

**T.C.
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**



**AKILLI ULAŞIM SİSTEMLERİ İÇİN KARAYOLU TİPİ,
KAVŞAK VE VİRAJLARIN DERİN ÖĞRENME YÖNTEMLERİ
İLE BELİRLENMESİ**

Vedat TÜMEN

Doktora Tezi

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

Kuramsal Temeller Bilim Dalı

ARALIK 2019

T.C.
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

Bilgisayar Mühendisliği Anabilim Dalı

Doktora Tezi

**AKILLI ULAŞIM SİSTEMLERİ İÇİN KARAYOLU TİPİ, KAVŞAK VE VİRAJLARIN DERİN
ÖĞRENME YÖNTEMLERİ İLE BELİRLENMESİ**

Tez Yazarı
Vedat TÜMEN

Danışman
Doç. Dr. Burhan ERGEN

ARALIK 2019
ELAZIĞ

T. C.
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

Bilgisayar Mühendisliği Anabilim Dalı

Doktora Tezi

Başlığı: **AKILLI ULAŞIM SİSTEMLERİ İÇİN KARAYOLU TİPİ, KAVŞAK VE
VİRAJLARIN DERİN ÖĞRENME YÖNTEMLERİ İLE BELİRLENMESİ**

Yazarı: Vedat TÜMEN

İlk Teslim Tarihi: 3.12.2019

Savunma Tarihi: 27.12.2019

TEZ ONAYI

Fırat Üniversitesi Fen Bilimleri Enstitüsü tez yazım kurallarına göre hazırlanan bu tez aşağıda imzaları bulunan jüri üyeleri tarafından değerlendirilmiş ve akademik dinleyicilere açık yapılan savunma sonucunda OYBİRLİĞİ ile kabul edilmiştir.

Danışman:	Doç. Dr. Burhan ERGEN Fırat Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği	 İmza	Onayladım
İkinci Danışman:			Onayladım
Başkan:	Prof. Dr. Mehmet KAYA Fırat Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği		Onayladım
Üye:	Doç. Dr. Murat KARABATAK Fırat Üniversitesi, Teknoloji Fakültesi, Yazılım Mühendisliği		Onayladım
Üye:	Doç. Dr. Tuğrul ÇAVDAR Karadeniz Teknik Üniversitesi, Mühendislik Fakültesi, Bilgisayar Müh.		Onayladım
Üye:	Dr. Öğr. Üyesi Abidin ÇALIŞKAN Batman Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği		Onayladım
Üye:	Unvan Adı SOYADI ... Üniversitesi, ... Fakültesi		Onayladım

Bu tez, Enstitü Yönetim Kurulunun/...../20..... tarihli toplantısında tescillenmiştir.

İmza

Prof. Dr. Soner ÖZGEN
Enstitü Müdürü

BEYAN

Fırat Üniversitesi Fen Bilimleri Enstitüsü tez yazım kurallarına uygun olarak hazırladığım “Akıllı Ulaşım Sistemleri için Karayolu Tipi, Kavşak ve Virajların Derin Öğrenme Yöntemleri ile Belirlenmesi ” Başlıklı Doktora Tezimin içindeki bütün bilgilerin doğru olduğunu, bilgilerin üretilmesi ve sunulmasında bilimsel etik kurallarına uygun davrandığımı, kullandığım bütün kaynakları atıf yaparak belirttiğimi, maddi ve manevi desteği olan tüm kurum/kuruluş ve kişileri belirttiğimi, burada sunduğum veri ve bilgileri unvan almak amacıyla daha önce hiçbir şekilde kullanmadığımı beyan ederim.

27/12/2019

Vedat TÜMEN



ÖNSÖZ

Sunulan doktora tez çalışmasında amaç, gerçek karayolu görüntülerinden elde edilmiş veriler üzerinden yeni bir görüntü işleme teknikleri ve derin öğrenme yöntemleri uygulanarak anlamlı sonuçlar ortaya çıkarmaktır. Önerilen yöntemde görüntü işleme aşamasında karayolunun ilgili bölgesine odaklanmış daha sonra elde edilen görüntünün geliştirilen yeni bir derin öğrenme yöntemi ile yüksek başarımlarla tanınması sağlanmıştır. Deneysel sonuçlar literatürde bilinen eşdeğer yöntemler kullanılarak test edilmiştir. Aynı zamandan var olan çalışmalar ile karşılaştırılmıştır. Uzun ve yorucu bir çalışmanın ürünü olarak ortaya çıkan bu çalışmada birçok yöntem ve uygulama geliştirilmiştir.

Doktora çalışmalarım kapsamında bana kendisiyle çalışma fırsatı vererek önümde yeni ufuklar açan, manevi desteğini, bilgi ve tecrübelerini eksik etmeyen, akademik çalışma ve disiplini kazanmamı sağlayan, sonsuz sabrı ve anlayışı dolayısıyla değerli danışman hocam Doç. Dr. Burhan ERGEN'e, tez çalışmasında verdiği sonsuz sevgi ve moralle her zaman yanımda olan kıymetli eşime ve çocuklarım Berhan Emir ve Mirza Ali'ye, tüm hayatım boyunca beni her konuda destekleyen, sevgi ve yardımlarını esirgemeyen aileme sonsuz teşekkürlerimi sunarım.

Vedat TÜMEN
ELAZIĞ, 2019

İÇİNDEKİLER

	Sayfa
ÖNSÖZ.....	iv
İÇİNDEKİLER	v
ÖZET	vii
ABSTRACT	viii
ŞEKİLLER LİSTESİ	ix
TABLolar LİSTESİ	xi
SİMGELER VE KISALTMALAR	xii
1. GİRİŞ	1
1.1. Tezin Amacı ve Katkıları	3
1.2. Tezin Yapısı.....	4
2. AKILLI ŞEHİRLERDE AKILLI ULAŞIM SİSTEMLERİ	5
2.1. Akıllı Ulaşım Sistemleri	5
2.2. Akıllı Ulaşım Sistemlerinde Görüntüleme, Algılama ve Tanıma Sistemleri.....	6
2.2.1. Elektronik Denetleme Sistemleri.....	7
2.2.2. Mobil Elektronik Sistem Entegrasyonu.....	8
2.2.3. Kavşak Kontrol Sistemleri	9
2.2.4. Yol Kenarı Kamera Tanıma Sistemi	9
2.2.5. Algılama Sensörleri.....	9
2.3. Akıllı Ulaşım Sistemlerinde Kullanılan Teknolojiler	10
2.3.1. İleri Sürücü Destek Sistemleri.....	10
2.3.2. Otonom Araç Sistemleri.....	11
3. GÖRÜNTÜ İŞLEME TEKNİKLERİ VE YAPAY ZEKA ÖĞRENME MODELLERİ.....	15
3.1. Görüntü İşleme Yöntemleri	15
3.1.1. Görüntüleri Yeniden Boyutlama	16
3.1.2. Görüntüde Alan Bulma	16
3.1.3. Görüntüde Kenar Tespit Algoritmaları	17
3.1.4. Görüntülerde İlgili Bölge için Kırpma	17
3.2. Görüntülerin Veri Setlerine Dönüştürülmesi	18
3.3. Yapay Zeka ve Makine Öğrenmesi Yöntemleri	19
3.3.1. Denetimli Öğrenme	19
3.3.2. Denetimsiz Öğrenme.....	20
3.3.3. Takviyeli Öğrenme.....	21
3.4. Klasik Makine Öğrenmesi Yöntemleri	21
3.4.1. Destek Vektör Makinesi.....	22
3.4.2. K-En Yakın Komşu Algoritması	23
3.4.3. Çok Katmanlı Algılayıcı	25
3.5. Derin Öğrenme	26
3.6. Derin Öğrenme Mimarileri.....	28
3.6.1. Boltzmann Makineleri.....	28
3.6.2. Kısıtlanmış Boltzmann Makineleri.....	30
3.6.3. Tekrarlayan Sinir Ağı.....	31
3.6.4. Uzun-Kısa Süreli Hafıza (Long Short-Term Memory - LSTM)	33

3.6.5. Evrişimsel Sinir Ağları (Convolutional Neural Networks - CNN).....	33
3.7. Evrişim Sinir Ağları Tabanlı Geliştirilen Derin Öğrenme Modelleri	41
3.7.1. AlexNet	41
3.7.2. LeNet.....	42
3.7.3. VggNet	43
3.7.4. ResNet	44
3.7.5. YOLO ve DarkNet	45
4. TRAFİKTE GÖRÜNTÜ İŞLEME VE GELİŞTİRİLEN MODEL.....	48
4.1. Geliştirilen Yol Tipini Tanıma Yöntemi	48
4.2. Geliştirilen Yöntemin Görüntü İşleme Aşaması.....	49
4.2.1. İlgilenilen Bölgenin Belirlenmesi	50
4.3. Geliştirilen Yöntemin Tanıma Aşaması	56
4.4. Eğitim, Doğrulama ve Test İşlemleri.....	57
4.4.1. Değerlendirme Kriterleri	57
4.4.2. Tanıma ve Sınıflandırma	58
5. UYGULAMA	59
5.1. Yol Tipi ve Yol Kalitesinin Tespiti	59
5.1.1. Yol Tipi ve Yol Kalitesinin Tespiti için Önerilen Model.....	61
5.1.2. Modelin Yol Tipini Tespiti için Değerlendirilmesi	63
5.1.3. Modelin Yol Kalitesi Tespiti için Değerlendirilmesi	65
5.2. Kavşak Tespiti.....	68
5.2.1. Kavşak Tespiti için Önerilen Model.....	69
5.2.2. Modelin Kavşak Tespiti için Değerlendirilmesi.....	70
5.3. Sağ ve Sol Yol Ayrımının Tespiti	74
5.3.1. Modelin Sol ve Sağ Yol Ayrımları Tespiti için Değerlendirilmesi	75
5.4. Yaya Geçitlerinin Tespiti.....	79
5.4.1. Modelin Yaya Geçitlerinin Tespiti için Değerlendirilmesi	79
5.5. Sağ ve Sol Tehlikeli Virajların Tespiti	82
5.5.1. Sağ ve Sol Tehlikeli Virajların Tespiti için Önerilen görüntü İşleme ve Tespit Yöntemi... 83	
5.5.2. Modelin Sağ ve Sol Tehlikeli Virajların Tespiti için Değerlendirilmesi.....	85
5.6. Videolarda bulunan karayolu parametrelerinin tespiti.....	86
5.6.1. YOLO Algoritması Kullanarak Yol Durumu ve Trafik İşareti Tespiti	88
6. SONUÇLAR.....	91
ÖNERİLER	93
KAYNAKLAR.....	94
ÖZGEÇMİŞ	

ÖZET

Akıllı Ulaşım Sistemleri için Karayolu Tipi, Kavşak ve Virajların Derin Öğrenme Yöntemleri ile Belirlenmesi

Vedat TÜMEN

Doktora Tezi

FIRAT ÜNİVERSİTESİ

Fen Bilimleri Enstitüsü

Bilgisayar Mühendisliği Anabilim Dalı

Aralık 2019, Sayfa: xii + 99

Son günlerde akıllı ulaşım sistemleri alanında geliştirilen otonom araçlar ve ileri sürücü destek sistemleri için yeni fikirler ve senaryolar üretilmektedir. Otonom araçlar ya da sürücü destek sistemlerinde, karayolunda tanıma işlemi için çoğunlukla bilgisayar görü teknikleri kullanılmakta ya da lazer sinyal işleme teknikleri uygulanmaktadır. Görüntü işleme yöntemleri, lazer sinyal işleme yöntemlerine göre daha düşük maliyetli çözümler üretebilmektedir. Görüntü işleme tekniklerine ek olarak derin öğrenme yöntemleri kullanılarak nesne ve yol tipinin tespiti yapılabilir.

Bu tez çalışmasında, karayolunda bulunan kavşaklar, yaya geçitleri, virajlar, yol ayrımları gibi önemli bölgelerin gerçek görüntüleri ilk defa bir tanıma problemi olarak ele alınmış, görüntü işleme ve derin öğrenme metoduyla tespit edilmiştir. Görüntü işleme aşamasında, yol görüntülerinin yapısı ve çekimine göre üst ve alt noktaları kırılarak ilgili bölgenin tespiti için bir metot geliştirilmiş ve elde edilen görüntülerin derin öğrenme mimarisi ile tanınması sağlanmıştır.

Geliştirmiş olduğumuz yöntem ve farklı tanıma algoritmaları ile görüntüler test edilmiştir. Modelimizin hem algoritmalara göre hem de literatürdeki farklı veri setleri kullanılarak geliştirilen çalışmalara göre oldukça iyi sonuçlar verdiği görülmüştür. Kamera görüntüleri kullanılarak gerçek zamanlı sistemler hazırlanabilir ve uygun maliyetli çözümler sunulabilir. Geliştirilen bu yöntemin sürücü destek sistemleri için uygulanabilir olduğu, hem araçlar hem de sürücülerin uyarılması gibi birçok alanda kullanılabilecek etkili bir yapı olduğu söylenebilir. Ayrıca bu çalışma, karayollarında bulunan trafik işaretleri ile gerçek yol durumlarının eşleşip eşleşmediği gibi tespit ve doğrulama sistemleri için de kullanılabilir.

Anahtar Kelimeler: Karayolu tipi tespiti, Kavşak tespiti, Viraj tespiti, Yaya geçidi tespiti, Görüntü işleme, Derin öğrenme

ABSTRACT

Determination of Highway Type, Intersections and Curves for Intelligent Transportation Systems by Deep Learning Methods

Vedat TMEN

Ph.D. Thesis

FIRAT UNIVERSITY
Graduate School of Natural and Applied Sciences
Department of Computer Engineering

December 2019, Pages: xii + 99

Recently, new ideas and scenarios have been produced for autonomous vehicles and advanced driver assistance systems developed in the field of intelligent transportation systems. In autonomous vehicles or driver assistance systems, computer vision techniques are often used for road recognition, or laser signal processing techniques are applied. Image processing methods can produce lower cost solutions than laser signal processing methods. In addition to image processing techniques, the object and path type can be determined using deep learning methods.

In this thesis, real images of important regions such as intersections, pedestrian crossings, bends, left-right intersections on the road have been dealt with for the first time as a recognition problem and determined by image processing and deep learning method. In the image processing stage, a method has been developed for the determination of the relevant region by clipping the top and bottom points according to the structure and attraction of road images and the acquired images were recognized with deep learning architecture.

Images have been tested with the method we developed and different recognition algorithms. It has been found that our model gives very good results both according to algorithms and studies developed using different datasets in the literature. Using camera images, real-time systems can be prepared and cost-effective solutions can be provided. It can be said that this developed method is applicable to driver support systems and it is an effective structure that can be used in many fields such as warning of vehicles and drivers. Also, this study can be used for detection and verification systems as well as whether traffic signs on highways match actual road conditions.

Keywords: Road type detection, Intersection detection, Curve detection, Crosswalk detection, Image processing, Deep learning

ŞEKİLLER LİSTESİ

	Sayfa
Şekil 2.1. Elektronik Denetleme Sistemi.....	7
Şekil 2.2. Mobese kamera sistemi	8
Şekil 2.3. Otonom araçlar	13
Şekil 3.1. RGB kanallı ve gri formatlı görüntülerin matris olarak gösterimi	15
Şekil 3.2. Görüntüde tanıma yöntemi ile parça-alan bulma-tanımlama	16
Şekil 3.3. Görüntü işleme aşamaları	18
Şekil 3.4. Denetimli öğrenme modelinin çalışma prensibi	20
Şekil 3.5. Denetimsiz öğrenme modelinin çalışma prensibi	20
Şekil 3.6. Takviyeli öğrenme modelinin çalışma prensibi.....	21
Şekil 3.7. Destek vektör makinalarının çalışma prensibi.....	22
Şekil 3.8. k-NN algoritmasının k=5,7 durumunda verilerin sınıflanması.....	24
Şekil 3.9. Örnek bir derin öğrenme mimarisi	27
Şekil 3.10. Boltzmann makinesi mimarisi.....	28
Şekil 3.11. Boltzmann makinesi tek yapay sinir hücresi	29
Şekil 3.12. Kısıtlı Boltzmann makinesi mimarisi	30
Şekil 3.13. Kısıtlı Boltzmann Makinesi sinir hücreleri yapısı	30
Şekil 3.14. Basit tekrarlayan ağ mimarisi	32
Şekil 3.15. Tekrarlayan sinir ağının açılması	32
Şekil 3.16. Tekrarlayan sinir ağı modeli.....	33
Şekil 3.17. LSTM bloğu	33
Şekil 3.18. Evrişimsel sinir ağları mimarisi.....	34
Şekil 3.19. Konvolüsyon işlem aşamaları	35
Şekil 3.20. Pooling katmanı örneği	36
Şekil 3.21. Tüm aktivasyon fonksiyonların grafikte gösterimi.....	40
Şekil 3.22. ResNet mimarisi ile normal mimari modelleri farkı.....	45
Şekil 3.23. YOLO nesne sınırları tahmini	45
Şekil 4.1. Geliştirilen sistemin prensip şeması	48
Şekil 4.2. Geliştirilen sistemin işlem aşamaları	49
Şekil 4.3. Görüntülerin veritabanına eklenmesi aşamaları	50
Şekil 4.4. VP metoduna göre karayolu görüntülerinin üst bitim noktasının tespiti ve kırılması	51
Şekil 4.5. Görüntülerde Canny kenar çıkarımı ve üst bitim noktasının tespit edilmesi.....	52

Şekil 4.6. Alt kesim noktasının tespit edilmesi.....	53
Şekil 4.7. Karayolu ve ilgisiz bölgelerin ufuk noktası kullanılarak bölgelere ayrılması	54
Şekil 4.8. Geliştirilen yöntemin karayolu tiplerine göre tespit edilen RoI bölgeleri	54
Şekil 4.9. Çalışmada karayolu bileşeni tanıma için geliştirilen yöntemin blok şeması	55
Şekil 4.10. Çalışmalar için kullanılan görüntülerin örnekleri.....	56
Şekil 4.11. Çalışmalar için kullanılan görüntülerin örnekleri.....	56
Şekil 5.1. Yol tipi ve kalitesi için hazırlanan görüntüler	60
Şekil 5.2. Yol tipi ve kalitesinin tespiti için önerilen yöntemin prensip şeması	61
Şekil 5.3. Yol tipi ve kalitesinin tespiti için önerilen model	62
Şekil 5.4. Yol tipi sınıflandırmanın performans grafiği.....	63
Şekil 5.5. Yol kalitesi sınıflandırma performans grafiği	66
Şekil 5.6. Bozuk- çukur yol veri setinden örnek görüntüler	66
Şekil 5.7. Kavşak tespiti için hazırlanmış önerilen model.....	69
Şekil 5.8. Modelin başarımlar grafiği ve kayıp(loss) fonksiyonu.....	71
Şekil 5.9. Bölme-4'e göre CNN modellerinin başarı oranları	72
Şekil 5.10. Modelin başarımlar grafiği	75
Şekil 5.11. Sol-sağ ve normal yol türünde örnek görüntüler	76
Şekil 5.12. Geliştirilen modelin normal yol ile sol – sağ yol ayrımları görüntülerin başarısının karşılaştırma matrisi	77
Şekil 5.13. Veri seti-5'te kullanmış olduğumuz örnek yaya geçitleri görüntüleri	79
Şekil 5.14. Önerilen modelin yaya geçidi tespitinde eğitim ve test aşamalarında göstermiş olduğu başarı grafiği.....	80
Şekil 5.15. Karayollarında bulunan tehlikeli viraj işaretleri	82
Şekil 5.16. Tehlikeli viraj için hazırlanan veri set	83
Şekil 5.17. Görüntülerin dönüşüm aşamaları	83
Şekil 5.18. Önerilen modelin Veri seti-6 üzerindeki eğitim ve test aşamalarının performans grafiği.....	85
Şekil 5.19. Önerilen modelin Veri seti-6 üzerindeki eğitim ve test aşamalarının performans grafiği.....	87
Şekil 5.20. reCAPTCHA spam önleme hizmeti	88
Şekil 5.21. Görüntüleri YOLO algoritmasında değerlendirme aşamaları	89
Şekil 5.22. Videolar üzerinde YOLO algoritması ile tanıma işlemi	90

TABLÖLAR LİSTESİ

Tablo 2.1. Otonom araç seviyeleri	12
Tablo 3.1. AlexNet derin öğrenme modelinin katman parametreleri	42
Tablo 3.2. LeNet derin öğrenme modelinin katman parametreleri	43
Tablo 3.3. VggNet-19 derin öğrenme modelinin katman parametreleri	43
Tablo 3.4. ResNet derin öğrenme modelinin katman parametreleri	44
Tablo 3.5. Darknet-53 mimarisinin katman parametreleri	46
Tablo 5.1. Hazırlanan veri setleri.....	59
Tablo 5.2. Veri seti-1’te bulunan yol tipleri görüntü dağılımları	63
Tablo 5.3. Veri seti-1 üzerinde yapılan CNN modellerinin başarımlar oranları	64
Tablo 5.4. Geliştirdiğimiz modelin başarısının literatürdeki diğer çalışmalarla karşılaştırılması	64
Tablo 5.5. Veri seti-2’de bulunan yol tipleri görüntü dağılımları	65
Tablo 5.6. Veri seti-2 üzerinde yapılan CNN modellerinin başarımlar oranları	67
Tablo 5.7. Geliştirdiğimiz modelin başarısının literatürdeki diğer çalışmalarla karşılaştırılması	68
Tablo 5.8. Kavşak, yol ayrımı ve yaya geçitlerinin tespiti için önerilen CNN modelinin parametreleri	70
Tablo 5.9. Kullanılan yol tiplerinin sayıları	71
Tablo 5.10. Farklı eğitimli modeller kullanılarak elde edilen sınıflandırma sonuçları ve ortalamaları	72
Tablo 5.11. CNN modellerinin kavşak tespiti için bölüm 4 veri setindeki performans sonuçları.....	73
Tablo 5.12. Kavşak veri setindeki CNN modellerinin ortalama performansı	73
Tablo 5.13. Geliştirdiğimiz modelin başarısının literatürdeki diğer çalışmalarla karşılaştırılması	74
Tablo 5.14. Kullanılan yol tiplerinin sayıları	75
Tablo 5.15. Önerilen modeli kullanarak yol ayrımlarının detaylı sınıflandırma performansları	76
Tablo 5.16. Modelimiz ve diğer CNN modellerinin yol ayrımları tespit sonuçları	77
Tablo 5.17. Sol ve Sağ kavşak tespiti için geliştirdiğimiz modelin literatürdeki diğer çalışmalarla karşılaştırılması	78
Tablo 5.18. Kullanılan yol tiplerinin sayıları	80
Tablo 5.19. Önerilen modeli kullanarak yaya geçitlerinin detaylı sınıflandırma performansları.....	80
Tablo 5.20. Yaya geçidi veri setindeki CNN modellerinin sınıflandırma performansları	81
Tablo 5.21. Yaya geçidi ile ilgili literatürdeki çalışmaların sonuçların karşılaştırılması	81
Tablo 5.22. Veri seti -6 da bulunan verilerin eğitim ve test olarak dağılım oranları.....	84
Tablo 5.23. Önerilen derin öğrenme tabanlı modelin katmanları ve özellikleri.....	84
Tablo 5.24. Virajların tespiti ile ilgili literatürdeki çalışmaların sonuçların karşılaştırılması	86

SİMGELER VE KISALTMALAR

Kısaltmalar

ADAS	: Advanced Driver Assistance Systems - İleri sürüş destek sistemleri
ALVINN	: Autonomous Land Vehicle in a Neural Network
Ar-Ge	: Araştırma Geliştirme
AUS	: Akıllı Ulaşım Sistemleri
AVNS	: Autonomous Navigation Systems - Otonom Navigasyon Sistemleri
AVRI	: Autonomous Vehicles Readiness Index- Otonom Araç Hazırlık Endeksi
BM	: Boltzmann Machines
CNN	: Convolutional Neural Networks
DVM	: Destek Vektör Makinesi
EDS	: Elektronik Denetleme Sistemleri
ELU	: Exponential Linear Unit
ESA	: Evrimsel Sinir Ağları
FC	: Fully Connected - Tam Bağlı katman
GPS	: Global Positioning System - Küresel Konum belirleme sistemi
GPU	: Graphics Processing Unit
GSV	: Google Street View
IoT	: Nesnelerin İnterneti
ITS	: Intelligent Transportation Systems - Akıllı Ulaşım Sistemleri
KGM	: Karayolları Genel Müdürlüğü
KKS	: Kavşak Kontrol Sistemleri
k-NN	: k- Nearest Neighbor - k en yakın komşu algoritması
LiDAR	: Laser Imaging Detection and Ranging
LSTM	: Long Short-Term Memory- Uzun-Kısa Süreli Hafıza
LTE	: Long-Term Evolution - Yüksek hızlı kablosuz iletişim
MLP	: Multi Layer Perceptron
MNIST	: Modified National Institute of Standards and Technology dataset
MOBESE	: MOBil Elektronik Sistem Entegrasyonu
NHTSA	: National Highway Traffic Safety Administration
RBM	: Restricted Boltzmann Machines
ReLU	: Rectified Linear Unit
RGB	: Red Green Blue
RNN	: Recurrent Neural Network- Tekrarlayan Sinir Ağı
RoI	: Region of Interest - İlgili bölge
SVM	: Support Vector Machine
V2I	: Vehicle to Infrastructure
V2V	: Vehicle to Vehicle
V2X	: Vehicle to Everything
VOC	: Visual Object Classes Dataset
VP	: Vanishing Point
WiBro	: Wireless Broadband - Geniş bant kablosuz internet teknolojisi
WiMAX	: Worldwide Interoperability for Microwave Access - Kablosuz hızlı iletişim aracı
YOLO	: You Look Only Once
YSA	: Yapay Sinir Ağları

1. GİRİŞ

Günümüzde gelişen teknolojilerin ana hedeflerinin başında otonom araçlar ve sürücü destek sistemleri gelmektedir. Otonom ve sürücü destek sistemlerinin kullanacağı veriler, karayollarında sürüş ortamlarından elde edilmekte ve bu veriler tanıma işlemine kullanılabilir. Otonom sistemler, insan-robot etkileşimli sistemler, Otonom araç yönlendirme sistemleri (Autonomous Vehicle Navigation Systems - AVNS), Akıllı ulaşım sistemleri (Intelligent Transportation Systems - ITS) gibi birçok uygulamada kullanılabilir önemli bir konu haline gelmiştir [1-3]. Akıllı ulaşım sistemleri (AUS) altyapı, araç, kullanıcılar ve trafik yönetimi de dahil olmak üzere karayolu taşımacılığının iletişim teknolojileri alanında uygulanan sistemler bütünüdür.

Son yıllarda “akıllı ulaşım sistemleri”, “akıllı trafik yönetim sistemleri” geliştirilmekte ve uygulanmaktadır. Bu sistemler yol ağında dolaşımın sürekliliğinin sağlanması için sürücüleri bilgilendirme, ıslıklandırma ve yönlendirme uygulamalarını içermektedir. Söz konusu çalışmaların başarılı biçimde uygulanabilmesi için yol ağının sistem özelliklerine, geometrik özelliklere ve taşıma kapasitesine sahip olması gerekmektedir.

Günümüzde karayolu araçlarının kullanımının hızlı bir şekilde artması ile özellikle şehir içinde ulaşım hizmetleri karayolu araçları ile karşılanmaktadır. Bu karayolları zamanla akıllı ulaşım sistemlerinin dahil edildiği yollara dönüştürülecektir. Son zamanlarda, gelişmiş ülkeler yatırımlarının çoğunu akıllı sistemler üzerinde toplamıştır. Akıllı ulaşım sistemleri hem trafik kazalarını minimum boyuta indirirken hem de büyük oranda ekonomik katkılar sağlamaktadır. Uluslararası bir değerlendirme şirketi olan KPMG’nin hazırlamış olduğu Otonom Araç Hazırlık Endeksi (Autonomous Vehicles Readiness Index-AVRI), (25 ülkenin özerk araçlara hazırlık seviyesini ölçmeye yardımcı bir araçtır) çeşitli kaynaklardan gelen 25 bireysel ölçümü tek bir puanda birleştiren bileşik bir endekstir. AVRI’ nin 2019 raporuna göre, seviye 5 otonom araçların kullanım altyapısına sahip olan ülkeleri sıralanmış ve Hollanda %25.05 oranla ilk sırada yer almıştır. Ülkemiz ise herhangi bir sıralamaya giremezken, Brezilya %6.41 ile 25. olmuştur[4]. Gelişmiş toplumlarda otonom araçlar için teknolojiler hızla geliştirilmekte ve kullanılmaktadır. Otonom araçların karayollarında yer almaları için altyapının mümkün olması gerekmektedir. Altyapı ile haberleşen sistemler otonom sürüşü mümkün hale getirmektedir.

Ulaşımın altyapısı olan karayollarının AUS sistemlerine dönüştürülmesi gerekmektedir. Günümüzde oluşan trafik kazaları ile yılda binlerce insan hayatını kaybetmekte, aynı zamanda ciddi boyutlarda bir ekonomik kayıp yaşanmaktadır. Bu kayıpların minimuma indirilmesi için araçların yeni teknolojiler ile güçlendirilmesinin yanında, altyapının da düzenlenmesi gerekmektedir. Karayollarında yayaların, araçların ve diğer unsurların karşılaşma ihtimalleri çoğunlukla kavşak, yol ayrımları, virajlar ve yaya geçitlerinde olmaktadır. Trafik kazaları da bu noktalarda daha sık görülmektedir. Amerika Ulusal Karayolu Trafik Güvenliği Başkanlığının (U.S.

Department of Transportation, National Highway Traffic Safety Administration -NHTSA) yayınlamış olduđu 2015 raporuna göre, meydana gelen ölümlü trafik kazalarının, %24'ü kavşaklarda, %15'i yaya geçitlerinde oluşmaktadır [5]. Trafik kazalarının bu alanlarda oluşma durumu da yıllara bakıldığında sürekli artmaktadır. Bu tür yerlerde sürücülerin veya araçların uyarılarak karar mekanizmalarının devreye girmesi istenmektedir. Otonom araçların en fazla kavşaklarda, yol ayrımlarında ve insanlarla sık karşılaşma bölgelerinde karar almada zorlandıkları görülmüştür. Otonom araçlar, karayollarındaki trafik işaretlerini algılayarak karar mekanizmalarında değerlendirir. Giriş değerleri sadece trafik işaretleri ile sınırlı değil aynı zamanda karayollarında bulunan yol çizgileri, hareket – park halindeki araçlar, yayalar ve yol kenarlarında bulunan diğer nesnelerdir. Karayolu üzerinde bulunan ve çoğunlukla şerit bilgisi içeren sürekli ve kesikli yol çizgileri aracın hareket konumunu ayarlamak için kullanılmakta ya da şerit ihlallerini tespit etmede ve şerit değişikliğinde kullanılmaktadır.

Yüksek hızda hareket eden araçlar için yolun durumunu belirten uyarı sistemlerinin önceden bırakılması oldukça önemlidir. Yollarda bulunan sol ve sağ tehlikeli virajların, yoldaki dönüşlerin, kavşak, yol ayrımları ve buna benzer önemli yol alanlarının tespiti için bu bölgelere uyarı amacıyla trafik işaretlerinin bırakılması gerekmektedir. Ülkemizdeki karayolları yönetmeliğine göre, trafik levhalarının yol kenarlarına bırakılması için bir dizi standartlar belirlenmiştir. Bu standartlar yolun tipine göre değişmektedir. Kavşak ve yol ayrımları için görüş alanı standartları belirlenmiştir: “Kavşaklardaki komşu yollarda, kavşak başlangıcından itibaren, Karayolları Genel Müdürlüğünün (KGM) yapım ve bakım aşında bulunan yollarda en az 150 metre ve diğer yollarda en az 50 metre mesafede taşıt yolunun tesis tarafındaki kenar çizgileri üzerinde alınacak iki noktanın birleştirilmesi ile bulunacak doğru parçasının, kavşak tarafında taşıt yolu kenar çizgileri ile teşkil ettiğı alandır” [6]. Bu tarife göre şehir dışındaki karayollarında bulunan tehlikeli virajlara, kavşaklara varmadan en az 150 metre, şehir içinde bulunan karayollarında ise en az 50 metre mesafede uyarı levhaları yerleştirilmesi gerekmektedir. Bu trafik işaret levhaları yol yapımı sırasında bu standartlara yakın olarak yerleştirilmektedir. Fakat bu levhalar zamanla bozulmakta, yıpranmakta ve bakımları yapılmamaktadır. Karayollarında trafik işaret levhalarının, bu nedenlerden dolayı tespit edilememesi otonom araçlar ya da sürücü destek sistemlerinin karar almasında büyük sorunlara sebep olmaktadır. Bu sorunların giderilmesi için trafik levhasının yanında gerçek yol durumunun tespiti ile karar mekanizmaları harekete geçirilebilir. Erken uyarı sistemleri ile gerçek yol görüntülerinin tespiti yapılarak tanıma işlemi yapılmış olur.

Karayollarında bölge-nesne tespiti yapması oldukça zor ve zahmetli bir iştir. Özellikle şehir içi sürüş ortamları, pek çok nesne, tutarsızlık ve ayrıntılı trafik akışı kalıplarıyla daha karmaşık hale gelmekte ve öngörülmezlik özelliğine sahip olmaktadır[7].

Karayolunun çizgi alanını belirlemek için bitiş noktasına ve paralel yol çizgileri tespitine dayalı algılama yaklaşımları kullanılmıştır [8, 9]. Fakat bu yöntemler, iyi döşenmiş yollarda yüksek

oranda doğru algılama sağlarken, yüksek nesne karmaşıklığına sahip yollarda istenen başarı elde edilememiştir. Ayrıca, bu yöntemler arka plan gürültüsüne duyarlıdır ve araçlar, yayalar ve diğer engeller gibi karmaşık yol sahnesinde beklenen performansı gösteremez.

Otonom araçlar veya sürücü destek sistemleri, bakımları düzgün olan iyi asfaltlanmış karayollarında test edilmektedir. Bu test çoğunlukla başarıyla sonuçlanmaktadır. Fakat gerçek trafik şartlarında test sınırlı tutulmaktadır. Gerçek yol koşullarında trafik işaretleri ve yol işaret bilgileri eksik-yanlış olabilmektedir. Otonom sistemler ya da sürücü destek sistemleri bulunan araçlar her yol şartlarında tanıma yapabilmelidir. Bu tez kapsamında karayolun varlığı belirlendikten sonra yaya geçidi, dönel kavşak, sağa veya sola yol ayrımları, T yol ayrımı (sağa ve sola ayırım) noktaları, U dönüşlü kavşaklar, sağ ve sol tehlikeli virajlar gibi önemli ve önceden tespit edilmesi gereken temel yol bileşenleri görüntüden elde edilecektir.

Bu tez çalışmasının amacı, karayolu üzerinde bulunan trafik uyarı-işaret levhalarının görüntü işleme ve makine öğrenmesi gibi yapay zeka teknolojilerini kullanarak, tanıma işlemi yapmak, otonom araçlar ya da sürücü destek sistemleri için uyarı mekanizmaları geliştirmektir.

1.1. Tezin Amacı ve Katkıları

Günümüzde insanlar tarafından kullanılan ve ihtiyaç olan yapıların tümü akıllı sistemler ile donatılmaktadır. Akıllı kent kavramı, gündelik hayatın bir parçası olan ve insan yaşamını kolaylaştırmayı amaçlayan birçok bileşene sahiptir. Akıllı sistemlerin en önemli ve ciddi bir bileşeni, trafikteki araçların altyapı ile haberleşme ve tanıma süreçlerini içeren akıllı ulaşım sistemleridir. Akıllı ulaşım sistemlerinin içinde bulunan otonom araçlar çeşitli seviyelere bölünmüş ve en üst seviye 5 ile tamamen sürücüdün bağımsız hareket eden araçlar olarak kategorilenmiş sistemler olarak değerlendirilir. Otonom sistemler ve sürücü destek sistemleri için karayolunun yapısını tanıma önemli bir parametredir. Özellikle yol kalitesinin belirlenmesi [10, 11], karayolundaki kavşakların [12-17] ve yaya geçitlerinin [18-23] algılanması gibi ileri sürüş destek sistemleri, otonom sistemler için başlıca araştırma alanlarıdır. Bu tez çalışmasının amacı, Akıllı ulaşım sistemleri bünyesinde bulunan otonom araç ve ileri sürüş destek sistemlerinin karayolunda hareket halindeyken yol durumunun önceden tanınması ve tespiti yapılarak çıkarımlar elde edilmesidir.

Tez çalışmasında önerilecek yöntem ile bu alanda yapılan çalışmalarda karşılaşılan birçok soruna çözüm üretilmiştir. Bu tezin literatüre katkıları genel olarak aşağıdaki gibi özetlenebilir.

- Geliştirilecek sistemlerin duruma göre karar alması için karayolu tiplerinin tanıma işlemi gerçek veriler ile yapılmıştır.
- Otonom sistemler için gerçek karayolu görüntüleri kullanılarak yol bilgisi elde edilmiştir.

- Sürücü destek sistemleri geliştirmek için karayolu görüntülerinden çıkarımlar elde edilerek makine öğrenme sistemi geliştirilmiştir.
- Gerçek karayolu görüntüleri oldukça karmaşık ve tutarsız yapılar içerdiğinden bunların tanınması ve anlamlandırılması için algoritmalar geliştirilmiş ve odaklanan bölgelerin tanınması sağlanmıştır.
- Yol kalitesinin tespiti yapılarak yol tipine göre araçların hareket davranışlarının belirlenmesi sağlanmıştır.
- Günümüzde oldukça popüler olan Evrişimsel sinir ağı modelleri geliştirilerek tanıma işlemi yapılmış, sonuçlar literatürdeki diğer modeller ile karşılaştırılmış ve model özgün bir çalışma olarak literatüre sunulmuştur.
- Geliştirilen tanıma yöntemi ile hem yol görüntüleri hem trafik işaretleri belirlenmiş ve bunların doğrulanması yapılarak daha güvenli sürüş tiplerinin geliştirilmesi önerilmiştir.
- Önerilen yöntemler, gerçek karayolu görüntülerinden oluşturulmuş ve akademik çalışmalarda kullanılmak üzere literatüre kazandırılmıştır.

1.2. Tezin Yapısı

Bu tez çalışmasının diğer kısımları aşağıda sıralanmıştır:

- İkinci bölümde Akıllı şehir uygulamaları, AUS, Otonom sistemler, sürücü destek sistemleri hakkında genel bilgiler verilmiş ve bu alanda yapılmış bilimsel çalışmalara değinilmiştir.
- Üçüncü bölümde, çalışma alanı ile ilgili görüntü işleme teknikleri ve yapay zeka öğrenme modelleri ile ilgili bilimsel çalışmalara değinilmiş, günümüzde oldukça popüler olan derin öğrenme yöntemleri ve makine öğrenmesi hakkında bilgiler sunulmuştur. Sınıflandırma ve tanıma süreçlerinde kullanılan algoritma ve yöntemler hakkında detaylı bilgiler verilmiştir.
- Dördüncü bölümde, trafikte görüntü işleme için geliştirilen yöntem anlatılmıştır. Yapılan tüm çalışmaların temel yöntemi ve parametreleri hakkında bilgi verilmiştir.
- Beşinci bölümde, karayolunda bulunan önemli bölgelerin tespitinde otonom sistemler ya da sürücü destek sistemlerinde kullanılabilecek uygulamalar gerçekleştirilmiş ve tüm uygulamaların başarımları değerlendirilmiştir.

2. AKILLI ŞEHİRLERDE AKILLI ULAŞIM SİSTEMLERİ

Son 50 yıldır hızla gelişen dünya nüfusu 7.7 milyarı geçmiş durumdadır ve 2050’de bu sayının 10 milyara ulaşması beklenmektedir [24]. Yaklaşık 10 yıl içinde 2030 yılına kadar dünya nüfusunun %60’ının şehirlerde yaşayacağı öngörülmektedir [25]. Toplam nüfusta payını her geçen gün artıran kentsel nüfus, beraberinde daha yoğun trafik ve çözülmesi gereken ulaşım sorunlarını getirmektedir. Enerji tüketimini ve çevre kirliliğini azaltarak insanlara daha etkin, kullanılabilir ve aktif bir yaşam alanı sunan akıllı şehirler, gün geçtikçe insan hayatında daha çok önem kazanmaktadır. Akıllı şehirler, pek çok ülkede başlamış bir dönüşüm ve geliştirme sürecindedir. Akıllı binalar, akıllı ofis yönetimleri, akıllı sağlık kuruluşları, akıllı resmi kurumlar gibi birçok kamu alanının birleşimi ile akıllı şehirleşme oluşmaktadır. Akıllı şehirleşmenin en büyük dönüşümü, gelişimi ve sürdürülebilirliği aslında ulaşım sektöründe olmuştur. Bu dönüşümün de en önemli ayağı akıllı ulaşım sistemleridir.

2.1. Akıllı Ulaşım Sistemleri

Günümüzde şehirlere yapılan göçlerin hız kazanması, motorlu araç sayısındaki hızlı artış ve ortaya çıkan trafik yoğunluğu, trafik kazaları oranının artması gibi birçok neden; araç içi güvenlik sistemleri, trafik yönetimi ve denetimine yönelik akıllı ulaşım uygulamalarına büyük bir gereksinim olduğunu göstermektedir. AUS insanın üzerindeki düşünme ve karar verme yükünü hafifletmeye yönelik ulaşım çözümleri olarak da tanımlanabilir [26]. AUS sistemleri 1928 yılında ilk defa elektrikli trafik ışıkları sistemlerini geliştirmek için kullanılmaya başlanmış ve daha sonra 1960’lı yılların başlarında bilgisayar kontrollü trafik ışıkları yapılmıştır [27]. Günümüze kadar süre gelen bir teknoloji olarak ifade edilebilir. Çünkü AUS alanında her geçen gün yeni bir teknoloji veya yöntem üretilmekte, aynı zamanda bu alan ticari ve sanayi alanında da karşılık bulmaktadır. Araç teknolojisinin gelişimi ile AUS sistemlerine verilen önem artmıştır.

AUS, bilgi ve iletişim teknolojilerinin, sensörler ve ileri optimizasyon tekniklerinin ulaşım sektörüne adapte edilmesi sonucu ortaya çıkan ve bu sistemin geliştirilmesiyle trafikte ulaşım sürelerini giderek kısaltan, yolcuların trafikte güvenliğini artıran, aynı zamanda çevreyle uyumlu ve çevre dostu ulaşım sistemleri olarak tanımlanabilir. AUS, bilgisayar ve iletişim teknolojilerinin yardımıyla hızlı ve etkin bir yapıyla veriye-bilgiye erişilmesine imkan tanımakta, ekonomik olarak sağlık alanında çevresel ve toplumsal açıdan sürdürülebilir çözümler üretmektedir. Dünyada birçok bölgede AUS kullanılmaya başlanmıştır ve gelişme aşamasında olduğu için birçok ülke bu alanda Ar-Ge çalışması yürütmektedir [28]. Ülke ekonomilerinde AUS alanında Ar-Ge faaliyetleri için yüksek oranda bütçe ayrılmıştır ve üretilen teknolojilerin katma değeri göz ardı edilemeyecek kadar önemlidir.

Günümüzde mobilite ve sürdürülebilir ulaşım kavramları, yaşamımızın birçok alanında yerini almaktadır. AUS sistemlerin geliştirilmesi ve kullanılması ile hedeflere ulaşmakta, aynı zamanda bu teknolojiler her alanda büyük önem kazanmaktadır [29]. Nitekim ülkemizin ulaşım ile ilgili otoritelerinin hazırladığı “2014-2023 Ulusal AUS Strateji ve Eylem Belgesi”[30] ortaya çıkmıştır ki; AUS teknolojileri gelecek için katma değer yaratabilecek önemli bir sektördür.

Ulaşım sistemlerindeki bütünlük ve çeşitliliğin yönetim ve planlamasının teknoloji ile bütünleşmesi şeklinde ortaya çıkan AUS’un belki de en önemli özelliği, birden fazla disiplinin bir arada olmasını gerektirmesidir. Bu da farklı uzmanlıkların bir arada çalışmasını ve üretken olmasını zorunlu kılmaktadır. Gerek trafik güvenliği gerekse trafik yönetimi konusunda etkin çözümler arayan Türkiye’de, AUS konusunda Ar-Ge ve ürün geliştirebilecek kapasitenin yaratılması önemlidir.

Geliştirilen AUS uygulamaları sayesinde, şehir içi ve şehir dışında özellikle yoğun bölgelerde araçların yolda geçirdiği sürenin kısaldığı sağlanabilmektedir. Kısa yolculuk süresi, hem ticari taşıtlar hem de normal bireysel kullanıcılar için ekonomik tasarruf demektir.

Ülkemizdeki toplu ulaşım sistemlerine yönelik teknolojilerdeki önemli gelişmeler AUS teknolojilerine örnek gösterilebilir. AUS ile ulaşım süreleri giderek kısaltılmakta, yolcuların güvenliği geçmişe kıyasla oldukça iyi bir konuma yükselmekte, çevreyle uyumlu ve çevre dostu ulaşım sistemleri geliştirilebilmektedir. Ayrıca ulaşım alternatiflerinin çoğaltılması sağlanmaktadır. AUS, emisyon salınım oranlarının düşürülmesi, seyahat sürelerinin kısaltılması, merkezi olarak arıza tespiti ve arızaya en kısa sürede müdahalenin sağlanması, sürücü ve yaya trafik güvenliğinin artırılması, yol kapasitelerinin yüksek verimde kullanılması gibi faydalar sağlamaktadır.

Bu alandaki uygulamalarının artışı, elektronik ve bilgi-işlem alanında yeni yatırım ürünlerinin oluşmasını sağlayacaktır. Birçok Batı ülkesi ve Uzak doğu ülkeleri, özellikle Güney Kore, Japonya, Singapur gibi ülkeler AUS ile karayollarının bir kısmını yönetmekte ve bu alanda Ar-Ge çalışmaları yapmaktadır.

AUS, birçok haberleşme teknolojisini kullanmakta ve gelişen teknoloji ile güncellenmektedir. Araçların kendisiyle ve çevresiyle haberleşmesi için küresel konum belirleme sistemi (GPS), kablosuz ağlar (Wi-Fi, WiMAX, WiBro), mobil iletişim araçları (4G,4.5G,5G,LTE), radyo dalgaları ve kızıl ötesi iletişim, yol kenarı kamera tanıma sistemi, sensör-algılama teknolojileri ve yakın mesafe iletişim teknolojileri gibi güncel teknolojiler kullanılmaktadır. [31].

2.2. Akıllı Ulaşım Sistemlerinde Görüntüleme, Algılama ve Tanıma Sistemleri

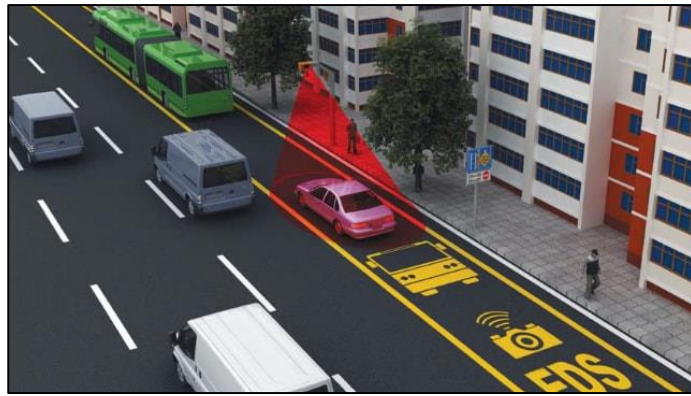
Akıllı ulaşım sistemlerinde araçların çevresiyle etkileşimi için, kamera araçlarıyla görüntü işleme teknikleri ve sensörlerden algılama yöntemleri kullanarak tanıma işlemi yapması gerekmektedir. Tanıma sistemi farklı kaynaklardan gelen verilerin birleştirilmesi ve işlenmesi ile

oluşmaktadır. Bir otonom araç, aracın çevresini gören kamera, etrafına yaydığı ve geri dönüş olarak nesnenin 3B yapısını çizdiği lazer ışınları, yakın engelleri ve nesneleri tespit ettiği sensörler, çevredeki ya da merkezi birimlerden gelen verilerin işlenmesi ile oluşmaktadır. Otonom araçlar veya ileri sürücü destek sistemleri için görüntü işleme aşamasında kamera verileri kullanılmalıdır. Kameradan elde edilen görüntülerin belirlenecek yöntemle göre önışlemeden geçirildikten sonra tanıma işlemi yapılması gerekmektedir. Kameraların gerçek veri işleme için kullanılacak tanıma araçlarına göre çekim yapması gerekmektedir ve anlık olarak modele iletim yapması için çekim formatı ayarlanmalıdır.

AUS'lu yollarda görüntü alımı ve işleme teknolojileri için sensörler ve kablosuz iletişim kullanılmaktadır. Karayollarında bulunan elektronik denetleme sistemleri, yoldaki araçların yoğunluk hesaplama sistemleri, yol kenarı kamera tanıma sistemleri gibi birçok teknoloji kullanılmaktadır. Bu teknolojilerin çoğu görüntü işleme teknikleri ile çalışmaktadır.

2.2.1. Elektronik Denetleme Sistemleri

Elektronik denetleme sistemi, karayollarında bulunan araçların oluşturduğu trafiğin kontrol edilmesi, ulaşım hizmetlerinin denetlenmesi ve yoğunlukla trafik kurallarına uymayan araçların tespit edilerek merkeze iletilmesi amacıyla geliştirilmiştir. Kural ihlali yapan araçların görüntülerinin kamera yardımıyla tespiti sonrası, görüntü işleme yazılımları ile plakaların kaydedilerek ilgili kanunda öngörülen cezaya tabi tutulması için tasarlanan sisteme Elektronik Denetleme Sistemleri (EDS) denir. Trafikte kural ihlali yapan araçların tespiti plaka tanıma sistemi ile yapılır ve hata oranı oldukça düşüktür.



Şekil 2.1. Elektronik Denetleme Sistemi

Bu sistemin, uluslararası standartlarda ve bağımsız akredite kuruluşlar tarafından güvenilirliği belgelendirilmiştir, Türkiye'nin tek yerli trafik ihlal denetim ve tespit ürünüdür. EDS'nin amacı sürücülerin trafik kurallarına uymasını sağlamak, özellikle trafiğin yoğun olduğu

ve hem yaya hem de araçlar için tehlike arz eden bölgelere yerleştirilerek kural ihlali yapan sürücülerini tespit edip cezai işlem uygulamaktır. Sayısal görüntüler üzerinde işlem yaparak daha doğru ve hızlı sonuçlar ortaya çıkarmaktadır. EDS kullanımının bir diğer özelliği ise karayolun ilgili bölgesinde saate göre sürücülerin davranışlarının tespit edilmesini sağlamaktır.

EDS trafikte birçok farklı uygulama alanında kullanılmaktadır. Bunlar;

- Kırmızı ışık ihlal tespit sistemi,
- Park ihlal tespit sistemi,
- Yaya yolu ihlal tespit sistemi,
- Hız tespit sistemi,
- Ortalama hız tespit sistemi,
- Plaka tanıma sistemi,
- Hemzemin geçiş ihlal tespit sistemi,
- Emniyet şeridi ihlal tespit sistemi,
- Ters yön ihlal tespit sistemidir ve yeni teknolojiler ile EDS sistemi kendisini güncellemektedir.

2.2.2. Mobil Elektronik Sistem Entegrasyonu

Mobil Elektronik Sistem Entegrasyonu (MOBESE), anlık iletişim altyapısı kullanan coğrafi bilgi sistemine bağlı olan ve çoğunlukla güvenlik amacıyla kullanılan sistemdir. MOBESE kurulduğu alandaki görüntüleme alt yapısı sayesinde bölgede oluşmakta olan olayların vuku anında kolayca gözlenmesine olanak verdiği gibi, görüntüleri kayıt altına alarak daha sonra etraflıca incelenmesini sağlamaktadır. Sistem bu yapıyla caydırıcılık özelliği kazanarak suç oranının azalmasına sebep olmaktadır.



Şekil 2.2. Mobese kamera sistemi

MOBESE sistemi, kablolu-kablosuz teknoloji kullanarak uzak merkez noktalarına veri gönderimi yapabilir. WiMax teknolojisi kullanarak çok daha uzak merkezi noktalara veri transferi

sağlar. MOBESE, bulunduğu bölgeyi güvenlik çemberi altına alarak, güçlü caydırıcılık özelliği ile koruyan güvenlik sistemidir. Plaka tanıma sistemleriyle uyumlu çalışmaktadır. AUS’nde MOBESE kullanılarak daha güvenli bir sürüş ortamı gerçekleştirilebilir. MOBESE ile araç yoğunluğu değil, yolun güvenlik olayları rahatlıkta takip edilebilir ve herhangi bir anomali durumunda araçlara iletmek üzere uyarı mekanizmaları oluşturulabilir. Özellikle kavşak bölgelerine yerleştirilen MOBESE sistemleri yakın çevrede trafikte bulunan araçlara yolun güvenliği ve diğer konularda bilgi aktarımını sağlayabilir. Bu nedenle MOBESE sistemleri AUS için önemli bir yere sahiptir.

2.2.3. Kavşak Kontrol Sistemleri

Kavşak Kontrol Sistemleri (KKS), karayollarının önemli bir bileşeni olan kavşaklarda araç sayılarına bağlı olarak trafik ışık sürelerinin optimize edilmesini sağlayarak araçlara güvenli ve adil bir şekilde hareket izni vermektedir. KKS, sinyalizasyon sistemine bağlı olarak çalışmakta ve araçlara yolun anlık değerine göre süreler tanımlamaktadır. Sabit ve dinamik KKS bulunmakta ve günümüzde AUS için artık dinamik temelli KKS sistemleri kullanılmaktadır. Kavşakta bekleyen araçlara, sürelerin adil dağıtımını, birbirini takip eden kavşaklarda yeşil ışığın belli hız limitiyle paralel çalışmasını vb. uygulamaları kontrol eder.

2.2.4. Yol Kenarı Kamera Tanıma Sistemi

Yol kenarı kamera tanıma sistemi, yol kenarında bulunan kamera veya etiket tabanlı şemalar, bölge bazlı trafik tıkanıklığı ücretlendirme sistemlerinde ve belirli yollarda ücretlendirme için kullanılabilir. Karayolları üzerine yerleştirilen kameralar, ücretli geçiş özelliği olan köprü veya tünellerde kullanılabilir. Bu tür sistemlerde sürücülerin ücretli bölgelere girdiği ve çıktığı yerlere kameralar yerleştirilerek plaka tanıma sistemi ya da etiket okuma işlemi gerçekleştirilir.

2.2.5. Algılama Sensörleri

Algılama sensörleri, sürücülere güvenli seyir için destek veren sensörlerden yol ve hava durumu sensörlerinden oluşan sistemlerdir. Bu sensörler, araç sensörleri, yol sensörleri, hava durumu sensörleri ve çevre algılama sistemleri olup aşağıda açıklanmıştır.

Araç Sensörleri

Araçlara yerleştirilen sensörler kısa mesafe engel tespiti, şerit algılama, kör nokta uyarısı, park etme, araç takip mesafesi uyarısı gibi fonksiyonlarla güvenli sürüş desteği sağlamak için geliştirilmiştir.

Yol Sensörleri

Kavşakları yönetmek ve sinyal sürelerini ayarlamak, şehir genelindeki ana arterler ve çevre yollardaki trafik akım bilgilerini tespit etmek, trafiği, mevcut alt yapının daha verimli kullanılması amacıyla yönlendirmek için gerekli verileri toplayan sensörlerdir.

2.3. Akıllı Ulaşım Sistemlerinde Kullanılan Teknolojiler

Bir ülkenin gelişmişlik düzeyi, katma değeri yüksek ürün imal etme ve bunu ihraç etmeye bağlı olarak değişmektedir. Otomobil üretimi her geçen gün artmakta ve teknolojisini iyileştirmektedir. Klasik sürücülü araçların yerine artık kısa vadede sürücü destek sistemleri ile donatılmış araçlar ve uzun vadede de yönetimin tamamen araçta olduğu otonom sistemli araçlar üretilmektedir. Tesla, Apple car, Google-Waymo vb dünyanın önde gelen teknoloji firmaları Ar-Ge bütçelerinden otonom araç üretimine ciddi oranlar ayırmış ve bu doğrultuda çalışmalar yapmaktadır. Halihazırda Mercedes, BMW, Ford, Volkswagen gibi dünya çapında ileri gelen otomobil üreticileri otonom araç teknolojileri ya da ileri sürücü destek sistemleri geliştirmektedir. Üretilen otonom araçlar karayollarında test edilmeye devam etmektedir. Bu araçların karayollarında bulunduğu durumlara, yayalara, nesnelere ve sürücülere vermiş olduğu tepkiler değerlendirilmektedir.

Otonom araçlar ya da sürücü destek sistemleri, görüntü işleme, lazer sinyal verilerini işleme, araç içi ve dışındaki sensörlerden elde ettiği verilerin işlenmesiyle tanıma yapmaktadır. Otonom ya da sürücü destek sistemlerinde araç içinde bulunan sensörler ile sürücünün (yorgunluk tespiti, gözün kapalılık tespiti gibi) ve kullanılan aracın durumu (Lastik basıncı, motorun durumu, farların açık-kapalılık durumu gibi) tespit edilebilir. Araç dışında bulunan sensörler, kameralar ve LiDAR verileri ile yol durumunu tespit edebilir.

Farklı karayolu ortamlarında aracın hareketi değişmeli ve sürücünün uyarılması gerekmektedir. Karayolu tipinin tespiti için sensör verilerinden ziyade görüntü işleme ve GPS harita bilgisi gerekmektedir.

2.3.1. İleri Sürücü Destek Sistemleri

İleri sürücü destek sistemleri, standart bir araç takip sisteminde bulunan özelliklerin yanı sıra şerit takip desteği, çarpışma uyarısı, trafik işaretlerinin tanınması ve park desteği gibi işlevleri yerine getirecek ve sürücülerin trafikte daha güvenli bir biçimde araç kullanımlarını sağlayacak tümleşik bir sistemdir.

İleri sürüş destek sistemleri - Advanced Driver Assistance Systems (ADAS) araç içine ve dışına yerleştirilen kameralar, sensörler sayesinde sürücülerin, araçların, yolun ve yayaların bulunduğu durumun tespitinin yapılarak sürücülerin uyarılması, araçların karar almasında yardımcı

olunması için geliştirilmiş bir dizi teknolojidir. Aslında ADAS, 1980 yıllarında başlayan ve sürekli yeni teknoloji ve algoritmaların eklendiği, sistemlerde güncellemelerin yapıldığı bir tanıma ve karar verme sistemidir.

Karayolu izlenmesi için tasarlanmış yapay zeka temelli ilk çalışma, 1989 yılında yapılmış olan Yapay sinir ağı(YSA) tabanlı bir otonom arazi aracı (Autonomous Land Vehicle In a Neural Network -ALVINN) [32] sistemidir. Bu çalışmada, yol izlemesi için tasarlanmış 3 katmanlı bir geri yayımlı YSA modeli kullanılmıştır. ALVINN sisteminde, bir kamera ve bir lazer mesafe bulucusunun elde ettiği görüntüler girdi olarak alınmış ve çıkış olarak aracın hareket etmesi için gereken yön belirlenmeye çalışılmıştır. Sistemin eğitimi, benzetimi yapılmış yol görüntüleri kullanılarak gerçekleştirilmiştir.

ADAS sistemi ile ilgili çalışmalar ilk olarak, emniyet kemerleri gibi basit mekanizmalardan oluşan iyileştirmelerle yapılmıştır, ancak daha sonra kilitleme önleyici sistemler, elektronik stabilizasyon programları ve hava yastıkları gibi daha karmaşık cihazlar geliştirilmiştir. Sonraki süreçlerde yol-şerit tanıma, araç takip mesafesini koruma ve şerit ihlal durumu tespiti gibi birçok çalışma yapılmış ve uygulanabilirliği kanıtlanmıştır. Trafik kazalarını önlemek veya azaltmak amacıyla tahmin etmeyi amaçlayan daha akıllı araç içi sistemler geliştirilmektedir.

ADAS, araç içerisinde bulunan kameralar sayesinde, sürücülerin sürüş anında uyuması ya da ön koltukta çocuk oturması gibi durumlar için uyarı üretebilmekte ve kaza durumu ile ilgili bilgileri otomatik olarak bir merkeze ve bu sistemi kullanan diğer araçlara iletebilmektedir. ADAS ile araçların rutin bakım zamanları sürücüye, araçta oluşan arızalar ise otomatik olarak servise bildirilmektedir. ADAS, tümleşik sistemin eğlence ve bilgilendirme işlevlerini bir araya getiren çoklu ortam uygulamalarını da desteklemektedir. Bu kapsamda Youtube, Dailymotion gibi çevrimiçi hizmetlere erişim sağlanması, trafik ve hava durumu bilgilerinin sunulması, sosyal medya entegrasyonu da sağlanmaktadır.

Sürücü destek sistemlerinin ilk örnekleri arasında 1970 yılında ABD’de ortaya çıkan araç hızını sabitlemeye yarayan seyir kontrol sistemleri yer almıştır. Bu durumu birçok Avrupalı otomotiv firmasının yer aldığı PROMETHEUS Projesi kapsamında geliştirilmeye başlayan adaptif seyir sistemleri takip etmiştir. Bu sistemlerinin araçlarda ilk olarak kullanılmaya başlamasının üzerinden 15 yıl geçmiş olmasına rağmen günümüzde henüz birçok otomotiv üreticisinin üst sınıf araçlarında uygulanmaktadır. Yakın gelecekte ise bu sistemin orta ve alt sınıf araçlarda standart bir özellik olarak bulunacağı ön görülmektedir.

2.3.2. Otonom Araç Sistemleri

Otonom araçlar, otomatik kontrol sistem donanımları kullanılmasıyla sürücü desteğine ihtiyaç duymadan kullandığı yolun, trafik durumunu ve çevre şartlarını algılayarak hareket eden ve

tüm algılamaları otomatik yapan otomobillerdir. Birçok araştırmacı insan davranışlarını temel olarak otonom araç sistemleri geliştirmektedir [33]. Yüksek frekanslarda farklı kablosuz iletişim teknolojilerindeki çalışmalar büyük bir seviyeye ulaştığından, otonom araç geliştirilmesine yönelik çalışmalar da bugün hızlı bir şekilde devam etmektedir. Kablosuz bağlantı teknolojisi, bağlı taşıt teknolojisi alanındaki gelişmeyi desteklemekte ve otonom araçların gelişiminde büyük değişimler sağlamaktadır. Aslında otonom araç fikri 1920'li yıllarda başlamış ve ilk adımlarını Tesla atmıştır. Birçok üniversitede proje konusu olarak geliştirilmiş olan sürücüsüz araç teknolojisi, günümüzde sensörler, radar sistemleri, GPS, araçlar arası kablosuz iletişim, bilgisayar görü sistemleriyle donatılmıştır. Otonom araçlarla ilgili yapılan ilk çalışmalar, otomobillerin sürücüsüz çalışmasını mümkün kılmaya odaklanırken, bugünlerde otonom araç teknolojileri seviye 4 ve seviye 5 üzerinde arttırmaya yönelmiştir. Otonom araçlarda teknolojinin seviye 5'e ulaşması ile sürücüsüz araçların sürücülü araçlarla beraber trafikte hareket edebileceği görülmektedir [34]. Tablo 2.1'de otonom araç teknolojilerinin seviyeleri verilmiştir.

Tablo 2.1.Otonom araç seviyeleri

Seviye	Durum	Özellikler
Seviye 0	Otonom yok	Direksiyon, frenleme, vites değişimi gibi en temel kontroller sürücüye aittir.
Seviye 1	Sürücü Asistanı	Sürücü çok az da olsa yönlendirilir, hızlanma, frenleme ve çevre kontrolü sürücünün denetimindedir.
Seviye 2	Kısmi Otonom	Araç direksiyon idaresinde ve hızlanma konusunda yönlendirme yapar. Araç idaresi sürücüdür.
Seviye 3	Koşullu Otonom	Aracın çevre kontrolünü sağlayabilir, frenleme gibi emniyet-kritik işlerde sürücünün serbest kalmasını sağlamıştır. 60 km/h hıza kadar aracı otomatik kontrolünü yapabilir.
Seviye 4	Yüksek Otonom	Aracın tüm direksiyon kontrolü, frenleme, hızlanma, nesne ve yol takibi gibi görevleri yerine getirebilir; karar gerektiren şerit değiştirme, dönme, sinyal verme gibi olaylara cevap üretebilir. Bu seviyede dinamik sürüşte karar verici değildir.
Seviye 5	Tam Otonom	Sürücü müdahalesi bulunmayan sistemlerdir. Direksiyon idaresi, gaz ve fren pedallarıyla hızlanma gibi tüm hareketler, kritik öneme sahip tüm durumlar, çevre kontrolü ve trafik sıkışıklığı gibi özel durumlar da dahil sürücü kontrolü bulunmamaktadır.

Sürücüsüz otomobiller çoğunlukla günlük yaşamda kullanılmayan ve yeni geliştirilen bir teknoloji olduğundan, otonom araçlarda daha düşük kullanıcı kabulü olması, araştırma yapmayı zorlaştırmaktadır. Halka açık bir ankette Kyriakidis ve ark., ankete katılanların %69'unun, otonom araçların 2050'lere kadar pazar payının %50'sine sahip olacağını ve katılımcıların %33'ünün de otonom araç kullanmayı tercih edeceğini düşündüğünü tespit etmiştir [35]. Başka bir araştırmada, Schoettle ve Sivak tarafından bir anket hazırlanmış ve ABD'deki 505 lisanslı sürücüye tercih ettikleri otomasyon seviyesi ve kendi kendine süren araçlarla ilgili endişeleri sorulmuştur. En çok tercih edilen seviyenin %43,8 oranında kendi kendini süren bir araç olmadığı belirtilmiştir. %40,6'sı kısmen otonom bir aracı tercih ederken, katılımcıların %15,6'sı tamamen kendi kendini süren

seviye 5 otonom bir aracı tercih etmiştir. Ayrıca, ankete katılanların %35,6'sının seviye 5 otonom araç kullanımı konusunda çok endişeli olduğu, ankete katılanların %14,1'inin kısmen otonom araç kullanımı konusunda endişeli olduğu belirtiliyor [36]. Seviye 4 ve Seviye 5 performanslarına ulaşmak için hala zamana ihtiyaç duyulmaktadır. Ayrıca araştırmacılar, otonom araçların 2025-2040'a kadar yollara sunulacağını düşünmediklerini belirtmişlerdir. Tahminlere göre, Seviye 4 veya 5 otonom araçlar 2020'lerde sınırlı kullanımda olacak ve bu sistemler ilerleyen zamanlarda daha çok kullanılacaklardır [33].



Şekil 2.3. Otonom araçlar

Otonom araçlar, çevreden gelen bilgileri (araçlar, yaya, yol) ve aracın kendi durumu ile ilgili bilgileri değerlendirmektedir. Aracın tekerleklerinde bulunan ultrasonik sensörler frenleyen - hız artıran veya park etmiş durumda olan araçların konumlarını tespit edebilir. Araçlar arasında iletişim kurarak yol durumunu öğrenir ve farklı güzergahlar seçebilir. Bunlar gibi çok çeşitli sensörlerden gelen verilerin merkezi bir bilgisayar sistemiyle analiz edilerek direksiyon kontrolü, frenleme, hızlanma gibi olaylar gerçekleştirilebilir.

Otonom araç teknolojileri ile birlikte literatüre giren Nesnelerin İnterneti (Internet of Things - IoT) fiziksel olarak kullanılan tüm nesnelerin birbirleri veya karmaşık büyük kompleks sistemlerle internet aracılığıyla belirli bir protokol içerisinde bağlanabildiği iletişim ağına denir.

IoT prensibinde; izleme cihazları, sensörler, biochip ve erişim düzenekleri nesne olarak kabul edilmektedir. Bunun yanı sıra nesne olarak kabul edilen bir birimin akıllı olduğu varsayılmaktadır. Bu kabulün yapılabilmesi için de birimin bir ID'ye sahip olması, bağlanabilir olması ve kendine ait bir sensörü olması gerekmektedir. Bu şartlar sağlandığında, nesne kontrol edilebilir ve erişilebilir hale gelmektedir. Otonom araçlarda bulunan fiziksel donanımları IoT teknolojisi ile iletişim kurulabilir.

Otonom araçlarda iletişim, insan katkısı olmadan çevrelerindeki tüm trafik koşullarıyla iletişim kurmayı sağlayan teknolojilerdir. Otonom araçlar çevredeki nesne ve araçlar ile herhangi

bir kaza yapmamak için daima diğer araçlara ve nesnelere bağlı kalmalı, aynı zamanda akışı korumak için gerekli reaksiyonları vermelidir. Araçlar arasında kurulan tüm iletişim türüne Araçtan Araca - Vehicle to Vehicle (V2V) denir. Otonom araçlar aynı zamanda hareket ettikleri ağın ortamına bağlantı sağlayabilir ve buna Araçtan Altyapıya - Vehicle to Infrastructure (V2I) adı verilir. Bu iki gelişmiş ve halen geliştirilmekte olan otonom taşıt teknolojileri, şu anda kullanılan normal araçlara göre farklılık göstermektedir. Üçüncü bağlantı olarak, Araçtan Her şeye - Vehicle to Everything - V2X) otonom araçların, herhangi bir çarpışmayı önlemek için, yayalar ve hareketli diğer nesneler (bisikletliler) gibi araç dışı nesnelere de bağlanmasını gerektirir. Araçlar arası iletişim sadece yakın araçlarda değil uzak mesafe araçlarda da bulut bağlantısı kurularak yapılır. Bir merkezdeki internet üzerinden trafik hakkında bilgi almak ve bu bilgileri otonom araçlara paylaşmak iletişimin bir diğer halidir [37].



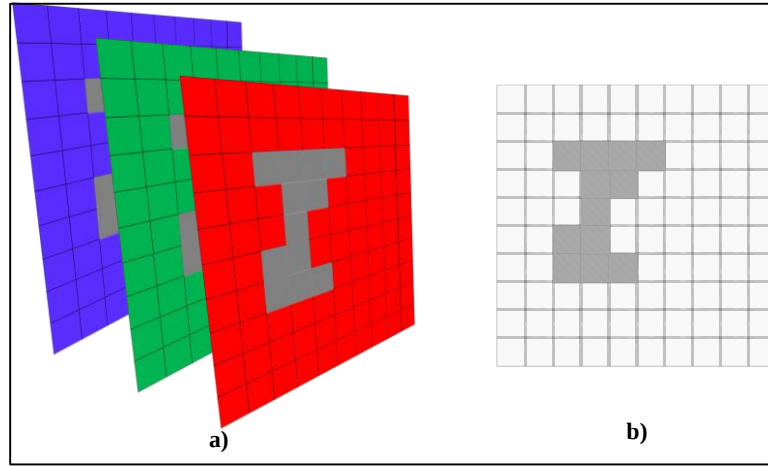
3. GÖRÜNTÜ İŞLEME TEKNİKLERİ VE YAPAY ZEKA ÖĞRENME MODELLERİ

Tez çalışmasında karayolu görüntülerinden yol durumunun tespiti için görüntü işleme ve yapay zeka teknikleri kullanılmıştır. Bu bölümde kullanılan teknikler verilmiştir.

3.1. Görüntü İşleme Yöntemleri

Görüntü işleme teknikleri, görüntüyü sayısal forma dönüştürmek ve görüntüde istenen parametrelerin belirlenmesi için sayısal formun farklı algoritmalar ile değişimini sağlayarak çıkarım yapmaktadır. Dönüştürülmüş verilerden spesifik görüntü elde etmek veya istenen bilgiler ortaya çıkarmak için kullanılan bir yöntemdir. Bu yöntemin giriş katmanı, video veya fotoğraf gibi sabit bir görüntüdür. Çıkış katmanı ise; görüntünün istenilen ya da dikkat edilmesi gereken bölümüdür.

Sayısal görüntülerin değerleri piksel sayısına göre matris olarak gösterilir. Kanal sayısına göre matris katmanı değişmektedir. Örneğin Red-Green-Blue (RGB) şeklindeki 3 kanallı görüntüler 3 katmanlı matris değerine sahiptir ve kırmızı, yeşil ve mavi için ayrı ayrı sayısal değerler barındıran görüntü boyutunda matrisler olarak nitelenir. Şekil 3.1’de RGB kanallı ve gri formatlı görüntüler matris olarak görülmektedir.



Şekil 3.1. RGB kanallı ve gri formatlı görüntülerin matris olarak gösterimi

Matrisin her bir hücresi bir piksele eşittir. RGB kanallı görüntülerin üç kanalda aynı piksellerde yansması varken, gri tek kanallı görüntü ise 256 bit derinliğinde tek katmanlı olmaktadır. Görüntünün çözünürlüğünü etkileyen temel faktör, piksel sayısındaki değişimdir.

3.1.1. Görüntüleri Yeniden Boyutlama

Farklı ortamlardan elde edilen görüntülerin, görüntü işleme ve tanıma sistemleriyle değerlendirilmesi için tek boyuta indirgenmiştir. Farklı veritabanlarında elde edilen görüntülerin kanal sayıları aynıken, boyutları farklıdır. Görüntülerde bozulma olmadan ve önemli bölgelerin kırılması yapılmadan görüntüler eşit boyutlara düşürülmelidir. Ayrıca, görüntülerin boyutunun düşürülmesi hem işlem süresini azaltmaya yardımcı olur hem de görüntülerin işgal ettiği alanın azalmasına sebep olur ve bu nedenle sistem performansını artırır. Bu tezde, kullanılan veri tabanları büyük ve çeşitli boyutlarda görüntüler içerdiği için, boyutlarının küçültülmesi yoluna gidilmiştir. Bunun için en küçük boyuta sahip ölçülere göre tüm görüntüler tekrar boyutlandırılmıştır. Aynı zamanda karayollarında bulunan üst yok olma noktaları tespit edilmiş ve noktanın üst kısmında kalan bölge kırılarak atılmıştır. Bu işlem sonunda kalan görüntü üzerinde tanıma işlemi yapılmıştır.

3.1.2. Görüntüde Alan Bulma

Verilen bir sabit görüntü içinde istenen bölgeyi bulma, günümüzde oldukça popüler bir konu olarak değerlendirilir ve görüntüde istenen bir bölgenin tespit edilmesi olarak ifade edilir. Şekil 3.2’de görüntü tanıma-eşleme örneği görülmektedir.



Şekil 3.2. Görüntüde tanıma yöntemi ile parça-alan bulma-tanımlama

Görüntülerde bulunan grup piksel değerleri incelenerek, aranan alandaki görüntüler eşleştirilir ya da bölge bazlı incelenerek, yakın değerlere sahip bölgeler işaretlenerek bulma işlemi gerçekleştirilir. Görüntüden bulunan nokta detaylar, kenarlar ya da alanlar tespit edilerek istenen bölgeye odaklanılır.

3.1.3. Görüntüde Kenar Tespit Algoritmaları

Görüntülerde kenar bulmak için birçok algoritma geliştirilmiştir. Bir RGB kanallı görüntüde kenar tespiti piksellerin parlaklık bileşeni kullanılarak elde edilmektedir. Gri seviyeli görüntülerde parlaklık değerleri kenar bölgelerde değişmektedir. Bu bölgelerde ani geçiş pikselleri üzerinden kenar tespiti yapılabildiği gibi birçok farklı yöntem de mevcuttur. Gradyan kullanarak görüntü içindeki değişimler tespit edilebilir. Gradyan temelli algoritmalar kullanılarak gri formattaki görüntülerden kenarlar elde edilebilir. Diğer bir alternatif ise, her bir renk bileşeni için bulunan kenarların her bir piksel için doğrusal olarak birleştirilmesidir [38]. Kenar tespiti için Canny, Sobel, Prewitt ve Robert algoritmaları geliştirilmiştir.

Canny Kenar Tespit Algoritması

Görüntüde nesnelerde bulunan keskin parlak kenarları bulmak için John F. Canny tarafından geliştirilmiştir, son derece etkin olarak kullanılan bir algoritmadır. Görüntünün gürültülerini azaltmak için Gaussian çekirdekle konvolüsyon kullanılır.

Sobel Kenar Tespit Algoritması

Gradyan operatörü (Bir skaler alanın yön türevi artımının en çok olduğu yere doğru yönelmiş ve büyüklüğü değişimin en büyük değerine eşittir) uygulanarak görüntünün gradyan büyüklüğü ve yönü hesaplanır [39].

Prewitt ve Robert kenar bulma metotları

İkili eşikleme yöntemi uygulanarak, görüntülerin istenmeyen bölgelerden ayırma işlemi gerçekleştirilir. Güçlü-zayıf ayrımı yapıldıktan sonra asıl kenarlarla görüntüye son hali verilir [40].

3.1.4. Görüntülerde İlgili Bölge için Kırpma

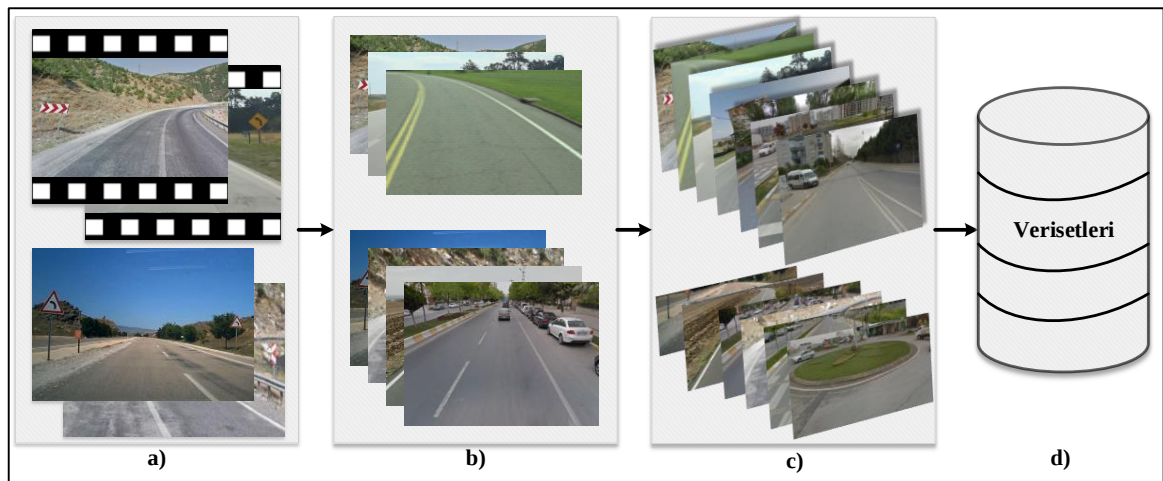
Bir veri kümesinde bulunan görüntülerin görsel kalitesinin artırılması ve ilgili bölgeye odaklanmak için kırpma yöntemi kullanılır. Tespit yapılabilmesi için bir görüntüde istenen bölgenin - Region of Interest (RoI), belirli bir amaç için tanımlanan bir veri kümesi içinden alınmasıdır [41, 42]. RoI kavramı, birçok alanda kullanıldığı gibi yol sınırlarının belirlenmesi alanında yaygın olarak kullanılmaktadır. Bu tez çalışmasında da görüntü işleme aşamasında RoI tespiti yapılmıştır.

Kenar tabanlı yöntemler düzgün yol türlerinde başarı gösterirken, özel bölge ve bozuk veya asfaltlı olmayan farklı yol tiplerinde aynı başarıyı gösterememektedir. Yol sınırlarının veya yol işaretlerinin yerleştirilmesi en yaygın kullanılan yaklaşımlardan biridir. Renk işareti [43, 44], Hough dönüşümü [45], yönlendirilebilir filtreler [46], Spline Modeli [47] vb. yol sınırlarını veya

işaretlerini bulmak için kullanılmıştır. Ancak bu yöntemler yol sınırları belli olan, trafik işaretleri düzgün olan ve düzgün asfalt döşeli yollarda başarı göstermektedir. Bu yöntemler bakımsız, yol işaretlerinin ve levhalarının eksikliği ve asfaltsız olan bozuk yollarda yanlış tespitler yapmaktadır. Bu durumun üstesinden gelmek için, yol tabanlı tespit yapabilmek amacıyla literatürde yaygın bulunan yöntemler (histogram analizi, renk işaretleri, filtreler) ve doku-bazlı yöntemler önerilmiştir. Öncül tabanlı yöntemlerin çoğu, değişen görüntüleme koşullarına, yol tiplerine ve etrafı çevreleyen ortamlara karşı dayanıklıdır. Bu nedenle, bu tür klasik temelli yöntemler, yol bölgesini tek bir yol görüntüsünden tespit etmek için uygulanamaz. Buna karşılık, doku bazlı yöntemler hem düzgün yollar hem de düzgün olmayan yolların istenen bölgelerinin tespiti için çok etkilidir. Bu yöntemler önce yerel yönelimli dokular arar ve sonra görüntülerde bulunan yol ufuk noktalarının bulunduğu yeri tespit eder. Genel olarak, bu doku bazlı yöntemlerin dezavantajları, ufuk noktalarının tespit işleminin hesaplama maliyetinin yüksek olması ve bazı karmaşık karayolu görüntülerinde yol alanı çıkarımının performansını etkileyebilecek noktalarının tespitinde yanlış tahmin yapılmasıdır.

3.2. Görüntülerin Veri Setlerine Dönüştürülmesi

Günümüzde geliştirilen otonom sistem teknolojileri ve sürücü destek sistemleri çoğunlukla bilgisayar görü sistemi kullanılarak yapılmaktadır. Görü temelli sistemlerde, makine algılaması ve öğrenmesi üzerinde çalışmalar mevcuttur. Makinenin farklı yöntemlerle öğrenmesi ve görüntüleme teknolojileri kullanarak karar vermesi sağlanmaktadır. Görüntü temelli çalışma prensibi Şekil 3.3'te görüldüğü gibidir.



Şekil 3.3. Görüntü işleme aşamaları

Şekil 3.3 incelendiğinde, 4 aşamadan oluştuğu görülmektedir. Bu çalışmada görüntü işleme sürecinde a) aşamasında video ve kamera görüntüleri toplanır, b) aşamasında tüm görüntüler RGB formatıyla sayısal değer olarak saklanır. c) aşamasında görüntüler ön işlemlerden geçirilir. d) aşamasında veri setlerine dönüştürülür.

Görüntü işleme tekniği olarak birçok yöntem bulunmaktadır. Bu tez çalışmasında kullanılan görüntülerin RoI alanlarının belirlenmesi için üst yok olma noktaları tespit edilmiş ve bu tespit ölçüsüne göre görüntülerin alt kısımları kırpılmıştır.

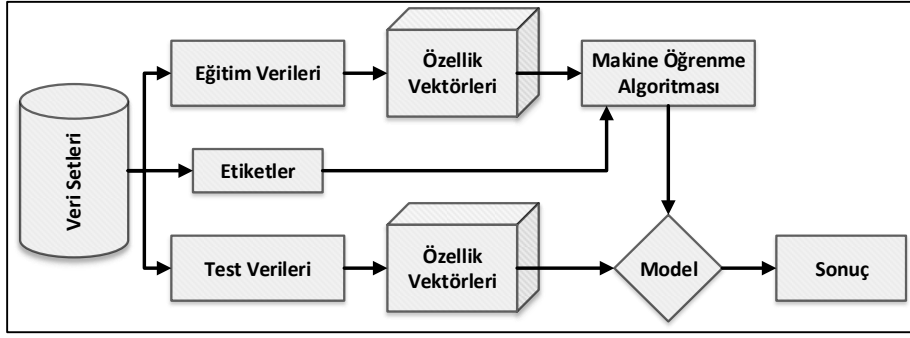
3.3. Yapay Zeka ve Makine Öğrenmesi Yöntemleri

Yapay zeka kavramını ilk olarak 20. yüzyılın ortalarında dönemin en başarılı bilim adamlarından olan Alan Mathison Turing ve John McCarthy ortaya atmıştır. Yapay zeka ile ilgili birçok çalışma yapılmış ve günümüzde oldukça popüler bir alan olmuştur. Yapay zeka adı altında farklı teknolojiler geliştirilmiş, insan gibi düşünen makinalar ve robotlar tasarlanmıştır. Yapay Zeka teknolojileri bilgisayar, bir robot ya da bir tanıma sisteminin giriş parametrelerine göre karar verme sürecini zeki canlılara benzer şekilde düşünen ve karar alma süreçleri olarak tanımlanabilir [48].

Derin öğrenmeye ihtiyaç duymamızın sebeplerinin başında yapay sinir ağlarının tek başına bazı problemleri çözememesi, yüksek başarılı sonuçlar elde edememesi veya geç cevap vermesi gelmektedir. İhtiyaca göre işlenecek verilerin büyümesine paralel olarak gelişen teknoloji sayesinde, grafik tabanlı işlemcilerin gücünün artması ile istenen veriye veya sonuca daha hızlı ulaşma imkânı sağlanmıştır.

3.3.1. Denetimli Öğrenme

Denetimli öğrenme, incelenen verinin barındığı sistemin önceden etiketlenen veriler kullanılarak eğitilmesi ile öğrenmenin sağlanmasıdır. Sistem eğitilirken veri setinde bulunan her bir örneğe ait giriş ve çıkışlar verilir. Metin Sınıflandırma çalışmalarında giriş, metnin içeriğini; çıkış ise kategorisini temsil eder. Test veri seti ise, sistemin doğrulanması amacıyla kullanılır. Sistemin doğrulanması aşamasında, öğrenme algoritması kategorisi bilinmeyen bir test verisine, eğitim verisinde bulunan çıkışlardan herhangi birini atar [49]. Denetimli öğrenme modeli süreci Şekil 3.4'te verildiği gibi gerçekleşmektedir [50].

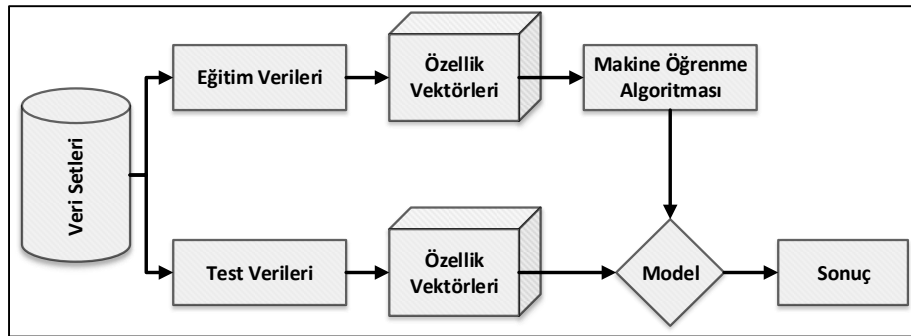


Şekil 3.4. Denetimli öğrenme modelinin çalışma prensibi

Denetimli öğrenme, Şekil 3.4’te görüldüğü gibi, ilk aşamada sistemin eğitimi için verinin bir kısmı eğitilerek özellik çıkarımları elde edilir ve daha sonra bu verilerin etiketleri ile eşleştirilir. İkinci aşamada ise test verilerinin özellik çıkarımları elde edildikten sonra eğitilmiş model ile karşılaştırılır ve verinin türü belirlenerek öğrenme gerçekleşmiş olur. Eğitim aşamasında etiketli veriler kullanıldığı için model, ham bir verinin sınıfını tahmin edebilmektedir. Destek Vektör Makineleri (DVM), YSA, Bayes, k-En yakın Komşu (k-NN), Lojistik Regresyon ve Karar Ağaçları gibi algoritmalar, denetimli öğrenme modeline örnek verilebilir. Bu modeller yaygın olarak kullanılmaktadır [51]. Günümüzde çok sık kullanılan sınıflandırma ve regresyon analizi, denetimli öğrenme yöntemleri ile yapılmaktadır.

3.3.2. Denetimsiz Öğrenme

Denetimsiz öğrenme, incelenen verilerin özniteliklerinin elde edildikten sonra etiketsiz veri kullanarak sistemin eğitilmesidir. Denetimsiz öğrenmede, veri setinde bulunan örneklerin sonuç-sınıf değerleri bilinmediği için tanıma veya sınıflandırma olarak nitelenemez. Verilerin öznitelikleri arasındaki ilişkiler tespit-öğrenme yapılarak veriler kümelerine ayrılabilir. Genellikle kümeleme, olasılık yoğunluk tahmini gibi makine öğrenmesi algoritmaları kullanılarak öznitelikler arasındaki ilişkiler bulunur. Şekil 3.5’te denetimsiz öğrenme çalışma prensibi görülmektedir.

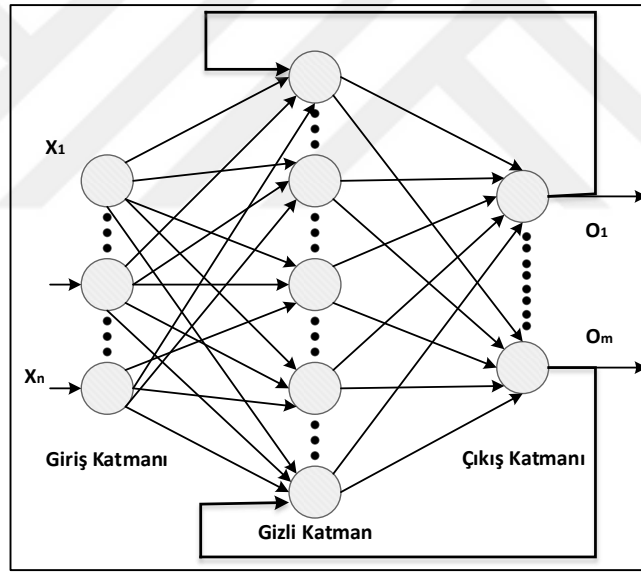


Şekil 3.5. Denetimsiz öğrenme modelinin çalışma prensibi

Denetimsiz öğrenme fonksiyonu kullanılarak toplanan parametreler denetimli öğrenme için de kullanılabilir [52]. Hiyerarşik kümeleme algoritmaları denetimsiz öğrenme modeli kullanılarak oluşturulur.

3.3.3. Takviyeli Öğrenme

Takviyeli öğrenme, denetimli ve denetimsiz öğrenmelerden farklı olarak, problemler veya örnekler bilinirken, doğru cevap bilinmemesi durumudur. Özellikle yapay zekalı bilgisayar oyunları bu öğrenme tipi ile hazırlanmaktadır. Bir oyunda yapılan hamleye karşılık, yeni bir hamle yapıldığı düşünüldüğünde bir süre sonra, karşılıklı oynanan oyunda iki taraftan birinin kazanacağından, burada elde edilen sonuç bir sonraki oyun için takviyeli öğrenmenin sinir sinyali olur. Takviyeli öğrenme ile kurulan yapay sinir ağlarının zamana bağlı değişen problemlerin çözümünde etkili olduğu görülmektedir. Takviyeli öğrenme modelinin şematik gösterimine Şekil 3.6'da yer verilmiştir.



Şekil 3.6. Takviyeli öğrenme modelinin çalışma prensibi

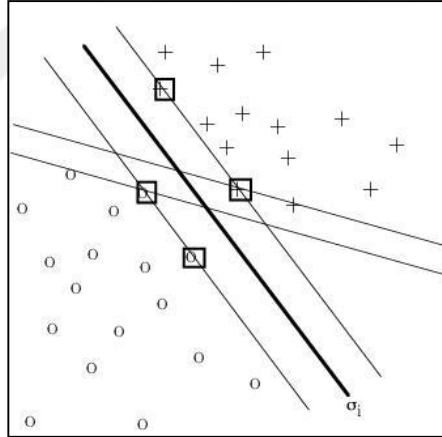
Eğer geri bildirim bilgisi kesin olarak biliniyorsa, YSA danışmalı öğrenme ile eğitilmesi, takviyeli öğrenmeye göre daha kısa ve daha az masraflı olacaktır.

3.4. Klasik Makine Öğrenmesi Yöntemleri

Günümüzde klasik makine öğrenmesi için birçok yöntem geliştirilmiştir. Literatürde en sık kullanılan algoritmalar DVM, k-NN ve Çok Katmanlı Algılayıcı (ÇKA) modelleridir.

3.4.1. Destek Vektör Makinesi

Temelleri 1963 yılına dayanan ve ilk olarak 1995 yılında Vladimir Vapnik, Bernhard Boser ve Isabelle Guyon tarafından geliştirilen Destek Vektör Makineleri, değerler üzerinden istatistiksel öğrenme teorisi yapan bir gözetimli öğrenme algoritmasıdır. DVM, bir veri grubunun karar sınırlarını belirlemeye yarayan karar düzlemleri ile oluşur. Değerlerin bulunduğu karar düzlemi farklı sınıf özelliklerine sahip değerleri birbirinden ayırmaktadır. Şekil 3.7’de gösterildiği gibi, nesneler daire ve artı sınıfından sadece birine aittir. Ortadaki sınır çizgisinin sağına ve soluna düşen yeni nesneler farklı değerler olarak tanımlanacaktır. DVM hem doğrusal hem de doğrusal olmayan sınıflandırıcı olarak kullanılabilen en güçlü denetimli öğrenme tekniğidir. DVM, maksimum marjı olan bir hiperdüzlemi bulmaya çalışır. Kenar boşluğu, hiperdüzlem ve en yakın etiketlenmiş vektör olan yarı alanlar arasındaki mesafedir. Doğrusal problemler için gerçekleştirilen bu yaklaşım doğrusal olmayan problemler için kernel dönüşümleri kullanılarak yapılabilir [53]. DVM kısaca elemanlar arasında bulunan örüntülerin bilinmediği veri setlerindeki sınıflandırma problemleri için önerilmiş bir makine öğrenmesi yöntemidir.



Şekil 3.7. Destek vektör makinalarının çalışma prensibi

DVM sınıflandırıcısının en sade hali ile amacı, iki veri sınıfını ayırabilmek için bir hiperdüzlemi veya karar sınırını bulmaktır. Şekil 3.7’de hiperdüzlem, ara bölgenin tam ortasına yerleştirilmiştir. Bunun yapılmasındaki amaç, ayırma sınırını maksimum yapmaktır. DVM’ de amaç, pozitif ve negatif olarak adlandırılan iki sınıfın verileri arasındaki optimum ayırıcı hiper düzlemi bulmaktır. İki sınıfın doğrusal ayrımını yapmak için Denklem 3.1 kullanılır [54].

$$f(x) = \omega^T x + b = 0 \quad (3.1)$$

Elemanların boyutunda ağırlık vektörünü temsilen ω parametresi, x eğitim verisidir ve ayırım yapan hiper düzlemin konumunu belirlemede b (bias) kullanılır[55]. DVM’ler de verilerin

doğrusal olarak ayrılacakları bir yapıda olup/olmamasına bağlı olarak karşılaşılabilecek iki durum söz konusudur.

$$J(W) = \|W\|^2/2 \quad (3.2)$$

Burada, $\|W\|$ ağırlık vektörünün uzunluğu ve y_i , alabilecek sınıfların etiketidir ve bu -1 veya 1 değer aralığındadır.

$$W^T x_i + b \geq 1, y_i = 1 \quad (3.3)$$

$$W^T x_i + b \geq -1, y_i = -1; i = 1, 2, \dots, n \quad (3.4)$$

$$y_i(W^T x_i + b) \geq 1, i = 1, 2, \dots, l \quad (3.5)$$

l eğitim örneği sayısıdır. Boşluk değişkenleri ξ_i ' den oluşan yanlış sınıflandırmayı sıfır olmayan gevşek değişken vektör ξ 'yi gidermek için Denklem aşağıdaki gibi hesaplanır.

$$y_i(W^T x_i + b) \geq 1 - \xi_i; \xi_i \geq 0, i = 1, 2, \dots, l \quad (3.6)$$

Maliyet fonksiyonu ξ_i 'ye göre değişmektedir.

$$J(W, \xi) = \frac{\|W\|^2}{2} + C \sum_i \xi_i \quad (3.7)$$

Burada C , kullanıcı tarafından atanan pozitif düzeltme parametresidir. DVM sınıflandırıcıda ayırıcı hiper düzlem ve en yakın eğitim örnekleri arasındaki mesafeyi en yüksek değerine çıkaran en uygun hiper düzlem Denklem 3.8 ile bulunur [56].

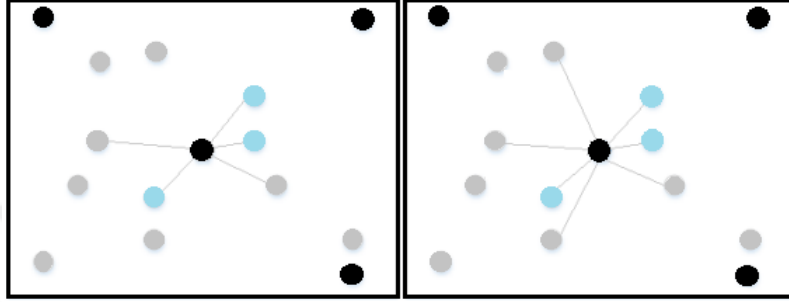
$$\begin{cases} \min \frac{\|\omega\|^2}{2} \\ y_i(\omega^T x_i + b) \geq 1 \end{cases} i = 1, 2, 3, \dots, M \quad (3.8)$$

DVM sınıflandırıcıda ayırıcı hiper düzlem ve en yakın eğitim örnekleri arasındaki mesafeyi en yüksek değerine çıkaran en uygun hiper düzlem Denklem 3.7 ile bulunur. Denklem 3.8'deki optimizasyon probleminin çözülmesiyle en uygun ayırıcı düzlem elde edilir [56].

3.4.2. K-En Yakın Komşu Algoritması

K-en yakın komşu algoritması, (k Nearest Neighbor - k-NN) sınıflandırmada en yaygın kullanılan eğitimli öğrenme algoritmalarından biridir. Algoritma yeni bir örnek geldiğinde daha önceden belirlemiş olduğu öğrenme çıktıları üzerinden sınıflandırma yapmaya çalışmaktadır. Algoritma, yeni gelen veriyi sınıflandırma işlemi sırasında bu veriye en yakın olan komşusuna göre

karar vermektedir [57]. Algoritmayı inceleyecek olursak; k-NN algoritmasının ilk aşaması, sınıflandırması yapılacak öznitelik vektörünün uzaklık değerlerine bakarak en yakın k sayısı kadar komşuyu bulmaktır. İkinci aşamada k kadar komşu en fazla hangi sınıfa ait ise, o sınıf elde edilen sonuç olarak atanır. Bu noktada k sayısını belirlenmesi için en uygun yol, k sayısını toplam eğitim örnekleri sayısının karekökünden daha az seçilebilir. Şekil 3.8’de k-NN algoritmasına ait k değeri 5 ve 7 seçildiğinde elde edilen sınıflandırma sonucu görülmektedir.



Şekil 3.8. k-NN algoritmasının k=5,7 durumunda verilerin sınıflanması

Benzerlik Ölçümü

Görüntü sınıflamada kullanılan birçok yöntem bulunmaktadır. Görüntü sınıflandırma, bir veri seti içinde belirli bir sınıfı oluşturan imgelerin benzerliğinden yola çıkarak ve özelliklerine göre seçilerek gruplandırılması olarak ifade edilebilir. Sınıflandırma işlemleri, geliştirilmiş çeşitli mesafe ölçüleri baz alınarak yapılmaktadır.

Görüntü işleme alanında imge sınıflandırma yapılabilmesi için elimizde olan mevcut verisetindeki verilerden öznitelik çıkartılması gerekmektedir. Belirlenmiş olan verisetindeki RGB renk uzayına sahip görüntülerin çeşitli dönüşümleri ile öznitelikleri elde edilebilir.

Görüntü analizi alanında kullanılan birçok özellik seçimi ve sınıflandırma yöntemleri bulunmaktadır. Sınıflandırma için kullanılacak özelliklerin seçiminde diskriminant özelliklerden de yararlanılmıştır.

Literatürde bulunan sınıflandırma algoritmalarında birçok farklı mesafe ölçüsü kullanılmıştır. Öklid uzaklığı, basit ve kolay uygulanabilir olması ve olumlu sonuçlar vermesi nedeniyle çok sık tercih edilen uzaklık ölçüsüdür. Manhattan mesafesinin ise literatürde sık kullanıldığı görülmektedir. Bir diğer sık kullanılan uzaklık ölçüsü ise Minkowski olarak değerlendirilebilir.

Öklid Mesafesi

Öklid uzaklığı ve Öklid uzaklığının karesi formülleri, veri işleme için kullanılan algoritma ve yöntemlerin çoğunda sıkça kullanılan hata parametresi (uzaklık) hesaplama yöntemleridir. Öklid

uzaklığı ve Öklid uzaklığının karesi formülleri Denklem 3.9'da görülmektedir. Bu denklemlerde, x_i ve y_i , n boyutlu verideki i. boyuta ait veri değerini temsil etmektedir. Öklid Uzaklık Formülü;

$$uzaklık(x, y) = \{\sum_{i=1}^n (x_i - y_i)^2\}^{1/2} \quad (3.9)$$

Manhattan Mesafesi

Manhattan uzaklık hesaplaması için, boyutlar arasındaki ortalama fark işlemi yapılmaktadır. Bu yöntem sıra dışı verilerden az etkilenmekte ve daha kararlı sonuçlar vermektedir. Manhattan uzaklık hesaplaması Denklem 3.10'da görülmektedir.

$$uzaklık(x, y) = \sum_{i=1}^n |x_i - y_i| \quad (3.10)$$

Minkowski Mesafesi

Minkowski uzaklık hesaplaması, p sayıda değişken göz önüne alınarak gözlem değerleri arasındaki uzaklığın hesaplanması için kullanılabilir. Söz konusu uzaklık şu şekilde hesaplanır.

$$d(i, j) = \left[\sum_{k=1}^p (|x_{ik} - x_{jk}|)^p \right]^{1/p} \quad i, j = 1, \dots, n; k = 1, 2, \dots, p \quad (3.11)$$

3.4.3. Çok Katmanlı Algılayıcı

Çok katmanlı algılayıcı (ÇKA) yapay sinir ağları ve mühendislik uygulamalarında en çok kullanılan sinir ağı modellerinden biridir. Birçok katmanlı algılayıcı modeli; bir giriş, bir veya daha fazla ara ve bir çıkış katmanından oluşur. Bir katmandaki bütün işlem elemanları bir üst katmandaki bütün işlem elemanlarına bağlıdır. Ağ yapısında, giriş katmanında herhangi bir bilgi işleme yapılmaz. Buradaki işlem elemanı sayısı tamamen uygulanan problemin giriş nöron sayısına bağlıdır. Ara katman sayısı ve ara katmandaki işlem elemanı sayısı ise deneme-yanılma yolu ile bulunur. Çıkış katmanındaki eleman sayısı ise yine uygulanan probleme dayanılarak belirlenir.

İlk katman her zaman giriş katmanı, son katman ise çıkış katmanları olarak adlandırılır. Giriş ve çıkış katmanları arasındaki diğer katmanlara gizli katman denir. Çok katmanlı algılayıcı yapay sinir ağlarında ağa bir örnek gösterilir ve örnekler sonucunda beklenen sonuçlar ağa verilir (danışmalı öğrenme). Örnekler giriş katmanına uygulanır, ara katmanda işlenir ve çıkış katmanından da çıkışlar elde edilir. Kullanılan eğitim algoritmasına göre, ağın çıkışı ile arzu edilen çıkış arasındaki hata geriye doğru yayılarak, hata minimuma düşünceye kadar ağın ağırlıkları değiştirilir. Girdileri, ağırlıkları ve önyargıları sağladıktan sonra, Multi Layer Perceptron (MLP)'lerin çıktısı aşağıdaki adımlar boyunca hesaplanır.

Girdilerin ağırlıklı toplamaları Denklem 3.12'deki gibi hesaplanır.

$$S_j = \sum_{i=1}^n (W_{ij} \cdot X_{ij}) - \theta_j, j = 1, 2, \dots, h \quad (3.12)$$

Burada n giriş düğümlerinin sayısıdır, W_{ij} , i . düğümden j . düğüme olan bağlantı ağırlığını gösterir θ_j , j . gizli düğümünün önyargısıdır (eşik) ve X_i , i . girişini gösterir. Her gizli düğümün çıktısı aşağıdaki şekilde hesaplanır:

$$S_j = \text{sigmoid}(s_j) = \frac{1}{(1+\exp(s_j))}, j = 1, 2, \dots, h \quad (3.13)$$

Son çıkışlar, gizli düğümlerin hesaplanan çıkışlarına dayanarak aşağıdaki şekilde tanımlanır

$$O_k = \sum_{j=1}^h (W_{jk} \cdot S_j) - \theta'_k, k = 1, 2, \dots, m \quad (3.14)$$

$$O_k = \text{sigmoid}(o_k) = \frac{1}{(1+\exp(-o_k))}, k = 1, 2, \dots, m \quad (3.15)$$

Buradaki W_{jk} , j . gizli düğümden k . çıkış düğümüne bağlantı ağırlığıdır ve O_k , k . çıkış düğümünün önyargısıdır (eşik değeri).

(3.14) ile (3.15) arasında görülebileceği gibi, ağırlıklar ve önyargılar, MLP'lerin nihai çıktısını verilen girdileri tanımlamaktan sorumludur. Girişler ve çıkışlar arasında istenen bir ilişki elde etmek için ağırlıklar ve önyargılar için uygun değerleri bulmak, MLP'lerin eğitiminin tam tanımıdır[58].

3.5. Derin Öğrenme

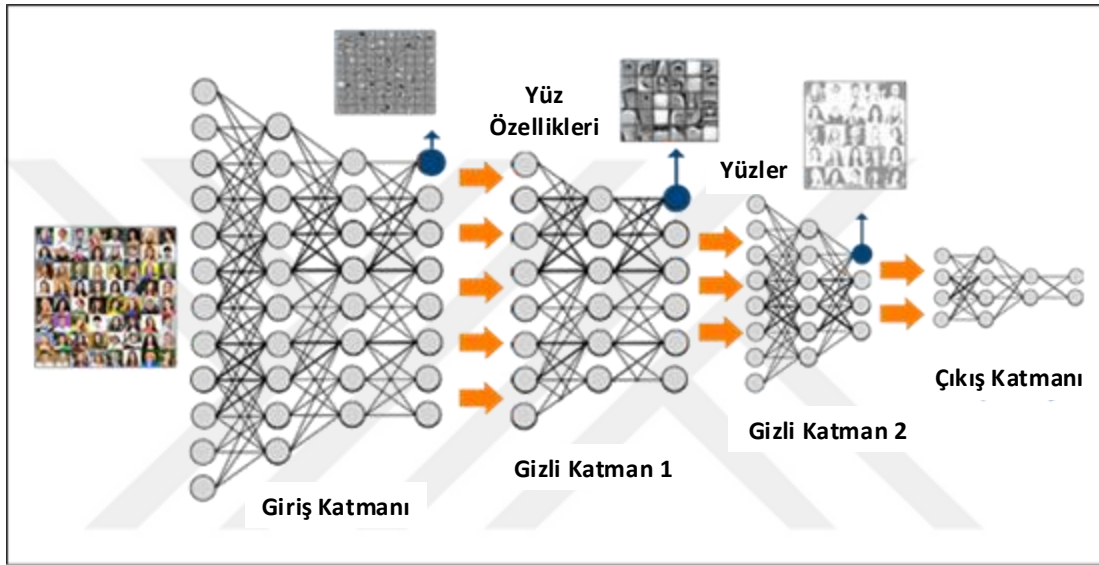
Yapay sinir ağlarındaki gizli katman ve düğüm sayılarının artırılmasına rağmen teknolojik gelişmelerin yetersiz olması, özellikle grafik kartların hız ve kapasitelerinin düşük olması ve hesaplama maliyetlerinin yüksek olması nedeniyle yapay zeka yöntemleri kullanılamamıştır. Fakat yüksek frekansta çalışan paralel işlemcili ve yüksek hafızalı belleğe sahip Grafik İşleme Ünitesi - Graphics Processing Unit (GPU) gibi teknolojilerin üretilmesi ile yüksek sayıda gizli katmandan ve ağırlık parametrelerinden oluşan yapay sinir ağları hesaplama maliyetleri düştüğünden, tekrar yapay zeka teknikleri kullanılmaya başlanmıştır [59].

Günümüzde tıp, uzay, havacılık, dil, eğitim, savunma, üretim ve pazarlama gibi hemen hemen hayatımızın tüm alanlarında kullanılan Derin Öğrenme (Deep Learning), görüntü, video, metinsel veri, ses ve sinyal verilerinin işlenerek anlamlı, tutarlı özniteliklerinin çıkarılmasını sağlayan, bir veri kümesi ile sonuçları tahmin etmeye yarayan makine öğrenmesi yöntemidir. Özellikle, büyük miktarlarda etiketlenmiş eğitim verilerinden özellik çıkarımı yapabilen sistemler oluşturmak için ileri teknoloji, çok seviyeli “derin” sinir ağların kullanılması olan Derin Öğrenme, önemli derecede yatırım ve araştırmanın yapıldığı bir alandır.

Makine öğrenmesi tekniklerinden biri olan derin öğrenme çok katmanlı bir YSA modeli olarak da tanımlanabilir. Yapay Zeka, Bilgisayarlı Görü ve Makine Öğrenme alanlarında birçok yöntem geliştirilmiştir [60-62]. Günümüzde de makine öğrenmesinde ve örüntü tanıma alanları gibi

yapay zekâ alanında çokça kullanılan bir yöntem haline gelmiştir. Derin öğrenme yöntemleri görüntü, ses, sinyal ve metinsel verileri kullanarak sınıflandırma, nesne tanıma, ses tanıma ya da dil çevirileri gibi önemli görevleri başarılı bir şekilde yerine getirmeye çalışmaktadır.

Bir makine öğrenmesi çeşidi olan derin öğrenme, özellik çıkarma ve dönüştürme için doğrusal olmayan birçok katman kullanır. Verileri işleyen katmanlar, önceki katmandaki çıktı parametrelerini girdi olarak alır ve işleyerek bir sonraki katmana işlenmek üzere verir [63]. Şekil 3.9’da örnek bir derin öğrenme mimarisi görülmektedir.



Şekil 3.9. Örnek bir derin öğrenme mimarisi

Derin öğrenme görüntü ya da verilerin alındığı bir giriş katmanı, gizli katmanlar ve çıkış katmanından oluşmaktadır. Giriş katmanında bulunan nöronlarda görüntülerin öznetelik çıkarımları yapılır, gizli katmanlarda öznetelik çıkarımları artırma ve eksiltme yaparak verilerin özellikleri tespit edilir. Son olarak çıkış katmanında verinin sınıf ya da kümesi belirlenmiş olur.

Derin öğrenmede verilerin üst düzey özellikleri, alt düzey özelliklerden türetilerek hiyerarşik bir yapı oluşturulur [64]. Derin öğrenme aslında verinin kendisinin işlenerek öğrenmesini temel alır. Üzerinde öğrenilecek bir görüntünün piksel başına yoğunluk değerlerinin bir vektörü, kenar özellikleri, renk veya boyut yapısı, özel şekiller gibi özelliklerin çıkarımları yapılabilir. Elde edilen özellik çıkarımlarının bazıları test edilen veriyi daha doğru ve üst başarıyla temsil ederken, bazıları da düşük oranda benzerlik sağlar [50]. Derin öğrenme yöntemlerinin başarısı, önerilen algoritmaya ve veriye göre değişmektedir. İncelenen veriye uygun algoritma yazılmalı, veriyi en iyi temsil eden özellik çıkarımları için doğru algoritmalar kullanılmalıdır. Derin öğrenme algoritmalarında, incelenen verinin özellikleri ile sınıf bilgisinin karşılaştırması denetimli öğrenme(supervised learning) olarak tanımlanır. Desen analizi gibi sonuç bilgisi içermeyen veri gruplarını, verinin kendi

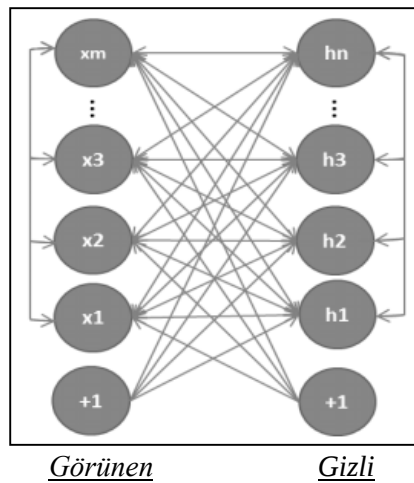
değerlerine göre kümeleme ya da ilişki kurma yöntemine göre ayırmaya denetimsiz öğrenme (unsupervised learning) denir.

3.6. Derin Öğrenme Mimarileri

Derin öğrenme mimarileri veri türüne, yapısına göre değişkenlik gösterdiği gibi veri miktarından da etkilenmektedir. Günümüzde kullanılan birçok derin öğrenme mimarisi bulunmaktadır. Ele alınacak problemlerin yapısına göre tercih yapılır. Verilerin sınıflandırılması, tanınması, tespiti ve tahmini gibi sonuçların üretiminde farklı derin öğrenme mimarileri kullanılır. Kullanılacak veri türleri olan görüntü, ses, metin, sinyal olması derin öğrenme mimarisini seçmede etkili olmaktadır.

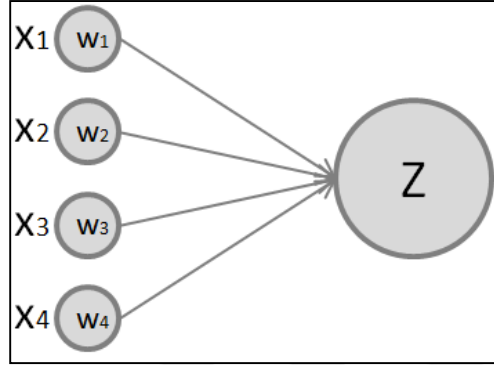
3.6.1. Boltzmann Makineleri

Boltzmann Makinesi (BM), karmaşık kararlar veren sinirlerden oluşan, simetrik bağlantılı bir yönlendirilmemiş olasılıksal devamlı veya ayrık grafik modellerden oluşan yapay sinir ağıdır. Boltzmann Makineleri, görünen veri katmanı ile gizli katman arasında ve katmanların kendi içindeki birimlerin etkileşiminin ortaya çıkarılması ve modellenmesinde kullanılmaktadır. Boltzmann mimarisinde bütün birimler birbirine bağlı olduğundan, özyinelemeli (recursive) bir yapıya sahiptir. Boltzmann Makinelerini oluşturan katmanlardan birinde bulunan bir birimin alacağı olasılıksal değer, bu birime bağlı olan diğer birimlerin değerlerine bağlı olarak hesaplanır. Boltzmann Makineleri verinin modellenmesinden sonra veriye uygun örneklemeler üretme yeteneğine sahiptir [49].



Şekil 3.10. Boltzmann makinesi mimarisi

Boltzmann Makinelerinde gizli katmanlarda öğrenilen değerler, bir sonraki hazırlanan mimari için girdi olmaktadır ve bu işlem birkaç defa tekrar edilerek istenen örüntüler elde edilmektedir. Bu sebeple derin sinir ağları için önemli bir mimaridir [49].



Şekil 3.11. Boltzmann makinesi tek yapay sinir hücresi

Şekil 3.11’de Boltzmann makinesindeki tek hücrenin mimarisi görülmektedir. Burada; Giriş değerleri: X_1, X_2, X_3, X_4 , Bağlantılar arasındaki ağırlıklar: W_1, W_2, W_3, W_4 , Bütün ağırlık değerlerinin toplam değeri Denklem 3.16 ile hesaplanır.

$$Z_i = b_i + \sum_j S_j W_{ij} \quad (3.16)$$

Denklem 3.16’ya göre; b_i giriş katmanından girilen sapma değerleri olarak ele alınır. Her birim için olan sapma değerlerinin toplamı, toplam giriş değerini etkiler.

W_{ij} , i ve j sinirleri arasındaki ağırlık değeridir. S_j , sinirlerin durum değerini temsil etmektedir.

Bütün sinirler arasındaki ağırlıklar simetriktr ve sinirler kendi kendilerine bağ oluşturmazlar. Boltzmann Makinesinin her bir birimi bir enerjiye sahip olduğundan yapay sinir ağındaki genel enerjiler toplamı aşağıdaki Denklem 3.17’de olduğu gibi ifade edilir:

$$E(v) = -\sum_i s_i^v b_i - \sum_{i < j} s_i^v s_j^v w_{ij} \quad (3.17)$$

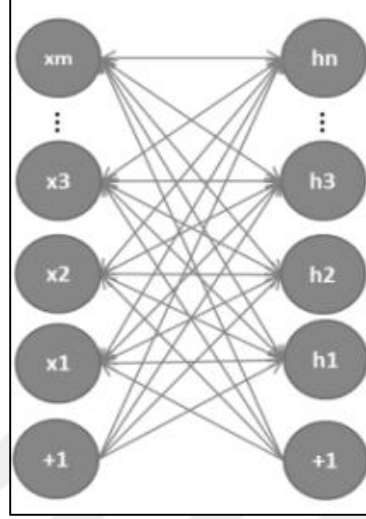
v , durum vektörünün olasılık değerini gösterir. Bu değer bütün ikili durum vektörleriyle bağlantılıdır. İkili durum vektörü aşağıda Denklem 3.18’ de görüldüğü gibi ifade edilir.

$$P(v) = \frac{e^{-E(v)}}{\sum_u e^{-E(u)}} \quad (3.18)$$

Boltzmann Makinesi birçok probleme çözüm olmakla birlikte bazı eksiklikleri bulunmaktadır. Bu eksiklikler kısıtlanmış Boltzmann makinesinin oluşumuna yol açmıştır [29].

3.6.2. Kısıtlanmış Boltzmann Makineleri

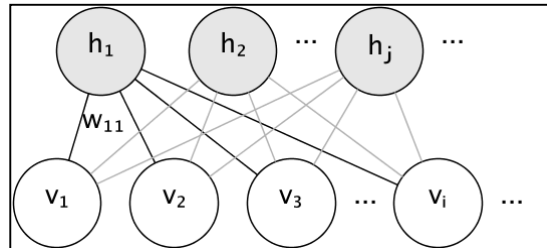
Boltzmann Makinesinin geliştirilmesiyle ortaya çıkmış bir yapay sinir ağ modelidir. Sinirler Boltzmann Makinesinde olduğu gibi birbirine tamamen bağlı değildir.



Şekil 3.12. Kısıtlı Boltzmann makinesi mimarisi

Kısıtlanmış Boltzmann Makinesi-(Restricted Boltzmann Machines-RBM) ters ıraksama yöntemiyle öğrenme yapar. Giriş ve çıkış katmanındaki birimlerde sayısal olarak herhangi bir sınırlandırma bulunmamaktadır. Görünür birimler arasında herhangi bir bağ bulunmamaktadır. Girilen problemin türüne göre ya denetimli ya da denetimsiz yollarla eğitilebilmektedir.

Kısıtlı Boltzmann Makinelerinin en büyük avantajı, katmanlara ait birimler diğer katmanın bilinmesi durumunda koşullu olarak bağımsızdır. Görünen veya saklı birimler dışarıda bırakıldığında diğer katmana ait birimlerin genel olarak çarpanlarına ayrılması mümkündür. Görünen katmanın tüm birimlerinin alabileceği olası tüm değerler üzerinden tek boyutlu integral şeklinde hesaplanabilir [49].



Şekil 3.13. Kısıtlı Boltzmann Makinesi sinir hücreleri yapısı

Şekil 3.13'te gösterildiği gibi görünür ve gizli birimler arasında çift yönlü bir bağlantı vardır. Her bağlantının bir ağırlık değeri vardır. Bu bağlantılar arasındaki ortak atamanın kalitesi aşağıdaki Denklem 3.19'da görüldüğü gibi enerji fonksiyonuyla ölçülür:

$$E(v, h) = \sum_i w_{ij} v_i h_j + \sum_i b_j v_i + \sum_i c_j h_i \quad (3.19)$$

b ve c görünür ve gizli birimlerdeki düğümlerin kesişen vektör değerlerini ifade etmektedir. Görünür ve saklı birimler arasındaki görünür vektörlerin olasılık dağılımları enerji fonksiyonuna göre aşağıdaki Denklem 3.20'de ifade edilir.

$$P(v, h) = \frac{1}{Z} e^{-E(v, h)} \quad (3.20)$$

Z , olasılık dağılımının "1" olmasını sağlayan bütün normalleştirme sabitlerinin toplamını sağlayan ayırma fonksiyonudur. Düğümler birbirine çift yönlü bağlandığından saklı ve görünür birimlerin aktivasyonları birbirinden bağımsızdır.

$$P(v|h) = \prod_{i=1}^m P(v_i|h) \quad (3.21)$$

Görünür birimleri m temsil etmektedir. Saklı birimlerin yapılandırma değeri h tır. Görünür birimdeki h değerinin koşullu olasılığı şu şekilde ifade edilir:

$$P(v|h) = \prod_{j=1}^m P(v_j|h) \quad (3.22)$$

n saklı birimleri temsil etmektedir. Bu formüllere göre bağımsız aktivasyon olasılıkları Denklem 3.23 ve 3.24'te ifade edilmektedir.

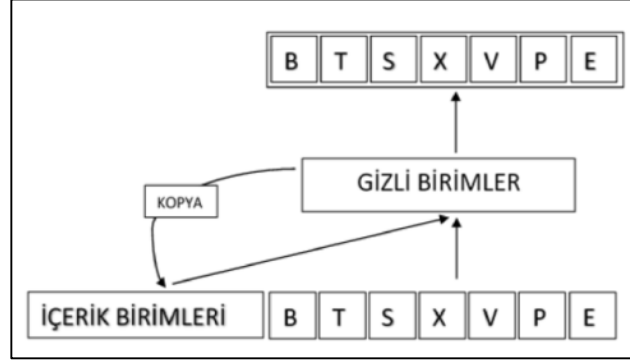
$$P(h_i = 1|v) = \partial(a_i + \sum_{j=1}^m w_{i,j} h_j) \quad (3.23)$$

$$P(h_j = 1|v) = \partial(b_j + \sum_{i=1}^m w_{i,j} v_i) \quad (3.24)$$

3.6.3. Tekrarlayan Sinir Ağı

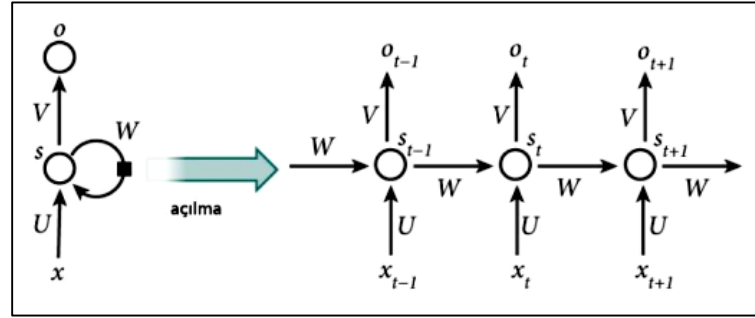
Elman tarafından tasarlanan basit tekrarlayan sinir ağları (Simple Recurrent Network-SRN) dil bilimciler ve psikanaliz için çığır açan bir yaklaşım olmuştur [65]. Elman'ın yayınladığı makalede konuşma akışı üzerindeki gizli yapı, üzerinde çalışılan bir öğrenme sürecini temsil ediyordu. Örüntü kümelemesinde fiil ve isim kategorizasyonu açık şekilde birbirinden

ayrılmaktaydı. Ayrıca canlı-cansız, insan-hayvan, avcı-yırtıcı gibi kategorilerde ayrılmıştı. Şekil 3.14'te Elman basit tekrarlayan sinir ağı örneği yer almaktadır.



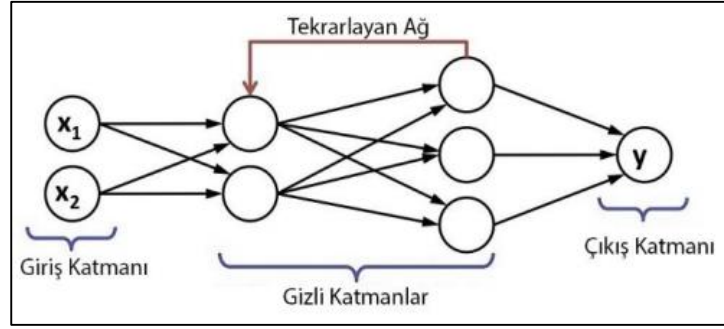
Şekil 3.14. Basit tekrarlayan ağı mimarisi

Tekrarlayan sinir ağları (Recurrent Neural Network-RNN), sadece ağı giren giriş örneklerini değil, daha önce zaman serisi içerisindeki giriş örneklerini de alırlar. Bu sinir ağının amacı, ardıl şekilde gelen verilerin kullanılmasıdır. Geleneksel sinir ağlarında girişler birbirlerinden bağımsız olarak ağı giriş yapar. Ancak tekrarlanan sinir ağlarında dizideki her verinin çıktısı önceki hesaplamalara bağlıdır. Şekil 3.15'te tekrarlayan sinir ağının nasıl açıldığı gösterilmektedir.



Şekil 3.15. Tekrarlayan sinir ağının açılması

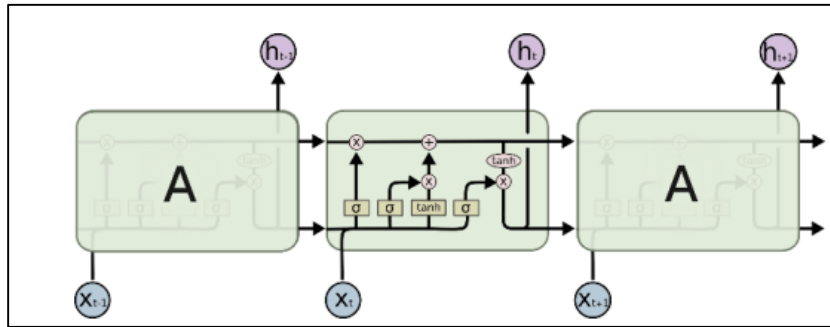
Tekrarlayan sinir ağları daha çok dil çevrimleri için kullanılmakla birlikte, arka arkaya meydana gelen bir dizi yapıdan bir sonraki noktanın tahmin edilmesi işlemini yapar. Buna basitçe örnek vermek gerekirse, finansal bir hareketin dizi şeklinde zaman serileri ile giriş verisi olarak kullanılarak sonraki bir zamanda hangi durumda olacağının tahmin edilmesi olarak gösterilebilir. Ya da kurulacak bir cümlede art arda gelen kelimelerin akabinde cümlemin devamının nasıl geleceğini gösteren kelimenin tahmin edilmesi işlemi örnek olarak verilebilir. İki tür RNN vardır bunlar; İki yönlü RNN'ler (Bidirectional RNNs) ve Derin RNN'lerdir (Deep RNNs). Şekil 3.16'da tekrarlayan sinir ağı modeli yer almaktadır.



Şekil 3.16. Tekerleyen sinir ağı modeli

3.6.4. Uzun-Kısa Süreli Hafıza (Long Short-Term Memory - LSTM)

RNN mimarilerinde zaman dizeleri aralarında bağlam boşlukları olması halinde sonraki dizinin tahmin edilmesi çok zor bir durumdur [49]. Bu durum RNN'ler için oldukça dezavantajlıdır. Hochreiter ve Schmidhuber yapmış oldukları çalışmada bu durumu ortadan kaldıracak uzun ve kısa süreli hafıza LSTM öne sürmüşlerdir [66], LSTM ağlarının RNN ağlarından bir farkı yoktur. Fakat gizli durumu hesaplamak için LSTM ağlarında bir yapı kullanılır. LSTM içerisinde hafıza hücreleri yer alır. Bu, önceki durumu ve girdi bilgisini tutan bir hücredir. Ağ mimarisi içerisinde yer alan bu hücreler hangi verinin tutulacağına ya da hangi verinin silineceğine karar verirler. Sonraki aşamada ise önceki durumun mevcut bellek ile giriş verisini birleştirirler [67]. Böyle bir yaklaşımla uzun vadeli bağımlılıkların ortadan kaldırılarak veri dizilerinin devam ettirilmesi mümkün kılınır. Şekil 3.17'de LSTM bloğu görülmektedir.

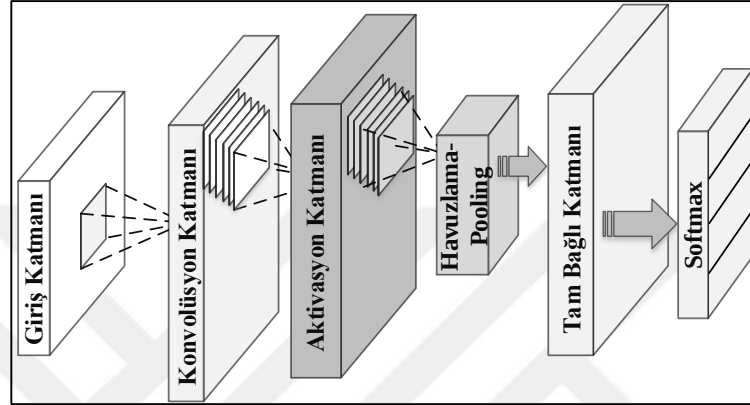


Şekil 3.17. LSTM bloğu

3.6.5. Evrimsel Sinir Ağları (Convolutional Neural Networks - CNN)

Evrimsel Sinir Ağları(ESA) aslında çok katmanlı algılayıcıların bir türüdür. Yann LeCun tarafından 1990 yılında yapılan çalışma ile geliştirilen ve günümüzde artan bir şekilde kullanılan CNN'ler çok katmanlı sinir ağlarının ileri yayılım algoritmasının özel bir türüdür [68]. CNN, en düşük sayıda önilem ile piksel görüntülerin işlenmesi ve doğrudan görsel kalıplar çıkarmak için

geliştirilmiştir. CNN'lerin, diğer sınıflandırma algoritmalarından daha iyi bir performans göstermesinin nedenlerinden birisi, öznetelik çoğaltma ve özetleme katmanlarına sahip olmasıdır. Bu özellikler yardımıyla mevcut veriyi daha iyi sınıflandırmakta ve tanımaktadır [11]. Evrişim süreci, bir nöronun kendi uyarı alanından uyarılara verdiği cevap olarak düşünülebilir. CNN katmanları gelişmekle birlikte temel olarak dört katmandan oluşmaktadır. Şekil 3.18'te örnek ESA mimarisi görülmektedir.



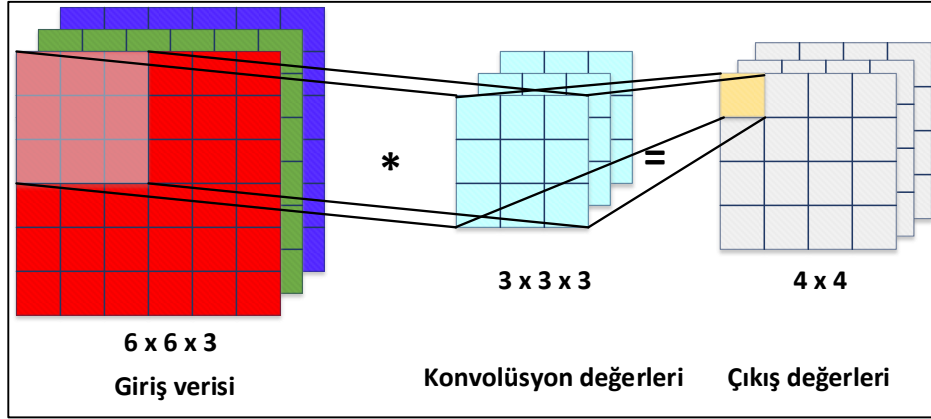
Şekil 3.18. Evrişimsel sinir ağı mimarisi

Şekil 3.18'de görüldüğü gibi CNN modelleri konvolüsyon katmanı, aktivasyon katmanı, alt-örnekleme (pooling) katmanı ve tam bağlı katmanlardan oluşmaktadır.

Konvolüsyon Katmanı

Bu katman temel olarak filtreler ve özellik haritalarından oluşmaktadır. Filtreler, katmanın nöronlarını temsil eder. Rastgele sayılar üretilerek ağırlıklar oluşturulur. Bu ağırlıklar ile imgenin belirtilen değerleri konvolüsyon işlemine tabi tutularak çıktı değeri elde edilir. Bu işlemler, belirli aralıklarla adım adım ilerler ve sisteme giren imgelerin tüm bölgesi taranana kadar devam eder.

Giriş katmanında bulunan imgenin sayısal değerleri konvolüsyon katmanında üretilmiş rastgele sayılara (0-1 arasında sayılar) sahip matriste elemanları ile çarpılır ve tüm değerlerin toplamı çapraz korelasyon yöntemiyle çıkış matrisinin ilgili elemanı olarak kaydedilir. Giriş verisi Şekil 3.19'da görüldüğü gibi çok kanallı da olabilir, tek kanallı da olabilir. Tek kanallı verilerde konvolüsyon işlemi basitçe yapılırken, çok kanallı özellikle görüntü verilerinde bu işlemler daha karmaşık olmaktadır.



Şekil 3.19. Konvolüsyon işlem aşamaları

Bu tez çalışmasında karayolu görüntülerinin analizi için renkli 3 kanallı görüntüler üzerinde çalışmalar yapılmıştır. Bu nedenle konvolüsyon işlemi 3 kanallı olarak yapılmıştır. Çıkış işaretinin kanal sayısı da uygulanan filtre kanalı/sayısı ile eşit olarak hesaplanır. Giriş verilerinden özellik çıkarımı için yapılan bu konvolüsyon işlemi aslında iki ayrı fonksiyonun çarpımından ve ardından bu çarpımlardan elde edilen değerlerin toplamından oluşur. Konvolüsyon işlemi basitçe filtre olarak adlandırılan kernel(çekirdek) fonksiyonun ana sayısal veri üzerinde piksek olarak kaydırılmasıdır. Her pencerede kaydırma işlemi, elemanların filtre matrisi ile birim çarpımının toplamı şeklindedir. Denklem 3.25'te f ana veri, h kernel olmak üzere konvolüsyon işleminin matematiksel hesaplaması görülmektedir.

$$f * h = \sum_k \sum_l f(k, l) h(i - k, j - l) \quad (3.25)$$

$$f = \begin{bmatrix} f_1 & f_2 & f_3 \\ f_4 & f_5 & f_6 \\ f_7 & f_8 & f_9 \end{bmatrix} \quad h = \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{bmatrix} \quad (3.26)$$

$$f * h = f_1 * h_9 + f_2 * h_8 + f_3 * h_7 + f_4 * h_6 + f_5 * h_5 + f_6 * h_4 + f_7 * h_3 + f_8 * h_2 + f_9 * h_1 \quad (3.27)$$

Tüm görüntü boyunca verinin matrisleri, belirlenen stride(matris kayma boyutu) parametresi ile kaydırılarak özellik haritası olarak çıktısı sonraki bir katmana verilmek üzere üretilir.

Yapay sinir ağı olan $f = \sum w_i x_i + b$ formülü ile x giriş değeri w ağırlık değerlerinin çarpılarak matris toplanması ve son olarak varsa b bias değerinin eklenmesi ile oluşur.

Konvolüsyon işleminde kernel için belirlenen giriş görüntüsü üzerinde kaydırıldığında (adım), bir özellik haritası meydana gelir. Tüm kernellerin değerleri oluşturduktan sonra değerler birleştirilir. Giriş verisinin boyutuna göre bir artış söz konusudur. 3 boyutlu matris veri giriş

parametresi ise çıkışında 4 boyutlu tensör oluşturacaktır. Yeni boyutun kernel boyutundan fazla olması boyut filtrenin kullanılmasından kaynaklanmaktadır. Tensör ve kernel(filtre)'nin derinlikleri aynı değerlere sahip olmalıdır. Denklem 3.28 ve 3.29'da çıkış boyutunun matematiksel hesaplaması görülmektedir.

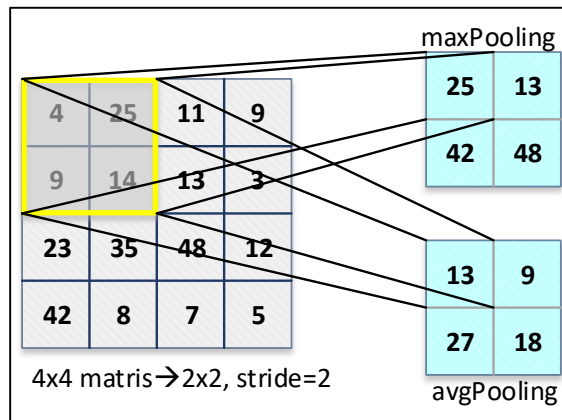
$$H_0 = \frac{H_i - f + 2p}{s} + 1 \quad (3.28)$$

$$W_0 = \frac{W_i - f + 2p}{s} + 1 \quad (3.29)$$

Özellikle görüntü işlemede kullanılan verilerin yükseklik ve genişliklerine göre kernel boyutu ayarlanmalıdır. Denklem 3.28 - 3.29'da çıkış boyutu için, H, W giriş veri boyutu, f kernel(filtre) boyutu, P padding(boş dolgu) ve S stride nitelemektedir.

Pooling (Havuzlama) katmanı

Alt-örneklem olarak da bilinen pooling(havuzlama) katmanı, konvolüsyon katmanı kullandıktan sonra elde edilen çıkış katmanındaki parametreler üzerinde farklı işlemler uygulanarak daha derin özelliklerin çıkarılması istenmektedir. Alt-örnekleme katmanı bu amaçla kullanılan bir katman olmasının yanında, giriş verisini daha düşük boyutlu temsillerine indirgemek için kullanılmaktadır. Pooling katmanı bir önceki katmandan elde edilen parametrelerin en belirginlerini tespit etmek için genellikle konvolüsyon katmandan sonra kullanılır. Bu katmanda yaygın olarak maksimum(maxPooling) ve ortalama(avgPooling) teknikleri kullanılmaktadır. Pooling işleminin matris boyutu ve stride olmak üzere iki önemli parametresi vardır. Şekil 3.20'de pooling örneği gösterilmiştir.



Şekil 3.20. Pooling katmanı örneği

Şekil 3.20 incelendiğinde, MaxPooling katmanı, matristeki 2x2 boyutunda en yüksek değeri veya ortalama değeri alarak stride=2 ile matrisi dolaşır. Sonuçta 16 elemanlı bir matristen 4 elemanlı bir matris üretilerek veri boyutu azaltılmış olur. Giriş matrisinin kanal sayısını sabit tutan Pooling katmanı sadece yükseklik ve genişlik değerini azaltarak verideki ayırt edilebilirliği yüksek parametreleri tespiti için kullanılır. Pooling katmanında ağı eğitimi bulunmamaktadır. Özellikle RGB gibi 3 boyutlu görüntülerde oluşan hesaplama karmaşıklığını azaltmak için kullanılan bir adımdır. Kapsül sinir ağlarının mimarı Hinton'ın [69] çalışmasında Pooling katmanının verideki önemli bazı bilgilerinde kaybolmasına sebep olduğundan başarıyı düşürdüğünü iddia etmiştir.

Aktivasyon katmanı

Aktivasyon işlemi ile bir özellik haritasının her bileşenine uygulanan doğrusal olmayan bir fonksiyonuyla doğrusal bir filtre elde edilir. ESA modellerinde kullanılan en yaygın yöntem, sıfıra eşikleme olarak bilinen ReLU'dur. Bu yöntem, sigmoid ve tanjant fonksiyonu ile karşılaştırıldığında daha hızlı sonuç üretmektedir. Bu durum özellikle işlem kapasitesi bakımından kısıtlı olan bilgisayarlar için belirgin bir hız artışına olanak sağlamaktadır [68, 70]. ReLU dışında yaygın olarak kullanılan aktivasyon fonksiyonları: Sigmoid, TanH(hiperbolik), PReLU, Softmax, SoftPlus, ELU, Swish'dir. Aktivasyon fonksiyonlarının matematiksel olarak ifadesi aşağıdaki denklemlerde görülmektedir.

Rectified Linear Unit (ReLU): En sık kullanılan ve $[0, +\infty)$ aralığında değer alabilen aktivasyon fonksiyonudur. Denklem 3.30'da ReLU aktivasyon hesaplaması görülmektedir.

$$f(x) = \begin{cases} x, & x \geq 0 \\ 0, & x < 0 \end{cases} \quad (3.30)$$

Denklem 3.30'da görüldüğü gibi, x sıfır veya üstünde bir değere sahipse çıkış kendisi olacaktır, diğer durumda çıkış sıfır olacaktır. Kurulacak modelin daha hızlı sonuç üretebilmesinde bu fonksiyonun negatif ekseninde 0 değerlerini alması oldukça etkili olacaktır. Hesaplama yükünün sigmoid ve hiperbolik tanjant fonksiyonlarına göre az olması, çok katmanlı ağlarda daha çok tercih edilmesine sebep olmuştur. Ancak 0 altındaki değerlerin sonucu 0 olacağından türevlenemediği için öğrenme tam olarak gerçekleşmeyecektir. Bu yüzden özellikle negatif değerli sinyal verilerinde kullanılması tercih edilmez.

Sigmoid: Doğrusal olmayan, hem 0'ın altında hem de üstünde değerlerde türevlenebildiği için öğrenme gerçekleşir. Denklem 3.31'de görüldüğü gibi özellikle x , -2 ile +2 arasında iken y değerleri diğer aralıklara göre hızlı şekilde değişir. Bu özellik x 'in diğer değer aralıklarında daha düşüktür.

$$f(x) = \frac{1}{1+e^x} \quad (3.31)$$

Bu fonksiyonun diğer avantajı $(-\infty, +\infty)$ karşılaşıldığında her zaman $(0,1)$ aralığında değer üretmesidir. Dezavantajı ise $-2, +2$ aralığı dışındaki diğer bölgelerde y değeri çok az tepki vermesidir. Bu da türev değerini düşük çıkarmaktadır.

Softmax: Softmax fonksiyonu x girdi değerinin belirli sınıfa ait olma olasılığını, $0-1$ aralığında y çıktı değerleri üretmek için kullanılır. Yani olasılıksal bir yorumlama gerçekleştirir. Denklem 3.32’de matematiksel hesaplaması görülmektedir. Sınıflayıcı olarak kullanıldığında yüksek performans gösterir.

$$f(x) = \frac{e_i^x}{\sum_i^k e_i^x} \quad (3.32)$$

$$i = [1..k] \quad (3.33)$$

Tanh (Hiperbolik tanjant): Fonksiyonun aralığı $(-1,+1)$ olarak tanımlanır ve türevinin daha dik olması ile daha çok değer alabilmesi diğer fonksiyonlardan farklıdır. Tanh fonksiyonu daha hızlı öğrenme ve daha iyi bir sınıflama için kullanılabilir. Özellikle çok farklı örüntü değerleri içeren verilerde daha verimli sonuçlar üretebilir. Denklem 3.34’te Tanh fonksiyonun matematiksel hesaplaması görülmektedir.

$$f(x) = \tanh(x) = \frac{2}{1+e^{-2x}} - 1 \quad (3.34)$$

Leaky (Sızıntı) ReLU: ReLU fonksiyonunda 0 değeri altındaki verilerde öğrenme gerçekleşmediği için buna alternatif olarak önerilen Leaky ReLU fonksiyonu negatif düzlemde 0.01 değerlik azalışla hesaplamaktadır.

$$f(x) = \max(0.1x, x) \quad (3.35)$$

ELU: Üstel Doğrusal Birim veya yaygın olarak bilinen adı ELU, maliyeti sıfıra daha hızlı bir şekilde dönüştüren ve daha doğru sonuçlar üreten bir işlemdir. Diğer aktivasyon fonksiyonlarından farklı olarak Denklem 3.36’da görüldüğü gibi x ’in 0’ın altında olması durumunda bir alfa sabitine sahiptir.

$$f(x) = \begin{cases} x, & x > 0 \\ \alpha(e^x - 1), & x \leq 0 \end{cases} \quad (3.36)$$

ELU, negatif girişler dışında ReLU'ya çok benzer. ELU çıkışı negatif değerde 0'ın altına düştüğünde yavaş pürüzsüzleşirken ReLU keskin bir şekilde pürüzsüzleşmektedir.

Swish (A Self-Gated - Kendinden Geçitli) Fonksiyonu: Negatif bölgede de değer alması ReLU ve diğer fonksiyonlardan farkı olarak değerlendirilebilir. Denklem 3.37'de Swish aktivasyon fonksiyonun matematiksel hesaplaması görülmektedir.

$$f(x) = 2x\sigma(\beta x) = \begin{cases} x, & \beta = 0 \\ 2 \max(0, x), & \beta \rightarrow x \end{cases} \quad (3.37)$$

Swish aktivasyon fonksiyonun Leaky ReLU'dan farkı ise, negatif bölgede almış olduğu değerlerin doğrusal olmamasıdır. Bu da bize negatif bölgede öğrenmenin daha iyi gerçekleşebileceğini göstermektedir. Aynı zamanda Swish fonksiyonuna özgü olarak girdinin artması, çıktının her zaman artacağı anlamına gelmemektedir. Denklem 3.38'de görüldüğü gibi β değeri öğrenilebilir bir parametre olarak karşımıza çıkmaktadır. β değeri çok büyük olduğunda veya 1'in altında çok küçük olduğunda fonksiyon ReLU fonksiyonuna yakınsar.

Basamak fonksiyonu:

$$f(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases} \quad (3.38)$$

Lineer (Doğrusal) fonksiyon:

$$f(x) = x \quad (3.39)$$

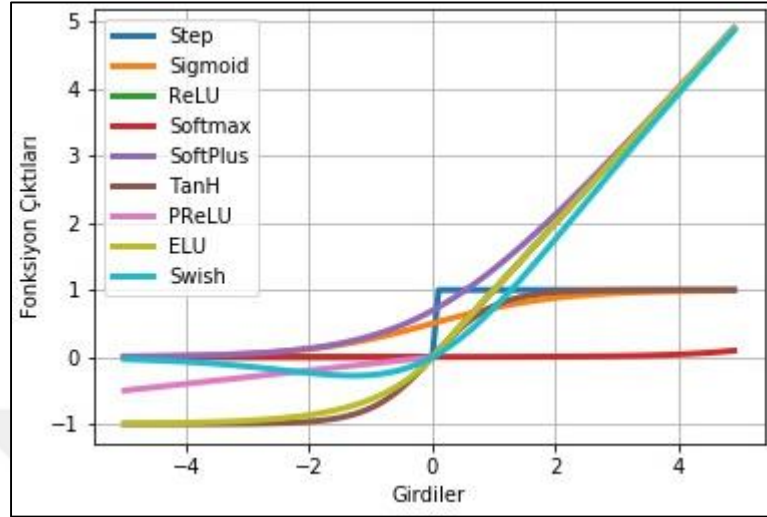
Softmax katmanı

Verileri sınıflamak için kullanılan bu katmanın genel itibarıyla en büyük özelliği, çok nitelikli verilerin sınıflandırılmasını başarılı bir şekilde yapmasıdır. Normal bir sınıflandırıcıda sınıf sayısı oldukça düşük olmasına karşın Softmax, verinin sayısı kadar veriyi sınıflara ayırabilir. Girdinin sınıfa ait olma olasılığını 0–1 aralığında değerler üretmekle belirlemektedir. Denklem 3.40'ta elde edilen çıkış parametresinin değerinin tüm sınıf değerlerinin toplamına bölünmesi ile ortaya çıkan $y_r(x)$ değerinin matematiksel hesaplaması görülmektedir. Denklem 3.41'de ise y_r 'nin değer aralığı görülmektedir.

$$y_r(x) = \frac{\exp(a_r(x))}{\sum_{j=1}^k \exp(a_j(x))} \quad (3.40)$$

$$0 \leq y_r \leq 1 \text{ ve } \sum_{j=1}^k y_j = 1 \quad (3.41)$$

Şekil 3.21’de tüm aktivasyon fonksiyonların matematiksel değerlere göre grafikleri görülmektedir.



Şekil 3.21. Tüm aktivasyon fonksiyonların grafikte gösterimi

Diğer Parametreler

Batch Boyutu: Yapay sinir ağlarında tüm veriler nörondan aynı anda işletilerek o ağı eğitmekte ve nöron ağırlıkları buna göre hesaplanmaktadır. Bu hesaplama, maliyet ve kapasite bakımından oldukça yük getirmektedir. Bu hesaplama işlemi veri sayısına bağlı olarak o oranda kat kat artmaktadır. Bu yüzden ağı eğitimi performansını artırmak ve belleği daha iyi kullanmak için tüm veri, belirli gruplar halinde alınmakta ve küçük gruplar halinde işlem yapılmaktadır. Bu yöntem mini-batch yöntemi denir. Model, her bir mini-batch üzerinde ileri besleme işlemi gerçekleştirir ve hata değerini hesaplar. Daha sonra her bir mini-batch üzerinde geri yayılım yöntemi kullanarak gradyan hesaplamalarını gerçekleştirir ve ağırlıkları günceller. Genel olarak mini-batch boyutlarının çoğunlukla aynı olup %1 - %5 arasında ideal olarak seçilmektedir. %1’den küçük seçilirse gürültü çok fazla olmakta, %5’ten fazla seçilirse modelin eğitim başarımı oldukça düşmekte ve eğitim tamamlanamamaktadır [71].

Epoch (Eğitim Tur Sayısı): Derin öğrenmede model eğitilirken mini-batch’de tüm veri tek seferde eğitilmez, parça parça eğitilir. Her parça eğitiminin sonunda geri yayılıma göre ağırlıklar hesaplanmaktadır. Her bir devir tüm veri setinin ileriye ve geriye doğru yöntemleri ile modelin tekrar eğitilip ağırlıkların güncellemelerinin yapılmasına Epoch yöntemi denir. Epoch değeri düşük tutulursa, model eksik öğrenir ya da yüksek tutulursa model aşırı öğrenme, ezber yapar. Epoch sayısı tahmini olarak belirlenmektedir, fakat tensorflow desteğiyle başarı, istenilen seviyeye ulaşıncaya kadar durur [72].

Learning Rate (Öğrenme Hızı): Derin öğrenmede ağırlık parametresi güncellenmesi geri yayılımla yapılmaktadır. Bu güncelleme ters türev alınarak farkın bulunması ve bu farkın öğrenme hızı değeri ile çarpılması ile elde edilebilir. Bu değer sabit değer ya da belirli düzeyde artan veya azalan şekilde ayarlanabilir olup 0,1 ile 0,000001 arasında seçilmektedir. Learning rate yüksek seçilirse, modelin eğitim verisinden çok etkilenebilir. Küçük değer seçilirse, yavaş ilerlemesine ve öğrenimin çok uzun sürmesine sebep olur [73].

Momentum Katsayısı (Momentum): Momentum katsayısı ile yeni üretilen kullanılacak değeri olduğu gibi almak yerine, bir önceki değeri de momentum katsayısı oranında işin içine katarak yeni değer hesaplanır [73]. Böylece gürültü ve grafikteki salınımlar azaltılarak daha hızlı bir yöntem oluşturulur. Çalışmalarda genel olarak momentum değeri 0.9 olarak belirlenir ve değer aralığı 0.8-0.99 dur.

Optimizasyon Yönteminin Seçimi: Derin öğrenme aslında bir optimizasyon problemidir. Optimizasyon fonksiyonu, doğrusal olmayan problemlerin çözümünde optimum değeri bulmak için kullanılan yöntemdir. Çalışmalarda kullanılan optimizasyon yönteminin en yaygın olanları: Adam, Adamax, Adadelata AdaGrad, Sgd optimizasyon algoritmalarıdır. Her optimizasyon fonksiyonunun diğerine göre artışı veya eksisi vardır.

3.7. Evrişim Sinir Ağları Tabanlı Geliştirilen Derin Öğrenme Modelleri

Günümüz teknoloji dünyasının en popüler konusu olan derin öğrenme, hemen hemen tüm alanlarda kullanılmaktadır. Çoğunlukla tanıma ve sınıflandırma için kullanılan modeller hazırlanacak veri setlerine göre değişebilmektedir. Bu modellerde giriş katmanından sınıflandırma ya da çıkış katmanına gidildikçe havuzlama (pooling) yöntemi kullanılarak öznetelik haritalarının çözünürlüğü düşürülür ve sadece çıkıştaki en küçük çözünürlüklü harita kullanılarak sınıflandırma yapılır. Literatürde farklı amaçlar için geliştirilmiş birçok derin öğrenme modelleri bulunmakta ve bu modeller ile görüntü, sinyal, ses verilerinin tanınması, sınıflandırılması için testler yapılmaktadır. Bu tezde önerilen model dışında, literatürde çok çalışılan ve başarımları oldukça yüksek çıkan AlexNet, LeNet, VggNet, ResNet gibi modeller incelenmiştir.

3.7.1. AlexNet

Alex Krizhevsky, Ilya Sutskever ve Geoffrey Hinton tarafından 2012 yılında geliştirilen bu ESA modeli derin öğrenmenin tekrar popüler hale gelmesini sağlamıştır. Geliştirmiş olduğu modelde [74], birbirini takip eden evrişim ve pooling katmanları bulunmasından dolayı diğer LeNet modeline çok benzemektedir. Aktivasyon fonksiyonu olarak ReLU fonksiyonunu ve pooling katmanında maxpooling kullanmıştır. Tablo 3.1’de AlexNet modelinin katman parametreleri görülmektedir.

Tablo 3.1. AlexNet derin öğrenme modelinin katman parametreleri

Katmanlar	Boyut	Kernel Sayısı	Nöron Sayısı
Giriş Katmanı	224×224×3		150.528
Konvolüsyon	11×11×3	96	253.440
ReLU			
Normalizasyon			
Pooling			
Konvolüsyon	5×5×48	256	
ReLU			
Normalizasyon			
Pooling			
Konvolüsyon	3×3×256	384	64.896
ReLU			
Konvolüsyon	3×3×192	384	64.896
ReLU			
Konvolüsyon	3×3×192	256	64.896
ReLU			
Pooling			
Tam Bağlı			4096
ReLU			
DropOut			
Tam Bağlı			4096
ReLU			
DropOut			
Tam Bağlı			80
Softmax			

Tablo 3.1’de görüldüğü gibi AlexNet modeli 5 konvolüsyon katmanı, 3 pooling katmanı, 7 ReLU aktivasyon katmanı, ara katmanlarda normalizasyon, 2 Dropout katmanı, tam bağlı katman ve son olarak Softmax sınıflandırıcı katmanından oluşmaktadır. GPU mimarisi üzerinde çalışan oldukça derin ve güçlü bir modeldir. Yaklaşık 60 milyon parametre hesabına ulaşmaktadır. Özellikle görüntü işleme alanında oldukça başarılı bir modeldir.

3.7.2. LeNet

İlk ESA modeli olarak değerlendirilir, 1998 yılında geliştirilmiş ve başarılı sonuçlar vermiştir. Yann LeCun ve ekibi tarafından posta numaraları, banka çekleri üzerindeki sayıların okunması için geliştirilmiştir [75]. MNIST veri seti üzerinde deneyler gösterilmiştir. Bu modelde sonradan geliştirilen diğer modellerden farklı olarak boyut azalma adımlarında max-pooling yerine average (ortalama) pooling işlemi yapılmaktadır. Ayrıca aktivasyon fonksiyonu olarak da sigmoid ve hiperbolik tanjant kullanılmaktadır. Tablo 3.2’de LeNet mimarisinin katmanları ve parametreleri verilmiştir.

Tablo 3.1. LeNet derin öğrenme modelinin katman parametreleri

Katmanlar	Boyut	Kernel Sayısı	Nöron Sayısı	Adım (Stride)	Aktivasyon fonksiyonu
Giriş Katmanı	32×32		150.528		
Konvolüsyon	28×28	5×5	253.440	1	tanh
AvgPooling	14×	2×2		2	tanh
Konvolüsyon	10×10	5×5		1	tanh
AvgPooling	5×5	2×2		2	tanh
Konvolüsyon	1×1	5×5		1	tanh
Tam Bağlı	84				tanh
Tam Bağlı	10				Softmax

Toplamda 3 konvolüsyon katmanı, 2 avgpooling katmanı ve tam bağlı katman ile softmax sınıflandırıcı katmanı kullanılmıştır. Son katman olan tam bağlı katmana giren parametre sayısı $5 \times 5 \times 16 = 400$ 'dür ve y çıkışında 0–9 arasındaki rakamları sınıflandırdığı için 10 sınıflı softmax bulunmaktadır. Bu ağ modelinde 60 bin parametre hesaplanmaktadır. Ağ boyunca matrisin yükseklik ve genişlik bilgisi azalırken, derinlik (kanal sayısı) değeri artmaktadır.

3.7.3. VggNet

Basit bir ağ modeli olup, öncesindeki modellerden en önemli farkı evrişim katmalarının 2'li ya da 3'lü kullanılmasıdır. Tam bağlantı (Fully connected-FC) katmanında $7 \times 7 \times 512 = 4096$ nöronlu bir öznitelik vektörüne dönüştürülür. İki FC katmanı çıkışında 1000 sınıflı softmax başarımı hesaplanır. Yaklaşık 138 milyon parametre hesabı yapılmaktadır [70]. Diğer modellerde olduğu gibi girişten çıkışa doğru matrislerin yükseklik ve genişlik boyutları azalırken derinlik değeri (kanal sayısı) artmaktadır. Tablo 3.3'te VggNet-19 derin öğrenme modelinin katman parametreleri verilmiştir.

Tablo 3.3. VggNet-19 derin öğrenme modelinin katman parametreleri

Katmanlar	Boyut	Kernel Sayısı	Adım (Stride)	Aktivasyon fonksiyonu
Giriş Katmanı	224×224×3			
2×Konolüsyon	224×224×64	3×3	1	ReLU
MaxPooling	112×112×64	3×3	2	ReLU
2×Konolüsyon	112×112×128	3×3	1	ReLU
MaxPooling	56×56×128	3×3	2	ReLU
2×Konolüsyon	56×56×256	3×3	1	ReLU
MaxPooling	28×28×256	3×3	2	ReLU
3×Konolüsyon	28×28×512	3×3	1	ReLU
MaxPooling	14×14×512	3×3	2	ReLU
3×Konolüsyon	14×14×512	3×3	1	ReLU
MaxPooling	7×7×512	3×3	2	ReLU
Tam Bağlı	25088			ReLU
Tam Bağlı	4096			ReLU
Tam Bağlı	4096			ReLU
Tam Bağlı	1000			Softmax

Modelin her evrişim katmanı çıkışında farklı ağırlıklara sahip filtreler hesaplanır ve katman sayısı artıka filtrelerde oluřan öznitelikler de artmaktadır.

3.7.4. ResNet

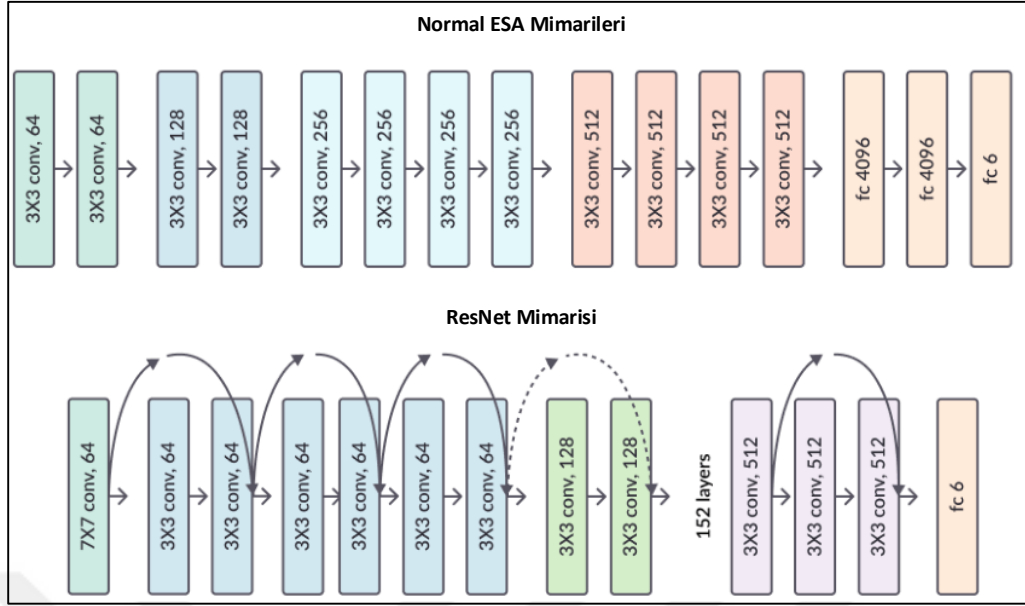
Kaiming He ve ark. tarafından 2016 yılında geliřtirilen bu ađ modelinin geręek anlamda derinleřmeye bařladıđı, kendinden önceki modellerden farklı bir mantıđı barındıran ResNet; artık deđerlerin (residual value) sonraki katmanları besleyen blokların (residual block) modele eklenmesiyle oluřmaktadır. ResNet bu özelliđi ile klasik bir model olmaktan çıkmaktadır [68].

Dođrusal ve ReLU arasında iki katmanda bir eklenen bu deđer ařađıdaki gibi sistemdeki hesabı deđerřtirir. Önceden gelen $a[l]$ deđerı, $a[l+2]$ hesabına eklenmiř olur. Tablo 3.4'te ResNet mimarisinin katman parametreleri verilmiřtir.

Tablo 3.4. ResNet derin öđrenme modelinin katman parametreleri

Katmanlar	Boyut	Kernel Sayısı	Adım (Stride)
Giriř	224×224×3		
Konolüsyon	112×112×64	7×7×64	2
Konolüsyon	56×56×64	3×3×64	2
MaxPooling	56×56×64	3×3	2
Konolüsyon	28×28×128	3×3×128	2
Konolüsyon	28×28×256	3×3×256	2
Konolüsyon	14×14×512	3×3×512	2
AvgPooling	1×1×512	3×3	
Tam Bađlı	1000	512×1000	
Softmax	1000		

Tablo 3.4'te görüldüđü gibi toplamda 5 konvolüsyon katmanı bir avgpooling katmanından sonra, tam bađlı katman ve son olarak Softmax sınıflandırıcı katmanı bulunmaktadır. 1000 sınıflı verileri ayırmaktadır. Diđer modellere göre farkı, önceki ağırlıklarının belli aralıklarla tekrar deđerlendirilmeye alınmasıdır. řekil 3.22'de ResNet mimarisinin klasik mimarilere göre farkı görülmektedir.

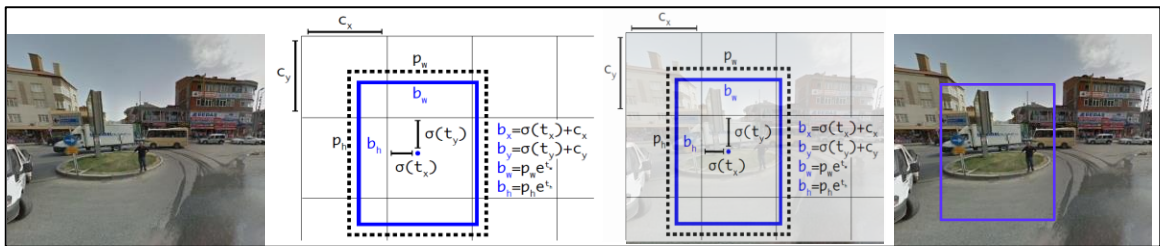


Şekil 3.22. ResNet mimarisi ile normal mimari modelleri farkı

Teorik olarak, modelde katman sayısı arttıkça başarımın artacağı düşünülür. Ancak gerçekte böyle olmadığı görülmüştür. Buradan hareketle ResNet modeli oluşturulmuştur. Böylece $w[l+2]=0$ olduğu durumda yeni teoriye göre $a[l+2]=b[l+2]$ olur. Bu (vanishing gradient) istenmeyen bir durumdur.

3.7.5. YOLO ve DarkNet

You Look Only Once (YOLO), tespit ve sınıflandırmayı içeren tek adımlı bir süreç oluşturmak için geliştirilmiştir. Sınırlama kutusu ve sınıf öngörülere, giriş görüntüsünün bir değerlendirmesinden sonra yapılır. YOLO fikri, sınırlayıcı kutu tahminleri ve sınıf tahminlerinin aynı anda yapılması bakımından diğer geleneksel sistemlerden farklıdır. Giriş görüntüsü önce $S \times S$ ızgarasına bölünür. Daha sonra, B sınırlama kutuları her bir ızgara hücresinde, her biri güven puanıyla tanımlanır. Buradaki güven, her sınırlayıcı kutuda bir nesnenin bulunma olasılığını ifade eder. Şekil 3.23'te YOLO algoritmasının görüntüde bulunan nesnelerin sınırları tespiti ölçeği görülmektedir.



Şekil 3.23. YOLO nesne sınırları tahmini

Şekil 3.23'te görüldüğü gibi görüntüde bulunan her bir nesne için özellik çıkarımı yaparak arama yapan YOLO nesnelerin sınırlarını belirler.

Darknet

Darknet, C dili ve CUDA mimarisi ile yazılmış açık kaynaklı bir sinir ağı modelidir. Hem CPU hem de GPU matematiksel hesaplamasını destekler. Darknet-53 mimarisi 52 konvolüsyon katmanından, 1'er AvgPooling, Tam Bağlı ve Softmax katmanlarından oluşmaktadır. DarkNet eğitilmiş ağ ortamında elde edilen parametreler son olarak YOLO algoritmasına sunulur ve görüntülerin sınıflandırılması sağlanır. YOLO algoritması için geliştirilen Darknet mimarisi, yüksek katmanlı bir CNN mimarisi ile hazırlanmıştır. ImageNet veritabanında bulunan görüntüler eğitimde kullanılarak bir derin öğrenme mimarisi oluşturulmuştur. Darknet mimarisi YOLO algoritmasının test amacıyla kullanılmasını sağlamak için geliştirilmiştir. YOLO'nun v1 [76] ve v2 [77] sürümleri Darknet-19'dan yapılmış 30 katmanlı bir mimarıdır ve nesne tanımayı 11 katmanda eklenerek yapmıştır. Bu mimari küçük görüntüleri tanımada yüksek başarımlı göstermezken, v3 [78] sürümü ile 53 katmanlı Darknet-53 mimarisini kullanarak daha küçük imgeleri tanımada yüksek başarı göstermiştir. YOLO V3 modelinde Darknet-53 mimarisine ek olarak 53 katmanlı ağ katmanı bulunmaktadır. Tablo 3.5'te Darknet mimarisinin katmanları ve parametreleri verilmiştir.

Tablo 3.5. Darknet-53 mimarisinin katman parametreleri

Katmanlar	Boyut	Filtre	Çıkış
Konvolüsyon	3×3	32	256×256
Konvolüsyon	3×3/2	64	128×128
Konvolüsyon	1×1	32	128×128
Konvolüsyon	3×3	64	128×128
Konvolüsyon	3×3/2	128	64×64
Konvolüsyon*2	1×1	64	64×64
Konvolüsyon*2	3×3	128	64×64
Konvolüsyon	3×3/2	256	32×32
Konvolüsyon*8	1×1	128	34×32
Konvolüsyon*8	3×1	256	32×32
Konvolüsyon	3×3/2	512	16×16
Konvolüsyon*8	1×1	256	16×16
Konvolüsyon*8	3×1	512	16×16
Konvolüsyon	3×3/2	1024	8×8
Konvolüsyon*4	1×1	512	8×8
Konvolüsyon*4	3×3	1024	8×8
AvgPooling			
Tam Bağlı	1000		
Softmax			

YOLO v3 modeli, Darknet-53 mimarisine ek olarak 53 katman ekleyerek toplamda 106 kat tamamen erişimli bir mimarisi bulunmaktadır. YOLO v3 aynı zamanda Pascal VOC veri kümesi ile eğitilmiştir. Bu sayede, gerçek zamanlı uygulamalar için hala hızlıyken, algılama algoritmasının

çoğunu geçerler. Artık bağlantılar ve örnekleme yardımıyla, yapının belirli katmanlarından 3 farklı ölçekte tespit yapılabilir. Bu, daha küçük nesnelerin tespitinde çok daha iyidir. Ancak karmaşıklığı nedeniyle önceki sürümlerden daha yavaş çalışır.



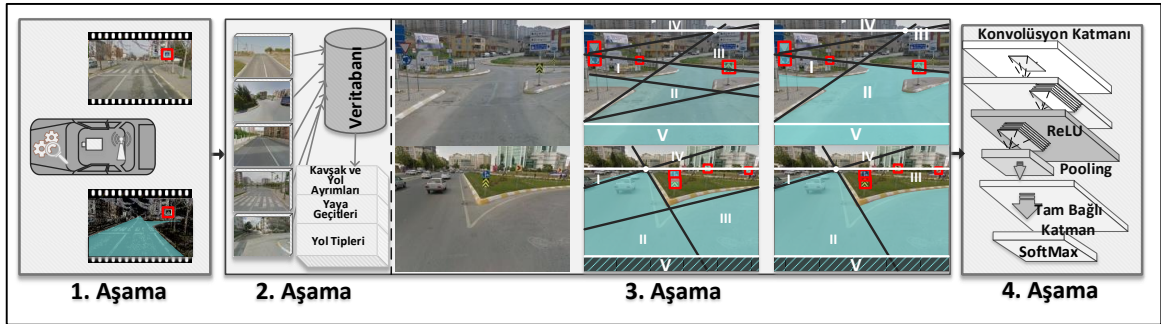
4. TRAFİKTE GÖRÜNTÜ İŞLEME VE GELİŞTİRİLEN MODEL

Bu tez çalışmasında, görüntü işleme teknikleri ve derin öğrenme metodu kullanılarak karayollarında bulunan önemli bölgelerin tespiti için bir yöntem geliştirilmiştir. Bu geliştirilen yöntemin, otonom araçlar veya sürücü destek sistemleri için bir uyarı mekanizması geliştirilmesine yardımcı olacağı düşünülmektedir. Araçlara bu sistemin yerleştirilmesi ile hem araç otonom özelliklere sahipse yönlendirme yapacak hem de ileri sürüş destek sistemleri için bir uyarı mekanizması oluşturmaya yardımcı olacaktır.

4.1. Geliştirilen Yol Tipini Tanıma Yöntemi

Çalışmada, karayollarında bulunan kavşaklar, yol ayrımları, yaya geçitleri, tehlikeli virajlar, yol tipi ve kalitesinin tespiti için bir bilgisayar görme sistemi geliştirilmiştir. Geliştirilen yöntem, otonom araç veya sürücü destek sistemleri için araçtan araca iletebilecek bilgi üretebilmektedir. Geliştirilen yöntemde, araçtan araca haberleşme yapan bu sistemlere yönelik olarak veri azaltmayı sağlayacak geliştirmeler de yapılmıştır.

Veri azaltma işlemi, tanımda kullanılan RoI bölgesinin kırılmasında gerçekleştirilmiştir. Önerilen kırpma yönteminin aynı zamanda yol tanıma gibi fazla imge içeren tanıma işlemleri için gerekli olan CNN'nin başarımını ve hızını artırdığı da gözlemlenmiştir. Yol tipinin tanınması için tasarlanan yapı prensip şema olarak Şekil 4.1'de verilmiştir.



Şekil 4.1. Geliştirilen sistemin prensip şeması

Şekil 4.1 incelendiğinde, geliştirilen yöntemin toplam 4 aşamadan oluştuğu görülmektedir. Yöntemin ilk aşamasında kullanılacak veri setleri için video ve görüntüler toplanmıştır. Bu aşamada farklı ortamlardan görüntüler elde edilmiştir. Hem kendimize ait hem de literatürde kullanılan gerçek görüntüler üzerinde çalışmalarımız yapılmıştır. Bu nedenle ilk olarak kendi veri setimizi oluşturmak için bir otomobile yerleştirilen kamera ile karayolu videosu çekilmiştir. İkinci

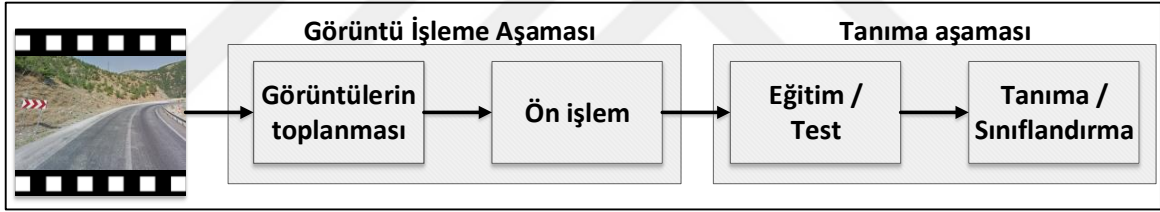
olarak Google Street View (GSV) ortamında karayolu görüntüleri toplanmıştır. Üçüncü olarak da [79] çalışmasında kullanılan video ve CMU-VASC veritabanındaki [80] görüntüler kullanılmıştır.

2. aşamada tüm görüntüler etiketlenmiş ve veri setlerine dönüştürülmüştür. Bu tez çalışmasında farklı oranlarda oluşturulmuş toplamda 6 veri seti hazırlanmıştır. Genel olarak karayolunu tanıma üzerine kurulmuş bu yöntem ile yolda bulunan kavşaklar, yol ayrımları, yaya geçitleri, tehlikeli virajlar, yol tipleri ve yol kalitesinin tespitlerindeki başarıyı görebilmek için uygun verisetler hazırlanmıştır.

3. aşamada hazırlanmış olan veri setleri, geliştirilmiş olan görüntü işleme yöntemine göre alt ve üst noktalardan kırılarak RoI alanı tespit edilmiştir. Bu yöntem tüm farklı yol tiplerine uygulanmıştır.

4. aşamada ise elde edilen tüm veri setlerinde bulunan yol tiplerine ve çalışmaya göre geliştirilmiş olan derin öğrenme mimarilerinin hem eğitimi sağlanmış hem de veri setinin kalan kısmı ile test işlemi yapılarak modellerin performansları ölçülmüştür.

Bu tez kapsamında genel olarak Şekil 4.2’de görüldüğü gibi iki aşamalı bir tanıma sınıflandırma çalışması yapılmıştır.



Şekil 4.2. Geliştirilen sistemin işlem aşamaları

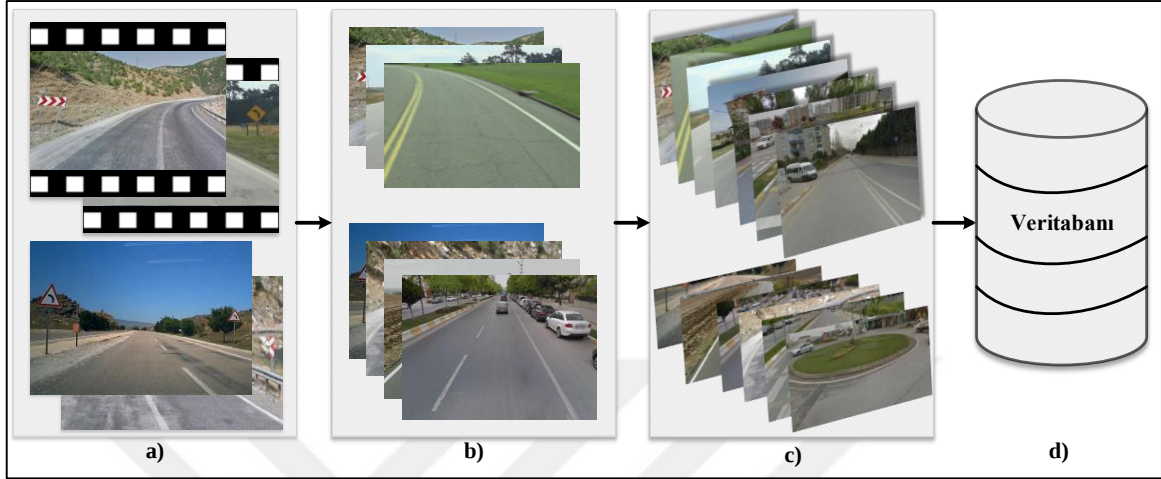
Şekil 4.2’de görüldüğü gibi görüntü işleme aşamasında verilerin video ve görüntülerden elde edilmesi ve ön işlemlerde değerlendirilmesi yapılmaktadır. Tanıma aşamasında ise, görüntüler eğitim ve test edildikten sonra tanıma ve sınıflandırma yapılmaktadır.

4.2. Geliştirilen Yöntemin Görüntü İşleme Aşaması

Görüntü işleme teknikleri, görüntüyü sayısal forma dönüştürmek ve görüntüde istenen parametrelerin belirlenmesi için sayısal formun farklı algoritmalar ile değişimini sağlayarak çıkarım yapmaktadır. Dönüştürülmüş verilerden spesifik görüntü elde etmek veya istenen bilgiler ortaya çıkarmak için ihtiyaca göre farklı görüntü işleme teknikleri kullanılır. Bu yöntemin giriş katmanı video veya imgelerdir. Çıktı katmanı ise, görüntünün istenilen ya da dikkat edilmesi gereken alanıdır.

Günümüzde geliştirilen otonom sistem teknolojileri ve sürücü destek sistemleri çoğunlukla bilgisayar görü sistemi kullanılarak yapılmaktadır. Görü temelli sistemlerde makine algılaması ve

öğrenmesi üzerinde çalışmalar mevcuttur. Makinenin farklı yöntemlerle öğrenmesi ve görüntüleme teknolojileri kullanılarak karar vermesi sağlanmaktadır. Görüntülerin veritabanına eklenmesi aşamaları Şekil 4.3'te görülmektedir.



Şekil 4.3. Görüntülerin veritabanına eklenmesi aşamaları

Şekil 4.3'te video ve kamera görüntülerinden elde edilen verilerin veritabanlarına dönüşüm aşamaları görülmektedir. Şekil 4.3(a)'da tüm görüntüler RGB kanallı video ve kameralardan toplanmıştır. Şekil 4.3(b)'de video verileri işlemek için 25fps ve kendi çözünürlükleri ile görüntülere dönüştürülmüş ve görüntü temelli bir veritabanı oluşturulmuştur. Şekil 4.3(c)'de ise görüntüler veritabanına eklenmeden önce boyutlandırma yapılmıştır. Şekil 4.3(d)'de ise tüm veriler belli oranlarda veritabanına dönüştürülmüştür. Bu aşamalardan sonra elde edilen veritabanında bulunan imgelerin içinde önemli alanların belirlenmesi için geliştirmiş olduğumuz yöntem kullanılmıştır.

4.2.1. İlgiilenilen Bölgenin Belirlenmesi

Karayolunda tespit etmeye çalıştığımız yol tipi ve özelliklerini bulmak için görüntülerin istenen bölgelerine odaklanmamız gerekmektedir. Bu nedenle görüntülerde bulunan ve tanımda bir öneme sahip olmayan, aynı zamanda tanıma işleminde performansı olumsuz yönde etkilemeyen kamera görüntülerinin üst ve altında belirlenecek bölgelerin kırılması gerekmektedir.

Karayolunda özel alan çıkarma, otonom sürüş, ileri sürücü destek sistemleri, kavşak tespiti, yol tipleri tespiti ve yaya geçidi tespiti gibi bilgisayarlı görmenin farklı alanlarında önemli bir rol oynar. Yol tespit algoritmasının sürekli değişen arka planları, farklı ortamlar (kentsel-kırsal yollar, rampa yollar, arazi dışı yollar gibi), farklı yol türleri (şekil, renk) ve farklı görüntüleme koşullarında

(değişken aydınlatma, farklı bakış açıları ve değişen hava koşulları) çalıştırmak oldukça zahmetlidir.

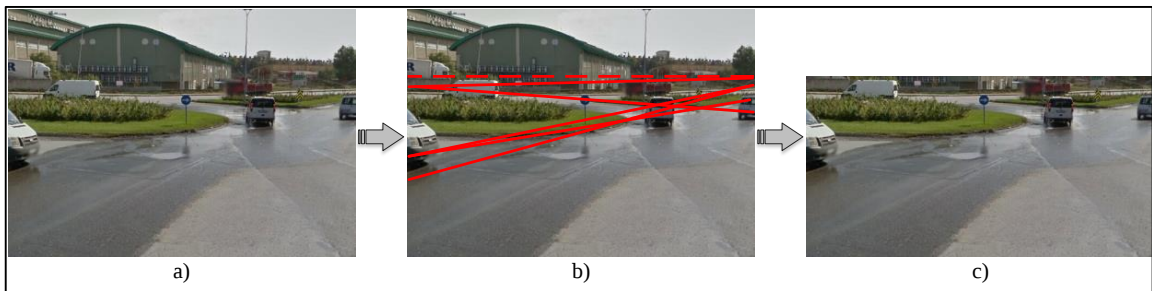
Karayolu görüntülerinden ilgili yol alanının çıkarımı, ileri sürüş destek sistemleri ve otonom araç sistemlerinin güvenliği için oldukça önemlidir. İlgili alanın tespiti ile araç rahatlıkla yönlendirme yapılabilir ve aynı zamanda ilgili bölgenin yüksek başarımla tanınması da sağlanmış olur. RoI ile görüntülerde gerekli olan ilgili maksimum alan tespit edilmiş olur. RoI'nin tespiti için [9, 81-85] çalışmalarında önerilen ve farklı bir parametre ile güncellenen karayolu üst ufuk noktası - Vanishing Point (VP) yöntemi ve görüntülerin alt kesim noktaların belirlenmesi için bir model geliştirilmiştir.

Karayollarında üst ufuk noktasının tespiti

Karayollarında bulunan üst noktaların tespiti için [8, 9, 81-83, 85, 86] çalışmalarında önerilen yöntem baz alınmış ve geliştirilmesi ile yeni bir model önerilmiştir.

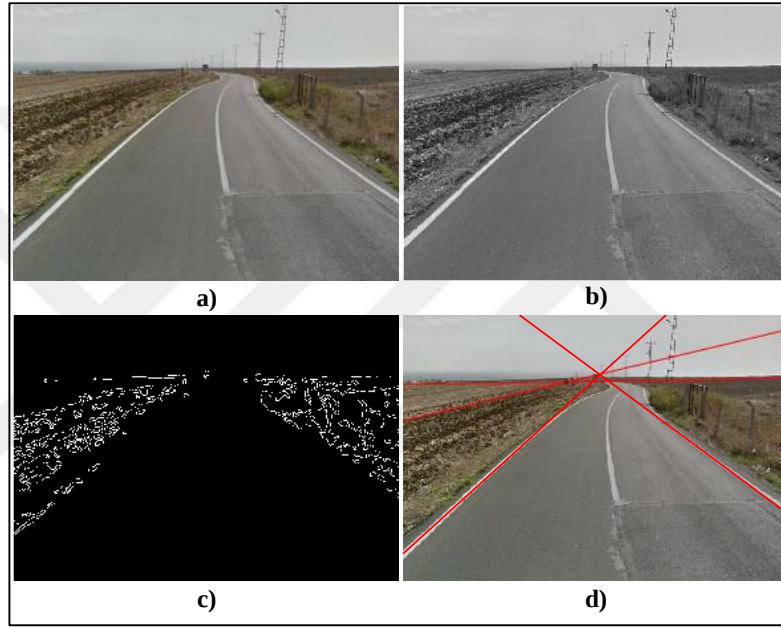
Otoyollar, yapısal yollar ve yapılandırılmamış yollar için sayısız yol çıkarma yöntemleri yaygın bir şekilde uygulanmıştır. Tüm bu çalışmalarda, VP yerini tahmin etmek kilit bir görevdir. Bir yolda bulunacak VP noktası eğer doğru bir şekilde yerleştirilmişse, yol alanını doğru tespit etmek daha kolaydır. VP tabanlı yol algılama yöntemleri temel olarak üç kategoride gruplandırılabilir: Kenar tabanlı yöntemler, önceki tabanlı yöntemler ve doku tabanlı yöntemler. Çoğu kenar tabanlı yöntemler, düzgün asfaltlanmış, yol çizgileri belirlenmiş ve aynı zamanda işaretlenmiş düz yollarda uygulanarak yüksek başarımlar elde edilmektedir. Yol çizgileri belli olmayan, bakımsız kalmış asfalt ya da iyi döşenmiş olmayan yollarda veya kavşak-kavis, yaya geçitleri gibi önemli alanları barındıran yollarda bu yöntem başarısız olmaktadır [81-85].

Ön işlem aşamasında, imge içerisinden tanımayı sağlayacak ilgili bölgenin (RoI) kırılması yapılmıştır. Kırılma adımında, ayırt edici yol bilgisi içermeyen üst ve alt kısımların çıkarılması yapılmıştır. İmgelerin üst bölgesinin kırılma işlemi, literatürde önerilen VP [8, 9, 81-83, 85, 86] yöntemine göre yapılmış ve uygun ölçülerdeki veri setlerine dönüştürülmüştür. Şekil 4.4'te örnek bir karayolu görüntüsünün VP metoduna göre üst bitim noktasının tespiti ve kırılması görülmektedir.



Şekil 4.4. VP metoduna göre karayolu görüntülerinin üst bitim noktasının tespiti ve kırılması

Şekil 4.4.'te görüldüğü gibi ham görüntü üzerinden VP metodu ile üst bitim noktası tespit edilmiş ve ardından en üst bitim noktasından kırpılma işlemi yapılmıştır. Şekil 4.4(a)'da ham görüntü görülmekte, Şekil 4.4(b)'de VP metoduna göre üst bitim noktaları ve kenar çıkarımları tespit edilmektedir. Şekil 4.4(c)'de ise tespit edilen VP noktasına göre kırpılma işlemi yapılarak işlenecek görüntü elde edilmiştir. Bu işlem sonucunda üst ilgisiz bölge atılmış ve odaklanılan alan daraltılmıştır. Veritabanındaki aynı görüntü Canny kenar tespit algoritmasından geçirilerek karayolunda bulunan kenarlar tespit edilmiştir. Şekil 4.5'te karayolu görüntülerinin Canny kenar tespit algoritmasında yapılan kenar tespitleri görülmektedir.



Şekil 4.5. Görüntülerde Canny kenar çıkarımı ve üst bitim noktasının tespit edilmesi

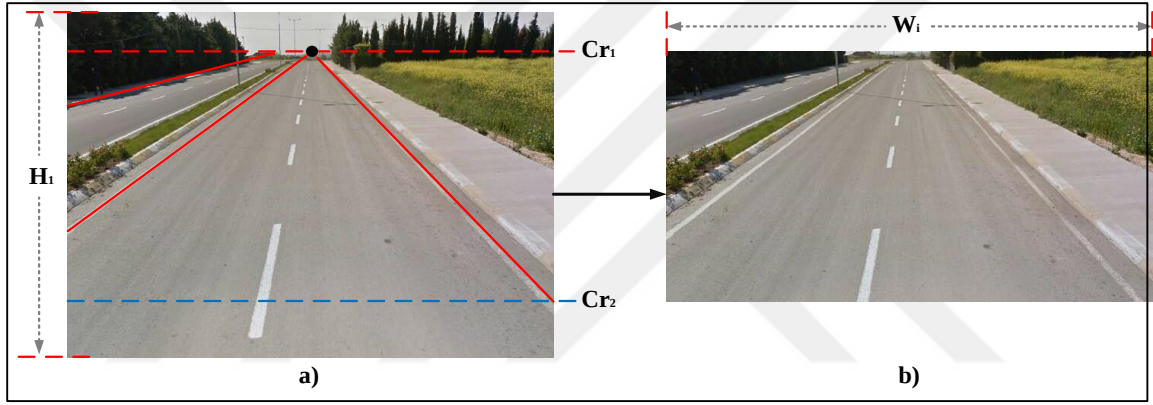
Şekil 4.5 incelendiğinde, görüntülerin 256 bit gri formata dönüşümü Şekil 4.5(b) görülmektedir. Şekil 4.5(c)'de Canny kenar tespit algoritması kullanılarak şerit tespiti yapılmıştır. Böylece VP yöntemi ve Canny kenar tespit algoritması kullanılarak karayolları görüntülerinde yol işaretleri ve ufuk noktaları tespit edilmiştir.

Çoğunlukla derin öğrenme mimarileri, standart ve sabit boyutlu görüntüler üzerinde tanıma işlemi yapmaktadır. Görüntüler üzerinde boyutlandırma işlemi yapılabilir, fakat geliştirilen modelin gerçek görüntülerin işlenmesiyle daha yüksek performans gösterdiği tespit edilmiştir. Bu nedenle alt gereksiz bölgenin de kırpılması gerekmektedir. VP metoduyla tespit edilen üst noktaya göre, imgenin boyutu dikkate alınarak görüntünün yüksekliği, genişliğinin yarısı olacak şekilde kırpılması yapılmıştır. Denklem 4.1 ve 4.2'de görüntülerin tespit edilen VP noktasına göre alt kırpm noktasının belirlenmesi ifade edilmiştir.

$$Cr_1 = Vp \quad (4.1)$$

$$Cr_2 = \begin{cases} H_i; & H_i - Cr_1 \leq \frac{W_i}{2} \\ H_i - (\frac{W_i}{2} - Cr_1); & H_i - Cr_1 > \frac{W_i}{2} \end{cases} \quad (4.2)$$

Burada, Cr_1 yolun VP bölgesinin üst kesim noktasını, Cr_2 ise yol görüntüsünün alt kesim noktasını temsil etmektedir. H_i imgenin yüksekliğini, W_i ise imgenin genişliğini temsil etmektedir. Yeni oluşan bu noktalar imgenin RoI alanını belirler ve tespit edilen bu noktalardan imgeler kırpılarak veri seti oluşur. Şekil 4.6'da imgelerin önerilen önileşim yöntemine göre kırılması gösterilmiştir.

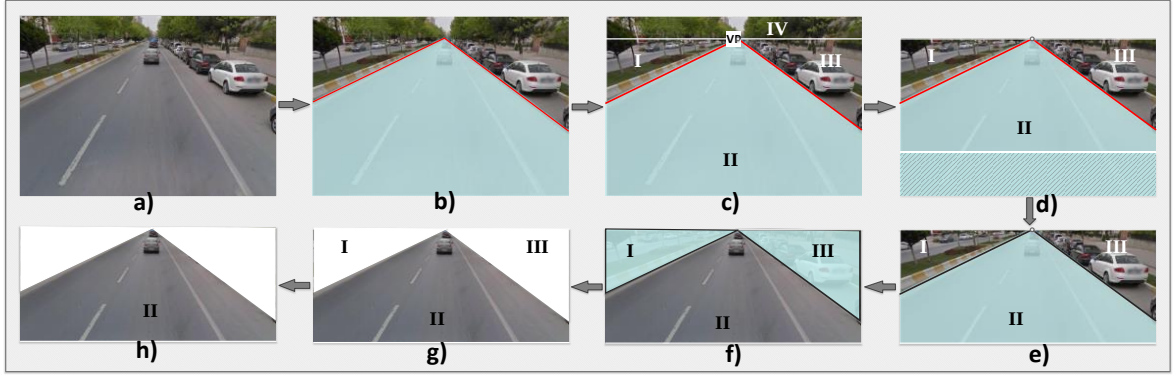


Şekil 4.6. Alt kesim noktasının tespit edilmesi

Şekil 4.6(a)'da VP noktası baz alınarak görüntünün alt kesim noktası tespit edilmiş ve bu noktadan görüntünün alt düzlemine göre paralel olacak şekilde kesim yapılarak Şekil 4.6(b)'deki görüntü elde edilmiştir.

Görüntü işleme aşamasında yol ve yol sınırları belirlendikten sonra, imge dört bölgeye ayrılarak analiz yapılmaktadır. Sola ve sağa dönüş yolları ile sol-sağ kavşaklar ve sol ve sağ tehlikeli virajların tespitinin daha yüksek başarımla yapılması için, imgeler yol sınırlarının VP noktası hedef alınarak belirlenmiştir.

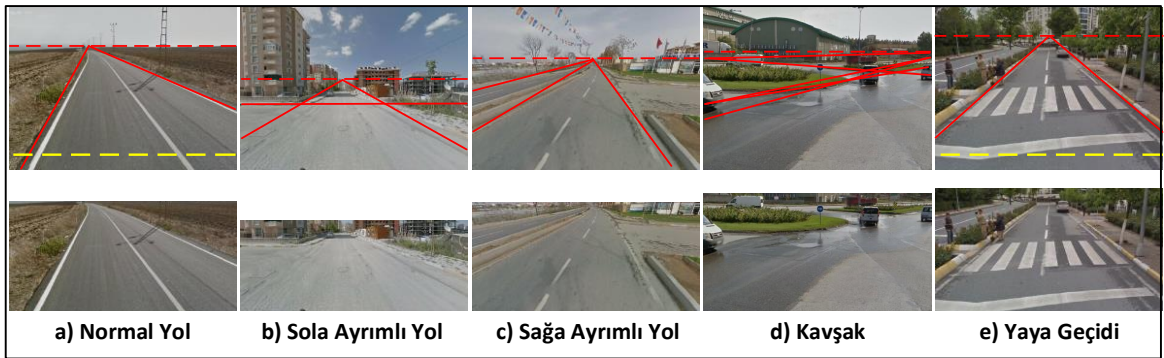
Şekil 4.7'de yol görüntüsünün dört bölgeye ayrılmasına örnek verilmiştir. Şekil 4.7(a)'da alınan ham imge ve Şekil 4.7(b)'de imge üzerinde yol bölgesinin tespiti görülmektedir. Şekil 4.7(c)'de ise, yol görüntüsünün dört bölgeye ayırma yöntemi görülmektedir. Şekil 4.7(d)'de ise yol bilgisi taşımayan üst alanın kırılmış hali ve yakın bölge bilgisi taşımayan alan işaretlemesi görülmektedir. Şekil 4.7(e)'de alt yakın bölge bilgisi taşımayan alanın kırılmış hali verilmiştir. Şekil 4.7(f, g, h)'de ise bölgeleri tamamen birbirinden ayırma aşamaları görülmektedir.



Şekil 4.7. Karayolu ve ilgisiz bölgelerin ufuk noktası kullanılarak bölgelere ayrılması

Şekil 4.7’deki görüntü işleme yöntemi kullanarak veritabanında bulunan tüm görüntüler önışlemden geçirilmiştir. Sayısal RGB formatındaki görüntü ve sınır belirlenmesi için imgede ufuk noktası VP ve kenar çıkarım algoritması kullanılarak bölgelere ayırım yapılmıştır. İlgisiz bölge olarak üst VP noktasından görüntünün tepe noktasına kadar olan alan belirlenmiştir. Daha sonra alt ilgisiz bölge tespiti için ise, görüntünün üst ufuk noktasından görüntünün alt noktasına kadar olan kısma oransal yaklaşarak Denklem 4.1 ve 4.2’ye göre hesaplama yapılmıştır. Hesaplama sonucunda yakın bölgeden çıkarılması gereken ilgisiz bölge tespit edilmiştir.

Önerilen bu kırpma yöntemi, elde edilen küçük boyutlu imgenin, araç ağı üzerinden diğer araçlara iletilmesi istendiğinde, V2V ağındaki veri trafiğini oldukça azaltacaktır. Aynı zamanda daha küçük bir imge üzerinde tanıma işlemi yapıldığından başarı oranı ciddi oranda artacaktır. VP noktasının tespiti ve ardından oluşan Şekil 4.8’de veritabanında bulunan imgelerin RoI alanının tespiti için yapılmış olan önışlem sonucunda oluşan örnek görüntüler verilmiştir.



Şekil 4.8. Geliştirilen yöntemin karayolu tiplerine göre tespit edilen RoI bölgeleri

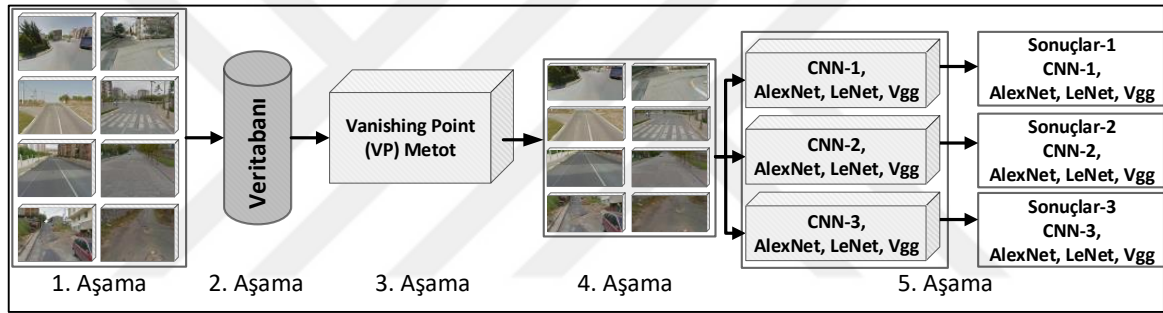
Şekil 4.8’de farklı türde karayolu görüntüleri görülmektedir. Görüntülerde normal yol, sol ve sağ yol ayrımları, kavşak ve yaya geçitleri görüntüleri bulunmaktadır. Şekil 4.8’de üst satırda bulunan ham görüntüler, 458×640 piksel ölçülere ve aynı zamanda RGB kanallı iki boyutlu

özelliklere sahiptir. İkinci satırda ise, ham görüntülerin önışlem aşamasından geçirildikten sonra oluşan kırılmış imgeler görölmektedir.

Bu tez çalışmamızda görüntülerdeki RoI alanın belirlmesine ek olarak, eş zamanlı çalışabilen karayollarının kenarlarında bulunan trafik işaretları de tespit edilmiştir. Böylece, gerçek yol durumu ile özellikle kavşak, yol ayrımı, yaya geçidi ve tehlikeli viraj işaretlarının eşleşip eşleşmediğı analizi yapılarak bir denetleme yöntemi de geliştirilmiştir.

Yapılan deneysel çalışmalarda toplamda 6 farklı yöntem gerçekleştirilmiştir. Her bir yöntem için farklı veri setleri oluşturulmuştur.

Karayolunun gerçek yapısının belirlenmesi ve ölkemizdeki karayollarının bir kısmının analizi için, kendi veri setimiz oluşturulmuş, aynı zamanda GSV ortamı ve daha önce yapılan literatürdeki çalışmalarda kullanılan veriler de değerlendirilmek üzere veri setlerine dönüştürölmüştür. Şekil 4.9’da oluşturulan tanıma sisteminin genel blok diyagramı verilmiştir.



Şekil 4.9. Çalışmada karayolu bileşeni tanıma için geliştirilen yöntemin blok şeması

Yöntemin ilk aşaması, probleme özgü imgelerin toplanıp etiketlenmesidir. Bu aşamada, incelenecek görüntüler kendi veritabanımız, GSV ortamından elde edilen veri tabanı ve [79] çalışmasında incelenen video verilerinin Şekil 4.9’da belirtilen yöntemden geçirilerek tanıma işlemi yapılmıştır.

İlk aşamada veriler toplanmış ve daha sonra veritabanına eklenmiştir. Şekil 4.9’un 3. aşamasında, tüm görüntüler geliştirilen VP metodundan geçirilmiş ve imgelerin RoI bölgeleri tespit edilmiştir. 4. aşamada, imgeler eğitim, doğrulama ve test olmak üzere derin öğrenmede sınıflandırma ile tanımaya yönelik olarak üç kategoriye tasnif edilmiştir. 5. Aşamada ise, görüntü türlerine göre tanınması için geliştirilen ve literatürde var olan CNN yöntemleri ve sonuçları görölmektedir.

Çalışmanın yerel şartlarda başarı durumunu değerlendirmek amacıyla, yol görüntüleri kayıt edilerek geliştirilen algoritmalar ile bu video üzerinde başarı durumu değerlendirilmiştir.



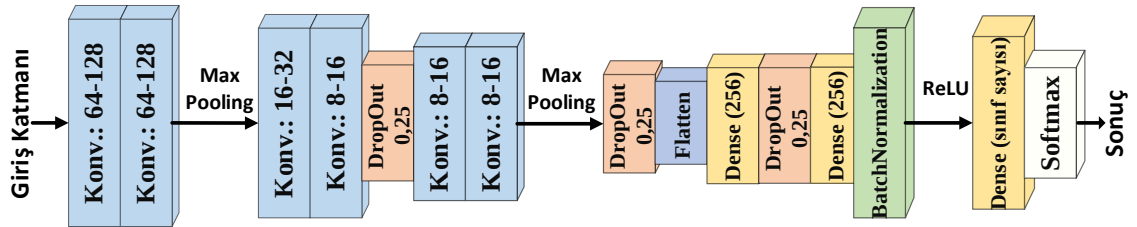
Şekil 4.10. Çalışmalar için kullanılan görüntülerin örnekleri

Şekil 4.10’da görüldüğü gibi soldaki 4’lü grup görüntüler özel bir araca yerleştirilen kamera ile elde edilen videolardan alınmıştır. Sağdaki 4’lü görüntüler ise [79] çalışmasının trafik işaretlerini tanımda kullanmış olduğu videolardan alınmıştır. Orta kısımda bulunan 4’lü görüntüler ise GSV ortamından elde edilmiştir.

Videolar, araç karayolunda hareket halinde iken 1 dakikalık süreyle çekimler yapılarak elde edilmiştir. Saniyede 25 çerçeve (25 fps) alınarak çekilen videolar daha sonra imgelere dönüştürülmüştür. Her saniye için elde edilen 25 kare görüntülerden öznitelik değerleri arasında belli bir fark olanlar çıkartılmış ve elde edilen görüntüler ile sistemin eğitim ve testi için belli oranlarda veri kümelerine dönüştürülmüştür.

4.3. Geliştirilen Yöntemin Tanıma Aşaması

Görüntüler farklı aşamalardan geçirildikten sonra hazırlanan veri setlerin yüksek başarımlı tanıma işlemi için, 15 katmanlı CNN tabanlı bir ağ modeli önerilmiştir. Bu modelin tüm görüntülerinin farklı gruplar altında sınıflandırması amaçlanmıştır. Bu çalışmada dört farklı veri set oluşturulmuş ve bu veri setlerin sınıflandırılması için birçok CNN modeli denenmiş, fakat en iyi sonuçların Şekil 5’te görülen 15 katmanlı CNN modelinden elde edildiği tespit edilmiştir.



Şekil 4.11. Çalışmalar için kullanılan görüntülerin örnekleri

Bu tez çalışmada genel olarak Şekil 4.11’deki modelden temel alınmış fakat her bir uygulama için hazırlanan model detaylı bir şekilde tüm parametreleri ile ilgili bölümlerde verilmiştir.

4.4. Eğitim, Doğrulama ve Test İşlemleri

Uygulamalar için hazırlanan veri seti üzerinde eğitim, doğrulama ve test işlemi yapılabilir. Bu amaçla veri setinin belli bir kısmı modelin eğitimi için ayrılır ve model eğitilir. Bu eğitim sonucunda oluşan ağırlık parametreleri ve eğitilmiş model doğrulama yapılmak üzere kullanılır. Denetimli model olarak isimlendirilen bu yöntem ile veri setinin geri kalan kısmı üzerinde doğrulama yapılır. Doğrulama sonucunda oluşan grafik test başarısı olarak isimlendirilir. Bu yöntem dışında test işlemi farklı şekilde de yapılabilir. Veri setinin eğitim kısmı için ayrılan bölüm değiştirilerek tüm veri seti test edilebilir. Çapraz doğrulama (Cross validation) olarak adlandırılan bu yöntem ile tüm test sonuçlarının ortalaması alınarak ortalama başarı elde edilmektedir.

Bu tez çalışmasında, kabul edilebilir ölçümler elde etmek için hazırlanan veritabanında bulunan etiketlenmiş ve önışlemden geçirilmiş tüm görüntüler, farklı oranlarda rastgele veri setlerine ayrılmıştır. Deneysel çalışmamızın eğitim, doğrulama ve test aşamaları için farklı oranlarda veri setleri hazırlanmıştır. Bu tez için geliştirilmiş uygulamaların veri sayıları ve nitelikleri farklı olduğu için hazırlanan veri setlerinin eğitim, doğrulama ve test aşamalarında veri sayılarının oranları değişmektedir. Oluşan başarımları ölçmek için farklı değerlendirme kriterleri belirlenmiştir.

4.4.1. Değerlendirme Kriterleri

Eğitilmiş CNN modelinin test aşamasında elde edilen sonuçların değerlendirilmesi için farklı kriterler altındaki sonuçları ölçülmelidir. Bu amaçla, test verileri için çeşitli değerlendirme kriterleri seçilmiştir. Bu kriterler Doğruluk, Kesinlik, Anma ve F1-skorudur. Bu hesaplamalarda kullanılan terimler Doğru(True)-Pozitif (TP), Yanlış(False)-Pozitif (FP), Doğru(True)-Negatif (TN) ve Yanlış(False)-Negatif (FN)'dir.

Doğruluk: Doğru tahmin edilen görüntülerin tüm görüntülere oranıdır ve Denklem 4.3'te gösterilmiştir.

$$Doğruluk(\%) = \frac{TP + TN}{TP + FP + TN + FN} \times 100 \quad (4.3)$$

Kesinlik: Doğru tahmin edilen pozitif görüntülerin tahmini toplam pozitif görüntülere oranıdır. Hesaplama Denklem 4.4'te gösterilmiştir.

$$Kesinlik(\%) = \frac{TP}{TP + FP} \times 100 \quad (4.4)$$

Anma: Bir sınıfta sınıflandırılan görüntülerin, o sınıftaki tüm görüntülere oranıdır ve hesaplama Denklem 4.5'te gösterilmiştir.

$$Anma(\%) = \frac{TP}{TP + FN} \times 100 \quad (4.5)$$

F1 - Skor: Denklem 4.6'da gösterildiği gibi kesinlik ve anma değerlerinin ağırlıklı ortalamasıdır.

$$F1 - Skor = \frac{(Anma) \times (Kesinlik) \times 2}{(Anma) + (Kesinlik)} \quad (4.6)$$

4.4.2. Tanıma ve Sınıflandırma

Test verilerinin eğitilmiş model üzerinden tanıma veya sınıflandırma kullanılarak değerlendirilmesi ile başarı sonuçları elde edilmiştir. Eğitim aşaması tamamlanan modeller yine aynı tür veriler ile doğrulanmış ve başarımları elde edilmiştir. Test işlemlerinde yapılmış olan çalışma literatürdeki benzer çalışmalar ile karşılaştırılmış, ayrıca aynı tür derin öğrenme mimarileri ile elde edilen sonuçlar ilgili tablolarda gösterilmiştir.

5. UYGULAMA

Çalışmalarımızda, otonom araç ve sürücü destek sistemleri için de tanınması gerekli, karayolunun önemli bileşenleri olan kavşaklar, yol ayrımları (sol-sağ kavşak), yaya geçitleri, tehlikeli virajlar, yol tipleri tespiti ve genel olarak yol kalitesinin imge işleme teknikleri ile belirlenmesi üzerine yoğunlaşmıştır. Tespit ve belirleme için görüntü tanıma işlemlerinde oldukça başarılı sonuçlar vermesi nedeniyle yaygınlaşmaya başlayan derin öğrenme yöntemlerine odaklanılmıştır.

Ayrıca, tez çalışmamızda bu belirlemelerle eş zamanlı çalışabilen, karayollarının kenarlarında bulunan trafik işaretleri de tespit edilmiştir. Böylece, gerçek yol durumu ile özellikle kavşak, yol ayrımı, yaya geçidi ve tehlikeli viraj işaretlerinin eşleşip eşleşmediğinin analizi yapılarak bir denetleme yöntemi de geliştirilmiştir.

Çalışmada toplamda 6 farklı tespit uygulaması yapılmıştır. Bu uygulamalarda farklı ortamlardan elde edilen görüntüler kullanılmıştır. Tablo 5.1’de uygulamalar için hazırlanan veri setleri görülmektedir.

Tablo 5.1. Hazırlanan veri setleri

Uygulama adı	Veri setleri	Sınıf sayısı	Görüntü sayısı
Yol tipi tespiti	Veri seti-1	5	640
Yol kalitesi tespiti	Veri seti-2	2	277
Kavşak tespiti	Veri seti-3	2	293
Sağ ve sol yol ayrımlarının tespiti	Veri seti-4	3	449
Yaya geçitlerinin tespiti	Veri seti-5	2	296
Sağ ve sol tehlikeli virajların tespiti	Veri seti-6	3	587

Tablo 5.1 incelendiğinde, 6 farklı çalışma için farklı veri sayısına sahip veri setleri hazırlandığı görülmektedir. Bu veri setlerinin bir kısmı eğitim, bir kısmı da doğrulama amacıyla kullanılmıştır. Görüntü sayıları ve görüntülerin özelliklerine göre veri setlerinin eğitim, doğrulama ve test olarak bölündüğü çalışmalar bulunmaktadır. Tüm bu çalışmalar için farklı CNN modelleri geliştirilmiş ve test işlemi bu modeller kullanılarak yapılmıştır.

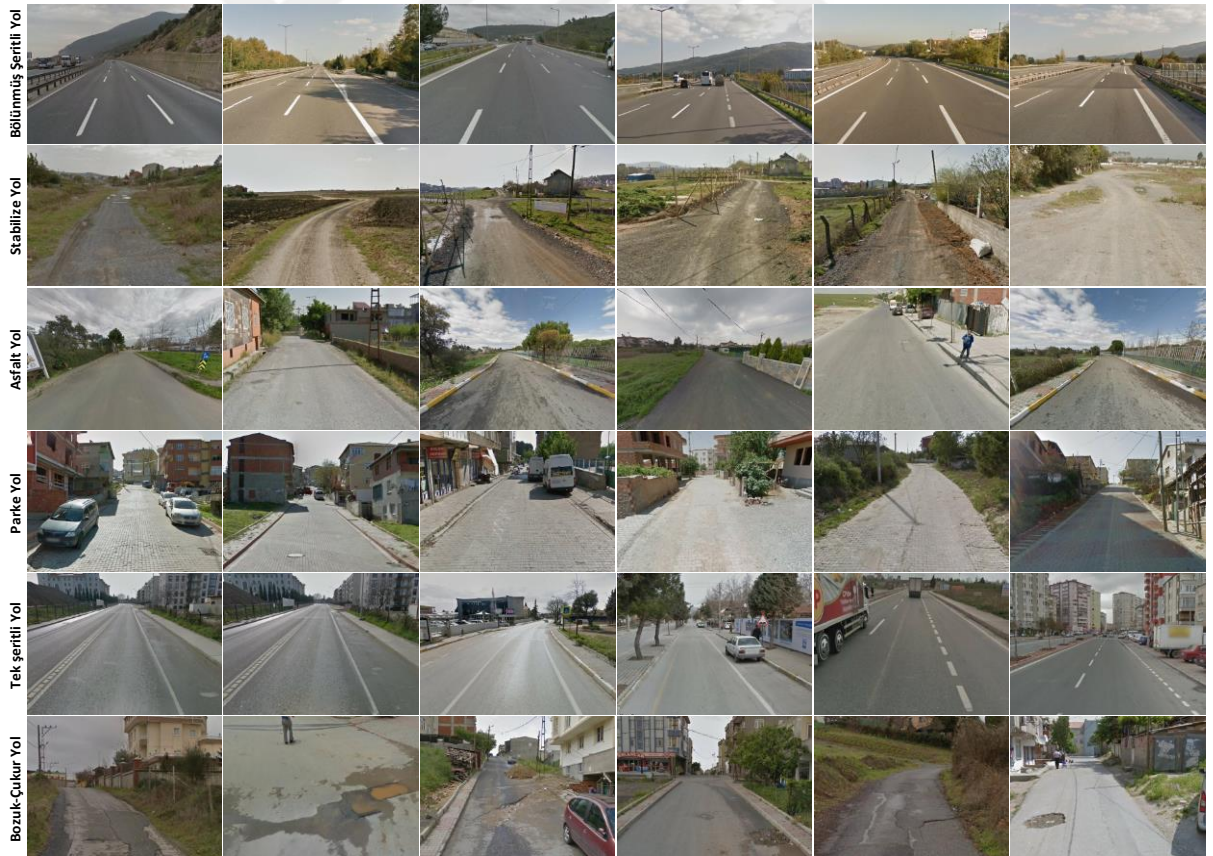
5.1. Yol Tipi ve Yol Kalitesinin Tespiti

Karayollarının özelliklerinin tespiti için hazırlanan tez çalışmasının bu bölümünde yol tipi ve kalitesinin tespiti yapılmıştır. Bu bölümde temel olarak iki ayrı deneysel çalışma yapılmıştır. Bunlar öncelikle yol tipinin otomatik olarak sınıflandırılmasıdır. Karayolu tipinin belirlenmesi, araçların hareket yöntemine karar vermede önemli bir parametre olarak değerlendirilir [83, 84].

Yol tipleri, engebeli yüzey, yolun genişliği, çevresindeki yapılar ve farklı fiziksel özelliklere sahiptir. Yerleşik bir kamera ile yol tipi sınıflandırma güvenliği, yakıt verimliliği ve yolcu konforu

hakkında önemli bilgiler sağlanabilir. Farklı karayolu tahminleri, otonom araçta kullanılacak tehlike önleme algoritmalarında güvenli gezinme rutinlerinin sınırlandırılmasında da faydalı olabilir. Yol tipinin veya yol kalitesinin (bozuk-delik yollar) tespiti, araçta motoru, hızı ayarlamak ve kullanıcı ara yüzünü duruma göre uyarmak gibi birçok kullanım alanına sahiptir. Ayrıca, mevcut yol türü ve çevresi, sürücünün karar almasına yardımcı bulunacaktır.

Karayollarının tiplerine göre hız ayarlaması yapılabilir ya da aracın diğer özellikleri için karar verilebilir. Yol tipi ve kalitesine göre aracın manevrası, hızı ve yavaşlaması dahil birçok şey değişmektedir. Örneğin; aracın otobanda hız sınırı yüksek iken, asfaltsız yollarda bu hızını düşürmesi gerekmektedir. Aynı zamanda aracın çukur, bozuk yollarda daha yavaş ve dikkatli gitmesi gerekirken asfaltlı, düzgün bir yolda araç hızını artırabilir. Karayolunun tip ve kalite tespiti, bu tez çalışmasının önemli bir uygulaması olarak değerlendirilmektedir. Bu deneysel çalışmada altı sınıfa (bölünmüş yol, otoban, asfaltlı sokak yolları, asfaltsız yollar, parke taşı döşeli yollar ve bozuk-çukur yollar) ait görüntü verileri kullanılmıştır. Şekil 5.1’de altı sınıfa ait yol tip görüntüleri görülmektedir.



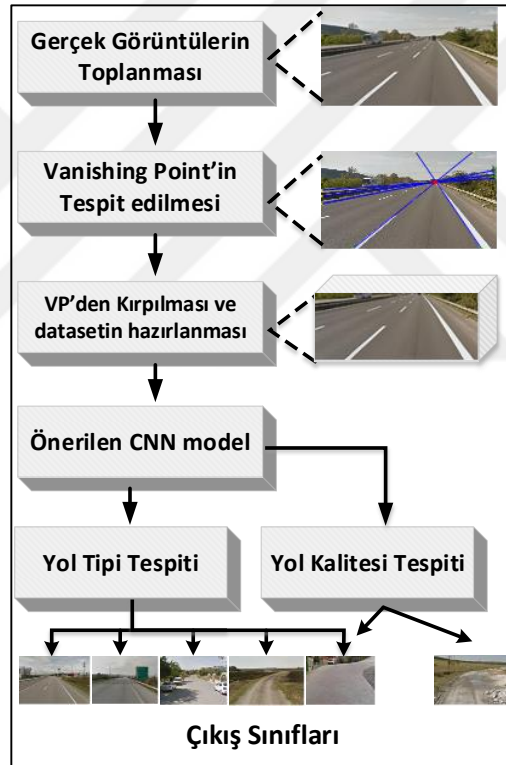
Şekil 5.1. Yol tipi ve kalitesi için hazırlanan görüntüler

Şekil 5.1’de görüldüğü gibi karayolları; bölünmüş yollar, otoban, asfaltlı sokak yolları, asfaltsız sokak yolları, parke döşeli yollar ve bozuk yollardan oluşmaktadır. Bu deneysel çalışma

için iki farklı veri seti (Veri seti-1, Veri seti-2) hazırlanmıştır. Veri seti-1 her sınıfta farklı sayıda toplanarak oluşturulmuş 640 yol görüntülerinden oluşmaktadır. Veri seti-2 ise bozuk yol görüntüleri (asfalt ve stabilize yol) ve diğer bozuk olmayan yol tiplerinin bir araya getirilmesi ile toplamda 277 (139 bozuk-çukur yol, 138 normal yol) görüntülerinden oluşturulmuştur.

5.1.1. Yol Tipi ve Yol Kalitesinin Tespiti için Önerilen Model

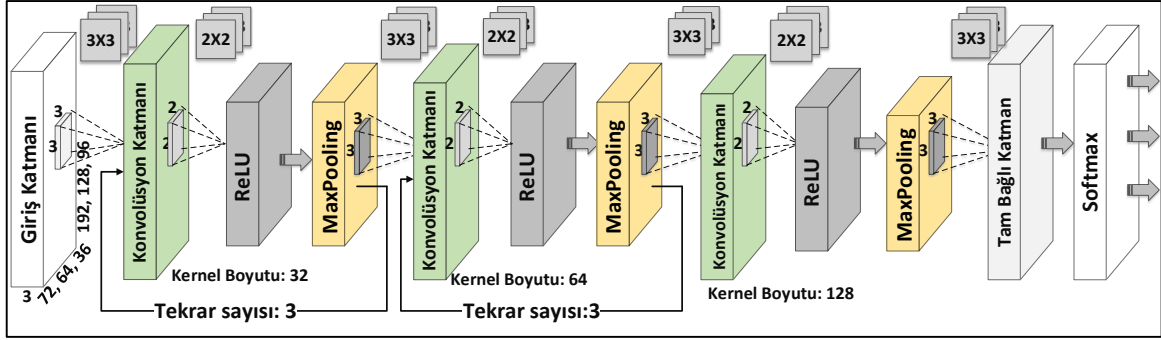
Bu çalışmada, yol tipi ve kalitesinin sınıflandırılması için bir yöntem geliştirilmiştir. Bu yöntemde görüntülerin elde edilmesi, görüntü işleme teknikleri ile istenmeyen bölgelerin kırılması ve veri setlerine dönüştürüldükten sonra önerilen modelden geçirilerek sınıflandırılması yapılmıştır. Geliştirilen yöntemin prensip şeması Şekil 5.2’de görüldüğü gibidir.



Şekil 5.2. Yol tipi ve kalitesinin tespiti için önerilen yöntemin prensip şeması

Şekil 5.2 incelendiğinde, elde edilen kamera görüntüleri yol ile birlikte birçok veriyi içermektedir. Sadece yol verisinin elde edilmesi amacı ile görüntüler üzerinde bir ön işlem yapılmıştır. Bu amaçla VP yöntemi giriş görüntülerine uygulanarak üst kesim noktaları tespit edilmiştir. Daha sonra bu kesim noktaları referans alınarak görüntü üzerinde kırma işlemi gerçekleştirilmiştir ve elde edilen görüntüler veri setlerine dönüştürülmüştür. Bu işlemlerden sonra yol tipi ve yol kalitesi şeklinde iki veri seti hazırlanmıştır. Yol tipi veri setinde 5 farklı yol görüntüleri bulunurken, yol kalitesi veri setinde ise 2 farklı yol görüntüleri bulunmaktadır. Veri

setlerin %80'i önerilen CNN tabanlı modelin eğitimi ve geriye kalan %20'si de doğrulama için kullanılmıştır. Yol tipi ve kalitesinin tespitinin test edilmesi için geliştirilen model Şekil 5.3'te verilmiştir.



Şekil 5.3. Yol tipi ve kalitesinin tespiti için önerilen model

Tasarlanan CNN modeli, toplamda 7'şer konvolüsyon, pooling ve ReLU katmanı içermektedir. Modelimiz Konvolüsyon, Pooling ve ReLU katmanlarının art arda üçer defa tekrar eden 2 blok ve son olarak kernel sayısının 128 olduğu son bloktan oluşmaktadır. Konvolüsyon katmanı ile elde edilen parametrelerin ReLU katmanı ile sonuçları alınmış ve negatif değerler 0'a çekilerek veri azaltılması ve tutarsız niteliklerin yok edilmesi sağlanmıştır. MaxPooling katmanı ile ReLU katmanında elde edilen verilerden matris olarak en yüksek anlamlı parametrelerin seçilmesi hedeflenmiştir. Veri boyutundan kaynaklı olarak bu birlikteliğin 3 defa tekrar edilmesi ile en anlamlı parametreler tespit edilmiştir. İlk konvolüsyon katmanı 3x3 boyutlu toplam 32 kernel görüntü verisi üzerinde 2 kaydırma adımı (stride) ile işlenmektedir. Bu katmanda elde edilen özellik haritaları üzerindeki 2x2 bölgelerde maksimum değerlerin alındığı MaxPooling katmanı ve sonrasında aktivasyon haritasının elde edildiği ReLU katmanı gelmektedir. Bloktaki tüm katmanlar benzer şekilde devam ederek tam bağlı katman ve softmax fonksiyonu ile sınıf skorları elde edilmektedir. İkinci uygulamada tasarlanan CNN modeline ek olarak giriş katmanına, VP yöntemi ile elde edilen görüntülerin boyutu verilmiştir ve hemen ardından gelen ilk konvolüsyon katmanının kernel boyutu 16 olarak değiştirilmiştir. Yol tipinin ve kalitesinin tespiti çalışmasında sınıf sayısının fazla olması nedeniyle daha derin bir model tasarlanmıştır. Bu modelle çok fazla özellik parametreleri elde edilmiş, aynı zamanda ara katmanlara bırakılan maxpooling katmanı ile farklılığı en yüksek elemanlar seçilmiştir.

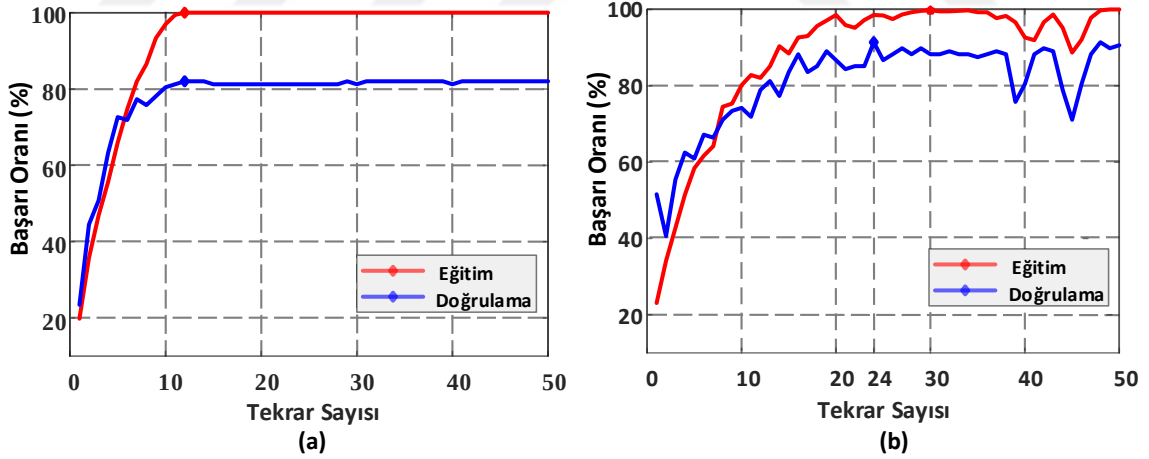
5.1.2. Modelin Yol Tipini Tespiti için Değerlendirilmesi

Bu deneysel çalışmada, modelin eğitimi için Veri seti-1'in %80'i (512) görüntü, geriye kalan %20'si doğrulama amacıyla kullanılmıştır. Tablo 5.2'de Veri seti-1'de bulunan yol tipi görüntülerinin sayısal olarak dağılımı görülmektedir.

Tablo 5.2. Veri seti-1'te bulunan yol tipleri görüntü dağılımları

Yol Tipi	Eğitim (%80)	Doğrulama(%20)
Bölünmüş Şeritli Yol	100	25
Stabilize Yol	99	25
Asfalt sokak Yolu	96	24
Parke Sokak Yolu	120	30
Tek Şeritli Yol	97	24
Toplam	512	128

Toplamda 640 görüntünün bulunduğu Veri seti-1'in eğitim ve doğrulama için ayrılmış yol tipleri rastgele seçilmiş ve eşit oranda dağılmamıştır. Rastgele seçildiklerinden sayıları farklı olmuştur. Bu veri set ile bir tanıma çalışması yapılmıştır. Bu amaçla, 5 yol türünün giriş katmanı için $128 \times 128 \times 3$ (genişlik $\times$ yükseklik $\times$ renk) boyutundaki görüntüler kullanılmıştır. Önerilen modelinin eğitim ve doğrulama süreçlerinin performans grafikleri, Şekil 5.4'te verilmiştir.



Şekil 5.4. Yol tipi sınıflandırmanın performans grafiği

Şekil 5.4(a)'da gösterildiği gibi, geliştirilen modelinin eğitim ve doğrulama aşaması 12. tekrarda en üst seviyeye çıkmıştır. Eğitim %100 olduğunda test performansı %82.03 olarak ölçülmüştür. Şekil 5.4(b)'de VP+Önerilen model ile görüntüler eğitilmiş ve doğrulama yapılarak oluşan performans grafiği verilmiştir. Eğitim aşaması 30. tekrarda tamamlanmış, en yüksek ve en erken test başarısına 24. tekrarda ulaşmıştır ve başarısı %91.41. olmuştur. VP+Önerilen modelin yaklaşık %9 performans artışı yaptığı gözlemlenmiştir.

Alandaki Farklı Modellerle Yapılan Çalışmaların Karşılaştırması

Şekil 5.4 incelendiğinde, VP yöntemi kullanılarak ön işleme verilerinin daha yüksek bir tanıma performansı değeri sağladığı görülmektedir. Çalışmada, literatürde yaygın olarak bilinen Vgg-5, Vgg-6, Vgg-8, AlexNet ve LeNet-5 CNN modelleri aynı veri setlerine uygulanmıştır. Bu modeller için birçok denemeye ve ağırlık ayarlamasına rağmen, performansın modelimizden düşük olduğu gözlenmiştir. Tablo 5.3'te, CNN modelleri tarafından veri seti için elde edilen başarımlar verilmiştir.

Tablo 5.3. Veri seti-1 üzerinde yapılan CNN modellerinin başarımlar oranları

Yol Tipi	RGB	RGB+VP
Vgg-8	71.88	88.28
Vgg-6	80.47	89.06
Vgg-5	81.25	89.84
AlexNet	81.67	92.67
LeNet	80.47	90.06
Önerilen Model	82.03	91.41

Yol Tipi Tespitinin Literatürdeki Yöntemlerle Karşılaştırılması

Yol tipinin tespiti ile ilgili literatürde çok fazla çalışma bulunmadığı görülmüştür. Genelde yolun dış kısmı ile ilgili çalışmalar olduğu ve bunların sınıflandırıldığı tespit edilmiştir. Görüntü işleme tekniklerinden ziyade ivmeölçer, donanımından elde edilen sinyal verilerinin sınıflandırılmasıyla sonuçlar elde edilmiştir.

Literatürde yol tipi tespiti ile ilgili birkaç çalışma tespit edilmiş ve bunlar incelenmiştir. Literatürde bulunan çalışmaların tek deneysel çalışma ile sonuç aldığı ve bu sonucu kullandığı belirlenmiştir. Bu nedenle yapılan bu uygulamada doğrulama aşamasında elde edilen en yüksek sonuç test başarımları olarak verilmiştir. Bu deneysel çalışmamızın başarımları ile literatürde bulunan benzer çalışmaların sonuçları Tablo 5.4'te görülmektedir.

Tablo 5.4. Geliştirdiğimiz modelin başarımlarının literatürdeki diğer çalışmalarla karşılaştırılması

Çalışma	Yıl	Toplam veri	Veri Tipi	Sınıf Sayısı	Metot	Başarı (%)
Taylor [25]	2012	8.618	Sinyal	4	DFT-DWT+J48,NB	89.00
Wang [87]	2012	-	Sinyal	2, 3, 4	FFT-FWT + SVM	99.00
Seeger [88]	2016	700 + 150	Sinyal	4	CNN	94.00
Slavkovikj [89]	2014	20.000	Görüntü	2	k-Means, SVM	85.30
Modelimiz		192 + 101	Görüntü	2	VP + CNN	91.41

Tablo 5.4 incelendiğinde, yol tipinin sınıflandırılması için toplamda 4 farklı çalışma incelenmiştir ve performans değerleri ile birlikte kullanmış oldukları metotlar ve veri sayıları görülmektedir. Taylor ve ark. [25] kendisinin topladığı 8.618 adet 4 sınıfa ait sinyal verisini DFT ve DWT tekniği ile önışlemeden geçirmiş ve farklı karar ağacı algoritmaları kullanarak en yüksek başarı olarak %89 elde etmiştir. Bu çalışma sadece sinyal verileri ile yapılmış ve gerçek hayata uyarlamasının yüksek hata payıyla mümkün olabileceği değerlendirilmiştir. Wang ve ark. [87] ivme ölçer sinyal verileri ile 2, 3 ve 4 sınıflı farklı çalışmalar yapmıştır. Çalışmasında veri önışlem için FFT, FWT kullanarak özellik çıkarımı yapılmış ve ardından elde edilen veri setleri SVM algoritmasına göre test edilmiş ve en yüksek başarısının %99 olduğunu ifade etmiştir. Seeger ve ark. [88] otoban, şehir içi park bölgesi ve sokak yollarında araç ile gezerek ivmeölçer cihazıyla veri toplamış ve bu verileri CNN metodunu kullanarak sınıflandırmıştır. 700 veri ile modeli eğitmiş, 15 veri ile modeli test ederek %94 başarı elde ettiğini ifade etmiştir. Slavkovikj ve ark. [89] GSV ortamında elde ettiği görüntüleri parçalara bölerek asfalt - stabilize yol ayrımı yapmak için SVM algoritması kullanmış ve başarısını %85.30 olarak tespit etmiştir. Görüntü sayımızın az olması ve gerçek verileri bütüncül olarak kullanmamıza rağmen geliştirmiş olduğumuz model oldukça yüksek başarı sağlamıştır.

5.1.3. Modelin Yol Kalitesi Tespiti için Değerlendirilmesi

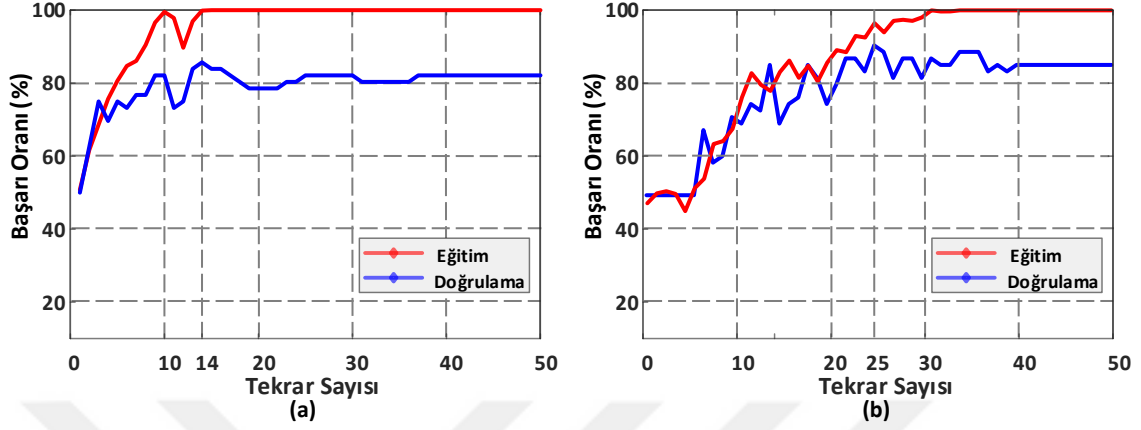
Kara yolları yüzeyinde oluşan delik ve çökmeleri tespit etmek için önermiş olduğumuz modelimizin giriş parametrelerini oluşturan batch size değerini 16 olarak belirleyerek her tekrarda daha az görüntünün işlenmesi sağlanmıştır. Böylelikle görüntüler arasındaki küçük farklılıklar daha rahat tespit edilmektedir. Görüntülerin daha derinlemesine incelenmesi için giriş katmanının kernel size değeri 16 olarak ayarlanmış ve daha küçük görüntüler elde edilerek daha fazla öznetelik oluşturulmuştur. Yol kalitesinin tespiti çalışmasında, Tablo 5.5'te görüldüğü gibi iki sınıftan oluşan toplam 277 görüntü içeren bir veri seti oluşturulmuştur.

Tablo 5.5. Veri seti-2'de bulunan yol tipleri görüntü dağılımları

Yol Tipi	Eğitim(%80)	Doğrulama(%20)
Bozuk Yol	111	28
Normal Yol	110	28
Toplam	221	56

Bu veri setindeki verilerin %80'i eğitim ve %20'si doğrulama için kullanılmıştır. Veri setinde yapılan tanıma çalışmaları için, bozuk yol görüntüleri içeren beş yol tipi örneği kullanılmıştır. Her sınıf için, görüntüler $128 \times 128 \times 3$ (genişlik $\times$ yükseklik $\times$ renk) olarak boyutlandırılmıştır. Eğitim süresi boyunca 221 görüntü (111 çukurlu yol ve 110 normal yol) ve doğrulama aşamasında 56

görüntü (2 sınıf $\times$ 28 örnek) kullanılmıştır. Önerilen modelinin öğrenme süreçlerinin performans grafikleri, Şekil 5.5'te görülmektedir.



Şekil 5.5. Yol kalitesi sınıflandırma performans grafiği

Şekil 5.5 (a)'da yol kalitesi tespiti için yapılan çalışmanın sonuçlarına baktığımızda, modelin performansının %87.50'ye ulaştığı ve eğitim aşamasının 14. tekrarda tamamlandığı görülmektedir. Şekil 5.5(b) incelendiğinde, VP yöntemiyle görüntülerin işlenmesinden sonra modelimizin daha iyi çalıştığı ve 25. tekrarda sınıflandırma performansın %91.07 olduğu görülmüştür. Bu deneyde, veri setindeki yol görüntülerinde çukurlar ve çatlaklar olduğu tespit edilmiştir. Şekil 5.6 bu veri setinde bulunan bazı görüntüleri göstermektedir.



Şekil 5.6. Bozuk- çukur yol veri setinden örnek görüntüler

Şekil 5.6’da görüldüğü gibi, yol kalitesi veri setinde bozuk yol verileri için, farklı yol tiplerinden görüntüler toplanmış ve rastgele seçilen görüntülerden eğitim ve test verisi oluşturulmuştur.

Alandaki Farklı Modellerle Yapılan Çalışmaların Karşılaştırması

Yol kalitesi çalışmasında, literatürde yaygın olarak bilinen Vgg-5, Vgg-6, Vgg-8, AlexNet ve LeNet-5 CNN modelleri aynı veri setlerine uygulanmıştır. Bu modeller için birçok denemeye ve ağırlık ayarlamasına rağmen, performansın modelimizden düşük olduğu gözlenmiştir. Tablo 5.6’da, CNN modelleri tarafından veri seti için elde edilen sonuçlar verilmiştir.

Tablo 5.6. Veri seti-2 üzerinde yapılan CNN modellerinin başarımları

Yol Tipi	RGB	RGB+VP
Vgg-8	83.93	82.14
Vgg-6	85.71	89.29
Vgg-5	83.93	87.50
AlexNet	88.63	92.29
LeNet	83.93	87.50
Önerilen Model	87.50	91.07

Tablo 5.6 incelendiğinde, VP metodunun yol kalitesini sınıflandırmada etkili olduğu ve performansı artırdığı görülmektedir. Vgg-8 modeli VP ile kullanıldığında performansın düştüğü, diğer modeller de ise yükseldiği görülmektedir. VP metodu, en yüksek etkiyi AlexNet mimarisinde yapmıştır ve performans değeri %92.29 olmuştur. Vgg-6, Vgg-5 ve LeNet mimarilerinde VP metodunun etkisi diğerlerine göre nispeten düşük çıkmıştır. Önermiş olduğumuz modelin VP metoduna tepkisi pozitif yönde yüksek olmuş ve başarı %91.07 olarak ölçülmüştür.

Yol Kalitesi Tespitinin Literatürdeki Yöntemlerle Karşılaştırılması

Yol kalitesinin tespiti ile ilgili literatür taramalarında çoğunlukla ivmeölçer modülle çalışma yapıldığı görülmektedir. İvme ölçer modül ile araç akıllı telefon uygulamalarıyla düz pürüzsüz yolda çukura girince modül pek noktalı sinyal üretir ve bu noktalarda kullanılan GPS verileri ile de yoldaki çukur noktalar tespit edilir. Bu yöntemin uygulanması için aracın çukura girmesi gerekmektedir, önceden çukur tespiti yapamamaktadır. Görüntü işleme ya da lazer verilerinin işlenmesi ile karayollarında önceden çukur tespiti yapılabilir ve bu sonuçlara göre uyarı sistemleri oluşturulabilir. Literatürde yapılan araştırma doğrultusunda karayolunda bulunan çukurların tespiti ile ilgili yapılan çalışmaların karşılaştırması Tablo 5.7’de verilmiştir.

Tablo 5.7. Geliştirdiğimiz modelin başarısının literatürdeki diğer çalışmalarla karşılaştırılması

Çalışma	Yıl	Toplam veri	Veri Tipi	Metot	Başarı (%)
Ryu ve ark. [90]	2015	90 - (100 video)	Görüntü	MHST + Otsu	73.50
Jo ve Ryu [91]	2015	100 - (20 video)	Görüntü	VP + Otsu	88.00
Koch ve Brilakis [92]	2011	120	Görüntü	Kendi metodu	86.00
Chen ve ark. [93]	2013	626,	Sinyal	CRSM + GMM	88, 85, 76, 92
Mednis ve ark. [94, 95]	2011	108+105	Sinyal	Z-THRESH, Z-DIFF, STDEV(Z), G-ZERO	84.00, 99.00, 87.00, 79.00
Modelimiz		192 + 101	Görüntü	VP + CNN	91.07

Tablo 5.7’de yollarda bulunan çukurlar, pürüzler ile ilgili literatürdeki çalışmaların önemli bir kısmı görülmektedir. Görüntü verileri ile yapılan çalışmalara bakıldığında Ryu ve ark. [90] 90 görüntünün histogram değerlerini alarak Otsu adlı algoritmada değerlendirmiş ve sınıflandırma başarısı %73.50 olmuştur. Yine Jo ve Ryu [91] yapmış olduğu çalışmada da aynı algoritmayı kullanmış, bu çalışmada ek olarak VP yöntemi ile görüntüler üzerinde önilem yaparak ilgisiz bölgeleri üst bitim noktasından kesmiştir. Bu çalışmada performans değeri %88 olmuştur. Koch ve Brilakis [92] 120 (normal ve çukur yol) görüntüyü kendi geliştirdiği yöntem ile sınıflamış ve %86 başarı elde etmiştir. Chen ve ark. [93] ivmeölçer modülü ile çukur yolların sinyal değerlerini gauss yöntemine göre değerlendirmiş ve başarısı %76 ile %92 arasında olmuştur. Mednis ve ark.[94, 95] farklı yöntemler geliştirerek toplamda 213 sinyal verisini uzun çukur, küçük çukur şeklinde çukur yüzey ve derinliğinin büyüklüğüne göre farklı algoritmalar ile test etmiş ve başarısı %79 ile %99 arasında çıkmıştır. Bizim çalışmamızın literatürdeki diğer çalışmalara göre avantajı, görüntü verileri kullanarak erken uyarı sistemi geliştirmektir. Farklı özelliklere sahip çukur ve bozuk yollar bu şekilde test edilmiş ve yüksek başarımlar elde edilmiştir.

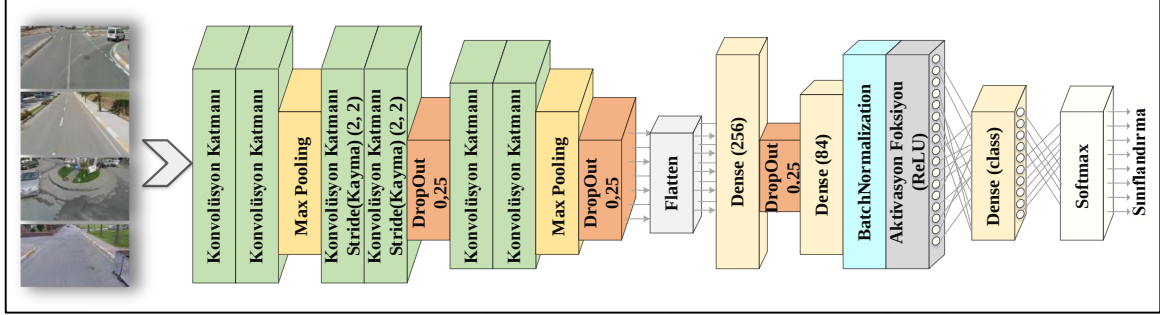
Yol tipi ve kalitesini tanımadaki başarıları incelendiğinde, VP metodunun kullanılması ile ciddi oranda artış görülmektedir. Bu nedenle bundan sonraki diğer çalışmaların tümünde sadece VP metodu kullanılarak sonuçlar alınmıştır.

5.2. Kavşak Tespiti

AUS için araçların ya da sürücülerin, karayolunda bulunan kavşakları tespit etmesi oldukça önemlidir. Uygulamanın bu bölümünde, karayollarında bulunan kavşakların, geliştirilen görüntü işleme ve derin öğrenme yöntemi ile tespiti yapılmıştır. Aynı zamanda tespit edilen karayolundaki kavşaklar, aynı karede bulunan trafik işaretleri ile eşleştirilmiştir.

5.2.1. Kavşak Tespiti için Önerilen Model

Karayollarında yol dokuları birbiriyle benzerlik göstermektedir. Yol kenarları, eğriliği, bitiş noktası gibi özelliklerin ortaya çıkarılması ve bu parametrelerin sınıflandırılması için model hazırlanmıştır. Şekil 5.7’de önerilen CNN modeli görülmektedir.



Şekil 5.7. Kavşak tespiti için hazırlanmış önerilen model

Şekil 5.7’de kavşak tespiti için geliştirmiş olduğumuz CNN modeli görülmektedir. Şekil 5.7’de önerilmiş olan CNN modeli hazırlamak için 6 konvolüsyon katmanı, 2 MaxPooling katmanı, 1 normalizasyon katmanı, 1 ReLU aktivasyon katmanı ve son olarak sınıflandırma katmanı için Softmax katmanı kullanılmıştır. Hazırlanan bu modelde katmanların bitiminde oluşan veri boyutunu azaltmak ve oluşabilecek modelin ezberlemesini (overfeed) engellemek için Dropout katmanı kullanılmış ve nöronlar arası çaprazlamayı yapmak için de Dense katmanı kullanılmıştır. Hazırlanan bu modelde 3 veri azaltma katmanı bulunmakta ve oluşan parametre yoğunluğunu %25 azaltmaktadır. Rastgele yapılan veri azaltma performans düşüklüğüne neden olmaktadır fakat kavşak türünde tek veri tipinin tanınması yapıldığı için veri azaltma katmanının sık kullanılması ciddi bir veri kaybı yaşatmamaktadır. Modelde maxpooling katmanı ile en yüksek parametreler elde edilmiştir. İlk maxpooling katmanı ilk iki konvolüsyon katmanından sonra oluşan veri karmaşıklığını azaltmaktadır. 2. maxpooling katmanı ise son konvolüsyon katmanından elde edilen parametrelerin içinde en iyi olanları seçecektir. Görüntülerin sınıflandırılması için Softmax katmanı kullanılmıştır. Bu model dönel kavşakların tespitinde, sağ ve sol yol ayrımlarının tespitinde ve yaya geçitlerinin tespitinde kullanılmıştır. Hazırlanan modelin parametreleri Tablo 5.8’de görülmektedir.

Tablo 5.8. Kavşak, yol ayrımı ve yaya geçitlerinin tespiti için önerilen CNN modelinin parametreleri

	Katman	Derinlik	Filtre boyutu	Adım (A) / Pooling(P)	İmge boyutu
1	Konvolüsyon	32	3×3	-	64×128×32
2	Konvolüsyon	32	3×3	-	62×126×32
3	Maxpooling			2×2 (P)	31×63×32
4	Konvolüsyon	32	3×3	2×2 (A)	15×31×32
5	Konvolüsyon	32	3×3	2×2 (A)	7×15×32
6	Dropout-Azaltma				7×15×32
7	Konvolüsyon	64	3×3	-	5×13×64
8	Konvolüsyon	64	3×3	-	3×11×64
9	Maxpooling			2×2 (P)	1×5×64
10	Dropout-Azaltma				1×5×64
11	Flatten	-	-	-	-
12	Dense				256
13	Dropout-Azaltma				256
14	Dense				84
15	Batch Normalization				-
16	Aktivasyon-ReLU				84
17	Dense				(2,3)
18	Softmax				(2,3)

Tablo 5.8 incelendiğinde 1,2,3 ve 4. Konvolüsyon katmanı 3×3'lük filtre ve 32 ağırlık vektöre sahiptir. Bu katmanların sonunda yüksek sayıda parametre elde edilir ve bu parametrelerin azaltılması için 2 maxpooling katmanı kullanılarak belirgin parametreler elde edilir. Maxpooling katmanında elde edilen çıkışta kavşak ayrıtların tespiti için belirgin parametrelerin artırımı sağlanmıştır. 3 ve 4. Konvolüsyon katmanları adım(stride) değeri 3×3 olarak ayarlanmıştır. 5 ve 6. Konvolüsyon katmanları 3×3'lük filtre ve 64 ağırlık vektörüne sahiptir. 1. dense katman değeri 256, 2. dense katmanı değeri 84 ve sınıflanan etiketli olan kategori sayısına göre dense katmanı 2 ve 3 olarak ayarlanmıştır, ayrıca Learning rate 10^{-5} ve Momentum değeri 0.5 olarak ayarlanmıştır.

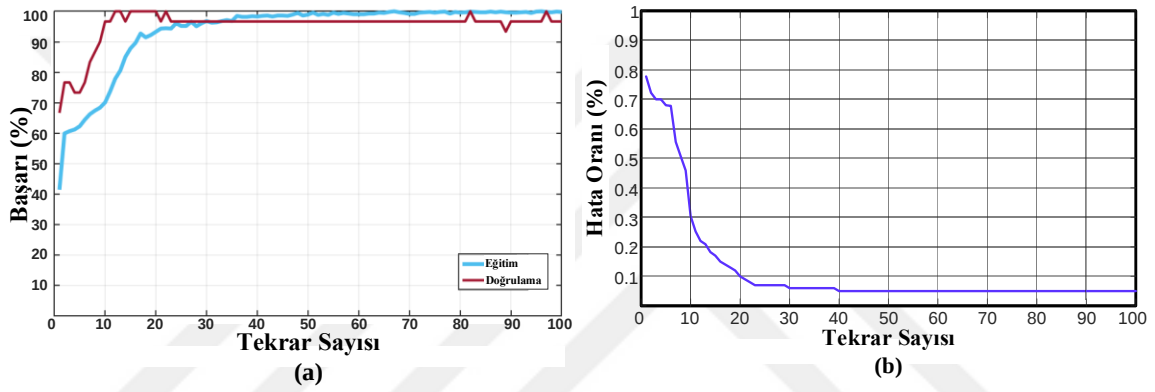
5.2.2. Modelin Kavşak Tespiti için Değerlendirilmesi

Bu deneysel çalışmada, kavşak içeren görüntüler ile normal yol görüntüleri içeren veriler kullanılmıştır. Çalışmamızda geliştirmiş olduğumuz CNN modeli ile bu veriler üzerinde tanıma işlemi yapılmıştır. Bu çalışma için toplamda 293 görüntü kullanılmış ve iki sınıflı bir başarımla elde edilmiştir. Tablo 5.9'da bu deneysel çalışma için kullanılan veri sayıları görülmektedir.

Tablo 5.9. Kullanılan yol tiplerinin sayıları

Sınıf	Toplam veri sayısı	Veri seti-3		
		Eğitim	Doğrulama	Test
Kavşak	192	153	20	19
Normal yol	101	81	10	10
Toplam	293	234	30	29

Tablo 5.9 incelendiğinde, kavşak veri sayısı ile normal yol görüntü sayısının eşit olmadığı görülmektedir. Hazırlanan modelde veri set' in %80'i eğitim için kullanılmış, doğrulama ve test aşamasının her birinde %10'u kullanılmıştır. Bu işlemlerin sonuçları Şekil 5.8'de performans ve kayıp fonksiyon grafiği verilmiştir.



Şekil 5.8. Modelin başarımlar grafiği ve kayıp(loss) fonksiyonu

Hazırlanan derin CNN modelinin yoldaki kavşakları tespit etmedeki başarısı oldukça yüksek olarak ölçülmüştür. Şekil 5.8'de görüldüğü gibi CNN modelin veri sette bulunan birbirine yakın benzerlikteki imgeleri %100 oranında doğruladığı görülmektedir.

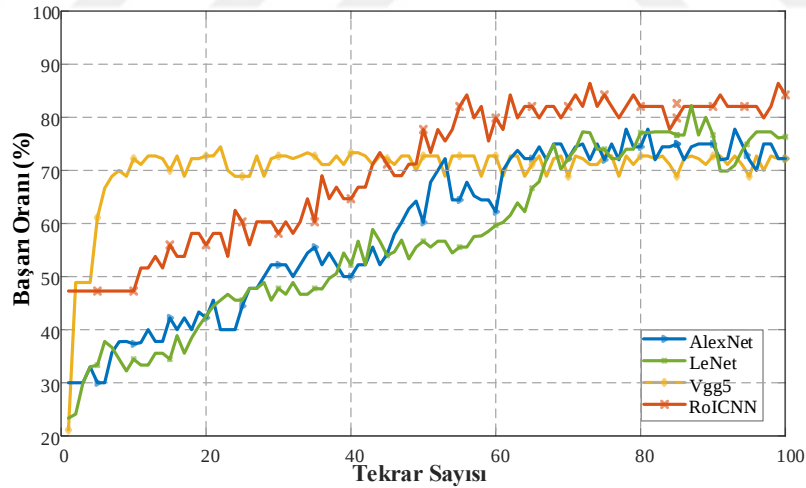
Alandaki Farklı Modellerle Yapılan Çalışmaların Karşılaştırılması

Kavşak tespiti için yapılan bu çalışmada elde edilen başarı değerleri ve literatürde oldukça sık rastlanan CNN modelleri ile yine aynı veri set üzerinden sonuçlar alınmış ve bu sonuçların karşılaştırılması yapılmıştır. Sınıf sayısının ve özellikle test edilen verilerin az olmasından dolayı güvenilir sonuçlar elde etmek için, veri setteki tüm görüntüler test edilmelidir [96]. Bu nedenle çapraz doğrulama (cross validation) yöntemi uygulanarak modelin eğitimi için veri setin %80'i rastgele seçilmiş ve geriye kalan veri setin %20'sinin yarısı doğrulama yarısı da test için kullanılmıştır. Bu işlem 5 defa tekrar edilerek farklı sonuçlar elde edilmiştir. Tüm bu işlemler literatürde sık kullanılan modeller (AlexNet, LeNet, Vgg-5) üzerinde de test edilmiş ve oluşan sonuçlar Tablo 5.10'da verilmiştir.

Tablo 5.10. Farklı eğitimli modeller kullanılarak elde edilen sınıflandırma sonuçları ve ortalamaları

Modeller	Performans Değerleri (%)					Ortalama
	Bölme-1	Bölme-2	Bölme-3	Bölme-4	Bölme-5	
AlexNet	93.33	90.00	86.67	96.67	90.00	91,11±5.09
LeNet	93.33	86.67	83.33	96.67	83.33	87.50±6.31
Vgg-5	83.33	86.67	76.67	93.33	76.67	83.34±7.06
Modelimiz	96.67	93.33	90.00	100,00	96.67	95.34±3.80

Tablo 5.10’da görüldüğü gibi önceden eğitilmiş modelin performansını karşılaştırmak için 5 kat çapraz doğrulama yöntemi uygulanmıştır. Her model, veri setin farklı bölümleri ile 5 kez eğitilmiş ve her bir eğitim için sınıflandırma doğruluğu hesaplanmıştır. Daha sonra tüm eğitilmiş modeller için, sınıflandırma performanslarının ortalaması hesaplanmıştır. Bu sonuçlara göre en düşük sınıflandırma performansının (%76.67) Bölüm-5 veri seti Vgg-5 modelinden elde edildiği ve en yüksek sınıflandırma performansının (%100) ile Bölme-4’te geliştirmiş olduğumuz modelden elde edildiği tespit edilmiştir. Modellerin performans ortalamaları incelendiğinde, en düşük ortalamanın (%83.34±7.06) Vgg-5 modelinde olduğu, en yüksek performans ortalamasının ise (%95.34±3.80) geliştirmiş olduğumuz modelde görülmüştür. Bölme-4 veri setinde tüm modeller üzerinden alınan sonuçların performans grafiği Şekil 5.9’da verilmiştir.



Şekil 5.9. Bölme-4’e göre CNN modellerinin başarı oranları

Şekil 5.9’da, en yüksek sınıflandırma performansının (%100) geliştirmiş olduğumuz modelden elde edildiği ve en düşük performansın (%93,33) Vgg-5 eğitimli modelden elde edildiği görülmektedir. AlexNet ve LeNet modellerinin sınıflandırma başarıları %96.67 olarak tespit edilmiştir.

Eğitimli modellerin ayrıntılı performans değerlendirmesi için, yaygın olarak kullanılan doğruluk, kesinlik, anma ve F1-skor değerleri ölçülmüştür. En yüksek performans parametreleri bölüm-4 veri setinde ölçülmüş ve AlexNet, LeNet, Vgg-5 modellerinin eğitilmesi ile elde edilen performans parametreleri Tablo 5.11'de verilmiştir.

Tablo 5.11. CNN modellerinin kavşak tespiti için bölüm 4 veri setindeki performans sonuçları

CNN Modelleri	Doğruluk (%)	Kesinlik(%)	Anma(%)	F1-skor(%)
AlexNet	96.67	97.00	97.00	95.00
LeNet	96.67	96.00	89.50	91.50
VggNet-5	93.33	96.66	90,50	86,25

Tablo 5.11 incelendiğinde Vgg-5 modeli hatırlama performansı hariç, diğer değerlendirmelerde en düşük ortalamaya sahipken, AlexNet ve LeNet modelleri nispeten daha yüksek performans göstermiştir. Modelimizin performansı ise tüm değerlendirmelerde en yüksek çıkmıştır. Tablo 5.12'de, tüm modellerin ortalama başarı değerleri görülmektedir.

Tablo 5.12. Kavşak veri setindeki CNN modellerinin ortalama performansı

Modeller	Doğruluk (%)
AlexNet	91.11
LeNet	87.50
VggNet-5	83.34
Geliştirilen Model	95.34

Tablo 5.12'ye göre, geliştirmiş olduğumuz modelin kavşak tanıma performansının diğer modellerden daha yüksek olduğu görülmüştür. VggNet-5 modelinin başarısı, en düşük başarı oranı olan %83,34 olmuştur. LeNet modelinin başarısı %87,50, AlexNet modelinin başarısı %91,11 olarak tespit edilmiştir. Modelimizin ortalama başarısı %95,34 olarak ölçülmüş ve bu ortalama sonuç, diğer modellere göre yüksek performans göstermiştir.

Kavşak Tespitinin Literatürdeki Yöntemlerle Karşılaştırması

Literatürde kavşak tespiti ile ilgili birkaç çalışma tespit edilmiş ve bunlar incelenmiştir. Bu deneysel çalışmamızın ortalama başarısı ile literatürde bulunan benzer çalışmaların sonuçları Tablo 5.13'te görülmektedir.

Tablo 5.13. Geliştirdiğimiz modelin başarısının literatürdeki diğer çalışmalarla karşılaştırılması

Çalışma	Yıl	Toplam veri	Veritabanı	Sınıf Sayısı	Metot	Başarı (%)
Rebai [14]	2014	10 + 88	Kendisine ait	4	SVM	71.42 – 93.61
Bhatt-Sodhi [12]	2017	880 - 15.500	Lara-Oxford	2	CNN+LSTM	78.25 - 92.16
Yi [97]	2017	200 + 200	KITTI-Kendi	2	CNN	90.50
Ballardini [98]	2017	2.527	KITTI	7	SUTSU-GIST	71.00-59.00
Modelimiz		192 + 101	Kendimize ait	2	VP & CNN	95.34

Tablo 5.13'e göre, Rebai [14], SVM sınıflandırıcısını kullanarak 88 kavşak görüntüsü ile birlikte 88 normal yol görüntülerini sınıflandırmış ve performansı (%91.21) olmuştur. Bhatt ve Sodhi [12] tarafından CNN ve LSTM yöntemi kullanılarak yapılan çalışmada, çok fazla veriye ve hazır veri setine rağmen, performansı %92.16 olarak ölçülmüştür. Yi [97], KITTI veri setini, model eğitimi için 200 kavşak noktası ve 200 kavşak olmayan normal yol görüntüsü ile kullanmış ve kendi ürettiği yakın mesafeli görüntülere testler uygulayarak %90.50 başarı elde etmiştir. Ballardini [98], SUTSU-GIST algoritmaları kullanarak %71-%59 performans ile sınıflandırmıştır. Önerilen modelin başarısının sınırlı sayıda veriye ve görüntülerin rastgele seçimine rağmen, diğer çalışmalara kıyasla yüksek olduğu ve ortalama başarının %95.34 olduğu görülmüştür. Bu yüksek başarı oranı, hem VP yönteminin görüntülerden RoI bölgesini doğru tespit ettiğini hem de önerilen modelin görüntülerdeki öz nitelikleri tespit etmedeki yüksek başarısını gözler önüne sermektedir.

5.3. Sağ ve Sol Yol Ayrımalarının Tespiti

Literatürde sol ve sağ kavşak olarak da değerlendirilen yol ayrımları, karayollarının önemli bileşenleridir. Özellikle şehir içlerinde aracın sola ya da sağa dönüşlerinde sık karşılaşılan bölgelerdir. Trafiğin sağdan aktığı ülkelerde (çoğunlukla sağdan akmakta) tek şeritli ve bariyerli karayollarında sağ yol ayrımı çok bulunurken, şehir içi bölgelerde hem sağ hem de sol yol ayrımları mevcuttur. Yakın doku değerlerine sahip asfaltlı karayollarının yol ayrımları, kavşaklarda olduğu gibi kaldırım, yol çizgileri gibi öznitelik değerleri farklı noktalardan tespit edilebilir. Sol ve sağ yol ayrımlarında araçların hem araçlarla hem de yayalarla karşılaşma ihtimalleri normal yola göre daha yüksektir. Araçların dönüş yapması ya da yayanın yolun karşısına geçmesi durumlarıyla sık karşılaşılmaktadır. Bu nedenle yol ayrımları karayollarında önemli bir yere sahiptir. Bu bölgelere varmadan yol kenarlarına, yol ayrımının ne olduğu ile ilgili trafik levhaları bırakılmaktadır. Sürücü destek sistemleri için ya da otonom araçlar için yolun hangi tarafa döndüğü bilgisinin tespiti oldukça önemlidir. Uyarı mekanizmalarını harekete geçirmek ve aracın karar almasına yardımcı olmak için bu bölgelerin tespit edilmesi gerekmektedir.

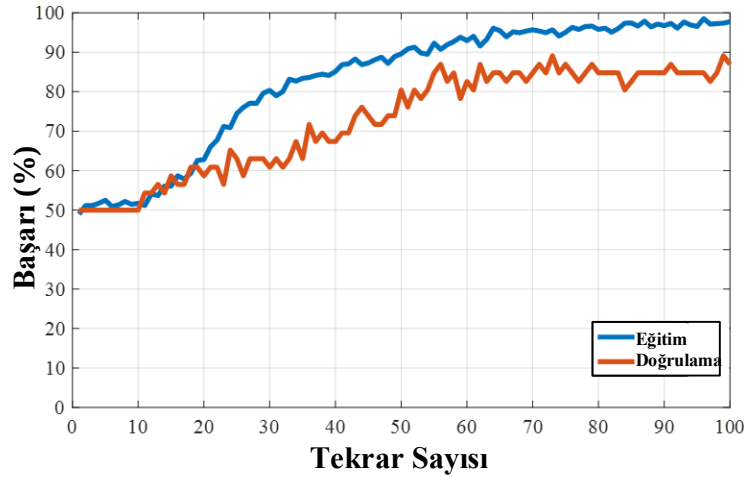
5.3.1. Modelin Sol ve Sağ Yol Ayrırıları Tespiti için Değerlendirilmesi

Karayolunda bulunan trafik akışına göre sol ve sağ kavşağın tespiti için, önermiş olduğumuz derin öğrenme tabanlı yöntem uygulanmıştır. Hazırlanan Şekil 5.7'deki modelin eğitimi için 449 görüntüden 359 (%80)'u kullanılmıştır. Tablo 5.14'te bu deneysel çalışma için kullanılan veri sayılarının dağılımı görülmektedir.

Tablo 5.14. Kullanılan yol tiplerinin sayıları

Sınıf	Toplam veri sayısı	Veri seti -4		
		Eğitim	Doğrulama	Test
Sol yol ayrımı	117	93	12	11
Sağ yol ayrımı	231	185	23	23
Normal yol	101	81	10	10
Toplam	449	359	45	44

Tablo 5.14'te görüldüğü gibi sol yol ayrımı için 117, sağ yol ayrımı için 231 adet imge kullanılmıştır. Türkiye ve birçok ülkede trafik akışı sağdan olduğu için sağ yol ayrımı görüntüleri, sol yol ayrımı görüntülerine göre daha fazladır. Normal yol görüntüleri için, ilk deneysel çalışmada bulunan imgeler kullanılmıştır. Tüm imgeler karayolunda hareket eden bir aracın bulunacağı farklı ortam ve koşullar gözetilerek elde edilmiştir. Veri setinin eğitimi ve doğrulaması işlemi sonucunda modelin başarı oranı Şekil 5.10'da görülmektedir.



Şekil 5.10. Modelin başarı oranı grafiği

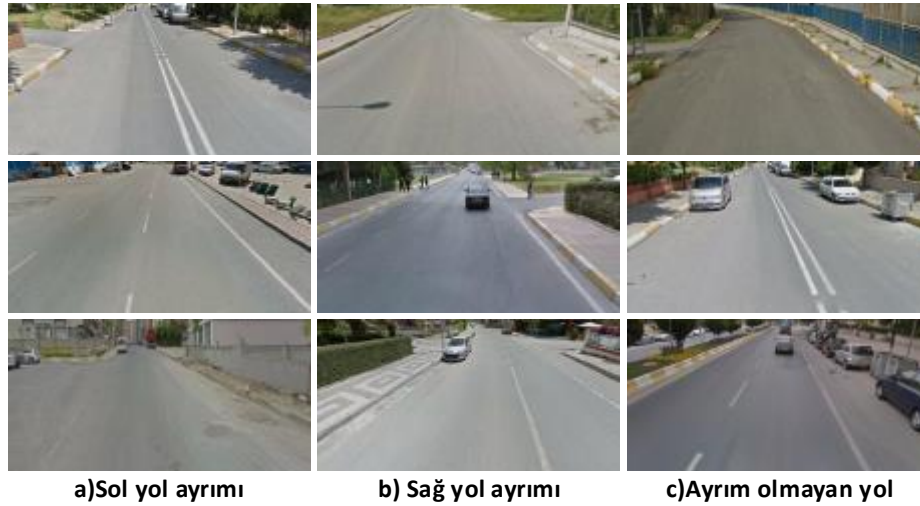
Şekil 5.10 incelendiğinde, modelin eğitim sürecinin uzun sürdüğü, sınıf ayrımının oldukça güç olduğu görülmektedir. Veri setindeki görüntülerin öznetelik değerleri benzerlik gösterdiği için eğitim aşaması uzun sürmüştür. Modelin veri setinde bulunan birbirine yakın benzerlikteki imgeleri %89.1 oranında doğruladığı görülmektedir. Test için ayrılan veri setinin sınıflandırma başarı oranı en

yüksek eğitilmiş model üzerinden yapılmıştır. Bu işlem sonucu geliştirilen modelin başarısı %86.36 çıkmıştır. Önerilen modelin veri set üzerindeki sınıflandırma ile ilgili tüm performans değerleri Tablo 5.15'te görülmektedir.

Tablo 5.15. Önerilen modeli kullanarak yol ayırımlarının detaylı sınıflandırma performansları

Sınıf	Veri sayısı	Doğruluk (%)	Kesinlik (%)	Anma(%)	F1 – Skor (%)
Sol yol ayrımı	11	81.80	82	82	82
Sağ yol ayrımı	23	95.20	95	87	91
Normal yol	10	75.00	75	90	82
Toplam	44	86.36	87	86	87

Tablo 5.15'te görüldüğü gibi en yüksek sınıflandırma başarısı %95.20 ile sağ yol ayrımında olduğu, en düşük başarının ise %75 ile normal yol ayrımında olduğu görülmektedir. Sağ yol ayrımında görüntü sayısının fazla olmasının, sınıflandırma başarısını bir nebze etkilemiş olduğu görülmektedir. Çalışmada algılama performansı en yüksek sağ yol ayrımında olurken, anma performansının en yüksek normal yol ayrımında olduğu görülmektedir. Sınıflandırılan görüntülerin birbirine yakın doku benzerliklerinin sınıflandırma performansını etkilemiş olduğu görülmektedir. Şekil 5.11'de çalışılan veri setindeki yakın görüntü değerlerine sahip sol yol ayrımı, sağ yol ayrımı ve normal yol görüntülerinin bir kısmı görülmektedir.



Şekil 5.11. Sol-sağ ve normal yol türünde örnek görüntüler

Şekil 5.11 incelendiğinde, veri setindeki görüntülerin birbirine yakın görüntü değerleri içerdiği görülmektedir. Benzerlik oranları yüksek olmasına rağmen %86.36 gibi yüksek bir başarı oranı ile görüntüleri doğru sınıflama yaptığı tespit edilmiştir.

Alandaki Farklı Modellerle Yapılan Çalışmaların Karşılaştırılması

Sol ve sağ yol ayrımlarının tespiti için yapılan bu deneysel çalışmada elde edilen başarı değerleri ve literatürde oldukça sık rastlanan CNN modelleri ile yine aynı veri seti üzerinden sonuçlar alınmış ve bu sonuçların karşılaştırılması yapılmıştır. Kavşak tespiti için hazırlanan model üzerinden yapılan bu çalışmanın sonucu ile AlexNet, LeNet ve Vgg-5 modelleri ile elde edilen sonuçlar karşılaştırılmış ve başarımlar sonuçları Tablo 5.16’da gösterilmiştir.

Tablo 5.16. Modelimiz ve diğer CNN modellerinin yol ayrımları tespit sonuçları

Modeller	Doğruluk (%)
VggNet 5	72.73
LeNet	77.27
AlexNet	74.99
Önerilen model	86.36

Tablo 5.16’da bulunan sonuçlarda önermiş olduğumuz modelin başarımının diğer CNN modellerine göre daha yüksek çıktığı görülmektedir. Bu çalışmada test edilen VggNet-5 başarısı %72.73 olarak ölçülürken, LeNet mimarisi görüntüleri %77.27 oranında doğru tespit etmiştir. AlexNet’in görüntüleri sınıflamadaki başarısı ise %74.99 olarak ölçülmüştür. Modelimizin ortalama başarısı, diğer modellerle göre oldukça yüksek ve %86.36 olarak gerçekleşmiştir. Bu çalışma sonucunda önermiş olduğumuz modelin veri setinde test edilen veriler üzerindeki karşılaştırma matrisi Şekil 5.12’de görülmektedir.

Doğruluk %86.36				
Tespit Edilen Sınıf	Sınıf	Normal Yol	Sol Yol Ayrımı	Sağ Yol Ayrımı
	Normal Yol	%75.0 (9)	%0.0 (0)	%4.8 (1)
	Sol Yol Ayrımı	%16.7 (2)	%81.8 (9)	%0.0 (0)
	Sağ Yol Ayrımı	%8.3 (1)	%18.2 (2)	%95.2 (20)
Gerçek Sınıf				

Şekil 5.12. Geliştirilen modelin normal yol ile sol – sağ yol ayrımları görüntülerin başarısının karşılaştırma matrisi

Şekil 5.12 incelendiğinde, 10 normal yol görüntüsünden sadece biri sağ yol ayrımı olarak yanlış sınıflandırılmışken, sol yol ayrımı görüntülerinden ikisi normal yol ve sağ yol ayrımı görüntülerinden biri normal yol olarak yanlış sınıflandırılmıştır. Aynı zamanda normal yol

görüntüsü olarak yanlış tespit ettiği 2 adet sol yol ayrımı ve 1 adet sağ yol ayrımı görüntüsü bulunmaktadır. Bu nedenle normal yol görüntülerinin sınıflandırılması %75 başarı göstermiştir. Öte yandan sağ yol ayrımı görüntülerinin sadece 3'ü yanlış sınıflandırılırken, 1 adet normal yol görüntüsü sağ yol ayrımı olarak yanlış tespit edilmiştir ve başarı %95.2 olmuştur. Sol yol ayrımının sınıflandırmadaki başarı ise %81.8 olarak ölçülmüştür. Sol yol ayrımı görüntülerinin sayısının düşük olması başarı oranını olumsuz etkilemiştir. Fakat yine de ortalama başarı %86.36 olmuştur.

Sol ve Sağ Yol Ayrımalarının Tespitinin Literatürdeki Yöntemlerle Karşılaştırılması

Sol ve sağ yol ayrımı ile ilgili yapılan çalışmalar genelde kavşak tespiti çalışmalarında bulunmaktadır. Bu deneysel çalışmamızda aldığımız sonuçların literatürde yapılan benzer çalışmalar ile karşılaştırılması Tablo 5.17'de verilmiştir.

Tablo 5.17. Sol ve Sağ kavşak tespiti için geliştirdiğimiz modelin literatürdeki diğer çalışmalarla karşılaştırılması

Çalışma	Yıl	Toplam veri	Veritabanı	Sınıf Sayısı	Metot	Başarı (%)
Rebai [14]	2014	50+88	Kendi verisi	4	SVM	73.33(Sol) - 88.57(Sağ)
Bhatt-Sodhi [12]	2017	880 – 15.500	Lara-Oxford	3	CNN & LSTM	91.7208
Yi [97]	2017	400 +	KITTI-Own	2	CNN	91.30 - 91.60
Ballardini [98]	2017	500(L), 952(R)	KITTI	7	SUTSU-GIST	66.00-55.00
Modelimiz		450	Kendi verimiz	3	VP & CNN	81.80(Sol) – 95.2(Sağ)

Tablo 5.17 incelendiğinde, Rebai [14] çalışmasında toplamda 138 adet görüntü kullanmış ve bunun 50 adedi sol ve sağ kavşak olarak seçmiştir. Çalışmasında kavşak, sol kavşak ve sağ kavşak ve kavşak yok şeklinde sınıflandırma yapmıştır. Sol kavşağı %73.33 ile doğru tespit ederken, sağ kavşağı %88.57 ile doğru sınıflandırmıştır. Bhatt-Sodhi [12] çalışmasında T- kavşak olarak nitelediği 47 görüntüyü sol-sağ kavşak şeklinde göstermiştir. Hazır veri set üzerinde yapmış olduğu çalışmada %91.72 bir başarı elde etmiştir. Yi [97] çalışmasında ise KITTI veri set üzerinden eğittiği modeli, araç farklı hızlarda hareket halinde iken almış olduğu görüntülerle test etmiştir. Ortalama 5 – 20 km/s hıza göre elde edilen görüntülerin sınıflandırma başarısını %91.30 – 91.60 olarak tespit etmiştir. Ballardini [98] SUTSU-GIST algoritmasını kullanarak sol ve sağ kavşak verilerini sınıflandırmış ve başarısı %71 ve %59 olarak bulunmuştur. Önermiş olduğumuz modelin başarısı; toplamda 449 rastgele seçilmiş veriler üzerinde yapılmıştır. Veritabanındaki görüntülerin ayırt ediciliği düşük olmasına rağmen modelin başarısı oldukça yüksek olmuştur.

5.4. Yaya Geçitlerinin Tespiti

Yaya geçitleri özellikle şehir içi karayollarında önemli bir yere sahiptir. Karayollarında araç ile yayanın en sık karşılaştığı bölgedir. Yayanın yolun diğer tarafına geçmek için kullandığı, genellikle yolun beyaz kalın şeritler ile çizildiği özel bölgelerdir. Bu bölgelere varmadan yol kenarlarına yaya geçidi ile ilgili trafik uyarı levhaları bırakılmaktadır. Şekil 5.13'te örnek yaya geçitleri görüntüleri bulunmaktadır.



Şekil 5.13. Veri seti-5'te kullanmış olduğumuz örnek yaya geçitleri görüntüleri

Sürücü destek sistemleri için ya da otonom araçlar için şehir içi karayolunda bulunan yaya geçitlerinin tespiti ölümcül-yaralanmalı trafik kazalarının önlenmesi için oldukça önemlidir. Sürücüler ya da araçların uyarılması için yaya geçidi tespit sistemi geliştirilmiştir. Karayolunda bulunan yaya geçitlerinin tanınması için Şekil 5.7'de görülen ve Tablo 5.8'de parametreleri detaylı bir şekilde verilen model üzerinde düzenleme yapılarak kullanılmıştır. Bu düzenlemelerde konvolüsyon katmanlarında bulunan stride parametreleri kaldırılmış, aynı zamanda dense parametresi artırılarak daha fazla gizli nöron oluşturulmuştur. Karayolunda bulunan beyaz kalın şeritlerin bütün olarak tespiti gerektiğinden parametre kaybı yaşanmaması için stride parametresi kaldırılmıştır. Yaya geçitlerinin bulunduğu görüntüler birbirine benzerlik göstermekte diğer sınıf için farklı veri tipleri bulunmaktadır. Bunların tespitinde yaya geçitleri işareti barındıran görüntüler sınıflandırmaya çalışılmıştır.

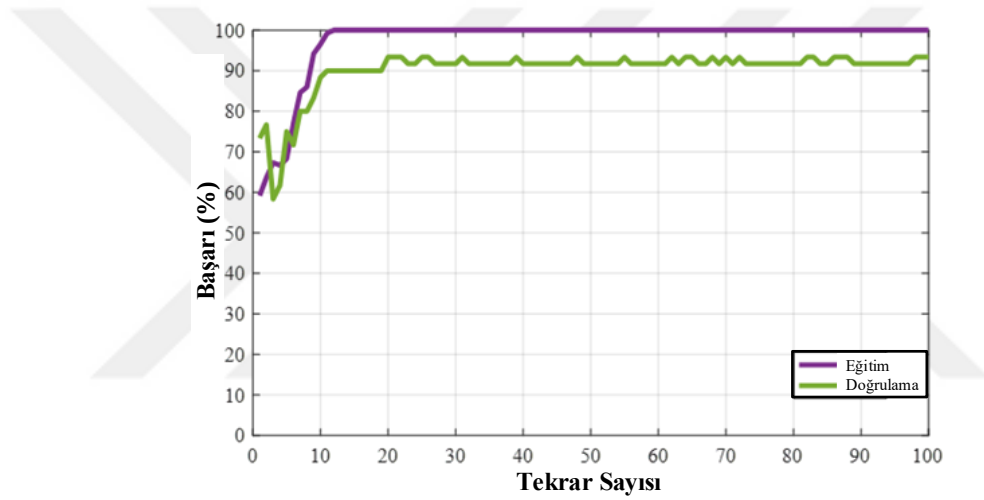
5.4.1. Modelin Yaya Geçitlerinin Tespiti için Değerlendirilmesi

Bu çalışmada 195 adet yaya geçidi işareti içeren ve 101 normal yol bölmesi olan toplamda 297 adet yol görüntüsü toplanarak bir veri set hazırlanmıştır. Tablo 5.18'de bu deneysel çalışma için kullanılan veri sayıları görülmektedir.

Tablo 5.18. Kullanılan yol tiplerinin sayıları

Sınıf	Veri Sayısı	Veri seti - 5		
		Eğitim	Doğrulama	Test
Yaya Geçidi	195	156	20	19
Normal Yol	101	81	10	10
Toplam	296	237	30	29

195 adet yaya geçidi görüntüleri hem normal yol görüntüleri ile yakın benzerlik göstermekte hem de standart bir yaya geçidi işaretleri bulunmamaktadır. Yaya geçidi işaretleri genelde koyu zemin üzerinde beyaz kalın şeritler şeklinde işaretlenmiştir. Fakat sarı renkte yol işaretlerine sahip yaya geçidi görüntüleri bulunmaktadır. Tüm bu yaya geçitleri görüntüleri veri setine eklenmiştir. Geliştirilen modelin başarı oranı Şekil 5.14'te görülmektedir.



Şekil 5.14. Önerilen modelin yaya geçidi tespitinde eğitim ve test aşamalarında göstermiş olduğu başarı grafiği

Yaya geçitlerinin tespiti için kullanılan modelin doğrulamadaki başarısı Şekil 5.14'te görüldüğü gibi %93.33 olarak ölçülmüştür. Bu çalışma sonucunda eğitilmiş model veri setinden rastgele ayrılmış veriler ile test edildiğinde %93.10 doğru tespit etmiştir. Önerilen modelin Veri seti-3 üzerinde tüm performans sonuçları Tablo 5.19'da görülmektedir.

Tablo 5.19. Önerilen modeli kullanarak yaya geçitlerinin detaylı sınıflandırma performansları

Sınıf	Veri sayısı	Başarı Oranı (%)	Kesinlik (%)	Anma (%)	F1-skör (%)
Yaya geçidi	19	94.73	95.00	95.00	95.00
Normal yol	10	90.00	90.00	90.00	90.00
Toplam-Ortalama	29	93.10	93.00	93.00	93.00

Alandaki Farklı Modellerle Yapılan Çalışmaların Karşılaştırılması

Yaya geçitlerin tespiti için yapılan bu deneysel çalışmada aynı tür modeller (AlexNet, LeNet, Vgg-5, Darknet) kullanılarak sonuçlar elde edilmiş ve karşılaştırması yapılmıştır. Bu sonuçların karşılaştırılması Tablo 5.20’de görülmektedir.

Tablo 5.20. Yaya geçidi veri setindeki CNN modellerinin sınıflandırma performansları

Modeller	Başarı (%)
LeNet	83.33
Vgg-5	86.67
AlexNet	86.66
Darknet	93.10
Modelimiz	93.10

Yaya geçitlerinin tespiti için yapılan bu çalışmada başarı oranı %93.10 ile diğer CNN modellerine göre yüksek çıkmıştır. Vgg-5 modeli ile AlexNet modeli %86.66 ile doğru sınıflandırırken, LeNet modeli %83.33 ile nispeten biraz daha düşük başarı göstermiştir. Bu çalışmada Darknet mimarisi ile de sonuç alınmış ve %93.10 başarı göstermiştir.

Yaya Geçitleri Tespitinin Literatürdeki Yöntemlerle Karşılaştırılması

Çalışmamızın bu bölümünde yaya geçitleri tespitinin literatürde yapılan benzer çalışmalar ile karşılaştırılması Tablo 5.21’de görülmektedir.

Tablo 5.21. Yaya geçidi ile ilgili literatürdeki çalışmaların sonuçların karşılaştırılması

Çalışma	Yıl	Toplam veri	Veritabanı	Sınıf Sayısı	Metot	Doğruluk (%)
Radvanyi [21]	2010	89	Kendi veri seti	2	CNN	89.2
Wang [23]	2014	122	Kendi veri seti	3	SVM	78.9 - 93.9
Chen [19]	2017	50	Kendi veri seti	2	Görüntü bölme	92
Berriel [99]	2017	657K,11K, 11K,	GSV, IARA, GOPRO	2	CNN	94.21, 93.06, 89.30
Modelimiz		195+101	Kendi veri setimiz	2	VP & CNN	94.73 - 93.10

Tablo 5.21 incelendiğinde Radvanyi [21], CNN yöntemi kullanarak toplamda 89 adet görüntü sınıflandırmış ve başarımı %89.2 olarak vermiştir. Wang [23], toplamda 122 adet merdiven ve yaya geçitleri görüntülerini SVM yöntemi kullanarak sınıflandırmış ve başarımı yaya geçidi için %78.9, ortalama başarısı %93.9 olarak vermiştir. Chen [19], farklı ışık altında elde etmiş olduğu toplamda 50 yaya geçidi görüntüyü sınıflandırmıştır ve başarısı %92 çıkmıştır. Berriel’in[99], videolardan çerçeveleri(frameleri) çekerek elde etmiş olduğu yüksek miktardaki veri setler ile yaya geçitlerini tanıma işleminde kullanmış ve %94.21 doğrulukta tespit etmiştir. Bu çalışmamızda 296

adet görüntünün %80'i sistemin eğitimi için kullanılmıştır. Bu çalışma sonucunda yaya geçidini %94.73 ile doğru tespit etmiş ve ortalama başarıyı %93.10 olarak ölçmüştür.

5.5. Sağ ve Sol Tehlikeli Virajların Tespiti

Otonom araçlar ya da sürücü destek sistemlerinde yol çizgisi takip etme veya yol sınır alanları belirleme sistemleri kullanılmaktadır. Yol takip sistemi ile sürücüler uyarılabilir ya da otonom sistem için araç yönlendirmesi yapılabilir. Karayolun bir bileşeni olan virajların tespiti önemli bir sorundur. Araçlar hareket halindeyken virajlara yaklaştığında, uyarı mekanizmalarının devreye girmesi ve varsa yüksek hızın düşürülmesi gerekmektedir. Yolun doğrultusunun hangi yöne olduğunun önceden tespit edilmesi ile oluşabilecek trafik kazalarının da önlenmesi sağlanır. Yol çizgileri ile ayrılmış, aynı zamanda belli bir sınır olan karayollarında asfaltın kendi renginden farklı olan şerit dışındaki rengi ve şerit çizgilerin doğrultusu tespit edilerek aracın yönü belirlenebilir. Tehlikeli viraj uyarısı için yolun sol ve sağ kenarlarına uyarı levhaları yol tipine göre yerleştirilmektedir. Şekil 5.15'te yola yerleştirilen tehlikeli viraj işaret levhaları görülmektedir.



Şekil 5.15. Karayollarında bulunan tehlikeli viraj işaretleri

Karayollarında bulunan tehlikeli viraj işaretleri hem yolun sağında hem de solunda bulunmaktadır. Bu işaretler zamanla yıpranmakta, hava şartlarına bağlı olarak çökme olmakta veya yolda bulunmamaktadır. Karayolunda özellik çıkarımları yaparak yolda tehlikeli viraj bulunup bulunmadığı tespit edilebilir.

Karayolunun türüne bağlı olarak hazırlanacak tanıma sistemi ile yolun gerçek durum tespitinin yapılması için bir yöntem önerilmiştir. Bu yöntemde CMU-VASC (Road Sequences) [80] veritabanında bulunan yol görüntülerin ön seçim işleminden geçirilerek 587 adet sol-sağ tehlikeli

viraj içeren ve viraj barındırmayan 3 tip karayolu görüntüleri ayrılmıştır. Şekil 5.16'da tehlikeli viraj tespiti için hazırlanan veri set görülmektedir.

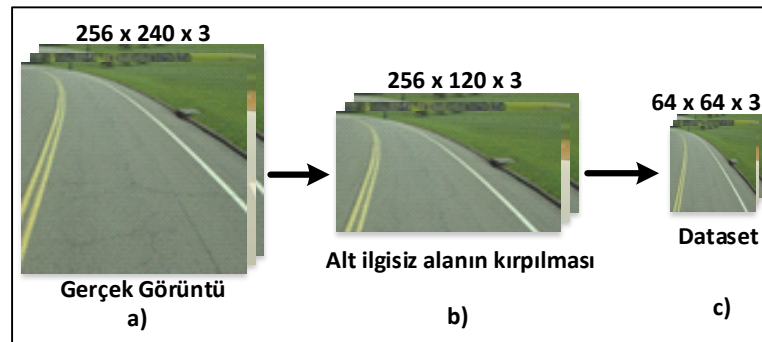


Şekil 5.16. Tehlikeli viraj için hazırlanan veri set

Şekil 5.16'da kullanılan görüntülerin 3 farklı yol tipine ait olduğu görülmektedir. Bu görüntülerin 149 adeti sol viraj yol görüntüsü, 113 adeti sağ viraj yol görüntüsü ve 325 tanesi düz yol olarak tek tek etiketlenmiştir.

5.5.1. Sağ ve Sol Tehlikeli Virajların Tespiti için Önerilen görüntü İşleme ve Tespit Yöntemi

Karayollarında bulunan tehlikeli virajların tespiti için bir yöntem geliştirilmiştir. Bu yöntemde hazırlanan veri setinin görüntü işleme teknikleri ile uygun boyutlara dönüştürülmesi ve önerilen bir sınıflandırma modelinin belirlenen oranıyla eğitilmesi ve performansını ölçmek için test edilmesi işlemi yapılmıştır. Hazırlanan veri setinde bulunan görüntülerin dönüştürme aşaması Şekil 5.17'de gösterilmiştir.



Şekil 5.17. Görüntülerin dönüşüm aşamaları

Şekil 5.17'de görüntülerin dönüşüm aşaması görülmektedir. Gerçek görüntü VP metoduna göre üst ve alttan kırıldıktan sonraki durumu Şekil 5.17(b)'de görülmektedir. Şekil 5.17(c)'de, bu görüntü geliştirilen CNN tabanlı modelde kullanılması için tekrar boyutlandırılmıştır. Bu çalışmada

CMU-VASC veritabanında bulunan 256x240 boyutundaki görüntülerin alt kısımları yüksekliğin yarı ölçüsünde kırılarak 256x120 boyutuna indirgenmiştir. Son aşamada bu görüntüler 64x64 boyutuna indirgenmiş ve değerlendirmek üzere veri setine dönüştürülmüştür. Hazırlanan veri set 6 eğitim ve test sayıları Tablo 5.22’de verilmiştir.

Tablo 5.22. Veri seti -6’da bulunan verilerin eğitim ve test olarak dağılım oranları

Yol Tipi	İmge Sayısı	Veri setindeki oranı (%)	Eğitim (%80)	Test (%20)
Sol virajlı yol	149	25.38	119	30
Normal yol	325	55.37	260	65
Sağ virajlı yol	113	19.25	90	23
Toplam	587	100	469	118

Veri seti-6’da toplamda 587 adet görüntünün sol ve sağa eğimi olmayan yol görüntülerin sayısı 325, sola virajlı yol görüntülerinin sayısı 149 ve son olarak sağ virajlı yol görüntüleri sayısı 113 olarak belirlenmiştir. Böylece CMU-VASC Veri seti toplamda 587 görüntüden oluşmaktadır.

Karayolunda bulunan virajların tespiti için derin öğrenme tabanlı bir model önerilmiştir. Bu modelin parametreleri Tablo 5.23’te verilmiştir.

Tablo 5.23. Önerilen derin öğrenme tabanlı modelin katmanları ve özellikleri

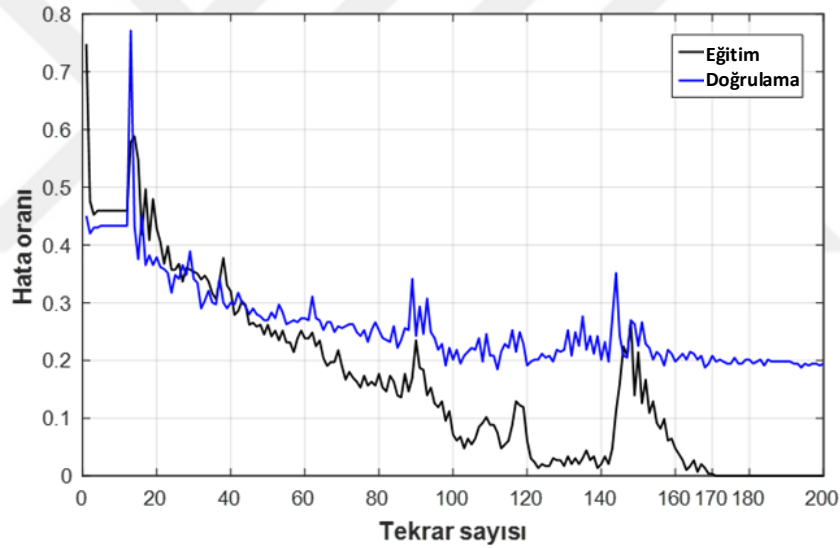
Katman Adı	Kernel Boyutu	Katman Tipi	Dış Boşluk	Adım
1 Konvolüsyon	12x12@32	-	2	3
2 Pooling	5x5	Max	0-1	3
3 Aktivasyon	-	ReLU		
4 Konvolüsyon	3x3@32	-	2	1
5 Aktivasyon	-	ReLU		
6 Pooling	3x3	Max	0-1	2
7 Konvolüsyon	5x5@32	-	2	1
8 Aktivasyon	-	ReLU		
9 Pooling	3x3	Max	0-1	2
10 Aktivasyon	-	ReLU		
11 Konvolüsyon	2x2@64	-	0	1
12 Aktivasyon	-	ReLU		
13 Konvolüsyon	1x1@64	-	0	1
14 Tam bağlı		Softmax		

Tablo 5.23’te görüldüğü gibi farklı parametrelere sahip 5 konvolüsyon, 3 pooling ve 5 ReLU aktivasyon katmanı ve sınıflandırma için Softmax katmanları kullanılmıştır. Modelin ilk konvolüsyon katmanı 12x12 boyutunda, 32 bit derinlikle ve 3 adım ile tüm görüntünün özellik çıkarımları yapılmaktadır. Bu katmandan hemen sonra gelen MaxPooling katmanı üzerinde 5x5 boyut değeri ile dolaşarak en büyük parametre alınır. 3. Katmanda olan ReLU aktivasyon katmanı kullanılmıştır. Viraj görüntülerin sol üst ve sağ üst alanları farklılık gösterirken alt kısımları benzerlik göstermektedir. Sınıf bilgisi çoğunlukla görüntünün üst kısmının doğru tespiti ile

gerçekleşeceğinden modelin ilk katmanında aşırı parametre artışı yapılmamıştır. İlk konvolüsyon katmanından sonra hemen pooling katmanı ile temel özellik parametreleri çıkarılmıştır. Hemn ardından eklenen ReLU aktivasyon katmanı ile en belirgin görüntüler elde edilmiştir. Bu şekilde devam eden model sona gelindiğinde parametre azaltma ve en belirgin özelliklerin tespiti sağlanmış olur. Tabloda belirtilen katmanların özellikleri ile tüm görüntüler işlenir ve tüm parametreler ile birlikte Softmax sınıflandırma katmanına ulaştırarak başarımlar değerlendirilmiştir.

5.5.2. Modelin Sağ ve Sol Tehlikeli Virajların Tespiti için Değerlendirilmesi

Sağ ve sol virajların tespiti için hazırlanan Veri seti-6' da bulunan 587 adet RGB kanallı görüntünün rastgele seçilmiş 469 adeti önerilen modelin eğitimi için kullanılmış ve geriye kalan 118 adet ile modelin test işlemi yapılmıştır. Yapılan bu çalışmanın performans grafiği Şekil 5.18'de görülmektedir.



Şekil 5.18. Önerilen modelin Veri seti-6 üzerindeki eğitim ve test aşamalarının performans grafiği

Şekil 5.18'i incelediğimizde, 170. tekrar sayısında eğitim tamamlanmışken, modelin performans başarımları %80.1 olarak tespit edilmiştir. Modelin eğitim aşamasında doğru bir eğilim gösterdiği ve özellikle 120. tekrar sayısından sonra eğitim aşaması neredeyse tamamlanmış durumda görülmektedir. Fakat 140. tekrar sayısından sonra modelin eğitiminde bozulma olmasına rağmen 150. tekrar sayısının eğitim değerinde artış görülmekte ve 170. tekrar sayısında eğitim aşaması başarıyla sonuçlanmıştır. Bu noktadaki testin de daha yüksek başarıma ulaşmadığı görülmektedir.

Sağ ve Sol Tehlikeli Virajların Tespitinin Literatürdeki ve Alandaki Yöntemlerle Karşılaştırılması

Yapılan bu çalışmanın literatür taramasında tespit edilen benzer çalışmalardaki başarımlarına bakıldığında, önışleme tabi tutulmadan ve özellik çıkarımı yapılmadan test işleminde oldukça yüksek bir başarıml elde edildiğı tespit edilmiştir. Tablo 5.24'te literatürdeki benzer çalışmalar ile almış olduğumuz sonucun karşılaştırması görölmektedir.

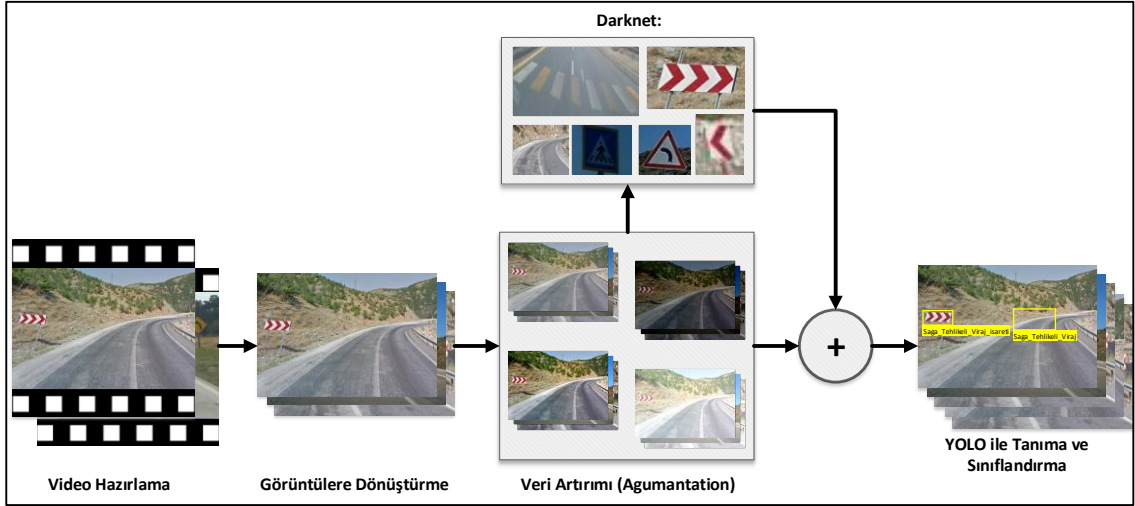
Tablo 5.24. Virajların tespiti ile ilgili literatürdeki çalışmaların sonuçların karşılaştırılması

Çalışma Adı	İmge Sayısı	Veri tabanı	Başarıml Oranı(%)	Kullanılan Yöntem
Neagoe ve Tudoran[100]	210	CMU	100	CSOM, MLP, k-Means
Surakka ve Heikkonen[101]	331	-	91/100 görüntü	Gabor + MLP
Önerilen Model	587	CMU	80.1	CNN

Literatür taramasında yol yönü ile ilgili çok çalışmanın olmadığı görölmüştür. Neagoe ve Tudoran [100] çalışmalarında 210 görüntü üzerinde çalışmış ve MLP sınıflandırıcı ile %100 başarıml elde ettiğini belirtmiştir. Surakka ve Heikkonen [101] çalışmasında ise, görüntülere gabor filtre uygulayarak özneteliklerini çıkarmış ve ardından MLP algoritmasına göre sınıflamış ve 100 görüntü üzerinden 91'ini doğru sınıflandırmıştır. Bu çalışmamızda 587 adet imgenin içinden rastgele seçilen 118 adet test görüntüleri önerilen model ile sınıflandırılmış ve başarımlı %80.1 olmuştur.

5.6. Videolarda bulunan karayolu parametrelerinin tespiti

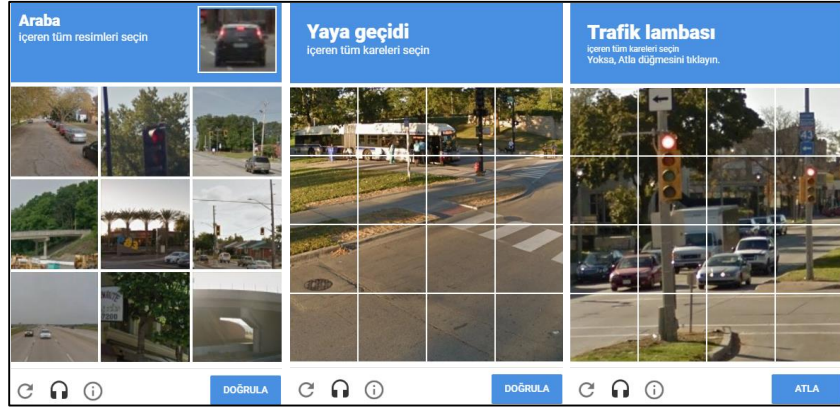
Karayolunda bulunan trafik işaretleri ve yolun gerçek durumunun tespiti için bir yöntem önerilmiştir. Bu yöntemde trafik işaretlerinin tanınması için görüntüler oluşturulmuş ve YOLO algoritmasına sunularak eğitim işlemi yapılmıştır. Diğer taraftan karayollarında bulunan ve tez çalışmasında üzerinde durduğumuz yol bileşenleri olan kavşaklar, yol ayrımları, yaya geçitleri ve sol-sağ tehlikeli virajların tanınması için bir yöntem geliştirilmiştir. Şekil 5.19'da video tabanlı tanıma sistemi için geliştirilmiş olan modelin çalışma aşamaları görölmektedir.



Şekil 5.19. Önerilen modelin Veri seti-6 üzerindeki eğitim ve test aşamalarının performans grafiği

Şekil 5.19’da önerilen yöntemin aşamaları incelendiğinde; ilk aşamada hazırlanan 2 dakikalık farklı yol ortamlarından birleştirilerek oluşturulan videolar 25(fps) çerçeve ayarı ile görüntülere dönüştürülmüştür. Görüntülerin, kavşak, yol ayrımları, yaya geçitleri, sol-sağ tehlikeli virajlar şeklinde farklı sınıflara ayırma işlemi yapılmıştır. Bu görüntülerin sayılarını artırmak için görüntülere yakın özellikler taşıyan karayolu görüntüleri toplanmış ve çeşitli görüntü işleme teknikleri (görüntüleri renklendirme, boyutlandırma gibi) uygulanarak veri artırımı (agumentation) yapılmıştır. Elde edilen görüntülerin tümü MATLAB programında bulunan image processing - annotation tool araçları ile etiketlenerek görüntüler ile birlikte DarkNet mimarisine eklenmiştir. YOLO algoritması eğitilmiş model olarak DarkNet’i kullanmakta ve bunun için belirlenen karayolu ile ilgili özel parametrelerin tespiti yapılması beklenmektedir. Bu nedenle DarkNet’in eğitim aşaması tamamlandıktan sonra oluşan ağırlık parametreleri ile YOLO algoritmasının test aşaması gerçekleşmiştir.

DarkNet mimarisi her türlü nesne, araç, gereç, simge vb ile ilgili görseller barındıran ve bunların otomatik tanınması üzerinde projeler geliştiren ImageNet [102] ortamında bulunan görüntüler kullanılarak oluşturulmuştur. Bu görseller kullanılarak modelin ağırlıkları oluşturulmuştur. Google’ın sponsorluğunda Standford üniversitesinin bir projesi olan ImageNet internet üzerinde son kullanıcıların özellikle web sitelerinde oturum açarken sisteme aşırı yüklenmeyi engellemek için Şekil 5.20’de görüldüğü gibi görüntü tabanlı sorular yöneltmekte ve kullanıcıdan gelen işaretlemeler ile görüntülerin doğruluğunu ve sınırlarını tespit etmektedir. Şekil 5.20’de görüldüğü gibi son kullanıcılara gönderilen pop up soru cevap spam önleme uygulamaları ile binlerce kullanıcıya sorular yönelterek veritabanında bulunan milyonlarca görüntüler eğitilmektedir. Bu yöntem oldukça ucuz, geniş kullanım ve global ürünler çıkarmada faydalı olmaktadır.



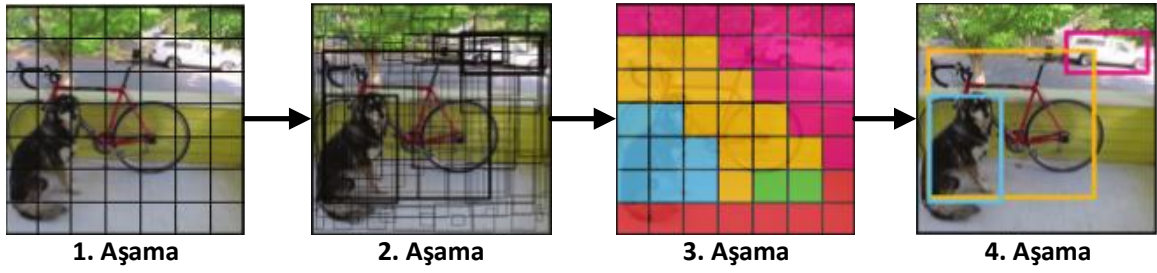
Şekil 5.20. reCAPTCHA spam önleme hizmeti

Tezin önceki uygulama bölümünde görüntülerin tanınması ve sınıflandırılarak başarımlarının hesaplaması yapılmıştır. Bu bölümde farklı karayollarında bulunan kavşak, sağ-sol yol ayrımları, yaya geçitleri, sağ-sol tehlikeli virajlar gibi önemli bölümlerin tanınması ve işaretlenmesi için bir sistem geliştirilmiştir. Geliştirilen bu sistem ile araç, karayolunda hareket halindeyken belirlenen trafik işaretlerinin tespit edilmesi, aynı zamanda yolun kendi öz niteliklerinin çıkartılması ile birlikte doğrulanması amaçlanmıştır. Çalışmada farklı öz niteliklere sahip iki karayolu görüntüsü hazırlanmıştır. Video 1 kendi imkanlarımızla Elazığ-Bingöl karayolunda 1080px çözünürlükte 1'er dakika video şeklinde herhangi özel bir düzenleme yapmadan çekmiş olduğumuz videolardır. Video 2 ise Algorry ve ark.[79] karayolunda bulunan trafik işaretlerini tanımak için geliştirmiş oldukları modeli test aşamasında kullandıkları videodur. Bu iki videoda farklı yol tipleri içeren görüntülerin %70'i çalışma konumuz olan bölgelerden tespit edilmiş, etiketlenmiş ve Darknet'e eklenmiştir. Her iki videonun çekildiği ortam, yol özellikleri ve trafik işaret parametreleri farklı olduğundan videolarda etiketleme işlemleri ayrı ayrı yapılmıştır.

Gerçek zamanlı trafik işareti ve gerçek yol bölgesini algılama ve sınıflandırma için bir görüntü işleme mimarisi geliştirilmiştir. Bu çalışma için, günümüzde oldukça popüler olan ve CNN mimarisinin bir ürünü olan, aynı zamanda hızlı ve gerçek zamanlı tanıma mimarisine sahip YOLO algoritması kullanılmıştır.

5.6.1. YOLO Algoritması Kullanarak Yol Durumu ve Trafik İşareti Tespiti

YOLO algoritması daha önceden çerçevesi belli olan görüntülerin bulunduğu bir eğitim ağırlıklarından faydalanarak verilen bir görüntüyü istenen sayıda (SxS) 2B olarak parçalara ayırır. Daha sonra parçaların çerçeveleri içine düşüp düşmediğine göre eğitim verisinde etiket hazırlanır. Her görüntü için etiketleme bir tesnor olarak hesaplanır. Şekil 5.21'de bir görüntünün YOLO algoritmasındaki işlem adımları gösterilmiştir.



Şekil 5.21. Görüntüleri YOLO algoritmasında değerlendirme aşamaları

Şekil 5.21’de ilk aşamada görüntü istenen sayıda ($S \times S$) parçalara ayrılır ve görüntü üzerinde gezinerek eğitim verisinde bir kutucuk görüntü çerçevesi merkezine düşüyorsa, o kutucuğa tekabül eden çerçeve, yani güven katsayısı için 1 değeri verilir. Diğer kutucuklar obje merkezine düşüp düşmediğine göre aynı matris üzerindeki diğerler de 1 olurlar. Bir kutucuk en fazla görüntülerde aranan nesne sayısı kadar tahmin yapabilir. Kutucuk ayrıca nesnenin görüntü içindeki koordinat ölçüsüne göre yerini ve büyüklüğünü de aynı tensor içinde tahmin etmektedir.

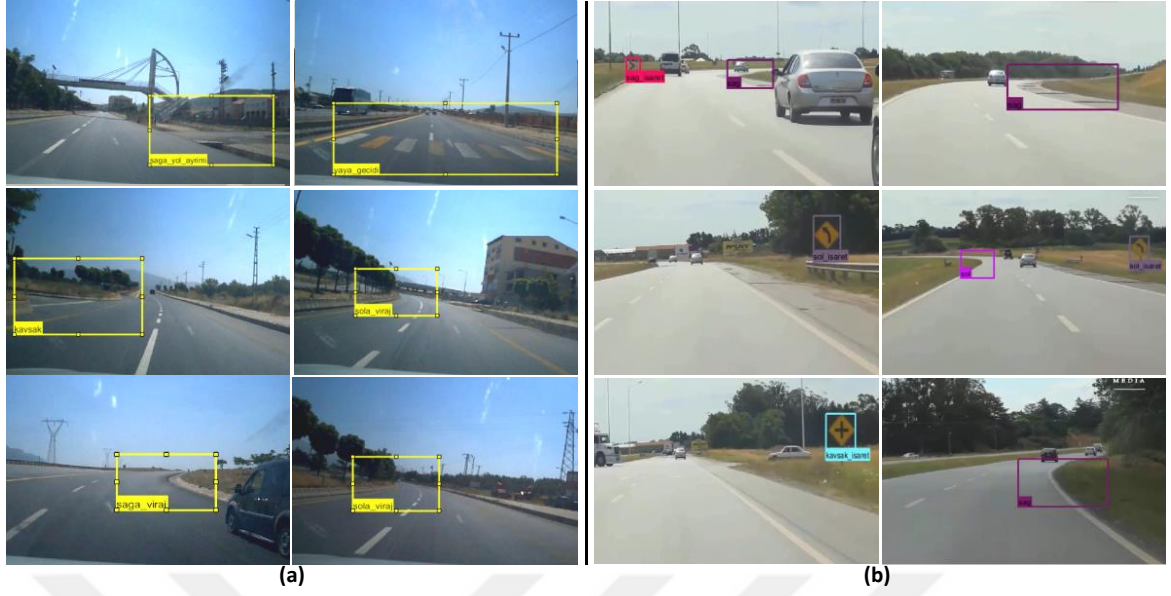
Aslında YOLO algoritması denetimli bir öğrenme süreci olarak değerlendirilebilir. YOLO algoritması diğer tanıma-sınıflandırma algoritmalarına göre oldukça hızlıdır. Bunun sebebi görüntüyü, tek seferde eğitilmiş sinir ağından geçirerek görüntüdeki tüm nesnelerin sınıfını ve koordinatlarını tahmin edebilmektir.

Nesne tespitinde YOLO algoritması, görüntüleri belirlenen ölçülere böldükten ve tensor’a dönüştürdükten sonra her çerçevede eğitilmiş nesnenin olup olmadığını belirler. Eğer nesne kutucukta varsa bunun orta noktasını, nesnenin ölçülerini ve hangi tür nesne olduğunu tespit eder. Daha sonra nesnenin orta noktasından başlayarak kutucuklar çizmektedir. Etrafına çizdiği kutucukları büyütürken dışarıya kadar çıkar ve üst kutucuk içinde nesneyi içine alacak şekilde çerçeveyi çizer. YOLO her kutucuk için ayrı tahmin vektörü oluşturur.

Tahmin vektöründe güven skoru parametresi vardır. Bu parametre ile çerçeve içinde nesnenin olup olmadığına karar vermek için 0-1 arasında değer üretir. Bu parametre nesnenin gerçekten o nesne olup olmadığını ve etrafındaki kutunun koordinatlarından ne kadar emin olduğunu gösterir.

Bu çalışmada YOLO V3 algoritması ve eğitilmiş Darknet mimari ile yol özellikleri ve ilgili trafik işaretleri tespiti yapılmıştır.

Bu tez çalışmasında özel eğitilmiş Darknet ve YOLO algoritması kullanılarak görüntülerin tanınması ve özellikle istenen nesnelerin işaretlenmesi yapılmıştır. Şekil 5.22’de yapılan çalışmanın sonucu olarak oluşan durum görülmektedir.



Şekil 5.22. Videolar üzerinde YOLO algoritması ile tanıma işlemi

Şekil 5.22’de farklı öz niteliklere sahip iki farklı video görüntülerinin olduğu görülmektedir. Şekil 5.22(a)’da Elazığ-Bingöl karayolunda çekmiş olduğumuz videonun YOLO algoritması ile değerlendirilmesi görülmektedir. Şekil 5.22(b)’de ise Algorry ve ark.[79] hazırlamış olduğu videonun YOLO algoritması ile değerlendirilmesi görülmektedir. Farklı öz niteliklere sahip görüntülerin eğitim aşaması için videolardan ayrı ayrı eğitilecek bölümler çıkarılmış ve Darknet’e yüklenerek eğitim aşaması tamamlanmıştır. Şekil 5.22(a)’da görüldüğü gibi hazırlanan veri setin Darknet+YOLO algoritması ile değerlendirme sonucunda videoda bulunan ilgili trafik işaretleri ve karayolun ilgili bölgesi tespit edilmiştir. Şekil 5.22(b)’de ise [79] çalışmasında trafik levhalarının tanınması için hazırlanan videonun Darknet+YOLO algoritması ile karayolunda bulunan ilgili bölgelerin ve işaretlerin tespiti yapılmıştır.

6. SONUÇLAR

Günümüzde otonom araç sistemleri ve ileri sürücü destek sistemleri hızla gelişmekte ve ticari anlamda ürünlerin pazarlanması için yeni teknolojiler üretilerek araçlara dahil edilmektedir. Dünya markası olmuş dev otomobil üreticileri, sürücü destek sistemleri ve otonom araç teknolojileri alanında ciddi Ar-Ge bütçeleri ayırmaktadır.

Geliştirilen otonom sistemler genellikle düzgün işaretlenmiş karayollarında test edilmekte ve yüksek başarımlar elde etmektedir. Fakat tahribata uğramış yol koşullarında yüksek başarımlar elde edilememektedir. Özellikle gelişmekte olan ülkelerde yol bakım onarımına fazla bütçe ayrılmadığından zamanla bozulmalar olmakta, trafik işaretlerinde kayıplar yaşanmaktadır. Otonom araçlar ya da sürücü destek sistemlerinin yolu tanıması için önemli olan bu işaretlerin zarar görmesi ile yanlış kararlar üretebilmektedirler. Bu sorunun önüne geçmek için geliştirilen yöntemler karayolun gerçek görüntüleri tanıma işlemi için kullanılmış ve bu görüntüler geliştirilen derin öğrenme modelleri ile test edilmiştir.

Bu tez çalışmasında, önemli yol bileşenlerinin tespiti için görüntü işleme ve derin öğrenme aşamalarını barındıran yöntemler geliştirilmiş ve gerçek veriler kullanılarak test edilmiştir. Karayollarında bulunan hem sürücü uyarı sistemleri hem de araç uyarı sistemlerinin gelişimi ve yolu tanıması için oldukça önemli olan yol tipi, yol kalitesi, kavşaklar, yaya geçitleri ve tehlikeli virajların tespiti yapay zeka temelli modeller ve görüntü işleme teknikleri kullanılarak yapılmıştır.

Görüntü işleme tekniği olarak VP yöntemine ek olarak alt bitim bölgesi ve kenar çıkarım algoritması ile ilgili bölgenin tespiti yapılmıştır. Tespit edilen bu verilerin her bir türü için geliştirilmiş olan derin öğrenme modeli eğitilmiş ve verinin bir kısmı ile de test edilmiştir. Test işlemi hem ham veri üzerinde hem de önerilmiş olan VP ve alt bitim noktası tespit edilmiş görüntüler üzerinde yapılmıştır. Önişlem olarak adlandırmış olduğumuz bu yöntem modelin başarısını yüksek oranda etkilemiştir. Geliştirmiş olduğumuz bu çalışmada görüntü işleme aşamasında, toplanan veriler daha iyi tanıma için önişlemden geçirilerek görüntüde bulunan özel bölgeler elde edilmiştir. Bu bölgeler daha sonra yapay zeka temelli derin öğrenme modellerinde test edilmiş ve tanıma işlemi yapılarak başarılı sonuçlar elde edilmiştir.

Tanıma sistemi için 7 farklı model geliştirilmiş ve bu modellerin eğitim ve test işlemlerinde daha doğru ölçümler için uygun görüntüler veri setlere dönüştürülmüştür. Geliştirilen modellerin yansırı literatürde oldukça sık kullanılan ve yüksek başarıya sahip AlexNet, LeNet, VggNet ve Darknet gibi derin öğrenme modelleri test edilmiş ve başarımları verilmiştir.

Yapılan çalışmaların video üzerindeki tepkilerini göstermek için Elazığ-Bingöl karayolu üzerinde çekilmiş bir video seçilmiş ayrıca daha önceki bir çalışmada kullanılan videolarda YOLO ve Darknet mimarisi kullanılarak test işlemi yapılmıştır. Bu çalışmayla karayolunda bulunan gerçek yol durumları ve trafik işaretleri tespit edilmiştir.

Tez çalışmasında gerçekleştirilen uygulamalarda; yol tiplerini tanıma için python dili, videolarda etiketleme ve tanıma işlemi için matlab ortamı kullanılmıştır. Görüntülerin sınıflandırılmasında 24GB RAM, 4.992 çekirdek ve 480GBps bellek bant genişliğine sahip Tesla K80 GPU kartı kullanılmıştır. Bu teknoloji kullanılarak geliştirilen sistemin eğitim aşamasının her epoch'ta bulunan batch değerine göre hızı değişmektedir. 32 batchlik bir epoch 28ms sürmüştür. 32x28ms ile bir epoch yaklaşık olarak 0.9s sürmektedir. Tanıma işleminde toplam 100 epoch ile test işlemi yapıldığından 90s sürmektedir.



ÖNERİLER

Karayollarında bulunan ve hem sürücüler hem de araçlar için önemli bölgeler olan kavşaklar, yaya geçitleri, tehlikeli virajlar, yol bozukluklarının tespiti ve yol tipinin belirlenmesi için hazırlanan sistem, otonom araç veya ileri sürücü destek sistemlerinde kullanılabilir. Geleceğin teknolojisi olarak lanse edilen otonom araçlar ticari pazarda büyük bir paya sahiptir. Otonom araç veya ileri sürücü destek sistemleri üreticileri yeni yapay zeka temelli ürünler geliştirmekte ve bu ürünlerin çoğunda bilgisayar görü teknikleri kullanmaktadır.

Yol tipi çalışması, yol sınırlarının belirlenmesi ve yol çizgilerinin varlığının denetimi için kullanılabilir. Böylece otonom araçlar ve ileri sürücü destek sistemleri sadece yol çizgilerine, işaretlerine, trafik levhalarına bağlı kalmadan yol alanının gerçek durumunu tespit edebilir. Böylelikle otonom araçlar ve ADAS sistemleri için genel bir karayolu denetleme sistemi yapılmış olur. Biz bu tez çalışmasında deneysel uygulamaları araçlar üstünde çalıştırabilecek Raspberry Pi gibi mini bilgisayarlar sistemlerine aktarılmasını planlamaktayız. Gelecek çalışmalarda gerçek zamanlı sistemler kurarak tüm yol tiplerini tanıyan genel bir ağ ve görüntü işleme teknolojisi üretmeyi amaçlamaktayız.

KAYNAKLAR

- [1] Dimitrakopoulos, G.; Demestichas, P. (2010). Intelligent transportation systems. *IEEE Vehicular Technology Magazine*, 2010. 5(1): p. 77-84.
- [2] El Faouzi, N.-E.; Leung H.; Kurian A. (2011). Data Fusion in Intelligent Transportation Systems: Progress and Challenges—A survey. *Information Fusion*, 2011. 12(1): p. 4-10.
- [3] Zhang J.; Wang F.-Y.; Wang K.; Lin W.-H.; Xu X.; Chen C. (2011). Data-driven intelligent transportation systems: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 2011. 12(4): p. 1624-1639.
- [4] International, K. 2019 Autonomous Vehicles Readiness, <https://assets.kpmg/content/dam/kpmg/xx/pdf/2019/02/2019-autonomous-vehicles-readiness-index.pdf>. Eriřim: 24 Kasım 2019
- [5] Group, M.L. Accidents Caused by Dangerous Intersections, <http://www.hg.org/article.asp?id=7652>. Eriřim: 24 Kasım 2019
- [6] Karayolları Genel Müdürlüğü, (1985). Karayolu Trafik Güvenlięinin Sağlanması Yönünden, Yolun Yapısında Yapılacak Her Türlü Çalışmalarda Alınacak Tedbirler ile Karayolu Dışında, Kenarında veya Üzerindeki Diğer Levhalar, Işıklar ve İşaretlemeler Hakkında Yönetmelik, <https://www.mevzuat.gov.tr/Metin.Aspx?MevzuatKod=7.5.4701&MevzuatIliski=0&sourceXmlSearch=>. Eriřim: 24 Kasım 2019
- [7] Huval, B. ve dięerleri, (2015). An empirical evaluation of deep learning on highway driving. *arXiv preprint arXiv:1504.01716*, 2015.
- [8] Ding, W.; Li, Y. (2015). Efficient Vanishing Point Detection Method in Complex Urban Road Environments. *IET Computer Vision*, 2015. 9(4): p. 549-558.
- [9] Li, Y.; Ding, W.; Zhang, X.G.; Ju, Z. (2016). Road detection algorithm for autonomous navigation systems based on dark channel prior and vanishing point in complex road scenes. *Robotics and Autonomous Systems*, 2016. 85: p. 1-11.
- [10] Maeda, H.; Sekimoto, Y.; Seto, T.; Kashiya, T.; Omata, H. (2018). Road Damage Detection and Classification Using Deep Neural Networks with Smartphone Images. *Computer-Aided Civil and Infrastructure Engineering*, 2018. 33(12): p. 1127-1141.
- [11] Tumen, V.; Yildirim O.; Ergen B. (2018). Recognition of Road Type and Quality for Advanced Driver Assistance Systems with Deep Learning. *Elektronika ir Elektrotehnika*, 2018. 24(6): p. 67-74.
- [12] Bhatt, D.; Sodhi, D.; Pal, A.; Balasubramanian, V.; Krishna, M. (2017). Have i reached the intersection: A deep learning-based approach for intersection detection from monocular cameras. *Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on*. 2017. IEEE.
- [13] Mukhija, P., Tourani, S.; Krishna, K.M. (2012). Outdoor Intersection Detection for Autonomous Exploration. *Intelligent Transportation Systems (ITSC), 15th International IEEE Conference on*. 2012. IEEE.
- [14] Rebai, K.; Achour, N.; Azouaoui, O. (2014). Road Intersection Detection and Classification Using Hierarchical SVM Classifier. *Advanced Robotics*, 2014. 28(14): p. 929-941.
- [15] Wang, J.; Wang, C.; Song, X.; Raghavanc, V. (2017). Automatic Intersection and Traffic Rule Detection by Mining Motor-Vehicle GPS Trajectories. *Computers, Environment and Urban Systems*, 2017. 64: p. 19-29.
- [16] Zhang, Y.; Zhang J.; Li, T.; Sun, K. (2016). Road Extraction and Intersection Detection Based on Tensor Voting. *Geoscience and Remote Sensing Symposium (IGARSS)*, 2016 IEEE.
- [17] Zhou, Q.; Li, Z. (2015). Experimental Analysis of Various Types of Road Intersections for Interchange Detection. *Transactions in GIS*, 2015. 19(1): p. 19-41.

- [18] Berriel, R.F.; Ross, F.S.; Souza A.F.; Santos, T.O.; (2017). Automatic Large-Scale Data Acquisition via Crowdsourcing for Crosswalk Classification: A deep learning approach. *Computers & Graphics*, 2017. 68: p. 32-42.
- [19] Chen, Z.S.; Zhang, D.F.; (2017). An Effective Detection Algorithm of Zebra-Crossing. *International Conference on Electrical and Information Technologies for Rail Transportation*. 2017. Springer.
- [20] Hariyono, J.; Jo, K.-H. (2017). Detection of pedestrian crossing road: A study on pedestrian pose recognition. *Neurocomputing*, 2017. 234: p. 144-153.
- [21] Radványi, M.; Varga, B.; Karacs K. (2010). Advanced Crosswalk Detection for The Bionic Eyeglass. *Cellular Nanoscale Networks and Their Applications (CNNA)*, 12th International Workshop on. 2010. IEEE.
- [22] Riveiro, B.; Riveiroa, B.; González-Jorgeb, H.; Martínez-Sánchez, J.; Díaz-Vilariño, L.; Arias, P. (2014). Automatic Detection of Zebra Crossings from Mobile LiDAR Data. *Optics & Laser Technology*, 2015. 70: p. 63-70.
- [23] Wang, S., Pan, H.; Zhang, C.; Tian, Y. (2014). RGB-D Image-Based Detection of Stairs, Pedestrian Crosswalks and Traffic Signs. *Journal of Visual Communication and Image Representation*, 2014. 25(2): p. 263-272.
- [24] Ajansı, A. Dünya Nüfusu, <https://www.aa.com.tr/tr/dunya/dunya-nufusu-2055te-10-milyari-bulabilir/1527644>. Erişim: 24 Kasım 2019
- [25] Taylor, P.; Anand, S.S.; Griffiths, N.; Adamu-Fika, F.; Dunoyer A.; Popham T. (2012). Road Type Classification Through Data Mining. *The 4th International Conference on Automotive User Interfaces and Interactive Vehicular Applications*. 2012. ACM.
- [26] Kenanoğlu, M.E.; Aydın M. (2018). Akıllı Ulaşım Sistemlerinin Dışsalılık Bağlamında Değerlendirilmesi: Seçilmiş Ülke Uygulamaları. *Çanakkale Onsekiz Mart Üniversitesi Uluslararası Sosyal Bilimler Dergisi*. 3(2): p. 363-387.
- [27] Yılmaz, Ö. (2012). *Karayolu ulaşımında akıllı ulaştırma sistemleri*, Uzmanlık tezi, Kalkınma Bakanlığı.
- [28] Tufan, H., *Akıllı Ulaşım Sistemleri Uygulamaları ve Türkiye için bir AUS Mimarisi Önerisi*. Ulaştırma ve Haberleşme Uzmanlığı Tezi, TC Ulaştırma Denizcilik ve Haberleşme Bakanlığı, 2014.
- [29] Yaman, H., (2014) Akıllı Ulaşım Sistemleri tanımı ve kapsamı. Akıllı Ulaşım ve Güvenlik Sistemleri, 2014: p. 54.
- [30] Ulaştırma ve Haberleşme Bakanlığı, Ulusal Akıllı Ulaşım Sistemleri Strateji Belgesi (2014-2023) ve Eki Eylem Planı (2014-2016). 2014, Ankara.
- [31] An, S.H.; Lee B.H.; Shin D.R. (2011). A Survey of Intelligent Transportation Systems. *Third International Conference on Computational Intelligence, Communication Systems and Networks*. 2011. IEEE.
- [32] Pomerleau, D.A. (1989). Alvin: An Autonomous Land Vehicle in A Neural Network. *Advances In Neural Information Processing Systems*. 1989.
- [33] Nordhoff, S., B.; Happee R. (2016). Conceptual Model to Explain, Predict, And Improve User Acceptance of Driverless Podlike Vehicles. *Transportation research record*, 2016. 2602(1): p. 60-67.
- [34] Dresner, K.; Stone, P.; Multiagent, A. (2008). Approach to Autonomous Intersection Management. *Journal of artificial intelligence research*, 2008. 31: p. 591-656.
- [35] Kyriakidis, M.; Happee R.; Winter, J.C. (2015). Public Opinion on Automated Driving: Results of An International Questionnaire Among 5000 Respondents. *Transportation Research Part F: Traffic Psychology and Behaviour*, 2015. 32: p. 127-140.
- [36] Schoettle, B.; Sivak, M. (2014). A Survey of Public Opinion about Autonomous and Self-Driving Vehicles in The US, the UK, and Australia. 2014, *University of Michigan, Ann Arbor, Transportation Research Institute*.

- [37] Lin, P.-S.; Wang, Z.; Guo R. (2016). Impact of Connected Vehicles and Autonomous Vehicles on Future Transportation. *Bridging the East and West*, 2016: p. 46-53.
- [38] Evans, A.N.; Liu, X.U. (2006). A Morphological Gradient Approach to Color Edge Detection. *IEEE Transactions on Image Processing*, 2006. 15(6): p. 1454-1463.
- [39] Gao, W., Zhang, X.; Yang, L.; Liu, H.; (2010). An improved Sobel Edge Detection. *3rd International Conference on Computer Science and Information Technology*. 2010. IEEE.
- [40] Chaple, G.N.; Daruwala, R.; Gofane, M.S. (2015). Comparisons of Robert, Prewitt, Sobel operator based edge detection methods for real time uses on FPGA. *International Conference on Technologies for Sustainable Development (ICTSD)*. 2015. IEEE.
- [41] Ma, M.;Guo, J.K. (2004). Automatic Image Cropping for Mobile Device with Built-In Camera. in *First IEEE Consumer Communications and Networking Conference, (CCNC)*. 2004, IEEE.
- [42] Yan, J., Lin, S.; Bing Kang, S.; Tang, X. (2013). Learning The Change for Automatic Image Cropping. *IEEE Conference on Computer Vision and Pattern Recognition*. 2013.
- [43] He, Y.; Wang, H.; Zhang, B. (2004). Color-Based Road Detection in Urban Traffic Scenes. *IEEE Transactions On Intelligent Transportation Systems*, 2004. 5(4): p. 309-318.
- [44] Sun, T.-Y.; Tsai S.-J.; Chan V. (2006). HSI Color Model Based Lane-Marking Detection. *IEEE Intelligent Transportation Systems Conference*. 2006. IEEE.
- [45] Southall, B.; Taylor, C.J. (2001). Stochastic Road Shape Estimation. *Eighth IEEE International Conference on Computer Vision*. ICCV 2001. IEEE.
- [46] Freeman, W.T; Adelson, E.H. (1991). The Design and Use of Steerable Filters. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 1991(9): p. 891-906.
- [47] McCall, J.C.; Trivedi, M.M. (2006). Video-Based Lane Estimation and Tracking for Driver Assistance: Survey, System, and Evaluation. *IEEE Transactions on Intelligent Transportation Systems*, 2006.
- [48] Russell, S.J.; Norvig, P. (2016). Artificial Intelligence: A Modern Approach. *Pearson Education Limited*. 2016: Malaysia;
- [49] Bengio, Y. (2009). Learning Deep Architectures for AI. *Foundations and trends in Machine Learning*, 2009. 2(1): p. 1-127.
- [50] Song, H.A.; Lee S.-Y. (2013). Hierarchical Representation using NMF. *International Conference on Neural Information Processing*. 2013. Springer.
- [51] Grefenstette, E.; Blunsom, P.; Freitas N.; Hermann K.M. (2014). A Deep Architecture for Semantic Parsing. arXiv Preprint arXiv:1404.7296, 2014.
- [52] Chao, W.-L. (2011). Machine Learning Tutorial. National Taiwan University, 2011.
- [53] Yabanova, İ.; Yumurtacı, M. (2018). Classification of Dynamic Egg Weight Using Support Vector Machine. *Journal of the Faculty of Engineering and Architecture of Gazi University*, 2018. 33(2): p. 393-402.
- [54] Adaminejad, H.; Shayegani, I.; Ohammadi, M.; Farjah, (2015). E., An Algorithm for Power Quality Events Core Vector Machine Based Classification. *The Modares Journal of Electrical Engineering*, 12(4), 50-59.
- [55] Aci, M.; Avci, M.; Aci, C.; (2017). Reducing Simulation Duration Of Carbon Nanotube Using Support Vector Regression Method. *Journal of the Faculty of Engineering and Architecture of Gazi University*, 2017. 32(3): p. 901-907.
- [56] Cheng, L.; Bao, W. (2014). Remote Sensing Image Classification Based on Optimized Support Vector Machine. *Telkomnika Indonesian Journal of Electrical Engineering*, 2014. 12(2): p. 1037-1045.
- [57] Murphy, K.; Ginnekena, B-van.; Schilham, A.M.R.; Hoop, B.J.; Gietema, H.A.; Prokop, M. (2009). A Large-Scale Evaluation of Automatic Pulmonary Nodule Detection in Chest CT Using Local Image Features and K-Nearest-Neighbour Classification. *Medical Image Analysis*, 2009. 13(5): p. 757-770.

- [58] Mirjalili, S. (2015). How Effective is The Grey Wolf Optimizer in Training Multi-Layer Perceptrons. *Applied Intelligence*, 2015. 43(1): p. 150-161.
- [59] Schmidhuber, J. (2015). Deep Learning in Neural Networks: An overview. *Neural networks*, 2015. 61: p. 85-117.
- [60] Yu, D.; Deng, L. (2010). Deep Learning and Its Applications to Signal and Information Processing. *IEEE Signal Processing Magazine*, 2010. 28(1): p. 145-154.
- [61] Hinton, G., Deng, L.; Yu, D.; Dahl, G. Mohamed, A. (2012). A Deep Neural Networks for Acoustic Modeling in Speech Recognition. *IEEE Signal Processing Magazine*, 2012. 29.
- [62] Arel, I.; Rose, D.C.; Karnowski, T.P. (2010). Deep Machine Learning-A New Frontier in Artificial Intelligence Research. *IEEE Computational Intelligence Magazine*, 2010. 5(4): p. 13-18.
- [63] Deng, L. Yu, D. (2014). Deep Learning: Methods and Applications. *Foundations and Trends in Signal Processing*, 2014. 7(3-4): p. 197-387.
- [64] Bengio, Y.; Courville, A.; Vincent, P. (2013). Representation Learning: A Review and New Perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2013. 35(8): p. 1798-1828.
- [65] Elman, J.L., (1990). Finding structure in time. *Cognitive science*, 1990. 14(2): p. 179-211.
- [66] Hochreiter, S.; Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, 1997. 9(8): p. 1735-1780.
- [67] Doğan, F.; Türkoğlu, İ. (2019). Derin Öğrenme Modelleri ve Uygulama Alanlarına İlişkin Bir Derleme. *DÜMF Mühendislik Dergisi*. 10(2): p. 409-445.
- [68] He, K., Zhang, X.; Ren, S.; Sun. J. (2016). Deep Residual Learning for Image Recognition. *The IEEE Conference on Computer Vision and Pattern Recognition*. 2016.
- [69] Sabour, S.; Frosst, N.; Hinton G.E. (2017). Dynamic Routing Between Capsules. In *Advances In Neural Information Processing Systems*. 2017.
- [70] Simonyan, K.; Zisserman, A. (2014). Very Deep Convolutional Networks For Large-Scale Image Recognition. *arXiv preprint arXiv:1409.1556*, 2014.
- [71] Bengio, Y. (2012). Practical Recommendations for Gradient-Based Training of Deep Architectures. In *Neural Networks: Tricks Of The Trade*. 2012, Springer. p. 437-478.
- [72] Derin Öğrenme Uygulamalarında En Sık kullanılan Hiper-parametreler. Available from: http://colinraffel.com/wiki/neural_network_hyperparameters, Erişim: 24 Kasım 2019
- [73] Çarkacı, N. Derin Öğrenme Uygulamalarında En Sık kullanılan Hiper-parametreler. <https://medium.com/deep-learning-turkiye/derin-ogrenme-uygulamalarinda-en-sik-kullanilan-hiper-parametreler-ece8e9125c4>. Erişim: 24 Kasım 2019
- [74] Krizhevsky, A.; Sutskever, I.; Hinton, G.E. (2012). Imagenet Classification With Deep Convolutional Neural Networks. *Advances in Neural Information Processing Systems*. 2012.
- [75] LeCun, Y., Bottou, L.; Bengio, Y.; Haffner, P. (1998). Gradient-Based Learning Applied to Document Recognition. *Proceedings of the IEEE*, 1998. 86(11): p. 2278-2324.
- [76] Redmon, J.; Divvala, S.; Girshick, R. (2016). You Only Look Once: Unified, Real-Time Object Detection. *The IEEE Conference on Computer Vision and Pattern Recognition*. 2016.
- [77] Redmon, J.; Farhadi, A. (2017). YOLO9000: Better, Faster, Stronger. *IEEE Conference on Computer Vision and Pattern Recognition*. 2017.
- [78] Redmon, J.; Farhadi, A. (2018). Yolov3: An Incremental Improvement. *arXiv preprint arXiv:1804.02767*, 2018.
- [79] Algotry, A.M.; García A.G.; Wofmann, A.G. (2017). Real-Time Object Detection and Classification of Small and Similar Figures in Image Processing. *International Conference on Computational Science and Computational Intelligence (CSCI)*. 2017. IEEE.

- [80] CMU-VASC, Road dataset. (2019). <https://www.scs.cmu.edu/tags/vasc>, Erişim: 24 Kasım 2019
- [81] Bui, T.H., Nobuyama E., T. Saitoh, A. (2013). Texture-Based Local Soft Voting Method for Vanishing Point Detection from a Single Road Image. *IEEE Transactions on Information and Systems*, 2013. 96(3): p. 690-698.
- [82] Bui, T.H.; Saitoh, T.; Nobuyama, E. (2014). Vanishing Point-Based Road Detection for General Road Images. *IEEE Transactions on Information and Systems*, 2014. 97(3): p. 618-621.
- [83] Hieu, B.T. (2013). Road Area Extraction Based on Vanishing Point Detection and Texture Orientation Estimation. 2013, *Kyushu Institute of Technology*.
- [84] Kong, H., Audibert, J.-Y.; Ponce J. (2010). General Road Detection from a Single Image. *IEEE Transactions on Image Processing*, 2010. 19(8): p. 2211-2220.
- [85] Rasmussen, C., (2008). Roadcompass: Following Rural Roads with Vision+ Ladar Using Vanishing Point Tracking. *Autonomous Robots*,. 25(3): p. 205-229.
- [86] Moon, Y.Y.; Geem Z.W.; Han, G.-T. (2018). Vanishing Point Detection for Self-Driving Car Using Harmony Search Algorithm. *Swarm and Evolutionary Computation*, 2018.
- [87] Wang, S.; Khushaba, R.; Kodagoda, S. (2012). Towards Speed-Independent Road-Type Classification. *12th International Conference on Control Automation Robotics & Vision (ICARCV)*. 2012. IEEE.
- [88] Seeger, C.; Muller, A.; Schwarz, L.; Manz, M. (2016). Towards Road Type Classification with Occupancy Grids. *IEEE Intelligent Vehicles Symposium (IV) Workshop: DeepDriving-Learning Representations for Intelligent Vehicles*. 2016.
- [89] Slavkovikj, V.; Verstockt S.; De Neve W.; Hoecke S. V.; Walle, R.V. (2014). Image-Based Road Type Classification. *22nd International Conference on Pattern Recognition*. 2014. IEEE.
- [90] Ryu, S.-K.; Kim T.; Kim Y.-R. (2015). Image-Based Pothole Detection System for ITS Service and Road Management System. *Mathematical Problems in Engineering*, 2015.
- [91] Jo, Y.; Ryu, S. (2015). Pothole Detection System Using A Black-Box Camera. *Sensors*, 2015. 15(11): p. 29316-29331.
- [92] Koch, C.; Brilakis, I. (2011). Pothole Detection in Asphalt Pavement Images. *Advanced Engineering Informatics*, 2011. 25(3): p. 507-515.
- [93] Chen, K.; Lu, M.; Tan, G.; Wu, J. (2013). CRSM: Crowdsourcing based road surface monitoring. *IEEE 10th International Conference on High Performance Computing and Communications & 2013 IEEE International Conference on Embedded and Ubiquitous Computing*. 2013. IEEE.
- [94] Mednis, A.; Strazdins, G.; Zviedris, R.; Kanonirs, G.; Selavo, L. (2011). Real Time Pothole Detection Using Android Smartphones with Accelerometers. *International Conference On Distributed Computing In Sensor Systems and Workshops (DCOSS)*. 2011. IEEE.
- [95] Madli, R.; Hebbar S.; Pattar P.; Golla, V. (2015). Automatic Detection and Notification of Potholes And Humps on Roads to Aid Drivers. *IEEE Sensors Journal*, 15(8): p. 4313-4318.
- [96] Talo, M.; Baloglu, UB.; Yıldırım Ö.; Acharya UR. (2019), Application Of Deep Transfer Learning For Automated Brain Abnormality Classification Using MR Images. *Cognitive Systems Research*, 2019. 54: p. 176-188.
- [97] Yi, Y.; Hao L.; Hao, Z.; Songtian, S.; Ningyi L. (2017). Song Wenjie Intersection Scan Model and Probability Inference For Vision Based Small-Scale Urban Intersection Detection. *Intelligent Vehicles Symposium (IV)*, 2017 IEEE.
- [98] Ballardini, A.L.; Cattaneo, D.; Fontana, S.; Sorrenti, D.G. (2017). An Online Probabilistic Road Intersection Detector. *IEEE International Conference on Robotics and Automation (ICRA)*. 2017. IEEE.

- [99] Berriel, R.F.; Lopes, A.T.; Souza A. F.; Santos, T.O. (2017). Deep Learning-Based Large-Scale Automatic Satellite Crosswalk Classification. *IEEE Geoscience and Remote Sensing Letters*, 2017. 14(9): p. 1513-1517.
- [100] Neagoe, V.-E.; Tudoran, C.-T. (2011). A Neural Machine Vision Model for Road Detection in Autonomous Navigation. *University Politehnica of Bucharest Scientific Bulletin, Series C: Electrical Engineering*, 2011. 73(2): p. 167-178.
- [101] Surakka, M.; Heikkonen, J. (1995). Road Direction Detection Based on Gabor Filters and Neural Networks, *Intelligent Autonomous Vehicle. 1995, Elsevier*. p. 283-288.
- [102] ImageNet. ImageNet. www.image-net.org, Erişim Tarihi: 20 Kasım.2019



ÖZGEÇMİŞ

Vedat TÜMEN

KİŞİSEL BİLGİLER

Doğum Yeri	: Bitlis-Tatvan
Doğum Yılı	: 1985
Uyruğu	: Türkiye Cumhuriyeti
Adres	: Bitlis Eren Üniversitesi, Rahva Yerleşkesi, Mühendislik - Mimarlık Fakültesi
E-posta	: vtumen@beu.edu.tr.
Yabancı Diller	: İngilizce (Düzey: YDS:65)

EĞİTİM BİLGİLERİ

Yüksek Lisans	: “Veri madenciliği yöntemleri ile güç kalitesi verilerinin incelenmesi”, Tunceli Üniversitesi, Fen Bilimleri Enstitüsü, Elektrik-Elektronik Mühendisliği Anabilim Dalı, 2013, Danışman: Doç Dr. Hüseyin ERİŞTİ
Lisans	: Muznür Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği, 2016
Lisans	: 19 Mayıs Üniversitesi, Eğitim Fakültesi, Bilgisayar ve Öğt. Tek. Bölümü, 2006
Lise	: Tuzla Anadolu Teknik Lisesi, İstanbul, 2002

ARAŞTIRMA DENEYİMİ

- ✓ Kullandığınız laboratuvar cihazları, bildiğiniz deney sistemleri gibi bilgiler veriniz
- ✓ Bildiğiniz bilgisayar programlama dillerini yazınız (C#, Java, MATLAB, Python)
- ✓ Kullanabildiğiniz bilgisayar programlarını yazınız (Netbeans, Visual Studio, Matlab)

İŞ DENEYİMİ

2019 – ...	: Bitlis Eren Üniversitesi, Bilgisayar Mühendisliği, Öğretim Görevlisi
2009 – 2019	: Munzur Üniversitesi, Öğretim Görevlisi
2007 – 2009	: Şanlıurfa MEB, Öğretmen

AKADEMİK FAALİYETLER

Makaleler:

1. Tümen, V., Ergen, B. (2019). Intersections and crosswalk detection using deep learning and image processing techniques. Physica A: Statistical Mechanics and its Applications, doi.org/10.1016/j.physa.2019.123510
2. Tümen V., Yıldırım Ö., Ergen B. (2018). Recognition of road type and quality for advanced driver assistance systems with deep learning. Elektronika Ir Elektrotechnika, 24(6), 67-74., doi.org/10.5755/j01.eie.24.6.22293
3. Algur, Ö., Tümen, V., Yıldırım, Ö. (2018). Dış Ortam Görüntülerindeki İnsan Hareketlerinin Hibrit Derin Öğrenme Yöntemleri Kullanarak Sınıflandırılması. Fırat Üniversitesi Mühendislik Bilimleri Dergisi, 30(3), 121-129.
4. Tümen, V., Yıldırım Ö. & Ergen, B. (2017). A Convolutional Neural Network Model for Road Flow Direction Detection. ASYU dergisi

Bildiriler:

1. Tümen, V., Yıldırım, Ö., Ergen, B. (2018). Detection of driver drowsiness in driving environment using deep learning methods. In 2018 Electric Electronics, Computer Science,

Biomedical Engineerings' Meeting (EBBT) (pp. 1-5). IEEE. DOI: 10.1109/EBBT.2018.8391427

2. Tümen, V., Söylemez, Ö. F., Ergen, B. (2017). Facial emotion recognition on a dataset using convolutional neural network. In 2017 International Artificial Intelligence and Data Processing Symposium (IDAP) (pp. 1-5). IEEE. DOI:10.1109/IDAP.2017.8090281

Projeler:

1. Föbap Proje No: MF.19.50, “Derin Öğrenme Yöntemleri Kullanılarak Akıllı Şehir Ulaşım Uygulamaları Geliştirmek”, Proje bütçesi: 17,853.52 TL, Proje Bap tarafından kabul edildi.

