Bilgisayar Bilimleri ve Mühendisliği Dergisi

Cilt-16
Sayı-1
2023

Video Verilerinde Bulunan Tehlikeli Nesnelerin Derin Öğrenme Yöntemleri ile Saptanması Üzerine Derleme

Review on Detection of Dangerous Objects in Video Data using Deep Learning Methods

Ayşe Berika Varol MALKOÇOĞLU
Beykoz Üniversitesi
Bilgisayar Programcılığı Bölümü İstanbul, Türkiye
ayseberikavarolmalkocoglu@beykoz.edu.tr
ORCID: 0000-0003-1856-9636

Ruya SAMLİ
İstanbul Üniversitesi-Cerrahpaşa
Bilgisayar Mühendisliği Bölümü İstanbul, Türkiye
ruyasamli@iuc.edu.tr
ORCID: 0000-0002-8723-1228

Öz

Bilgisayarla görme tekniklerinden biri olan nesne saptaması son yıllarda hem akademik hem de ticarî potansiyeli sayesinde büyük ilgi görmektedir. Günümüzde teknolojinin gelişimi ile birlikte güvenlik ya da kişisel amaçlarla çekilen video görüntülerinin artması ve donanım elemanlarının gelişmesi, ihtiyaç duyulan kaynaklara erişimi kolaylaştırmış dolayısıyla nesne saptama sistemlerinin gelişimini hızlandırmıştır. Bu alanda yaya saptaması, yüz tanıma gibi bazı klasikleşmiş konularda çok sayıda çalışma bulunmaktadır. Fakat bu çalışmada farklı nesne gruplarının getirdiği zorlukları gözlemlemek adına tehlikeli nesneler üzerine yapılan ve güvenlik güçlerine yardımcı sistemlerin tasarlanmasına katkı sağlayan çalışmalar araştırılıp derlenmiştir. Çalışmalarda kullanılan nesne saptama yöntemleri geleneksel yöntemler ve derin öğrenme tabanlı modern yöntemler olarak iki kısımda incelenmiş olup avantajları ve dezavantajları tartışılmıştır. Ayrıca literatürdeki eksiklikler belirlenip, gelecekteki çalışmalar için araştırmacılara yönergeler sunulmuştur.

Anahtar Kelimeler: Nesne saptaması, nesne algılama, tehlikeli nesneler, derin öğrenme

Abstract

Object detection, which is one of the computer vision techniques, has been very interested in both academic and commercial potential in recent years. Today, the development of technology, combined with the increased video images for security or personal purposes, and the development of hardware elements, made it easier to access the resources needed, thereby accelerating the development of object detection systems. There are many studies in some classics such as pedestrian detection, face recognition etc. in this area. However, this studies on dangerous objects and contributing to the design of safety-aid systems have been researched and compiled to observe the challenges of different groups of objects in the study. The methods of object detection used in the studies have been studied in two parts as traditional methods and deep learning modern methods, discussing the advantages and disadvantages. In addition, deficiencies in the literature have been identified and guidelines have been provided to researchers for future studies.

Keywords: Object detection, object recognition, dangerous objects, deep learning

1. Giriş

Ateşli silahlar ya da kesici aletlerin kişi/kişilere karşı kullanılması ya da tehdit amacıyla taşınması kanunen suç sayılmaktadır. Fakat ülkemizde ve dünyada bireysel silahlanma ile suç işleme oranı hızla artmaktadır. Gerçekleştirilen bu silahlı saldırılar genellikle sokakta, evde ya da daha sakin alanlarda olsa da insanların bir arada bulunduğu okul, alış-veriş merkezi, düğün salonları, oyun parkı, hastane gibi yerlerde de silahların kullanıldığı ve masum insanların kurban gittiği görülmektedir. Bu durum özellikle kanuna başvurma oranının düşük olduğu bölgelerde ya da kalabalık şehirlerde yaşanmaktadır. Birçok ülkede "bireysel silahsızlanmayı" özendirmek amacıyla çeşitli vakıflar kurulmasına rağmen bu durum hala bir güvenlik sorunu olarak devam etmektedir [1-3]. Olası tehlikelerin önüne geçmek amacıyla yetkililer özellikle kalabalık ortamlardaki suç oranını

Gönderme, düzeltme ve kabul tarihi: 06.10.2022 - 11.01.06.2023 –04..02.2023
Makale türü: Derleme

azaltmak ve caydırıcılığı arttırmak için güvenlik kameraları kurmaktadır. Mevcut kamera sistemleri sayesinde suçlunun saptamasını sağlayabilmektedir. Fakat suçlunun işlediği eylemi geri alamamaktadır. Dolayısıyla silahlı saldırılara kurban giden kişiler zamanında müdahale edilemedikleri için yaralanma ya da hayatlarını kaybetme riskleri ile karşı karşıya kalmaktadır. Bu gibi durumların önüne geçmek amacıyla tercih edilen kapalı devre televizyon sistemlerinin (Close Circuit TeleVision - CCTV) ve şehir izleme sistemlerinin (MOBESE) kullanımı kamu kuruluşlarında, alış-veriş merkezlerinde, bankalarda, otoparklarda, hastane ve fabrikalarda yaygınlaştırılmaktadır. Kameralardan gelen veriler operatörler tarafından 7/24 izlenerek, insanlar ve mülkler için zararlı olma potansiyeli yüksek bireyler ve durumlar incelenmektedir [4]. Bu güvenlik kameralarını izleyen operatörlerin temel görevi olası tehditleri saptamak, kontrol etmek, gözlemlemek ve tehditlere karşı önlem alarak insanların güvenliğini sağlamaktır [5]. Güvenliğin sağlanabilmesi için operatörler 4, 9 ve 16 ekranı aynı anda izlemektedirler. Fakat insanın doğası gereği, aynı anda birden fazla video akışını izlemenin oldukça zor olduğu bilinmektedir. Bu operatörlerin algılama oranları yaşlarına bağlı olarak değişmekle birlikte, yaştan bağımsız her bireyin 1 saat sonra algılama oranı önemli ölçüde düştüğü öne sürülmektedir [6]. Dolayısıyla gün geçtikçe artan CCTV verilerinin manuel olarak incelenmesinin ve analiz edilmesinin oldukça zor olduğu görülmektedir. Bu tarz sistemlerin otomatikleştirilerek operatörlere yardımcı sistemler haline getirilmesi bir ihtiyaç haline geldiği araştırmacılar tarafından fark edilmiştir [7]. Ancak son yıllarda bu tarz çalışmalar yapılmaya başlanabilmiştir. Bunun temel nedeni kameralardan elde edilen verilerin gerçek zamanlı işlenmesi günümüzdeki teknolojilerle (son 10 yılda) ancak mümkün hale gelmesinden kaynaklanmaktadır [8].

Gerçek zamanlı verilerin işlenmesi özellikle derin öğrenme tabanlı modeller ile daha kolay bir şekilde gerçekleştirilmektedir. Bu modeller büyük veriler ile çalışmaktadır. Gerekli olan verilerin saklanması için ihtiyaç duyulan depolama miktarına ve işleme hızına günümüzdeki mevcut yöntemler ile ulaşmak mümkündür. Dolayısıyla 10-15 sene önce donanım yetersizliğinden kaynaklanan problemlerin günümüzde teknolojinin gelişmesiyle aşıldığı ve güvenlik ya da kişisel amaçlarla çekilen video görüntülerinin arttığı bilinmektedir. Hareketli ya da sabit kameralar ile çekilen ve zaman içerisinde artış gösteren bu video görüntüleri, büyük bir veri havuzu oluşturmaktadır. Bu sayede çok sayıda video verilerinin bir araya getirilip incelenmesi ve analiz edilmesi akademik ve ticari potansiyeli ile artan bir ilgi görmektedir.

Bu çalışmada özellikle tehlikeli (silah/bıçak) nesnelerin saptanmasını gerçekleştirip kaynaklara katkı sağlayan derin öğrenme tabanlı çalışmalar incelenerek, kullanılan modeller analiz edilmiş, kaynaklardaki katkıları değerlendirilerek vars eksiklikleri ya da yenilikleri belirtilmiştir. Kaynaklarda bazı benzer çalışmaların yapıldığı görülmektedir. [8]'de araştırmacılar nesne saptama çalışmalarının son 20 yıldaki gelişimini ele alarak kullanılan geleneksel ve derin öğrenme tabanlı modelleri incelemiş, yüz tanıma, metin tarama, yaya algılama gibi çalışmaları değerlendirmiştir. [9] ve [10]'da derin öğrenme tabanlı nesne saptama teknikleri incelenip eksiklikleri ve gelecekte yapılabilecek öneriler sunulmuştur. Video görüntülerindeki zorluklar ve tıkanıklıklar için önerilerde bulunarak araştırmacılara yol göstermişlerdir. [11]'de derin öğrenme yöntemleri ile gerçekleştirilen yaya saptama çalışmaları incelenerek değerlendirilmiştir. Yaya algılamaya yönelik modern yöntemleri incelemiş olan araştırmacılar aynı zamanda farklı veri setlerindeki özellik tabanlı yaklaşımları karşılaştırarak veri setlerinin önemini vurgulamışlardır. Bu çalışmada ise benzerlerinden farklı olarak sadece tehlikeli nesnelere odaklanılmıştır. Dolayısıyla çalışmadaki hedef nesnelerin boyutlarının küçük olması özellikle hareketli görüntüler üzerinde algılamasını zorlaştırmaktadır. Faklı nesnelerin farklı zorlukları beraberinde getirdiği bilindiği için bu çalışma araştırmacılara diğer derleme çalışmalarından daha farklı bir bakış açısı sunacağı düşünülmektedir. Yalnızca tehlikeli nesnelere odaklanılarak yapılmış çalışmalar dikkate alınarak; kullanılan teknikler, veri setleri, çalışma ortamları ve donanım gereksinimleri incelenip detaylı analiz edilmiştir.

Çalışma 6 bölüm şeklinde organize edilmiştir. 2. Bölümde nesne saptama kavramı açıklanmış ve hangi amaçlar için kullanıldığı konusuna değinilmiştir. 3. Bölümde derin öğrenme tabanlı algoritmalar incelenmiştir. 4. Bölümde bu algoritmalar kullanılarak tehlikeli nesnelerin saptanmasını gerçekleştiren çalışmalar incelenip kategorize edilmiştir. 5. Bölümde yapılan çalışmaların eksiklikleri ve yayınlara katkıları üzerine bir tartışma gerçekleştirilip son bölümde sonuç ve gelecekteki çalışmalardan bahsedilmiştir.

2. Nesne Algılama

İnsanlar önceden bildiği bir nesne ile tekrar karşılaştığında bunun ne olduğunu anında algılayabilir ve tanımlayabilir. Nesne algılama işlemi insanın bu özelliğinden esinlenerek tasarlanmış ve bilgisayarlara bu eylemi yaptırmak için geliştirilmiş bir bilgisayarla görü tekniğidir. Temel amacı görüntü veya videolardaki nesneleri saptama edip konumlandırarak sınıflandırabilmektir. Yıllar içerisinde bu alanda birçok algoritma geliştirilse de 2001 yılında Viola ve Jones tarafından geliştirilen ve kendi isimleri ile anılan Viola-Jones algoritması bu alanda devrim yaratmıştır [12]. Araştırmacılar, bu yöntem ile, 24×24 beneklik insan yüzü içeren ve insan yüzü içermeyen eğitim seti oluşturup tasarladıkları modeli eğiterek görseldeki insan yüzlerini algılayabilen bir model tasarlamışlardır [13]. Zaman içerisinde donanımın gelişmesiyle birlikte tasarlanan nesne algılama modellerinin veri işleme hızı ve başarısı artarak devam etmiştir. Günümüzde yaygın olarak kullanılan nesne algılama modellerinden, otonom sürüş [14-16], yaya saptaması [17-20], yüz tanıma [12, 21-23] gibi pek çok alanda yararlanılmaktadır.

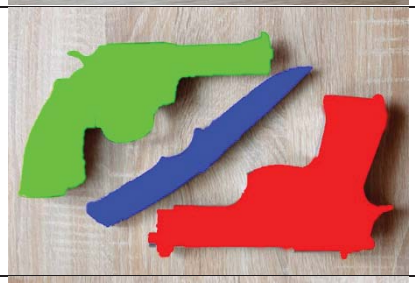
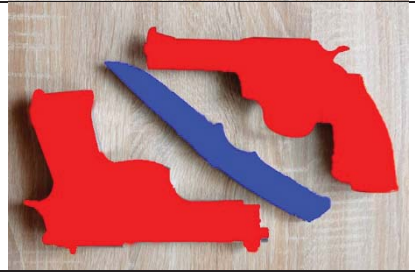
2.1 Nesne Algılamanın Görevleri

Literatürde nesne algılama ve nesne saptaması üzerine birçok çalışma mevcuttur. Ancak zaman zaman bu terimlerin karıştırıldığı ve birbiri yerine kullanıldığı görülmektedir. Çünkü bu iki terim nesneleri tanımlamaya yönelik benzer

tekniklerdir. Temelde nesne saptaması görüntüdeki hedef nesne/nesneleri bulma işlemini gerçekleştirir ve nesne algılamanın alt kümesidir. Bu çerçevede nesne algılama birçok görevin çatısını oluşturmaktadır. Görüntü içerisindeki tüm nesneleri tanıma ve konumlandırma işlemi yapan nesne algılama yönteminin gerçekleştirilmesi sırasında bazı teknik görevler mevcuttur [24, 25]. Bu teknikler örneklerle birlikte Çizelge-1'de açıklanmaktadır.

Tekli ve çoklu nesnenin saptanması sırasında sınıflandırma + yerelleştirme ya da nesne saptaması işlemi kullanılır. Bu iki yöntem benzer mantıkta çalışır fakat sınıflandırma + yerelleştirme görüntüde tek bir nesne varsa ya da nesnelerin sayısı biliniyorsa doğru çalışabilmektedir ancak bu durum ağ performansının düşmesine sebep olur. Çünkü amaç sadece nesnenin sınıfını belirlemek değil aynı zamanda sınırların çizilebilmesi için 4 koordinat noktasının tahmini bilgilerini de belirlemektir. Dolayısıyla nesne sınıflandırma + yerelleştirme modelinin eğitimi için; önceden etiketlenmiş görüntü verileri ve bu görüntü verilerinde yer alan nesnelerin sınır koordinatları yer almaktadır.

Çizelge-1: Nesne algılamada yer alan teknik görevler

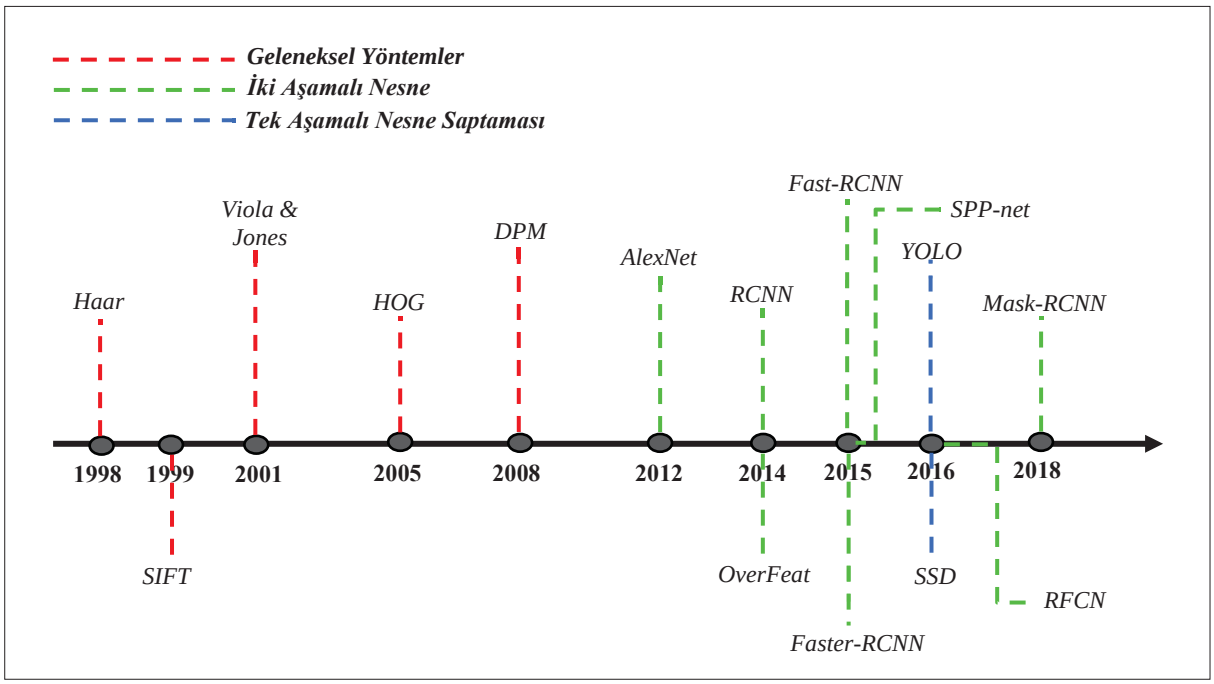
Görev		**Açıklama**		**Örnek**
Tekli Nesne	Nesne Sınıflandırma	Önceden etiketli veriler ile eğitilmiş modele bir girdi verildiğinde bu girdi verisini bazı metriklere göre sınıflandırır.		
	Nesne Yerelleştirme	Görüntüdeki nesnenin konumunu saptama ederek sınırlayıcı bir kutu ile temsil eder. Sınırlayıcı bir kutu çizmek için x koordinatı, y koordinatı, yükseklik ve genişlik olmak üzere 4 sürekli sayı kümesini tahmin eder.		Bu eylem ancak sınıflandırılma işleminden sonra hedef nesne/nesneler için gerçekleşir.
	Sınıflandırma + Yerelleştirme	Görüntüde yalnızca bir nesne varsa ya da verideki nesnelerin sayısını biliniyor ise nesnelerin yerleri saptama edilip sınırlarının çizilmesini sağlar.		
Çoklu Nesne	Nesne Saptaması	Görüntüdeki tüm nesnenin saptanıp sınırlarının belirlenmesini sağlar.		
	Görüntü Bölütleme	Görüntüdeki her nesne için oluşturulan benek bazında maskeler, bir nesnenin varlığının işaretlenmesini sağlar. Sınırlayıcı kutudan daha etkili nesne algılama yöntemidir.	*Örnek bölütleme: Aynı sınıfın birden çok örneği ayrı bölütlenir. Tüm nesneler aynı sınıfa ait olsalar bile farklı renklerle renklendirilir.*	
			Anlamsal bölütleme: Aynı sınıftaki tüm nesneler için tek bir sınıf oluşturulur, bu nedenle aynı sınıftaki tüm nesneler aynı renkle renklendirilir.	

Nesne sınıflandırma sırasında nesnenin hangi sınıfa ait olduğu tahmin edilirken, yerelleştirme sırasında nesnenin çevresindeki sınırlayıcı kutunun x koordinatı, y koordinatı, yükseklik ve genişlik bilgileri olmak üzere 4 sürekli sayı kümesi

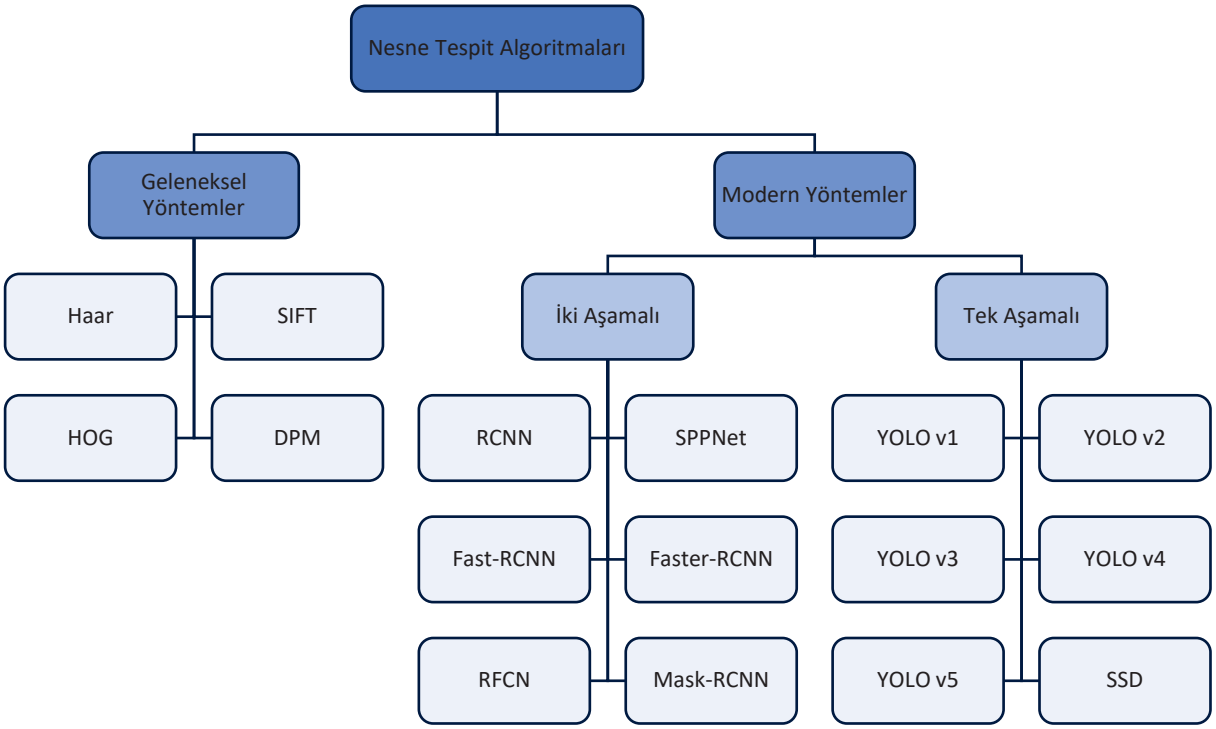
tahmin edilmelidir. Bu nedenle evrişimli sinir ağı hem nesne sınıfını hem de sınırlayıcı kutuya ait koordinat bilgisini tahmin etmeyi öğrenecektir [26, 27]. Eğer görüntüde birden fazla nesne varsa ve nesnenin mevcut sayısı bilinmiyorsa, sistemin üreteceği sınırlayıcı kutu sayısı dolayısıyla koordinat sayısı bilinemeyecektir. Bu noktada nesne saptama yöntemi kullanılması gerekmektedir. Bu yöntemi etkili bir şekilde uygulayabilmek için çeşitli derin öğrenme tabanlı algoritmalar mevcuttur.

3. Nesne Saptama Algoritmaları

Nesne saptama yöntemleri geleneksel ve derin öğrenme tabanlı yöntemler olarak iki kısımda incelenebilir. Şekil-1'de nesne saptaması için geliştirilen algoritmaların zaman çizelgesi gösterilirken Şekil-2'de kaynaklarda kullanılan algoritmalar gösterilmiştir. Şekil-1'de görüldüğü üzere, 2014 yılından önceki dönemde geleneksel yöntemler, sonraki dönemde ise derin öğrenmeye dayalı yöntemler ortaya çıkmıştır [8].



Şekil-1: Nesne saptaması için geliştirilen algoritmaların zaman çizelgesi



Şekil-2: Kaynaklarda taranan algoritmaların diyagramı

3.1 Geleneksel Yöntemler

Geleneksel nesne saptama algoritmaları görüntü içerisindeki nesnelerin saptanıp özelliklerinin çıkartılmasından sonra sınıflandırılmasıyla oluşan 3 temel adımdan oluşmaktadır.

3.1.1 Bölge Seçimi

Çok ölçekli sürgülü pencere yardımı ile tüm görüntü taranarak farklı boyutlarda ve farklı konumlarda olan nesnelerin saptanması gerçekleştirilir. Sabit boyutlu bir pencere görüntü üzerinde kaydırılarak ilgili bölge aranmaktadır. Fakat bu yöntem oldukça zaman almakta ve hatalı sonuçlar üretebilmektedir [28]. Buna rağmen hala bazı nesne saptama işlemlerinde kullanılan bir yöntemdir.

3.1.2 Özelik Çıkarımı

Bölge seçiminde belirlenen nesneye ait özelliklerin çıkartılması, doğru sınıflandırma için oldukça önemlidir. Fakat nesneye ait özellik çıkarımı yapılırken farklı aydınlatma koşulları ve farklı arka planlar yapılan işlemleri zorlaştırabilir. Bu nedenle görüntülerin özelliklerini elle tanımlamak yerine Haar, Ölçek Değişmez Unsur Dönüşümü (Scale-Invariant Feature Transform - SIFT), Yönlü Gradyanlar Histogramı (Histograms of Oriented Gradients - HOG), Değişime Uğrayabilen Parça Bazlı Model (Deformable Part Model - DPM ve benzeri özellik çıkarım teknikleri kullanılmaktadır [9]. Bu teknikler aşağıda kısaca açıklanmıştır.

Haar: Bir görüntüde belirlenmiş iki veya daha fazla bitişik dikdörtgen bölgenin benek değerlerinin toplamının farkı ile elde edilen özellik çıkarım algoritmasıdır. Görüntüyü dikdörtgenler yardımı ile tarayarak dikdörtgenlerin içerdiği beneklerin toplamlarına ya da farklarına eşik değeri uygulanır [29,30].

SIFT: Özellik çıkarımı, bir nesnenin veya sahnenin Gaussian filtresi sonucunda elde edilen farklı görünümleri arasındaki değişmeyen özelliklerini keşfedilmesiyle sağlanır. Görüntünün döndürme, ölçeklendirme ve aydınlatma gibi işlemler sonucunda değişmeyen bölgesel özelliklerini Gaussian filtresi yardımıyla keşfeden 4 adımlı etkili bir algoritmadır [31].

HOG: Görüntü gradyanını bir görüntüde meydana gelen yoğunluktaki veya renkteki yönlü değişiklikleri temsil etmektedir. Görüntü gradyan vektörleri, en büyük yoğunluk artışının meydana geldiği yönü simgeleyen işaretçilerdir ve temel amaç görüntülerden bilgi almaktır. HOG algoritması, görüntüdeki beneklerin yönelim ve büyüklük değerlerini kullanarak özellik haritası çıkaran bir algoritmadır. Görüntünün yatay ve dikey gradyanları hesaplanarak kenarlar ve önemli noktalar bulunur. Görüntü hücrelere bölünür, hücre içerisindeki her bir beneğin gradyan büyüklüğü sahip olduğu açıya göre histogram bölgelerine dağıtılır. Sonuç olarak, bir blok içerisinde oluşturulan tüm histogramların birleştirilerek büyük bir histogram elde edilir [32,33]. HOG algoritması 3 temel adımdan oluşmaktadır:

- Yeniden boyutlandırma ve renk normalleştirme
- Her beneğin gradyan vektörü, büyüklüğü ve yönün hesaplanması
- Görüntünün birden fazla 8×8'lik beneklere bölünmesi

DPM: Aydınlatma, bakış açısı, tıkanıklık gibi çeşitli sebeplerden dolayı nesne algılama sırasında çeşitli varyanslar olabilmektedir. DPM bu değişimleri yakalayarak nesne saptama işlemini başarılı bir şekilde gerçekleştirebilmektedir [34]. Model 3 ana kısımdan oluşmaktadır:

- Kök (root) filtresi, tüm nesneyi kaplayan bir pencereye sahiptir bu pencere ile bölge özelliği vektörü için ağırlıkları belirler.
- Parçalı filtre, nesnenin daha küçük bölümlerini kapsar.
- Mekânsal model, köke göre parça filtrelerinin konumlarını puanlamak için kullanılır.

Ayrıca filtrelemeden önce piramit seviyelerinde HOG özelliklerini ve bir nesnenin farklı parça konumlarını bulmak için eğitim sırasında doğrusal Destek Vektör Makinesi (Support Vector Machine - SVM) kullanır.

3.1.3 Nesnenin Sınıflandırılması

Özellikleri çıkartılan nesnelerin hangi sınıfa ait olduklarının belirlenmesi gerekmektedir. Bu işlem için SVM veya Adaboost algoritması kullanılmaktadır.

SVM: Sonsuz boyutlu alanda oluşturulan bir hiper-düzlem kümesi ile verilerin sınıflandırma yapılarak ayrıştırılmasını sağlayan denetimli makine öğrenme algoritmasıdır. Bu algoritma eldeki verileri kullanarak iki veya daha fazla sınıfı birbirinden ayırabilen bir yapıya sahiptir. 2 boyutlu düzlemlerde çözülemeyen problemler Kernel hilesi yöntemi (boyut arttırıyormuş gibi yapar) kullanılarak çözümlenebilmektedir [35,36].

AdaBoost: Topluluk yöntemini kullanarak sınıflandırma yapan makine öğrenmesi algoritmasıdır. Zayıf sınıflandırıcıların hatalarından ders çıkarmak ve onları güçlü olanlara dönüştürmek için yinelemeli bir yaklaşım kullanmaktadır [13].

Geleneksel nesne algılama yöntemleri hala kullanılmasına rağmen bazı eksiklikler içermektedir. Örneğin; bölge seçimi için kullanılan sürgülü pencere tekniği görüntüyü tarayabilmek için çok fazla pencere kullanır. Bu sebepten hesaplama maliyeti oldukça yüksektir. Ayrıca özellik çıkarım ey öntemleri görüntülerde oluşabilecek morfolojik çeşitlilikler yüzünden her zaman iyi haritalama gerçekleştiremeyebilir. Bu gibi işlevsel sorunların önüne geçilmesine 2014 yılı itibari ile gerçekleştirilmeye başlanmıştır. Araştırmacılar nesne saptaması sırasında ağırlıklı olarak derin öğrenme tabanlı yöntemler kullanmaya ve geliştirmeye odaklanmıştır.

3.2 Modern Yöntemler

1995 yılında geliştirilmiş olan SVM algoritmasından elde edilen sonuçlar problemlere çözüm getirebilecek doğruluk oranlarına sahip olduğu için makine öğrenmesi algoritmalarının gelişimi bir süre durağan şekilde varlığını

sürdürmeye devam etmiştir. Bu durum 2009 yılında Hinton tarafından geliştirilen çok katmanlı Derin Sinir Ağı (Deep Belief Networks - DBNs) modeli sayesinde değişerek denetimsiz öğrenme modelini ortaya çıkarmıştır [37]. Ardından veri kümelerinin büyümesi, bilgisayar donanımının gelişmesi ve doğru aktivasyon fonksiyonlarının tercih edilmesi ile makine öğrenmesi yönteminin bir alt kümesi olan çok katmanlı yapılardan meydana gelen derin öğrenme yöntemleri geliştirilmiştir. ImageNet tarafından nesneleri sınıflandırabilmek için yapılan yarışmalar ile daha popüler olmaya başlayan derin öğrenme yöntemi, Krizhevsky ve ark. tarafından 2012'de tasarlanan 7 katmanlı AlexNet ağı ile araştırmacıların dikkatini çekmiştir [38]. Görüntülerin sınıflandırılması için kullanılan bu yöntemin nesne saptama işlemlerinde de kullanıp kullanamayacağını sorgulamaya başlamış ve 2014 yılında Bölge Tabanlı Evrişimli Sinir Ağı (Region Based Convolutional Neural Networks - RCNN) ve OverFeat adında iki farklı nesne saptama mimarisi tanıtılmıştır [39-41]. Tanıtılan bu iki mimari sayesinde nesne saptama işlemleri derin öğrenme yöntemlerine dayalı tekniklerle gerçekleştirilmeye başlanmıştır. Zaman içerisinde tasarlanan yeni teknikler ile nesne saptaması güçlü bir şekilde geliştirilmiş ve geliştirilmeye devam etmektedir [42].

Bu çalışmada belirtilen algoritmaların içerikleri, doğruluk ve veri işleme hızlılarının avantajları – dezavantajları algoritmaların özelliklerine göre iki farklı başlık altında tartışılmıştır.

3.2.1 İki Aşamalı Nesne Saptaması

Geleneksel yöntemde olduğu gibi modern yöntemlerde de bir nesnenin saptanması için görüntüdeki yeri, özellikleri ve ardından hangi sınıfa ait olduğu bulunmalıdır. Nesnelerin potansiyel konumlarını gösteren yerleri keşfedebilmek için bölge öneri yöntemi kullanılmaktadır. Sabit boyutlu özellikler bu bölgelerden beslenerek elde edilmektedir. Ardından özellikler sınıflandırıcılara gönderilerek nesnenin sınıfını tahmin etmektedir. İki aşamalı yöntemlerin özellik çıkarım sürecinde yavaş olduğu fakat geleneksel yöntemlere göre daha yüksek doğrulukta sonuç ürettiği bilinmektedir [42, 43].

3.2.1.1 Bölge Tabanlı Evrişimli Sinir Ağları

Girshick vd. tarafından 2014 yılında geliştirilmiş olan bu mimari temelde bölge önerisi, özellik çıkarımı ve sınıflandırma + yerelleştirme olarak 3 ana bölümden oluşmaktadır (Şekil-3).

Bölge önerisi: Bölge önerisi bir görüntüdeki olası nesnelerin yerini belirleme işlemidir. Bu yöntem ile görüntüde çok sayıda bölge seçilmektedir. Seçilen bölgeler Seçici Arama algoritması ile 2000 aday bölgeye indirgenerek evrişimli katmana gönderilmektedir [43]. Belirlenen bölgelerin seçici arama algoritması ile azaltılması işlemi ise temelde Şekil-4'te bir örneği gösterildiği gibi nesnelere farklı renkler atayıp görüntüden ayıran bir yöntemdir [44]. Algoritmanın çalışma prensibi aşağıdaki gibi ifade edilebilir:

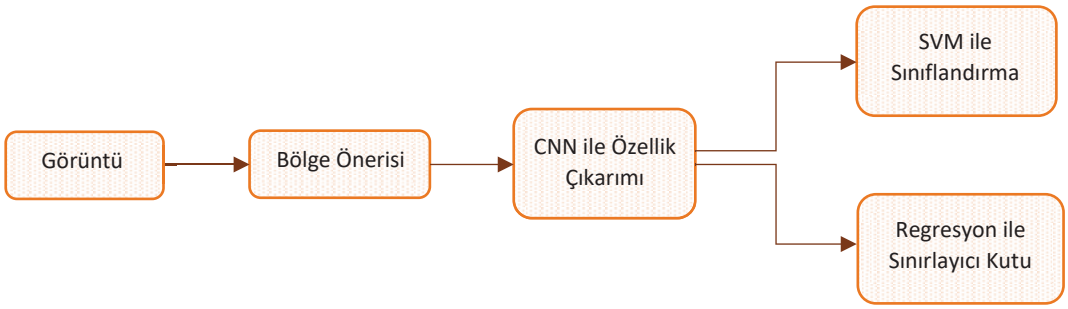
- Görüntü bölütlenerek birçok aday bölge belirlenir.
- Açgözlü özelliğe sahip bir algoritma kullanılarak benzer renklerdeki bölgeler tekrar tekrar birleştirilir.
- Nihai aday bölge önerilerini üretmek için oluşturulan bölgeler kullanılır.

Şekil-4: Seçici arama algoritması ile insan saptaması [44]

Özellik çıkarımı: Bölge önerisi ile belirlenen yaklaşık 2000 adet aday bilginin her biri ayrı ayrı Evrişimli Sinir Ağı (Convolutional Neural Networks - CNN) modeline girdi olarak gönderilmektedir. CNN modeli ile her bölgeden sabit uzunlukta özellik çıkarma işlemi gerçekleştirilerek sınıflandırılması için SVM modeline ve sınırlayıcı kutuların belirlenebilmesi için bir regresyon modeline gönderilmektedir [39].

Sınıflandırma + yerelleştirme: Evrişimli katmanlardan elde edilen özellikler sınıflandırılması için bir SVM algoritmasına gönderilmektedir. Sınıflandırılacak nesnenin çerçevelenebilmesi için sınırlayıcı kutunun koordinatlarını tahmin edecek bir regresyon modeli belirlenmektedir [39]. RCNN modeli derin öğrenme tabanlı nesne saptama algoritmalarının atası olarak bilinmektedir. RCNN algoritmasının performansı geleneksel nesne saptama algılarıma göre daha başarılı sonuçlar ürettiği bilinse de bu modelin bazı dezavantajları bulunmaktadır [8]. Bu dezavantajlar aşağıdaki gibi ifade edilebilir:

- Görüntülerin bölge önerisi ile birlikte saklanması gerektiği için çok fazla depolama alanı gerektirmektedir [46].
- Her görüntüden 2000 aday bölge önerisinin sınıflandırması, zaman kaybına sebep olmaktadır [46].
- Model sadece 227×227 boyutunda girdi verisi alabildiği için nesne bilginin kaybolmasına yol açabilmektedir [8, 10].
- Bölge önerisi çıkarımı sırasında gereksiz birçok teklif ürettiği için zaman karmaşıklığının artmasına sebep olmaktadır [45, 46].



Şekil-3: RCNN mimarisinin akış diyagramı

3.2.1.2 Mekânsal Piramit Havuzlama Ağı

He ve ark. tarafından 2015 yılında geliştirilen bu mimarinin temel amacı sabit boyutlu girdilere mecburiyeti ortadan kaldırmak ve görüntü boyutu ne olursa olsun çıktı olarak sabit uzunlukta bir temsil oluşturabilmektir. CNN mimarileri giriş verisi olarak görüntüleri sabit boyutta almaktadır. Fakat bu yaklaşımı gerçekçi ve kolay bulmayan araştırmacılar Mekansal Piramit Havuzlama (SPP-net) adını verdikleri bir strateji kullanarak tam bağlantı katmanından kaynaklanan sabit boyutlu girdi problemini ortadan kaldırmışlardır.

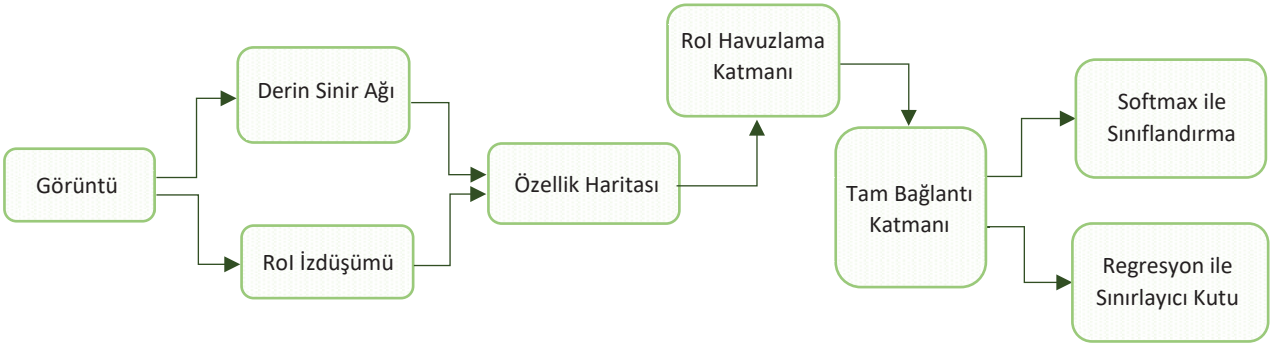
Mekansal piramit havuzlama: Rastgele boyutlarda görüntüler ile ağı besleyebilmek için son havuzlama katmanı ile mekansal piramit havuz katmanı yer değiştirilmektedir. SPP katmanları yalnızca bir havuzlama işlemi uygulamaz. Birkaç farklı çıktı boyutunda havuzlama işlemi uygular ve sonuçları bir sonraki katmana göndermeden önce birleştirir. Bu sayede tam bağlantılı katmanı için (veya diğer sınıflandırıcılara) sabit uzunlukta çıktılar üretilmektedir [47].

RCNN'deki özellik hesaplaması görüntü başına binlerce bölgenin ham beneklerine tekrar tekrar uyguladığı için zaman alır, oysa SPP-net evrişim katmanlarını yalnızca bir kez çalıştırdığı için daha hızlıdır. Ayrıca RCNN'ye kıyasla sınırlayıcı kutu tahmin hızı daha başarılıdır. Bu iyileştirmelere rağmen mevcut dezavantajları aşağıdaki gibi ifade edilebilir:

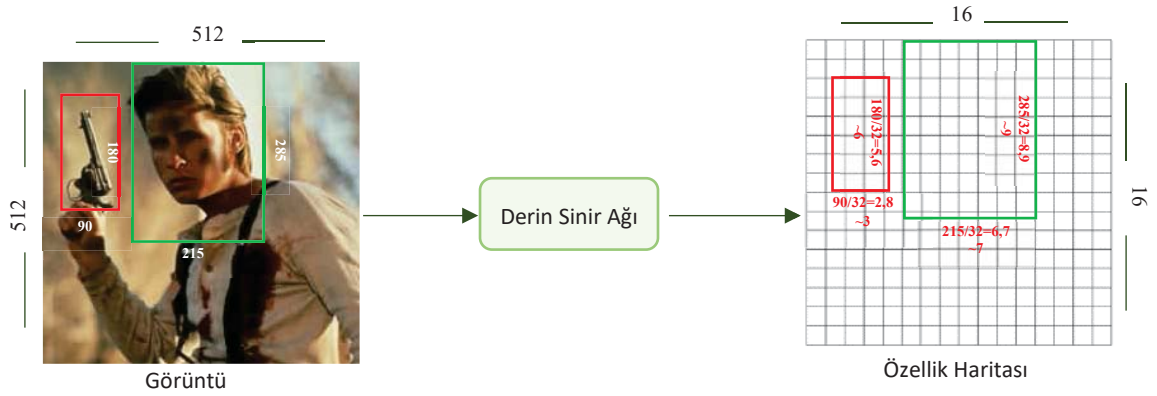
- Özellik çıkarma ve ağın ince ayarlarının yapılandırılması zaman almaktadır.
- Bölge önerilerinden dolayı fazla yer kaplamaktadır.
- RCNN gibi, özelliklerin çıkarılması, ince ayar yapılması, SVM ile eğitmeyi ve son olarak sınırlayıcı kutu yerleştirmeyi içeren çok aşamalı bir işlem hattından oluşmaktadır [46].

3.2.1.3 Hızlı Bölge Tabanlı Evrişimli Sinir Ağları

RCNN'de kullanılan yaklaşık 2000 adetlik aday bölgeler eğitim zamanını ve maliyeti arttırdığı bilinmektedir. Bu soruna çözüm üretebilmek adına Girshick 2015 yılında RCNN mimarisini geliştirerek zamandan ve maliyetten tasarruf etmeyi başaran Hızlı Bölge Tabanlı Evrişimli Sinir Ağı (Fast-RCNN) mimarisini tasarlamıştır. Girshick bu sefer görüntüyü bölge tavsiyelerine göre ayırmak yerine tüm görüntü ve bölge tavsiyelerini CNN mimarisine girdi olarak vermektedir. Şekil-5'de gösterildiği gibi görüntü CNN'e gönderilerek özellik haritası üzerinde bölge önerilerinin izdüşümleri alınmaktadır. Buradan elde edilen bilgiler RoI (Region of Interest - İlgi Alanı) havuzlama katmanına gönderilerek her bölge önerisi için sabit uzunlukta $(3 \times 3,\ 5 \times 5,\ 7 \times 7)$ bir özellik vektörü elde edilmektedir. Her özellik vektörü önce tamamlama katmanına sonra sınıflandırma için Softmax ve sınırlayıcı kutu için regresyon modeline gönderilmektedir. Böylece özellik haritalarını ayrıca saklamaya gerek kalmadığı için depolamadan da tasarruf sağlamaktadır [46].



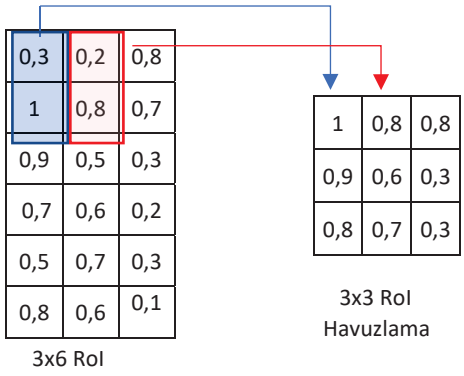
Şekil-5: Fast-RCNN mimarisinin akış diyagramı



Şekil-6: Özellik haritasında RoI'lerin izdüşümü, özellik haritasında silah ~3 × 6, yüz ~7 × 9 büyüklüğünde yer alır. Buna ek olarak x ve y koordinatlarında benzer şekilde hesaplanmaktadır.

RoI havuzlama katmanı: Görüntüden önerilen bölgelere RoI denilmektedir. Orijinal görüntü bir derin sinir ağına (VGG16, ResNet-50 vs.) verilerek özellik haritası çıkartılmaktadır. Bu özellik haritası üzerinde olası nesnelerin izdüşümleri bulunarak RoI'ler elde edilmektedir (Şekil-6).

Harita üzerindeki konumu ve sınırlayıcı kutunun büyüklüğü özellik haritası boyutunda hesaplanır. Bu yapının tam bağlantı katmanına gönderilebilmesi için farklı boyuttaki RoI'lerin aynı boyutta olması gerekmektedir. Şekil-6'da görüldüğü gibi belirlenen nesnelerin büyüklükleri birbirinden farklıdır. Her birinin sabit boyutta olabilmesi tam bağlantı katmanı için önemlidir. Bu sebeple elde edilen tüm RoI'ler, havuzlama katmanından geçirilerek eş boyutlara indirgenmektedir. RoI havuzu temelinde maksimum havuzlama yaparak farklı boyutlardaki girdileri sabit boyutlu (genelde 7×7) özellik haritalarına dönüştürmektedir. Şekil-7'de bir RoI'nin havuzlama katmanından geçirilmesine ilişkin örnek görsel yer almaktadır. Bu örnekte gösterilen RoI ~3×6 boyutundadır. Özellik haritasını 3×3 olarak belirledikten sonra uygun boyutlu filtre (bu örnek için 1×2) hesaplanarak maksimum özellikli havuzlama katmanına gönderilir. Ardından elde edilen $3 \times 3 \times 512$ boyutlu bu bilgi, tam bağlantı katmanına gönderilmektedir [46]. Özellik çıkarımı sırasında RoI'nin harita üzerinde kapladığı alan benek bazında belirlenirken bazı veri kayıpları yaşanabilmektedir. Bu veri kayıpları beneklerin tam sayı ile ifade edilmesinden kaynaklanmaktadır. Bu sebepten RoI boyutları belirtilirken yaklaşık (~) ifadesi kullanılmıştır. Yaşanan bilgi kayıplarının yanı sıra bilgi kazanımları da yaşanabilmektedir fakat kayıplar nesne sınıflandırma için daha önemli olabilmektedir. Bu durumun önüne geçip daha hassas özellik haritalarının çıkartıla bilinmesi için niceleme yöntemi kullanmayan RoIAlign yöntemi kullanılabilir [48].



Şekil-7: ~3×6 boyutundaki RoI'nin havuzlama katmanından sonraki dönüşümü

Tam bağlantı katmanı: Bu katman kendisine gelen bilgileri tek boyutlu vektöre dönüştürerek sınıflandırma ve regresyon katmanına göndermektedir.

Sınıflandırma ve regresyon: Tam bağlantı katmanında tek boyutlu vektöre dönüştürülen bilgiler sınıflandırma için Softmax, sınırlayıcı kutu için lineer regresyon algoritmasına gönderilmektedir.

Fast-RCNN, RCNN, SPP-net'ten daha yüksek algılama kalitesine başka bir deyişle Ortalama Hassasiyete (mAP) sahip olduğu bilinmektedir. Özellikle depolama alanından tasarruf ettiği ve diğer iki modele göre görüntü başına yalnızca bir işlem yaptığın için daha hızlı eğitim sürecinin olduğu bilinmektedir [46]. Fakat bu yöntemin kendine göre bazı dezavantajları bulunmaktadır. Fast-RCNN, RCNN'ye benzer şekilde, bölgenin tekliflerini bulmak için seçici arama algoritması kullanmaktadır. Dolayısıyla RCNN'den daha hızlı olsa bile hala yavaş ve zaman alıcı bir sürece sahiptir.

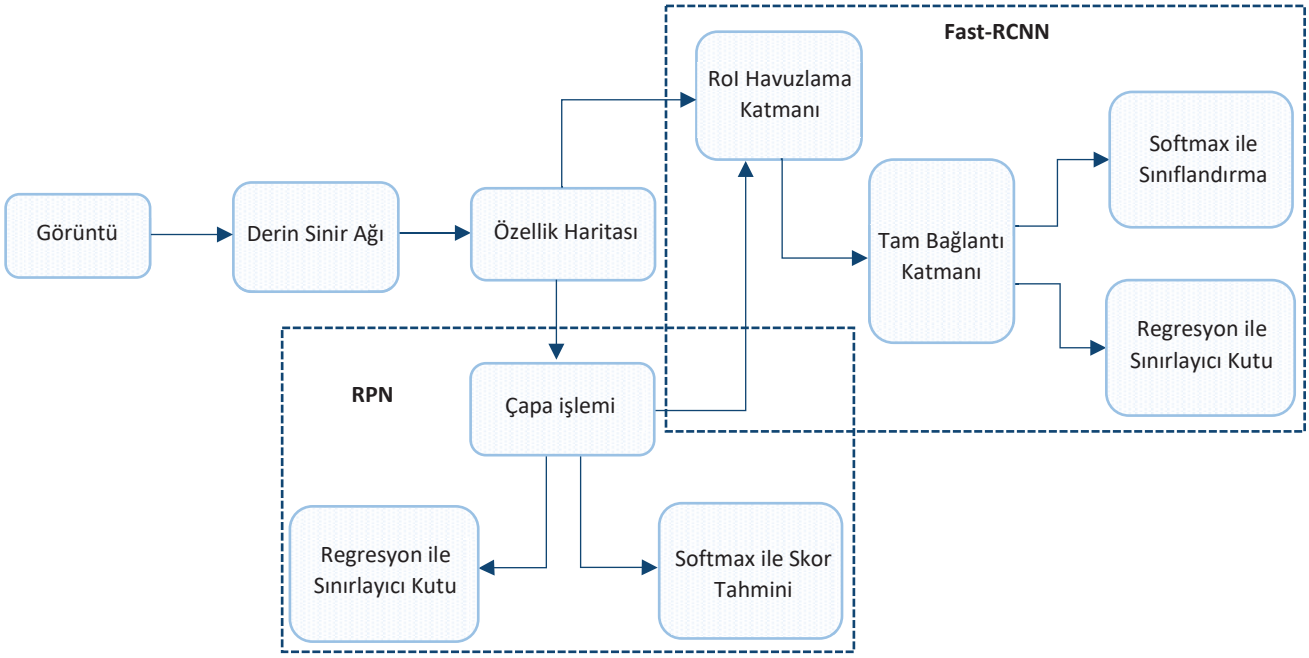
3.2.1.4 Daha Hızlı Bölge Tabanlı Evrişimli Sinir Ağları

SPP-net ve RCNN'de bir darboğaz haline gelen bölge önerisinin maliyetini ortadan kaldırmak için Ren ve diğerleri Bölge Öneri Ağını (RPN) önermişlerdir. Bu mimari sayesinde bölge önerilerinin ürettiği maliyet yok denecek kadar azaltılarak bölge öneri süresini 2 saniyeden 10 milisaniye'ye düşürmeyi başarmışlardır. Aynı zamanda bölge önerisi aşamasının özellik temsilinde genel bir iyileşmeyi sağlamışlardır. Daha hızlı Bölge Tabanlı Evrişimli Sinir Ağı (Faster-RCNN) yapısı Şekil-8'de gösterildiği gibi RPN ve Fast-

RCNN yapılarının birleştirilmiş halini temsil etmektedir [49]. Faster-RCNN modelinin çalışma şekli aşağıdaki gibi ifade edilebilir:

- Görüntünün herhangi bir derin sinir ağı (VGG16, ResNet50, AlexNet vs.) ile özellik haritası çıkartılır.
- RPN katmanına gönderilen özellik haritası, $n \times n$ ($3 \times 3, 5 \times 5$ vs.) boyundaki filtre ile N (224, 512 vs.) adet evrişim katmanından geçirilir.
- Elde edilen bilgiler olası nesnenin saptanması için Softmax katmanına ve sınırlayıcı kutu için regresyon katmanına gönderilir.
- RPN katmanından çıkan veriler ve derin sinir ağından elde edilen özellik haritası, RoI havuzlama katmanına gönderilir.
- Farklı boyutlardaki verilerin tamamlama katmanı tarafından anlaşıla bilinmesi için her bölgeden sabit uzunlukta özellik vektörü çıkarılır.
- Çıkarılan öznitelik vektörleri Fast-RCNN'deki Softmax katmanı sayesinde sınıflandırılır, regresyon katmanı sayesinde ise sınırlayıcı kutuları çizilir.

RPN: Nesne sınırlarını ve her bir konumdaki puanları aynı anda tahmin etme yeteneğine sahip olan tamamen evrişimli bir ağ yöntemidir. Herhangi bir ölçekteki görüntüden bölge önerileri çıkartmakta ve önerilen bölgelere tahmin etme doğruluğu için skor ataması yapmaktadır. RPN yönteminin temelinde çıpa kutuları yer almaktadır. Bu kutular sayesinde görüntüde algılanan farklı boyuttaki nesnelerin olası konumları saptanabilmektedir. Uçtan uca eğitilebilir bir model olduğu için seçici arama algoritmasına göre daha iyi bölge önerileri sunmaktadır. Ayrıca RPN ve Fast-RCNN mimarisinde aynı evrişimli katmanlar kullanılarak görüntü işlenebildiği için seçici arama algoritmasına göre daha hızlı bölge önerileri üretebilmektedir. Aynı evrişim katmanlarını kullanmaları sayesinde ise, iki yöntem birleştirilerek eğitim sadece bir kere gerçekleştirilmektedir [49].



Şekil-8: Faster-RCNN mimarisinin akış diyagramı

3.2.1.5 Bölge Tabanlı Tam Evrişimli Ağlar

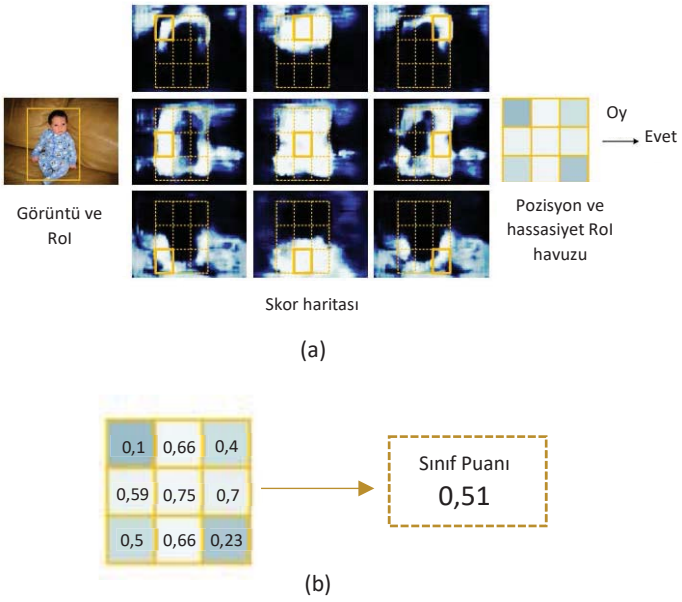
Etkili ve verimli nesne tanıma algoritması sunmayı hedefleyen Dai ve ark. 2016 yılında RFCN mimarisini geliştirmişlerdir. Bu algoritma temelde Tam Evrişimli Ağlar (FCN) mimarisine dayanmakta olup benzer şekilde paylaşılan evrişimli mimarilerden oluşmaktadır. Bir dizi özel evrişimsel katman ile birçok konuma duyarlı puan haritası oluşturarak her biri göreceli bir uzamsal konuma göre hesaplanan özellik bilgisi sayesinde Fast/Faster RCNN'den daha hızlı ve verimli çalıştığını göstermişlerdir [50].

Bu algoritma şu şekilde çalışmaktadır:

- Görüntünün ResNet derin öğrenme ağı ile özellik haritası çıkartılır.
- Özellik haritası RPN ve RFCN arasında özellikler paylaşılır.
- RPN ile özellik haritalarında aday bölgeler belirlenir.
- Belirlenen her bir aday bölgenin 3×3'lük haritası çıkartılır. Bu harita sol üst, orta üst, sağ üst, sol orta, sağ alt alanını algılayan 9 bölge tabanlı özellik haritasıdır.

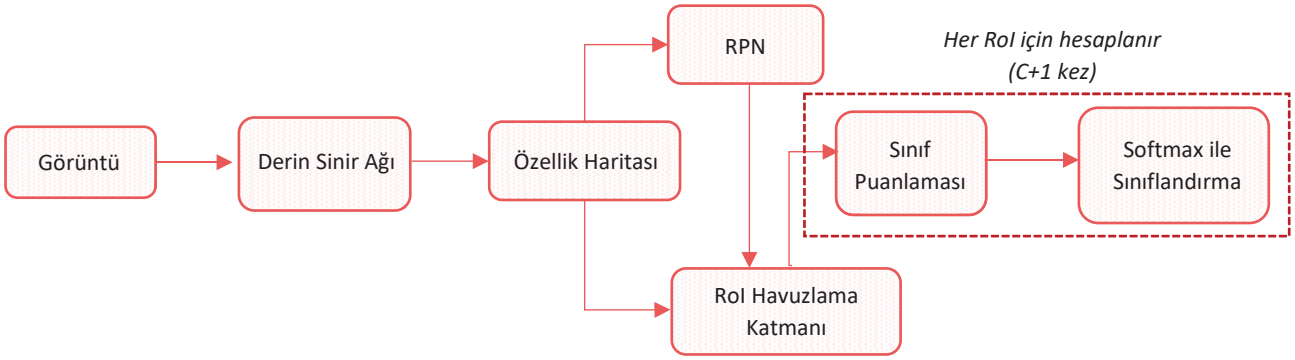
- Elde edilen 3×3'lük özellik haritasındaki hücrelerin ortalaması alınarak nesneyi algılama puanını elde edilir.
- C adet sınıf varsa, nesne olmayan yerler içinde bir sınıf eklenerek toplamda C+1 tane 3×3'lük harita oluşturulur ve C+1 tane sınıf puanı elde edilir.
- Ardından, her bir sınıfın olasılığını hesaplamak için elde edilen puanlar Softmax algoritmasına gönderilir.

RPN işleminden sonra RoI havuzlama katmanına gönderilen bilgiler burada bir süzgeçten geçmektedir. Belirlenen bir aday bölgenin 3×3'lük haritasının elde edilip sınıf puanının hesaplanması Şekil-9'da gösterilmektedir.



Şekil-9: (a) RoI'lerin 3×3'lük harita ile başarılı bir şekilde örtüşmesi [50], (b) Örnek bir skor tablosu ve Sınıf Puanının hesaplanması

Şekil-10'daki akış diyagramına göre Şekil-9'da "RoI havuzlama Katmanından-Sınıf Puanlamasına" giden kısım temsil edilmektedir. Bu işlem her bir aday bölge için yapılmaktadır.



Şekil-10: RFCN mimarisinin akış diyagramı

3.2.1.6 Maske Bölgesi Tabanlı Evrişimli Sinir Ağları

He ve ark. tarafından 2018 yılında geliştirilmiş olan Maske Bölge Tabanlı Evrişimli Sinir Ağı (Mask-RCNN) mimarisi bir evrişimsel sinir ağıdır ve segmentasyon açısından son teknoloji olarak bilinmektedir. Faster-RCNN'e segmentasyon işleminin eklenmesiyle elde edilen Mask-RCNN mimarisi içeresinde Semantik ve Örnek olmak üzere 2 ana görüntü segmentasyon türü bulunmaktadır [51].

Şekil-11'de gösterildiği gibi Mask-RCNN temelde Faster-RCNN'den türetilmiş bir yapıdır. Faster-RCNN'de yer alan iki çıkışa ek olarak üçüncü bir çıkış yer almaktadır. Bilinen çıkışlardan biri aday nesne, diğeri ise sınırlayıcı kutudur. Mask-RCNN'de bu çıktılara ek olarak nesnenin maskesinin bulunduğu üçüncü bir çıkış eklenmektedir. Bu da yapının iki aşamadan meydana gelmesine sebep olur.

İlk aşama:

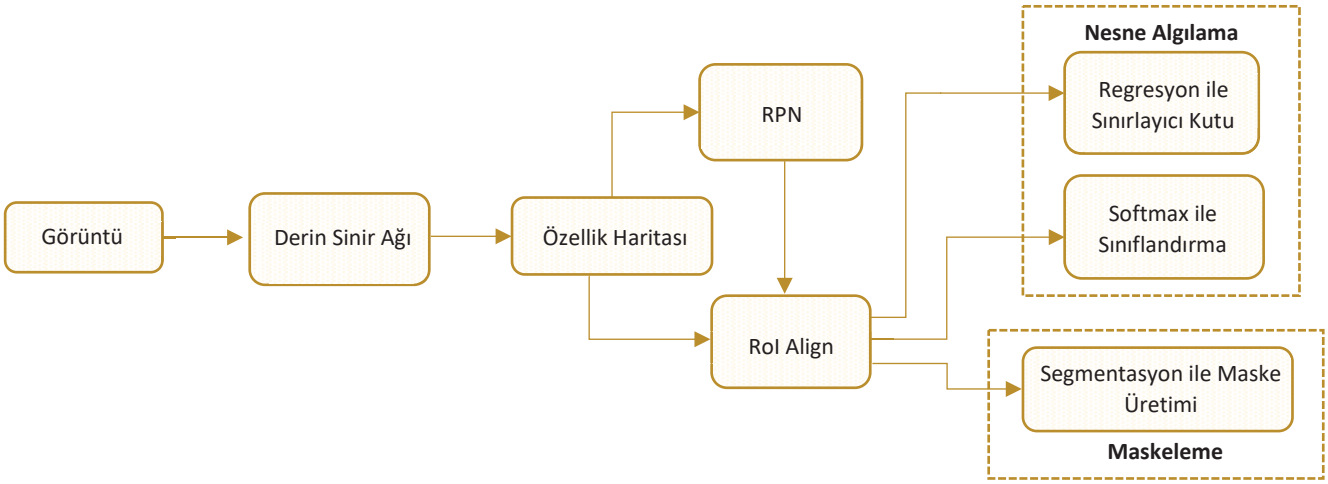
- Görüntünün bir derin öğrenme ağı ile özellik haritası çıkarılır.
- Özellik haritası kullanılarak RPN ile görüntüdeki RoI'ler saptanır.

İkinci aşama:

- Önerilen bölgelerin her biri için sınırlayıcı kutular ve nesne sınıfının tahmini gerçekleşir.
- Paralel olarak her bir RoI için $m \times m$ boyutlu ikili maske çıktısı üretilir.

Önerilen her bölge farklı boyutta olabilmektedir fakat ağlardaki tam bağlantılı katmanları, tahminlerde bulunmak için her zaman sabit boyutlu vektör gerektirir. Bu yüzden önerilen bu bölgelerin boyutu, RoIAlign yöntemi kullanılarak sabitlenir.

RoIAlign: Bu işlem RoI havuzlama işlemine benzemektedir. Aralarındaki temel fark ise RoIAlign'in niceleme gerçekleştirmemesidir. Niceleme işlemi büyük bir değer kümesindeki girdiyi ayrı bir kümeye sınırlama (yuvarlama) işlemidir. Sınırlama işlemi kolay hesaplama yapılmasını sağlarken veri kaybetmemize sebep olmaktadır. Bu durumu aşabilmek için RoIAlign yönteminde filtre boyutuna göre (3×3 ise 9, 4×4 ise 16 kutu vs.) saptanan nesneyi eşit parçalara bölme işlemi gerçekleştirilerek her birinin içine çift enterpolasyon uygulanmaktadır. RoIAlign, ortalama olarak daha iyi hassasiyet sağladığı için nesne saptama işlemleri daha başarılı bir şekilde gerçekleştirilebilmektedir [52].



Şekil-11: Mask-RCNN mimarisinin akış diyagramı

3.2.2 Tek Aşamalı Nesne Saptaması

Tek aşamalı nesne algılama algoritmaları, bölge önerisi vasıtasıyla nesne konumlarını önceden belirlemeye ihtiyaç duymamaktadır. Bunun yerine nesne sınıflandırma ve algılama işlemlerini doğrudan görüntü seviyesinde gerçekleştirmektedir. Bu sayede, nesnenin sınıflandırma doğruluğunu ve koordinat konumunu doğrudan elde edebildiği için iki aşamalı nesne saptama yöntemlerine göre daha hızlı çalışmaktadır [42].

İki aşamalı nesne saptama yöntemlerinde; eğitim birden fazla aşamadan oluşur, bu sebepten verileri işlemek ve sonuç üretme zamanı oldukça uzundur. Buna istinaden, tek aşamalı nesne saptama yöntemlerinin motivasyonu bu problemleri ortadan kaldırarak, iki aşamalı nesne saptama yöntemlerinde yaşanan sorunların önüne geçmektir.

3.2.2.1 Sadece Bir Kez Bak

Redmon ve ark. tarafından 2016 yılında evrişimli sinir ağlarını kullanarak geliştirilen gerçek zamanlı nesne algılama mimarisidir. İki aşamalı nesne saptama mimarilerinde bir nesnenin saptanması bir sınıflandırma problemi, sınırlayıcı kutusu ise bir regresyon problemi iken, Sadece Bir Kez Bak (You Only Look Once- YOLO) mimarisi tamamen bir regresyon problemidir. YOLO mimarisi görüntülerdeki/videolardaki nesneleri ve bu nesnelerin koordinatlarını aynı anda saptanmaktadır. YOLO'nun zaman içerisinde gelişen çeşitli versiyonları mevcuttur. Her bir versiyon bir öncekinde var olan probleme çözüm getirmesi amacıyla geliştirilmiştir [53]. YOLO'nun temelini oluşturan YOLOv1 algoritması şu şekilde çalışmaktadır:

- Görüntü SxS boyutunda eşit büyüklükte ($3 \times 3, 5 \times 5, 19 \times 19$ gibi) parçalara (ızgara/hücre) bölünür.
- Izgara hücrelerinin her biri sınırlayıcı kutu ve bu kutular için güven skoru tahmin eder.
- Hücreler, her nesnenin sınıfını belirlemek için tek bir evrişimsel sinir ağından geçirilir. Omurga olarak GoogleLeNet mimarisi kullanılır.
- YOLO her ızgara için ayrı bir tahmin vektörü $[x, y, w, h,$ güven skoru] oluşturur.
- Hücreler ağdan geçirilirken nesnenin orta noktası hücre içinde bulup bulunmadığı kontrol edilir.
- Eğer nesnenin orta noktası (x, y) hücre içinde ise, saptadığı nesnenin yüksekliğini (h), genişliğini (w), sınıfını ve güven skorunu hesaplar.
- Her bir bölgedeki nesneleri çevreleyen sınırlayıcı kutular hesaplanan koordinatlar [x, y, w, h, güven skoru] sayesinde çizilir.
- Öngörülen sınırlayıcı kutu gerçek kutuyla aynıysa, Birleşim Üzerinden Kesişme (Intersection Over Union - IOU) 1'e eşittir. Gerçek kutuya eşit değilse sınırlayıcı kutular ortadan kaldırılır.

Güven Skoru: Sınırlayıcı kutu ile çevrilen hücre içindeki nesnenin içeriğinden ne kadar emin olduğunu gösteren skor türüdür. Eğer hücrede nesne bulunmuyorsa skor değeri 0'dır.

P(nesne): Hücre içinde nesnenin bulunma olasılığıdır.

IoU: Gerçekte nesnenin bulunduğu kutu ile tahmin edilen kutunun kesişimidir.

Bu elemanların arasındaki ilişki aşağıdaki denklem ile ifade edilmiştir.

$$Güven\ skoru = P(nesne)\ x\ IoU$$

Şekil-12'de gösterilen görselde silah nesnesinin orta noktası 4. hücrede yer almaktadır. Eğer bu hücre içerisinde başka bir nesnenin de orta noktası olsaydı sınırlayıcı kutu sadece güven skoru yüksek olan için çizilecektir [51].

Şekil-12: Görüntü 3x3'lük hücrelere bölünmüştür. Silah nesnesinin orta noktası ve koordinatları hesaplanmıştır. Sınırlayıcı kutu ile çerçevelenmiştir.

Bu durumda var olan diğer nesnenin sınırlayıcı kutusu çizilmeyecektir. Bunun için farklı yöntemlerin geliştirilmesine ihtiyaç duyulmuştur. Buradan yola çıkarak YOLOv1'in dezavantajları aşağıdaki gibi ifade edilebilir:

- Küçük nesneleri ya da birbirine yakın nesneleri saptamaya zorlanır.
- Genelleme yeteneği zayıftır (aynı nesne farklı boyutlarında sınıflandırma yapamamaktadır).
- Birden fazla nesnenin orta noktası aynı hücrede ise birini seçmek zorunda kalır.

YOLOv1'deki problemlere istinaden Redmon ve Farhadi tarafından bir sonraki yıl YOLO9000 diğer adıyla YOLOv2 mimarisi tanıtılmıştır [54]. YOLOv2'deki yenilikler aşağıdaki şekilde ifade edilebilir:

- Toplu normalleştirme yaparak modelin düzenlenmesi sağlanmıştır.
- Giriş boyutu yükseltilerek ortalama hassasiyet $\sim\%4$ oranında yükseltilmiştir.
- Birden fazla orta nokta durumunu aşmak için çıpa kutuları eklenmiştir. Bu sayede her bir hücre için vektör hesaplamak yerine her bir çapa kutusu için vektör hesaplanmıştır.
- 13×13'lük ızgaralar kullanarak görüntüdeki küçük nesnelerin saptanması gerçekleştirilmiştir.
- Çok ölçekli eğitim sayesinde nesnelerin genelleştirilmesi sağlanmıştır. Başka bir deyişle giriş görüntüsünün boyutunu sabitlemek yerine, birkaç yinelemede bir ağı değiştirip, her 10 grupta bir ağ rastgele yeni görüntü boyutları seçerek eğitimini tamamlamıştır.
- Omurga olarak DarkNet19 mimarisi kullanılmıştır.

YOLOv2 mimarisi nesne saptaması alanında oldukça başarılı sonuçlar üretmektedir. Fakat daha da başarılı sonuçlar alabilmek için Redmon ve Farhadi 2018 yılında YOLOv3 mimarisini tanıtarak artırımlı iyileştirmeler olarak isimlendirdikleri bazı yenilikleri açıklamıştır [55]. YOLOv3'teki yenilikler aşağıdaki şekilde ifade edilebilir:

- Her sınırlayıcı kutu için skor üretimi lojistik regresyon yöntemi ile gerçekleştirilmiştir.
- YOLO v2'de kullanılan Softmax yerine her sınıf için lojistik sınıflandırıcılar kullanılmıştır.
- Nesnelerin tahmini sırasında daha fazla bilgi elde etmek için "neck (boyun)" adında ara katmanı eklenerek, farklı ölçeklerde daha iyi sonuçlara sahip olmak için Özellik Piramit Ağı (Feature Pyramid Network- FPN) [56] yöntemi kullanılmıştır.
- Omurga olarak DarkNet53 mimarisi kullanılmıştır.

2020 yılının başlarında Redmon bilgisayarla görü alanındaki çalışmalarını durdurduğunu açıkladıktan sonra YOLO'nun geliştirilmesi diğer araştırmacılar tarafından gerçekleştirilmeye devam etmiştir. Bochkovskiy ve ark. 2020 yılının ortasına gelindiğinde YOLOv4 mimarisini geliştirdiklerini duyurarak v3'e göre daha hızlı, performanslı ve maliyetli olduğunu belirtmişlerdir [57]. YOLOv4'deki yenilikler aşağıdaki şekilde ifade edilebilir:

- Özellik çıkarım süresini arttırmadan BoF (Bag-of-Freebies - Hediye Çanta) tekniği ile veri artırımı yapılmıştır.
- Çıkarım maliyetini yalnızca küçük bir miktar artıran ancak nesne algılamanın doğruluğunu önemli ölçüde artırabilen BoS (Bag-of-Specials - Özel Çanta) tekniği kullanılmıştır.
- YOLOv3'te kullanılan FPN yerine Yol Toplama Ağı (Path Aggregation Network - PAN) [58] ve SPP kullanılmıştır.
- Omurga olarak CSPDarkNet53 mimarisi kullanılmıştır.

YOLOv4'ün tanıtıldığı aynı sene içerisinde Jocher tarafından YOLOv5 mimarisi GitHub üzerinden tanıtılmıştır. YOLOv5, diğer YOLO versiyonlarından farklı olarak bir PyThoch uygulamasıdır. YOLOv4'teki gibi omurga olarak CSPDarkNet53 kullanmaya devam ederken neck katmanında yalnızca PAN kullanmaktadır.

YOLO versiyonlarına bakıldığında YOLOv4 ve YOLOv5'in v3, v2, v1 mimarilerinden daha hızlı ve başarılı sonuçlar ürettiği çeşitli kaynaklarda gösterilmiştir [57, 59, 60]. Fakat YOLOv4 ve YOLOv5 hakkında net bir sonuç bulunmamaktadır. Bazı kaynaklarda YOLOv4'ün bazılarında ise YOLOv5'in daha performanslı sonuçlar üretildiği gösterişe de bu durum kullanılan veri kümeleri, değiştirilmiş hiper-parametreler gibi birçok faktöre bağlı olabilmektedir [60-63]. Aşağıdaki Çizelge-2'de YOLO versiyonlarında kullanılan teknikler ve parametreler yer almaktadır.

Çizelge-2: YOLO versiyonlarının özellikleri

	YOLOv1	YOLOv2	YOLOv3	YOLOv4	YOLOv5
Omurga	GoogleLeNet	DarkNet19	DarkNet53	CSPDarkNet53	CSPDarkNet53
Boyun K.	-	-	FPN	SPP & PAN	PAN
Aktivasyon F.	Linear	Logistic	Linear	Mish	Leaky ReLU & Sigmoid
Kayıp F.	Sum-Squared Error	Sum-Squared Error	Binary Cross-Entropy	Binary Cross-Entropy	Binary Cross-Entropy & Logits loss function

3.2.2.2 Tek atışta çoklu kutu algılama

Liu ve ark. tarafından 2016 yılında arka plan bilgisini kullanılarak Tek Atışta Çoklu Kutu Algılama (Single shot multi-box detector – SSD) nesne algılama mimarisi geliştirilmiştir. SSD mimarisi temelde VGG16 mimarisi üzerine inşa edilmiştir. Ancak tamamen bağlı katmanları kullanmak yerine bir dizi evrişim katmanı ekleyerek; birden çok ölçekteki özelliklerin çıkarılması ve her katmanda girdi boyutunun kademeli olarak azaltılması sağlanmıştır. Nesnenin yerelleştirilmesi ve sınıflandırılması ağın tek bir ileri geçişinde gerçekleştirilmektedir [64]. Sınırlayıcı kutu için Szgedy'nin MultiBox üzerindeki çalışmasından ilham alarak inception (başlangıç) tarzı bir evrişim ağı kullanılmıştır. 1×1 boyutlu filtreler sayesinde boyut sayısının azalması sağlanmıştır [65].

MultiBox: Nesne sınırlama kutularının koordinatlarını doğrudan çıkaran inception tabanlı evrişimli ağ mimarisidir. İlgili nesnenin doğru bir sınırlayıcı kutu içinde olma olasılıkları hesaplamak için *Güven Kaybı* ve *Konum Kaybı* adı verilen iki bileşeni kullanır. Güven kaybı, sınırlayıcı kutunun nesnelliğinden ne kadar emin olduğunu çarpraz entropi ile hesaplarken, konum kaybı tahmin edilen sınırlayıcı kutuların gerçek değerlerinden ne kadar uzakta olduğu L2-Norm ile hesaplamaktadır. α değeri, konum kaybının sonuca katkısının belirlenmesinde yardımcı olmaktadır [65].

$$MBKayıpDeger = GüvenKaybı + \alpha \; x \; KonumKaybı$$

MultiBox mimarisinde IoU değerlerinin hesaplanması için Faster-RCNN mimarisinde kullanılan çapalara benzer *priors adında* yapılar eklenmiştir. MultiBox, nesne sınıflandırması yapmazken SSD yapar. Bu nedenle, tahmin edilen her sınırlayıcı kutu için, veri kümesindeki her olası sınıf adına bir dizi *c* sınıfı tahminleri hesaplanır.

4. Tehlikeli Nesnelerin Saptanması

Bir sahnedeki tüm nesneleri ya da belirlenen özel bir nesneyi bulmak için kullanılan nesne saptama yöntemi birçok alanda kullanılmaktadır. Çalışmada kaynak taraması daraltılarak, sadece video görüntüleri üzerinden tehlikeli nesneleri (silah, kesici, delici aletler vs.) saptayan çalışmalar ve bu aşamada kullanılan yöntemler iki kısma ayrıştırılarak incelenmiştir.

4.1 Geleneksel Yöntemler ile Tehlikeli Nesnelerin Saptanması

Tehlikeli nesnelerin, yetkisiz kişilerce kullanımı her zaman bir sorun olmuştur. Bu sorunun önüne geçmek adına 2000'li yılların başında gizli silah ve bıçak saptama sistemleri geliştirilmeye başlanmıştır. Bu sistemler kızılötesi görüntüleme, milimetre dalga görüntüleme gibi bazı görüntüleme tekniklerini kullanarak kişinin kıyafetlerinin altında taşıdığı gizli nesneyi tespit edebilmektedir. Ancak nesnenin tehlikeli olup olmadığını ve türünü saptayamamaktadır [66-70]. Zaman içerisinde yapay sinir ağları ile bu nesnelerin türü de saptanmaya çalışılmıştır [71]. Ancak burada kullanılan görüntüleme teknikleri kalabalık alanlarda kullanıma uygun olmadığı düşünülmektedir. Bunun yerine günümüzde hemen hemen her yerde bulunan MOBESE, CCTV, IP gibi kamera sistemlerinden elde edilen veriler kullanılarak neredeyse gerçek zamanlı tespitler gerçekleştirmek hem maliyet açısından hem de uygulanabilirlik açısından daha kolay olduğu düşünülmektedir.

Buna istinaden 2007 yılında hem silahları hem de silahla suç işleme niyetinde olan kişileri CCTV aracılığıyla otomatik olarak saptanması amaçlanmış ve bu doğrultuda MEDUSA adında bir program oluşturulmuştur [7]. Ertesi yıl operatörler ile silah saptamanın sınırlı olduğu belirtilerek ateşli silahların saptamasını otomatikleştirmek için model geliştirmişlerdir. Fakat bu model aynı sahne içerisinde yer alan farklı silahları algılamada başarılı olamamıştır [72]. İlerleyen yıllarda yüksek kapasiteli donanımlara erişim, veri saklama miktarındaki artış ve derin öğrenme tabanlı mimarilerin geliştirilmesiyle birlikte nesne saptama sistemleri daha da önem kazanmıştır. Bu süreç içerisinde nesne algılama yöntemleri kullanılarak yaya saptaması, yüz tanıma gibi çeşitli problemlere çözüm aranmıştır. Ancak nesne algılama yöntemleri kullanılarak tehlikeli nesnelerin saptamaya başlanması, 2013 yılında Grega ve ark. tarafından gerçekleştirilmiştir. Grega ve ark. arka plan alglama, kenar algılama, sürgülü pencere, temel bileşen analizi ve sinir ağları yönteminden yararlanarak kendi oluşturdukları veri setleri ile silah saptama işlemi gerçekleştirmişlerdir [73]. Ardından 2015 yılında Tiwari ve Verma görüntüden ilgisiz nesneleri ortadan kaldırmak için

renk tabanlı bölütlemeden yararlanmıştır. Hedef nesneyi bulmak için Harris ilgi noktası dedektörü ve Hızlı Retina Anahtar Noktası (Fast Retina Keypoint - FREAK) yöntemlerini kullanmıştır. %84,26 sınıflandırma doğruluk oranı ile çalıştığını belirtmelerine rağmen arka plan değişimlerine duyarlı olduğu için gerçek zamanlı çalışabilecek bir sistem olmadığı düşünülmektedir [74]. Aynı yıl yazarlar benzer şekilde ilgisiz nesneleri ortadan kaldırma adına yine renk tabanlı bölütlemeden yararlanmış ve bu sefer Hızlandırılmış Sağlam Özelliklerin (SURF) ilgi noktası dedektörünü parçalı görüntülerde hedef nesneyi bulmak için kullanmışlardır. Veri setini kendilerinin topladığını ve 15 adet silah, 10 adet silah olmayan görüntülerden oluştuğunu belirtmişlerdir. Sistemleri %88,67 oranında başarı elde etmesine rağmen veri setinin azlığı ve silahların sabit bir şablonla eşleştirilmesi sistemin gerçekçi çalışmadığını düşündürmektedir [75]. Grega ve ark. 2016 yılında kendi verilerini toplayıp, nesne saptaması için sürgülü pencere ve Canny kenar algılama tekniklerinden yararlanarak bıçak ve silah saptama sistemi geliştirmişlerdir. Bıçak algılama algoritmasının özgüllüğü ve duyarlılığı sırasıyla %94,93 ve %81,18 iken ateşli silah algılama algoritmasının özgüllüğü ve duyarlılığı sırasıyla %96,69 ve %35,98'dir. Düşük çözünürlükteki verilerde çalışabildiği için gerçek zamanlı verilerde çalışabileceğini belirtmişlerdir. Fakat sabit boyutlu sürgülü pencere kullanmaları hatalı tespitlere ve zaman kaybına yol açabilmektedir [76]. Geleneksel yöntemlerden yararlanan bir diğer çalışmada ise Vajhala ve ark. öznitelik vektörünü çıkartırken HOG tekniğini, sınıflandırma için ise yapay sinir ağından yararlanmıştır. Nesne saptaması sırasında %83 oranında doğruluk değeri elde ettiklerini belirtmişlerdir. Fakat bu çalışmada kullanılan silah ve bıçaklardan oluşan veri seti dengesizdir, görüntü kalitesi yüksektir ayrıca silah resimlerinin arka planı beyazdır dolaysıyla gerçek zamanlı veriler üzerinde iyi bir sonuç üretemeyeceği düşünülmektedir [77]. Kaynaklarda karşılaşılan bir diğer çalışmada ise; Buckchash ve Raman farklı ölçek ve konumdaki bıçakların çevrimiçi videolarda saptanması için bir çerçeve sunmaktadır. Çalışmalarında ön plan bölütleme ile özellik çıkarımı, Hızlandırılmış Bölütleme Test (Features from Accelerated Segment Test - FAST) tekniği ile nesne saptaması ve Çoklu Çözünürlük Analizi (Multi-Resolution Analysis - MRA) tekniği ile nesne sınıflandırma işlemini yaparak %98,48 doğruluk değerine ulaşmışlardır [78]. Farklı konum ve pozisyonlarda bulunan bıçaklardan oluşan veri kümesi kullanmaları başarı oranını arttırdığı görülmektedir. Fakat bununla beraber eğitim ve test sırasında kullanılan veri sayısının azlığı üretilen performans sonucun ve süresinin gerçekçi olmadığını düşündürmektedir.

4.2 Modern Yöntemler ile Tehlikeli Nesnelerin Saptanması

2014 yılında tanıtılan RCNN ve OverFeat mimarileri nesne saptama işlemlerinde çığır açarak saptama işlemlerinin başarısını ve hızını arttırmıştır. Bu dönemden sonra yapılan çalışmaların birçoğu derin öğrenme tabanlı mimariler çerçevesinde gerçekleştirilmiştir. Sunulan bu mimariler ile birçok (yaya saptaması, otonom sürüş, yüz tanıma vs.) nesne tanıma sistemi geliştirilse de tehlikeli nesneler özelinde bilinen ilk çalışma Lai ve Mapes tarafından 2017 yılında hazırlanmıştır. Yapının omurgası OverFeat mimarisinden oluşturulmuş ve gerçek zamanlı sınıflandırma yapabilmek adına film sahnelerinden alınan veriler kullanılmıştır. En iyi performansı Overfeat-3'te %93 eğitim doğruluğu ve %89 test doğruluğu ile elde etmişlerdir. Ancak veri setlerinin çözünürlüğü yüksek olduğu için gerçek veriler üzerinde çalışmasının zor olduğu düşünülmektedir. Bunun yanı sıra çalışmada da belirtildiği gibi her sınıflandırmanın yaklaşık 1,3 saniye sürmesi gerçek zamanlı gözetlemenin yapılamayacağını göstermektedir [79]. Aynı yıl Verma ve Dhillon tarafından karmaşık sahnelerde otomatik silah algılama için Faster-RCNN tabanlı model geliştirilmiştir. Ayrıca saptanan nesnelerin sınıflandırılması için SVM, K En Yakın Komşu (K Nearest Neighbors - KNN) ve Topluluk Ağacı (Ensemble Tree - ES) algoritmalarından yararlanmışlardır. Veri setlerini film, televizyon şovları, video oyunları ve animelerden oluşturmuşlardır. En iyi sonucu %93 sınıflandırma doğruluğu ile topluluk ağaçları algoritmasıyla elde etmişlerdir. Kullanılan eğitim verilerinin çözünürlükleri yüksek olduğu için gerçek zamanlı sistemlerdeki başarısının benzer şekilde yüksek olacağı düşünülmemektedir [80].

2018 yılında tehlikeli nesnelerin derin öğrenme tabanlı mimariler ile saptanması birçok araştırmacı tarafından gerçekleştirilmiş ve bu alanda önemli katkı sağlamışlardır. Olmos ve ark. 2018 yılında Faster-RCNN tabanlı silah saptama modeli geliştirmişlerdir. Modelin CNN omurgasında önceden eğitilmiş VGG16 mimarisi kullanarak kendi oluşturdukları 3000 adetlik veri seti ile ince ayar yaparak modelin eğitimini tamamlamışlardır. Çalışmanın sonucunda %100 hatırlama ve %84,21'lik bir hassasiyet değeri elde ederek düşük çözünürlükteki videolarda da başarılı sonuçlar üretebildiklerini göstermişlerdir. Genel olarak iyi sonuç üreten bu çalışma fatura, cüzdan, kart gibi benzer büyüklükteki nesnelerden silah nesnesini ayırt etmekte zorlanmaktadır [81]. Akcay ve ark. X-RAY verilerini kullanarak sürgülü pencere tabanlı CNN, Faster-RCNN, RFCN, YOLOv2 mimarilerinin performanslarını karşılaştırmışlardır. YOLOv2 mimarisinin hem doğruluk hem de hız açısından diğer mimarilerden daha iyi olduğunu belirterek 544×544 boyutlu girişlerde %88,5'lik mAP, 416×416 boyutlu girişlerde %97,4'lük mAP değeri elde ettiklerini göstermişlerdir [82]. Singleton ve ark.. çeşitli yönlerde, şekillerde ve boyutlarda tabancaları tanımlamak için yaklaşık 5 bin veri kullanarak MobileNetv1 ağını eğitmişlerdir. Modeli test ederken 30 silahlı fotoğraftan 26'sını başarı ile saptama ettiklerini belirterek %86,67'lik sınıflandırma doğruluğu elde ettiklerini göstermişlerdir. Fakat kullandıkları test seti çok küçük olduğu için bu oranın gerçekçi olmadığı düşünülmektedir [83].

2019 yılında Gelena ve Yadav, CCTV operatörlerini hem görsel hem de sesli bir şekilde uyarılması için CNN tabanlı model önerisinde bulunmuştur. Görüntü verilerini siyah-beyaza çevirerek, filtreleme, kenar algılama, arka plan çıkarımı ve sürgülü pencere işlemlerini gerçekleştirmişlerdir. Elde edilen özellikleri CNN mimarisine göndererek bir nesnenin silah ya da silah olmamasını tahmin etmişlerdir. Önerilen yaklaşımla elde edilen sınıflandırma doğruluğu %97,78, hassasiyet değeri %93,84 ve özgüllük değeri %99,73 oranında olduğunu göstermişlerdir. Başarı oranları yüksek olmasına rağmen bu

çalışma yalnızca tek bir silah türünü saptayabilmektedir. Ayrıca kenar çıkarma yöntemi ile gerçek veriler üzerinde anlık çıkarım yapılamayacağı düşünülmektedir [84]. Aynı yıl Romero ve Salamea YOLO mimarisini kullanarak iki bölümden oluşan ateşli silah saptama modeli tasarlamışlardır. İlk bölümde YOLO ile nesne algılama ve yer belirleme işlemini, ikinci bölümde ise CNN ile ateşli silah algılama modelini gerçekleştirmişlerdir. CNN modeli olarak VGGNet ve ZFNet algoritmaları kullanmış ve performanslarını karşılaştırmışlardır. Buna göre gri tonlamalı görüntülerin renkli görüntülerden, VGGNet tabanlı modelin, ZFNet tabanlı modelden daha başarılı olduğunu göstermişlerdir [85]. Fernandez-Carrobles ve ark. silah ve bıçak saptaması için Faster-RCNN tabanlı model önermişlerdir. Modelin omurgasında sırasıyla bir GoogleNet ve bir SqueezeNet mimarisi alınarak iki yaklaşımı karşılaştırmışlardır. Silah saptaması için en iyi sonuç %85,44 ortalama hassasiyet değeri ile SqueezeNet mimarisi kullanılarak elde edilirken, bıçak saptaması için en iyi sonuç %46,68 ortalama hassasiyet değeri ile GoogleNet yaklaşımı kullanarak elde etmişlerdir. Bu sonuçların önceki çalışmalardan daha başarılı sonuç ürettiğini belirtmişlerdir [86]. Iqbal ve ark. silahların ve tüfeklerin algılanması için iki aşamalı Yönelim Duyarlı Nesne Dedektörü (OAOD) modelini önermişlerdir. Aşama-1'de, nesnenin yönünü tahmin ederken, aşama-2'de yeniden sınıflandırılan ve yerelleştirilen döndürülmüş nesne önerilerinden maksimum alan dikdörtgenlerini kırpmaktadırlar. Önerilen OAOD modelini, aynı veri seti üzerinde YOLOv2, YOLOv3, SSD, Evrişimsiz Tek Atım Dedektörü (DSSD) ve Faster-RCNN dahil olmak üzere beş mevcut model ile değişen IoU ve mAP metriklerine göre karşılaştırmışlardır. Sonuç olarak önerilen OAOD modelinin diğerlerinden daha başarılı sonuçlar ürettiğini göstermişlerdir. Fakat bu sistemin tüfek vb. uzun ve ince nesneler için daha uygun olduğu düşünülmektedir. Ayrıca eğitim ve test amacıyla yüksek kalitede görüntüler kullanılmıştır, bu da onu gerçek zamanlı senaryolar için daha az uygun hale getirebilir [87]. Azevedo Kanehisa ve Almeida Neto tarafından YOLO mimarisinin ateşli silah tespitindeki başarısı gözlemlenmiştir. Veri setini %90 eğitim %10 test şeklinde ayırarak eğittikleri modelin sonucunda %70,72 mAP, %95,73 duyarlılık, %97,30 özgüllük ve %96,26 sınıflandırma doğruluğu elde etmişlerdir.
Bu çalışmada doğrulama işlemi yapılmamıştır. Ayrıca veri setindeki görüntülerin kalitesi yüksek olduğu için gerçek zamanlı veriler üzerindeki başarısının düşük olabileceği düşünülmektedir [88].

2020 yılında González ve ark. eğitimin öğrenme başarısını arttırmak için gerçek ve sentetik veri seti kullanarak Faster-RCNN-FPN tabanlı silah saptama modeli önermişlerdir. Omurgası ResNet50 olan Faster-RCNN mimarisini FPN mimarisi ile birleştirip daha hızlı çalışmasını sağlamışlardır. Sentetik veriler kullanılarak yapılan eğitimin, modele çeşitlilik sağlayarak doğruluğu artırdığını ve gerçek verilerin CCTV görüntülerinin algılanmasını kolaylaştırdığını belirtmişlerdir. Fakat gerçek zamanlı görüntülerde uzaktaki nesneleri algılamayı çözemediklerini de ifade etmişlerdir [89]. Perez-Hernandez ve ark. küçük nesneler (tabanca, akıllı telefon, bıçak, fatura, cüzdan ve kart) arasındaki algılamayı geliştirmek amacıyla iki aşamalı ODeBiC metodolojisini önermişlerdir. Birinci aşamada, hedef nesneleri içeren aday bölgeleri elde etmek için omurgası ResNet101 olan Faster-RCNN mimarisini eklemişlerdir. İkinci aşamada ise, her bir bölgeyi binarizasyon tekniği ile sınıflandırır ve bunu takiben çıktı çerçevesini algılama sonuçlarıyla birlikte üretmek için bir toplama yöntemi izlemişlerdir [90]. Raturi ve ark. gizli silahların saptanması ve sınıflandırılması için derin öğrenme tabanlı yeni bir çerçeve önermişlerdir. Veri setindeki gürültüyü gidermek için Canny kenar algoritması, istenilen nesnelere sahip olup olmadığını belirlemek için sürgülü pencere tekniği kullanmışlardır. Ardından Faster-RCNN algoritması ile eğitim gerçekleştirmişlerdir [91].

2021 yılında Noor ve Isa, YOLOv4 Darknet mimarisi ile tabanca ve bıçak gibi tehlikeli nesneleri saptamaya çalışmıştır. Tek sınıflı ve çok sınıflı iki eğitim seti oluşturarak sistemin etkinliğini test etmişlerdir. Tek sınıflı nesne algılamada %66-77 arasında sınıflandırma doğruluğu elde ederken, çoklu sınıflı nesne algılamada %100'e kadar sınıflandırma doğruluğu elde etmeyi başarmışlardır [92]. Narejo ve ark. daha az hesaplama kaynağı ile ateşli silahları kısa sürede saptanabilen bir çerçeve önermiştir. IP kameralar aracılığı ile güvenlik güçlerine ulaşarak girişleri kilitleme sistemini programlamışlardır. Nesne saptaması için YOLOv3 Darknet53 mimarisini kullanmışlar ve YOLOv2 ile karşılaştırarak YOLOv3'ün daha başarılı sonuç ürettiğini göstermişlerdir [93]. Hashmi ve ark. CCTV verilerini kullanarak görüntüdeki silahın varlığını saptamak için YOLOv3 ve YOLOv4 mimarilerinin karşılaştırmalı analizini yapmıştır. YOLOv4 mimarisinde hatırlama metriği %78, F1 puanı %82, hassasiyet %85 ve mAP %84,85 değerinde iken YOLOv3 mimarisinde %71, F1 puanı %77, hassasiyet %84 ve mAP %77,30 değerinde olduğunu hesaplamışlardır. Buna istinaden YOLOv4 mimarisinin işlem süresi ve hassasiyet açısından YOLOv3'ten bariz bir şekilde üstün performans gösterdiğini belirtmişlerdir [94]. Kayalvizhi ve ark. X-ışını görüntüleme sistemlerinde bıçak, makas, İngiliz anahtarı, pense gibi nesnelerin saptanması için derin öğrenme tabanlı bir çözüm önermiştir. YOLOv3 mimarisinin omurgasında DarkNet53 yerine Inceptionv3 ve ResNet50 mimarilerini kullanarak karşılaştırma yapmışlardır. ResNet50 omurgası kullanılarak elde edilen sonuçlar Inceptionv3'den elde edilen sonuçlara göre daha başarılı olmasına rağmen mAP metriğinin herhangi bir nesne için dahi %70'i aşamadığı görülmüştür. Bu durumda bu modelin gerçek zamanlı sistemlerde başarılı bir şekilde çalışabileceği düşünülmemektedir [95]. Bhatti ve ark. gerçek zamanlı CCTV videolarında silah saptamasını algılamayı başarabilen ve düşük çözünürlük ve parlaklıkta bile iyi çalışan modeli geliştirmek için kendi oluşturdukları veri setinden iki yöntemi ve çeşitli sınıflandırıcıları karşılaştırmışlardır. Sürgülü pencere yaklaşımında VGG16, InceptionV3 Inception ResnetV2 sınıflandırıcılarını ve bölge önerisi yaklaşımında SSDMobileNetV1, Faster-RCNN Inception-ResnetV2 (FRIRv2), YOLOv3 ve YOLOv4 mimarilerini karşılaştırmışlardır. Sonuçta, YOLOv4 mimarisinin, diğer mimarilerden daha başarılı olduğunu %91,73 mAP değeri ve %91'lik bir F1 puanı ile göstermişlerdir [96]. Salido ve ark. silah saptaması için üç farklı mimari kullanarak, bu mimarilerin karşılaştırmalı analizini gerçekleştirmişlerdir. Çalışmada, omurgası VGG16 olan Faster-RCNN, omurgası ResNet50 olan RestinaNet,

omurgası DarkNet53 olan YOLOv3 olmak üzere 3 mimari kullanmışlardır. ResNet50 omurgasıyla ince ayar yapılmış RetinaNet tarafından elde edilen en iyi ortalama hassasiyet değeri %96,36 ve hatırlama değeri %97,23 olarak hesaplamışlardır. Ayrıca en iyi kesinlik değerini %96,23 ve F1 puan değerini %93,3 ile YOLOv3 mimarisini kullanarak elde etmişlerdir [97]. Iqbal ve ark. geliştirdiği sistem ile drone gözetimi sonucunda herhangi bir olağandışı faaliyetin saptanmasından sonra, sistemin güvenlik görevlileri için bir uyarı oluşturmasını sağlamışlardır. Gözetim görüntülerinin ilgili nesneleri belirleyebilmesi için Faster-RCNN mimarisi için 4 farklı omurga (SqueezeNet, GoogleNet, ResNet-18 ve ResNet-50) kullanarak karşılaştırmışlardır. Sonuçta ResNet50 omurgasına sahip Faster-RCNN mimarisinin en iyi sonucu ürettiğini gözlemlemişlerdir [98]. Sivakumar suç oluşumu hakkında güvenlik güçlerini uyaran bir sistem önermiştir. Sistemde YOLOv4 mimarisini kullanıp Faster-RCNN mimarisi ile performans karşılaştırması yapmıştır. YOLOv4 mimarisinin test sonuçları, mAP %78,3, hassasiyet %98,56 ve hatırlama %91,21 iken Faster-RCNN mimarisinin mAP %62,4, hassasiyet %92,56 ve hatırlama %89,33 değerinde olduğunu göstererek YOLOv4 mimarisinin daha başarılı olduğunu belirtmişlerdir [99]. Kaya ve ark. derin öğrenme yöntemi kullanarak 7 farklı silah türünün saptaması için yeni bir model önerilmiştir. Bu modelin VGGNet mimarisine dayalı silah sınıflandırmasına yeni bir yaklaşım sunduğunu belirterek, VGG16, ResNet50 ve ResNet101 mimarileri ile karşılaştırmışlardır. Karşılaştırma sonucunda önerilen modelin %98,40 sınıflandırma doğruluğuyla ve diğer metriklere göre de en iyi sonuca sahip olduğunu göstermişlerdir [100].

2022 yılında Bushra ve ark. tarafından web kameraları üzerinden silah saptaması yaparak suçluların yüz tanıma ile birlikte tam konumunu ve meydana geldiği olay zamanını YOLOv5 mimarisi kullanılarak bulmayı hedeflenmişlerdir. Modelin düşük çözünürlüklü ve küçük nesneleri doğru bir şekilde yakalayabildiğini belirtmişlerdir. YOLOv5 modelinin mAP değeri %95, hassasiyet değerini %98 ve hatırlama değerini %87 olarak hesaplamışlar ve YOLOv4'e göre daha başarılı sonuçlar elde ettiklerini belirtmişlerdir [101].

Literatür taraması sonucunda elde edilen önemli çalışmalar Çizelge-3'de görüldüğü gibi kısaca derlenmiştir.

Çizelge-3 Çalışmaların karşılaştırmalı analizi

Çalışma	Model	Veri Seti Kaynağı	Veri Seti Boyutu	Performans Değerleri				
				mAP	*Doğruluk*	*Hatırlama*	*Hassasiyet*	*Özgüllük*
[73]	Sürgülü p. +Canny+Sinir ağı	Kendi çekimleri ve üretimleri	4500	-	-	-	-	%96,69
[74]	Harris+FREAK	Çeşitli kaynaklardan	89	-	%84,26	-	-	-
[75]	SURF	Çeşitli kaynaklardan	25	-	%88,67	-	-	-
[76]	Sürgülü p. +Canny	CCTV	12.899	-	-	-	-	%96,69
[77]	HOG+YSA	Google, INRIA	10.508	-	%83,05	-	-	%85,7
[78]	FAST+MRA	KDSDatasets	1448	-	%98,48	%98	%97	-
[79]	OverFeat	YouTube, IMFDB	2735	-	%89	-	-	-
[80]	Faster-RCNN	IMFDB	-	-	%93,1	-	-	-
[81]	Faster-RCNN+ VGG16	Çeşitli kaynaklardan	3000	-	-	%100	%84,21	-
[82]	Sürgülü p.+ YOLOv2	Dbp2, Dbp 6, FFOB, FPOB	19.398 - 9680 13.770	%97,4	-	-	-	-
[83]	MobileNetv1	ImageNet	4794	-	%86,67	-	-	-

[84]	Sürgülü p.+ CNN	[73]	5869	-	%97,78	-	%93,84	%99,73
[85]	YOLO+VGGNet	Google, Instagram ve YouTube	17.684	-	%86,11	%86	%86	-
[86]	SqueezeNet	COCO ve [79]	15.000	-	-	-	%85,44*	-
[87]	OAOD	ITUF	13.647	%89,9	-	-	~%90*	-
[88]	YOLO	IMFDB	20.000	%70,72	%96,26	-	-	%97,3
[89]	Faster-RCNN+ ResNet50+FPN	Kendi çekimleri ve üretimleri	-			%100	%88,12	
[90]	Faster-RCNN+ ResNet101	İnternet ve kendi çekimleri	-	-	-	%93,09	%93,87	-
[91]	Canny+ Faster-RCNN	Çeşitli kaynaklardan	9084	-	%93,6	-	-	-
[92]	YOLOv4+ DarkNet	Open Images V6	3000	-	%66-77	-	-	-
[93]	YOLOv3+ DarkNet53	Google	~500	-	%98,89	-	-	-
[94]	YOLOv4	Google, CCTV ve filmler	7800	%84,85	-	%78	%85	-
[95]	YOLOv3+ ResNet50	SIXray10	1.059,231	%63,7	-	-	-	-
[96]	YOLOv4	CCTV, GitHub, YouTube ve IMFDB	1732 5254 8237	%91,73	%99	-	%93	-
[97]	RetinaNet+ ResNet50	Google ve YouTube	1220	-	-	%97,23	%96,36*	-
[98]	Faster-RCNN+ ResNet50	Çeşitli kaynaklardan	6655	-	-	-	%79*	-
[99]	YOLOv4	Çeşitli kaynaklardan	~200	%78,3	-	%91,21	%98,56	-
[100]	VGGNet tabanlı model	Çeşitli kaynaklardan	5214	%87,3	%98,4	-	-	%99,28
[101]	YOLOv5	Roboflow	1294	%95	-	%87	%98	-

*ifadesi değerin ortalama olduğunu belirtmek için kullanılmıştır.
IMFDB: Internet Movie Firearms Database- İnternet Film Ateşli Silahlar Veri tabanı

5. Bulgular ve Tartışma

Geçtiğimiz son 10 yılda özellikle derin öğrenme tabanlı mimarilerin gelişmesiyle birlikte yapılan nesne saptama işlemlerinin geleneksel yöntemlere kıyasla daha hızlı ve başarılı sonuçlar ürettiği araştırmacılar tarafından defalarca kanıtlanmıştır. Fakat bu alanda eksik kalan ve üzerinde çalışılması gerekildiği düşünülen bazı problemler hala

bulunmaktadır. Nesne saptaması sırasında yaşanan bazı problemler aşağıdaki gibi ifade edilebilir:

- Küçük nesnelerin algılanması,
- Benzer nesnelerin karıştırılması,
- İlgili nesnenin farklı ölçeklerindeki saptamasının zorlaşması,
- Gerçek zamanlı nesne saptaması,
- Düşük çözünürlükteki görüntüden nesnenin algılanabilmesi,
- Doğrusal olmayan hareketler sırasında hedef nesnenin belirlenebilmesi,
- Arka plandaki ani değişimler sonucu hedef nesnenin belirlenebilmesi şeklinde ifade edilebilir.

Bu problemlere dayanarak gelecekteki araştırmalar için bazı öneriler şunlardır:

(1) Maliyeti düşük ve hızlı modeller oluşturmak için; yüzlerce katmana kadar çıkabilen CNN mimarilerinin yerine daha kompakt ve hafif ağlar kullanılabilir. Başka bir deyişle tek veya iki aşamalı nesne saptama modellerinin kullanılması tercih edilebilir. Fakat bu yöntemlerde dahi ağın hızı seçilen yönteme göre değişiklik gösterebilmektedir. Özellikle iki aşamalı nesne saptama modellerinin yüksek doğrulukta sonuç üretmesine rağmen, özelliklerin çıkarımı sırasında ağın yavaş çalıştığı bilinmektedir. Bu durumda ağın hızlanması için özellik çıkarımı sırasında kullanılacak olan teknikler oldukça önem arz etmektedir. Yapılan araştırmalar neticesinde ağın hızlanması için tek aşamalı nesne saptama mimarilerinin kullanılması veya iki aşamalı mimarilerde RPN kullanılması tavsiye edilmektedir. Ayrıca modelden bağımsız olarak Grafik İşlem Biriminden (Graphics Processing Unit - GPU) faydalanılması ağın çalışma hızını arttırmaktadır.

(2) Daha başarılı sonuçlar için; RCNN, Mask-RCNN, Faster-RCNN gibi bölge tabanlı mimariler kullanılabilir. Bunun yanı sıra son yıllarda sıkça kullanılan YOLO mimarisinin son versiyonları hem hızlı ve basit hem de bölge tabanlı mimariler kadar başarılı sonuç üretmektedir. Bununla beraber seçilen mimarilerde kullanılan omurga yapıları, hiperparametreler, eğitim sırasında kullanılan verinin kalitesi, dengesi ve sayısı modelin performansını etkilemektedir. Özellikle modelde kullanılacak olan veriler dengesiz ya da yetersiz olduğu durumlarda sentetik veriler üretilerek veri seti zenginleştirilebilir. Bu sayede modelin ezberlemesinin önüne geçerek farklı veriler üzerinde sistemin doğru çalışması sağlanabilir.

(3) Gerçek zamanlı sistemler oluşturmak için; kullanılan eğitim setlerinin önemli olduğu bilinmektedir. Maliyeti düşük, hızı ve başarısı yüksek model ile yüksek kalitedeki donanımın yanı sıra modelin eğitimi için kullanılan veri setinin geniş ve çeşitli olması oldukça önemlidir. Daha fazla kategoriye sahip yeni ve büyük ölçekli veri setlerinin geliştirilmesi gereklidir. Ayrıca kullanılacak olan verilerin gerçek dünyadaki problemleri yansıtan veriler olması (düşük çözünürlük, karışık arka plan, ışık yetersizliği vb.), ağın eğitimini geliştirmektedir. Bu sayede sistemin gerçek zamanlı veriler üzerinde yüksek performans göstermesi sağlanabilir.

(4) Hızlı bir eğitim süreci için; Transfer Öğrenme yöntemi kullanılarak önceden eğitilmiş ağlardan faydalanılabilir. ImageNet ve COCO gibi veri setleri ile önceden eğitilmiş ağlar kullanılıp yapılması gereken ince ayarlar, farklı veri setleri sayesinde gerçekleştirerek modelin eğitim süreci hızlandırılabilir.

6. Sonuç

Nesne saptama işlemleri akademi dünyasında son 20 yıldır tartışılan önemli konulardan biridir. Bilgisayarların donanım yeteneklerinin artması ile bu alanda yapılan çalışmalar ve gelişmelerin hızla arttığı görülmüştür. Gelişen bu alanda yapılan birçok çalışma bulunmaktadır. Ancak mevcut çalışma için alan daraltılarak belirli özellikteki çalışmalar incelenmiştir. İncelenen çalışmaların özellikleri ve değinilen konular şu şekildedir:

- Sadece tehlikeli nesnelerin nesne saptama yöntemleri ile algılanmasını içeren çalışmalar incelenmiştir.
- Nesne saptaması sırasında kullanılan algoritmalar geleneksel ve modern olarak iki kısma ayrılıp detaylı bir şekilde analiz edilmiştir.
- Yapılan çalışmalarda kullanılan yöntemler ve sonuçları karşılaştırılmıştır.
- Eğer varsa çalışmada eksik olduğu düşünülen durumlardan bahsedilmiştir.
- Nesne saptama alanındaki mevcut problemler ve gelecekteki araştırmalar için bazı yönergeler sunulmuştur.

Çalışmada tehlikeli nesneler üzerine odaklanılmasının temel sebebi bu alanda yapılmış çalışmaların daha az olmasından kaynaklanmaktadır. Genellikle kaynaklarda, yaya saptaması, otonom sürüş ve yüz saptaması üzerine yapılan çalışmalar görülmektedir. Fakat farklı nesneler faklı problemleri beraberinde getirir dolayısıyla bu alandaki çalışmaların gelişmeye daha açık olduğu ve olası tehditlere karşı otomatik sistemler geliştirilmesinin topluma faydalı olabileceği düşünülerek bu kapsamda yapılan çalışmalar incelenmiştir. Fakat onlarca yıllık araştırmadan sonra dahi, bu alanda incelemek için çok sayıda araştırma fırsatı bulunmaktadır.

Çalışmada, kaynaklardaki çeşitli problemlere dikkat çekerek, çeşitli araştırma yönergeleri sunulmuştur. Bu çalışma sayesinde, yeni başlayanlar için bakış açısı sunulduğu, çalışma yapmak isteyenler için araştırma alanlarının ve kaynaklardaki eksikliklerin belirlendiği düşünülmektedir. Ayrıca bu alanda yapılmış en geniş Türkçe kaynak olduğu ve literatüre katkı sağlandığı düşünülmektedir.

Kaynakça

[1] Umut vakfı, http://www.umut.org.tr/ (22.08.2022)

[2] UNIDIR, https://unidir.org/ (22.08.2022)

[3] United Nation, https://www.un.org/disarmament/ (22.08.2022)

[4] Piza, E. L., Welsh, B. C., Farrington, D. P., & Thomas, A. L., "CCTV surveillance for crime prevention: A 40-year systematic review with meta-analysis", *Criminology & Public Policy*, *18*(1):135-159, (2019)

[5] Cohen, N., Gattuso, J. & MacLennan-Brown, K., "CCTV operational requirements manual 2009", *St. Albans: Home Office Scientific Development Branch,* (2009)

[6] Tickner, A. H., & Poulton, E. C., "Monitoring up to 16 synthetic television pictures showing a great deal of movement", *Ergonomics*, *16*(4):381-401, (1973)

[7] Darker, I., Gale, A., Ward, L., & Blechko, A., "Can CCTV reliably detect gun crime?", *In 2007 41st Annual IEEE International Carnahan Conference on Security Technology,* 264-271, (2007)

[8] Zou, Z., Shi, Z., Guo, Y., & Ye, J., "Object detection in 20 years: A survey", *arXiv preprint* arXiv:1905.05055, (2019)

[9] Zhao, Z. Q., Zheng, P., Xu, S. T., & Wu, X., "Object detection with deep learning: A review", *IEEE transactions on neural networks and learning systems*, 30(11):3212-3232, (2019)

[10] Xiao, Y., Tian, Z., Yu, J., Zhang, Y., Liu, S., Du, S., & Lan, X., "A review of object detection based on deep learning", *Multimedia Tools and Applications*, 79(33):23729-23791, (2020)

[11] Brunetti, A., Buongiorno, D., Trotta, G. F., & Bevilacqua, V., "Computer vision and deep learning techniques for pedestrian detection and tracking: A survey", *Neurocomputing*, 300:17-33, (2018)

[12] Wu, X., Kumar, V., Ross Quinlan, J. et al., "Top 10 algorithms in data mining", *Knowledge and information systems,* 14:1–37, (2008)

[13] Viola, P., & Jones, M.," Rapid object detection using a boosted cascade of simple features", *In Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001,* 1: I-I, (2001)

[14] Chen, C., Seff, A., Kornhauser, A., & Xiao, J., "Deepdriving: Learning affordance for direct perception in autonomous driving", *In Proceedings of the IEEE international conference on computer vision,* 2722-2730, (2015)

[15] Chen, Y., Zhao, D., Lv, L., & Zhang, Q., "Multi-task learning for dangerous object detection in autonomous driving", *Information Sciences*, 432:559-571, (2018)

[16] Chen, X., Kundu, K., Zhang, Z., Ma, H., Fidler, S., & Urtasun, R., "Monocular 3d object detection for autonomous driving", *In Proceedings of the IEEE conference on computer vision and pattern recognition,* 2147-2156, (2016)

[17] Ren, S., He, K., Girshick, R., & Sun, J., "Faster r-cnn: Towards real-time object detection with region proposal networks", *Advances in neural information processing systems*, 28:91-99, (2015)

[18] Brunetti, A., Buongiorno, D., Trotta, G. F., & Bevilacqua, V., "Computer vision and deep learning techniques for pedestrian detection and tracking: A survey" *Neurocomputing*, 300:17-33, (2018)

[19] Yang, B., Huang, C., & Nevatia, R., "Learning affinities and dependencies for multi-target tracking using a CRF model", *In CVPR 2011, 1233-1240,* (2011)

[20] Wojke, N., Bewley, A., & Paulus, D., "Simple online and realtime tracking with a deep association metric", *In 2017 IEEE international conference on image processing (ICIP)*, 3645-3649, (2017)

[21] Yang, Z., & Nevatia, R., "A multi-scale cascade fully convolutional network face detector", *In 2016 23rd International Conference on Pattern Recognition (ICPR),* 633-638, (2016)

[22] Coşkun, M., Uçar, A., Yildirim, Ö., & Demir, Y., "Face recognition based on convolutional neural network", *In 2017 International Conference on Modern Electrical and Energy Systems (MEES)*, 376-379, (2017)

[23] Kamencay, P., Benco, M., Mizdos, T., & Radil, R., "A new method for face recognition using convolutional neural network", *Advances in Electrical and Electronic Engineering*, *15*(4):663-672, (2017)

[24] Salari, A., Djavadifar, A., Liu, X. R., & Najjaran, H., "Object recognition datasets and challenges: A review", *Neurocomputing*, (2022)

[25] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., ... & Fei-Fei, L., "Imagenet large scale visual recognition challenge", *International journal of computer vision*, *115*(3):211-252, (2015)

[26] Karagiannakos S., "Localization and Object Detection with Deep Learning", *Towards Data Science,* (2019)

[27] Stanford University, "Lecture 11: Detection and Segmentation"*, Stanford University,* (2019)

[28] Gürbüz, M. E., & Gangal, A., "Döndürülmüş Kayan Pencereler Kullanarak İyileştirilmiş Hibrid Nesne Tespit Yöntemi ", *Eleco 2014 Elektrik – Elektronik – Bilgisayar ve Biyomedikal Mühendisliği Sempozyumu*, 573-577, (2014)

[29] Lienhart, R., & Maydt, J., "An extended set of haar-like features for rapid object detection", *In Proceedings. international conference on image processing*, 1: I-I, (2002)

[30] Papageorgiou, C. P., Oren, M., & Poggio, T., "A general framework for object detection", *In Sixth International Conference on Computer Visio,* 555-562, (1998)

[31] Lowe, D. G., "Object recognition from local scale-invariant features", In *Proceedings of the seventh IEEE international conference on computer vision*, 2:1150-1157, (1999)

[32] Dalal, N., & Triggs, B., "Histograms of oriented gradients for human detection", *In 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, 1: 886-893, (2005)

[33] Emrullah, A. C. A. R., & Özerdem, M. S., "Tarımsal imge dokularından HOG algoritması ile öznitelik çıkarımı ve öznitelik tabanlı toprak neminin tahmini" *Anatolian Science-Bilgisayar Bilimleri Dergisi*, *1*(1):1-7, (2016)

[34] Felzenszwalb, P. F., Girshick, R. B., McAllester, D., & Ramanan, D., "Object detection with discriminatively trained part-based models", *IEEE transactions on pattern analysis and machine intelligence*, *32*(9):1627-1645, (2010)

[35] Berwick R. & Idiot V., "An Idiot's guide to Support vector machines (SVMs)", *MIT, (*2022)

[36] Vapnik, V. N., "The Nature of Statistical Learning Theory (Second edition)", *Springer Science & Business Media*, 314, New York, (1995)

[37] Hinton, G. E., "Deep Belief Works", *Scholarpedia*, 4:5, (2009)

[38] Krizhevsky, A., Sutskever, I., & Hinton, G. E., "Imagenet classification with deep convolutional neural networks", *In Advances in neural information processing systems. In Advances in Neural İnformation Processing Systems 25 (NIPS 2012)*, 25(1):1097-1105, (2012)

[39] Girshick, R., Donahue, J., Darrell, T., & Malik, J., "Rich feature hierarchies for accurate object detection and semantic segmentation", *In Proceedings of the IEEE conference on computer vision and pattern recognition*, 580-587, (2014)

[40] Sermanet P., Eigen D., Zhang X., Mathieu M., Fergus R., and LeCun Y., "Overfeat: Integrated recognition, localization and detection using convolutional networks", *2nd International Conference on Learning Representations,* (2014)

[41] Varol Malkoçoğlu, A. B., "Akut lenfoblastik lösemi hücrelerinin derin öğrenme yöntemleri ile sınıflandırılması", *Yüksek lisans tezi,* Ondokuz Mayıs Üniversitesi, (2020)

[42] Popescu, A., & Coatrıeux, G., "Large-Scale Object Detection for Social Media User's Visual Privacy Protection", *Master Thesis,* Paris-Saclay University, (2020)

[43] Du, L., Zhang, R., & Wang, X., "Overview of two-stage object detection algorithms", *In Journal of Physics: Conference Series*, 1544(1):1-6, (2020)

[44] Uijlings, J. R., Van De Sande, K. E., Gevers, T., & Smeulders, A. W., "Selective search for object recognition", *International journal of computer vision*, *104*(2):154-171, (2013)

[45] Mohan S., *"*6 Different Types of Object Detection Algorithms in Nutshell", *Machine Learning Knowledge,* (2020)

[46] Girshick, R., Fast R-CNN. *In Proceedings of the IEEE international conference on computer vision,* 1440-1448, (2015)

[47] He, K., Zhang, X., Ren, S., & Sun, J., "Spatial pyramid pooling in deep convolutional networks for visual recognition", *IEEE transactions on pattern analysis and machine intelligence*, *37*(9):1904-1916, (2015)

[48] Erdem K., "Understanding Region of Interest (RoI Pooling)", (2020)

[49] Ren, S., He, K., Girshick, R., & Sun, J, "Faster R-CNN: Towards real-time object detection with region proposal networks", *Advances in neural information processing systems*, *28*, (2015)

[50] Dai, J., Li, Y., He, K., & Sun, J., "R-FCN: Object detection via region-based fully convolutional networks", *Advances in neural information processing systems*, *29*, (2016)

[51] He, K., Gkioxari, G., Dollár, P., & Girshick, R., "Mask R-CNN." In *Proceedings of the IEEE international conference on computer vision*, 2961-2969, (2017)

[52] Bozdağ, Z., "Histopatolojik Görüntülerde Tümör Bölütlenmesi", *Doktora Tezi*, İnönü Üniversitesi, (2021)

[53] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A., "You only look once: Unified, real-time object detection", *In Proceedings of the IEEE conference on computer vision and pattern recognition*, 779-788, (2016)

[54] Redmon, J., & Farhadi, A., "YOLO9000: better, faster, stronger." *In Proceedings of the IEEE conference on computer vision and pattern recognition*, 7263-7271, (2017)

[55] Redmon, J., & Farhadi, A., "Yolov3: An incremental improvement", *Cornell University*, (2018)

[56] Lin, T. Y., Dollár, P., Girshick, R., He, K., Hariharan, B., & Belongie, S., "Feature pyramid networks for object detectio", *In Proceedings of the IEEE conference on computer vision and pattern recognition*, 2117-2125, (2017)

[57] Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. M., "Yolov4: Optimal speed and accuracy of object detection", *arXiv e-prints*, arXiv-2004, (2020)

[58] Liu, S., Qi, L., Qin, H., Shi, J., & Jia, J., "Path aggregation network for instance segmentation", *In Proceedings of the IEEE conference on computer vision and pattern recognition*, 8759-8768, (2018)

[59] Aly, G. H., Marey, M. A. E. R., El-Sayed Âmin, S., & Tolba, M. F.," YOLO V3 and YOLO V4 for masses detection in mammograms with resnet and inception for masses classification", *In International Conference on Advanced Machine Learning Technologies and Applications*, 145-153, (2021)

[60] Sozzi, M., Cantalamessa, S., Cogato, A., Kayad, A., & Marinello, F.," Automatic bunch detection in white grape varieties using YOLOv3, YOLOv4, and YOLOv5 deep learning algorithms", *Agronomy*, *12*(2):319, (2022)

[61] Rahman, E. U., Zhang, Y., Ahmad, S., Ahmad, H. I., & Jobaer, S., "Autonomous vision-based primary distribution systems porcelain insulators inspection using UAVs", *Sensors*, *21*(3):974, (2021)

[62] Ge, Z., Liu, S., Wang, F., Li, Z., & Sun, J., "Yolox: Exceeding yolo series in 2021", *arXiv preprint*, (2021)

[63] Nepal, U., & Eslamiat, H., "Comparing YOLOv3, YOLOv4 and YOLOv5 for autonomous landing spot detection in faulty UAVs", *Sensors*, *22*(2):464, (2022)

[64] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C., "Ssd: Single shot multibox detector", *In European conference on computer vision*, 21-37, (2016)

[65] Szegedy, C., Reed, S., Erhan, D., Anguelov, D., & Ioffe, S., "Scalable, high-quality object detection", *arXiv preprint,* arXiv:1412.1441, (2014)

[66] Uner, M. K., Ramac, L. C., Varshney, P. K., & Alford, M. G., "Concealed weapon detection: an image fusion approach", *In Investigative image processing*, 2942:123-132. (1997).

[67] Sheen, D. M., McMakin, D. L., & Hall, T. E., "Three-dimensional millimeter-wave imaging for concealed weapon detection", *IEEE Transactions on microwave theory and techniques*, *49*(9):1581-1592, (2001)

[68] Xue, Z., Blum, R. S., & Li, Y. "Fusion of visual and IR images for concealed weapon detection", *In Proceedings of the Fifth International Conference on Information Fusion. FUSION 2002*, 2:1198-1205, (2002)

[69] Blum, R., Xue, Z., Liu, Z., & Forsyth, D. S., "Multisensor concealed weapon detection by using a multiresolution mosaic approach" *In IEEE 60th Vehicular Technology Conference, 2004.* 7:4597-4601, (2004)

[70] Upadhyay, E. M., & Rana, N. K., "Exposure fusion for concealed weapon detection", *In 2014 2nd International Conference on Devices, Circuits and Systems (ICDCS),* 1-6, (2014)

[71] O'reilly, D., Bowring, N., & Harmer, S., "Signal processing techniques for concealed weapon detection by use of neural networks", *In 2012 IEEE 27th Convention of Electrical and Electronics Engineers in Israel*, 1-4, (2012)

[72] Darker, I. T., Gale, A. G., & Blechko, A., "CCTV as an automated sensor for firearms detection: Human-derived performance as a precursor to automatic recognition", *In Unmanned/Unattended Sensors and Sensor Networks V*, 7112:208-219, (2008)

[73] Grega, M., Łach, S., & Sieradzki, R., "Automated recognition of firearms in surveillance video", In *2013 IEEE International Multi-Disciplinary Conference on Cognitive Methods in Situation Awareness and Decision Support (CogSIMA)*, 45-50, (2013)

[74] Tiwari, R. K., & Verma, G. K., "A computer vision based framework for visual gun detection using harris interest point detector", *Procedia Computer Science*, 54:703-712, (2015)

[75] Tiwari, R. K., & Verma, G. K., "A computer vision based framework for visual gun detection using SURF", *In 2015 International Conference on Electrical, Electronics, Signals, Communication and Optimization (EESCO),* 1-5, (2015)

[76] Grega, M., Matiolański, A., Guzik, P., & Leszczuk, M., "Automated detection of firearms and knives in a CCTV image", *Sensors*, 16(1):47, (2016)

[77] Vajhala, R., Maddineni, R., & Yeruva, P. R., "Weapon detection in surveillance camera images", *Electrical Engineering,* (2016)

[78] Buckchash, H., & Raman, B., "A robust object detector: application to detection of visual knives", *In 2017 IEEE International Conference on Multimedia & Expo Workshops (ICMEW),* 633-638, (2017)

[79] Lai, J., & Maples, S., "Developing a real-time gun detection classifier", *Course: CS231n, Stanford University*, (2017)

[80] Verma, G. K., & Dhillon, A., "A handheld gun detection using faster r-cnn deep learning", *In Proceedings of the 7th international conference on computer and communication technology*, 84-88, (2017)

[81] Olmos, R., Tabik, S., & Herrera, F. "Automatic handgun detection alarm in videos using deep learning", *Neurocomputing*, *275*:66-72, (2018)

[82] Akcay, S., Kundegorski, M. E., Willcocks, C. G., & Breckon, T. P., "Using deep convolutional neural network architectures for object classification and detection within x-ray baggage security imagery", *IEEE transactions on information forensics and security,* 13(9):2203-2215, (2018)

[83] Singleton, M., Taylor, B., Taylor, J., & Liu, Q. "Gun identification using tensorflow", *In International Conference on Machine Learning and Intelligent Communications*, 3-12, (2018)

[84] Gelana, F., & Yadav, A., Firearm detection from surveillance cameras using image processing and machine learning techniques, *In Smart innovations in communication and computational sciences*, 25-34, (2019)

[85] Romero, D., & Salamea, C., "Convolutional models for the detection of firearms in surveillance videos", *Applied Sciences*, *9*(15):2965, (2019)

[86] Fernandez-Carrobles, M., Deniz, O., & Maroto, F., "Gun and knife detection based on faster R-CNN for video surveillance", *In Iberian conference on pattern recognition and image analysis*, 441-452, (2019)

[87] Iqbal, J., Munir, M. A., Mahmood, A., Ali, A. R., & Ali, M., "Orientation aware object detection with application to firearms*", arXiv preprint arXiv:1904.10032,* 22. (2019)

[88] de Azevedo Kanehisa, R. F., & de Almeida Neto, A., "Firearm Detection using Convolutional Neural Networks", *11th International Conference on Agents and Artificial Intelligence (ICAART), 2:* 707-714, (2019)

[89] González, J. L. S., Zaccaro, C., Álvarez-García, J. A., Morillo, L. M. S., & Caparrini, F. S., "Real-time gun detection in CCTV: An open problem", *Neural networks*, 132:297-308, (2020)

[90] Pérez-Hernández, F., Tabik, S., Lamas, A., Olmos, R., Fujita, H., & Herrera, F., "Object detection binary classifiers methodology based on deep learning to identify small objects handled similarly: Application in video surveillance", *Knowledge-Based Systems*, *194*, (2020)

[91] Raturi, G., Rani, P., Madan, S., & Dosanjh, S., "ADoCW: An automated method for detection of concealed weapon", *In 2019 Fifth International Conference on Image Information Processing (ICIIP)*, 181-186, (2019)

[92] Noor, W. E. I. B. W., & Isa, N. M., "Object Detection: Harmful Weapons Detection using YOLOv4", *In 2021 IEEE Symposium on Wireless Technology & Applications (ISWTA),* 63-70, (2021)

[93] Narejo, S., Pandey, B., Rodriguez, C., & Anjum, M. R., "Weapon detection using YOLO V3 for smart surveillance system", *Mathematical Problems in Engineering*, (2021)

[94] Hashmi, T. S. S., Haq, N. U., Fraz, M. M., & Shahzad, M., "Application of Deep Learning for Weapons Detection in Surveillance Videos", *In 2021 International Conference on Digital Futures and Transformative Technologies (ICoDT2)*, 1-6, (2021)

[95] Kayalvizhi, R., Malarvizhi, S., Choudhury, S. D., Topkar, A., & Vijayakumar, P., "Detection of sharp objects using deep neural network based object detection algorithm", *In 2020 4th International Conference on Computer, Communication and Signal Processing (ICCCSP),* 1-5, (2020)

[96] Bhatti, M. T., Khan, M. G., Aslam, M., & Fiaz, M. J., "Weapon detection in real-time cctv videos using deep learning", *IEEE Access*, 9:34366-34382, (2021)

[97] Salido, J., Lomas, V., Ruiz-Santaquiteria, J., & Deniz, O., "Automatic handgun detection with deep learning in video surveillance images", *Applied Sciences*, 11(13):6085, (2021)

[98] Iqbal, M. J., Iqbal, M. M., Ahmad, I., Alassafi, M. O., Alfakeeh, A. S., & Alhomoud, A. "Real-Time Surveillance Using Deep Learning", *Security and Communication Networks*, (2021)

[99] Sivakumar, P., "Real Time Crime Detection Using Deep Learning Algorithm", *In 2021 International Conference on System, Computation, Automation and Networking (ICSCAN)*, 1-5, (2021)

[100] Kaya, V., Tuncer, S., & Baran, A., "Detection and classification of different weapon types using deep learning", *Applied Sciences*, 11(16):7535, (2021)

[101] Bushra, S. N., Shobana, G., Maheswari, K. U., & Subramanian, N., "Smart Video Survillance Based Weapon Identification Using Yolov5", *In 2022 International Conference on Electronic Systems and Intelligent Computing (ICESIC)*, 351-357, (2022)