

KOCAELİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

ELEKTRONİK VE HABERLEŞME MÜHENDİSLİĞİ
ANABİLİM DALI

YÜKSEK LİSANS TEZİ

GÖMÜLÜ PLATFORMLAR İÇİN KUŞBAKIŞI GÖRÜNTÜLER
ÜZERİNDEN DERİN ÖĞRENME TEMELLİ GERÇEK
ZAMANLI ŞERİT TESPİT SİSTEMİ

GÖKHAN TARIM

KOCAELİ 2019

KOCAELİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

ELEKTRONİK VE HABERLEŞME MÜHENDİSLİĞİ
ANABİLİM DALI

YÜKSEK LİSANS TEZİ

GÖMÜLÜ PLATFORMLAR İÇİN KUŞBAKIŞI GÖRÜNTÜLER
ÜZERİNDEN DERİN ÖĞRENME TEMELLİ GERÇEK
ZAMANLI ŞERİT TESPİT SİSTEMİ

GÖKHAN TARIM

Prof. Dr. Oğuzhan URHAN
Danışman, Kocaeli Üniversitesi
Dr. Öğr. Üyesi Orhan AKBULUT
Jüri Üyesi, Kocaeli Üniversitesi
Dr. Öğr. Üyesi Ramazan DUVAR
Jüri Üyesi, Doğu Üniversitesi


.....


.....


.....

Tezin Savunulduğu Tarih: 29.07.2019

ÖNSÖZ VE TEŞEKKÜR

Bu tez çalışmasında, derin öğrenmenin günümüz teknolojileri için ortak hazırlanan kütüphane ve metodlar yardımıyla hızlı şekilde platformdan bağımsız şeritten ayrılma uyarı sistemleri için gerçekleştirilmiş şerit tespit sistemi, gerçek zamanlı ve başarı oranı yüksek bir şekilde uygulanmıştır. Bu tür çalışmaların açık kaynak kod yazarlarına yararlı olmasını temenni ederim.

Uzun yıllar boyunca yanında çalıştığım, mesleğimde kendimi geliştirmemde yardımcı olan ve her gün daha fazla geliştirmem gerektiğini gösteren değerli hocam Prof.Dr. Oğuzhan URHAN'a teşekkür ederim.

Bana her zaman güvenen ve maddi manevi desteğini esigemeyip arkamda her zaman durup gurur duyan annem Lale TARIM, babam Ersin TARIM, kardeşim Haktan TARIM'a teşekkür ederim.

Çalışmamın her anında, her satırında ve her konuda yardımcı olan uzun yıllar çalışma arkadaşlığı yaptığım Ayhan KÜÇÜKMANİSA'ya, çalışmam sırasında anlayış gösterip her zaman güvenen ve arkamda duran kız arkadaşım Didem BENLİOĞLU'na çok teşekkür ederim.

Çalışma Kocaeli Üniversitesi Bilimsel Araştırma Projeleri destek kapsamında 2019/047 nolu çalışma kapsamında desteklenmektedir. Kocaeli Üniversitesi'ne desteklerinden dolayı teşekkür ederim.

Haziran - 2019

Gökhan TARIM

İÇİNDEKİLER

ÖNSÖZ VE TEŞEKKÜR	i
İÇİNDEKİLER	ii
ŞEKİLLER DİZİNİ.....	iii
TABLolar DİZİNİ	v
SİMGELEr VE KISALTMALAR DİZİNİ	vi
ÖZET.....	vii
ABSTRACT	viii
GİRİŞ	1
1. ARAÇ TEKNOLOJİLERİ	3
2. ŞERİTTEN AYRILMA UYARI SİSTEMLERİ.....	5
3. YAPAY ZEKA.....	11
3.1. Makine Öğrenmesi.....	13
3.2. Derin Öğrenme.....	16
3.2.1. Nesne tespiti	20
3.2.1.1. You only look onces (YOLO).....	23
4. TEZ ÇALIŞMASINDA GERÇEKLENE N ŞERİT TESPİT SİSTEMİ.....	28
4.1. Ters Perspektif Dönüşümü.....	30
4.2. YOLO Eğitimi	37
4.3. YOLO Test.....	38
4.4. YOLO ile Eğitim Sırasında Elde Edilen Sonuçlar.....	41
4.5. Son İşlem	43
4.6. Deneysel Sonuçlar	46
5. SONUÇLAR VE ÖNERİLER	50
KAYNAKLAR	51
KİŞİSEL YAYINLAR VE ESERLER	56
ÖZGEÇMİŞ	57

ŞEKİLLER DİZİNİ

Şekil 2.1.	Otoyol şerit işaretleri	5
Şekil 2.2.	Araçlar seyir halinde iken kamera ile alınan bozucu etkilere sahip görüntüler (a) ve (b) Karanlıkta ışık patlaması (c) ve (d) Gölge ve bozucu etkenler	6
Şekil 3.1	Basit bir yapay sinir ağı.....	12
Şekil 3.2.	Makine öğrenmesinin çalışma blokları	14
Şekil 3.3.	Derin öğrenmenin çalışma blokları	14
Şekil 3.4.	Klasik makine öğrenmesi yöntemleri ile Derin öğrenme yöntemlerinin çalışma prensipleri	15
Şekil 3.5.	ImageNet yarışmasında 2011-2016 yılları arasında alınan sonuçlar	16
Şekil 3.6.	Hold-out yöntemi toplam veri setini eğitim ve test veri setlerine bölünmesi.....	17
Şekil 3.7.	(a) Tekrarlayan sinir ağı (b) Uzun kısa süreli bellek (c) Üretken ters ağlar	18
Şekil 3.8.	Derin öğrenme metodlarının çeşitlerine göre çalışma şekli. (a) Nesne sınıflandırma (b) Nesne lokalizasyonlama (c) Nesne tespiti (d) Nesne segmentasyonu	20
Şekil 3.9.	Faster R-CNN yönteminin blok diyagramı	21
Şekil 3.10.	SSD yönteminin blok diyagramı	22
Şekil 3.11.	Önerilen nesne tespit yöntemlerinin birbirlerine göre karşılaştırılması	22
Şekil 3.12.	YOLO 'nun genel çalışma prensibi.....	23
Şekil 3.13.	YOLO 'nun eğitim için oluşturduğu etiketli veri formatı	25
Şekil 3.14.	Etiketli veri oluşturma programı	25
Şekil 3.15.	YOLO ile eğitim için etiketleme programı ile işaretlenmiş örnek.....	26
Şekil 4.1.	Şerit işaretlerinin etiketlenmesi sonucu oluşan sınırlayıcı kutu kaplama alanı örneği	28
Şekil 4.2.	Tez çalışmasında gerçekleştirilen yöntemin detayları a) Eğitim sürecindeki yöntem b) Nihai geliştirilen yöntem	29
Şekil 4.3.	Orjinal kamera pozisyonunun sanal kamera pozisyonlarına dönüşümü	30
Şekil 4.4.	Ham şerit işaretleri barındıran görüntünün IPM işlemi sonrası dönüşümü (a) Ham görüntü (b) IPM işlemi sonrası oluşan görüntü	31
Şekil 4.5.	Araç içi kamera ile alınan görüntüde ufuk noktasının belirlenmesi	33
Şekil 4.6.	Şerit işaretleri barındıran imgenin ters perspektif dönüşümü sonrası ufuk noktasından uzaklaştıkça oluşan doğrusal olmayan bölge	34

Şekil 4.7.	Ham görüntünün ters perspektif dönüşümü (a) Ham görüntü (b) Ters perspektif dönüşümü sonrası elde edilen görüntü	35
Şekil 4.8.	YOLO ile eğitim için gerekli etiketli giriş imgesinin işlem aşamaları şablonu (a) Ham imge (b) Ufuk noktasından (siyah doğru) kesilen imge (c) Ters perspektif sonucu imge (d) Şeritin var olduğu bölgenin kesildiği imge	36
Şekil 4.9.	YOLO ile eğitim için hazırlanmış giriş imgesi örnekleri.....	36
Şekil 4.10.	Etiketli eğitim imge örnekleri	37
Şekil 4.11.	YOLO eğitim konfigürasyonları	38
Şekil 4.12.	YOLO test sonuçları.....	39
Şekil 4.13.	Sağ ve sol bölge ayrımı yapım şekli.....	40
Şekil 4.14.	Şerit tespitinin eğitimi için hazırlanılan veri setinin oluşturulması (a) İki sınırlayıcı kutunun yüksek oranda kesişmesi, (b) İki sınırlayıcı kutunun düşük oranda kesişmesi	41
Şekil 4.15.	Eğitim işlemi sırasında elde edilen hata oranı grafiği	43
Şekil 4.16.	YOLO ile Şerit tespit test çıktıları (a) Şerit tespit sonuç görüntüsü (b) Şerit tespit sonuçlarının sınıf benzerlik oranları.....	43
Şekil 4.17.	Borkar veri setinden üretilen şerit değişimi sırasında oluşan görüntü (a) Şerit değişim başlangıcı görüntüsü (b) Şerit değişim görüntüsü (c) Şerit değişim bitiş görüntüsü.....	44
Şekil 4.18.	YOLO tespit sonuçlarının dönüşümü (a) YOLO test sonucu (b) YOLO test sonucunun merkezinde çizilmiş doğru.....	45
Şekil 4.19	Geliştirilen yöntemin birinci çıktıları (a) Derin öğrenme test sonrası tespit edilen şeritler (b) Perspektif dönüşümü sonrası ham görüntüde işaretlenmesi (c) Deneysel sonuçlar için uzatılmış tespit sonuçları	45
Şekil 4.20.	Geliştirilen yöntemin ikinci çıktıları (a) Derin öğrenme test sonrası tespit edilen şeritler (b) Perspektif dönüşümü sonrası ham görüntüde işaretlenmesi (c) Deneysel sonuçlar için uzatılmış tespit sonuçları	45
Şekil 4.21.	Geliştirilen yöntemin deneysel sonuçlara hazırlanması (a) YOLO tespit sonucu (b) Tespit sonuçlarından alınan örnek noktalar (c) PM dönüşümü sonrası elde edilen doğrular.....	47

TABLULAR DİZİNİ

Tablo 4.1. YOLO eğitim parametre denemeleri hata sonuçları	42
Tablo 4.2. Şerit işareti tespit yöntemlerinin veri seti üzerinde performans değerlendirilmesi	48
Tablo 4.3. Yöntemin çalıştırıldığı makinelerin teknik özellikleri	49
Tablo 4.4. Geliştirilen yöntemin çalıştırıldığı makinelerdeki süreleri.....	49
Tablo 4.5. Karşılaştırılan yöntemlerin PC-3 ile çalıştırıldığında elde edilen süreler	49



SİMGELER VE KISALTMALAR DİZİNİ

Kısaltmalar

ACC	: Accuracy (Doğruluk)
ADAS	: Advanced Driver Asistan Systems (İleri Sürücü Destek Sistemleri)
AEB	: Auto Emergency Braking (Otomatik Acil Frenleme)
ANN	: Artificial Neural Network (Yapay Sinir Ağı)
CNN	: Convolutional Nerual Network (Evrışimsel Sinir Ağı)
CPU	: Central Processing Unit (Merkezi İşlem Birimi)
DNN	: Deep Neural Network (Deri Sinir Ağı)
FOV	: Field Of View (Görüş Açısı)
FPS	: Frame Per Second (Saniyedeki Çerçeve Sayısı)
GAN	: Generative Adversarial Network (Üretken Ters Ağlar)
GPU	: Graphics Processing Unit (Grafik İşlem Birimi)
HDD	: Hard Disk Drive (Kalıcı Hafıza Birimi)
ILSRVC	: Imagenet Large Scale Visual Recogni Challenge
IOU	: Intersection Over Union (Kesişen Bilgi)
IPM	: Inverse Perspective Mapping (Ters Perspektif Dönüşümü)
LIDAR	: Ligh Detection and Ranging (Işık Tespit ve Mesafe Hesaplayıcı)
LSTM	: Long Short Term Memory (Uzun Kısa Süreli Bellek)
LWDS	: Lane Warning Departure Systems (Şerit Takip Uyarı Sistemleri)
MSE	: Mean Square Error (Ortalama Karesel Hata)
MSER	: Maximally Stable Extremal Regions (Maksimum Kararlı Ekstremal Bölgeler)
NCAP	: New Car Assessment Programme (Yeni Araç Değerlendirme Programı)
OICA	: Organisation Internationale des Constructeurs d'Automobiles (Uluslararası Otomotiv Sanayicileri Birliği)
PC	: Personal Computer (Bilgisayar)
RAM	: Random Access Memory (Geçici Erişimli Hafıza Birimi)
RCNN	: Region Convolutional Nerual Network (Bölgesel Evrışimsel Sinir Ağı)
RESNET	: Resudial Neural Network (Resudal Yapay Sinir Ağı)
RNN	: Recurrent Neural Network (Tekrarlayan Sinir Ağı)
ROI	: Region of Interest (İlgi alanı)
RPN	: Regional Proposal Network (Bölgesel Öneri Ağı)
SSD	: Single Shot MultiBox Detector (Üretken Ters Ağlar)
TPU	: Tensor Processing Unit (Tensor İşlem Birimi)
YOLO	: You Only Look Onces
WHO	: World Health Organisation (Dünya Sağlık Örgütü)

GÖMÜLÜ PLATFORMLAR İÇİN KUŞBAKIŞI GÖRÜNTÜLER ÜZERİNDEN DERİN ÖĞRENME TEMELLİ GERÇEK ZAMANLI ŞERİT İŞARETİ TESPİT SİSTEMİ

ÖZET

Otonom araçlar son yıllarda büyük önem kazanmaktadır. Şerit takip sistemleri otonom araçların en önemli bileşenleridir. Şerit takip işlemi için şerit işaretlerinin başarılı bir şekilde tespit edilmesi gerekmektedir. Bu çalışmada otonom araçlar için şerit işaretlerinin tespitine yönelik derin öğrenme temelli gerçek zamanlı bir yöntem önerilmektedir.

Önerilen yöntemde kameradan alınan görüntüler öncelikle kuş bakışı görüntülerine çevrilmekte ve şerit işaretlerinin bulunması nesne tespit problemi olarak ele alınmaktadır. Nesne tespiti işleminde derin öğrenme kullanan YOLO yönteminden faydalanılmaktadır. YOLO yöntemiyle şerit işaretleri sınırlayıcı kutular içine alınarak tespit edilmektedir. Bu sınırlayıcı kutuların ortasında oluşturulan çizgiler sayesinde ters kuş bakışı dönüşümü sonrası şerit işaretlerinin orijinal görüntüdeki yeri tespit edilmektedir.

Son işlem aşamasında tespit edilen şerit işareti konumları filtrelenerek işlenmektedir. Böylece anlık şerit işareti tespit edilmeme durumlarının performansına olası olumsuz etkisi giderilmektedir. Ayrıca önerilen yöntem gömülü platformlarda gerçek zamanlı çalışma yeteneğine sahiptir.

Anahtar Kelimeler: Derin Öğrenme, Gömülü Platform, Otonom Araçlar, Şerit İşareti Tespiti.

REAL-TIME DEEP LEARNING BASED LANE MARKING SYSTEM OVER BIRD'S EYE VIEW FOR EMBEDDED PLATFORMS

ABSTRACT

Autonomous vehicles have gained importance in recent years. Lane detection systems are most important components of autonomous vehicles. Lane markings are need to be detected successfully for lane detection systems. In this study, deep learning based real time method is proposed for lane detection for autonomous vehicles.

In the proposed method, images that captured from camera is converted to bird's eye view perspective and the presence of lane markings are considered as object detection problem. In object detection process, YOLO method that using deep learning is used. The lane markings area is detected in bounding boxes thanks to YOLO method. With the lines that are center position of bounding boxes, lane markings in the location of original image is detected after reverse bird's eye view image transformation.

At the last step of processing, filtering to the detected lane marking locations. Thus, the negative impact of the instances not detected instantly are eliminated. Additionally, the proposed method is capable of working real-time on embedded platforms.

Keywords: Deep Learning, Embedded Platforms, Autonomous Vehicles, Lane Markings Detection.

GİRİŞ

Günümüzde araç kullanımı geçmiş yıllara göre büyük bir ivme ile artmaktadır. OICA'nın verilerine göre 2018 yılında dünyada toplam 10 milyon araç üretilmiştir. Bu araçların 1,5 milyonu ülkemizde üretilmiştir [1]. Üretim miktarlarının bu denli artması ile beraber tüketicilerin yeni ihtiyaçları doğmakta ve bu ihtiyaçları karşılamak amacıyla geliştirilen yeni otomobil teknolojileri günden güne artmaktadır. Otomobil üreticileri yeni istekler ve genişleyen otomobil piyasasında farklılık yaratmak amacıyla gerek elektrikli gerek fosil yakıtlı araçlarda sürücü hayatını koruyan, diğer araçlarla iletişimde olan ve araç sürüş keyfini artıran teknolojiler geliştirmektedir.

Dünyaya daha az zarar veren, daha az yakıt tüketimi ve maliyeti olan elektrikli araçlar oldukça ön planda bulunmaktadır. Bu tür gelişmelere rağmen artan araç sayısı ile beraber kaza oranları da oldukça yükselmektedir. Dünya Sağlık Örgütü'nün (World Health Organisation - WHO) yayınladığı rapora göre 2030 yılında kayıplı trafik kazaları dünya genelinde ilk beşe girecektir ve her yıl bu oluşan kazalarda insan kayıplarının 100 binde 18'ini (1,4 milyon yolcu) trafik kazaları oluşturmaktadır. Ülkemizde oluşan kayıplı ve yaralanmalı trafik kaza oranı 2006'dan beri oluşan trafik kazalarına oranla neredeyse iki kat artmıştır. Yine bu rapora göre oluşan kazalarda ölenlerin yüzde 43,2'sinin sürücüler oluşturduğu göstermektedir [2].

Trafik kazalarının önlenmesi ve yolcuların daha güvenli seyahat edebilmesi günümüzde en önemli konular arasında yer almaktadır. Bununla beraber araç kullanıcılarının istekleri ve tercihleri de sürücüye yardımcı olan bu tür sistemlerden yana olmaktadır. Bu tür sistemlere genel olarak İleri Sürüş Destek Sistemleri (Advance Driver Assistance Systems - ADAS) olarak adlandırılmaktadır. Üreticiler için gerek otonom araçlar, gerekse ileri sürücü destek sistemleri popülerleşen dünyamızın olmazsa olmazları arasında görülmektedir.

Bu tez çalışmasında ileri sürüş destek sistemlerinden şeritten ayrılma tespit sistemlerine alternatif bir yöntem geliştirilmiştir. Tez çalışmasının birinci bölümde araç teknolojileri hakkında bilgiler verilmiştir. Teknolojinin hangi yönde ilerlediği

hakkında detaylardan bahsedilmiştir. Çalışmanın ikinci bölümde şeritten ayrılma tespit sistemlerinin dünyaca belirlenmiş standartları ve sistemler hakkında genel bilgiler verilir, yaygın olarak kullanılan algılayıcılar hakkında bilgiler sunulmuştur. Çalışmanın üçüncü bölümünde yapay zeka teknolojilerinin kullanım alanları ve genel olarak kullanım amaçları hakkında bilgi verilmiştir. Trend bilimin teknolojiye etkisi ve avantajları hakkında detaylar verilmiştir. Çalışmanın dördüncü bölümünde ise derin öğrenme ile nesne tespiti yapabilen YOLO algoritması hakkında detaylar sunulmuştur. Eğitim, test ve son işlem süreçlerinin geliştirme aşamaları, literatürde yapılan diğer çalışmalara göre karşılaştırmaları ve deneysel sonuçlar anlatılmıştır. Bu tez çalışmasının son bölümünde ise genel olarak çalışmada elde edilen sonuçlar ve kazanımlar hakkında bilgi verilmiş olup geliştirilen yönteme yapılabilecek olası iyileştirmelere yer verilmiştir.

1. ARAÇ TEKNOLOJİLERİ

100 yılı aşkın süredir geliştirilen araç teknolojileri uzun yıllar üzerinde yapılan çalışmalar ve geliştirmeler sonucunda ortaya çıkmış olup kullanıcılarının sadece bir noktadan diğerine ulaşımını sağlamaktan öte rahatını ve konforunu da daha iyi bir noktaya taşıması amacıyla geliştirilen sistemler topluluğu olarak anılmaktadır. Araç teknolojileri temel olarak mekanik, kimyasal, elektrik ve elektronik teknolojilerden oluşmaktadır. Bu teknolojiler kimi zaman bir araç içi eğlence sistemi teknolojisi olmakta, kimi zaman doğaya daha az zarar veren bir motor teknolojisi olarak karşımıza çıkmaktadır. Bu çalışmada araç içi sürücü emniyeti için kullanılan elektronik sistemler ve teknolojiler üzerine odaklanılmıştır.

Günümüzde araçlarda elektrik ve elektronik sistemler, birçok kimyasal ve mekanik teknolojinin önüne geçmekte ve araçların içinde kullanılan mekanik teknolojilerin gitgide elektrik ve elektronik temelli teknolojilere dönüşmesine neden olmaktadır. Dünya güvenlik standartlarını belirleyen Global-NCAP kuruluşunun önemli alt kuruluşlarından Euro-NCAP'ın belirlediği sürücü, yolcu güvenliğini ve sürücü destek sistemlerinden aldığı puanlara göre üreticiler araçlarını üretebilmekte ve belirlenen puanın altındaki araçların üretimine izin verilmemektedir. Euro-NCAP araçları;

- Yolcu koltuğu güvenliği,
- Çocuk koltuğu güvenliği,
- Yaya güvenliği,
- Güvenlik asistanları

şeklinde 4 ana başlık altında incelemektedir [3].

Güvenlik asistanları her geçen gün yeni eklenen ve yenilenen teknolojilerle sürücü ve yolcu güvenliği standartlarını yukarıya taşımasıyla dikkat çekmektedir. Güvenlik asistanları sürücülere seyir halinde temel seviyede uyarı ve kontrol amaçlarıyla kullanılmaktadır. Bu sistemler oluşabilecek dikkatsizlikleri önceden tespit edip can ve mal kayıplarına sebep olacak kazaların önüne geçmeyi hedeflemektedir.

Dünyaca kabul edilen araç güvenlik sistemlerinden elektronik denge kontrolü, otomatik acil durumda frenleme (AEB), kör nokta takibi, şerit takip uyarı sistemleri (LDWS), hız takibi ve uyarı sistemleri sıkça kullanılan bazılarıdır. Bu teknolojiler sensör, radar, konum, görü ve bu gibi algılayıcıların karmaları kullanılarak geliştirilmektedir.

Her algılayıcının avantajları ve dezavantajları bulunmaktadır. Literatürde yapılan algılayıcı çalışmaları hakkında bilgiler [4]'te detaylı olarak anlatılmıştır. Sensör tipi algılayıcılar, genel olarak fiziksel ortamdan aldığı ölçüm sonuçlarını analiz ederek çalışmaktadır. Araçlarda sıkça kullanılan dahili ve harici sensörlerden alınan verilerin ani değişimine göre kazanın etkisini azaltan sistemler bu tür sistemlere örnek olarak gösterilebilmektedir. Radar tipi algılayıcılar, belirli dalga boylarında yaydıkları ve fiziksel cisimlerden dönen ses, ışık ve radyo sinyalleri üzerinden yapılan ölçüm analizleri ile çalışmaktadır. Popüler olarak uzun menzilli radar ve Lidar sistemler aracın bulunduğu şeritteki konumu ve diğer araçlarla olan mesafelerini hesaplayan sistemler örnek gösterilmektedir. Konum tipi algılayıcılar, uydular ve baz istasyonları gibi dünya üzerindeki lokasyonu oldukça hassasiyetle belirlenmiş noktalar üzerinden mesafe hesaplama sonuçlarını analiz ederek çalışmaktadır. Görü tipi algılayıcılar, kamera gibi görsel sensörlerin oluşturduğu, gerçek dünyadaki cisimlerin görsel öğelerle ilişkilendirilen çıktılarının analiz edilmesiyle çalışmaktadır. Bu tip algılayıcılar diğer sensörlere göre başarı oranları yüksek olması sebebiyle araç teknolojilerinde oldukça sık kullanılmaya başlanmıştır. Karma temelli algılayıcılar, birden fazla algılayıcının birlikte kullanıldığı böylelikle daha yüksek performans elde etmeyi hedefleyen sistemlerdir. Ancak birden çok algılayıcının birlikte kullanılması karmaşıklık problemlerine yol açabilmektedir.

Bu çalışmada araç içi kamera ile görü temelli algılayıcı sistemler kullanılarak Şerit takip uyarı sistemleri üzerine özgün bir çalışma gerçekleştirilmiştir.

2. ŞERİTTEN AYRILMA UYARI SİSTEMLERİ

Şeritten ayrılma tespit sistemleri sürücülerin dikkatsizlik, dalgınlık, uykusuzlukla beraber uykuya dalma gibi istemsiz şekilde oluşan buldukları şeritten ayrılma eylemlerinin sonucunda oluşabilecek kazaları önlemeyi amaçlamaktadır.



Şekil 2.1. Otoyol şerit işaretleri [24]

Otoyollarda bulunan şerit işaretleri tam düz veya kesikli halde Şekil 2.1’de gösterildiği gibi bulunmaktadır. Trafik kurallarına göre araçlar şerit değişimlerini kesik çizgili şeritlerde yapabilmektedir. Şerit takip sistemleri, otoyolda bulunan şerit çizgilerinin konumlarını ve aracın bu şeritlere yaklaşma hızlarını analiz ederek şeritten ayrılma durumları tespit edilebilmektedir. Bu tür şerit değişim durumlarında araç kullanıcıları uyarılarak aracın sabit şeritte bulunması sağlanmaya çalışılmaktadır. Şeritten ayrılma uyarı sistemleri, Avrupa Birliği Komisyonu tarafından geliştirilen otomotiv standardına göre geliştirilmektedir. Bu standarda göre istemsiz şerit değişim hızı 0,1 m/sn – 0,8 m/sn olarak belirlenmiştir. Şeritten ayrılma uyarı sistemleri

60km/saat hızın altında çalışmamakta ve yanal bölgelerdeki şeritlere 30cm'den daha fazla yaklaşıldığında uyarı üretmektedir.

Şeritten ayrılma tespit sistemlerinde çoğunlukla görü temelli sistemler kullanılmaktadır. Bu sistemlerin ilk aşaması şerit tespiti olup ikinci aşamada şeritten ayrılma tespiti yapılmaktadır. Dolayısıyla yol üzerindeki şeritlerin tespiti sistemin başarısını doğrudan etkilemektedir. Görü temelli sistemler, Şekil 2.2'de gösterilen hava şartları, ışık yansımaları, lens ışık yoğunluğunun artması, yoldaki aşınmalar, yoldaki çatlaklar, az ışıklılık, yolda oluşan gölge gibi oluşan durumlardan etkilenebilmektedir. Bu durum sistemlerin düzgün çalışmamasına ve hatta bazı durumlarda aksamasına sebep olabilmektedir. Bu tür sistemler, kalibrasyon ayarlarına çok bağımlıdır ve herhangi bir fiziksel kalibrasyonun bozulmasından olumsuz etkilenmektedir.



Şekil 2.2. Araçlar seyir halinde iken kamera ile alınan bozucu etkilere sahip görüntüler (a) ve (b) Karanlıkta ışık patlaması (c) ve (d) Gölge ve bozucu etkenler

Literatürdeki çalışmalarda kullanılan yöntemler kamera ile alınan bu görümlerin sistemler tarafında işlenmesiyle oluşmaktadır. Bu tür çalışmalar;

- Geleneksel görüntü işleme temelli yöntemler,
- Öğrenme temelli yöntemler

olarak iki ana grupta toplanabilir. Bu çalışmada günümüzde kullanımı oldukça yaygınlaşan öğrenme temelli bir yöntem geliştirilmiştir.

Literatürde geleneksel görüntü işleme temelli yöntemler, [24,55,58,59,61]'deki çalışmalarda ve [4]'teki çalışmada anlatılmıştır. [24]'te yapılan çalışmada geleneksel görüntü işleme yöntemleri için kesikli şerit işaretlerinin devamlı şerit işaretlerinin tespitine göre yarattığı zorluğa çözüm üretmektedir. Kesikli şerit işaretlerinin tespiti için zamanda sürekli yöneme giren çerçevelerin ardışık şekilde alınan görüntülerde bir ortalaması alınarak şerit işaretlerinin görüntüdeki tespit problemlerini çözmektedir. Ortalaması elde edilen imgede ters perspektif dönüşümü yapılarak şerit işaretlerinin girdi imgeye dik ve birbirlerine paralellik özelliği kullanılmaktadır. Yöntemde şerit işaretlerinin tespiti için Hough dönüşümü sayesinde olası şerit işaret noktaları elde edilmektedir. Oluşabilecek yol yüzeyindeki bozucu etmenlerden etkilenmemek amacıyla girdi 1-boyutlu korelasyon işleminden geçirilmekte ve korelasyon çıkışı elde edilen veri RANSAC verilerek olası şerit işaretlerinin tespiti yapılmaktadır.

[55]'daki çalışmada, [56]'de yapılan Hough dönüşümü ile MSER (Maximally Stable Extremal Regions) tabanlı bir şerit tespit yöntemi önerilmektedir. Yöntemde başlangıç olarak MSER ile şerit işaretlerinin tespiti yapılmaktadır. MSER, ürettiği sonuçların tamamında şerit işaretleri bulunmamaktadır. Bu sonuçların bir kısmını elemek için öncelikle yöntemde geliştirilen bir arıtma algoritması ile özellik noktaları elde edilmekte ve bu özelliklerle gerçek şerit işaretlerinin tespiti PPHT (Progressive Probabilistic Hough Transform) [57] ile sağlanmaktadır. Olası şerit işaretlerinin tespit edilememe problemini çözmek adına Kalman fitresi uygulanmaktadır.

[59]'da önerilen yöntemde başlangıçta yol yüzeyi tespit edilmekte ve yol yüzeyi dışındaki olası işlemler engellenmektedir. Bu yol yüzeyi tespiti için Gabor dokusal özellikler (Gabor textural features) ve ortalama-kaydırma kümelemesi (mean-shift clustering) kullanılmaktadır. Şerit işaretlerinin tespiti için sırasıyla yönlendirmeli filtre (steerable filter), IPM, Hough dönüşümü ve MSAC (M-estimator Sample and Consensus) [60] yöntemlerinin kullanılmaktadır. Şerit işaretlerinin tespit edilememe

durumlarını engellemek için genişletilmiş Kalman filtresi (extended Kalman filter) kullanılmaktadır.

[58]'de önerilen yöntemde istenmeyen şerit değişimlerini tespit etmektedir. Yöntemde kameradan alınan görüntüler ilk olarak PLSF (Piecewise Linear Stretching Function) yöntemi ile renkli görüntü gri seviyeli [0, 1] değerleri aralığında gri seviyeli görüntüye çevrilmektedir. Şerit işaretlerinin tespit aşamasında işlemsel yükün azaltılması için belirlenen bir ROI'de, sol ve sağ şerit çizgileri tespit etmek amacıyla bölünmektedir. Ardından Hough dönüşümü yardımıyla şerit işaretleri ile ilgili parametreleri kestirmek için Öklid uzaklığı tabanlı aday özellikler çıkarılmaktadır. Böylece önerilen şerit işaretleri arasındaki değişikliği tanımlanabilmekte ve oluşturulan şeritten ayrılma ölçü değeri yorumlanarak sürücü uyarılmaktadır.

[61]'de önerilen yöntemde mobil bir robot sistem üzerinde gerçek zamanlı Hough dönüşümü ile şerit işaretlerinin tespiti gerçekleştirilmiştir. Yöntemde ilk olarak 1-boyutlu Sobel filtresinin ile eşikleme uygulanmaktadır. Ardından Hough dönüşümü ile girdideki devamlı çizgiler tespit edilmektedir. Bu çizgiler incelenerek şerit işaretlerinin görüntüdeki yeri iki aşamada tespit edilmektedir. Bu inceleme sırasında ilk aşamada şerit işaretlerinin birbirine olan paralellik özelliği faydalanılarak ters perspektif dönüşümü kullanılmaktadır. Bu dönüşümün sonucunda şerit işaretlerinin paralel olması özelliğiyle bu paralel doğrulardaki noktaların birbirine olan uzaklıkları eşit olduğu kabul edilmektedir. İkinci aşamada şerit işaretlerinin genişliği bilinen değerlerde olup olmadığı kontrol edilmektedir. Ters Perspektif sonucu Hough skoru yüksek olan paralel çiftler şerit işareti olarak belirlenmektedir. Paralel çiftler bulunmadığında tek şerit işareti olup olmadığı kontrol edilmekte ve Hough skoru en yüksek değere sahip olan çizgiler şerit işareti olarak tespit edilmektedir.

Literatürde derin öğrenme kullanılarak şerit tespiti yapılan çalışmalar incelendiğinde birçok farklı yaklaşım ile karşılaşılmaktadır. [12]'deki çalışmada, derin öğrenme algoritmalarının yüksek işlem harcama kapasitesini azaltmak amacıyla farklı yaklaşım olarak sistemi iki evrişimsel sinir ağ olarak ele almaktadır. Bu ağların birincisi sınıflandırma amaçlı kullanılmaktadır. İkincisi ağda yapılan şerit sınıflandırma ve şerit regresyon kayıplarının toplamından oluşmaktadır. Diğer nesne tespit algoritmalarına göre sınırlayıcı kutu kullanmak yerine piksel tabanlı nesne tespiti önerilmiştir.

[13]'deki çalışmada, uzun kısa zamanlı bellek (LSTM) metodundan türetilmiş evrişimsel sinir ağlarıyla (CNN) tekrarlayan sinir ağlarını karıştıran bir ConvLSTM yöntemi içeren yöntem önerilmiştir. Bu yöntem kodlama ve kod çözücü bloklarından oluşmaktadır. Yöntemde öncelikle veri setinden ardarda CNNEncoder kodlayıcısına alınan imgeler, girdinin zaman-özellik haritası çıkarılmak üzere girmektedir. Bu kodlayıcı çıkışındaki özellik haritası ConvLSTM tabanlı yöntemle sokulup burada şeri işaretlerinin tespiti tahmini için işlenmektedir.. İşlenen bu imge çerçeveleri CNNDecoder kod çözücüsüne sokularak şeri işaretlerinin tespit tahmini yapılmaktadır.

[14]'deki çalışmada, araçların ön bölümünden bulunan kameradan alınan görüntüler üzerinden yapılan Ters Perspektif Dönüşümü (IPM) sonrası şeri işaretlerinin belirgin renk geçişi kullanılarak öneri haritası oluşturulmaktadır. Bu öneri haritasında elde edilen sonuçlara göre şeri işaretlerinin konumunun bulunması gerçekleştirildikten sonra LSTM tabanlı şeri tespit ağına sokularak eğitim için gerekli olan etiketli veri maliyetini düşürmekte ve daha az işlem yükü sağlayan bir sistem elde etmektedir.

[15]'deki çalışmada önerilen yöntemde araçların yanal bölümlerine yere bakacak şekilde yerleştirilen kameralardan alınan imgeler öncelikle belirli aralıklarla ızgara şeklinde bölünmektedir. Her bir bölünen ızgara kendi içerisinde şeri işaretleri aramaktadır. Eğer şeri işareti bulursa, işaretin imgedeki oryantasyonunu ve bulunduğu pozisyonu belirlemektedir. Bu yöntem sayesinde bulunan şeri işaretlerinin pozisyonlarını yüksek doğrulukla bulabildiği iddia edilmektedir.

[16]'da önerilen çalışmada, hem şeri işaretleri hem de trafik işaretleri tespit edilmesine odaklanılmıştır. Yöntemde bir aracın etrafını 360 derece kapsayacak şekilde konumlandırılmış kameralardan alınan görüntüler üzerinde çalışılmıştır. Bu yöntem görüntüler üzerinde öncelikle birden fazla ağa yardım edebilen bir evrişimsel sinir ağı geliştirilmiştir. Bu geliştirilen ağ hem şeri işaretlerinin tespitinde, hem de trafik işaretlerinin tespitinde rol almaktadır. Bu evrişimsel sinir ağının çıkışında bir tekrarlayan sinir ağı bulunmaktadır. Bu tekrarlayan sinir ağı geçmişten bilgiler bulundurarak tespit sonuçlarının doğruluğunu artıran bir rol almaktadır.

[17]'deki çalışmada araç etrafına yerleştirilmiş 360 derece kameralar ile araç park yeri problemlerini çözmesi için şeri işaretlerinin tespit temelli yöntem önermiştir. PSV

veri setleri kullanılarak geliştirilen bu yöntemde ilk olarak CNN katmanı kullanılmakta ve çıkışında VH-stage olarak adlandırılan dikey ve yatay sınırlandırma ağı bulunmaktadır. Bu iki ağın çıkışı toplayıcılar ve bir dizi CNN ile çıkışta sonlandırılmaktadır. Çıkıştaki görüntüde bu ağ sayesinde dikey ve yatay şekilli şerit işaretleri tespit edilebilmektedir.

[18]'de önerilen çalışmada giriş imgesi (640×480) çok küçük kafes şeklinde sınırlayıcı kutulara (4×4) bölütleme yapılmaktadır. Bölütlenen r kafes şeklindeki sınırlayıcı kutular yöntem geliştiricileri tarafından toplanan etiketli verilerle evrişimsel sinir ağına girilmektedir. Her bir kafes etiketli verinin kendi içerisinde bulunup bulunmadığını belirlemektedir. Bu evrişimsel sinir ağı çıktısına göre şerit işaretinin bulunduğu bitişik sınırlayıcı kutular aynı şerit işaretini tasvir ederek şerit işaretlerinin tespiti yapılmaktadır.

[41]'de önerilen çalışmada bir aracın ön bölümüne yerleştirilmiş açılı bir kameradan alınan görüntüyle şerit işaretlerini ve yol yüzeyini tek imgeden üç boyutlu uzayda tespit edebilmektedir. Yöntem sırasında sentetik üç boyutlu şeritler olarak adlandırılan veri seti oluşturulmuş ve kullanılmıştır. Yöntemde iki ağ kullanılmaktadır. Ham imge CNN temelli geliştirilen bir ağı girerek yol yüzeylerinin tespitini sağlamaktadır. Geliştirilen bu ağın belirli boğumlarından paralel şekilde alınan çıkışlar, ters perspektif dönüşümüyle kuş bakışı görünüme projekte edilmektedir. Dönüşüm sonrasında başka bir CNN temelli geliştirilen ağ kuş bakışı görüntüler üzerinde çalışmaktadır. Üretilen her çıkış kılavuzlu biçimde şerit işaretlerinin öğrenilmesini sağlamaktadır. Bu sürülen ağ şerit işaretlerinin tespitini üç boyutlu olarak tespit edebilmektedir. Bu yöntemle 0-30 metre aralığında doğrusal olan çizgiler üretilmekte ancak 30-80 metre aralığında üç boyutlu uzaydan şerit işaretlerinin tespiti ile doğrusal olmayan eğik şekilde şerit işaretleri üretilmektedir.

3. YAPAY ZEKA

Yapay zekanın tarihi oldukça yeni olmakla beraber 1950’li yıllarda Alan Turing ile başlayan serüven ilk 1956 yılında yapılan ilk yapay zeka konferansıyla yol almaya başlamış olup, 2002 yılında geliştirilen bir otonom süpürge engellere takılmadan çarpmadan temizlik yapabirmiştir. Bu tür gelişmelerden sonra son 15 yılda Google, Amazon, Nvidia ve Tesla gibi firmalar bu alandaki potansiyeli görerek yapay zeka teknolojilerine milyarlarca dolar yatırım yapmıştır. 2017 yılında Google’ın geliştirdiği bir yazılım olan AlphaGo, uluslararası oyun olan Go oyun sanatında dünyanın en iyi oynayan insanını yenmesi ve Tesla’nın geliştirdiği tam otonom otomobillerin sokaklarda görmemiz yeni bir çağ açmıştır.

Yapay zeka, günümüzde geleneksel çözümlerin yetersiz kaldığı yerlerde insan gibi deneme yanılma yöntemiyle sistemler hakkında bilgi edinebilen, hatalardan ders çıkararak görevleri öğrenmeyi başarabilen ve kendisini her geçen gün geliştirebilen bir teknolojidir. Yapay zeka genel olarak bilgisayar gibi bir makinenin insana özgü olduğu düşünülen bir konuda ya da genel bağlamda anlam çıkarma, fikir yürütme, belirli sezilere cevap verme gibi yetenekleri yerine getirebilen metodlar ve algoritmalar topluluğu olarak anılmaktadır. Yapay sinir ağları (YSA), geleneksel yöntemlerin aksine doğrusal olmayan veri odaklı yaklaşımlardır. YSA’lar insan beyninin öğrenme sırasında karmaşık ve doğrusal olmayan verilerin analizini yapması gibi girdi verilerin altında yatan veri ilişkisini tanımlayabilir öğrenebilir, çıktılarla arasında bağlar kurabilir. Bu gibi özelliklerinden dolayı günümüzde tercih edilmektedir [50].

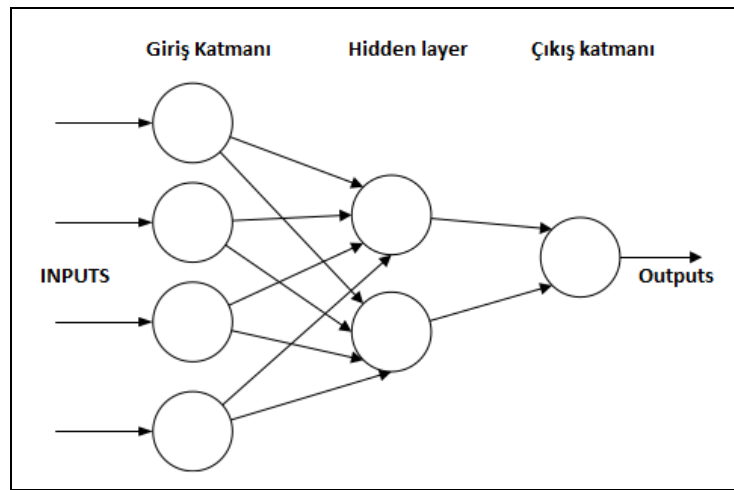
Yapay zeka ile geliştirilen metodlar ve algoritmalar, sadece bir sisteme özel olmayıp aynı metoda cevap verebilen birden fazla sistem bulunabilmektedir. Buna en iyi örnek olan aynı metodla kullanılan sistemler, kimi zaman bir uzay aracının yere inişini hesaplamada kimi zaman bir saç tanesinin büyüme oranını hesaplamada kullanılmaktadır.

Yapay sinir ağları, geliştiriciler tarafında insan sinir hücrelerinin yapısından ilham alınarak geliştirilmiştir. İnsan vücudunda ve beyinde bulunan sinir ağı yapısından

esinlenerek geliştirilmiştir. Nispeten insan beyninin elektronik modellerine benzemektedir. Yapay sinir ağları, veri setlerini sınıflandırmak, anlamlandırmak, tahminde bulundurmak gibi doğrusal olmayan işlevlerin yerine getirilmesini sağlayan ampirik yapılar olarak görülmektedir.

Günümüzde yapay sinir ağları insan beyninden daha az karmaşık olduğu kabul edilmektedir. Yapay sinir ağları birbirlerine birebir bağlanmış birden fazla oluşabilen bağların belirli boğumlarda birleşerek girdilere cevap üreten yapılardır.

Yapay sinir ağı, birbirlerine bağlı ve birbirleri arasında bilgi alışverişi yapan nöron yapıları barındırmaktadır. Bu nöronlar birden fazla özelliği algılayan ağ katmanlarından oluşmaktadır. Her katman kendinden önceki katmandan aldığı farklı girdilere cevap veren bir çok nörona sahiptir. Birden fazla oluşan katmanlar, çok katmanlı sinir ağları (multi-layer feedforward networks) olarak adlandırılmaktadır. Bu tip sinir ağları, Şekil 3.1'de gösterildiği gibi üç katmandan oluşmakta ve bunlar girdi (input layer), gizli (hidden layer), çıktı (output layer) katmanlarıdır. Girdi bölümü genellikle büyük vektör biçiminde ham bilgiler bulundurmaktadır. Girdi katmanı, gizli katmana bu ham bilgileri aktarmaktadır. Girdi ve çıktı katmanı arasındaki bölüm “gizli katman” olarak adlandırılmakta ve gizli katman n adet katmandan oluşabilmektedir. Gizli katman, girdi katmanındaki bilgileri ve özellikleri çıkararak öğrenmeyi sağlamakta ve çıktı katmanına bu bilgileri anlamlandırmaya çalışmaktadır. Çıktı katmanı, girdilere cevap alınan katmandır [52, 53].



Şekil 3.1 Basit bir yapay sinir ağı [51]

Yapay zeka, aşağıdaki gibi kullanım alanlarına sahiptir.

- Doğal dil işleme [43],
- Konuşma [44],
- Planlama, zamanlama, optimizasyon [46],
- Robotik [47],
- Görü [45]

Makine öğrenmesi, kendi içerisinde Derin öğrenme ve Tahmine dayalı analitik olarak ayrılmaktadır. Doğal dil işleme, kendi içerisinde çeviri, sınıflama, kümeleme, bilgi çıkarma olarak ayrılmaktadır. Konuşma, kendi içerisinde konuşmadan yazıya, yazıdan konuşmaya çevirme olarak ayrılmaktadır [48]. Bu tez çalışmasında yapay zeka türlerinden makine öğrenmesine odaklanılmıştır.

3.1. Makine Öğrenmesi

Makine öğrenmesi adını sıkça duyduğumuz, gitgide teknolojinin her alanında etkin şekilde kullanılmaya başlanan, geleneksel lineer yöntemlere alternatif olarak geliştirilmiş ve yapay zekanın ilk örneklerinin görüldüğü bilim dalıdır. Amacı geleneksel yöntemlerin başarısız, sistemsiz ve işlevsiz olarak yetersiz kaldığı durumların üstesinden gelebilmektir.

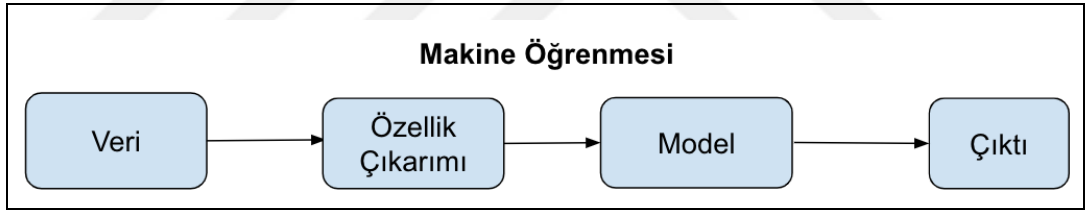
Bazı makine öğrenmesi yöntemleri, insan kadar hızlı ve geçmişte yapılan tercihlerden ders alarak sonuçlar çıkarabilmektedir. Geleneksel yöntemler kullanılan sistemlerde başlangıç zamanında verdiği sonuçla, belirli bir süre geçtikten sonra verdiği sonuç zamanda değişmez ve doğrusal olmaktadır. Ancak makine öğrenmesi yöntemleri kullanılan sistemlerde eğer hata oranı yüksekse aynı zamandaki girdilere aynı sonuçları tahmin edemeyebilmektedir. Bu durumun oluşmasını sağlayan aşağıdaki gibi etmenlerden dolayı doğru yöntemin geliştirilmesi önem arz etmektedir.

- Yöntemin yeteri kadar efektif sonuç üreten bir metoda sahip olmaması,
- Eğitim sırasında yöntemin yeteri kadar doğrulukla eğitilememesi,
- Yeteri düzeyde öznitelliklerinin çıkarılamaması,
- Yöntemin sınıflandırıcı ve bölütleyici gibi metod parametrelerinin doğru ayarlanamaması,
- Yeterli verinin olmaması

Makine öğrenmesi Makine öğrenmesi kendi içerisinde

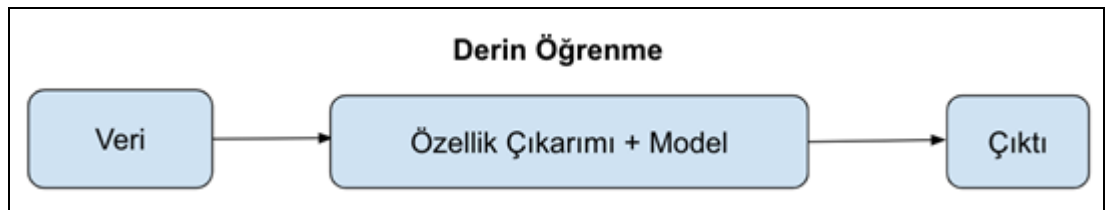
- Gözetimli öğrenme (Supervised Learning),
- Gözetimsiz öğrenme (Unsupervised Learning),
- Pekiştirmeli öğrenme (Reinforcement Learning),
- Yarı denetimli öğrenme (Semi-supervised Learning)

şeklinde türlerine ayrılmaktadır. Bunlardan gözetimli öğrenme, eğitim için hazırlanan etiketlenmiş veri girdilerinin belirli bir sınıfa, kümeye ait olduğunu (classification) ya da girdilerin belirli bir aralık içerisinde bulunduğunu (regression) belirlendiği öğrenme türüdür. Gözetimsiz öğrenme, eğitim için veri girdilerinin hazırlanmadığı ve tamamen girdiler üzerinden tahmine dayalı eğitim (clustering) ile öğrenme türüdür. Pekiştirmeli öğrenme ise belirli bir amaca odaklı eğitim girdilerinin eğitim aşamasında elde edilen veri girdilerine göre alınan çıktılara dayalı eğitim ile öğrenme türüdür. Yarı denetimli öğrenme denetimsiz ve denetimli öğrenmenin arasında bir öğrenme türüdür. Daha çok etiketlenmemiş verilerden denetim sağlanarak öğrenmeyi sağlamaktadır.



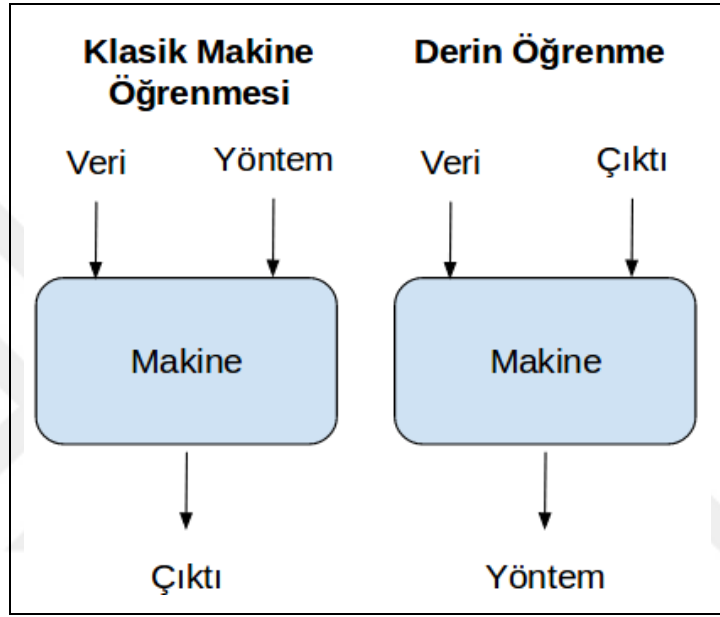
Şekil 3.2. Makine öğrenmesinin çalışma blokları

Makine öğrenmesinde eğitim süreci basit olarak Şekil 3.2’te gösterildiği gibi öncelikle veri girdisi sisteme sokulur ve özellik çıkarımı işlemi gerçekleştirilir. Ardından geliştirilen modele sokularak bir girdiye üretilen çıktı elde edilmiş olur. Ancak bu durum derin öğrenme eğitim sürecinde farklı ilerlemektedir. Derin öğrenme yöntemlerinde eğitim süreci Şekil 3.3’te gösterildiği ekstra bir özellik çıkarımına aşamasına gereksinim duyulmadan gerçekleştirilmektedir.



Şekil 3.3. Derin öğrenmenin çalışma blokları

Günümüzde üreticiler ürün gamlarını artırabilmek için sistemlerine sürekli yeni algılayıcılar eklemekte ve bu da sistemleri daha da karmaşıklştırmaktadır. Bu sebeple karma algılayıcılar oldukça yaygın hale gelmektedir. Derin öğrenme, bu tür karmaşıklştan sistemlerde klasik makine öğrenmesi yöntemlere göre çoğu yönden ön plana çıkmaktadır.



Şekil 3.4. Klasik makine öğrenmesi yöntemleri ile Derin öğrenme yöntemlerinin çalışma prensipleri

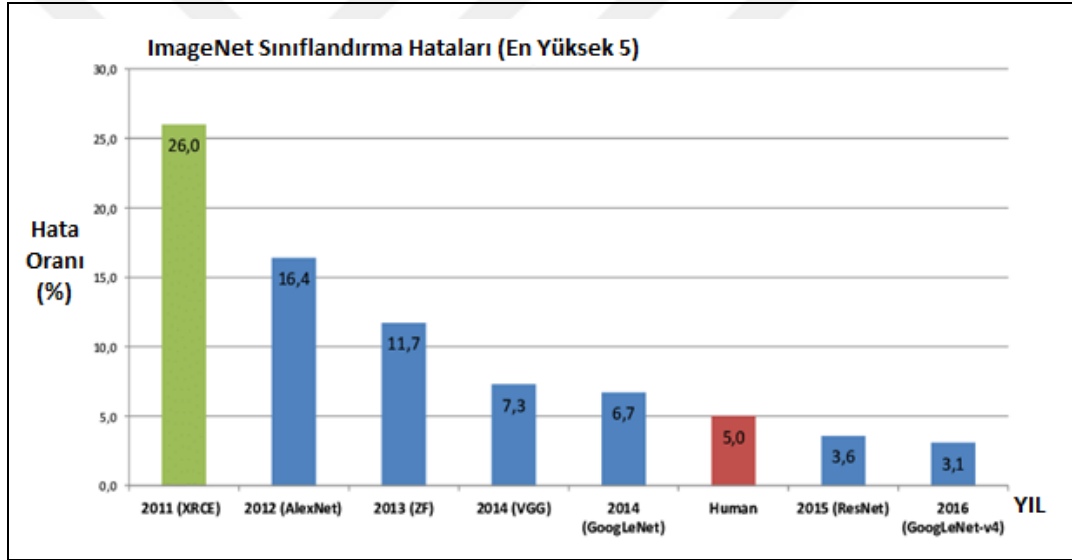
Şekil 3.4’de gösterildiği gibi klasik makine öğrenmesi yöntemlerinde belirli veri girdileri ile geliştirilen yöntem sayesinde makine ilgili çıktıları üretmektedir. Ancak bu işlem derin öğrenmede sistemin ilgili girdi ve çıktıları üreten yöntem belirli olmadığı için öncelikle girdi ve çıktılarından bir yöntem elde edilmektedir. Bu süreç yöntemin öğrenildiği eğitim aşaması olarak adlandırılmaktadır. Eğitim sonucunda elde edilen yöntemle klasik makine öğrenmesi yöntemleri aynı süreçte devam etmektedir. Bu noktada ilk süreç olan eğitim aşamasında elde edilen yöntemin girdilere göre çıktıları karşılayan bir yapıda olması önemlidir [62].

Kullanılacak yöntemde eğer bir yapay sinir ağı uygulanacaksa, ağ uygulanacak probleme ve problemin zorluğuna göre belirlenmesi gerekmektedir. Bu durum hem başarıyı hemde performansı etkilemektedir. Bu çalışma kapsamında literatürde yapılan çalışmalar incelendiğinde derin öğrenme yöntemleri şerit takip sistemlerinde

daha çok tercih edildiği gözlemlenmiştir [12-24]. Derin öğrenme yöntemleri, klasik makine öğrenmesi yöntemleri özellik çıkarımı konusunda ekstra iş yükünden kurtaracağı da düşünülerek bu çalışmada tercih edilmiştir.

3.2. Derin Öğrenme

Günümüzde hayatın her anında anlamlı veya anlamsız veri kümeleri ve karmaşık verilerle karşılaşmaktadır. İnsan bilgi işleme mekanizmaları net bir şekilde katmanlı hiyerarşik yapılarla donatılmıştır. Bu yapılar karmaşık bilgileri anlamak ve bu bilgilerden anlamlı sonuçlar oluşturmak için derin mimarilere ihtiyaç duymaktadır. Tıpkı insan beyninde olduğu gibi derin öğrenme yapıları, giriş ve çıkış arasındaki katman dizilerini öğrenme ve işleme özelliğine sahiptir [54].

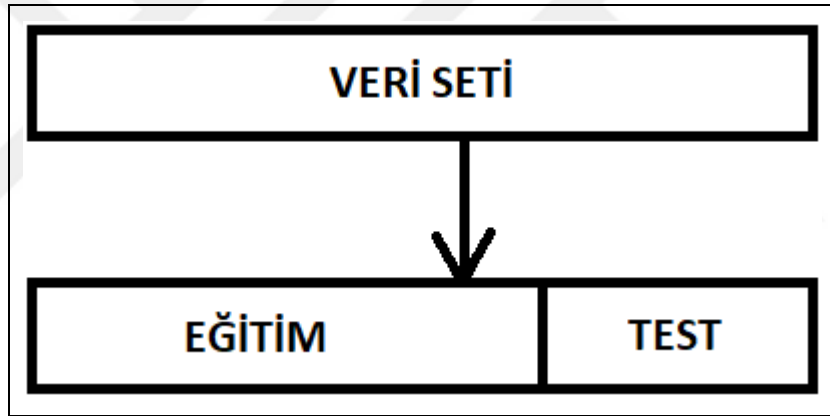


Şekil 3.5. ImageNet yarışmasında 2011-2016 yılları arasında alınan sonuçlar [6]

Derin öğrenmenin bu kadar gelişmesinde en önemli etken ImageNet [5] yarışmasıdır. ImageNet ILSVRC (ImageNet Large Scale Visual Recognition Competition), 2010 yılında makine öğrenmesi biliminin gelişmesi amacıyla başlatılan bir yarışmadır. Bu yarışmada katılımcılardna PASCAL-VOC [10] veri setinin üzerinde nesne tespiti ve sınıflandırma yapabilecek algoritmaların geliştirilmesi beklenmektedir. 2012 yılına kadar geleneksel makine öğrenmesi temelli yaklaşımlarla elle özellik çıkarımı temelli çözümler geliştirilmesinin ardından 2012 yılında ilk derin öğrenme temelli çözüm oldukça yüksek bir performans artışı sağlayarak ortaya çıkmıştır. Yarışmada sonraki yıllarda derin öğrenme odaklı algoritmalar geliştirilmiştir.

Şekil 3.5'te ImageNet yarışmasının sınıflandırma kategorisinde alınan hata oranlarının 2011-2016 yıllarında başarılı olan metodlar gösterildiği gibi yıldan yıla ortalama büyük bir ivmeyle hata oranları düşmüştür. 2015'te ResNet [7] ve 2016'da GoogleLeNet-v4 [8] ile alınan sonuçlar bir insanın karar verirken yaptığı hatanın da altına düşmüştür. Bu gibi sonuçlar sayesinde derin öğrenme alanına ilgi büyük bir hızla artmış ve gelişmiştir.

Derin öğrenme algoritmalarının geliştirilmeleri sırasında üç süreçten geçilmektedir. Eğitim, modelin etiketli verilerle eğitilerek sisteme en uygun ağı bulunmasına ve bulunan ağı en iyi modeli oluşturmasını sağlamaktadır. Doğrulama, eğitim sırasında geliştirilen ağı geriye yayılım (back propagation) anında test edilmesine ve ağı parametrelerinin hassas şekilde ayarlanabilmesini sağlamaktadır [63]. Test, geliştirilen modelin ne kadar doğrulukla oluşturulduğunu belirlenmesini sağlamaktadır.

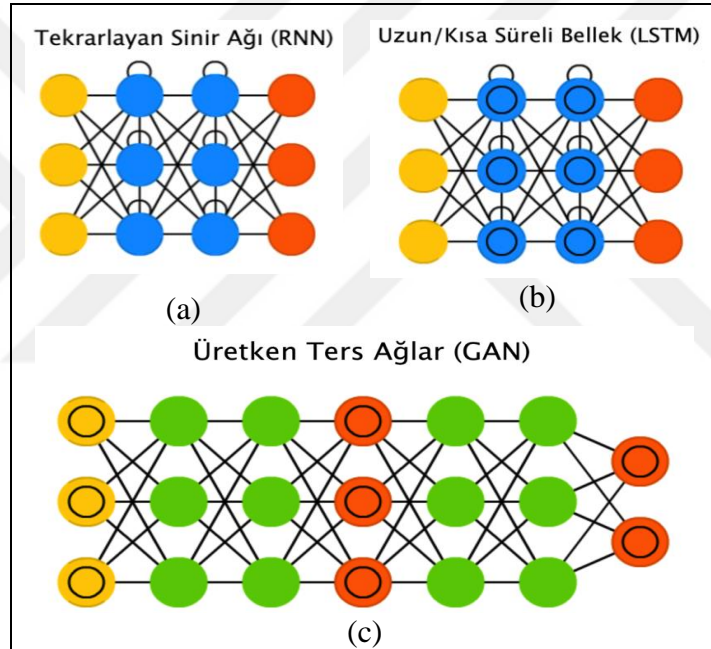


Şekil 3.6. Hold-out yöntemi toplam veri setini eğitim ve test veri setlerine bölünmesi

Geliştirilen ağı modelinin doğruluğunu belirleyen yöntemlere göre veri setinin kullanım şekli farklılık gösterebilmektedir. Literatürde kullanılan Hold-out, Cross validation, Leave-one-out metodları doğrulama sürecinde kullanılmaktadır [35-36]. Bu tez çalışmasında veri setlerinin öğrenme süreçlerinde Şekil 3.6'da gösterildiği gibi hold-out doğrulama yöntemi kullanılmıştır. Bu doğrulama yöntemine göre toplam veri setinin %75'lik bölümü ile eğitim, %25'lik bölümü ile test yapılmaktadır.

Derin öğrenme, yapay zekada kullanılan sinir ağı yapıları gibi birbirlerine sinapslarla bağlanmış yapay sinir ağlarını kullanmaktadır. Ancak yapay zekada ve makine öğrenmesinde kullanılan yapay sinir ağları genellikle birkaç gizli katmandan

oluşmaktadır. Derin öğrenme, genellikle derin sinir ağları (DNN) olarak adlandırılan birçok gizli katmana sahip, ileri beslemeli mimariye sahip modellerdir [54]. Derin öğrenmede, genellikle oldukça fazla gizli katman kullanılmaktadır. Bu gizli katmanlar girdilerle, çıktılarla ve birbirleri arasında birebir sinaps bağlantısına sahiptirler. Ancak derin öğrenmede kullanılan sinapslar gizli katmanlara birebir bağlama çok tercih edilmemektedir. Derin öğrenme de gizli katmanlar ve bu katmanların sinapslar yardımıyla birbirlerine bağlanma şekilleri metodun vereceği sonucu çok büyük etkilemekte ve en iyi sonucu veren (en az hata) model eğitim sürecinde belirlenmektedir. Bu da eğitim sürecinde oluşabilecek varyasyonlar ve bu varyasyonların metodun geliştirilme süresinin artmasına sebep olmaktadır.



Şekil 3.7. (a) Tekrarlayan sinir ağı (b) Uzun kısa süreli bellek (c) Üretken ters ağlar [9]

Şekil 3.7’de derin öğrenme yöntemlerinde en çok tercih edilen yapay sinir ağlarının kullanım çeşitleri bulunmaktadır. Solda bulunan tekrarlayan sinir ağı (RNN), klasik yapay sinir ağlarının eğitimi sırasında kullandığı ileri yönlü besleme (feed-forward) yapısındaki gibi etiketli girdinin ağı girdikten sonra sadece çıkıştaki sonuçla karşılaştırılarak eğitim sürecinin tamamlanamamaktadır. Tekrarlayan sinir ağlarında eğitim aşaması sırasında ağın t anında aldığı çıktılar $t - 1$ anında girdilere aldığı çıktılardan kararlardan etkilenebilmektedir.

Uzun/Kısa Süreli Bellek (LSTM) türü sinir ağlar bir tekrarlayan sinir ağı (RNN) türüdür. Uzun vadede RNN'in çözemediği önceki kararlara fazla bağımlılığı anlamakta ve bu bağımlılığı azaltmaktadır. Üretken ters ağlar (GAN), iki yapıdan oluşmaktadır. Birinci yapı üretici (generator), eğitim sırasında girdiye göre sanal sonuçlar üretmektedir. İkinci yapı ayırt edici (discriminator), üretici tarafından üretilen sonuçların gerçek olup olmadığını belirlemektedir.

Literatürde, şeritten ayrılma tespit sistemleri için tüm bu yapılar kullanılarak metodlar ve modeller geliştirilmiştir. Şeritten ayrılma uyarı sistemlerinde derin öğrenme ile görüntü işleme metodları geliştirildiği sistemlere genel olarak;

- Sınıflandırma (Classification),
- Nesne lokalizasyonlama (Object Localization),
- Nesne tespiti (Object Detection),
- Segmentasyon (Segmentation)

gibi türlerde uygulanmaktadır. Burada kullanılan her bir yapı kendi içerisinde farklı avantajlar ve dezavantajlar sağlamakla beraber ortak problemlere de çözüm olabilmektedir. Bu yaklaşımların avantajları ve dezavantajları sırasıyla aşağıda açıklanmıştır.

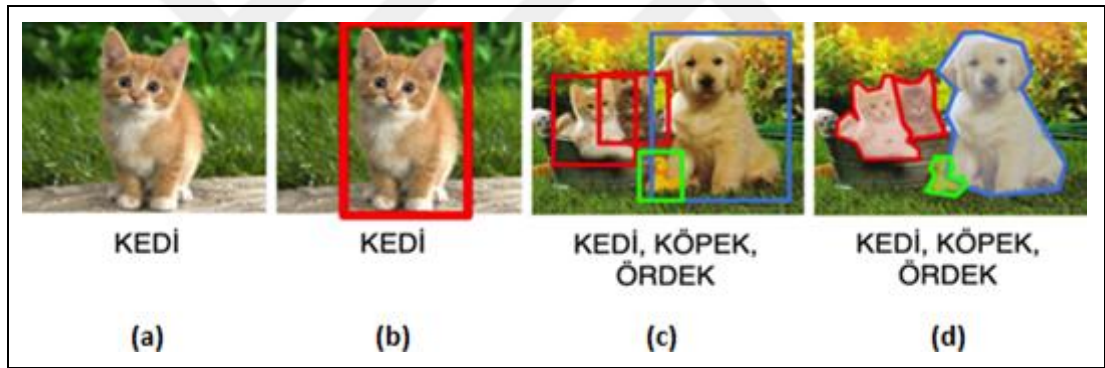
Sınıflandırma, girdi imgede şerit işareti bulunup bulunmadığını belirlemede kullanılmaktadır. Sınıflandırma çoğunlukla sınıfların içerisinde imgenin sadece bir sınıfa ait olduğu uygulamalar kullanılmaktadır. Birden fazla sınıfın çözümü olarak çok etiketli sınıflandırma metodu kullanılmaktadır. Oldukça hızlı ve doğru sonuçlar üretebilmektedir. Ancak şeritten ayrılma sistemlerinde çok tercih edilmemektedir. Bunun sebebi şerit işaretlerinin bulunduğu yerleri detaylı şekilde verememesi ve birden fazla şerit işareti tespit edememesidir.

Nesne lokalizasyonu nesnenin imgenin hangi pikselleri arasında olduğu sınırlayıcı kutularla (bounding boxes) tespit edilebilmektedir. Nesne lokalizasyonu çoğunlukla sınıflandırma ile beraber kullanılmaktadır. Bu tür uygulamalar genellikle bir imgedeki bir nesnenin hangi pikseller arasında bulunduğunu belirleyen uygulamalar olmaktadır. Ancak şeritten ayrılma sistemlerinde kullanılamamaktadır. Sınıflandırma ve

lokalizasyonlama ile beraber kullanılırken sadece bir şeritin tespiti ile uygulamalarda model ağıın en az iki kez çalıştırılacak olması tercih edilmeme sebep olmaktadır.

Nesne tespiti yaklaşımları imgedeki nesnelere her birinin sınıflandırıp hangi pikseller arasında bulunduğunu sınırlayıcı kutularla ayrı ayrı belirlemektedir. Nesne sınıflandırmayla beraber kullanılan lokalizasyonlamadan farkı birden fazla sınıfın bulunduğu piksellerin sınırlayıcı kutularla tespit edilmesidir. Oldukça hızlı modeller bulunmaktadır. Bu tez çalışması çerçevesinde, nesne tespiti yaklaşımı ile şerit işaretlerini tespiti üzerine çalışılmıştır.

Segmentasyon (bölütme), girdi imgedeki her bir pikselin hangi sınıfa ait olduğunu belirlemektedir. Segmentasyon temelli yaklaşımlarda piksel tabanlı eğitim gerçekleştirilmektedir. Bu tez çalışmasında tercih edilmemesinin en büyük sebebi de budur. Ancak yöntem olarak yüksek performans verebilmektedir.



Şekil 3.8. Derin öğrenme metodlarının çeşitlerine göre çalışma şekli. (a) Nesne sınıflandırma (b) Nesne lokalizasyonlama (c) Nesne tespiti (d) Nesne segmentasyonu [11]

Bu derin öğrenme metodlarının uygulanmasına Şekil 3.8'de gösterildiği gibi Şekil 3.8 (a) imge sınıflandırma, Şekil 3.8(b) nolu imge nesne sınıflandırma ve lokalizasyonlama, Şekil 3.8(c) nolu imge nesne tespitleme, Şekil 3.8(d) nolu imge piksel tabanlı nesne segmentasyonlamaya örnek olarak gösterilmektedir.

3.2.1. Nesne tespiti

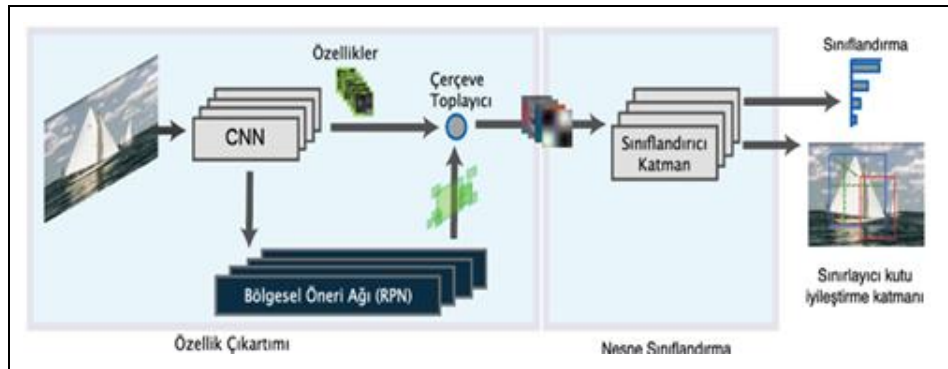
Nesne tespiti yaklaşımları bir objenin imgedeki yerini ve hangi sınıfa ait olduğunu belirlemektedir. Ancak sınıflandırma ve lokalizasyonlama yaklaşımlarının beraber kullanıldığı gibi kullanılmamaktadır. Sınıflandırma ve lokalizasyonlama bir imgedeki

tek objeyi sınıflandırıp lokalizasyonlamaktadır. Ancak nesne tespiti bir imgedeki birden fazla belirlenmiş sınıfa ait nesneyi sınıflandırıp, imgedeki pozisyonunu tespit edebilmektedir. Aslında nesne tespiti diğer bir deyişle çok etiketli sınıflandırma ve lokalizasyonlama olarak da adlandırılabilir. Eğitim ve test aşamalarında hızlı sonuçlar görülebilmekte ve eğitim sırasında oluşturulan veri setinin oluşturulması segmentasyon yöntemlerine göre oldukça kolay olmaktadır. Tespit sonuçlarını ağır eğitim çıkışında belirlenmiş sınırlar içerisinde hem objenin sınıfını hem de objenin imgede bulunduğu pozisyon bilgisi elde edilmektedir.

Literatürde yakın geçmişte önerilmiş ve nesne tespiti için yüksek başarımlar sağlayan;

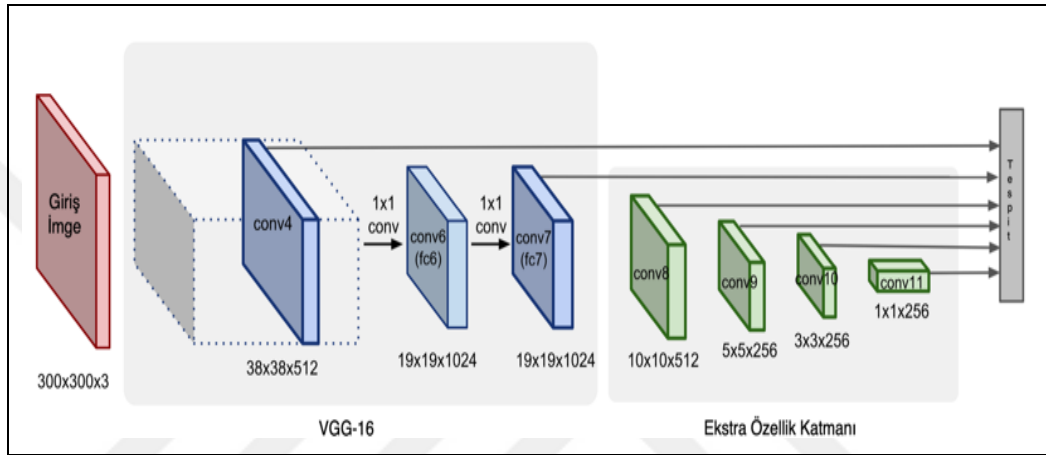
- Faster R-CNN [19],
- Single Shot MultiBox Detector (SSD) [20],
- You Only Look Once (YOLO) [21,22]

gibi birçok yöntem bulunmaktadır. Bu yöntemler nesne tespit oldukça başarılı ve en çok tercih edilen derin öğrenme ile nesne tespit yöntemleridir. Faster R-CNN, kendisinden önce gelen Fast R-CNN'e göre oldukça hızlı ve karmaşıklığı azaltılmış nesne tespit yöntemidir. Bu nesne tespit yöntemi Şekil 3.9'da gösterildiği gibi öncelikle veri seti evrimsel sinir ağları ile öz nitelikleri çıkarılmakta ve diğer yandan bu özelliklerin yerlerini tespit etmek amacıyla bölgesel öneri ağı ile paylaşılmaktadır. Bölgesel öneri ağının çıkardığı bilgiler özelliklerle birleştirilerek özelliklerin imgenin hangi bölgesinde olduğu belirlenmektedir. Özellik çıkarımı evresinde elde edilen sonuçlar sınıflandırıcı katmana sokularak imgedeki objeler sınıflandırılmakta ve sınıflandırıcı katman çıkışındaki sınırlayıcı kutular, iyileştirme katmanını çıkışında objenin nerede bulunduğu tespit edebilmektedir.



Şekil 3.9. Faster R-CNN yönteminin blok diyagramı [38]

SSD yöntemi, Faster R-CNN'in gerçek zamanlı çalışma için yeterli olmadığı durumlarda ve düşük çözünürlüklerde daha hızlı sonuç üretmek amacıyla geliştirilmiş nesne tespit yöntemidir. SSD Şekil 3.9'da gösterildiği gibi ağ Faster R-CNN gibi RPN kullanmadan VGG-16 ağının çıkışında geliştirilen varsayılan sınırlayıcı kutular evrimsel sinir ağıla eğitimi sayesinde doğrudan bu varsayılan kutulardan alınan ara sonuç çıktıları son katmana ileterek özellik haritası elde edilmektedir. Böylelikle özellik çıkarımıyla uzun zaman kaybetmeden nesnenin tespiti mümkün olmaktadır.



Şekil 3.10. SSD yönteminin blok diyagramı [39]

YOLO, diğer derin öğrenme ile nesne tespit yöntemlerine göre popüler yöntemdir. Bunun sebebi oldukça basit ve diğer yöntemlerin eksik kaldığı yanları kapatmaktadır. YOLO yönteminin detaylarına bir sonraki alt başlıkta yer verilmiştir.



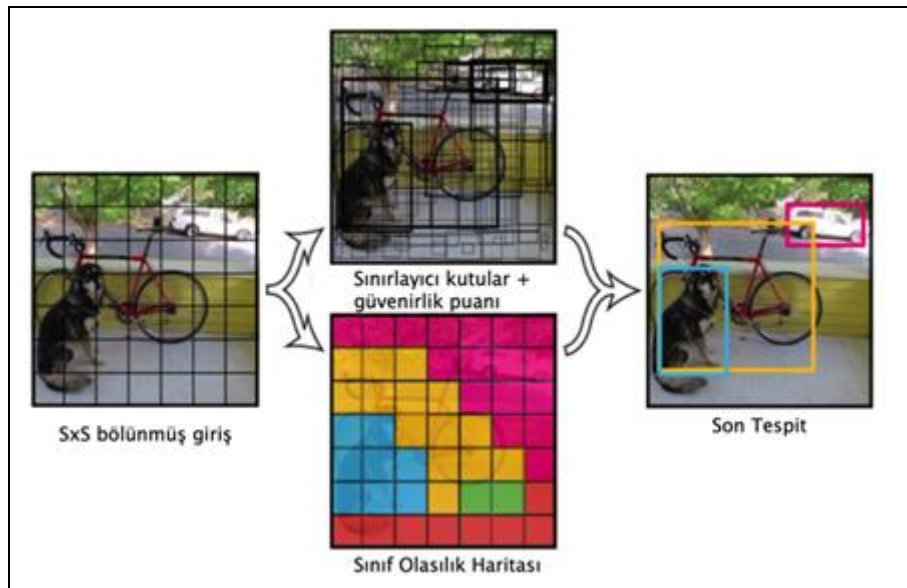
Şekil 3.11. Önerilen nesne tespit yöntemlerinin birbirlerine göre karşılaştırılması [23]

Bu çalışmada geliştirilen şeritten ayrılma sistemlerine en iyi sonuç verebileceği düşünülen YOLO 'nun tercih edilme sebebi aşağıda sıralanmıştır.

- Faster R-CNN, diğer iki yönteme göre hata oranı oldukça düşük olmasına rağmen yüksek seviyede doğruluk elde edebilmek için fazla zaman kaybetmesinden dolayı gerçek zamanlı sistemlerde uygulanması mümkün olmayan bir yöntem olmasıdır.
- SSD, Faster R-CNN yöntemine alternatif olarak geliştirilmesine rağmen YOLO kadar optimum hız ve doğruluk oranında bulunmamasıdır.
- YOLOv3 versiyonunun önceki versiyonlarındaki eksik yanlarını kapatan bir güncelleme almasından dolayı başarı ve hız olarak yeterli olmasıdır.

3.2.1.1. You only look onces (YOLO)

YOLO, literatürde bulunan derin öğrenme temelli nesne tespiti yöntemlerine benzer şekilde evrimsel sinir ağlarından faydalanmaktadır. Ancak diğer derin öğrenme yöntemlerinden farklı olarak girişteki imge tek seferde ağdan geçirilerek çıkışta nesne tespiti sağlanmaktadır. YOLO, Faster R-CNN ve SSD'ye kıyasla mAP oranı ile yaklaşık başarımlar sağlamakta ve gerçek zamanlı nesne tespiti sağlayabilmektedir [22]. Bunun sayesinde diğer yöntemlerden daha popüler ve tercih edilmesine neden olmaktadır.



Şekil 3.12. YOLO 'nun genel çalışma prensibi [21]

Şekil 3.12’de YOLO ’nun nesne tespiti yaparken kullandığı genel yapısı gösterilmiştir. Öncelikle giriş imgesi $S \times S$ ($S = 7$) boyutlarında ızgara kutu şeklinde bölütlenmektedir. Ardından her bir kutu kendi içerisinde bir güven oyu ve içerisinde nesnenin merkezini bulunuyorsa nesnenin yatay düzlemdeki x ve dikey düzlemdeki y koordinatlarını bulundurmaktadır. Ayrıca içerisinde merkezi bulunan nesnenin yükseklik, genişlik bilgisini ve nesnenin tespit edildiği sınıfa benzerlik oranını tutmaktadır. Güvenoyu geçerli olduğu ızgara içinde nesne olup olmadığını bulduktan sonra bulunan nesnelerin etiketli eğitim nesneleriyle ne kadar kesiştiğini (IoU) belirlemektedir. Tespit edilen objenin merkezi eğer var olan sınırlayıcı kutu içerisinde ise x , y , yükseklik, genişlik ve benzediği sınıf bilgisini tutmaktadır. Bu yaklaşım sayesinde bir nesnenin görüntüde bulunduğu yeri ve nesnenin belirlenen sınıflar içerisinde hangi sınıfa ait olduğu bulunabilmektedir.

YOLO’nun ilk versiyonunda sabit bölütlenmiş kutular tek bir nesnenin koordinatları ve sınıf bilgisini tutmaktadır. YOLOv2’de Faster R-CNN’de kullanılan yardımcı kutulara (anchor box) benzer şekilde birden fazla nesnenin merkezinin bulunduğu bilgisini tutmaktadır. Böylelikle YOLOv2 ile sabit bölütlenmiş kutuların tek bir sınıf bilgisi tutulması probleminin önüne geçilmiştir.

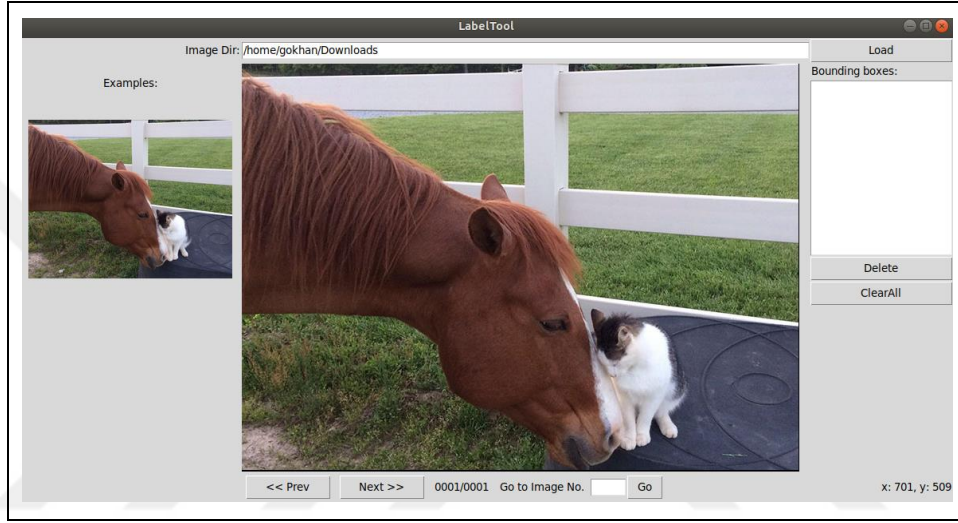
YOLO ilk versiyonunu tanıttığı yıllarda çok hızlı ve doğruluğu fena olmayan bir yöntem olarak sunulmuştur. Ancak YOLOv3’te ağıın daha hızlı olmasından ziyade, ağıın başarımının yüksek ve verimli sonuçlar vermesine önem verilmiştir. YOLOv3, resudal atlayıcı bağlantılar ve çoğullama (upsampling) gibi özellikleri ağı yapısına sahiptir. Aynı zamanda YOLOv2’ye göre 10 kat daha fazla sınırlayıcı kutu tahmin etmektedir. YOLOv3, giriş imgesinin boyutlarını sırasıyla 32, 16 ve 8 oranında aşağıya doğru (downsampling) bölerek üç ölçekte tahmin yapmaktadır. COCO veri seti için 50-60 mAP değerlerine 50ms civarlarında elde etmektedir [49].

Bir derin öğrenme ağıının kendisini eğitebilmesi için öncelikle etiketli veri ihtiyacı bulunmaktadır. Etiketli veri, insan gücüyle işaretlenen ve ağı neyin doğru neyin yanlış yada ağıın neyi öğrenmesi gerektiğini belirten girdilerdir. Model ağı etiketli veriyi bir kılavuz olarak kabul ederek sistemden çıktı üretmektedir. Üretilen çıktıyla, girdi arasında benzerlik (IoU) kurarak ağıın ağırlıklarının güncellenmesini sağlamaktadır.

Bir imgedeki nesnenin eğitilebilmesi için nesnenin eğitim verisinin sisteme etiketli veri olarak verilmesi gerekmektedir. YOLO eğitim sırasında her image girdisi için belirli bir formatta etiketli veri ihtiyacı duymaktadır. YOLO tarafından belirlenen Şekil 3.13’deki formata göre etiketli veriler oluşturulması gerekmektedir.

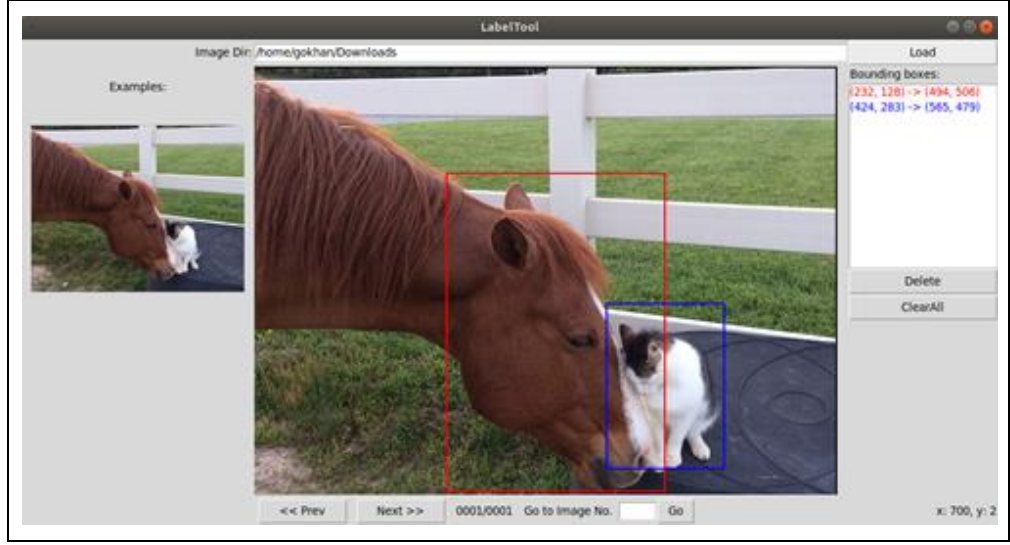
[sınıf_türü] [kutu_merkez_x] [kutu_merkez_y] [kutu_yüksekliği] [kutu_genişliği]

Şekil 3.13. YOLO ‘nun eğitim için oluşturduğu etiketli veri formatı



Şekil 3.14. Etiketli veri oluşturma programı

Şekil 3.14’te bu çalışmada eğitim için kullanılan açık kaynak kodlu etiketleme programına ait görsel bulunmaktadır. Bu program verilen üst kısımda bulunan kutunun içerisine yazılan dosya yolundaki tüm “.png” uzantılı imgeleri karışık sıralı getirebilmektedir. Bu program sayesinde etiketli veri insan gücü ile işaretlenerek YOLO formatında eğitim veri setleri oluşturulabilmektedir. Bu görsel üzerinde işaretlenen sınırlayıcı kutunun merkez noktasının koordinatları, yükseklik ve genişlik bilgisi ayrıca hangi sınıfa ait olduğu belirlenmektedir. Program ile işaretlenmiş bir imgedeki objeler Şekil 3.15’te gösterilmiştir.



Şekil 3.15. YOLO ile eğitim için etiketleme programı ile işaretlenmiş örnek

Derin öğrenme algoritmalarında tıpkı yapay sinir ağlarında kullanılan geriye yayılımdaki (back propagation) gibi ağı gradyan hesaplaması yaparak güncellemektedir. Bu güncelleme işlemi çok büyük boyutlardaki eğitim veri setleri kullanılırken çok yüksek işlem sürelerinin harcanmasına sebep olmaktadır. Bu problemi çözmek amacıyla imge veri seti, grup (batch) halde işlendikten sonra ağı güncellemesi yapılmaktadır. Böylelikle ağı güncellemesi için kaybedilen işlem zamanı telafi edilmektedir. Batch boyutu seçilirken oldukça dikkatli olunmalıdır. [42]'deki çalışmada da bahsedildiği üzere batch boyutundaki ufak değişiklikler öğrenmenin başarısında çok büyük değişimlere neden olmaktadır.

Öğrenme oranı, sistemin veri setini öğrenme hızını belirlemektedir. Derin öğrenme algoritmalarında genellikle tercih edilen öğrenme oranı eğitimin başlangıcında yüksek olup eğitimin ilerleyen süreçlerinde bu oran düşürülmekte ve böylelikle ağı daha az hatayla ve daha hızlı öğrenmesi sağlamaktadır. Bunun sebebi ise öğrenme oranının başlangıta düşük olması veya sürekli düşük kalmasının hata oranının belirli bir hata aralığında sıkışmasına neden olmasıdır. Öğrenme hızının başlangıta veya sürekli yüksek olması ise ağı sürekli doğrusal ve pürüzsüz bir hata oranına oturamamasına neden olmaktadır.

Bu parametrenin doğru değerde bulunabilmesi birden fazla etkene bağlı olmaktadır. Bu etkenler aşağıda açıklanmıştır.

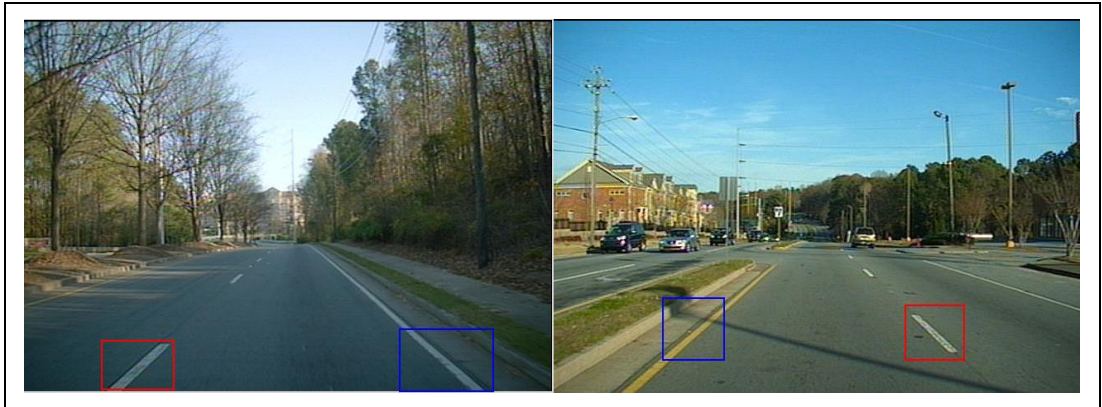
- Batch boyutu büyüdükçe model ađın öğrenimi sırasında ađın nesnelere ezberlemesi (overfitting) kolaylaşmakta, batch boyutu küçüldükçe model ađın öğrenmesi yavaşlamakta ve zorlaşmaktadır.
- Ađın öğrenme hızının (learning rate) yüksek olması öğrenim sırasında doğru öğrenim katsayılarına erişememesine sebep olmakta, düşük olması çok uzun zamanda ađın öğrenmesine neden olmaktadır. Bu da batch boyutunun öğrenme hızıyla beraber denenerek bulunması gerektiđini göstermektedir.
- Ađın eğitimi için hazırlanan etiketli imgelerin içindeki nesnelere birbirleri ile çok fazla kesişen (intersection) nesnelere eğitim sırasında sistemin hızlı ezberleme problemini ortaya çıkarmaktadır. Olabildiğince batch boyutu optimumlaştırılarak işlenen nesnelere farklı imgelerde alınarak işlenmesi gerekmektedir.
- Batch boyutunun büyük olması hata kaybı (error loss) grafiğinde düzgün bir düşüş elde edilememesine neden olabilmektedir. Bunun sebebi karışık şekilde alınan eğitim veri seti gruplarında oluşan farklılardan dolayı doğru batch boyutu ve öğrenme hızı parametrelerinin oluşturulamamasına neden olmaktadır. Ancak eğitim iterasyonu arttıkça bu doğru pürüzsüzleşmesi olađan bir durum olmaktadır.

Test sürecinde ekstra bir işlem gerektirmemektedir. Test için ayrılan imgeler yöntem girdi olarak verilmekte ve yöntem çıkışta doğrudan sistemin modelinin doğruluk oranına göre test sonuçlarını vermektedir.

4. TEZ ÇALIŞMASINDA GERÇEKLENEN ŞERİT TESPİT SİSTEMİ

Bu bölümde tez çalışmasında yapılan çalışmalar hakkında bilgiler verilecektir. Bu tez çalışmasında şeritten ayrılma sistemlerinin derin öğrenme yöntemlerinden YOLO ile nesne tespiti yöntemi kullanılarak bir çözüm üretmeyi hedeflemiştir. Şerit tespiti için geliştirilen yöntemde yüksek çözünürlükteki imgelerde yüksek doğrulukla sonuç verebilmesi ve gerçek zamanlı nesne tespiti yapabilmesinden dolayı YOLO tercih edilmiştir. Ancak düşük çözünürlüklerde diğer derin öğrenme metodları gibi YOLO da yüksek çözünürlüklerdeki doğruluk oranına göre zayıf kalmakta ve öğrenmesi zorlaşmaktadır.

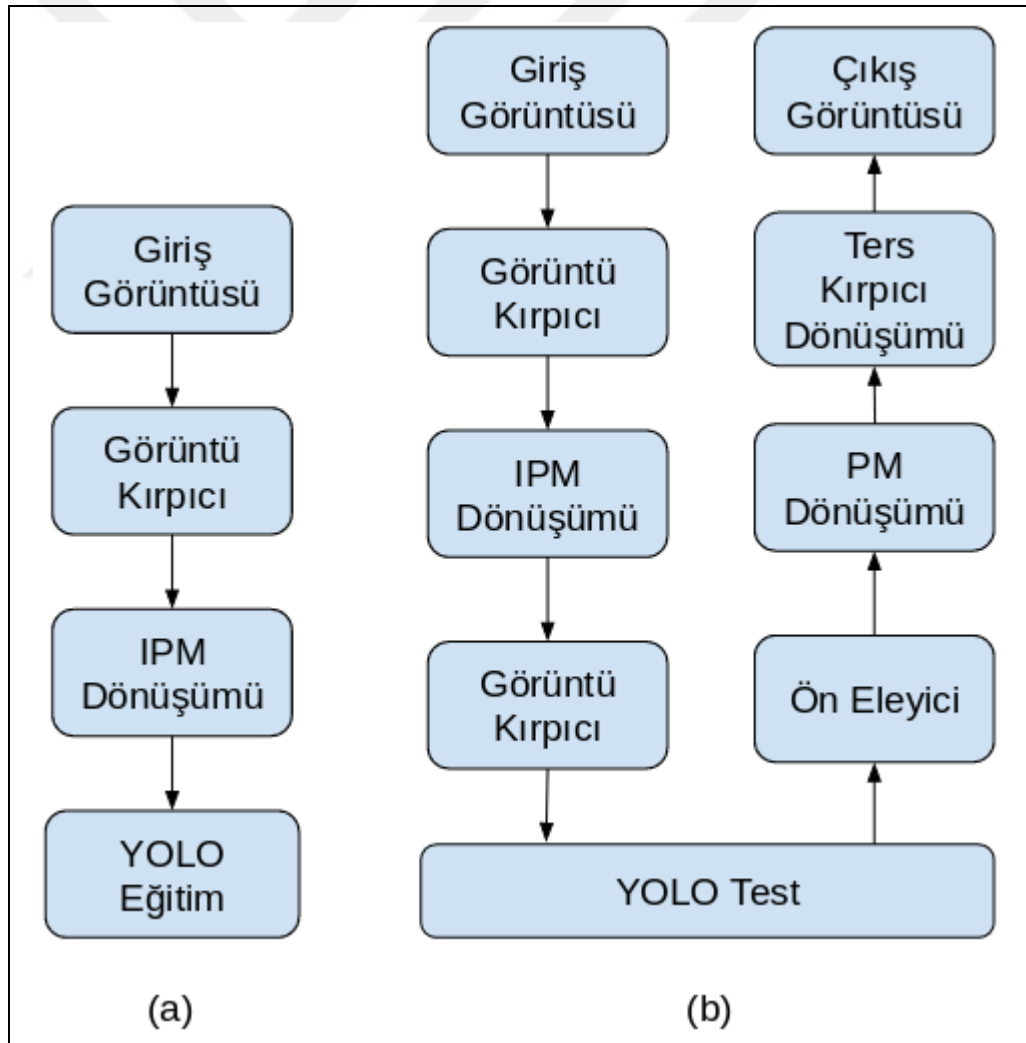
Literatürde şeritten ayrılma tespit sistemleri için geliştirilen nesne tespit çalışmalarında şerit işaretlerinin eğitimi için oluşturulan araç içi kameralar ile çekilen birçok ham şerit veri setleri bulunmaktadır. Bu çalışmada hali hazırda oluşturulmuş olan veri setlerinden Borkar [24] veri seti kullanılmıştır. Bu veri setinde işaretlenen etiketli veriler kamera pozisyonunun eğik olması sebebiyle Şekil 4.1’de gösterildiği gibi ham verinin sınırlayıcı kutu içerisinde çok küçük bir kısmı kaplamasına sebep olmaktadır. Bu problem eğitim sırasında sisteme verilen etiketli veri setinin yeterince öznetelik çıkaramamasına ve eğitim verisinin büyük boyutlarda sisteme girdi olarak girmesi sebebiyle işlem süresinin artmasına neden olmaktadır.



Şekil 4.1. Şerit işaretlerinin etiketlenmesi sonucu oluşan sınırlayıcı kutu kaplama alanı örneği

Bu gibi problemleri çözmek adına ham veri setlerinin eğitim sırasında sınırlayıcı kutuda daha fazla alan kaplaması ve imgenin özneliklerini yüksek başarımla çıkarabilmesi gerekmektedir. Aynı zamanda çoğu geleneksel görüntü işleme algoritmaları kamera gibi görü temelli algılayıcılarla kullanıldığı zaman kalibrasyon parametreleri oldukça hassas şekilde ayarlanmaktadır. Geliştirilen algoritmalar kalibrasyon değişimlerinden çok fazla etkilenmektedir. Bununla beraber şerit işaretlerinin görüntüye dik ve şerit işaretlerinin birbirine paralel olması etiketli verileri YOLO ile eğitim sırasında daha efektif şekilde kullanılmasını sağlamaktadır.

Geliştirilen sistemin bozucu etkilerden daha az etkilenmesi amacıyla çalışmada şeritten ayrılma tespit uygulamalarında sıkça tercih edilen ters perspetif dönüşümü, YOLO ile derin öğrenmeye girmeden önce uygulanmaktadır.

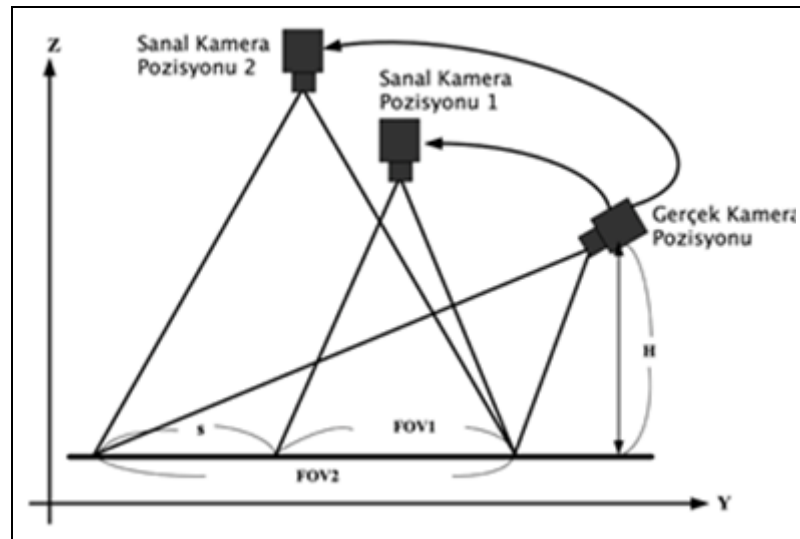


Şekil 4.2. Tez çalışmasında gerçekleştirilen yöntemin detayları a) Eğitim sürecindeki yöntem b) Nihai geliştirilen yöntem

Bu tez çalışmasında geliştirilen yöntem Şekil 4.2’de gösterildiği gibi iki bölümden oluşmaktadır. Birinci bölümde geliştirilmek istenilen nihai yöntemde kullanılacak yapay sinir ağının şerit tespit işlemini yüksek doğruluk ile yapabilmesi için eğitilmesi gösterilmektedir. İkinci bölümde ise çalışmanın geliştirildiği yöntem genel olarak gösterilmektedir.

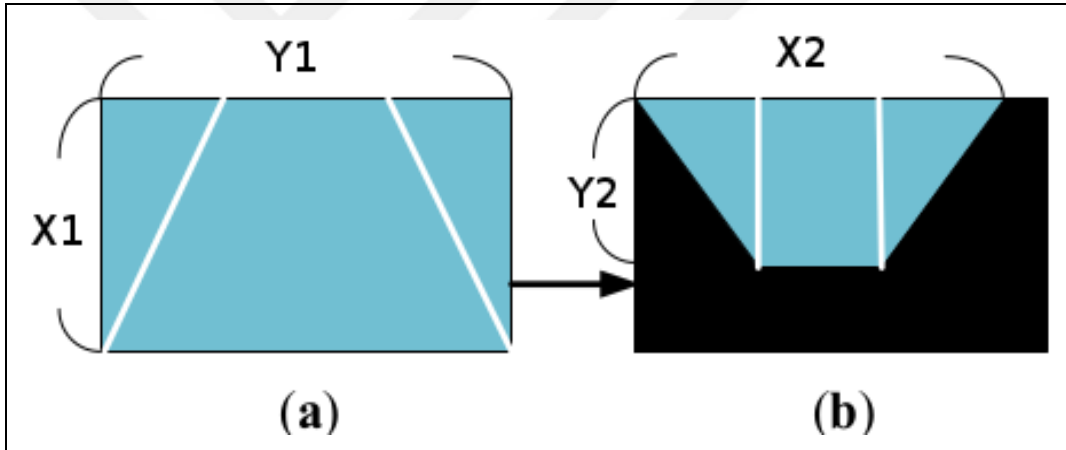
4.1. Ters Perspektif Dönüşümü

Bir çok derin öğrenme temelli şeritten ayrılma tespit sistemi aracın ön bölgesinde ve içine yerleştirilmiş kamera ile yatay şekilde alınan görüntü setleri üzerinden kendi yöntemlerini uygulamaktadır. Gerçek dünyada birbirine paralel olan şerit işaretleri kameradan alınan bu görüntülerde perspektif etkisi nedeniyle paralel olarak gözükmemektedir. Bu paralel olmamasının yöntemlerin çalışmasına dezavantajı bulunduğundan ham görüntüler genelde ön işleme tutulmaktadır. Bu tez çalışma kapsamında önerilen yöntemde, şerit işaretlerinin araca dik ve paralel olması özelliğinden faydalanılmıştır. Bu nedenle kamera alınan görüntülerin kamera düzleminde dünya düzlemine geçmesi için ters perspektif dönüşümü (Inverse Perspective Mapping - IPM) kullanılmıştır. Şekil 4.3’te gösterildiği gibi açılı şekilde araçlara konumlandırılan kameraların perspektiften elde edilen görüş açısı değişimiyle kuş bakışı perspektife geçilmektedir.



Şekil 4.3. Orjinal kamera pozisyonunun sanal kamera pozisyonlarına dönüşümü [25]

Ters Perspektif Dönüşümü kamera ve lens kalibrasyonuna bağlılığı oldukça az olan bir yöntemdir. Yöntem temel olarak kamera çekim perspektifinin bakış açısından faydalanarak farklı bakış açısına geçmesiyle görüntünün değiştirilmesini amaçlamaktadır. Bu işlemi giriş görüntüsünden aldığı 4 nokta üzerinden kameranın bakış açısının değiştirileceği uzayın tersi ile transformasyon matrisi oluşturulmaktadır (Eşitlik (4.1), Eşitlik (4.2) ve Eşitlik (4.3)). Bu eşitliğe göre eşitliğin sağ tarafı dönüştürülmek istenen noktanın homografi dönüşümü ile eşitliğin sağındaki uzayın perspektifinin değişimi sonrasında ortaya çıkan noktayı vermektedir. Şekil 4.4(a)'da şerit çizgileri bulunan imgenin genişliği X_1 ile yüksekliği Y_1 ile tanımlanmakta, Şekil 4.4(b)'deki ise şerit çizgileri bulunan imgenin genişliği X_2 ile yüksekliği Y_2 ile tanımlanmıştır. Ters transformasyon matrisi sayesinde eğimli şerit işaretleri doğrusal şekle Şekil 4.4'teki şekilde dönüştürülmüştür.



Şekil 4.4. Ham şerit işaretleri barındıran görüntünün IPM işlemi sonrası dönüşümü (a) Ham görüntü (b) IPM işlemi sonrası oluşan görüntü [25]

Bu matrisin hesaplanması görsel öğelere bağlıdır. Bu matris perspektifi dönüştürülmek istenen imgenin dört köşesi için ayrı ayrı uygulanarak kullanılmaktadır. Böylece kameranın perspektifinden olabildiğince az etkilenilerek doğrusal bir görüntü elde edilmiş olmaktadır. Bu hesaplanmanın doğru şekilde yapılabilmesi için her kameranın konumlandırılması, lens açısı ve bakış açısı (Field of View, FoV) üzerinden transformasyon matrisi değerlerinin çıkarılması gerekmektedir.

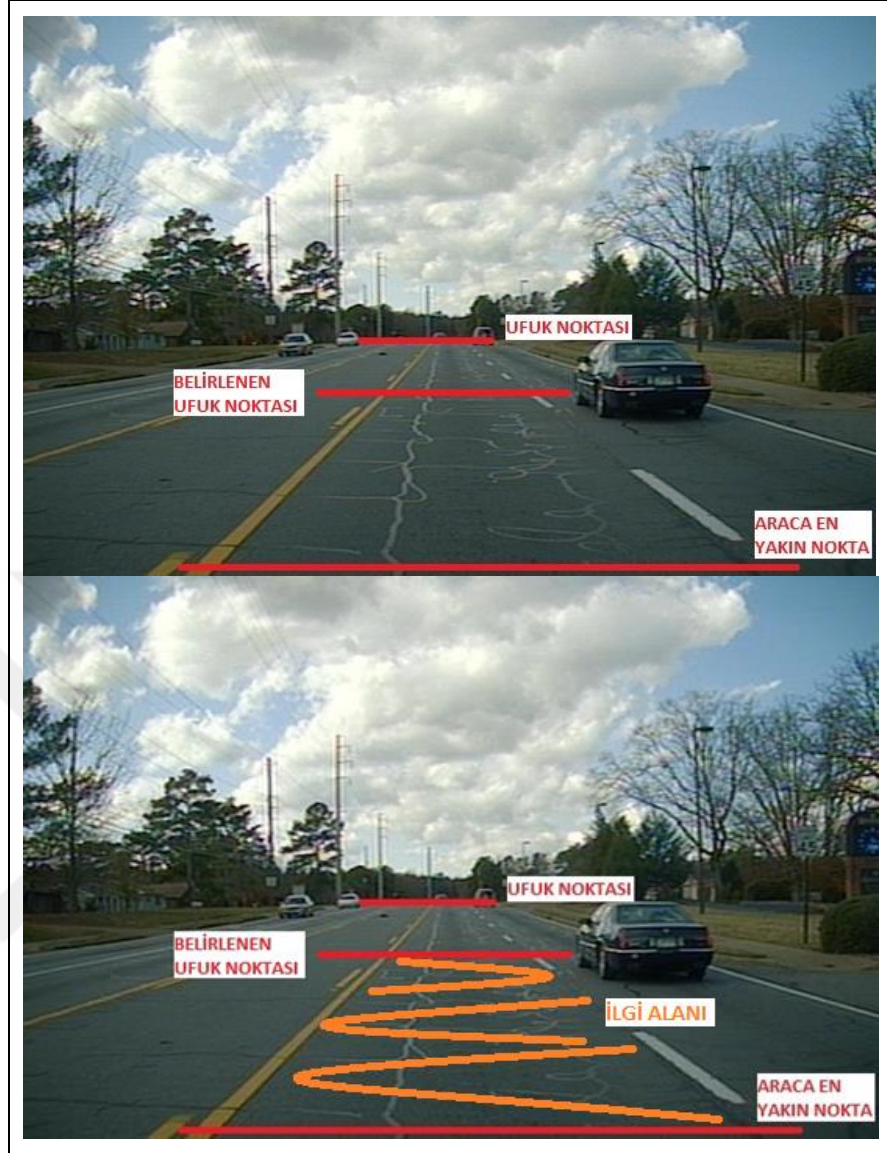
$$\begin{pmatrix} X_2 \\ Y_2 \\ Z_2 \end{pmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{pmatrix} X_1 \\ Y_1 \\ Z_1 \end{pmatrix} \Rightarrow \begin{pmatrix} X_2 \\ Y_2 \\ Z_2 \end{pmatrix} = H \begin{pmatrix} X_1 \\ Y_1 \\ Z_1 \end{pmatrix} \quad (4.1)$$

$$x_2 = \frac{x_1}{z_1} = \frac{h_{11}x_1 + h_{12}y_1 + h_{12}}{h_{21}x_1 + h_{22}y_1 + h_{22}} \quad (4.2)$$

$$y_2 = \frac{y_1}{z_1} = \frac{h_{21}x_1 + h_{22}y_1 + h_{23}}{h_{31}x_1 + h_{32}y_1 + h_{33}} \quad (4.3)$$

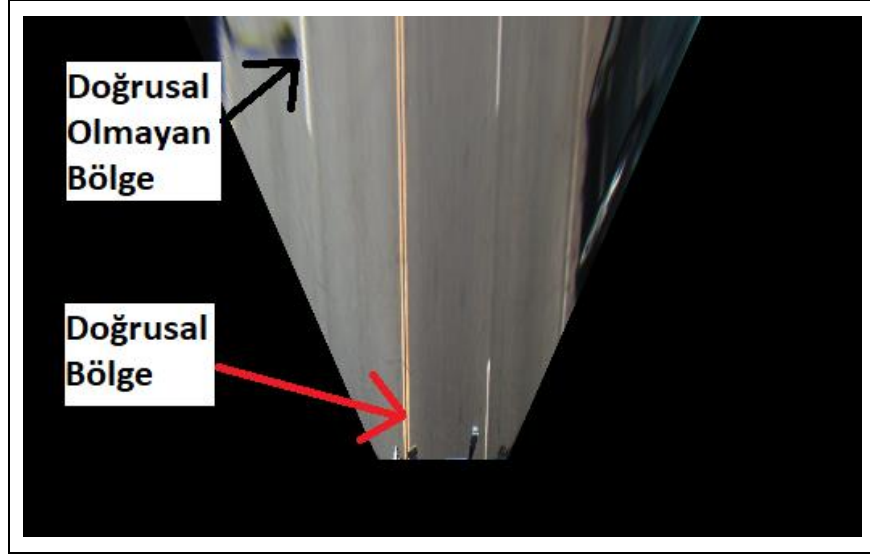
Ters perspektif dönüşümü yapılmadan önce ham imge kırpıcıdan geçmesi gerekmektedir. Bu işlemin yapılmasının sebebi şerit işaretlerinin ham görüntünün başladığı alt noktasından (araca yakın nokta) ufuk noktasına (araca uzak nokta) kadar ki bölümde bulunmaktadır. Bu bölümün dışındaki kısımlarda Şekil 4.5'te gösterildiği gibi şerit işaretlerinin bulunmamasından dolayı geliştirilen yöntemde ekstra yük olmaktadır. Ayrıca bu kırpıcı sayesinde tüm görüntünün perspektif dönüşümünü yapmak yerine sadece şerit işaretlerinin bulunduğu bölgenin perspektif dönüşümünü yapmanın yöntemin performansına olumlu etkisi bulunmaktadır.

Literatürde ufuk noktasının yerini bulabilen bir çok yöntem geliştirilmiştir [32-34]. Ancak bu çalışmada ufuk noktası literatürdeki yöntemler gibi belirlenmemektedir. Şekil 4.5'te belirlenen ufuk noktası literatürdeki yöntemler gibi belirlenmemektedir. Bu belirlenen ufuk noktası, Şekil 4.5'teki araca en yakın nokta ile arasında oluşturulan yamuk sayesinde ilgi alanı ile bulunmaktadır. İlgi alanı Şekil 4.4(b)'de gösterilen yamuğun alt kenarı olan araca en yakın nokta ve üst kenarı olan ufuk noktasından Şekil 4.5'teki gibi oluşmaktadır. Bu ilgi alanı araç içine yerleştirilen kameranın açısından (FOV) faydalanılarak oluşturulan bir yamuk alandır. Bu yamuk alan projekte edildiği yüzeyde piksel genişleme problem yaratmaması için ufuk noktası bu ilgi alanındaki yamuğun kenar eğimine ve alt kenar uzunluğuna bakılarak belirlenmektedir. Bu alt uzunluk tüm şerit işaretini kapsamaları amacıyla Borkar veri setinde ham imgenin yatayda 240. ile 320. pikselleri arasında belirlenmiş olup yanal kenarların eğimi ortalama 47 derece boyutlarında bulunmuştur. Bu bilgilerden faydalanılarak ufuk noktası Borkar veri setinde ham görüntünün 290. pikseli belirlenmiştir.



Şekil 4.5. Araç içi kamera ile alınan görüntüde ufuk noktasının belirlenmesi

Ham görüntünün ters perspektif dönüşümü işlemi sonrasında şerit işaretlerinin birbirlerine göre paralel ve ham görüntüye dik şekilde uzanması Şekil 4.6'da gösterildiği gibi ufuk noktasına yaklaştıkça bozulmaktadır. Bunun sebebi perspektif dönüşümü sonrası imgenin Şekil 4.4(a)'da gösterildiği gibi düz yamuk görüntüden Şekil 4.4(b)'ye benzer şekilde ters yamuk görüntüye geçişi sırasında esnetilmesinden oluşan doğrusal olmayan genişlemelerdir.

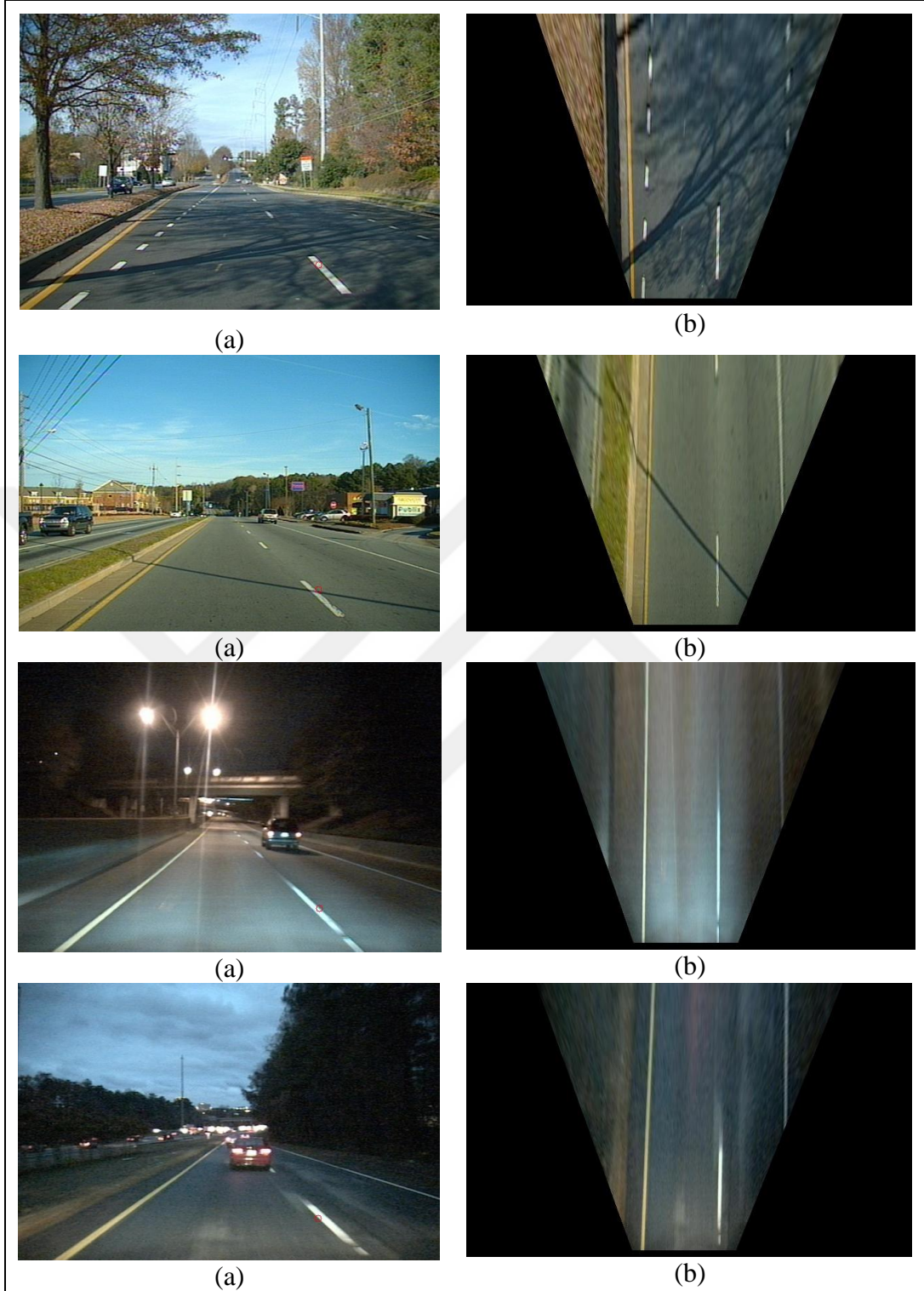


Şekil 4.6. Şerit işaretleri barındıran imgenin ters perspektif dönüşümü sonrası ufuk noktasından uzaklaştıkça oluşan doğrusal olmayan bölge

Genişleme problemini çözmek için el yordamı ile bozulmanın en az olduğu araca en uzak nokta olarak belirlenen ufuk noktasından araca en yakın noktaya kadar ki bölüm kesilerek ters perspektif dönüşümü işlemine sokulmaktadır. Şekil 4.7’de örnek bir giriş görüntüsü ve ters perspektif dönüşümü sonrası görüntüsü gösterilmektedir.

Şerit işaretlerinin ters perspektif dönüşümü sonrası elde edilen görüntü Şekil 4.7(b)’de gösterildiği gibi yamuk şeklinde bulunan gerçek dünya imgesi ve yamuğun dış bölgesinde bulunan siyah bölgeden oluşmaktadır. Şerit işaretlerinin ters perspektif dönüşümü sonrası elde edilen görüntünün Şekil 4.7(b)’deki gibi YOLO’ya eğitim işlemi sırasında etiketli imge olarak ve test süresince giriş imgesi olarak kullanmanın yöntem geliştirme aşamasında karşılaşılan bazı dezavantajları aşağıdaki gibi tespit edilmiştir.

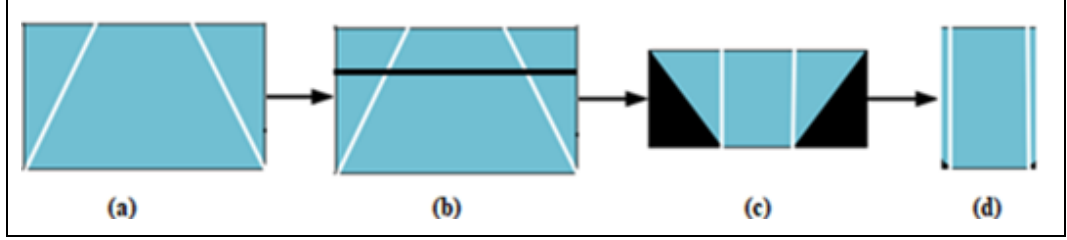
- Dönüştürülen imgede yamuk şekilde bulunan gerçek dünya görüntüsünün küçük bir kısmında şerit işaretleri bulunmaktadır. Bu bölgenin bütün imgenin küçük bir kısmını kaplamasından dolayı ters perspektif dönüşümü öncesi yapılan kırıcı işlemdeki gibi problemler oluşmaktadır.
- Dönüştürülen imgedeki yamuk bölümün dış kısmındaki kalan siyah bölgenin YOLO ile şerit tespitinin eğitimi sırasında öğrenmeye bozucu etkiye bulunabilmektedir.



Şekil 4.7. Ham görüntünün ters perspektif dönüşümü (a) Ham görüntü (b) Ters perspektif dönüşümü sonrası elde edilen görüntü

YOLO ile eğitim sırasında şerit tespiti için geliştirilen yüksek seviyeli parametrelerin (hyper parameters) ayarlanması zorlanmakta ve eğitimin yeterli doğrulukta yapılamamasına neden olmaktadır. Bu gibi sebeplerden dolayı ham girdinin ters

perspektif sonrası oluşan görüntünün ekstra bir kırpıcıdan daha geçerek YOLO ile eğitim ve teste sokulmaktadır.



Şekil 4.8. YOLO ile eğitim için gerekli etiketli giriş imgesinin işlem aşamaları şablonu (a) Ham imge (b) Ufuk noktasından (siyah doğru) kesilen imge (c) Ters perspektif sonucu imge (d) Şeritin var olduğu bölgenin kesildiği imge

Şekil 4.8(c)'deki şablon görüntüye göre belirlenen imgenin alt köşede oluşturulan beyaz renki bir ROI'yle giriş imgesi kesilmekte ve Şekil 4.8(d)'deki gibi bir giriş imgesi elde edilmektedir. Bu yöntem sayesinde elde edilen etiketlenebilir görüntünün çözünürlüğü Borkar veri setinde 174×290 olmaktadır. Dönüşümün tüm evreleri sırasıyla Şekil 4.8'de gösterilen akıştaki gibi Şekil 4.8(a)'da ham giriş imge Şekil 4.8(b)'deki gibi ufuk noktasından kesilerek Şekil 4.8(c)'deki gibi ters perspektif dönüşümü yapılmaktadır. Ardından imge şeritt işaretlerinin bulunduğu ROI'den Şekil 4.8(d)'deki gibi kesilmiştir. Böylece YOLO ile eğitim sırasında etiketlenebilecek imge ve test sırasında giriş imgesi çıktısı Şekil 4.9'daki gibi üretilmiştir.

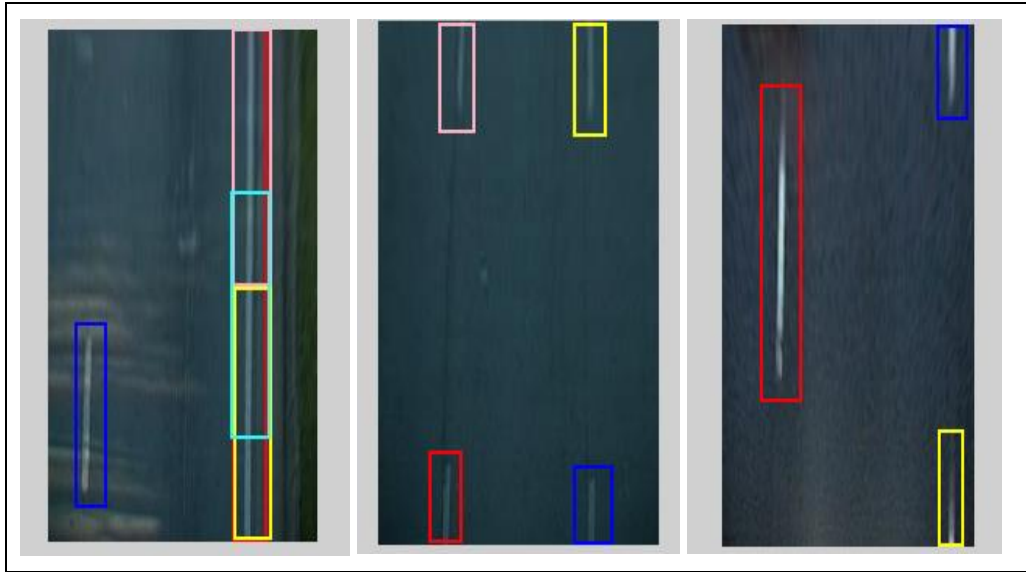


Şekil 4.9. YOLO ile eğitim için hazırlanmış giriş imgesi örnekleri

4.2. YOLO Eğitimi

Bu bölümde eğitim sırasında oluşturulan sistem hakkında bilgiler verilecektir. Ters perspektif dönüşümü sonrası belirli bir ROI ile küçültülen giriş imgesi sayesinde ağ modeli daha hızlı ve efektif eğitimle oluşturulmaktadır.

Etiketli veri işaretleme işlemi için Bölüm 3.2.1.1’de açıklanan yardımcı programı ile manuel olarak yapılmıştır. İşaretleme sırasında şerit işaretlerinin sınırlandırıcı kutulara dik ve iki şerit işaretinin birbirine paralel olmasından faydalanılarak sınırlayıcı kutunun yüzdesel olarak olabildiğince büyük alanını kaplayacak şekilde işaretlenmektedir. Şerit işaretlerinin eğitimi sırasında özniteliklerinin öğrenilebilmesi için etiketli verilerde dikkat edilmesi gereken hususlar bulunmaktadır. Şerit işaretleri genellikle otoyollarda beyaz ve sarı işaretler şeklinde bulunması ve otoyolların koyu siyah ve türevlerinden oluşması özniteliklerin çıkarılmasında kullanılabilir. Otoyolda kamera ile Şekil 4.9’deki gibi alınan bir görüntü renk uzayında incelendiğinde şerit olan yerlerin renk değerleri maksimum, olmayan yerlerin renk değeri minimum bulunmasıyla Gauss eğrisine benzer bir özniteliği bulunmaktadır. Bu özniteliklerin YOLO ile eğitim sırasında daha iyi öğrenilebilmesi için el yordamıyla yapılan etiketli veri işaretlemenin olabildiğince şerit işaretlerine ve arka plandaki otoyola fit biçimde Şekil 4.10’da gösterildiği gibi yapılması gerekmektedir.



Şekil 4.10. Etiketli eğitim imge örnekleri

YOLO eğitim verilerinin grup halde işlenebilmesi için geliştirdiği yapılar bulunmaktadır. Bu yapılarca göre eğitim imge setleri ve etiketli veri setinde belirlenmiş toplam sınıf sayısı, eğitim, test ve doğrulama imge ve etiketli veri setlerinin bulunduğu dosya yolu gibi bilgilerin bildirilmesi gerekmektedir. Bu çalışmada belirlenen bu yapıya göre Şekil 4.11’de gösterildiği gibi bir adet sınıf oluşturulmuştur. Yine başka bir yapı olan eğitim veri setlerinin dosya yolunun gösterilmesi ve bu eğitim dosya yolunda bulunan imgelerin “.png”, aynı klasörde etiketli veri setlerinin “.txt” uzantılı olması gerekmektedir.

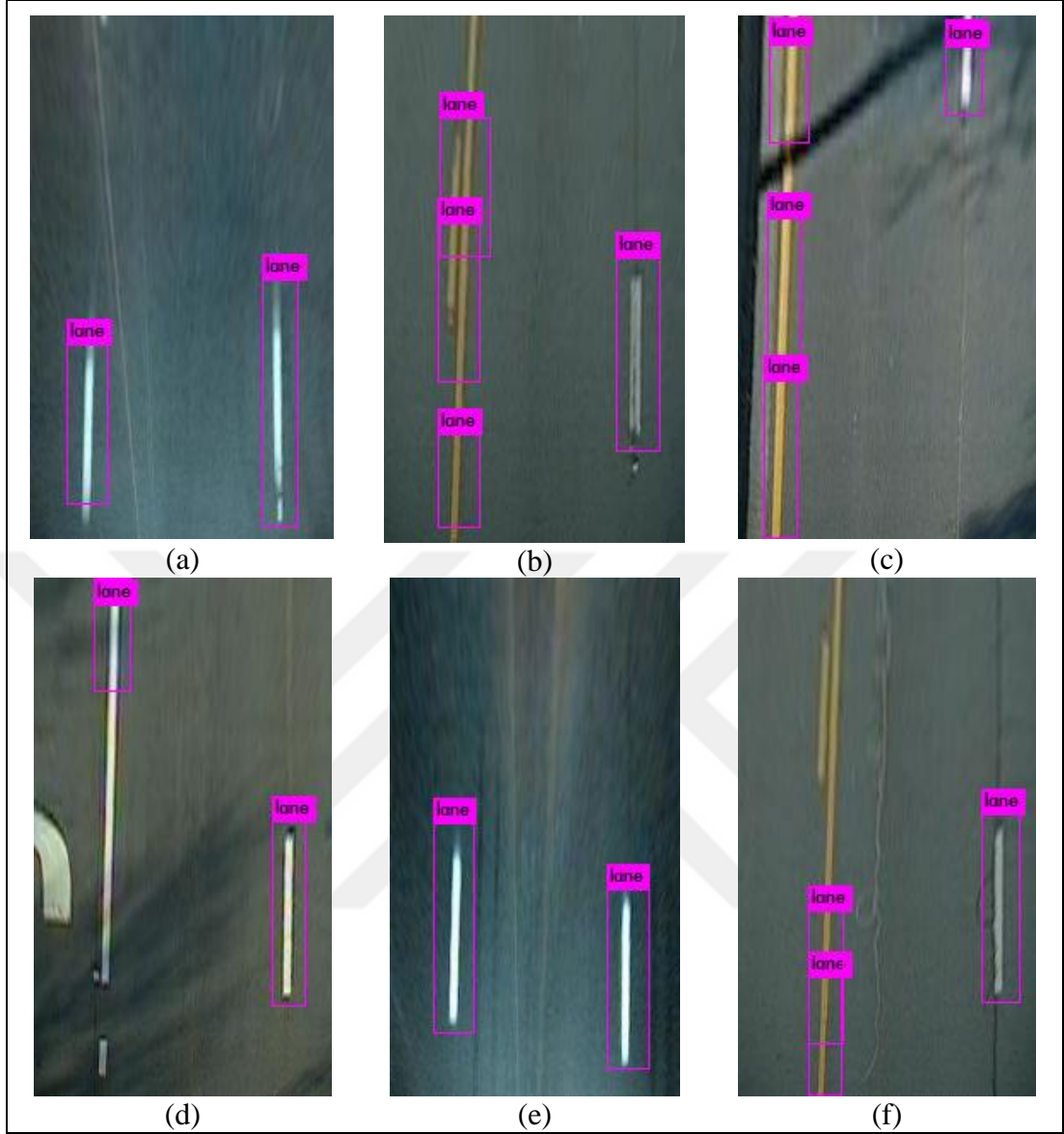
```
classes=1
train = /home/gokhan/Desktop/Trash/DL/Images/train.txt
valid = /home/gokhan/Desktop/Trash/DL/Images/valid.txt
backup = /home/gokhan/Desktop/Trash/DL/backup/
train = /home/gokhan/Desktop/Trash/DL/labels.txt
```

Şekil 4.11. YOLO eğitim konfigürasyonları

4.3. YOLO Test

Derin öğrenme ile geliştirilen modellerin doğru şekilde test edilmesi oldukça yüksek önem arz etmektedir. Bir sistemin iyi eğitim sonucu verip vermediği sadece hata oranına bakılarak anlaşılabilir. Örneğin bu tez kapsamında geliştirilen şeritten ayrılma uyarı sistemi için YOLO ile gerçekleştirilen eğitimin hata oranı 0,8 - 1,2 aralığında iken yapılan testlerde görece daha iyi şerit işareti tespiti başarımları sağlanırken eğitimin hata oranı 0,05 - 0,1 aralığında olduğu durumda şerit tespit başarımları düşmekte ve model daha çok yoldaki bozucu etkenlere odaklanmaktadır.

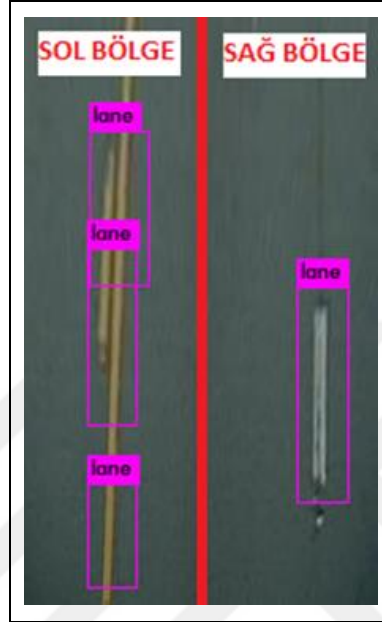
Eğitimin tamamlandığını modelin test sonucunda daha iyi karar verilmektedir. Eğitimi tamamlanan ağ modelinin testi, eğitim sırasında yapıldığı gibi test imgeleri YOLO ile geliştirilen ağ modeline sokulmaktadır. Ancak eğitim sırasında ağ modelini güncellemek için yapılan geri yayılım (back propagation) yapılmamaktadır. Elde edilen benzerlik sonuçları görsel ve grafiksel olarak kontrol edilmektedir. Bu çalışmada eğitilen ağ modelinden oluşturulan bir kaç test sonucu Şekil 4.12’de gösterilmektedir.



Şekil 4.12. YOLO test sonuçları

Bu sonuçlara göre oluşan sınırlayıcı kutular arasından görüldüğü üzere sisteme otoyol kırıklığı, gölge ve çatlaklardan ayrı olarak aynı şerit üzerinde birden fazla tespit yapması ve şerit işaretlerinin tespit edilemediği yerler gibi bazı bozucu etkiler yer almaktadır. Tespit edilemeyen bölgeler için önceki giriş imgelerine bağlı bir kalman filtresi gerekmektedir. Geliştirilen yöntemin çıktısında tek bir ham girdiden en fazla iki adet şerit işaretinin tespit edilmesi gerekmektedir. Bu tespit edilen şerit işaretlerinin sağ ve sol için maksimum birer adet tespit edilmesi gerekmektedir. Birden fazla tespit edilen sınırlayıcı kutuların elenmesi gerekmektedir. Eleme işlemi yapılabilmesi için sınırlayıcı kutunun sınıf benzerlik oranı ve ham imgedeki konumu oldukça önemlidir. Bunun için görüntüdeki tespit edilen sınırlayıcı kutuların konumlarına bakılarak Şekil

4.13’de gösterildiği gibi sol bölgedeki şerit işareti için imgenin yatayda başladığı noktadan orta noktasına kadarki bölüme kadar, sağ bölgedeki şerit işareti için orta noktasından imgenin bittiği noktaya kadarki bölümde ayrı ayrı eleme işlemi yapılmaktadır.



Şekil 4.13. Sağ ve sol bölge ayrımı yapım şekli

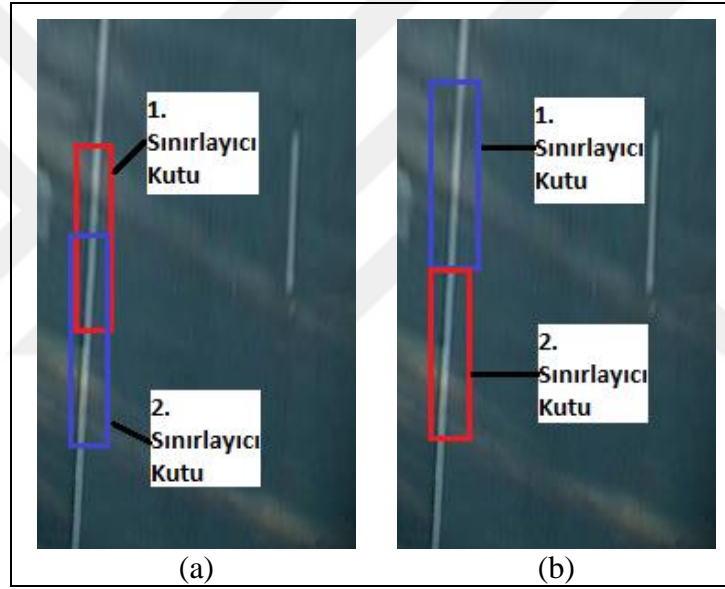
Geliştirilen ağ modeli testi sırasında el yordamıyla gözden geçirilmiştir. Test ile elde edilen sınırlayıcı kutulara göre aşağıdaki gibi sonuçlar elde edilmiştir.

- Karanlık imgelerde şerit işaretlerinin tespiti Şekil 4.12(a) ve Şekil 4.12(e)’deki gibi çoğunlukla tespit edilirken görüntünün karanlık olup bozucu etkilerinde kararması sebebiyle daha az etkilenmektedir.
- Şerit işaretleri üzerinde oluşan Şekil 4.12(c) ve Şekil 4.12(d)’dekine benzer gün batımıyla oluşan gölgeler sırasında bozucu etkenlerden etkilenilmektedir. Gölgelere dayanıklılığı ek sistemler eklenmesi gerekmektedir.
- IPM sonrası oluşan görüntünün alt sınırdan kırılan imge sayesinde şerit işaretleri olabildiğince imgenin köşe sınır bölgelerinde bulunmaktadır.

4.4. YOLO ile Eğitim Sırasında Elde Edilen Sonuçlar

Bu bölümde YOLO ile eğitim sırasında geliştirilen derin öğrenme yüksek parametrelerinin (hyper parameters) son ayarlamaları (fine tuning) sırasında elde edilen bu sonuçlar toplu olarak aşağıdaki gibi ele alınmıştır.

- Şekil 4.14'te Borkar veri setinden alınan bir imgenin yardımcı programla iki adet sınırlayıcı kutunun işaretlenmesi gösterilmektedir. Bu işaretleme sırasında sınırlayıcı kutuların birbirleriyle kesişen noktaları Şekil 4.14(a)'daki gibi fazla olan eğitim sırasında elde edilen hata oranı, kesişen nokta oranı Şekil 4.14(b)'deki gibi az olan eğitime göre daha fazla hata ve tespit sırasında sonuç elde edilmiştir.



Şekil 4.14. Şerit tespitinin eğitimi için hazırlanan veri setinin oluşturulması (a) İki sınırlayıcı kutunun yüksek oranda kesişmesi, (b) İki sınırlayıcı kutunun düşük oranda kesişmesi

- Batch boyutunun mini batch boyutuna oranı 3'e 1 (batch boyutu 96, mini batch boyutu 32) iken sınırlayıcı kutuların birbirlerine daha çok benzeyen etiketli veriler ile modelin eğitimi artacağından modelin hata oranında sürekli farklı değerlere zıplamasına ve belirli iterasyondan sonra ezberlemesine sebep olmuştur.
- Batch boyutunun mini batch boyutuna oranı 1,5'a 1 (batch boyutu 48, mini batch boyutu 32) iken sınırlı etiketli veri elde edilmesinden dolayı lokal minimumda kalmasına ve öğrenememesine neden olmuştur.

- Öğrenme oranı olarak birçok denemeler sonucunda literatürde de en çok tercih edilen 0,001 tercih edilmiştir. Eğitimin başlangıcında bu oran YOLO tarafından polinomal olarak yükseltilmektedir [22].

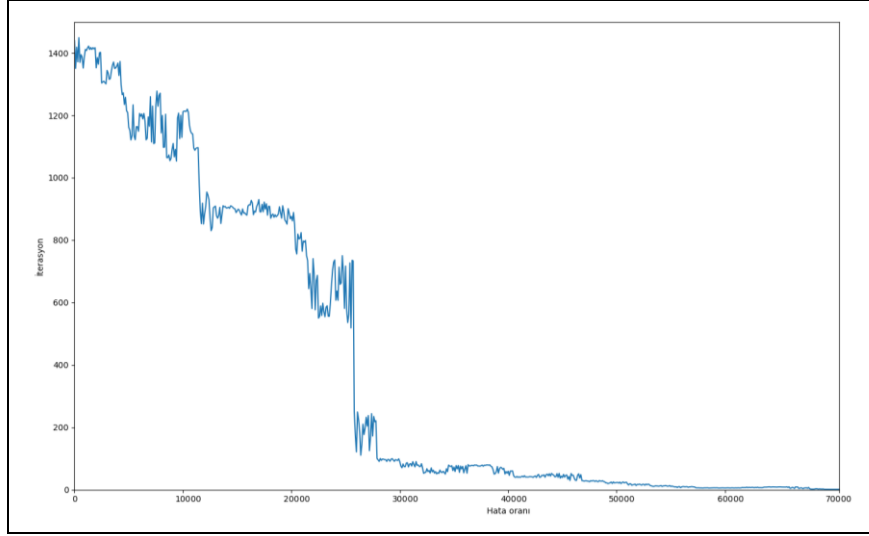
En iyi Batch, mini batch boyutu olarak 2'ye 1 oran bulunmuştur. Bu oranda da oldukça dikkatli şekilde belirli aralıklarla sistem eğitim ağırlıkları elde tutulup test edilmiştir. Bu çıktılarına göre eğitim hata oranı 0,09 bölgesinde iken eğitime son verilmiş olup, batch 64 mini batch boyutu 32 ve öğrenme oranı 0,001 olarak seçilerek bu hata oranı elde edilmiştir.

Çok yüksek seviyede iterasyonla eğitilen modelde başlangıçta istenilen 0,1 yakınlarında oluşan hata oranında model en iyi sonucunu vermekte ancak 0,01 oranlarına düşmesi istenildiği zaman eğitim modeli şerit işaretlerinin tespitinden daha çok kırıklara, bozukluklara ve gölgeler gibi otoyollarda oluşan bozucu etkilere odaklanmaktadır.

Tablo 4.1. YOLO eğitim parametre denemeleri hata sonuçları

Deneme	Batch Boyutu	Mini-batch Boyutu	Öğrenme oranı	Veri Miktarı	Hata Oranı
1	96	32	0,001	4500 (750×6)	1,98
2	48	32	0,001	4500 (750×6)	2,21
3	64	32	0,001	4500 (750×6)	1,58
4	64	32	0,001	1200 (200×6)	0,56
5	96	32	0,001	1800 (300×6)	0,74
6	64	32	0,005	4500 (750×6)	1,21
7	96	32	0,005	1800 (750×6)	1,44
8	64	32	0,001	1800 (300×6)	0,21

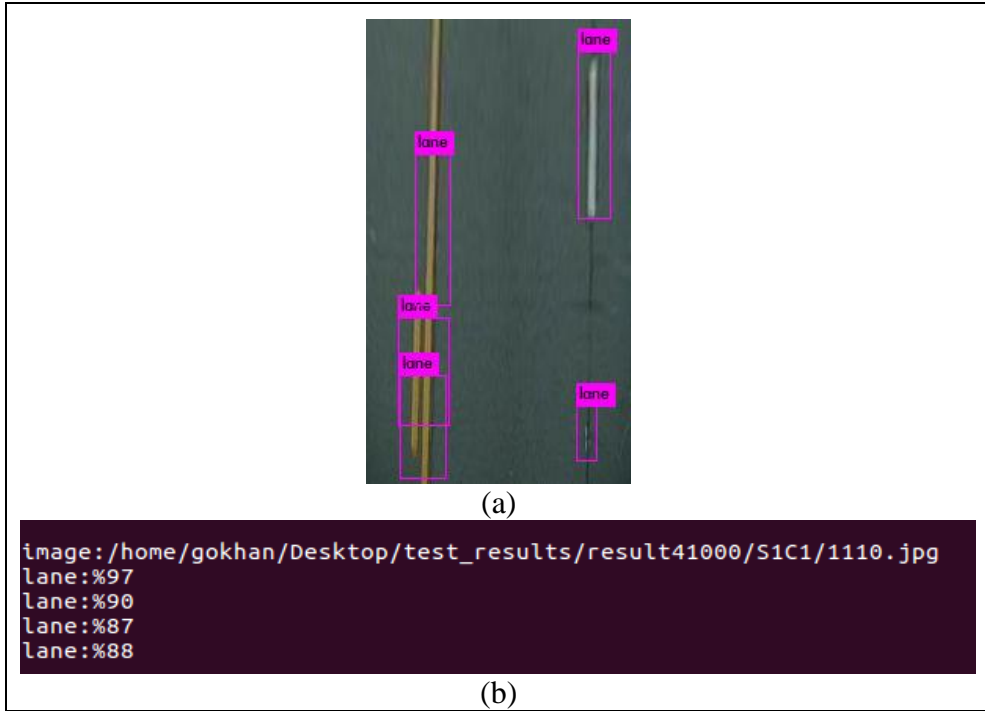
Çalışmada gerçekleştirilen eğitim işlemi sırasında hassas ayarlamalar tamamlanmıştır. Bu ayarlamalar içerisinde Tablo 4.1'de verildiği üzere en az hata oranını Deneme 8 satırındaki ayarlamalar sayesinde alınmıştır. Veri setinin testi sırasında elde edilen iterasyon hata oranı grafiği Şekil 4.15'te gösterildiği gibidir. Test sırasında elde edilen sonuçlar ve Şekil 4.12'de gösterilen tespit edilen sonuçlar karşılaştırılarak gözden geçilmiştir. Bu sonuçlara göre geliştirilen ağ modelinin eğitimine 29000. iterasyon bölgesinde tamamlanmasına karar verilmiştir.



Şekil 4.15. Eğitim işlemi sırasında elde edilen hata oranı grafiği

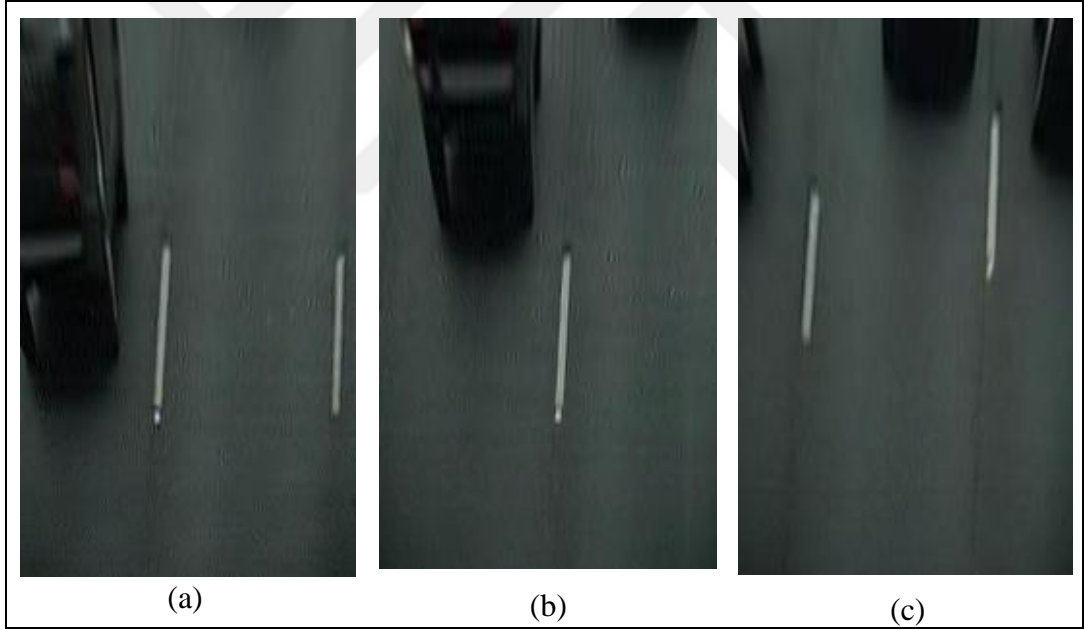
4.5. Son İşlem

Son işlem bölümünde YOLO test bölümünde eğitim sonucu oluşturulan modelin eksik yanlarını kapatmak amacıyla bazı işlemler uygulanmıştır. Bu işlemler genel olarak geleneksel görüntü işleme temelli olup merkezi işlem biriminde (CPU) gerçekleştirilmiştir.



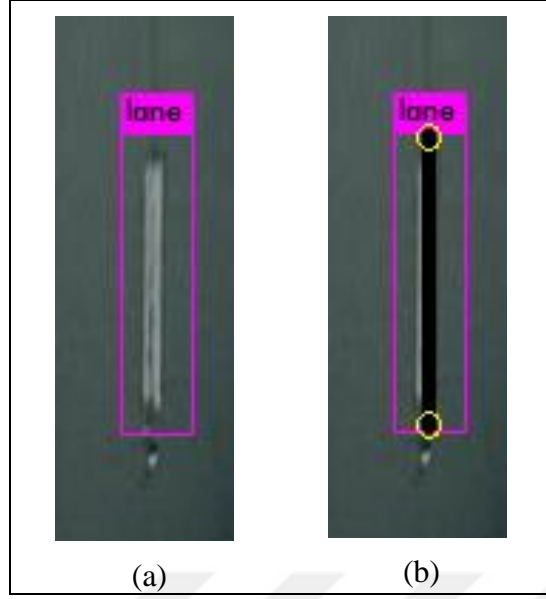
Şekil 4.16. YOLO ile Şerit tespit test çıktıları (a) Şerit tespit sonuç görüntüsü (b) Şerit tespit sonuçlarının sınıf benzerlik oranları

YOLO test sonuçlarına göre şerit işaretlerinin eleme işlemi için köşelere en yakın ve test sonucu yol sınıfına benzerlik oranı en yüksek sınırlayıcı kutu seçilmektedir. Şekil 4.16'da görüldüğü üzere bir imgede gerçekleştirilen imge bazlı test sonucu ve bu sonuca göre şerit sınıfına benzerlik oranı gösterilmektedir. Böylece iki şerit (sağ ve sol) işareti arasında oluşan bozucu etkiler YOLO ile eğitim sırasında elemine edilmektedir. Ancak şerit işareti ile imge sınır çizgisi arasında kalan bozucu etkiler elemine edilememektedir. Bu bozucu etkilerin sistem genel performansı açısından önemli bir problem olmadığı kanaatine varılmıştır. Araçlar sabit şeritte seyir halinde iken şerit işaretleri Şekil 4.11'deki gibi imge sınırlarına olabildiğince yakın şekilde bulunduğundan bozucu etkiler şerit işareti gibi davranmaktadır. Şekil 4.17'de gösterildiği gibi şerit değişimi sırasında imge sınırında olan şerit işaretlerinden herhangi biri kaybolacağı ve sadece bir şerit tespit edileceğinden dolayı sisteme büyük etkisi bulunmamaktadır.

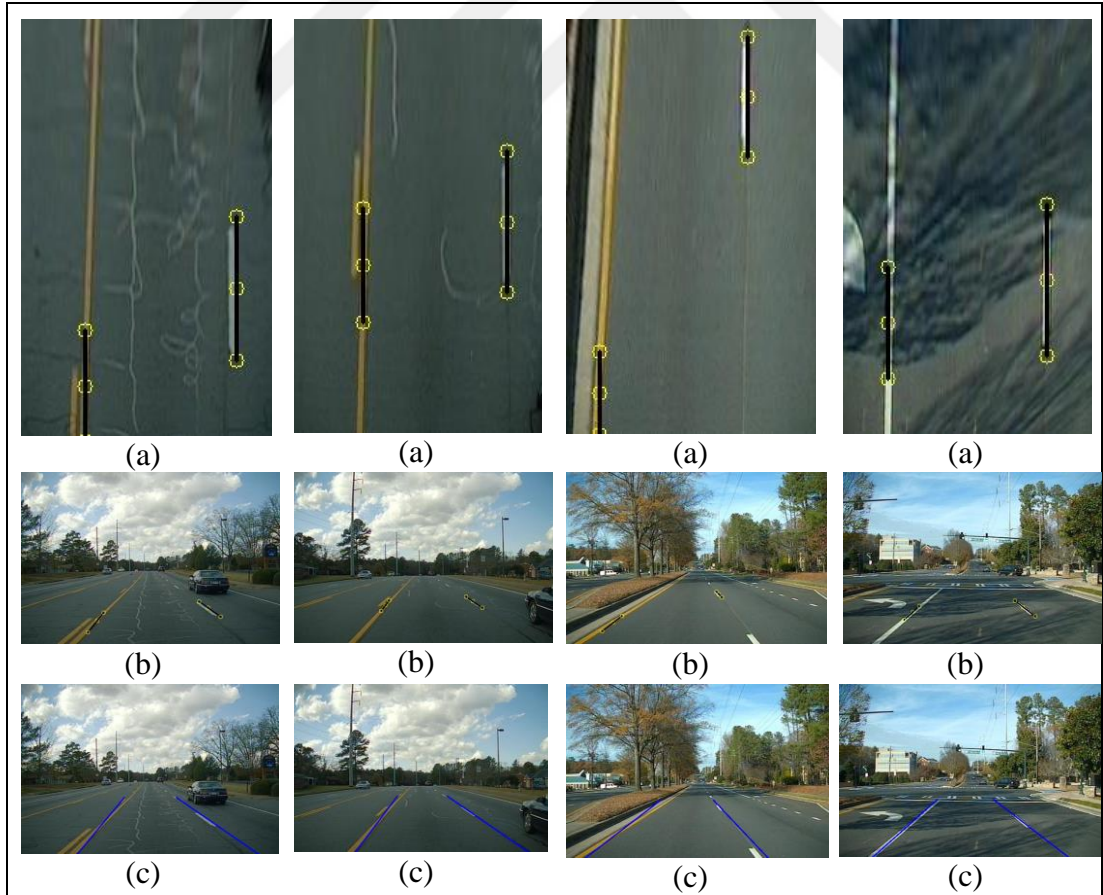


Şekil 4.17. Borkar veri setinden üretilen şerit değişimi sırasında oluşan görüntü (a) Şerit değişim başlangıcı görüntüsü (b) Şerit değişim görüntüsü (c) Şerit değişim bitiş görüntüsü

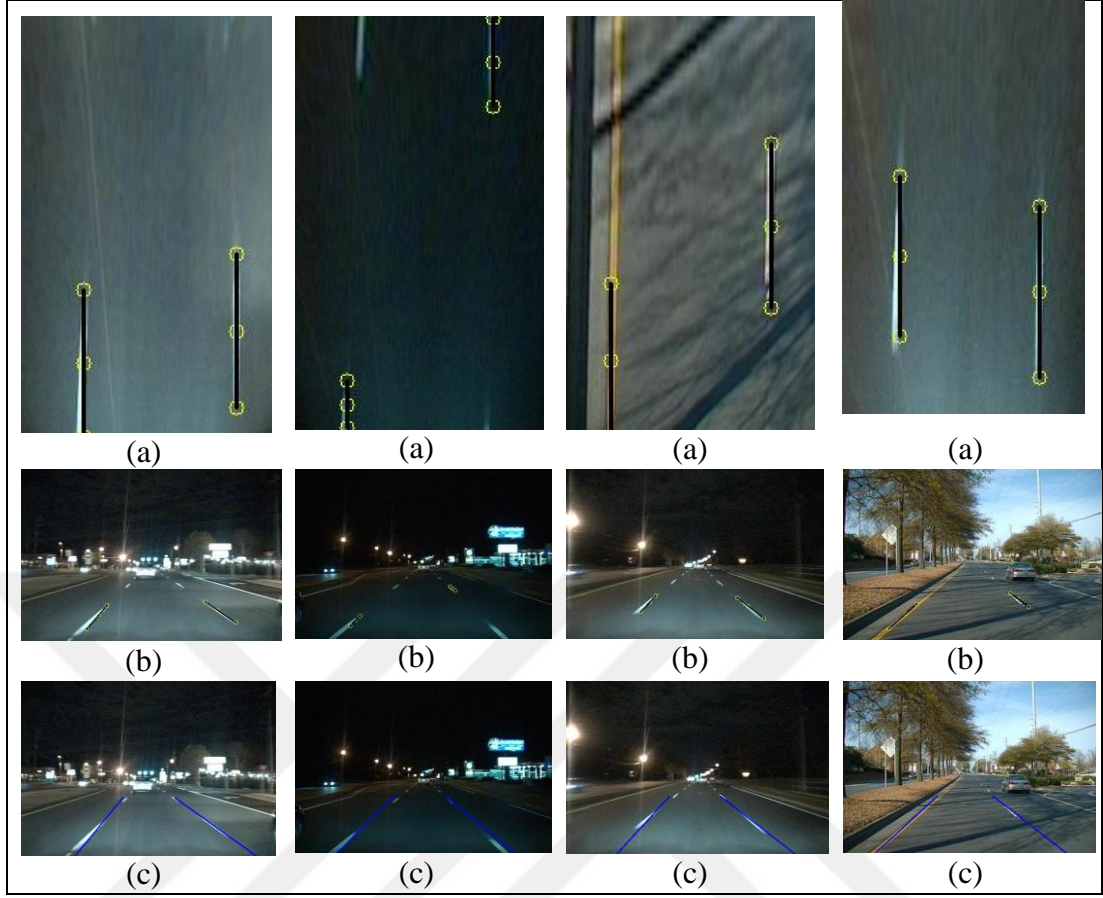
Elenmiş şekilde bulunan sınırlayıcı kutular sayesinde şerit işaretlerinin tespiti sağlanmıştır. Ancak tespit sonucunda elde edilen sınırlayıcı kutular görsel olarak elverişli gözükmemesinden dolayı sınırlayıcı kutunun üst ve alt sınırının yataydaki merkez noktaları arasında oluşturulan bir doğru ile Şekil 4.18(b)'deki gibi şerit işaretleri görselleştirilmiştir.



Şekil 4.18. YOLO tespit sonuçlarının dönüşümü (a) YOLO test sonucu (b) YOLO test sonucunun merkezinde çizilmiş doğru



Şekil 4.19 Geliştirilen yöntemin birinci çıktıları (a) Derin öğrenme test sonrası tespit edilen şeritler (b) Perspektif dönüşümü sonrası ham görüntüde işaretlenmesi (c) Deneysel sonuçlar için uzatılmış tespit sonuçları



Şekil 4.20. Geliştirilen yöntemin ikinci çıktıları (a) Derin öğrenme test sonrası tespit edilen şeritler (b) Perspektif dönüşümü sonrası ham görüntüde işaretlenmesi (c) Deneysel sonuçlar için uzatılmış tespit sonuçları

4.6. Deneysel Sonuçlar

Çalışmanın son işlem bölümüne kadarki kısmında sistem genel hatları ile anlatılmış olup çalışma kapalı döngü olarak anlatımı tamamlanmıştır. Bu bölümde bu kapalı döngü ile geliştirilen çalışmanın literatürde geliştirilen diğer çalışmalarla karşılaştırılması anlatılacaktır.

Bu çalışmada 6 eğitim veri setiyle toplam 1800 adet görüntü (6×300) üzerinden şerit işaretlerinin eğitimi için el yordamıyla şeritler işaretleri işaretlenmiştir. Eğitim ve test süreçlerinde Borkar veri seti kullanılmış olup literatürde bulunan şerit işareti tespit yöntemleriyle sistem performansı karşılaştırılmıştır. Bu çalışmada nesnel bir performans değerlendirmesi için Eşitlik (4.4)'de verilen ortalama karesel hata (Mean Square Error -MSE) karşılaştırma formülü olarak kullanılmıştır.

$$MSE = \frac{1}{N} \sum_{i=1}^N (X_{tespit} - X_{yer_dogrusu})^2 \quad (4.4)$$

MSE değerinin hesaplanması tespit edilen şerit çizgisinin bulunduğu nokta (x_{tespit}) ile yatay düzlemde tespit edilen şerit çizgisi (ground truth) noktası ($x_{yer\ dogrusu}$) arasında hesaplanmaktadır. Çalışmanın literatürdeki diğer yöntemlerle karşılaştırılabilmesi için Şekil 4.21(a)'daki gibi YOLO test çıktısı olan sınırlayıcı noktaların Şekil 4.21(b)'deki gibi üst ve alt noktasından alınan noktalar Şekil 4.21(c)'deki gibi nihai şerit işaretlerini gösteren doğrular elde edilmiştir. Bu elde edilen şerit işaretlerini gösteren doğrular Borkar veri setinin deneysel sonuçları için yükseklik ekseninde 280. ve 420. pikselleri kestiği noktalar doğrusu şeklinde uzatılarak deneysel sonuçlara hazırlanmıştır.



Şekil 4.21. Geliştirilen yöntemin deneysel sonuçlara hazırlanması (a) YOLO tespit sonucu (b) Tespit sonuçlarından alınan örnek noktalar (c) PM dönüşümü sonrası elde edilen doğrular

Tespit edilen şerit işaretleri ve oluşturulan yer doğrularının arasında MSE hesaplaması yapılarak bu değerin belirlenen eşik değerinden (yatay şerit işareti büyüklüğünün yarısı kadar piksel) küçük olması durumunda şerit işareti tespit edildi kararı verilmektedir. Önerilen yöntemin ve literatürde bulunan yöntemlerin şerit işareti tespit performansının değerlendirilmesi Eşitlik (4.5), Eşitlik (4.6) ve Eşitlik (4.7)'te gösterildiği gibi Tablo 4.1'de verilmektedir.

$$p = \frac{TP}{(TP+FN)} \quad (4.5)$$

$$r = \frac{FP}{(FP+TN)} \quad (4.6)$$

$$F_1 = 2 \times \left(\frac{p \times r}{p + r} \right) \quad (4.7)$$

TP (True Positive), yol yüzeyinde şerit işaretinin var olduğu ve yöntemin başarılı bir şekilde şerit tespit etmesi, FP (False Positive), yol yüzeyinde şerit işaretinin var olmadığı yöntemin şerit işareti tespit etmesi, TN (True Negative), yol yüzeyinde şerit işaretinin var olduğu durumda yöntemde şerit işareti tespit etmemesi, FN (False Negative) ise yol yüzeyinde şerit işaretinin var olduğu durumda yöntemin şerit işaretini tespit edememesi durumudur.

Tablo 4.2. Şerit işareti tespit yöntemlerinin (Borkar ve diğ., 2012) veri seti üzerinde performans değerlendirilmesi

	(Gaikwad ve Lokhande, 2015) [58]	(Borkar ve diğ., 2012) [24]	(Mammeri vd., 2016) [55]	(Küçük yıldız ve Ocak, 2014) [61]	Önerilen Yöntem
S1C1	0,84	0,75	0,81	0,82	0,64
S2C1	0,60	0,23	0,36	0,45	0,37
S4C6	0,31	0,57	0,32	0,42	0,65
Ortalama	0,43	0,43	0,37	0,43	0,46

Şerit işaretlerinin tespitinin sonuçları, literatürdeki çalışmalar üzerinde Borkar veri seti kullanılarak elde edilmiş F1-skorumları ile Tablo 4.2'deki gibi gözükmektedir. Önerilen yöntemin F1-skoru literatürdeki karşılaştırılan diğer yöntemlerden daha başarılı olduğu gözlemlenmiştir. S1C1 vs S2C1 görüntülerinde elde edilen sonuçlar S4C6'da elde edilen sonuçlara göre göreceli olarak daha kötü bulunmuştur.

Bunun sebebi S1C1 ve S2C1 görüntülerinde bozucu etkiler S4C6'ya göre oldukça fazla oranda bulunmaktadır. Bu da geliştirilen ağıın bozucu etkileri tanıması için eğitiminin geliştirilmesi gerektiğini göstermektedir. Ekstradan ek bozucu etkiler içeren eğitim verileri ile ağıın geliştirilmesine devam edilmesi gerekmektedir.

Tablo 4.3'te teknik özellikleri verilen Linux işletim sistemi altında C ve C++ ile kodlanmış şekilde çalışan makinelerde denenmiştir. Yazılım dili görüntü işleme bölümlerinde (IPM, PM) OpenCV ile C++ dili, YOLO ile drin öğrenme bölümlerinde C dili ve CUDA kütüphanesi kullanılmıştır.

Tablo 4.3. Yöntemin çalıştırıldığı makinelerin teknik özellikleri

Özellikler	PC-1	PC-2	PC-3
İşlemci (CPU)	Intel i9-7900X, 3,3GHz, 20 Cores	Intel i7-6700K, 4GHz, 8 Cores	Intel i7-5960HQ, 2,90 GHz
Ekran Kartı (GPU)	Nvidia Geforce TitanXp, Nvidia Geforce RTX 2080	Nvidia Geforce RTX 2060	Nvidia Quadro K1100M
Geçici Hafıza (RAM)	64 GB	20 GB	16 GB
Kalıcı Hafıza (HDD)	512 GB	1TB	512 GB

Geliştirilmiş olan yöntemin test edildiği makinelerdeki çalışma süreleri Tablo 4.3'te gösterilmektedir. Bu testler sırasında ilk makinede iki adet ekran kartı bulunmaktadır. İki ekran kartında toplam 50ms (20 fps) civarında süre harcanmaktadır. Bu sürenin büyük kısmında şerit işaretlerinin tespit işlemi için kullanılmaktadır. Her iki makinada da ters perspektif dönüşümü sırasında ortalama yakın süreler harcanmaktadır. Tablo 4.2'de başarımları karşılaştırılan yöntemlerin aynı makine üzerinde çalışma süreleri Tablo 4.5'te gösterilmiştir. Elde edilen bu süreler doğrultusunda önerilen yöntem diğer yöntemlerden daha yüksek FPS aynı zamanda daha alçak çalışma süresi elde edilmiştir.

Tablo 4.4. Geliştirilen yöntemin çalıştırıldığı makinelerdeki süreleri

Cihaz	IPM işlemi (ms)	Şerit tespit işlemi (ms)	PM işlemi (ms)	Toplam (ms)
PC-1 (RTX2080)	10,467	39,595	0,039	50,101
PC-1 (TitanXp)	10,1	38,421	0,024	48,545
PC-2	10,79	65,311	0,04	76,141
PC-3	11,42	127,155	0,08	138,655

Tablo 4.5. Karşılaştırılan yöntemlerin PC-3 ile çalıştırıldığında elde edilen süreler

	(Gaikwad ve Lokhande, 2015) [58]	(Borkar ve diğ., 2012) [24]	(Mammeri vd., 2016) [55]	(Küçük yıldız ve Ocak, 2014) [61]	Önerilen Yöntem
Time (ms)	8520	2520	169	180	139
Fps	0,12	0,40	5,92	5,56	7,21

5. SONUÇLAR VE ÖNERİLER

Bu tez çalışması kapsamında ileri sürücü destek sistemlerinde kullanılacak bir şerit takip sistemi geliştirilmiştir. Bu sistem gerçek zamanlı çalışabilme kapasitesine sahip olup ARM mimarisine sahip Linux işletim sistemi ve Nvidia CUDA destekli ekran kartı içeren tüm platformlarda çalışabilme kapasitesine sahiptir.

Geliştirilen sistem kamera ile gerçek dünyadan alınan görüntülerinin ters perspektif dönüşümü sayesinde derin öğrenme ile test edilebilir hale getirilmiştir. Elde edilen bu görüntü, şerit işaretlerinin kabaca bulunduğu bölgeden kırılarak şerit işaretlerinin tespiti için teste sokulmaktadır. Devamında elde edilen sınırlayıcı kutulardan görsel niteliği yüksek doğrulara çevrilmektedir. Böylece başarımlı hesapları daha kolay yapılabilmektedir.

Geliştirilen yöntem genel amacı şeritten ayrılma uyarı sistemlerine takviye bir yöntem olarak önerilmektedir. Yöntem, karşılaştırılan diğer yöntemlere kıyasla daha az işlem gücü ve zaman harcamaktadır.

Şerit tespit işlemi için kameradan alınan otoyol görüntülerinde bulunan yol kırıkları, çatlaklıkları, ışık yansımaları, gölgeler gibi bozucu etkilerden etkilenmemesi amacıyla geliştirilen sistemde eğitim sırasında bu durumlarda olan görüntüler de girdi olarak verilmiştir. Yöntemde şerit tespiti için kullanılan ağ modelinin her senaryoyu karşılaması mümkün değildir. Yöntemin bu gibi senaryolara karşı daha kararlı olabilmesi için sürekli öğrenme yapılabilen yöntemlerin dahil edilmesi ileriki süreçte gerekebilmektedir.

KAYNAKLAR

- [1] Organisation Internationale des Constructeurs d'Automobiles, Production Statistics 2018, <http://www.oica.net/category/production-statistics/2018-statistics/>, (Ziyaret tarihi: 22 Mayıs 2019).
- [2] World Health Organization, Global status report on road safety 2018, <https://apps.who.int/iris/bitstream/handle/10665/276462/9789241565684-eng.pdf?ua=1>, (Ziyaret tarihi 22 Mayıs 2019).
- [3] Euro New Car Assessment Programme, Assessment Protocol, <https://cdn.euroncap.com/media/20876/euro-ncap-assessment-protocol-sa-v70.pdf>, (Ziyaret tarihi 19/05/2019).
- [4] Küçükmanisa A., Tarım G., Urhan O., Real-time Illumination and Shadow Invariant Lane Detection on Mobile Platform, *Journal of Real-time Image Processing*, DOI: 10.1007/s11554-017-0687-2.
- [5] ImageNet Large Scale Visual Recognition Challenge (ILSVRC), <http://www.image-net.org/challenges/LSVRC/>, (Ziyaret tarihi: 29 Nisan 2019).
- [6] Zitzewitz G., Survey of neural networks in autonomous driving, *Advanced Seminar SS*, 2017.
- [7] He K., Zhang Shaoqing X., Sun R., Deep Residual Learning for Image Recognition, *arXiv preprint arXiv:1512.03385*, 2015.
- [8] Szegedy C., Liu w., Jia Y., Sermanet P., Reed S., Anguelov D., Erhan D., Vanhoucke V., Rabinovich A., Going deeper with convolutions, *arXiv preprint arXiv:1409.4842*, 2014.
- [9] Tch A., The mostly complete chart of Neural Networks, Towards Data Science, <https://towardsdatascience.com/the-mostly-complete-chart-of-neural-networks-explained-3fb6f2367464>, (Ziyaret tarihi: 20 Mayıs 2019).
- [10] The PASCAL Visual Object Classes (VOC), <https://host.robots.ox.ac.uk/pascal/VOC/>, (Ziyaret tarihi: 26 Mayıs 2019).
- [11] Ouaknine A., Review of Deep Learning Algorithms for Object Detection, <https://medium.com/zylapp/review-of-deep-learning-algorithms-for-object-detection-c1f3d437b852>, (Ziyaret tarihi: 29 Mayıs 2019).
- [12] Zang J., Zhou W., Zhang G., Duan Z., Traffic Lane Detection using Fully Convolutional Neural Network, *Annual Summit and Conference 2018*, 12-15 Kasım 2018, Hawaii

- [13] Zou Q., Jiang H., Dai Q., Yue Y., Chen L., Wang Q., Robust Lane Detection from Continuous Driving Scenes Using Deep Neural Networks, *arXiv preprint arXiv:1903.02193*, 2019.
- [14] Wang Z., Ren W., Qiu Q., LaneNet: Real-Time Lane Detection Networks for Autonomous Driving, *arXiv preprint arXiv:1807.01726*, 2018
- [15] Gurghian A., Koduri T., Bailur S., Carey K., Murali J., DeepLanes: End-To-End Lane Position Estimation using Deep Neural Networks, *2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2016, DOI: 10.1109/CVPRW.2016.12.
- [16] Li J., Mei X., Prokhorov D., Tao D., Deep Neural Network for Structural Prediction and Lane Detection in Traffic Scene, *IEEE Transactions on Neural Networks and Learning Systems*, 2017, DOI: 10.1109/TNNLS.2016.2522428.
- [17] Wu Y., Yang T., Zhao J., Guan L., Jiang W., VH-HFCN based Parking Slot and Lane Markings Segmentation on Panoramic Surround View, *arXiv preprint arXiv:1804.07027*, 2018.
- [18] Huval B., Wang T., Tandon S., Kiske J., Song W., Pazhayampallil J., Andriluka M., Rajpurkar P., Migimatsu T., Cheng-Yue R., Mujica F., Coates A., Ng A., An Empirical Evaluation of Deep Learning on Highway Driving, *arXiv preprint arXiv:1504.01716*, 2015
- [19] Ren S., He K., Girshick R., Sun J., Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, *arXiv preprint arXiv:1506.01497*, 2015
- [20] Liu W., Anguelov D., Erhan D., Szegedy C., Reed S., Fu C., Berg A., SSD: Single Shot MultiBox Detector, *arXiv preprint arXiv:1512.02325*, 2015
- [21] Redmon J., Divvala S., Girshick R., Farhadi A., You Only Look Once: Unified, Real-Time Object Detection, *arXiv preprint arXiv:1506.02640*, 2015
- [22] Redmon J., Farhadi A., YOLOv3: An Incremental Improvement, *arXiv preprint arXiv:1804.02767*, 2018
- [23] Sachan A., Zero to Hero: Guide to Object Detection using Deep Learning: Faster R-CNN, YOLO, SSD, <https://cv-tricks.com/object-detection/faster-r-cnn-yolo-ssd/>, (Ziyaret tarihi: 30 Mayıs 2019).
- [24] Borkar A., Hayes M., Smith M. T., A novel lane detection system with efficient ground truth generation, *IEEE Transactions on Intelligent Transportation Systems*, 2012, **13**(1), 365–374.
- [25] Lin C., Wang M., A Vision Based Top-View Transformation Model for a Vehicle Parking Assistant, *Sensors*, 2012, **12**(4), 4431-4444.

- [26] Visvikis C., Smith T.L., Pitcher M., Smith R., Study on Lane Departure Warning and Lane Change Assistant System, <https://trl.co.uk/sites/default/files/PPR374.pdf>, (Ziyaret tarihi: 29 Mayıs 2019).
- [27] COWI, Cost-Benefit Assessment and Prioritisation of Vehicle Safety Technologies (TREN-ECON2-002), *European Commission, Directorate General Energy and Transport*, Brüksel, Belçika, 2006.
- [28] Aly M., Real time Detection of Lane Markers in Urban Streets, *2008 IEEE Intelligent Vehicle Symposium*, 2008, 7-12.
- [29] Bar Hillel A., Lerner R., D. Levi, Raz G., Recent Progress in Road and Lane Detection: A survey, *Mach. Vis. Appl.*, 2014, **25**(3), 727–745.
- [30] Thakur R., Scanning LIDAR in Advanced Driver Assistance Systems and Beyond, *IEEE Consumer Electronics Magazine*, 2016, **5**(3), 48-54.
- [31] Hata A., Wolf D., Road Marking Detection using LIDAR Reflective Intensity Data and Its Application to Vehicle Localization, *17th Int. IEEE Conf. Intell. Transp. Syst.*, 2014, 584–589.
- [32] Zhou Z., Farhat F., Wang J. Z., Detecting Dominant Vanishing Points in Natural Scenes with Application to Composition-Sensitive Image Retrieval, *arXiv preprint arXiv:1608.04267*, 2016.
- [33] Mistry H. V., Makwana R., Optimized Voting Scheme for Efficient Vanishing Point Detection in General Road Images, *International Journal of Advanced Computer Science and Applications (IJACSA)*, 2016, DOI: 10.14569/IJACSA.2016.070818.
- [34] Ding W., Li Y., Liu H., Efficient vanishing point detection method in unstructured road environments based on dark channel prior, *IET Computer Vision*, 2016, DOI: 10.1049/iet-cvi.2015.0390.
- [35] Kohavi R., A study of cross-validation and bootstrap for accuracy estimation and model selection, *14th International Joint Conference on Artificial Intelligence (IJCAI)*, Montreal, 20-25 Ağustos 1995
- [36] Refaeilzadeh P., Tang L., Liu H., Cross-Validation. In: Liu L., Özsu M.T., *Encyclopedia of Database Systems*, 1st ed., Springer, Boston, 532-538, 2009
- [37] Cross-Validation, <https://machine-learning-course.readthedocs.io/en/latest/content/overview/crossvalidation.html>, (Ziyaret tarihi: 8 Haziran 2019).
- [38] R-CNN, Fast R-CNN, and Faster R-CNN Basics, <https://www.mathworks.com/help/vision/ug/faster-r-cnn-basics.html>, (Ziyaret tarihi: 8 Haziran 2019).
- [39] Weng L., Object Detection Part 4: Fast Detection Models, <https://lilianweng.github.io/lil-log/2018/12/27/object-detection-part-4.html>, (Ziyaret tarihi: 8 Haziran 2019).

- [40] Das S., Day A., Pak A., Roy N., Applications of Artificial Intelligence in Machine Learning: Review and Prospect, *International Journal of Computer Applications*, 2015, DOI: 10.5120/20182-2402
- [41] Garnett N., Cohen R., Peer T., Lahav R., Levi D., 3D-LaneNet: end-to-end 3D multiple lane detection, *arXiv preprint arXiv:1811.10203*, 2018.
- [42] Hoffer R., Hubara I., Soudry D., Train longer, generalize better: closing the generalization gap in large batch training of neural networks, *arXiv preprint arXiv:1705.08741*, 2017.
- [43] Luong M. T., Pham H., Manning C. D., Effective approaches to attention-based neural machine translation, *arXiv preprint arXiv:1508.04025*, 2015.
- [44] Amodei D., Anubhai R., Battenberg E., Deep speech 2: End-to-end speech recognition in english and mandarin, *arXiv preprint arXiv:1512.02595*, 2015.
- [45] He, K., Zhang, X., Ren, S., Sun, J., Deep residual learning for image recognition, *arXiv preprint arXiv:1512.03385*, 2015.
- [46] Qureshi H. A., Simeonov A., Bency J. M., Yip M. C., Motion Planning Networks, *arXiv preprint arXiv:1806.05767*, 2018.
- [47] Lee J., Hwangbo J., Hutter M., Robust Recovery Controller for a Quadrupedal Robot using Deep Reinforcement Learning, *arXiv preprint arXiv:1901.07517*, 2019.
- [48] Kumar C., Artificial Intelligence: Definition, Types, Examples, Technologies, <https://medium.com/@chethankumargn/artificial-intelligence-definition-types-examples-technologies-962ea75c7b9b>, (Ziyaret tarihi: 8 Haziran 2019).
- [49] Kathuria A., What's new in YOLO v3?, <https://towardsdatascience.com/yolo-v3-object-detection-53fb7d3bfe6b>, (Ziyaret tarihi: 8 Haziran 2019).
- [50] Kukreja H., Bharath N., Siddesh C. S., Kuldeep S., An Introduction to Artificial Neural Network, *International Journal Of Advance Research And Innovative Ideas In Education*, **1**(5):27-30, 2016
- [51] O'Shea K., Nash R., An Introduction to Convolutional Neural Networks, *arXiv preprint arXiv: 1511.08458*, 2015.
- [52] Haykin S., *Neural networks and learning machines*, 3rd ed., Pearson Education, Upper Saddle River, NJ, 2009
- [53] Hijazi S., Kumar R., Rowen C., Cadence, Using Convolutional Neural Networks for Image Recognition, https://ip.cadence.com/uploads/901/TIP_WP_cnn_FINAL-pdf, (Ziyaret tarihi: 9 Haziran 2019).
- [54] Deng L. Yu D., Deep Learning: Methods and Applications, *Foundations and Trends in Signal Processing*, 2014, **7**, 197-387.

- [55] Mammeri A., Boukerche A., Tang Z., A Real-time Lane Marking Localization, Tracking and Communication System, *Computer Communications*, 2016, **73**, 132-143.
- [56] Matas J., Chum O., Urban M., Pajdla T., Robust Wide Baseline Stereo from Maximally Stable Extremal Regions, *Image and Vision Computing*, 2004, **22**(10), 761-767.
- [57] Stephens R. S., Probabilistic Approach to the Hough Transform, *Image and Vision Computing*, 1991, **9**(1), 66-71.
- [58] Gaikwad V., Lokhande S., Lane Departure Identification for Advanced Driver Assistance, *IEEE Transactions on Intelligent Transportation Systems*, 2015, **16**(2), 910-918.
- [59] Tapia-Espinoza R., Torres-Torriti M., Robust Lane Sensing and Departure Warning under Shadows and Occlusions, *Sensors*, 2013, **3**(1), 3270-3298.
- [60] Torr P. H. S., Zisserman A., MLESAC: A New Robust Estimator with Application to Estimating Image Geometry, *Computer and Vision Image Understanding*, 2000, **78**(1), 138-156.
- [61] Küçükyıldız G., Ocak H., Development and Optimization of a DSP-based Real-time Lane Detection Algorithm on a Mobile Platform, *Turkish Journal of Electrical Engineering and Computer Sciences*, 2012, **22**(6), 1484-1500
- [62] Bahmani M. J., AI vs Machine Learning vs Deep Learning, <https://medium.com/datadriveninvestor/ai-vs-machine-learning-vs-deep-learning-ba3b3c58c32>, (Ziyaret tarihi: 19 Haziran 2019).
- [63] Shah T., About Train, Validation and Test Sets in Machine Learning, <https://towardsdatascience.com/train-validation-and-test-sets-72cb40cba9e7>, (Ziyaret tarihi: 9 Haziran 2019).

KİŞİSEL YAYINLAR VE ESERLER

- [1] Küçükmanisa A., **Tarım G.**, Urhan O., Real-time Illumination and Shadow Invariant Lane Detection on Mobile Platform, *Journal of Real-time Image Processing*, DOI: 10.1007/s11554-017-0687-2.
- [2] Küçükmanisa A., **Tarım G.**, Urhan O., Akıllı Telefonlar İle İleri Sürücü Destek Sistemleri, *7.Karayolları Trafik Güvenliği Sempozyumu*, Ankara, Türkiye, 17-19 Kasım 2016.
- [3] **Tarım G.**, Küçükmanisa A., Urhan O., Gömülü Platformlar için Kuşbakışı Görüntüler Üzerinden Derin Öğrenme Temelli Gerçek Zamanlı Şerit İşareti Tespit Sistemi, *Uluslararası Marmara Fen ve Sosyal Bilimler Kongresi*, Kartepe, Kocaeli, 26-28 Nisan 2019.

ÖZGEÇMİŞ

1991 yılında Erzurum’da doğdu. 1995 yılında Yalova’ya taşındı. İlköğretim eğitimini Yalova Atatürk İlköğretim Okulunda tamamladıktan sonra liseyi Yalova Anadolu Lisesinde tamamlamıştır. 2010 ve 2015 yılları arasında Kocaeli Üniversitesi Mühendislik Fakültesi Elektronik ve Haberleşme Mühendisliği Bölümü’nden mezun oldu. Lisans tezi olarak Görünür ışık haberleşmesi ile yüksek hızda görüntü ve veri aktarımı adlı projeyi 2015 yılında yapmıştır. Bu süre zarfında Kocaeli Üniversitesi Elektronik ve Haberleşme Mühendisliği bölümü KULE laboratuvarında araştırmacı öğrenci olarak bir çok çalışmada bulunmuştur. 2019 yılına kadar Kocaeli Üniversitesi spin-off şirketi olan EVSTEK Bilişim Teknolojilerinde Yazılım geliştirme mühendisi olarak çalıştıktan sonra yaklaşık 6 aydır CTECH Bilişim Teknolojilerinde Gömülü yazılım mühendisi olarak çalışmasına devam etmektedir.