VERİ BİLİMİ DERGİSİ
www.dergipark.gov.tr/veri

Bir Küçük Nesne Tespit Zorluğu Olarak Hava Görüntülerinden Araç Tespiti

Ömer ER[1]* , Hasan Şakir BİLGE[2]

[1]*Gazi Üniversitesi, Mühendislik Fakültesi, Elektrik - Elektronik Mühendisliği, Ankara*
[2] *Gazi Üniversitesi, Fen Bilimleri Enstitüsü, Elektrik - Elektronik Mühendisliği A.B.D., Ankara*

Özet

Hava görüntüleri üzerinde araç tespiti; istihbarat, keşif ve gözetleme açısından önemlidir. Ancak bu görev; düşük uzamsal çözünürlük, karmaşık arka plan, nesne üzerine düşen ışık/gölge farklılıkları ve nesnelerin çevre tarafından kamufle olması gibi sebeplerle zordur. Son zamanlarda geliştirilen CNN tabanlı ağlar umut vericidir ancak bu ağlar doğrudan küçük nesnelerin tespiti için yeterli değildirler ve ince ayara ihtiyaç duyarlar. Bu çalışmada daha hızlı RCNN algoritması ve görece büyük nesnelerin tespitinde başarısı kanıtlanmış ResNet ağı ile VEDAI veri kümesi üzerinde çalışılmıştır. Nesnelerin toplam görüntüdeki piksellerin 0.5×10^{-3}'ü kadar az yer kapladığı görüntüler üzerinde başarım artırımı için daha hızlı RCNN algoritmasında değişiklikler ile çeşitli deneyler yapılmıştır. Deneyler sonucunda %74.9 ortalama hassasiyet elde etmenin mümkün olduğu gösterilmiştir.

Makale Bilgisi

Başvuru:
16/07/2020
Kabul:
04/12/2020

Anahtar Kelimeler: *Nesne Tespiti, Derin Öğrenme, Hava Görüntüleri*

Vehicle Detection From Aerial Imagery As A Small Object Detection Difficulty

Abstract

Vehicle detection on aerial images is important for intelligence, reconnaissance and surveillance. However, this task is difficult for reasons such as low spatial resolution, complex background, light / shadow differences on the object, and the objects being camouflaged by the environment. Recently developed CNN-based networks are promising, but they are not sufficient for direct detection of small objects and need fine tuning. In this study, with the VEDAI dataset, faster RCNN algorithm and proven ResNet network in the detection of relatively large objects are investigated. Various experiments have been carried out with changes in the faster RCNN algorithm to increase performance on images where objects occupy as little as 0.5×10^{-3} of pixels in the total image. As a result of the experiments, it has been shown that it is possible to achieve an average sensitivity of 74,9%.

Keywords: *Object detection, deep learning, aerial images*

* İletişim e-posta: iletişim. omerer.eee@gmail.com
* Bu çalışmanın bir kısmı III. International Conference on Data Science and Applications 2020'de sözlü olarak sunulmuştur.

1 Giriş

İnsansız hava aracı (İHA) ve uydu teknolojisinin ilerlemesi, bunların kullanımının yaygınlaşması; elde edilen hava görüntü verilerinin artmasına ve bu alanda çeşitli uygulamaların geliştirilmesine olanak sağlamıştır. Bu uygulamaların büyük bir kısmı geniş alanları kapsayan hava görüntülerinin taranmasına dayanmaktadır ve ilgili tüm nesnelerin doğru algılanması ihtiyacını paylaşırlar. Buradaki temel amaç, hedef tanıma görevinden insan faktörünü çıkarmak ve otomatik bir hedef tanıma sistemi geliştirmektir. Özel olarak hava görüntüleri üzerinden hedef tanıma görevlerinden bir tanesi de araç tespitidir. Hava görüntüleri üzerinden araç tespiti; trafik yönetimi, akıllı ulaşım sistemlerinin tasarımı, afet yönetimi, otopark yönetimi, şehir planlama gibi birçok alan için ve özellikle istihbarat, askeri keşif ve gözetleme için önemlidir. Ancak gelişen uydu ve İHA elektro optik sistemlerine rağmen halan bu zorlu bir iştir. Bunun sebeplerinden ilki, yatay görüntüleme ile alınan görüntülere kıyasla hava görüntülerinde nesnelerin çok daha az bilgi içerebiliyor olmasıdır. Bunun temel etmeni uzak mesafedir. Ayrıntılar çok yüksek çözünürlüklü görüntüde dahi çok sınırlıdır. Bir diğer sebep, hava görüntülerinde araç benzeri nesnelerin çokluğudur. Evlerin çatıları, klima üniteleri, çöp kutuları ve elektrik üniteleri gibi birçok nesne yukarıdan bakıldığında araçlar ile karıştırılabilmektedir. Sebeplerden bir tanesi de nesnelerin geleneksel nesne algılama çalışmalarında olduğu gibi yerçekimin etkisi ile tek bir yön doğrultusunda bulunmamasıdır. Hava görüntülerinde araçlar rastgele yönlerdedir ve bu durum tespitlerini zorlaşmaktadır [1]. Araçları hava görüntülerinde tespit etmenin diğer zorlukları olarak; karmaşık arka plan, araçların üzerine düşen ışık ve gölge farklılıkları ve araçların bazen bir ağaçla bazen bir trafik işareti ile çevre tarafından kamufle olması belirtilebilir.

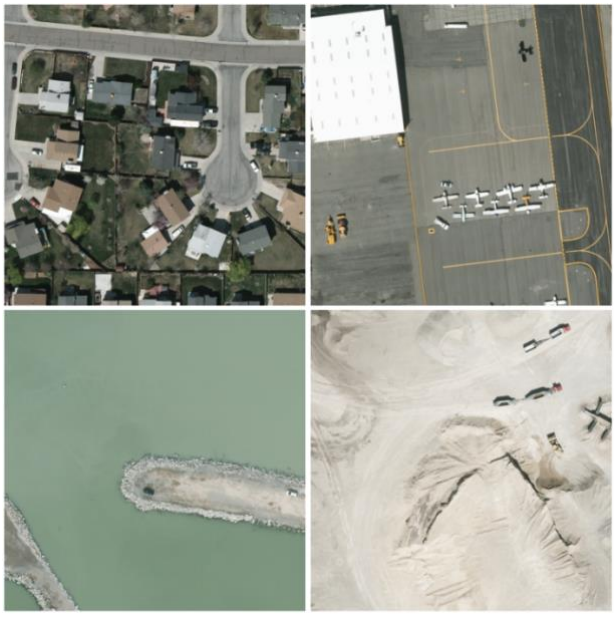
Nesne tespit teknolojisi ile ilgili olarak bugüne kadar önerilen yöntemler; sığ ve derin öğrenmeye dayalı olarak ikiye ayrılabilir. Sığ öğrenmeye dayalı yöntemler, el yapımı özellik çıkarma metotları ve sınıflandırıcı veya sınıflandırıcıların kombinasyonlarından oluşan yöntemler uygulanılarak geliştirilmiştir. Bu yöntemlere göre; ilk önce yönelimli gradyan histogramı (HOG) [2] ve ölçekten bağımsız öznitelik dönüşümü (SIFT) [3] gibi çok yaygın özellik çıkarma metotları ile nesnelerin öznitelikleri çıkarılmış, ardından destek vektör makinası (SVM) [4] ve Ada-Boost [5] gibi sınıflandırıcıların bir tanesi veya birden fazlası kullanılarak nesnelerin tespiti sağlanmıştır. Bu yöntemler nesneyi tek bir katı şablon olarak modellerken, değişime uğrayabilen parça bazlı model (DPM) [6] nesneyi uzaysal olarak organize edilmiş parçaların kombinasyonu şeklinde modellemiştir. DPM de dahil olmak üzere sığ öğrenmeye dayalı yöntemler, kayan pencere prensibi ile çalışırlar. Kayan pencere prensibine göre bir pencere tüm görüntü üzerinde kayarak nesnelerin tespitini sağlar, bu çok zaman alıcıdır. Derin öğrenmeye dayalı yöntemler ise Krizhevsky ve arkadaşlarının 2012 yılında ILSVRC2012 [7] yarışmasında büyük ses getiren derin evrişimli sinir ağları (CNN) [8]'nı önermesiyle başlamaktadır. İlerleyen dönemlerde bölge tabanlı evrişim sinir ağları (RCNN) [9] nesne tespitinde önemli performans kazanımı elde etmiştir. Ardından RCNN'in görece hız sorununu çözen hızlı RCNN [10] önerilmiştir. Bir sonraki adımda bölge önerilerini oluşturmak için kendi içinde bölge teklif ağı (RPN) ismi verilen, tamamen evrişimli bir bölge teklif ağı kullanan, daha hızlı RCNN [11] önerilmiştir. Bölge teklif tabanlı ağlar birkaç nedenden dolayı sığ öğrenmeden daha iyi performans göstermektedir [12]. Bunlardan biri; CNN'in, el yapımı özellik çıkarma metotlarının aksine özellik çıkarma işlemini daha güçlü bir şekilde otomatik olarak yapmasıdır. Diğer sebep ise sığ öğrenmeye dayalı yöntemlerin nesne tespiti için görüntünün tamamını arayan kayar pencere kullanmasının aksine bölge tabanlı CNN'lerin yalnızca önerilen bölgeler üzerinde nesneyi aramasıdır. Bu metot çok daha az zaman alıcıdır.

Son birkaç yılda [13], [14], [15] ve [16] gibi CNN tabanlı yeni algoritmalar geliştirilmiş olsa da bu yazıda daha hızlı RCNN algoritmasının ve ILSVRC 2015 [17] sınıflandırma yarışmasında birincilik elde eden artık ağ (ResNet) [18] modelinin kullanımına odaklanılmıştır. Bahsedilen algoritmalar, büyük ölçekli nesneleri barındıran ImageNet [19] ve MSCOCO [20] gibi veri kümlerinde başarı sağlamıştır. Ancak küçük nesnelerin tespiti için yeterli değildirler ve bu görev için bu algoritmalara ince ayar yapılması gerekmektedir. Bu kapsamda daha hızlı RCNN algoritması ve ResNet modeli üzerinde küçük nesnelerin tespiti için bir takım deneysel değişiklikler yapılmıştır. Bu yazının geri kalanı aşağıdaki gibi düzenlenmiştir: Bölüm 2'de kullanılan veri kümesi ve başarım artırımı için yapılan veri artırımı işlemi anlatılmaktadır. Bölüm 3'te daha hızlı RCNN

algoritması ve ResNet modeli tanıtılmaktadır. Bölüm 4'te daha hızlı RCNN ve ResNet üzerinde yapılan değişiklikler; Bölüm 5'te yapılan deneyler ve sonuçları açıklanmıştır. Son olarak, Bölüm 6'da sonuçların değerlendirilmesi yapılmıştır.

2 Veri Kümesi ve Veri Artırımı

Giriş bölümünde belirtildiği gibi bu çalışmanın motivasyonu küçük nesnelerin otomatik tespiti; özel olarak hava görüntüleri üzerinden araçların tespitidir. Bu motivasyonun gerçekleşmesi için uygun veri kümesinin seçimine ihtiyaç duyulmaktadır. Veri kümesinin özellikleri birtakım kısıtları karşılamalıdır: İstihbarat, keşif, gözetleme gibi çeşitli uygulamaların ihtiyaçlarını karşılayacak kadar hedeflerin yeterli tür çeşitliliğine ve sayısına sahip olmalıdır. Arka planlar; yol ve binaların olduğu yalnızca kentsel alanlar olmamalıdır, mümkün olduğunca çeşitli olmalıdır. Hedefler görüntü üzerinde piksel olarak küçük yer kaplamalıdır. Çalışmamız için bu kıstasları sağlayan VEDAI [21] veri kümesi seçilmiştir. VEDAI, halka açık Utah AGRC [22] veri tabanından alınarak orto-normalize edilmiş görüntüler içermektedir. Görüntülerin yerden uzaklığı sabittir. Bu özellik kasten seçilmiştir. Çünkü istihbarat, keşif ve gözetleme senaryolarında gözetim sistemi genellikle belirli yükseklik verilerine sahiptir ve bu durum ilgili uygulamalar için aranan bir özelliktir. VEDAI veri kümesinin bir diğer aranan özelliği de çok farklı arka planları barındırmasıdır. Göl, tarla, orman, dağlar, çöl gibi kırsal alanların yanında otoban, otopark, yerleşim alanları ve şantiyeler gibi kentsel arka plan alanlarını içermektedir.



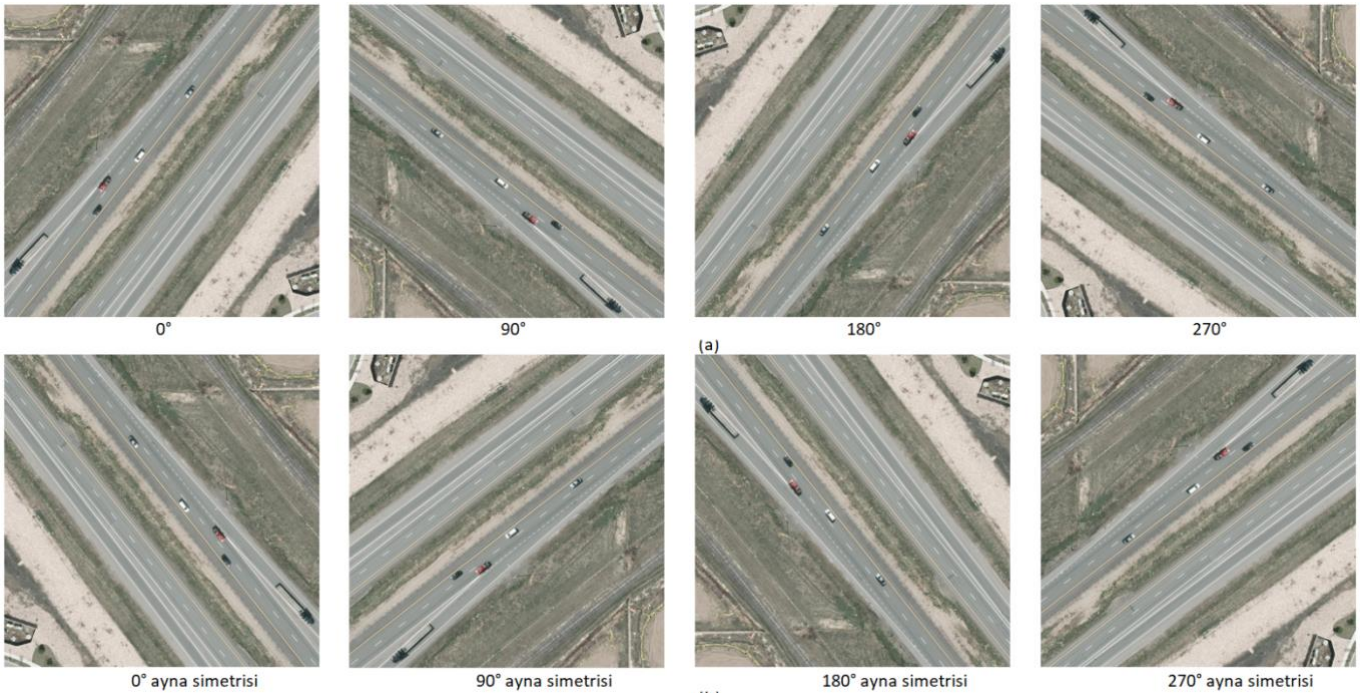
Şekil1. VEDAI veri kümesinden örnekler

VEDAI her biri 1239 adet olmak üzere 512×512 ve 1024×1024 renkli ve kızılötesi görüntüden oluşmaktadır. Bu çalışmada 1024×1024 çözünürlüğündeki renkli görüntüler üzerinde çalışılmıştır. Dokuz farklı araç türü mevcuttur. Bunlar: uçak, araba, kamyon, panelvan, pikap, kamp arabası, traktör, gemi ve iş makinesidir. Veri kümesi görüntü başına ortalama 5.5 araç sunmaktadır. Tablo 1 her bir araç türünün veri kümesindeki bulunma adedini göstermektedir. Bir görüntüde araçlar, görüntünün ortalama %0.7'sini kaplamaktadır. En küçük boyuta sahip araba sınıfı ise görüntüde toplam piksellerin 0.5×10^{-3}'ünü kaplamaktadır. 1024×1024 görüntülerde zemin örnekleme mesafesi 25 cm/piksel'dir. Çalışmalarımızda bazı uygulamaların ([23 ve 24]) aksine farklı görüntü ve boyuttaki uçak, traktör, tekne ve iş makinesi araç türleri de kullanılmıştır. Bu durumun eğitim başarısında bir miktar düşüşe sebep olacağı ön görülse de gerçek zamanlı bir araç tespit uygulamasında bu tür araçların da tespit edilmesi gerekliliğinden dolayı mevcut tür çeşitliliği korunmuştur.

Tablo 1.VEDAI veri kümesi araç sayısı

Sınıf	Adet
araba	1340
uçak	47
kamyon	300
pikap	950
traktör	190
kamp arabası	390
tekne	170
panelvan	100
iş makinesi	200

Ek olarak, VEDAI veri kümesinin yeterli büyüklüğe sahip olmaması ve bunun eğitim başarısına olumsuz etki edeceği düşünüldüğü için veri kümesi üzerinde veri artırma işlemi uygulanmıştır. Artırma işlemi için iki yöntem kullanılmıştır. İlk olarak, veri kümesindeki her bir görüntü saat yönünün tersinde 90°,180° ve 270°'lik açılarla döndürülmüştür. Ardından orijinal görüntünün ve döndürülmüş görüntülerin dikey eksende ayna simetrileri alınmıştır (Şekil 2'de gösterilmiştir). Böylece orijinal veri kümesinden sekiz kat büyüklükte bir veri kümesi elde edilmiştir. Bu işlemler uygulanırken görüntü üzerindeki araçların koordinatları da döndürme ve ayna simetrisi işlemlerine tabi tutulmuştur.



Şekil 2. Veri artırımı için a) Mevcut görüntüler saat yönünün tersinde dört açı ile döndürülmüştür b) Ardından her bir görüntünün dikey eksende ayna simetrisi alınmıştır

3 Daha Hızlı RCNN ve ResNet Modeli

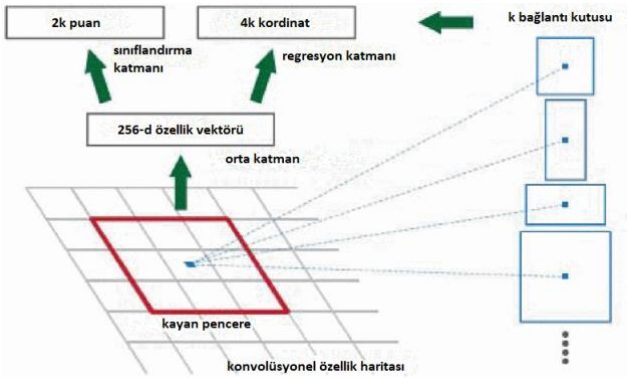
3.1 Daha Hızlı RCNN

Daha hızlı RCNN [11]; RCNN [9] ve hızlı RCNN[10] algoritmalarının dezavantajlı yönlerini çözerek üretilmiş bir algoritmadır. RCNN ve hızlı RCNN'de önce seçici arama algoritması [25] ile bölge önerileri oluşturulmakta, daha sonra bölge önerileri üzerinde nesne sınıfını sınıflandırmak ve sınırlayıcı kutuları oluşturmak için CNN tabanlı bir ağ kullanılmaktadır. Aralarındaki temel fark ise; RCNN bölge önerilerini piksel düzeyinde doğrudan girdi olarak aldığı görüntüden üretirken, hızlı RCNN bölge önerilerini girdi olarak aldığı özellik haritasından üretmektedir. Ancak her ikisinde de bölge önerisinin yapıldığı algoritma ile tespit ağı birbirinden ayrıştırılmıştır. Bu yaklaşım Shaoqing Ren vd.'ne göre iyi bir yaklaşım değildir. Bu yaklaşımda seçici arama yanlış negatif ürettiğinde, tespit ağı bundan zarar görmektedir. Ayrıca RCNN ve hızlı RCNN'de kullanılan seçici arama algoritması basit bir algoritmadır. Bölge önerisinin oluşturulduğu aşamada, hiçbir öğrenme gerçekleşmez. Bu, kötü aday bölgelerinin oluşmasına yol açabilmektedir. Aynı zamanda bu algoritma yavaş çalışmaktadır. Tüm bunları göz önünde bulunduran Shaoqing Ren ve arkadaşları "daha hızlı RCNN" ismini verdikleri algoritmalarında, seçici arama yerine ağın bölge önerilerini öğrenmesini sağlayan bölge önerileri ağı (RPN)'nı önermişlerdir. Bu CNN tabanlı ağ, tespit ağı ile paylaşımlıdır. Hızlı RCNN'de olduğu gibi daha hızlı RCNN'de de görüntü doğrudan özellik haritasının oluşturulması için evrişimsel bir ağın girişi olarak ağa beslenir. Bölge önerilerinin özellik haritası üzerinden üretilmesi için ayrı bir ağ kullanılır. Son olarak öngörülen bölge önerileri, ilgili bölgedeki görüntüyü sınıflandırmak ve sınırlama kutularının oluşturulması için ROI maksimum havuzlama katmanı kullanılarak yeniden şekillendirilir.

3.1.1 Bölge Önerileri Ağı (RPN)

Bölge öneri ağı girdi olarak aldığı özellik haritası üzerindeki her konumda 9 bölge önerisi üretmek için, k=9 adet bağlantı kutusu kullanmaktadır. Bu kutular standart bir Daha hızlı RCNN'de 128, 256 ve 512 ölçeklerindedir ve 1:1, 1:2, 2:1 en-boy oranlarındadır. Bir sınıflandırma katmanı özellik haritası üzerindeki her bir noktada k kutuları içinde nesne olup olmadığına dair 2×k adet puan üretmektedir. Bir regresyon katmanı da yine her nokta için k bağlantı kutuları içerisinde merkez koordinatlarını, genişlik ve yüksekliği içeren 4×k adet koordinat önerisi üretmektedir. Bir W×H özellik haritasında toplam W×H×k adet bağlantı kutusu vardır. Böylece bölge öneri ağı hangi konumun nesne içerdiğini önceden kontrol etmektedir. Ardından bağlantı kutuları, nesne sınıfını algılamak ve o nesnenin sınırlama kutusunu

oluşturmak için tespit ağına geçecektir. Şekil 3 bu ağın çalışma prensibini göstermektedir.



Şekil 3. Daha Hızlı RCNN Bölge Teklif Ağı [8]

3.2 ResNet Ağı

2012 yılında Krizhevsky ve arkadaşlarının AlexNet ağı [8] ile ILSVRC2012 [7] sınıflandırma yarışmasını kazanmasından bu yana CNN mimarisi derinleşme eğiliminde olmuştur. AlexNet yalnızca 5 konvolüsyon katmanına sahipken, VGG [26] ve GoogleNet [27] sırasıyla 19 ve 22 katmana sahiptir. Ancak katman sayısı arttıkça; gradyanın, geri yayılımda tekrar eden çarpımlar sonucunda sıfıra yaklaştığı, kaybolan gradyan sorunu ile karşılaşılır. Bu durum, ağın derinleşmesiyle performansın doyuma ulaşmasına ve hatta hızla bozulmaya başlamasına neden olmaktadır [18]. Çok derin ağları eğitmedeki bu sorun, ResNet ağı ile tanıtılan artık bloklarla çözülmüştür. Bu bloğun temel işlevi, önceki katmanın çıktısını bir sonraki katmana eklemektir. Buna göre; girdi X'in ağırlık katmanından geçmesiyle oluşan F(X) çıktısı, bir sonraki ağırlık katmanında önceki katmanın girdisi olan X ile birleşir. Ancak, çoğu kez konvolüsyon işlemine maruz kalan görüntüde olduğu gibi F(X)'de boyut daralması olmaktadır. Bu sebeple X ve F(X) aynı boyuta sahip olmaz. X; F(X) ile eşleşecek şekilde genişlemesi için doğrusal W ile çarpılır. Bu, X ve F(X) 'in bir sonraki katmana giriş olarak birleştirilmesini sağlar. Denklem (1) bu durumu matematiksel olarak ifade etmektedir.

$$Y = F(X, W_i) + W_s X \quad (1)$$

ResNet ağı, ILSVRC2015 [17] sınıflandırma yarışmasında ImageNet veri kümesi ile %3.57'lik hata oranı ile birinci olmuştur. Tablo 2 ResNet ağının yapısını göstermektedir.

4 Daha hızlı RCNN ve ResNet Modifikasyonları

Bu bölümde, daha hızlı RCNN algoritmasının ve ResNet ağının hava görüntüleri üzerinde başarılı bir şekilde uygulanabilmesi için yapılan değişiklikler açıklanmaktadır. İlk değişiklik, daha hızlı RCNN algoritmasının RPN modülünde kullanılan bağlantı kutularının sayısı ve boyutudur. İkinci değişiklik, ResNet ağının konvolüsyon katman sayısının değiştirilmesini içermektedir.

4.1 Bağlantı Kutu Sayısı ve Boyutu

Bölüm 3.1.1'de belirtildiği gibi daha hızlı RCNN algoritması, orijinalde bölge önerilerinin üretildiği

Tablo 2. ResNet ağı katman yapısı

Katman İsmi	Çıkış Boyutu	18 Katman	50 Katman	101Katman
Conv1	112×112	7×7,64, stride 2		
Conv2_x	56×56	3×3 maks. Havuzlama, stride 2		
		$\begin{bmatrix}3x3,64\\3x3,64\end{bmatrix} x2$	$\begin{bmatrix}1x1,64\\3x3,64\\1x1,256\end{bmatrix} x3$	$\begin{bmatrix}1x1,64\\3x3,64\\1x1,256\end{bmatrix} x3$
Conv3_x	28×28	$\begin{bmatrix}3x3,128\\3x3,128\end{bmatrix} x2$	$\begin{bmatrix}1x1,128\\3x3,128\\1x1,512\end{bmatrix} x4$	$\begin{bmatrix}1x1,128\\3x3,128\\1x1,512\end{bmatrix} x4$
Conv4_x	14×14	$\begin{bmatrix}3x3,256\\3x3,256\end{bmatrix} x2$	$\begin{bmatrix}1x1,256\\3x3,256\\1x1,1024\end{bmatrix} x6$	$\begin{bmatrix}1x1,256\\3x3,256\\1x1,1024\end{bmatrix} x23$
Conv5_x	7×7	$\begin{bmatrix}3x3,512\\3x3,512\end{bmatrix} x2$	$\begin{bmatrix}1x1,512\\3x3,512\\1x1,2048\end{bmatrix} x3$	$\begin{bmatrix}1x1,512\\3x3,512\\1x1,2048\end{bmatrix} x3$
	1×1	Ort.Havuzlama,1000 tam bağlantı,softmax		
Flop(işlem) Sayısı		1.8×10^9	3.8×10^9	7.6×10^9

RPN ağında {128,128; 128:256; 256,128; 256,256; 256,512; 512,256; 512,512; 512,1024 ve 1024,512} boyutlarında dokuz adet bağlantı kutusu kullanmaktadır. Bu bağlantı kutuları; geleneksel büyük boyutlu farklı en-boy oranlarına sahip nesne tespit uygulamaları için yeterlidir, ancak nesnenin görüntü sensöründen sabit uzaklıkta olduğu ve daha küçük nesneler içeren bir veri kümesi için uygun değildir. Orijinal daha hızlı RCNN algoritmasının kullandığı bağlantı kutularından daha küçük boyutlardaki bağlantı kutularının kullanılması daha yararlı olabilir. Ayrıca kullanılan bağlantı kutu sayısı, üretilen bölge önerileri ile orantılı olduğu için eğitim başarısına doğrudan etki edecektir. Bu sebeplerle YOLOv2 algoritmasında [15] kullanılan metot daha hızlı RCNN algoritmasına uygulanmıştır. Daha hızlı RCNN'in RPN ağı, farklı bağlantı kutu sayılarını ve boyutlarını kullanacak şekilde değiştirilmiştir. VEDAI veri kümesi üzerinden uygun bağlantı kutularının boyutlarını belirlemek için IoU mesafe metriği kullanılmıştır. Bağlantı kutularının IoU mesafe metriği kullanılarak veri kümesi üzerinden tahmin edilmesi, temel olarak benzer en boy oranlarına sahip kutuların birlikte kümelenip verilere uyan bağlantı kutularının tahmin edilmesi prensibi ile çalışır [15]. Bağlantı kutuları; bir, iki, üç, dört ve beş adetlerinde olacak şekilde her biri için IoU mesafe metriği ile oluşturulmuştur. Tablo 3 oluşturulan bağlantı kutularını göstermektedir. Bir, iki, üç, dört ve beş adetlerinde bağlantı kutularının kullanıldığı her bir senaryo için daha hızlı RCNN ResNet50 ağ modelleri oluşturulmuştur.

Tablo 3. Kullanılan bağlantı kutuları

Adet	Bağlantı Kutuları
1	[40,40]
2	[36,36;80,80]
3	[72,76;25,46;47,27]
4	[43,24;77,94;26,47;69,37]
5	[42,23;103,103;66,34;41,68;24,44]

4.2 Konvolüsyon katman sayısı

Tablo 2'de gösterildiği gibi ResNet ağı içerdiği katman sayısına göre isim almaktadır. Örneğin ResNet50; kırk dokuz konvolüsyon katmanı ve bir tam bağlantılı katmandan oluşmaktadır. Benzer şekilde ResNet18; on yedi konvolüsyon katmanı ve bir tam bağlantılı katmandan oluşmaktadır. ResNet ağının tüm varyasyonları için giriş boyu 224×224'tür. Standart ResNet ağını kullanan daha hızlı RCNN algoritmasında özellik haritasının çıktısı ve RPN'nin girişi olarak conv4_x'in son konvolüsyon katmanı kullanılmaktadır. ResNet ağının tüm varyasyonları için conv4_x'ün çıktısı, 14×14'lük özellik haritası çözünürlüğüne sahiptir. Bu, ağ derinliğinin özellik haritası boyutunu değiştirmediğini gösterir. Ancak özellik haritasını oluştururken kullanılan konvolüsyon katmanı sayısı değiştiği için bir özellik haritası noktasının temsil edildiği piksel derinliği de değişmektedir. Farklı derinlikteki ResNet ağlarını kullanmak faydalı olacağı değerlendirilmektedir. Bu sebeple ImageNet'te daha önceden eğitilmiş ResNet18, ResNet50 ve ResNet101 ağlarının yanı sıra bu ağlara farklı sayıda konvolüsyon katmanlarını ekleme ve çıkarma yaparak yeni ağ modelleri oluşturulmuştur.

5 Deneysel Çalışmalar

Deneyler için ağ eğitim ve test kodları Matlab2019b yazılımında yazılmıştır. Tüm deneyler Google bulut sistemi üzerinden uzaktan bağlanılan Intel Xeon CPU(4 çekirdekli, 2.30 GHz) 15GB ram, NVIDIA Tesla T4 GPU (16GB ram) ve Windows Server 2019 ile donatılmış bir bilgisayar üzerinden gerçekleştirilmiştir. Veri kümesinin %80'i eğitim, %20'si test için kullanılmıştır. Eğitimlerde mini yığın boyutu ismi verilen, bir görüntüden pozitif ve negatif nesne örneklerinin rastgele üretilmesine dayanan örnekleme stratejisi kullanılmıştır. Mini yığın boyutu, eğitimi hızlandıran aynı zamanda GPU bellek kullanımını önemli ölçüde artıran bir parametredir. Kullanılan GPU bellek boyutu sınırlamasından dolayı eğitimlerde mini yığın boyutu 4 olarak seçilmiştir. Böylece her iterasyonda rastgele 4 pozitif ve negatif nesne önerisi, bir alt veri kümesi oluşturmak için üretilmiştir. Pozitif ve negatif nesne önerileri, Denklem 2'de tanımlanan IoU Jaccard indeksi kullanılarak belirlenmektedir:

$$IoU = \frac{A_{hedef} \cap A_{tahmin}}{A_{hedef} \cup A_{tahmin}} \quad (2)$$

Denklemde A_{hedef}, veri kümesinde nesnenin belirtilen koordinatlarını yani olması gereken alanı; A_{tahmin} ise tahmin edilen alanı temsil etmektedir. Eğitimlerde IoU indeksi; 0.7'den büyük değerler için pozitif nesne, 0.1 ile 0.3 arasında olan değerlere negatif nesne atfında bulunulmuştur. Bu değerler, *Negatif Kesişim Aralığı* ve *Pozitif Kesişim Aralığı* parametreleri ile eğitimden önce ağda belirtilmiştir. Maksimum epok sayısı 25 seçilmiştir. Başlangıç öğrenme oranı 0.001 kullanılmıştır. Bu değer her 12 epok'ta bir 0.1çarpanı ile düşürülerek, 12'den 24'e kadar 0.0001 öğrenme oranı, 25'inci epokta 0.00001 öğrenme oranı kullanılmıştır. Ayrıca

momentum değeri olarak 0.9 kullanılmıştır. Tablo 4 kullanılan eğitim parametrelerini göstermektedir.

Modellerin performansının nicel olarak değerlendirilmesi için yaygın olarak kullanılan hassasiyet, geri çağırma oranı, F1 skoru ve ortalama hassasiyet (AP) metrikleri kullanılmıştır. Hassasiyet, geri çağırma oranı ve F1 skoru sırasıyla Denklem 3, Denklem 4 ve Denklem 5 ile hesaplanır:

$$Geri\ Çağırma(R) = \frac{Doğru\ Pozitif}{Doğru\ Poz. + Yanlış\ Neg.} \quad (3)$$

$$Hassasiyet(P) = \frac{Doğru\ Pozitif}{Doğru\ Poz. + Yanlış\ Poz.} \quad (4)$$

$$F1\ skoru = \frac{2 * Geri\ Çağırma * Hassasiyet}{Geri\ Çağırma + Hassasiyet} \quad (5)$$

Test sırasında nesne tespit tahminleri, eğitilmiş ağın oluşturduğu sınırlayıcı kutuların veri kümesinde nesnenin konumunu belirten alan ile çakışmasına göre doğru veya yanlış pozitif olarak atanmaktadır. Yine bu atamada IoU Jaccard indeksi kullanılmıştır. IoU değerinin 0.5'ten büyük olduğu tahminler, doğru pozitif olarak atanır. Bir nesne için birden fazla sınırlayıcı kutu oluşturulması cezalandırılır. Örneğin eğitilmiş ağ test sırasında bir nesne için dört adet sınırlayıcı kutu oluşturmuş ise bunun biri IoU değeri 0.5'ten büyük olan doğru pozitif, geri kalan üçü yanlış pozitif olarak değerlendirilir. Değerlendirmeler, hassasiyet ve geri çağırma oranlarının hesaplanmasında kullanılır. Hassasiyet doğru pozitif tespitlerinin, geri çağırma oranı ise pozitif tespitlerin tamamının bir değerlendirmesidir. Ortalama hassasiyet ise hassasiyet-geri çağırma grafiğinin altında kalan alanı tanımlamaktadır. Ortalama hassasiyet ve F1 skoru detektörün performansını ortaya koyan iki metriktir. Yüksek ortalama hassasiyet ve F1 skor değerleri, yüksek performans anlamına gelir.

Tablo 4. Eğitim Parametreleri

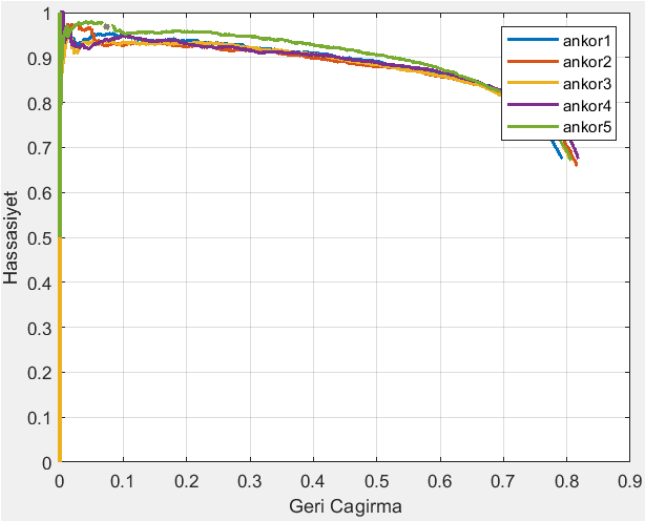
Parametreler	Değerler
Mini Yığın Boyutu	4
Başlangıç öğrenme hızı	0.001
Öğrenme Hızı Azaltma Periyodu	12 epok
Öğrenme Hızı Azaltma Çarpanı	0.1
Maksimum Epok Sayısı	25
Negatif Kesişim Aralığı	[0,0.3]
Pozitif Kesişim Aralığı	[0.7,1]
Momentum	0.9

5.1 Bağlantı Kutu Sayısı ve Boyutunun Etkisi

İlk deneyde, bağlantı kutu sayısı ve bu kutuların boyutlarının algılama performansı üzerindeki etkisi araştırılmıştır. ImageNet ile önceden eğitilmiş ResNet50 ağı ile daha hızlı RCNN'in RPN modülünde Tablo 3'te gösterilen beş farklı bağlantı kutusunu kullanarak beş farklı eğitim yapılmıştır. Ardından test için ayrılan veri kümesi kullanılarak Bölüm 5'te belirtilen metrikler hesaplanmıştır. Ayrıca her birinin eğitim süreleri kayıt edilmiştir. Tablo 5, beş farklı bağlantı kutusu ile oluşturulan modellerin test sonuçlarını ve eğitim sürelerini göstermektedir. Şekil 4 ise her bir modelin geri çağırma-hassasiyet grafiğini göstermektedir. Eğitim süresi saat cinsinden verilmektedir.

Tablo 5. Farklı bağlantı kutularına ait sonuçlar

Model	R	P	AP	F1	Eğitim Süresi
Ankor1	0.791	0.676	0.705	0.729	74.6
Ankor2	0.815	0.660	0.719	0.729	80.8
Ankor3	0.806	0.672	0.712	0.733	82.2
Ankor4	0.818	0.676	0.724	0.740	82.4
Ankor5	0.806	0.673	0.728	0.734	84.1



Şekil 4.Farklı bağlantı kutularına ait R-P grafiği

Buna göre, birkaç istisna dışında bağlantı kutu sayısı arttıkça; ortalama hassasiyet ve F1 skor değerleri artmaktadır. Ancak bununla birlikte işlem yükü de arttığı için eğitim süreleri de artış eğilimindedir. Modeller, bağlantı kutu sayılarına göre ankor1, ankor2, ankor3, ankor4 ve ankor5 isimlerini almışlardır. Ankor4 modeli, algılama başarısında (hem ortalama hassasiyet hem de F1 skor açısından değerlendirildiğinde) ankor1, ankor2 ankor3 modellerine göre radikal bir artış göstermiştir. Ancak ankor4'ten sonraki ankor5'in eğitim başarısında, ortalama hassasiyet açısından

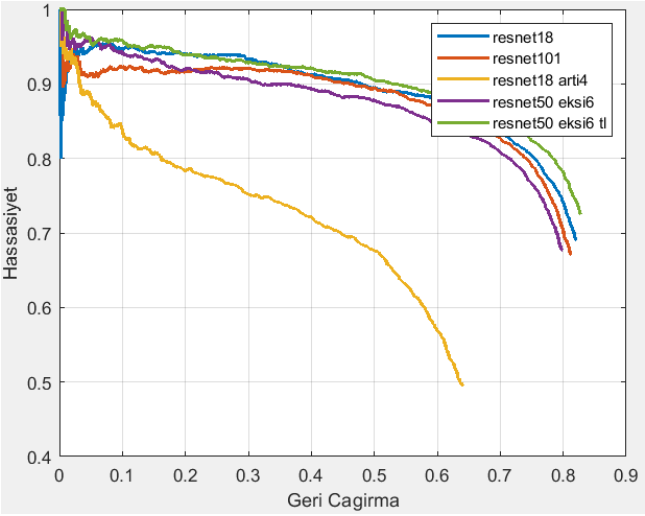
bakıldığında radikal bir artış görülmemiş aksine F1 skoru açısından bakıldığında bir düşüş görülmektedir. Bu sonuçlar ve eğitim süreleri dikkate alınarak bundan sonraki deneylerde ankor4 kullanılmasına karar verilmiştir.

5.2 Konvolüsyon Katman Sayısının Etkisi

Bir sonraki eğitimde, ResNet ağ derinliğinin algılama performansına etkisi araştırılmıştır. ResNet50'den daha sığ olan ResNet18 ve ResNet50'den daha derin olan ResNet101 ağları ile eğitimler yapılmıştır. Bu bölümde yapılan tüm test sonuçları Tablo 6 ve Şekil 5'te gösterilmiştir. Buna göre ResNet18, hem ResNet50'den hem de ResNet101'den daha iyi performans göstermiştir. Bu sonuç [26,28] gibi daha derin ağların daha iyi olduğunu savunan önceki tecrübelere ve hava görüntülerinde de bu bilginin iyi çalıştığını gösteren [24,29] gibi çalışmalara aykırıdır, ancak [30] ile paralellik göstermektedir. ResNet50'den daha sığ bir ağın, VEDAI veri kümesinde daha başarılı bir sonuç elde etmesinin ardından; ResNet18 ile ResNet50 ağ derinlikleri arasında bir ağın algılama başarısına etkisi araştırılmıştır. Bunun için ResNet18'in conv2 bloğundaki iki artık bloğa çıktı boyutu korunacak şekilde (64 tane 3×3 filtreye sahip) 2'şer konvolüsyon katmanı eklenerek, ResNet18'den daha derin bir model eğitilmiştir. Eklenen konvolüsyon katmanlarının ağırlık ilklendirilmesi, sıfır ortalama ve sıfır varyans ile bir dağılım sağlayan, Xavier ilklendirici olarak da bilinen, Glorot [31] yöntemi ile yapılmıştır. Ayrıca ResNet50'nin conv4 bloğundan 2 artık blok yani 6 konvolüsyon katmanı çıkarılarak oluşturulan, ResNet50'den daha sığ bir model eğitilmiştir. Tablo 6'da görüldüğü üzere ResNet18_arti4 modeli, daha önceden eğitilmiş standart ResNet18 modelinden çok daha az başarı sağlamıştır. Bu durum; standart ResNet18'in tüm ağırlıklarının daha önceden eğitilerek optimize olmuş olmasına rağmen, ResNet18_arti4 modelinin araya sonradan eklenen dört konvolüsyon katmanı nedeni ile eklenen konvolüsyon katmanı ve sonraki katmanlardaki ağırlıkların yeniden optimize olması gerekmesinden kaynaklanmaktadır. Aynı şekilde, ResNet50_eksi6 için de ResNet50 modelinin ortasından bir kısım katman çıkarıldığı için yeni modelde çıkarılan katmandan sonraki katman ağırlıkları yine optimize olmaya ihtiyaç duyar. Ancak atılan katmanlardan sonra birleştirilen iki tarafın ağırlıkları arasındaki fark, görece az olduğu için ağırlıkların optimizasyonu çok daha az maliyetle olmaktadır. Bu sebepten ResNet50_eksi6 modelinin ağ başarısı, ResNet50 ve ResNet18'den az ancak hatırı sayılır bir değerde çıkmıştır.

Tablo 6. Farklı derinlikteki ağlara ait sonuçlar

Model	R	P	AP	F1	Eğitim Süresi
ResNet18	0.820	0.689	0.734	0.749	46.8
ResNet101	0.812	0.671	0.718	0.735	109.5
ResNet18 artı4	0.641	0.495	0.475	0.559	51.2
ResNet50 eksi6	0.800	0.677	0.702	0.732	86.5
ResNet50 eksi6_tl	0.827	0.725	0.749	0.773	87.6



Şekil 5. Farklı derinlikteki ağlara ait R-P grafiği

Bu aşamaya kadar yapılan tüm eğitimler, ILSVRC yarışmasında önceden eğitilmiş ResNet ağlarına ilk önce daha hızlı RCNN algoritmasına uygun olarak RPN katmanlarının eklenmesi ve çıktı katmanlarının güncellenmesi ile yapılmıştır. Yani ResNet18_artı4 modeli için ilk olarak önceden eğtilmiş hazır ResNet18 modeli, daha hızlı RCNN algoritmasına uygun olarak değiştirilmiş ardından 4 katman çıkarılarak eğitilmiştir. Benzer işlem ResNet50_eksi6 için de uygulanmıştır. Bu son iki deneyde görülmüştür ki önceden eğitilmiş bir ağı bir algoritmaya uyarlayıp birkaç katmanın eklenmesi veya çıkarılması işlemlerinden sonra eğitilmesi, önceden eğitilmiş bir ağın bir algoritmaya uyarlandıktan sonra eğitilmesine göre ağırlıkların çok daha büyük maliyetlerde optimizasyonuna ihtiyaç duyulduğu için algılama başarısında düşüşlere neden olmaktadır. Optimizasyon maliyetini azaltmak için bir sonraki deneyde *öğrenmenin aktarılması (transfer learning)* yöntemi denenmiştir. Önceki deneylerde eğitilen

daha hızlı RCNN_ResNet50 modelinin conv4 bloğu üzerinden 6 konvolüsyon katmanı çıkarılarak tekrardan eğitimi sağlanmıştır. Yani daha hızlı RCNN_ResNet50 modelinin ağ kat sayıları eğitim için kullanılmıştır. Algılama test sonuçlarına bakıldığında görülmüştür ki öğrenmenin aktarılması ile yapılan daha hızlı RCNN ResNet50_eksi6_tl modeli, hem daha hızlı RCNN ResNet50 modelinden hem de daha hızlı RCNN ResNet50_eksi6 modelinden başarılı bir sonuç elde etmiştir. Bu modelin VEDAI veri kümesi üzerindeki algılama örnekleri Şekil 6'da verilmiştir. Yapılan deneylerde eğitim süreleri değerlendirildiğinde, beklendiği gibi ResNet50'den daha sığ olan ResNet18 ile oluşturulan modelin eğitiminin çok daha az zaman aldığı; ResNet50'den daha derin olan ResNet101 ile oluşturulan modelin eğitiminin çok daha fazla zaman aldığı görülmüştür. Katmanların orijinal ağlardan eklenerek veya çıkarılarak deforme edilen modellerin eğitimlerinin ise orijinallerinden bir miktar daha fazla sürdüğü gözlenmiştir.

5.3 Sonuçların Karşılaştırılması

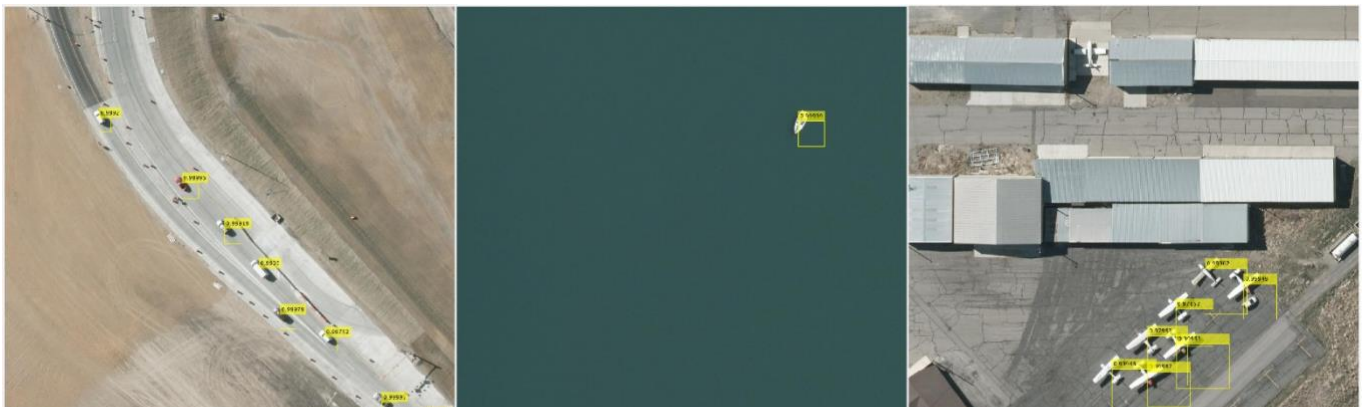
Elde edilen sonuçlar [23] ve [24]'teki sonuçlar ile karşılaştırılmıştır. [23]'te VEDAI veri setinden tekne, traktör ve uçak sınıfları örnek azlığı sebebi ile çıkarılmıştır. Ardından bu yazıda olduğu gibi döndürme ve ayna simetrisi alma yöntemleri ile mevcut veriler sekiz katına çıkarılmıştır. Hiçbir değişiklik yapılmadan orijinal daha hızlı RCNN algoritması ile farklı önceden eğitilmiş ağ modelleri ve kendi önerdikleri daha hızlı RCNN benzeri, bölge önerilerinin ayrı bir ağ tarafından yapıldığı ikili bir ağ yapısı içeren bir modelin eğitim sonuçları Tablo 7'de paylaşılmıştır. Bu yazıda elde edilen sonuçların, hiçbir değişiklik yapılmadan kullanılan daha hızlı RCNN ve benzeri algoritmalardan çok daha başarılı olduğu görülmektedir. Bu da mevcut algoritmaların küçük nesne tespitlerinde doğrudan kullanımının yetersiz olduğu, küçük nesne tespiti için bu algoritmaların ince ayara ihtiyaç duyduğu tezini doğrulamaktadır. [24]'te VEDAI veri kümesinden yalnızca üstten bakıldığında birbirine çok benzeyen, benzer boyutlara sahip araba, pikap ve panelvan araç sınıfları kullanılmış, diğer sınıflar atılmıştır. Ayrıca veri kümesinin sağladığı görüntülerdeki araçların koordinatları, merkez noktaları korunarak yüksekliği ve genişliği 40'a 40 olacak şekilde bir değişikliğe gidilmiş. Eğitimlerde de daha hızlı RCNN 'in bağlantı kutuları 40×40 kullanılmıştır. Farklı olarak, VGG-16 ağı kullanılmış ve veri artırımı yapılmamıştır. RPN'e bölge önerilerini üretmesi için giriş olarak beslenen özellik haritasının oluştuğu katmanlar, ağın farklı derinliklerinden seçilmiş ve yapılan deney sonuçları Tablo 7'de paylaşılmıştır. Sonuçların, bu yazıda elde edilen algılama başarılarından daha başarılı olduğu görülmektedir. Bu çalışmada eğitimlerde birbirinden çok farklı görünen uçak, iş makinesi, traktör, kamyon gibi dokuz farklı araç sınıfı kullanılmıştır. [4]'te ise birbirine her açıdan çok benzeyen üç farklı araç sınıfı kullanılmıştır. Bunun öğrenmeyi kolaylaştırdığı ve algılama başarısını artırdığı değerlendirilmektedir.

Tablo 7.Farklı çalışmaların karşılaştırılması

Model	AP
Daha Hızlı R-CNN (Z&F)[23]	0.308
Daha Hızlı R-CNN (VGG-16)[23]	0.421
Önerilen Model(VGG-16)[23]	0.546
Daha Hızlı RCNN ResNet50_eksi6_tl	**0.749**
Daha Hızlı RCNN(VGG-16) conv3[24]	0.972
Daha Hızlı RCNN(VGG-16) conv4[24]	0.974
Daha Hızlı RCNN(VGG-16) conv5[24]	0.967

6 Sonuçlar

Bu çalışmada, küçük nesne tespiti için uygun zorluktaki bir veri kümesi olan VEDAI ile araç algılama başarısını artırmak üzere ResNet ağı ve



Şekil 6. Daha Hızlı RCNN ResNet50_eksi6_tl modeline ait algılama sonuçları

daha hızlı RCNN algoritmasında değişiklikler yapılmıştır. Daha hızlı RCNN algoritmasının bölge öneri ağında kullanılan bağlantı kutu sayısı ve boyutunun, küçük nesneleri tespit etmekte olan etkisini araştırmak üzere yaptığımız deneysel çalışmalarda; genel olarak, kullanılan bağlantı kutu sayısının artmasının algılama başarısını arttırdığı ancak işlem yükünün artmasından dolayı ağ eğitim süresini de arttırdığı gözlenmiştir. Daha hızlı RCNN algoritmasında önerilen orijinal boyutlarındaki bağlantı kutularının küçük nesne tespitinde yetersiz oldukları, daha küçük boyutlu ve veri kümesine uygun bağlantı kutularının IoU mesafe metriği ile hesaplanarak kullanılmasının algılama başarısında daha iyi sonuç gösterdiği; algılama başarı sonuçlarının orijinal bağlantı kutuları kullanan diğer çalışma sonuçları ile karşılaştırılmasında gösterilmiştir. Böylece küçük nesne tespiti için orijinal algoritmaların yetersiz olduğu, bu görevler için orijinal algoritmalara ince ayar yapılması gerektiği savı doğrulanmıştır. Ağ katman sayısının, küçük nesneleri tespit etmekte olan etkisini araştırmak üzere yaptığımız deneysel çalışmalarda ise; küçük nesne tespiti için daha sığ ağların daha faydalı olabileceği sonucuna ulaşılmıştır. Beklenildiği gibi ağ derinliği arttıkça işlem sayısı da arttığı için ağ eğitim süresinin arttığı gözlenmiştir. Ayrıca öğrenmenin aktarımı yöntemi ile ortalama hassasiyette %74.9'a kadar algılama başarısının sağlandığı gösterilmiştir.

Yapılan deneylerde hiper parametreler üzerinde eniyileme çalışmaları yapılmamıştır. Ayrıca istihbarat, keşif ve gözetleme görevlerinde gözetleme sistemine belirli bir yükseklik verisi tanımlandığı için, kasten seçilen, görüntülerin yerden uzaklığının sabit olduğu tek bir veri kümesi kullanılmıştır. Hiper parametreler üzerinde eniyileme çalışmaları yapılarak ve eğitimlere yerden yükseklik verileri aynı olan farklı kaynaklardan elde edilen hava görüntülerini içeren veri kümeleri dahil edilerek, önerilen modelin bir ürün haline getirilebileceği değerlendirilmektedir.

Kaynaklar

[1] Liu C, Ding Y, Zhu M, Xiu J, Li M, Li Q. "Vehicle Detection in Aerial Images Using a Fast Oriented Region Search and the Vector of Locally Aggregated Descriptors". *Sensors,* 19(15), 3294, 2019.

[2] Dalal N., Triggs B. Histograms of oriented gradients for human detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*; San Diego, CA, USA. 20–25 June 2005.

[3] Lowe D. "Distinctive image features from scale-invariant keypoints". *International Journal of Computer Vision*, 60, 91–110, 2004.

[4] Chang C, Lin C. LIBSVM: A library for support vector machines. ACM Trans. Intell. Syst. Technology, 389–396, 2011.

[5] Viola P, Jones M. Rapid object detection using a boosted cascade of simple features. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Kauai, USA, 2001.

[6] Felzenszwalb P, Girshick R, McAllester D, Ramanan D. Object detection with discriminatively trained part based models. *IEEE Trans. Pattern Anal. Mach. Intell*, 32, 1627–1645, 2010.

[7] Deng J, Berg A, Satheesh S, Su H, Khosla A, Li F. *ImageNet Large Scale Visual Recognition Competition 2012,* accessed on 10 July 2017.

[8] Krizhevsky A, Sutskever I, Hinton G. ImageNet classification with deep convolutional neural networks. *Proceedings of the 25th International Conference on Neural Information Processing System*, Lake Tahoe, NV, USA, 3–6 December 2012.

[9] Girshick R, Donahue J, Darrell T, Malik J. Rich feature hierarchies for accurate object detection and semantic segmentation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, OH, USA, 23–28 June 2014.

[10] Girshick R. Fast R-CNN. 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, 1440-1448, 2015

[11] Ren S, He K, Girshick R, Sun J. Faster R-CNN: Towards real-time object detection with region proposal networks. *Neural Information Processing Systems (NIPS),* 2015.

[12] Tayara H, Soo K, Chong K. Vehicle Detection and Counting in High-Resolution Aerial Images Using Convolutional Regression Neural Network. IEEE Access, 1-1, 2017

[13] Redmon J, Farhadi A. "An incremental improvement". *CoRR*, vol. 3, 2018.

[14] Redmon J, Divvala S, Girshick R, Farhadi A. "You only look once: Unified real-time object detection". *IEEE Conference on ComputerVision and Pattern Recognition*, 2016.

[15] Redmon J, Farhadi A. "YOLO9000: Better, Faster, Stronger". *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* Honolulu, HI, 6517-6525, 2017

[16] Liu W, Anguelov D, Erhan D, Szegedy C, Reed S. "Ssd: Single shot multibox detector". *European Conference on ComputerVision (ECCV),* 2016.

[17] Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, Huang Z, Karpathy A, Khosla A, Bernstein M, Berg A. "Large Scale Visual Recognition Competition". *International Computer Vision Journal,* 2015.

[18] He K, Zhang X, Ren S, Sun J. "Deep residual learning for image recognition". *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[19] Deng J, Dong W, Socher R. "Imagenet: A large-scale hierarchical image database". *CVPR*, 248-255, 2009.

[20] Lin T, Maire M, Belongie S, Hays J, Perona P, Ramanan D, Dollár P. "Microsoft COCO: common objects in context". *ECCV*,740-755, 2014.
[21] Razakarivony S, Jurie F. Vehicle detection in aerial imagery: A small target detection benchmark. *J. Vis. Commun. Image Represent,* 34, 187–203, 2016
[22] İnternet: Utah agrc URL:http://gis.utah.gov/. 2012.
[23] Zhong J, Lei T, Yao G. Robust Vehicle Detection in Aerial Images Based on Cascaded Convolutional Neural Networks. *Sensors*, 17(12), 2720, 2017.
[24] Sakla W, Konjevod G, Mundhenk T. "Deep Multi-modal Vehicle Detection in Aerial ISR Imagery". *2017 IEEE Winter Conference on Applications of Computer Vision*, Santa Rosa, CA, 916-923, 2017.
[25] Uijlings J, Van de Sande K, Gevers T, Smeulders A. Selective search for object recognition. *Int. J. Comput. Vision*, 104,154–171, 2013
[26] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv,* 1409, 1556, 2014.
[27] Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A. "Going deeper with convolutions*". In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1–9, 2015.
[28] Szegedy C, Ioffe S, Vanhoucke V. Inception-v4 inception-ResNet and the impact of residual connections on learning. *CoRR*, vol. abs/1602.07261, 2016
[29] Huster T, Deep learning for pedestrian detection in aerial imagery. *In MSS Passive Sensors*, 2016.
[30] Carlet J, Abayowa B. Fast vehicle detection in aerial imagery. *CoRR*, vol. Abs/1709.08666, 2017.
[31] Glorot X, Bengio Y. "Understanding the difficulty of training deep feedforward neural networks". *Journal of Machine Learning Research - Proceedings Track*, 9, 249-256, 2010.