



T.C.
KIRŞEHİR AHİ EVRAN ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
İLERİ TEKNOLOJİLER ANABİLİM DALI

İNSANSIZ HAVA ARACI İLE YOL GÜVENLİĞİ

EYÜP ENSAR YILDIZ

YÜKSEK LİSANS TEZİ

KIRŞEHİR / 2022



T.C.
KIRŞEHİR AHİ EVRAN ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
İLERİ TEKNOLOJİLER ANABİLİM DALI

İNSANSIZ HAVA ARACI İLE YOL GÜVENLİĞİ

EYÜP ENSAR YILDIZ

YÜKSEK LİSANS TEZİ

DANIŞMAN
Dr. Öğretim Üyesi Murat IŞIK

KIRŞEHİR / 2022

TEZ BİLDİRİMİ

Tez içindeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edilerek sunulduğunu, ayrıca tez yazım kurallarına uygun olarak hazırlanan bu çalışmada bana ait olmayan her türlü ifade bilginin kaynağına eksiksiz atıf yapıldığını bildiririm.

EYÜP ENSAR YILDIZ

20.04.2016 tarihli Resmi Gazete’de yayımlanan Lisansüstü Eğitim ve Öğretim Yönetmeliğinin 9/2 ve 22/2 maddeleri gereğince; Bu Lisansüstü teze, Kırşehir Ahi Evran Üniversitesi’nin aboneli olduğu intihal yazılım programı kullanılarak Fen Bilimleri Enstitüsü’nün belirlemiş olduğu ölçütlere uygun rapor alınmıştır.



ÖNSÖZ

Yüksek Lisans sürecinde danışmanlığımı yürüten, çalışmamın her aşamasında önerileri ile beni yönlendiren, tezimin başından sonuna kadar çok önemli katkılar yapan, kendisini tanıdığım günden bu yana kaliteli kişiliği, yardımsever ve sabırlı hali, mesleğindeki son derece başarılı ve özverili duruşuyla her zaman bana örnek olmasının yanı sıra bir bilim adamının nasıl çalışması gerektiğini kendisinden öğrendiğim değerli danışmanım Dr. Öğr. Üyesi Murat IŞIK'a büyük bir içtenlikle teşekkür ederim.

Tezimin oluşum sürecinde, ilerlemesinde, yazma aşamasında ve sonuçlanması süreçlerinde bana destek olan Doç. Dr. Mustafa YAĞCI ve Dr. Öğr. Üyesi Mikail KOÇ'a teşekkürlerimi içtenlikle sunarım.

Tezimi, ailem başta olmak üzere tüm sevdiklerime ve beni sevenlere ithaf ederim.

Haziran, 2022

EYÜP ENSAR YILDIZ

İÇİNDEKİLER

Sayfa No

ÖNSÖZ	iv
İÇİNDEKİLER	v
ŞEKİL LİSTESİ	vii
TABLO LİSTESİ	viii
SİMGE VE KISALTMA LİSTESİ	ix
ÖZET.....	x
ABSTRACT.....	xi
1. GİRİŞ	1
1.1. Amaç	3
1.2. Önem.....	3
1.3. Benzer Çalışmalar	4
1.4. Tez Organizasyonu.....	7
2. TEMEL KAVRAMLAR.....	8
2.1. Görme Kavramı.....	8
2.1.1. Görüntü	8
2.1.2. Renk Uzayları.....	9
2.1.3. Görüntü İşleme	10
2.1.4. Morfolojik Operatörler	12
2.1.5. Eşikleme	13
2.1.6. Görüntüler Üzerinden Akıllı Çıkarımlar	15
2.2. İnsansız Hava Aracı (Drone Teknolojisi).....	16
2.2.1. Kullanım Alanları.....	16
2.2.2. Çeşitleri	16
2.2.3. Kontrol Yöntemleri	17
2.2.4. Drone Seçimi.....	17
2.3. Nesne Tespiti.....	18
2.3.1. Nesne Tespiti Algoritmaları	19
3. MATERYAL VE YÖNTEM	20
3.1. OpenCV.....	20
3.2. Python	20
3.3. YOLO (You Only Look Once) Algoritması	22
3.3.1. YOLO Algoritması Çalışma Prensipleri.....	23
3.3.2. Bounding Box (Sınırlandırıcı - Çevreleyici Kutu) Tahmin İşlemi.....	24
3.3.3. Non – Maximal Suppression Algoritması	25
3.5. Plaka Okuma	26

3.5.1. Tesseract Sınıfı.....	26
3.6. Viola-Jones Algoritması ve Haar Cascade Sınıflandırıcı.....	26
3.6.1. Viola-Jones Algoritması ve Haar Cascade Alt Bileşenleri.....	27
3.6.2. Haar Benzeri Özellikler.....	27
3.6.3. Resim İntegrali.....	29
3.6.4. AdaBoost (Sınıflandırıcının Eğitilmesi).....	30
3.6.5. Basamaklandırma Yöntemi (Basamaklı Sınıflandırıcı).....	31
3.6.6. Viola-Jones ve Haar Cascade Sınıflandırıcı ile Yüz Tanıma İşlemi.....	32
4. MODEL TASARIMI VE UYGULAMASI.....	36
4.1. Veri Seti Elde Edilmesi.....	36
4.2. Yazılımın Geliştirilmesi.....	36
4.3. Yazılımın Test Edilmesi.....	38
5. TARTIŞMA VE SONUÇ.....	41
5.3. Öneriler ve Sonraki Çalışmalar.....	47
KAYNAKLAR.....	48
ÖZGEÇMİŞ.....	51

ŞEKİL LİSTESİ

	Sayfa No
Şekil 2.1. Sayısal görüntü işleme adımları [24].	10
Şekil 2.2. (a) Binary görüntü (b) Genişleme işlemi sonucunda oluşan görüntü [26].	13
Şekil 2.3. (a) Binary görüntü (b) Aşınma işlemi sonucunda oluşan görüntü[26].	13
Şekil 2.4. (a) Orijinal görüntü (b) 0.6 sınır eşik değeri sonucu oluşan görüntü[26].	14
Şekil 2.5. OpenCV eşikleme operatörleri kullanımı sonucu oluşan görüntüler	15
Şekil 2.6. Görüntü işleme ile diğer disiplinler arası ilişki	16
Şekil 3.1. YOLO ve Diğer Algoritmaların Karşılaştırılması[36].	23
Şekil 3.2. YOLO ile Nesne Tespiti[37].	24
Şekil 3.3. Bounding boxes konum ve boyut hesaplama[36].	25
Şekil 3.4. IoU Değerinin Hesaplanması[38].	25
Şekil 3.5. Hesaplanan (a) ve Kabul Edilen (b) Öznitelik Değerleri	28
Şekil 3.6. Kenar Özellikleri ve örnek resim üzerinde uygulanması	28
Şekil 3.7. Çizgi Özellikleri ve örnek resim üzerinde uygulanması	28
Şekil 3.8. Merkez - Çevre Özellikleri	29
Şekil 3.9. Resim integral hesabı	29
Şekil 3.10. Resim integrali hesaplama örneği (a) orijinal resim ve (b) resmin integrali	30
Şekil 3.11. Zayıf Sınıflandırıcıların minimize edilmesi örnek 1	30
Şekil 3.12. Zayıf Sınıflandırıcıların minimize edilmesi örnek 2	30
Şekil 3.13. AdaBoost Sınıflandırıcısı [43].	31
Şekil 3.14. Basamaklandırma Yöntemi[42].	32
Şekil 3.15. Yüz Tanıma İşlemi Akış Diyagramı [44].	33
Şekil 3.16. Adım adım yüz tespit işlemi	34
Şekil 3.17. Deer One Pro (Kullanılan İnsansız hava aracı)	34
Şekil 3.18. Vs Gps Pro (Kullanılan Mobil Arayüz)	35
Şekil 4.1. Kırşehir Ahi Evran Üniversitesi Merkez Kampüsü	36
Şekil 4.2. Yazılıma ait akış diyagramı	37
Şekil 4.3. Yaya geçidi dikey mesafe hesaplama	39
Şekil 5.1. Yaya geçidi, araç ve insan tespitleri- 1	41
Şekil 5.2. Yaya geçidi, araç ve insan tespitleri- 2	41
Şekil 5.3. Yaya geçidi, araç ve insan tespitleri- 3	42
Şekil 5.4. Yaya geçidi, araç ve insan tespitleri- 4	43

TABLO LİSTESİ

	Sayfa No
Tablo 3.1. Deer One Pro Özellikleri.....	35
Tablo 5.1. YOLO Başarım Sonuçları	43
Tablo 5.2. Geliştirilen Modelin Yaya Tespitinde Başarım Sonuçları.....	44
Tablo 5.3. Geliştirilen Modelin Araç Tespitinde Başarım Sonuçları	45
Tablo 5.4. Modelin Yaya Geçidini Kullanmayan Yayaları Tespit Etmekteki Karışıklık Matrisi ...	46
Tablo 5.5. Modelin Yaya Geçidine Araç Sayısını Tespit Etmekteki Karışıklık Matrisi	46
Tablo 5.6. Yaya Geçidini Kullanmayan Yayaların Yüzlerinin Tespit Edilmesindeki Başarım	47
Tablo 5.7. Yaya Geçidine Park Etmiş Araçların Plakalarının Tespit Edilmesindeki Başarım.....	47



SİMGE VE KISALTMA LİSTESİ

Simgeler	Açıklama
b_h	: Nesnenin yüksekliği
b_x	: Nesnenin orta noktasının x koordinatı
b_y	: Nesnenin orta noktasının y koordinatı
b_w	: Nesnenin genişliği
c_x, c_y	: Izgaranın resmin sol üst köşesine olan uzaklıkları
f	: Bir öznitelik
$H(x, f, p, \theta)$: Bir zayıf sınıflandırıcı
IoU	: Intersection Over Union
km	: Kilometre
m	: Metre
m/s	: Metre / Saniye
mAP	: Genel Ortalama Kesinlik Değeri
p	: Eşitsizlik işaretinin yönünü belirten polarite
p_w, p_h	: Önceden belirlenen kutunun genişliği ve yüksekliği
t_x, t_y, t_h, t_w	: Her bir kutucuk için ağ tarafından tahmin edilen Bounding box koordinatları ve boyutları
x	: İmgenin 80x32'lik bir penceresi
σ	: Sigmoid fonksiyonu temsilidir, t_x, t_y değerlerini 0-1 aralığına düşürür.
θ	: Bir eşik değeri
(x, y)	: Resmin koordinatları
Kısaltmalar	Açıklama
CMYK	: Cyan – Magenta – Yellow – Key (Yeşil ve mavi karışımı – Kırmızı ve morun karışımı – Sarı – Siyah olarak tanımlanır)
FPS	: Frame Per Second (Saniyedeki frame sayısı)
GHz	: Giga Hertz (İşlemci saat hızı)
GPU	: Graphics Processing Unit (Ekran kartı işlemcisi)
GPS	: Global Positioning System (Küresel Konumlama Sistemi)
Hd	: High Definition (Yüksek Çözünürlük)
HSV	: Hue – Saturation – Value (Rengin özü – Doygunluk – Değer)
HSL	: Hue – Saturation – Value (Rengin özü – Doygunluk – Parlaklık)
İHA	: İnsansız Hava Aracı
JVM	: Java Virtual Machine
M/s	: Metre / Saniye
Nm	: Nanometre
OCR	: Optical Character Recognition (Optik Karakter Tanıma)
RAM	: Random Access Memory (Rastgele erişimli bellek)
R-G-B / RGB	: Red – Green – Blue (Kırmızı – Yeşil – Mavi)
Sd	: Secure Digital (Güvenli Sayısal)
TR	: Türkiye
TÜİK	: Türkiye İstatistik Kurumu
UAV/UAS	: Unmanned Aerial Vehicle/Systems
YOLO	: You Only Look Once (Tek Bir Sefer Bak)

ÖZET

YÜKSEK LİSANS TEZİ

İNSANSIZ HAVA ARACI İLE YOL GÜVENLİĞİ

EYÜP ENSAR YILDIZ

Kırşehir Ahi Evran Üniversitesi

Fen Bilimleri Enstitüsü

İleri Teknolojiler Anabilim Dalı

Danışman: Dr. Öğr. Üyesi Murat IŞIK

Gündelik yaşamda yaya geçidine park eden araçlar ve yaya geçidini kullanmayan insanlar trafik akışını bozmakta ve yaşam kalitesini olumsuz etkilemekle beraber çeşitli trafik kazalarına sebep olabilmektedir. Bu çalışmanın amacı İnsansız Hava Aracı (İHA) kullanarak yaya geçidini kullanmayan yayaların ve yaya geçidi üzerine park eden araçların tespit edilmesinde ekonomik, hızlı ve başarılı bir yöntem geliştirmektir. İHA'nın belirlenen yol güzergâhında, sabit yükseklik ve açıyla kullanılmasıyla elde edilen video akışı kaydedilmiş ve geliştirilen bir model ile analiz edilmiştir. Geliştirilen model Ahi Evran Üniversitesi Merkez Yerleşkesi içerisinde test edilmiştir. Yaya geçidini kullanmayan kişilerin yüzlerinin tespitinde oldukça yüksek bir başarı elde edilmiştir. Park etmiş araçların plakalarının tespitinde ve ayrıca park yasağı olan yerlerde park eden araçlar ile duraklama yapan araçların ayırımında farklı sonuçlar elde edilmiştir. İHA'ların bu şekilde gündelik yaşamda kullanılmasıyla trafikteki aksaklık ve kazaların belirli bir ölçüde önüne geçilebileceği öngörülmektedir.

Haziran 2022, 64 Sayfa

Anahtar Kelimeler: insansız hava aracı, görüntü işleme, python, yolo, viola-jones

ABSTRACT

M.Sc. THESIS

ROAD SAFETY WITH UNMANNED AERIAL VEHICLE

EYÜP ENSAR YILDIZ

Kırsehir Ahi Evran University

Graduate School of Sciences and Engineering

Advanced Technologies Department

Supervisor: Assist. Prof. Murat IŞIK

In daily life, vehicles parked at the pedestrian crossing and people who do not use the pedestrian crossing disrupt the traffic flow, negatively affecting the quality of life, and cause various traffic accidents. The aim of this study is to develop an economical, fast and successful method for detecting pedestrians who do not use the pedestrian crossing and vehicles parked incorrectly by using an Unmanned Aerial Vehicle (UAV). The video stream obtained from the use of the UAV on the determined road route, with a fixed height and angle, was recorded and analyzed with a developed model. The developed model was tested in Ahi Evran University Central Campus. A very high success has been achieved in detecting the faces of people who do not use the pedestrian crossing. Different results have been obtained in detecting the license plates of parked vehicles and also in distinguishing between vehicles parked in places where parking is prohibited and vehicles that make a stop. It is predicted that by using UAVs in daily life in this way, traffic disruptions and accidents can be prevented to a certain extent.

June 2022, 64 pages

Keywords: unmanned aerial vehicle, image processing, python, YOLO, viola-jones

1. GİRİŞ

İçerisinde bulunduğumuz yirmi birinci yüzyılda teknolojik gelişmeler çok hızlı ilerlemekte ve insanlar için büyük kolaylık ve kazanımlar sağlamaktadır. Bu kazanımlar ciddi problemleri de beraberinde getirmektedir. Özellikle gelişmekte olan ülkelerde hızlı nüfus artışı, kentleşme ve kırsal kesimlerden büyük kentlere doğru tek yönlü göç yaşanmaktadır. Buna paralel olarak şehirlerde yaşayan insan sayısı, insanların ihtiyaçları ve talepleri sürekli artmaktadır. Bu artışlar kalabalık nüfusa ve çevre sorunlarına sebebiyet vermektedir. Böylece altyapı yetersizliği, çarpık kentleşme ve düzensiz yaşam alanları ortaya çıkmaktadır. Bu durum şehirlerin problemleri ile birlikte büyümesine sebep olmaktadır.

Teknolojik gelişmelerin sağladığı önemli kazanımlardan olan ulaşım ve otomotiv sektöründeki gelişmeler beraberinde önemli artışları getirmiştir. Gelişmekte olan ülkemiz örneği incelendiğinde ülkemizde bulunan araç ve araç kullanan kişi sayısı hızla artmakta ve araca olan talep de hızla artmaya devam etmektedir. Emniyet Genel Müdürlüğü 2020 yılı istatistikleri incelendiğinde son on yılda motorlu araç sayısında %59,71, sürücü sayısında ise %45 artış olduğu görülmüştür [1]–[4].

İçerisinde bulunduğumuz araç ve araç kullanan kişi sayısının artması durumu kazanımlarıyla birlikte birçok sorunu da beraberinde getirmektedir. Şehir içlerindeki ulaşım ve park problemi çok hızlı büyümekte ve çözümü her geçen gün zorlaşan bir sorun haline dönüşmektedir. Hızla artan araç sayısı, buna paralel olarak ortaya çıkan trafik yoğunluğuyla birlikte trafik kazalarını çözülmesi gereken önemli bir problem haline getirmiştir. Trafik kazaları, birey ve toplum yaşamı üzerindeki olumsuz etkileriyle günümüzde çok önemli bir sosyal problemdir [5]. Bu problemin altında yatan sebepleri incelediğimizde ise bazı insan davranışlarının buna sebebiyet verdiğini görmekteyiz. Bu davranışlar incelendiğinde iki farklı örnek davranış göze çarpmaktadır.

Birinci örnekte insanlar günlük işlerini yaparken araçlarını emniyetli ve park etmeye uygun alanlara park etmek isterler. Ancak düzenli park yerlerinin ihtiyacı gidermemesi sürücülerini yasak alanlara park etmeye zorlar. Taşıt yollarında yayaların güvenli bir şekilde karşıdan karşıya geçmelerini sağlamak üzere hazırlanmış ve trafik işaretleri ile belirlenmiş alan olan yaya geçitlerine zaman zaman sürücüler araçlarını park edebilmektedirler. Her ne kadar yasalar çerçevesinde yaya geçidine ve diğer yasak alanlara park etmek suç olsa ve cezası bulunsada, çoğunlukla cezalar uygulanmamakta veya bu cezalar caydırıcı olamamaktadır.

Park yasağı olan yerleri ve yaya geçitlerini güvenlik kameraları ile sürekli kontrol etmek mümkün olmamakta ve yapılan park hatalarının tespiti ve ispatı zorlaşmaktadır. Araçların başta yaya geçidi olmak üzere yasak alanlara park etmeleri trafik güvenliğini tehlikeye düşürmekte, trafikte bir düzensizliği sebep olabilmekte, hatta kazalara sebebiyet vermektedir.

İkinci örnek ise yayaların yaya geçidini kullanmadan karşıdan karşıya geçmesidir. Yaya geçitleri yerine göre ışıklı ya da çizgilerle belirlenmiş durumda olup her biri için farklı kurallar geçerlidir. Fakat her iki durumda da temel prensip trafik akışı içerisinde yayaların otoyolu kesen bir hatta güvenle karşıdan karşıya geçmelerini sağlamaktır. İnsanlar bazen keyfi olarak, bazen yaya geçidine kısmi uzaklıktan ötürü, bazen de farkında olmadan yaya geçidini kullanmayıp doğrudan karşıdan karşıya geçmeyi tercih etmektedirler. İnsanların yaya geçidi olduğu halde kullanmayıp rastgele bir yerden karşıya geçmeleri trafiğin aksamasına ya da trafik kazalarına neden olmaktadır.

Her iki davranışın sonuçları incelendiğinde yaya geçidi ve diğer yasak olan yerlere park eden araçlar ile yaya geçidini kullanmadan karşıya geçen insanların trafik akışını bozdukları ve çeşitli trafik kazalarına sebep oldukları bilinmektedir. 2017 yılından 2020'ye kadar TÜİK Karayolu Trafik Kaza İstatistikleri incelendiğinde; kaza oranlarının yaklaşık olarak yüzde %20 - %23 arasında yaya sebepli olduğu görülmektedir [1]–[4]. Buna ek olarak hatalı park etmeden kaynaklanan kazaların oranının sürekli olarak arttığı ile ilgili birçok haber mevcuttur [6], [7].

Ayrıca 2017 yılından 2020'ye kadar TÜİK Karayolu Trafik Kaza İstatistikleri incelendiğinde; “hatalı park” cezasının en çok kesilen cezalar arasında ilk üç sırada yer aldığı görülecektir [1]–[4]. “Hatalı Park” cezasının en çok uygulanan ilk üç ceza içerisinde yer alması ve kaza oranlarının hızla artması bu durumun çözüme kavuşturulması gerekliliğini ortaya çıkarmaktadır. Hatalı yapılan parklar hem trafik düzenini bozmakta hem de kazalara sebebiyet verebilmektedir.

Gerek yaya geçidine park eden araçlar gerekse de yaya geçidini kullanmadan karşıdan karşıya geçen insanlardan dolayı ortaya çıkan zararların giderilmesinde yeni çözümler arayışları bulunması fikri öne çıkmaktadır. Bu çözüm arayışında insansız hava aracı kullanımı akla gelen bir seçenektir. İnsansız hava araçlarının kullanım alanları gün geçtikçe artmaktadır. İlk zamanlar hobi ve askeri amaçlar için kullanılan insansız hava araçları daha

sonra kamera yardımıyla havadan görüntü çekimlerinde kullanılmaya başlanmış olup bu çalışma içinde kullanımı uygun görülmüştür.

Bu tez çalışmasında insansız hava aracı ile uygun bir muhitte içerisinde yaya geçidi, sabit ve hareket halinde insanlar, park halinde ve hareket halinde araçların olduğu bir yol üzerinde görüntü kaydı yapılmıştır. Bu görüntüler üzerinde görüntü işleme teknikleri kullanılarak bilgisayar ortamında geliştirilen model ile nesnelerin tespiti yapılmıştır. Geliştirilen model ile yaya geçidi kullanmayan insanlar ve yaya geçidine park eden araçlar tespit edilmiştir.

1.1. Amaç

Ahi Evran Üniversitesi Merkez Kampüsü içerisinde; öğrencilerin yaya geçitlerini kullanmadığı ve araçların yasak olmasına rağmen yaya geçidi üzerinde hatalı park ettikleri gözlemlenmiştir. Yaya güvenliğinin ve trafiğin olumsuz etkilendiği bu durumlar çalışmanın ana motivasyonunu oluşturmuştur.

Bu çalışmada, yaya geçidini kullanmayan yayaların ve bu geçide park eden araçların tespit edilmesi için insansız hava aracı ile kampüs içerisinde toplanan verileri girdi olarak kullanan bir model geliştirilmiştir. Trafik düzeninin ve yaya güvenliğinin sağlanmasına ekonomik, hızlı ve başarılı bir şekilde katkı sağlayarak insan yaşamının kalitesinin artırılması amaçlanmaktadır. Bu amaca yönelik geliştirilen modelin aşağıdaki hedeflere sahip olması amaçlanmaktadır;

- Hem kumanda ile hem de önceden belirlenen koordinatlar üzerinde otonom uçuş yapabilecek şekilde programlanabilir insansız hava aracı ile belirli bir yükseklikte verilerin elde edilmesi ve filtrelenmesi,
- Toplanan verilerin geliştirilen modelle analiziyle, yaya geçidi kullanmayanların yüzlerinin ve yaya geçidi üzerine park eden araçların plakalarının tespit edilmesi.

1.2. Önem

Her geçen gün artan trafik ve park problemleri hayatın akışını zorlaştırmakta ve insan yaşamının kalitesinin düşmesine sebep olmaktadır. Yaya geçidini kullanmayıp karşıdan karşıya geçen insanlar ve yaya geçidine park edilen araçlar trafik akışını olumsuz etkilemektedirler. Bu durumun önüne bir insansız hava aracı ile geçilebilir. İnsansız hava

aracı ile sabit kameraların ve trafik polislerinin olmadığı birçok noktaya kolaylıkla erişilebilir ve kaydedilen görüntüler ile yapılan yanlışlar tespit edilebilir. Rastgele zamanlarda otonom çalışabilen bir insansız hava aracı ile görüntü kaydı yapılması insanlar üzerinde caydırıcı etki yaparak yapabilecekleri hataları azaltabilir. Bu çalışma ile bir insansız hava aracı tarafından kaydedilen görüntülerin bilgisayar ortamında görüntü işleme algoritmaları kullanılarak görüntüler üzerinden insan, araç, yaya geçidi tespitleri yapılarak trafik ve yol güvenliği sağlanmasına katkı sunulabilecektir.

1.3. Benzer Çalışmalar

Peker ve Zengin [8] ortam güvenliği sağlama amacıyla gerçek zamanlı olarak hareket analizi, yüz belirleme ve yüz tanıma işlemlerini gerçekleştirmek için yaptıkları çalışmalarında bilgisayar kamerası ile ortama giren kişinin hareketini ve akabinde yüzünü tespit etmişlerdir. Çalışmalarında mekanik bir düzen, bilgisayar ve bilgisayar kamerası, C# programı ile hazırlanmış bir yazılım kullanmışlardır. Yaptıkları çalışmada ışık koşullarındaki yetersizliklerin ve resmin ön cepheden çekilme zorunluluğunun dezavantajları olduğunu belirtmişlerdir. Hazırladıkları yazılım ile %70 ile %95 arası başarı sağladıklarını belirtmişlerdir. Yüz tanıma işlemi olarak çalışmamıza benzer bir çalışma olup kullandıkları yazılımlar noktasında çalışmamızdan farklılık göstermektedirler.

Murat [9] yaptığı çalışmada insansız hava aracılığıyla havadan alınan görüntüler üzerinden nesne tespiti yapmıştır. Nesne tespitinde farklı evrimsel sinir ağı modelleri üzerinde performans denemeleri yapmıştır. Yaptığı çalışmada kullandığı modelleri (YOLOv5, YOLOv3, RetinaNet, EfficientDet) karşılaştırmış ve YOLOv5 modelinin en iyi seviyede ve YOLOv3 modelinin ikinci seviyede başarılı olduğunu tespit etmiştir. Bu çalışma, havadan alınan görüntülerin analizinde ve görüntüde bulunan nesnelerin tespitinde daha başarılı neticeler almak için YOLOv5 modelinin kullanımı noktasında çalışmamıza yol gösterici olmuştur.

Özalp ve Düğenci [10] hazırladıkları plaka tanıma sisteminde C# yazılım dili ve Java platformunu kullanmıştır. Yaptıkları çalışmada kalın, ince, birbirine yakın, TR logolu, TR logosuz, mavi logo yüzeyli, kırmızı logo yüzeyli gibi etkenlerde zaman zaman okumada başarının düştüğünü belirtmişlerdir. Yaptıkları çalışma plaka tespiti noktasında çalışmamıza benzerlik göstermekte olup kullandıkları yazılımlar noktasında ve çalışmalarının amaçları konusunda farklılıklar göstermektedir.

Kaplan ve diğ. [11] arkadaşları hazırladıkları araç plaka tanıma sisteminde sabit kamera ve bariyer önünden neredeyse durma hızında olan araçların yaklaşık 2-4 metreden plakalarını yaklaşık %99 oranda tespit etmişlerdir. Bu çalışmanın otopark ya da okul girişi gibi kontrol noktalarında kullanılması mümkünken hareketli araçlarda ve mesafenin arttığı durumlarda verimli olarak kullanılmasıyla ilgili bir çalışma yapılmamıştır.

Şanlıalp [12] yaptığı çalışmada trafik polislerinin trafik cezai işlemlerini mobilize şekilde yapmalarını sağlamak için iPhone akıllı telefonlarda kullanılabilir olan bir mobil yazılım geliştirmiştir. TR plakalı sivil araçların plakalarını okuyan bu yazılım uygulaması OpenCV görüntü işleme kütüphanesi ve Tesseract karakter tanıma sınıfı kullanmakta ve plakaları gündüz %88, gece ise %80 oranında başarı elde etmektedir. Uygulamanın dezavantajı olarak; yalnızca iPhone markalı cihazlarda kullanılabilmesi ve yalnızca yakın mesafeden üst düzey başarı elde etmesi gösterilebilir.

Şimşek ve diğ. [13] doğal ortamlarda bulunan canlıların hareketi ile tetiklenen fotokapanlar aracılığıyla elde edilen görüntüler üzerinden canlıların tespitini yapmışlardır. Yaptıkları çalışmada PASCAL VOC veri seti ile eğitilmiş YOLO mimarisi kullanmışlardır. Yaptıkları uygulama ile canlı tespitinde yüksek bir başarı elde etmişler ancak çeşitli doğal sebeplerden ötürü tespit edemedikleri canlı sayısı da azımsanmayacak düzeydedir. Yapılan çalışma YOLO modelinin eğitilmesi ve kullanılması açısından çalışmamıza benzerlik gösterse de çalışma amacı bakımından farklıdır.

Kaplan [14] yaptığı çalışmada Haar Cascade Sınıflandırıcısı kullanarak resim, video ve kamera üzerinde gerçek ve yarı zamanlı yüz tanıma işlemi gerçekleştirmiştir. Kaplan çalışmasında OpenCV Kütüphanesi ve C# programlama dilini kullanmıştır. Yaptığı çalışmada insan yüzü tespiti yaparken kullandığı kütüphane ve sınıflandırıcı ile çalışmamıza benzerlik göstermekte olup geliştirdiğimiz modelde Python programlama dili tercih etmemiz farklılık oluşturmaktadır.

İnce ve diğ. [15] yaptıkları uzun mesafeli hareketli görüntülerde yayaların biyometrik özelliklerine göre yaşlarının sınıflandırılması çalışmalarında OpenCV Kütüphanesi üzerinde Haar sınıflandırıcı ile insan tespiti yapmışlardır ve insanları yetişkin ve çocuk olarak kategorize etmişlerdir. Yetişkin tespitinde %74.7 ve çocuk tespitinde ise %68.1 başarı sağlamışlardır. Geliştirilen model sadece insan tespiti amaçlıdır ancak Haar sınıflandırıcısını kullanılma şekli ile çalışmamıza oldukça yakındır.

Ayvaz ve Gürüler [16] yaptıkları çalışmada öğrencilerin sınıf içinde yüz ifadelerinden duygusal durumlarının gerçek zamanlı olarak tespit edilmesini sağlayan model hazırlamışlardır. OpenCV kütüphanesi üzerinde Python programlama dili kullanarak ve Haar Cascade sınıflandırıcı ile yaptıkları çalışmada yakın mesafeden aldıkları yüz görüntüleri üzerinden duygusal durum çıkarımı yapmışlardır. Daha önceden hazırlanan veri setinde bulunan yüz işaretçilerine dayanan öznitelikler ile alınan görüntüler karşılaştırılmış ve öğrencilerin yüz ifadesinde duygusal durumu %97.15 oranında tespit edilmiştir. Bu başarı 2 metreden daha kısa mesafeden, uygun bakış açısı ve uygun ışık miktarı ile sağlanmıştır. Uygun ışık miktarı sağlanmayan durumlarda ve değişken mesafelerden alınan görüntüler üzerinde başarısı bilinmemektedir.

Karaimer ve Baştanlar [17] yaptıkları çalışmada tüm yönlü kameralar ile binek araç tespiti yapmışlardır. Çalışmalarında Haar-benzeri öznitelikler ve basamaklı hızlandırma kullanmışlardır. Yaptıkları çalışmada basamaklı hızlandırmanın tek bir sınıflandırıcıya göre kazandırdığı süreyi gözlemlemişlerdir. Ayrıca küresel kesitli ve silindirik panoramik imgelerin tespit başarı oranlarını kıyaslayarak küresel kesitli panoramalarda başarı oranının daha yüksek olduğunu tespit etmişlerdir. Çalışma sadece araçların tespit edilmesi içindir ayrıca araç içerisinde yüz taraması yapılmamıştır.

Zhao ve Chen [18] yaptıkları çalışmada geliştirilmiş YOLO modeline dayalı gerçek zamanlı yaya tespiti yapmışlardır. Kendi modellerinde yaptıkları iyileştirmeler ile yaya tespit oranlarını YOLOv3 modeli kullanarak artırmışlardır. Çalışma sadece yaya tespiti içindir ve çalışmamıza YOLO algoritmasının kullanımı konusunda benzerlik göstermektedir.

Karaduman [19] yaptığı çalışmada başlangıçta insan kontrolü altında, devamında ise otonom olarak belirli bir uzaklığa giden ve geri gelen insansız hava aracı kullanmıştır. İnsansız hava aracı görüntü işleme teknikleri sayesinde istenen hedefe otonom kontrol ile giderek ve rota belirleyip bu rotayı takip ederek başlangıç noktasına geri gelmiştir. İnsansız hava aracından alınan görüntülerin işlenerek yolun ve rotanın tespit edildiği bu çalışma kullandığı materyaller ve yöntemler olarak farklılık gösterse de insansız hava aracının otonom olarak yol kontrolü yapabilmesi yönüyle çalışmamızla benzerlik göstermektedir.

Kutlu [20] yaptığı çalışmada insansız hava aracı ile elde edilen görüntüleri derin öğrenme yöntemleri ile analiz etmiştir. Yaptığı deneysel çalışmada insansız hava aracı ile yaklaşık 100 metre ve daha üzeri yükseklikten kaydettiği 240 adet görüntü üzerinde evler ve arabalar başta olmak üzere 7541 adet nesne etiketlemesi yapmıştır. Kullandığı materyal ve yöntemler

farklılık arz etse de temeli nesne tespitine dayalı olan bu çalışma, çalışmamızla ortak temele dayanmaktadır.

Literatürde İnsansız Hava Aracı'ndan elde edilen görüntülerle güvenlik amacıyla yapılmış birçok çalışma mevcuttur. Bu çalışmaların ortak yanlarından birisi insansız hava araçlarının yol güvenliği konusunda kullanılabilirliğinin ve başarısının oldukça yüksek olması gösterilebilir.

1.4. Tez Organizasyonu

Tez çalışması 5 bölümden oluşmakta olup, bu bölümlere ait içeriklerden aşağıda kısaca bahsedilmiştir.

Bölüm 1. Giriş: Problemin tanımı, yapılan çalışmanın amacı, yapılan tez çalışmasını diğerlerinden farklı kılan yönler, bilime katkısı ve tez organizasyonu hakkında bilgi verilmektedir

Bölüm 2. Temel Kavramlar: Görüntü, renkli görme kavramı, renk uzayları, görüntü işleme, morfolojik operatörler, eşikleme, görüntüler üzerinden akıllı çıkarımlar, insansız hava araçları, İHA kullanım alanları, İHA çeşitleri, İHA kullanım yöntemleri Hakkında bilgiler verilmiştir.

Bölüm 3. Materyal ve Yöntem: Bu çalışma için geliştirilen modelde kullanılan paketler, Algoritmalar ve Platformlar hakkında bilgi verilmiştir. YOLO algoritması, OpenCV paketi, Tesseract ve Haar Cascade sınıflandırıcısının nasıl kullanıldıkları ve avantaj/dezavantajları ile ilgili bilgiler verilmiştir.

Bölüm 4. Model Tasarımı ve Uygulaması: Veri toplama, adım adım modelin geliştirilmesi süreci ve nasıl çalıştığı bilgisi bu bölümde verilmiştir.

Bölüm 5. Tartışma ve Sonuç: Yapılan çalışma sonucu elde edilen bulgular, çalışmanın sonucu, sonuca göre ortaya çıkan tartışma ve öneriler bu kısımda verilmiştir.

2. TEMEL KAVRAMLAR

2.1. Görme Kavramı

Göz, dış ortamdan maksimum seviyede bilgi alınmasını sağlayan, dış ortamdaki varlıkların görüntüsünü inceleyen ve akabinde bu görüntüleri uygun sinyallere dönüştürerek beyne gönderen duyu organıdır. Görme olayının gerçekleşmesi için ilk önce göze ışık düşmesi gereklidir. Ardından gelen ışık retina üzerinde bulunan fotoreseptörler aracılığıyla sinirsel sinyallere dönüştürülürler. Bu sinyaller optik sinirler vasıtasıyla karmaşık işlemlerin ve değerlendirmelerin yapılacağı beyne iletilir [21]. Bu aşamada beynin daha önce yaşamış olduğu tecrübe ve bilgiler ile birlikte psikolojik durum ve etiksel kazanımlar gibi daha birçok etken devreye girer ve görme gerçekleşir [22]. Tıpkı bir fotoğraf makinası misali gözlerde başlayan görme sistemi beynin görme işlemini yapan bölümlerinde tamamlanır ve bu durum aslında bir görüntü işleme sistemi olarak kabul edilebilir.

Renk, ışığın değişik dalga boylarının gözün retinasına ulaşması ile ortaya çıkan bir algılama olarak tanımlanmaktadır. Işık maddeler üzerine düştüğünde bir kısmı soğurulup bir kısmı ise yansıtılır ve buradaki çeşitlilikten dolayı renk tonları ve renkler oluşur. Madde renklerin tamamını yansıtırsa göz bunu beyaz olarak, hiç ışık yansıtmazsa ise siyah olarak algılar. Aradaki dalgalarda ise diğer tüm renkler oluşur. İnsan gözünün algılayabileceği dalga boyu 380 nm ile 780 nm arasında olup bu aralığın dışındaki ışığı insan gözü algılayamaz. İnsan gözünde 3 çeşit renk algılayıcısı vardır. Bu algılayıcılar kırmızı, yeşil ve mavi renkleri algılar. Doğada bulunan tüm renkler aslında red (kırmızı), green (yeşil) ve blue (mavi) renklerin farklı oranlarda karışmasıyla meydana gelmektedir. Bu durumda renkli görme olayının tamamen beyinde gerçekleşen bir algıdan ibaret olduğu söylenebilir.

2.1.1. Görüntü

Kameraların insan hayatına girmesiyle içinde bulunduğumuz anı ölümsüzleştirmek adına telefon, fotoğraf makinası kamera ya da içerisinde görüntü alma özelliği olan herhangi bir cihazla anında fotoğraf ya da video kaydı alınabilmektedir. Bu görüntüler bilgisayar ortamında dosyalanabilmekte, çok uzun yıllar saklanabilmekte ve istenildiği zaman gün yüzüne çıkarılabilmektedir. Kameralar vasıtasıyla aldığımız görüntüler yalnızca saklanmak ve izlenmek amacıyla kullanılmamakta olup bu görüntüler gelişen teknoloji imkânları ve görüntü işleme yöntemleri sayesinde bilimsel alanlarda ve diğer disiplinler arası mühendislik alanlarında faal bir şekilde kullanılmaktadır.

Görüntü, sürekli gibi görünse de aslında insan gözünün görebileceği en küçük parçalar olan piksellerden oluşur. İki boyutlu piksel dizisi bir araya gelip bir görüntüyü oluşturur. Görüntüyü piksellerin matrisi şeklinde de ifade etmek mümkündür. Görüntüler pikseller üzerinde kırmızı, yeşil ve mavi renklerin farklı oranlarda bir araya gelmesiyle oluşmakta ve daha sonra bilgisayarların ve diğer makinaların anlayabileceği şekilde ikilik sayma sisteminde bulunan '0' ve '1' sayıları ile ifade edilmektedirler.

2.1.2. Renk Uzayları

Renkler ışığın dalga boylarına göre tepki vermeleri sonucu kırmızı, yeşil ve mavi renklerin belirli oranlarda karışımından oluşmaktadır. Bu renklerin karışımında sonra çok fazla alternatif renk oluşacağından dolayı renkleri gruplandırma ihtiyacı doğmuştur. Böylece renk uzayı kavramı ortaya çıkmıştır. Renk uzayı renklerin koordinat düzleminde bir nokta olarak temsil edildiği bir renk organizasyonu olarak tanımlanabilir. Renkleri ölçmek, gruplandırmak ve farklı amaçlar gerçekleştirmek adına farklı kişi ya da firmalar tarafından geliştirilmiş olan çok sayıda renk uzayı kendine has bir yapıda literatürde yerini almıştır. Yani renk uzayları için "resimlerin ve görüntülerin dijital bir ifadesidir" demek doğru olacaktır [22].

RGB Renk Uzayı: Bir rengi belirlemek için birbirinden bağımsız üç değişkene gerek vardır. Bu yüzden 3 farklı değişkenden oluşan bir matris sayesinde renkler matematiksel olarak ifade edilir [23].

En sık kullanılan renk uzaylarının başında Red (kırmızı), Green (yeşil) ve Blue (mavi) renklerinin bir küp olarak tanımlandığı ve 3 boyutlu bir dizi olarak kırmızı, yeşil ve mavinin belirli oranlarda karıştırılıp elde edilen renk kodunun bu dizide tutulduğu RGB renk uzayı gelmektedir.

CMYK Renk Uzayı: Cyan (C) (yeşil ve mavinin karışımı), Magenta (M) (kırmızı ve morun karışımı), Yellow (Y) (sarı), Key (K) (siyah olarak tanımlanır) renklerinin karışımı ile yapılan CMYK renk uzayıdır.

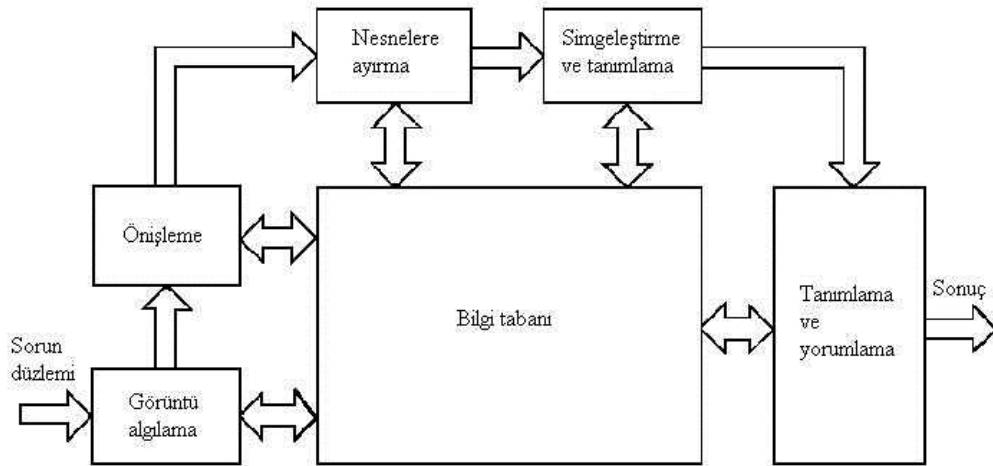
HSV Renk Uzayı: Hue (H), Saturation (S), Value (V), Türkçesi ile rengin özü, doygunluğu ve değeri şeklinde tanımlanan HSV renk uzayı sık kullanılan renk uzaylarının başında gelir.

HSL Renk Uzayı: Hue (H), Saturation (S), Lightness (L), Türkçesi ile rengin özü, doygunluğu ve açıklığı (parlaklığı) şeklinde tanımlanan HSL renk uzayı yine sık kullanılan renk uzaylarının başında gelir.

Literatürde yer edinmiş birçok renk uzayı bulunmaktadır. Bunlardan en çok kullanılan renk uzayı RGB olmasına rağmen, bu renk uzayının en büyük dezavantajı bir rengin değişmesi için 3 parametrenin birden değişmesi gerekmektedir. Yani en basit şekliyle bir rengi koyudan açığa doğru değişmesi için bile 3 parametre değiştirilmelidir. 3 parametrenin aynı anda kontrol edilmesi hem işlem yükünü hem de işlem karmaşıklığını artırmaktadır. Ancak HSL veya HSV renk uzaylarında ise sadece bir parametrenin değiştirilmesi ile rengin değişmesi veya açıklık/koyuluk ayarlaması mümkündür.

2.1.3. Görüntü İşleme

Görüntü işleme, pikseller üzerinde gerçekleştirilen matematiksel işlemler sayesinde görüntülerin düzeltilmesi veya görüntülerden anlamlı ifadeler çıkarılmak üzere bilgisayarlarda işlenmesi sürecidir. Şekil 2.1’de gösterildiği üzere bu süreç elde edilen görüntülerin veya videoların sayısallaştırılarak bilgisayar ortamına aktarılması ve istenen amaca göre geliştirilmiş bir yazılımla işlenmesidir.



Şekil 2.1. Sayısal görüntü işleme adımları [24].

Görüntü işleme işlemleri temel olarak üç adımda gerçekleştirilmekte ve incelenmektedir.

- Gerekli ekipman ile görüntü imajının aktarılması
- Görüntünün analiz edilmesi ve istenilen amaç doğrultusunda işlenmesi
- Analiz edilen verilere ait raporun, çıktının ya da sonucun alınması

Görüntü İşleme Yöntemleri

Görüntü işleme sisteminde analog görüntü işleme ve dijital görüntü işleme olmak üzere iki yöntem kullanılmaktadır. Her iki yöntemde de görüntü işleme için verinin geçmesi gereken temel aşamalar bulunmaktadır. Bu aşamalar sırasıyla aşağıda listelenmiştir.

- Ön işleme
- Geliştirme ve görüntüleme
- Bilgi çıkarımı

Bu aşamaların uygulanmasının ardından ilgili veriden ihtiyaç ve hedefler doğrultusunda sonuçlar elde edilebilmektedir.

Görüntü İşleme Teknikleri Kullanılarak Yapılabilecek İşlemler

Görüntü işleme teknikleri birçok işlem yapılabilmektedir. Görüntü işleme, kaydedilen görüntülerin işlenmesinden sonra devamında bulanık mantık, yapay zekâ veya yapay sinir ağları gibi pek çok algoritma ile değerlendirilen bir teknoloji haline gelmiştir. Çağımızda birçok işlemde çeşitli sebeplerden ötürü insan yerine otonom sistemlerin tercih edilmesi, görüntü işlemenin önemini daha da artırmaktadır. Görüntü işleme teknikleri ile en başta aşağıda verilen maddeler olmak üzere birçok işlem yapılabilmektedir. Bunlar;

- Bir görüntüde bulunan ve insan tarafından algılanamayan ya da algılanmakta zorlanılan nesnelerin tespit edilmesi,
- Görüntülerin veya videoların gürültülerinden arındırılarak temizlenmesi
- Görüntü kalitesinin artırılması,
- Nesne takibi,
- Nesne tanıma.

Teknolojinin gelişmesi ve görüntü işleme tekniklerinin artmasıyla her geçen gün yapılabilecek işlemlerin sayısı artmaktadır.

Görüntü İşleme Platformları

Görüntü işleme için birçok platform ve paket kullanılabilir. Örneğin Matlab, Halcon, Fiji, Cimg, C++, Java ve Python bu platformlara; OpenCV, SimpleCV ve BoofCV paketleri ise paketlere örnek olarak verilebilir. Her platformun kendine özgü avantaj ve dezavantajları vardır. Zaman içerisinde yapılan çalışmalar, ilerlemeler, güncellemeler bu platform ve

paketler için deęişim ve yenileşme sağlayabilmektedir. Platform ve paket tercihi yaparken tercih edilecek başlıca avantajlar:

- Platformun faal olması,
- İyi performans göstermesi,
- Ücretsiz olması, hızlı çalışması,
- İçerisinde bulunan hazır fonksiyonlar,
- Açık kaynak kodlu olması,
- Yeterince doküman ve kaynak bulunması, farklı dillere destek vermesi,
- Kod satırının basit ve sade olması

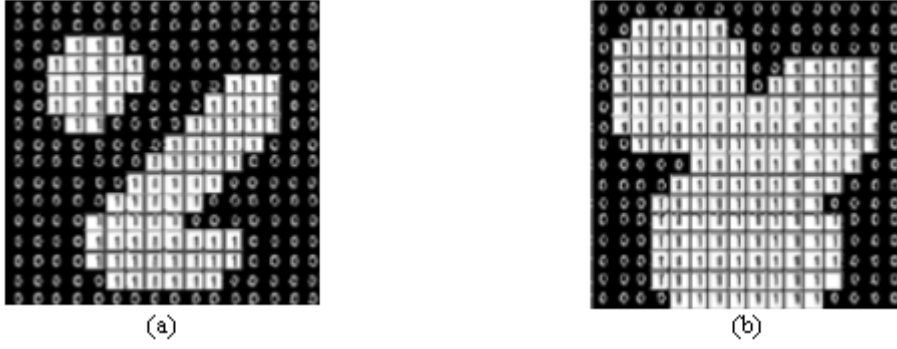
gibi durumlar varsayılır.

2.1.4. Morfolojik Operatörler

Morfoloji, biyolojide canlıların şekillerini ve yapılarını inceleyen, matematikte ise bir görüntüde bulunan ana öğeleri muhafaza edip hatalı kısımları ortadan kaldırmak veya belirli özelliklerini ortaya çıkarmak için kullanılan yöntemler olarak ifade edilebilir.

Morfoloji bilimi, üzerinde işlem yapılan görüntü üzerindeki şekillerin tanımlanması, analizi, yorumlanması, bölümlenmesi, köşelerin tespit edilmesi, genişletilmesi, daraltılması, gürültü giderilmesi gibi birtakım işlemleri yerine getirerek nesnelere ayırt etme veya gruplandırma yapmaya imkân verir. Görüntü işlemede morfolojik operatörler en temelde aşındırma (erosion) ve genişletme (dilation) işlemlerinden türetilmiştir. Aşındırma ve genişletme, kullanılan yapı elemanını görüntü üzerindeki çeşitli noktalarda yer değiştirerek yapı elemanı ile görüntünün koordinatlarının kesişme noktalarını inceleme şeklinde çalışır [25]. Aşındırma ve genişletme işlemleri ihtiyaca göre birbirinin tersi işlemleri yaparak görüntüyü hedeflenen hale getirmede aracılık ederler.

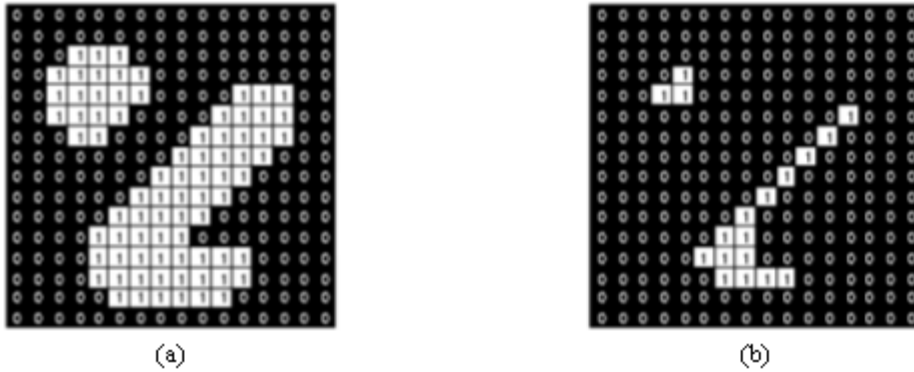
Genişleme (dilation) operatörü: Nesneyi kalınlaştıran, büyüten ve alanını artıran operatördür. Ayrıca kullanılan ince bir gürültü ile nesnenin bölünmesi ve iki farklı nesne gibi görünmesinin önüne geçmek için bu operatörden faydalanılır. Genişleme operatörü Şekil 2.2’de verilen örnekte olduğu gibi çalışmaktadır.



Şekil 2.2. (a) Binary görüntü (b) Genişleme işlemi sonucunda oluşan görüntü [26].

Verilen örnek incelendiğinde binary görüntüye genişleme işlemi uygulanmış olduğu görülmektedir. Genişleme işlemi sonucunda binary görüntüde bulunan beyaz piksel grupları büyümüş, birbirleri arası mesafe azalmış ve birbirlerine yaklaşmışlardır. Bu işlem bir nevi görüntüyü gürültülü hale getirme işlemidir.

Aşındırma (erosion) operatörü: Nesneyi incelten, küçülten ve alanını azaltan operatördür. Ayrıca kullanılan ince bir gürültü ile birbirine bağlı nesnelere ayırmak için bu operatörden faydalanılır. Aşındırma operatörü Şekil 2.3’de verilen örnekte olduğu gibi çalışmaktadır.



Şekil 2.3. (a) Binary görüntü (b) Aşınma işlemi sonucunda oluşan görüntü[26].

Verilen örnek incelendiğinde binary görüntüye aşındırma işlemi uygulanmış olduğu görülmektedir. Aşındırma işlemi sonucunda binary görüntüde bulunan beyaz piksel grupları küçülmüş, birbirleri arası mesafe artmış ve birbirlerinden uzaklaşmışlardır. Bu işlem bir nevi görüntüyü gürültülerden temizlenme işlemidir.

2.1.5. Eşikleme

Eşik kelimesi günlük hayatta sıkça kullanılan alt sınır ya da üst sınırdan sınırlamak anlamında kullanılan bir kelimedir. Dijital görüntü işlemede sıkça kullanılan eşikleme ifadesi eşik değere benzer anlamda kullanılır. Bir görüntüdeki pikseller hakkında nihai karar

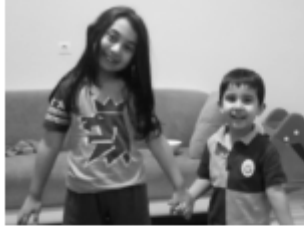
vermek ya da istediğimiz değerdeki pikselleri korumak için bu pikselleri bir değer altında veya üzerinde bir dereceye kadar reddetmek istenebilir [27]. Burada asıl hedeflenen üzerinde çalışılan görüntüde bulunan nesnelere arka plandan ayırma işlemidir. Çeşitli işlemlerde kullanılan eşikleme yöntemleri vardır.

Sık kullanılan eşikleme yöntemlerinden biri olan threshold binary yöntemi Şekil 2.4'te bulunan orijinal görüntü üzerine uygulandığında giriş görüntüsü içerisindeki her bir pikselin değeri önceden belirlenen bir eşik değeri ile kıyaslanır. Pikselin değeri eşik değerinin altında ya da üzerinde ise bu piksellere farklı değerler atanabilir. Herhangi bir görüntüye gri tonlama eklendikten sonra eşik değerinin altında olan pikseller siyah, eşik değerinin üstünde olan pikseller ise beyaz olarak atanır. Burada piksel bir nesne gibi düşünülüp, 1 ise nesne 0 ise arka plan gibi değerlendirilir. Şekil 2.4'ten de anlaşılacağı üzere renkli bir görüntüden binary (ikili) yani siyah beyaz görüntü elde edilmiş olur. Bu sayede eşikleme yöntemi görüntü üzerinde bulunan nesne tespiti veya farklı amaçla kullanılabilir.



Şekil 2.4. (a) Orijinal görüntü (b) 0.6 sınır eşik değeri sonucu oluşan görüntü[26].

Siyah beyaz görüntü elde edildikten sonra amaca OpenCV kütüphanesinde farklı threshold yöntemleri mevcuttur. Bunlar; THRESH_BINARY, THRESH_BINARY_INV, THRESH_TRUNC, THRESH_TOZERO ve THRESH_TOZERO_INV olmak üzere sıralanabilir [28]. Her bir threshold operatörü uygulandığı görüntünün piksel değerlerinde farklı işlemler yapar. Orijinal resim üzerinde ayrı ayrı OpenCV threshold operatörleri kullanılmış ve her bir operatörün yaptığı değişiklik Şekil 2.5'te gösterilmiştir.



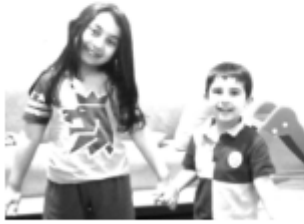
Orijinal resim



THRESH_BINARY



THRESH_BINARY_INV



THRESH_TRUNC



THRESH_TOZERO

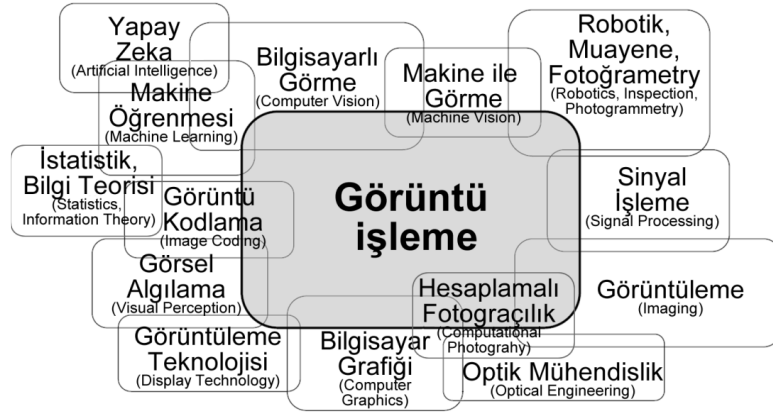


THRESH_TOZERO_INV

Şekil 2.5. OpenCV eşikleme operatörleri kullanımı sonucu oluşan görüntüler

2.1.6. Görüntüler Üzerinden Akıllı Çıkarımlar

Günümüzde teknolojinin geldiği noktada görüntü işleme yöntem ve teknikleri çok yaygın bir şekilde kullanılmaktadır. Görüntüler işleme sayesinde teknoloji kullanılarak görüntüler üzerindeki nesnelere hakkında bilgi çıkarımı oldukça ilerlemiştir. Her geçen gün yeni yöntemler keşfedilmekte ve mevcut yöntemler daha ileriye taşınarak ilerleme kaydedilmektedir. Görüntüler üzerinden bilgi okuma, plaka okuma, sinyal tespit etme, yazı karakterlerini metne çevirme, barkod okuma gibi birçok uygulama hali hazırda kullanılmakta ve her geçen gün yenileri piyasaya sürülmektedir. Mevcut görüntüler işlenerek ya da iyileştirilerek ortaya anlamlı bilgiler üretilmekte ve akıllı çıkarımlar yapılabilmektedir. Örneğin görüntülerin yorumlanmasıyla çevresel değişiklikler incelenebilmekte, şehirler planlanabilmekte, mekânlar dekore edilebilmekte ve daha sayısız uygulama yapılabilmektedir. Şekil 2.6 Şekil 2.6 da anlaşılacağı üzere günümüzde artık görüntü işleme disiplinleri arası bir çalışma alanı haline gelmiştir.



Şekil 2.6. Görüntü işleme ile diğer disiplinler arası ilişki

2.2. İnsansız Hava Aracı (Drone Teknolojisi)

İHA'lar uluslararası literatürde "UAV/UAS (Unmanned Vehicle/Systems)" olarak tanımlanan içerisinde insan olmayan, yerden bir pilot yardımı ile uzaktan kontrol edilen veya önceden belirlenen rota üzerinde otonom uçuş yapabilen bir uçan araç çeşididir [29]. Belirli yüksekliğe çıkabilen ve belirli bir alan ulaşabilen bu insansız hava araçları ilk olarak askeri maksatlarla üretilmişlerdir. Günümüzde ise teknolojik gelişmeler ile daha küçük boyutlarda ve daha ekonomik fiyatlarda üretilmekte ve drone ismiyle insanlığın hizmetine sunulmaktadır.

2.2.1. Kullanım Alanları

İnsansız hava araçları yapabildikleri işlere göre oldukça ekonomik çözümler sunabilmekte ve hayatın birçok alanın da hizmet etmektedirler. İnsansız hava araçları eğlence, görüntü alma, nesne taşıma, tarım, sağlık sektöründe, kargo hizmetlerinde, fotoğrafçılık, film ve sinema sektöründe, haritacılıkta, lojistik, ulaşım, güvenlik, trafik, askeri ve sivil birçok amaçla kullanılmakta ve popülerlikleri de her geçen gün artmaktadır.

2.2.2. Çeşitleri

İnsansız hava araçları teknik yapıları bakımından incelendiğinde kullanılacağı alanlara göre farklı kanat yapısı veya farklı sayıda motora sahip olacak şekilde tasarlanmakta ve üretilmektedirler. İnsansız hava araçları motor sayılarına göre de sınıflandırmalara sahiptir. Örneğin QuadCopter, bağımsız dört adet farklı motorla çok hızlı hareket kabiliyeti olan bir insansız hava aracı türüdür. Drone, doğru akım ile çalışan motoru, elektronik hız kontrol

ünitesi, mikro denetleyici, denge kontrol ünitesi, sensörler ve güç ünitesi ile birlikte toplam 6 kısımdan oluşur.

Drone, insansız hava çeşitlerinden en çok bilinen modelidir. Drone, üzerinde bulunduğu motor sayısına, ağırlığına, kanat yapısına, kullanım amacına, kontrol edilme şekline ve kullanıldığı alana göre alt kategorilere ayrılmaktadır.

2.2.3. Kontrol Yöntemleri

İnsansız hava araçlarını diğer hava araçlarından ayıran en önemli özelliklerinden biri kontrol yöntemidir. İhtiyaca göre insansız hava aracının nasıl kontrol edilmesi gerektiğine karar verilebilir.

Günümüzde insansız hava araçları yerde bulunan bir pilot tarafından gönderilerek uzaktan kumanda ile kontrol edilen ya da önceden oluşturulmuş uçuş programı yüklenerek belirlenen rotasında otomatik olarak uçurulan araçlardır [30].

Kullanılan kontrol sisteminin çeşidine göre: Doğrudan kontrol, gözlenen kontrol, uyumsuz - otonom kontrol ve uyumlu - otonom kontrol olarak alt başlıklara ayrılır [31].

Doğrudan Kontrol: Yerde bulunan bir merkezden belirli bir uzaklığa kadar bir operatör tarafından yapılan insansız hava aracı kontrol yöntemidir.

Gözlenen Kontrol: İnsansız hava aracı önceden programlanır ve bağımsız hareket eder ancak operatörün istediği şekilde gözlemlenebilir hareketler sergileyebildiği kontrol yöntemidir.

Uyumsuz – Otonom Kontrol: İnsansız hava aracı önceden programlanır ve bu programın dışına çıkmadan uçuşunu yapar. Herhangi bir dış etken varsa insansız hava aracı bu duruma uyum sağlayamaz ve olumsuz etkilenebilir.

Uyumlu – Otonom Kontrol: İnsansız hava aracı önceden programlanarak uçuşunu yapar ve bununla birlikte dış etkenlere karşı uyum sağlayabilir.

2.2.4. Drone Seçimi

Sahada tarama yapabilen bir insansız hava aracından elde edilecek verilerin miktarının oldukça büyük olmasının yanı sıra bu verilerin analizi ve değerlendirilmesi için harcanacak insan gücü ve zamanın da oldukça fazla olacağı aşikârdır. Bu durum maliyeti düşük, hızlı ve

akıllı bir deęerlendirmenin nasıl yapılacağı sorusunu akla getirmektedir. İnsansız hava araçlarından alınacak görüntülerin uygun görüntü işleme algoritmaları kullanarak işlenmesi bu soruların cevabı olacaktır. Son yıllarda insansız hava araçları ve görüntü işleme üzerine çok sayıda araştırma yapılmakta ve uygulamalar geliştirilmektedir Çünkü bilim insanları ve endüstri geliştiricileri yaptıkları çalışmalarda insansız hava araçlarını ve görüntü işleme tekniklerini daha fazla kullanmakta ve bu alanda bulunan boşlukları deęerlendirmektedirler. Yapılan her yeni çalışma insan hayatını kolaylaştırmaktadır.

Günümüzde görüntü işleme teknolojileri kabiliyetlerinden dolayı neredeyse insansız hava araçları ile bütünleşik hale gelmiştir. Nitekim günümüzde görüntü alma, görüntü işleme ve görüntü analiz etme işlemleri ile uğraşan şirketlerin neredeyse tamamı görüntü işleme algoritmalarından faydalanmakta ve birçok uygulamada insansız hava araçlarını tercih etmektedirler.

Drone seçimi yapılırken dikkat edilmesi gereken özellikler:

- Ağırlık,
- Dayanıklılık,
- Uçuş mesafesi,
- İrtifa
- Kanat yükü,
- Kamera özellikleri,
- Motor tipi.

2.3. Nesne Tespiti

Günümüzde yaygın bir şekilde üzerinde çalışma yapılan alanlardan bir tanesi de görüntü tabanlı nesne tespiti ve nesnelerin sınıflandırılması alanıdır. Bu konu üzerine yapılan birçok çalışma mevcuttur ve yeni çalışmalar ve tasarımlar üzerinde de çalışılmaktadır.

Görüntülerde ve videolarda karmaşık bir resim içerisinde bulunan ve bir anlamı olan nesnelerin görüntü işleme teknikleri kullanılarak hangi bölgede olduğunun tespit edilmesi ile ilgilenen bilgisayar teknolojisi alanına nesne tespiti adı verilir. Aynen insanlarda bulunan görme sistemi gibi bir sistemle hedef bilgilere ulaşılması amaçlanmaktadır. Görüntülerde hareketli nesnelerin takip edilmesi bilgisayarlı görme uygulamalarındaki önemli konulardan biridir [8]. Hareketli nesnelerin tespit edilmesinde sabit nesnelerin tespitine göre daha fazla

zorluk yaşanmaktadır. Hareketli hedeflerin bulunması yol trafik kontrolü, otopark kontrolü gibi durumlarda da önem kazanmaktadır [32].

Nesne tespiti için temel çalışma prensibi tespit edilecek nesnenin insan, hayvan, araç, yaya geçidi, ev, plaka, insan yüzü, yaya gibi bir sınıfa ait örnek olması mantığına dayanmasıdır. Nesne tespitinde işlenen görüntüler üzerinde bulunan nesnelerin bulunduğu alanın bir çerçeve içerisine alınması ve koordinatlarının tespiti gerekmektedir. Nesne tespiti duruma göre anlık görüntüler ya da daha önceden kayıtlı görüntüler üzerinden önemli bilgilerin ortaya çıkartılması veya sınıflandırılması şeklinde yapılabilir. Görüntü üzerinde bulunan nesnelerin tespiti için birçok yöntem ve teknik geliştirilmiştir.

2.3.1. Nesne Tespiti Algoritmaları

Nesne tespit algoritmaları genel olarak 2 kısımda incelenebilir. Bunlar sinir ağı tabanlı ve sinirsel olmayan yaklaşımlardır. Sinir ağı yaklaşımlarında YOLO, SSD, RefineDet, Retina-Net, R-CNN ve R-CNN'ye ait daha geliştirilmiş modelleri olan Fast R-CNN, Faster R-CNN ve Mask R-CNN çoğunlukla tercih edilen yaklaşımlardır. Sinirsel olmayan yaklaşımlarda ise Haar Cascade, SIFT, HOG ilk akla gelen yaklaşımlardır. Her iki yaklaşım da başarılı sonuçlar alan sistemlerdir. Son yıllarda yapay zekâ tabanlı bileşenlerle güçlendirilen bu sistemler hem daha hızlı hem de daha kesin hedef tespiti yapmayı sağlamaktadır[33]. Bu sistemler kendilerine özgü farklı metot ve teknikler kullanarak nesne tanımada her geçen gün aşama kaydetmektedirler.

3. MATERYAL VE YÖNTEM

Bu başlık altında çalışmamızda kullandığımız paketlerin ve platformların kısaca açıklamaları yapılacak ve ayrıca bu paket/platformların neden seçildiği hakkında bilgi verilecektir.

3.1. OpenCV

OpenCV (Open Source Computer Vision Library, Açık Kaynak Bilgisayar Görüşü Kütüphanesi) açık kaynaklı görüntü işleme ve yüz tanıma kütüphanesidir [34]. Görüntü işleme çalışmalarında sıklıkla kullanılan algoritmaları bünyesinde bulunduran OpenCV, platform bağımsız ve yüksek performans gösteren oldukça faal açık kaynak kodlu ve tamamen ücretsiz bir kütüphanedir. C++, Python, Java ve MATLAB başta olmak üzere birçok programlama diline doğrudan ya da geliştirilen wrapperlar aracılığıyla desteği vardır. Çok fazla doküman ve kaynak bulmak mümkündür.

Görüntü işleme yazılımları için performans en önemli kriterlerden birisidir. Görüntü işleme yazılımları işlemci, ekran kartı ve bellek ihtiyacı olarak en yüksek olanlara ihtiyaç duyarlar. Bu sebepten ötürü yazılımın yüksek performans göstermesi için seçilen platform ve kullanılacak olan programlama dili oldukça kritik öneme sahiptir. Görüntü işleme üzerine yapılan çalışmalarda geçmişten günümüze C, C++ ve JAVA başta olmak üzere çeşitli programlama dilleri başta performans olmak üzere çeşitli sebeplerden dolayı tercih sebebi olmuştur. Son yıllarda ise OpenCV dokümanlarına bakıldığında ve örnekler incelendiğinde hızlı uygulama geliştirme ve yazılım kolaylığı gibi durumlardan dolayı Python programlama dili görüntü işlemede en çok tercih edilen dillerden biri haline gelmiştir.

3.2. Python

1991 yılında ilk sürümüyle çoğunlukla nesne tabanlı kısmen de fonksiyonel programlamayı destekleyen genel amaçlı bir programlama dili olarak ortaya çıkmıştır. Diğer dillere göre oldukça kolay olması, geniş standart kütüphanesi ve dinamik yapısı sebebiyle nedeniyle günümüzde oldukça popüler hale gelmiş ve kullanıcı kitlesini epey artırmıştır. Yüksek seviyeli bir dil olması ve bilgisayar hafızasında çok yer tutma probleminden dolayı önceleri pek tercih edilmezken, günümüzde yüksek bellek seviyelerine ulaşmanın kolaylaşmasından dolayı dezavantajlı olmaktan çıkmıştır. 2000'li yıllardan itibaren donanım ve yazılım

alanındaki gelişmelerin ilerlemesine paralel olarak bilimsel ve hesaplamalı mühendislik konularında sıkça tercih edilen Python programlama dili çok daha yaygın ve geniş bir kullanım alanına kavuşmuştur. Python programlama dilinin avantajları aşağıda sıralanmıştır.

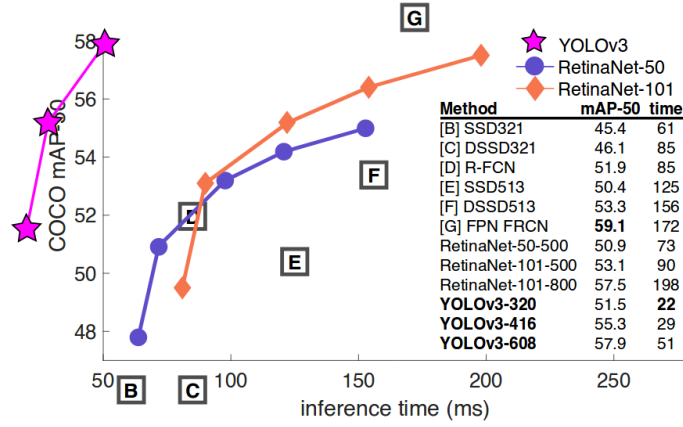
- Ücretsiz bir şekilde kullanılması,
- Söz dizimi olarak sade, kolay okunup kolay yazılabilen, insan diline yakın olması ve kolay öğrenilmesi,
- Günümüzde açık kaynak kod ve özgür yazılım akımının popüler hale gelmesi ve kitlelere yayılması,
- Bu dile ait yeni kütüphanelerin kolay öğrenilmesi ve kolay keşfedilmesi,
- Yeni program yazım aşamasında hızlı deneme yapmaya imkân sağlaması,
- Diğer dillerde sıkça karşılaşılan tırnak işareti ve parantez gibi detay hata sebeplerinin bu dilde karşımıza çıkmaması,
- Yazılan programlarda yapılan hesaplama işlemlerinin sonucunu her adımda görmeye olanak vermesiyle üzerinde çalışılan işlemlerin kolayca şekillenmesini sağlaması ve böylece yüksek verim elde edilmesi,
- Web geliştirmede, yapay zekâ ve makine öğrenmesinde kullanılabilmesi,
- Diğer dillerde yazılan çok uzun satırlı kodların çok daha az satırda yazılabilmesi,
- Farklı işletim sistemlerinde çalışması ile çeşitlilik ve esneklik sunması,
- Hızlı ve sade olması sebebiyle güvenlik sorunu ortaya çıktığında büyük bir hızla sorunun düzeltilmesi yapılabildiğinden siber güvenlik alanında rahatlıkla kullanılabilmesi,
- Tekrar tekrar kullanılabilen çevik tasarımı sayesinde çok az kodla önemli işler başarılabilmesi ve bu kodların diğer projelere uyarlanabilmesi.

3.3. YOLO (You Only Look Once) Algoritması

YOLO, konvolüsyonel (evrişimsel) sinir ağıları kullanarak görüntülerde veya videolarda bulunan nesnelere ve bu nesnelere ait koordinatları tespit eden bir algoritma olarak tanımlanmaktadır.

Joseph Redmon ve arkadaşlarının geliştirdikleri YOLO, darknet ile çalışabilen bir sinir ağıdır ve nesne tespitinde yeni bir yaklaşımdır [35]. OpenCV platformu üzerinde kullanılabilen YOLO, görüntülerdeki nesnelere ve bu nesnelere ait koordinatları tespit eden ve dünya genelinde oldukça popüler olan bir algoritmadır. YOLO algoritmasını ön plana çıkaran durumlar ise oldukça hızlı, iyi sonuçlar veren ve gerçek zamanlı nesne tespiti yapabilen bir algoritma olmasıdır. Diğer algoritmalar genelde nesne bulunması muhtemel alanlarda bölge bazlı çalışıp ayrı ayrı sınıflandırıcıları yürütmekte ve iyi sonuçlar almaktalar, ancak bir resmi birden fazla işleme yaptıkları için iş yükleri artmakta ve daha yavaş sonuç almaktadırlar. YOLO ise kendi çalışma prensibi ile ön plana çıkmakta hem doğru hem de hızlı sonuç almaktadır.

Nesne tespiti yapan algoritmalar görüntü üzerinde önce nesne bulunma ihtimali yüksek alanları belirleyip o bölgelerde farklı sınıflandırıcıları yürütmektedir. Resmin üzerinde ölçeklendirmenin uygulandığı bu sistemlerde modelde en yüksek ölçüm değerini alan kısımda nesne olduğu düşünülmektedir. Bu yöntemin avantajı iyi sonuçlar ortaya koyması, ancak dezavantajı ise bir görüntüye birden fazla işlem yapıldığı için zaman kaybı oluşturmasıdır. YOLO ise tamamen farklı ve kendine özgü bir yaklaşımla görüntünün tamamına tek bir sinir ağı uygulamaktadır. YOLO'nun kullandığı bu sinir ağı görüntüyü bölgelere ayırır ve her bölge için sınırlayıcı kutuları ve olasılıkları tahmin etmektedir. Bu sınırlayıcı kutular tahmin edilen olasılıklara göre ağırlıklandırılır. YOLO'nun sınıflandırıcı tabanlı sistemlere karşı da bir takım avantajları ortaya çıkmıştır. YOLO test esnasında resmin bütününe bakar ve tahminler görüntüdeki küresel bağlam tarafından bilgilendirilir. Her görüntü bir kez sinirsel ağdan geçirildiği için resim içerisinde bulunan nesnelere ait koordinatları tek seferde tespit edilmiş olur. Böylece YOLO sonuca hızlı ulaşabilmektedir.



Şekil 3.1. YOLO ve Diğer Algoritmaların Karşılaştırılması[36].

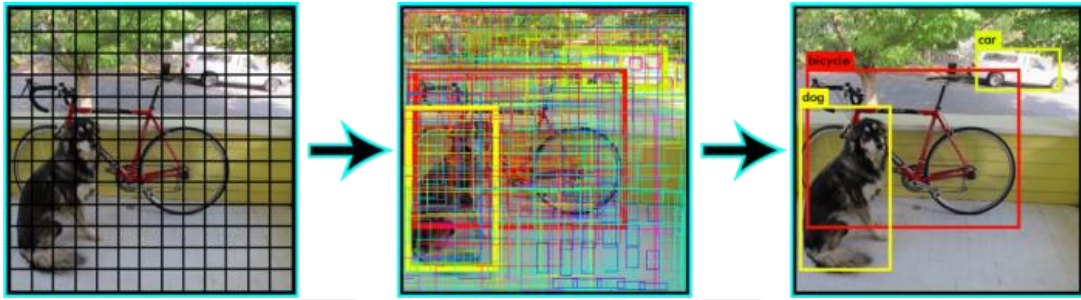
YOLO algoritması ile diğer iyi sonuçlar verdiği bilinen geleneksel algoritmaların, COCO veri seti üzerinde karşılaştırılması Şekil 3.1’de sunulmuştur. COCO veri seti 1.500.000 nesne örneği ve 80 farklı nesne kategorisi içermektedir. YOLO algoritmasının diğer algoritmalara göre üstünlüğü şekilde gözükmektedir. İnference time (çıkarma zamanı) değerlerine göre YOLO versiyonlarının diğer algoritmalara göre çok hızlı olduğu ayrıca mAP (genel ortalama kesinlik değeri) değerleri incelendiğinde YOLO algoritmasının süreye kıyasla doğruluğunun oldukça iyi durumda olduğu söylenebilir.

3.3.1. YOLO Algoritması Çalışma Prensibi

Resimler tek bir kareden (frame) oluşur. Videolar ise birçok kareden oluşur. YOLO algoritması resim işleme yaparken tek bir kare için çalışırken videolarda ise tüm kareler için tekrar tekrar çalışır, ancak temel de her ikisi için de aynı mantıkla hareket eder.

Şekil 3.2’de görüldüğü üzere YOLO algoritması, ilk önce görüntüyü ızgara biçiminde bölgelere ayırır. Bu ızgaralar 3x3, 5x5, 13x13, 26x26 gibi değerlerde olabilir. Her bir ızgara kendi içinde nesne olup olmadığını kontrol eder. Eğer nesne varsa nesnenin orta noktasının kendi içerisinde olup olmadığını, orta noktası kendi içerisindeyse nesnenin uzunluğunu, yüksekliğini ve sınıfını tespit eder. Ardından bulunduğu nesnelere çevreleyecek şekilde çevreleyici kutu (bounding box) içerisine alır. O bölgede bulunduğu nesne için bulunma olasılığı hesaplama işlemi yapar. Bu hesaplama işlemi bir güven skorudur. Güven skoru o nesnenin yüzde kaç olasılıkla tahmin edilen nesne olduğunu bildirir. Örneğin, tespit edilen bir kamyon için güven skoru 0,25 ise bunun anlamı o nesnenin kamyon olma olasılığının oldukça düşük olduğu, 0,80 ise bunun anlamı o nesnenin kamyon olma olasılığının çok yüksek olduğudur. Yani YOLO yaptığı tahminin ne kadar güvenilir ya da ne kadar güvenilir olmadığını bildirir. Her bir bölgede nesne olup olmadığı araştırılır. Eğer bir nesne

tespiti yapılırsa, o nesneye ait yükseklik, genişlik ve orta noktalarının yatay ve dikey koordinatları belirlenerek nesne çevresine bounding box (çevreleyici - sınırlayıcı kutu) çizilir. Bu arada bir takım alt işlemlerde yapılmaktadır. Her bir bölge için içerisinde güven skoru yer alan bir tahmin vektörü oluşturulur. Güven skorunun 0 olması nesnenin olmadığı 1 olması ise o bölgede nesnenin olduğu anlamına gelir. Aynı bölge içerisinde aynı nesne için birden fazla bounding box çizdirilebilme durumu problemine karşı ise non-maximal suppression tekniği kullanılır. Bu teknik ile en yüksek güven skoruna sahip olan bounding box'ın kalması ve diğer bounding box'ların görüntüden çıkarılması sağlanır.



Şekil 3.2. YOLO ile Nesne Tespiti[37].

3.3.2. Bounding Box (Sınırlandırıcı - Çevreleyici Kutu) Tahmin İşlemi

YOLO algoritmasının bir ızgara içerisinde nesne tespiti yaptığında bulunduğu nesnelere çevreleyecek şekilde çizdiği çevreleyici kutuya bounding box denir. Bounding box çizilmesi için gerekli bir takım hesaplama işlemleri vardır ve bu hesaplamalar sonucunda bir güven skoru ortaya çıkar. Şekil 3.3'te bounding boxes konum ve boyut hesaplamaları için kullanılan değerler aşağıda verilmiştir.

$$b_x = \sigma(t_x) + c_x \quad \text{Denklem 3.1}$$

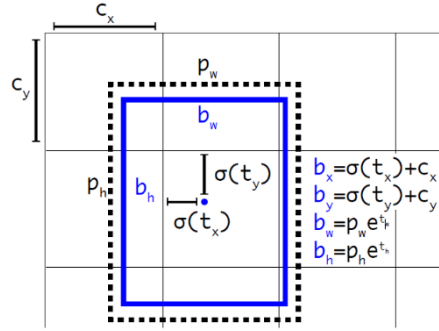
$$b_y = \sigma(t_y) + c_y \quad \text{Denklem 3.2}$$

$$b_w = p_h e^{t_w} \quad \text{Denklem 3.3}$$

$$b_h = p_h e^{t_h} \quad \text{Denklem 3.4}$$

t_x, t_y, t_h, t_w : her bir kutucuk için ağ tarafından tahmin edilen Bounding box koordinatları ve boyutları, c_x, c_y : ızgaranın resmin sol üst köşesine olan uzaklıkları, p_w, p_h : önceden belirlenen kutunun genişliği ve yüksekliği, b_x : nesnenin orta noktasının x koordinatı, b_y : nesnenin orta

noktasının y koordinatı, b_w ; nesnenin genişliği, b_h ; nesnenin yüksekliği, σ ; sigmoid fonksiyonunu temsil eder ve t_x , t_y değerlerini 0 - 1 aralığına düşürür.

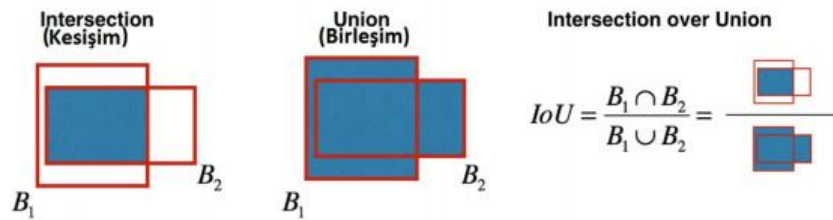


Şekil 3.3. Bounding boxes konum ve boyut hesaplama[36].

3.3.3. Non – Maximal Suppression Algoritması

Aynı bölge içerisinde aynı nesne için birden fazla bounding box çizdirilebilme durumu problemine karşı geliştirilmiş olan non-maximal suppression tekniği ile en yüksek güven skoruna sahip olan bounding box'ın kalması ve diğer bounding box'ların görüntüden çıkarılması sağlanır. Bu algoritma ile güven skoru 0,5 gibi belli bir seviyenin altında olan tüm kutular ilk önce devreden çıkartılır, halen kutu varsa en yüksek güven skorlu kutu seçilir. Bu değer IoU (Intersection Over Union) değeri 0,5 ten fazla olan diğer tüm kutular çıkartılır. Bir sınır kutusunda bulunan bir nesne için hesaplanan nesnellik puanı bir sonraki nesne için hesaplanan nesnellik puanı ile karşılaştırılır. Nesnellik puanı düşük olan elenir ve nesnellik puanı yüksek olan bir sonraki nesnenin nesnellik puanı ile karşılaştırılır ve her defasında büyük olan kalır. Bu işlem ile her nesne için sadece bir bounding box kalmış olur.

IoU değeri ise Şekil 3.4'te olduğu gibi bounding boxların konumsal olarak kesiştikleri bölgenin yine konumsal olarak birleşikleri bölgeye bölünmesiyle hesaplanmaktadır.



Şekil 3.4. IoU Değerinin Hesaplanması[38].

3.5. Plaka Okuma

Optical Character Recognition (Optik Karakter Tanıma) kısaca (OCR) olarak bilinen resim dosyalarında bulunan yazıları tespit etmeye yarayan teknolojidir. OCR sistemi, elektronik görüntüler üzerindeki karakterlerin okunarak ASCII koda dönüştürülme işlemidir[39]. Plaka okuma işlemi aslında OCR teknolojisinin kullanıldığı bir alan olup plaka üzerinde bulunan karakterleri okumak ve plakayı tespit etmek için bu teknolojiden faydalanılır.

3.5.1. Tesseract Sınıfı

Tesseract optik karakter tanıma sistemi, 1984 ve 1994 yıllarında HP firması tarafından geliştirilmiştir [40]. Tesseract sınıfı komut satırından kullanılabilen, genel amaçlı, karakter tanımda başarısı çok yüksek, açık kaynak kodlu, ücretsiz, Türkçe destekli, esnek, bir OCR kütüphanesidir. Plaka tanıma işlemlerinde sıklıkla kullanılmaktadır.

Tesseract sınıfı harf, rakam, sembol ve türevlerini algılayıp tanıyabilecek amaçla geliştirilmiş bir algoritmadır. Görüntü üzerinde bulunan karakter topluluklarını birbirinden ayrı tutarak üzerinde çalıştığı görüntüyü satırlara bölerek çalışır. Algoritmanın kendisine ait veri deposu vardır. Bu depoda farklı karakterlere özgü parametreler bulunur. Çalıştığı görüntü üzerinde algıladığı pikseller veri deposunda bulunan parametreler ile uygunluk gösterdiğinde hedef karakteri tanır ve ortaya çıkarır. Üzerinde çalışılan görüntünün arka planında ya da üzerinde farklı desen, renk ya da işaretlemeler olduğu durumlarda ise karakterlerin ortaya çıkarılması için kullanılan yöntemler vardır. En iyi performansı beyaz arka plan üzerinde bulunan yatay yönlü ve siyah renkli metinleri okumada gösterir. Birçok yerde farklı amaçlarla karakter okumada kullanılmasıyla birlikte ideal bir plaka okuma algoritması olarak kabul edilir.

3.6. Viola-Jones Algoritması ve Haar Cascade Sınıflandırıcı

Viola-Jones Algoritması OpenCV Kütüphanesi üzerinde çalışabilen bilgisayar görmesi üzerine oldukça işlevsel, çok güçlü ve popüler bir algoritmadır. Paul Viola ve Michael Jones tarafından 2001 yılında geliştirilmiştir. Bu algoritmanın sınıflandırma kısmında literatürde haar dalgacıkları olarak bilinen matematikçi Alfred Haar'a ait olan matematiksel işlemler kullanılmıştır. Haar Cascade sınıflandırıcısı görüntü işleme için çok kullanışlı ve çok gerekli olan video veya resimlerdeki nesnelere bulmak ve onları takip etmek için OpenCV

Kütüphanesi içinde bulunan Haar Cascade sınıflandırıcısı; Paul Viola ve Micheal Jones tarafından, bu soruna çözüm için oluşturulmuştur [41].

Günümüzde halen sıkça kullanılmakta olan bu algoritma gerçek zamanlı yüz tanıma işlemleri başta olmak üzere derin öğrenme üzerine çok güçlü bir yazılımdır. En çok yüz tanıma amaçlı kullanılsa da diğer nesnelere bulmak için de kullanılır. Viola-Jones algoritmasını diğer algoritmalarından ayıran en belirgin özellikleri doğruluk oranının yüksekliği ve kullanılan donanımın özellikleri de göz önünde bulundurularak insan gözünün yakalayamayacağı bir hızda çalışabiliyor olmasıdır.

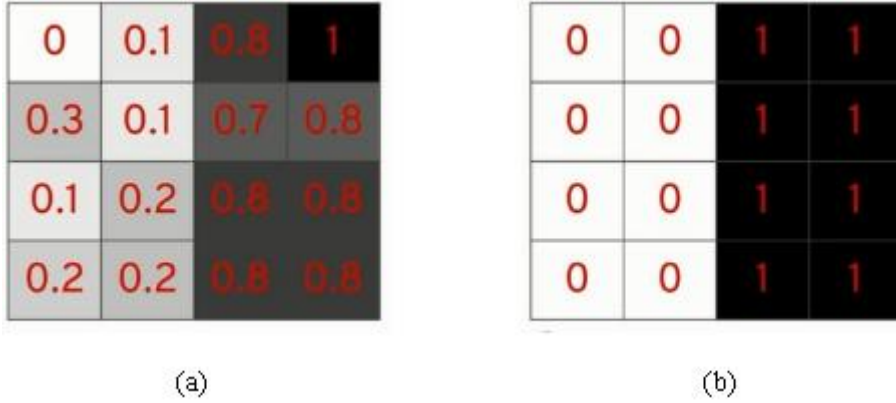
3.6.1. Viola-Jones Algoritması ve Haar Cascade Alt Bileşenleri

Viola-Jones Algoritması 4 ana alt bileşenden oluşmaktadır [42]. Bu bileşenler Haar Benzeri Öznitelikleri, Resim İntegrali, AdaBoost (Sınıflandırıcının Eğitilmesi) ve Basamaklandırma Yöntemi (Basamaklı Sınıflandırıcı) olarak detaylıca incelenmiştir.

3.6.2. Haar Benzeri Öznitelikleri

Nesne tespitinde görüntülerin dijital özelliklerine Haar özellikleri denir. Bir resmin dikdörtgenler ile tarandıktan sonra dikdörtgenlerin içerdikleri küçük resimlerin toplamlarına veya farklarına eşik değerleri uygulanmasına Haar öznitelikleri denir. Haar filtreleri uygulanarak istediğimiz görüntülerden öznitelikler elde edilmektedir. Filtre resim üzerinde adım adım ilerler. Beyaz ve siyah kısımların piksel yoğunluk değerleri toplanır. Beyaz ve siyah kısımların yoğunluklarının toplamları ve farkları hesap edilerek öznitelik değeri ortaya çıkar. Hesap edilen öznitelik değerleri Şekil 3.5’de görüldüğü üzere 1’e yakınsa 1 ve 0’a yakınsa 0 kabul edilir. Çıkan öznitelik değerinin yüksekliği nesneye ait özellik olma olasılığını artırır, çıkan öznitelik değerinin düşüklüğü yani siyah alan ile beyaz alanın toplamları arasındaki farkın az olması ise nesneye ait olma olasılığını düşürür. Eğer bu değer düşük çıkarsa algoritma o alan üzerindeki özellikleri dikkate almaz, yüksek çıkarsa algoritma sonraki aşamaya geçer. Örnek olarak insan yüzünü ele alacak olursak gözler yanak bölgesine göre daha koyudur. Göz ve yanak bölgesine yan yana iki dikdörtgen koyup, bu dikdörtgenlerin altı kısmında kalan bölgelerin piksellerinin yoğunluklarının farkı tespit edilip, bu fark insan yüzü tespit edilmesinde kullanılabilir. İnsan yüzünde bunun gibi farklı özellikler kullanılarak insan yüzünü diğer nesnelere ayırt edebilecek özellikler tespit edilebilir. Elde edilen özellikler bir araya gelip bir grup oluşturduğunda ise bu grup bir sınıflayıcı yardımıyla eğitilerek insan yüzünü diğer nesnelere ayırt edebilecek hale gelir.

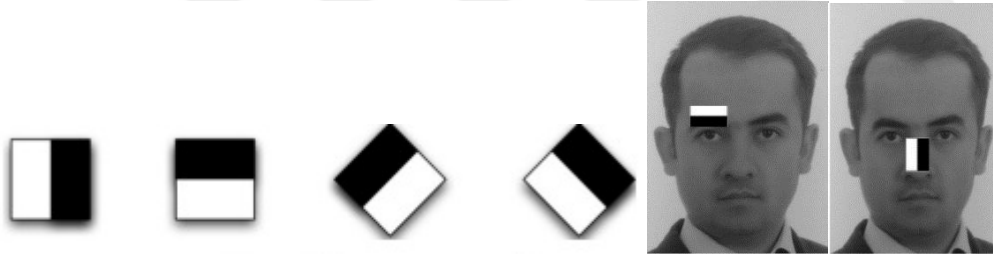
Bir başka örnekte ise yol üzerinde bulunan yaya geçidi açık ve koyu renklere göre yoldan ayırt edilebilir.



Şekil 3.5. Hesaplanan (a) ve Kabul Edilen (b) Öznitelik Değerleri

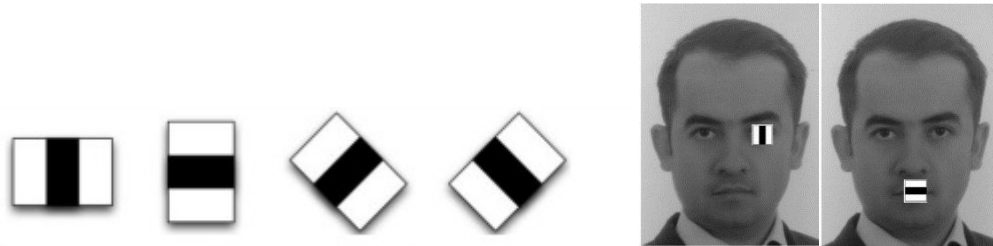
Aşağıda bulunan Haar benzeri öznitelikleri, görüntüler üzerinde kullanılarak nesne tespit edilmektedir.

Kenar Özellikleri: Görüntü üzerinde alt alta veya yan yana koyu ve açık rengin olduğu durum kenar özelliği olduğunu belirtir. Şekil 3.6'da kenar özellikleri ve bu kenar özelliklerinin resim üzerine uygulanma biçimi gösterilmiştir.



Şekil 3.6. Kenar Özellikleri ve örnek resim üzerinde uygulanması

Çizgi Özellikleri: Görüntü üzerinde sırayla açık, kapalı ve açık renklerin olduğu durum çizgi özelliği olduğunu belirtir. Şekil 3.7'de çizgi özellikleri ve bu çizgi özelliklerinin resim üzerinde uygulanma biçimi gösterilmiştir.



Şekil 3.7. Çizgi Özellikleri ve örnek resim üzerinde uygulanması

Merkez – Çevre Özellikleri: Merkezde koyu ve kenarı açık renklerin olduğu kare ve 45 derece eğik karelerin olduğu durumu belirtir. Şekil 3.8’de merkez ve çevre özellikleri gösterilmiştir.



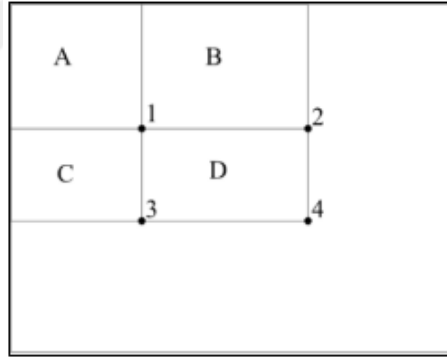
Şekil 3.8. Merkez - Çevre Özellikleri

3.6.3. Resim İntegrali

Algoritma ile oluşturulan dikdörtgenlerde tespit edilen küçük resimlerin sayılarının hızlı bir şekilde toplanması için integral imge kullanılır. Konum olarak (x,y)’nin üstünde ve solunda kalan alanın piksel değerleri toplanarak integral imgenin (x,y) konumundaki değeri hesaplanır. Şekil 3.9’ da resim integral hesabı gösterilmiştir. Aşağıda Denklem 3.5 ile verilen formül ile ifade edilir [42].

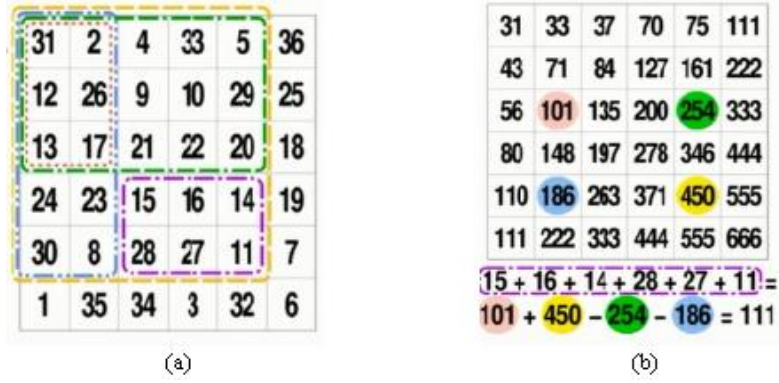
$$ii(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y')$$

Denklem 3.5



Şekil 3.9. Resim integral hesabı

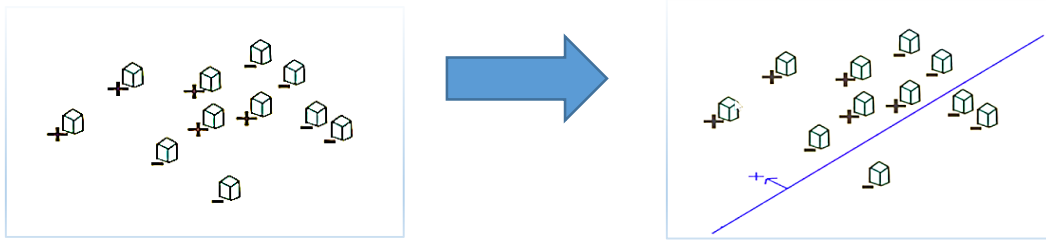
A dikdörtgeni içindeki piksel değerleri toplamı, integral imgenin 1 konumundaki değeridir. 2 konumundaki değer A+B, 3 konumundaki değer A+C, 4 konumundaki değer A+B+C+D dir. D içindeki piksel değerleri toplamı 4+1-(2+3) olarak hesaplanır [20]. Şekil 3.10’ da bu hesaplama işlemi örneği gösterilmiştir.



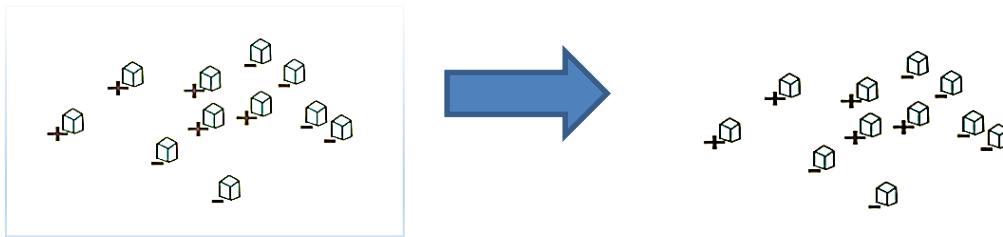
Şekil 3.10. Resim integrali hesaplama örneği (a) orijinal resim ve (b) resmin integrali

3.6.4. AdaBoost (Sınıflandırıcının Eğitilmesi)

Adaboost algoritması sınıflandırıcıyı eğitmek ve özniteliklerin seçilmesi için kullanılır. Burada hedef Şekil 3.11 ve Şekil 3.12’de verilen örneklerde olduğu gibi imge üzerinde bulunan negatif değerlerin ele alınarak minimize edilip ayıklanması ve pozitif değerlere ulaşılmasıdır. Bu algoritmada zayıf sınıflandırıcılar birer öznitelik gibi ele alınır ve farklı düzenlerde bir araya gelerek güçlü sınıflandırıcıları oluştururlar. Resim üzerinde kayan pencere ile nesne taraması yapılır. Sınıflandırıcı aranılan nesneyi bir pencere üzerinde nesnedir veya değildir şeklinde ayırt etmek üzere eğitilir. Pencerelerden gelen çok sayıda dikdörtgen özniteliğin çok az bir kısmı etkili bir sınıflandırıcı oluşturmak için kullanılır. Zayıf sınıflandırıcılar bir öznitelik olarak düşünülür. Akabinde içerisinde nesne bulunan ve bulunmayan örnekleri en ideal şekilde ayıran öznitelikler aranır.



Şekil 3.11. Zayıf Sınıflandırıcıların minimize edilmesi örnek 1



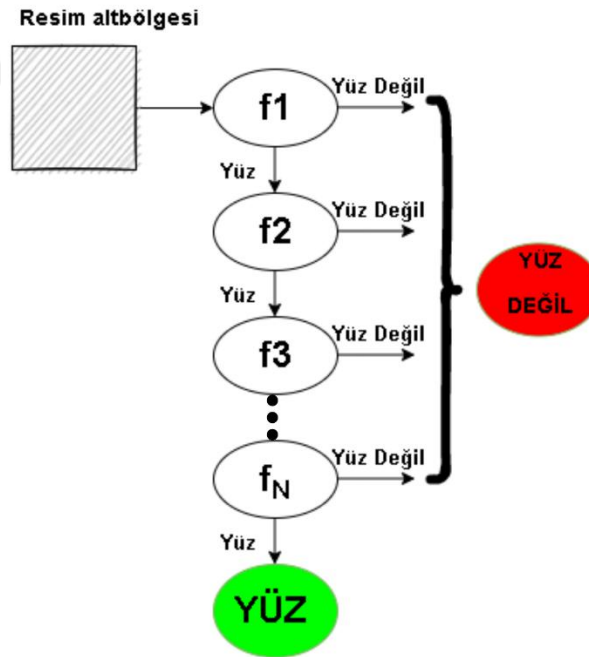
Şekil 3.12. Zayıf Sınıflandırıcıların minimize edilmesi örnek 2

Her bir zayıf sınıflandırıcı aslında içerisinde asgari sayıda yanlış sınıflandırılmış örnek bulunan eşik değerli bir fonksiyondur. Bu fonksiyon aşağıda Denklem 3.6'da verilmiştir.[42]

$$h(x, f, p, \theta) = \begin{cases} 1 & p \cdot f(x) < p \cdot \theta \text{ ise} \\ 0 & \text{aksi halde} \end{cases} \quad \text{Denklem 3.6}$$

$H(x, f, p, \theta)$: bir zayıf sınıflandırıcı, f ; bir öznitelik, θ ; bir eşik değeri, p ; eşitsizlik işaretinin yönünü belirten polarite, x ; İmgenin 80x32'lik bir penceresi olarak verilmektedir.

Zayıf sınıflandırıcılar bir araya gelerek esas sınıflandırıcıyı oluştururlar. Eğitim kümesine yeni bir zayıf sınıflandırıcı ilave edildiğinde zayıf sınıflandırılan örneklerin sayısı artırılmış, doğru sınıflandırılan örneklerin sayısı ise azaltılmış olur. Bu şekilde yeni sınıflandırıcı ekleme işlemlerine devam edilir ve nihayetinde en az sayıda hata veren zayıf sınıflandırıcıya ulaşılır. Aslında zayıf sınıflandırıcılar birleşerek hatayı minimize eden nihai öznitelige ulaşılmış olur. Amaçlanan doğru ve yanlış tespit oranlarına ulaşıldığında ise yeni zayıf sınıflandırıcı ilavesi durdurulur. Şekil 3.13 yüz tespiti için AdaBoost sınıflandırıcısının adım adım çalışmasını göstermektedir.

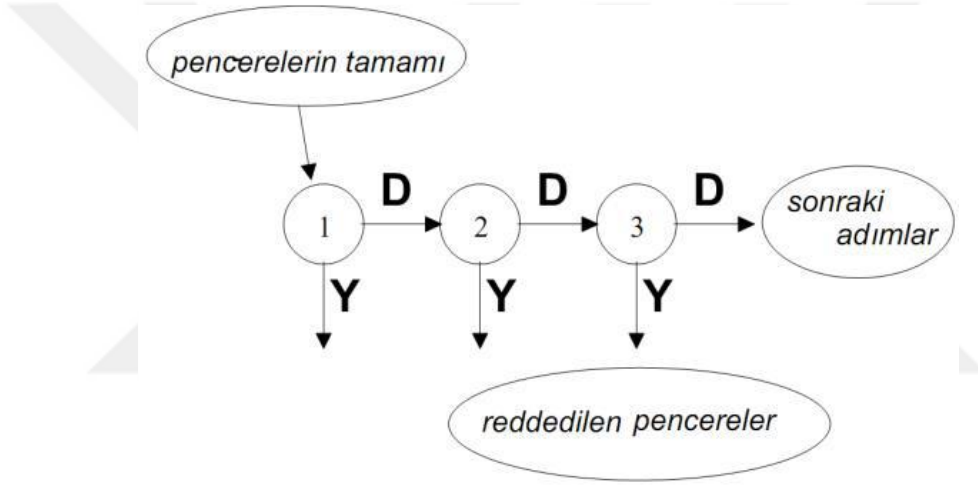


Şekil 3.13. AdaBoost Sınıflandırıcısı [43].

3.6.5. Basamaklandırma Yöntemi (Basamaklı Sınıflandırıcı)

Bu yöntem hızlı bir şekilde küçük ve etkili bütün pozitif örnekleri tespit etmek ve negatif örnekleri elemek için kullanılmaktadır. Sınıflandırıcılar taşıdıkları özniteliklere göre azdan

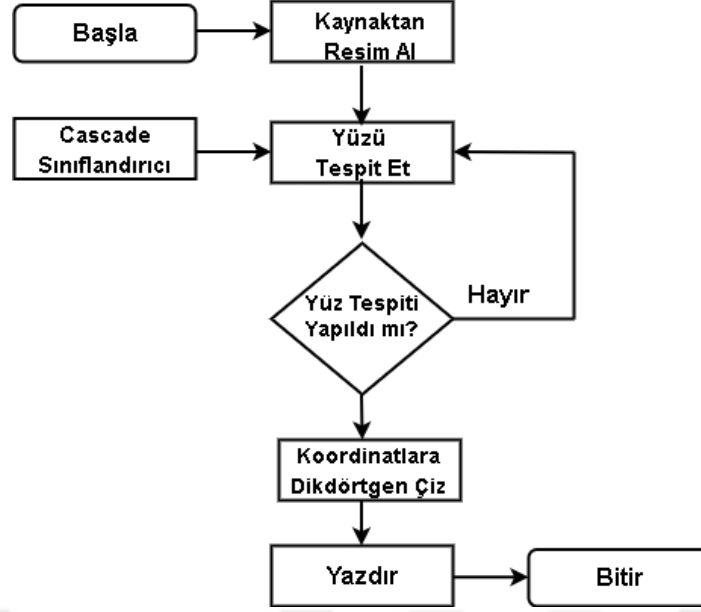
çoğa doğru boyutlarına ve hassasiyetlerine göre basamaklandırılırlar. İçerisinde aranılan nesne bulunmayan pencereler, çok az öznitelik bulunduran çok basit bir sınıflandırıcı ile elenebilirler. Bir pencere Şekil 3.14’te gösterildiği gibi birinci dereceden bir sınıflandırıcıyı geçerse ikinci dereceden bir sınıflandırıcı ile tetiklenir, onu da geçerse üçüncü dereceden bir sınıflandırıcı ile tetiklenir. Algoritma bu şekilde devam eder. Burada her pencere yoğun öznitelik bulunduran sınıflandırıcıdan geçmek zorunda kalmaz. Adaboost ile eğitilen üst basamaklardaki karmaşık sınıflandırıcılar önceki aşamalarda daha basit sınıflandırıcıları geçen pencereler için çalıştırılırlar. Örneğin insan yüzünün arandığı bir pencerede göz, kaş burun dudak gibi organlardan biri ya da birkaçı yoksa o pencere hızlıca elenir ve diğer pencereye geçilir. Böylece iş yoğunluğu çok önemli miktarda azalır ve zamandan büyük oranda kazanç elde edilir.



Şekil 3.14. Basamaklandırma Yöntemi[42].

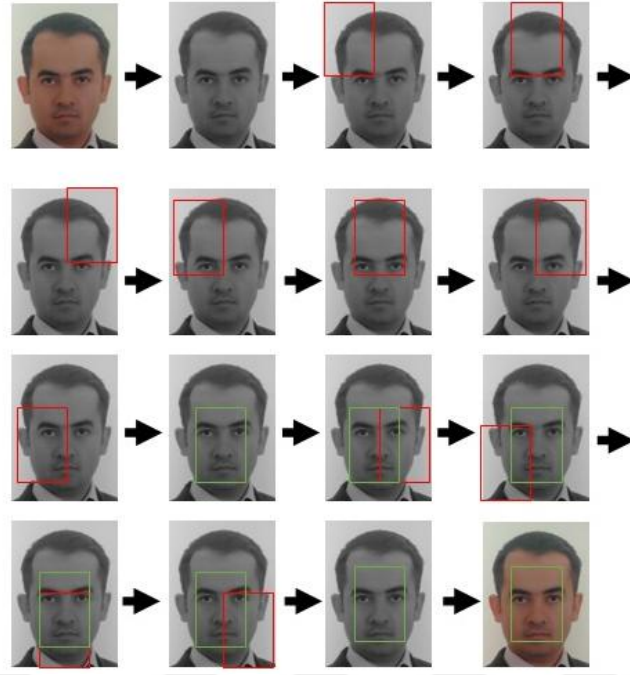
3.6.6. Viola-Jones ve Haar Cascade Sınıflandırıcı ile Yüz Tanıma İşlemi

Yüz tanıma işlemine ait akış diyagramı Şekil 3.15’ de gösterilmiştir.



Şekil 3.15. Yüz Tanıma İşlemi Akış Diyagramı [44].

Algoritma yüz tespitinde çalışırken ilk önce resmi gri tonlamalı (grayscale) yapar. Haar özniteliklerini kullanarak kenar, çizgi ve merkez – çevre gibi özellik durumlarını belirler. Karşıdan yüz tespitinde önce kaş ve üzeri alın kısmı yani başın üst ve sol köşe kısmı dikdörtgen çerçeve içine alınır. Kaşın siyaha ve üzerindeki alın kısmının beyaza yakın olma durumunu kontrol eder. Devamında dikdörtgen çerçeveyi başın diğer kısmına doğru kaydırır ve diğer kaş ve alın kısmını kontrol eder. 2 farklı koyu renkte kaş ve daha açık renkte alın kısmını tespit ettikten sonra dikdörtgen çerçeveyi bir gözü içine alacak şekilde aşağı indirir ve yine diğer kısma doğru kaydırmaya başlar. Burada gözün iç kısmı siyah ve iki yanı beyaza yakın olacak şekilde gözü tespit eder. Ardından burun kısmını dikey bir dikdörtgen içerisine alır ve ışık düşme açısına göre bir kısmı siyaha yakın bir kısmı da beyaza yakın tonların olup olmadığını kontrol eder. Diğer göz içinde önceki göz ile aynı renk durumu olup olmadığını kontrol eder. Devamında dudaklara bakar ve dudaklarda kenarlar siyah orta kısmı beyaz olan yatay bir dikdörtgen şeklinde dudak tespiti yapar. İki kaş, iki göz, burun ve ağız tespiti ile bir bütünlük sağlar ve böylece yüzü tespit etmiş olur. Tam bütünlük olmasa bile belli bir oranın üzerinde bu tespit sağlansa bile yüzü tespit eder. Burada bu oran değeri değiştirilerek yüz olma ihtimali zayıf piksellerin elenmesi ya da belirginliği daha az olan yüzlerin tespiti yapılabilir. Bu çalışma sistemi Şekil 3.16’da adım adım gösterilmiştir.



Şekil 3.16. Adım adım yüz tespit işlemi

Bu çalışmada model geliştirilirken OpenCV Kütüphanesi ve Python programlama dili tercih edilmiştir. Nesne tespiti için YOLO algoritması, insan yüzü tespiti ve plaka konumu tespiti için Viola-Jones Algoritması (Haar Cascade Sınıflandırıcı), araç plakalarının tespiti için Tesseract sınıfı kullanılmıştır.

Çalışmada, 4k kamerası olan 3000 metreye kadar görüntü iletimi yapabilen, katlanabilir, 4 adet fırçasız motora sahip, Mobil veya PC üzerinden bir yazılımla kontrol edilebilir bir quadcopter drone seçilmiştir. Seçilen insansız hava aracına ait görsel Şekil 3.17’de, insansız hava aracını kontrol etmek için kullanılan mobil arayüz Şekil 3.18’de ve insansız hava aracına ait özellikler Tablo 3.1’de gösterilmiştir.



Şekil 3.17. Deer One Pro (Kullanılan İnsansız hava aracı)



Şekil 3.18. Vs Gps Pro (Kullanılan Mobil Arayüz)

Tablo 3.1. Deer One Pro Özellikleri

Hız	Batarya	GPS	Mesafe	Kamera
4 m/s – 16 m/s	25 dk.	Dual GPS Way point (ara nokta) uçuş	3.2 km	120 derece geniş açı 60 derece dikey hareket Titreme giderici 1080p çözünürlük

4. MODEL TASARIMI VE UYGULAMASI

4.1. Veri Seti Elde Edilmesi

Bu çalışmada veriler Şekil 4.1’de bir görseli verilen Kırşehir Ahi Evran Üniversitesi Merkez kampüsünden toplanmıştır. Görüntülerin toplanmasında sabit yükseklik ve belirlenen sabit bir hız kullanılmıştır. Bu hızın ve yüksekliğin belirlenmesinde İHA üzerinde bulunan kamera özellikleri ana etken olmuştur. En ideal görüntü kalitesine ulaşılabilecek parametreler belirlenmiş ve kullanılmıştır. Farklı İHA modellerinde hız ve yükseklik kavramları değiştirilebilir. İHA daha önceden programlanan güzergâhta, yaklaşık 15 m yükseklikten 4 m/sn hızla ve sabit açıyla video kaydını üzerinde takılı olan bir harici belleğe depolamıştır. Her bir döngü 1200 m mesafe olacak şekilde planlanmış ve yaklaşık 7 dakika sürede tamamlanmıştır. Videolar 30 FPS olarak kaydedilmiştir.

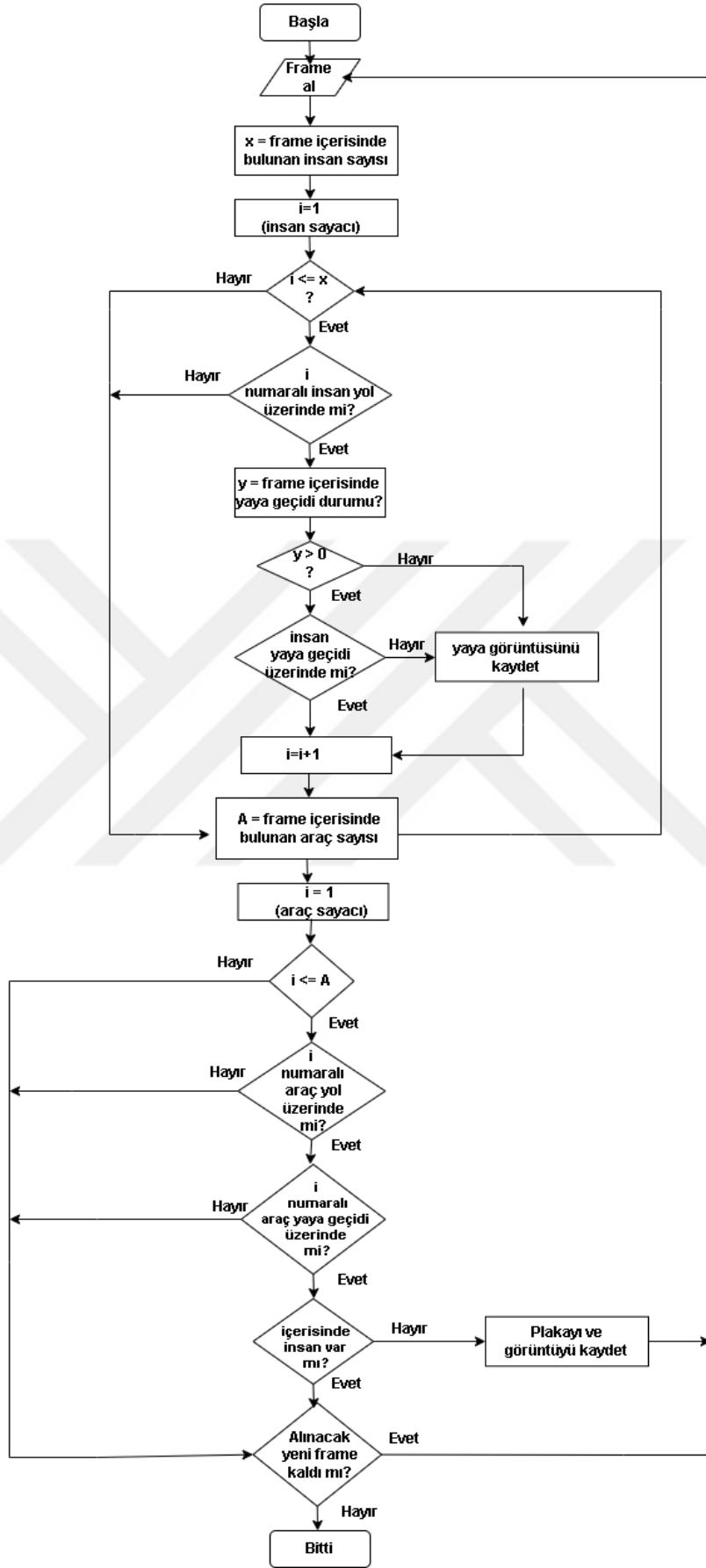


Şekil 4.1. Kırşehir Ahi Evran Üniversitesi Merkez Kampüsü

4.2. Yazılımın Geliştirilmesi

Elde edilen videoların analizi için bir model yazılmıştır. Bu yazılımın geliştirilmesinde 32GB RAM, 8GB Nvidia GPU ve 5.0 GHz işlemcili bir bilgisayar kullanılmıştır.

3000 araç, 2000 yaya geçidi ve 6000 yaya görüntüsü ile eğitilen “YOLO large” (YOLOv5l) modeli video akışında yer alan yaya geçitleri, araçlar ve yayaların tespitinin x,y,w,h (x,y konum için ve w,h ise boyut içindir) biçimiyle elde edilmesi için kullanılmıştır. Sadece 3 sınıfı ayırt edici olarak kullanılan bu modelde bir sonraki adım elde edilen sonuca göre 2 farklı şekilde devam etmektedir.



Şekil 4.2. Yazılıma ait akış diyagramı

Geliştirilen modelde Şekil 4.2’de verilen akış diyagramı kullanılmıştır. Akış diyagramına göre resimlere dönüştürülen video görüntüleri işlenmiştir. Elde edilen görüntülerde ilk adım insan ve yaya açısından taranmasıdır. Bulunan nesnelere göre işlem adımı 2 farklı şekilde ilerlemektedir.

Eğer insan bulunmuşsa; bulunan insanın nesne alanın yaya geçidinde ve kaldırımında olup/olmadığı tespit edilmiştir. Eğer yaya kaldırım veya yaya geçidinde değilse görüntüsü konum ve zaman bilgisiyle birlikte kaydedilmiştir.

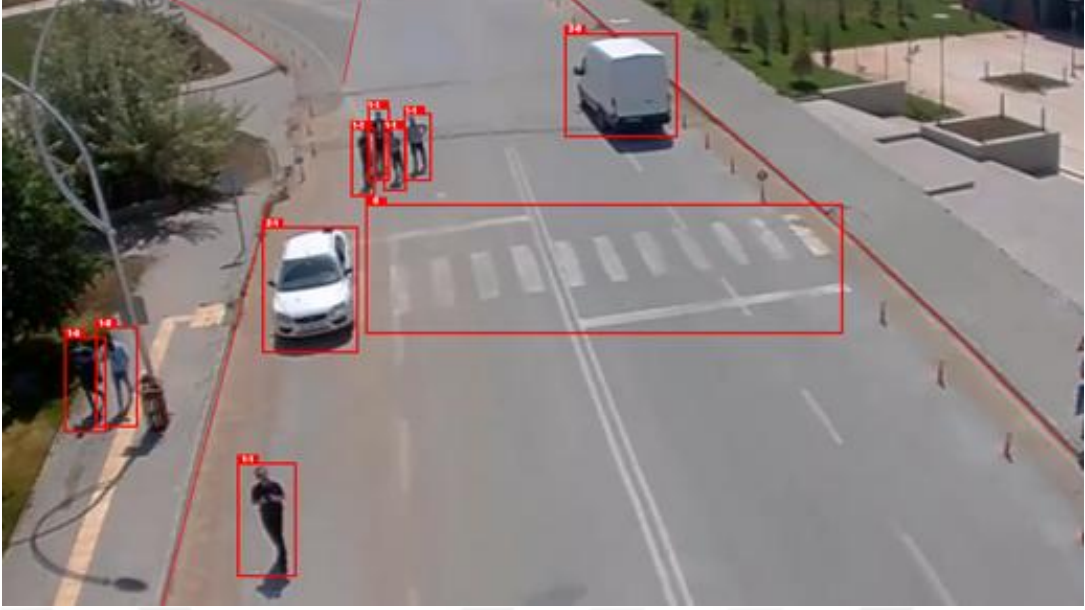
Eğer araç bulunmuşsa ve bulunan bu araç yaya geçidinde ise park etme/etmeme durumu içerisinde şoför bulunma/bulunmama durumuna göre karar verilmiştir. Eğer park ettiği sonucuna varıldı ise plaka bilgisi, konumu, görüntüsü ve zamanı kaydedilmiştir.

Eğer görüntülerde insan veya araç tespiti yapılmamış ise herhangi bir işlem sağlanmamıştır.

Algoritmanın akışında yaya, araç ve yaya geçitlerinin tespitinde YOLO algoritması ile eğitilmiş model kullanılmıştır. Yaya geçidini kullanmayan yayaların ve araç içerisinde bulunan yüzlerin tespiti için ise Viola-Jones algoritması kullanılmıştır. Araç plakalarının konumlarını bulmak için yine Viola-Jones algoritması, plaka karakterlerinin okunması içinse Tesseract sınıfı kullanılmıştır.

4.3. Yazılımın Test Edilmesi

Şekil 4.3’de veri seti içerisinde bulunan bir çerçeve bulunmaktadır. İlgili çerçevede hem araç hem de yaya bulunmaktadır. Bu yüzden ilgili çerçeve sonraki adımlara aktarılarak devam edilecektir. Geliştirilen program hız açısından daha verimli olması için komut satırı tabanlı olarak çalışmakta ve sadece sonuçlar çıktı olarak üretilmektedir. Bu sebepten dolayı Şekil 4.3 üzerinde bulunan işaretlemeler bu çalışmadaki anlatım için üretilmiştir.



Şekil 4.3. Yaya geçidi dikey mesafe hesaplama

Geliştirilen modelin çalışmasını adımlara bölecek olursak;

İlk adım YOLO ile eğitilmiş modelimizden, yayaların, araçların ve yaya geçidi konum bilgilerinin elde edilmesidir. Konum bilgileri için ilgili nesnenin sol üst köşesi başlangıç olarak kabul edilirse sırasıyla aşağıdaki bilgiler alınmaktadır:

x: tespit edilen nesnenin yatay eksenindeki konumu

y: tespit edilen nesnenin dikey eksenindeki konumu

w: tespit edilen nesnenin genişliği

h: tespit edilen nesnenin yüksekliği

İkinci adım tespit edilen nesnenin türüne göre iki farklı şekilde ilerlemektedir:

- İşlenmesi için alınan çerçevede araç görüntüsü bulunduğu için aracın yaya geçidine park etme/etmeme durumunun tespiti yapılmıştır. Bu tespit için ilk aşama aracın yaya geçidinin üzerinde olup/olmadığının belirlenmesidir. Nitekim Şekil 4.3’de araçlardan birisi yaya geçidi üzerindedir. Bundan sonraki aşama ise araç görüntüsü içerisinde yüz taraması yapılarak park etme/etmeme durumunun belirlenmesidir. Aynı araç için 1 defa yüz tespiti yapıldıktan sonra 5 sn. boyunca bir daha aynı araç için işlem yapılmamaktadır. Ancak yaya geçidi ile örtüşen bu araçta, araç hala görüntü

alanındayken bir defa bile yüz tespiti yapılmamışsa ilgili aracın plakası, görüntüsü, zaman ve konum bilgileri dışarıya çıktı olarak aktarılmaktadır.

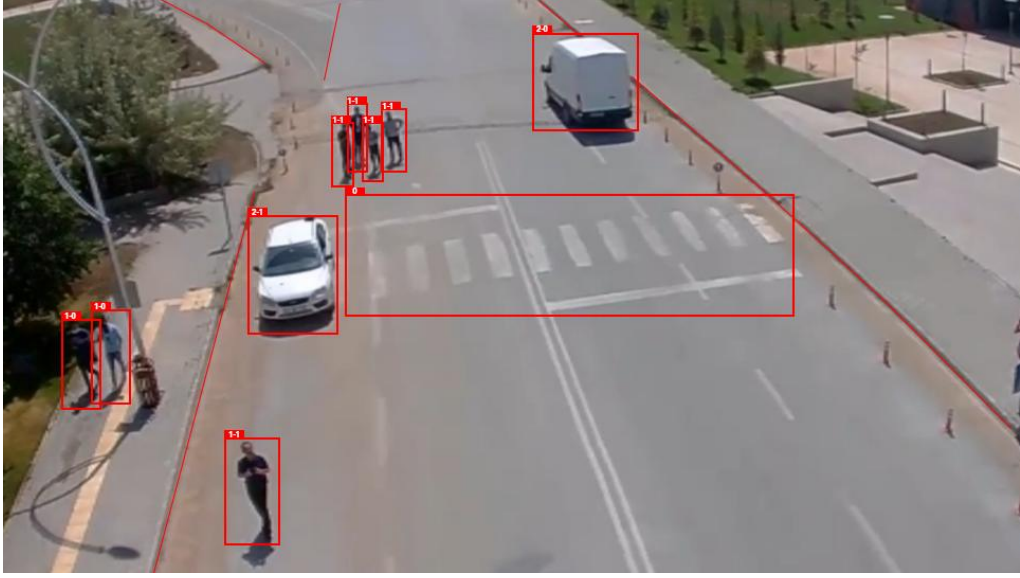
Burada 5 sn. ölçütü 4 m/s ile 15 m yükseklikte hareket eden kameramızda bir nesnenin ilk görüntüye girdikten sonra tekrar çıkması için gerekli süre olarak hesaplandığı içindir.

- b. İşlenmesi için alınan çerçevede yaya görüntüleri bulunduğu için ilk aşamada yol ve yaya geçidi taraması yapılır. Tespit edilen yaya görüntülerinden yol kenarında (kaldırım) olanlar ve yaya geçidi ile alanları örtüşenler değerlendirme dışı bırakılır. Diğer yayaların ise görüntüleri, zaman ve konum bilgileri dışarıya aktarılır.

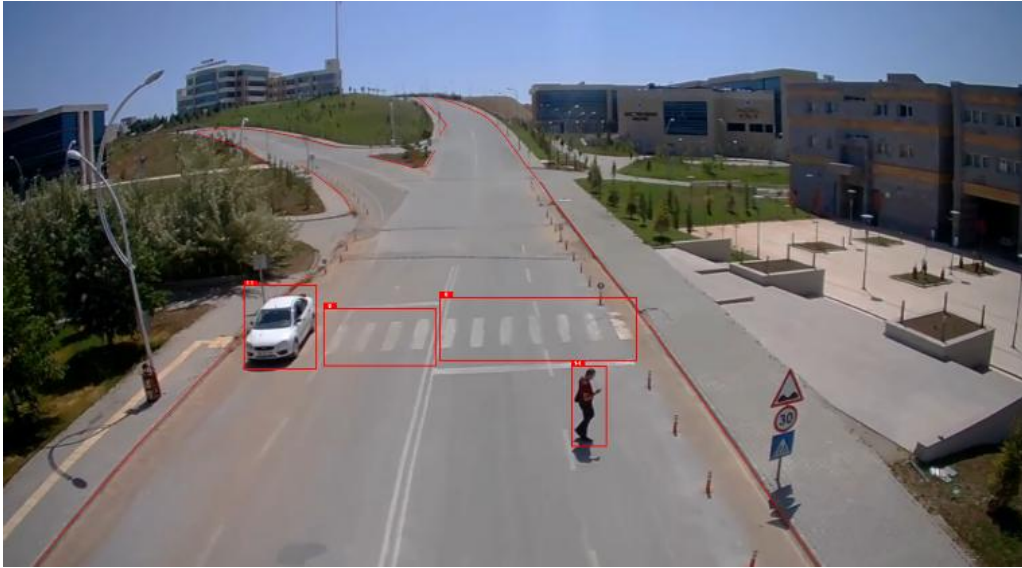


5. TARTIŞMA VE SONUÇ

Geliştirilen program hız açısından daha verimli olması için komut satırı tabanlı olarak çalıştığı için çalışma anında görsel olarak çıktı vermemektedir. Bunun yerine sadece tespit edilen durumlar uygulama çıktısı olarak üretilmektedir. Bu sebepten ötürü bu bölümde sunulan görsel çıktıların tümü geliştirilen algoritmanın çalışması sırasında, bu çalışma için görselleştirilmiş örnek çıktılardır. Bu örnekler Şekil 5.1, Şekil 5.2, Şekil 5.3 ve Şekil 5.4' te bulunan görsellerdir.



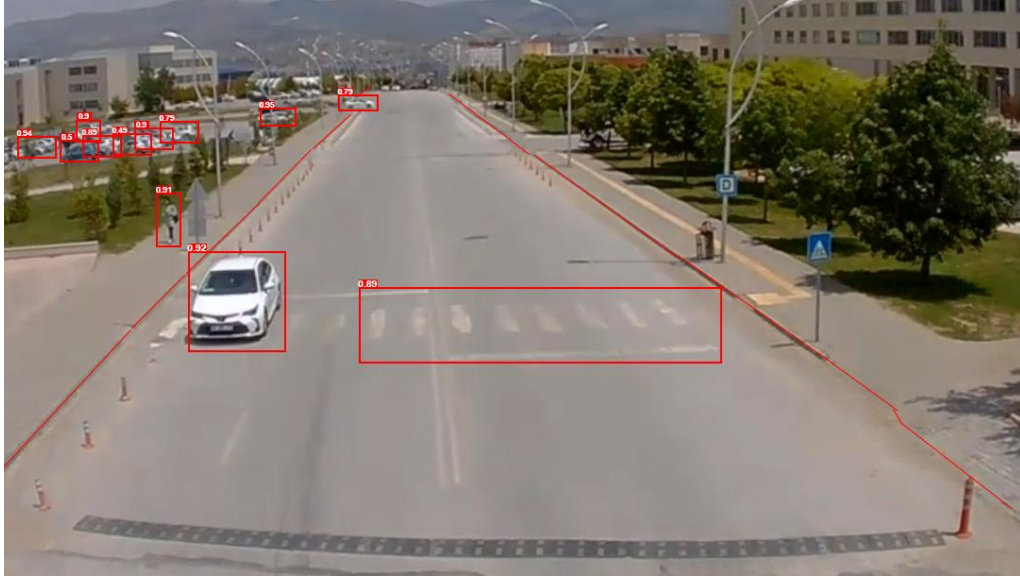
Şekil 5.1. Yaya geçidi, araç ve insan tespitleri- 1



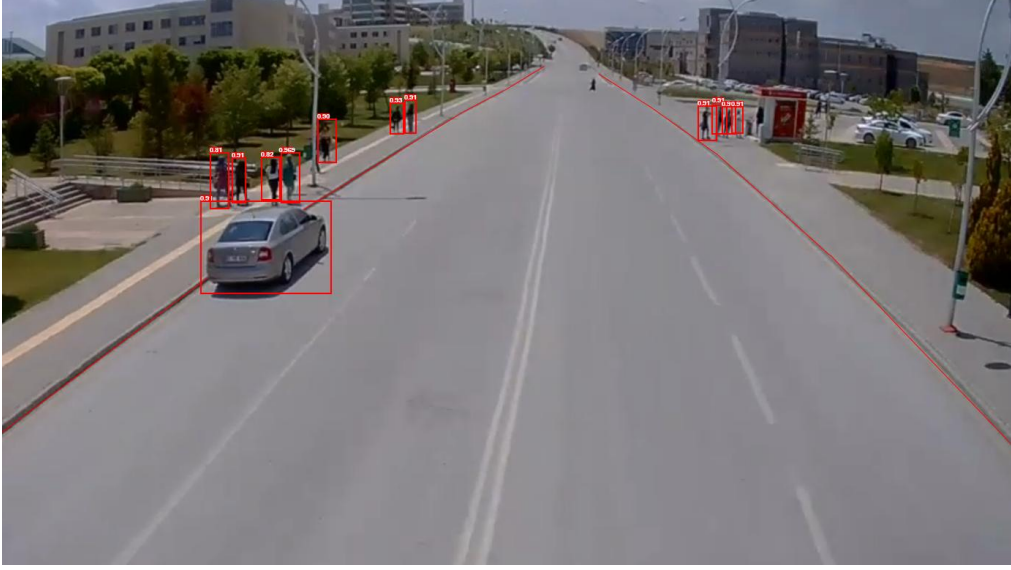
Şekil 5.2. Yaya geçidi, araç ve insan tespitleri- 2

Geliştirilen model Kırşehir Ahi Evran Üniversitesi merkez Yerleşkesi içerisinde belirli bir güzergâh üzerinde test edilmiştir. Park etmiş araçların park etme/duraklama durumları ve içerisinde insan bulunma/bulunmama durumlarına göre karar verilmiştir. Ancak bu durum teorikte mümkün görülse de uygulamada modeli en çok zorlayan kısım olmuştur. Özellikle camları filmli olan araçlarda neredeyse tüm tespitler başarısız olmuştur. Normal camlı araçlarda ise modelin başarısı düşük irtifalarda oldukça yüksektir. Ancak günün farklı zamanlarında farklı açı ile düşen güneş ışığının yansıdığı bazı durumlarda plaka karakterlerinin çıkarılmasında model başarısı oldukça düşmüştür. Yaya geçitlerinin ve yaya geçidini kullanmayan insanların yüzlerinin tespitinde oldukça yüksek başarı elde edilmiştir. Buna ek olarak kaldırımda yürüyen kişilerin tespitinde kısmen sıkıntılar yaşanmıştır.

Ayrıca günün farklı saatlerinde güneş ışığı açısından kaynaklı olarak plakanın elde edilmesinde meydana gelen parlamalar karakterlerin okunmasında bazı karakterlerin atlanmasına veya hatalı okumalara sebep olmuştur. Rüzgâr seviyesinin 5 km/saat'in üzerinde olduğu durumlarda görüntü kayıt kalitesi oldukça olumsuz etkilenmiş ve işaretlemelerde ve konum tespitlerinde hatalı sonuçlar meydana gelmiştir.



Şekil 5.3. Yaya geçidi, araç ve insan tespitleri- 3



Şekil 5.4. Yaya geçidi, araç ve insan tespitleri- 4

Çalışmada kullanılan insansız hava aracına ait kameraya göre daha yüksek çözünürlüklü bir kamera kullanılarak çözünürlükten kaynaklı sıkıntının giderilmesi ve ağırlığı daha yüksek insansız hava aracı kullanılarak rüzgârın etkisinin azaltılması şüphesiz modelin başarısını artıracaktır.

Görüntüler üzerinde Araç, Yaya ve Yaya Geçidi bulması için kullanılan YOLO algoritmasının eğitilmesinden sonra elde edilen başarımlar ölçütleri Tablo 5.1’de paylaşılmıştır.

Tablo 5.1. YOLO Başarımlar Sonuçları

Açıklama	Ölçüt
Model	YOLO v5
Hassasiyet (Precision)	%84,7
Keskinlik (Recall)	%81,2
F-Skoru (F1-Score)	%83,4
Eğitim Süresi	6 Saat 52 Dakika

Eğitilen YOLO modeli sadece Araç, Yaya ve Yaya Geçidi olmak üzere 3 sınıfı ayırt etmek için kullanıldığından dolayı Hassasiyet, Keskinlik ve F1-Skoru değerleri çok sınıfı ayırt etmesi için eğitilen modellerden daha yüksektir.

Geliştirilen modelin test edilmesi amacıyla, veri toplamak için İHA toplamda belirlenen güzergâhta gündüz vakti günün farklı zamanlarında 10 tam tur kullanılmıştır. Elde edilen

veriler YOLO algoritması ile analiz edildikten sonra elde edilen sonuçlar Tablo 5.2’de ve Tablo 5.3’te paylaşılmıştır. Elde edilen görüntülerde normalde 323 yaya, 79 araç sayılmıştır. Yayalardan 199’u yol harici bölgede yer alırken, 39’u yaya geçidi üzerinde ve 85’i ise yol üzerinde gözlemlenmiştir. Geliştirilen modelin bu sayıları tespiti üzerinde başarısı incelenerek başarımların analizi yapılacaktır.

Tablo 5.2. Geliştirilen Modelin Yaya Tespitinde Başarım Sonuçları

Açıklama	Gerçekte olan	Modelin Tespit Ettiği
Yaya geçidini kullanan kişi sayısı	39	37
Yaya geçidini kullanmayan kişi sayısı	85	79
Harici alanda yürüyen kişi sayısı	199	156
Toplam yaya sayısı	323	272

Geliştirilen modelde amaçlarımızdan birisi mevcutta 323 yayanın tümünün tespit edilmesi ve bunlardan yol üzerinde bulunanlar ile yaya geçidini kullanmayan 85 kişinin algılanarak, yüzleriyle birlikte kaydedilmesini sağlamaktır. Ancak elde edilen sonuçlar incelendiğinde 51 kişinin modelimiz tarafından hiç tespit edilmediği görülmektedir. Buna neden olarak özellikle kaldırımda birbirlerine oldukça yakın ve grup halinde yürüyen kişiler olduğu gözlemlenmiştir. Benzer şekilde yaya geçidini kullanan kişilerden sadece 2 tanesi tespit edilememiştir ve bu kişilerin birbirine oldukça bitişik yürüdükleri için tek kişi olarak algılandığı görülmüştür. Yaya geçidini kullanmayan ve yol üzerinde bulunan kişilerden ise 6 tanesi tespit edilememiştir. Bunlara ek olarak, Yaya yüzlerinin elde edilmesindeki başarı ise %60 tan yukarıya çıkarılamamıştır. Bunun temel nedenlerinden birisi ise yayaların yüzlerini İHA kamerası objektifine doğru tam olarak göstermemesi olduğu sonucuna varılmıştır. Ayrıca güneş ışıklarının farklı açı ile düşmesiyle yüzdeki parlaklık seviyesinin artması da yüzleri elde etmekteki başarımın düşmesinin sebeplerinden birisi olarak gösterilebilir.

Tablo 5.3. Geliştirilen Modelin Araç Tespitinde Başarım Sonuçları

Açıklama	Gerçekte olan	Modelin Tespit Ettiği
Yol üzerinde araç sayısı	26	26
Yol harici alanda araç sayısı	53	30
Yaya geçidi üzerinde park eden araç sayısı	11	7
Yaya geçidi üzerinde hareket eden araç sayısı	6	5

Geliştirilen modelde amaçlarımızdan birisi mevcutta 79 aracın tümünün tespit edilmesi ve bunlardan yaya geçidine park etmiş olan 11 aracın algılanarak plakalarıyla beraber kaydedilmesini sağlamaktır. Ancak geliştirilen modelle toplamda 56 araç tespit edilmiştir. 23 araç hiç tespit edilememiştir. Buradaki farkın temel nedeni ise otopark alanlarının kamera objektifine girdiği ve kamera açısından kaynaklı olarak birçok aracın başka diğer araçlar arkasında kaldığından dolayı tespit edilemediğidir. Şekil 5.3 incelendiğinde görüntü çerçevesinin sol üst köşesinde 1’den fazla tespit edilen araçlar gözükmemektedir. İlgili alanda 14 araç olmasına rağmen modelimiz bu araçlardan sadece 6 tanesini tespit etmiştir. Ancak bu teze çalışma olan kısım sadece yol üzerinde bulunan araçlar olduğu düşünüldüğünde ve Tablo 5.2 incelendiğinde, yol üzerinde bulunan bütün araçların eksiksiz tespit edildiği görülecektir. Yol üzerinde tespit edilen 26 aracın 11 tanesi yaya geçidi üzerine park etmiş oldukları halde bunlardan sadece 7 tanesi park etmiş olarak tespit edilmiştir. Bunun temel sebeplerinden birisi park eden araçların park etme/etmeme durumlarının içerisinde insan bulunma/bulunmama durumuna göre karar verilmesidir. Ancak araç camlarından yansıyan güneş ışıklarının yüz tespitini neredeyse imkânsız hale getirmesi olarak gösterilebilir. Buna ek olarak İHA üzerinde kullanılan kamera çözünürlüğünün düşük olması da yine sebeplerden birisi olarak sayılabilir. Sonuç olarak geliştirilen modelin çalışmasında her bir adım için bir Karışıklık Matrisi çıkarıldığında Tablo 5.4 ve Tablo 5.5 elde edilmiş olacaktır.

Tablo 5.4. Modelin Yaya Geçidini Kullanmayan Yayaları Tespit Etmekteki Karışıklık Matrisi

		Gerçek	
		Pozitif	Negatif
Tahmin	Pozitif	79	6
	Negatif	0	238
Doğruluk		0,98	
Duyarlılık		1,00	
Seçicilik		0,98	
F1-Score		0,79	

Tablo 5.5. Modelin Yaya Geçidine Araç Sayısını Tespit Etmekteki Karışıklık Matrisi

		Gerçek	
		Pozitif	Negatif
Tahmin	Pozitif	7	4
	Negatif	1	67
Doğruluk		0,94	
Duyarlılık		0,8	
Seçicilik		0,94	
F1-Score		0,86	

Tablo 5.6. Yaya Geçidini Kullanmayan Yayaların Yüzlerinin Tespit Edilmesindeki Başarım

Açıklama	Gerçekte olan
Gerçekte yaya geçidini kullanmayanların sayısı	85
Tespit edilen yaya geçidini kullanmayanların sayısı	79
Tespit edilen yüz sayısı	48
Tespit edilemeyen yüz sayısı	31

Tablo 5.7. Yaya Geçidine Park Etmiş Araçların Plakalarının Tespit Edilmesindeki Başarım

Açıklama	Gerçekte olan
Gerçekte yaya geçidine park eden sayısı	11
Tespit edilen yaya geçidine park eden sayısı	7
Tespit edilen plaka sayısı	4
Tespit edilemeyen plaka sayısı	3

Tablo 5.6 ve Tablo 5.7 incelendiği zaman geliştirilen modelin yaya geçidini kullanmayan yayaların ve yaya geçidine park eden araçların tespitinde oldukça başarılı olduğu görülmektedir. Ancak süreç ilgili yayaların yüzlerinin ve araç plakalarının tespit edilmesine geldiğinde bu başarım sonucu oldukça düşmektedir.

5.3. Öneriler ve Sonraki Çalışmalar

Trafikte yapılan hatalardan dolayı meydana gelen zararı azaltmanın en etkili yolu, koruyucu önlemlerin yanı sıra hatalı hareketleri hızlı bir şekilde tespit etmek ve müdahalede bulunmaktır.

İnsansız hava araçlarının bu şekilde eğitilmiş bir model ile birlikte gündelik yaşamda kullanılmasının özellikle yaya geçitlerinde meydana gelecek trafik aksamlarının önüne geçeceği ve trafiğin aksamasına sebep olan kişiler üzerinde caydırıcı bir etkiye sahip olacağı değerlendirilmektedir. Gerek önleyici bir tedbir olması gerekse hatalı araç ve şoförlerin tespitinde insansız hava araçlarının bu model ile kullanılmasıyla birlikte trafikte yaşanan aksaklıkların ve kazaların belirli bir ölçüde önüne geçilebileceği öngörülmektedir.

KAYNAKLAR

- [1] TÜİK, 2021, Emniyet Genel Müdürlüğü, *Trafik Eğitim ve Araştırma Dairesi Başkanlığı*, Trafik İstatistik Bülteni – 2019.
- [2] TÜİK, 2018, Emniyet Genel Müdürlüğü, *Trafik Eğitim ve Araştırma Dairesi Başkanlığı*, Trafik İstatistik Bülteni – 2017.
- [3] TÜİK, 2019, Emniyet Genel Müdürlüğü, *Trafik Eğitim ve Araştırma Dairesi Başkanlığı*, Trafik İstatistik Bülteni – 2018.
- [4] TÜİK, 2020, Emniyet Genel Müdürlüğü, *Trafik Eğitim ve Araştırma Dairesi Başkanlığı*, Trafik İstatistik Bülteni - 2018.
- [5] Erjem, Y., 2005, Trafik Sisteminin İşleyişi ve Trafik Kazaları Üzerine Sosyolojik Bir Araştırma, *Polis ve Sosyal Bilimler Dergisi*, 3, 1, 69-94.
- [6] Zeren, R., 2021, *Kasımda meydana gelen trafik kazalarında 180 kişi hayatını kaybetti*, <https://ilkha.com/guncel/kasimda-meydana-gelen-trafik-kazalarinda-180-kisi-hayatini-kaybetti-181047>, [Ziyaret tarihi: 28 Ocak 2022].
- [7] Antep, G., 2022, *Hatalı park kazaya neden oldu:3 yaralı*, <https://www.sabah.com.tr/gaziantep/2022/02/03/hatali-park-kazaya-neden-oldu-3-yarali>, [Ziyaret tarihi: 10 Mart 2022].
- [8] Peker, M. ve Zengin, A., 2011, Gerçek zamanlı harekete duyarlı bir görüntü tanıma sistemi, *In 6th International Advanced Technologies Symposium (IATS'11)* (pp. 16-18).
- [9] Murat, S., 2021, *İnsansız hava aracı görüntülerinden derin öğrenme yöntemleriyle nesne tanıma*, Yüksek Lisans, Maltepe Üniversitesi, Lisansüstü Eğitim Enstitüsü.
- [10] Özalp, A. N., ve Dügenci, M., 2011, Plaka Tanıma Sistemlerine Yeni Yaklaşımlar, *XIII. Akademik Bilişim Konferansı*, 2-4 Şubat 2011 Malatya İnönü Üniversitesi.
- [11] Kaplan, O., Sağiroğlu, Ş., ve Çolakoğlu, Ö. F., 2014, *Erciyes Üniversitesi Mühendislik Fakültesi*, Araç Plaka Tanıma Sistemi.
- [12] Şanlıalp, İ., 2015, *IOS ile trafik ceza bildirim uygulaması*, Yüksek Lisans, Haliç Üniversitesi, Fen Bilimleri Enstitüsü.
- [13] Şimşek, E., Özyer, B., ve Özyer, G. T., 2020, Foto-Kapan Görüntülerinde Derin Öğrenme Tabanlı İnsan Tespiti, *Bayburt Üniversitesi Fen Bilimleri Dergisi*, 3(1), 1-8.
- [14] Kaplan, A., 2018, Gerçek ve yarı gerçek zamanlı yüz tespit etme, *Face detection on real and semi-real time*.1-8.
- [15] İnce, Ö. F., İnce, İ. F., ve Park, J. S., 2017, Uzun Mesafeli Hareketli Görüntülerde Yayaların Biyometrik Özelliklerine Göre Yaşlarının Sınıflandırılması, *Bilişim Teknolojileri Dergisi*, 10(2), 123-128.
- [16] Ayvaz, U., ve Gürüler, H., 2017, Öğrencilerin Sınıf içi Duygusal Durumlarının Gerçek Zamanlı Tespit Edilmesi, *Real-Time Detection of Students' Emotional States in the Classroom*, 1-8

- [17] Karaimer, H. C., ve Bařtanlar, Y., 2014, Haar-Benzeri Öznitelikler ve Basamaklı Hızlandırma Kullanarak Tümyönlü Kameralar ile Binek Araç Tespiti, *Conference Paper*,1-4.
- [18] Zhao, C., ve Chen, B., 2019, Real-time pedestrian detection based on improved YOLO model. In 2019 *11th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)* (Vol. 2, pp. 25-28). IEEE.
- [19] Karaduman, M., 2016, *İnsansız hava araçları görüntüleri üzerinden optimum rota tespiti*, Yüksek Lisans Tezi, Fırat Üniversitesi, Fen Bilimleri Enstitüsü.
- [20] Kutlu, Ö., 2019, *İnsansız hava aracı ile elde edilen görüntülerin derin öğrenme yöntemleri ile analizi*, Yüksek Lisans Tezi, Marmara Üniversitesi, Fen Bilimleri Enstitüsü.
- [21] Zollinger, H., 1999, *Color: A Multidisciplinary Approach*, Verlag Helvetic Chimica Acta, Postfach, CH8042 Zürich, Switzerland.
- [22] Iřık, M., 2017, *Renk körlüğünün tanısına yönelik yeni bir arayüz*, Doktora Tezi, Sakarya Üniversitesi, Fen Bilimleri Enstitüsü.
- [23] Yılmaz, İ., 2002, *Renk Uzayları ve Dönüřüm Algoritmaları*. Doktora Tezi, Selçuk Üniversitesi, Fen Bilimleri Enstitüsü.
- [24] Gonzales R. C., Woods R. E., 1993, *Digital Image Processing*, Addison-Wesley Publishing Company.
- [25] Deniz, A. 2003, *Morfolojik Görüntü İşleme*, Yüksek Lisans Tezi, Yıldız Teknik Üniversitesi, Fen Bilimleri Enstitüsü.
- [26] Perihanođlu, G. M., 2015, *Dijital görüntü işleme teknikleri kullanılarak görüntülerden detay çıkarımı*, Yüksek Lisans Tezi, İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü.
- [27] Küçükpara, Z., 2019, *Görüntü işleme yöntemi ile optik işaret tanıma ve değerlendirme sistemi*, Yüksek Lisans Tezi, Necmettin Erbakan Üniversitesi, Fen Bilimleri Enstitüsü.
- [28] opencv.org, 2022, *Image Thresholding*, <https://docs.opencv.org>, [Ziyaret tarihi: 28 Ocak 2022].
- [29] Bayraktar, O., Özdemir, F., Çetin, Ö., ve Yılmaz, G., 2012, İnsansız Hava Araçları İçin Otonom İniř Sistemi Simülatörü Tasarımı, *Biliřim Teknolojileri Dergisi*, 5(2), 1-8.
- [30] Kahveci, M., ve Nazlı, C. A. N, 2017, İnsansız Hava Araçları: Tarihçesi, Tanımı, Dünyada ve Türkiye'deki Yasal Durumu, Selçuk Üniversitesi Mühendislik, *Bilim ve Teknoloji Dergisi*, 5(4), 511-535., Selcuk Univ. J. Eng. ,Science Technol., vol. 5, no. 4, pp. 511–535, 2017.
- [31] Yücel, P., 2020, *Sabit kanatlı insansız hava araçlarında çoklu otopilot sistemleri*, Yüksek Lisans Tezi, TOBB Üniversitesi, Fen Bilimleri Enstitüsü.

- [32] Chan, P.K.M., Li, C.K, 2000, Motions of Multiple Objects Detection based on Video Frames, *IEEE International Symposium on Consumer Electronics*, Hong Kong, s.62-66.
- [33] Daş, R., Polat, B., ve Tuna, G., 2019, Derin öğrenme ile resim ve videolarda nesnelerin tanınması ve takibi. *Fırat Üniversitesi Mühendislik Bilimleri Dergisi*, 31(2), 571-581.”
- [34] Tombak, M. E., 2019, *Python ve OpenCv ile yüz tanıma ve otomatik Blur uygulaması*, Yüksek Lisans Tezi, Kırıkkale Üniversitesi, Fen Bilimleri Enstitüsü.
- [35] Yanci, M., 2020, *Derin Öğrenme Yöntemleri İle Medikal Görüntülerde Kanserli Doku Tespiti*, Yüksek Lisans Tezi, Marmara Üniversitesi, Fen Bilimleri Enstitüsü.
- [36] Redmon, J., ve Farhadi, A, 2018, *Yolov3: An incremental improvement*, arXiv preprint arXiv:1804.02767.
- [37] Redmon, J., Divvala, S., Girshick, R., ve Farhadi, A, 2016, You only look once: Unified, real-time object detection, *In Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 779-788).
- [38] Kılıç, B., 2020, *Panorama ile üretilen plevral efüzyon sitopatoloji görüntüleri üzerinde yolov3 ile otomatik çekirdek algılama*, Yüksek Lisans Tezi, Karadeniz Teknik Üniversitesi, Fen Bilimleri Enstitüsü.
- [39] Çan, Z., ve Taşçı, T. *Market-Fiyat Analizi için Orc Tabanlı Mobil Uygulama*, https://www.academia.edu/49021680/Market_Fiyat_Analizi_%C4%B0%C3%A7in_OCR_Tabanlı%C4%B1_Mobil_Uygulama_2019_, [Ziyaret tarihi: 12 Mart 2022].
- [40] Smith, R., 2007, An Overview of the Tesseract OCR Engine, *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)* 629-633.
- [41] Taş, Y., 2018, *Lazer kesim tezgâhları için görüntü işleme tabanlı kontrol ve güvenlik sistemi*, Yüksek Lisans Tezi, Erciyes Üniversitesi, Fen Bilimleri Enstitüsü.
- [40] Jones, P., Viola, P., ve Jones, M., 2001, *Rapid object detection using a boosted cascade of simple features*, In University of Rochester. Charles Rich.
- [41] Kaplan, A., 2018, *Gerçek ve Yarı Gerçek Zamanlı Yüz Tespit Etme*, Doctoral dissertation, Yüksek Lisans Tezi, Fırat Üniversitesi, Fen Bilimleri Enstitüsü.
- [42] Taunk, P., Jayasri, G., Priya, J. P., ve Kumar, N. S., 2020, *Face detection using Viola Jones with Haar cascade*. Test Engineering and Management, 83.

ÖZGEÇMİŞ

Kişisel Bilgiler	
Adı Soyadı	Eyüp Ensar YILDIZ
Doğum Yeri	
Doğum Tarihi	
Uyruğu	<input checked="" type="checkbox"/> T.C. <input type="checkbox"/> Diğer:

Eğitim Bilgileri	
Lisans	
Üniversite	Zonguldak Karaelmas Üniversitesi
Fakülte	Karabük Teknik Eğitim Fakültesi
Bölümü	Bilgisayar Öğretmenliği
Mezuniyet Yılı	2009

Yüksek Lisans	
Üniversite	Ahi Evran Üniversitesi
Enstitü Adı	Fen Bilimleri Enstitüsü
Anabilim Dalı	İleri Teknolojiler
Programı	Tezli Yüksek Lisans Programı
Mezuniyet Tarihi	2022

Makale ve Bildiriler	