Evrişimsel Sinir Ağları Kullanılarak Araç, Yaya ve Trafik İşaretlerinin Algılanması

Recognition of Vehicles, Pedestrians and Traffic Signs Using Convolutional Neural Networks

Gülyeter ÖZTÜRK
Mekatronik Mühendisliği
Sakarya Uygulamalı Bilimler Üniversitesi
Sakarya, Türkiye
gulyeterozturk@subu.edu.tr

Raşit KÖKER
Elektrik ve Elektronik Mühendisliği
Sakarya Uygulamalı Bilimler Üniversitesi
Sakarya, Türkiye
rkoker@subu.edu.tr

Osman ELDOĞAN, Durmuş KARAYEL
Mekatronik Mühendisliği
Sakarya Uygulamalı Bilimler Üniversitesi
Sakarya, Türkiye
eldoğan@subu.edu.tr, dkarayel@subu.edu.tr

Özetçe*—Robotik dünyasının ve otomasyonun bugünü ve geleceği olan otonom arabalar ile otonom arabaların koordineli bir şekilde çalıştığı gelişmiş sürücü destek sistemleri (ADAS), sürüş ortamı bilgisi yoluyla sürücülere fayda sağlayabilecek önemli teknolojilerdir. Otonom arabaların temel unsurlarından bazıları; çevreyi, engelleri, yayaları, trafik işaretlerini ve diğer araçları algılamaktır. Bu çalışmada makine öğrenmesi konularındaki problemlerin çözümünde son yıllarda büyük doğruluk oranı ve hız ile kendinden çokça söz ettiren derin öğrenme kullanılarak otonom arabanın işlevlerinden biri olan çevredeki nesneleri algılama uygulaması yapılmıştır. Araçtan çekilen video görüntülerinden çeşitli ortamlarda, farklı görüş açılarında ve boyutlarda olan işaretler ve nesneler algılanabilmektedir. Yaya, araba, bisiklet ile 7 trafik işaretinden oluşan 10 nesneye ait 517 görüntü, derin öğrenmenin temel mimarisi kabul edilen evrişimsel sinir ağları modellerinden SSD Inception V2, Faster R-CNN Inception V2, Faster R-CNN Resnet 50 ve Faster R-CNN Resnet 101 kullanılarak nesne algılama çalışmaları gerçekleştirilmiştir. COCO veri seti üzerinden önceden eğitilmiş olan bu modeller transfer öğrenimi yöntemi ile bir kısmı GRAZ-01 ve GRAZ-02 veri setlerindeki görüntülerden bir kısmı da cep telefonu kamerası kullanılarak elde edilen görüntülerden oluşturulan yeni veri seti üzerinde tekrardan eğitilerek değerlendirilmiştir. Performans analizleri sonucunda, Faster R-CNN Resnet 101 modelinin hem görüntü hem de video üzerinde nesne algılama çalışmalarında %85.1 doğruluk oranıyla başarı gösterdiği gözlemlenmiştir.

Anahtar Kelime—Derin öğrenme, Evrişimsel sinir ağları, Nesne algılama, Transfer öğrenimi

Abstract*—As the present and future of the robotic world and automation, autonomous vehicles and Advanced Driver Assistance Systems (ADAS) that work in conjunction with autonomous vehicles are important technologies that can benefit drivers through current driving environments. Some elementary factors of these autonomous cars are recognizing surroundings, barriers, pedestrians, traffic signs and other vehicles. In this study, as one of the functions of an autonomous car, the operation of peripheral object recognition is carried out through the use of deep learning which has been mentioned with great accuracy and speed these years in the field of solving problems in machine learning. Signs and objects in various environments, different viewing angles and dimensions can be recognized through the video images taken from the vehicle. Application of object recognition is achieved through the use of 517 images of 10 objects consisting of pedestrians, cars, bicycles and 7 traffic signs, and of convolutional neural networks models including SSD Inception V2, Faster R-CNN Inception V2, Faster R-CNN Resnet 50 and Faster R-CNN Resnet 101, which are known as the basis of deep learning. The models previously trained on the COCO data set are retrained and evaluated on the new data set with the transfer learning method. The new data set is formed by part of the image from the GRAZ-01 and GRAZ-02 data sets and part of the image from the mobile phone camera. As a result of performance analyzes, Faster R-CNN Resnet 101 model is found to be successful in object detection on both images and videos with 85.1% accuracy.

Keywords—Deep learning, Convolutional neural networks, Object recognition, Transfer learning

I. GİRİŞ

Günümüzde araba kazaları dünya çapında artmakta ve ayrıca trafik kurallarına uymada sürücü ihmali de artmaktadır. Bu tür olaylar sürücüsüz arabalar geliştirilerek kontrol edilebilir. Otonom araba, sürücünün yardımı olmadan veya herhangi bir insan müdahalesi olmadan dış çevreyi tek başına algılayabilen ve gezinebilen bir araçtır. Son yıllarda, birkaç teknolojik firma ve üniversite, araştırma ve geliştirme için hem teknik hem de finansal olarak büyük kaynaklar yatırarak otonom araçlara büyük ilgi göstermiştir. Devam eden kapsamlı araştırmalardan bazıları Alphabet'in yan kuruluşu Waymo, NVIDIA'dan PilotNet, General Motor'un Cruise, Tesla'nın Tesla 'S' modeli, Ford'un Agro AI 'dir [1]. Otonom araba ADAS gibi altyapı sistemleri ile koordineli olarak çalışarak yoldaki mevcut trafik durumunu, araç yakınındaki nesneleri, tehlikeleri ve zorlukları analiz ederek daha güvenli, rahat ve sağlıklı bir sürüş sağlar. Otonom araba, nesne algılama, şerit algılama, engel algılama, trafik işareti algılama, kendi kendine park etme, tümsek algılama, çukur algılama, kaza algılama vb. özelliklere sahip olmalıdır. Ancak aydınlatma ve hava koşullarından kaynaklanan varyasyonlar, gerçek zamanlı ortamın karmaşıklığı, görüş yakalama açısı nedeniyle trafik sahnelerinde nesne algılama işlevinin zorlukları artmaktadır. Çok yüksek doğruluk ve gerçek zamanlı hız ile nesne algılamaların gerçekleştirilmesi için çalışmalar devam etmektedir. Bu çalışmada yukarıda belirtilen zorlukların üstesinden gelmek için geleneksel makine öğrenimi ve bilgisayarlı görme modellerinin aksine son zamanlarda nesne algılama çalışmalarında yaygın bir rol almaya başlayan ve daha iyi performanslar sağlayan derin öğrenme modelleri kullanılmıştır.

978-1-7281-9090-7/20/$31.00 ©2020 IEEE

Derin öğrenme, beynin yapısal ve işlevsel özelliklerinden esinlenerek tasarlanan, çok katmanlı ağ yapıları olan yapay sinir ağları üzerinde çalışan algoritmalar ve modeller kümesi ile ilgili bir makine öğrenmesi alt alanıdır. 2012 yılında Krizhevsky ve arkadaşlarının nesne tanıma alanında büyük bir yarışma olan ImageNet Büyük Ölçekli Görüntü Tanıma Yarışması'nı (ILSVRC) derin öğrenmenin temel mimarisi kabul edilen Evrişimsel Sinir Ağı'nı (CNN) kullanarak kazanmaları ile derin öğrenme bilim dünyasında büyük bir etkiyi oluşturarak CNN'lere olan ilgiyi canlandırdı [2]. Makine öğreniminde, programcıların oluşturduğu algoritmalarla verilerden özellikler çıkarma gerçekleştirilirken yani özellik çıkarma mühendisliği yapılırken, derin öğrenmede özellik çıkarma konvolüsyon katmanlarında gerçekleşmektedir. Ham veriler derin öğrenme modeline verilir ve konvolüsyon katmanlarında özellik çıkarma işlemleri gerçekleştirilir. Elde edilen özellikler sınıflandırma katmanına verilerek sonuç elde edilir, böylece model kendi kendine öğrenmeyi sağlamış olur. Derin öğrenmenin sağladığı bu yararlı işlevinden dolayı bu çalışmada CNN modelleri nesneleri algılama amaçlı kullanılmıştır. Nesne algılama, belirli bir görüntüdeki bir veya daha fazla nesnenin varlığını, konumunu ve türünü tanımlamayı içeren bir işlevdir. Nesneler genellikle resimlerden veya video beslemelerinden tanımlanabilir. Bu çalışmada nesne algılama adı altında yaya, araba, bisiklet ile dur, yaya geçidi, yol ver, döner kavşak, kavisli yol, 20 hız sınırı ve 30 hız sınırı trafik işaretleri algılanmaya çalışılmıştır. Daha önceki çalışmalarda ayrı ayrı araba algılama, trafik işareti algılama ve yaya algılama uygulamaları yapılmışken bu çalışmada içerisine kamera yerleştirilen bir arabanın seyir halindeyken veya durağan iken görüş alanı içerisinde bulunan yayaları, arabaları, bisikletleri ve belirlenen trafik işaretlerini tek bir model üzerinde algılama çalışması yapılmıştır. Önceden eğitilmiş 4 model üzerinde gerçekleştirilen nesne algılama çalışmasında doğruluk oranı olarak en iyi performansı sağlayan model tespit edilmeye çalışılmıştır.

II. LİTERATÜR ÇALIŞMASI

Trafik işaretlerini algılama, yayaları algılama ve arabaları algılama sistemleri için literatürde çeşitli çalışmalar gerçekleştirilmiştir. Gerçekleştirilen çalışmaların ortak amacı algılama sistemlerinin yüksek doğrulukta ve gerçek zamanlı hızda çalışmalarını sağlamaya yöneliktir. Geleneksel bilgisayarlı görme işlevleri derin öğrenmenin ortaya çıkışı ile tamamen değişmiştir. Derin öğrenme teknolojisi uygulanarak, nesne algılama performansı önemli ölçüde artmıştır ve nesne algılamada en sık CNN tabanlı yöntemler kullanılmıştır.

Trafik işareti algılama sistemleri için çeşitli yaklaşımlar incelenmiştir. Liang ve arkadaşları iki modülden oluşan bir dedektör önermiştir. İlki, işaret sınırlarının ortak özelliklerini kullanır ve ilgilenilen bölgeleri (ROI) çıkarır. İkincisi, ROI'ler üzerinde daha ince doğrulamalar gerçekleştirir ve Yönlendirilmiş Gradyanların Histogramları (HOG) ve Destek Vektör Makineleri (SVM) kombinasyonunu kullanarak trafik işaretlerini tespit eder [3]. Chauhan ve arkadaşları 2020 yılında yaptıkları çalışmada 43 trafik işaretini sınıflandırmak için CNN modelini TensorFlow derin öğrenme kütüphanesini kullanarak eğitmiş ve %95 doğruluk oranı elde etmiştir [4]. Jung ve arkadaşları 2016 yılında trafik işareti bölgelerinin çıkarılmasının ilk aşamada yapıldığı ve LeNet-5 CNN mimarisi ile sınıflandırmanın daha sonraki aşamada gerçekleştiği CNN tabanlı trafik işareti tanıma algoritması geliştirmiştir [5]. Arcos-Garcia ve arkadaşlarının 2018 yılında yaptıkları çalışmada daha önce COCO veri seti üzerinde eğitilmiş olan 8 derin öğrenme modeli transfer öğrenimi yoluyla Alman trafik işaretleri veri seti (GTSDB) üzerinde yeniden eğitilmiştir. Modellerin değerlendirmesi ve karşılaştırması doğruluk (mAP), bellek tüketimi, hız, kayan nokta işlemlerinin (FLOPs) sayısı, modelin parametre sayısı ve trafik işareti görüntü boyutlarının etkisi gibi temel ölçütler üzerine yapılmıştır [6]. 2016 yılında Fan ve arkadaşları Fast R-CNN modelini, Almanya'nın Karlsruhe şehrinde trafik sahnelerinde araç tespiti için kullanmıştır [7]. Dalal ve Triggs 2005 yılında, algoritmanın özelliklerinin yaya tespitini bir dereceye kadar iyileştirebileceği HOG algoritmasını önermiştir [8]. Yönlü gradyan histogramı filtrelerinin ve makine öğrenimi algoritmaları gibi görüntü sınıflandırması için geleneksel filtre tabanlı tekniklerin, büyük hacimli yaya giriş görüntüleri için iyi performans göstermekte zorlandığı tespit edilmiştir [9]. Zhang ve arkadaşları 2017 yılında INRIA veri seti üzerinde Faster R-CNN modelini kullanarak yaya algılamada %92.7 doğruluk oranı elde etmiştir. Görüntüler CNN 'den geçirildikten sonra elde edilen özeliklerden yaya olma ihtimali olan öneri bölgelerini elde etmek için K-means küme analizi kullanılmıştır [10]. Benenson ve arkadaşları 2014 yılında derin konvolüsyonel sinir ağı kullanarak 2D ve 3D veri kümesi ile bilgisayar vizyonu tabanlı yaya algılama metodolojileri üzerine çalışma yapmıştır [11]. Zhang ve arkadaşları yayaları etkin bir şekilde algılayabilen, gerçek zamanlı ve doğruluğu iyileştirebilen LeNet-5 CNN yapısına dayalı yeni bir yaya algılama yöntemi önermiştir [12]. Kim ve arkadaşları yaya algılama etkinliğini artırmak için bir CNN modeli olan VGG-16 'nın optimize edilmiş mimarisini önermiştir. INRIA veri seti kullanılan çalışmada önerilen derin öğrenme modeli %98.5 doğruluk oranı sağlayarak makine öğrenimi modellerinden ve hibrit modellerden daha iyi performans gösterdiği tespit edilmiştir [9]. Kim ve arkadaşları 2016 yılında karayolu ortamında nesne tespitini sağlamak için SSD modelini kullanmıştır. Eğitimde KITTI veri setini kullanan araştırmacılar SSD modeli üzerinde ince ayarlar gerçekleştirerek ve veri setini artırma yoluna giderek nesne tespiti çalışmasında performans artışı sağlamıştır [13]. 2018 yılında Shetty ve arkadaşları, SSD modelini kullanarak, sonuçların üretilmesinde çok hızlı olan ancak doğruluk oranının düşük olduğu bir nesne tespitini gerçekleştirmiştir. Faster-RCNN modelini kullanarak gerçekleştirdikleri nesne tespiti çalışmasında ise tespit edilen nesnenin doğruluğunun SSD 'ye kıyasla daha yüksek olduğunu ve sonuçları üretmek için gereken zamanın SSD 'ye kıyasla daha fazla olduğunu tespit etmiştir [14].

III. METODOLOJİ

Derin öğrenme modellerinden SSD Inception V2, Faster R-CNN Inception V2, Faster R-CNN Resnet 50 ve Faster R-CNN Resnet 101 aynı veri seti üzerinde çalıştırılıp nesne algılamadaki doğruluk değerleri (mAP) karşılaştırılarak en iyi performansa sahip model bulunmaya çalışılmıştır.

A. Veri Seti

Derin öğrenme ve genel olarak makine öğrenmenin önemli bir unsuru ağı besleyen verilerdir. Veriler ses verileri, metin verileri, görüntü verileri veya video verileri gibi farklı türlerde olabilir. CNN modellerinin beslendikleri veri setlerinden yararlı özelikler elde edebilmesi için kaliteli ve

çok miktarda verinin olması gerekir ve bunu sağlamak için de çaba ve zaman gerekir. Transfer öğrenimi, bir modelin büyük veri setleri üzerinde kapsamlı eğitiminden elde ettiği bilgiyi mevcut modele aktarma yöntemidir. Derin ağ modelleri, transfer öğrenimi yöntemiyle daha az veri ile eğitilebilir ve önemli ölçüde performans elde edebilir. Bu çalışmada 91 nesne kategorisine sahip 328.000 resim içeren COCO veri setinden elde edilen bilgilerden yararlanmak için transfer öğrenimi kullanılmıştır. Bu çalışmada ağı eğitmede kullanılacak insan, araba ve bisiklet görüntülerinin bir kısmı nesne sınıflandırma ve nesne tanıma çalışmaları için oluşturulan GRAZ-01 ve GRAZ-02 [15] veri setlerinden bir kısmı da cep telefonu kamerasıyla elde edilmiştir. Trafik işaretlerinden eğitimde kullanılacak yaya geçidi, dur, yol ver, döner kavşak, kavisli yol, 20 hız sınırı ve 30 hız sınırı işaretleri cep telefonu kamerasıyla çekilmiştir. 517 görüntüden oluşan veri setinden bir kesit Şekil 1'de gösterilmektedir.

Şekil 1. Veri setinden örnekler

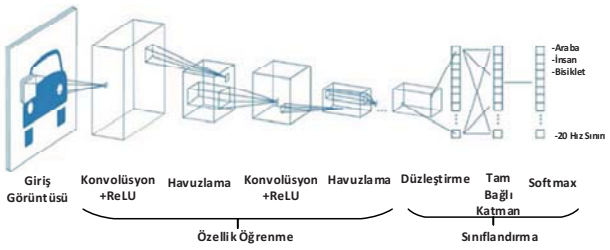
TABLO I. VERI SETINDE KULLANILAN NESNELER VE SAYILARI

Nesne	*Eğitim Verileri*	*Test Verileri*
İnsan	169	35
Araba	503	107
Bisiklet	96	25
Dur	30	12
Yaya geçidi	87	28
Yol ver	59	15
Döner kavşak	54	12
Kavisli yol	30	5
30 hız sınırı	25	7
20 hız sınırı	13	7

Resimler farklı açılar ve farklı ışıklandırmalar altında elde edilmiştir. Farklı boyutlara sahip olan resimlerin tümü jpg formatındadır ve bir görüntü birden fazla nesne içerebilmektedir. 517 görüntünün %80 'ni eğitim verisine, %20 'si test verisine ayrılmıştır ve ayrılan verilerde bulunan nesne sayıları Tablo I'de gösterilmektedir.

B. Nesne Algılama Modelleri

Klasik sinir ağları tek boyutlu özellik vektörleri ile çalışırken CNN'ler, verileri matris formatta alır ve her konvolüsyon katmanında bulunan eğitilebilir filtreler ile işler. CNN, bir görüntüdeki mekansal ve zamansal bağımlılıkları ilgili filtreler kullanarak başarılı bir şekilde yakalayabilir. Basit bir CNN mimarisi oluşturmak için Şekil 2'de gösterildiği gibi temel olarak konvolüsyon katmanı, havuzlama katmanı ve tam bağlı katman olmak üzere üç ana katman türü kullanılır. CNN aldığı giriş verilerini ardı ardına gelen katmanlarda filtreler kullanarak işler. Filtreler, eğitim sırasında değerlerini kendi kendine öğrenir ve verilerdeki belirli desenleri ortaya çıkarır. Havuzlama katmanı, verileri işleme kolaylığı sağlamak için belirli yöntemler kullanarak konvolüsyon katmanlarından gelen verilerin boyutunda azaltmaya gider. Son adım olarak, elde edilen veriler vektör haline getirilir ve çok katmanlı algılayıcılar kullanılarak sonuç elde edilir. Elde edilen sonuç ile istenen sonucun farkı kadar bir hata oluşur. Bu hatanın minimum olması istenir. Ağırlıkların güncellenerek hatanın azaltılması için geri yayılım algoritması kullanılarak hata tüm ağırlıklara aktarılır. Her bir iterasyonla ağırlıkların güncellenmesi yapılarak hatanın azaltılması sağlanır.



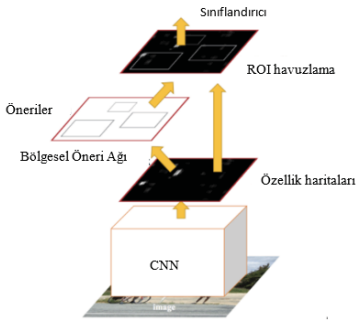
Şekil 2. Evrişimsel sinir ağının genel mimarisi [16]

CNN'ler, görüntü özelliklerini öğrenme konusunda güçlü bir yeteneğe sahiptir ve sınıflandırma ve sınırlayıcı kutu regresyonu gibi birden çok görevi yerine getirmektedir. Algılama yöntemi genel olarak iki kategoriye ayrılabilir. İki aşamalı yöntem, çeşitli algoritmalar aracılığıyla nesnenin bir bölge önerisini oluşturur ve ardından nesneyi bir evrişimli sinir ağı ile sınıflandırır. Tek aşamalı yöntem bir bölge önerisi oluşturmaz, ancak nesne sınırlayıcı kutusunun konumlandırma problemini işlemek için bir regresyon problemine doğrudan dönüştürür. Tek aşamalı nesne algılama modelleri SSD ve YOLO, iki aşamalı nesne algılama modelleri Bölgesel Evrişimsel Sinir Ağları (R-CNN) ve ailesidir [14].

SSD (Single Shot Multibox Detector) [17], farklı boyuttaki nesneleri işlemek için farklı çözünürlükteki özellik haritalarından üretilen tahminleri birleştirme kapasitesine sahiptir. Bir SSD modelinin ilk ağ katmanları, yüksek kaliteli görüntü sınıflandırması için kullanılan standart bir mimariye dayanmaktadır. Daha sonra, algılama amacıyla çok ölçekli özellik haritalarını üretmek için ağa bir yardımcı yapı eklenir. Bu yapı, özellik haritalarının boyutunu aşamalı olarak küçültmek ve birden çok ölçekte algılama tahminlerine izin vermek için evrişimli özellik katmanlarından oluşur.

Faster R-CNN (Faster Region Based Convolutional Networks) [18], CNN'i kullanarak giriş görüntüsünden özellik haritaları çıkarır ve daha sonra bu haritaları nesne önerilerini döndüren bir bölge teklif ağından (RPN'den)

geçirir. Sonrasında RPN 'den elde edilen tüm önerileri aynı boyuta getirmek için ilgi bölgesi havuzlama katmanı uygular. Son olarak, aynı boyuta getirilen öneriler, sınırlayıcı kutuları sınıflandırmak ve tahmin etmek için tam bağlı katmana iletilir. Faster R-CNN modeli Şekil 3'te gösterilmektedir.



Şekil 3. Faster R-CNN [18]

C. Özellik Çıkarıcılar

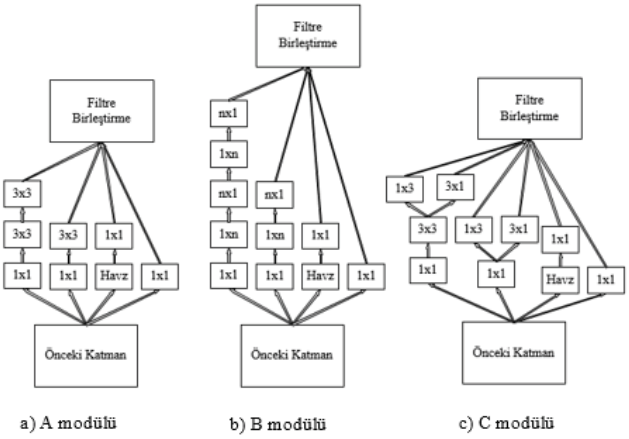
Görüntü sınıflandırma işlevinde giriş görüntülerinden üst düzey özellikler elde etmek için özellik çıkarıcılar olarak Resnet 50, Resnet 101 ve Inception V2 evrişimsel sinir ağları kullanılmıştır.

Resnet 50 ve Resnet 101, ILSVRC 2015 ve COCO 2015 yarışmalarında algılama, bölümleme ve sınıflandırma gibi birçok zorluğu başarıyla tamamlayarak birinciliği kazanan derin artık (residual) ağlardır [19]. Teorik olarak, modelde katman sayısı arttıkça başarımın artacağı düşünülür. Ancak yapılan deneyler, daha derin modellerin derinlikleri artıkça iyi performans göstermediğini ortaya koymuştur. Resnet kullanılarak bu sorun çözülmeye çalışmıştır. Faster R-CNN modelinin özellik çıkarıcıları olarak kullanılmak üzere bu ağlar iki aşamaya ayrılmıştır. İlki, RPN özelliklerinin çıkarılmasını gerçekleştirir ve ikincisi, sınırlayıcı kutu sınıflandırıcı özelliklerini çıkarır. Bu özellik çıkarıcıların her ikisi de dört artık blokla oluşturulmuştur: ilk üçü (konv2x, konv3x ve konv4x) RPN özelliklerini çıkarırken, konv4 x 'ın son katmanı bölge önerilerini tahmin etmek için kullanılır. Ek olarak, kutu sınıflandırıcı özellikleri, dördüncü artık bloğun son katmanı (konv5x) tarafından çıkarılır. Şekil 4, farklı katmanlardan oluşan Resnet mimarilerini göstermektedir.

katman adı	çıkış boyut	18-katman	34-katman	50-katman	101-katman	152-katman
konv1	112×112	7×7, 64, adım 2				
		3×3 maks havuzlama, adım 2				
konv2_x	56×56	[3×3, 64; 3×3, 64] ×2	[3×3, 64; 3×3, 64] ×3	[1×1, 64; 3×3, 64; 1×1, 256] ×3	[1×1, 64; 3×3, 64; 1×1, 256] ×3	[1×1, 64; 3×3, 64; 1×1, 256] ×3
konv3_x	28×28	[3×3, 128; 3×3, 128] ×2	[3×3, 128; 3×3, 128] ×4	[1×1, 128; 3×3, 128; 1×1, 512] ×4	[1×1, 128; 3×3, 128; 1×1, 512] ×4	[1×1, 128; 3×3, 128; 1×1, 512] ×8
konv4_x	14×14	[3×3, 256; 3×3, 256] ×2	[3×3, 256; 3×3, 256] ×6	[1×1, 256; 3×3, 256; 1×1, 1024] ×6	[1×1, 256; 3×3, 256; 1×1, 1024] ×23	[1×1, 256; 3×3, 256; 1×1, 1024] ×36
konv5_x	7×7	[3×3, 512; 3×3, 512] ×2	[3×3, 512; 3×3, 512] ×3	[1×1, 512; 3×3, 512; 1×1, 2048] ×3	[1×1, 512; 3×3, 512; 1×1, 2048] ×3	[1×1, 512; 3×3, 512; 1×1, 2048] ×3
	1×1	ortalama havuzlama, 1000-d tam bağlı,softmax				

Şekil 4. Farklı katmanlardan oluşan Resnet mimarileri

Inception V2, ILSVRC 2014 yarışmasında Google ekibinin evrişimsel sinir ağlarının hesaplama karmaşıklığını azaltmada ve performansı artırmada çığır açan Inception(başlangıç) modülünün geliştirilmesiyle elde edilmiştir. Bir giriş görüntüsü için, inception modülü paralel olarak birden fazla konvolüsyon işlemini gerçekleştirir. Sonra onların çıktıları tek bir çıktı vektöründe birleştirilir. Farklı ölçeklerde bilgiler çıkarmak için 1x1, 3x3 ve 5x5 gibi farklı filtre boyutları kullanılır. Boyutsallığı azaltmak için 1x1 konvolüsyonlar kullanılarak hesaplama karmaşıklığı azaltılır. Inception V2 modülü Şekil 5'te gösterildiği gibi üç farklı tip modüle sahiptir. İlk modül (A), 5x5 konvolüsyonu 2 katmanlı 3x3 konvolüsyon olacak şekilde değiştirilerek, hesaplama yükünü %28 azaltmış ve performansı artırmıştır [20].



Şekil 5. Inception V2 modülleri [20]

3x3 konvolüsyonu 3x1 ve 1x3 konvolüsyonlu alt katmanlara ayrıldığında %33 kazanç sağlanmıştır (B). Filtre genişletilerek daha yüksek boyutlu gösterimler ilkesinin bir ağ içinde yerel olarak işlenmesi daha kolaylaşmıştır ve burada modül genişletilmiştir (C) [20]. Inception V2 ağının mimarisi Tablo II'de gösterilmektedir.

TABLO II. INCEPTION V2 MIMARISI [20]

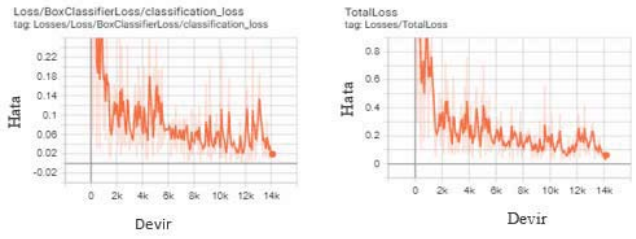
İşlem türü	***Filtre boyutu/ Adım Sayısı***	***Giriş boyutu***
konvolüsyon	3x3/2	229x229x3
konvolüsyon	3x3/1	149x149x32
konvolüsyon	3x3/1	147x147x32
havuzlama	3x3/2	147x147x64
konvolüsyon	3x3/1	73x73x64
konvolüsyon	3x3/2	71x71x80
konvolüsyon	3x3/1	35x35x192
3xinception	A modülündeki gibi	35x35x288
5xinception	B modülündeki gibi	17x17x768
2xinception	C modülündeki gibi	8x8x1280
havuzlama	8x8	8x8x2048
lineer	logits	1x1x2048
softmax	sınıflandırıcı	1x1x1000

IV. DENEY

Yapılan literatür çalışmaları sonucunda görüntü sınıflandırmada başarılarını kanıtlamış önceden eğitilmiş Faster R-CNN Inception V2, SSD Inception V2, Faster R-CNN Resnet 50 ve Faster R-CNN Resnet 101 modellerinin kullanılması uygun görülmüştür. Önceden eğitilmiş

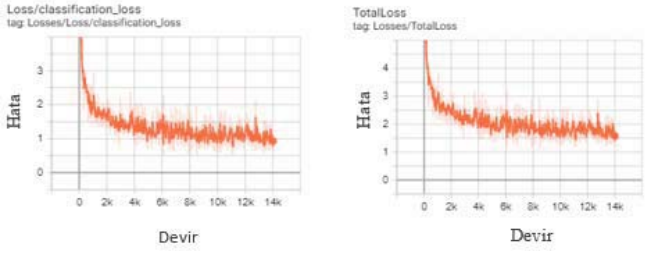
modellerin konfigürasyon dosyaları üzerinde değişiklikler yapılmış ve modeller TensorFlow Nesne Algılama API [21] kullanılarak tekrardan eğitilmiştir. 10 nesneyi algılamadaki performansları değerlendirilmiştir ve bu çalışma için hem hız açısından hem de doğru tespitler (mAP) gerçekleştirme açısından en iyi model seçilmeye çalışılmıştır. Modellerin eğitimi Intel Core i7 CPU işlemciye sahip bilgisayar üzerinde gerçekleştirilmiştir.

Her dört modelde sınıflandırıcı olarak softmax fonksiyonu kullanılarak nesne sınıflarının olasılıkları elde edilmiştir. Faster R-CNN Inception V2 modelinde optimizasyon yöntemi olarak 'momentum' kullanılmış, ilk öğrenim değeri (learning rate) 0.0002, batch size değeri 1 ve momentum değeri 0.9 olarak belirlenmiştir. Eğitim 14000 devir ile gerçekleştirilerek TensorBoard üzerinde elde edilen hata değerlerinin değişimi Şekil 6'da gösterilmiştir. TensorFlow tarafından sağlanılan TensorBoard; yapılan çalışmaların grafiklerini görselleştirmeye, grafiklerin çalışmasıyla ilgili nicel ölçümleri çizdirmeye yarayan görselleştirme aracıdır.



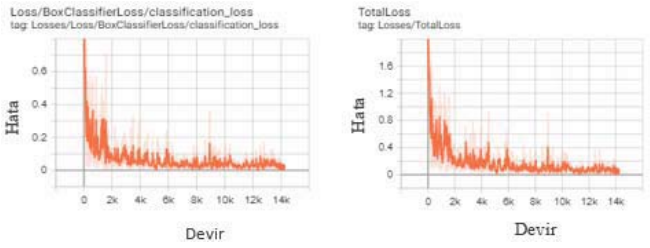
Şekil 6. Faster R-CNN Inception V2 modelinin 14000 eğitiminde hata değerinin değişimi

SSD Inception V2 modelinin konfigürasyon dosyasında optimizasyon yöntemi olarak 'RMS Prob' kullanılmıştır. İlk öğrenme değeri 0.004, batch size değeri 24 ve momentum değeri 0.9 olarak belirlenmiştir. Bu model giriş görüntülerini 300x300 boyutuna getirerek işlemleri gerçekleştirir. Model 14000 devir (epoch) ile eğitilmiş ve eğitim sürecinde hata değerinin değişimi Şekil 7'de gösterilmiştir. Eğitim sürecinde devir sayısı artarken hata değerindeki azalım diğer modellere oranla yavaş olmaktadır.



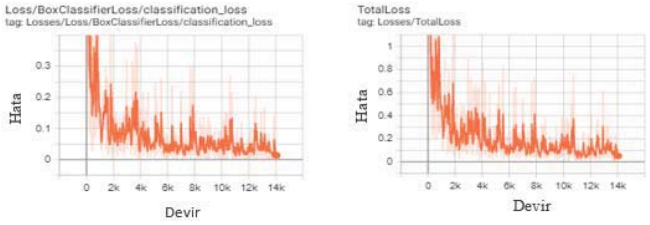
Şekil 7. SSD Inception V2 modelinin 14000 devir eğitiminde hata değerinin değişimi

Optimizasyon yöntemi olarak 'momentum' kullanan ve ilk öğrenme değeri 0.0003, batch size değeri 1, momentum değeri 0.9 olarak belirlenen Faster R-CNN Resnet 101 modelinin 14000 devir sonucunda elde edilen hata değeri ve değişimi Şekil 8'de gösterilmiştir.



Şekil 8. Faster R –CNN Resnet 101 modelinin 14000 devir eğitiminde hata değerinin değişimi

Özellik çıkarıcı olarak Resnet 50 kullanan Faster R-CNN Resnet 50 modelinin konfigürasyon dosyasında ilk öğrenme oranı 0.0003, batch size 1, momentum değeri 0.9 olarak belirlenmiş ve eğitim 14000 devirde gerçekleştirilmiştir. Optimizasyon yöntemi olarak 'momentum' kullanan modelin eğitim sürecinde hata değerindeki değişimler Şekil 9'da gösterilmiştir.



Şekil 9. Faster R-CNN Resnet 50 modelinin 14000 devir eğitiminde hata değerinin değişimi

Modellerin eğitimi yapıldıktan sonra değerlendirme aşamasında test görüntülerinde bulunan nesne veya nesnelerin tespit etme başarımına ilişkin verilerin hesaplamasında, TensorFlow Şekil 10 'da gösterilen karışıklık (confusion) matrisinden yararlanır.

Modeller değerlendirilirken karışıklık matrisinden türetilen metrik değerleri kullanılır. Matristen elde edilen veriler kullanılarak elde edilen hassasiyet, özgünlük, doğruluk ve kesinlik gibi metrikler aşağıda belirtildiği gibi hesaplanmaktadır.

		Tahmin Edilen Sınıf	
		Pozitif (P)	Negatif (N)
Gerçek Sınıf	Pozitif (P)	Gerçek Pozitif (TP)	Yanlış Negatif (FN)
	Negatif (N)	Yanlış Pozitif (FP)	Gerçek Negatif (TN)

Şekil 10. Modellerin değerlendirilmesinde kullanılan karışıklık matrisi

Kesinlik (precision) metriği 'tahmin edilen örnekler arasında gerçekten kaç tanesi doğrudur' sorusuna cevap verir. Hassasiyet (recall) metriği 'pozitif sınıfa ait örneklerden kaç tanesi doğru tahmin edildi' sorusuna cevap verir. Özgünlük metriği yanlış pozitif oranıdır. Denklem 1'de doğruluk değerinin, (2)'de özgünlük değerinin, (3)'te hassasiyet değerinin ve (4)'de kesinlik değerinin nasıl hesaplandığı belirtilmiştir.

Doğruluk (accuracy):

$$(TP+TN) / (FP+FN+TP+TN) \quad (1)$$

Özgünlük (Specificity):

$$TN / (TN+FP) \quad (2)$$

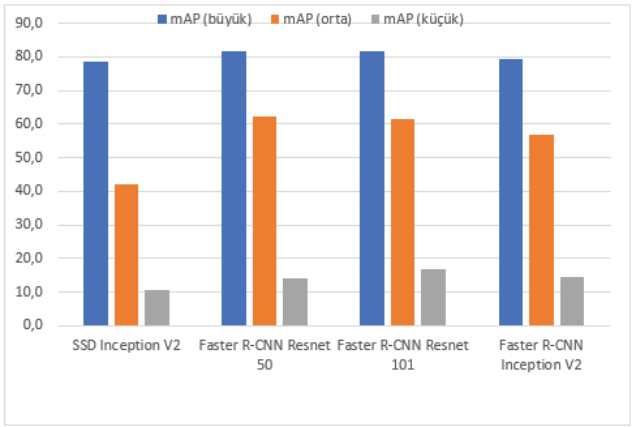
Hassasiyet (recal):

$$TP / (TP+FN) \quad (3)$$

Kesinlik (precision):

$$TP/ (TP+FP) \quad (4)$$

mAP (Mean Average Precision) nesne algılayıcıların doğruluğunu ölçen ölçüttür. mAP(small) 32x32 pikselden küçük olan nesneleri algılama başarısını, mAP(medium) 32x32 ile 96x96 piksel arasında olan nesneleri yakalama başarısını ve mAP(large) 96x96 ile 10000x10000 piksel arasında olan nesneleri yakalama başarısını belirtir.



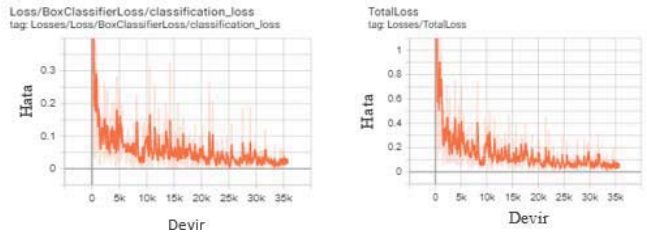
Şekil 11. 4 modellin mAP ile hesaplanan doğruluklarının karşılaştırılması

14000 devirde eğitilen modellerin birbirlerine yakın değerlerde doğruluk değerleri verdikleri tespit edilmiştir ve Şekil 11'de gösterilmiştir. Video görüntüleri üzerinde test edilen 4 modelden SSD Inception V2'nin nesne tespit etmede hızlı olmasına rağmen tespit ettiği nesne sayısının az olduğu gözlemlenmiştir. Daha iyi performans elde edebilmek için Faster R-CNN Inception V2, Faster R-CNN Resnet 50 ve Faster R-CNN Resnet 101 modelleri 35000 devirde yeniden eğitilmişlerdir. Yeniden eğitilen modellerin eğitim parametreleri Tablo III'de belirtilmiştir.

TABLO III. 35000 DEVIRDE EĞITILECEK MODELLERIN PARAMETRELERI

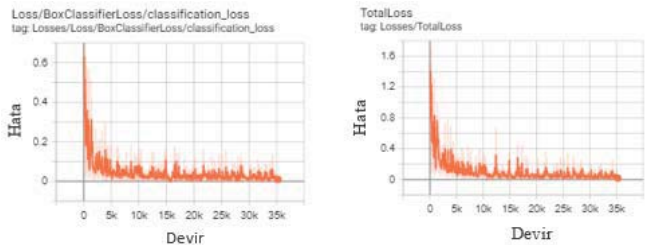
Model ismi	*Optimizasyon yöntemi*	*Öğrenme değeri*	*Batch size*	*Momentum değeri*
Faster R-CNN Inception V2	Momentum	0.0002	1	0.9
Faster R-CNN Resnet 50	Momentum	0.0003	1	0.9
Faster R-CNN Resnet 101	Momentum	0.0003	1	0.9

Faster R-CNN Inception V2 modelinin 35000 devirde gerçekleşen eğitimindeki hata değişim grafiği Şekil 12'de gösterilmiştir.



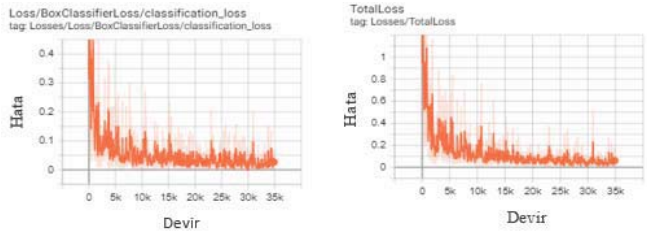
Şekil 12. Faster R-CNN Inception V2 modelinin 35000 devir eğitiminde hata değerinin değişimi

Faster R-CNN Resnet 101 modelinin 35000 devir eğitim sürecinde hata değerinin değişimi Şekil 13'te gösterilmiştir.



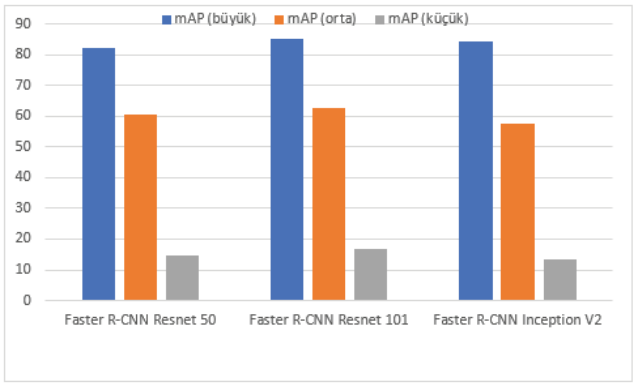
Şekil 13. Faster R-CNN Resnet 101 modelinin 35000 devir eğitiminde hata değerinin değişimi

35000 devirde eğitilen Faster R-CNN Resnet 50 modelindeki hata değerinin değişimi Şekil 14'te gösterilmiştir.



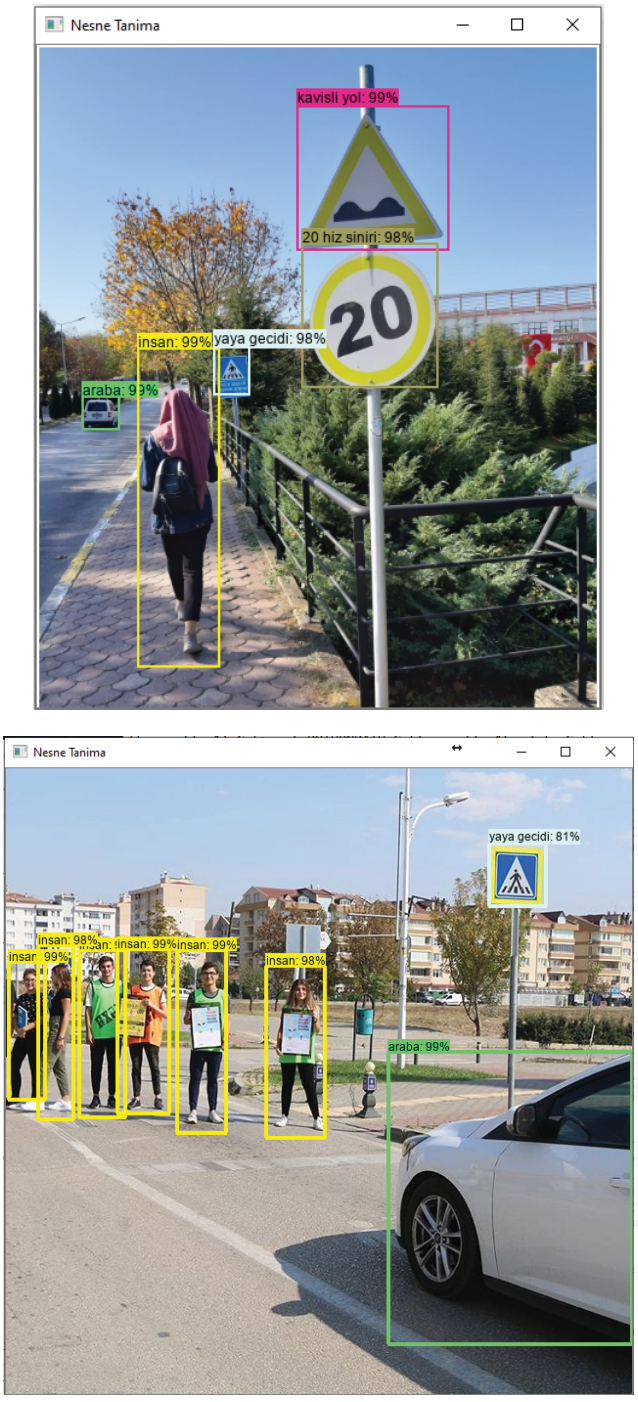
Şekil 14. Faster R-CNN Resnet 50 modelinin 35000 devir eğitiminde hata değerinin değişimi

35000 devirde gerçekleştirilen eğitimde performansta iyileşme gösteren modellerden en yüksek doğruluk değerini Faster R-CNN Resnet 101 göstermiştir. Şekil 15'te 35000 devirde eğitilen modellerin doğruluk değerlerinin karşılaştırılması gösterilmiştir.



Şekil 15. 3 modelin mAP ile hesaplanan doğruluklarının karşılaştırılması

Faster R-CNN Resnet 101 modelinin Şekil 16' da görüntüler üzerinde, Şekil 17'de videolar üzerinde gerçekleştirilen nesne algılama işlemleri gösterilmiştir. Yapılan çalışmalarda araçtan yaklaşık 50 metre uzaktaki işaretler doğru bir şekilde algılanabilmiştir. Veri seti oluşturulduğunda nesnelerin bir kısmı kapatılmış durumdaki görüntüler de kullanılmıştır. Uygulama aşamasında bir kısmı kapatılmış trafik işretlerini algılamada sorun yaşanmıştır. Bu durumun ağı eğitmede kullanılan trafik işaretlerinin sayısının diğer nesnelere oranla az olmasından kaynaklı olduğu düşünülmüştür.



Şekil 16. Görüntü üzerinde nesne algılama

Çalışma, saatte 30-70 km hızları arasında giden bir taşıt içerisinden cep telefonu kamerasıyla kaydedilmiş video görüntüleri üzerinde test edilmiştir.

Şekil 17. Videoda nesne algılama

V. SONUÇLAR

Daha önce COCO veri setiyle eğitilen Faster R-CNN Inception V2, SSD Inception V2, Faster R-CNN Resnet 50 ve Faster R-CNN Resnet 101 modelleri; 10 nesneye ait 517 görüntü ile transfer öğrenimi yöntemi kullanılarak yeniden eğitilmiştir. Intel Core i7 CPU işlemciye sahip bilgisayarda; 14000 devirde gerçekleştirilen eğitimde modellerin birbirine yakın doğruluk değerinde oldukları tespit edilmiştir. Eğitilen modeller görüntü ve videoya uygulandığında SSD Inception V2 modelinin hız olarak iyi performansa sahip olmasına rağmen doğruluk olarak diğer modellerden geride kaldığı gözlemlenmiş ve özellikle küçük nesneleri tespit etmede zayıf kaldığı görülmüştür.

Faster R-CNN Resnet 50, Faster R-CNN Resnet 101 ve Faster R-CNN Inception V2 modellerinin 35000 devirde gerçekleştirilen eğitimlerinde %85.1 doğruluk değeri ile en iyi sonucun Faster R-CNN Resnet 101 modelinin verdiği tespit edilmiştir. Yapılan testler sonucunda genel olarak, dört modelin de büyük boyuta sahip nesnelerin tespitinde iyi sonuçlar verdiği gözlemlenmiştir.

Otonom arabanın koordineli bir şekilde çalıştığı ADAS sistemlerinin en kısa zamanda en doğru kararları alabilmesi için nesneleri ve işaretleri tespit etmede iyi performans göstermesi önemli bir kriterdir ve bu şartları sağlayan modelin elde edilmesi gerekir. Çalışmamızda doğruluk değerinde iyi performans elde eden model tespit edilmiştir. Fakat hız açısından daha iyi performansın elde edilebilmesi için CPU yerine resim veya video karelerini gerçek zamanlı olarak işleyen GPU'nun kullanılması önerilir. Gelecekteki çalışmalarda, nesneleri iyi algılamak veya sınıflandırmak için yeni çıkan modeller araştırılarak iyi çalıştığı kanıtlanmış olan derin öğrenme mimarileri kullanılabilir ve otonom arabanın şerit algılama ve engel algılama gibi farklı görevleri de bu çalışmaya eklenebilir.

REFERENCES

[1] Kulkarni, R., Dhavalikar, S., & Bangar, S. (2018, August). Traffic Light Detection and Recognition for Self Driving Cars Using Deep Learning. In *2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)* (pp. 1-4). IEEE.

[2] 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).

[3] Liang, M., Yuan, M., Hu, X., Li, J., & Liu, H. (2013, August). Traffic sign detection by ROI extraction and histogram features-based recognition. In *The 2013 international joint conference on Neural networks (IJCNN)* (pp. 1-8). IEEE.

[4] Chauhan A., Rastogi A., Gaur A., & Singh A. (2020). Traffic sign detection using deep learning. *In International Journal of Engineering Applied Sciences and Technology*. (pp. 355-358).

[5] Jung, S., Lee, U., Jung, J., & Shim, D. H. (2016, August). Real-time Traffic Sign Recognition system with deep convolutional neural network. In *2016 13th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)* (pp. 31-34). IEEE.

[6] Arcos-Garcia, A., Alvarez-Garcia, J. A., & Soria-Morillo, L. M. (2018). Evaluation of deep neural networks for traffic sign detection systems. *Neurocomputing*, 316, 332-344.

[7] Fan, Q., Brown, L., & Smith, J. (2016, June). A closer look at Faster R-CNN for vehicle detection. *In 2016 IEEE intelligent vehicles symposium (IV)* (pp. 124-129). IEEE.

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

[9] Kim, B., Yuvaraj, N., Sri Preethaa, K. R., Santhosh, R., & Sabari, A. (2020). Enhanced pedestrian detection using optimized deep convolution neural network for smart building surveillance. *Soft Computing*, 1-12.

[10] Zhang, H., Du, Y., Ning, S., Zhang, Y., Yang, S., & Du, C. (2017, December). Pedestrian detection method based on Faster R-CNN. *In 2017 13th International Conference on Computational Intelligence and Security (CIS)* (pp. 427-430). IEEE.

[11] Benenson, R., Omran, M., Hosang, J., & Schiele, B. (2014, September). Ten years of pedestrian detection, what have we learned?. *In European Conference on Computer Vision* (pp. 613-627). Springer, Cham.

[12] Zhang, C. W., Yang, M. Y., Zeng, H. J., & Wen, J. P. (2019). Pedestrian detection based on improved LeNet-5 convolutional neural network. *Journal of Algorithms & Computational Technology*, *13*, 1748302619873601.

[13] Kim, H., Lee, Y., Yim, B., Park, E., & Kim, H. (2016, October). On-road object detection using deep neural network. *In 2016 IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia)* (pp. 1-4). IEEE.

[14] Shetty, J., & Jogi, P. S. (2018, May). Study on Different Region-Based Object Detection Models Applied to Live Video Stream and Images Using Deep Learning. *In International Conference on ISMAC in Computational Vision and Bio-Engineering* (pp. 51-60). Springer, Cham.

[15] "GRAZ-01, GRAZ-02". http://www-old.emt.tugraz.at/~pinz/data/, [Erişim tarihi 10 Ekim 2019].

[16] "A comprehensive quide to Convolutional Neural Networks". https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53. [Erişim tarihi: 07 Kasım 2019].

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

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

[19] 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).

[20] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the inception architecture for computer vision. *In Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2818-2826).

[21] Huang, J., Rathod, V., Sun, C., Zhu, M., Korattikara, A., Fathi, A., ... & Murphy, K. (2017). Speed/accuracy trade-offs for modern convolutional object detectors. *In Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 7310-7311).