

DERİN ÖĞRENME YÖNTEMLERİ KULLANARAK GERÇEK ZAMANLI ARAÇ TESPİTİ

Özet

İnsansız hava araçları, sağlamış olduğu hareketlilik ve yüksek irtifa sayesinde günümüzde; alan tespiti, trafik izleme ve trafik kontrol gibi birçok alanda artan bir kullanıma sahiptir. Bu alanda hedeflenen önemli işlerden birisi de alan fotoğrafları yardımıyla gerçek zamanlı araç tespiti ve araç sayımı olarak görülmektedir. Bu amaç doğrultusunda derin öğrenme, makine öğrenmesi ve birçok görüntü işleme tekniği ön plana çıkmaktadır. Bu çalışma kapsamında derin öğrenme mimarilerinden konvolüsyonel sinir ağları yapısı kullanılarak, gerçek zamanlı nesne tespit algoritması olan YOLO algoritması yardımıyla bir araç tespit uygulaması geliştirilmiştir. YOLO algoritmasının performansı, konvolüsyonel sinir ağları yapısı yardımıyla artırılmaya çalışılmıştır. Çalışma sonucunda YOLO'ya yardımcı olarak sunulan yapının farklı veri setleri kullanılarak gerçekleştirilen testlerinde, YOLO'nun başarısı %4,3 artırılmış ve 400x400 boyutlarındaki giriş değerlerinde 60 fps işlem değerine ulaşılabilirdiği görülmüştür. Gerçek zamanlı uygulamalarda araç tespiti için kullanılacak bir yapı ortaya konmuştur.

Anahtar Kelimeler: Araç tespiti, derin öğrenme, makine öğrenmesi.

REAL-TIME VEHICLE DETECTION BY USING DEEP LEARNING METHODS

Abstract

Thanks to unmanned aerial vehicles' mobility and high altitude; it has an increasing use in many areas such as area detection, traffic monitoring and traffic control in today. Real time vehicle detection and vehicle count are the one of the important works to be done by using unmanned aerial vehicles. For this purpose, deep learning, machine learning and many image processing techniques come to the fore. Within the scope of this study, a vehicle detection application has been developed by using the convolutional neural networks structure from deep learning architectures and with the help of real-time object detection algorithm, YOLO. YOLO algorithm performance is tried to be increased with the help of convolutional neural network structure. As a result of the study, the success of YOLO has been increased by %4,3 in the tests performed using different data sets and it has been observed that the input values of 400x400 can reach 60 fps transaction value. The structure that can be used for vehicle detection in real time applications has been introduced.

Key Words: Vehicle detection, deep learning, machine learning.

1. Giriş

Araç tespiti; trafik yoğunluk izleme, ulaşım analizi yapma, park yeri tespiti, askeri alanda hedef tespiti, otonom araç uygulaması geliştirme gibi birçok alanda ihtiyaç duyulan bir konudur. İnsansız hava araçları, sağladıkları yüksek çözünürlüklü fotoğraflar, gerçek zamanlı video aktarımları ve hareketlilik imkânlarıyla araç tespiti problemlerinde aktif olarak kullanılmaya başlanmıştır. Fakat performans ve ölçeklenebilirlik sorunlarından dolayı, drone kameraları ve diğer trafik kameralarından gelen görüntülerin gerçek zamanda işlenmesi hayata geçirilememektedir. Bu anlamda ortaya konan derin öğrenme, makine öğrenmesi yöntemleri ve görüntü işleme algoritmaları, araç tespit uygulamalarının daha hızlı ve daha doğru bir şekilde tanımlanabilmesini sağlamıştır.

Derin öğrenme yardımıyla geliştirilen araç tespit uygulamaları, genel olarak ilk aşamada girdi görüntülerinden konvolüsyonel özellikli bir özellik kümesinin ortaya konmasıyla başlar. Daha sonra bu küme yardımıyla sınıflandırıcı olarak tanımlanmış ağlar, özellik nesnelarini tanımak için kullanılır. Genel olarak bu ağ yapısı seçilen yöntemle bağlı olarak kayan bir pencere yapısında tüm resmi tarayarak veya resim üzerindeki seçilmiş bölgelerde çalıştırılarak tespit işlemini gerçekleştirir.

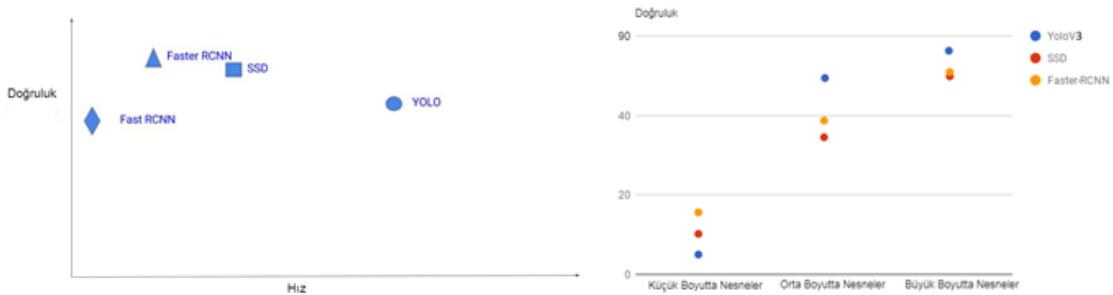
Çalışma kapsamında, literatürde araç tespiti üzerine odaklanan derin öğrenme ve görüntü işleme bazı yöntemler incelenmiştir. Derin öğrenme yöntemlerinin konvolüsyonel ağ temelli, görüntü işleme ve nesne tespiti yöntemlerinin ise YOLO (You Only Look Once) ve benzeri algoritmalar temelli oldukları gözlenmiştir.

CNN (convolutional neural network) temelli yapılarda; AlexNet, GoogleNet, Vgg16, Vgg19 ve ResNet50 algoritmaları ve Faster R-CNN yapısı önce çıkmaktadır. AlexNet (Krizhevsky ve ark. 2012) algoritması 25 katmandan oluşan, giriş katmanında 227x227x3 boyutlarındaki görüntüyü işleyebilen bir yeteneğe sahiptir. ImageNet veri tabanında doğruluk oranı %80 olarak gözlendiği belirtilmiştir. GoogleNet, AlexNet'e göre 12 kat daha fazla katman içeren bir yapıdır. (Szegedy ve ark. 2015) Aktivasyon için doğrusal aktivasyon kullanmaktadır. ImageNet üzerinde yapılan testte %93 doğruluk oranına ulaşmış olduğu gözlenmiştir. Vgg16 derin öğrenme algoritması ise 13 konvolüsyon 3 tam bağlı katmanından oluşan bir ağıdır. ImageNet veri tabanında %89 doğruluğa ulaşmıştır. (Chen ve ark. 2016; Simonyan ve Zisserman 2014) Vgg19 derin öğrenme algoritması, Vgg16 algoritmasının akabinde oluşturulan toplamda 47 katmandan oluşan bir yapıya sahiptir. Aynı veri tabanında %88 doğruluk oranı yakalamıştır. (Simonyan ve Zisserman 2014). ResNet50 mikro mimari modülü yapısına sahiptir, bünyesinde 177 katmandan oluşan bir ağ vardır. (Doğan, 2018) Giriş katmanı olarak 224x224x3 boyutunu kabul etmektedir. (He ve ark. 2016) CNN temelli bu yapılar, doğruluk oranı açısından yüksek değerlere sahip olsalar da tespit hızı açısından düşük değerlere sahiptirler. Bu anlamda Faster R-CNN algoritması, CNN yapısının yavaşlığını azaltmak amacıyla tasarlanmış bir ağ yapısıdır. CNN göre 250 kat daha hızlı çalışmaktadır. (İnik & Ülker, 2017) Hız ve doğruluk yüzdesi CNN temelli algoritmalarla göre daha yüksek bir değere (%91) sahip olsa da gerçek zamanlı uygulamalarda

gerçekleştirilen testlerde 7 fps değerleri gözlenmiştir. Bu da algoritmanın gerçek zamanlı uygulamalar için yeterli olmadığını göstermektedir. CNN temelli yapıların doğruluk olarak yüksek değerlere sahip olmalarına rağmen, gerçek zamanlı uygulamalar için tespit hızlarında yetersiz olduğu gözlenmektedir.

Nesne tespit ve görüntü işleme algoritmalarında ise, son zamanlarda, YOLO (You Only Look Once) ve SSD (Single Shot Detector) algoritmaları ön plana çıkmaktadır. SSD algoritması farklı katmanlardaki özellik haritalarından küçük nesnelere belirleyen bir algoritmadır. Faster R-CNN algoritmasına göre daha kötü başarımla sahip olmasına karşın, daha hızlı bir sonuç üretmektedir. YOLO algoritması (Redmon, Divvala, Girshick, & Farhadi, 2016) ise resmi tamamen konvolüsyonel olarak ele alınarak, daha hızlı işlemeyi hedeflemektedir. COCO veri tabanı üzerinden yapılan çalışmada (Lin & Girshick 2017) algoritmanın büyük objelerde %98 oranında başarımla gösterdiği, objeler küçüldükçe ise bu oranın %60'lara düştüğü ifade edilmiştir.

Derin öğrenme yöntemlerinden Faster R-CNN ile görüntü işleme ve nesne tespit yöntemlerinden SSD ve YOLO karşılaştırıldığında; Faster R-CNN'nin en yüksek başarımla sahip olduğu, fakat en düşük hızı gösterdiği gözlenmiştir. YOLO'nun ise büyük objelerde başarımla oranının yüksek olduğu, objeler küçüldükçe başarımla oranının düştüğü görülmüştür. Hız konusunda ise diğer algoritmalarından öndedir.



Şekil 1- Faster RCNN, SSD ve YOLOv3 Karşılaştırmaları

Drone görüntülerindeki araçların boyutlarının küçük olması, YOLO algoritmasının başarımla oranını etkilediğinden, bu anlamda tek başına kullanılması istenilen sonucu vermemektedir. Bu yüzden, drone görüntülerinden gerçek zamanlı araç tespiti için; YOLO algoritmasının sunduğu hıza yakın ve R-CNN derin öğrenme yönteminin başarımla oranına sahip bir yaklaşıma ihtiyaç duyulmaktadır. Bu anlamda gerçekleştirilen çalışmada, "Münih Veri Seti" (Liu, K., Mattyus, G., 2015) ile özel olarak oluşturulmuş; bünyesinde araba, kamyon, kamyonet, otobüs objelerini içeren, bir veri seti birleştirilerek araç tespiti için veri seti olarak kullanılmıştır. Çalışmada YOLOv3 algoritması ile Faster R-CNN derin öğrenme ağı kullanılmıştır. Oluşturulan gerçek zamanlı araç tespiti uygulaması yardımıyla, araçların hızlı ve başarımla oranı yüksek bir şekilde tespit edilmesi sağlanmıştır.

Çalışmanın geri kalanı şu şekilde planlanmıştır. 2. Bölümde ilk olarak araç tespiti ve sınıflandırma konusundaki literatürdeki çalışmalar ve kullanılan veri setlerinin özellikleri sunulmuştur. Sonrasında, kullanılan algoritma ve ağ yapısı paylaşılmıştır. Algoritma testleri gerçekleştirilerek diğer çalışmalar ile karşılaştırılmalı bir değerlendirme yapılmıştır. 3. Bölümde ise sonuçlar sunulurken gelecek çalışmalardan bahsedilmiştir.

2. Literatür Araştırması

Yapılan bu çalışma ile benzer olarak drone, uydu gibi sistemlerden gelen görüntüler veya fotoğraflar üzerinde birçok araç tespit ve sınıflandırma uygulaması mevcuttur. Bu çalışmalar incelendiğinde genel olarak çalışmaların donanım ve yazılım temelli olarak ayrıldığı gözlenmektedir. Bu çalışma kapsamında, çalışmayı doğrudan ilgilendiren yazılım alanındaki çalışmalar referans alınmıştır. Literatür araştırması

kapsamında araç tespiti, trafik düzenlemesi, makine öğrenmesi, derin öğrenme anahtar kelimeleri kullanılarak ilgili makaleler araştırılmıştır. Bu kapsamda öne çıkan çalışmalar aşağıda kısaca özetlenmiştir.

Chen (Chen, Xiang, Liu, & Pan, 2014) tarafından mevcut algoritmalarından farklı olan yeni bir model ve yaklaşım olan hibrit derin yapay sinir ağı (HDNN) ortaya konmuştur. Bu model kayan pencere ve derin CNN (convolutional neural network) yöntemini içerisinde barındırmaktadır. Ana fikir, konvolüsyonel katmanları farklı boyutlarda ayarlayarak farklı boyutlardaki araçları tespit etmektir. Çalışma içerisinde Google Earth yardımıyla San Francisco üzerinden toplanmış özel bir veri seti kullanılmıştır. Her ne kadar çalışma diğer çalışmalardan farklı bir yöntem ortaya koysa da işlem bakımından en güçlü işlemcilerde bile 7 saniye kadar süre almaktadır. Zamandan kaybettirdiği için çok tercih edilen bir yöntem olmadığı belirtilmiştir.

Ammour ve arkadaşları (Ammour et al., 2017) tarafından araç tespiti için iki aşamalı metod ortaya konmuştur. İlk aşamada bölge çıkarımı ve ortalama-kaydırma algoritması yardımıyla resim kısımlara bölünmüştür. İkinci aşamada ise araç tespiti için VGG16 modeli kullanılarak özellik çıkarımı yapılmıştır. SVM (support vector machine) yardımıyla özelliklerin araç mı, değil mi tespitine karar verilmiştir. Çalışma içerisinde Trento Üniversitesi, Fen Fakültesi Kampüsünden özel olarak elde edilmiş veri seti kullanılmıştır. Doğruluk bakımından birçok algoritmadan önde olsa da gerçek zamanlı kullanım için kullandığı zaman ve küçük objelerdeki yanıltıcı yaklaşımından dolayı uygun olmadığı düşünülmektedir.

Deng ve arkadaşları tarafından yapılan çalışmada (Deng, Sun, Zhou, Zhao, & Zou, 2017), R-CNN (region based) temelli özellik haritası üretmeyi hedefleyen bir yöntem ortaya konmuştur. CNN yapısının sunmuş olduğu yavaşlığın önüne geçmek istenmiştir. Çalışmada çiftlenmiş ağ yapısı (iki farklı CNN metodunun birleşimi) kullanılmıştır. Veri seti olarak “Münih Araç Veri Seti” tercih edilmiştir. Doğruluk oranı ve hız bakımından benzer çalışmalara göre daha iyi sonuçlar verse de küçük objelerde hatalı sonuçlar üretmektedir. CNN yapısını içerdiğinden gerçek zamanlı uygulamalar için daha yavaş bir tespit mekanizmasına sahip olduğu belirtilmiştir.

Başka bir çalışmada (Ren, He, Girshick, & Sun, 2017), CNN ağ yapısının çalışma süresini kısaltmak hedeflenmiştir. Özellik çıkarımı için VGG16 modeli kullanılmış, ağ yapısı bu model ile beslenerek genişletilmiştir. Yapılan çalışma sonucunda CNN yapısının hızlandırıldığı ifade edilmiştir. Fakat küçük objelerde ortaya çıkan hatalı tespit önüne geçilemediği belirtilmiştir. Çalışmada drone resimleri yerine yanal resimler kullanılmış olup performans değerleri 5-17 fps aralığında ölçülmüştür.

Yu ve arkadaşları tarafından yapılan başka bir çalışmada (Yu, Westfechtel, Hamada, Ohno, & Tadokoro, 2017) Fast R-CNN ve Faster R-CNN yöntemleri kullanılmıştır. Bu ağ yapıları için uygulama alanı ortaya konmuş, küçük objeler için optimize bir yaklaşım geliştirmek istenmiştir. Vedai ve Münih Araç Veri Seti kullanılmıştır. Çalışmada başarısı kanıtlanmış diğer algoritmalara yer verilmemiştir.

Farklı bir çalışmada (Kyrkou, Timotheou, Kolios, Theocharides, & Panayiotou, 2018) ise trafik izleme üzerine yoğunlaşarak CNN algoritması kullanıp, yol ve araç çıkarımı yapılmıştır. Yol çıkarımı için maskeleme işlemi kullanılmış olup, CNN algoritması karar verme için tercih edilmiştir. Çalışma içerisinde Kıbrıs içerisinde çekilmiş drone ve uydu fotoğraflarının kullanıldığı belirtilmiştir. Çalışmanın başarı oranının yüksek, maskeleme tekniği ile farklı yöntemleri bir araya getiren bir yapısı olduğu gözlenmektedir.

Liu ve arkadaşları tarafından gerçekleştirilen çalışmada (Liu, Yang, & Li, 2018), kızılötesi görüntüler referans alınarak araç tespiti gerçekleştirilmiştir. Veri seti olarak “NPU_CS_UAV_IR_DATA” kullanılmıştır. CNN temelli bir tespit uygulaması geliştirilmiş, kızılötesi fotoğraflar gerçek zamanlı olarak çekilerek sisteme gönderilmiştir. Sistem üzerinden önceden test ve doğrulama adımları gerçekleştirilmiş olan yazılım koşarak CNN temelli bir tespit işlemi gerçekleştirilmiştir. Siyah-beyaz fotoğraflar üzerinden başarı oranı yüksek tespitlerin gerçekleştirildiği aktarılmıştır.

Yang ve arkadaşları tarafından gerçekleştirilen bir çalışmada ise (Yang, Liao, Li, Cao, & Rosenhahn, 2019), Double Focal Lose CNN adı verilen bir ağ yapısı geliştirilmiştir. Veri seti olarak ITCVD veri seti tercih edilmiştir. Temel olarak CNN kullanan ağ yapısı yardımıyla tespit hatasını azaltmak hedeflenmiştir. İşlem süresinin uzun olması sebebiyle gerçek zamanlı uygulamalar için uygun bir yapı sunulamadığı görülmektedir.

Literatürdeki çalışmalardan gözlemlendiği üzere genel olarak uygulanan yaklaşım; nesne tespit doğruluğu arttırmak ve tespit süresini kısaltmak üzerinedir. Sunulan yöntemler doğruluk oranı olarak düzgün sonuçlar verse de gerçek zamanlı uygulamalar için, süre bakımından geride kalmaktadır. Bu anlamda, süre

bakımından gerçek zamanlı işlemeye uygun, doğruluk payı yüksek bir yaklaşıma ihtiyaç duyulduğu gözlenmiştir.

3. Materyal ve Metot

Yapılan çalışmalardan da görüldüğü üzere, derin öğrenme ve nesne tespiti alanındaki makalelerin, iki temel yapı üzerinde yoğunlaştığı görülmektedir. Bu yapılardan ilki konvolüsyonel sinir ağı ve türevlerini içerirken diğeri YOLO (You Only Look Once) adı verilen yapıyı içermektedir.

Literatür taramasında kullanılan yöntemler haricinde, çalışmaya katkısı olabilecek veri setleri de ayrıca incelenmiştir. Bu anlamda literatürde sıkça kullanılan veri setleri aşağıdaki tabloda verilmiştir;

Tablo 3.1- Literatür Veri Setleri Özellikleri

<i>Veri Seti</i>	<i>Kaynak</i>	<i>Farklı Aç</i>	<i>Çözünürlük</i>	<i>Format</i>	<i>Araç Sayısı</i>
OIRDS	Uydu	Mevcut	Düşük	Sınırlayıcı Kutu	180
VEDAI	Uydu	Mevcut	Düşük	Sınırlayıcı Kutu	295
COWC	İHA	Mevcut	Düşük	Araç Merkezli Seçim	32.716
PUCPR	Kamera	Mevcut	Yüksek	Sınırlayıcı Kutu	192.216
CARPK	Drone	Mevcut Değil	Yüksek	Sınırlayıcı Kutu	89.777
Münih	Uydu	Mevcut	Yüksek	Sınırlayıcı Kutu	-

Bu çalışmada, literatürde başarı oranı ve çalışma hızıyla öne çıkan CNN temelli Faster R-CNN derin öğrenme algoritması ve YOLO görüntü işleme algoritmasının 4 versiyonundan (v1,v2,v3,v4 (Mahto, Garg, Seth, & Panda, 2020)) v3 versiyonu referans alınmıştır.

Faster R-CNN, hızlı bölgesel temelli konvolüsyonel sinir ağı olup, benzer bölgeleri birleştirerek işlem yapan bir yapıya sahiptir. Faster R-CNN de amaç seçici bir arama yöntemiyle belirli sayıda bir bölge oluşturup, tüm resim üzerinden arama yapmak yerine bu bölgeler üzerinden arama yapıp, doğru objeyi bulmaktır.

YOLO algoritması ise bölgesel temelli yaklaşım yerine, resmi tamamen konvolüsyonel olarak ele alarak daha hızlı olarak işleyerek gerçek zamanda işlemeye uygun bir yapı sunmaya yöneliktir. Resim boyutuna bağlı olarak $m \times m$ ’lik bir ızgaraya bölünerek, benzerliklerine göre ayırım gerçekleştirilir.

Bu çalışma kapsamında araç tespitinin daha hızlı yapılması ve tespit doğruluk oranının artırılması hedeflenmiştir. Bu doğrultuda tespit için YOLO algoritması tercih edilirken, doğruluğu arttırmak adına hızlı bölgesel konvolüsyonel temelli bir ağ YOLO algoritmasının öğrenme evresinde yardımcı olarak sunulmuştur. Veri seti olarak Tablo 3.1 ‘de bulunan veri setlerinden, bünyesinde ve yüksek çözünürlüklü, 100m üzerinden farklı araç tiplerinin görüntülerini içermesi sebebiyle Münih Araç Veri Seti kullanılmış olup, “Google Earth” uygulaması ve DJI drone görüntülerinden elde edilen veriler de veri setini genişletmek amacıyla sete dâhil edilmiştir.

3.1 Kullanılan Yöntem

Kullanılan yöntemin ortaya koyulmasında ilk olarak hızlı bölgesel konvolüsyonel sinir ağının tanımı gerçekleştirilmiştir. Konvolüsyonel sinir ağları, içerisinde, giriş görüntüsünün özelliklerini çıkaran konvolüsyonel operatöre sahiptir. Bu operatör, giriş verisinin görüntü özelliklerini öğrenerek pikseller arasındaki ilişkiyi korur. Temel yapısı; konvolüsyonel operatör, ReLU (Rectified Linear Units), alt örnekleme ve tam bağlantılı katman olmak üzere 4 yapıdan oluşmaktadır. Her konvolüsyonel işlemden sonra, ReLU aktivasyon fonksiyonu kullanılarak her piksel için negatif değerler taşıyan alanlar sıfır değeri ile değiştirilir. Alt örnekleme katmanı ise her bir özellik haritasının boyutunu küçülterek önemli olan

bilgileri diğer katmana aktarır. Tam bağlantılı katman, giriş görüntüsü ile çıkarılan görüntüler arasında bağlantı kurarak, eğitim veri setine dayalı giriş görüntüsünü sınıflandırır.

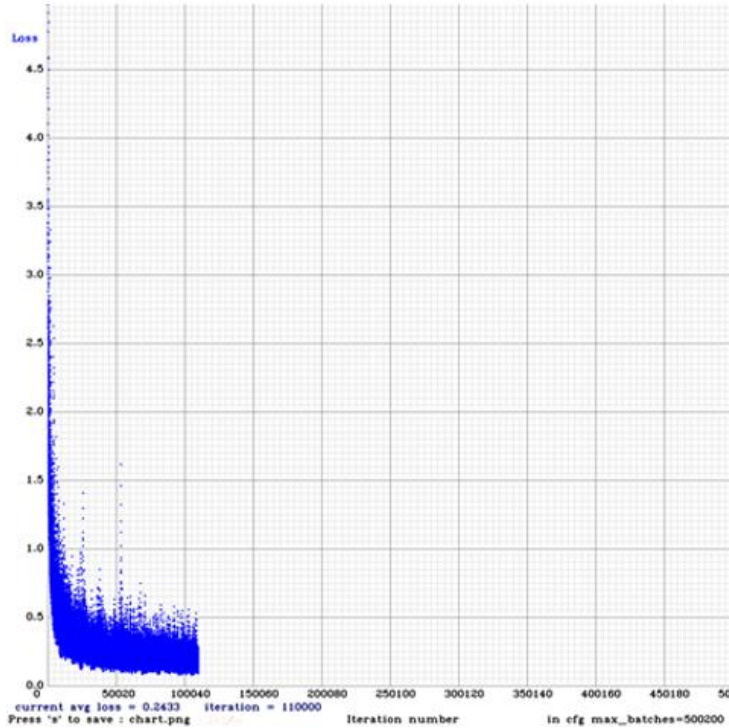
Konvolüsyonel sinir ağı tanım yapılırken, eldeki veri setlerinden küçük araç nesnelere seçilerek asıl nesne tespitini gerçekleştirecek algoritmanın düzgün beslenmesi hedeflenmiştir. Ayrıca yapılan tanımlamanın algoritma performansını karşılaştırmak amacıyla da referans olması planlanmıştır. Materyal ve metod başlığı altında belirtilen veri setleri algoritmaya uygun halde etiketlenmiştir. Eldeki veriler %70 öğrenme %30 test olacak şekilde, hazırlanan bir C# programlama dili ile yazılmış masaüstü uygulaması yardımıyla rastgele olarak bölümlenmiştir. Python programlama dili ile geliştirilen kod yardımıyla, Faster R-CNN ağı eğitilerek YOLO'yu besleyecek şekilde hazırlanmıştır. Sonrasında YOLO algoritması Windows üzerinde yine tespit için python programlama dili üzerinde yazılmış bir kod ile çalıştırılarak, sonuçlar ortaya konmuştur.

4. Deneysel Sonuçlar ve Bulgular

YOLO, konvolüsyonel sinir ağları ve türevlerine göre performans olarak daha hızlı cevap verebilen bir algoritmadır. Çalışma kapsamında gerçek zamanlı tespitte en yakın performans sergilenmek istendiğinden, YOLO algoritması seçilmiştir.

YOLO kendi yapısında kullanıcıya örnek ağırlık matrisleri ve yapılandırma dosyaları sunmaktadır. Ağırlık matrisleri algoritmanın öğrenme sonucunda çıkardığı yapılarıdır. Kullanıcılar basit tespit işlemlerini yapabilmek ve YOLO'nun öğrenme sürecini kısaltmak amacıyla örnek ağırlık matrislerini paylaşmaktadırlar. Ağırlık matrisleri yapılandırma dosyalarının test ve doğrulama sürecinden ortaya çıkmaktadır. YOLO'nun iyi bir öğrenme sürecinin olabilmesi için yapılandırma dosyalarının güzel kurgulanmış olması gerekmektedir.

Çalışma kapsamında araç setleri etiketleme işlemi hızlı bölgesel konvolüsyonel ve YOLO algoritması için ayrı ayrı yapılmıştır. Etiketleme ve öğrenme işlemleri sonucunda toplamda 110000 iterasyon yapılarak, 7 gün 12 saat 23 dakika sonucunda ortalama kayıp 0,243 civarına düşürülmüştür. Bundan sonrasındaki seyir 0,3 ile 0,14 arasında salınım yaptığından öğrenme aşaması durdurulmuştur. Öğrenme eğrisi Şekil 4.1 'YOLO Öğrenme Eğrisi' altında verilmiştir.



Şekil 4. 1- YOLO Öğrenme Eğrisi

YOLO algoritmasının küçük obje tespitleri konusunda çok başarılı olmadığı bilinmektedir. Bu anlamda algoritmanın küçük obje tespitlerini yapabilmesi adına öğrenme aşamasında CNN algoritmasının özellik çıkarımı kısmı kullanılarak YOLO 'ya referans olarak gösterilmiştir. Bu anlamda özellikle 100 m ve üzeri fotoğraflarda başarılı tespit oranı arttırılmaya çalışılmıştır. Fakat özellikle gölgenin düştüğü alanlarda, arazi rengiyle araç renginin yakın olduğu anlarda, algoritmanın başarılı tespit oranının düştüğü gözlenmiştir.

Faster R-CNN ve YOLO algoritmalarının her ikisi de GeForce 1060 GPU Ekran kartı ve 16GB Ram'e sahip bilgisayar üzerinde koşulmuştur. Algoritmaların karşılaştırılabilmesi için kesinlik (precision), duyarlılık (recall), F1-Skoru (F1-Score) ve kalite (quality) metrikleri kullanılmıştır.

Kesinlik metriği (1); pozitif olarak tahminlediğimiz değerlerin gerçekten kaç adedinin pozitif olduğunu göstermek için kullanılmıştır.

Duyarlılık metriği (2); pozitif olarak tahmin etmemiz gereken işlemlerin ne kadarını pozitif olarak tahmin ettiğimizi göstermek için kullanılmıştır.

F1-Skoru metriği (3); bize kesinlik ve hassaslık değerlerinin harmonik ortalamasını göstermektedir. Harmonik ortalama tercih edilmesinin sebebi, uç durumları da işleme almak içindir.

Kalite metriği (4); algoritmanın yeni bir girdi de göstereceği muhtemel başarı oranını ifade etmektedir.

Metrikler tanımları şu şekildedir;

$$Kesinlik = \frac{DP}{DP + YP} \quad (1)$$

$$Duyarlılık = \frac{DP}{DP + YN} \quad (2)$$

$$F1 Skoru = \frac{2 * Kesinlik * Hassaslık}{Kesinlik + Hassaslık} \quad (3)$$

$$Kalite = \frac{DP}{DP + YP + YN} \quad (4)$$

DP: Doğru Pozitif (Gerçekte pozitif olup modelin pozitif tahmin ettiği objeler)

YP: Yanlış Pozitif (Gerçekte pozitif olmayıp modelin pozitif tahmin ettiği objeler)

YN: Yanlış Negatif (Gerçekte pozitif olup modelin negatif tahmin ettiği objeler)

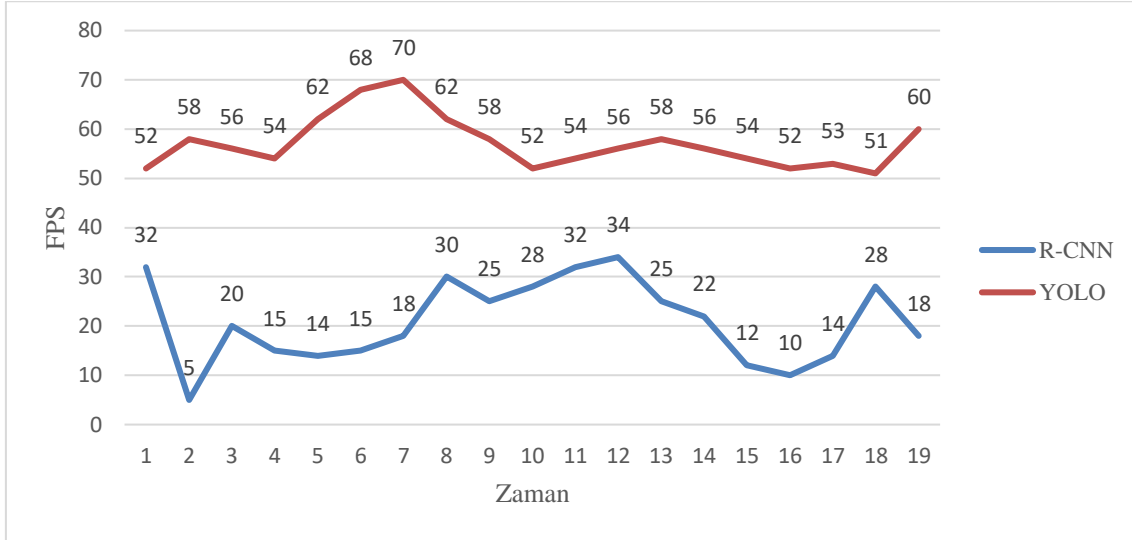
Metrikler doğrultusunda ilk olarak Faster R-CNN ve YOLO algoritması karşılaştırılmıştır. Karşılaştırma için eldeki verilerden 50m ve 100m'den çekilmiş fotoğraflar seçilerek 500 adet fotoğrafı içeren bir küme kullanılmıştır. Bu küme üzerinden yapılan karşılaştırma "Tablo 4.1 – Algoritma Karşılaştırmaları" içerisinde paylaşılmıştır.

Tablo 4.1- Algoritmaların Performanslarının Karşılaştırması

Metot	DP	YP	YN	Kesinlik (1)	Duyarlılık (2)	F1 Skoru (3)	Kalite (4)
Faster R-CNN	390	26	44	%93,75	%89,86	0.9176	%84,78
YOLO	458	18	24	%96,21	%95	0.9561	%91,6

Testler doğrultusunda YOLO algoritmasının skor olarak daha başarılı olduğu gözlenmiştir. Fotoğraf girdisinin yanısıra, algoritmalar aynı mesafeleri içeren videolar üzerinden de karşılaştırılmıştır. Faster R-CNN uygulamasının gerçek zamanlı video işlemesi yapıldığında ortalama 20.89 FPS (frame per second)

aralığında sonuç sağladığı gözlenmiştir. YOLO ise tek bir seferde sinir ağından geçirerek resimdeki tüm nesnelerin sınıfını ve koordinatlarını tahmin edebildiğinden fps değerleri daha yükseklerde seyretmiştir. YOLO algoritmasında alınan sonuç ortalama 57.16 FPS olmuştur. Aşağıdaki şekilde algoritmaların aynı video üzerinde yapılan FPS karşılaştırmaları gözlenmektedir.



Şekil 4.2- FPS Karşılaştırması

YOLO algoritmasının küçük objeleri tespiti konusundaki başarısını arttırmak adına, Faster R-CNN ağı ile referans sınıflandırma beslemesi sağlanmıştır. Bu kapsamda YOLO algoritmasının Faster R-CNN ile beslenmiş ve beslenmemiş durumları karşılaştırılarak kurulan yapının getirisi ortaya konmuştur. Algoritma testini gerçekleştirmek adına 5 farklı tipte (tek araç içeren, araca benzer yapı içeren, karmaşık ve küçük araç yapıları barındıran vb.) 50m, 75m, 100m ve 100m üzeri farklı yüksekliklerden çekilmiş fotoğraflarla testler gerçekleştirilmiştir. Elde edilen sonuçlar aşağıdaki tablo 4.2’de metre bazında gösterilmiştir.

Tablo 4.2- YOLO ve Faster R-CNN Beslemeli YOLO algoritmalarının karşılaştırması

Mesafe	Test Edilen Fotoğraf Sayısı	Algoritma	Toplam Araç Sayısı	DP	YN	YP	Kesinlik	Hassaslık	F1 Skoru	Kalite
50m	50	YOLO	100	100	0	0	1	1	1	1
	50	Faster R-CNN Beslemeli YOLO	100	100	0	0	1	1	1	1
75m	50	YOLO	100	78	16	6	0,929	0,830	0,876	0,780
	50	Faster R-CNN Beslemeli YOLO	100	82	10	4	0,954	0,891	0,921	0,854
100m	50	YOLO	360	290	40	30	0,906	0,879	0,892	0,805
	50	Faster R-CNN Beslemeli YOLO	360	320	25	15	0,955	0,928	0,941	0,889
100m+	50	YOLO	1060	840	140	80	0,913	0,857	0,884	0,792
	50	Faster R-CNN Beslemeli YOLO	1060	880	120	60	0,936	0,880	0,9075	0,830

Çalışma literatürdeki benzer çalışmalarla (Zhong, Lei, & Yao, 2017) resimler yeniden boyutlandırılarak (702×468) karşılaştırıldığında, tanımlanan YOLO algoritmasının başarı oranının tek başına Faster R-CNN(Z&F) algoritmasının gerisinde kaldığı, Faster R-CNN beslemeli YOLO algoritmasının ise Faster R-CNN(Z&F) algoritmasının başarı oranının üzerinde olmasına rağmen Faster R-CNN(VGG-16), Cascaded CNN algoritmasının altında kaldığı gözlenmiştir. Her iki algoritmanın da hız olarak diğerlerinden daha yüksek değerleri sahip olduğu gözlenmiştir. Bu kapsamda geliştirilen algoritmanın doğruluk oranı olarak iyileştirme sağladığı ortaya konmuştur. Münih veri seti üzerinden yapılan karşılaştırma sonuçları “Tablo 4.3- Farklı Nesne Tespit Modellerinin Münih Veri Seti Üzerinden Karşılaştırmaları” altında verilmiştir.

Tablo 4.3-Farklı Nesne Tespit Modellerinin Münih Veri Seti Üzerinden Karşılaştırmaları

<i>Nesne Tespit Modeli</i>	<i>Hassaslık</i>	<i>F1-Skoru</i>	<i>FPS</i>
Faster R-CNN(Z&F)	66.8%	0.657	5,2
Faster R-CNN(VGG-16)	78.3%	0.779	4,9
Cascaded CNN	80.3%	0.782	3,2
Yolov3	58.2%	0.561	28
Faster R-CNN Beslemeli YOLO	70.1%	0.694	10

Çalışma sonucunda YOLO'ya yardımcı olarak sunulan yapının farklı veri setleri kullanılarak gerçekleştirilen testlerinde, YOLO algoritmasının tespit oranını %4,3 arttırdığı ve 400x400 giriş değerlerinde 60fps değerine ulaşabildiği görülmüştür. Çalışma kapsamında gerçek zamanlı uygulamalarda araç tespiti için kullanılacak bir yapı ortaya konmuştur.

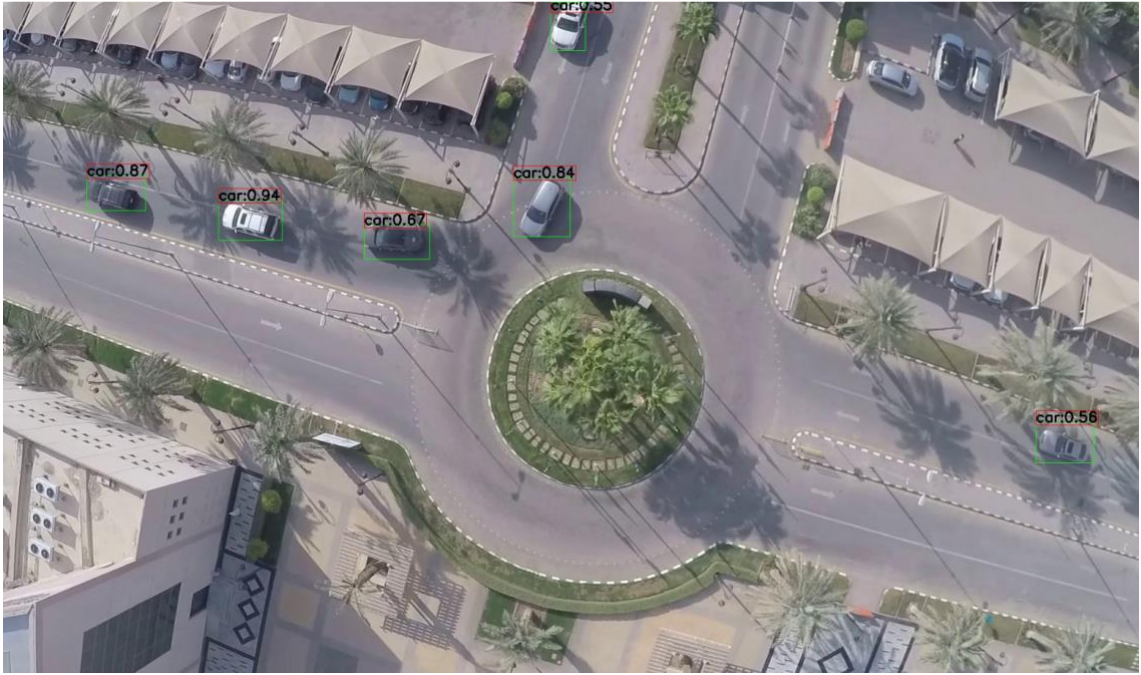
Algoritmanın tespit ettiği farklı yüksekliklerden elde edilmiş örnek sonuçlar ise Şekil 4.3, 4.4, 4.5, 4.6 'da gösterilmiştir.



Şekil 4.3- 50m yükseklikten çekilmiş drone görüntüsü üzerinden Faster R-CNN Beslemeli YOLO algoritmasının araç tespit başarısı



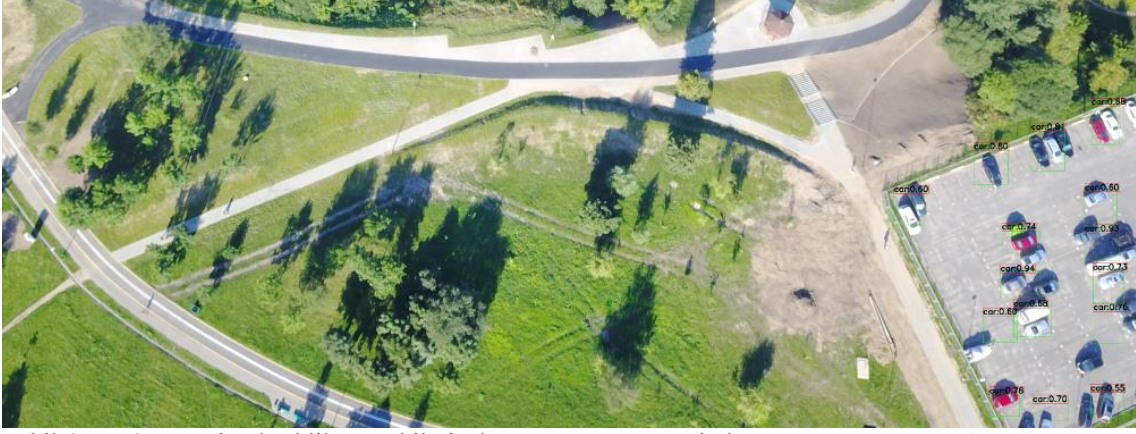
Şekil 4.4- 75m yükseklikten çekilmiş drone görüntüsü üzerinden Faster R-CNN Beslemeli YOLO algoritmasının araç tespit başarısı



Şekil 4.5 -100m yükseklikten çekilmiş drone görüntüsü üzerinden Faster R-CNN Beslemeli YOLO algoritmasının araç tespit başarısı



Şekil 4.6 100m üzeri (Yaklaşık 150m) yükseklikten çekilmiş drone görüntüsü üzerinden Faster R-CNN Beslemeli YOLO algoritmasının araç tespit başarısı



Şekil 4.7 150m üzeri yükseklikten çekilmiş drone görüntüsü üzerinden Faster R-CNN Beslemeli YOLO algoritmasının araç tespit başarısı

5. Değerlendirme ve Sonuç

Bu çalışma kapsamında gerçek zamanlı uygulamalarda kullanılabilecek algoritma hızına sahip, doğruluk oranı yüksek bir araç tespit uygulaması geliştirilmek hedeflenmiştir. Bu kapsamda, ilk olarak; araç tespitini gerçekleştirebilmek adına “Münih Veri Seti” (Liu, K., Mattyus, G., 2015) ile, özel olarak oluşturulmuş; bünyesinde araba, kamyon, kamyonet, otobüs objelerini içeren, bir veri seti birleştirilerek genişletilmiş bir veri havuzu oluşturulmuştur. Böylece derin öğrenme ve makine öğrenmesi algoritmalarında kullanılacak test ve öğrenim setleri ortaya konmuştur. Veri setinin %70’i öğrenmeye %30’u ise teste ayrılmıştır.

Literatürdeki çalışmalar analiz edilerek, öne çıkan derin öğrenme algoritmaları Faster R-CNN ve YOLO olarak belirlenmiştir. İlk olarak eldeki veri seti etiketlenerek Faster R-CNN algoritmasına uygun hale getirilmiştir. Sonrasında algoritma araç tespiti amacıyla denenmiştir. YOLO algoritması için eldeki veri seti aynı şekilde etiketlenerek algoritmanın testi gerçekleştirilmiştir. Karşılaştırma için eldeki verilerden 50m ve 100m’den çekilmiş fotoğraflar seçilerek 500 adet fotoğrafı içeren bir küme kullanılmıştır. Testler doğrultusunda YOLO algoritmasının başarı oranının %95,61 olduğu, Faster R-CNN algoritmasının ise %91,76 başarı oranına ulaştığı gözlenmiştir. Video üzerindeki testlerde Faster R-CNN uygulaması ortalama 20.89 FPS (frame per second) aralığında seyretmiştir. YOLO ise tek bir seferde girdileri sinir ağından geçirerek resimdeki tüm nesnelere sınıfları ve koordinatlarını tahmin edebildiğinden fps değerleri daha yükseklerde (ortalama 57.16 FPS) sonuç vermiştir. Bu sonuçlar doğrultusunda YOLO algoritması seçilerek, algoritmanın küçük araçlarda verdiği yanlış sonuçların giderilmesi hedeflenmiştir.

Bu doğrultuda YOLO algoritması basit bir Faster R-CNN sinir ağı yardımıyla beslenerek geliştirilmiştir. Bu yardımcı yapı ağı 100m ve 100m üzeri farklı yüksekliklerden çekilmiş fotoğraflardaki araçlarla eğitilmiştir. YOLO’ya girdi olarak tespit edebildiği sınıflandırma verisini sunmuştur. Sonrasında iyileştirilen bu ağ yapısı önceden denenilen YOLO algoritması ile karşılaştırılmıştır. Karşılaştırma için 50m, 75m, 100m ve 100m üzeri farklı yüksekliklerden çekilmiş araç fotoğrafları girdi olarak verilmiştir.

Çalışma sonucunda YOLO’ya yardımcı olarak sunulan yapının farklı veri setleri kullanılarak gerçekleştirilen testlerinde, YOLO’nun tespit oranını %4,3 arttırdığı ve 400x400 giriş değerlerinde ise 60fps değerine ulaşabildiği görülmüştür. Girdi boyutu büyüdükçe, geliştirilen algoritma performansı ve doğruluk oranının düştüğü gözlenmiştir. Literatür üzerinde yapılan karşılaştırmada hız olarak en yüksek tepki değerine sahip olmasına rağmen, algoritma başarı oranı olarak Faster R-CNN(VGG-16) ve Cascaded CNN algoritmalarının gerisinde kalmıştır. Tespit hızını arttırmanın doğruluk payını düşürdüğü gözlenmiştir.

Yapılan çalışmanın ulaştığı fps değerleri ve 50m – 150m aralığında ortaya koyduğu doğruluk oranlarıyla, trafik izleme, hedef tespiti, araç sayımı, park uygulamaları gibi birçok alanda kullanılabileceği düşünülmektedir.

Bu çalışmada henüz şartlar oluşmadığı için gerek zaman gerekse elde edilen veri setlerinden kaynaklanan bazı yapılamayan çalışmalar; algoritmanın hata oranının düşürülmesi, daha büyük boyutlu girdilere daha doğru sonuçlar üretebilmesi ve araç çeşitliliğinin artırılarak algoritmanın daha fazla aracı tanıyabilmesinin sağlanması, hata oranını düşürmek amacıyla farklı ağ yapılarının küçük objelerdeki performansının

değerlendirilmesi, YOLO'ya bütünleştirilmesinin sağlanması ve öğrenme ve test aşamalarında ise, eldeki veri setinin daha da genişletilerek, 5 fold cross validation ile de seçilimi daha da rastgele bir hale getirerek daha doğru sonuçlara ulaşılabilirliği söylenebilir.

KAYNAKLAR

Ammour, N., Alhichri, H., Bazi, Y., Benjdira, B., Alajlan, N., & Zuair, M. (2017). Deep learning approach for car detection in UAV imagery. *Remote Sensing*. <https://doi.org/10.3390/rs9040312>

Chen, L. C., Papandreou, G., Kokkinos, I., Murphy, K., & Yuille, A. L. (2016). Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *arXiv preprint arXiv:1606.00915*.

Chen, X., Xiang, S., Liu, C. L., & Pan, C. H. (2014). Vehicle detection in satellite images by hybrid deep convolutional neural networks. *IEEE Geoscience and Remote Sensing Letters*. <https://doi.org/10.1109/LGRS.2014.2309695>

Deng, Z., Sun, H., Zhou, S., Zhao, J., & Zou, H. (2017). Toward Fast and Accurate Vehicle Detection in Aerial Images Using Coupled Region-Based Convolutional Neural Networks. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 10(8), 3652–3664. <https://doi.org/10.1109/JSTARS.2017.2694890>

DOĞAN, F. (2018). Öğrenme Algoritmalarının Yaprak Sınıflandırma Başarımlarının Karşılaştırılması. *SAKARYA UNIVERSITY JOURNAL OF COMPUTER AND INFORMATION SCIENCES*.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).

Ioffe, S., & Szegedy, C. (2015, June). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning* (pp. 448-456).

İnik, Ö., & Ülker, E. (2017). Derin Öğrenme ve Görüntü Analizinde Kullanılan Derin Öğrenme Modelleri. *Gaziosmanpaşa Journal of Scientific Research*.

Kyrkou, C., Timotheou, S., Kolios, P., Theocharides, T., & Panayiotou, C. G. (2018). Optimized vision-directed deployment of UAVs for rapid traffic monitoring. In *2018 IEEE International Conference on Consumer Electronics, ICCE 2018* (Vol. 2018-Janua, pp. 1–6). <https://doi.org/10.1109/ICCE.2018.8326145>

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097-1105).

Liu, K.; Mattyus, G. DLR 3k Munich Vehicle Aerial Image Dataset. Available online: http://pba-freesoftware.eoc.dlr.de/3K_VehicleDetection_dataset.zip (accessed on 31 December 2015)

Liu, X., Yang, T., & Li, J. (2018). Real-time ground vehicle detection in aerial infrared imagery based on convolutional neural network. *Electronics* (Switzerland), 7(6). <https://doi.org/10.3390/electronics7060078>

Mahto, P., Garg, P., Seth, P., & Panda, J. (2020). Refining Yolov4 for vehicle detection. *International Journal of Advanced Research in Engineering and Technology*. <https://doi.org/10.34218/IJARET.11.5.2020.043>

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. <https://doi.org/10.1109/CVPR.2016.91>

Ren, S., He, K., Girshick, R., & Sun, J. (2017). Faster R-CNN: Towards Real-Time Object Detection with. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6), 1137–1149. <https://doi.org/10.1109/TPAMI.2016.2577031>

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Sommer, L. W., Schuchert, T., & Beyerer, J. (2017). Fast deep vehicle detection in aerial images. In *Proceedings - 2017 IEEE Winter Conference on Applications of Computer Vision, WACV 2017* (pp. 311–319). <https://doi.org/10.1109/WACV.2017.41>

T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar. Focal loss for dense object detection. *arXiv preprint arXiv:1708.02002*, 2017. 1, 3, 4.

Yang, M. Y., Liao, W., Li, X., Cao, Y., & Rosenhahn, B. (2019). Vehicle detection in aerial images. *Photogrammetric Engineering and Remote Sensing*, 85(4), 297–304. <https://doi.org/10.14358/PERS.85.4.297>

Zhong, J., Lei, T., & Yao, G. (2017). Robust vehicle detection in aerial images based on cascaded convolutional neural networks. *Sensors (Switzerland)*. <https://doi.org/10.3390/s17122720>