi.org/10.1016/j.trc.2019.05.024>.”
- [26] “Park, W. J., Kim, B. S., Seo, D. E., Kim, D. S., & Lee, K. H. (2008, June). Parking space detection using ultrasonic sensor in parking assistance system. In *2008 IEEE intelligent vehicles symposium* (pp. 1039-1044). IEEE.”
- [27] “Rahiman, W., & Zainal, Z. (2013, June). An overview of development GPS navigation for autonomous car. In *2013 IEEE 8th Conference on Industrial Electronics and Applications (ICIEA)* (pp. 1112-1118). IEEE.”
- [28] “Borenstein, J., Everett, H. R., & Feng, L. (1996). Where am I? Sensors and methods for mobile robot positioning. *University of Michigan*, 119(120), 27.”
- [29] “Limsoonthrakul, S., Dailey, M. N., & Parnichkun, M. (2009, October). Intelligent vehicle localization using GPS, compass, and machine vision. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 3981-3986). IEEE.”
- [30] “Chen, Q., & Wang, H. (2006, June). A real-time lane detection algorithm based on a hyperbola-pair model. In *2006 IEEE intelligent vehicles symposium* (pp. 510-515). IEEE.”
- [31] “Lin, Q., Han, Y., & Hahn, H. (2010, May). Real-time lane departure detection based on extended edge-linking algorithm. In *2010 Second International Conference on Computer Research and Development* (pp. 725-730). IEEE.”
- [32] “Carreira, J., Li, F., & Sminchisescu, C. (2012). Object recognition by sequential figure-ground ranking. *International journal of computer vision*, 98(3), 243-262.”
- [33] “Schulte, S., De Witte, V., Nachtegaal, M., Van der Weken, D., & Kerre, E. E. (2006). Fuzzy two-step filter for impulse noise reduction from color images. *IEEE transactions on image processing*, 15(11), 3567-3578.”
- [34] “Nachtegaal, M., Schulte, S., Van der Weken, D., De Witte, V., & Kerre, E. E. (2005, May). Fuzzy filters for noise reduction: The case of Gaussian noise. In *The 14th IEEE International Conference on Fuzzy Systems, 2005. FUZZ'05*. (pp. 201-206). IEEE.”
- [35] “McDonald, J. B., & Mc Donald, J. B. (2001). Application of the hough transform to lane detection and following on high speed roads. In *in Motorway Driving Scenarios*”, in *Proceeding of Irish Signals and Systems Conference*.”
- [36] “Kreucher, C., Lakshmanan, S., & Kluge, K. (1998, October). A driver warning system based on the LOIS lane detection algorithm. In *Proceedings of IEEE international conference on intelligent vehicles* (Vol. 1, pp. 17-22). Stuttgart, Germany.”
- [37] “Roushdy, M. (2007). Detecting coins with different radii based on hough transform in noisy and deformed image. *GVIP Journal*, 7(1), 25-29.”

- [38] “Kim, W., Son, Y. S., & Chung, C. C. (2015). Torque-overlay-based robust steering wheel angle control of electrical power steering for a lane-keeping system of automated vehicles. *IEEE Transactions on Vehicular Technology*, 65(6), 4379-4392.”.
- [39] “M. B. A. D. a. J. F. K. Franziska Hartwich, «Customized Automated Driving Styles for Younger and Older Driver,» Germany,Braunschweig.”.
- [40] “Ahmad, A., Arshad, R., Mahmud, S. A., Khan, G. M., & Al-Raweshidy, H. S. (2014). Earliest-deadline-based scheduling to reduce urban traffic congestion. *IEEE Transactions on Intelligent Transportation Systems*, 15(4), 1510-1526.”.
- [41] “Ortega Serrano, E. (2020). Diseño e implementación de un sistema motorizado para telescopio basado en microcontrolador (Doctoral dissertation, Universitat Politècnica de València).”.
- [42] “Carreira, J., & Sminchisescu, C. (2010, June). Constrained parametric min-cuts for automatic object segmentation. In 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (pp. 3241-3248). IEEE.”.
- [43] “Lin, T., Maire, M., Belongie, S. J., Bourdev, L. D., Girshick, R. B., Hays, J., ... & Zitnick, C. L. (2014). Microsoft COCO: common objects in context. *CoRR abs/1405.0312* (2014).”.
- [44] “Hnewa, M., & Radha, H. (2020). Object detection under rainy conditions for autonomous vehicles: A review of state-of-the-art and emerging techniques. *IEEE Signal Processing Magazine*, 38(1), 53-67.”.
- [45] “Li, J., Liang, X., Shen, S., Xu, T., Feng, J., & Yan, S. (2017). Scale-aware fast R-CNN for pedestrian detection. *IEEE transactions on Multimedia*, 20(4), 985-996.”
- [46] *Zotero nasıl kullanılır? (6 dakikada öğren)*, (2020). Erişim: 05 Eylül 2021. [Çevrimiçi Video]. Erişim adresi: <https://www.youtube.com/watch?v=01eZ-JzLql4>
- [47] “Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. M. (2020). YOLOv4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*.”.
- [48] “He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).”.
- [49] “Dai, X. (2019). HybridNet: A fast vehicle detection system for autonomous driving. *Signal Processing: Image Communication*, 70, 79-88.”.
- [50] “Giusti, A., Cireşan, D. C., Masci, J., Gambardella, L. M., & Schmidhuber, J. (2013, September). Fast image scanning with deep max-pooling convolutional neural networks. In *2013 IEEE International Conference on Image Processing* (pp. 4034-4038). IEEE.”.

ÖZGEÇMİŞ

Kişisel Bilgiler		
Adı Soyadı	Md Nasrullah	
Doğum Yeri		
Doğum Tarihi		
Uyruğu		
Telefon		
E-Posta Adresi		

	Eğitim Bilgileri
	Lisans
Üniversite	
Fakülte	
Bölümü	
Mezuniyet Yılı	

Yüksek Lisans	
Üniversite	Bandırma Onyedli Eylül Üniversitesi
Enstitü Adı	Fen Bilimleri Enstitüsü
Anabilim Dalı	Fen Bilimleri
Programı	Akıllı Ulaşım Sistemi ve Teknoloji

Makale ve Bildiriler
Derin Öğrenme ve YOLO Algoritmaları ile Nesne Tespiti ve Şerit Belirleme