



**GERÇEK SÜRÜCÜ DAVRANIŞLARINDAN ŞERİT DEĞİŞTİRME
EYLEMİNİN YSA İLE TAHMİN MODELİ**

Ali GÜNEŞ

**DOKTORA TEZİ
KAZALARIN ÇEVRESEL VE TEKNİK ARAŞTIRMASI
ANA BİLİM DALI**

**GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

KASIM 2022

ETİK BEYAN

Gazi Üniversitesi Fen Bilimleri Enstitüsü Tez Yazım Kurallarına uygun olarak hazırladığım bu tez çalışmasında;

- Tez içinde sunduğum verileri, bilgileri ve dokümanları akademik ve etik kurallar çerçevesinde elde ettiğimi,
- Tüm bilgi, belge, değerlendirme ve sonuçları bilimsel etik ve ahlak kurallarına uygun olarak sunduğumu,
- Tez çalışmasında yararlandığım eserlerin tümüne uygun atıfta bulunarak kaynak gösterdiğimi,
- Kullanılan verilerde herhangi bir değişiklik yapmadığımı,
- Bu tezde sunduğum çalışmanın özgün olduğunu,

bildirir, aksi bir durumda aleyhime doğabilecek tüm hak kayıplarını kabullendiğimi beyan ederim.

Ali GÜNEŞ

25/11/2022

GERÇEK SÜRÜCÜ DAVRANIŞLARINDAN ŞERİT DEĞİŞTİRME EYLEMİNİN YSA İLE TAHMİN MODELİ

(Doktora Tezi)

Ali GÜNEŞ

GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

Kasım 2022

ÖZET

Trafik kazaları sürücü, yol, çevre ve araç özelliklerine bağlı olarak bir veya birden fazla etkenin birleşimiyle meydana gelir. Kazaların oluşumunda en büyük etken sürücülerdir. Sürücülerin trafik güvenliğini riske sokan davranışlarından birisi de hatalı şerit değişimidir. Sürücülerin şerit değiştirme niyetleri eyleme dönüşmeden önce belirlenebilirse bu hatanın azaltılmasında büyük katkı sağlanabilecektir. Bu çalışmada sürücünün öndeki aracı geçmeden 3 s kadar önceki görsel hareketleri, aracın hız ve çevresel şartları ve öndeki araç ile olan mesafe dikkate alınarak bir model geliştirilmiştir. Oluşturulan model yapay sinir ağlarında iki çıktılı sınıflandırma modeli olarak kurulmuştur. Gerçek yol şartlarında 8 katılımcı ile yapılan yol testlerinde elde edilen veriler ışığında veri seti elde edilmiştir. Veri setinin oluşumunda öndeki aracın tespiti ve araç ile olan mesafesi, sürücü yüz ve göz tespiti için görüntü işleme yöntemlerinden, aracın hızı ve direksiyon açısı bilgisi için ise taşıt CAN Bus sisteminden bilgi alınmıştır. Bu veriler ile hazırlanan yapay sinir ağı sınıflandırma modeli ile %93,1 doğruluk oranıyla sürücünün öndeki aracı geçme niyetinin olup olmadığının tahmini gerçekleştirilmiştir. Oluşturulan modelin diğer makine öğrenmesi, Naive Bayes, en yakın komşular algoritması, destek vektörleri algoritması, karar ağaçları algoritmalarına göre daha başarılı sonuç verdiği görülmüştür. Ayrıca bu çalışmayla kazaların oluşmasını önlemek adına yapılan araştırmalara da katkı sağlanmıştır.

Bilim Kodu : 93006

Anahtar Kelimeler : Sürücü davranışı, yapay sinir ağları, trafik kazaları, şerit değiştirme, görüntü işleme, nesne tespiti

Sayfa Adedi : 105

Danışman : Prof. Dr. Can ÇINAR

PREDICTION MODEL OF LANE CHANGE ACTION FROM REAL DRIVER
BEHAVIORS WITH ANN

(PhD. Thesis)

Ali GÜNEŞ

GAZİ UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

November 2022

ABSTRACT

Traffic accidents occur with the combination of one or more factors depending on the driver, road, environment and vehicle characteristics. The biggest factor in the occurrence of accidents is drivers. One of the behaviors of drivers that puts traffic safety at risk is incorrect lane change behaviors. If the lane change intentions can be determined before they turn into action, a great contribution can be made to reduce this fault. In this study, a model was developed considering the visual movements of the driver 3 s before passing the vehicle in front, the speed and environmental conditions of the vehicle, and the distance from the vehicle in front. The developed model is established as a two-output classification model in artificial neural networks. The data set was obtained in the light of the data obtained in the driving tests conducted with 8 participants in real road conditions. In the formation of the data set, information was obtained from image processing methods for the detection of the vehicle in front and its distance from the vehicle, driver face and eye detection, and from the vehicle CAN Bus system for the vehicle's speed and steering angle information. With the artificial neural network classification model prepared with these data, it was estimated whether the driver intended to pass the vehicle in front with an accuracy rate of 93.1%. It has been seen that the created model gives more successful results than other machine learning, Naive Bayes, nearest neighbors algorithm, support vectors algorithm, and decision tree algorithms. In addition, this study contributed to the researches to prevent the accidents.

Science Code : 93006

Key Words : Driver behavior, artificial neural networks, traffic accidents, lane change, image processing, object detection

Page Number : 105

Supervisor : Prof. Dr. Can ÇINAR

TEŞEKKÜR

Çalışmalarım boyunca beni her zaman destekleyen ve yönlendiren tez danışmanım Sayın Prof. Dr. Can ÇINAR'a sonsuz teşekkürlerimi sunarım.

İÇİNDEKİLER

	Sayfa
ÖZET	iv
ABSTRACT.....	v
İÇİNDEKİLER	vii
ÇİZELGELERİN LİSTESİ.....	x
ŞEKİLLERİN LİSTESİ.....	xi
SİMGELER VE KISALTMALAR.....	xv
1. GİRİŞ.....	1
2. TRAFİK KAZALARI.....	3
2.1. Trafik Kazalarında Sürücü Etkisi.....	6
3. GÖRÜNTÜ İŞLEME	11
3.1. Görüntü İşleme Temel Bileşenleri	11
3.1.1 Analog görüntü	12
3.1.2. Sayısal (Dijital) görüntü.....	13
3.2. Görüntü İşleme Teknikleri	14
3.2.1. Morfolojik işlemler	14
3.2.2. Filtreleme işlemleri	17
3.2.3. Kenar belirleme.....	19
3.2.4. Geometrik işlemler.....	24
3.2.5. Haar Cascade nesne takibi ve Viola Jones nesne dedektörü.....	27
3.2.6. Yolov3 ile nesne tanıma.....	31
4. YAPAY SİNİR AĞLARI VE MAKİNE ÖĞRENMESİ.....	33
4.1. Yapay Sinir Hücresi	33

	Sayfa
4.2. Yapay Sinir Ağlarının Yapısı.....	36
4.3. Yapay Sinir Ağlarının Çalışması	37
4.4. YSA'nın Sınıflandırılması	37
4.4.1. YSA'nın yapılarına göre sınıflandırılması	38
4.4.2. YSA'nın öğrenme algoritmalarına göre sınıflandırılması.....	38
4.4.3. YSA temel öğrenme kuralları	38
4.4.4. YSA'da en çok kullanılan modeller ve öğrenme kuralları.....	39
4.5. Çok Katmanlı Ağlar	40
4.5.1. Çok katmanlı ağların yapısı	40
4.5.2. Çok katmanlı ağların öğrenme kuralı.....	41
4.5.3. Çok katmanlı ağının çalışma prosedürü.....	44
4.6. Makine Öğrenmesi ile Sınıflandırma Yöntemleri.....	46
4.6.1. Naive Bayes sınıflandırma algoritması	46
4.6.2. En yakın komşu (KNN) sınıflandırma algoritması	47
4.6.3. Destek vektör makineleri (SVM) sınıflandırma algoritması.....	48
4.6.4. Karar ağaçları sınıflandırma algoritması.....	48
5. LİTERATÜR ÖZETLERİ	51
6. MATERYAL VE METOT	71
6.1. Direksiyon Açısı ve Taşıt Hızının Elde Edilmesi	71
6.2. Kameralardan Görüntü Verilerinin Alınması.....	72
6.3. Test Güzergahı ve Sürücülerin Seçimi.....	74
6.4. Öndeki Aracın Tespiti	77
6.5. Öndeki Araç İle Mesafe Tespiti	80
6.6. Sürücünün Yüz, Göz ve Burun Tespiti	83

	Sayfa
6.7. Test Verilerinin Oluřturulması.....	85
7. YSA MODELİ VE SONUÇLARIN DEĞERLENDİRİLMESİ	87
7.1. YSA Modelinin Oluřturulması	87
7.2. Makine Öğrenmesi ile Bazı Sınıflandırma Yöntemlerinin Uygulanması	89
7.2.1. Naive Bayes sınıflandırma algoritması	89
7.2.2. En yakın komřu (KNN) sınıflandırma algoritması	90
7.2.3. Destek vektörü (SVM) sınıflandırma algoritması.....	91
7.2.4. Karar ağaçları sınıflandırma algoritması.....	92
8. SONUÇ VE ÖNERİLER	97
KAYNAKLAR	99
ÖZGEÇMİŐ	105

ÇİZELGELERİN LİSTESİ

Çizelge	Sayfa
Çizelge 2.1. Trafik kazalarından ölenlerin 100 000 kişiye oranı - sayısal (2020)	3
Çizelge 2.2. Türkiye’de trafik kazaları, ölü ve yaralı sayıları 2007–2019	4
Çizelge 2.3. Ölümlü ve yaralanmalı trafik kazalarına neden olan kusurlar	5
Çizelge 2.4. 2019 yılı trafik kazalarında sürücü kusurlarının oranları	6
Çizelge 5.1. Ölçüm yapılan parametreler	51
Çizelge 5.2. Sürücülerin hız ihlali yapma nedenleri	52
Çizelge 5.3. Sürücülerin hız ihlali yapmamak için almayı düşündükleri önlemler	52
Çizelge 5.4. Duygusal durumun sürüşe etkisi.....	57
Çizelge 6.1. Test ve eğitim verisi.....	85

ŞEKİLLERİN LİSTESİ

Şekil	Sayfa
Şekil 2.1. 2007 – 2019 yılları arası meydana gelen trafik kazaları, ölümlü, yaralanmalı ve maddi hasarlı trafik kazası sayısı	5
Şekil 2.2. Sürücünün dikkatini dağıtan olayların gruplandırılması	8
Şekil 2.3. Sürücülerin dikkatini dağıtan olaylar	8
Şekil 2.4. Yerleşim yeri dışında sürücü hatalarının dağılımı	10
Şekil 3.1. 16 x 16 bölümlenmiş ızgarada 256 adet gri seviyenin gösterimi	12
Şekil 3.2. Sayısal görüntü fonksiyonu	13
Şekil 3.3. Sürekli bir görüntünün sayısallaştırılması	14
Şekil 3.4. Aşınma işlemi	15
Şekil 3.5. Genleşme işlemi.....	15
Şekil 3.6. Açma işlemi	16
Şekil 3.7. Kapama işlemi	16
Şekil 3.8. Ortanca filtre çalışma şekli	17
Şekil 3.9. Gauss dağılımı	18
Şekil 3.10. Roberts operatörü	20
Şekil 3.11. Sobel operatör maskesi	20
Şekil 3.12. Prewitt operatör maskesi	21
Şekil 3.13. Fotoğraf ve histogram görüntüsü	22
Şekil 3.14. Histogram eşitleme fotoğraf – eşitleme	22
Şekil 3.15. Kartezyen koordinat sistemi ve nokta için koordinat sistemi	23
Şekil 3.16. Doğrunun ρ ve θ gösterimi	24
Şekil 3.17. Döndürme işlemi	25
Şekil 3.18. Affine dönüşümü gösterimi	26
Şekil 3.19. Kartezyen ve homojen koordinatlar	26

Şekil	Sayfa
Şekil 3.20. Haar Cascade çerçeveleri	27
Şekil 3.21. Haar Cascade resim üzerinde çerçeve hareketleri	28
Şekil 3.22. Viola - Jones dört nokta gösterimi	29
Şekil 3.23. Viola – Jones dedektör tipleri	30
Şekil 3.24. Özellik haritası ve tahmin vektörü	32
Şekil 3.25. Nesne tanıma işleminin aşamaları	32
Şekil 4.1. Yapay sinir ağı hücresi	33
Şekil 4.2. Sigmoid tipi fonksiyon	35
Şekil 4.3. Çok katmanlı ağ yapısı	36
Şekil 4.4. YSA bilgi akışı	36
Şekil 4.5. Çok katmanlı ağ yapısı	40
Şekil 4.6. Öğrenmenin hata uzayındaki gösterimi	45
Şekil 4.7. Öğrenen bir ÇKA'nın öğrenme eğrisine örnek	45
Şekil 4.8. KNN ile bir sınıflandırma	47
Şekil 4.9. Destek vektörleri algoritması gösterimi	48
Şekil 4.10. Karar ağaçları algoritmasının genel yapısı	49
Şekil 5.1. Ses efektlerinin etkisinin ölçümü	56
Şekil 5.2. Yorgunluk tespiti için kullanılan YSA ağ yapısı	64
Şekil 5.3. Akıllı telefon ve araç eksenleri	69
Şekil 6.1. ODIS (Offboard Diagnostic Information System) test cihazı	71
Şekil 6.2. ODIS cihazından alınan sensör verisi.....	72
Şekil 6.3. Test güzergahı.....	75
Şekil 6.4. Google maps trafik görüntüsü	75
Şekil 6.5. Cumartesi günleri belirli saatlere göre trafik yoğunluğu	76
Şekil 6.6. Ayar paneli.....	81
Şekil 7.1. YSA modeli ve eğitim paneli	87

Şekil	Sayfa
Şekil 7.2. YSA eğitim sonu regresyon grafikleri	88
Şekil 7.3. YSA eğitim performans grafiği	89
Şekil 7.4. Naive Bayes sınıflandırma sonuçları	90
Şekil 7.5. En yakın komşu sınıflandırma sonuçları	91
Şekil 7.6. Destek vektörü sınıflandırma sonuçları	92
Şekil 7.7. Karar ağaçları sınıflandırma sonuçları.....	93
Şekil 7.8. YSA eğitim hata matrisi	94
Şekil 7.9. YSA test hata matrisi	94
Şekil 7.10. Test ve YSA verilerinin karşılaştırılması	95

RESİMLERİN LİSTESİ

Resim	Sayfa
Resim 5.1. Deneyde kullanılan ölçüm araçları	54
Resim 5.2. Test simülasyon platformu	54
Resim 5.3. Saldırganlık düzeyini ölçmek için kullanılan simülatör ve bir sürüş senaryosu.....	55
Resim 5.4. Simülatör ve test düzeneği.....	57
Resim 5.5. Doğru pedal seçimi belirleme ölçümü.....	61
Resim 5.6. Doğru pedal seçimi test simülatörü	61
Resim 5.7. Ölçüm aracı.....	62
Resim 5.8. Sürücü aktivitelerinin 4 kamera görüntüsü ile ölçümü.....	63
Resim 5.9. Sürücü pozisyonları	65
Resim 5.10. Simülasyon test düzeneği	67
Resim 5.11. Anlık verilerin toplanması	68
Resim 5.12. Veri toplamak için kullanılan cihazlar.....	70
Resim 6.1. Sürücünün yüz hareketleri ve öndeki aracın tespiti için kullanılan kameraların montaj görüntüsü	73
Resim 6.2. Havana HVNA-8081 marka kamera	73
Resim 6.3. ODIS bağlantı modülünün araca bağlanması	73
Resim 6.4. Ön ve sürücü kamera görüntüsü	74
Resim 6.5. Analiz program görüntüsü	77
Resim 6.6. Analiz program görüntüsü öndeki araç tespiti.....	78
Resim 6.7. Kuşbakışı görüntü şerit çizgileri.....	80
Resim 6.8. Kuşbakışı görüntü tespiti alan belirlenmesi.....	81
Resim 6.9. Öndeki araç mesafe tespiti.....	82
Resim 6.10. Sürücünün yüz, göz ve burun tespiti.....	84

SİMGELER VE KISALTMALAR

Bu çalışmada kullanılmış simgeler ve kısaltmalar, açıklamaları ile birlikte aşağıda sunulmuştur.

Simgeler	Açıklamalar
A	Ara katman
B	Beklenen çıktı
Ç	Çıktı
D	Doğru cevaplanan örnek sayısı
E	Hata
F	Aktivasyon fonksiyonu
g	Yerçekimi ivmesi (m/s^2)
G	Girdi
P	Performans
$P(x C)$	Örneğin x olma olasılığı
T	Toplam örnek sayısı
TH	Toplam hata
W	Ağırlık
X	Girdi
α	Momentum katsayısı
β	Eşik değeri ağırlıkları
σ	Standart sapma
δ	Hata
λ	Öğrenme katsayısı
Φ	Eşik değeri
Kısaltmalar	Açıklamalar
CMYK	Camgöbeği, magenta, sarı, anahtar
CNN	Evrişimli sinir ağları
ÇKA	Çok katmanlı ağlar

Kısaltmalar	Açıklamalar
ÇKA	Çok katmanlı ağlar
DBQ	Sürücü davranış anketi
DPI	İnç kareye düşen piksel sayısı
GMM	Gauss karışım modeli
HSB	Ton/doygunluk/parlaklık
KNN	K-En yakın komşuluk
NHTSA	Ulusal Karayolu Trafik Güvenliği İdaresi
ODB2	On-Board Diagnostik II
ODIS	Offboard Diagnostik Bilgi Sistemi
RGB	Kırmızı, yeşil, mavi
ROC	Alıcı çalışma karakteristiği
SDS – ADAS	Gelişmiş sürücü yardım sistemi
SVM	Destek vektör makinaları
YSA	Yapay sinir ağları

1. GİRİŞ

Trafik kazası Karayolları Trafik Yönetmeliği'nde; karayolunda hareket eden bir veya birden fazla aracın karıştığı ölüm, yaralanma ve/veya zararlı sonuçlanmış olay olarak tanımlanmıştır. Dünyada motorlu taşıt sayısındaki artışın olumsuz sonuçlarından birisi de trafik kazalarındaki artıştır. Yıllardır trafik kazalarının azaltılması ile ilgili kurallar, sürücü eğitimleri ve taşıt güvenlik sistemleri gibi birçok alanda ilerleme kaydedilmesine karşın kaza sayısındaki artışın önüne geçilememektedir.

Dünya Sağlık Örgütü verilerine göre 2000 ile 2016 yılları arasında motorlu taşıt sayısı 850 milyondan 2,1 milyar adede çıkmıştır. Raporda 100 bin araç başına kazalardaki ölüm sayısının 135'ten 65'e gerilediği de ifade edilmiştir. Araç başına göre düşme görülse de 2000 yılında dünya genelinde trafik kazalarında 1,15 milyon insan hayatını kaybederken, 2016 yılında bu rakam 1,35 milyona çıkmıştır. Trafik kazalarındaki ölümler, ölüm nedenleri arasında sekizinci sırada yer alırken 5 ile 29 yaş arasında ilk sırada bulunmaktadır. Ekonomik açıdan bakıldığında gelişmemiş ülkelerde trafik kazaları nedeniyle meydana gelen ölümlerin oranı gelişmiş ülkelere göre 3 kat daha fazladır [1].

Türkiye'de motorlu taşıt sayısı 2009 yılında 14 316 700 iken bu sayı 2020 yılında 24 144 857 adede yükselmiştir. Ölümlü/yaralanmalı trafik kazası sayısı da 111 121 den 150 275'e çıkmıştır. Araç sayısı ve kaza sayısının artış oranlarına karşın trafik kazalarında meydana gelen ölüm sayısı 4 324'ten 4 866'ya çıkarak nispeten daha az bir artış olmuştur [2]. Ülkemizde meydana gelen trafik kazalarının ekonomik kaybının 13 milyar dolar civarında olduğu öngörülmektedir [3].

Dünyada ve Türkiye'de trafik kazası sayısı yıllara göre artış gösterse de ölümlü kazalarda artışın aynı oranda olmadığı görülmektedir. Gelişmiş ülkelerin trafik kazalarını önleme konusuna devlet politikası boyutundaki eğilimlerinin olumlu gelişmelerin önünü açtığı söylenebilir. Otomotiv üreticilerinin araç güvenlik sistemlerinde sağladıkları geliştirmeler, bölünmüş karayolu sayısının artması, trafik cezalarının caydırıcı olması ve sürücülerin daha da bilinçlenmesi bu sorunun önüne geçilmesindeki en önemli etkenler olarak düşünülebilir.

Trafik kazalarının sebepleri arasında ilk beş unsur olarak sürücü, yolcu, yaya, yol ve taşıt gösterilmektedir. Yıllara göre bakıldığında, trafik kazalarının sebepleri arasında en fazla

kusur %88 ile %90 aralığında sürücülerde görülmektedir [2]. Maalesef bu oranın yıllardır pek değişmediği de görülmektedir.

Her geçen yıl trafik kazalarının ve etkilerinin araç sayısı ve kaza sayısının artışına göre daha az olmasına karşın kaza sayısının azaltılamaması ve kazalarda en büyük hata payının sürücüde olması durumu değişmemiştir. Aktif güvenlik sistemleri arasında yer alan sürücü destek sistemleri (SDS–ADAS) ile sürücü, taşıt ve çevre ilişkisini temel alarak yapılan geliştirmelerin sürücü hatalarını azaltma konusunda etkili olmaya başladığı söylenebilir. Sürücü destek sistemleri, sürücüye uyarıda bulunan veya öncesinde risk analizi ve yorumlama yaparak kaza riskini azaltabilen yöntemler olarak iki bölümde incelenmektedir.

Bu çalışmada, sürücüye uyarıda bulunabilmek için, sürücü davranışları ile şerit değiştirme eylemi arasındaki ilişki kurulmaya çalışılmış, öndeki aracı geçme amaçlı şerit değiştirme eyleminin gerçekleşmesinden önce eylemin tahmin edilmesi amaçlanmıştır. Sürücü göz ve yüz hareketinin tespiti, taşıtın önündeki araçların algılanması için nesne tespiti ve takibi yöntemlerinden yararlanılmıştır. Sürücü hareketleri, gerçek yol şartlarında yapılan test sürüşleri ve kameralarla kayıt altına alınmıştır. Testlerin trafik yoğunluğunun şerit değişimlerine izin verecek saatlerde yapılması sağlanmıştır. Kullanılan iki kameranın aynı zamanda kayıt almasının sağlanması için Visual Studio ile video kayıt programı hazırlanmıştır. Testte kullanılan aracın hızı ve direksiyon hareketinin tespiti için aracın ODB2 bağlantısından yararlanılarak ODIS test cihazı kullanılmıştır. Elde edilen verilerle yapay sinir ağlarından yararlanarak bir tahmin modeli oluşturulmuştur.

2. TRAFİK KAZALARI

Dünya Sağlık Örgütü'nün yayımladığı raporda, dünyada yılda yaklaşık 1,3 milyon kişi trafik kazalarında can vermekte ve bu ölümlerin %94'ü düşük gelirli ülkelerde olmaktadır. Ayrıca 20-50 milyon arasında da yaralanmaların olduğu tahmin edilmektedir. Bahsi geçen ekonomik gelir düzeyi düşük ülkelerdeki araç sayısı dünya genelinin %54'ünü oluşturmaktadır. Kazaların ekonomik açıdan yarattığı maliyet ise ülkelerin gayri safi yurtiçi hasılasının yaklaşık %3'üne denk gelmektedir. 2030 yılına kadar trafik kazalarının oluşmasına karşı önlem alınmazsa dünyadaki ölümlerin en büyük 7. nedeni olacağı tahmin edilmektedir [4].

Çizelge 2.1'de görüldüğü gibi, Dünya Sağlık Örgütü'nün 2019 yılındaki çalışmasında dünya üzerindeki trafik kazalarında ölümlerin nüfustaki 100 000 kişiye oranına göre, 61,9 ölümle en fazla ölümler Zimbabve, 2,03 ile de en az İsveç'te meydana geldiği görülmektedir. Türkiye bu sıralamada 8,2 ölümle 130. sırada yer almıştır.

Çizelge 2.1. Trafik kazalarından ölenlerin 100 000 kişiye oranı (2020) [4]

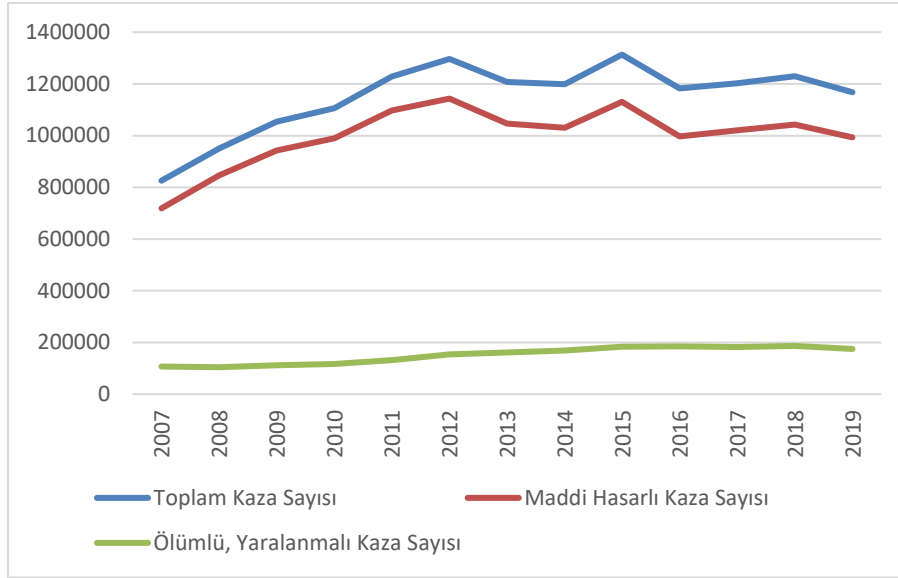
Sıra	Ülke	Oran	Sıra	Ülke	Oran	Sıra	Ülke	Oran
1	Zimbabve	61,9	87	Endonezya	18,3	174	İspanya	2,8
2	Liberya	52,03	88	Yeni Gine	17,9	175	Malta	2,7
3	Malavi	51,62	89	Çin	17,7	176	Danimarka	2,6
4	Gambya	47,51	90	Afganistan	17,7	177	Hollanda	2,6
5	Togo	46,62	91	Samoa	17,6	178	Norveç	2,5
6	Tanzanya	46,17	92	Laos	17,6	179	Birleşik Krallık	2,4
7	Ruanda	45,90	93	Honduras	17,4	180	Kiribati	2,4
8	Sao Tome	45,52	94	Guyana	17,3	181	İsviçre	2,4
9	Burkina Faso	45,94	95	Pakistan	17,1	182	Japonya	2,4
10	Burundi	43,7	130	Türkiye	8,2	183	İsveç	2,3

Çizelge 2.2'de 2007 ile 2019 yılları arasında Türkiye'de meydana gelen trafik kazalarının ölümlü ve yaralanmalı ve maddi hasarlı adet sayıları yer almaktadır. Şekil 2.1'de de kaza sayılarının yıllara göre değişimi görülmektedir. Toplam kaza sayısının, maddi hasarlı kaza

sayısı ile aynı eğilimde olduğu görülmektedir. Ölümlü ve yaralanmalı kaza sayısının ise maddi hasarlı kazalara göre daha düşük sayılarda olduğu görülmektedir. Şekil 2.1’de 2011 ile 2019 yılları arasında toplam kaza sayısında çok önemli bir değişim olmadığı görülmektedir. Ölümlü ve yaralanmalı kaza sayılarında da yıllara göre artış olmadığı görülmektedir. Çizelge 2.2 ve Şekil 2.1’de görüldüğü gibi, ülkemiz için kazaları önleme amaçlı gerçekleştirilen teknolojik ve altyapı çalışmalarının etkili olduğu düşünülebilir. Bunun yanı sıra her ne kadar trafik kazalarının artışının önü kesilmiş olsa da kazaların en aza indirilmesi henüz söz konusu değildir.

Çizelge 2.2. Türkiye’de meydana gelen trafik kazaları, ölü ve yaralı sayıları 2007 – 2019 [5]

Yıl	Toplam Kaza Sayısı	Maddi Hasarlı Kaza Sayısı	Ölümlü, Yaralanmalı Kaza Sayısı	Toplam Ölümlü Kaza	Kaza Yerinde Ölü Sayısı	Kaza Sonrası Ölü Sayısı	Yaralı Sayısı
2007	825 561	718 567	106 994	5 007	5 007	-	189 057
2008	950 120	845 908	104 212	4 236	4 236	-	184 468
2009	1 053 346	942 225	111 121	4 324	4 324	-	201 380
2010	1 106 201	989 397	116 804	4 045	4 045	-	211 496
2011	1 228 928	1 097 083	131 845	3 835	3 835	-	238 074
2012	1 296 634	1 143 082	153 552	3 750	3 750	-	268 079
2013	1 207 354	1 046 048	161 306	3 685	3 685	-	274 829
2014	1 199 010	1 030 498	168 512	3 524	3 524	-	285 059
2015	1 313 359	1 130 348	183 011	7 530	3 831	3 699	304 421
2016	1 182 491	997 363	185 128	7 300	3 493	3 807	303 812
2017	1 202 716	1 020 047	182 669	7 427	3 534	3 893	300 383
2018	1 229 364	1 042 832	186 532	6 675	3 368	3 307	307 071
2019	1 168 144	993 248	174 896	5 473	2 524	2 949	283 234



Şekil 2.1. 2007 – 2019 yılları arası meydana gelen trafik kazaları, ölümlü, yaralanmalı ve maddi hasarlı trafik kazası sayısı

Çizelge 2.3'e göre 2012 ile 2019 yılları arasında trafik kazalarına sebep olan faktörlere baktığımızda en büyük payı %88 ile %89,9 aralığında sürücü kusurunun aldığı görülmektedir. Trafik kazalarının azaltılması için sürücü kusurları veya etkileri mutlaka azaltılmalıdır. Son yıllarda özellikle bu amaçla birçok çalışma yapılmaktadır. Bu çalışmaların en önemlisinin otonom araçlar olduğu söylenebilir.

Çizelge 2.3. Ölümlü ve yaralanmalı trafik kazalarına neden olan kusurlar [5]

Yıl	Toplam Kusur	(%)	Sürücü Kusuru	(%)	Yolcu Kusuru	(%)	Yaya Kusuru	(%)	Yol Kusuru	(%)	Taşıt Kusuru	(%)
2012	181 266	100	161 076	88,9	797	0,4	17 672	9,7	1 124	0,6	597	0,3
2013	183 030	100	162 327	88,7	774	0,4	16 458	9,0	1 913	1,0	1 558	0,9
2014	193 215	100	171 236	88,6	901	0,5	18 115	9,4	1 841	1,0	1 122	0,6
2015	210 498	100	187 980	89,3	915	0,4	18 522	8,8	1 916	0,9	1 165	0,6
2016	213 149	100	190 954	89,6	869	0,4	18 612	8,7	1 717	0,8	997	0,5
2017	213 325	100	191 717	89,87	782	0,4	18 095	8,5	1 619	0,7	1 112	0,5
2018	217 898	100	194 928	89,46	1 916	0,9	18 394	8,4	1 300	0,6	1 360	0,6
2019	204 538	100	180 042	88,02	2 572	1,3	16 726	8,2	1 045	0,5	4 153	2,0

2.1. Trafik Kazalarında Sürücü Etkisi

Trafik kazalarına sebep olan 5 ana unsur olarak sürücü, yolcu, yaya, yol ve taşıt gösterilmektedir. Kazanın oluşmasında en büyük kusurun %89 ile %90 aralığında sürücü kusuru olduğu ve bu oranın yıllar içerisinde çok değişmediği görülmektedir.

Çizelge 2.4’de 2019 yılında meydana gelen trafik kazalarında sürücü kusurlarının içerikleri listelenmiştir. Bu çizelgeye göre en fazla hatanın araç hızını yol, hava ve trafiğin gerektirdiği şartlara uydurmama olduğu görülmektedir. Sonraki hatalar sırasıyla, sürücünün diğer kusurlu halleri, kavşaklarda geçiş üstünlüğüne uymamak, manevraları düzenleyen genel şartlara uymamak, arkadan çarpmak olarak görülmektedir. Sıralamadaki bu dört kusur toplam kaza kusurlarının yaklaşık %74’üne denk gelmektedir.

Çizelge 2.4. 2019 yılı trafik kazalarında sürücü kusurlarının oranları [5]

Taşıtın hızını hava, yol ve trafiğin gerektirdiği diğer koşullara uydurmamak	70 062
Sürücünün diğer kusurlu halleri	26 203
Kavşaklardaki geçiş önceliklerine uymamak	23 616
Manevraları düzenleyen kurallara uymamak	14 669
Öndeki araca arkadan çarpmak	13 405
Dönüş gibi doğrultu değiştirme kurallarına uymamak	11 652
Taşıt girmesi yasak işareti olan yerlere girmek	4 425
Kırmızı ışığa veya görevlilerin dur işaretlerine uymamak	4 286
Park halindeki araçlara çarpmak	3 897
Şerit takip ve şerit değiştirme kurallarına uymamak	2 564
Alkollü araç kullanmak	2 334
Taşıtı çok yüksek hızda kullanmak	1 836
Geçme yasağı olan yerlerden geçmek	1 093
TOPLAM	180 042

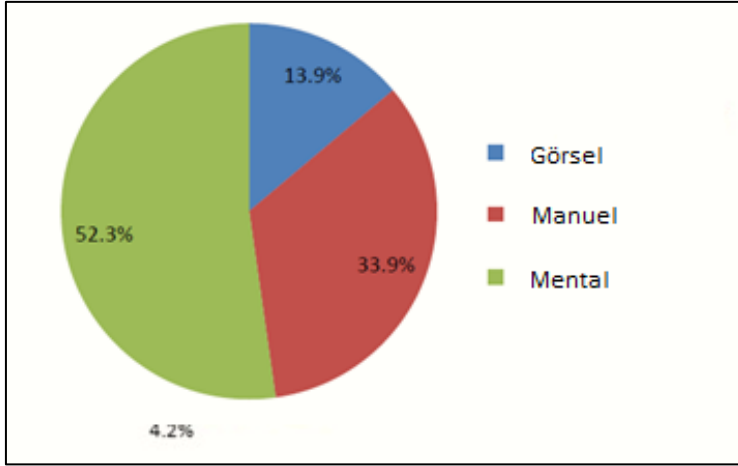
Trafik kazalarında sürücü kusurlarının ortaya konulması amacıyla kullanılan kaza tutanaklarında birçok durum sorgulanmaktadır. Bunlar;

- Araç sürücüsünün kırmızı ıfıkta geip gemediėi,
- Girilmesi yasak olan yola girmek,
- Yanlıř řeritte hareket etmek,
- Sollama yasaėı olan blgede geiř yapmak,
- Kavřaklarda geiř nceliėine aykırı hareket etmek,
- Grevlinin uyarı iřaretlerine uymamak,
- Aynı ynde giden iki aratan arkada bulunan aracın ndeki araca arpması,
- ndeki aracı gemek iin sollama, sola ve geri dnř manevralarında kurallara uymamak,
- Geiř nceliėini olan aralara yol vermemek,
- Anayolda duraklama kurallarına uygun hareket etmemek,
- Park etmiř aralara arpmaktır. [6]

Tutanaktan alınan bilgiler doėrultusunda srcnn hata yaptıėı alanlar ortaya konulmaktadır. Gnmzde bu hataların nne gemek amacıyla birok alıřma yapılmaktadır.

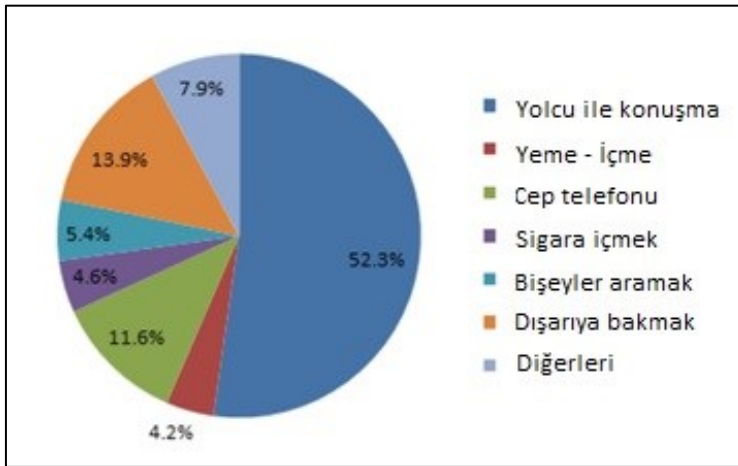
Tung tarafından yapılan doktora tez alıřmasında trafik kazalarında nemli derecede kusuru bulunan srclerde srř esnasında dikkatini daėıtan olaylar 3 tanımda toplanmıřtır [7].

- Grsel: Srř esnasından gzleri yoldan alan anlık olaylardır. Navigasyon gibi ekipmanlara veya yol kenarında bulunan reklam panolarına bakmak rnek olarak gsterilebilir.
- Manuel: Srcnn ellerini direksiyondan ekmesiyle oluřan durumdur. Cep telefonu ile mesaj yazma ve radyo kullanımı gibi davranıřlar rnek olarak verilebilir.
- Mental: Srcnn zihinsel olarak konsantrasyonun bozulması, dikkatinin daėılmasıdır. Arataki yolcularla veya cep telefonuyla konuřmak rnek verilebilir (řekil 2.2).



Şekil 2.2. Sürücünün dikkatini dağıtan olayların gruplandırılması [7]

Tung tarafından yapılan çalışmada ayrıca belirli 3 güzergâhta yerleştirilen kameralarla sürücülerin davranışları incelenmiştir. Bu kameralardan 3150'si erkek 1295'i kadın olmak üzere toplam 4485 sürücü görüntüsü elde edilmiştir. Çalışma sonucunda Şekil 2.3'de görüldüğü gibi en dikkat dağıtıcı olay %52,3 ile yolcu ile konuşma çıkmıştır. Sırasıyla %13,9 ile dışarıya bakma ve %11,6 ile cep telefonu en çok dikkat dağıtan olaylar olarak sıralanmıştır [7].



Şekil 2.3. Sürücülerin dikkatini dağıtan olaylar [7]

Bunun yanı sıra 2001 yılınca Stutts ve Rodgman hazırladıkları raporda sürücülerin dikkatinin dağılmasının sebeplerini aşağıdaki gibi sıralamışlardır [8].

- Yemek veya içmek,
- Araç dışındaki biri, nesne veya bir olay,

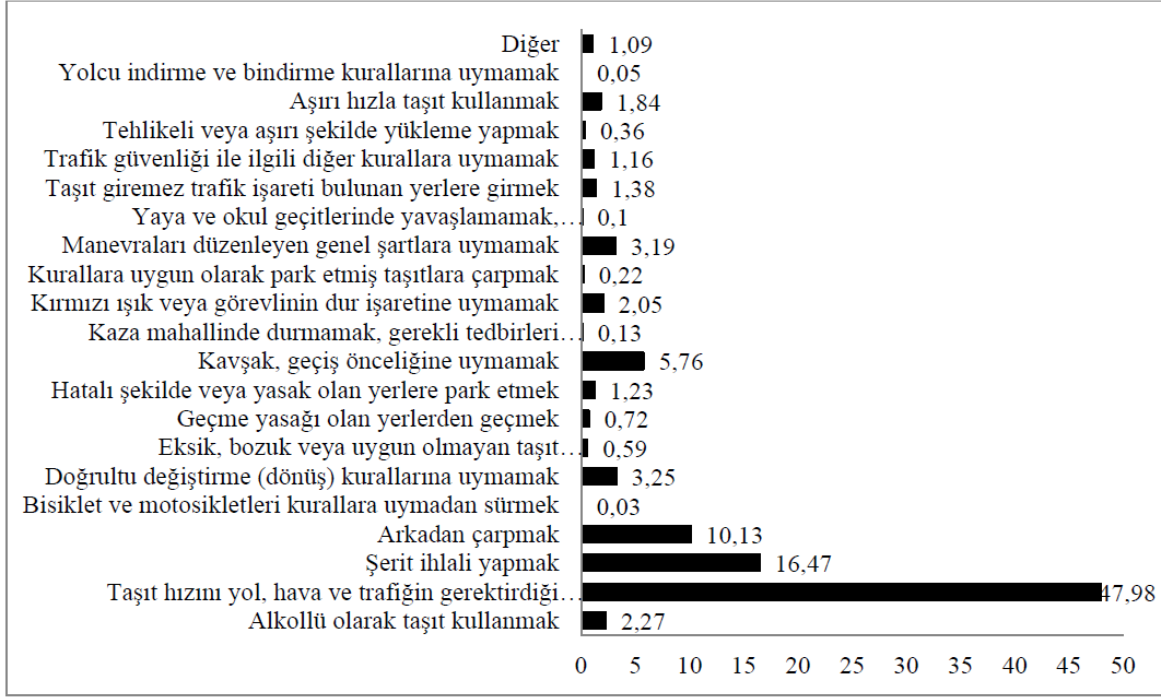
- Radyo veya CD oynatıcı ayarlamak,
- Araç içerisindeki diğer olaylar,
- Araç içerisinde hareket eden bir nesne,
- Sigara içmek,
- Cep telefonu ile konuşmak,
- Araç içerisinde cihaz kullanmak,
- Taşıt hızlanmasını veya kullanım kolaylığı sağlayan kontrol mekanizmaları,
- Diğer dikkat dağıtıcılar.

2003 yılında Young ve Regan tarafından yapılan çalışmada, dikkat dağıtıcı olaylar teknoloji odaklı ve teknolojik olmayan dikkat dağıtıcılar olarak 2 gruba ayrılmıştır. Cep telefonu ile konuşmak, eğlence ve navigasyon cihazları teknolojik, yeme-içme, sigara kullanmak, araçtaki diğer kişiler ile konuşmak ise teknolojik olmayan dikkat dağıtıcılar için örnek olarak gösterilebilir [9].

Sadeqi tarafından 2016 yılında yapılan tez çalışmasında, Amerika Birleşik Devletleri'nde sürücülerin dikkatinin dağılmasında cep telefon kullanımının %88,5 ile %95,7 arasında etkili olduğunu belirtmiştir. Bu çalışmaya göre cep telefonuna gelen mesajın okunması veya mesaj göndermek için sürücünün gözleri 4,6 s kadar yoldan ayrılmaktadır. 2014 yılında yayımlanan bir makaleye atıfta bulunarak, sürücünün dikkatinin dağılması nedeniyle oluşan kazaların maliyetinin 129 milyon dolar olduğunu ifade etmiştir. Buna karşın, trafik tutanaklarında görevli memurun raporuna dayanarak, kazanın dikkat dağınlığından kaynaklanıp kaynaklanmadığının kesin olarak ifade edilemediğini de belirtmiştir [10].

Ülkemizde Karayolları Trafik Kanunu'nun 84. Maddesinde, trafik kazalarında sürücü kusurlarının belirlenmesi ve asli kusurlu sayılan haller belirtilmiştir [11]. Kaza tutanaklarında bulunan maddelerin bu kanunda ifade edilen maddelerden oluştuğu görülmektedir. Kanunda belirtilen maddeler çerçevesinde trafik kazalarında sürücülerin hangi hallerde kusurlu sayılacağı belirlenmektedir. Kazada bahsi geçen maddelerden herhangi biri veya birkaçı, kazaya karışan sürücülerden birden fazlası tarafından yapılmışsa kusur oranları yönetmeliklere bağılı olarak sürücüler için ayrı ayrı belirlenir [11].

Kuyumcu ve arkadaşları tarafından yapılan çalışmada yerleşim yeri dışında sürücü hatalarının dağılımı belirlenmiştir. Şekil 2.4’de görüldüğü gibi sürücü hataları arasında şerit ihlalinin, yerleşim yeri dışında olan kazalardaki payı %16,47 olarak gösterilmiştir [12].



Şekil 2.4. Yerleşim yeri dışında sürücü hatalarının dağılımı [12]

Trafik kazalarına sebep olan en önemli etkenin sürücü olduğu görülmektedir. Sürücü davranışı özelinde yapılacak araştırmaların kazaları önleme açısından etkisinin yüksek olacağı görülmektedir. Yapılan bu çalışmada da sürücü davranış veya hataları arasında yer alan hatalı sollama / şerit ihlali durumu incelenmiştir. Sürücünün şerit değiştirme veya öndeki aracı geçme isteğinin belirlendiği durumlar için bir tahmin modeli oluşturulmuştur.

3. GÖRÜNTÜ İŞLEME

Son yıllarda teknolojinin gelişimiyle görüntüden veri elde etme kolaylaşmış ve kullanım alanı artmıştır. Bu artışın etkisi ile görüntüden elde edilen bilginin de önemi artmıştır.

3.1. Görüntü İşleme Temel Bileşenleri

Piksel: Görüntüyü oluşturan kare şeklindeki noktalardır. Renkli görüntünün oluşabilmesi için pikseller içerisinde kırmızı, yeşil ve mavi renkler bulunmalıdır. Bu üç rengin belirli miktarlarda karışmasıyla diğer renkler oluşur.

Nokta ve Nokta Aralığı: Piksel üç ana rengi barındırır. Bu üç rengin her birine nokta, aralarındaki mesafeye de nokta aralığı denir. Bu mesafe görüntü kalitesine etki eder. Aralık ne kadar az olursa görüntü de o kadar kaliteli olmaktadır.

Çözünürlük: Ekranda görüntülenebilen piksel sayısına çözünürlük denir. Sütun ve satırda bulunan piksel sayılarının çarpılmasıyla bulunur. Bu değer arttıkça görüntü kalitesi artar.

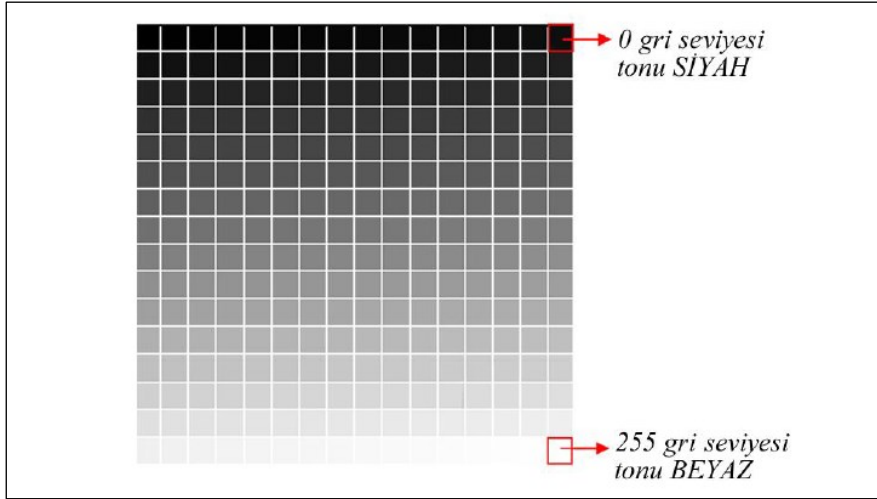
Piksel Yoğunluğu: Bir inç karede bulunan piksel sayısını ifade eder.

Çerçeve: Gerçek zamanlı olarak elde edilen videolar saniyede birbiri ardına çekilen 25 veya daha fazla görüntünün birleştirilmesiyle oluşur. Videoyu oluşturan bu görüntülerin her birine çerçeve denir.

İnç Kareye Düşen Piksel Sayısı: Görüntülerin kâğıda alınan baskılarında bir inç kareye basılan piksel sayısını ifade eder. İnç kareye düşen piksel sayısı (DPI) arttıkça görüntünün kalitesi de artar.

Renk Modelleri: Sayısal görüntülerdeki renkleri ifade eder. RGB, CMYK ve HSB renk modelleri bulunmaktadır. Birden fazla modele ihtiyaç duyulma nedeni renk çeşidinin fazla olmasıdır [13].

Gri Tonlamalı Renk Uzayı: Bu renk uzayı gri renginin 256 tonunu barındırır. Piksellerin parlaklık değerlerini 0 ile 255 değerleri arasındaki rakam temsil eder. Şekil 3.1’de görüldüğü gibi siyah renk 0, beyaz renk ise 255’tir [13].



Şekil 3.1. 16 x 16 bölümlenmiş ızgarada 256 adet gri seviyenin gösterimi [13]

Kırmızı – Yeşil – Mavi (RGB) Renk Uzayı: Adından da anlaşılacağı üzere kırmızı, yeşil ve mavi renklerinin belirli oranlarda karıştırılmasıyla oluşan renkleri barındırır. Bu üç rengin %100 oranında karıştırılmasıyla beyaz renk elde edilirken, üç renginde %0 oranında karışımından siyah renk elde edilir [13].

Renkli Görüntü: Renkli görüntü kırmızı (R), yeşil (G) ve mavi (B) yerleştirilmiş nesneye ait görüntüsünün üst üste gösterilmesiyle oluşur. Renkli görüntünün değer aralığı da RGB = (0,0,0) ile (255,255,255) arasında gösterilir [13].

Resim: İçinde konum ve renk bilgisi bulunan görsel temsillere denir [13].

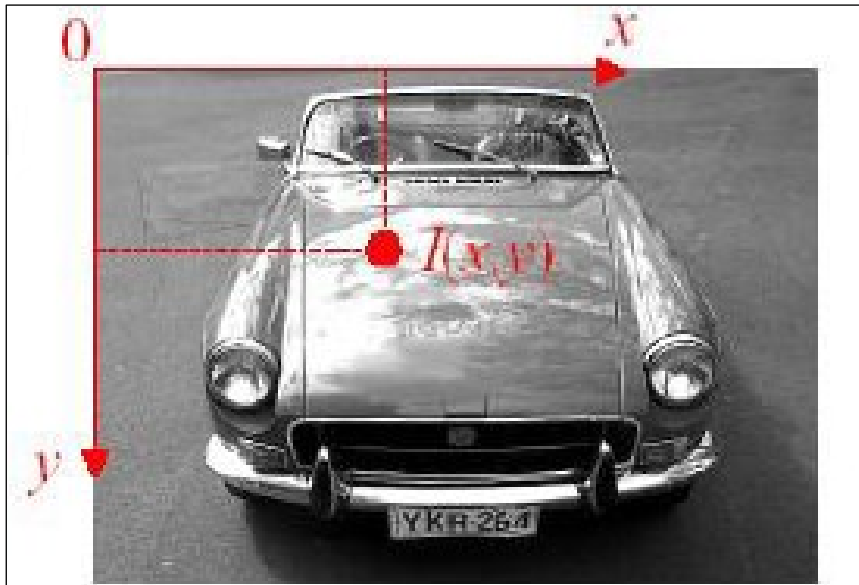
3.1.1. Analog görüntü

Işık geçirgenliği olmayan nesnelere, ışığın yansımalarıyla bir yüzeye düşerek oluşturduğu sürekli izdüşümü meydana gelmesidir. Maksimum ve minimum aralığı belirlenmiş $f(x,y)$ fonksiyonunda x ve y değerleri sürekli olarak reel sayılar alır. Analog bir görüntüde nesneye çok yakından bakılsa da görüntüyü oluşturan renkler bulunur [13].

3.1.2. Sayısal (Dijital) görüntü

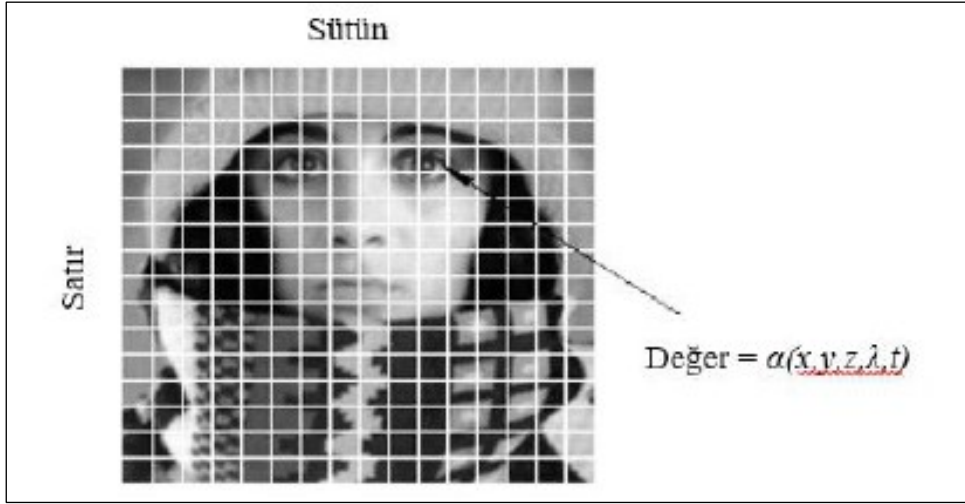
Sayısal görüntü, analog görüntünün piksel adı verilen küçük karelere ayrılmasıyla oluşur. $f[x,y]$ biçiminde gösterilir. Burada pikselin değeri, piksel alanında kalan ortalama parlaklık seviyesidir [14].

Dijital görüntü renkleri temsil eden 256 farklı renk yoğunluğundan oluşur. Pikseller iki değişkenden oluşan bir fonksiyon ve bu fonksiyonun ortaya koyduğu sayı değerine sahiptir. $I(x,y)$ şeklinde bir fonksiyonda x ve y koordinatları ile gösterilir. Görüntü üzerindeki yerleşimi Şekil 3.2’de gösterilmiştir. I ile gösterilen görüntünün özellik fonksiyonudur. Bu fonksiyon sayesinde görüntünün tümü veya bazı bölgelerinde işlemler yapılabilir [13].



Şekil 3.2. Sayısal görüntü fonksiyonu [14]

Şekil 3.3’de sayısal görüntünün koordinat düzleminde gösterimi yer almaktadır. Satır ve sütunların kesiştiği bölge pikseldir. Burada “z” derinliği, “ λ ” rengi ve “t” zamanı gösteren bir fonksiyondur. Piksellerde bulunan tamsayı ise o pikselin parlaklık değeridir.



Şekil 3.3. Sürekli bir görüntünün sayısallaştırılması [14]

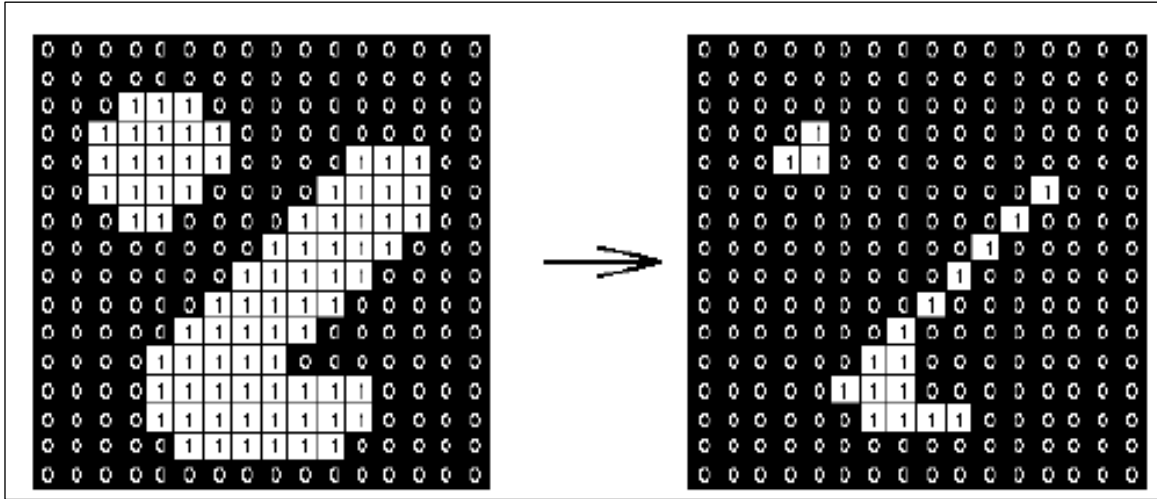
3.2. Görüntü İşleme Teknikleri

3.2.1. Morfolojik işlemler

Morfoloji biçim bilimi olarak ifade edilir ve gri seviyeye dönüştürülen görüntünün analizinde kullanılır. Görüntüde bulunan gürültü ve sorunların giderilmesinde kullanılır. İşlemler ikili görüntü üzerinde yapılır. Bu işlemler açma, kapama, genişleme ve aşınma işlemleridir [15].

Aşınma

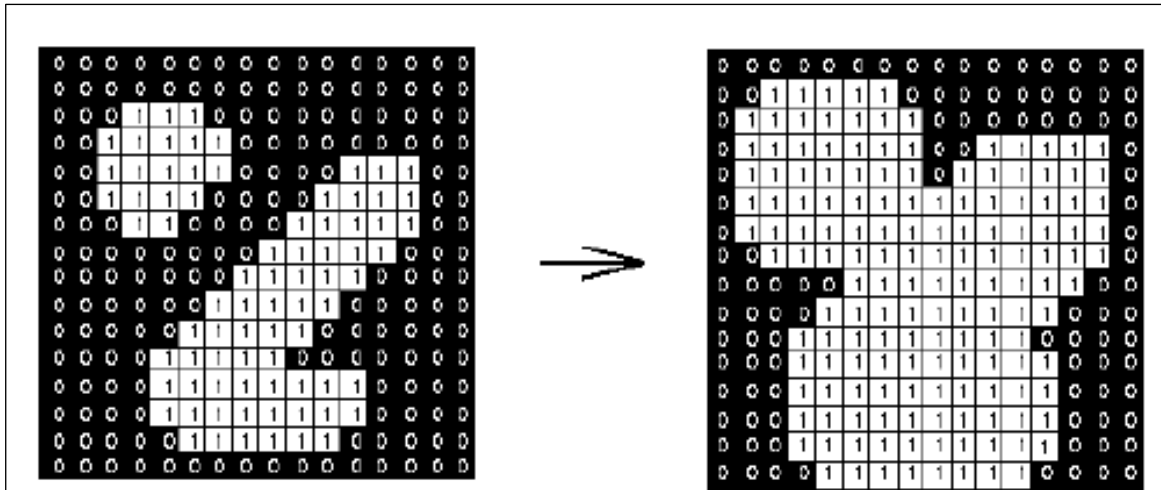
Görüntü içerisinde yer alan bir objeyi inceltmek ya da küçültmek için kullanılan yöntemdir. İkili moda çevrilen görüntüye aşınma uygulanırsa nesne ölçülerinde küçülme meydana gelir, önemsiz alanlar yok olur ve delik varsa genişler. Şekil 3.4’de bir görüntüde aşınma işleminin sonucu görülmektedir.



Şekil 3.4. Aşınma işlemi [15]

Genleşme

Aşınmanın tersidir. İkili görüntüde yer alan bir objeyi büyütme veya kalınlaştırma işlemidir. Görüntüdeki bu kalınlaştırmayı kontrol etmek için yapısal eleman kullanılır. Şekil 3.5’de genleşme filtresinin etkisi görülmektedir.

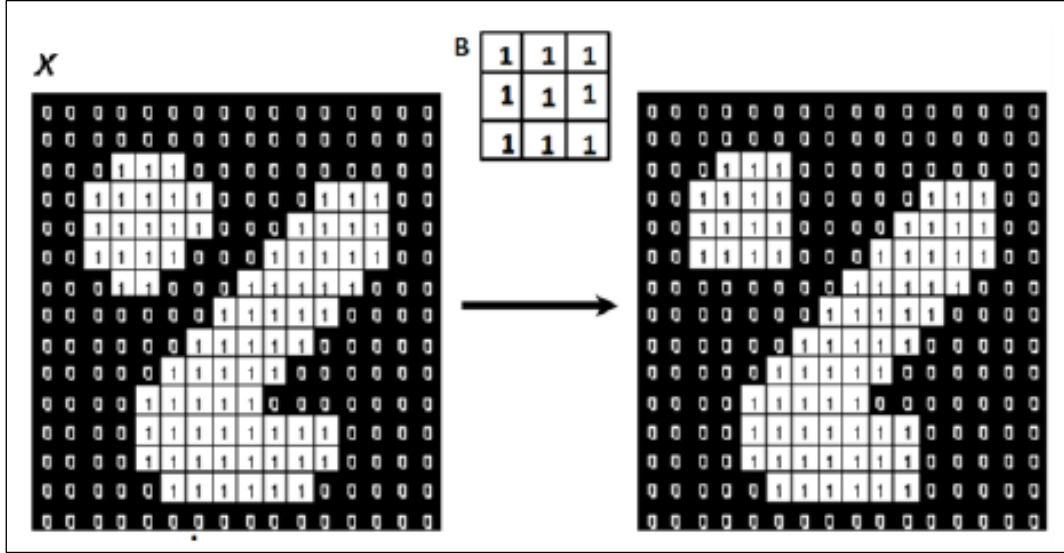


Şekil 3.5. Genleşme işlemi [15]

Açma

Görüntü üzerinde aşınma yönteminden sonra genleşme uygulandığında açma işlemi meydana gelmiş olur. Görüntü üzerindeki gürültüleri gidermek amacıyla kullanılır. Bu

işlemde küçük nesneler yok edilir. Şekil 3.6’da açma işleminin görüntüde yaptığı değişiklik gösterilmektedir.

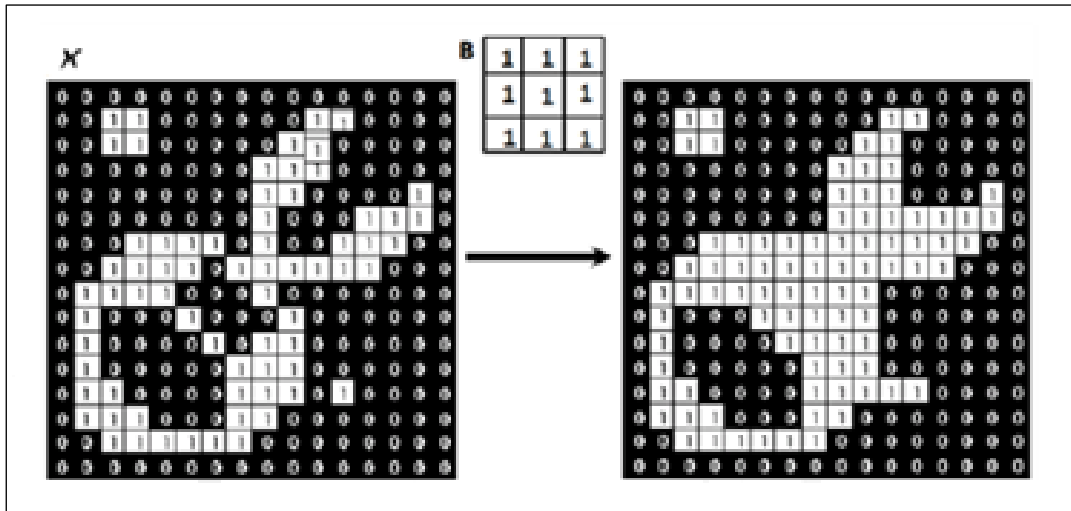


Şekil 3.6. Açma işlemi [15]

Kapama

Açma işleminin tersidir. Genleşme uygulandıktan sonra aşınma işleminin yapılmasıyla elde edilir. Kapama işleminde birbirine yakın nesneler birleşir ve boşluklar ortadan kalkar [15].

Şekil 3.7’de kapama işleminin etkisi görülmektedir.



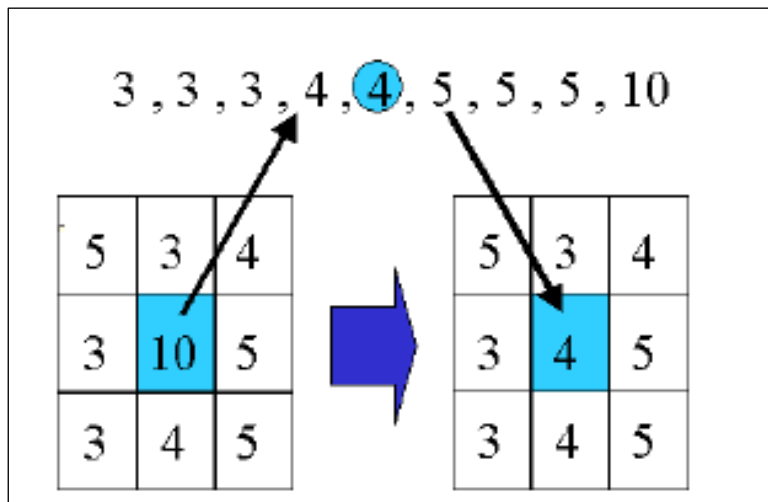
Şekil 3.7. Kapama işlemi [15]

3.2.2. Filtreleme işlemleri

Görüntü üzerinde piksele komşu olan her piksel için uygulanan işlemleri ifade eder. Görüntüdeki gürültünün giderilmesi, yumuşatılması, istenmeyen durumların ortadan kaldırılması ve talep edilen belirli verinin elde edilmesini amaçlar. Ortanca (median), ortalama (mean), Gauss filtreleme ve eşikleme en çok kullanılan filtreleme işlemleri arasındadır.

Ortanca Medyan

Filtreleme işlemlerinin ilk basamağı medyan filtresidir. Bu filtrelemede amaç görüntünün bulanıklığını azaltmak ve görüntüye netlik kazandırmaktır. Bunun yanında görüntüde oluşan nokta hatalarını gidermek için de kullanılır. Medyan filtresinin olumsuz etkilerinden biri görüntüdeki detay bilgisini azaltmasıdır. Uygulanması ortanca piksel değerinin komşu piksel değeri ile değiştirilmesi şeklindedir [16]. Piksel değerlerinin tümü küçükten büyüğe sıralanır ve ortadaki değer merkezde yer alan piksele yazılır. Şekil 3.8’de ortanca filtrenin çalışma şekli verilmiştir [15].



Şekil 3.8. Ortanca filtre çalışma şekli [15]

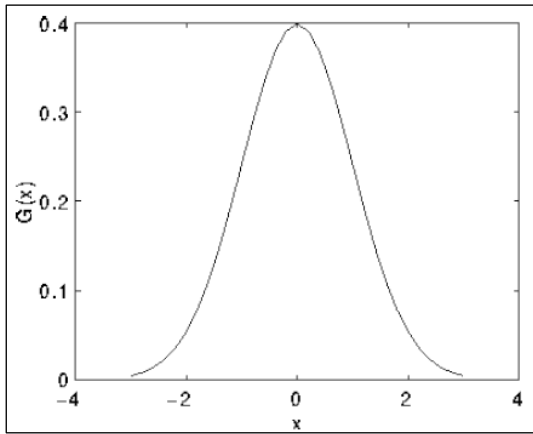
Ortalama

Ortalama filtre ile görüntü içinde yer alan gürültü yok edilerek görüntü bulanıklaştırılır. Bu işlemde bir pikselin etrafında bulunan bütün piksellerin ortalama değeri alınarak orta noktadaki değer belirlenir [15].

Gauss Filtresi

Görüntüde düzleştirme işleminde kullanılır. Görüntüdeki gürültüyü kaldırma ve bulanıklaştırma amacıyla uygulanır. Bulanıklaştırmanın derecesi standart sapmayla belirlenir. Bu yöntemde medyan filtrelemeye göre daha yumuşak bir işlem uygulanır. Daha geniş kenar çıkarır [16]. Bu operatör, bir konvolüsyon operatörüdür. Konvolüsyon, belirli bir çekirdek matris ile tüm piksellerin çarpımı ile oluşur. Bu yönden ortalama filtreye benzemektedir. Bunun yanında ortalama filtreye göre farklı olarak çan şeklinde bir çekirdek şablon kullanır. Gauss filtrenin çan şeklindeki grafiği Şekil 3.9’da, eşitliği de Eşitlik 3.1’de gösterilmektedir. Bu eşitlikte σ dağılımın standart sapmasıdır [17].

$$G(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}} \quad (3.1)$$



Şekil 3.9. Gauss dağılımı [17]

Eşikleme ve Eşik Değer Belirleme

Görüntüyü binary şeklinde dönüştürme işlemidir. Bunu da belirlenen bir sınır değeri kullanarak yapar. Genellikle görüntü özelliklerinin çıkartılmasında kullanılır [16].

Filtreleme yöntemlerinin temel tekniği olarak düşünülebilir. Öncelikle bir eşik değeri belirlenir ve bu değer çıkarmaya çalıştığımız nesnelerin özelliklerinin piksel değerleri olabilir. Sonrasında eşik değerinin üzerinde kalan beyaz (0) altında kalan siyah (1) olarak ifade edilir. Çıkarılan özellik ikili görüntülerdir. Tüm resimler aynı olmadığından iki eşik belirleme yöntemi, merkezdeki piksellere komşu olan piksellere etki eden yerel ve tüm görüntüyü etkileyen global yöntemi vardır [15].

3.2.3. Kenar belirleme

Görüntünün gri seviyelerindeki ani değişimin olduğu alanlar kenar olarak tanımlanır. Kenar belirleme renk farklılığının gözlemlendiği ve bu şekilde resimdeki nesnelerin ayrımının kolaylaştığı bir yaklaşımdır. Resimden faydalı bilgiler elde etmeye ve resmin özelliklerini ortaya çıkarmaya yarar. Kenarlar özellikle görüntülerin sınırlarına ve gölge geçişlerine uygulanır [16].

Görüntünün kenar tespiti için farklı yöntemler kullanılmaktadır. Bu yöntemlerde filtrelerle piksel değerleri tekrar hesaplanır [18]. Hesaplama işlemi Eş. 3.2’de gösterilmiştir.

$$\sum_{i=-\infty}^{+\infty} \sum_{j=-\infty}^{+\infty} h(i,j) \times f(x-i,-j) \quad (3.2)$$

Bu eşitlikte h fonksiyonu filtredir. Bu işlemden sonra tek bir değer elde edilir. Bu değer filtreleme yapılan kısmın merkezindeki piksele atanır. Bu filtrelemede dikkat edilecek husus resmin ilk ve son satırı ile sütunlarına işlem yapılmaması gerektiğidir [18].

Filtre boyutu genellikle 3x3 matris şeklinde seçilir. Filtrenin boyutunun artması işlem yükünü arttırdığı gibi ince detayların kaybolmasına da neden olur. Bu sebeple büyük boyutlu filtreler tercih edilmemektedir [18].

Birçok kenar belirleme yöntemi vardır. Bu yöntemler birinci ve ikinci dereceden türevler alınması bakımından ikiye ayrılmıştır. Genel olarak birinci mertebeden türevler daha kalın kenarları tespit ederken, ikinci mertebeden türevler daha ince kenarların tespitinde kullanılır. Bu iki yöntemden yönsüz kenar maskelerinin kullanıldığı Roberts, Sobel ve Prewitt

operatörleri birinci mertebeden türev, Gauss Laplası ve Canny operatörleri de ikinci mertebeden türev kategorisine ayrılır [17].

Roberts

En çok kullanılan ikili görüntüler üzerinde, basit ve hızlı çalışan bir yöntemdir. Kenarları bulmak için ilk defa türev yöntemini kullanan operatörlerindendir. Bu yöntem satır ve sütun kenarlarını ayrı ayrı bulmaktadır. Sonrasında bulunan kenarlar birleştirilir ve nesnelerin tüm kenarları ortaya çıkar. 2x2 maskeler ile görüntüyü x ve y koordinatlarında tarayarak kenarlar belirlenir. Şekil 3.10'da maskeleme ve Roberts operatörü gösterilmiştir.

m_1	m_2	-1	0	0	-1
m_3	m_4	0	1	1	0

Şekil 3.10. Roberts operatörü [17]

Sobel

Bu operatör görüntü yoğunluğu fonksiyonunun yakınsamasını hesaplamak için kullanılan ayrık türev operatörü olarak da tanımlanabilir. Bu yöntemde 3x3 maskeler ile x ve y koordinatlarında görüntü taranarak kenarlar belirlenir. Şekil 3.11'de 3x3 maskelemeli Sobel operatörü gösterilmektedir.

m_1	m_2	m_3	-1	-2	-1	-1	0	1
m_4	m_5	m_6	0	0	0	-2	0	2
m_7	m_8	m_9	1	2	1	-1	0	1

Şekil 3.11. Sobel operatör maskesi [17]

Prewitt

3x3 maskeler ile f görüntüsü x ve y koordinatlarında taranarak kenarlar belirlenir. Prewit operatör maskesi Şekil 3.12’de bulunmaktadır.

m_1	m_2	m_3	-1	-1	-1	-1	0	1
m_4	m_5	m_6	0	0	0	-1	0	1
m_7	m_8	m_9	1	1	1	-1	0	1

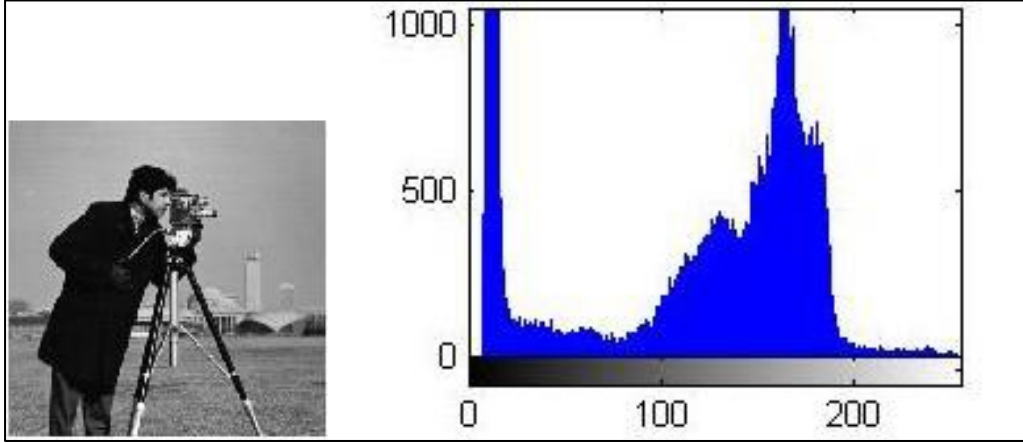
Şekil 3.12. Prewitt operatör maskesi [17]

Canny

Kenarları görüntü gradyanının yerel maksimum olduğu yerleri bularak belirleyen karmaşık bir yöntemdir [17]. Görüntünün gri seviyeli şeklini giriş görüntüsü olarak alır ve kesikliklerin belirtildiği görüntüye çevirir. Bu işlemden önce resmin yumuşatılması için Gauss filtresini kullanır. Sonrasında görüntünün parlak bölümlerini göstermek için yumuşatılan görüntüye uygulanır. Görüntüde kenarlarda tepeler oluşur. Algoritma bu tepe zirvelerinde bulunan pikselleri sıfıra ayarlar [19].

Histogram Eşitleme

Histogram görüntü yoğunluğunu ifade eder. Görüntüdeki nesnelerin veya bileşenlerin piksellerinin toplam piksellere bölünmesi ile normalize edilir. Bunun sonucuda tüm bileşenlerinin toplamı bire eşit olmalıdır. Histogram üzerinde görüntünün koyu renkli bölümleri grilik değerinin düşük olduğu tarafta, açık renkli bölümler ise yüksek ölçekli gri ton tarafında yer alır. Düşük kontrastlı bir görüntü ise histogram ortalarında dar bir grilik şeklinde olur. Şekil 3.13’de bir fotoğraf ve bu fotoğrafın histogramı görülmektedir [15].

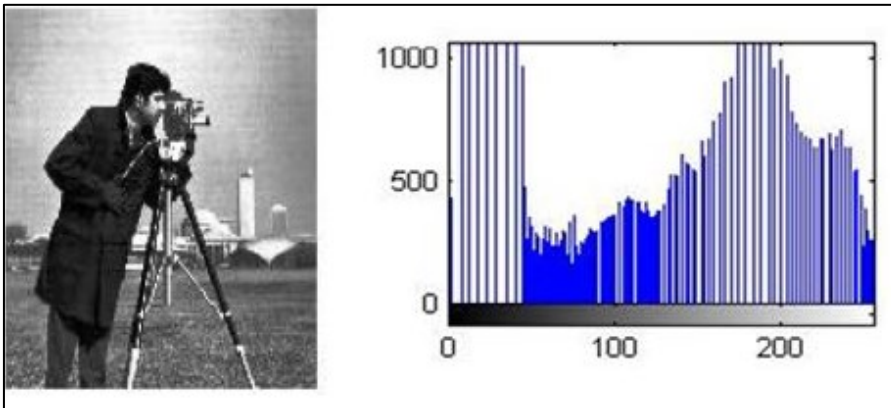


Şekil 3.13. Fotoğraf ve histogram görüntüsü [15]

Histogram eşitleme, görüntüdeki gri alandaki görüntünün histogramını kullanarak karşıtlığın ayarlanması işlemini ifade eder. İstenen görüntü, yoğunluğun eşit şekilde dağılması şeklindedir. Dağılım grafik seviyesinin 0-255 arasında olması istenir [15]. Eş. 3.3’de kontrastı iyileştirmek için gri seviye dönüşüm değeri verilmiştir.

$$s_k = \sum_{i=1}^k \frac{n_j}{n} * (L-1) \quad k = 0, 1, 2, \dots, L-1 \quad (3.3)$$

Bu eşitlikte, n giriş bölümünün piksel sayısını, n_j gri seviyedeki piksel sayısını ve L toplam gri seviye sayısını ifade etmektedir. Şekil 3.14’te histogram eşitleme yapılmış bir fotoğraf ve histogramı görülmektedir.



Şekil 3.14. Histogram eşitleme fotoğraf – eşitleme

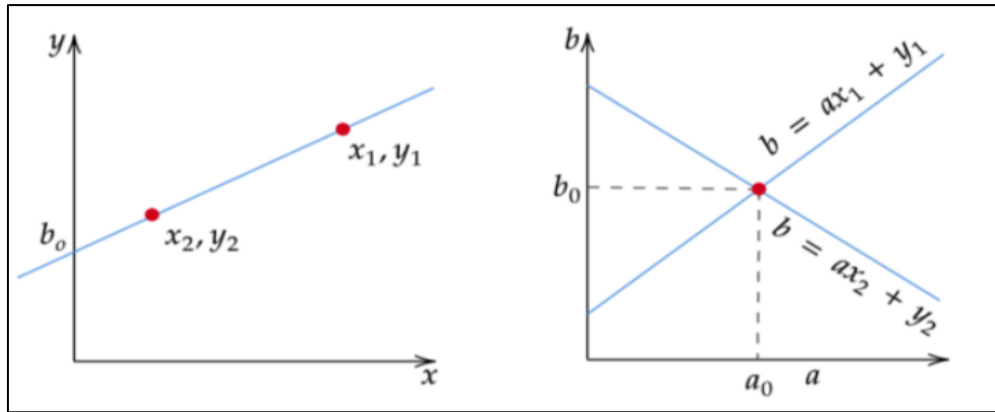
Hough Dönüşüm Tabanlı Metotlar

Hough dönüşüm metodu çizgi tespit etmek gereken durumlarda kullanılır. Özellikle gürültülü görüntülerdeki başarısı nedeniyle kullanımı yaygındır. Bu metotta her piksel üzerinde en az bir kez işlem yapıldığı için hesaplama süresi uzundur. Hough dönüşüm tabanlı metotlarda, ikili görüntüdeki (siyah-beyaz) kartezyen koordinat sisteminden farklı koordinat sistemine, parametre uzayı da denir. Görüntüdeki düz çizgileri tespit etmek için de uygulanan bu yöntemde çizgilerin eğim ve kesişim noktaları kullanılır. Eş. 3.4'de düz çizgiler tanımlanmıştır [20].

$$y=ax+b \quad (3.4)$$

Koordinat eksenindeki çizgiler (x,y) ile ifade edilir. Her çizgi (a,b) ile gösterilen başka bir eksendeki noktaya karşılık gelmektedir. (x,y) eksenindeki bir noktadan birden fazla sayıda doğru geçebilir ve bu doğruların eğimleri ve sonlandırılma noktaları ise (a,b) düzleminde bir çizgi oluşturmaktadır. Bu çizgi ise Eş. 3.5'te verilmiştir. Şekil 3.14'de (x,y) koordinat sistemindeki çizginin (a,b) koordinat sistemindeki konumu gösterilmiştir [20].

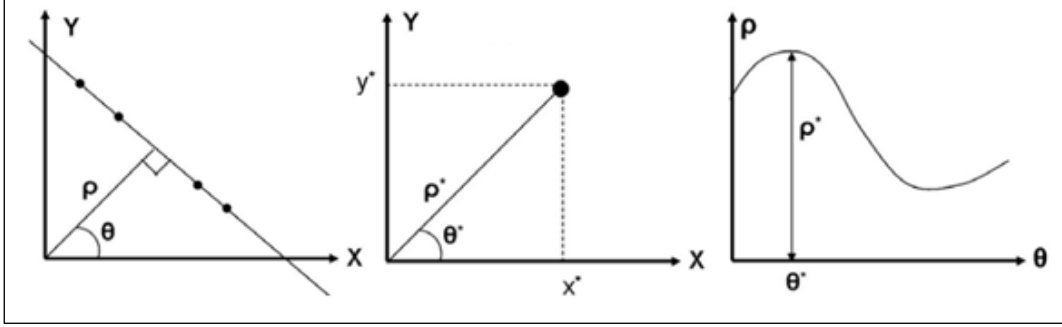
$$b=ax+y \quad (3.5)$$



Şekil 3.15. Kartezyen koordinat sistemi ve nokta için koordinat sistemi

(x,y) koordinat sisteminde seçilen her doğru (a,b) koordinatından bir noktaya denk geldiği gibi (a,b) koordinatında seçilen her nokta (x,y) düzleminde bir doğruya denk gelir. Görüntünün tüm pikselleri taranarak doğru olabilecek noktalar tespit edilmeye çalışılır. Bu tarama esnasında sinüzoid eğriler oluşur. Resimdeki çizginin birden fazla noktasından

geçildiğinde bu sinüzoid eğride tepe noktaları oluşur. Tepe noktaları görüntü üzerindeki doğrunun tespit edildiği noktalardır. Şekil 3.16'da doğrunun orijinle mesafesi olan ρ ve x eksenini ile yaptığı açı θ gösterilmektedir. Ayrıca doğrunun görüntü uzayı ve hough uzayındaki görüntüleri de verilmiştir [21].



Şekil 3.16. Doğrunun ρ ve θ gösterimi [21]

$f(x,y)$ giriş görüntüsünün a x b boyutundan olduğu düşünülecek olursa, giriş görüntüsündeki noktaların x ve y yönünde sırasıyla Δx ve Δy aralıklarına sahip olduğu kabul edilerek Eş. 3.6'daki gibi ifade edilir.

$$H(\rho, \theta) = \sum_a \sum_b f(a, b) \delta(\rho - a\Delta x \cos\theta - b\Delta y \sin\theta) \quad (3.6)$$

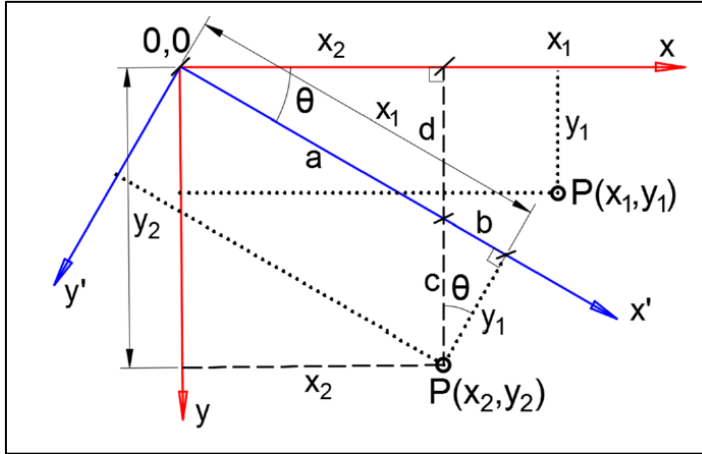
Giriş görüntüsündeki her bir noktanın parametre uzayında bir sinüzoid dönüşmesi Eş. 3.7 şeklinde ifade edilir [21].

$$\rho = a\Delta x \cos\theta + b\Delta y \sin\theta \quad (3.7)$$

3.2.4. Geometrik işlemler

Taşıma: Resimdeki her pikseli yeni konuma yerleştirme işlemidir. Diğer bir ifadeyle resimdeki (x_1, y_1) koordinatındaki her pikseli $(\beta x, \beta y)$ boyunca taşıyarak yeni konuma, (x_2, y_2) koordinatına yerleştirmektir.

Döndürme: Resimde belirlenen (x_1, y_1) koordinatını bir nokta etrafında belirli bir açıyla (x_2, y_2) koordinatına döndürme işlemidir. Şekil 3.17'de $P(x_1, y_1)$ noktasının $P(x_2, y_2)$ koordinat noktasına θ açısı kadar döndürüldüğü gösterilmiştir.



Şekil 3.17. Döndürme işlemi [22]

Aynalama: Döndürme işlemini belirlenen eksen veya bir nokta etrafında yansıtarak yapmasıdır.

Ölçekleme: Komşu pikseller arasında istatistiksel bir örnekleme yaparak oluşturulacak pikselin değerini belirler.

Küçültme / Büyültme: Giriş resmini x ve y yönünde birer piksel atlayarak okunan değerleri yeni resimde sırayla yerleştirirsek 1/2 x lik bir küçültme sağlamış oluruz.

Eğme / Kaydırma: Resmin bir tarafı sabit dururken diğer tarafının x eksenini ya da y eksenini doğrultusunda kaydırmak için bir matris kullanılır [22].

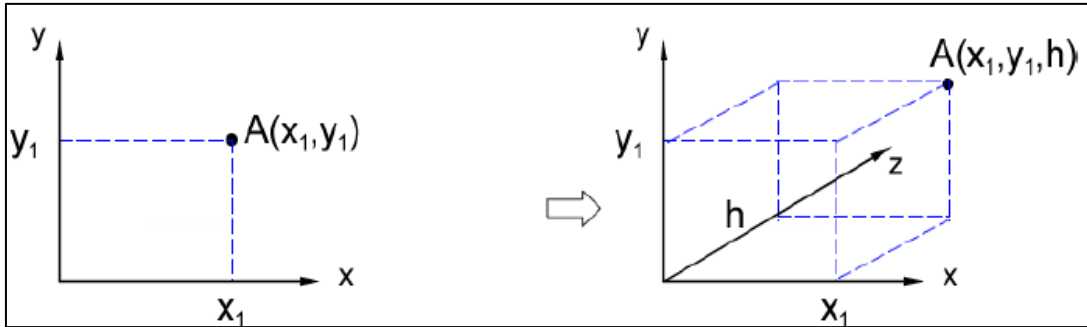
Affine Dönüşüm: Görüntünün köşe noktalarına itme ve çekme işlemleri uygulanarak, görüntüye döndürme veya yeniden ölçeklendirme gibi işlemlerin yapılmasını sağlayan dönüşüm işlemidir. Affine dönüşümü sonrasında noktaların paralellikleri aynı kalır. Bu dönüşümle kamera açılarından kaynaklanan geometrik bozulmaları veya deformasyonları düzeltilebilir. Bir görüntü üzerinden seçilen x, y noktasındaki pikselin geometrik dönüşümü, bu pikselin yeni bir konuma x_1, y_1 taşınmasıdır [23]. Şekil 3.18’de affine dönüşümü resmedilmiştir.



Şekil 3.18. Affine dönüşümü gösterimi [23]

Perspektif Düzeltme: Görüntü içerisindeki bir nesnenin 2 boyutlu bir düzleme izdüşümünün alınmasıdır. Bu işlemle resim üzerindeki görüntülere farklı açılardan bakabilmemiz sağlanır.

Nesnelerin koordinatları matris olarak yazılır ve çarpım notasyonu uygulanarak yeni konumu elde edilir. Kartezyen koordinat sistemindeki nesnenin koordinatları $A(x_1, y_1)$ şeklindedir. Bu koordinatlara üçüncü bileşen eklenmesi boyutu arttırarak yeni bir homojen koordinat sistemini oluşturur. Şekil 3.19’da kartezyen ve homojen koordinat sistemleri gösterilmiştir.



Şekil 3.19. Kartezyen ve homojen koordinatlar [23]

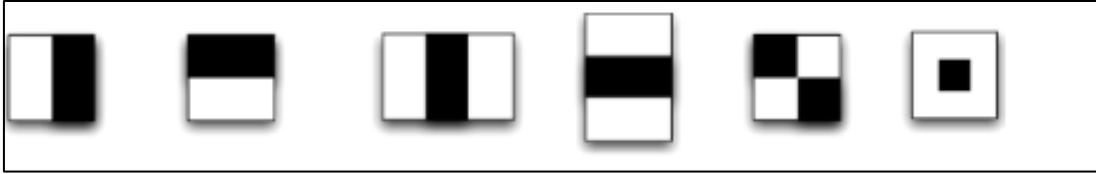
Kartezyen koordinat sisteminde $A(x_1, y_1)$ koordinatlarına h değerinin eklenmesiyle $A(x_1, y_1, h)$ elde edilmesidir.

Döndürme matrisinin homojen koordinatlar kullanılarak gösterimi Eş. 3.8’de verilmiştir. Perspektif dönüşümü matrisin çarpılması ile yapılmaktadır. x_1 ve y_1 giriş koordinatlarını x_2 ve y_2 ise çıkış resmindeki koordinatları vermektedir [23].

$$\begin{bmatrix} x2 \\ y2 \\ z \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x1 \\ y1 \\ 1 \end{bmatrix} \quad (3.8)$$

3.2.5. Haar Cascade nesne takibi ve Viola Jones nesne dedektörü

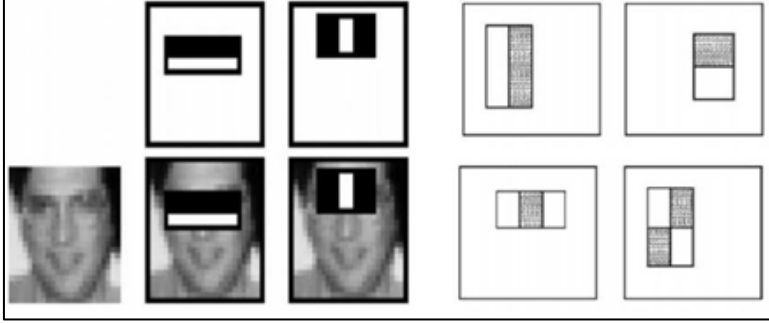
Haar Cascade algoritması nesne tespiti için en çok kullanılan algoritmalarındandır. Görüntü üzerinde yapılacak işlemlerin hızı işlemin zorluğu ve çokluğuna bağlıdır. Videoların görüntü karelerinden oluştuğu düşünüldüğünde her kare üzerinde tekrar tekrar yapılan işlemler doğruluğu ve doğru sonuca ulaşma hızını artırır. Haar Cascade sınıflandırıcı da bu amaçla doğmuştur. Temel yapısı görüntü üzerinde bir çerçevenin gezdirilmesidir. Bu çerçeve en boy oranın fazla değişmeyen nesnelerin tespiti ve bu nesnelerin birbirleriyle olan ilişkilerini eğitmeyi amaçlar. Görüntü üzerinde nesnelerin boyutlarında modifiye işlemi yapılırken en boy oranları değişmemektedir. Görüntülerde en boy oranı değişen nesneler için farklı bir eğitilmiş sınıf kullanılması gerekir. Haar Cascade sınıfları, eğitilmiş sınıflandırıcılara sahiptir. Şekil 3.20’de Haar Cascade örnek çerçeveleri gösterilmektedir.



Şekil 3.20. Haar Cascade çerçeveleri [23]

Haar Cascade algoritması, eğitilen Haar sınıflarını kullanır. Sınıflandırıcı, pozitif ve negatif örnekler kullanılarak eğitilir. Pozitif örnekler, nesneye ait bölgelerden alınmış aynı boyuttaki alanları temsil ederken negatif örnekler, nesnenin olmadığı diğer bölgelerden alınmış aynı boyuttaki alanları temsil etmektedir. Sınıflandırıcı bu örnekler üzerinde pozitif veya negatif etiketleme yapan ve görüntüdeki nesneleri algılayan çerçeveyi yana ve aşağı kaydırarak tarama yapar. Her tarama sonucunda çerçeve görüntü üzerinde bulunduğu yeri, negatif olarak etiketlemesi, bu alanın sınıflandırılmasının bittiğini alanda hiçbir nesnenin bulunmadığını ve çerçevenin bir sonraki alana kaydırılması gerektiği anlamına getirir. Fakat çerçeve bulunduğu yeri, pozitif olarak etiketler ise; bu alanda bir nesnenin bulunduğunu ve sınıflandırıcının bu alanı diğer işlemler için sonraki aşamaya geçirmesi gerektiği anlamına gelir. Eğitimden geçmiş sınıflandırıcıya daha önceden bulunması istenen nesne giriş olarak

tanımlandığında sınıflandırıcı nesnenin bulunduğu alan için 1 diğer alanlar için 0 çıkışı üretir. Haar Cascade resim üzerinde çerçeve hareketleri Şekil 3.21’de verilmiştir [23].



Şekil 3.21. Haar Cascade resim üzerinde çerçeve hareketleri [23]

Haar Cascade eğitimi dört aşamada gerçekleşir;

1. Haar özellik seçimi
2. İntegral görüntü oluşturma
3. Adaboost eğitimi
4. Basamaklı sınıflandırıcılar

Haar Özellik Seçimi

Haar benzeri özellikleri temel olarak 3 ana bölgeye ayrılır.

- Kenar özellikleri; Görüntü üzerinde belirli alanlar koyu, belirli alanlar açık renklerden oluşan özelliktir.
- Çizgi özellikleri; Görüntü üzerindeki alanlar sırasıyla açık, koyu, açık renklerden oluşan özelliktir.
- Dört kare veya merkez çevre özellikleri; Kare içindeki renklerin çaprazlama olarak açık ve koyu renklerden oluşması dört-kare özelliğini, bir karenin merkezdeki rengi koyu diğer renklerin açık olmasıyla oluşan özelliktir [23].

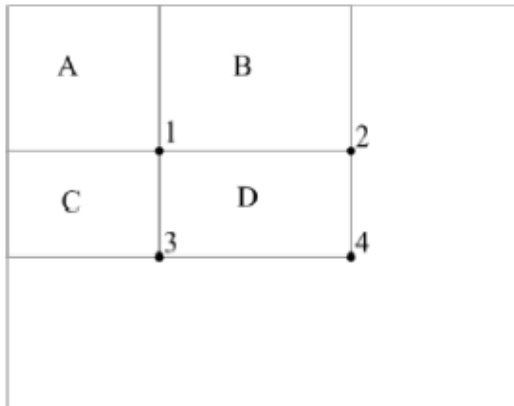
Tarama yapıldıktan sonra görüntü alt bölümlere ayrılır. Bu alanda nokta yoğunlukları toplanır ve toplamlar arasındaki fark hesaplanarak kategorilere ayrılır.

İntegral Görüntüler Oluşturma

Bu öznelikleri hızlı bulabilmek için integral imge kullanılır. Yaptığı işlem görüntü üzerindeki herhangi bir (x,y) konumundaki yeri, görüntünün üstten ve soldan piksellerin toplanmasıdır. İntegral imgenin hesaplanması Eşitlik 3.9'da gösterilmektedir [23].

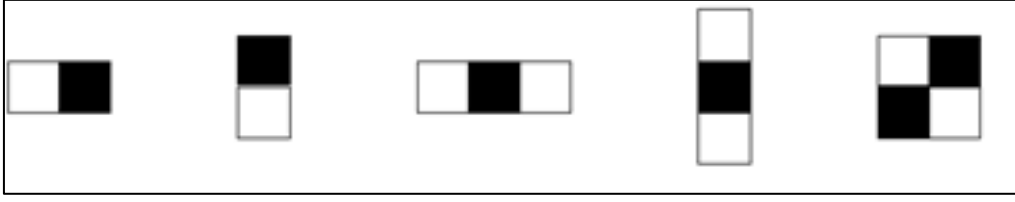
$$ii(x,y) = \sum_{x' \leq x, y' \leq y} i(x',y') \quad (3.9)$$

Şekil 3.22'de Viola-Jones nokta gösterimi verilmiştir. A dikdörtgen bölgenin içerisindeki piksel değerlerinin toplamı 1 konumundaki değerdir. 2 konumundaki değer A ve B bölümlerindeki piksel değerlerinin toplamına eşittir. 3 konumundaki değer ise A ve C nin toplamıdır. Son olarak 4 konumundaki değeri de A+B+C+D'dir. D bölümündeki piksellerin toplam değeri $4+1-(2+3)$ şeklinde hesaplanır [24].



Şekil 3.22. Viola - Jones dört nokta gösterimi [24]

Viola-Jones detektörü iki veya daha fazla dikdörtgenden oluşan bir alt pencere vererek analiz yapar. Şekil 3.23'de dedektör tipleri gösterilmektedir. Her tip, siyah alanlardaki toplam pikselden beyaz alandaki toplam pikselin çıkarılmasıyla elde edilen tek bir değeri ortaya koyar. Sonrasında AdaBoost yöntemi kullanılarak elde edilen verilerle sınıflandırma yapılır. Böylece işlem hızı daha da artmaktadır.



Şekil 3.23. Viola – Jones dedektör tipleri [24]

AdaBoost Eğitimi (Sınıflandırıcının Eğitilmesi)

Adaboost, bir sınıflandırıcıyı hızlandırmak için kullanılmasına rağmen Viola ve Jones Adaboost işlemi, hem sınıflandırıcıyı eğitmek hem de özneliliklerin seçilmesi için kullanılmıştır. Haar Cascade algoritmasının ilk kullanım zamanlarına karşın sonraki zamanlarda grafik işlemci kartlarının da sınıflandırma sürecine dâhil edilmesi algoritmanın hızını arttırmıştır. İmgeyi farklı ölçeklendirilmiş çerçevelerle görüntü üzerinde hareket ettirildiğinde sınıflandırıcı o çerçevenin aranan nesne olup olmadığını ayırt edecek şekilde eğitebilecek kapasiteye sahip olmuştur. Eğitim işleminde kullanılan bütün çerçevelere de elde edilen özneliliğin az eğitilmiş bir sınıflandırıcı için kullanılır. Adaboost algoritmasındaki zayıf sınıflandırıcıların her biri doğru özneliliğe sahip eğitilmiş sınıflandırıcı gibi düşünürler, bunların içinde pozitif ve negatif örnekleri ayırabilen en iyi sınıflandırıcıyı tespit etmeye çalışmışlardır [23].

Basamaklı Sınıflandırma

Zayıf olan bir sınıflandırıcı, görüntü içinde aranan bir nesneyi bulma yeteneği olmasına rağmen, yine de nesneyi bulmak için yetersiz kalmaktadır. Bunun için aranan nesneyi en iyi şekilde bulacak çok sayıda Haar özelliği gerekir. Bulunan bu Haar özellikleri birleştirilerek kademeli sınıflandırıcılar halinde düzenlenir. Amaç, eğitim setindeki örneklerin içindeki negatif örnekleri en kısa sürede reddedip aranan nesnenin görüntülerin çoğunda olmadığını varsaymaktır. Görüntü üzerinde aranan nesneyi bulabilen bir sınıflandırıcıdan gelen pozitif sonuç, ilk sınıflandırıcının yanlış-artıları (aranan nesnenin olmadığı pozitif örnekleri) eleyebilmek için daha az işlem ile eğitilebilen ve tespit oranı yüksek olan ikinci sınıflandırıcıyı tetikler. Eğer ikinci sınıflandırıcı pozitif sonuç üretirse, eğitilebilen ve tespit oranı yüksek olan üçüncü sınıflandırıcıyı tetikler ve bu işlem herhangi

bir negatif sonuç alınana kadar devam eder. Böylece buraya kadarki bütün sınıflandırıcılar AdaBoost ile eğitilmiş olur [23].

3.2.6. Yolov3 ile nesne tanıma

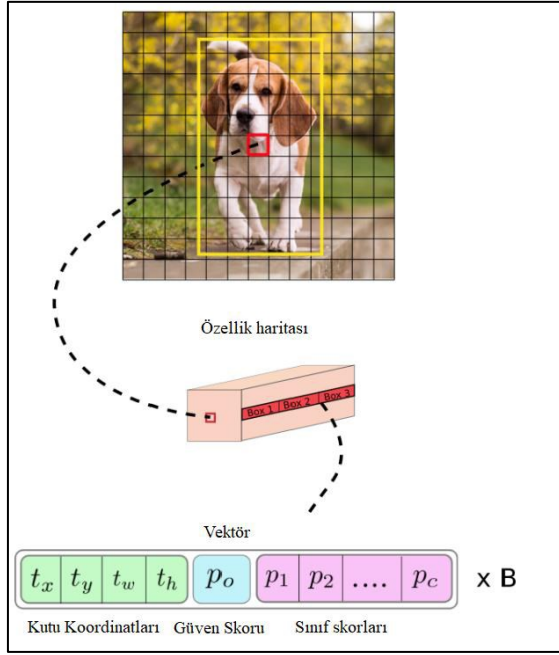
YOLO, “*You Only Look Once*” kelimelerinin baş harflerinden oluşmuş, günümüzde en hızlı ve yüksek doğrulukla çalışan, insan gözünün görme sistematüğinden esinlenerek geliştirilmiş derin öğrenme tabanlı bir nesne tanıma mimarisidir. Konvolüsyon sinir ağlarında görüntünün tamamından nesnenin tanıma ve belirlenme işlemlerini tek seferde yaparak, nesne sınırlama ve sınıflandırma için resmin tamamını tek seferde işler [25].

YOLOv3-320 metodu 320x320 giriş çözünürlüğüne sahip YOLOv3 modeli olarak adlandırılmaktadır. Giriş çözünürlüğünün artması ile ortalama doğruluk artmakta ancak algılama süresi gecikmektedir. Bu sebeple yapılandırma aşamasında YOLOv3 modelini eğitmeden önce parametre seçimi büyük önem taşımaktadır. Hızlı çalışan bir model istendiğinde doğruluktan fedakârlık yapılması gerekirken, doğruluk hassasiyeti istenen durumlarda da hızdan feragat edilmesi gerekir. YOLO algoritması görüntüyü tek seferde sinir ağından geçirerek resimdeki nesneleri ve koordinatları tahmin etmesi sayesinde hızlı bir algoritmadır [25].

YOLO giriş görüntüsünü SxS boyutunda ızgaraya bölmektedir. Konfigürasyon dosyasında 416x416 şeklinde bir çözünürlük değeri ayarlanmışsa ve her ızgaranın piksel değeri 32 ise, oluşturulan model 13x13'lük bir ızgara oluşturur. Her ızgara içerisinde nesne barındırıp barındırmadığını eğer bulunuyorsa orta noktasının ızgara içerisinde olup olmadığını bulmakla sorumludur. Tahmin işleminde her ızgara bir tahmin vektörü oluşturmaktadır. Her bir ızgara için oluşturulabilecek tahmin vektörü sayısı Eş. 3.10'da verilmiştir.

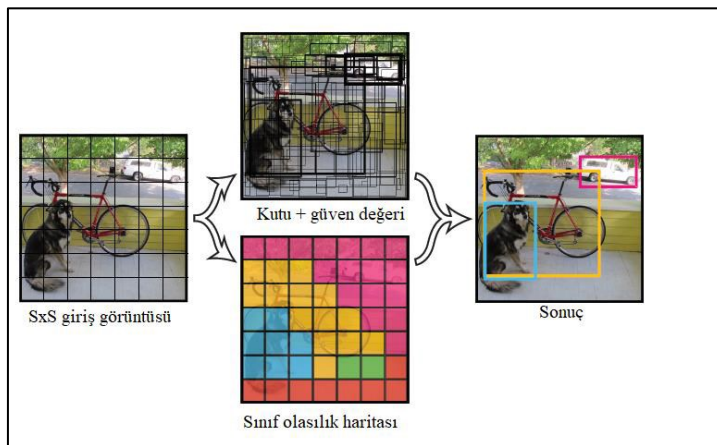
$$\text{Tahmin Vektör Sayısı} = B \times (5+C) \quad (3.10)$$

Bu eşitlikte kullanılan B , bağlama kutusu sayısıdır. Bu kutular nesne etrafındaki sınır kutucuğunun çizilmesinde kullanılan geometrileri önceden belirlenmiş kalıplardır. C , ise YOLO'nun tanıyabildiği nesne sınıfı sayısını göstermektedir. YOLO modeli kaç sınıfa göre eğitilirse C değeri de o olmaktadır. Vektörde her bir sınıf için bir olasılık değeri üretilir. Vektörler Şekil 3.24'de gösterildiği gibi içerisinde belirtilen değerleri tutmaktadır.



Şekil 3.24. Özellik haritası ve tahmin vektörü [25]

Şekil 3.24’de yer alan güven skoru (p_0), modelin geçerli ızgara içerisindeki nesne bulunma olasılığıdır. Izgarada kesinlikle bir nesne yoksa 0, bir nesne varsa 1 değeri almaktadır. t_x ve t_y , nesnenin orta noktasının x ve y koordinatlarını, t_w genişliğini ve t_h ise yüksekliğini göstermektedir. Nesne tanımda girişe uygulanan her bir görüntü için $S \times S \times B$ adet tahmin vektörü algoritma tarafından üretilir. Her bir tahmin vektörü için bir kutu çizilir. Ancak kutuların büyük bir kısmı elimine edilir ve sadece yüksek güven skoruna sahip olan kutular kalır. Sonrasında yüksek güven skoruna sahip olan bu kutuların içerisinde üst üste çakışanlardan en yüksek skorlu olanı seçilir. Nesne tanıma işleminin aşamaları Şekil 3.25’te görülmektedir.



Şekil 3.25. Nesne tanıma işleminin aşamaları [25]

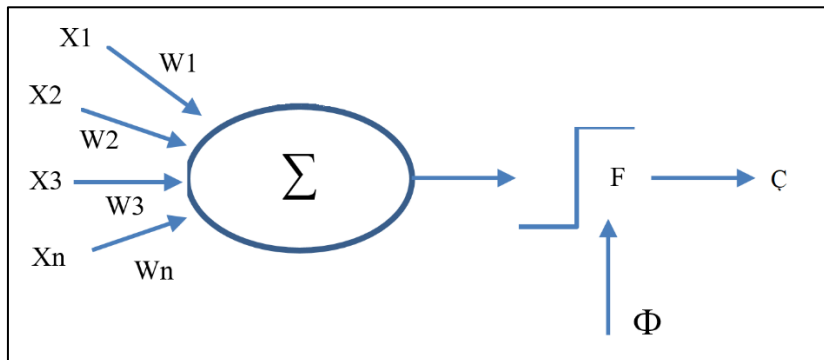
4. YAPAY SINIR AĞLARI VE MAKİNE ÖĞRENMESİ

Yapay sinir ağları (YSA), insan beyninden ilham alınarak bir olay veya durumun içeriğinden öğrenme yoluyla çıkarımda bulunup, farklı bilgiler türetebilen bunu da destek almadan gerçekleştirebilen sistemlerdir [26]. YSA insan beynine iki yönüyle benzemektedir. Birincisi, bilgiyi elde eder, ikincisi ise bilgiyi nöronlar (sinir hücresi) arası ağırlıklar saklar [27].

Yapay sinir ağları süreç elemanlarının birbiri ile bağlanması sonucu oluşur. Ağ temelde bir veri setini işleyerek sonuç veri seti ortaya koyacak şekilde çalışır. Genel özellikleri arasında; bilginin bağlantılar arasında konumlanması, örneklerle eğitimi gerçekleştirmesi, görülmemiş verilerden bilgi üretmesi, verilerin bir bölümü eksik olsa da çalışma, bilgilerin dağınık olması ve yalnızca sayısal bilgilerle çalışabilmesi bulunur.

4.1. Yapay Sinir Hücresi

YSA birbirine bağlı sinir hücrelerinden oluşur. Sinir hücresi temel bilgi işleme birimidir. Alınan bilgiler (X) belirlenen ağırlıklarla (W) çarpılarak toplanır. Elde edilen toplam aktivasyon fonksiyonundan (F) geçerek çıktı oluşturulur. Elde edilen çıktı önceden belirlenen çıktılar ile karşılaştırılır ve fark oluşmuşsa fark olarak kabul edilir. Ortaya çıkan fark olması gereken değerle oranlanarak hata oranı elde edilmiş olur. Hata oranına göre ağırlıklar (W) tekrar belirlenir. İşlem yeni ağırlıklarla tekrar yapılır. Ağın eğitilmesi ağırlıkların güncellenip sürecin ilerletilmesi işlemidir. Yapay sinir hücresini oluşturan elemanlar Şekil 4.1’de gösterilmektedir. Buradaki Φ ifade eşik değeridir.



Şekil 4.1. Yapay sinir ağı hücresi [27]

Girdiler: Hücreye gelen ve başka bir yapay sinir ağı hücresinden olduğu gibi ağından dışından da elde edilebilen bilgilerdir. X ile gösterilir.

Ağırlıklar: Hücreye iletilen bilginin önemini ve hücreye etkisini gösterir. Öneminin göstergesi değerin büyük ya da küçük olması ile ilgili değildir. Yine aynı şekilde artı veya eksi olması yalnızca yönünün pozitif ya da negatif olduğunu gösterir. Ağırlıklar sabit veya değişken olabilir ve nöronlar arasında iletimini sağlayan tüm bağlantılarında bulunurlar. Böylece her girdi üzerinde etkisi bulunmaktadır [26].

Birleştirme Fonksiyonu: Birleştirme işlemi, toplama işlevi olarak tanımlanmaktadır. Hücreye gelen net girdiyi hesaplar. Ağırlıklı toplamı bulmayı amaçlar. Eş. 4.1'de işlem gösterilmektedir.

$$Net = \sum_i^n X_i W_i \quad (4.1)$$

Burada, X : girdi, W : ağırlık ve Net ise hücreye gelen toplam girdilerdir. Genel kullanımda fazlaca yer alsaydı da birleştirme fonksiyonu olarak yalnızca toplama işlemi kullanılmamaktadır. Buna ek olarak kullanılan farklı yöntemlerde bulunmaktadır. Hangi yöntemin kullanılacağına dair seçim deneme yanılma ile yapılabilmektedir. Kullanılan diğer birleştirme fonksiyonları aşağıdaki gibidir [26].

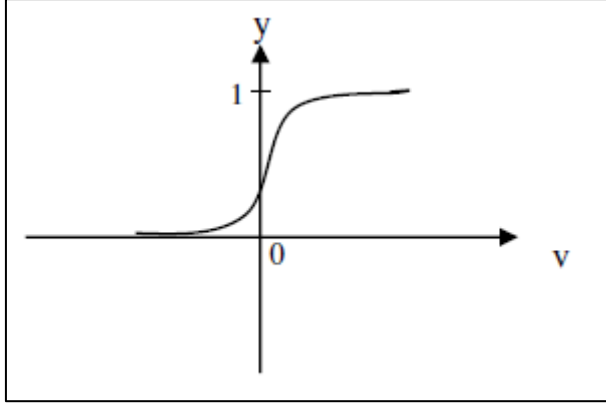
Çarpım fonksiyonu;	$Net\ Girdi = \prod G_i A_i$
Maksimum fonksiyonu;	$Net\ Girdi = \max (G_i A_i), i = 1 \dots N$
Minimum fonksiyonu;	$Net\ Girdi = \min (G_i A_i), i = 1 \dots N$
Çoğunluk fonksiyonu;	$Net\ Girdi = \sum \text{sgn} (G_i A_i)$
Kümülatif fonksiyonu;	$Net\ Girdi = Net(\text{eski}) + \sum (G_i A_i)$ [26]

Uygulamaların bazılarında eşik değeri Φ olan bu toplamaya katılmıştır. Eşik değerinin ağı yapısı içindeki görevi; bütün girişler sıfır olursa çıkış değerini sıfır olmasından kurtararak belirlenen eşik değerine eşitleyecektir [27].

Aktivasyon Fonksiyonu: Yapay sinir ağının önemli faktörüdür. Farklı ağ yapıları için farklı aktivasyon fonksiyonları kullanılabilir. Burada da olması gereken veya en iyi aktivasyon

fonksiyonu şudur deme şansı bulunmaktadır. Yine deneme yanılma yöntemiyle sonuca gidilebilmektedir. Sigmoid fonksiyonu yaygın olarak kullanılmaktadır [26]. Sigmoid fonksiyonu Eş. 4.2’de, grafik gösterimi de Şekil 4.2’de verilmiştir.

$$F(\text{NET}) = \frac{1}{1 + e^{-\text{NET}}} \quad (4.2)$$



Şekil 4.2. Sigmoid tipi fonksiyon [26]

Lineer Fonksiyonu: $F(\text{NET}) = \text{NET}$, girdi bilgileri çıkış bilgilerine eşit olarak kabul edilir.

Step fonksiyonu: Eşik değerinin devrede olmadığı durumdur.

$$F(\text{Net}) \begin{cases} 1 & \text{Eğer } \text{Net} > \text{eşik değeri} \\ 0 & \text{Eğer } \text{Net} \leq \text{eşik değeri} \end{cases}$$

Sigmoid fonksiyonu: $F(\text{NET}) = \sin(\text{NET})$ öğrenilecek örneklerin, eğitim veri seti de denebilir, sinüs eğrisine uygun dağılım gösterdiği durumlarda kullanılır. En aktif bölgesi 0,2 - 0,8 arasındadır.

Eşik fonksiyonu: Gelen bilgilerin 0 veya 1 aralığında değer olmasına göre sınıflanır.

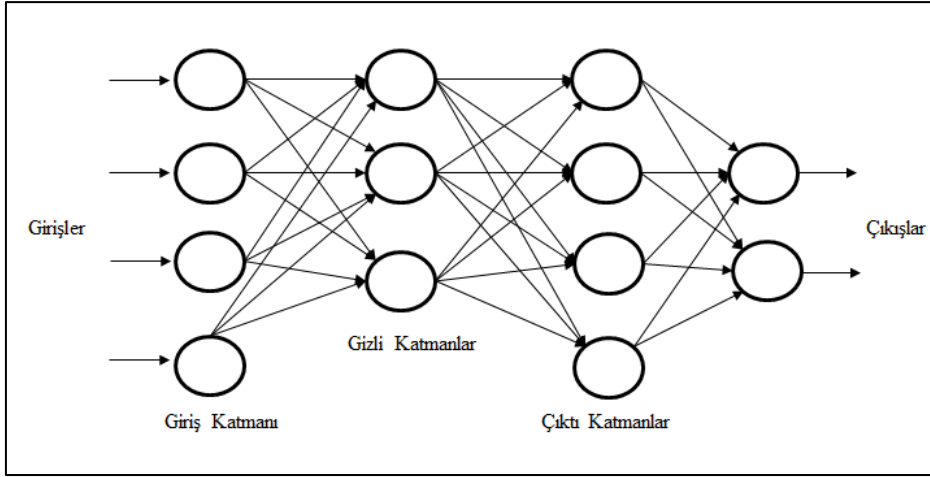
$$F(\text{Net}) \begin{cases} \text{Eğer } \text{Net} \leq 0 \\ \text{Eğer } 0 < \text{Net} < 1 \\ \text{Eğer } \text{Net} \geq 1 \end{cases}$$

Hiperbolik tanjant fonksiyonu: Net girdi değerlerinin tanjant fonksiyonu ile hesaplanmasıdır [26].

Çıktı: Aktivasyon fonksiyonu sonrasında ortaya konulan çıktı değeridir. Sonuçlanan değer bir başka sinir ağ hücresine veya dış dünyaya gönderilir. Ağın sadece bir çıktısı mevcut olur.

4.2. Yapay Sinir Ağlarının Yapısı

Yapay sinir ağları, yapay sinir hücrelerinin bir araya gelmesiyle oluşur. Bu sinir hücreleri belirli katmanlar içerisinde barındırılır. Bu birliktelik bir düzen içerisinde oluşturulur. Genellikle üç katmandan oluşur. Bu katmanlar birbirleri arasında paralel yapı ile oluşturulur. Her katmanda bulunan düğümler bir önceki katmandan giriş alabilir. Son katmanda yer alan düğümler ise ağın çıkışıdır. Şekil 4.3'te çok katmanlı bir ağın yapısı yer almaktadır [27].

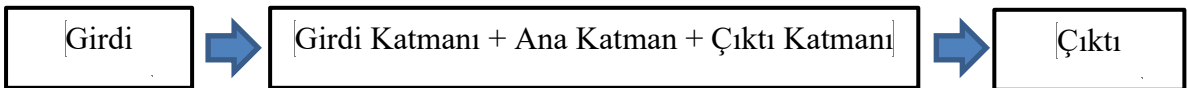


Şekil 4.3. Çok katmanlı ağ yapısı [27]

Girdi Katmanı: Bu katman süreç elemanları ağın dışından gelen bilgileri ara katmanlara iletmekle sorumludur.

Ara katmanı: Girdi katmanından aldığı bilgileri işleyerek çıktı katmanına gönderir. Birden fazla ara katman bulunabilir. Gizli katman olarak ta adlandırılır [28]. Ara katman ve katmandaki nöron sayısı ağı oluşturan uygulamacının tasarrufundadır. Gereğinden az nöron kullanımı ağın eğitim doğruluk performansını azaltırken, gereğinden fazla nöron kullanımı ise yeni girdi değerlerine göre çıktı tahminlerinin doğruluk oranları azalacaktır [27].

Çıktı katmanı: Ara katmanlardan gelen bilgiyi işleyerek çıktıyı, sonuç bilgisini üretirler. Elde edilen çıktı bilgisi bu katmanda ortaya konur. Şekil 4.4'te bilgi akışı gösterilmiştir [28].



Şekil 4.4. YSA bilgi akışı [28]

4.3. Yapay Sinir Ağlarının Çalışması

Ağ için belirlenmiş veriler vektöre dönüştürülerek ağa iletilir ve ağ da bu bilgi vektörüne göre çıktı vektörünü oluşturur. Elde edilen çıktıların istenen çıktı değerleri olmaması durumunda, katmanlar arasındaki nöronların bağlantılarında bulunan ağırlık değerleri değiştirilerek tekrar işlem yapar. İstenilen sonuca ulaşılan kadar bu süreç devam eder. Bu işleme “ağın eğitilmesi” denir. Ağırlık değerleri ağın ilk çalışmasında rasgele belirlenir. Eğitimde kullanılan girdi ve çıktı değerlerinin tümüne de “eğitim kümesi” denir [27].

Girdi ve çıktı vektörün tanımı tasarımı gerçekleştiren tarafından belirlenir. YSA çıktıların nasıl hazırlandığının yorumunu vermez [26].

İterasyon (epoch) kelimesi, tüm eğitim kümesinin ağa verilmesi olarak ifade edilir. Eğitim kümesine göre istenilen çıktı değerlerine ulaşıldığında mevcut ağırlık değerleri korunur. Artık ağın problem çözümünde genelleme yapabileceği söylenebilir ve bu genelleştirme özelliğini taşıma işlemine de “ağın öğrenmesi” denilir. Ağırlıkların değişimi durdu ise öğrenme işlemi de durmuş demektir [27].

Ağın eğitimi tamamlandıktan sonra eğitimin başarılı olup olmadığını ölçme işlemine “ağın test edilmesi” denir. Test işleminde önceden hiç kullanılmamış test setleri kullanılır.

Ağın eğitilmesi sonrasında hata miktarlarında azalma ve test sonuçlarında performansın artmasına sebebiyet veren durumlardan biri “ağın ezberlemesi” olarak tanımlanır. Bu kullanıcıyı yanıltan bir durumdur. Ezberleyen ağ gerçek hattaki görüntüyü temsil etmediğinden kullanılamaz [27].

4.4. YSA'nın Sınıflandırılması

Ağın hücre yapısı öğrenme algoritmasıyla ilişkilidir. Diğer bir ifadeyle hücreler arasındaki bağlantı öğrenmenin nasıl olacağı ile alakalıdır. Eğitim de hatanın istenilen düzeye çekilmesinde ağırlıkların değiştirilmesidir. Bunlara bağlı olarak algoritmalar ağın yapılarına göre ve öğrenme algoritmalarına göre sınıflandırılırlar.

4.4.1. YSA'nın yapılarına göre sınıflandırılması

İleri Beslemeli Ağlar: İşlemciler katmanlara ayrılmıştır ve bilgi akışı giriş katmanından çıkış katmanına olacak şekilde tek yönlü olur. Bu akış sırasında orta ve çıkış katmanlarında işlenen bilginin çıktısı elde edilir [26].

Geri Beslemeli Ağlar; Bu ağın yapısı, çıkış ve ara katmanlarda oluşan çıktı verisini giriş veya ara katmanlara geri yönlendirmesi şeklindedir. Bilginin akışını ileri ve geri yönde akışını sağlamış olur. Bu ağ yapısının bir özelliği de çıkış verisi olarak ortaya konulan değer aynı zamanda diğer katman için girdi verisi olmaktadır [26].

4.4.2. YSA'nın öğrenme algoritmalarına göre sınıflandırılması

Danışmanlı Öğrenme: Ağa örnek olarak bir veri seti verilir ve sonrasında ağda elde edilen sonuçlar ile hedef çıktılar arasındaki fark hatayı ifade ettiği için bu hatayı en aza indirmeye çalışılır. Bunun için de ağların ağırlıkları değiştirilerek sonuçlar tekrar kontrol edilir. İleri beslemeli ağlarda çoğunlukla bu yöntem kullanılır.

Danışmansız Öğrenme: Bu öğrenme modelinde danışmanlı modeldeki gibi örnek sonuç veri seti verilmez, yalnızca girdiler verilir. Girişe verilen örnek veriler çıkış bilgisine göre sınıflandırma kendi kendine geliştirir.

Takviyeli Öğrenme: Bu yöntem danışmanlı öğrenme yöntemine benzer, ağın performansını ifade eden skor veya derece bildirir [26].

4.4.3. YSA temel öğrenme kuralları

Hebb Kuralı: Öğrenme algoritmalarının temeli sayılır. Bu öğrenme kuralında girdi ve çıktı arasındaki işaretler aynı ise nöronlar arası ağırlıklar artar. Bunun sonucu olarak nöronlar arasındaki bağlantı güçlenir.

Hopfield Kuralı: Zayıflama ve kuvvetlendirme büyüklüğü dışında Hebb kuralına benzemektedir. Bu yöntemde bağlantıların kuvvetlendirilme veya zayıflatma miktarı belirlenir. Bu işlemi öğrenme katsayısı kadar azaltıp artırarak yapar. Öğrenme katsayısı genellikle 0-1 arasındadır ve uygulayıcı tarafından belirlenir.

Delta Kuralı: Bu kuralda amaç ağ hatasının karelerinin ortalamasını minimize etmektir. Hataların karelerinin en küçük olduğu nokta dereceli azaltma yöntemi ile tespit edilir. Bunu da; hataların kareleri toplamının her bir ağırlığa göre kısmi türevlerini alarak ve negatif türev değerlerini kullanarak yapmaktadır.

Kohonen Kuralı: Bu eğitim kuralında ağırlıkların ayarlanması için işlemci elemanlar aralarında yarışır. Bunun sonucu olarak, tek seferde, yalnızca kazanan işlemci elemanın ağırlığı değişir. Kohonen kuralı hedef çıkışa gereksinim duymayıp danışmansız öğrenme ile eğitimi gerçekleştirilir [27].

4.4.4. YSA’da en çok kullanılan modeller ve öğrenme kuralları

Modeller:

- Basit algılayıcı – Tek katmanlı algılayıcılar (Perseptron)
- Adaline
- Çok katlı algılayıcılar (ÇKA)
- Radyal tabanlı sinir ağı
- LVQ (Learning Vector Quantisation)
- Ceza Mekanizmalı (LVQ)

Öğrenme Kuralları:

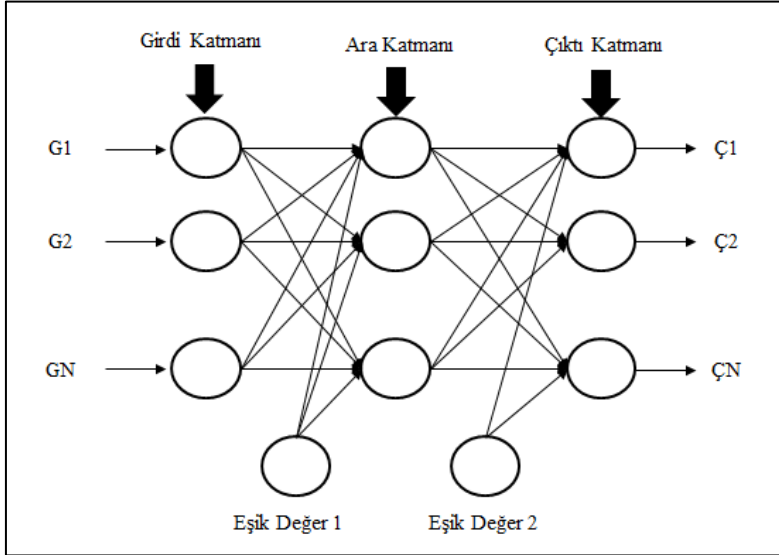
- Levenberg Marquat öğrenme kuralı
- BFGS öğrenme Kuralı
- Kohonen kuralı
- Hebb kuralı
- Hopfield kuralı
- Delta kuralı
- Öğrenme algoritmaları
- Geri yayılım algoritması
- Genetik algoritma [27].

4.5. Çok Katmanlı Ağlar

Tek katmanlı ağların doğrusal olmayan problemleri çözme yeteneği olmadığından bu sorunu çözebilmek için birçok model geliştirilmiştir. Çok katmanlı algılayıcılar bu modellerden birisidir. Olayın doğrusal olmaması, problem çıktılarının grafik üzerinde bir doğru ile iki veya daha fazla sınıfa ayırılamaması durumudur. Genelde delta öğrenme kuralı öğrenme yönetimi kullanılmaktadır [27].

4.5.1. Çok katmanlı ağların yapısı

Şekil 4.5'te çok katmanlı ağın hücre ve bağlantı yapısı görülmektedir.



Şekil 4.5. Çok katmanlı ağ yapısı [27]

Girdi Katmanı: Gelen girdi bilgileri ($G_1, G_2, \dots, G_N$) olarak ara katmana gönderir. Birden fazla girdi gelebilir.

Ara Katmanlar: Girdi ile çıktı katmanları arasında kalan katmandır. Girdi katmanından gelen bilgiler burada işlenerek sonraki katmana gönderilir. Ara katman sayısı ve bu katmanlardaki süreç elemanı birden fazla olabilir.

Çıktı Katmanı: Bu katman, ara katmandan aldığı bilgilerden oluşturduğu çıktıları ($\Ç_1, \Ç_2, \dots, \Ç_N$) belirleyerek ağ dışında iletir [26].

4.5.2. Çok katmanlı ağların öğrenme kuralı

Çok katmanlı ağlar öğretmenli diğer bir ifadeyle danışmanlı öğrenme yöntemini kullanır. Ağa örneklerle birlikte örneklerden elde edilmesi istenen sonuçlarda verilmektedir. Problem veri setini temsil eden çözüm ağ tarafından bu örneklerden genellemeler yapılarak üretilmektedir. Sonrasında benzer örnekler için üretilmiş model ile çözümler ortaya konulabilmektedir.

Çok katmanlı ağların öğrenme kuralına delta kuralı da denilmektedir. Bu kural en küçük kareler metodunu kullanmaktadır. Bu sebeple bu öğrenme kuralına genelleştirilmiş delta kuralı da denilir. Öğrenme iki basamakta olur [27].

İleri Doğru Hesaplama

Eğitim için hazırlanmış veri setinden bir örneğin girdi katmanına iletilmesiyle başlar (G1, G2, G3...). Burada iletilen girdi bilgileri değişikliğe uğramadan ara katmana gönderilir. Girdi katmanındaki k ve süreç elemanın çıktısı ζ_k , Eş. 4.3'deki gibi ifade edilebilir.

$$\zeta_k^i = G_k^i \quad (4.3)$$

Ara katmandaki işlem için kullanılacak elemanlar, girdi katmanındaki işlem elemanlarından gelen bilgiyi bağlantı ağırlıklarının etkisiyle almaktadır ($A_1, A_2..$). Eş. 4.4'de gösterildiği gibi hesaplanan net girdi (NET_j^a), öncelikle ara katmandaki işlem elemanlarına gelmektedir [27].

$$NET_j^a = \sum_{k=1}^n A_{kj} \zeta_k^i \quad (4.4)$$

Bu eşitlikte, $A_{kj} \zeta_k^i$ girdi katmanı elemanına ait ağırlık değerini ifade etmektedir. Bu net girdi aktivasyon fonksiyonundan (genellikle sigmoid fonksiyonundan) geçirilerek ara katman elemanın çıktısı hesaplanır. Sigmoid fonksiyonunun kullanılmasıyla çıktı değerine ulaşılmış olur [27].

Eşitlik 4.5’de, β_j , ara katmanında bulunan eşik değer elemanının ağırlık değeri gösterilmektedir. Eşik değeri her daim 1 olarak belirlenir. Ağırlık değerinin belirlenmesinde ise sigmoid fonksiyonunu kullanılmıştır. Ara katmanı ve çıktı katmanlarında işlem elemanları NET girdinin hesaplanması ve sigmoid fonksiyondan geçirilmesiyle belirlenir [27].

$$C_j^a = \frac{1}{1 + e^{-(NET_j^a + \beta_j^a)}} \quad (4.5)$$

Geriye Doğru Hesaplama

Ağın verilen girdilere göre ürettiği çıktının hedeflenen çıktılar ($B_1, B_2, \dots$) ile karşılaştırılır. Çıktılar ile hedeflenen değerler arasında ortaya çıkan far hata olarak kabul edilir. Geriye hesaplama işleminde, bu hata ağdaki ağırlıklara dağıtılarak sonraki iterasyonda azaltılması sağlanır. Çıktı katmanındaki h. süreç elemanı için oluşan hata (E_h) olacaktır. Eş. 4.6’da hata gösterilmektedir [27].

$$E_h = B_h - C_h \quad (4.6)$$

Çıktı katmanında hataların toplamının bulunması için de tüm hatalar toplanır (TH). Bazı hata değerleri negatif olduğunda toplam hatanın sıfır olma ihtimali vardır. Bunun önüne geçmek için hataların kareleri hesaplanır ve sonucun karekökü alınır. Bu işlem Eş. 4.7’de gösterilmektedir [27].

$$TH = \frac{1}{2} \sum_m E_m^2 \quad (4.7)$$

Süreç elemanlarının ağırlıkları değiştirilerek toplam hatayı en aza indirmeye çalışılır. Herhangi bir t zamanında t. iterasyonda ağırlığın değişim miktarı Eş. 4.8’de gösterildiği gibi hesaplanmaktadır.

$$\Delta A_{jm}^a(t) = \lambda \delta_m \Delta C_j^a - \alpha \Delta A_{jm}^a(t-1) \quad (4.8)$$

Burada süreç elemanına bağlayan bağlantının ağırlığındaki değişim miktarı ΔA^a olarak gösterilmiştir. λ ve α ise öğrenme ve momentum katsayılarıdır. Ağırlıkların değişim miktarı

öğrenme katsayısı olarak adlandırılır. Çok katmanlı ağın öğrenmesi sırasında bir noktaya takılıp kalmaması amacıyla ağırlık değişim değerinin belli bir noktada bir sonraki değişime eklenmesi momentum katsayısı ile olmaktadır. Eşitlikteki δ_m ise m. çıktı ünitesinin hatasıdır. Bu hata miktarı Eş. 4.9’da gösterildiği gibi hesaplanır.

$$\delta_m = \int (NET) \cdot E_m \quad (4.9)$$

Bu eşitlikte, $\int (NET)$ aktivasyon fonksiyonunun türevini göstermektedir. Eş. 4.10’da gösterildiği gibi, sigmoid fonksiyonu kullanılarak hesaplanabilir.

$$\delta_m = C_m (1 - C_m) \cdot E_m \quad (4.10)$$

Ağırlıkların t. iterasyondaki yeni değerleri değişim miktarı hesaplandıktan sonra Eş. 4.11’de gösterildiği gibi hesaplanabilir.

$$A_{jm}^a(t) = A_{jm}^a(t-1) + \Delta A_{jm}^a(t) \quad (4.11)$$

Benzer şekilde eşik değere bağlı ağırlıkların da değiştirilmesi gerekir. Bu amaçla öncelikle değişim miktarı hesaplanır. Çıktı katmanına ait eşik değer ağırlıkları β^c ile gösterildiğinde değişim miktarı Eş. 4.12’de gösterildiği gibi olur.

$$\Delta \beta_m^c(t) = \lambda \delta_m + \alpha \Delta \beta_m^c(t-1) \quad (4.12)$$

t. iterasyonda eşik değer ağırlığının yeni değeri de Eş. 4.13’te gösterildiği gibi hesaplanır.

$$\beta_m^c(t) = \beta_m^c(t-1) + \Delta \beta_m^c(t) \quad (4.13)$$

Ağ yapısına göre geriye doğru gidildiğinde en son girdi katmanı ile ara katman arasındaki ağırlıkların değiştirilmesi gereklidir. Bu kısımdaki ağırlıkların değiştirilmesinde çıktı katmanındaki işlem elemanlarının da etkisi olmaktadır. Bu ağırlıklardaki değişimi ΔA_{kj}^i gösterilirse değişim miktarı Eş. 4.14’teki gibi olur. Hata değeri aktivasyon fonksiyonu olarak sigmoid fonksiyonu kullanıldığında Eş. 4.15’deki gibi hesaplanır.

$$\Delta A_{kj}^i(t) = \lambda \delta_j^a C_k^i + \alpha \Delta A_{kj}^i(t-1) \quad (4.14)$$

$$\delta_j^a = \int (\text{NET}) \sum_m \delta_m A_{jm}^a \quad (4.15)$$

Hata değeri elde edildikten sonra, değişim miktarı Eş. 4.16 ile, ağırlıkların yeni değeri de Eş. 4.17'deki gibi hesaplanır.

$$\delta_j^a = C_j^a (1 - C_j^a) \sum_m \delta_m A_{jm}^a \quad (4.16)$$

$$A_{kj}^i(t) = A_{kj}^i(t-1) + \Delta A_{kj}^i(t) \quad (4.17)$$

Ara katman eşik değer ağırlıklarının değişim miktarı ($\Delta\beta_j^a$) Eş. 4.18 ile hesaplanır.

$$\Delta\beta_j^a(t) = \lambda\delta_j^a + \alpha\Delta\beta_j^a(t-1) \quad (4.18)$$

Ağırlıkların yeni değerleri de eşitlik Eş. 4.19'daki gibi olacaktır.

$$\beta_j^a(t) = \beta_j^a(t-1) + \Delta\beta_j^a(t) \quad (4.19)$$

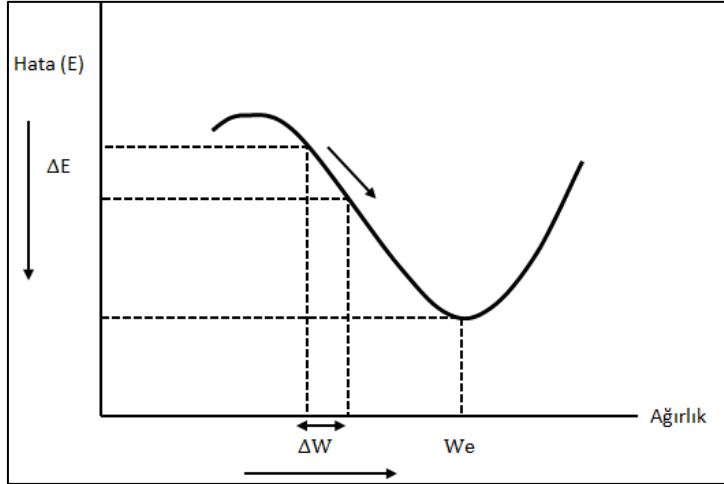
Bu işlemler sonrasında tüm ağırlıklar değiştirilmiş olacaktır. İterasyon yeni bir örnek veri girdisi ile başlatılır. İterasyon öğrenme tamamlanıncaya kadar yinelenir [27].

4.5.3. Çok katmanlı ağının çalışma prosedürü

Ağın eğitilmesi

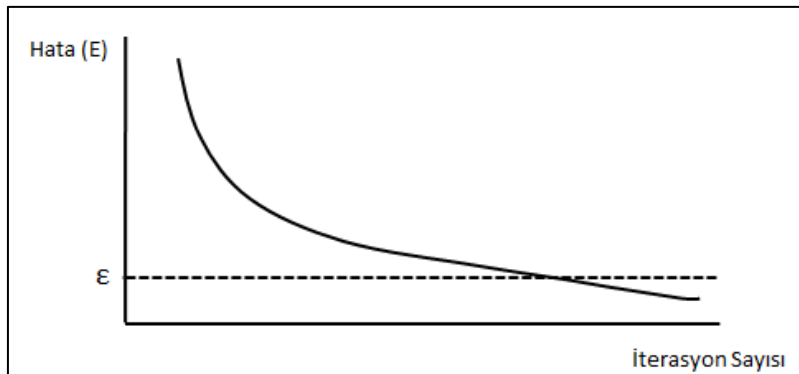
Ağın kendisine verilen girdilerle istenen çıktıların elde edilmesini sağlayacak ağırlıkların belirlenmesidir. Sürecin başlangıcında rasgele değer atanır ve iterasyonlarla istenen değere ulaşması sağlanır. İstenen hata değerine ulaşıldığında ağırlık değerlerin tespit edilmiş olur.

Şekil 4.6'da gösterilen W_e değeri istenilen en düşük hata değerini ifade eder. İterasyonla ağırlıkta ΔW kadar değişimle hatanın ΔE kadar değişmesi sağlanır. Grafikte W_e değeri ağırlıkların değişimiyle elde edildiği gözükse de gerçek uygulamalarda çoğu zaman istenilen hata değerine ulaşmak kolay değildir. Bu sebeple belir bir seviyede tolerans sağlanır.



Şekil 4.6. Öğrenmenin hata uzayındaki gösterimi [27]

Eğitimde karşılaşılan sorunlardan ilki öğrenme süresinin uzun olmasıdır. Bunun nedenlerinden biri de başlangıçta seçilen ağırlık değerinin seçimidir. Büyük değerler seçildiğinde eğitim veri setindeki değişkenlere düşmesine, küçük değerler seçilmesi de ağırlıkların doğru değerleri bulması için zaman alacaktır. Şekil 4.7’de iterasyon hata eğrisi yer almaktadır. İterasyonla beraber hatanın düştüğü görülür. Bir noktadan sonra hatadaki değişimin azaldığı görülür. Bu nokta eğitimin bittiğini ifade eder. Eğer çıkan sonuç istenen düzeyde değil ise yerel bir çözüme takılmış denir [27].



Şekil 4.7. Öğrenen bir ÇKA’nın öğrenme eğrisine örnek [27]

Ağın performansının ölçülmesi

Performans burada öğrenme sonucunun göstergesidir. Eğitim sırasında istenen değer ile elde edilen sonuç arasındaki performans olarak ta söylenebilir. Elde edilen sonuçların istenen kadar doğru sonuç vermesi performansın iyi olduğu çıkarımını sağlayamaz. Bunun önüne

geçmek için ağın eğitim verileri arasında yer almayan ve ağa hiç gösterilmemiş örnekler kullanılır. Böylece ağ görmediği örneklerle ortaya çıkardıkları sonuç arasında oran ölçülerek performans belirlenmiş olur. Performans değerini veren denklem Eş. 4.20'de verilmiştir.

$$P = \frac{D \times 100}{T} \quad (4.20)$$

D doğru cevaplanan örnek sayısı, T toplam örnek sayısı, P de performans oranıdır. Performans oranı önceden belirlenmiş bir düzeyde değilse ağ tüm örnek testleri doğru cevaplandırırsa bile iyi öğrendiği söylenemez. Bu durumda eğitime devam etmek gerekir. Bunu da iterasyon artırılarak yapılır. İterasyon artırılmasına karşın hala performans artmıyorsa örnek test verisinin temsil yeteneğinin düşük olması, ağ parametrelerinin veya topolojilerinin iyi seçilmediği anlaşılr. Daha önce belirtilen değişiklikleri yaparak ağı yeniden eğitmek gerekebilir [27].

4.6. Makine Öğrenmesi ile Sınıflandırma Yöntemleri

Makine öğrenmesini basit tanımıyla, yapay zeka uygulamaları içinde veri bilimine konsantre olup ve algoritmalar geliştiren yapay zeka birimi olarak ifade edebiliriz. Mevcut verilerden yola çıkarak olayları tahmin etme veya belirlenen alanlara sınıflandırma için kullanılır. Naive Bayes, en yakın komşu (KNN), destek vektör makineleri (SVM) ve karar ağaçları algoritmaları bu sınıflandırma algoritmaları arasında yer alır.

4.6.1. Naive Bayes sınıflandırma algoritması

Örnek veya değişkenin mevcuttaki her sınıfa ait olma olasılığını, koşullu olasılık temeline dayanarak sınıflandırma yapar. Sınıflandırma işlemini de değişkenlerin sonuca etkisini birleşmesiyle yapar [29]. Kullanım kolaylığı ve basit yapısı nedeniyle birçok alanda sınıflandırma algoritması olarak kullanılabilir. Naive Bayes algoritmasının formülü Eş. 4.21'de gösterilmektedir.

$$P(C_j|x) = \frac{p(X|C_j)p(C_j)}{p(x)} = \frac{p(X|C_j)p(C_j)}{\sum_k p(X|C_k)p(C_k)} \quad (4.21)$$

Bu eşitlikte, $P(C_j|x)$, x olan bir örneğin sınıf j 'den olma olasılığı (son olasılık), $P(x|C_j)$, sınıf j 'den bir örneğin x olma olasılığı, $p(x)$, herhangi bir örneğin x olma olasılığı ve $P(C_j)$, sınıf j 'nin ilk olasılığıdır [29].

4.6.2. En yakın komşu (KNN) sınıflandırma algoritması

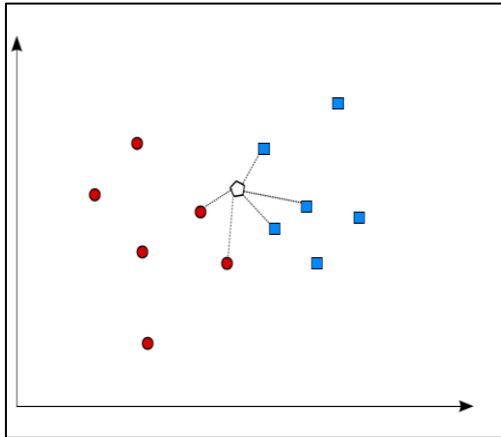
K en yakın komşu algoritması, tek bir aşamada öğrenmeyi tamamlayan denetimli öğrenme algoritmasıdır. İstenen adette komşunun belirlenmesi olarak ifade edilebilir. Yeni bilginin hangi örnek sınıfta yer alması gerektiğini ortaya koyan ve bunu da bilginin hangi sınıfa en yakın komşu olma olasılığını gösteren K adet veri seçerek yapar. K adet verinin yeni veri ile olan mesafesinde en yakın olma özelliği aranır. Mesafe hesaplamaları için Öklid, Manhattan ve Minkowski fonksiyonları kullanır. Eş. 4.22, 4.23 ve 4.24'de gösterilmektedir [30].

$$\text{Öklid} = \sqrt{\sum_{i=1}^N (x_i - y_i)^2} \quad (4.22)$$

$$\text{Manhattan} = \sum_{i=1}^N |x_i - y_i| \quad (4.23)$$

$$\text{Minkowski} = \left(\sum_{i=1}^N (|x_i - y_i|)^k \right)^{\frac{1}{k}} \quad (4.24)$$

Mesafe hesaplama için literatürde genellikle Öklid eşitliğinin kullanıldığına rastlanılmaktadır. Hesaplamaların her veri arasındaki uzaklığı ölçme gereksinimi nedeniyle çok fazla bellek alanına gereksinim duyar.

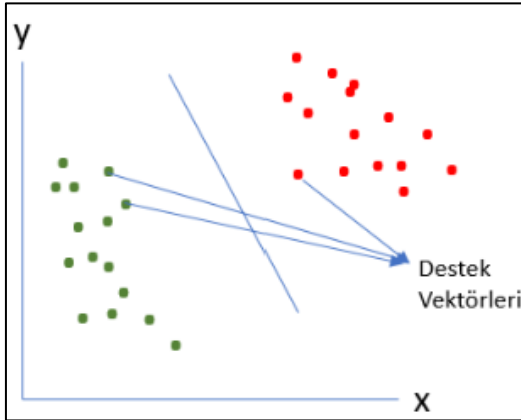


Şekil 4.8. KNN ile bir sınıflandırma [31]

Şekil 4.8’de k sayısı, en benzer örnek sayısı olarak 5 belirlenmiş bir eğitim verisinin grafik üzerinde gösterimi verilmiştir. Grafikte de görüldüğü gibi en yakın ve en fazla örnekler mavi grubunda olduğundan sınıflandırma sonucu mavi çıkmıştır. K sayısının seçimi eğitimin doğruluğu için önem kazanmaktadır. Bu sayı küçük seçilirse az miktarda tahmin bölgesi sınırları daha dar alanda bulunacağından daha hassas olacak bu kez de doğru olmayan sınıflandırma riski artacaktır. K sayısı çok büyük seçildiğinde ise en çok elemanı olan sınıf kümesine yönelim olacaktır.

4.6.3. Destek vektör makineleri (SVM) sınıflandırma algoritması

Genellikle sınıflandırma problemlerinde kullanılan ancak tahmin modelleri için de kullanılabilen denetimli öğrenme algoritmasıdır. Temelde iki sınıfı birbirinden ayırmak için kullanılır. Sınıfların ayrılmasında sınır çizgisine en yakın olan veriler destek vektörleri olarak adlandırılır. Yüksek boyutlu verilerde etkili sonuçlar üretebilmektedir [32].



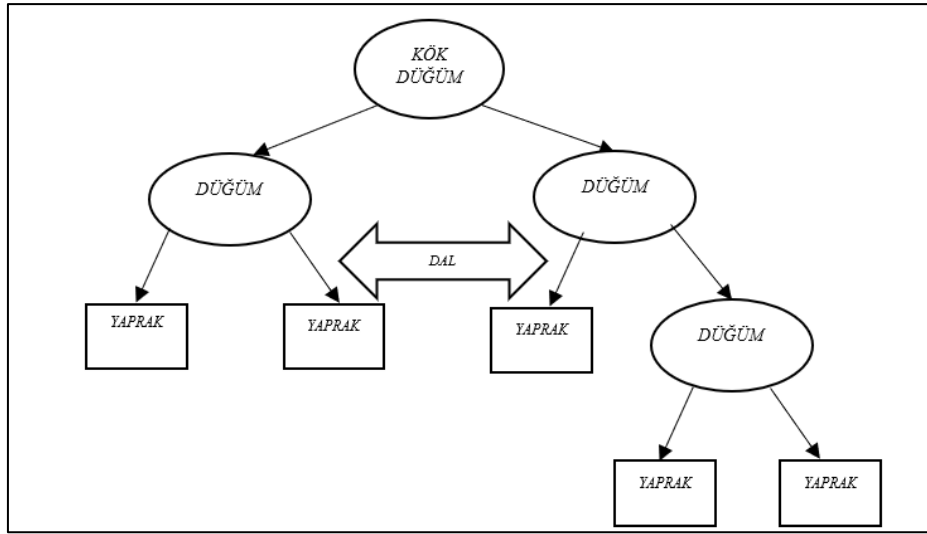
Şekil 4.9. Destek vektörleri algoritması gösterimi [32]

Sınıflandırma için örnekler hakkında tam bilgiye sahip olunmadığında kullanılabilmesi nedeniyle seçilmektedir. Doğrusal ve doğrusal olmayan verilerde kullanılabilir [33].

4.6.4. Karar ağaçları sınıflandırma algoritması

Sınıflandırma ve tahmin problemlerinde kullanılan denetimli bir öğrenme algoritmasıdır. En çok kullanılan ve uygulanması kolay algoritmalarından biridir. Ağaç şekline benzetilen yapısı vardır. Bu algorithmada değişkenlerdeki ayrı özellikler öne çıkarılarak sıralı şekilde ayrılmaktadır. Önce ağacın kurulumunu yapar sonrasında da sınıflandırma için işlemleri gerçekleştirir. İşlemi kökten uca doğru ilerleyerek gerçekleştirir. Karar ağacının en

tepesindeki kök düğümü sınıflandırmada en fazla ağırlığı olan noktadır. Düğüm, veriye uygulanacak testi ifade eder. Verinin bütününe ulaşması amacıyla dal yapısı oluşturulmalıdır. Dal, testin sonucunun alındığı bölgedir. Bu bölgede sınıflandırma ortaya çıkartılır. Sınıflandırmanın yeterli olup olmadığı karar düğümlerinden elde edilen sonuçlara göre karar verilir. Yaprak ise sınıflandırma oluşturulabiliyorsa bu sınıflardan her birine denir [34]. Şekil 4.10’da kök düğüm, düğüm, dal ve yaprak oluşumu gösterilmektedir.



Şekil 4.10. Karar ağaçları algoritmasının genel yapısı [34]

5. LİTERATÜR ÖZETLERİ

Lajunen ve arkadaşları tarafından Manchester Üniversitesi'nde yapılan çalışmada, İngiliz, Fin ve Hollandalı sürücülerin sürüş davranışları incelenmiştir. Kazalarda sürücü etkisinin %90'ın üzerinde olduğundan yola çıkarak, sürücü davranışlarını araştırmak için bir sürücü davranış anketi (DBQ) hazırlanmış ve posta yoluyla 2000 kişiden veri toplanmıştır. Ankette 6 sınıftan oluşan toplam 27 soru kullanılmıştır. Çizelge 5.1'de katılımcılar hakkında bazı bilgiler verilmiştir. Çalışmanın sonucunda bu üç batı Avrupa ülke sürücülerinin benzer ve nispeten güvenli sürüş kültürüne sahip oldukları görülmüştür [35].

Çizelge 5.1. Ölçüm yapılan parametreler [35]

	İngiliz (n=831)	Fin (n=1123)	Hollandalı (n=703)
Ortalama yaş	39,55 (14,51)	37,52 (15,12)	45,90 (14,75)
Ortalama sürüş deneyimi	18,35 (12,91)	16,93 (12,94)	23,62 (13,11)
Yıllık sürüş yapılan km.	20 894	20 435	24 637
Katılımcıların erkek oranı	51,3	45,9	70,4
Son 3 yılda trafik kazası yapıldı mı?	31,3	22,5	28,2

Yüksel tarafından sürücülerin trafik kazasına karışmasına sebep olan davranışların belirlenmesine yönelik olarak yapılan çalışmada, sürücülerin trafik kazasına karışmasında ya da karışmamasında belirleyici olan değişkenler diskriminant analizi modeli ile belirlenmiştir. Başlangıçta 32'den daha az bağımsız değişkenle trafik kazasına karışan ya da karışmayan sürücüler arasındaki farklılıklar ortaya konulmuştur. Çalışma Ankara'da okul servisi işi yapan bir şirkette çalışan servis sürücülerinden 63'üne anket uygulaması şeklinde yapılmıştır. Araştırma sonucunda, istatistiksel açıdan anlamlı bulunup, modele giren değişkenler, günlük araç kullanım süresi, kaza yapma sıklığı, en son trafik kazasını ne zaman yaptığı, kazadan sonra sürüş biçiminde değişiklik yapıp yapmadığı ve yapmış oldukları sürüş aktivitelerinden zevk alma olarak belirtmiştir. Yüksel tarafından ortaya konulan bu model ile sürücülerin trafik kazasına karışıp karışmamasındaki ayırt etme %93,7 oranında belirleyicilikle tahmin edilmiştir [36].

Kaçaroğlu ve arkadaşları tarafından yapılan çalışmada, hız ihlali sebebiyle psiko-teknik değerlendirmeye alınan ve ehliyetlerine el konulan sürücülerin hız ihlali yapmalarına sebep olan faktörler ve hız ihlalinin tekrarlanmaması için alınacak önlemler incelenmiştir. Çizelge 5.2 ve 5.3'de gösterildiği gibi sürücüler hız ihlali yapmalarının nedeni olarak, %51,4'lük oranla zaman baskısı faktörünü belirtmişlerdir. %80,3 oranla da hız kurallarına uyacaklarını

belirtmişlerdir. Sürücülerin %66'sı bu ihlalin sebebini dış etkenlere bağlamış olup, bu sürücülerin hatanın sebebini kendisinde gören sürücülere oranla aynı ihlal sebebiyle daha fazla ceza tekrarı yaptıkları belirlenmiştir [37].

Çizelge 5.2. Sürücülerin hız ihlali yapma nedenleri [37]

İhlal Nedenleri	Sayı	%
İşlerin yoğunluğu ve işyeri baskısı	204	51,4
Dikkatsizlik	115	29
Yolun boş olması	91	2,9
Aracın özellikleri	82	20,7
Kişinin aceleci olması	64	16,1
Radarların tuzak olması	59	14,9
Acil haller	57	14,4
Trafik tabelasının olmaması	50	12,6
Kasten ihlal	50	12,6
Sebebi yok	31	7,8
Sollama yaparken	26	6,5

Çizelge 5.3. Sürücülerin hız ihlali yapmamak için almayı düşündükleri önlemler [37]

Önlemler	Sayı	%
Kurallara kesinlik uyacak	317	80,3
Tekrar ihlal edilecek	47	11,9
Radarların yerini öğrenecek	44	11,1
Yola erken çıkacak	40	10,1
Cezalı olduğu süreçte kurallara uygun kullanmayı öğrendi	39	9,9
Artık araç kullanmayacak	19	4,8
Polisle anlaşacak	16	4,1
Aracına hız kontrol donanımı taktıracak	12	3
Aracını değiştirecek	10	2,5

Kaçaroğlu ve arkadaşları tarafından yapılan çalışmada, araştırmaya katılan sürücülerin yaş ortalaması 37,13'tür. Katılımcıların %29,9'u ilkokul, %33,5'i ortaokul ve lise %36,5'i de üniversite mezunu sürücülerden oluşmaktadır. Ayrıca %30,9'u 10 yıldan az, %69,1'i ise 10 yıldan daha fazla araç sürme tecrübesine sahiptir. Ehliyetlerine el konulmadan önce sürücülerin aylık ortalama 4096 km yol kat ettiklerini belirtilmiştir. Çalışmanın sonuçlarından biri olarak; ihlalin nedenlerine ve tekrar ihlal yapmamak için alınacak önlemlere ilişkin atıfların belirlenmesinde 11 neden saptanmıştır. Bu maddeler araştırmacılar tarafından incelenmiş ve ihlalin sorumluluğunun kendi hatası olduğunu belirtenler iç atıf, ihlalin sorumluluğunun dış koşullara bağlanmasına ilişkin nedenler ise dış atıf olarak gruplandırılmışlardır. Alınacak önlemler 9 madde olarak gruplandırılmıştır. Bu maddeler, dikkatli olma, yola erken çıkma, aracı değiştirme, radarların yerini öğrenme, hız kontrol

donanımı kullanma, rüşvet verme, tekrar ihlal ve bu süreçte araç kullanarak kurallara uygun kullanmayı öğrenme olarak sıralanabilir [37].

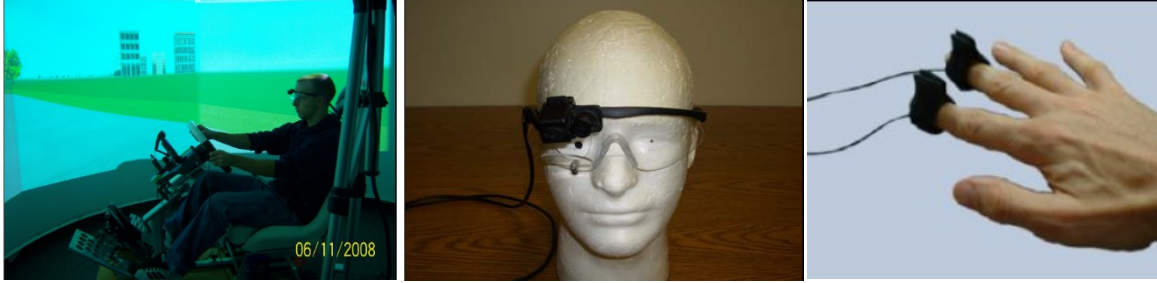
Eboli ve arkadaşları tarafından yapılan çalışmada, sürücülerin fiziksel ve duygusal şartlarının araç sürüşüne olan etkileri incelenmiştir. Bir bilgi formu yardımıyla elde ettikleri veriler ile sürüş şartları ve stillerini, dikkatli veya saldırgan olarak tanımlamışlardır. Her sürücü için 30 sorudan oluşan formlarla farklı günlerde bilgi toplanmıştır. Ayrıca 10 km uzunluğunda bir güzergâh belirlenerek yapılan sürüşlerle de sürüş bilgileri toplanmıştır. Böylece katılımcıların sürüş stilindeki muhtemel fiziksel ve duygusal değişimler ortaya konulmuştur. Çalışmaya, birinci grupta taşımacılık sektöründe deneyimli 2 kişi, ikinci grupta belirli bir deneyime sahip 2 polis memuru ve son grupta ise 8 üniversite öğrencisi katılmıştır. Katılımcıların yaşları 25 ile 50 arasında değişmektedir. Sorular sonrasında yaklaşık 400 cevaptan oluşan bir veri elde edilmiştir. Sürüşler farklılık gösterse de 10 kilometrelik güzergâh yaklaşık 15 dakika sürmüş ve sürüş sonrası 1 ile 4 arasında, 1 dikkatli, 4 saldırgan sürücü olarak belirlenmiştir. Sürücülerin fiziksel özellikleri de tembel, kararsız, sabırlı, tezcanlı ve yüzeysel olarak 5 sınıf ayrılmıştır [38].

Riyahi tarafından yapılan çalışmada sürüş esnasında sürücülerin dikkatini çekebilecek etkenlerin beyinde oluşturduğu sinyaller araştırılmıştır. Çalışmada bilgisayar simülasyonunda katılımcılara yaklaşık 10-15 dakikalık test sürüşleri hazırlanmıştır. Katılımcı olarak yaş aralığı 18-65 arasında değişen 30 kişi belirlenmiştir. Sonuç olarak beyin sinyalleri korelasyonları kullanılarak sürücülerin izlenebileceği ve sürücü yardımcı sistemleri ile uyarılabileceği ortaya konulmuştur [39].

Kondyli tarafından yapılan çalışmada, rampada ve yol birleşimlerinde yaşanan trafik sıkışıklıklarında sürücü karakteristikleri modellenerek, davranışları ölçümlenmiştir. Çevre şartları ve trafik durumu gibi hususları bir tabloda ağırlıklarını belirleyerek sınıflandırmıştır. Sürüş deneylerini yapmak için değişik yaş, sürüş deneyimi, araç kullanım sıklığı, haftalık araç sürüş saati ve sahip olduğu aracın tipi gibi bilgiler toplanmıştır [40].

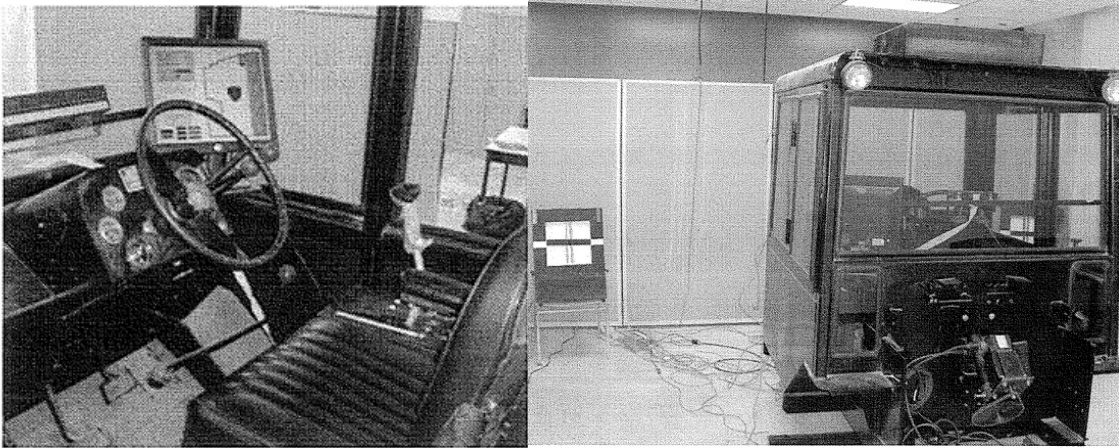
Gupta tarafından yapılan tez çalışmasında sağa ve sola dönemeçli 250 m yarıçaplı virajlarda göz hareketleri referans alınarak sürücü davranışları incelenmiştir. Çalışma aracın yönüne göre dönebilen bir koltuğa sahip bilgisayar simülatörü kullanılarak yapılmıştır (Resim 5.1). Katılımcılar 5000 milden fazla araç kullanma deneyimine sahip olup, yaşları 20 ile 25

arasında değişmektedir. 12 katılımcıdan, 16 viraj bulunan yaklaşık 5,5 millik mesafeyi yaklaşık 8 dakikada tamamlamaları istenmiştir. Tehlikeli durumlarda sürücülerin fizyolojik tepkileri ortaya konulmuştur.



Resim 5.1. Deneyde kullanılan ölçüm araçları [41]

Karimi tarafından yapılan doktora tez çalışmasında, 14 deneyimci traktör sürücüsü ile tarım traktörlerinde sürüş esnasında sürücünün görsel davranışları simülator ve saha testleri ile incelemiştir. Sürücüler öncelikle saha deneyimleri ile aynı manevra ve dönüşleri yapmış, ardından 15 dakika boyunca paralel çizgiler arasında düz sürüş gerçekleştirmişlerdir. Resim 5.2’de testlerde kullanılan simülator görülmektedir [42].



Resim 5.2. Test simülasyon platformu [42]

Testler sürüş güzergahı için yönlendirme ipuçları kullanılarak ve kullanılmadan yapılmıştır. Yönlendirme ipuçları kullanıldığında sürücülerin daha rahat bir sürüş performansı gösterdikleri, kullanılmadığı durumda sürüş esnasında kontrol aktivitelerini arttırdıkları ve sürüş performansının kötüleştiği görülmüştür [42].

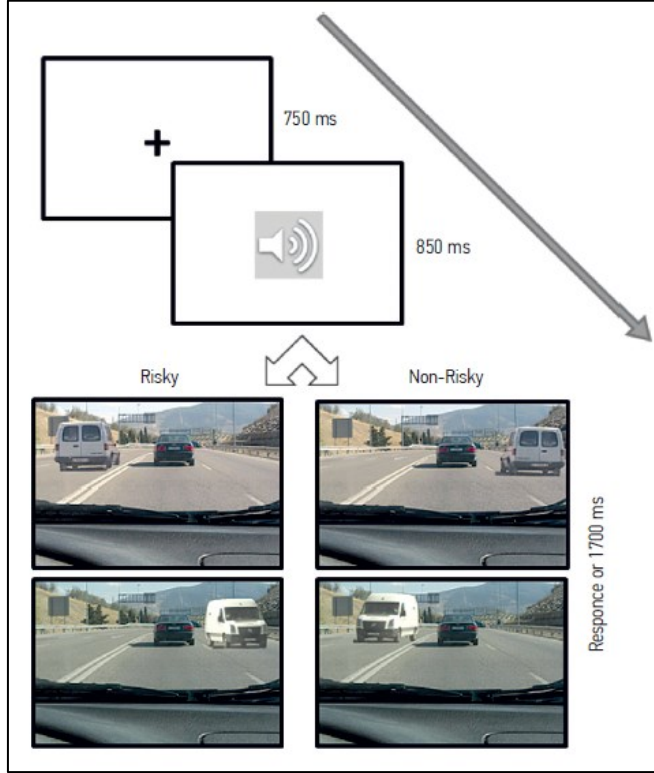
Golshani tarafından yapılan çalışmada, bir sürüş simülatörü kullanılarak 9 farklı sürüş senaryosu için 55 sürücünün katılımıyla yapılan testlerde sürücü özellikleri ile saldırgan sürüş davranışları arasındaki ilişkiler belirlenmiştir. Çalışmada ikili lojistik regresyon analizi ve yapısal eşitlik modellemesi kullanılarak sürücü özellikleri, sürücü performans ölçümleri ve saldırgan sürüş davranışları arasındaki ilişkiler incelenmiştir. Simülatör ve üzerinde gerçekleştirilen test senaryolarından biri Resim 5.3'te gösterilmektedir. Golshani tarafından yapılan ölçümler sonucunda, sürücü saldırganlıklarının derecelendirilmesi yapılmış olup, en çok yapılane en fazla, en az yapılane ise en az ağırlık puanı verilerek bir değerlendirme tablosu hazırlanmıştır. Sonuç olarak saldırgan sürüş davranışlarının ve sürücü özelliklerinin farklı sürüş olayları arasında değişiklik gösterdiğini ve araç dinamiği değişkenlerinin de saldırgan sürüş davranışlarını etkilediği ortaya konulmuştur [43].



Resim 5.3. Saldırganlık düzeyini ölçmek için kullanılan simülatör ve bir sürüş senaryosu [43]

Serrano ve arkadaşları tarafından yapılan çalışmada, his uyandıran seslerin sürücü kararlarının hızı üzerindeki etkileri araştırılmıştır. Çalışmada katılımcılara sürüş esnasında uygulanan duyguları etkileyici seslerin, sürücülerin hareketlerinde değişikliğe sebep olup olmadığı ve bunun frenleme gibi eylemlere etkisi olup olmadığı araştırılmıştır. Serrano ve arkadaşları, yaşları 18 ile 27 arasında olan 3 erkek ve 16 kadın katılımcı ile deney yapmışlardır. Bir odada ekran üzerinde resimler gösterilerek yapılan uygulamada 4 farklı

trafik koşuluna göre deney sonuçları değerlendirilmiştir. Deneylerde ekranda görüntü oluştuktan 750 ms sonra bebek gülmesi ve bebek bağırması gibi sesler 850 ms süresince uygulanmış, sürücülerin ekrandaki durumu bilgisayar faresinin sağ ve sol butonlarına tıklamasıyla ölçümü yapılmıştır. Ses efektlerinin ölçümü Şekil 5.1’de görülmektedir [44].



Şekil 5.1. Ses efektlerinin etkisinin ölçümü [44]

Elde edilen test sonuçlarına göre bir trafik sahnesinin riskli olup olmadığı his uyandıran bir ses duyulduktan sonra daha hızlı değerlendirildiği belirtilmiştir. Bununla birlikte acil frenleme gibi tepkilerin seslerin duygusal durumundan etkilenmediği de ortaya konulmuştur. Ayrıca, Serrano ve arkadaşları tarafından 2006 yılında yapılan bir çalışmaya da atıfta bulunularak sürücülerin duygusal durumunun sürüşe etkileri Çizelge 5.4’te verilmiştir [44].

Çizelge 5.4. Sürücülerin duygusal durumunun sürüş davranışına etkisi [44]

Heyecanlı	Dikkatsiz sürüş
Kızgın	Saldırgan sürüş
Üzgün	Yavaş tepkili sürüş
Karamsar	Yavaş tepkili sürüş
Telaşlı	Yavaş tepkili sürüş
Stresli	Saldırgan sürüş
Endişeli	Dikkatsiz sürüş
Hüsran uğramış	Saldırgan sürüş

Chakrabarty ve Gupta tarafından yapılan çalışmada, simülatör ve gerçek yol şartlarında olumsuz hava koşullarının sürücülerin görsel ve psiko-motor davranışları üzerindeki etkileri incelenmiştir. Yaşları 35 ile 45 arasında değişen 21 ticari araç sürücüsü ile yapılan testlerde sürücülerin hız, tepki süresi ve yağmurlu havalarda şerit takibi gibi davranışları incelenmiştir. Bu amaçla V-Box adında 3 kameradan oluşan bir sistem kullanmış ve sürücülerin göz hareketleri ölçülmüştür. Test düzeneği ve kullanılan simülatörün yapısı Resim 5.4'te gösterilmiştir [45]. Her biri 30 dakikadan oluşan toplam 3 bölüm olarak yapılan testlerde, açık hava, sisli ve yağmurlu hava koşulları simüle edilmiştir. Simülatörde yapılan testlerde deneyimli sürücülerin daha yüksek hızlarda ve trafik işaretlerini ihlal ederek araç sürdükleri görülmüştür.



Resim 5.4. Simülatör ve test düzeneği [45]

Chen ve arkadaşları tarafından yapılan çalışmada, önden çarpışma uyarı sistemlerinin etkinliğine yardımcı olmak amacıyla doğal sürüş verileri kullanılarak şerit değiştirme sırasında sürücü davranışları karakterize edilmiştir. Çalışmada 108 araç sürücüsünün 43 000 saat ve 1,2 milyon millik sürüş verisi kullanılmıştır. Geniş bir sürüş verisi örneğinden faydalanılarak sollama manevralarını belirlemek için, sollama olaylarını otomatik olarak tanımlayan bir algoritma geliştirilmiştir. Bu algoritma, yaklaşık 126 sürüşten 1 425 şerit

değiştirme olayı kullanılarak video analizi ile doğrulanmıştır. Çalışmada sürücülerin şerit değiştirme sıklığının taşıt hızına göre değiştiği ve en fazla 40-50 mil/saat sürüş hızında şerit değiştirdikleri görülmüştür [46].

Stavrinov ve arkadaşları tarafından bir sürüş simülatörü kullanılarak yapılan çalışmada, çeşitli yaş gruplarından katılımcıların reklam panolarına tepkileri araştırılmıştır. Katılımcıların reklam panolarına göz teması kurma süresi, panoların adedi, uzunluğu ve bunların etkileri değerlendirmeye alınmıştır. Katılımcıları 16-19 ve 35-55 olarak 2 yaş grubuna ayırmışlar ve simülatörde toplam 16 millik bir yol üzerinde deney yaparak, göz hareketlerini FaceLab programı ile kayda almışlardır. Çalışmalarında, 2 farklı boyutta toplam 16 reklam panosu göstererek katılımcıların reklam panolarına bakışları kaydedilmiştir. Yaş grubu reklam panosu etkileşimine bakıldığında, genç grubun panolara daha fazla tepki gösterdiği, ayrıca panolara bakış süresinin 0,75 s ile 2 s arasında değiştiği görülmüştür [47].

Shi tarafından yapılan çalışmada, aracın yolun merkezinden yanal sapmasına ve/veya yol eğriliğine göre direksiyon açısının ayarlanmasına bağlı olarak sürücülerin davranışları incelenmiştir. 3 yılı aşkın sürüş deneyimine sahip ve son 5 yılda 10 000 milden fazla araç sürmüş 21 katılımcıyla sürüş testleri yapılmıştır. Belirlenen güzergâhta ve araç üzerine yerleştirmiş kamera, radar ve gps cihazları kullanılarak veriler toplanmıştır. Çalışmada yanal sapma, yol eğriliği ve direksiyon açısı arasındaki ilişki Armax modeli kullanılarak tanımlanmıştır. Gerçek sürüş sırasında giriş ve çıkış verileri gerçek zamanlı olarak işlenerek regresyon katsayıları tahmin edilmiştir [48].

Harbluk ve arkadaşları tarafından yapılan çalışmada, şehir içi trafikte ses tabanlı ve eller serbest olarak adlandırılan teknolojilerin kullanımında bilişsel talebin sürücü davranışı ve performansı üzerindeki etkileri incelenmiştir. Minimum 5 yıllık ehliyete ve yıllık en az 10 000 km sürüş deneyimine sahip, 21-34 yaşları arasında, 9 kadın ve 12 erkek katılımcıyla 8 km'lik yol güzergâhında yapılan testlerde, göz takibi yapılarak sürücüler üzerindeki görsel değişimler incelenmiştir. Testler kolay ve zor bilişsel görevlerin yerine getirilmesi ve normal sürüş şartlarında olmak üzere 3 aşamalı olarak yapılmıştır. Performans değerlendirmesi önde bulunan nesne ile olan mesafe ve frenleme arasında bağlantı kurularak yapılmıştır. Testler sırasında araç içerisinde bulunan cep telefonu kullanılarak sürücü ile konuşulmuş ve çeşitli sorular sorulmuştur. Konuşma sırasında bilişsel görev olarak çeşitli matematik problemleri

sorulmuştur. Her ne kadar eller serbest olarak kullanılan cihazların tasarım amacı sürüş sırasında bu cihazları elle çalıştırmadan kaynaklanan dikkat dağınıklığını azaltmak ya da ortadan kaldırmak olsa da, bu cihazların kullanımı sırasında araç kontrolünün etkilendiği özellikle en zor bilişsel görevler sırasında daha fazla sert frenlemenin meydana geldiği görülmüştür. Bu cihazların kullanımlarıyla ilişkili bilişsel dikkat dağınıklığı potansiyeli de dikkate alınmalı ve uygun şekilde değerlendirilmelidir [49].

Thompson ve arkadaşları tarafından yapılan çalışmada, ağır vasıta sürücülerine yapılan ödeme yöntemlerinin sürücü yorgunluğu ve kaza riski üzerindeki etkileri aracı tabanlı modelleme ile bir taşıma sistemi simüle edilerek incelenmiştir. 2013 verilerine göre Avusturalya’da yılda 250’ye yakın ağır vasıta sürücüsünün kazalarda hayatını kaybettiğine atıfta bulunarak oluşturulan modelde 200 kamyon sürücüsü ile çalışılmış ve 50 rasgele teslimat alanı seçilmiştir. Kamyon sürücülerinin ücretlendirilmesi kilometre başına, hatta göre veya sabit fiyatlandırma şeklinde yapılmıştır. Çalışmada sürücülerin son mola verdiği saat sonrasında geçen zaman, güncel gelir, varış noktası, mevcut konum, son uyunan saat sonrası geçen süre, uyku durumu, dinlenme durumu, sürücü ehliyet durumu ve kalkış sonrası alınan mesafeler numaralandırılarak sınıflanmışlardır. Elde edilen sonuçlara göre, parça başına ücret ödeme yöntemlerinin sürücüler üzerindeki ekonomik baskıyı artırdığı ve bu durumun riskli durumlarda karar vermeyi zorlaştırdığı görülmüştür [50].

Demir ve Çavuşoğlu tarafından yapılan çalışmada, hiyerarşik eş zamanlı durum makineleri modellemesi ile farklı sürüş kategorileri için çeşitli sürüş davranışlarını simüle eden yeni bir sürücü davranış modeli geliştirilmiştir. Deneylere yaşları 19 ile 51 arasında değişen 15 deneyimli (6-31 yıl sürüş tecrübesine sahip) ve 15 acemi sürücü (0-12 ay sürüş tecrübesine sahip) katılmıştır. Testlerde TRAFIKENT sürüş simülatörü kullanılarak çeşitli sürücü hata ve ihlallerini temsil eden tehlikeli bir şehir içi trafik ortamı oluşturulmuş ve katılımcıların trafik kurallarına uyarak belirlenen rotada 20 dakika araç sürmeleri istenmiştir. Sonuç olarak toplanan veriler üzerinde yapılan ANOVA analizlerinde trafik kaza sayısı, kırmızı ışıkta geçme, şerit ihlali ve geçiş üstünlüğüne uymama gibi davranışlarda deneyimin büyük öneminin olduğu görülmüştür. Ayrıca deneyimli sürücülerin acemi sürücülere göre çok daha düşük puan aldığı ve daha fazla kırmızı ışık ihlali yaptığı görülmüştür [51].

Guéguen ve arkadaşları tarafından yapılan çalışmada, yayaların gülümsemesinin sürücü davranışları üzerindeki etkileri araştırılmıştır. Çalışmada yayaların gülümsemesinin

sürücülerin yayanın karşıdan karşıya geçmesine izin vermek için durma kararına ve sürücülerin hızlarına etkileri araştırılmıştır. Deneyler, Güney Fransa’da Atlantik sahilinde, yaz mevsiminde ve saat 2 ile 5 arasında yapılmıştır. 10 araç sürücüsü ile yapılan deneylerde yaya olarak 2 erkek ve 2 kadın görev almıştır. Dört farklı yaya geçidinde deneyler yapılmıştır. Deneylerde araştırmacılar yaya geçidinde beklerken sürücünün yüzüne bakarak gülümsemekte ve sonrasında araç durduğunda yaya geçidinden karşıya geçmektedir. Sonuç olarak, hem yaya geçitlerinde hem de dışında yayaların gülümsemesinin duran sürücü sayısını artırdığı tespit edilmiştir. Bu çalışmada ayrıca, sürücülerin yayaların gülümsemesini gördükten sonra daha yavaş sürdüğü görülmüştür [52].

Hyodo ve arkadaşları tarafından yapılan çalışmada, otoyolda uzun süreli taşıt kullanımının sürücü davranışlarına etkisi araştırılmıştır. Çalışma 16 adet iniş ve çıkış eğimli bölümü bulunan Matsuyama otoyolunda yaklaşık 36 km’lik mesafede yapılmıştır. Sürüş süresine göre sürücünün durumu sağlıklı veya normal, normal ile yorgun arasındaki geçiş evresi ve yorgunluk durumu olmak üzere üç evreye ayrılmıştır. Uzun süreli sürüş saatlerinde sürücülerin bu eğimli yolda uygun hızı koruması ve öndeki araçla olan mesafesi incelenmiştir. Sonuç olarak sürücülerin yaklaşık 5 000 saniyede bir dinlenmesi gerektiği sonucu ortaya konulmuştur [53].

Wu ve arkadaşları tarafından yapılan çalışmada sürücülerin ayak hareketleri incelenerek, pedallara yanlış basmalarına sebep olan faktörler belirlenmiştir. Çalışmada, 2004 yılında yayınlanan “Trafik Kazalarını Araştırma ve Veri Analizi Enstitüsü” raporuna atıfta bulunularak, 2004 yılında Japonya’da meydana gelen trafik kazalarının yaklaşık 7000 adedinin pedallara doğru basılmamasından kaynaklandığı belirtilmiştir. Çalışmada, Resim 5.5’de görüldüğü gibi sürücülerin sağ ayaklarını pedallara doğru nasıl hareket ettirdiklerini görmek için bir simülatör kullanılmıştır.



Resim 5.5. Doğru pedal seçimi belirleme ölçümü [54]

Deneyler 18 ile 83 yaş arasında, 20 erkek 23 kadın olmak üzere toplam 43 katılımcı ile gerçekleştirilmiş olup, video analizleri ile pedal uygulamaları direkt pedala kuvvet uygulama, kararsız hareket, doğru ancak kesik kesik uygulama ve yanlış uygulama olmak üzere dört sınıfa ayrılmıştır. Sürücülerin önlerinde görülen trafik işaretlerine uygun pedala basarak mümkün olduğunda hızlı ve doğru bir şekilde yanıt vermeleri istenmiştir. Pedal seçimi şu şekilde belirlenmiştir; eğer yeşil ışık yanarsa gaz pedalına basılmalı, sarı ışık yanarsa hiçbir işlem yapılmamalı, kırmızı ışık yanarsa fren pedalına basılmalıdır. Resim 5.6’da pedal seçimi ve simülasyon görüntüsü gösterilmektedir. Sonuç olarak katılımcıların %68,87’si kırmızı ışıktaki doğru şekilde, %28,36’sı kararsız ve %3,48’i de yanlış pedala bastığı görülmüştür. Ayrıca 21 yaş ve altı sürücülerin diğer yaş gruplarına göre daha az pedal hatası yaptıkları, daha az tereddüt ettikleri ve pedala daha doğru bastıkları görülmüştür. En yaşlı grubun tereddüt etme ve pedal hatalarının daha fazla olduğu görülmüştür [54].



Resim 5.6. Doğru pedal seçimi test simülatorü [54]

Ryder ve arkadaşları tarafından yapılan çalışmada, İsviçre’de 266 000 trafik kaza verisinden yararlanılarak kaza yapma olasılığı olan kritik bölgeler için araç içi uyarı sistemi (DSS) geliştirilmiştir. Çalışmada öncelikle kaza noktaları, geçmiş kaza verilerinden bölgesel olarak belirlenmiş ve bu kaza noktaları çeşitli kategorilere ayrılmıştır. Sonrasında, sıcak noktaların

konumları ve sınıflandırmaları ile araç içi bir DSS geliştirilerek sürücülere uyarılar şeklinde görsel destek sağlanmıştır. Geliştirilen sistem prototip bir araç kullanılarak ülke çapında 57 profesyonel sürücü ile toplam 170 000 km'lik mesafe ve 4 haftalık periyotta test edilmiştir. Sonuç olarak kaza noktalarının araç içi uyarı ile belirtilmesi zaman içinde sürücü davranışlarında gelişme sağlamıştır [55].

Pérez ve arkadaşları tarafından yapılan çalışmada, sürücü davranışlarını incelemek için Argos adı verilen araç içi veri kayıt sistemi geliştirilmiştir (Resim 5.7). Geliştirilen sistem ile taşıt, sürücü ve çevresel bazı veriler ile dokuz adet eşzamanlı video görüntüsü kaydedilebilmektedir. Argos ayrıca farklı türlerde araç içi ışık ve ses uyarıları üretip kaydedebilmektedir. Bu sayede bir deney yöneticisinin etkileşime girmesine veya bir deney sırasında gerçekleşecek belirli eylemleri planlamasına olanak tanımaktadır. Bu sistemle, serbest sürüş, araç ile öndeki aracı takip etme, şerit takibi, bağlantılı manevralar, yavaşlama ve trafik lambaları gibi konular üzerinde çalışılabilmektedir. Argos sistemi ile yapılan yol testleri sonucunda, sistemin İspanyol Trafik Kurumu'nun belirlenmiş olduğu gereklilikleri ve beklentilerini karşıladığını görülmüştür [56].

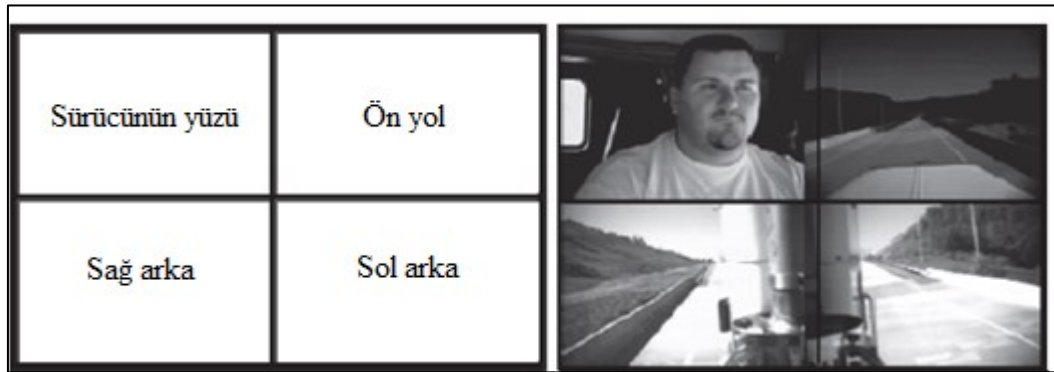


Resim 5.7. Ölçüm aracı [56]

Sato ve Akamatsu tarafından yapılan çalışmada, belirli bir kavşakta sağa dönüş yapmadan önce trafik koşullarının sürücülerin hızlanma, frenleme ve sinyal verme gibi çeşitli davranışları üzerindeki etkileri araştırılmıştır. Çalışmada gerçek yol şartlarında sürücü davranışını ölçmek ve kaydetmek için çeşitli sensörler ve sürüş kayıt sistemi ile donatılmış araçlar kullanılmıştır. Deneyler 8 haftalık sürede doğal sürüş koşullarında gerçekleştirilmiştir. Tsukuba'da yapılan deneyler sağa ve sola dönüşler içeren 15 km'lik mesafede profesyonel olmayan 22 ile 52 yaş aralığında 3 erkek ve 1 kadın sürücü ile

gerçekleştirilmiştir. Deneylerde sağ ayağın gaz pedalını bırakma ve fren pedalına basma anı, dönüşlerde sinyal vermeye başlama anı ve öndeki araçla olan mesafe ana veri olarak kaydedilmiştir. Sonuç olarak kavşağa yaklaşan aracın hızının ve sürücünün fren pedalına basma zamanlamasının ön ve arkasında bulunan araçlardan etkilendiğini göstermiştir. Ancak dönüş sinyali verme zamanlamasının önde ve arkada bulunan araçlardan etkilenmediği görülmüştür [57].

Abbas ve arkadaşları tarafından yapılan çalışmada sürüş güvenliğini riske atan olayların birkaç saniye önceden tespit edilebilmesi amacıyla bir metot geliştirilmiştir. Bu amaçla Virjinya Teknoloji Taşımacılık Enstitüsü tarafından yürütülen “100-car Naturalistic Driving” çalışmasından yararlanmışlardır. 100-car programıyla yaklaşık 43000 saat ve 2 000 000 mil sürüş ile video, ses ve dinamik performans bilgileri toplanmıştır. Yapılan testlerde Resim 5.8’de görüldüğü gibi 4 kamera kullanarak, sürücü aktiviteleri gözlenmiştir. Sonrasında elde edilen bilgiler anket verileri ile desteklenmiştir. Sonuç olarak sürüş esnasında kritik düzeyin veya diğer bir ifadeyle sürüşü riske atan olayların tanımlanabilmesi için 6 saniye gibi bir süreye ihtiyaç olduğu ortaya konulmuştur. Bu sonuç, doğru boyunca hızlanma, yanal hızlanma, taşıt hızı, sürüş doğrultusu, yanal hareket açısı, mesafe ve mesafe oranı gibi değişkenler kullanılarak elde edilmiştir [58].



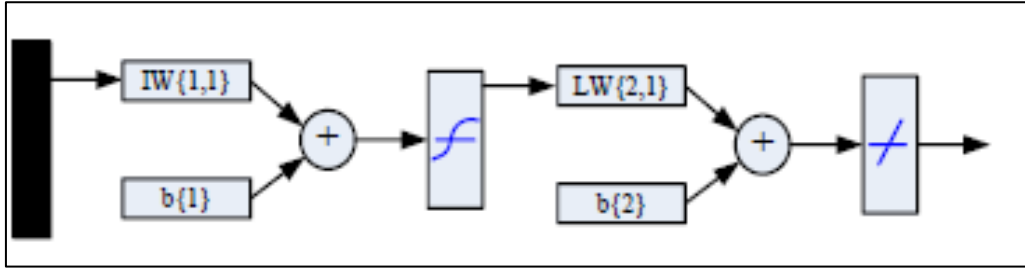
Resim 5.8. Sürücü aktivitelerinin 4 kamera görüntüsü ile ölçümü [58]

Li ve Meng tarafından yapılan çalışmada birden fazla değişken kullanılarak sürücü yorgunluğunu algılamak için bir yöntem geliştirilmiştir. Analizlerde, sürücünün yüz özellikleri, sürüş rotası ve direksiyon kullanım şekli değişken olarak belirlenmiştir. Bir sürüş simülatörü kullanılarak bu üç değişkene ait veriler toplanmış ve öznel değerlendirmeye dayalı olarak sürücünün uyanıklık ve yorgunluk düzeyi belirlenmiştir. 4 kadın ve 4 erkekten oluşan katılımcı ile yapılan çalışmada sürücülerin testlere başlamadan önce en az 8 saat

uyumaları sağlanmıştır. Belirli saat periyodlarında sürüş yaparak, yorgunluk düzeylerini belirten formları doldurmaları sağlanmıştır. Sonrasında elde edilen değerler yapay sinir ağına girdi olarak kullanılmış ve geri yayımlı algoritma kullanılarak eğitilmiştir [59].

Şekil 5.2’de gösterildiği gibi yapay sinir ağını 5 giriş, 11 gizli ve 2 çıkış olacak biçimde yapılandırmışlardır. Çıkışlarında [1 0] dinç, [1 0] yorgunu temsil etmektedir. Ağın eğitiminde hazırladıkları 500 örnekten 50 tanesini kullanmışlardır. Çalışmanın sonucunda yorgun sürücüyü tespit etmek için direksiyon açısından aldıkları verilerin daha anlamlı olacağı sonucuna varmışlardır. Yorgun insanların göz kırpmaları azalırken, gözlerinin kapalı kaldığı süre uzamaktadır. Bunun tespiti sürücünün yüzüne odaklanan bir kamera ile yapılabilir. Bununla birlikte normal sürücüler direksiyonu daha yumuşak hareketlerle kullanırken, yorgun kullanıcılar daha keskin direksiyonlama yapmaktadırlar. Bu hareketlerin tespiti de direksiyon açısının ölçülmesi ile elde edilebilir [59].

Sonuç olarak yorgunluk tespiti için kullanılan yöntemlerden biri olan göz kapaklarının kapalı kalma süresi ve kaç kez açılıp kapandığının yorgunluk tespitinde yeterli olamayacağı, trafikte araç kullanım şekli ve trafiğin durumu ile bilgilerin de dikkate alınması gerektiği ortaya konulmuştur [59].



Şekil 5.2. Yorgunluk tespiti için kullanılan YSA ağ yapısı [59]

Ulusal Otoyol Trafik Güvenliği Kurumu’nun 2013 raporunda, 2011 yılında Amerika Birleşik Devletleri’nde ölümlü trafik kazalarının %10’unun, yaralanmalı kazaların %17’sinin sürücülerin dikkat dağınıklığı nedeniyle meydana geldiği belirtilmiştir. Ayrıca, Avustralya Ulusal Kaza çalışmasında sürücülerin dikkat kaybının kazaların oluşumunda %70 oranında etkili olduğu belirtmiştir. Bu amaçla, Liberty tarafından bir sürüş simülatörü kullanılarak yapılan çalışmada sürücünün bilerek veya istem dışı olarak dikkatinin dağılmasının sürüş performansına etkileri incelenmiştir. 25-39 yaş aralığında 19 erkek ve 19 kadın olmak üzere 38 katılımcı ile yapılan testler “NADS quarter-cab MiniSim” marka sürüş

simulatoründe gerçekleştirilmiştir. Dikkat dağıtıcı olarak ekranda kelime eşleştirme görevi ve bir takım görsel animasyonlar ile aniden ortaya çıkan zil sesi gibi etkenler kullanılmıştır. Sonuç olarak bilerek veya istem dışı dikkat dağınıklığının sürüş performansını farklı etkilediği görülmüştür. Sürücülerin bilerek ikincil bir görevle meşgul olduklarında dikkatlerinin dağıldığının farkında oldukları ve sürüş esnasında bazı olumsuzlukları telefı edebildikleri görülmüştür [60].

Yan ve arkadaşları tarafından yapılan çalışmada sürücülerin sürüş esnasında telefon etme, yemek yeme, sigara içme, frenleme gibi davranışlarını tanımlamak için bir model geliştirilmiştir. Deneyler Southeast Üniversitesi'nde sürücü pozisyonu veri seti kullanılarak 20 katılımcı ile gerçekleştirilmiştir. Sürücü davranışları, telefon görüşmesi, yemek yeme, frenleme, direksiyonu doğru pozisyonda tutma, telefonla oyun oynama ve sigara içme olmak üzere 6 sınıfa ayrılmıştır (Resim 5.9). Çalışmada Gauss karışım modeli (GMM) kullanılarak derin öğrenme evrişimsel sinir ağları (R*CNN) oluşturulmuştur. Ten rengine göre sürücü davranışları tespit edilmiştir. Bu tespit için GMM yönteminden yararlanılmıştır. Sağlıklı bir tespit ile eğitim ve test işlemleri için görüntüler RGB yerine YCbCr renk uzayına dönüştürülmüştür. Kullandıkları Gauss modelinin pratikte çıkabilecek birçok problemin önüne geçtiğini ve bu sayede ten rengine göre görüntü tespitinde başarılı olduklarını ifade etmişlerdir [61].



Resim 5.9. Sürücü pozisyonları [61]

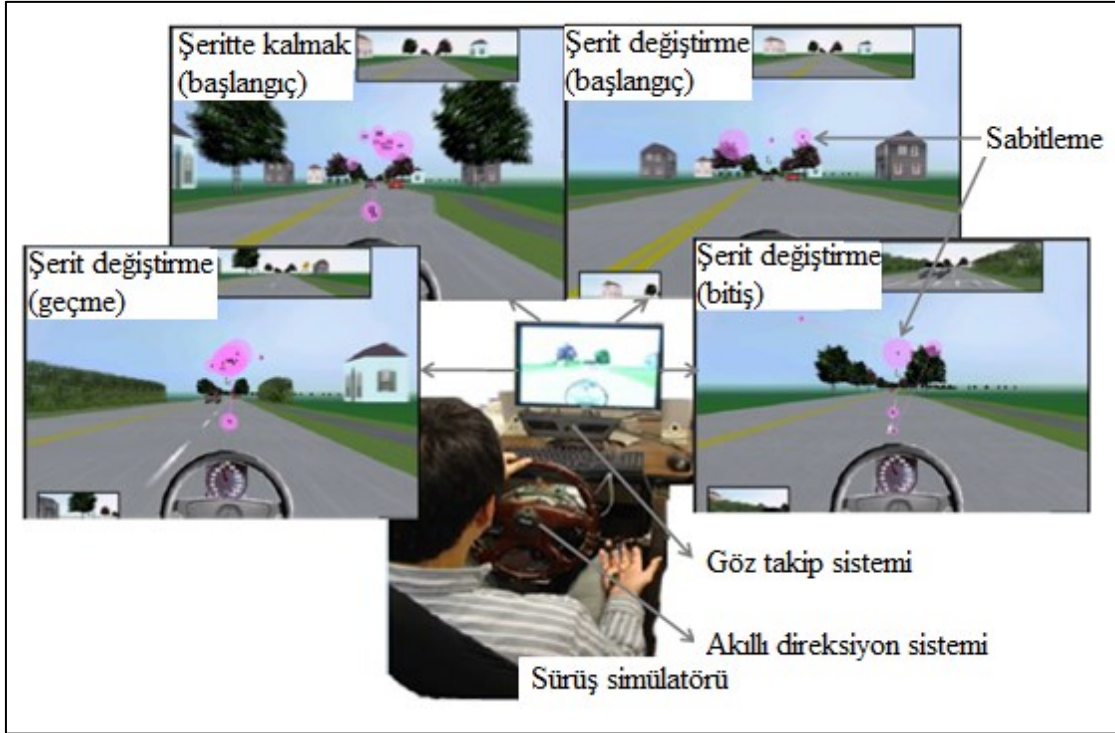
Doğan tarafından yapılan çalışmada görüntü işleme metotları kullanılarak kuş bakışı dönüşümü temelli bir araç çevresi izleme sistemi geliştirilmiştir. Bu model geliştirilirken

OpenCV açık kaynak kodlu bilgisayarlı görü kütüphanesinden yararlanılmıştır. Çalışmada, kameralardan alınan görüntüler kamera kalibrasyonu ile düzeltilmiş ve bu görüntüler girdi olarak kullanılmıştır. Kullanılan kalibre görüntülere kuş bakışı görüntü dönüşüm işlemi uygulanarak araç çevresini görüş konusunda olumlu sonuçlar elde edilmiştir [62].

Satzoda ve Trivedi tarafından yapılan çalışmada, etrafındaki ortamı algılayan sensörlerle donatılmış bir araç (ego araç) referans alınarak, bu aracı sollayan ve bu araçtan uzaklaşan araçları tespit etmek için görüntü temelli yeni bir yöntem geliştirilmiştir. Önerilen yöntemde Adaboost sınıflandırıcı kullanılmış olup Haar benzeri özelliklere dayanmaktadır. Çalışmada 27 000 video karesinde meydana gelen 51 adet sollama ve uzaklaşma eylemi %100 doğrulukla tespit edilmiştir [63].

Martin tarafından yapılan çalışmada, soldan ve sağdan şerit değiştirme ile aracın şeritte tutulması davranışları sürücünün göz hareketlerinden faydalanarak modellenmiştir. Testler farklı sürüş deneyimlerine sahip katılımcılar ile farklı güzergâhlarda ortalama 40 dakikalık sürüşler şeklinde gerçekleştirilmiştir. Sürücü göz hareketlerini almak için dikiz aynasına ve A sütununa yerleştirilen iki kameradan faydalanılmıştır. Çalışmada sürücülerin sol ve sağ şerit değiştirme manevraları yaklaşık 1 000 ms önceden %75'in üzerinde doğrulukla tespit edilmiştir [64].

Huang tarafından yapılan çalışmada makine öğrenme yöntemlerinden destek vektör makineleri metodu kullanılarak sürücülerin göz hareketleri ile şerit değiştirme niyetleri tahmin edilmiştir. Bir sürüş simülatörü kullanılarak yapılan deneylerde, 6 erkek ve 6 kadın olmak üzere 12 lisansüstü öğrenci katılımcı yer almıştır. Deneylerde katılımcılar kendilerine verilen öndeki aracı geçme ve şerit değiştirme senaryosunu gerçekleştirmiştir. Resim 5.10'da görüldüğü gibi, bu sırada akıllı direksiyon sistemi ile sürüş performans verileri ve sürücünün göz hareketleri kaydedilmiştir. Destek vektör makineleri metodu kullanılarak sürüş manevraları aracın şeritte tutulması ve şerit değiştirme olmak üzere iki sınıfta tanımlanmıştır. Sonuç olarak sürücülerin şerit değiştirme davranışları 0,6 saniye önceden %88,78 doğrulukla tahmin edilmiştir [65].



Resim 5.10. Simülasyon test düzeneği [65]

Salemi tarafından yapılan çalışmada, araçların çalınmasını engellemek amacıyla, sürücü davranışına dayalı bir kimlik doğrulama sistemi geliştirilmiştir. Çalışmada sürücünün araç kullanım karakteristiğinin algılanması ve farklı sürücü karakteristiklerinin tespit edilmesini sağlayacak bir model ortaya konulmuştur. Çalışmada 18 erkek ve 12 kadından oluşan 30 katılımcı, birbirinden tamamen farklı A ve B rotalarında yaklaşık 9,8 mil araç kullanmıştır. Belirlenen güzergâhta yapılan testler, yoğun trafikte sürekli frenleme ve hızlanma, şerit değiştirme ve dönüş manevraları gibi sürüş koşullarını içermektedir. Sürüş verileri, bluetooth özellikli OBDII adaptörü, akıllı telefon ve bir android uygulaması kullanılarak kaydedilmiştir. Sürücülerin kimlik doğrulaması için, araç hızı, motor devri ve jiroskop verileri kullanılmıştır. Çalışmada sürücülerin kimlik doğrulaması %4,44'lük bir eşit hata oranı ile başarılı bir şekilde gerçekleştirilmiştir [66].

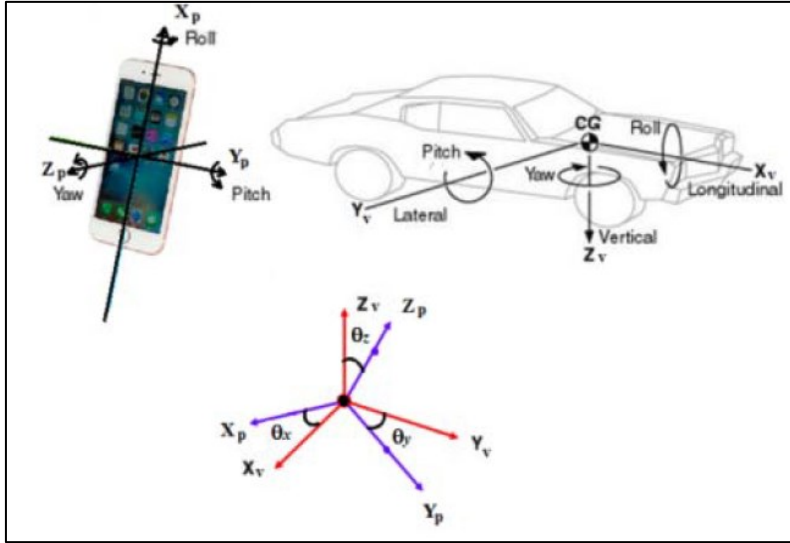
Yüksel ve Atmaca tarafından yapılan çalışmada makine öğrenmesi algoritmaları ile sürücü davranışları sınıflandırılmıştır. Sürücü davranışları, ani hızlanma, ani sağa veya ani sola dönüş hareketleri elde edilerek sınıflandırılmıştır. Elde ettikleri veriyi rasgele orman, Bayes ağları, yapay sinir ağları ve destek vektör makinelerini kullanarak bir model oluşturmuşlardır. Verilerin elde edilmesi için Raspberry pi aracılığı ile jiroskop ve ivmeölçer ölçümleri yapılmıştır. Çalışmada kullanılan test düzeneği Resim 5.11'de görülmektedir.

Çalışmalarının sonucunda sınıflandırma için en yüksek doğruluk performansı Bayes ağları ve destek vektör makineleri [67].



Resim 5.11. Anlık verilerin toplanması [67]

Al-Din ve Al-Mashakbeh tarafından yapılan çalışmada akıllı telefon tabanlı otoyol sürüş tanımlama ve sınıflandırma sistemi geliştirilmiştir. Sadece akıllı telefon kullanılarak yapılan çalışmada, öncelikle sürüş tanımlama ve sınıflandırma sisteminin kalitesini arttırmak için istatistiksel filtreme teknikleri kullanılmış ve telefon sensörlerinin kalibrasyonu için yeni bir prosedür geliştirilmiştir. Çalışmanın ikinci kısmında öğrenme sürecini basitleştirdiği, doğruluk ve güvenilirliğini arttırdığı için sürüş olayı tanımlama ve sınıflandırma şeklinde ikiye ayrılarak incelenmiştir. Şekil 5.3'te akıllı telefon ile araç konumunun koordinatları görülmektedir. Akıllı telefonda bulunan ivmelenme, jireskop ve manyetik sensör vasıtasıyla elde edilen veriler makine öğrenimi metotları kullanılarak bir algoritma geliştirilmiştir. Doğru bir sınıflandırma ve tanımlama modeli geliştirebilmek için, testler 10 katılımcı ile 15 kilometrelik şehir dışı yolda gerçekleştirilmiştir. Bu testlerde hızlanma, frenleme, sol şerit değişikliği, sağ şerit değişikliği, otoyola girme ve çıkma eylemlerini her sürücünden 5 kez yapması istenerek sürüş verileri toplanmıştır. Elde edilen veriler yapay sinir ağları kullanılarak sürüşleri sert, normal ve yumuşak olarak sınıflandıracak bir model ortaya konulmuştur. Sonuç olarak %100'e yakın bir tanımlama oranı ve %95'e yakın bir sınıflandırma oranı elde edilmiştir [68].



Şekil 5.3. Akıllı telefon ve araç eksenleri [68]

Olabiyyi ve arkadaşları tarafından yapılan çalışmada, sürücü davranışlarını tahmin etmek için gerçek zamanlı veri toplama, işleme ve öğrenme çerçevesi içeren bir sistem ortaya konulmuştur. Sürücü davranışlarının tahmini zaman serileri ile anomali tespit problemi şeklinde formüle edilmiştir. Çalışmada farklı sürüş koşulları altında güney San Fransisco körfezi bölgesinde 5 sürücünden yaklaşık 35 saatlik sürüş verisi toplanmıştır. Veri toplamak için 3 kamera, GPS ve CAN Bus veri kaydedici kullanılmıştır. Sürücü hareketini tahmin etmek amacıyla, girdi olarak taşıtın fren ve gaz pedal basıncı, vites konumu, direksiyon açısı, hızlanma, ivmelenme ve motor hızı ile sürücünün yüz, kafa ve kol hareketi bilgisini alınarak frenleme, yön değiştirme ve dönüş eylemleri için tahminleme modeli oluşturulmuştur. Sonuç olarak önerilen sistemin sürücü hareketlerini tahmin etmede diğer mevcut sistemlerden daha iyi performans gösterdiği ve sürücü tarafından eylem gerçekleştirilmeden 5 saniye öncesinde hızlanma, frenleme, şerit değiştirme ve dönüş gibi temel sürücü davranışlarını doğru bir şekilde tahmin edebildiği ortaya konulmuştur [69].

Serttaş ve arkadaşları tarafından yapılan çalışmada, bir akıllı telefonda bulunan sensörler kullanılarak gerçek zamanlı sürüş verileri ile sürücü davranışları incelenmiştir. 2 katılımcı ile yapılan testlerde, akıllı telefonda bulunan jiroskop, ivmeölçer, yakınlık ve ortam ışığı gibi sensörler kullanılarak sürücülerin hızlanma, şerit değiştirme ve ani yavaşlama gibi sürüş karakteristikleri tanımlanmıştır. İki farklı senaryo şeklinde yapılan testlerde öncelikle aynı yol, güzergâh ve saatte aynı araçla iki sürücü belirlenen parkuru tamamlamıştır. İkinci senaryoda ise, aynı güzergâh, araç ve sürücü ile farklı saatlerde yapılan sürüşte trafik

yoğunluğunun sürücü davranışlarına etkileri incelenmiştir. Elde edilen verilerle sürücü davranışları istatistiksel özellikler kullanılarak tanımlanmıştır. Çalışmanın sonucunda aynı yol ve trafik şartlarında sürüş karakteristiklerinin değişkenlik gösterebileceği ortaya konulmuştur [70].

Peng ve arkadaşları tarafından yapılan çalışmada şerit değiştirme yardım sisteminde kullanılmak üzere gerçek yol deneyleri ile sürücünün şerit değiştirme davranışını tahmin etmek için bir model geliştirilmiştir. Deneyler 9 erkek ve 7 kadın olmak üzere 16 katılımcı ile iki yönlü ve dört şeritli bir otoyolda gerçekleştirilmiştir. Deneylerin yapıldığı otoyolda hız sınırı 110 km/h olup, deneyler yaklaşık 25 kilometrelik bir mesafede yapılmıştır. Resim 5.12’te deney düzeneği görülmektedir [71].

Yapılan sürüşlerde 406 şerit değişimi örnek alınarak bir veri tablosu oluşturulmuştur. Çalışmada sinyal verme, araç hızı, öndeki ve arkadaki araçlar ile olan mesafe ve sürücülerin baş konum açısı girdi verisi olarak kullanılmış ve geri yayımlı yapay sinir ağı modeli oluşturulmuştur. Oluşturulan model kullanılarak sürücülerin şerit değiştirme davranışı %85,54 doğrulukla 1,5 saniye önceden tahmin edilebilmiştir. Ayrıca çalışmanın sonuçları, sürücülerin şerit değiştirme davranışlarını tahmin etmek için geliştirilen modelin, şerit değiştirme yardım sisteminin performansını önemli ölçüde iyileştirebileceği ortaya konulmuştur [71].

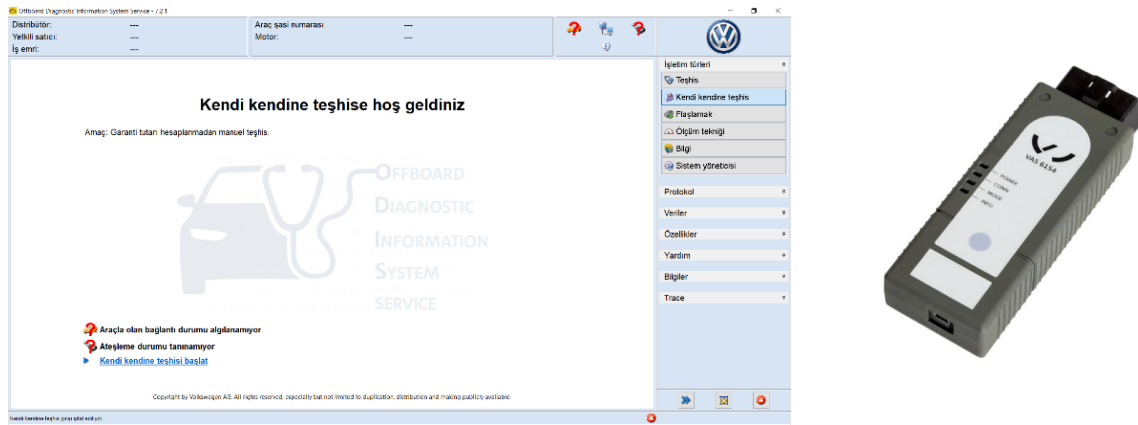


Resim 5.12. Veri toplamak için kullanılan cihazlar [71]

6. MATERİYAL VE METOT

6.1. Direksiyon Açısı ve Taşıt Hızının Elde Edilmesi

Bu çalışmada, sürücü davranışları ile şerit değiştirme arasında ilişki kurularak, şerit değiştirme eyleminin gerçekleşmeden önce tahmin edilmesi amaçlanmaktadır. Bu amaçla, sürücünün yüz ve göz hareketleri ile taşıtın önündeki araçların belirlenmesi için nesne tespiti ve takibi yönteminden faydalanılmıştır. Test sürüşleri gerçek yol şartlarında yapılmış ve sürücü davranışlarının belirlenmesi için kamera kayıtlarından yararlanılmıştır. Şekil 6.1’de görülen ODIS Vas5054 test cihazı ile CAN Bus hattı üzerinden araç ile ilgili belirli bilgilere erişilebilmektedir. Direksiyon açısı bilgisi, sinyal kolunun konumu, taşıt hızı, motor devri, fren butonu ve gaz pedalı potansiyometre bilgisi bu bilgilerden bazılarıdır.



Şekil 6.1. ODIS (Offboard Diagnostic Information System) test cihazı

Şekil 6.2’de örnek bilgilerin ekran görüntüsü yer almaktadır. Vas5054 cihazı ile anlık değerlere ulaşılabilen ve veriler deney sonrası yorumlama için kaydedebilmektedir. Çalışmada ihtiyaç duyulan direksiyon açısı bilgisi ve araç hızı bu test cihazından yazılım ile kayıt altına alınmıştır. Excel programında açılabilir şekilde csv formatında veriler elde edilmiştir.

Vas5054 test cihazı aşağıdaki özelliklere sahiptir.

- Oki chipset
- K line (ISO9141-2) protokolü
- High-speed CAN (ISO 11898-2) protokolü

- Low-speed CAN (ISO 11898-3) protokolü
- J1850 (SAE J1850) protokolü
- UDS Protokolü
- PWM Protokolü

003 Ölçüm değerleri		23.10.2021 12:14:52
Name	Wert	Adr.
Araç hızı [\$2B16]	82,83 km/s	0044
Motor devir sayısı [\$F40C]		0044
[LO]_measurement_value	1.565 1/min	
Direksiyon açısı sensörü açısı [\$1812]		0044
Direks.açısı	-2,8 °	
Durum	geçerli	
Direksiyon açısı sensörü açısı hızı [\$1815]		0044
Direksiyon açısı hızı	-1,0 °	
Durum	geçerli	

Şekil 6.2. ODIS cihazından alınan sensör verisi

6.2. Kameralardan Görüntü Verilerinin Alınması

Bu çalışmada sürücünün öndeki aracı geçme kararı ve şerit değiştirme davranışının tahmin edilmesi üzerine bir model önerilmiştir. Bu model yapay sinir ağlarında, çok katmanlı ağlar modeli ile oluşturulmuştur. Modelde kullanılacak veri bilgisi de araca yerleştirilmiş 2 kamera ve araç CAN Bus üzerinden Vas5054 test cihazı ile alınmıştır. Bu amaçla aracın ön camına dış ve iç görüntü alacak şekilde 2 kamera yerleştirilmiştir. Kameraların yerleşim düzeni Resim 6.1’de, kullanılan Havana HVNA-8081 marka kameranın görüntüsü Resim 6.2’de verilmiştir. Testlerde kullanılan Havana HVNA-8081 marka kameranın özellikleri aşağıda verilmiştir.

- 12 Megapiksel CMOS sensörü
- 10M/ 5M / 3M /2M /1M görüntü çözünürlüğü
- 1082P, 720P, WVGA video çözünürlüğü
- Avi video dosya formatı
- JPEG görüntü dosya formatı
- 140° açılı lens



Resim 6.1. Sürücünün yüz hareketleri ve öndeki aracın tespiti için kullanılan kameraların montaj görüntüsü



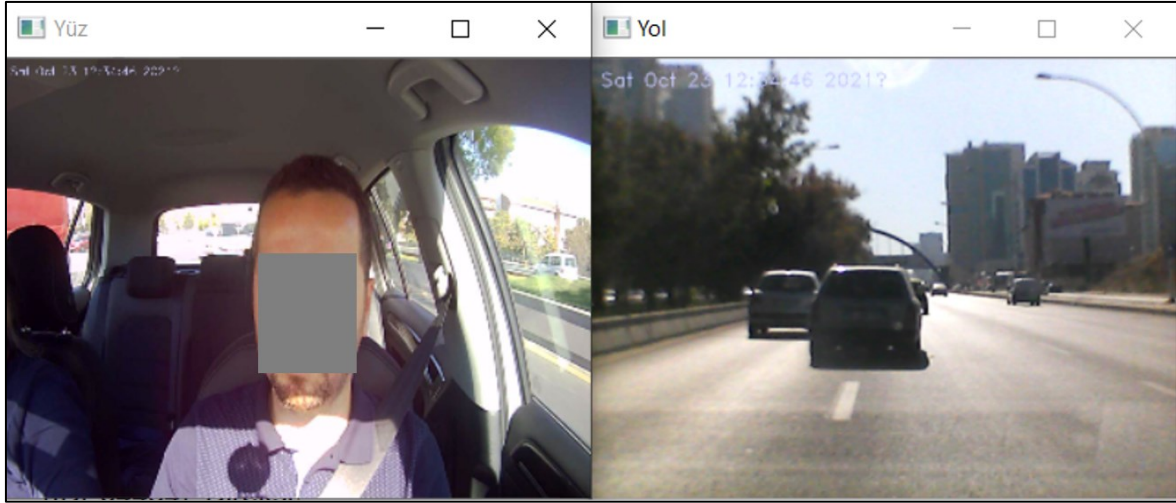
Resim 6.2. Havana HVNA-8081 marka kamera

Taşıt bilgilerine ulaşmak için kullanılan ODIS Vas5054 cihazının bağlantısı taşıtta bulunan ODB2 soketi ile yapılmıştır. Soketin bağlantı görüntüsü Resim 6.3'te verilmiştir.



Resim 6.3. ODIS bağlantı modülünün araca bağlanması

Her iki kameradan alınan görüntülerin eşzamanlı olması için Visual Studio ortamında C++ ile bir program hazırlanmış ve bu program ile video kaydı alınmıştır. Resim 6.4'te kameralardan alınan örnek görüntü yer almaktadır. Hazırlanan bu program ile sürücü yüz ve göz hareketleri ile öndeki araç görüntüsü arasında zaman farkının önüne geçilmiştir. Alınan görüntüler 640 x 480 boyutlarında ve saniyede 30 çerçeve şeklinde elde edilmiştir.

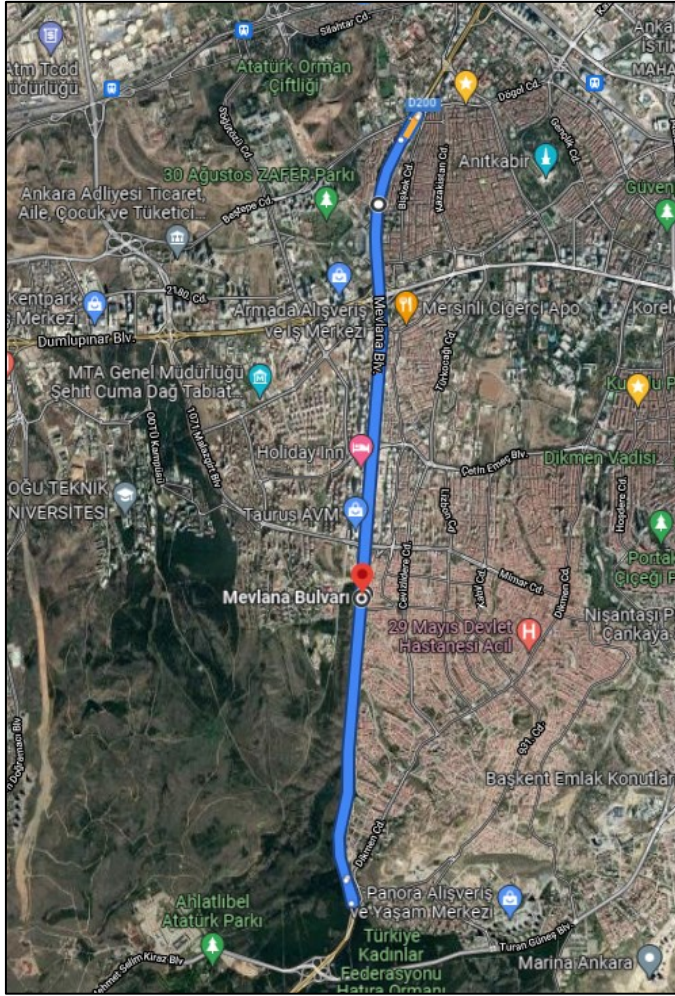


Resim 6.4. Ön ve sürücü kamera görüntüsü

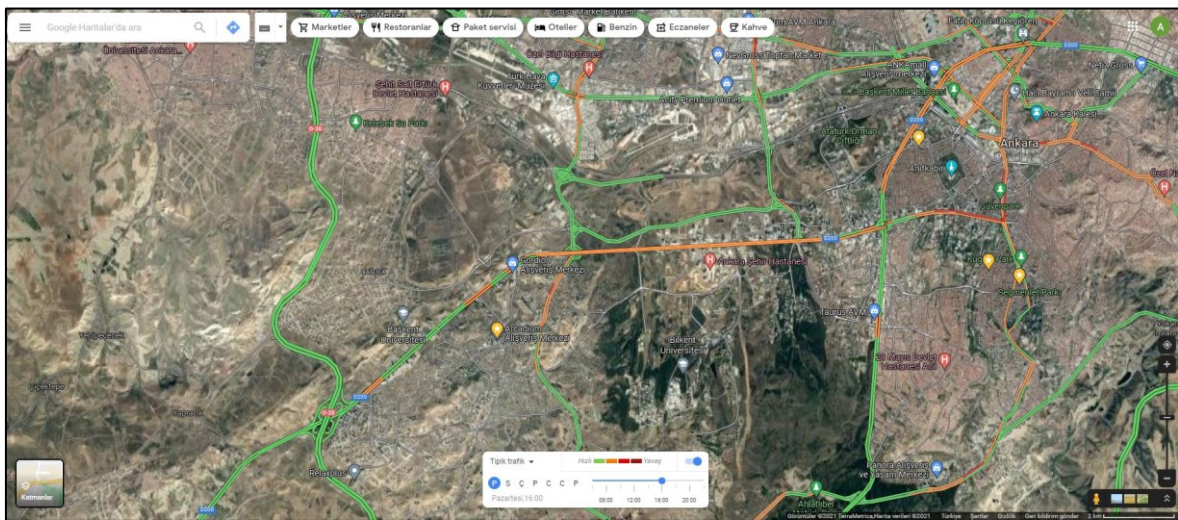
6.3. Test Güzergahı ve Sürücülerin Seçimi

Sürücü davranışlarını ölçümlemek için yapılan yol testleri 8 katılımcı ile gerçekleştirilmiştir. Test güzergahı için, şehir içi trafiğin çok sıkışık olmadığı ve araç ile manevra yapılabilecek zaman aralığı içerisinde yer alan bir bölge seçilmiştir. Şekil 6.3'te testlerin yapıldığı güzergah görüntüsü bulunmaktadır.

Güzergâh seçiminde Google maps internet sitesinden yararlanılmıştır. Şekil 6.4'te gösterilen Google maps görüntüsünden anlık ve günlük ortalama trafik durumuna göre trafik yoğunluğu okunabilmektedir. Sürüşün yapılacağı gün olarak cumartesi saat 12.00 ile 16.00 arası seçilmiştir. Test sürüşleri bu zaman aralığında yapılmıştır.

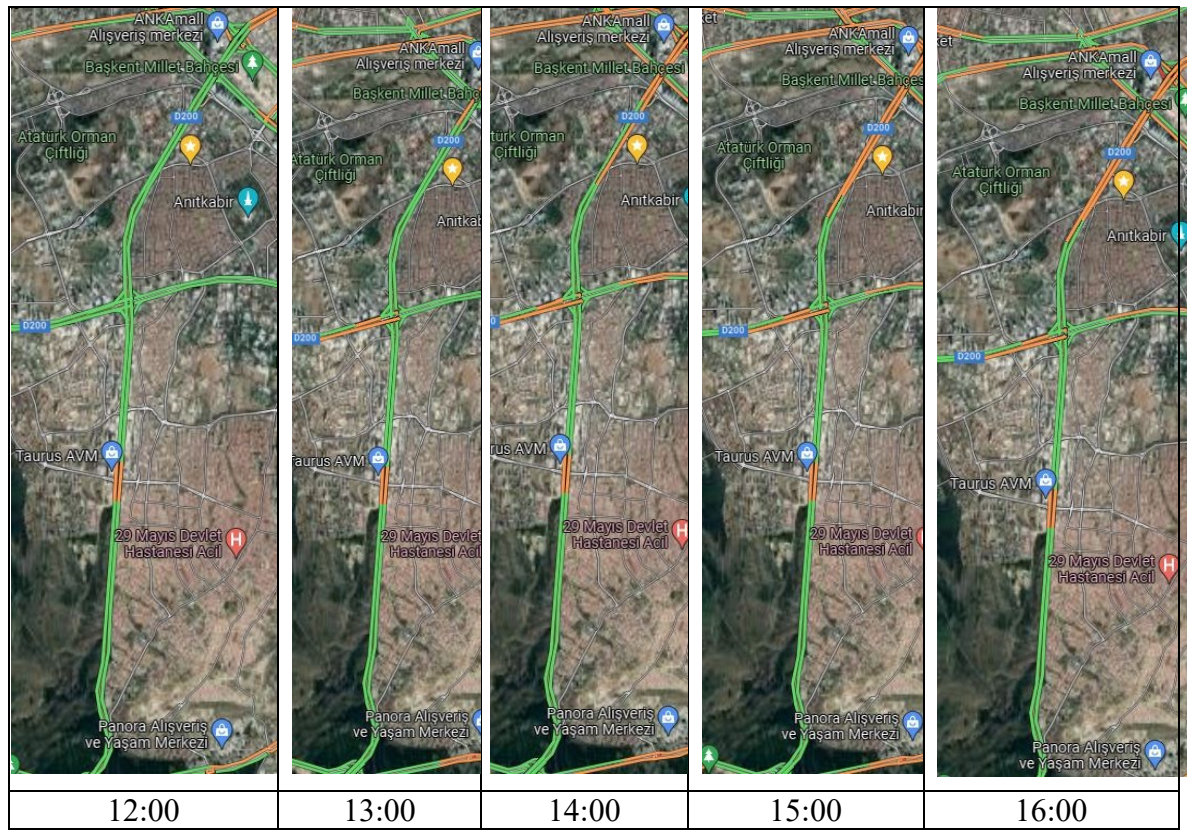


Şekil 6.3. Test güzergahı



Şekil 6.4. Google maps trafik görüntüsü [72]

Şekil 6.3'te gösterilen güzergâh 17,6 km uzunluktan oluşmaktadır. Testlerde 2020 model otomatik vitesli dizel bir araç kullanılmıştır. Şekil 6.5'te test sürüş gününde saatlere göre güzergâhtaki trafik yoğunluk durumu gösterilmektedir. Testler bütün katılımcılarla aynı trafik yoğunluğunda gerçekleştirilmiştir. Böylece trafikte dur-kalkların çok olmadığı sürüş şartlarında sağlıklı bir şekilde öndeki aracı geçme verileri elde edilebilmiştir. Katılımcılardan trafik kurallarına uyarak belirlenen güzergahta normal şartlarda araç kullanmaları ve normal sürüş esnasında uygun koşullar oluştuğunda öndeki aracı geçmeleri istenmiştir. Bu şekilde, öndeki aracı geçtikleri her zaman aralığı sollama yapılan zaman olarak belirlenmiştir.



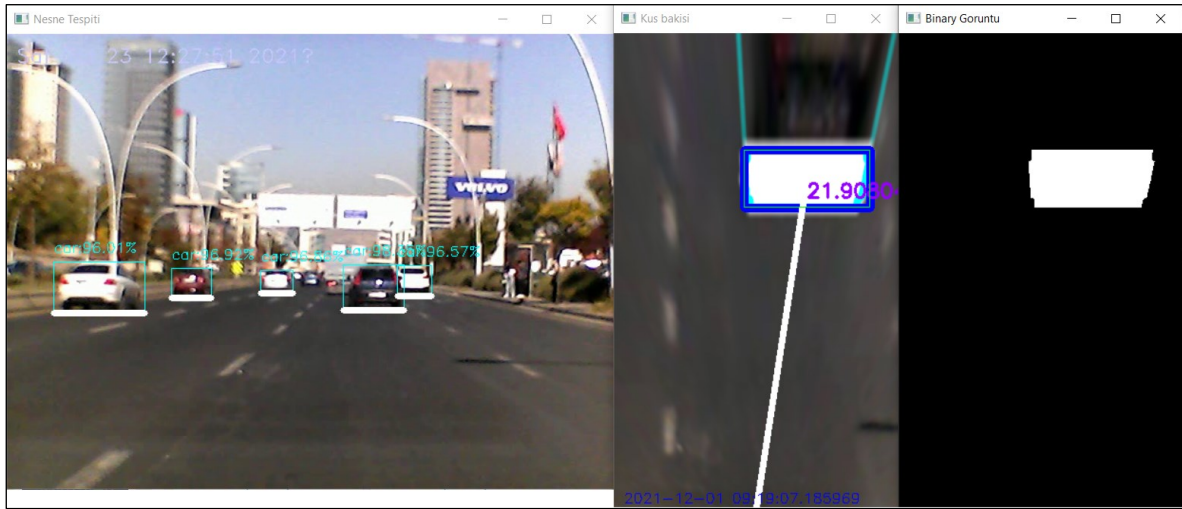
Şekil 6.5. Cumartesi günleri belirli saatlere göre trafik yoğunluğu

Test sürüşüne katılan sürücülerin yaş ortalaması 39'dur. En yaşlı sürücü 52, en genç sürücü 32 yaşındadır. Tüm katılımcılar erkek sürücülerden oluşmaktadır. Sürücülerin ortalama 22,5 yıllık bir sürüş deneyimi bulunmaktadır. En deneyimli sürücü 35 yıllık deneyime sahipken en az deneyimli sürücünün ise 14 yıllık deneyimi bulunmaktadır. Eğitim durumlarına bakıldığında da en az lise eğitimi en çok yüksek lisans eğitiminin olduğu grupta ortalama lisans mezunlarından oluşmaktadır. Katılımcıların yıllık araç kullanımları açısından

ortalama 15 000 km kullandıkları görülmüştür. Tüm katılımcıların son bir sene içerisinde trafik kazasına karışmış olup, yalnızca birisi sadece yaralanmalı trafik kazasına karışmıştır. Ayrıca son bir sene içerisinde hemen hemen tüm katılımcıların hız sınırını geçme cezasını bir kez almış oldukları görülmüştür.

Bu bilgiler doğrultusunda katılımcı grubunu deneyimli sürücülerden oluşan, yıllık araç kullanımları makul düzeyde, en az bir trafik kazasına karışmış ve yılda bir kez de hız sınırı cezasına çarptırılmış test grubu olarak tanımlayabiliriz.

Alınan kamera kayıtlarından matematiksel veri elde etmek amacıyla Python dilinde Anaconda platformunda Spyder derleyicisi ile bir program hazırlanmıştır. Bu programda görüntü işleme yöntemleri uygulanması amacıyla Opencv kütüphanesinden yararlanılmıştır. Görüntüden konum ve mesafe bilgisi edinilmiş ve bu bilgiler text dosyasına aktarılmıştır. Resim 6.5'te hazırlanan programla elde edilen sonuç görüntüsü bulunmaktadır.



Resim 6.5. Analiz program görüntüsü

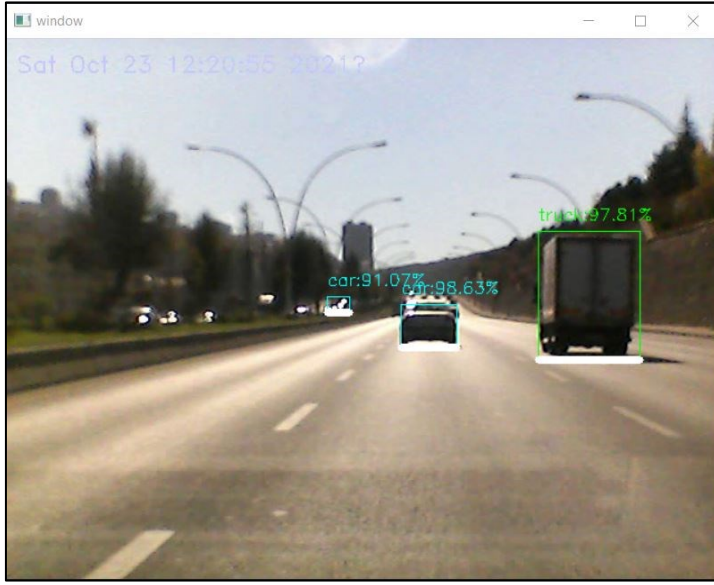
6.4. Öndeki Aracın Tespiti

Öndeki aracı tespit etmek için ayrıca Yolo kütüphanesinden faydalanılmıştır.

- Tespit edilecek nesne alanı = cv2.dnn.blobFromImage(resim çerçevesi,1/255, (416,416), swapRB = True, crop = False);

Kullanılan tespit edilecek nesnenin alanı için değişkeninin atanmasıyla, Resim 6.6'da bulunan resmi 4 boyutlu formata dönüştürür. Buradaki 1/255 ifadesi resmin yeniden boyutlandırılması için kullanılan ölçektir. Genelde 1/255 olarak kullanılır. Seçilen resim 416, 416 modeldedir. Ek olarak resmin renklendirmesinin RB formatında, ekstra resim kırpmalarının olmayacağı bilgisi bulunmaktadır.

- Etiketler = ["person","bicycle","car","motorcycle"....];



Resim 6.6. Analiz program görüntüsü öndeki araç tespiti

Yolo' nun resmi sitesinde tanımlanmış 80 nesne yer almaktadır. Bu nesneler arasında bizim testlerimizde faydalandığımız; Car, Truck, Bus da mevcuttur. Tespit edildikten sonra nesne başlıklarında bu etiketler belirecektir.

- Model =cv2.dnn.readNetFromDarknet("Yolo_Model/yolov3.cfg","Yolo_Model/yolov3.weights");

Eklenen metod ile “configuration” ve “weight” dosyalarına ulaşılır. Bu dosyalarda Yolo' nun hazırlamış olduğu veriler bulunmaktadır.

- Katmanlar=model.getLayerNames()
- Çıktı katmanı = [katmanlar[katman[0]-1] for katman in model.getUnconnectedOutLayers()] model.setInput(nesne alanı)

- Dedektör katmanları =model.forward(çıktı katmanı)

Oluşturulan katmanlar değişkeni ile modelde bulunan katmanlar çekilir. Bu katmanlar sayesinde nesne tespiti yapılabilir. Katmanlar arasında yalnızca çıktı katmanlarını almak için de çıktı katmanı değişkeni tanımlanmıştır. Modelin girişi nesne alanı olarak tanımlanan değişken atanır. Model.forward metodu ile tespit edilen çıktı katmanları gösterilerek tespit işlemi yapılmış olur.

- Sonuc=object_detection[5:]
- Tespit edilen nesnenin indeks numarası=np.argmax(sonuc)
- Güvenilirlik skoru=sonuc[tespit edilen nesnenin indeks numarası]

Tespit edilen katmanlarla, tespit edilmiş olan nesnelerin listelenmesi yapılır. Tespit edilen nesnelerin ilk beş adedinin değeri nesnenin çevresine çizilecek olan dörtgen ile ilgili bilgiler olduğundan beş değerden sonraki değerlere ulaşılır. Güvenilirlik skoru tahmin edilen nesnenin doğruluk oranının yüzde olarak verecektir.

- if Güvenilirlik skoru> 0.90:
- Etiket = Etiketler[tespit edilen nesne]

Eğer tespit edilen nesnenin güvenilirlik skoru %90 dan büyük ise nesne etrafına dörtgen çizimine izin verilecektir.

- Sınırlayıcı dörtgen = nesne dedektörü [0:4] * np.array([çerçeve genişliği, çerçeve uzunluğu, çerçeve genişliği, çerçeve uzunluğu])
- Dörtgen merkez x koordinatı, dörtgen merkez y koordinatı, dörtgen genişliği, dörtgen yüksekliği = Sınırlayıcı dörtgen.astype("int")
- Başlangıç x koordinatı = int(dörtgen merkez x koordinatı - (dörtgen genişliği /2))
- Başlangıç y koordinatı = int(dörtgen merkez y koordinatı - (dörtgen yüksekliği /2))

Nesne dörtgenini çizmek için belirlenen değerlerin, koordinat düzleminde ve en, boy çizilecek noktaları verecek şekilde olması için sayı (integer) tipine dönüşümleri yapılır.

- Dörtgenler listesi.append([başlangıç x koordinatı, başlangıç y koordinatı, int(dörtgen genişliği),int(dörtgen yüksekliği)])

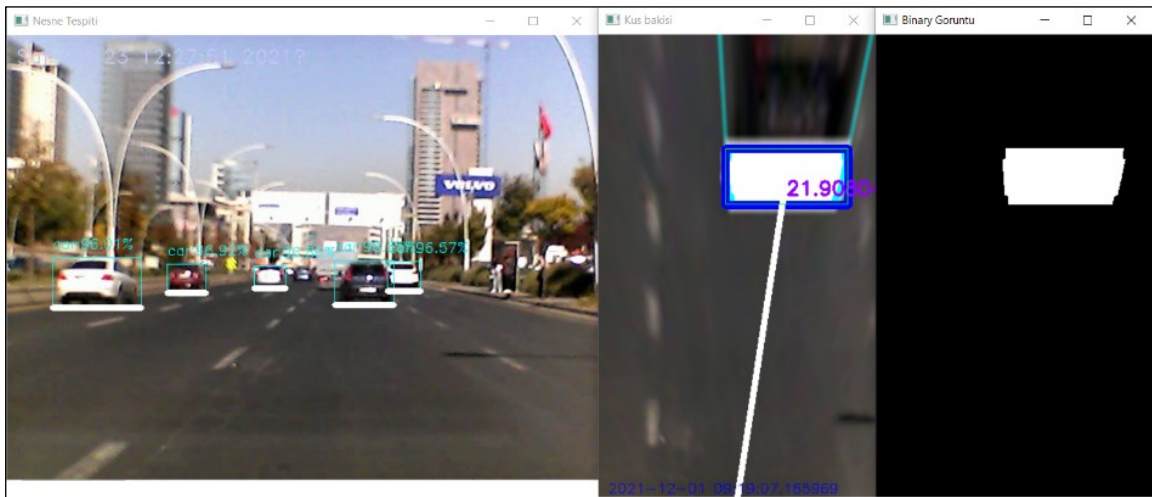
- Maksimum indeks sayısı=cv2.dnn.NMSBoxes(dörtgen listesi, güvenilirliği sınır üstündekiler ,0.5,0.4)

Bu bölümde tespit edilen nesnelerin indeks bilgisi, güvenilirlik değeri ve nesne dörtgeninin başlangıç, bitiş, yükseklik ve genişlik bilgileri alınır. Alınan bu değerler cv2.dnn.NMSBoxes() metodunda kullanılarak en yüksek güvenilirliğe sahip dörtgenlerin indeks değerlerini verecektir. 0,5 güvenlik değeri, 0,4 ise eşik değeridir. Bu süreçten sonra belirlenen dörtgenler çizilecektir.

6.5. Öndeki Araç İle Mesafe Tespiti

Taşıtın önünde tespit edilen araç ile olan mesafeyi ölçmek için kameradan alınan görüntü kuşbakışı görünümüne alınarak hesaplama yapılmıştır (Resim 6.7). Perspektif düzeltme yöntemi ile bu görüntüye ulaşılabilmektedir. Resim üzerinden belirlenen 4 nokta seçilir ve bu noktalar yeni bir resime aktarılarak izdüşüme ulaşılır (Resim 6.8).

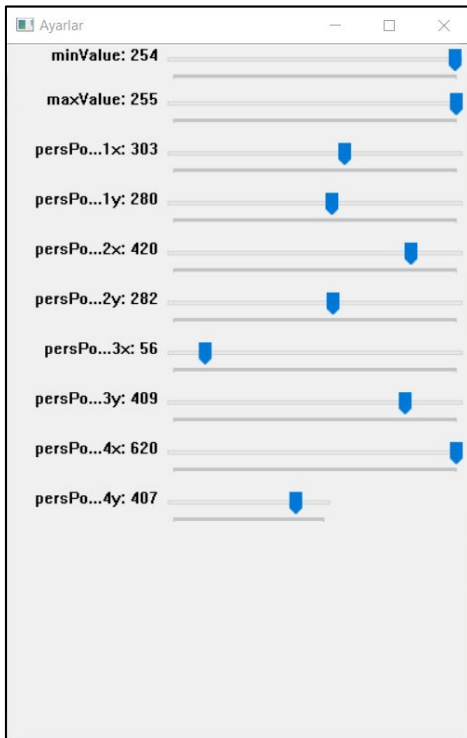
Kameranın konumuna doğru ayarlamamanın yapılabilmesi için ayar paneli oluşturulmuş ve bu panelde seçilecek x ve y noktalarının video kaydı izlenirken yapılabilmesi ve doğru görüntülerin alınması sağlanmıştır. Bu dört noktanın belirlenmesi için hazırlanan ayar paneli Şekil 6.6'da görülmektedir.



Resim 6.7. Kuşbakışı görüntü şerit çizgileri



Resim 6.8. Kuşbakışı görüntü tespiti alan belirlenmesi



Şekil 6.6. Ayar paneli

Amaçlanan şerit çizgilerinin piksel değerlerinin ölçülüp gerçek değerleri ile aralarındaki oranın tespit edilmesidir.

- `pts1=np.float32([[persPoint1x,persPoint1y],[persPoint2x,persPoint2y],[persPoint3x,persPoint3y],[persPoint4x,persPoint4y]])`
- `pts2=np.float32([[0,0],[width,0],[0,height],[width,height]])`

Görüntüden belirlenen 1, 2, 3, 4 noktalarına ait x ve y noktaları belirlenir. Sonrasında bu noktaların taşınacağı ikinci resim çerçevesinin noktaları belirlenir.

- `matrix=cv2.getPerspectiveTransform(pts1,pts2)`
- `output_perspective=cv2.warpPerspective(frame,matrix,(width,height))`

Tanımlanan görüntü noktaları ikinci noktalara taşıyacak matris atanır ve bu matrise göre ön görünüşün perspektifi alınarak görüntü kuşbakışı görüntü alınmış olur. İzdüşümü alınan görüntüde öndeki aracı tespit etmek için beyaz çizgiden yararlanılmıştır (Resim 6.9).



Resim 6.9. Öndeki araç mesafe tespiti

Tespiti yapabilmek için görüntü işleme yöntemlerinden morfoloji uygulamaları kullanılmıştır.

- `output=cv2.GaussianBlur(output_perspective,(5,5),cv2.BORDER_DEFAULT)`
- `output_gray=cv2.cvtColor(output,cv2.COLOR_BGR2GRAY)`

- `ret,thresh=cv2.threshold(output_gray,minTresh,maxTresh,cv2.THRESH_BINARY)`
- `contours,hierarchy=cv2.findContours(thresh,mode=cv2.RETR_TREE,method=cv2.CHAIN_APPROX_NONE)`
- `cv2.drawContours(output,contours,contourIdx=-1,color=(255,255,0),thickness=5,lineType=cv2.LINE_AA)`
- GaussianBlur: Resimdeki gürültüyü azaltmak için kullanılan komuttur.
- `output_gray`: Elde edilmiş üst görüntüdür.
- `minTresh`, `maxTresh`: Resimde siyah beyaz görüntü elde etmek için minimum ve maksimum eşik değerleridir.
- `thresh`: Resmin filtrelenmesinde kullanılan eşik değeridir. Amaç resimde bulunan arka planı çıkarmak ve resimi nesne tespiti için hazırlamaktır.
- `findContours`: Resim içerisinde nesne aramak için kullanılan komuttur.
- `drawContours`: Bulunan nesne beyaz olarak için dolduran komuttur.

Alınan kuşbakışı görüntüsü önce blur yöntemi ile resim yumuşatılıp ve sonrasında gri tona dönüştürülmüştür. Görüntüde eşik değeri belirlenerek siyah beyaz dönüşümle görüntüdeki nesneler belirlenmiştir. Tespit edilen aracın alt bölümüne çizilen beyaz çizgi, nesne olarak gösterip merkezinin başlangıç noktasına uzaklığı piksel değeri belirlenmiştir. A ile gösterilmiş şerit çizgisinin gerçek değeri 4,5 metre, resim üzerindeki uzunluğu ise 87 pikseldir. B ile gösterilen uzunluk ise piksel değeri 630'dur.

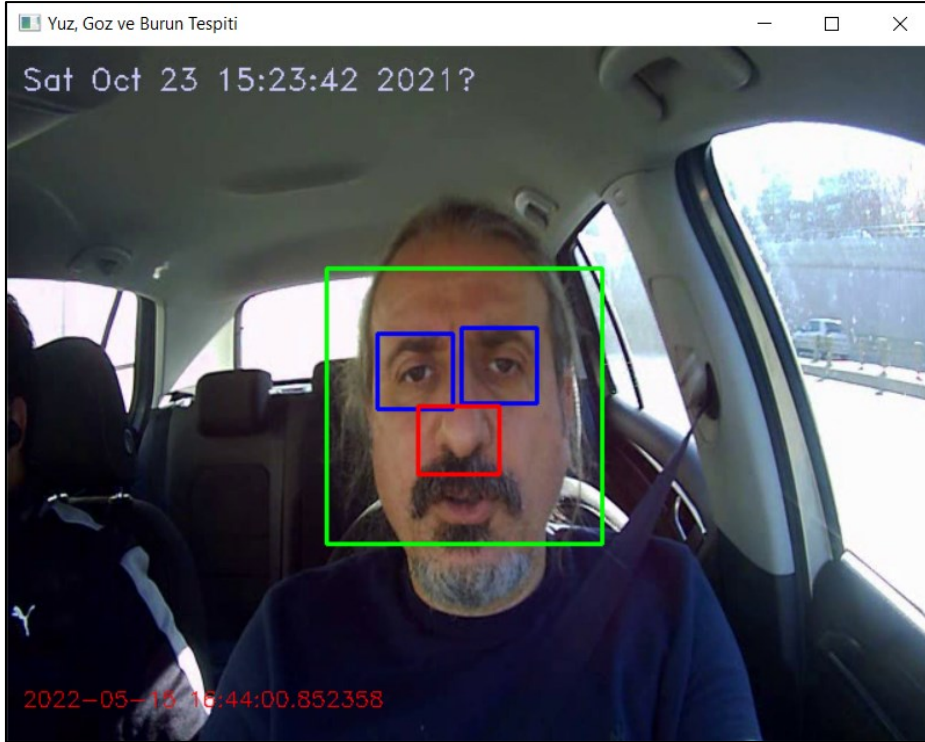
Tespit edilen noktanın belirlenen başlangıç noktasına olan uzaklığı Eşitlik 6.1'deki gibi hesaplanmıştır.

$$\text{Mesafe} = (630 - \text{Nesnenin Merkez Y Noktası}) * 4,5/87 \quad (6.1)$$

6.6. Sürücünün Yüz, Göz ve Burun Tespiti

Çalışmada sürücünün seyir halindeyken bakış doğrultusunu varsaymak amacıyla göz ve yüz konum tespiti yapılmıştır. Her iki göz ayrı ayrı tespit edilmiş ve bu değerlere ek olarak burun konum bilgisi de eklenmiştir. Böylece yüzün konumu kendi aralarında mesafesi sabit olan 3 nokta ile tespit edilmiştir (Resim 6.10). Haar Cascade yönteminden yararlanarak bu tespitler yapılmıştır. Yüz, göz ve burun tespiti için opencv kütüphanelerinde bulunan xml dosyaları kaynak sağlamaktadır.

- $Yüz = yuz_kademelitespit.detectMultiScale(Görüntünü\ gri\ şekli, 1.1, 5, Minimum\ dörtgen\ çerçeve = (150, 150))$
- $Çerçeve = cv2.rectangle(çerçeve, (x, y), (x+w, y+h), (0, 255, 0), 2)$



Resim 6.10. Sürücünün yüz, göz ve burun tespiti

Üzerinde arama yapılacak görüntü gri forma dönüştürülür. DetectMultiScale metodu ile de bu çerçeve üzerinde arama tarama yapar. Bu esnada kullanılacak veri kütüphanesi “face_cascade” dir. 1.1 arama işleminde resimlerdeki ebat farklarını ortadan kaldırmak için belirtilen katsayı kadar küçültülür. 5 değeri en düşük komşuluk ilişkisini ifade etmektedir. 150,150 değeri de tespit edilen yüz için çizilecek en küçük dörtgenin yükseklik ve genişliğini belirtir.

Yüz tespiti yapıldıktan sonra, bulunan alan içerisinde göz ve burun taraması yapılır. Burada göz ve burun için çizilmesi istenen dörtgen 50x50 boyutlarındadır. Yüz, göz ve burun tespitleri yapılmasıyla bu nesnelerinin merkez noktaları da bulunmuş olur. Test verisi olarak sağ göz ve burun merkezi değerlendirmeye alınmıştır.

6.7. Test Verilerinin Oluşturulması

Test için girdi olarak; göz hareketleri ve burun konumu ile birlikte öndeki aracın konumu ve mesafesi ve kullanılan aracın hız bilgisi alınmıştır. Veriler öndeki aracı geçme eyleminden 3 saniye öncesine kadar olan bilgilerin ortalaması alınarak oluşturulmuştur. Her bir satırdaki eylemin sonu öndeki aracı sollama ile geçilmiş ise 1 tersi ise 2 olarak sınıflandırılmıştır. Hazırlanan veri listesi Çizelge 6.1’de verilmiştir.

Çizelge 6.1. Test ve eğitim verisi

Sürücü	Mes. Ort	X Ort	Y Ort	Brn_Ort X	Brn_Ort Y	Gz_Ort X1	Gz_Ort Y1	Dir. Açı	Hız	Sonuç
AG1	22,5	159,7	141,4	60,6	71,3	59,1	44,2	-1,2	89,48	1
AG10	21,1	218,7	170,5	52,8	77,5	62,5	44,5	-0,5	68,56	1
AG11	23,5	163,4	118,9	54,0	72,9	49,2	44,5	1	62,36	1
AG12	9,1	193,3	432,4	59,7	63,4	58,9	45,5	-0,2	63,05	1
AG13	8,9	189,9	436,9	56,0	72,6	54,2	45,2	0,8	56,23	1
AG2	24,6	209,4	95,9	61,5	64,2	56,7	43,2	0,3	91,8	1
AG4	22,7	189,1	135,8	64,8	74,3	48,4	35,7	-8,4	85,61	1
AG5	17,0	129,0	259,7	59,2	73,0	49,5	45,6	0,7	78,34	1
AG6	7,9	190,5	457,3	59,4	70,6	53,5	42,9	-6,3	99,2	1
AG9	13,5	245,6	336,8	56,1	61,1	46,0	45,2	-2,3	62,62	1
CS1	24,9	83,3	88,1	64,0	36,5	35,8	33,0	1	95,06	1
CS11	8,2	156,0	451,5	63,8	64,7	63,6	43,2	0,6	82,5	1
CS3	20,0	193,1	195,8	55,6	52,8	56,0	41,1	-0,1	89,58	1
CS4	17,5	190,0	250,4	54,7	66,4	59,4	43,6	0	89,73	1
CS5	18,8	177,3	220,8	50,0	65,6	48,9	44,4	0,8	66,59	1
CS6	18,9	175,0	219,5	55,9	62,6	44,2	44,0	1	95,64	1
CS9	22,0	175,8	150,8	52,0	64,3	48,5	45,1	0,1	59,12	1
EY3	19,7	172,0	200,7	56,2	48,2	65,5	44,0	1,5	72,75	1
EK1	14,8	184,5	309,0	47,6	66,8	49,9	45,2	0	58,98	1
EK11	11,8	149,9	373,5	62,5	52,4	35,7	43,0	-2,1	62,33	1
EK12	26,5	11,3	53,7	27,1	31,8	32,8	26,8	1,2	88,69	1
EK13	18,8	132,8	220,5	55,3	63,5	50,3	42,2	2,2	91,66	1
EK14	15,4	220,7	295,5	55,5	62,5	50,8	42,9	0,1	80,66	1
EK3	18,8	234,2	220,4	54,5	59,3	50,3	42,4	0,3	95,62	1
EK4	22,2	152,3	146,3	54,0	56,7	52,3	45,6	-3,6	68,08	1
EK7	12,7	152,7	354,8	51,0	62,0	53,5	46,0	0,9	65,28	1
EK8	11,9	227,8	371,5	55,0	57,3	60,1	44,7	-0,4	55,67	1
EK9	18,3	169,2	232,6	54,3	51,8	57,4	43,0	0	71,52	1
HE1	12,9	137,2	350,4	55,5	64,1	48,2	39,7	-1	71,81	1

- Sürücü; Testi gerçekleştiren kişi bilgisidir.
- Mes. Ort: Öndeki araç ile olan mesafeyi metre olarak gösterir.
- X Ort; Öndeki aracın merkezinin x koordinatıdır.
- Y Ort; Öndeki aracın merkezinin y koordinatıdır.

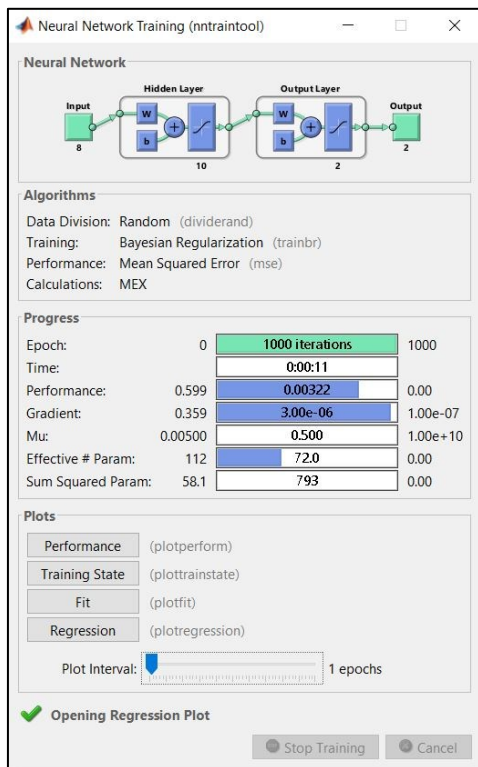
- Brn_Ort X; Burun merkezinin x koordinatıdır.
- Brn_Ort Y; Burun merkezinin y koordinatıdır.
- Gz_Ort X1; Sağ gözün merkezinin x koordinatıdır.
- Gz_Ort Y1; Sağ gözün merkezinin y koordinatıdır.
- Dir.Açı; Direksiyon açısı
- Hız; Test aracının hızı. (km/saat)

Öndeki araç bilgisi ve yüz, göz burun bilgisi ayrı ayrı analiz edilmiş. İstenen koordinat ve mesafe bilgisi excel dosyalarına kaydedilmiştir. Sonrasında analiz zaman bilgisi ile sürüş zaman bilgisi eşleştirilerek excel dosyaları birleştirilmiş tek dosya haline getirilmiştir. Odis test cihazında elde edilen kayıtlardaki zaman bilgisi ile video kayıtlarındaki zaman bilgisi aynı olduğu için yine excel dosyalarına direksiyon açısı ve araç hız bilgisi eklenmiştir.

7. YSA MODELİ VE SONUÇLARIN DEĞERLENDİRİLMESİ

7.1. YSA Modelinin Oluşturulması

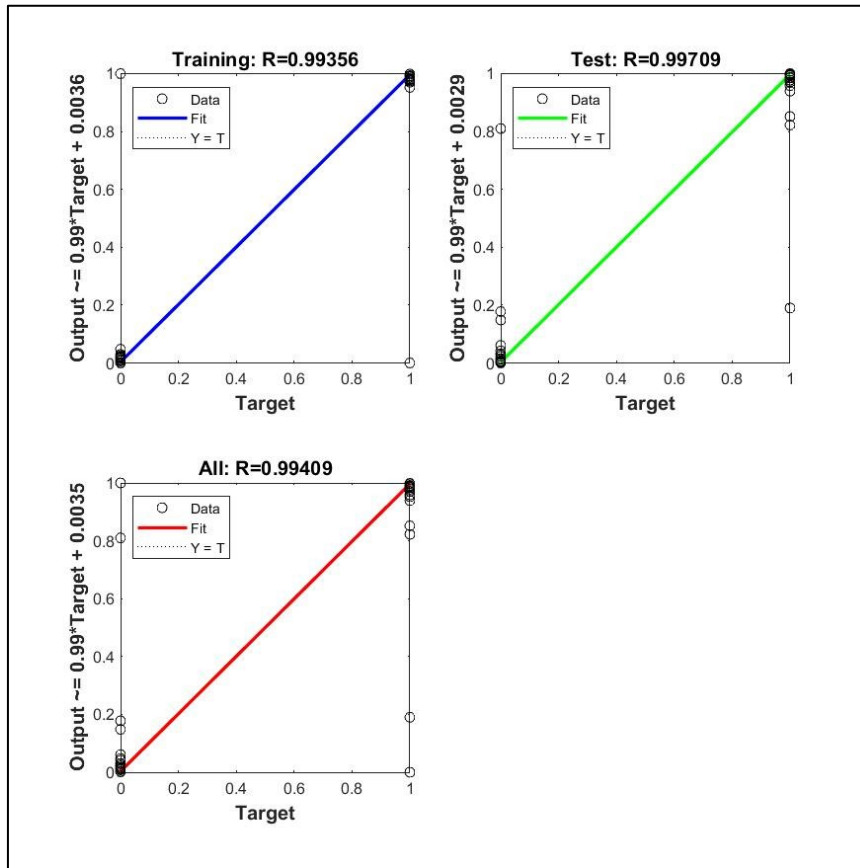
Bu çalışmada sürücünün öndeki aracı geçme isteğinin oluşması ya da oluşmaması şeklinde iki seçenekli bir sınıflandırma modeli kurulması amaçlanmıştır. Bu doğrultuda elde edilen değişkenler çok katmanlı yapay sinir ağ modelinde kullanılmıştır. Yapay sinir ağ modelinin oluşturulmasında Matlab 2020a programından yararlanılmıştır. Yapılan test sürüşlerinde öndeki araca geçme süreçleri tespit edilerek bu süreçlerin verileri pozitif olarak alınmıştır. 17,6 km'lik sürüş deneyimi içerisinde 53 kez yapılan bu eylem diğer süre aralıklarından ayrıştırılmıştır. Bu eylem zaman aralığındaki konum bilgilerinin ortalaması ve hareket bilgisinin en yüksek olduğu değer alınarak bir pozitif değer tablosu hazırlanmıştır. Ek olarak bir de negatif eğitim verisi de oluşturulmuştur. Eğitim sonrasında eğitim veri setinde olmayan 29 veri seti hazırlanmıştır. Şekil 7.1'de yapay sinir ağ modeli ve eğitim paneli görülmektedir.



Şekil 7.1. YSA modeli ve eğitim paneli

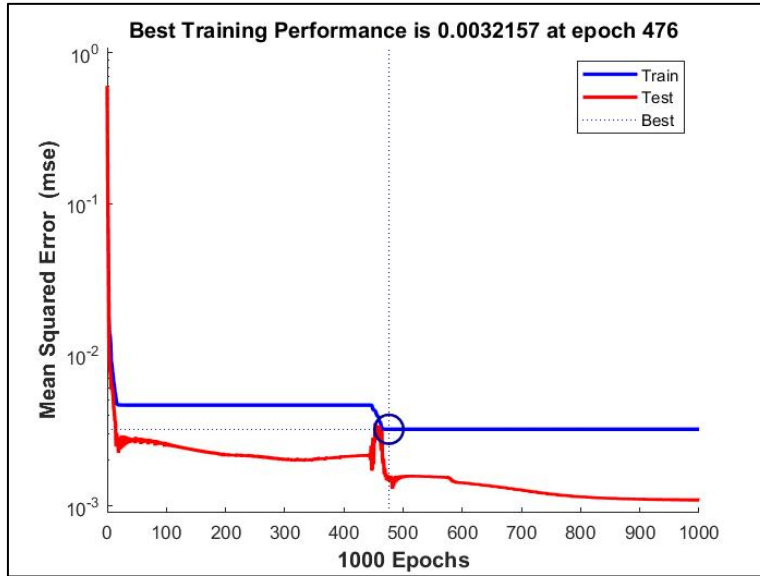
Alınan bilgiler doğrultusunda 10 gizli katmana ve 2 çıkış katmanına sahip çok katmanlı ağ oluşturulmuş ve eğitimi gerçekleştirilmiştir. Eğitim algoritması olarak “Bayesian Regularization” seçilmiştir. Bu eğitim algoritması ile diğer eğitim algoritmalarına (Levenberg-Marquardt vb.) göre daha iyi eğitim sonucuna ulaşıldığı için seçilmiştir.

Eğitim sonrasında regresyon grafikleri Şekil 7.2’deki gibi elde edilmiştir. Burada eğitim regresyonu katsayısı 0,99, test regresyon katsayısı 0,98 ve tüm veri setinin regresyonun da 0,99 çıktığı görülmüştür. Yüz konum hareketi, öndeki araç ile olan mesafe, aracın hızı ve direksiyon açısını göz önünde bulundurduğumuzda öndeki aracı geçme isteği arasında bağıntının yüksek olduğu söylenebilmektedir.



Şekil 7.2. YSA eğitim sonu regresyon grafikleri

Matlab programı tarafından her iterasyonda ağın ne kadar eğitildiğini görmek için belirlediği validasyon değerinin 476. iterasyonda en iyi değerine geldiği Şekil 7.3’de görülmektedir.



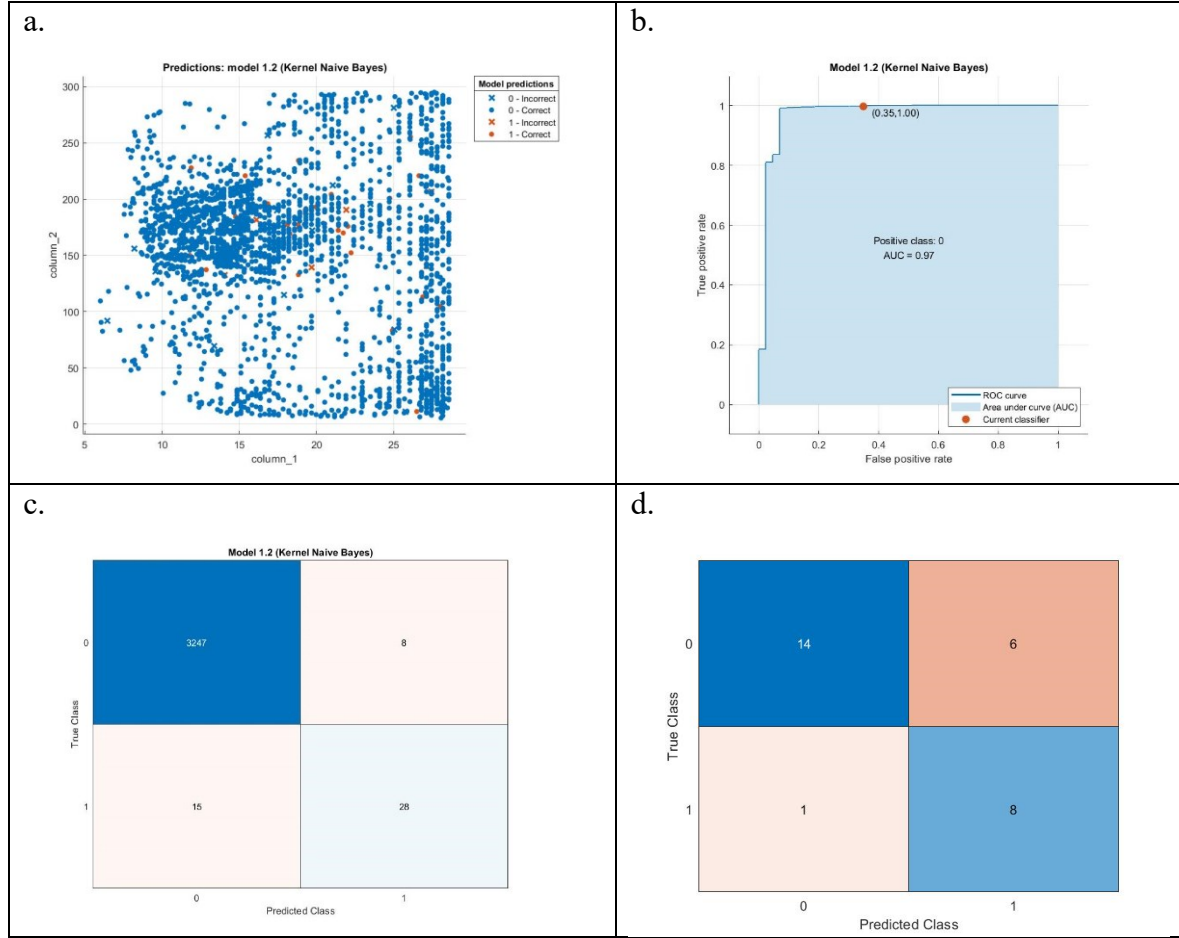
Şekil 7.3. YSA eğitim performans grafiği

7.2. Makine Öğrenmesi ile Bazı Sınıflandırma Yöntemlerinin Uygulanması

Çalışmada yapay sinir ağı modelinin yanı sıra makine öğrenmesinde en çok karşımıza çıkabilecek olan Naive Bayes, en yakın komşu, destek vektörü ve karar ağaçları sınıflandırma algoritmalarında da denemeler yapılmıştır.

7.2.1. Naive Bayes sınıflandırma algoritması

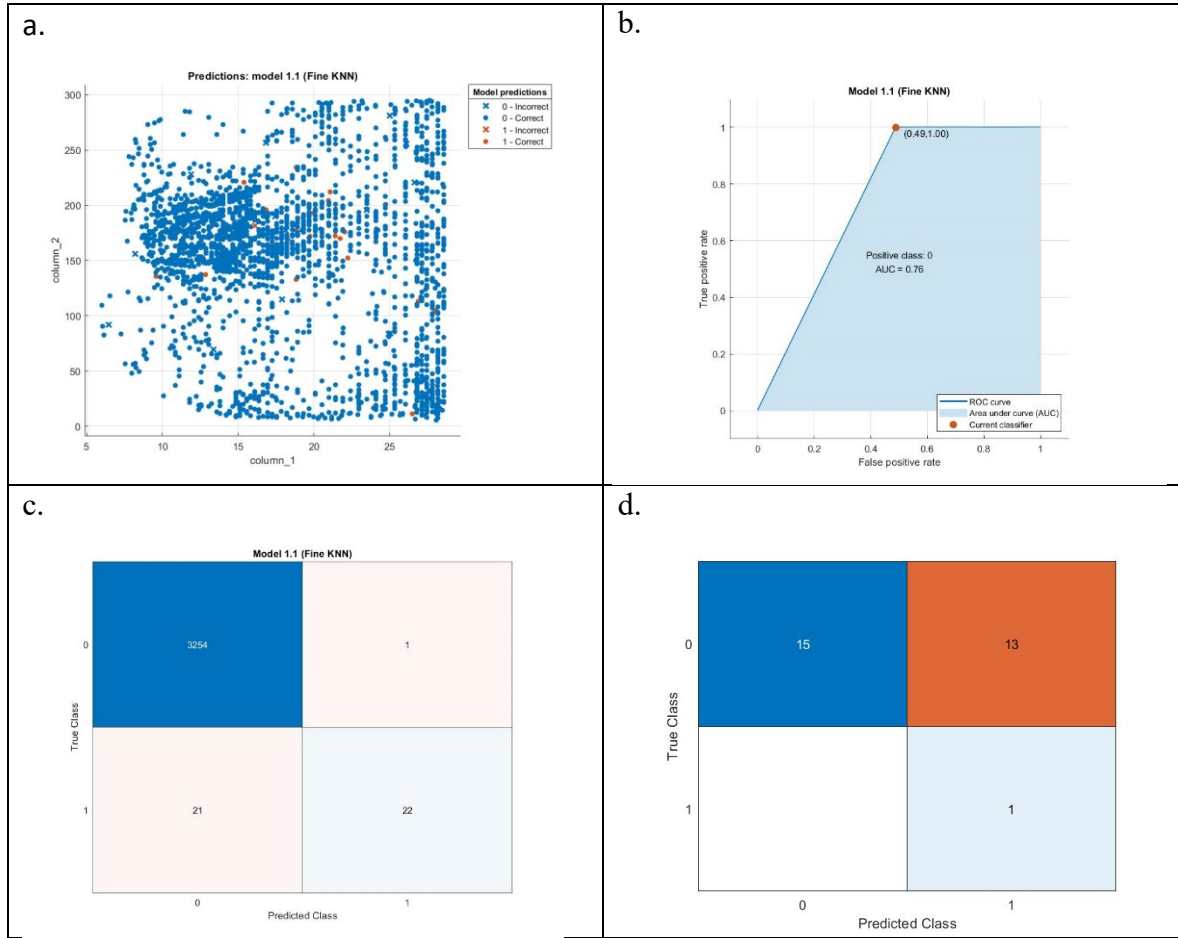
Şekil 7.4’de (a) bölümünde sollamanın olduğu kırmızı, olmadığı mavi noktaların dağılımı görülmektedir. (b) kısmında ROC eğrisinin altında kalan alanın en yüksek olduğu eğitim sonucu gözükmemektedir. (c) ile gösterilen alanda ise eğitimin karmaşıklık matrisine göre başarı oranı %99 çıkmasına karşın (d) bölümündeki test karmaşıklık matrisinde sonuç %75,8 çıkmıştır.



Şekil 7.4. Naive Bayes sınıflandırma sonuçları

7.2.2. En yakın komşu (KNN) sınıflandırma algoritması

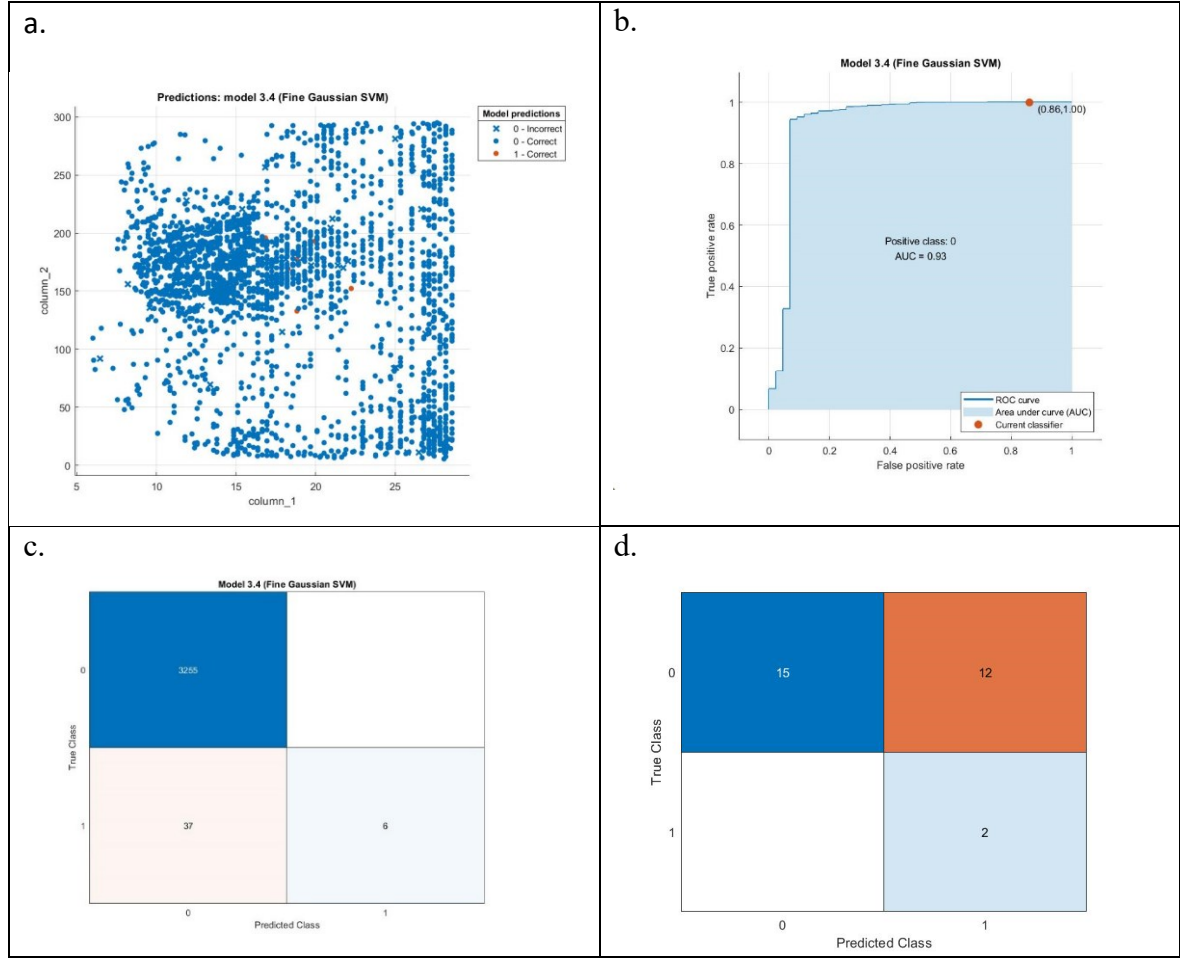
Şekil 7.5’de (a) bölümünde sollamanın olduğu kırmızı verinin daha az hatalı bilgi aldığı, geçişin olmadığı mavi noktaların dağılımı görülmektedir. (b) kısmında ROC eğrisinin altında kalan alandan performansının iyi olmadığını söyleyebiliriz. (c) ile gösterilen bölümde de eğitimin karmaşıklık matrisine göre başarı oranı %99 çıkmasına karşın (d) bölümündeki test karmaşıklık matrisine göre doğru sonuç performansı %55,1 çıkmıştır.



Şekil 7.5. En yakın komşu sınıflandırma sonuçları

7.2.3. Destek vektörü (SVM) sınıflandırma algoritması

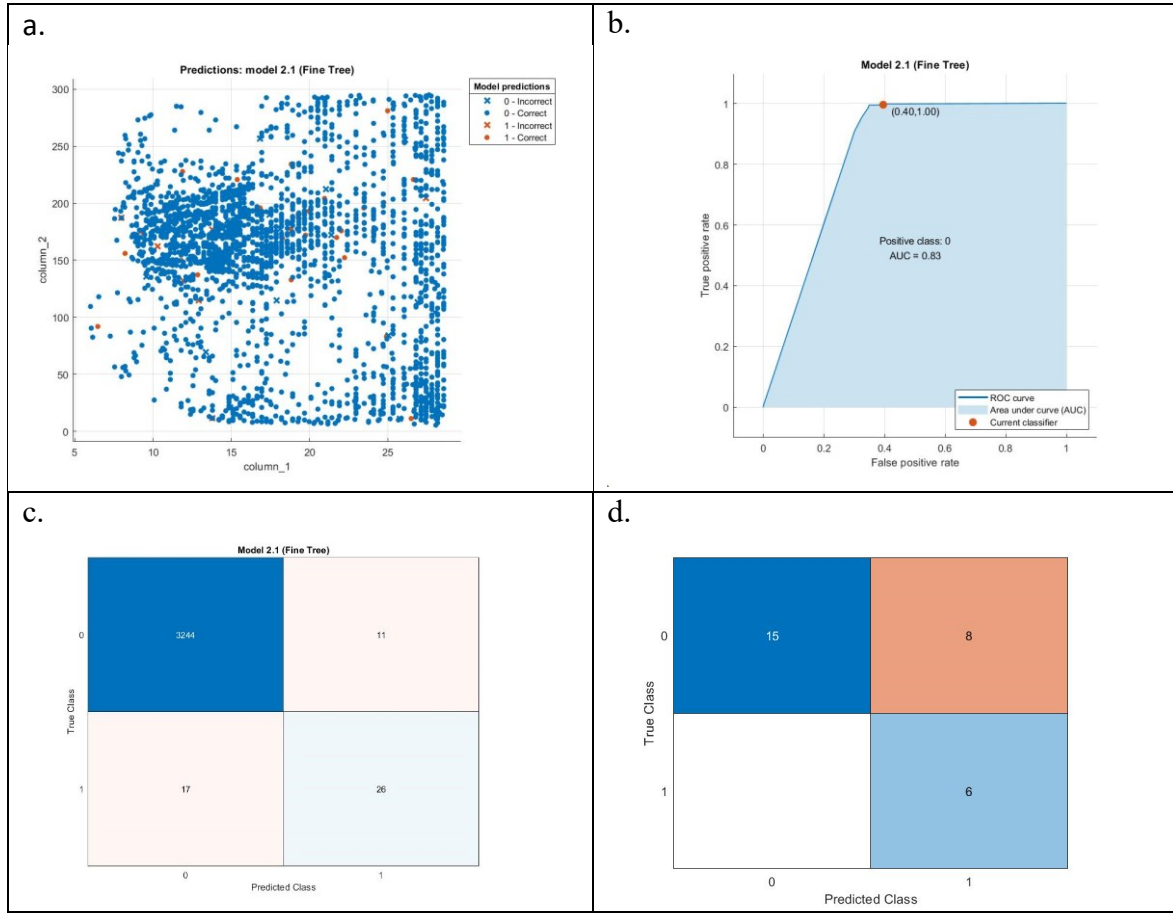
Şekil 7.6'da (a) kısmında solumanın olduğu kırmızı verinin hatasız şekilde eklenebildiği görülmektedir. Geçişin olmadığı eylemler ise mavi noktalar olarak gösterilmektedir. (b) kısmında ROC eğrisinin altında kalan alandan performansının iyi olmadığını söyleyebiliriz. (c) ile gösterilen bölümde de eğitimin karmaşıklık matrisine göre başarı oranı %99 çıkmasına karşın (d) bölümündeki test karmaşıklık matrisine göre doğru sonuç performansı %58,6 çıkmıştır.



Şekil 7.6. Destek vektörü sınıflandırma sonuçları

7.2.4. Karar ağaçları sınıflandırma algoritması

Şekil 7.7’de (a) kısmında sollamanın olduğu kırmızı verinin grafiğe dağınık şekilde yayıldı görülmektedir. Sollamanın olmadığı eylemler ise mavi noktalar olarak gösterilmektedir. (b) kısmında ROC eğrisinin altında kalan alandan da performansının iyi olmadığını söyleyebiliriz. (c) ile gösterilen bölümde de eğitimin karmaşıklık matrisine göre başarı oranı %99 çıkmasına karşın (d) bölümündeki test karmaşıklık matrisine göre doğru sonuç performansı %72,4 çıkmıştır.



Şekil 7.7. Karar ağaçları sınıflandırma sonuçları

Yapay sinir ağları çok katmanlı yapısı ile sınıflandırma algoritması oluşturulmuştur. Oluşturulan ağ 8 girdi, 10 gizli ve 2 çıktı katmandan oluşmaktadır. Direksiyon açısı bilgisini de değişkenlere ekleyerek kurulan ağın eğitim ve test sonuçlarının çok yüksek performans değerine ulaşılmış ve %100'e yakın çıkmıştır. Direksiyon açısı bilgisinin ağ eğitiminde manipülasyona neden olduğu görülmüş ve değişkenler arasından çıkarılarak çalışma sürdürülmüştür. Tüm test sürüşünden elde edilen 31 314 satır veri, anlamlı veri olarak 3332 satıra indirgenmiştir. Veri setin 59 adet pozitif 3273 adet negatif sonuçlu ölçümden oluşturulmuştur. Sınıflandırma problemlerinde etkili olduğu bilinen Bayesian regülasyonu ile eğitim gerçekleştirilmiştir.

Eğitim hata matrisine göre 3 288 doğru, 10 yanlış tahmin sonucuna Şekil 7.8’de gösterildiği şekilde ulaşılmıştır.

$$\text{Hata Oranı} = \frac{\text{Yanlış Tahmin Edilenler}}{\text{Toplam Tahmin Edilenler}} = \frac{10}{3332} = \% 0,3$$

EĞİTİM		
	TAHMİN	
GERÇEK	34	9
	1	3288

Şekil 7.8. YSA eğitim hata matrisi

Bu sonuçlara göre hata oranı % 0,3 gibi düşük bir değer çıkmıştır. Diğer bir ifadeyle %99,7 başarı değerine ulaşılmıştır. Oluşturulan modele veri setinde yer almayan 29 adet veri gösterilerek test edilmiştir.

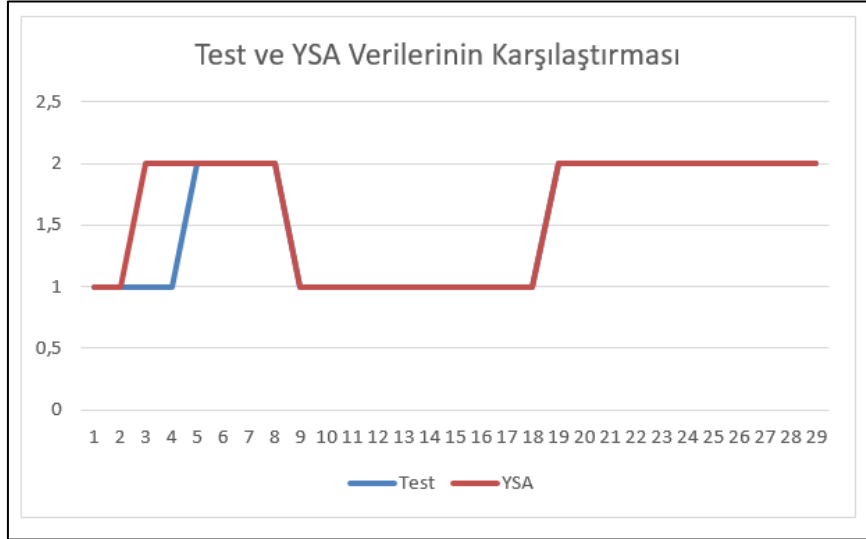
$$\text{Hata Oranı} = \frac{\text{Yanlış Tahmin Edilenler}}{\text{Toplam Tahmin Edilenler}} = \frac{2}{29} = \% 6,9$$

Test sonuçlarının hata oranı Şekil 7.9’daki hata matrisine göre %6,9 gibi düşük bir değer çıkmıştır. Diğer bir ifadeye %93,1 doğruluk oranı çıkmıştır. Literatür çalışmasında bahsedilen Peng [71] ve arkadaşlarının çalışmalarında elde ettikleri %85,5 performansa göre yüksek çıktığı görülmüştür.

TEST		
	TAHMİN	
GERÇEK	12	0
	2	15

Şekil 7.9. YSA test hata matrisi

Şekil 7.10'da test için hazırlanmış 29 veri setinin sonuç değerleri ile bu test verilerinin kullanıldığı yapay sinir ağının sonuçlarının karşılaştırıldığı grafik yer almaktadır. 3 ve 4. satırlarda hatalı diğer satırlarda doğru sonuç verdiği görülmektedir.



Şekil 7.10. Test ve YSA verilerinin karşılaştırılması

Elde edilen veri setinin makine öğrenmesi sınıflandırma algoritmalarında da modelleme çalışması yapılmıştır. Naive Bayes, en yakın komşu, destek vektörü ve karar ağaçları sınıflandırma algoritmalarında modeller kurulmuştur. Kurulan modellerin eğitim sonuçları %99 başarı elde etmesine karşın, testlerde Naive Bayes için %75,8, en yakın komşular algoritması için %55,1, destek vektörleri algoritması için %58,6 ve karar ağaçları algoritması için %72,4 değerleri elde edilmiştir.

8. SONUÇ VE ÖNERİLER

Dünya Sağlık Örgütü'ne göre son yıllarda dünya genelinde yılda 1,35 milyon kişi trafik kazalarında hayatını kaybetmektedir. Trafik kazalarının dünya üzerindeki ölüm nedenleri arasında en çok görülen 7. sebep olduğu ifade edilmektedir [1]. Trafik kazalarının oluşumundaki en büyük payı %88 ile %90 aralığında sürücünün aldığı görülmüştür [2]. Sürücü hatalarını en aza indirerek kazaların önüne geçilebileceği düşünülebilir. Öndeki aracı geçmek için yapılan sollama eyleminde yapılan hatalar, erken şerit değişimi, öndeki araç ile mesafeyi ayarlayamama, yan geri görüş aynasına bakmadan eyleme başlama, yeterli hızlanmayı sağlamama bu hatalar arasında yer almaktadır. Bu amaçla sürücü davranışları ile şerit değiştirme arasındaki ilişki kurulduğunda ve şerit değiştirme eyleminin gerçekleşmesinden önce tahmin edilebilirse olası hatalar öncesinde uyarıda bulunulabileceği düşünülebilir.

Bu çalışmada, sürücü davranışlarından öndeki aracı geçme eylemini tahmin modeli oluşturulmaya çalışılmıştır. 8 katılımcıyla, gerçek yol ve sürüş şartlarında testler gerçekleştirilmiştir. Test güzergâhının, öndeki aracı normal sürüşte geçebilecek zamanı sağlayacak uzunlukta ve trafik yoğunluğunun buna müsaade edebilecek düzeyde olacağı zamanlar seçilmiştir. Sürücülere araçları nasıl kullanacakları, hangi eylemleri yapacakları vb. kullanım şekilleri ile ilgili bildirimler yapılmamış, yalnızca trafik kurallarına uymaları istenmiştir.

Modelleme için gerekli olan bilgi, sürüş esnasında sürücünün yüz hareketlerinden, aracın hızı ve direksiyon açısı bilgisinden ve öndeki araç ile olan mesafenin ölçülmesiyle elde edilmiştir. Sürücü yüz hareketleri ve öndeki aracın tespiti için iki kamera, araç hızı ve direksiyon açısı bilgisi için taşıt CAN Bus hattından test cihazı ile elde edilmiştir. İki kameradan alınan video kaydının zaman senkronizasyonunda hata olmaması amacıyla Visual Studio C++ üzerinde video kayıt programı hazırlanmıştır. Sürücünün yüz ve göz hareketlerini, öndeki aracın tespiti ve mesafe ölçümü için ise Python programı ile Yolo ve Opencv kütüphaneleri kullanılarak program hazırlanmış, görsel bilgi matematiksel veriye dönüştürülmüştür. Elde edilen veriler excel dosyasında birleştirilmiştir.

Oluşturulan veriden sürücü testleri incelenerek sollama yapmaya başladığı andan önceki 3 saniye içerisindeki bilgilerin ortalaması alınarak veri seti oluşturulmuştur. Tüm

katılımcıların toplam 53 sollama eylemi belirlenmiştir. Öndeki aracı geçme ile ilgili yapılmış çalışmalarda belirli saniyede bu tahminin yapılması üzerinde çalışıldığı görülmüştür. Bu çalışmada ise hareket kararının vermeden 3 saniye öncesine kadar olan zaman dilimindeki bilgiler kullanılmıştır.

Çalışmanın sonucunda sürücülerin hatalı sollama yapmadan 3 saniye öncesinde sollama yapıp yapmayacağını tahmin modeli ortaya konulmuştur. Model sınıflandırma problemi çerçevesinde çok katmanlı ağ yapısı oluşturulmuştur. Ağ 10 gizli katman ve 2 çıktı katmanı ihtiva etmiştir. Bayes Regülasyonu ile ağ eğitimlerinde başarılı sonuç elde edilmiştir. Ağın eğitimi sonrasında test veri setinin ağa gösterilmesiyle %93,1 doğrulu ortaya çıkmıştır.

Oluşturulan yapay sinir ağı sınıflandırma modeli diğer makine öğrenmesi, Naive Bayes, en yakın komşular algoritması, destek vektörleri algoritması, karar ağaçları algoritmalarına göre daha başarılı sonuç vermiştir. Modellerin eğitim sonuçları %99 başarı elde etmesine karşın, testlerde Naive Bayes için %75,8, en yakın komşular algoritması için %55,1, destek vektörleri algoritması için %58,6 ve karar ağaçları algoritması için %72,4 değerleri elde edilmiştir.

Yapay sinir ağı ile kurulan modelin makine öğrenmesi modellerine göre daha başarılı sonuç verdiği görülmüştür. Oluşturulan ağın doğruluk oranının rakamsal değerinin yüksek olmasına karşın, kullanılan algılayıcılar ve veri setinin farklı sürücülerde uzun araç sürüşlerinden elde edilmesi bu oranı daha da artırabilecektir.

Sürücülerin öndeki aracı geçme isteğinin eylemden çok önce tahmin edilebilmesi erken uyarı sistemlerine doğru bilgi akışı sağlayabilecektir. Böylece trafikte yaşanan hatalı sollama kazalarının aza indirilmesine fayda sağlayacaktır. Ayrıca bu tip sürücü hareketlerinin önceden tahmin edilebilmesi diğer riskli sürüş (ani fren yapma, ani hızlanma, trafik ışık kurallarına uymama vb.) eylemlerinin önüne geçilmesinde önemli pay sahibi olacaktır.

KAYNAKLAR

1. World Health Organization (2018). Global Status Report On Road Safety, *World Health Organization*, İsviçre, 3-7.
2. Türkiye İstatistik Kurumu (2021). Karayolu Trafik İstatistikleri 2020, *Türkiye İstatistik Kurumu*, Ankara, 1-3.
3. Karaoğlu, R. (2019). *Motorlu Kara Taşıtlarında Meydana Gelen Maddi Hasarlı Trafik Kazalarının Ülke Ekonomisine Etkisi*. Yüksek Lisans Tezi, Bursa Uludağ Üniversitesi Fen Bilimleri Enstitüsü, Bursa, 16.
4. İnternet: World Road Traffic Accident Report. (2019). URL: <https://www.worldlifeexpectancy.com/world-road-traffic-accidents-report>, Son Erişim Tarihi: 05. 02. 2020.
5. İnternet: Yıllara Göre Kaza, Ölü ve Yaralı Sayısı. (2020). URL: <http://www.tuik.gov.tr/UstMenu.do?metod=temelist>, Son Erişim Tarihi: 05. 02. 2020.
6. Karayolları Trafik Kanunu (1983). *T.C. Resmi Gazete Sayı* (18195, 18 Ekim 1983)
7. Tung, C. (2014). *The Investigation Of Distracted Driving At Highway Rail Grade Crossing*, Doctoral Thesis, The Faculty of The Graduate College at the University of Nebraska, Nebraska, 22-27.
8. Stutts, J.C., Reinfurt, D.W., Staplin, L., Rodgman, E. A. (2001). The Role of Driver Distraction in Traffic Crashes. AAA Foundation for Traffic Safety, May 2001, Washington, 8.
9. Young, K., Regan, M. (2003). Driver Distraction: A Review of the Literature. *Australasian College of Road Safety*, 379-405.
10. Sadeqi, D. (2016). *Data Analysis of Crashes Involving Distracted Drivers in Texas*, Master's Thesis, The University of Texas at San Antonio, Texas, 1-13.
11. Karayolları Trafik Yönetmeliği. (1997). *T.C. Resmi Gazete*. (23053, 18. Temmuz .1997)
12. Kuyumcu, Z. Ç., Aslan, H., Yose, A., M., Ahadi, S. (2020, 23-25 Ekim). *Türkiye’de Trafik Kazaları ve Sürücülerin Kazalardaki Payı*. 8th International Symposium on Innovative Technologies in Engineering and Science, Bursa.
13. İnternet: Çayıroğlu, İ. (2022). Görüntü İşleme. URL: http://www.ibrahimcayiroglu.com/Dokumanlar/GoruntuIsleme/Goruntu_Isleme_Ders_Notlari-1.Hafta.pdf, Son Erişim Tarihi: 01.11.2022.
14. Dilek, B. G. (2012). *Video Görüntüleri İçinden Hareketli Nesne Ayıklanması ve İzlenmesi*. Yüksek Lisans Tezi, Fırat Üniversitesi Fen Bilimleri Enstitüsü, Elazığ, 3-4.

15. Önal, R. (2019). *Görüntü İşleme Teknikleri ve Yapay Sinir Ağları Kullanılarak İrmiğin Seçilmiş Kalite Parametrelerinin Belirlenmesi*. Yüksek Lisans Tezi, Kahramanmaraş Sütçü İmam Üniversitesi Fen Bilimleri Enstitüsü, Kahramanmaraş, 22-26.
16. Katırcıoğlu, G. (2019), *Denim Kumaşlarının Görüntü İşleme ile İncelenip Kumaş Hava Geçirgenliğinin Yapay Sinir Ağı İle Tahmin Edilmesi*. Yüksek Lisans Tezi, Erciyes Üniversitesi Fen Bilimleri Enstitüsü, Kayseri, 15-17.
17. Yurtsever, A. (2018), *Bölütlenmiş Histopatolojik Görüntüler Üzerinde Derin Öğrenme Yöntemiyle Kolon Kanseri Tespiti*, Yüksek Lisans Tezi. Sakarya Üniversitesi Fen Bilimleri Enstitüsü, Sakarya, 62.
18. Ceylan, A. (2018), *Dört Rotorlu Hava Aracı ile Görme Tabanlı Nesne Takibi*. Doktora Tezi, Fırat Üniversitesi Fen Bilimleri Enstitüsü, Elazığ, 48.
19. Dinçer, S. (2009), *Dijital Görüntü İşleme Teknikleri ile Matlab ve Mikrokontrolör Kullanılarak İki Boyutlu Görüntünün Çizdirilmesi*. Yüksek Lisans Tezi, İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü, İstanbul, 42.
20. Asmaz, K. (2006), *Görüntü İşleme İle İki Boyutlu Cisimlerden Grafik Modeller İçin Veri Elde Edilmesi*. Yüksek Lisans Tezi, Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü, İstanbul, 35-36.
21. Aydoğan, D. (2008). Hough Dönüşüm Yöntemi İle Gravite Anomali Haritalarından Çizgiselliklerin Saptanması: Orta Anadolu Uygulaması. *E-Journal of New World Sciences Academy*, 3(4), 644-647.
22. İnternet: Çayıroğlu, İ. (2022). Görüntü İşleme. URL: http://www.ibrahimcayiroglu.com/Dokumanlar/GoruntuIsleme/Goruntu_Isleme_Ders_Notlari-3.Hafta.pdf, Son Erişim Tarihi: 01.11.2022
23. Halıcı, M. (2019). *Kaskad Nesne Algılama ve Yapay Sinir Ağları*. Yüksek Lisans Tezi, Fırat Üniversitesi Fen Bilimleri Enstitüsü, Elazığ, 29-37.
24. Viola, P., Jones, M. (2001). *Rapid Object Detection using a Boosted Cascade of Simple*. Accepted Conference on Computer Vision and Pattern Recognition, 1-9.
25. Sağın, C. (2019). *Koruyucu Baret Kullanım İhlallerinin Tespiti*. Yüksek Lisans Tezi, Gebze Teknik Üniversitesi Fen Bilimleri Enstitüsü, Gebze, 17-21.
26. Öztemel, E. (2016). *Yapay Sinir Ağları* (4. Baskı). İstanbul: Papatya Yayıncılık, 45-114.
27. Kargı, V. S. A. (2015). *Yapay Sinir Ağ Modelleri ve Bir Tekstil Firmasında Uygulama*. Bursa: Ekin Basım Yayın Dağıtım, 23-102.
28. Duman, N. (2006). *Yapay Sinir Ağları ve Bir Uygulama*. Yüksek Lisans Tezi, Cumhuriyet Üniversitesi Sosyal Bilimler Enstitüsü, Sivas, 28.

29. Gül, E. (2020). *Ağır Vasıta Hava Kompresörü Arızalarının Makine Öğrenmesi Yöntemleri Kullanılarak Analizi*, Yüksek Lisans Tezi. Konya Teknik Üniversitesi, Konya, 22-23.
30. Karataş, G. (2020). *Derin Öğrenme Tabanlı Saldırı Tespiti*. Doktora Tezi, Marmara Üniversitesi Fen Bilimleri Enstitüsü, İstanbul, 24-25.
31. Korkusuz, R. (2019). *Futbola İlişkin Twitter Paylaşımlarının Duygu Analizi*. Yüksek Lisans Tezi, Trakya Üniversitesi Fen Bilimleri Enstitüsü, Edirne, 54.
32. Çetiner, M. (2020), *Makine Öğrenmesi Yöntemleri İle Yazılım Hata Tahmini*. Yüksek Lisans Tezi. İstanbul Kültür Üniversitesi Fen Bilimleri Enstitüsü, İstanbul, 9-10.
33. Yılmaz, İ. G. (2021), *Makine Öğrenmesi Teknikleri İle Havacılıkta Yakıt Taşımacılığı Kararının Alınması ve Bir Uygulama*. Doktora Tezi, İstanbul Üniversitesi Fen Bilimleri Enstitüsü, İstanbul, 53-55.
34. Altunkaynak, B. (2019). *Veri Madenciliği Yöntemleri ve R Uygulamaları*. Ankara: Seçkin Yayınevi, 35-38.
35. Lajunen, T., Parker, D., Summala, H. (2004). The Manchester Driver Behaviour Questionnaire: a cross - cultural study. *Accident Analysis and Prevention*, 36, 231-238.
36. Yüksel, İ. (2004). Trafik Kazasına Karışan Sürücülerin Öngören Değişkenlerin Belirlenmesi. *Öneri Dergisi*, 6(21), 227-233.
37. Kaçaroğlu, G., Amado, S., Akün, E. (2004). Hız İhlali Yapan Sürücülerin İhlale İlişkin Nedensel Atıflarının ve Kişilik Özelliklerinin İncelenmesi. *Türk Psikoloji Yazıları*, 7(13), 1-23.
38. Eboli, L., Mazzulla, G., Pungillo, G. (2017). The influence of physical and emotional factors on driving style of car drivers: A survey design. *Travel Behaviour and Society*, 7, 43-51.
39. Riyahi, P. (2014). *A Steady - State Visual Evoked Potential Brain Computer Interface System Evaluation as an In Vehicle Warning Device*. Master's Thesis, University of Tehran, Tehran, 99-136.
40. Kondyli, A. (2009). *Breakdown Probability Model At Freeway Ramp Merges Based On Driver Behavior*. Doctoral Thesis, University of Florida, Florida, 126-153.
41. Gupta, S. R. K. (2008). *Driver Performance on Curves Using a Driving Simulator*. Master's Thesis, Northeastern University Boston, Massachusetts, 64-67.
42. Karimi, D. (2008). *Driving Simulation to Study the Role of Different Sensory Cues in Operating an Agricultural Vehicle*. Doctoral Thesis, University of Manitoba, Manitoba, 44-46.

43. Golshani, N. (2015). *Analysis of Aggressive Driving Behavior: A Driving Simulation Study*. Master's Thesis, State University of New York, New York, 3-18.
44. Serrano, J., Stasi, L. L., Megias, A., Catena, A. (2013). Affective-sound Effects on Driving Behaviour. *Transport*, 29(1), 100-106.
45. Chakrabartya, N., Guptab, K. (2013). Analysis of Driver Behaviour and Crash Characteristics during Adverse Weather Conditions. *Procedia-Social and Behavioural Sciences*, 104, 1048 - 1057.
46. Chen, R., Kusano, K., Gabler, H. (2015). Driver Behavior During Overtaking Maneuvers from the 100-Car Naturalistic Driving Study. *Traffic Injury Prevention*, 16, 176-181.
47. Stravrinos, D., Mosley, P., Wittig, S., Johnson, H., Decker, J., Sisopiku, V., Welburn, S. (2016). Visual Behavior Differences in Drivers Across the Lifespan: A digital billboard simulator study. *Transportation Research*, 41, 19-28.
48. Shi, M. (2008). *Real Time Human Driver Identification for Improved Safety*. Doctoral Thesis, University of California, California, 15-24.
49. Harbluk, J., Noy, I., Trbovich, P., Eizenman, M., (2007). An On-road Assessment of Cognitive Distraction: Impacts on Drivers' Visual Behavior and Braking Performance, *Accident Analysis and Prevention*, 39, 372-379.
50. Thompson, J., Newman, S., Stevenson, M. (2015). A Model for Exploring The Relationship Between Payment Structures, Fatigue, Crash Risk, and Regulatory Response in A Heavy-Vehicle Transport System. *Transportation Research*, A(82), 204-215.
51. Demir, M., Çavuşoğlu, A. (2012). New Driver Behavior Model to Create Relastic Urban Traffic Environment. *Transportation Research*, F(15), 289-296.
52. Gueguen, N., Eyssartier, C., Meineri, S. (2016). A Pedestrian's Smile and Drivers' Behavior: When a Smile Increases. *Journal of Safety Research*, 56. 83-88.
53. Hyodo, S., Yoshii, T., Satoshi, S. (2016). An Analysis of The Impact of Driving Time on The Driver's Behavior Using Probe Car Data. *Transportation Research Procedia*, 21, 169-179.
54. Wu, Y., Boyle, L., Cheryl, M., Foley, J. (2015). Modeling Types of Pedal Applications Using a Driving Simulator. *Human Factors*, 57(7), 1276-1288.
55. Ryder, B., Gahr, B., Egolf, P., Dahlinger, A., Wortmann, F. (2017). Preventing Traffic Accidents With in-Vehicle Decision Support System - The Impact of Accident Hotspot Warnings on Driver Behaviour. *Decision Support Systems*, 99. 64-74.

56. Perez, A., Garcia, I., Nieto, M., Pedraza, J., Rodriguez, S., Zamorano, J. (2010). Argos: An Advanced In-Vehicle Data Recorder on a Massively Sensorized Vehicle for Car Driver Behavior Experimentation. *IEEE Transactions On Intelligent Transportataion System*, 11(2), 463-473.
57. Sato, T., Akamatsu, M. (2007). Influence of Traffic Conditions on Driver Behavior Before Making a Right Turn at an Intersection: Analysis of Driver Behavior Based on Measured Data on An Actal Road. *Transportation REsearch*, F(10), 397-413.
58. Abbas, M., Higgs, B., Medina, A., Yang, C. (2011). Discriminant Analysis of Driver Behavior. *Advances in Transportation Studies*, Special Issue, 77-86.
59. Hui-ling, M., Man, L. (2015). A Method of Driver Fatigue Detection Based on Multi-features. *International Journal of Signal Processing, Image Processin and Pattern Recognition*, 8(10), 107-114.
60. Hoekstra, L. (2015). *Driving Under Involuntary and Voluntary Distraction: Individual Differences and Effects on Driving Performance*. Master's Thesis, University of Toronto, Toronto, 21-38.
61. Yan, S., Bailing, Z., Smith, J. (2016). *Driver Behavior Recognition Based on Deep Convolutional Neural Networks*. 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery, 636-640.
62. Doğan, T. (2013). *Açık Kaynak Kodlu Bilgisayarlı Görü Kütüphanesi Kullanarak Kuş Bakışı Görüntü Dönüşümü*. Yüksek Lisans Tezi, Trakya Üniversitesi Fen Bilimleri Enstitüsü, Edirne, 52-56.
63. Satzoda, K. R., Trivedi, M. M. (2014). *Overtaking & Receding Vehicle Detection for Driver Assistance and Naturalistic Driving Studies*. 17th. IEEE Conference on Intelligent Transportation Systems, 697-702.
64. Martin, S., Vora, S., Yuen, K., Trivedi, M. M., (2018). Dynamics of Driver's Gaze: Explorations in Behavior Modeling & Maneuver Prediction. *IEEE Transactions on Intelligent Vehicles*, 3(2), 141-150.
65. Huang, X. (2012). *Driver Lane Change Intention Recognition by Using Entropy- Based Fusion Techniques and Support Vector Machine Learning Strategy*. Master's Thesis, Northeastern University Boston, Massachusetts, 22-37.
66. Salemi, M. (2015). *Authenticating Drivers Based On Driving Behavior*. Master's Thesis, The State University of New Jersey, New Jersey, 21-32.
67. Yüksel, A., Şerafettin, A. (2019). Sürücü Davranışlarının Makine Öğrenmesi Algoritmaları ile Sınıflandırılmasında Pencereleme Yönteminin Etkisi. *Teknik Bilimler Dergisi*, 9(2), 75-80.

68. Al-Din, N., Salim, M. (2020). Development of a Highway Driving Events Identification and Classification Using Smartphone. *International Journal of Nanoparticles*, 12(2), 152-173.
69. İnternet: Olabiyi, O., Martinson, E., Chintalapudi, V., Guo, R. (2019). Driver Action Prediction Using Deep (Bidirectional) Recurrent Neural Network. URL: <https://arxiv.org/abs/1706.02257>, Son Erişim Tarihi: 01.11.2022
70. Serttaş, N. T., Gerek, N. Ö., Hocaoglu, O. F., (2019). Gerçek Zamanlı Sürüş Verileri ile Sürücü Davranışlarının Belirlenmesi. *Afyon Kocatepe Üniversitesi Fen ve Mühendislik Bilimleri Dergisi*, 19, 676-681.
71. Peng, J., Guo, Y., Fu, R., Yuan, W., Wang, C. (2015). Multi-parameter Prediction of Drivers' Lane-Changin Behaviour With Neural Network Model. *Applied Ergonomics*, 50, 207-217
72. İnternet: Google Maps (2021), URL: <https://www.google.com/maps/@39.9028191,32.7184867,14558m/data=!3m1!1e3!5m1!1e1>, Son Erişim Tarihi: 01.11.2022



Gazili olmak ayrıcalıktır