

T.C.
GEBZE TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

KORUYUCU BARET KULLANIM İHLALLERİNİN TESPİTİ

CANİBEK SAĞIN
YÜKSEK LİSANS TEZİ
ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI

GEBZE
2019

T.C.
GEBZE TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

KORUYUCU BARET KULLANIM
İHLALLERİNİN TESPİTİ

CANİBEK SAĞIN
YÜKSEK LİSANS TEZİ
ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI

DANIŞMANI
DR. ÖĞR. ÜYESİ KÖKSAL HOCAOĞLU

GEBZE
2019

T.R.
GEBZE TECHNICAL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

**DETERMINATION OF VIOLATIONS OF
PROTECTIVE HELMET USAGE**

CANİBEK SAĞIN
**A THESIS SUBMITTED FOR THE DEGREE OF
MASTER OF SCIENCE**
DEPARTMENT OF ELECTRONIC ENGINEERING

THESIS SUPERVISOR
ASSIST. PROF. DR. KÖKSAL HOCAOĞLU

GEBZE
2019

GTÜ Fen Bilimleri Enstitüsü Yönetim Kurulu'nun 26/06/2019 tarih ve 2019/29 sayılı kararıyla oluşturulan jüri tarafından 03/10/2019 tarihinde tez savunma sınavı yapılan Canibek Sağın'ın tez çalışması Elektronik Mühendisliği Anabilim Dalında YÜKSEK LİSANS tezi olarak kabul edilmiştir.

JÜRİ

ÜYE

(TEZ DANIŞMANI) : DR. ÖĞR. ÜYESİ A. KÖKSAL HOCAOĞLU

ÜYE

: DR. ÖĞR. ÜYESİ SULTAN ALDIRMAZ ÇOLAK

ÜYE

: DOÇ. DR. KORAY KAYABOL

ONAY

Gebze Teknik Üniversitesi Fen Bilimleri Enstitüsü Yönetim Kurulu'nun
...../...../..... tarih ve/..... sayılı kararı.

ÖZET

Tehlikeli çalışma alanlarında çalışan kişiler için koruyucu baret kullanımı hayati öneme sahiptir ve kullanımları yasal bir zorunluluktur. Baret kullanılması gereken alanlarda çalışanların bu kuralı ihlal edip etmediklerinin tespiti görüntü tanıma teknikleri kullanılarak yapılabilmektedir. Bu çalışmada, derin öğrenme yöntemi ile gerçek zamanlı nesne tanıma modeli kullanılarak baret kullanım ihlali tespiti problemine bir çözüm geliştirilmiştir. Geliştirilen algoritma güvenlik kameralarından alınan video görüntüleri üzerinde gerçek zamanlı olarak çalışabilmektedir ve iki temel aşamaya sahiptir. İlk aşamada, bilinen bir veri seti ile eğitilmiş derin öğrenme modeli kullanılarak video karesindeki kişilerin yerleri tespit edilmektedir. Kişilerin bulunduğu bölgeler kırılarak oluşturulan görüntüler ikinci aşama için giriş olarak uygulanmaktadır. İkinci aşamada ise bu çalışma için oluşturulan özgün veri seti ile eğitilen You Only Look Once (YOLO) olarak adlandırılan gerçek zamanlı nesne tespiti modeli kullanılarak kişilerin baret kullanıp kullanmama durumu tespit edilmektedir. Önerilen yöntem, test setinde baretli sınıf için %96, baretsiz sınıf için ise %97 oranında doğru sınıflandırma başarısına sahiptir.

Anahtar Kelimeler: Derin öğrenme, Baret tanıma, YOLO.

SUMMARY

Usage of protective helmets is vital for persons, who working in hazardous working areas and also it is a legal obligation. In areas, where helmets should be used, it is possible to determine whether the employees violate this rule by using image recognition techniques. In this study, a solution to the problem of helmet usage violation detection was developed by using deep learning based real time object detection model. The developed algorithm can work in real time on video footage taken from security cameras and it has two basic stages. In the first stage, the locations of the people in the video frame were determined by using a deep learning model trained with a known data set. The images created by cropping the areas where the people are located were applied as an input to the second stage. In the second stage, the real-time object detection model called YOLO, which was trained with the original data set created for this study, was used to determine whether or not the helmets were used. According to the test results, the correct detection success was 96% for the helmet class and 97% for the no helmet class.

Key Words: Deep learning, Helmet detection, YOLO.

TEŐEKKÜR

Yüksek lisans eğitimim boyunca yardımları, bilgileri ve tecrübeleri ile bana sürekli destek olan, tez çalışmam boyunca ilgisini, yönlendirmelerini esirgemeyen danışmanım Dr. Öğr. Üyesi Köksal HOCAOĞLU hocama, manevi destekleri ve bana olan inançları için aileme, her anımı paylaştığım, her zorlukta yanımda olan sevgili eşim Ayşe Nur SAĞIN'a en içten teşekkürlerimi sunarım.

İÇİNDEKİLER

	<u>Sayfa</u>
ÖZET	v
SUMMARY	vi
TEŞEKKÜR	vii
İÇİNDEKİLER	viii
SİMGELER ve KISALTMALAR DİZİNİ	x
ŞEKİLLER DİZİNİ	xi
TABLolar DİZİNİ	xii
1. GİRİŞ	1
2. BARET KULLANIMI	3
2.1. Koruyucu Baretin Çalışan Sağlığı ve Güvenliği İçin Önemi	4
2.1.1. Yasal Zorunluluklar	5
2.2. Koruyucu Baret İhlali Tespiti İle İlgili Yapılan Çalışmalar	6
2.3. Nesne Tanıma	10
2.3.1. Makine Öğrenmesi	10
2.3.2. Derin Öğrenme	11
2.3.3. Evrişimsel Sinir Ağları	12
2.3.3.1. Evrişimsel Katman	13
2.3.3.2. Etkinleştirme Katmanı	13
2.3.3.3. Örnekleme Katmanı	14
2.3.3.4. Tam Bağlantı Katmanı	15
2.4. Derin Öğrenme Tabanlı Nesne Tanıma Sistemleri	16
2.4.1. Faster R-CNN	16
2.4.2. SSD	17
2.4.3. YOLO	18
3. BARET İHLALİ TESPİTİ İÇİN ÖNERİLEN YAKLAŞIM	22
3.1. YOLO Uygulaması	22
3.1.1. YOLO'nun Derlenmesi	23
3.1.2. Veri Setinin Oluşturulması	24

3.1.2.1. Etiketleme İşlemi	24
3.1.2.2. YOLO Etiket Dosyası Formatı	25
3.1.2.3. Pascal VOC Etiket Dosyası Formatı	26
3.1.3. Eğitim İçin YOLO'nun Konfigürasyonu	27
3.1.4. Veri Setinin Eğitimi	29
3.1.5. Eğitim Sonuçları	30
3.2. OpenCV	32
3.2.1. OpenCV'nin Temel Modülleri	32
3.2.2. OpenCV ve Derin Öğrenme	33
3.3. Problemin Çözümü	34
4. DENEYLER VE SONUÇLAR	38
4.1 Algoritmanın Testi	39
4.2 İnsansız Ortam Testi	41
4.2.1 İnsansız Ortamda Birinci Aşamanın Testi	42
4.2.2 İnsansız Ortamda İkinci Aşamanın Testi	43
4.3 İnsanlı Ortam Testi	46
4.4 Orijinal Test Setini Kullanarak İkinci Aşamanın Testi	48
4.5 Deney Sonuçlarının Değerlendirilmesi	50
5. SONUÇ VE ÖNERİ	51
KAYNAKLAR	52
ÖZGEÇMİŞ	55

SİMGELER ve KISALTMALAR DİZİNİ

Simgeler ve Açıklamalar

Kısaltmalar

CNN	:	Evrişimsel Sinir Ağları (Convolutional Neural Networks)
KKD	:	Kişisel Koruyucu Donanım
YSA	:	Yapay Sinir Ağı
RFID	:	Radyo Frekansı ile Tanımlama (Radio Frequency Identification)
HOG	:	Odaklı Gradient Histogram (Histogram of Oriented Gradient)
SVM	:	Destek Vektör Makinesi (Support Vector Machine)

ŞEKİLLER DİZİNİ

<u>Şekil No:</u>	<u>Sayfa</u>
2.1: Kang Li çerçeve mimarisi.	7
2.2: Bilgisayarlı görme çözüm önerisi.	8
2.3: Bilgisayarlı görme tespit işlemi sonucu.	9
2.4: Temel YSA modeli.	11
2.5: Derin öğrenme ve yapay zekâ ilişkisi.	12
2.6: Evrişim işlemi.	13
2.7: ReLU etkinleştirme fonksiyonu.	14
2.8: Maksimum örnekleme.	14
2.9: Tam bağlantı katmanı.	15
2.10: Düzleştirme işlemi.	16
2.11: R-CNN'in yapısal görünümü.	17
2.12: SSD mimarisi.	18
2.13: YOLOv3 performansı.	18
2.14: Özellik haritası ve tahmin vektörü.	20
2.15: Modelin tahmin işlemi.	21
3.1: LabelImg programı ara yüzü ve etiketleme işlemi.	25
3.2: YOLO etiket dosyası örneği.	26
3.3: XML etiket dosyası örneği.	26
3.4: Transfer öğrenimi ve eğitimin performansı.	29
3.5: MS COCO veri setindeki bazı kategorilerin örnek sayıları.	35
3.6: Çözümün blok şeması.	36
3.7: Algoritmanın akış diyagramı.	37
4.1: Başarılı test sonucu örnekleri.	40
4.2: Yanlış sınıflandırma örnekleri.	41
4.3: Sonuç alınamayan örnekler.	41
4.4: Birinci aşamanın hatalı test sonucu örnekleri.	43
4.5: İkinci aşamanın hatalı test sonucu örnekleri.	45
4.6: İnsanlı ortam testi hatalı sonuç örnekleri.	47
4.7: Farklı obje hatası örnekleri.	49

TABLÖLAR DİZİNİ

<u>Tablo No:</u>	<u>Sayfa</u>
3.1: Üretilen bazı ağırlıkların doğruluk seviyeleri.	31
4.1 Algoritmanın test sonucu.	39
4.2 Birinci aşamanın test sonucu.	42
4.3 İkinci aşamanın test sonucu.	43
4.4 İnsanlı ortam testi sonucu.	46
4.5 Orijinal test seti ile ikinci aşamanın testi.	48

1. GİRİŞ

Tehlikeli çalışma alanlarında çalışan personelin kişisel koruyucu donanım (KKD) kullanması yasal bir zorunluluktur. İş güvenliği açısından KKD kullanımını düzenleyen yasal mevzuat mevcuttur [1]. Bir veya birden fazla sağlık ve güvenlik riskine karşı çalışanları korumak için tasarlanmış giyilmek, takılmak veya taşınmak amacıyla kullanılan herhangi bir cihaz, alet veya malzemeye KKD denir. Baret, KKD kapsamı içerisinde yer alır ve tehlikeli çalışma alanlarında herhangi bir kaza anında çalışanın baş bölgesini darbelere, cisim çarpmasına ve elektrik çarpmasına karşı koruyan güvenlik malzemesidir. Ülkemizde 2004-2016 yılları arasında sadece inşaat sektöründe meydana gelen iş kazası sayısı 215,949'dür. Bu kazalarda hayatını kaybeden çalışan sayısı ise 5647'dir [2]. İnşaat sektörü dışındaki diğer tehlikeli çalışma şartlarının olduğu madencilik, imalat vb. gibi sektörler de düşünüldüğünde iş güvenliği için koruyucu baret kullanımının önemi ortaya çıkmaktadır. Amerika Birleşik Devletleri işçi istatistikleri bürosunun verilerine göre iş kazası sonucu oluşan travmatik beyin yaralanmalarının %90'ı koruyucu baret kullanımı ile önlenilecek türden yaralanmalar olduğunu göstermektedir [3]. İnşaat alanlarında çalışan personelin baret kullanımının hayati önemine yönelik farkındalıklarının düşük olduğu gözlemlenmektedir. Gerek ağırlığı sebebiyle ve gerekse sıcak havalarda serinlemek için koruyucu baret çıkarma davranışları çalışanlar arasında sıklıkla görülür [4].

Dünyada ve ülkemizde çalışma alanında personelin koruyucu baret kullanım ihlalleri tespitinin teknolojidene yararlanılarak yapılması günümüzün önemli problemlerinden biridir. Bu problemin çözümü için sensör tabanlı tespit veya görüntü üzerinden tespit olmak üzere iki kategoride çözüm metotları bulunmaktadır.

Bu çalışmada, derin öğrenme yöntemi ile gerçek zamanlı nesne tanıma modeli kullanılarak baret kullanım ihlali tespiti problemine görüntü tabanlı bir çözüm geliştirilmiştir. Önerilen algoritma, güvenlik kameralarından alınan video görüntüleri üzerinde gerçek zamanlı olarak çalışabilmektedir. Algoritma birbirini takip eden iki aşamadan oluşmaktadır. İlk aşamada, Microsoft COCO veri seti ile eğitilmiş derin öğrenme modeli kullanılarak video karesindeki kişilerin yerleri tespit edilmiştir. Kişilerin bulunduğu alanlar kırılarak oluşturulan görüntüler ikinci aşamaya giriş olarak uygulanmıştır. İkinci aşamada ise bu çalışma için oluşturulan özgün veri seti

ile eğitilen YOLO olarak adlandırılan gerçek zamanlı nesne tespiti modeli kullanılarak kişilerin baret kullanıp kullanmama durumu tespit edilmiştir.

Bu çalışmada geliştirilen özgün veri seti genel kullanıma açık olarak sunulurken baret kullanım ihlali tespiti konusunda diğer araştırmacıların çalışmasını teşvik edici ve kolaylaştırıcı olacaktır. Önerilen yöntem, test setinde baretli sınıf için %96, baretsiz sınıf için ise %97 oranında doğru sınıflandırma başarısına sahiptir. Veri setinin genel kullanıma açık olması, önerdiğimiz yöntemin iyileştirilmesine de katkı sunacaktır.

Bu tezin ikinci bölümünde baret kullanımına yönelik yasal mevzuat özetlenmekte ve baret kullanım ihlaline yönelik daha önce yapılmış çalışmalar hakkında bilgi verilmektedir. Ülkemizdeki yasal düzenlemeler ve zorunluluklardan bahsedilip, baret kullanımının önemini altı çizilmiş, gerçek zamanlı obje tanıma sistemleri üzerinde durulmuş ve bu çalışmada önerilen yöntemin birinci aşamasında kullanılan derin öğrenme mimarisinin detayları verilmiştir. Önerilen algoritmanın adımları anlatılmıştır. Üçüncü bölümde seçilen mimarinin konfigürasyonu ve eğitim aşamaları anlatılmıştır. OpenCV'nin derin öğrenme mimarilerinin kullanımına sunduğu destek açıklanmıştır. Dördüncü bölümde, bu çalışmada oluşturulan özgün veri setinin genel özelliklerinden bahsedilmiş ve önerilen algoritmanın performansı değerlendirilip sonuçların analizi sunulmuştur. Beşinci bölümde ise elde edilen sonuçlar irdelenmiş, çalışma üzerinde gelecekte ne gibi iyileştirmelerin yapılabileceği tartışılmıştır.

2. BARET KULLANIMI

KKD yürütülen işten kaynaklanan, çalışanın sağlığı ve güvenliğini etkileyen bir veya birden fazla riske karşı koruma sağlayan, çalışan tarafından giyilen, takılan veya tutulan, bu maksada uygun olarak tasarımı yapılmış tüm alet, araç, gereç ve cihazları ifade eder.

KKD hem kaza yaşanmadan önce tedbir almak hem de kazanın oluşmasını engellemek için iş sağlığı ve güvenliği adına önemli bir uygulamadır. KKD sayesinde herhangi bir iş kazası, önceden önlenilmekte, ya da herhangi bir kaza anında işçinin bu kazayı en az hasarla atlatabilmesi sağlanmaktadır.

Kişisel koruyucu malzemelerin kullanım amaçları şöyle sıralanabilir;

- Baş bölgesinin korunması
- Göz ve çevresinin korunması
- Çalışanın işitme duyusunun korunması
- Çalışanın solunum sisteminin korunması
- Çalışanın hijyen şartlarının sağlanması

Ülkemizde büyüklük bakımından lider sektörlerden birisi inşaat sektörüdür. Bu sektör özelinde yapılan araştırmalar, diğer sektörlerle göre bu sektör çalışanlarının iş kazası sonucu iki kat daha fazla sakatlanma ve üç kat daha fazla ölüm riskinin bulunduğunu göstermiştir. İnşaat sektörü çalışanlarının çalışma esnasında maruz kaldıkları başlıca olumsuzluklar gürültü, titreşim, sıcaklık, biyolojik mikroplar, kimyasal maddeler ve ergonomik yoksunluklardır. Yine bu sektör çalışanlarında sık karşılaşılan mesleki hastalıklar asbestoz, mezotelyoma, kas-iskelet hastalıkları, dermatit, işitme kaybı ve el-kol titreşim sendromu olarak sayılmaktadır. Sektörler arası yapılan incelemede, inşaat sektörünün ülkemizde hem ölümlü hem de sürekli iş göremezlik ile sonuçlanan iş kazası sıklığının en fazla yaşandığı sektör olduğu görülmektedir. SGK verilerine göre inşaat sektöründe gerçekleşen iş kazalarının, ülkemizde gerçekleşen tüm iş kazalarının içerisindeki oranı yaklaşık %9, sürekli iş göremezlik ile sonuçlanan tüm iş kazaları içerisindeki oranı %18 ve ölümlü iş kazaları içerisindeki oranı da %28 dir [2].

Eldiven kullanılmadığı için oluşan el kesikleri, gözlük kullanılmadığı için göze çapak kaçması, çelik tabanlı iş ayakkabısı verilmediği için ayağa çivi batması, emniyet kemeri takılmadığı için yüksekten düşme ve baret takılmadığı için başa malzeme düşmesi veya baş çarpması vb. şantiyelerde birincil riskli kazalardan bir kaçıdır. Oysa KKD kullanımı, toplu koruma sağlanamadığı hallerde, bireysel koruma için alınması gerekli önlemler arasındadır ve bu tarz yüksek riskli kazalara karşı çalışanın son koruma tedbiridir [5].

Bu bilgiler ışığında KKD kullanılmama durumlarında en hayati yaralanma riskine sahip KKD'lerden birinin koruyucu baret olduğu görülmektedir. Baret kullanımının önemsenmediği çalışma alanlarında çalışanların güvenliği riskli bir hale gelmektedir. Bu durum koruyucu baret kullanım ihlalinin otomatik olarak tespitini zorunlu hale getirmektedir.

2.1. Koruyucu Baretin Çalışan Sağlığı ve Güvenliği İçin Önemi

Baretler başı vurmalara karşı, yukarıdan düşen objelere karşı, düşme sonucu yaralanmalara karşı; hatta yağmur ya da soğuk gibi iklimsel olaylara karşı korumaktadır. Örneğin; araçların altında çalışırken emniyet kaskı giyilmesi, aracın altındaki çıkıntılara vurulması ya da kısmen de araçtan sıvılar ve metal parçalar düşmesi ihtimallerine karşı koruma sağlar. Baretin başa tam oturması güvenlik açısından gereklidir. İş güvenliği açısından baretlere başka objelerin eklenip çıkarılabilmektedir. Örneğin kulakların gürültüden korunması amacıyla kullanılan kulaklık barete takılabilmekte veya bazı iş kollarında baretlerin üzerine lamba monte edilebilmektedir.

İş sağlığı ve güvenliğinde baretler esas olarak yukarıdan düşen ağır cisimlere karşı veya çarpmalardan başı korumaktır. Başı elektrik çarpmalarından korumak amacıyla da baretler en etkili kişisel koruyucu ekipmanlardır. Soğuğa karşı, baretlerin içine giyilen özel şapkalar da üretilmektedir.

Baretler, güneşe karşı vücudun baş ve yüz kısmını korurken, kimyasal maddeler ve UV ışınlarına karşı koruyuculukta da büyük bir işlevsellik sunmaktadır. Makinelerle çalışan işçilerin saçlarını da makinelere kaptırma ihtimaline karşı baret kullanması gerekmektedir. Özellikle madenler, inşaatlar, demir-çelik sanayi,

endüstri sektörü gibi yüksek tehlikeli ve güvenlik gerektiren iş alanlarında çalışan kişiler için baret kullanımı hayati derecede önem taşımaktadır.

2.1.1. Yasal Zorunluluklar

İşyerlerinde sağlıklı ve güvenli çalışmayı yürütebilmek için devlete, işverene ve işçilere çeşitli görevler düşmektedir. 6331 Sayılı İş Güvenliği Kanunu'nun 19. maddesi çalışanların uymak zorunda olduğu görevlerini belirlemektedir. Çalışanlar, aldıkları eğitim ve işverenin bu konudaki hazırladığı talimatları doğrultusunda, kendilerinin ve diğer çalışanların sağlık ve güvenliklerini, yaptığı davranışlarla tehlikeye düşürmemekten sorumludur. Çalışanların, KKD kullanımı ile ilgili diğer yükümlülükleri şu şekildedir:

i) İşyerindeki tüm makine ve teçhizatı kurallara ve amacına uygun şekilde kullanmak, bunların güvenlik ekipmanlarını doğru olarak kullanmak, keyfi olarak çıkarmamak ve değiştirmemek çalışanın sorumluluğundadır.

ii) İş veren tarafından sağlanan KKD'yi amacı doğrultusunda düzgün kullanmak yine çalışanın sorumluluğundadır.

KKD yönetmeliğinde ise KKD'nin tanımı, niteliği, KKD üreticisinin sorumlulukları, işverenin sorumlulukları, dağıtıcısının sorumlulukları, kullanıcısının sorumlulukları, AB standartları gibi birçok görev ve sorumluluk ayrıntılı olarak açıklanmaktadır [1].

İş verenin yükümlülükleri şu şekildedir.

- KKD'nin kullanma süreleri belirlenirken, yapılan iş ile ilgili şartlar, riskin seviyesi, risklerle karşılaşma sıklığı, işin özelliği ve KKD'nin koruma performansı dikkate alınarak iş veren tarafından belirlenir.

- KKD'nin temini, bakımı ve uygun şartlarda muhafazası iş verenin sorumluluğundadır.

- İşveren çalışana KKD'nin hangi risklere karşı, nasıl kullanacağı konusunda bilgilendirmek iş verenin sorumluluğundadır.

- İş veren KKD kullanımı konusunda çalışanına uygulamalı eğitim verir.

- KKD'nin yeterli sayıda ve kolayca erişilecek şekilde bulundurulmalıdır.
- KKD kullanımı ile ilgili talimatlar iş veren tarafından hazırlanır. Talimatlar anlaşılır ve kolay okunabilir olmalıdır.
- KKD'nin amacına uygun olarak kullanımını sağlamak da iş verenin sorumluluğundadır.

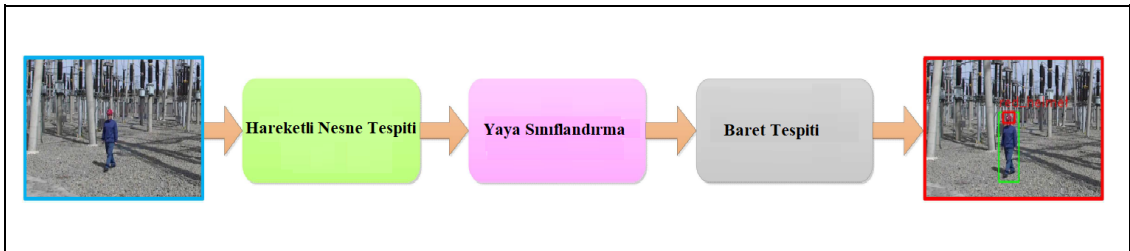
2.2. Koruyucu Baret İhlali Tespiti İle İlgili Yapılan Çalışmalar

Bu alanda yapılan çalışmaları sensör tabanlı çözümler ve görüntü tanıma tabanlı çözümler olarak iki temel kategoride değerlendirebiliriz.

Sensör tabanlı çözümler temelde işçi ve baretin yerini belirlemek için konumlandırma teknolojisi üzerinden geliştirilmiştir. Kelm, geliştirdiği çözümünde bir Radyo Frekanslı ile tanımlama (RFID) portalı tasarlamıştır [6]. KKD'ler içerisine yerleştirilen pasif RFID etiketleri yardımıyla, çalışma sahası girişinde çalışanın üzerindeki tüm KKD'lerin tam olup olmadığı kontrolü yapılmakta, çalışanın giriş saati kaydedilmekte, KKD'lerin kullanım ömürlerinin kontrolü gerçekleştirilmektedir. Bu kontrol sadece çalışma sahası girişinde yapıldığı için gerçek zamanlı bir kontrolden söz edemeyiz. Ek olarak bu çalışmada KKD'ler üzerindeki RFID etiketler ile çalışan kimliği arasındaki mesafe ölçülerek KKD'lerin çalışan üzerinde veya yakınında olup olmadığı kontrolü yapılmaktadır. Ancak bu çözüm bize KKD'lerin düzgün bir şekilde giyilip giyilmediği bilgisini vermemektedir. Barro-Torres, geliştirdiği çözümünde CPS (cyber-physical system) adını verdiği RFID ve Zigbee teknolojilerini barındıran bir çözüm sunmuştur [7]. Gerçek zamanlı olarak RFID etiketlerden alınan konumlandırma bilgisini Zigbee ile ilgili merkeze aktarılmaktadır. Ancak bu problemde KKD'lerin düzgün bir şekilde giyilip giyilmediği bilgisini verememektedir. Dong, ise Virtual Construction Technology adını verdiği bir çözüm geliştirmiştir [8]. Koruyucu baret içerisine yerleştirilen basınç sensörü yardımıyla baretin kullanım durumu tespit edilmekte, bu bilgiyi bluetooth haberleşme teknolojisi ile merkeze iletmektedir. Bu çözümde bluetooth kapsama alanının dışına çıkılması durumunda bu bilginin iletimi mümkün olmamaktadır. Bu durum bilgi kayıplarına sebep olacaktır. Bu çözümdeki diğer bir sorunlu nokta ise bluetooth ile haberleşmeyi sağlayan elektronik kartın düzenli olarak şarj edilmesi ihtiyacıdır. Çalışma sahalarında şarj ihtiyacı çözümün

uygulanabilirliğini ortadan kaldıran bir durumdur. Genel olarak sensör tabanlı tespit ve takip teknolojileri işçilerin her birinin bir etiket veya cihazı üzerlerinde bulundurmasını gerektirmektedir. Bu ciddi bir yatırım maliyeti oluşturmakta ayrıca çalışanlar giyilebilir tarzda olan takip ve tespit sistemlerinin kendi sağlıklarını ve mahremiyetlerini riske atacağını düşünebildiklerinden kullanmakta isteksiz davranabilmektedir.

Sensör tabanlı çözümlere kıyasla görüntü tanıma tabanlı çözümler sahada daha fazla uygulanabilirlik potansiyeline sahiptir. Bu çalışmada yüz özellikleri, deri rengi ve hareket tespiti verilerini kullanarak koruyucu baret kullanım durumunu tespit etmektedir [9]. Bu çalışma tüm çalışanların yüzlerini kameralara dönük olduğu varsayımı ile yapılmıştır. Bu durum çözümün uygulanabilirliğini kısıtlamaktadır. Xinhua JIANG düşük çözünürlüklü güvenlik kamerası görüntülerinde baret kullanım durumunu tespit etmek için bir çalışma gerçekleştirmiştir [10]. Düşük çözünürlüklü görüntülerde baretli ve baretsiz kişilerin görüntülerinden *Local Binary Pattern* (LBP) ve *Gray Level Co-occurrence Matrix* (GLMC) özelliklerini çıkarıp, bu özellikler ile bir YSA'yı eğitip, eğitilen bu YSA ile baretli ve baretsiz insanların tespit işlemi gerçekleştirilmiştir. Yaptığı çalışma sonucunda %94 seviyesine kadar bir başarı seviyesine kadar ulaşmıştır. Kang Li çalışmada koruyucu baret kullanım durumunu tespit eden bir yapı tasarlanmıştır. Aşağıda Şekil 2.1'de görüldüğü gibi üç aşamalı bir sistem şeklinde tasarlanmıştır [11].



Şekil 2.1 Kang Li çerçeve mimarisi.

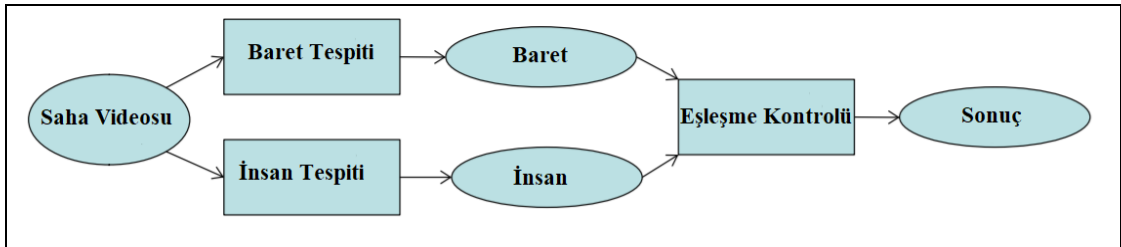
Hareketli Nesne Tespiti; bu aşamada video çerçeve'lerini arka plan bilgisi ile karşılaştırıp değişen piksellerin durumuna bakılarak hareket eden nesnelere tespit edilmektedir. Bu aşamada amaç ilgilenilen bölgenin (Region Of Interest – ROI) belirlenmesidir. Arka plan bilgisi ViBe isimli bir algoritma kullanılarak elde edilmektedir [12].

Yaya tespit; yaya tespit işlemi C4 olarak adlandırılan algoritma kullanılmıştır [13]. Bu algoritma, tüm görüntü üzerinde değil sadece hareketin olduğu bölgede koşturularak, yaya tespiti yapılmıştır.

Baret kullanım tespiti; son aşamada çalışanın baret kullanıp kullanmadığı bilgisini elde etmek için renk uzayı dönüşümü gerçekleştirilmekte ve renk özellikleri ayırımı algoritması kullanılmaktadır. Renk özellikleri ayırımı daha kolay yapılabildiği için *Red Green Blue* (RGB) olan renk uzayı *Hue, Saturation, Value* (HSV) uzayına dönüştürülmektedir. Renk özellikleri ayırımı algoritmasının İngilizce tanımı *Color Feature Discrimination* (CFD) şeklindedir.

Yapılan bu çalışmada ViBE + C4 + CFD algoritmalarının birbirini takip edecek şekilde kullanımı ile %94 doğrulukla çalışan bir algoritma elde edilmiştir.

Zhenhua Zhu, bilgisayarlı görme teknikleri kullanarak bu çalışmayı gerçekleştirmiştir [14]. Mimarisi Şekil 2.2'de görülen bu çalışma, inşaat alanında kaydedilen video görüntüleri üzerinde gerçek zamanlı olarak insan vücudu ve koruyucu baret tespitini ayrı ayrı yapıp, tespit edilen sonuçların birbiri ile eşleşme gerçekleştirip gerçekleştirmediğine bakarak, baret kullanımını hakkında bir sonuç elde edilmektedir.



Şekil 2.2 Bilgisayarlı görme çözüm önerisi.

İnsan vücudunun tespiti, iki aşamalı olarak gerçekleştirilmiştir. İlk aşamada arka plan ile olan farka bakarak hareketli objelerin olduğu bölgeler tespit edilmektedir. Hareketli nesnelere olduğu bu bölgelerde kayan pencere işlemi uygulanarak insan vücudunun tespit işlemi gerçekleştirilmiştir. Her kayan pencerede *Histogram of Oriented Gradients* (HOG) özellikleri çıkarılıp ardından bunlar *Support-vector machine* (SVM) ile sınıflandırma işlemine tabi tutularak pencere içerisinde insan olup olmadığı bilgisi elde edilmiştir. Kullanılan SVM insan vücudu pozitif örnekleri olan fotoğraflar ile içerisinde insan olmayan negatif fotoğraflardan oluşan bir veri seti kullanılarak denetimli olarak eğitilmiştir.

Koruyucu baret tespiti insan tespitine benzer aşamalarda gerçekleştirilmektedir. Öncelikle inşaat sahasında farklı açılardan ve farklı şartlarda elde edilmiş yüzlerce baret fotoğrafı içeren bir veri setinin HOG özellikleri çıkarılıp, bu özellikleri kullanarak SVM eğitilmiştir. Eğitilen SVM derlenerek bir baret tespit modeli elde edilmiştir. Video görüntüsünün HOG özellikleri çıkarılıp baret tanıma modeline uygulayarak, baretin bulunduğu bölge tespit edilmektedir. Tespit modeli sadece HOG özelliklerini kullandığı için baretin tespit işleminde renk ile ilgili bir sınırlama meydana gelmemektedir.

Tespit edilen İnsan vücudu ve baret bilgilerini kullanarak sahadaki insanların baret kullanıp kullanmama durumları belirlenmektedir. Görüntüde insanın bulunduğu bölümde özellikle vücudunun üst kısmında bir baret tespiti yapılmamışsa o insan baret kullanmamaktadır, şeklinde işaretlenmekte ve alarm bilgisi üretilmektedir. Şekil 2.3'te sistemin ürettiği sonuç görülmektedir.



Şekil 2.3: Bilgisayarlı görme tespit işlemi sonucu.

2.3. Nesne Tanıma

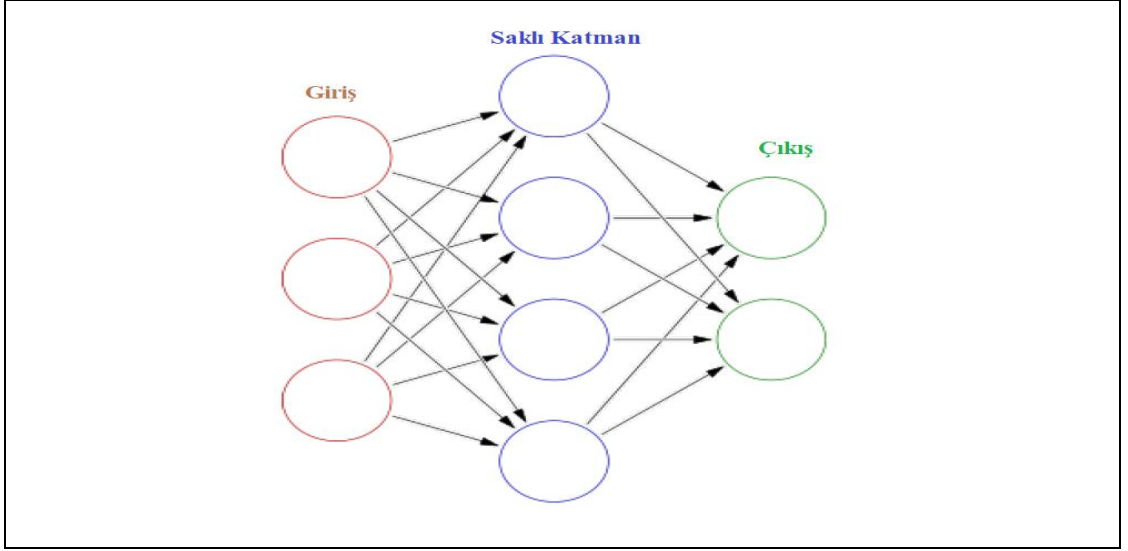
Son yıllarda bilgisayarlı görü arařtırmalarında hızlı ve başarılı bir büyüme yaşanmıştır. Bunun ardındaki temel nedenlerden ikisi makine öğrenmesi ve derin öğrenme alanlarındaki gelişmelerdir. Özellikle derin öğrenme tabanlı mimarilerin sunduğu yüksek doğrulukta çalışan sınıflandırma ve nesne tanıma algoritmaları geçtiğimiz son 4 yıla damgasını vurmuştur.

Koruyucu baret kullanım durumunun tespiti özünde tehlikeli çalışma alanlarında çözülmeye çalışılan bir nesne tanıma problemidir. Bu problem, görüntü içerisindeki kişinin kafasında bir baret ekipmanının takılı olup olmama durumunun doğru şekilde tespitidir. Yapılan bu çalışmada derin öğrenme tabanlı bir mimari olan YOLO kullanılarak çözüm geliştirildiği için bu bölümün devamında derin öğrenme teknolojinin temelleri ve derin öğrenme tabanlı nesne tanıma mimarileri anlatılmıştır.

2.3.1. Makine Öğrenmesi

Biyolojik sinir hücrelerinden esinlenerek tasarlanan, bilgisayarların giriş verilerinden öğrenmesini sağlayan programlama mimarilerine yapay sinir ağları (YSA) denir. Şekil 2.4'de temel YSA modeli görülmektedir. YSA'ların temel bileşeni algılayıcı (perceptron) olarak tanımlanan bileşendir. Lineer fonksiyonlar $y=Wx+b$ şeklinde tanımlanır. Bu fonksiyonlarda y değeri x 'in değerine bağlı olduğu için bağımlı değişken, x değeri ise herhangi bir girdi olabileceği için bağımsız değişken olarak tanımlanır. W ve b değerleri ise fonksiyonun parametreleri olarak tanımlanmaktadır.

Yapay sinir ağı modelinin en küçük parçası olan algılayıcı $y = Wx+b$ lineer fonksiyonuyla tanımlanır. Algılayıcı yapay sinir ağlarının en küçük öğrenme birimidir. Bu fonksiyonda yukarıda gösterildiği üzere W değeri ağırlık parametresi, x değeri girdi, b değeri bias ve y değeri ise ağın çıktısı olarak tanımlanmaktadır. YSA'ların eğitiminin temel amacı giriş verisine göre en iyi çıkışı verecek, w ağırlık parametreleri ve b bias değerlerinin hesaplanmasıdır.

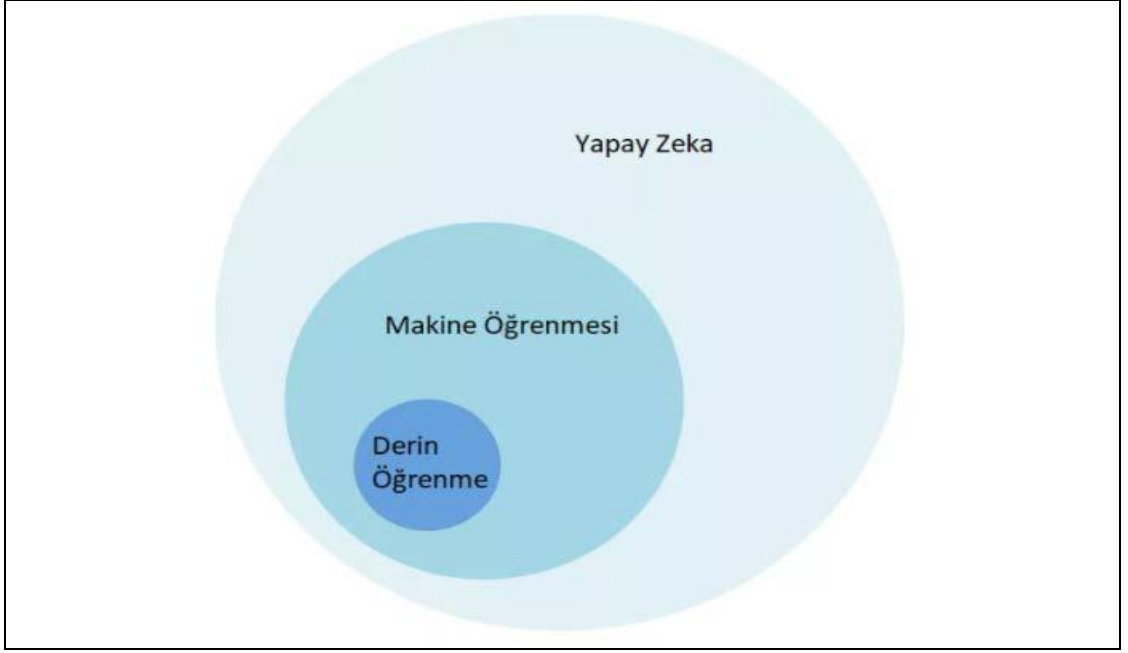


Şekil 2.4: Temel YSA modeli.

2.3.2. Derin Öğrenme

Derin öğrenme terimi 2006 yılında Hinton tarafından çok katmanlı YSA'ların verimli bir şekilde eğitilebileceği ispatlandıktan sonra literatürde sıkça kullanılmaya başlanmıştır [15]. Derin öğrenme, beynin yapısal ve işlevsel özelliklerinden esinlenerek tasarlanmış, çok katmanlı ağ yapıları üzerinde çalışan algoritmalar ve modeller kümesidir. Derin öğrenme yapay zekâ alanında yeni bir mimaridir, Şekil 2.5'de de görüldüğü gibi, makine öğrenmesinin bir alt kümesinde bulunur. Bu tanımda “Derin” ifadesi birden fazla saklı katmanı (İng. “hidden layer”) ifade eder.

Makine öğrenmesi ile bir model oluşturmak için öncelikle öznelik vektörünün elimizdeki veriden çıkartılması gerekmektedir. Özellik vektörünün çıkartılması için uzman kişilere ihtiyaç duyulmaktadır. Bu işlemler oldukça uzun sürede yapıldıklarından uzman kişilerin çalışma zamanını almaktadır. Makine öğrenme teknikleri, işlenmemiş bir bilgiyi, uzman yardımı olmadan ve ön işlem yapmadan işleyemezler. Derin öğrenme mimarisi ise öğrenme işlemini işlenmemiş veri üzerinden yapmaktadır. Ham veriyi işlerken içerisinden elde ettiği veriyi farklı katmanlarda oluşturduğu temsillerle elde etmektedir.



Şekil 2.5: Derin öğrenme ve yapay zekâ ilişkisi.

2012 yılında ImageNet nesne sınıflandırma yarışmasında bir derin öğrenme modeli olarak tasarlanan Alexnet evrişimli sinir ağının elde ettiği büyük başarı çok dikkat çekmiştir. ImageNet ILSVRC yarışmasında sınıflandırma doğruluk oranını %74,3'ten %83,6'ya yükseltmesi görüntü sınıflandırma alanında bir kırılma noktasıdır [16].

Derin öğrenmenin başlangıcı geçmiş yıllara dayansa da özellikle son zamanlarda kullanımının bu kadar yayılmasının en önemli sebeplerinden birincisi eğitim kümesi olabilecek kadar çok verinin olması, ikinci olarak bu verileri işleyebilecek ve depolayabilecek donanımların geliştirilmiş olmasıdır [17]. Verinin büyümesi ile veriden daha anlamlı bilgilere ulaşmak, öznel kestirimleriyle ilgili optimizasyon yapmayı zorunlu kılmaktadır. Görüntü üzerindeki işlemlerde bu iş için en çok kullanılan mimari evrişimsel sinir ağları (CNN) dir.

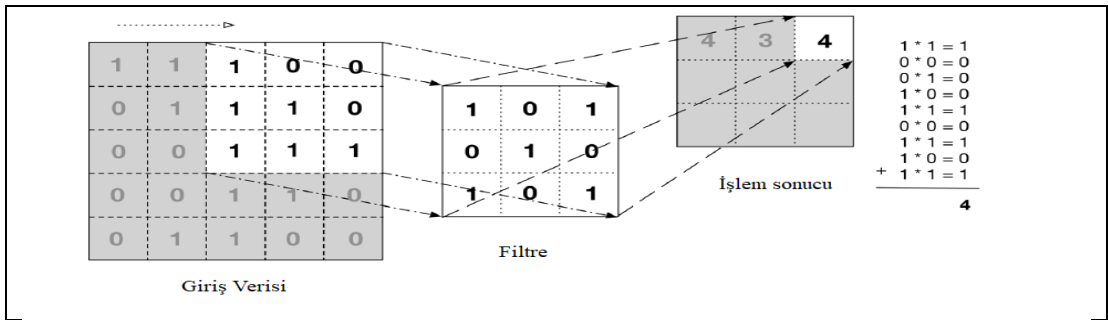
2.3.3. Evrişimsel Sinir Ağları

Evrişimsel sinir ağları (CNN), eğitilebilir ağırlıkları ve biasları olan algılayıcılardan oluşan ileri beslemeli bir yapay sinir ağı türüdür [18]. CNN ifadesi ağın evrişim matematik işlemini kullanmasını ifade eder. CNN birçok alanda kullanılmakla beraber görüntü ve fotoğraf gibi iki boyutlu verilerde üstünlüğünü

göstermiştir [19]. CNN normal ileri beslemeli sinir ağı öncesi bir anlamda öznitelik çoğaltma ve özetleme katmanlarını da içerecek şekilde, öznitelik çıkarma işlevini kendi yapısına almıştır. CNN temelde evrişimsel katmanı (İng.“convolutional layer”), etkinleştirme katmanı (İng.“activation layer”) ve örnekleme katmanı (İng.“pooling layer”) olmak üzere üç temel yapıdan oluşur. Eğer CNN sınıflandırma işlemi için kullanılacaksa tam bağlantı katmanı (İng.“fully connected layer”) sistemin sonuna eklenir.

2.3.3.1 Evrişimsel Katman

Görüntü sınıflandırma gibi uygulamalar için çok katmanlı algılayıcı mimarileri çok fazla parametre içerecekleri için verimli değildirler. Örneğin 64x64x3 (genişlik,yükseklik ve kanal sayısı) çözünürlüklü bir fotoğrafı giriş olarak alabilen bir çok katlı algılayıcı 12288 adet ağırlığa sahip olacaktır. Bunu yapmak yerine CNN fotoğraf üzerindeki uzaysal yerleşime odaklanır. CNN giriş fotoğrafı üzerinde bir filtre kaydırarak evrişim işlemi gerçekleştirir, her evrişim işlemi sonrası filtrenin kaydırılma miktarına adım kaydırma (İng.”stride”) denir. Evrişim işlemi sonucu elde edilen yeni matris ilk gizli katmana giriş olarak uygulanır. Evrişim işleminin temel işlemleri Şekil 2.6’de görülmektedir.

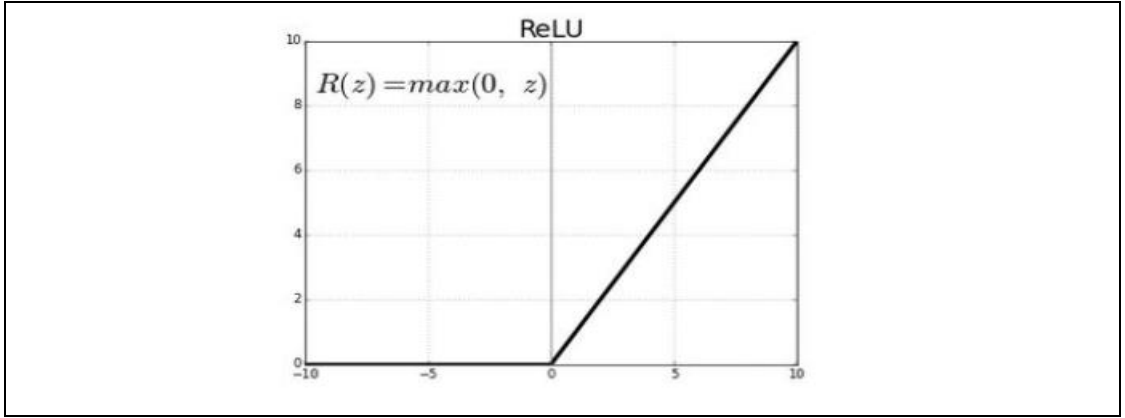


Şekil 2.6: Evrişim işlemi.

2.3.3.2 Etkinleştirme Katmanı

Evrişimsel katmandan sonra genel olarak etkinleştirme katmanı gelir. Etkinleştirme katmanının temel görevi bu mimarinin non lineer olmasını sağlamaktır. Non lineerlik YSA’ya karmaşık problemleri modelleyebilme yeteneği kazandırır.

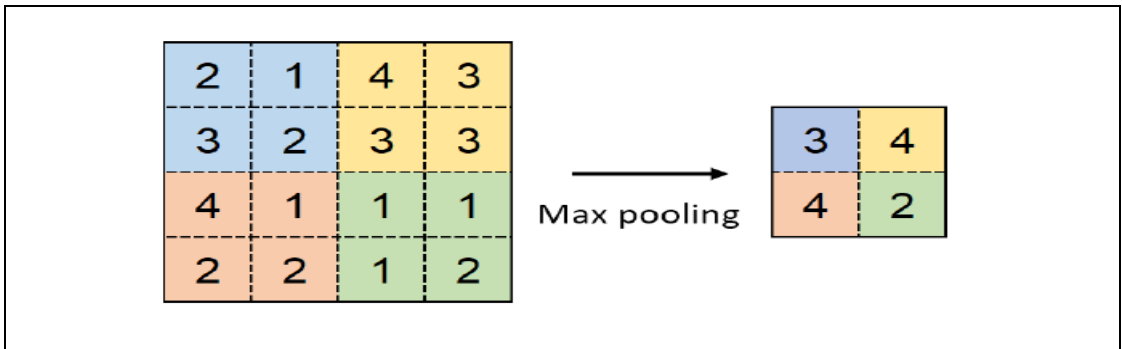
Etkinleştirme fonksiyonu olarak kullanılan bir dizi fonksiyon vardır. Örnek olarak ReLU (Rectified Linear Unit) fonksiyonu Şekil 2.7’de gösterilmiştir.



Şekil 2.7: ReLU etkinleştirme fonksiyonu.

2.3.3.3 Örneklem Katmanı

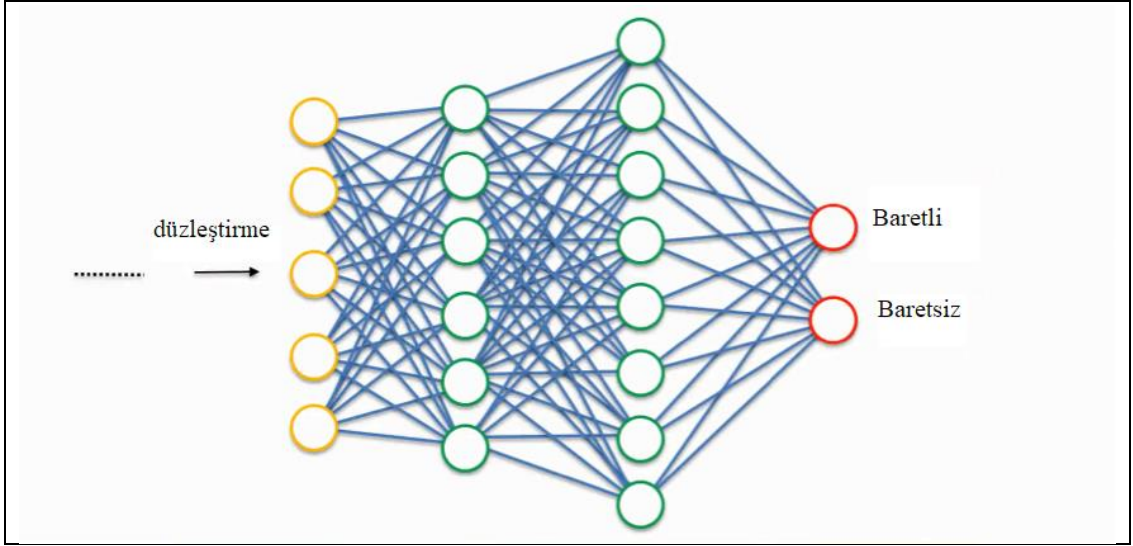
Verinin uzaysal boyutunu düşürmek için kullanılır. Genellikle 2x2 boyutunda örneklem filtresi kullanılarak bu işlem gerçekleştirilir. Bu operasyon sistemin daha sağlam ve girişteki küçük değişimlerden daha az etkilenir bir şekilde çalışmasını sağlar [20]. Örneklem işlemi, ilgili verinin öz istatistiğini çıkarır. Birkaç çeşit örneklem çeşidi vardır, ancak en çok kullanılan çeşidi, Şekil 2.8’de görülen maksimum örneklem (İng. “Max pooling”) dir.



Şekil 2.8: Maksimum örneklem.

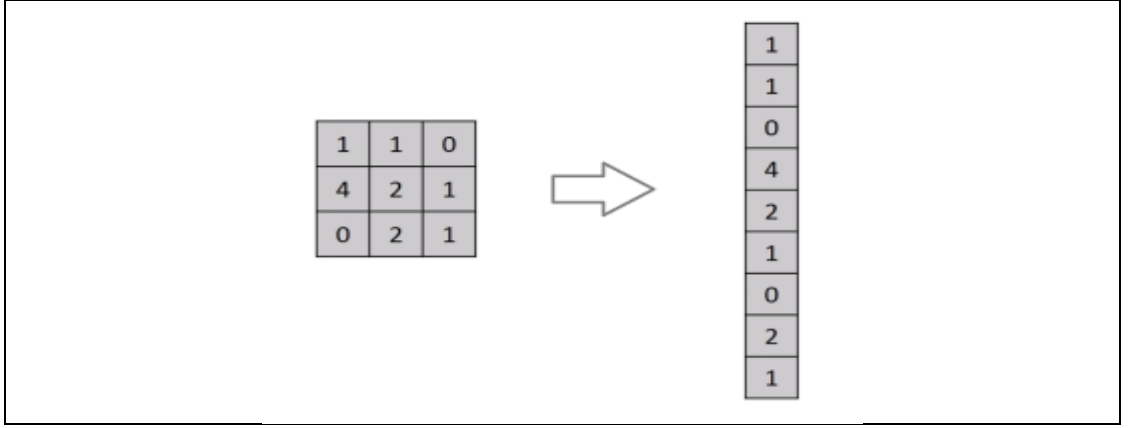
2.3.3.4 Tam Bağlantı Katmanı

Bu katmanlar, genellikle sınıflandırma için kullanılan CNN'lerin sonuna eklenir ve standart bir ileri besleme sinir ağı ile benzer yapıya sahiptir. Yani tüm girişler ve çıkışlar birbirine ağırlıklar üzerinden bağlıdır. Bu katman sınıflandırma işleminin gerçekleştirildiği katmandır. Örnekleme katmanının çıkışından alınıp düzleştirme işlemi (İng. "flattening") yapılan veri daha sonraki adımda bu katmanın girişine uygulanır. Bu katmanda bulunan ağırlıklar ve onların doğru bir şekilde eğitilmesi, sınıflandırma performansını direkt olarak etkilemektedir.



Şekil 2.9: Tam bağlantı katmanı.

Yapay sinir ağlarının, giriş verileri genel olarak tek boyutlu bir diziden alınır. Düzleştirme (İng. "flattening") işlemi örnekleme katmanından gelen matrislerin tek boyutlu dizilere çevrilmesi işlemidir. Tam bağlantı katmanı temel bir yapay sinir ağı olduğu için girişi sadece dizi alacak şekilde yapılandırılmıştır.



Şekil 2.10: Düzleştirme işlemi.

2.4. Derin Öğrenme Tabanlı Nesne Tanıma Sistemleri

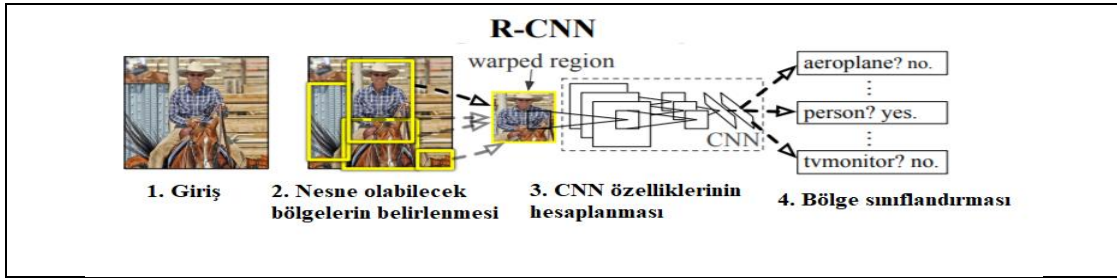
Sınıflandırma problemi ile nesne tanıma probleminin arasındaki temel farklardan biri, nesne tanıma işlemi sonunda nesnenin bulunduğu yer ve nesnenin sınıfı belirlenir, sonra nesne bir kutucuk içerisine alınır. Nesne tanıma işleminde sadece tek bir nesne değil, bazı durumlarda birden fazla birbirinden farklı nesnelere de kutucuklar içine alınabilir. Birden fazla nesne bulunabilme durumundan dolayı nesne tanıma işlemi için standart bir CNN sonuna da bir tam bağlantı katmanı kullanarak bu problem çözülememektedir. Günümüzde nesne tanıma işlemi için derin öğrenme tabanlı yaygın olarak kullanılan üç adet algoritma vardır. Bu algoritmalar Faster-RCNN, SSD ve YOLOv3 şeklindedir.

2.4.1 Faster R-CNN

Faster R-CNN bölge bazlı nesne tanıma sistemlerinin en son geliştirilen ve en hızlı çalışan örneğidir. Faster R-CNN'e gelmeden önce eski versiyonlarını açıklayarak gelişimini de görebiliriz.

R-CNN bölge bazlı nesne tanıma sistemlerinin ilk örneğidir [21]. Bu modelde sistem girişine uygulanan görüntü üzerinde seçici arama (İng. "selective search") algoritması çalıştırılarak 2000 adet nesne olabilecek bölge belirlenir [22]. Bölge belirleme işlemi sırasında sınıflandırma ile ilgili bir işlem yapılmaz sadece görüntü üzerinde nesne olabilecek bölgeler belirlenir. Bu işleme bölge teklifi (İng. "regional

proposal”) denir. Belirlenen bu bölgeler bir kare içerisine (İng. “warped region”) alınarak ayrı ayrı CNN girişine uygulanır ve sınıflandırma işleminden geçirilir.



Şekil 2.11: R-CNN’in yapısal görünümü [21].

Bu model’de görüntü üzerinde belirlenmiş 2000 adet bölgeyi tek tek CNN’den geçirip ardından sınıflandırma işlemi oldukça zaman alıcı ve yavaş olduğu için gerçek zamanlı olarak kullanılamaz. Şekil 2.11’de R-CNN’in yapısal görünümü görülmektedir.

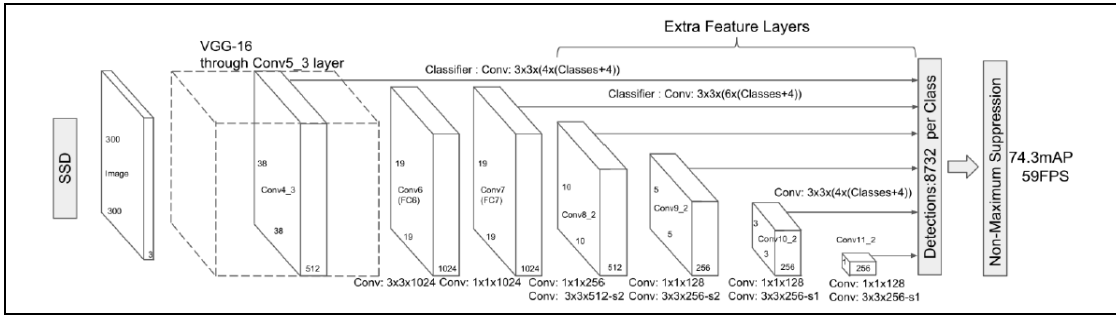
Fast R-CNN bir önceki model olan R-CNN’e göre ciddi bir hızlanma sağlamıştır [23]. Bu modelde girişteki görüntünün tamamı bir kez CNN’den geçirilir ve evrişimsel özellik haritası elde edilir. Evrişimsel özellik haritası üzerinden nesne olabilecek bölgeler seçici arama (İng. “selective search”) ile belirlenip kare içerisine alınır. Daha sonra örnekleme katmanından geçirilerek tam bağlantı katmanına uygulanabilecek hale getirilir.

Bu başlığın konusu olan Faster R-CNN mimarisine gelecek olursak, bu mimari görüntüde nesne olan bölgelerin belirlenmesi işleminde kullanılan seçici arama (İng. “selective search”) algoritmasının kullanılmasını kaldırıp, onun yerine Bölge teklif ağı (İng. “Region Proposal Network”, RPN) yapısını getirmiştir. Fast R-CNN’den farkı sadece bu değişim olsa da elde edilen hız, bize gerçek zamanlı nesne tanıma olanağı sağlamıştır.

2.4.2 SSD

SSD (Single Shot Multibox Detection) mimarisi ağ üzerinde ileri yönlü tek bir gidiş ile, nesnelerin çevresindeki kutu koordinatlarını, nesnelerin sınıfını ve sonuçların güven değerini, elde eder. Bu çıktıdaki güven değeri bir threshold ile

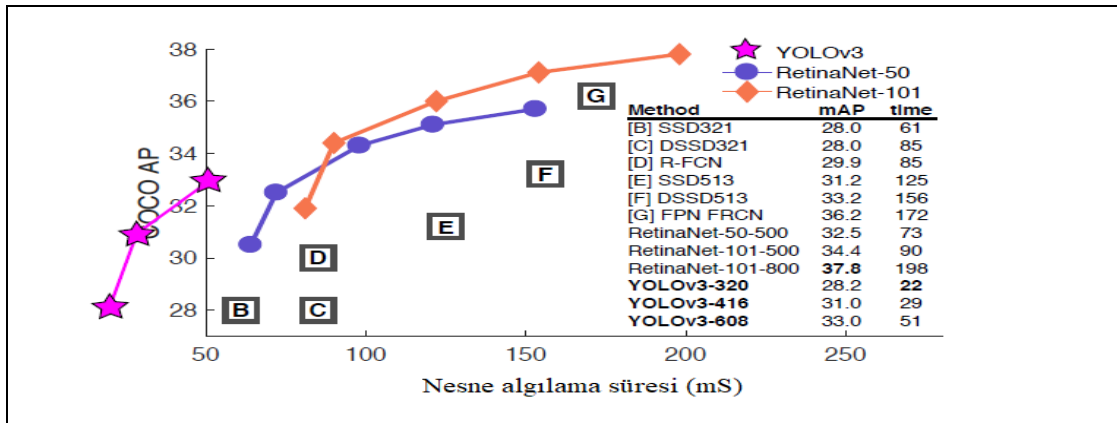
filtrelenir, ayrıca örtüşen kutu koordinatlarına non-max suppression (NMS) uygulanarak, en uygun olan değer hesaplanır diğerleri elimine edilir [24].



Şekil 2.12: SSD mimarisi [24].

2.4.3 YOLO

YOLO gerçek dünyada insan gözünün görme sistematığından ilham alınarak geliştirilmiş derin öğrenme tabanlı nesne tanıma mimarisidir. YOLO ismi *You Only Look Once* cümlesindeki kelimelerin baş harflerinin birleşmesinden oluşur. Bu cümleden de anlaşılacağı üzere sadece bir CNN kullanarak görüntünün tamamına bakarak tüm nesne tespit ve teşhis işlemlerini tek seferde gerçekleştirir. YOLO nesne sınırlayıcı kutu tahmini ve sınıflandırma için tüm resmi tek seferde işler. Bu durum resmin genelinden bağlamsal bilgi üretmesine de imkân tanır. YOLO nesne tanıma mimarisi günümüzde en hızlı çalışan ve yüksek doğrulukla çalışan mimaridir [25]. Aşağıdaki grafikte YOLO mimarisinin en son çıkan modeli olan YOLOv3'ün diğer nesne tanıma çözümleri karşısındaki üstünlüğü görülmektedir.



Şekil 2.13: YOLOv3 performansı [25].

Şekil 2.13’de de görüldüğü gibi ortalama doğruluk (İng. “Mean Average Precision”, mAP) seviyesi ile nesne algılama süresi arasında ters orantı görülmektedir. YOLOv3-320 isimli metod giriş çözünürlüğünün 320x320 olarak konfigüre edilmiş YOLOv3 modeli olduğu anlamına gelmektedir. Giriş çözünürlüğü arttıkça ortalama doğruluk artarken, algılama süresindeki uzama görülmektedir. Bu bilgi bize eğiteceğimiz YOLOv3 modelini eğitmeden önce konfigürasyon aşamasındaki, parametre seçiminin önemini göstermektedir. Çok yüksek hızlı çalışan bir model isteniyorsa ortalama doğruluk değerinden biraz fedakârlık yaparak bu yönde bir sonuç elde edebilirken, yüksek hassasiyetin önemli olduğu ve hızın çokta kritik olmadığı yerlerde ise tersi yönde bir seçim yapılabilir. YOLO algoritmasının bu kadar hızlı olmasının sebebi resmi tek bir seferde nöral ağdan geçirerek resimdeki tüm nesnelerin sınıfını ve koordinatlarını tahmin edebiliyor. Yani bu tahmin işleminin temeli, nesne tespitini tek bir regresyon problemi olarak ele almalarında yatıyor.

YOLO girdi görüntüsünü SxS ızgaraya böler, eğer konfigürasyon dosyasında çözünürlük değerlerini 416x416 olarak ayarlanmışsa ve stride (her bir ızgaranın pixel değeri) değeri 32 ise, biz 13x13 lük ızgaraya sahip bir model oluşturmuşuz demektir. Her bir ızgara kendi içinde nesne olup olmadığını varsa orta noktasının içinde olup olmadığını bulmaktan sorumludur. Örneğin fotoğraf içindeki bir nesnenin merkezi 7. ızgaraya denk geliyorsa, o nesnenin çevresindeki kutucuğu ve sınıflandırma işlemini 7. ızgara yapar. Tahmin işlemi sırasında ızgara bir tahmin vektörü oluşturur. Her ızgaranın oluşturabileceği tahmin vektörü sayısı $B \times (5 + C)$ formülü ile bulunur.

B = Anchor box sayısıdır. Bu kutucuklar geometrileri önceden belirlenmiş kalıplardır, nesne etrafındaki sınır kutucuğu çizme işleminde kullanılır.

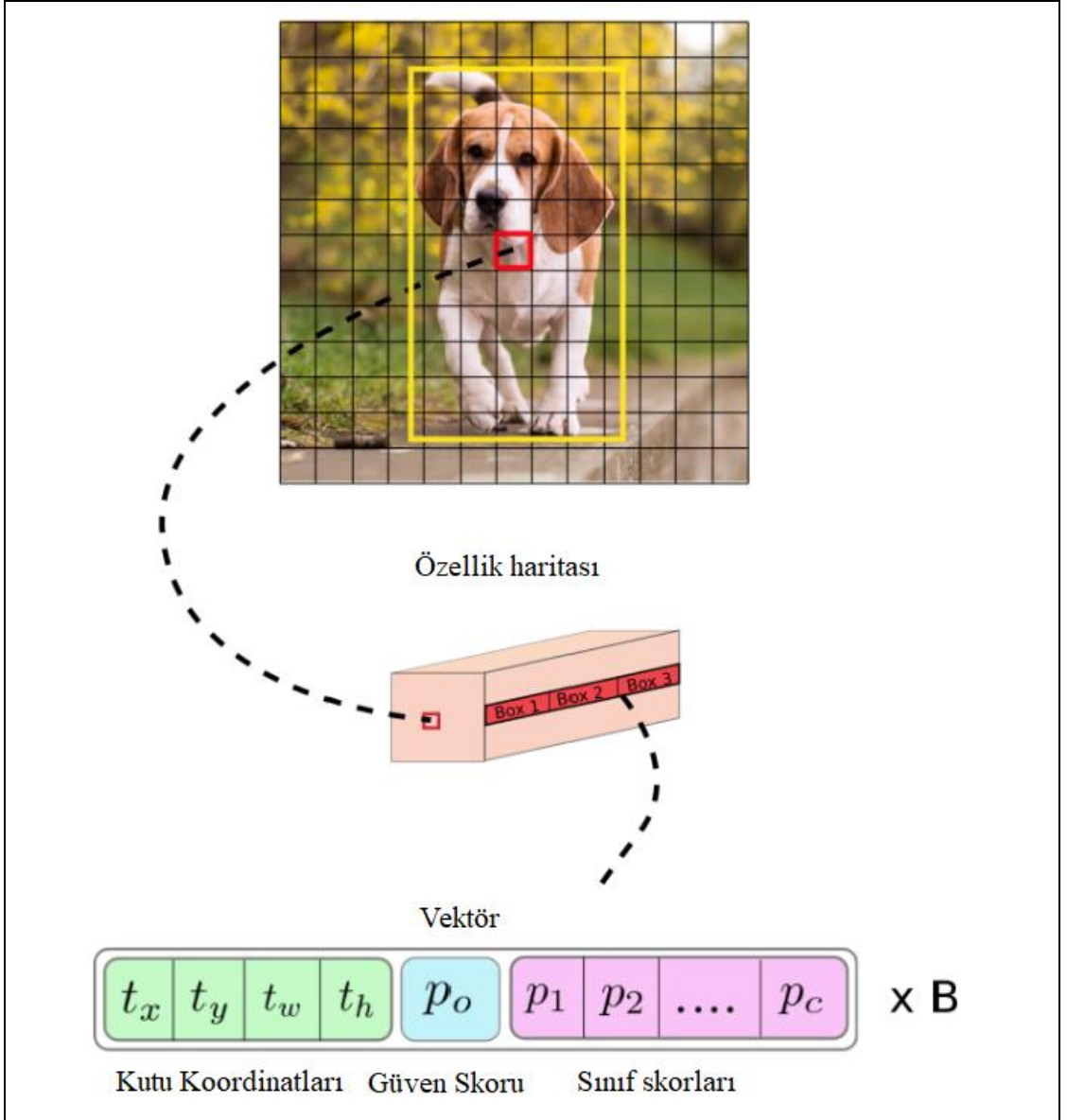
C = YOLO’nun tanıyabildiği nesne sınıfı sayısıdır. Eğittiğimiz YOLO modelinin kaç sınıfa göre eğitsek C değerimiz o olur. Vektörde her bir sınıf için bir olasılık değeri üretilir.

5 sabit sayısı ise her bir vektörde bulunan, güven skoru (P.Object), t_x, t_y, t_w ve t_h değerleridir.

- i) Güven skoru (P0): Bu skor modelin geçerli ızgara içinde nesne bulunup bulunmadığından ne kadar emin olduğunu gösterir. (0 kesinlikle bir nesne yok, 1 ise kesinlikle bir nesne var).

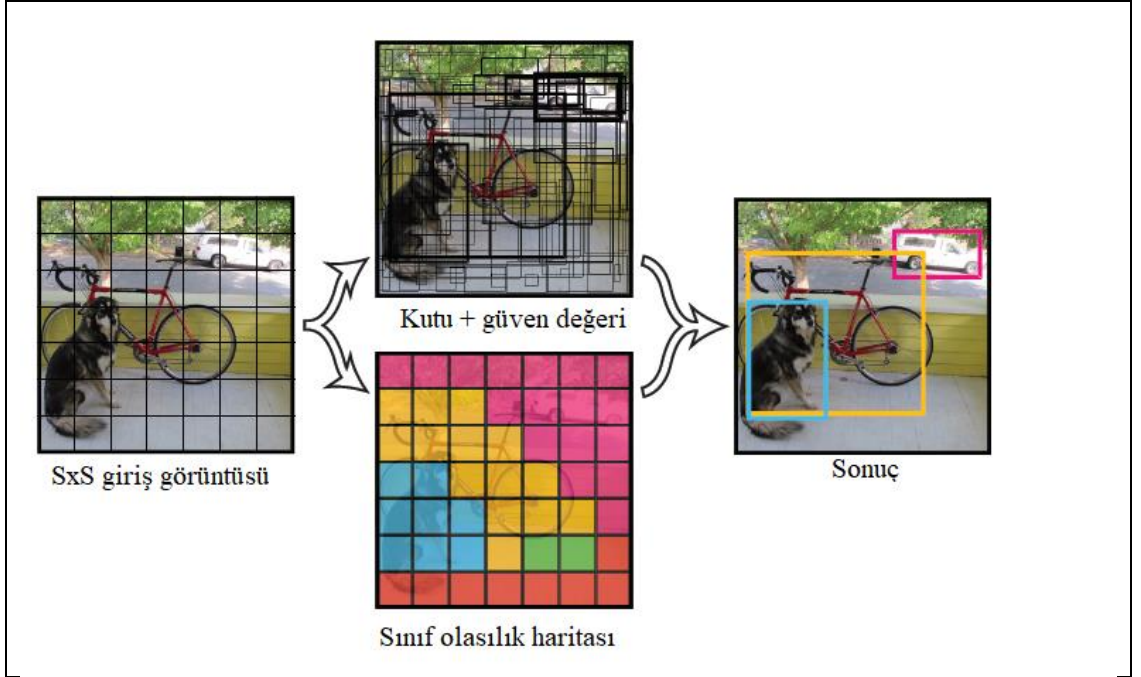
- ii) t_x : Nesnenin orta noktasının x koordinatı.
- iii) t_y : Nesnenin orta noktasının y koordinatı.
- iv) t_w : Nesnenin genişliği.
- v) t_h : Nesnenin yüksekliği.

Şekil 2.14’de 13x13 ızgaraya sahip sistemin özellik haritası ve vektör detayları görülmektedir.



Şekil 2.14: Özellik haritası ve tahmin vektörü.

Nesne tanıma işlemi sırasında algoritma girişe uygulanan her bir görüntü için $S \times S \times B$ adet tahmin vektörü üretir. Her bir tahmin vektörü bir kutu çizmekten sorumludur. Fakat ortaya çıkan bu kadar kutunun büyük bir kısmı elimine edilerek sadece yüksek güven skorlu olanlar kalır. Bu yüksek güven skor'lu kutuların içerisindeki üst üste çakışmış olanlara Non-Max Suppression (NMS) uygulanarak, çizdirilecek kutucuk sayısı iyice düşürülmüş olur. Kutu çizimi sonunda sınıflandırma işlemi, sınıf skorları içerisinde en yüksek skorlu olan seçilerek yapılır. Şekil 2.15'de işlemin aşamaları gösterilmiştir.



Şekil 2.15: Modelin tahmin işlemi.

YOLO algoritmasında hata fonksiyonlarını üç temel ana başlıkta inceleyebiliriz:

- Sınıflandırma kaybı: Tahmin edilen nesnenin ne kadar yanlış olduğu.
- Konum kaybı: Tahmin edilen kutunun ne kadar yanlış olduğu.
- Güven kaybı: Izgaranın içinde nesne olup olmadığının ne kadar yanlış olduğu

Bunların hepsi genel kaybımızı etkileyen faktörlerdir. Bu üç kayıp fonksiyonunun toplamı bize genel kayıp değerini verir.

3. BARET İHLALİ TESPİTİ İÇİN ÖNERİLEN YAKLAŞIM

Nesne tanıma işlemi nesnenin yerinin tespiti ve tespit edilen nesnenin sınıflandırılması olmak üzere iki bölümden meydana gelmektedir. Günümüzde nesne tanıma işlemi, robotik, güvenlik, tüketici elektroniği, lojistik gibi birçok alanda etkin olarak kullanılmaktadır. Derin öğrenme tabanlı nesne tanıma algoritmaları, sunduğu yüksek doğruluk performansı ve gerçek zamanlı çalışma özellikleri ile giderek bilgisayarlı görme alanında yaygınlaşmaktadır.

Bilgisayarlı görme alanında en çok kullanılan kütüphanelerden biri olan OpenCV platformu da artık derin öğrenme mimarilerini desteklemeye başlamıştır. YOLO desteklediği derin öğrenme mimarilerinden biridir. Bu çalışmada YOLO mimarisinde eğitimi gerçekleştirdiğimiz ağırlıkların, OpenCV fonksiyonlarında kullanılması ve yorumlanması ile nesne tanıma işlemi gerçekleştirilmiştir.

Bu bölümde, veri setinin tanıtımı yapılmış, etiketleme işlemi anlatılmış ve YOLO'nun baret veri seti ile eğitilmesi adım adım izah edilmiştir.

3.1. YOLO Uygulaması

YOLO açık kaynak kodlu bir nesne tanıma algoritmasıdır.Çeşitli veri setlerinin YOLO mimarisinde eğitilmiş modellerinin internette hazır olarak indirilip kullanımı mümkündür. Ancak kendi oluşturacağımız bir veri setinin eğitimi çeşitli adımlardan oluşan bir işlemdir. Bu çalışmada YOLOv3 isimli YOLO'nun 2018 yılında yayınlanan versiyonu [25] kullanıldığı için bu versiyon üzerinden işlem detayları anlatılacaktır.

Hazırlanan veri setinin eğitiminin süresi, veri setinin içerdiği örneklerin miktarı ile doğru orantılı olarak artmaktadır. Derin öğrenme mimarilerinin en temel özelliği içerdikleri büyük miktardaki verilerdir. YOLO'nun eğitim süresini kısaltmak için GPU desteğini aktif hale getirerek derleyip, GPU desteği olan bir bilgisayarda eğitim gerçekleştirilmelidir. Eğer eğitim işleminin gerçekleştirileceği bilgisayar üzerinde Nvidia marka GPU'lu bir ekran kartı varsa eğitim öncesi CUDNN ve CUDA kütüphaneleri işletim sistemine kurulmalıdır. Bu iki kütüphane Nvidia firmasının

kendi ekran kartları için geliştirdiği, eğitimin daha hızlı yapılmasını ve GPU'nun daha optimize kullanılmasını sağlayan yazılım kütüphaneleridir.

3.1.1. YOLO'nun Derlenmesi

Ubuntu veya farklı bir linux tabanlı işletim sisteminde *git* uygulaması kullanılarak YOLO'nun internette indirilmesi *git clone* komutunun linux komut satırında çalıştırılması ile gerçekleştirilir [26].

İndirme işlemi gerçekleştirildikten sonra darknet klasörüne girilerek Makefile dosyasındaki GPU ve CUDNN parametreleri "1" olarak ayarlanmalıdır. Bu ayar sayesinde GPU desteği aktif hale getirilmiştir. Eğer NVIDIA marka bir ekran kartına sahip bir bilgisayarda eğitim işlemi gerçekleştirilecekse çok daha kısa sürede veri setinin eğitimi tamamlanacaktır.

Daha fazla fotoğraf dosyası formatı desteği ve OpenCV programının kullanımı için Makefile dosyası içerisindeki OPENCV parametresi 1 olarak ayarlanmalıdır. Bu parametreyi aktif ederek derleme işlemini gerçekleştirmek için öncesinde OpenCV programının bilgisayarımıza kurulmuş olması gerekmektedir.

Tüm bu işlemlerin ardından darknet klasöründe *make* komutu çalıştırılarak derleme işlemi gerçekleştirilir. Derleme işlemi başarılı bir şekilde gerçekleştirildiğinde, GPU desteği ile daha hızlı eğitim gerçekleştirebilen bir YOLOv3 alt yapısı elde edilmiş olacaktır. Derleme işlemi tamamlandığında darknet klasörü içerisinde darknet isimli çalıştırılabilir bir dosya belirecektir. Daha sonraki aşamalarda tüm eğitim işlemleri bu dosya kullanılarak gerçekleştirilmiştir.

Eğitime geçmeden önce darknet'in internet sitesi incelenip site içerisinde tanımı yapılan çeşitli demo uygulamaları çalıştırılarak, derlenen programın düzgün çalışıp çalışmadığı kontrol edilmelidir [26].

Demo işlemi için ihtiyaç duyulabilecek çeşitli test dosyaları darknet klasöründe bulunmaktadır.

3.1.2. Veri Setinin Oluřturulması

Bu alıřmada oluřturulan veri seti, baretli ve baretsiz insanların fotoęraflarının internetten veya farklı veri setlerinin ierisinden alınması ile oluřturulmuřtur. rneęin insanların alkıřlama hareketini tanımak iin kullanılan veri setinden, insan tespiti amalı hazırlanmıř inra veri setinden alınan fotoęraflar baretsiz sınıfının tespiti iin kullanılırken. İnřaat ortamındaki baretli iřileri ieren video grntlerinden alınan rnekler, internetten elde edilen fotoęraflardan da baretli sınıfının verileri oluřturulmuřtur.

Veri seti sadece fotoęraflardan ibaret deęildir. Bu fotoęraflar zerinde yapılacak etiketleme iřlemi, fotoęraf ierisindeki nesnelere konumu ve sınıf bilgisini ieren veri dosyaları retmektedir. Bu alıřmada labelImg programı kullanarak etiketleme iřlemi gerekleřtirilmiřtir.

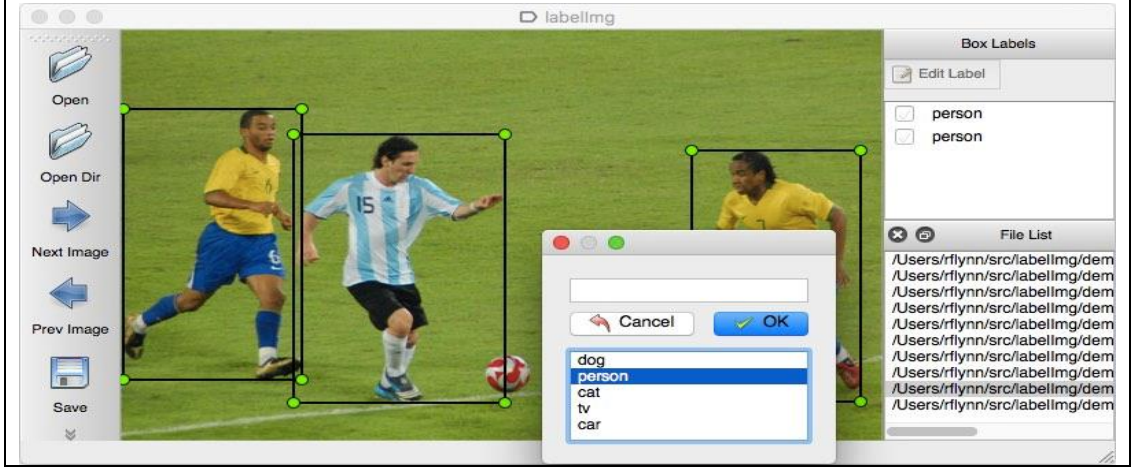
Yapılan alıřma sonucunda veri seti, baretli ve baretsiz sınıfını ieren fotoęraflar birer klasre, bu sınıfların nesne konum verilerini ieren etiket (İng. “annotation”) dosyaları da birer klasre kayıt edilerek toplam 4 klasrede oluřturulmuřtur.

3.1.2.1. Etiketleme İřlemi

Etiketleme iřlemi, tanınmasını istedięimiz nesnelere sınıflandırma ve fotoęraf ierisindeki konum bilgilerini ieren bir dosyanın retilmesidir. Bu alıřmada labelImg etiketleme programı kullanılarak etiketleme yapılmıřtır. Bu program sunduęu bir ok pratik kısa yol ve kullanıcı dostu arayz ile etiketleme iřlemini olduka kolaylařtırmıřtır. alıřma boyunca 11505 fotoęraf tek tek etiketlenmiřtir. Bir adet fotoęrafın etiketleme sresi, fotoęrafın ierdięi baretli ve baretsiz kiři sayısı ile doęrudan iliřkilidir. Fotoęraf ierisindeki etiketlenecek nesne sayısı arttıka fotoęraf bařına etiketleme sresi artmaktadır. 11505 adet fotoęrafın tek tek etiketleme iřlemi yaklaşık 90 saat srmřtir.

LabelImg programı etiketleme iřleminde hem YOLO formatında hem de Pascal VOC formatında etiket dosyası retebilmektedir [29]. YOLO formatı seildięinde ilgili fotoęrafın dosya ismi ile aynı isme sahip “.txt” uzantılı bir dosya retilerek etiketlemeye ait bilgiler kaydedilmektedir. Pascal VOC seilmesi durumunda ise, fotoęraf ile aynı isme sahip bir XML dosyası retilmektedir. Hangi

formatta etiket dosyası üretilmesi isteniyorsa labelImg programı ara yüzünden o format seçilmelidir.



Şekil 3.1: LabelImg programı ara yüzü ve etiketleme işlemi.

Etiketleme işlemi, fotoğraftaki nesnenin bir kutucuk içerisine alınması ile yapılır. İşaretlenen bölgeye ait sınıf bilgisi Şekil 3.1’de gösterildiği gibi programın ara yüzü aracılığı ile girilir. Bu çalışmada baretli ve baretsiz olmak üzere iki sınıf olduğu için etiketleme yaparken insan vücudunun baş kısmını baz alarak, baret’in takılı olup olamama durumuna göre baretli veya baretsiz şeklinde etiketleme işlemi gerçekleştirilmiştir. Yapılan çalışmada etiketleme işlemi sonucu Pascal VOC formatında kaydedilmiş, daha sonra “convert2YOLO” programı ile YOLO etiket bilgisi formatına dönüştürülmüştür [30]. Bu format seçiminin amacı bu veri setinin farklı nesne tanıma mimarilerinde de kullanımına imkân sağlamaktır.

3.1.2.2. YOLO Etiket Dosyası Formatı

labelImg programında bir etiketleme işlemi sonunda aşağıdaki bir satırlık veri üretilir. Etiketlenen her obje yeni bir satır olarak ilgili etiketleme dosyasına bu formatta kayıt edilir. Bu dosya formatında X ve Y parametreleri etiketleme kutusunun orta noktasının X ve Y koordinat verisidir. Şekil 3.2’de bir etiketleme dosyası örneği görülmektedir. Örnekteki her satır bir adet etiketleme işleminin verisini içermektedir. Bir satırdaki veri içeriği sırasıyla *Obje sınıfı*, *X*, *Y* parametreleri, *genişlik* ve *yükseklik* verisi şeklindedir. Bu veriler birer boşluk ile birbirinden ayrılmaktadır.

Obje-sınıfı verisi 0 ile (sınıf sayısı – 1) arasında bir değer almaktadır. Bu çalışmada baretli ve baretsiz olmak üzere iki sınıf olduğu için obje-sınıfı sadece 0 veya 1 değerini alır. X, Y, genişlik ve yükseklik verileri ise 0 ile 1 arası ondalıklı sayı olarak hesaplanır.

- $X = X \text{ koordinatı} / \text{fotoğrafın genişliği}$.
- $Y = Y \text{ koordinatı} / \text{fotoğrafın yüksekliği}$.
- $\text{genişlik} = \text{Etiketleme kutusunun genişliği} / \text{fotoğraf genişliği}$.
- $\text{yükseklik} = \text{Etiketleme kutusunun yüksekliği} / \text{fotoğrafın yüksekliği}$.

```
0 0.559 0.3 0.09 0.168
0 0.267 0.392 0.075 0.136
0 0.828 0.303 0.098 0.168
```

Şekil 3.2: YOLO etiket dosyası örneği.

3.1.2.3. Pascal VOC Etiket Dosyası Formatı

Bu dosya formatı aslında XML bir dosyadır. İçerisinde etiketleme ile ilgili tüm veriler bulunur. Diğer nesne tanıma mimarilerinde bu dosya formatı kullanıldığı için etiketleme işlemi bu dosya formatı ile kayıt edilip, sonrasında YOLO formatına dönüştürülmüştür. Bu şekilde gelecekte YOLO dışındaki diğer nesne tanıma mimarilerinde de bu veri setinin kullanım imkânı sağlanmıştır. Aşağıda XML dosya formatının bir örneği görülmektedir.

```
<?xml version="1.0" encoding="UTF-8" ?>
<annotation>
  <folder>train_image_folder</folder>
  <filename>Video_9_frame_sec_392_Crop_2.jpg</filename>
  <path>/home/csagin/Academic/New_Dataset/train_image_folder/Video_9_frame_sec_392_Crop_2.jpg</path>
  <source>
    <database>Unknown</database>
  </source>
  <size>
    <width>144</width>
    <height>297</height>
    <depth>3</depth>
  </size>
  <segmented>0</segmented>
  <object>
    <name>helmet</name>
    <pose>Unspecified</pose>
    <truncated>1</truncated>
    <difficult>0</difficult>
    <bndbox>
      <xmin>44</xmin>
      <ymin>1</ymin>
      <xmax>96</xmax>
      <ymax>66</ymax>
    </bndbox>
  </object>
</annotation>
```

Şekil 3.3: XML etiket dosyası örneği.

3.1.3. Eğitim İin YOLO'nun Konfigürasyonu

YOLO algoritmasının bizim hazırladığımız veri setini kullanarak eğitimi için yapılması gereken konfigürasyon, algoritmanın ihtiyacı olan bazı bilgilerin çeşitli dosyalarda ayarlanması ile gerçekleştirilir. Değişikliklerin yapılacağı dosyalar “obj.names”, “obj.data”, “train.txt”, “validation.txt” ve “yolov3.cfg” dosyalarıdır.

Obj.names, bu dosya içerisinde sınıf isimlerini içermektedir. Etiketlemeye ilk başlanan sınıf, yani YOLO etiket formatına göre 0 indeksine sahip sınıf ilk satıra yazılmalıdır. İndeks numarasına göre diğerleri her biri bir satıra gelecek şekilde eklenmelidir. Bu çalışmada baretli sınıfı 0 nolu indekse sahipken, baretsiz sınıfı 1 nolu indekse sahiptir.

Obj.data, bu dosyanın içeriği şu aşağıdaki şekildedir.

- i) classes= 2
- ii) train = data/train.txt
- iii) valid = data/validation.txt
- iv) names = data/obj.names
- v) backup = backup/

İsminden de anlaşıldığı gibi classes parametresi, sınıf sayısını belirtmektedir. Diğer parametreler ise eğitim sırasında ihtiyaç duyulan diğer dosyaların hangi klasör altında olduğu verisini içermektedir. Eğer bu dosyaların yerleri değişmiş ise burada gerekli düzeltme yapılmalıdır.

Hazırladığımız veri seti dosyalarının konumları ve dosya isimleri train.txt ve validation.txt dosyalarında tutulmaktadır. Bizim çalışmamızda etiketleme işlemi yaptığımız 11505 fotoğrafın 8280 adedi eğitim amacıyla kullanılırken, eğitim sırasında her 1000 iterasyonda hesaplanan ortalama doğruluk (İng. “mean Average Precision”, mAP) ölçümü için ise geri kalan 3225 fotoğraf kullanılmıştır. Ortalama doğruluk ölçümü işlemi eğitim sırasında üretilen ağırlıklar için yapılmaktadır, tasarlanan konfigürasyonda her 1000 iterasyonda bir ağırlık üretimi yapılmaktadır. Eğitimin ne zaman sonlandırılacağı ve üretilen hangi ağırlığın seçilerek kendi çözümümüzde kullanılacağı kararı bu ölçümler ile alınmaktadır.

Yolov3.cfg ise bir konfigürasyon dosyasıdır. Bu dosya içerisinde eğitim süresini ve eğitim sırasında üretilecek ağırlıkların çalışma performansını etkileyen önemli parametreler bulunmaktadır. Ayarlanan parametrelerin değerleri ve bunların etkileri şu şekildedir.

- batch = 64
- width=416
- height=416
- momentum=0.9
- decay=0.0005
- learning_rate=0.001

batch parametresi, her bir iterasyonda eğitimde kullanılacak fotoğraf sayısını belirtir. GPU'nun RAM kapasitesine göre bu parametre eğitim öncesi ayarlanmalıdır. Çok yüksek bir miktar seçilirse GPU RAM miktarı yetersiz kalacağı için hata vererek eğitim yarıda kesilecektir, eğer çok düşük bir miktar seçilirse eğitim çok uzun sürebilir. Bu çalışmada 16 GB ram'e sahip, bulut tabanlı bir sanal bilgisayar kullanıldığı için 64 rakamı ayarlanmıştır.

width ve height parametreleri YOLO ağının çözünürlük parametreleridir. Bu parametreler eğitim için ağın girişine uygulanan fotoğrafların çözünürlüklerinin yeniden boyutlandırılacağı değerlerdir. Bu parametrelerin çok yüksek seçilmesi ağın çalışma hızını yavaşlatır, gerçek zamanlı nesne tanıma performansını olumsuz etkiler. Çok düşük bir değer seçilmesi durumunda ise ağın tahmin doğruluk seviyesini olumsuz yönde etkiler.

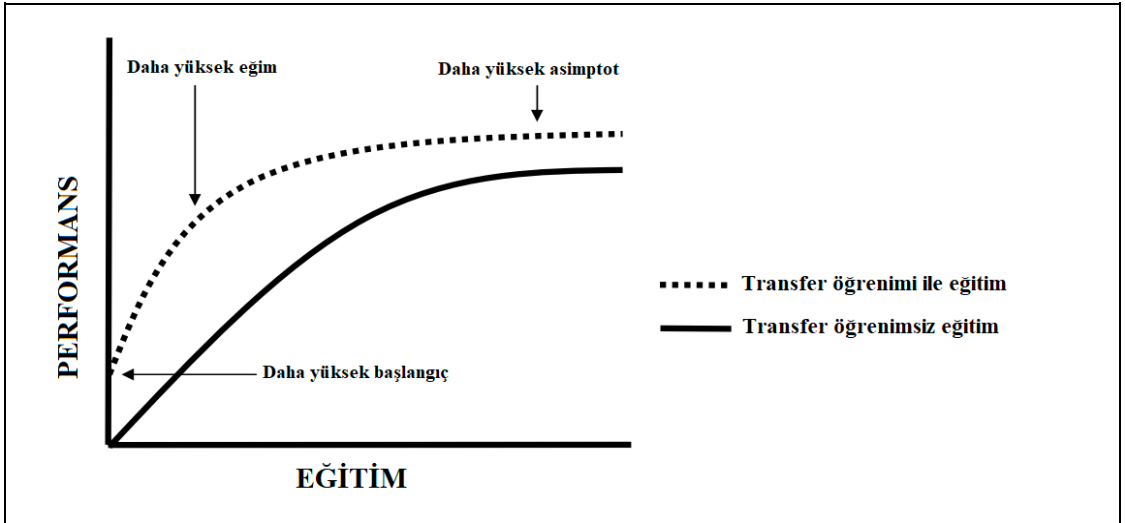
Bu parametreler dışında konfigürasyon dosyasında belirlenen ve sınıf sayısına göre değişen, evrişimsel filtre parametresi ayarlanmıştır. Yolov3.cfg dosyasında bir çok filtre bulunmaktadır. Burada değişiklik yapılan filtre parametresi yolo katmanlarından önceki filtrelerdir. $filters=(classes + 5) * 3$ formülü ile hesaplanır. Bu çalışmada iki adet sınıf olduğu için $filters = 21$ olarak ayarlanmıştır.

3.1.4. Veri Setinin Eğitimi

Baret veri seti, baretli ve baretsiz olmak üzere iki sınıfa sahip bir veri setidir. Bu veri seti için gerekli tüm parametre ayarları yapıldıktan sonra eğitim aşağıdaki komut linux komut satırına yazılarak eğitim işlemi başlatılır.

- Komut: `./darknet detector train data/obj.data yolov3.cfg darknet53.conv.74`

Bu komut sırasında kullanılan “darknet53.conv.74” dosyası, eğitimi hızlandırmak ve eğitim performansını arttırmak için kullanılan, ön-eğitimli (pre-trained) bir modeldir. Ön-eğitimli model kullanılması işlemine literatürde Transfer öğrenimi (İng. “Transfer learning”) denir. Transfer öğrenimi, görevi modellerken hızlı ilerleme ve gelişmiş performans sağlayan bir optimizasyondur. Öğrenmeyi transfer etmek, yeni bir görevde öğrenmenin, önceden öğrenilmiş olan ilgili bir görevden bilginin aktarılması yoluyla iyileştirilmesidir.



Şekil 3.4: Transfer öğrenimi ve eğitimin performansı [31].

Transfer öğreniminin özgün veri setinin eğitimi esnasında üç önemli noktada eğitim performansını arttırıcı etkisi vardır.

- Daha yüksek başlangıç noktası: Transfer öğrenimi uygulanan modelin başlangıç yetenekleri, uygulanmayan modele göre daha yüksektir.

- Daha yüksek eğime sahip bir eğitim-performans eğrisi: Transfer öğrenimi uygulanan modelin eğitim sırasında elde edilen eğitim-performans eğrisi, diğerine göre daha diktir.
- Daha yüksek asimptot seviyesi: Transfer öğrenimi uygulanan modelin eğitim sonundaki performans seviyesi, uygulanmayan modele göre daha yüksek bir seviyededir.

“darknet53.conv.74” isimli ön-eğitilmiş model dosyası, daha önce image.net veri seti ile eğitilerek oluşturulmuş bir modeldir. Image.net veri seti ile eğitilen evrişimsel ağırlıkları içermektedir. Bu model YOLO’nun internet sitesinden indirilmiştir [26].

3.1.5. Eğitim Sonuçları

Eğitim sırasında train.txt dosyasında belirlenen dosyalar kullanılarak eğitim gerçekleştirilmekte, her 1000 iterasyonda bir eğitim işlemi devam ederken ağırlık üretilmektedir. Üretilen bu ağırlığın ortalama doğruluk (İng. “mean Average Precision”) hesaplamasında, yani doğrulama işlemi ise validation.txt dosyasında belirlenen veriler kullanılmıştır. Eğitim ve doğrulama için kullanılan veriler birbirinden farklıdır ve ayrı klasörlerde bulunmaktadır.

Eğitim işlemi 50000 iterasyona kadar çalışılmıştır, eğitim sırasında 34000. iterasyonda ortalama olarak en yüksek doğruluk oranına erişildiği için ve bu noktadan sonra ortalama doğruluk seviyesinde düşüşler başladığı için 34000. iterasyonda üretilen ağırlık seçilerek OpenCV fonksiyonlarında nesne tanıma işlemi için kullanılmıştır.

Eğitim işlemi Tesla V100 GPU’lu bir sanal makine üzerinde gerçekleştirilmiş ve 28 saat sürmüştür. Eğitim 50000. iterasyona ulaştığında üretilen ağırlıkların ortalama doğruluk oranlarında daha yüksek bir oran yakalanamadığı için bu noktada durdurulmuştur.

Tablo 3.1: Üretilen bazı ağırlıkların doğruluk seviyeleri.

İterasyon	1000	5000	10000	15000	30000	32000	33000	34000	35000
Baretili Sınıfı	67,74	82,46	86,85	88,16	87,98	87,32	87,85	87,8	87,57
Baretsiz Sınıfı	56,56	82,19	85,21	85,52	86,6	86,7	87,01	87,18	87,04
Ağırlığın Ortalaması	62,15	82,32	86,03	86,84	87,29	87,01	87,43	87,49	87,3

3.2. OpenCV

OpenCV (Open Source Computer Vision) açık kaynak kodlu görüntü işleme kütüphanesidir. 1999 yılında Intel tarafından geliştirilmeye başlanmış daha sonra Itseez, Willow, Nvidia, AMD, Google gibi şirket ve toplulukların desteği ile gelişim süreci devam etmektedir. İlk sürüm olan OpenCV alfa 2000 yılında piyasaya çıkmıştır. İlk etapta C programlama dili ile geliştirilmeye başlanmış ve daha sonra birçok algoritması C++ dili ile geliştirilmiştir. Open source yani açık kaynak kodlu bir kütüphanedir ve BSD lisansı ile altında geliştirilmektedir.

OpenCV platform bağımsız bir kütüphanedir, bu sayede Windows, Linux, FreeBSD, Android, Mac OS ve iOS platformlarında çalışabilmektedir. C++, C, Python, Java, Matlab gibi programlama dilleri ile kolaylıkla OpenCV uygulamaları geliştirilebilir. OpenCV kütüphanesi içerisinde görüntü işlemeye (İng. “image processing”) ve makine öğrenmesine (İng. “machine learning”) yönelik 2500’den fazla algoritma bulunmaktadır. Bu algoritmalar ile yüz tanıma, nesnelere ayırt etme, insan hareketlerini tespit edebilme, nesne sınıflandırma, plaka tanıma, üç boyutlu görüntü üzerinde işlem yapabilme, görüntü karşılaştırma, optik karakter gibi işlemler rahatlıkla yapılabilmektedir.

3.2.1. OpenCV’nin Temel Modülleri

OpenCV kütüphanesini oluşturan ana bileşenlerden bazıları aşağıdaki gibidir.

highgui: Resim görüntüleme, pencereleri yönetme ve grafiksel kullanıcı arabirimleri için gerekli olabilecek metotları barındırır.

imgcodecs: Dosya sistemi üzerinden resim ve video okuma/yazma işlemlerini yerine getiren metotları barındırmaktadır.

videoio: Kameralara ve video cihazlarına erişmek ve görüntü almak ve görüntü yazmak için gerekli metotları barındırır.

core: OpenCV’nin temel fonksiyonları ve matris, point, size gibi veri yapılarını içerir. Ayrıca görüntü üzerine çizim yapabilmek için kullanılacak metotları barındırır.

imgproc: Görüntü işleme ile ilgili filtreleme operatörleri, kenar bulma, nesne belirleme, renk uzayı yönetimi, renk yönetimi ve eşikleme gibi neredeyse tüm fonksiyonları içine alan bir pakettir.

3.2.2. OpenCV ve Derin Öğrenme

Bilgisayarlı görme alanı derin öğrenmenin gelişiminden itibaren büyük bir değişim yaşıyor. Bu alanda kullanılan , OpenCV'ye entegre edilmiş bir çok metod ve fonksiyon derin öğrenme tabanlı çözümler tarafından geride bırakılmıştır. Derin öğrenme tabanlı çözümler çalışma doğrulukları ile büyük bir fark yaratmıştır. Buna ek olarak derin öğrenmedeki bu gelişimler ile birlikte daha öncesinde yıllarca süren bilgisayarlı görme tabanlı sistem geliştirme işlemleri, aylar mertebesindeki kısa sürelerde yapılabilir hale gelmiştir. Derin öğrenme metodları çok kısa bir süre içerisinde bilgisayarlı görme alanında çok büyük değişiklikler gerçekleştirmiştir.

Derin öğrenme alanında yaşanan bu hızlı gelişim ile birlikte, bu alanda kullanılan yazılım kütüphaneleri ve yazılım mimarileri de hızla çeşitlenmiş ve gelişmiştir. Tensorflow, Pytorch, Caffe, MxNet, CNTK, Keras gibi kütüphaneler, derin öğrenme mimarilerinin tasarlanmasında sıkça kullanılmaktadır. OpenCV yaşanan bu gelişimi takip ederek, bu mimarilerde eğitilmiş modellerin (ağırlıkların), OpenCV içerisinde kullanımına imkan veren esnek bir model sunmuştur.

Bu çözüm özellikle makine öğrenmesi tabanlı nesne tanıma fonksiyonları kullanan OpenCV uygulamalarının, daha yüksek doğrulukta çalışan derin öğrenme tabanlı nesne tanıma fonksiyonlarına geçişine imkan sunmaktadır. Kullanıcılar kendi oluşturdukları veri setini istedikleri mimaride eğiterek, elde ettikleri eğitilmiş modelleri (ağırlıkları) OpenCV'de kullanabilmektedir. Bu sayede kullanıcılar mevcut iş akışlarını bozmadan eğittikleri derin öğrenme modelini altyapılarına yerleştirebilmektedirler.

OpenCV 3.3.1 versiyondan itibaren Caffe, Torch, Tensorflow ve Darknet (YOLO) mimarilerinin kullanımını desteklemektedir.

3.3 Problemin Çözümü

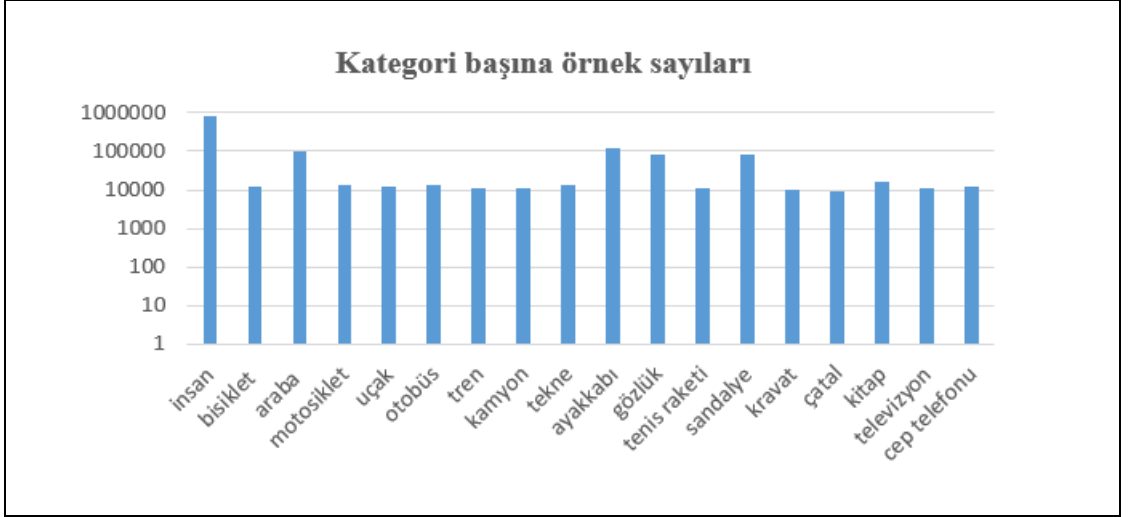
Koruyucu baret kullanım ihlallerinin tespiti problemini çözmek için görüntü üzerinden nesne tanıma işleminin yapılması hedeflenmiştir. Bu çalışmada incelenen derin öğrenme tabanlı görüntü tanıma mimarileri arasından YOLOv3 mimarisi seçilerek, çalışmaya devam edilmiştir. Bu seçimin ardındaki nedenler şu şekildedir.

- YOLO obje tanıma mimarisi açık kaynak kodlu bir mimaridir. İnternet sitesinden indirilip çalışmaya hemen başlanabilir [26].
- İncelenen diğer mimarilere göre daha hızlı ve yüksek doğrulukta çalışmaktadır.
- Açık kaynak kodlu olduğu için hakkında bilgi bulmak daha kolaydır.
- OpenCV platformunda desteklenmektedir.
- Bir çok veri setinin YOLOv3 ile eğitilmiş ağırlıkları internette bulunabilmektedir.

Bu problemde görüntüde bulunan insanların baret kullanıp kullanmadığının belirlenmesi istenmektedir. Yani baretli ve baretsiz sınıfının tespiti istenmektedir. Çözüm olarak iki aşamalı çalışan bir algoritma tasarlanmıştır. İlk aşamada MS COCO veri seti ile eğitilmiş bir YOLOv3 modeli kullanılarak görüntüdeki insanların yerleri belirlenmiştir [27]. Belirlenen bu bölgelerde baret veri seti ile eğitilen YOLOv3 modeli kullanılarak baret kullanım durumunun tespiti yapılmıştır.

MS COCO veri seti içerisinde 80 adet sınıf bulunan ve 200,000'den fazla görüntü barındıran bir veri setidir. Bu veri setine göre eğitilmiş YOLOv3 ağırlıkları internette indirilip algoritmanın ilk aşamasında kullanılmıştır. Bu veri setinin sınıflarından “person” (insan) sınıfı kullanılarak işçilerin yerlerinin tespiti gerçekleştirilmiştir.

MS COCO veri setindeki 80 sınıfın içinden örnek olarak 18 tanesinin sınıflara göre veri dağılımı Şekil 3.5’de görülmektedir.



Şekil 3.5: MS COCO veri setindeki bazı kategorilerin örnek sayıları.

Görüntüde insan tespit işleminin ardından, insan tespit edilen bölge kırılarak yeni görüntüler oluşturulmuş ve bunlar üzerinde ikinci aşamada baret tanıma işlemi gerçekleştirilmiştir. Yapılan çözümün blok şeması Şekil 3.6'da algoritmanın akış diyagramı Şekil 3.7'de görülmektedir. Algoritma python dilinde OpenCV kütüphaneleri kullanılarak yazılmıştır.

İki aşamalı tasarlanan algoritmanın ilk aşaması görüntüdeki çalışanların yerlerinin tespit edilmesi işlemini gerçekleştirmektedir. İkinci aşamadaki baretli veya baretsiz durumunun tespit işlemi sadece birinci aşamadaki çalışan tespit işlemi başarılı olması durumunda çalışanların tespit edildiği bölge üzerinde gerçekleştirilmektedir. Bu sayede görüntünün tamamında baret arama işlemi yapılmamakta ve insanların olmadığı bölümlerde bulunan baret veya baret benzeri cisimlerin sonucu etkilemesinin önüne geçilmektedir.

Baret veri seti oluşturulurken sadece baret takmış veya baret takmamış insanların baş kısmı kutucuk içerisine alınarak etiketleme işlemi yapıldığı için, görüntüdeki çalışanın baretini takmayıp elinde tutması gibi durumların baretsiz olarak sınıflandırılması sağlanmıştır.

Veri seti bir çok farklı tip ve renkteki baretleri içerecek şekilde oluşturulduğu için, baretli sınıfı tespitinde baretin tipi veya rengi ile ilgili herhangi bir kısıtlama bulunmamaktadır. Her renkteki baretin tespiti gerçekleştirilmektedir.



İnsan Algılama : YOLOv3 (COCO veri seti)



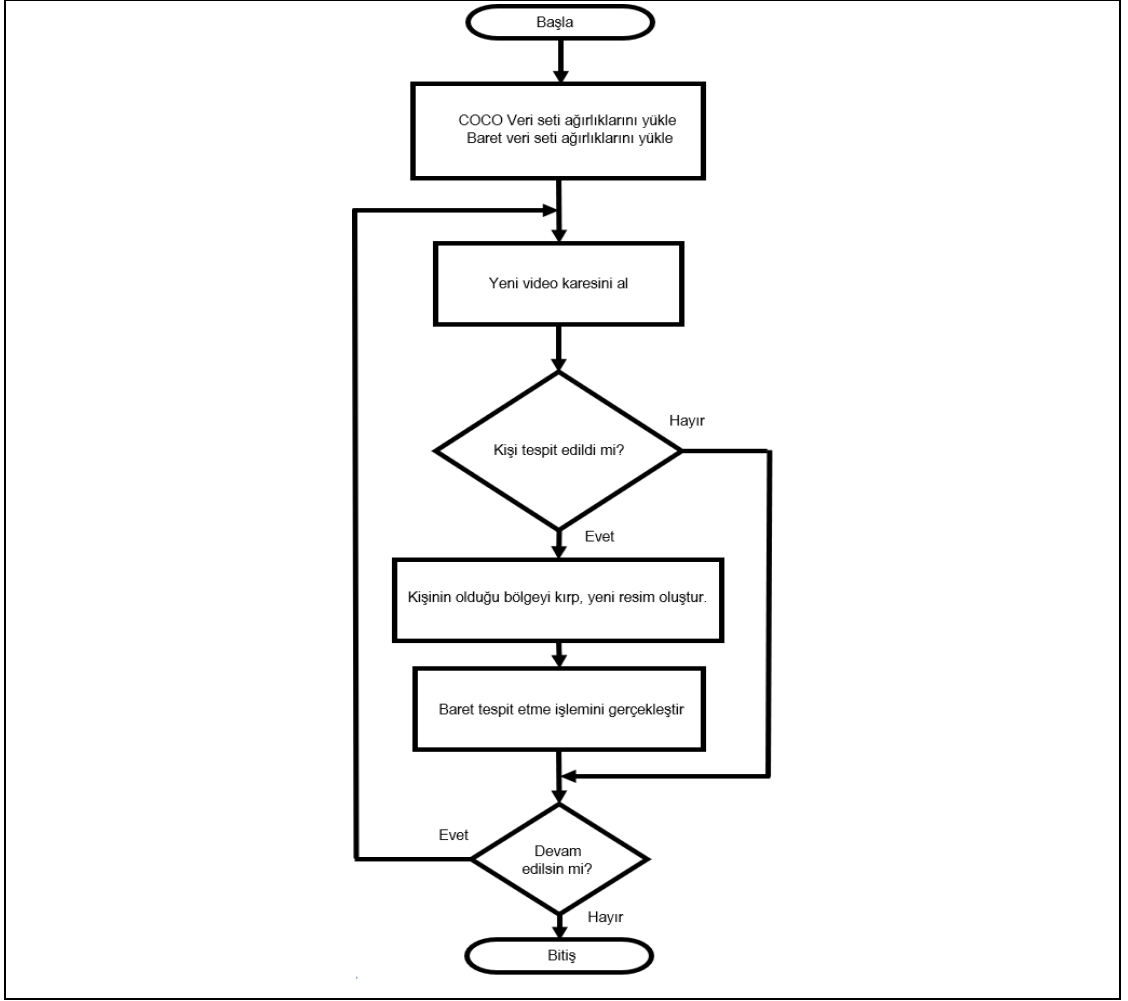
Ara İşlem (Kırpma işlemi)



Baret kullanımı tespiti: YOLOv3 (Baret veri seti)



Şekil 3.6: Çözümün blok şeması.



Şekil 3.7: Algoritmanın akış diyagramı.

Yazılan programda çalışma esnasında en iyi sonucu elde edebilmek için bazı parametreler testler esnasında belirlenmiştir. Güven skorunun threshold değeri ve Non-Max Suppression (NMS) threshold değeri de 0,4 olarak belirlenmiştir. İki aşamalı olarak çalışan bu algorithmada COCO veri seti'nin "person" sınıfı dışındaki diğer sınıflar dikkate alınmamıştır. Algoritmanın çalışma hızını YOLO konfigürasyon dosyasındaki genişlik ve yükseklik değerlerinin değişimine bağlı olarak artış ve azalış gösterdiği gözlemlenmiştir.

4. DENEYLER VE SONUÇLAR

Bir önceki bölümde detayları anlatılan, baretli ve baretsiz kişilerin tespiti için tasarlanan algoritmanın testleri bu bölümde gerçekleştirilmiştir. Algoritmanın testi için baretli ve baretsiz fotoğrafları içeren bir test seti oluşturulmuştur. Test setindeki fotoğraflar internette çeşitli veri setleri taranarak bulunmuştur. Testte kullanılan baretli fotoğraflar *Harvard dataverse*'de bulunan *hardhat* veri setinden alınmıştır [32]. Baretsiz fotoğraflar ise *Inria person* veri setinden temin edilmiştir [33]. Test sırasında kullanılan baretli fotoğraf adedi 1518, baretsiz fotoğraf adedi ise 1617 dir. Bu test setindeki fotoğraflar algoritmanın ilk aşamasında tespit edilen kişi bölgelerinin kırılması ile oluşturulmuştur. Bu yüzden her fotoğrafta bir adet insan görüntüsü bulunmaktadır.

Problemin çözümü için tasarlanan algoritma iki aşamalı bir yapıya sahiptir. İlk aşamada görüntü içerisindeki kişiler tespit edilmekte, ikinci aşamada tespit edilen bu kişilerin baret takıp takmama durumunun tespiti yapılmaktadır. Her iki aşamanın ayrı ayrı performanslarını görebilmek için üç adet test seti oluşturulmuştur.

Bu test setlerinin ilki *insansız ortam* test setidir. Bu test seti içeriğinde hiç insan görüntüsü olmayan 1030 adet fotoğrafın bir araya getirilmesi ile oluşturulmuştur. Bu test seti ile birinci ve ikinci aşamanın insansız görüntülerdeki çalışma performansı ayrı ayrı test edilerek tablo halinde sonuçlar sunulmuştur.

İkinci test seti *kişi tespit* test setidir. Birinci aşamanın insanlı görüntülerdeki insanları tespit performansının analizi için hazırlanmıştır. Bu test seti içerisinde baretli veya baretsiz insanların olduğu 1019 adet fotoğrafın bir araya getirilmesi ile oluşturulmuştur.

Üçüncü test seti ise ikinci aşamanın testi için hazırlanmıştır. Bu test seti elimizdeki baretli ve baretsiz fotoğrafların bir araya getirilmesi ile oluşturulmuştur. Bu test seti 994 adedi baretli insanların görüntülerini içeren fotoğraflardan, 1125 adedi ise baretsiz insan görüntülerini içeren fotoğraflardan olmak üzere toplam 2119 fotoğraftan meydana gelmektedir. Üçüncü test seti, algoritmanın testi için kullanılan test setinin ilk aşama uygulanmadan önceki orijinal fotoğraflarıdır. Algoritmanın tamamının testinde orijinal fotoğraflar üzerinde kişilerin yerleri tespit edilmekte, bu tespit edilen kişilerin olduğu bölge kırılarak yeni fotoğraflar oluşturulmaktadır. Ancak bu testte orijinal fotoğraflar üzerinde herhangi bir kırılma işlemi yapmadan

kişilerin baret takıp takmama durumlarının tespiti yapılmıştır. Yani algoritmanın testi ile benzer bir test bu bölümde sadece birinci aşama olmadan orijinal fotoğraflar üzerinde gerçekleştirilmiştir. Yapılan bu tüm testler ile elimizdeki YOLOv3 ağırlıklarının çeşitli durumlardaki çalışma performansları tespit edilmiştir.

4.1 Algoritmanın Testi

Baret veri setinin eğitimi esnasında her 1000 iterasyonda bir adet ağırlık üretilmiştir. Üretilen bu ağırlıklar arasında yapılan karşılaştırma ile en performanslı doğrulama sonucunu veren 34000. Iterasyonda üretilen ağırlık seçilerek bu aşamada kullanılmıştır. Bu testte her biri içerisinde bir adet insan görüntüsü olan ve algoritmanın birinci aşaması tarafından üretilmiş 1518 adet baretli, 1617 adet ise baretsiz insanın fotoğrafının olduğu test seti kullanılmıştır.

Test esnasında üretilen tüm fotoğraflar tek tek gözle kontrol edilmiştir. Bu testin sonuçlarını kontrol ederken temel başarı kriterimiz doğru sınıflandırma ve nesnenin fotoğraf içerisindeki yerinin doğru bir şekilde dikdörtgen içerisine alınmış olunmasıdır. Bu testin çıktısı olarak elde edilen başarılı çıktılar bazılarını şekil 4.1'de görülmektedir. Örnekler tek tek incelendiğinde farklı renge sahip baretlerin sorunsuz bir şekilde tespit edildiği görülmektedir. Şekil 4.1 d) ve f) örneklerinde yüzü tam olarak görünmeyen insanların sınıflandırılmasının doğru bir şekilde yapıldığı görülmektedir.

Tüm örnekler incelendiğinde baretli veya baretsiz sınıfının tespiti esnasında fotoğraftaki kişilerin yüzlerinin görünür olması şartı, kişilerin görüntüde dik durması şartı veya bazı baret renklerini algılayamama gibi sınırlayıcı bir durum ile karşılaşılmamıştır.

Tablo 4.1: Algoritmanın test sonucu.

Test seti	TEST SONUCU		
	Baretli tespiti	Baretsiz tespiti	Sonuç yok
Baretli sınıfı	1452	23	43
Baretsiz sınıfı	11	1562	44



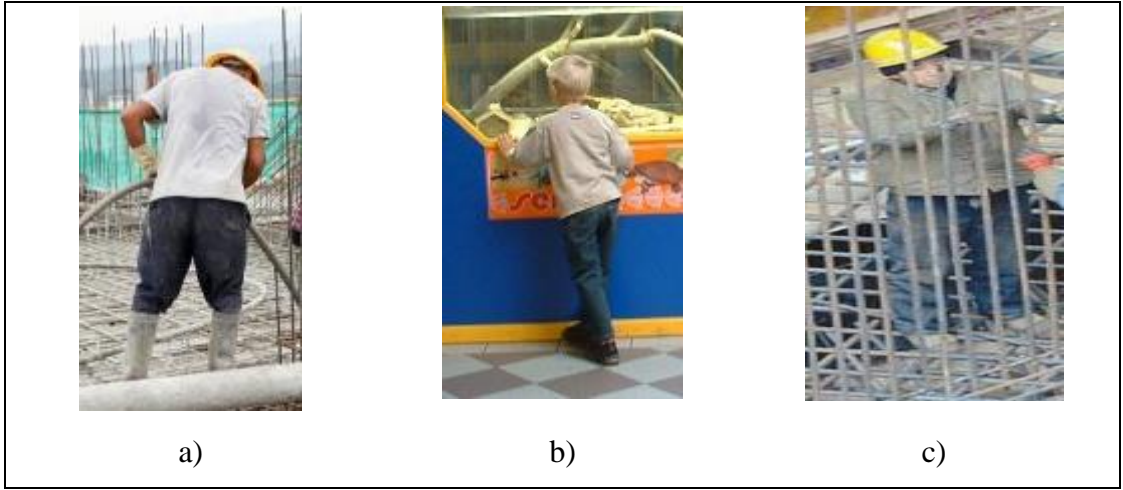
Şekil 4.1: Başarılı test sonucu örnekleri.

Test sonucu elde edilen veriler tablo 4.1’de bir araya getirilmiştir. Tablo incelendiğinde baretili sınıfına ait 1518 fotoğrafın, 1452 adedi doğru bir şekilde baretili sınıfı olarak sınıflandırılırken, 23 adedi baretsiz olarak sınıflandırılmıştır. 43 adet fotoğrafta ise herhangi bir sınıflandırma sonucu elde edilememiştir. Başarısız test sonucu örnekleri şekil 4.2 ve şekil 4.3’de görülmektedir. Şekil 4.2’de yanlış sınıflandırma örnekleri görülürken, şekil 4.3’de herhangi bir sınıflandırma sonucu alınamayan örnekler bulunmaktadır.

Hatalı örneklerin sayısı az olsa da bu hatalı örneklerin daha da azaltılması için eğitim kümesinin genişletilmesi gerekmektedir. Derin öğrenme mimarilerinin eğitimi esnasında oldukça geniş veri setleri kullanılmaktadır.



Şekil 4.2: Yanlış sınıflandırma örnekleri.



Şekil 4.3: Sonuç alınamayan örnekler.

4.2 İnsansız Ortam Testi

İnsansız ortam test seti 1030 adet fotoğraftan meydana gelmektedir. Bu test seti insanların kamusal veya özel yaşam alanlarının fotoğraflarından meydana getirilmiştir. Bu fotoğrafların tek ortak yanı hiçbirinde canlı insan görüntüsünün olmamasıdır. Bu fotoğrafların çekildiği ortamlarda heykel ve insan figürlerine sahip objeler bulunabilmektedir.

Bu testte, çözüm için tasarlanan algoritmanın birinci aşamasını oluşturan ve insan tespitini yapan bölüm ile algoritmanın ikinci aşamasını oluşturan ve baretili veya baretsiz sınıfını tespit eden bölüm ayrı ayrı test edilmiştir.

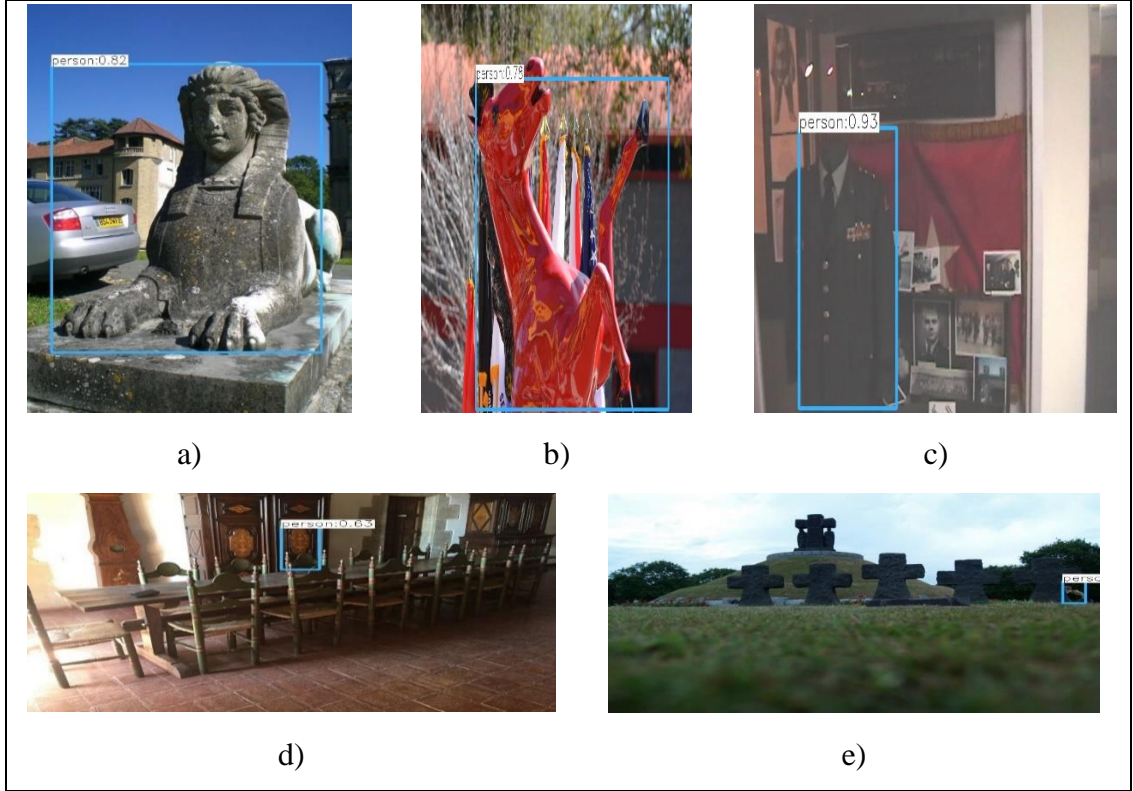
4.2.1 İnsansız Ortamda Birinci Aşamanın Testi

İnsansız ortam test setinde algoritmanın birinci aşamasının çalıştırılması ile bu test gerçekleştirilmiştir. Tablo 4.2’de test sonucu görülmektedir. Şekil 4.4’de ise hatalı sonuçların örnekleri görülmektedir.

Bu testte amaç fotoğraftaki insanların tespit edilmesi işlemidir. Test setinde hiç insan olmadığı için 13 adet insan tespiti sonucunun elde edilmesi hatalı bir durumdur. Hatalar incelendiğinde heykel, vitrin mankeni veya insan yüzü silüeti gibi objelerin fotoğrafta bulunması sonucu hatanın olduğu görülmüştür. Şekil 4.4 a) ve b) de iki adet heykel’in insan olarak sınıflandırıldığı görülmektedir. Şekil 4.4 c) resminde ise bir adet vitrin mankeninin insan olarak sınıflandırılmıştır. Şekil 4.4 d) ve e) örneklerinde ise insan görüntüsü ile alakası olmayan bazı objelerin insan olarak tespit işlemi yapıldığı görülmüştür.

Tablo 4.2: Birinci aşamanın test sonucu.

	Test Sonucu	
	İnsan tespiti	Tespit yok
İnsansız ortam test seti	13	1017



Şekil 4.4: Birinci aşamanın hatalı test sonucu örnekleri.

4.2.2 İnsansız Ortamda İkinci Aşamanın Testi

İnsansız ortam test setinde algoritmanın ikinci aşamasının çalıştırılması ile bu test gerçekleştirilmiştir. Tablo 4.3’de test sonucu görülmektedir. Şekil 4.5’de ise hatalı sonuçların örnekleri görülmektedir.

Tablo 4.3: İkinci aşamanın test sonucu.

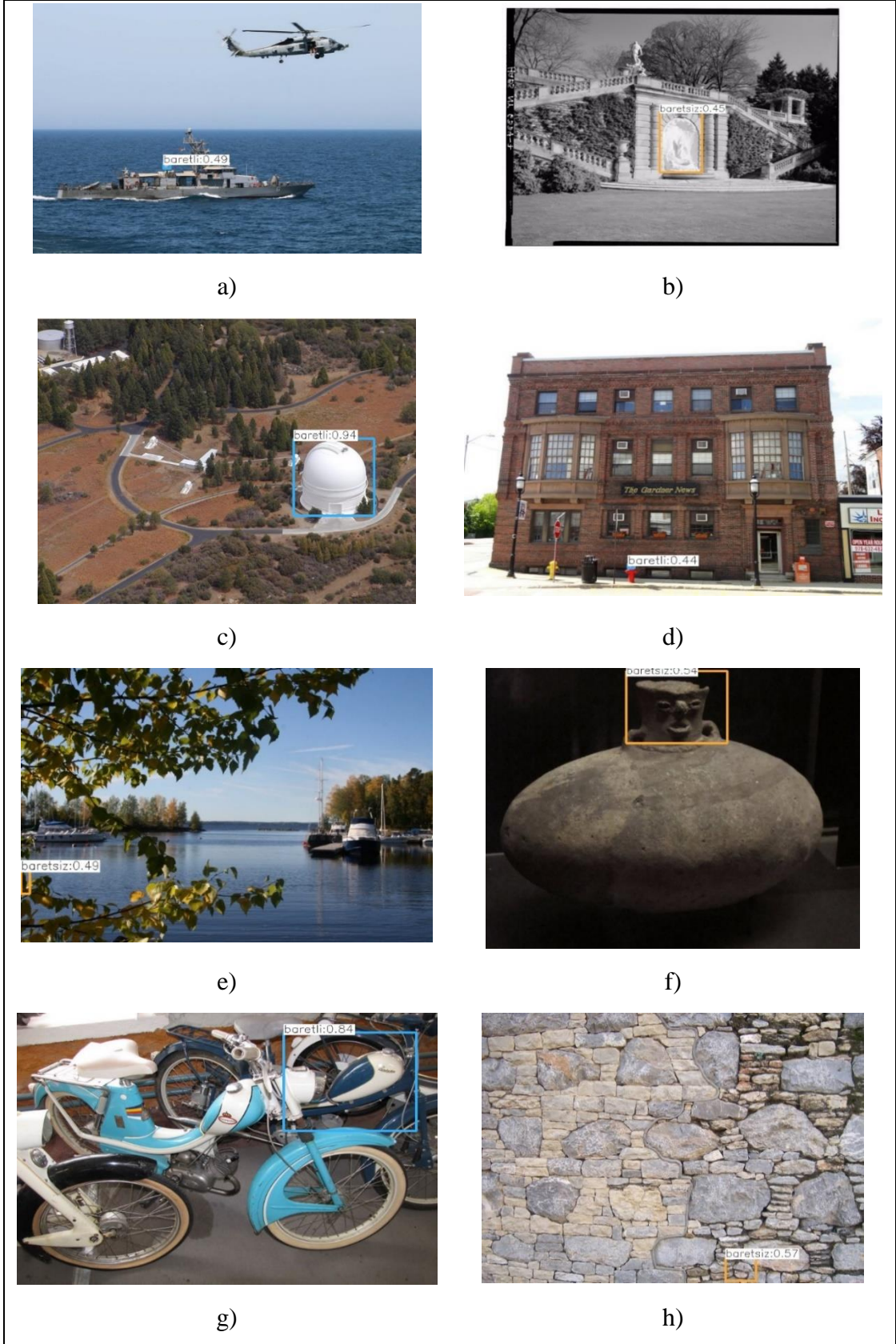
	Test Sonucu		
	Bareti tespiti	Baretsiz tespiti	Tespit yok
İnsansız ortam test seti	13	21	996

Aynı test seti üzerinde yapılan testte ikinci aşama birinci aşamaya göre daha fazla hatalı sonuç vermiştir. Birinci aşamada kullanılan ağırlığın eğitildiği veri setinin, ikinci aşamada kullanılabileceğine göre çok daha fazla veri ile eğitildiği göz önüne alındığında ortaya çıkan bu sonucun normal olduğu görülmektedir. Hatalı durumlar incelendiğinde, insan silüeti veya küresel objelerin baretili veya baretsiz olarak

sınıflandırıldığı görülmüştür. Bu hatalar incelendiğinde tasarladığımız algoritmanın ilk aşamasının önemi görülmektedir.

İlk aşama uygulanmadığında, ortamda hiç insan olmaması durumunda bile ikinci aşama'da %3,3 oranında hatalı bir sınıflandırma performansı görülmektedir. İnsan ile alakası olmayan objelerin baretli veya baretsiz olarak sınıflandırıldığı görülmüştür. Şekil 4.5 a) da gemi üzerindeki küresel objenin baretli olarak sınıflandırıldığı görülmektedir. Şekil 4.5 b) de bahçe duvarındaki bir bölümün baretsiz olarak sınıflandırılmıştır. Şekil 4.5 c) ve d) de küresel objeler baretli sınıfı olarak tespit edilmiştir. Şekil 4.5 e), f) ve h) de insan silüeti içeren görüntüler baretsiz sınıfı olarak sınıflandırılmıştır. Şekil 4.5 g) de ise motosikletin yakıt deposunun baretli olarak sınıflandırıldığı tespit edilmiştir.

Bu test seti üzerinde önce algoritmanın ilk aşaması, ardından ikinci aşamasının çalıştırılmış ve herhangi bir baretli veya baretsiz sınıflandırma sonucu elde edilmemiştir. Bu durum hiç insan içermeyen bir veri seti için istenilen durumdur. Bu test setinde tasarlanan algoritmanın hatasız çalıştığı görülmektedir.



Şekil 4.5: İkinci aşamanın hatalı test sonucu örnekleri.

4.3 İnsanlı Ortam Testi

Bu testte amaç algoritmanın kişi tespitinden sorumlu birinci aşamasının, kişi tespit test seti ile test edilerek performansının ölçülmesidir. Kişi tespit test seti baretili veya baretsiz insan görüntülerini içeren 1019 adet fotoğrafın bir araya getirilmesi ile oluşturulmuştur. Bu test gerçekleştirildikten sonra, görülen hataların üç grupta toplandığı görülmüştür.

Yanlış sınıflandırma hatası: Fotoğrafta insan olmayan bir nesnenin insan olarak sınıflandırılması durumudur.

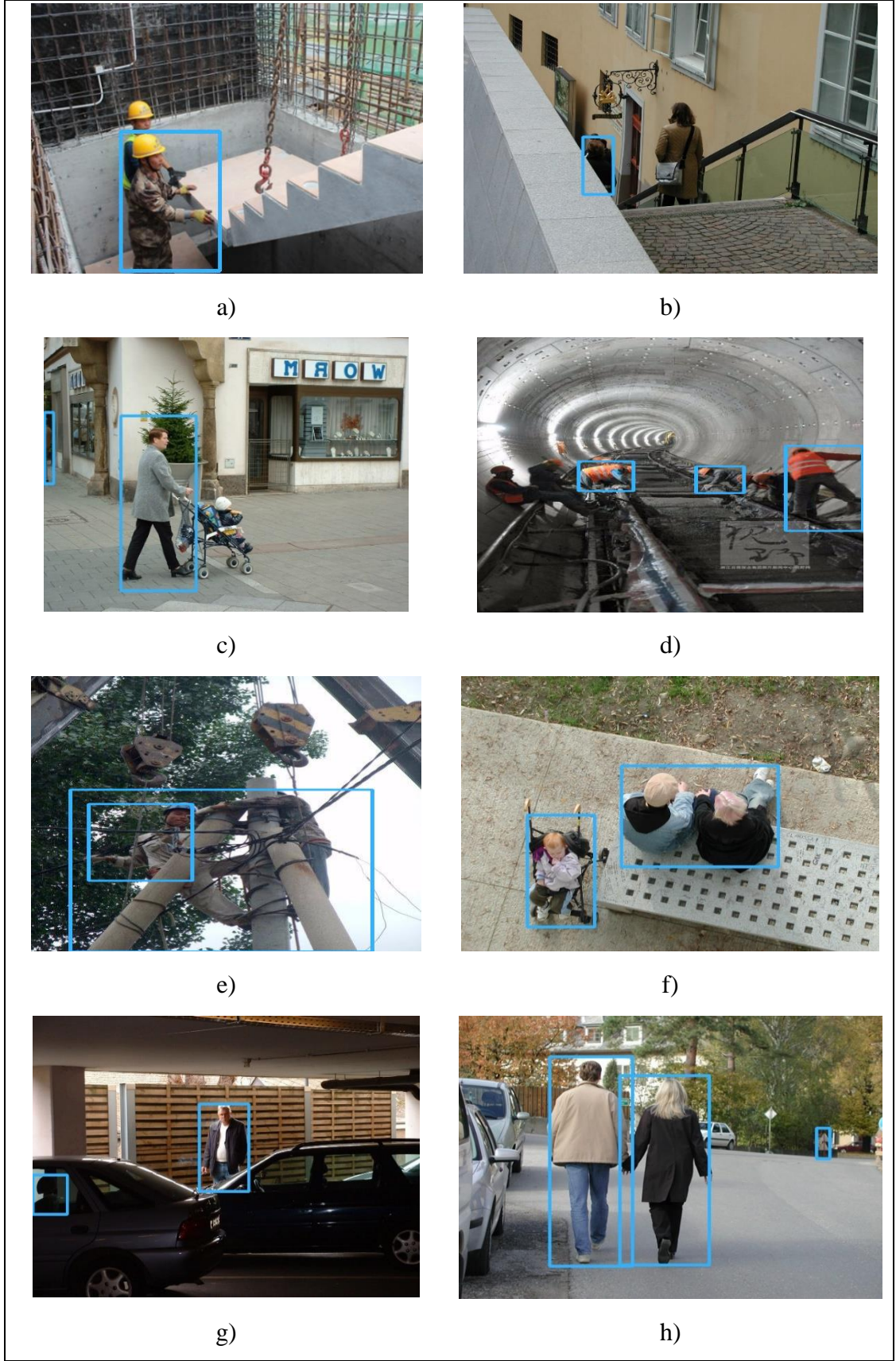
Tespit edilemeyen kişi hatası: Testte beklenen sonuç fotoğraftaki tüm insanların tam olarak tanınıp kutucuk içerisine alınmasıdır. Eğer eksik olarak sınıflandırma yapılmışsa, yani tespit edilemeyen insanlar olmuşsa bu hata tanımı içerisinde değerlendirilmiştir.

Nesne sınırı çizim hatası: Tespit edilen kişilerin içerisine alındığı kutucuğun yanlış olarak konumlandırılması durumları bu hata kapsamında değerlendirilmiştir.

Test sonucu elde edilen çıktı fotoğrafları tek tek incelenerek tablo 4.4’de görülen test sonucu elde edilmiştir. Şekil 4.6’da ise hatalı test sonucu örnekleri görülmektedir. Test sonucu örneklerinin ilk dört örneği tespit edilemeyen kişi olarak adlandırılan hata kapsamına giren örneklerdir. Şekil 4.6 a) ve b) fotoğraflarında birden fazla insan olmasına rağmen sadece birer insan tespit edilmiştir. Aynı şekilde c) ‘de bebek arabasındaki bebek tespit edilememiştir. Şekil 4.6 d) fotoğrafında ise sadece üç işçi tespit edilebilmiştir. Şekil 4.6 e) ve f) ise nesne sınırı çizim hatası örnekleridir, geri kalan son iki örnek ise yanlış sınıflandırma hatası örnekleridir. Şekil 4.6 g) fotoğrafında araba koltuğunun kafalık kısmı insan olarak hatalı tespit edilirken, h)’da ise yol kenarındaki insan olmayan obje insan olarak sınıflandırılmıştır.

Tablo 4.4: İnsanlı ortam testi sonucu.

	TEST SONUCU			
	Başarılı insan tespiti	Yanlış sınıflandırma	Tespit edilemeyen kişi	Nesne sınırı çizim hatası
İnsanlı ortam testi	940	11	65	3



Şekil 4.6: İnsanlı ortam testi hatalı sonuç örnekleri.

Test sonuçları incelendiğinde özellikle tespit edilemeyen kişi hatasının ağırlıkta olduğu görülmüştür. Çok kalabalık insan gruplarını içeren fotoğraflarda bu hata görülebilmektedir.

4.4 Orijinal Test Setini Kullanarak İkinci Aşamamın Testi

Algoritma testi aşamasında orijinal test setindeki tespit edilen insan görüntüleri birinci aşamada kırılarak test seti oluşturulmuş ve baretli veya baretsiz sınıfı için test edilmiştir. Tablo 4.1’de görülen sonuç orijinal test setinin kırılması ile elde edilmiştir. Bu testte ise algoritmanın birinci aşamasını kullanmadan orijinal fotoğraflar üzerinde ikinci aşamanın testi gerçekleştirilmiştir.

Orijinal test seti 994’ü baretli 1125’i baretsiz olmak üzere toplam 2119 fotoğraftan meydana gelmektedir. Testi gerçekleştirdikten sonra çıktı olarak elde edilen fotoğraflar kontrol edilmiştir. Sadece bölüm 4.1’deki algoritma testinde kırılarak elde edilen insan görüntüleri ile bu testte elde edilen orijinal test seti sonucu karşılaştırılmıştır. Elde edilen test sonucu tablo 4.5’de verilmiştir.

Yapılan bu testte elde edilen sonuçlar aşağıdaki açıklandığı şekilde gruplanmıştır.

Farklı obje hatası: Bu hata farklı bir objenin baretli veya baretsiz sınıfı olarak sınıflandırılmasıdır. Bu testte sadece ikinci aşama çalıştırıldığı için çevrede insan ile ilişkili olmadığı halde baretli veya baretsiz olarak sınıflandırılmış objeler gözlemlenmiştir.

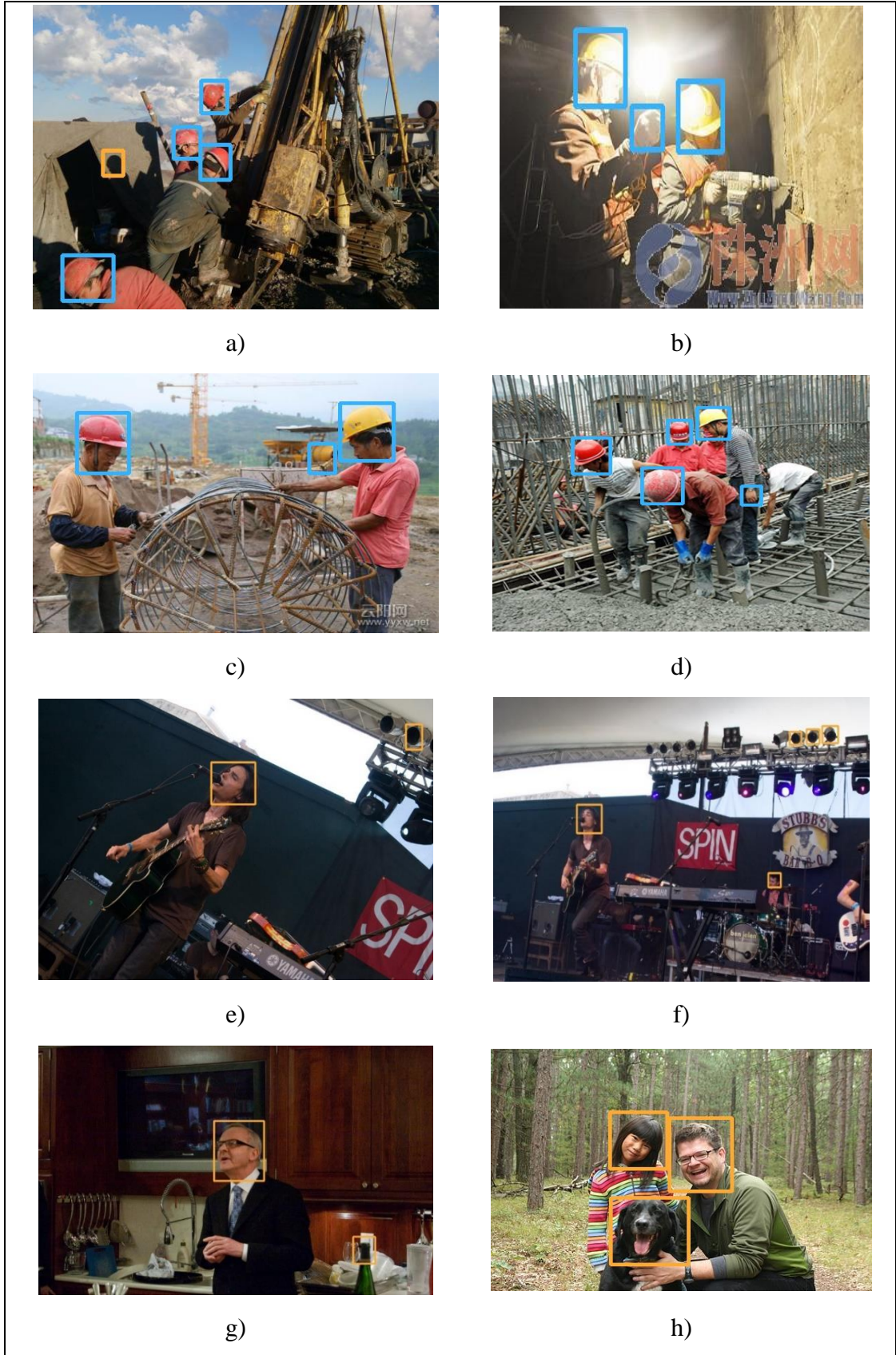
Sonuç yok: Herhangi bir sınıflandırma sonucu elde edilemeyen fotoğraflar için bu ifade kullanılmıştır.

Baretsiz tespiti: Baretsiz olarak sınıflandırma.

Baretli tespiti: Baretli olarak sınıflandırma.

Tablo 4.5: Orijinal test seti ile ikinci aşamanın testi.

	TEST SONUCU			
	Baretli tespiti	Baretsiz tespiti	Sonuç yok	Farklı obje
Baretli sınıfı	1428	5	56	29
Baretsiz sınıfı	1528	6	70	13



Şekil 4.7: Farklı obje hatası örnekleri.

Test orjinal test seti üzerinde sadece ikinci aşamanın çalıştırılması ile gerçekleştirilmiştir. Baretli sınıfının tespiti durumunda obje mavi kutucuk ile baretsiz sınıfı tespitinde ise obje turuncu renkli kutucuk ile çerçeveslendirilmiştir.

Şekil 4.7’de farklı obje hatası örnekleri görülmektedir. Şekil 4.7 a)’da çalışan işçilerin gölgesi baretsiz sınıfı olarak tespit edilmiştir. Şekil 4.7 b) ve d)’de ise çalışan işçinin eli baretli sınıfı olarak işaretlenmiştir. Şekil 4.7 c)’de iş makinesinin bir bölümü baretli sınıfı olarak sınıflandırılmıştır. Şekil 4.7 e) ve f)’de ise sahne aydınlatma aparatlarının bazılarının baretsiz sınıfı olarak işaretlendiği görülmektedir.

Şekil 4.7 g)’de şişenin uç kısmının baretsiz olarak işaretlenmiştir. Şekil 4.7 h)’de ise bir köpeğin baretsiz sınıfı olarak işaretlendiği görülmüştür.

Tablo 4.5 incelendiğinde en çok dikkat çeken farklı obje hatasıdır. Bu hatanın temel nedeni ikinci aşama testinin tüm fotoğrafta yapılması ve çevredeki bazı objelerin bu hataya sebep olmasıdır. Tablo 4.5’deki test sonucunu tablo 4.1’deki sonuç ile kıyasladığımızda %2 daha fazla hatalı sonuç oluştuğu gözlenmektedir.

4.5 Deney Sonuçlarının Değerlendirilmesi

İnsansız ortam testleri incelendiğinde özellikle ikinci aşama olan baretli ve baretsiz sınıflarını tespit eden bloğun tek başına kullanılması durumunda %3,3 oranında yanlış tespitler yaptığı tablo 4.3’de görülmektedir. Bu çalışmada çözüm olarak sunduğumuz algoritma ise insansız ortam testinde hiç hatalı sonuç üretmeden, herhangi baretli veya baretsiz sınıfı tespiti yapmadan bu testi geçmiştir.

İkinci aşamanın tek başına uygulandığı insanlı ortam testleri sonuçları tablo 4.5’de görülmektedir. Farklı obje hataları %2 oranında görülmektedir. Bu çalışmada önerilen algoritma yapısı gereği ikinci aşamayı sadece insan görüntüleri üzerinde çalıştırdığı için farklı obje hatası gibi bir problem görülmemektedir. Algoritmanın performansını gösteren tablo 4.1 ile insanlı ortam test sonucunu içeren tablo 4.5’in sonuçları kıyaslandığında, tablo 4.5’de %2 daha fazla hatalı sonuç oluştuğu görülmektedir.

Elde edilen sonuçlara göre, bu çalışmada çözüm önerisi olarak sunulan algoritmanın farklı obje hatalarını ortadan kaldırdığı ve daha yüksek bir nesne tespit performansı sunduğu görülmüştür.

5. SONUÇ VE ÖNERİ

Bu çalışmada derin öğrenme mimarisi kullanılarak baret kullanım ihlalinin tespitine yönelik bir algoritma önerilmiştir. Baret ihlaline yönelik yasal mevzuat incelenmiş ve baret kullanım ihlaline yönelik daha önce yapılan çalışmalar özetlenmiştir. Derin öğrenme tabanlı bir nesne tanıma algoritması olan YOLO mimarisi esas alınarak iki aşamalı olarak çalışan algoritmanın baret kullanım ihlali konusunda performansını test etmek üzere özgün test setleri oluşturularak çeşitli testler gerçekleştirilmiştir. Farklı test setleri ile hem birinci hem de ikinci aşama test edilmiştir. İnsansız ve insanlı performansları ayrı ayrı ölçülmüştür. Birinci ve ikinci aşamanın arka arkaya kullanımının çevredeki nesnelere kaynaklanabilecek yanlış tespit problemlerini önlediği görülmüştür.

Bu çalışmada oluşturulan veri setleri genişletilerek, yeni sınıflar eklenerek baret dışındaki diğer KKD'lerin kullanım durumlarının tespitinin yapılması mümkündür.

KAYNAKLAR

- [1] ResGaz 1, (2019), Kişisel Koruyucu Donanım Yönetmeliği, 1 Mayıs 2019 tarih ve 30761 sayılı Resmi Gazete.
- [2] Hacıbektaşoğlu S. E., (2018), “İnşaat sektöründe yaşanan iş kazalarının analizi ve bu kazalara neden olan etkenlerin incelenmesi”, “Stratejik ve Sosyal Araştırmalar Dergisi”, 2 (3), 159 – 177.
- [3] Hulme A., Mills N. J., (1995), “Industrial head injuries and the performance of helmets”, Proceedings of the 1995 International IRCOBI Conference on the Biomechanics of Impact, 217, Brunnen, Switzerland, 13-15, September.
- [4] Farhang A., (1998), “Factors contributing to discomfort or dissatisfaction as a result of wearing personal protective equipment”, Journal of Human Ergology, 27 (1-2), 70-75.
- [5] Yılmaz F., Tan O., (2015), “Bir inşaat şantiyesinde iş kazalarının neden olduğu işgünü kayıplarının işverene maliyetinin belirlenmesi”, International Journal of Economic and Administrative Studies, 14, 153.
- [6] Kelm A., Laußat L., (2013), “Mobile passive radio frequency identification (RFID) portal for automated and rapid control of personal protective equipment (PPE) on construction sites”, Autom. Constr. 36, 38–52.
- [7] Barro-Torres S., Fernández-Caramés T.M., Pérez-Iglesias H.J., Escudero C.J., (2012), “Realtime personal protective equipment monitoring system”, Comput. Commun. 36 (1), 42–50.
- [8] Dong S., He Q., Li H., Yin Q., (2015), “Automated PPE Misuse Identification and Assessment for Safety Performance Enhancement”, ICCREM, 3 (1), 204–214.
- [9] Du S., Shehata M., Badawy W., (2011), “Hard hat detection in video sequences based on face features, motion and color information”, 2011 3rd International Conference on Computer Research and Development, 25–29, Shanghai, China, 11-13 March.
- [10] Xinhua J., Heru X., Lina Z., Yanqing Z. (2016), “A Study of Low-resolution Safety Helmet Image Recognition Combining Statistical Features with Artificial Neural Network”, International Journal of Simulation Systems, Science & Technology., 17 (37), 1-6.
- [11] Kang L., Xiaoguang Z., Jiang B., Min T., (2017), “Automatic Safety Helmet Wearing Detection”, IEEE 7th Annual International Conference on CYBER Technology in Automation, 617-622, Hawaii, USA, 31 July-4 August.

- [12] Barnich O., Van Droogenbroeck M., (2009), "ViBe: a powerful random technique to estimate the background in video sequences", 2009 IEEE International Conference on Acoustics, Speech and Signal Processing, 945-948, Taipei, Taiwan, 19-24 Apr.
- [13] Wu J., Geyer C., Rehg J. M., (2011), "Real-time human detection using contour cues", 2011 IEEE International Conference on Robotics and Automation (ICRA), 860-867, Shanghai, China, 9-13 May.
- [14] Zhu Z., Park M., Elsafty N., (2015), "Automated monitoring of hardhats wearing for onsite safety enhancement", International Construction Specialty Conference of the Canadian Society for Civil Engineering (ICSC), 610-615, British Columbia, Canada, 7-10 June.
- [15] Hinton G. E., (2007). "Learning multiple layers of representation". Trends in Cognitive Sciences. 11 (10), 428–434.
- [16] Krizhevsky A., Sutskever I., Hinton G. E., (2012), "ImageNet Classification with Deep Convolutional Neural Networks", Advances in neural information processing systems 25 (2).
- [17] Anwer. A.M.O., (2017), "Derin öğrenme ile Göğüs Kanseri Teçhizi", Yüksek Lisans Tezi, THK Üniversitesi Fen Bilimleri Enstitüsü.
- [18] LeCun Y., Bengio Y., (1995) "Convolutional Networks for Images, Speech, and Time", Neural Information Processing: Research and Development, 1 (1), 276-279.
- [19] Arel I., Rose D. C., Karnowski T. P., (2010), "Deep Machine Learning- A New Frontier in Artificial Intelligence", IEEE Computational Intelligence Magazine, 5 (4), 13 - 18.
- [20] Goodfellow I., Bengio Y., Courville A., Deep Learning, 2th Edition, MIT Press, 2016.
- [21] Girshick R. Jeff D. Trevor D., Jitendra M., (2014), "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation", Computer Vision and Pattern Recognition, Ohio, USA, 24 - 27 June.
- [22] Uijlings J., Van de Sande K., Gevers T., Smeulders A., (2013) "Selective search for object recognition.", IJCV, 73 (3), 154 – 171.
- [23] Girshick R. B., (2015), "Fast R-CNN", IEEE International Conference on Computer Vision, 121-129, Santiago, Chile, 7-13 Dec.
- [24] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu and A. C. Berg, (2016), "SSD: Single Shot MultiBox Detector", ECCV 2016, 21 - 37, Amsterdam, Holland, 11–14 October.

- [25] Joseph R., Farhadi A., (2018), “YOLOv3: An Incremental Improvement”, arXiv, 1804, 2767.
- [26] Web 1, (2019), <https://pjreddie.com/darknet/yolo/>, (Erişim tarihi: 23/06/2019).
- [27] Lin T., Maire M., Belongie S., Bourdev L., Girshick, J., Hays R., Perona P., Ramanan D., Dollár P., Zitnick C., (2014), “Microsoft COCO: common objects in context”, ECCV 2014, 740-755, Zurich, Switzerland, 6-12 September.
- [28] Web 2, (2019), <https://github.com/HCIILAB/SCUT-HEAD-Dataset-Release>, (Erişim tarihi: 23/06/2019).
- [29] Web 3, (2019), <https://github.com/tzutalin/labelImg>, (Erişim tarihi: 23/06/2019).
- [30] Web 4, (2019), <https://github.com/ssaru/convert2Yolo>, (Erişim tarihi: 23/06/2019).
- [31] Web 5, (2019), <https://machinelearningmastery.com/transfer-learning-for-deep-learning/>, (Erişim tarihi: 01/09/2019).
- [32] Web 6, (2019), <https://dataverse.harvard.edu/dataverse/hardhat>, (Erişim tarihi: 23/06/2019).
- [33] Web 7, (2019), <http://pascal.inrialpes.fr/data/human/>, (Erişim tarihi: 23/06/2019).

ÖZGEÇMİŞ

Canibek Sağın 1981 yılında Adana’da doğdu. 2004 yılında Marmara Üniversitesi Elektronik ve Haberleşme Öğretmenliği bölümünden mezun oldu. 2012 yılında Gebze Teknik Üniversitesi Fen Bilimleri Enstitüsü Elektronik Mühendisliği Anabilim Dalında yüksek lisans eğitimine başladı. 2010 yılından beri Arçelik Elektronik işletmesinde Uzman Ar-Ge mühendisi olarak çalışmaktadır.

