



MobileNet based traffic sign detection system for mobile mapping: Crowdsourced geographical data collection system

Ceren Özcan Tatar^{1*}, Emrah Yılmaz¹, Abdullah Efe², Berk Sönmez², Yalçın Özdemir², Burak Danişan², Hale İrem Beyaz², Engin Yeğnidemir³

¹Department of Remote Sensing and Geographical Information Systems, Graduate School of Science, Eskisehir Technical University, 26555, Eskisehir, Türkiye

²Basarsoft Information Technologies Inc., 06520, Ankara, Türkiye

³Gebze Technical University, Computer Engineering Building, 41400, Kocaeli, Türkiye

Highlights:

- A new method of mobile data collection for map database updating.
- In the proposed method, MobileNet V1 MNAS FPN and MobileNet V2 FPNLite were used.
- A 4-times increase in data production was achieved.

Keywords:

- Crowdsourced Geographical Data Collection System
- Mobile Mapping Systems
- Machine learning
- Artificial neural network
- Traffic signs
- Object detection

Article Info:

Research Article

Received: 08.02.2023

Accepted: 17.09.2023

DOI:

10.17341/gazimmfd.1249165

Acknowledgement:

This study was funded by The Scientific and Technological Research Council of Türkiye (TUBITAK) 1505 program under project number 3200751 and 2244 program under project number 119C200.

Correspondence:

Author: Ceren Özcan Tatar
e-mail:

ceren_ozcan@eskisehir.edu.tr
phone: +90 554 525 6898

Graphical/Tabular Abstract

Mapping systems evolve by integrating object detection and classification, now extending to mobile devices. Utilizing various artificial neural network algorithms for object recognition enhances efficiency in map development and database updates. This study introduces a novel method combining crowdsourcing and machine learning for mobile mapping, tested with data from ITU Racing Driverless TTVS and Turkey, doubling data production speed and reducing data transfer to the central server. Results were evaluated based on Precision, Recall, F1 Score, and mAP values (Figure A).

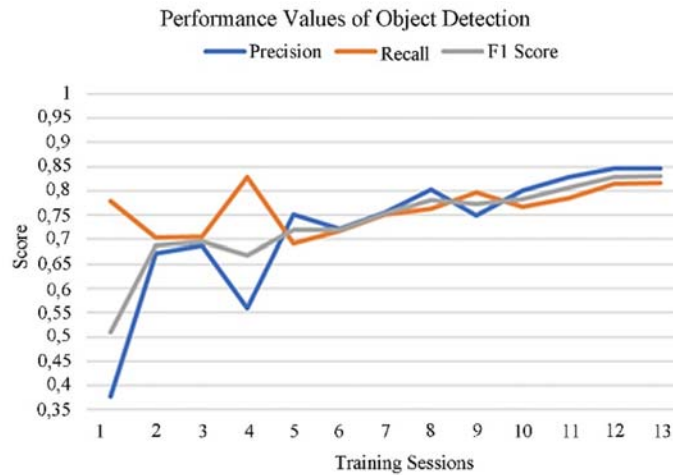


Figure A. Performance values of developed object detection model

Purpose: In this study, it is aimed to develop a novel, low cost, and faster method for updating map databases with machine learning supported crowdsourced data collection system.

Theory and Methods To define traffic signs on maps, ITU Racing Driverless TTVS and GoPro images from various Turkish locations were utilized for model training. TensorFlow and TensorFlow Lite were employed for CBDCS, with MobileNet V1 MNAS FPN and MobileNet V2 FPNLite chosen for object detection. COCO 2017 dataset's pre-trained weights were applied, and Transfer Learning facilitated knowledge transfer in the training pipeline.

Results: It has been seen that MobileNet V1 MNAS FPN is more successful in terms of generating more data as it increases the number of recognized images per second by 4 times. When analyzed in terms of transferring all data to the center, it is measured that 40,295 MB of data is transferred for 1 camera in 1 day when the model is not used, while an average of 79 MB of data is transferred per day when inference is made on a mobile device with machine learning object detection and only the relevant data is transferred to the center.

Conclusion: The inference transmission of the detected objects to the central database by performing the inference on mobile devices reduces the time to obtain the information and make it available. Although the model accuracy is less than that performed on powerful computers in the inference made on mobile devices, it provides an advantage in data generation and updating in terms of shortening the time to access information and speeding up the work of the operators.



Mobil haritalama amaçlı mobilenet tabanlı trafik işaretleri tespit sistemi: Kitlesel coğrafi bilgi toplama sistemi

Ceren Özcan Tatar^{1*}, Emrah Yılmaz¹, Abdullah Efe², Berk Sönmez², Yalçın Özdemir², Burak Danışan², Hale İrem Beyaz², Engin Yeğnidemir³

¹Eskişehir Teknik Üniversitesi, Lisansüstü Eğitim Enstitüsü, Uzaktan Algılama ve Coğrafi Bilgi Sistemleri A.B.D., 26555, Eskişehir, Türkiye

²Başarsoft Bilgi Teknolojileri A.Ş., 06520, Ankara, Türkiye

³Gebze Teknik Üniversitesi, Bilgisayar Mühendisliği Binası, 41400, Kocaeli, Türkiye

ÖNEÇİKANLAR

- Bu makale, harita veri tabanı güncellenmesinde mobil veri toplama için yeni bir yöntem sunmaktadır
- Yapay sinir ağı modellerinden MobileNet V1 MNAS FPN ve MobileNet V2 FPNLite kullanılmıştır
- Önerilen KCVTS ile birim zamanda coğrafi veri üretiminde 4 kat artış sağlanmıştır

Makale Bilgileri

Araştırma Makalesi

Geliş: 08.02.2023

Kabul: 17.09.2023

DOI:

10.17341/gazimmfd.1249165

Anahtar Kelimeler:

Kitlesel coğrafi bilgi toplama sistemi (KCVTS), mobil haritalama sistemleri, makine öğrenmesi, yapay sinir ağları, trafik işaretleri, nesne tespiti

ÖZ

Mobil haritalama sistemleri (*Mobile Mapping Systems*- MMS) coğrafi veri toplama yetenekleri ile birlikte, gelişmiş sürücü destek sistemleri (*Advanced Driver Assistance Systems*- ADAS) ve akıllı ulaşım sistemleri (*Intelligent Transportation Systems* - ITS) gibi birçok uygulama alanının sayısal harita ihtiyacını karşılayabilmektedir. Üretilen haritalarda özellikle trafik işaretlerinin konum ve sınıf bilgilerinin bulunması, bahsi geçen uygulama alanları için önem arz etmektedir. Ancak, MMS tarafından toplanan verilerin geniş ölçekli ve karmaşık olması, trafik işaretlerinin konum-sınıf çıkarımlarını zorlaştırmaktadır. Bu nedenle araştırmacılar, trafik işareti verilerinin işlenmesi için yapay zekâ tabanlı yöntemler geliştirmiştir. Bu çalışmada, trafik işaretlerinin konum ve sınıf bilgilerinin yapay zekâ ile çıkarımına yönelik tasarlanan Kitlesel Coğrafi Bilgi Toplama Sistemi (KCVTS) açıklanmıştır. KCVTS; MobileNet tabanlı mobil cihazlarda etkinlik gösteren, cihazın gerçek-zamanlı kamera görüntülerinde bulunan trafik işaretlerini tespit eden ve sınıflandıran ve böylece, işaretlerin konum-sınıf bilgilerini veri tabanına aktaran hafif-yapılı bir sistemdir. Çalışmada KCVTS'nin manuel işlem gerektiren geleneksel yöntemlerden, trafik işaretlerinin şekil ve renk gibi özelliklerinin çıkarımına dayanan yarı-geleneksel yöntemlerden ve saha verilerinin merkezdeki güçlü bilgisayarlarda, bilgisayarlı görü ve makine öğrenmesi teknikleri ile işlendiği YZ tabanlı yöntemlerden birçok noktada daha pratik ve verimli olduğu gösterilmiştir.

Mobilenet based traffic sign detection system for mobile mapping: Crowdsourced geographical data collection system

HIGHLIGHTS

- This article presents a new method of mobile data collection for map database updating
- MobileNet V1 MNAS FPN and MobileNet V2 FPNLite, which are artificial neural network models were used
- With the proposed CGDCS, a 4-times increase in geographical data production per unit time was achieved

Article Info

Research Article

Received: 08.02.2023

Accepted: 17.09.2023

DOI:

10.17341/gazimmfd.1249165

Keywords:

Crowdsourced geographical data collection system (CGDCS), mobile mapping systems, machine learning, artificial neural networks, traffic signs, object detection

ABSTRACT

Mobile mapping systems (MMS) have gained increasing interest as a cost-effective means of collecting geospatial data, catering to the digital mapping needs of various domains such as advanced driver assistance systems (ADAS) and intelligent transportation systems (ITS). In the generated maps, the location and class information of traffic signs are particularly crucial for the aforementioned applications. However, the extensive and complex nature of data collected by MMS makes it challenging to infer the location and class of traffic signs. Consequently, researchers have developed artificial intelligence-based methods for processing traffic sign data. In this study, a Crowdsourced Geographical Data Collection System (CGDCS) which is designed for the inference of traffic sign location and class information using artificial intelligence is introduced. CGDCS is a lightweight system that operates on mobile devices, leveraging the MobileNet architecture to detect and classify traffic signs present in real-time camera images, thereby transferring the location and class information of the signs to a database. The study demonstrates that CGDCS is more practical and efficient than traditional methods involving manual processing, semi-traditional methods based on the extraction of shape and color features of traffic signs, and AI-based methods that process field data in high-performance computers using high computer vision and machine learning techniques.

*Sorumlu Yazar/Yazarlar / Corresponding Author/Authors : *ceren_ozcan@eskisehir.edu.tr, emrah@basarsoft.com.tr, abduallah.ef@basarsoft.com.tr, berk.sonmez@basarsoft.com.tr, yozdemir@basarsoft.com.tr, burak.danisan@basarsoft.com.tr, hale.beyaz@basarsoft.com.tr, engin@yegnidemir.net / Tel: +90 554 525 6898

1. Giriş (Introduction)

Mobil Haritalama Sistemleri (*Mobile Mapping Systems- MMS*) coğrafi verileri uygun maliyetli yöntemleri ile toplayan bir araç olarak giderek artan bir ilgiye sahiptir. MMS, gelişimleriyle beraber birçok uygulamayı da mümkün kılmaktadır. Bu uygulamalar otonom araçlar, gelişmiş sürücü destek sistemleri (*Advanced Driver Assistance Systems - ADAS*) [1] konum-farkında mobil uygulamalar, acil durum müdahale planlaması, karayolu haritalaması, tesis yönetimi [2], altyapı yönetimi, akıllı ulaşım sistemleri (*Intelligent Transportation Systems - ITS*), akıllı şehirler [3] ve navigasyon sistemleri için 3D şehir modelleri [4] olarak sıralanabilir. Bu uygulamaların birçoğu trafik işaretlerinin konum ve sınıf verilerine ihtiyaç duymaktadır [4]. Fakat, MMS tarafından toplanan verilerin geniş ölçekli ve karmaşık olması, verilerinin işlenmesini zorlaştırmaktadır [3]. Bunun yanında, bu durum, MMS verilerinden trafik işaretlerinin algılanması (*detection*), tanınmasını-sınıflandırılmasını (*recognition-classification*) ve envanterlenmesini (*inventorying*) de zorlu bir süreç haline getirmiştir. Dolayısıyla araştırmacılar trafik işareti verilerinin işlenmesi amacıyla, özellikle yapay zekâ (YZ) tabanlı birçok yöntem geliştirmişlerdir [1]. Bu çalışmada, MMS'de trafik işaretleri işleme sistemlerinin (TİİS) nasıl ele alındığını önce geleneksel yöntemler, sonrasında ise YZ tabanlı yöntemleri açısından incelemekte, yöntemler karşılaştırılmakta ve geliştirilen trafik işaretlerinin konum ve sınıf bilgilerinin yapay zekâ ile çıkarımına yönelik tasarlanan Kitlelesel Coğrafi Bilgi Toplama Sistemi (KCVTS) açıklanmaktadır. Bu kapsamda, çalışma boyunca, "saptama" ve tanıma-sınıflandırma" süreçleri toptan "saptama" olarak adlandırılacaktır. Çalışmada ele alınan diğer modeller ve araştırmacıların önerdiği KCVTS modeli trafik işaretlerinin "saptanmasını ve tanınmasını-sınıflandırılmasını" içermektedir ve "envanterlenmesini içermemektedir. Fakat MMS uygulamalarında amaçlanan trafik işaretlerinin konum ve sınıf bilgileriyle envanterlenmesidir.

Harita verilerinin toplanma yöntemleri geleneksel, yarı geleneksel ve YZ Tabanlı Yöntemler olarak 3 başlıkta incelenebilir:

- **Geleneksel Yöntemler (Traditional Methods):** Yaygın uygulanan geleneksel yöntemlerde saha aracında bir şoför ve personel bulunmakta ve araç ilerledikçe personel, gördüğü trafik işaretleri gibi coğrafi unsurları, mağaza, okul, hastane, gibi önemli yerleri ve diğer şehir öğelerine ait bilgileri manuel olarak araçtaki bilgisayara kaydetmektedir [5, 6]. Bir başka geleneksel yöntemde ise, araç üzerindeki kameralarla çekilen video daha sonra masa başındaki personel tarafından izlenerek coğrafi veri tabanına işlenmektedir [7-9]. Buna ek olarak lazer tarama (LIDAR) ile veri toplama ve ardından işleme de yapılabilmektedir [10-13]. Ayrıca lazer tarama teknikleri iki kategoriye ayrılarak incelenebilir: yersel lazer tarama (*Terrestrial Laser Scanning - TLS*) ve giyilebilir mobil lazer tarama (*Wearable Mobile Laser Scanning - WMLS*). TLS, yüksek doğruluk ve doğrudan ölçüm sağlar, ancak karmaşık ve büyük iç mekanlarda sonuç elde etmek için gereken veri toplama süresini oldukça uzatır. Bu sebeple, TLS'in doğruluk seviyelerine daha az veri ile daha kısa sürede ulaşılması için WMLS sistemleri geliştirilmiştir. Giyilebilir olması, cihazların hafif yapıda olup, sırta giyilebilmesinden veya elde taşınabilmesinden kaynaklanmaktadır. Yapılan kıyaslamalarda, WMLS sistemlerinin karmaşık iç mekanların haritalanmasına alternatif bir çözüm olabileceği görülmüştür. Yapılan gözlemlerde, WMLS'in potansiyeline rağmen insan müdahalesine hala ihtiyaç duyduğu ve işlem sonrası eylemlerin istenen sonuçları elde etmek için gerekli olduğu not edilmiştir [14, 15]. Türkiye'de veri üreticisi şirketlerden biri olan Başarsoft ile yapılan görüşmede günlük olarak araçtan 150-160 görüntü alındığı, görüntü başına yaklaşık olarak 3 etiketleme yapıldığı ve günlük etiketleme sayısının yaklaşık 500 adet olabildiği bilgisi alınmıştır. Dolayısıyla, bu yöntemlerin, YZ tabanlı ve yarı-geleneksel yöntemlere kıyasla daha maliyetli olduğu gözlemlenmiştir.

- **Yarı-Geleneksel Yöntemler (Semi-Traditional Methods):** Yarı-geleneksel algoritmalar YZ tabanlı değildir ve trafik işaretlerinin şekil-renk gibi görsel özelliklerinin çıkarımlarına (*feature extraction*) dayanırlar. Öte yandan, bu algoritmalar aydınlatma, kamera açısı, engeller ve sürüş hızı gibi etkenler tarafından kolayca kısıtlanabilmektedir, dolayısıyla gerçek-zamanlı testlerde başarılı performanslar elde edilememektedir [16, 17].
- **YZ Tabanlı Yöntemler (AI Based Methods):** Görece yeni sayılabilecek YZ tabanlı uygulamalarda ise doğrudan merkeze iletilen görüntü ya da video formatındaki saha verileri güçlü bilgisayarlarda bilgisayarlı görüş (*computer vision*) [18], görüntü işleme (*image processing*), makine öğrenmesi (*Machine Learning - ML*), derin sinir ağları (*Deep Neural Networks - DNN*) ve evrişimli sinir ağları (*Convolutional Neural Networks - CNN*) teknikleriyle işlenip veri tabanına eklenmektedir ve veri tabanı operatörler tarafından denetlenmektedir.

Literatürde trafik işaret levhalarının doğru, hızlı ve uygun maliyetli şekilde işlenmesine yönelik geliştirilen YZ tabanlı farklı çözümler bulunmaktadır. Fakat, bu çözümler tüm verinin merkeze aktarılması ve toplu olarak işlenmesi nedeniyle, süreçte büyük bir gecikmeye sebep olmaktadır. Ayrıca tüm verilerin merkeze aktarılmasının maliyeti de yüksektir. Dolayısıyla bu yöntemlerin gerçek-zamanlı işlev göstermemektedir. Çalışma, YZ tabanlı yöntemlerle ilgili bu sorunları aşabilmek amacıyla TİİS için yeni bir yöntem olan KCVTS'i önermektedir.

Çalışma kapsamında coğrafi veri toplamaya yönelik olarak yapılan literatür taramasında, geleneksel yöntemlerin YZ tabanlı yöntemlerin gerisinde kaldığı fakat YZ tabanlı yöntemlerin de maliyetli olduğu görülmüştür. Çalışma, literatürdeki YZ tabanlı yöntemlerden daha verimli olduğu gözlemlenen MobileNet tabanlı bir trafik işaretleri tespit sistemini tanıtmaktadır. KCVTS, mobil haritalamaya yönelik tasarlanmış, mobil cihazlarda etkinlik gösteren ve trafik işaretlerini gerçek-zamanlı (*real-time*) işleyen bir sistemdir. MobileNet mobil cihazlarda etkinlik göstermesi ve hafif-yapılı (*light-weighted*) olması ile pratiklik sağlamasıyla ve DNN kullanması ile araştırmacılar tarafından tercih edilen bir model olmuştur.

Trafik işaretlerinin daha hızlı ve pratik yöntemlerle işlenmesi amacıyla Seifert, 2004 yılında OptaGIS isimli bir sistem geliştirmiştir. OptaGIS, trafik işareti envanterlerinin Coğrafi Bilgi Sistemleri (CBS) yazılımı ile güncellenmesine yönelik çalışan bir sistemdir ve mobil cihazlarda etkinlik göstermektedir. OptaGIS üç bileşene sahiptir: GPS algılayıcısı, kamera ve standart mobil CBS yazılımı. Bu sistem, trafik işaret levhalarının envanterlerinin tutulması ve bakımının sağlanması işlemlerini gerçekleştirmek amacıyla levha konumunun GPS ile belirlenmesi, kamera ile trafik işaretinin algılanması ve ardından sınıflandırılarak CBS veri tabanında konum ve öznitelik bilgisinin saklanması süreçlerinden oluşmaktadır. Trafik işaretlerinin işlenmesi makine öğrenmesi ML teknikleri ile gerçekleştirilmiştir ve herhangi bir halka açık veri seti kullanılmamıştır. Geliştirilen sistem ile görüntü yakalama ve tanıma testlerinde %98 oranında sınıflandırma başarısı elde edilmiştir [19]. Temelde, OptaGIS'in KCVTS'den iki farkı bulunmaktadır. Öncelikle OptaGIS, ML tekniklerine bağlı kalmak KCVTS ise MobileNet kullanmaktadır. KCVTS'nin OptaGIS'ten ayrıştığı bir diğer nokta ise trafik işaretlerini mobil cihazlarda işleyebilmesidir. OptaGIS trafik işareti verilerini bir mobil cihaza bağlı dahil ya da harici bir kamera yardımı ile alsa da verileri bir sunucuda işlemekte ve direkt mobil cihazda işlememektedir. Bunlarla birlikte, OptaGIS'in mobil haritalama amacıyla inşa edilmiş bir sistem olması KCVTS ile aralarındaki en önemli ortak noktadır ve OptaGIS bu özelliğiyle literatürdeki diğer TİİS çalışmalarından ayrılmaktadır.

TİİS sadece MMS'nin konusu olmadığı ve trafik işaretlerini tespitine odaklanan çoğu çalışmanın ADAS'a yönelik olarak hazırlandığı görülmektedir [20]. Bu kapsamda CNN ve DNN gibi makine öğrenmesi algoritmaları sıklıkla çalışmalarda kullanılmaktadır. Örneğin, Arcos-García'nın 2018'de yaptığı çalışmada, evrişimli ve mekânsal dönüştürücü (*convolutional-spatial transformer*) modülü içeren; ReLU, yerel kontrast normalizasyonu (*local contrast normalisation*) ve max-pooling hiperparametrelerini kullanan bir CNN modeli geliştirilmiştir. Önerilen CNN yaklaşımı ile Alman Trafik İşaretleri Tanıma Veri Seti'nde (*German Traffic Sign Recognition Benchmark – GTSRB*) %99,71 oranında sınıflandırma başarısı yakalanmıştır. Ayrıca gösterilen performans değerlendirildiğinde yöntemin bellek gereksinimi açısından daha verimli olduğunun da altı çizilmiştir [20]. KCVTS, MobileNet'in hafif yapısından yararlanmakta ve mobil cihazlarda etkinlik göstermektedir. Bu noktada, Arcos-García'nın önerdiği CNN modelinden ayrılmaktadır.

Nuakoh tarafından 2019 yılında yapılan çalışmada ADAS'a yönelik olarak trafik işaretlerini tespitine odaklanılmıştır. Çalışmada LISA veri seti kullanılarak ABD trafik işaretlerinin tamamının sınıflandırılması amaçlanmıştır. CNN'den yararlanarak yapılan çalışmada, GTSRB veri seti üzerinde gerçekleştirilen testler sonucunda %98,7 oranında bir sınıflandırma başarısı sağlanmıştır. Daha sonra LISA üzerinde yapılan testler ile birlikte %99,04 oranında bir sınıflandırma başarısı elde edilmiştir [21]. KCVTS'nin Nuakoh'un modelinden ayrıldığı en önemli nokta, ADAS'a yönelik değil, mobil haritalamaya yönelik tasarlanmış olmasıdır. Aynı zamanda KCVTS mobil cihazlarda etkinlik gösterirken Nuakoh çalışmasında tasarlanan modelin böyle bir işlevine atıfta bulunmamıştır.

Pei'nin 2018 yılında yaptığı çalışmada, yerelleştirilmiş trafik işaretlerinin gerçek-zamanlı saptanması için çok-ölçekli evrişimsiz ağlar (*Multi-scale Deconvolutional Networks – MDN*) temelli bir model önerilmiştir. Model üç bileşenden oluşmaktadır: evrişimli artık ağ (*Convolutional Residual Network- CRN*), modifiye edilmiş nitelik piramit ağı (*Modified Feature Pyramid Network- MFPN*) ve çok-ölçekli sınıflayıcı ve saptayıcı (*multi-scale classifier and detector*). GTSRB veri seti yanında Çin Trafik İşaretleri Veri Seti (*Chinese Traffic Sign Dataset- CTSD*) ile de test edilen model, sınıflamada %99,1 başarıya ulaşmıştır [22]. Pei'nin tanıttığı model trafik işaretlerini gerçek-zamanlı işlemesi açısından KCVTS'ye benzemekle beraber, Pei'nin modeli MDN kullanırken KCVTS DNN tabanlı olan MobileNet kullanmaktadır.

CNN ve DNN tekniklerini kullanan çalışmalarda, trafik işaretlerinin saptanması ve sınıflandırılmasının iyileştirilmesi için, son zamanlarda sıklıkla YOLO (*You Look Only Once*) modelleri de tercih edilmektedir. Örneğin Yao vd. 2022 yılında yaptığı çalışmada, trafik işaretlerinin gerçek zamanlı işlenmesi amacıyla, geliştirilmiş bir YOLOv4-Tiny modeli önermektedir. YOLOv4-Tiny, YOLOv4 modelinin hafif-yapılı versiyonu olan tek-aşamalı bir sinir ağı modelidir. Bununla birlikte çalışma geliştirilmiş bir nitelik piramit ağı (Feature Pyramid Network – FPN) olan adaptif nitelik piramit ağını (Adaptive Feature Pyramid Network- AFPN) önermektedir. YOLOv4-Tiny tabanlı model GTSRB veri seti yanında, Çin Trafik İşaretleri Saptama Veri Seti (*Chinese Traffic Sign Detection Benchmark – CTSD*) ile de test edilmiştir. Model CTSD veri setinde %92,53, GTSRB veri setinde ise %87,6 oranında sınıflandırma başarısı yakalamıştır [23]. Yao'nun modelinin gerçek-zamanlı ve hafif-yapılı bir TİİS olması, KCVTS ile arasında ortak nokta teşkil etmektedir. Ek olarak Yao çalışmasında, önerdiği modelin mobil cihazlarda da etkinlik gösterme potansiyeline vurgu yaparken modelin uygulama alanı olarak MMS'i göstermemiştir.

Zhu ve Yan tarafından 2022 yılında yapılan çalışmada DNN ile birlikte YOLOv5 modeli kullanılmış ve araştırmacıların kendilerinin

oluşturduğu veri setinde %94 oranında bir sınıflandırma başarısı yakalanmıştır [18]. Çalışma, YOLOv5 modeli ile görüntülerde nesne tespiti (*object detection*) için tek bir derin sinir ağı (*Single Neural Network*) kullanan SSD'i (*Single Shot Multibox Detector*) karşılaştırmanın yanında, önerdikleri YOLOv5 modelinin ITS'deki kullanım potansiyeline dikkat çekmektedir. Dolayısıyla, YOLOv5 modeli ile KCVTS kullanım alanları ve uyguladıkları metotlar bakımından ayrılmaktadır.

Wu ve Cao tarafından 2022'de yapılan bir başka çalışma ise geliştirilmiş hafif-yapılı bir YOLOv4 modeli ile insansız sürüş sistemlerindeki (*Unmanned Driving Systems*) yüksek parametre sayısı, düşük gerçek-zaman performansı ve düşük isabetlilik oranları gibi problemleri çözmeyi amaçlamaktadır. MobileNetV2 ağını YOLOv4 ile birlikte kullanan model Tsinghua-Tencent 100K (TT100K) görüntü verilerinde %82,8 oranında bir sınıflandırma başarısı elde etmiştir [24]. Wu ve Cao'nun önerdiği model, MobileNetV2 ağını kullanması bakımından KCVTS ile benzeşmektedir. Öte yandan, araştırmacılar çalışmalarında YOLOv4 tabanlı modelin mobil cihazlarda MMS ile bütünleşmiş bir şekilde kullanılabilmesine yönelik bir atıfta bulunularak KCVTS'den ayrılmaktadırlar.

Alanda özellikle son yıllarda yapılan çalışmalar, uygun olmayan hava koşullarında da trafik işaretlerinin işlenmesine yönelik olarak yapılmıştır. Bu kapsamda yapılan çalışmalar, önceki çalışmalarda olduğu gibi, CNN ve diğer ML algoritmalarının bir arada kullanılması ile gerçekleştirilmiştir. Velamati 2021 yılında yaptığı çalışmada CNN tabanlı modelini Python kütüphaneleri olan TensorFlow (TF) ve OpenCV kullanarak GTSRB veri setinde yer alan karanlık, yağışlı vb. bozuk görüntüleri üzerinde %96 oranında bir sınıflandırma başarısı yakalamıştır [25]. Çalışmanın, TF kullanması KCVTS ile arasındaki ortak noktadır. Velamati'nin çalışmasında geliştirilen model, mobil haritalama amacıyla geliştirilmemiştir ve Velamati, modelin mobil cihazlarda uygulanabilirliğine bir atıfta bulunmamıştır.

Zhou ise 2021 yılında yaptığı çalışmasında HRNetv2p ve Libra-RCNN algoritmaları ile inşa edilen modeli, araştırmacıların kendilerinin oluşturduğu iki veri setinde test etmiştir. Bu veri setleri, Buzlu Çevre Trafik İşaretleri Tanıma-Saptama Değerlendirme Veri Seti (*Ice Environment Traffic Sign Recognition-Detection Benchmark- ITRB, ITSDB*), üzerinde test etmiş ve %94 oranında bir başarı ile trafik işaretlerini sınıflandırmıştır. Bahsi geçen veri setleri *Common Objects in Context* 2017 (COCO2017) formatında hazırlanmıştır [26]. KCVTS'nin de kullandığı hibrit veri seti COCO2017 veri setinin ağırlıklarını (*weight*) kullanmaktadır. Dolayısıyla Zhou'nun çalışmasında kullandığı yöntem ile KCVTS'nin yöntemi bu açıdan benzeşmektedir. Fakat, Zhou modelini mobil haritalama amaçlı olarak tanıtmamaktadır.

Batool'un 2021 yılında yaptığı bir çalışmada ise sisli, yağmurlu, bulanık hava koşullarında trafik işaretlerinin işlenmesine yönelik olarak DensNet, CNN, RPN (*Region Proposal Network*) ve A2PM (*Accurate Anchor Prediction Model*) algoritmaları kullanılarak iELMNet (*improved Extreme Learning Machine Network*) yöntemi geliştirilmiştir. Geliştirilen bu yöntem ile CURE-TSD veri setinde %98,6, GTSRB veri setinde %99,5 ve TT100K veri setinde %95,2 oranlarında sınıflandırma başarısı yakalanmıştır [27]. Geliştirilen yöntem, gerçek-zamanlı işleme yapabilmesi yönünden KCVTS'ye benzemektedir. Lakin Batool'un çalışmasında, model mobil haritalama amaçlı geliştirilmemiştir.

YOLO gibi modeller görünürlüğü düşük olduğu koşullarda trafik işaretlerinin işlenmesinde de yer almaktadır. Wan tarafından 2021 yılında yapılan gölgeli, karanlık, aşırı güneşli ve yağmurlu gibi görünürlüğü düşük olduğu hava koşullarında trafik ışıklarının

işlenmesine yönelik olarak Trafik İşaretleri- Yolo (*Traffic Sign-Yolo*, TS-Yolo) modelini önermiştir. YOLOv5 ve MixConv tabanlı TS-Yolo, GTSRB ve TT100K veri setleri üzerinde test edilmiştir. Testler sonucunda %75-83 oranında sınıflandırma başarısına ulaşılmıştır [28]. TS-Yolo hem kullanım alanı açısından hem de kullandığı yöntemler açısından KCVTS'den ayrılmaktadır.

İlgili çalışmalar değerlendirildiğinde KCVTS'nin, literatürde yer alan trafik işaretlerini işleme sistemlerinden birçok noktada farklı olduğu görülmektedir. Spesifik olarak mobil haritalama amaçlı yapılan çalışmalar kısıtlı sayıdadır [29]. OptaGIS'in tanıtıldığı Seifert'in çalışması bunlardan biridir. Öte yandan, çalışmalarda kullanılan modellerin bir kısmı gerçek-zamanlı işleme yapabilmektedir. Gerçek-zamanlı işleme yapabilen modellerden de sadece Yao'nun YOLOv4-Tiny tabanlı modeli hafif-yapılı olarak değerlendirilebilir. Dolayısıyla KCVTS; gerçek-zamanlı, mobil haritalama amaçlı ve hafif-yapılı olmasıyla birlikte tüm trafik işaretleri tanıma sistemleri içerisinde kendine özgün bir yer edinmektedir. Ayrıca, ilgili çalışmalar çeşitli veri setlerini kullanırken, bu çalışmada KCVTS'nin eğitim ve test sürecinde ITU Racing Driverless Türkiye Trafik İşaretleri Veri Seti'nde (TTVS) [29] yer alan görüntüler ile Türkiye'nin farklı yerlerinde sahadan toplanarak oluşturulan veri setindeki görüntüler kullanılmıştır. Bunun yanında, COCO2017 veri setinin ağırlıklarından yararlanılmıştır. KCVTS; TF ve TF Lite kütüphaneleri, MobileNetV1 MNAS FPN ve MobileNetV2 FPNLite modelleri ile inşa edilmiştir. Geliştirilen yöntemle akıllı telefon, tablet gibi mobil cihazlarda trafik işaretleri işlendikten sonra ilgili verinin anlık olarak merkez işlemciye aktarılmasıyla mobil haritalama süreci hızlandırılmıştır.

2. Materyal ve Yöntem (Material and Methodology)

Çalışma kapsamında yavaş, yüksek maliyetli ve iş gücü gerektiren geleneksel, yarı-geleneksel hatta YZ tabanlı trafik işaretlerini işleme yöntemlerine hızlı ve uygun maliyetli bir çözüm olarak KCVTS geliştirilmiştir. KCVTS, mobil cihazların kameralarıyla kaydedilen yol görüntüsünü trafik işaretlerini işlemek amacıyla yine mobil cihaz üzerinde MobileNet'e dayalı görüntü işleme teknikleriyle analiz eden ve bu yolla mobil haritalama yapan, harita verilerini sunuculara ileten bir yöntemdir. Bu kapsamda, KCVTS hem halka açık veri seti hem de araştırmacıların kendilerinin oluşturduğu veri seti ile trafik işaretlerini ait görüntüler ile eğitilmiş, daha sonra mobil cihaz üzerinde test edilmiş ve sonuçlar performans ölçümlerine göre ortaya konmuştur.

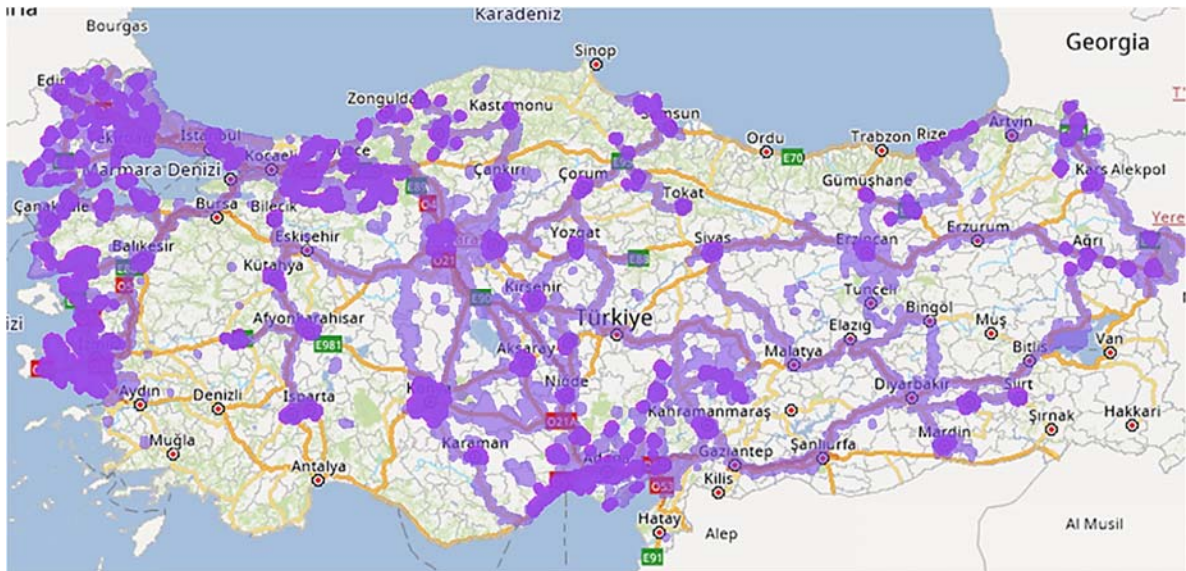
2.1. Veri ve Ön İşleme (Data and Preprocessing)

Modeller, COCO2017 veri seti ile önceden eğitilmiştir (*pre-trained*). Bu nedenle, modellerin eğitiminde COCO2017'nin ağırlıkları kullanılmıştır. Önceden eğitilmiş modeller, belirli görüntü özelliklerini saptama konusunda uzmanlaşmıştır ve bu özellikler, yeni bir sınıflandırma görevinde de kullanılabilir. Bu yaklaşım, eğitim aktarımı (*transfer learning*) olarak adlandırılır. Eğitim aktarımı ile COCO2017 veri seti ile önceden eğitilmiş bu görüntü sınıflandırma modellerinin, trafik işaretlerini sınıflandırma görevinde de kullanılması mümkün kılınmıştır. Böylece, daha az veriyle daha yüksek bir doğruluk oranı elde edilebilmektedir.

COCO veri seti, nesne tespiti, görüntü ayrımı (*image segmentation*) ve görüntü altyazısı (*image captioning*) oluşturma görevlerinde kullanılan büyük ölçekli bir görüntü tanıma (*image recognition*) veri setidir. 330,000'den fazla resim içerir, her resim 80 nesne kategorisiyle ve görüntüyü tasvir eden 5 altyazıyla etiketlenmiştir. Bu veri seti, bilgisayarlı görü araştırmalarında, nesne tespiti ve görüntü ayrımı modellerinde yaygın olarak kullanılmaktadır [29].

KCVTS'nin eğitim verisi temini kapsamında başlangıç olarak ITU Racing Driverless TTVS veri seti tercih edilmiştir. ITU Racing Driverless TTVS, ITU Racing Driverless ekibinin öncülüğünde, Türkiye'nin çeşitli üniversitelerinde etkinlik gösteren 30 farklı ekibin katkısı ile oluşturulan ve otonom araçlar üzerine çalışan, güncel olarak 32661 görüntü içeren bir veri setidir [30]. Veri setinden 9293 adet görüntü kullanılmıştır.

ITU Racing Driveless TTVS ve COCO2017 veri setlerinin yanında, araştırmacılar tarafından bir otomobile üç adet GoPro Hero 4 model kamera yerleştirilerek ikinci bir veri seti oluşturulmuştur. Kameralar, ileri bakacak şekilde bir tanesi ön kısma ve aracın iki yanına orta kamerayla yaklaşık 45° açı yapacak şekilde yerleştirilmiştir. Tercih edilen kamera 12 MP (4000 x 3000) sabit görüntü yakalayabilmektedir ve görüntüleri JPEG formatında depolamaktadır. Seyir esnasında her kamera saniyede 1 görüntü çekecek şekilde ayarlanmış ve bu şekilde elde edilen görüntüler operatörler tarafından Labellmg yazılımıyla, ITU Racing Driverless'in da kullandığı COCO2017 formatında etiketlenmiştir (veri toplanan alanlar için (Şekil 1)) [31]. Veri setinde boyutu 320x320 olan resimler için öncelikle tekrardan boyutlandırıp 640x640 haline getirme işlemi yapılmaktadır. Daha sonrasında, etiketleme yapılmadan önce görüntü setindeki bazı görüntülerin netliği kötü olmasından dolayı görüntü



Şekil 1. Veri Toplanan Alanlar (Data Collected Areas)

çözünürlüğünün iyileştirilmesi ve işaretleri daha net hale getirmek için *super resolution* algoritması kullanılmaktadır. Ardından ikinci veri setinin veri temizliği ve kontrolleri yapılmıştır. İkinci veri seti ile ITU Racing Driverless TTVS'den oluşan melez veri seti, eğitim ve test kümelerine dağıtılmıştır. 28627 görüntü eğitim, 2817 görüntü test için ayrılmıştır. Eğitim setinde 46394 etiket, test setinde 4986 etiket bulunmaktadır. Bu etiketler, 156 sınıfa ayrılmıştır. Sınıfların dağılıma ilişkin grafik Şekil 2'de verilmiştir.

2.2. Kitlesel Coğrafi Bilgi Toplama Sistemi- KCVTS (Crowdsourced Geographical Data Collection System (CGDCS))

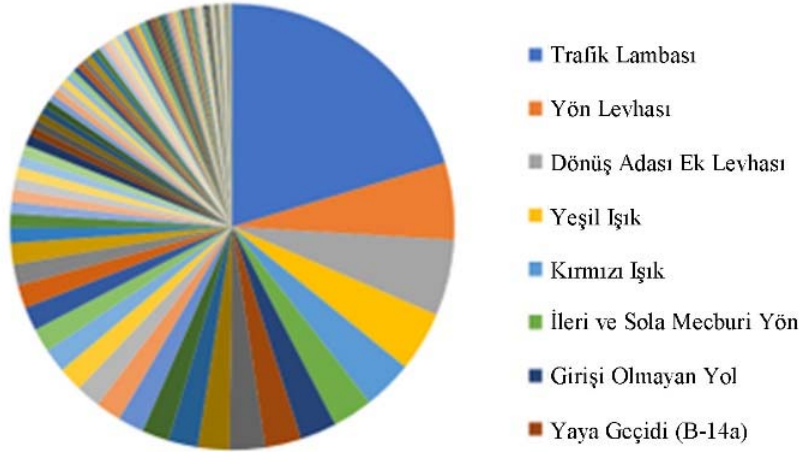
Çalışmada güçlü topluluk desteğiyle öne çıkan açık kaynak kodlu TF kütüphanesi ve mobil cihazlar için geliştirilmiş sürümü TF Lite kütüphanesi kullanılmıştır. Trafik işaretlerini işlemek için SSD MobileNetV1 MnasFPN 320x320 ve SSD MobileNetV2 FPNLite 640x640 modelleri tercih edilmiştir. Model bileşenleri Tablo 1'de yer almaktadır.

SSD MobileNetV1 MnasFPN 320x320, ana model olarak MobileNetV1 kullanılmaktadır. Modelin nesne tespit bileşenleri ise SSD ve MnasFPN yapılarıdır. Bunun yanında model, 320x320 şeklinde ölçeklendirilmiş görüntülerle eğitilmiş ve test edilmiştir. MobileNetV1, derinlemesine ayrılabilir evrişim (*Depthwise Separable Convolution - DSC*) tekniği ile karmaşıklığı (*complexity*) ve model boyutu (*model size*) azaltılmış, hafif-yapılı, mobil ve kaynakların kısıtlı olduğu ortamlarda kullanılmak üzere tasarlanmış bir CNN yapısıdır [32]. Bu yönüyle, trafik işaretlerini mobil ortamda işlemek için uygun bir tercihtir. MobileNetV1, modelde omurga ağı (*backbone network*) olarak kullanılmaktadır ve girdi (*input*) görüntülerin nitelik çıkarımından sorumludur. SSD ise tek bir derin sinir ağı kullanan bir nesne tespit metodudur. COCO gibi çeşitli veri setleri üzerinde gerçekleştirilen deneyler göstermiştir ki SSD, YOLO ve Faster R-CNN gibi tekniklere kıyasla daha hızlıdır ve rekabetçi oranlarda sınıflandırma başarısına sahiptir [33]. Bu sebeplerden dolayı, nesne tespit bileşeni olarak SSD tercih edilmiştir.

FPN ise nesne tespiti ve bilgisayarlı görü görevlerinde semantik segmentasyon için geliştirilen bir CNN yapısıdır. VGG ve ResNet gibi geleneksel CNN yapıları sabit ölçekli (*fixed scale*) görüntülerde işlev gösterirken FPN birden fazla ölçekte (*multi-scale*) işlev gösterebilmektedir [33]. FPN yapısının bir eklentisi olan MnasFPN ise nesne tespit modellerinin saptama uç kısmı (*detection head*) için tasarlanmış mobil-dostu bir arama uzayı (*search space*) ve yapı arama (*architecture search*) tekniğidir. MnasFPN arama uzayını, yapı arama teknikleriyle sistemli bir şekilde keşfeder, saptama uç kısmı için en verimli ve yüksek performanslı yapıyı bulur. MnasFPN'in araştırmacılar tarafından özellikle tercih edilmesinin sebebi, KCVTS'de kullanılacak mobil cihazlar gibi etkinlik göstereceği cihazın donanım özelliklerinin farkında olarak en optimize saptama uç kısmı yapısını bulabilmesidir. Dolayısıyla MnasFPN, mobil cihazlar gibi kaynakların kısıtlı olduğu cihazlarda en etkili seçimde bulunur [34]. Bu prensip ile SSD yapısındaki, saptama uç kısmının verimliliğini ve performansını artırır.

SSD MobileNetV2 FPNLite 640x640 modeli ise, ana model olarak MobileNetV2 kullanılmaktadır. Nesne tespit bileşenleri olarak da SSD ve FPNLite barındırır. Bununla birlikte model, 640x640 şeklinde ölçeklendirilmiş görüntülerle eğitilmiş ve test edilmiştir. MobileNet'in ikinci versiyonu olan MobileNetV2 ters artık yapısı (*inverted residual structure*) ile MobileNetV1'den ayrılmaktadır ve ImageNet sınıflandırması, COCO nesne tespiti ve VOC görüntü ayrımı gibi çeşitli görevlerde daha yüksek başarı göstermektedir [35]. Bu sebeple MobileNetV2, modelde omurga ağı olarak kullanılmaktadır ve girdi görüntülerin nitelik çıkarımından sorumludur. FPNLite ise FPN'nin hafif-yapılı varyantıdır. Mobil cihazlar gibi kaynakların kısıtlı olduğu ortamlarda kullanılmak üzere tasarlanmıştır ve bu yönü FPNLite'in tercih sebebi olmuştur.

SSD MobileNetV1 MnasFPN modelinde 320x320 ölçeğinin seçilmesinin sebebi ise, bu ölçeğin modelde en optimize şekilde çalıştığının gözlemlenmiş olmasıdır. Aynı şekilde SSD MobileNetV2



Şekil 2. Veri Sınıfları Dağılımı (Distribution of Data Classes)

Tablo 1. Model Bileşenleri (Model Components)

	SSD MobileNetV1 MnasFPN 320x320	SSD MobileNetV2 FPNLite 640x640
Ana Model (<i>Base Model</i>)	MobileNetV1	MobileNetV2
Nesne Tespit Bileşenleri (<i>Object Detection Components</i>)	SSD ve MnasFPN	SSD ve FPNLite
Ölçek (<i>Scale</i>)	320x320	640x640

FPNLite için de 640x640 ölçeği en optimize ölçek olduğundan seçilmiştir. Çalışma kapsamında seçilen modellerin diğer modellerle başarı oranı ve işlem süresi bakımından literatürdeki kıyaslama ölçütleriyle (*benchmarks*) karşılaştırıldığı görülmüş ve en optimize modellerin çalışma kapsamında seçilen modeller olduğu görülmüştür [36].

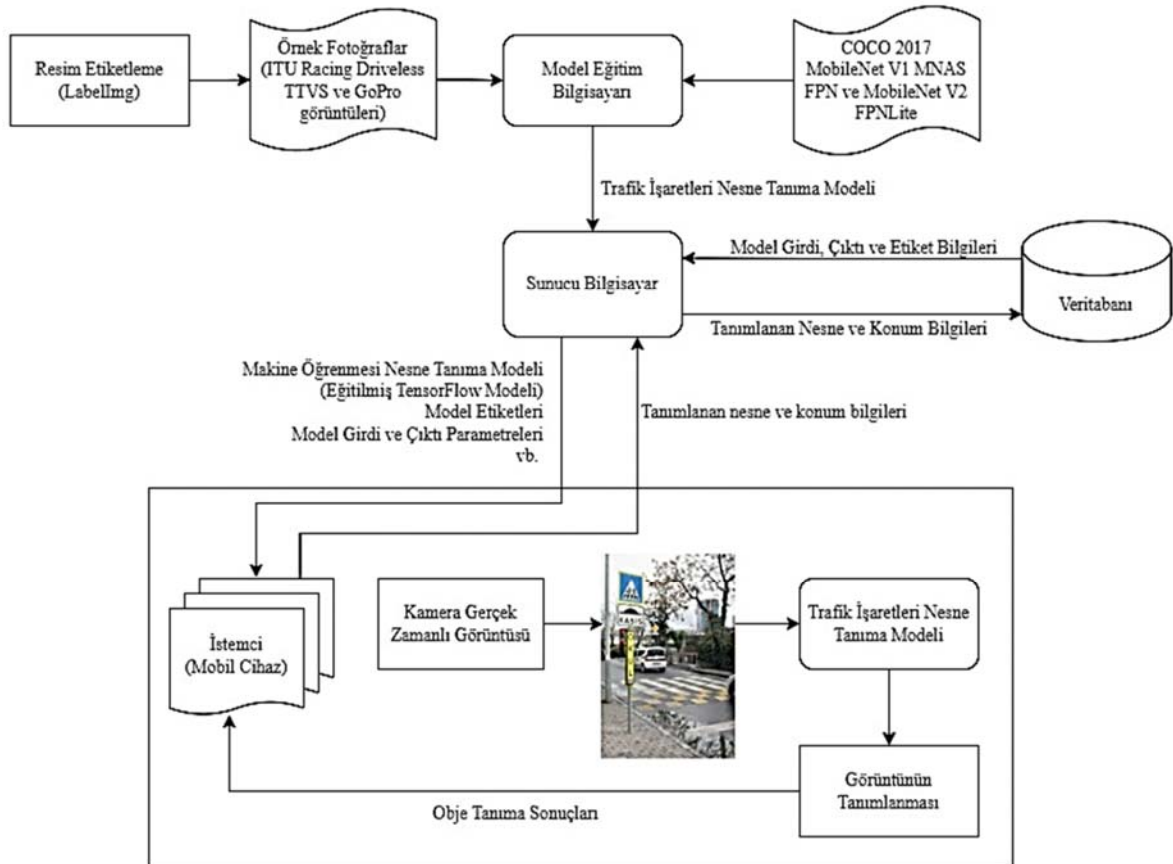
Jaccard indeksi (*birleşim üzerinde kesişim, Intersection over Union-IoU*), batch boyutu (*batch size*), öğrenme katsayısı (*learning rate*), warm-up öğrenme katsayısı (*warm-up learning rate*) hiperparametre olarak seçilip kullanılmıştır. Bu hiperparametre değerlerinde en yüksek performansı veren değerler: 0.6000000238418579 ile Jaccard indeksi ve 128 batch boyutu, 0.0799999821186066 ile öğrenme katsayısı ve 0.026666000485420227 ile warm-up öğrenme katsayısı olarak belirlenip kullanılmıştır.

Modellerin eğitiminde, LabelImg yazılımıyla COCO2017 formatında etiketlenen görüntülerden oluşan melez veri seti (ITU Racing Driverless TTVS + GoPro seti) kullanılmıştır. SSD MobileNetV1 MnasFPN 320x320 ve SSD MobileNetV2 FPNLite 640x640 modelleri, eğitim bilgisayarında COCO2017 ağırlıklarıyla eğitildikten sonra sunucu bilgisayara aktarılmıştır. Sunucu Bilgisayar (SB) hem veri tabanı hem de istemci (mobil cihaz) ile etkileşim halindedir. SB, istemciye eğitimli modeli sağlarken istemci de modeli çalıştırmaktadır. Model, kameradan gerçek-zamanlı elde edilen görüntüleri işlemekte, trafik işaretleri sınıf ve konum bilgilerini edinmektedir. Son olarak istemci, SB'ye sınıf ve konum bilgilerini aktarmaktadır. Bu şekilde, trafik işaretleri işleme görevi model tarafından mobil cihaz içerisinde gerçekleştirilir. Anlatılan süreç, KCVTS'in çalışma akışıdır. Süreci özetleyen diyagram Şekil 3'de görülebilir.

Performans ölçümlerinde, "*Precision*", "*Recall*", "*F1 Score*" ve "*Mean Average Precision (mAP)*" metrikleri kullanılmıştır. Metrikler, 0,50 ve üzeri *Intersection over Union (IoU)* değerlerinin görüldüğü örnekler baz alınarak hesaplanmıştır. Metriklerin veri bilimi açısından matematiksel ifadeleri Tablo 2'de bulunmaktadır.

Precision, saptanan trafik işaretlerinin fiziksel olarak var olup olmadığını bir ölçütükün *Recall*, mevcut tüm işaretlerin saptanıp saptanmadığının ölçüsüdür. *Precision* ve *recall* arasında ters orantılı bir ilişki vardır. Model nesne tespitte duyarlıysa, daha fazla *false positive* değerine sahip olacaktır ve dolayısıyla *recall* değeri düşecektir. Model, işaretlerin varlığına yeterince duyarlı değilse, o zaman daha fazla *false negative* değerine sahip olacaktır ve dolayısıyla *recall* daha yüksek bir değerde olacaktır. *Precision* ve *recall* arasındaki bu potansiyel dengesizlik senaryoları nedeniyle, *F1 Score* gözetilerek, *precision* ve *recall* arasında bir denge aranması tavsiye edilir [28, 37].

F1 Score, *precision* ve *recall* değerlerinin harmonik ortalamasıdır ve modelin sınıflandırma başarısının dengeli bir ölçüsü olarak değerlendirilmektedir. Farklı işaretler arasındaki *F1 Score* değişimi, modelin belirli trafik işaretlerini "ezberleyip ezberlemediğini" gösterebilir. Bu durum, eğitim veri setinde belirli bir işaretin daha yüksek temsilinin sonucu olabilir. Diğer bir önemli metrik, her bir trafik işareti sınıfın ortalama *precision* değerinin, sınıflar bakımından tekrar ortalaması alınarak hesaplanan *mAP* metriğidir. *mAP*, *precision-recall* eğrisinin altındaki alan olarak da ifade edilebilir. IoU metriği ise her bir görüntü örneği için tahmini sınırlayıcı kutu



Şekil 3. Sistem Mimarisi (System Architecture)

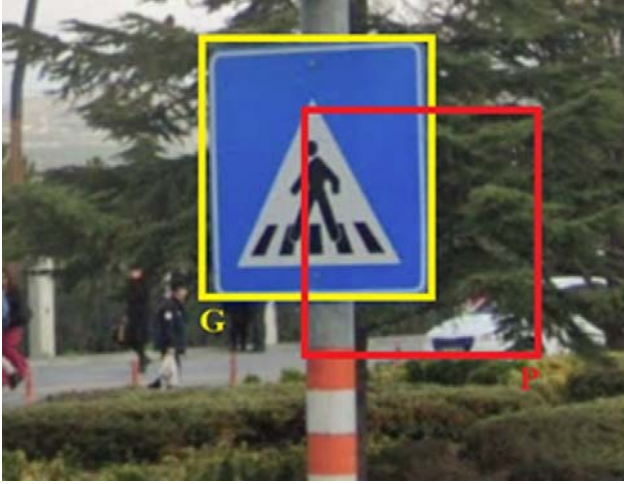
Tablo 2. Metrik İfadeleri (Metric Expressions)

Precision*	$\frac{tp}{tp + fp}$
Recall*	$\frac{tp}{tp + fn}$
F1 Score*	$2 \frac{precision \cdot recall}{precision + recall} = \frac{2tp}{2tp + fp + fn}$
Mean Average Precision (mAP)**	$\int_0^1 prec(rec)d(rec)$
Intersection over Union***	$IoU(G, P) = \frac{ G \cap P }{ G \cup P }$

* *tp*: true positive, *fp*: false positive
tn: true negative, *fn*: false negative

** *prec(rec)* precision-recall grafiği olmak üzere
*** *G*: tahmini sınırlayıcı kutu (*predicted bounding box*)
P: gerçek sınırlayıcı kutu (*ground-truth bounding box*)

(*predicted bounding box*) ile gerçek sınırlayıcı kutunun kesişim alanı ile birleşim alanının oranıdır (Şekil 4). Çalışmada, 0,50 ve üzeri IoU değerine sahip örnekler kullanılmıştır. Bu, yalnızca tahmini sınırlayıcı kutu ile gerçek sınırlayıcı kutu arasında %50'den fazla örtüşme olan tespitlerin dikkate alınacağı anlamına gelir [28, 37].



Şekil 4. Sınırlayıcı Kutular: *G* gerçek sınırlayıcı kutu, *P* tahmini sınırlayıcı kutu
(Bounding Boxes: *G*: Ground-truth bounding box, *P* predicted bounding box)

3. Bulgular (Results)

Önerilen bütünleşmiş metodolojinin sonuçları, *IoU*, *mAP*, *recall* ve *F1 Score* metriklerine göre ele alınmıştır. Şekil 5, test veri setinde toplanan görüntülerde saptanan işaretlerin örneklerini göstermektedir.

İlk 6 eğitim turunda SSD MobileNetV1 MnasFPN 320x320 modeli kullanılırken geriye kalan 7 eğitim turunda SSD MobileNetV2 FPNLite 640x640 modeli kullanılmıştır. 1. eğitim turunda trafik işaretleri sınıflarına göre 33 grupta birleştirilmiş, sonraki eğitim turlarında birleştirme yapılmadan eğitim gerçekleştirilmiştir ve metrikler o şekilde değerlendirilmiştir. Bazı trafik işareti sınıflarında çok az veri olmasına rağmen sınıflar birleştirilmediğinde (150-156 sınıf kullanıldığında) daha iyi sonuçlar alınmıştır. Bu değişimlerin izlenebilmesi amacıyla 1. eğitim turunda 33 sınıf, 2-7 eğitim turlarında 150 sınıf ve 8-13 eğitim turlarında 156 sınıf kullanılmıştır.

TF içerisinde, eğitim verilerinin artırılmasını amaçlayan bir aynalama işlevi bulunmaktadır. Lakin, tespit edilmesi beklenen trafik işaretlerinin bir kısmı (örn. “sağa dönülmez”) yönlü olduğu için TF

kütüphanesindeki varsayılan aynalama işlevinden olduğu gibi faydalanılmamıştır. Bunun nedeni, aynalama yapıldığında yönlü işaretlerin değişecek olmasıdır. Örneğin, “sağa dönülmez” işareti aynalama işleminden sonra “sola dönülmez” işareti olacaktır. Fakat, “sola dönülmez” (aynalanmış “sağa dönülmez”) işaretinin etiketi asıl görüntünün etiketi, yani “sağa dönülmez” olarak kalacaktır çünkü TF’deki aynalama işlemiyle etiketler değişmeyecek, sadece görüntüler aynalanacaktır. 3. eğitim turunda veri çoğaltmaya yönelik olarak aynalama için bir yaklaşım geliştirilmiş ve görüntüleri aynalarken etiketleri de yöne göre değiştirilmiştir. Yani, “sağa dönülmez” işareti önce aynalanmış ve “sola dönülmez” işaretine dönüşmüştür, sonra da “sola dönülmez” (aynalanmış “sağa dönülmez”) işaretinin etiketi “sola dönülmez” olarak değiştirilmiştir. Yönlü olmayan tabelaların aynalanmış görüntülerine ait etiketler değiştirilmemiştir. Yapılan bu değişikliklerle sonuç az da olsa iyileştirilmiştir.

4. eğitim turunda veri miktarı artırılmıştır. Bunun ardından, *recall* önemli ölçüde iyileşirken *precision* değerinin kötüleştiği görülmüştür. 5 ve 6. eğitim turlarında veri miktarı artırılmaya devam edilmiş ve daha dengeli *precision* ve *recall* değerleri elde etmek için güven eşiği (*confidence threshold*) değeri artırılmıştır. 7. eğitim turundan itibaren SSD MobileNetV2 FPNLite 640x640 modeli kullanılmıştır. 8., 9. Ve 10. Eğitim turlarında daha fazla veri eklenip aynalama yapılmadan eğitim yapılmıştır. *F1 Score* değeri iyileşirken *mAP* değerinin önemli ölçüde düştüğü görülmüştür. 11. eğitim turunda sadece 200 adetten az örneği bulunan sınıfların görüntüleri aynalanarak görüntü çoğaltma ile eğitim yapılmıştır. 12 ve 13. eğitim turlarında veri eklenmesi yapılmıştır. Verilerin artırılması ve karmaşık örneklerin eklenmesi ile beklenenin aksine modelde iyileşme gerçekleştirilmemiş, eğitim veri setinde daha az bulunan ve tanınması kolay olan karakteristik trafik işaretleri için öncesinde daha yüksek olan *precision* değerinin düşmesine sebep olmuştur. Dolayısıyla gelen model bazında *mAP* değeri de düşmüştür. Eğitim turları sonucunda elde edilen nesne tespit modeli performans verileri Tablo 2 ve Şekil 6’da verilmiştir.

SSD MobileNetV1 MnasFPN 320x320 modeli ile akıllı telefonda gerçekleştirilen testte bir trafik işaretinin ortalama tespit süresi 221ms, SSD MobileNetV2 FPNLite 640x640 modeli ile aynı cihazda gerçekleştiren testte ise ortalama tespit süresi 798ms olarak ölçülmüştür. SSD MobileNetV1 MnasFPN 320x320 modelinin, SSD MobileNetV2 FPNLite 640x640 modelinden yaklaşık 4 kat daha hızlı olduğu anlaşılmıştır. Ayrıca, SSD MobileNetV1 MnasFPN 320x320 kullanan KCVTS’in bir günde 2500’ün üzerinde trafik işaretinin konum ve sınıf bilgilerini veri tabanına aktardığı belirlenmiştir. Son olarak, KCVTS’in bir günde sunucu bilgisayara aktardığı ilgili sınıf-konum verilerinin ortalama boyutunun 79 MB olduğu ölçülmüştür.

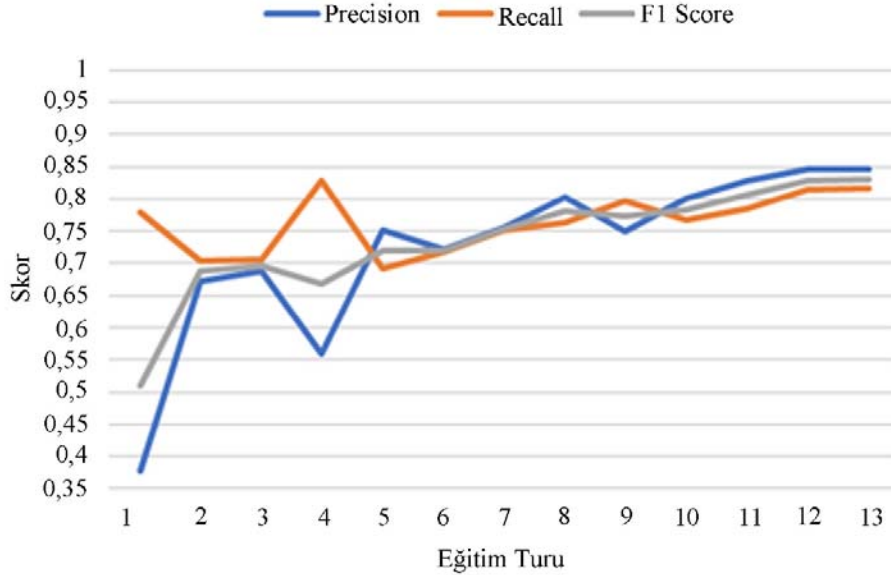


Şekil 5. İşaret Saptama (Sign Detection)

Tablo 3. Nesne Tespit Modeli Performans Değerleri (Object Detection Model Performance Values)

Eğitim	Sınıf Sayısı	Güven Eşik Değeri	Precision	Recall	F1 Score	mAP	Model
1	33	0,15	0,378492	0,778740	0,5094		SSD MobileNetV1 MnasFPN 320x320
2	150	0,15	0,672833	0,704288	0,6882		SSD MobileNetV1 MnasFPN 320x320
3	150	0,15	0,687431	0,706231	0,6967		SSD MobileNetV1 MnasFPN 320x320
4	150	0,15	0,559889	0,829122	0,6684		SSD MobileNetV1 MnasFPN 320x320
5	150	0,23	0,751436	0,692036	0,7205		SSD MobileNetV1 MnasFPN 320x320
6	150	0,23	0,721165	0,717853	0,7195		SSD MobileNetV1 MnasFPN 320x320
7	150	0,23	0,754577	0,750284	0,7524	56,6	SSD MobileNetV2 FPNLite 640x640
8	156	0,23	0,802428	0,762214	0,7818	52,3	SSD MobileNetV2 FPNLite 640x640
9	156	0,20	0,749391	0,797686	0,7728		SSD MobileNetV2 FPNLite 640x640
10	156	0,23	0,800912	0,767423	0,7838		SSD MobileNetV2 FPNLite 640x640
11	156	0,23	0,827865	0,785209	0,8060	58,4	SSD MobileNetV2 FPNLite 640x640
12	156	0,23	0,845567	0,813690	0,8293	60,3	SSD MobileNetV2 FPNLite 640x640
13	156	0,23	0,845316	0,816140	0,8305	59,8	SSD MobileNetV2 FPNLite 640x640

Nesne Algılama Modeli Performans Değerleri



Şekil 6. Nesne Tespit Modeli Performans Sonuçları (Object Detection Model Performance Results)

4. Sonuçlar ve Tartışmalar (Results and Discussions)

Bu çalışmanın amacı, makine öğrenmesi (ML) ve sinir ağları (NN) alanındaki son gelişmelerden yararlanarak mobil haritalama sistemlerine bütünleşmiş, daha hızlı ve pratik bir yaklaşım geliştirmektir. Çalışmada tanıtılan KCVTS, en gelişmiş CNN yapılarının trafik işaretlerinin işlenmesi görevinde kullanımına ve bu

CNN yapılarının MobileNet kullanarak mobil cihazlara taşınmasına bir örnek oluşturmaktadır. KCVTS, geriye kalan YZ tabanlı TIİS'lerinden birçok noktada ayrılmaktadır. Öncelikle, bu YZ tabanlı uygulamaların büyük bir kısmı mobil haritalama sistemlerine yönelik geliştirilmemiştir. KCVTS bu açığı kapatmak amacıyla, tamamen mobil haritalama görevine odaklanmış bir sistemdir. Bu açıdan, tüm TIİS uygulamaları arasında spesifik olarak mobil haritalama amaçlı

çalışan bir azınlığı temsil etmektedir. Geliştirilen TIİS sistemlerinin çok az bir kısmı mobil cihazlarda gerçek-zamanlı etkinlik göstermektedir. YZ tabanlı uygulamaların çoğunun, mobil cihazlarda etkinlik gösterememesinin sebebi mobil cihazların, kaynak-kısıtlı (*resource-constrained*) ortamlar sunmasıdır. Fakat ağır-yapılı olan CNN gibi derin öğrenme uygulamaları, kaynak-kısıtlı ortamlarda yeterince etkinlik gösterememektedir. Çalışma kapsamında bu sorun, hafif-yapılı CNN yapısıyla ve mobil cihazlar gibi kaynak-kısıtlı ortamlarda kullanılabilmesiyle öne çıkan MobileNet'i modellerinde tercih edilerek aşılmıştır. Bu yönünden güç alan KCVTS, mobil cihazlarda gerçek-zamanlı etkinlik gösterebilmekte ve kısa sürelerde trafik işaretlerini tespit edip sınıflandırabilmektedir.

Literatürdeki çalışmaların birçoğunun önerdiği modellerin, kırılmış görüntülerde test edildiği görülmüştür. Modeller, kırılmış görüntülerde daha yüksek bir başarı sınıflandırması göstermektedir. Bu durumun sebebi, kırılmış görüntülerin neredeyse sadece bir adet trafik işaretini içermesidir. Dolayısıyla modelin trafik işaretlerini, örneğin bir otoyol ya da sokak görüntüsünde, tespit etmesine (*detection*) gerek kalmamaktadır. Model, tespit yapmaksızın direkt sınıflandırma (*classification*) yapabilmektedir. Öte yandan KCVTS, gerçek-zamanlı kamera verileri ile eğitilmiş ve test edilmiştir. Bu veriler, örneğin bir otoyolun ya da sokağın geniş açılı bir görüntüsüdür ve sadece trafik işaretleri değil, birçok başka nesne de bulundurmaktadır. Dolayısıyla, KCVTS önce trafik işaretlerini tespit etmekte, sonrasında sınıflandırmaktadır. Bu görev, diğer TIİS uygulamalarının yerine getirdiği “tek bir trafik işaretinin bulunduğu görüntülerden işaretin sınıflandırılması” görevinden daha zorlu bir görevdir. Buna rağmen, KCVTS performans ölçümlerinde kayda değer bir sınıflandırma başarısı göstermiştir. Sonuç olarak, KCVTS dışındaki modellerin azımsanmayacak bir kısmının daha yüksek sınıflandırma başarıları göstermelerinin sebebi KCVTS'e göre “daha kolay” bir görevi yerine getirmeleridir. Bu kapsamda, elde edilen sonuçlar TF'nin COCO veri seti üzerinde yaptığı kıyaslama ölçütü (*benchmark*) sonuçları karşılaştırılmıştır. Bu sonuçlara göre Speed (ms) ve mAP (*Mean Average Precision*) değerleri incelendiğinde en optimize algoritmanın MobileNet v2 FPNLite 640x640 olduğu görülmüştür. MobileNet v2 FPNLite 640x640 algoritmasında hız (*Speed*) değeri 39 ms, mAP değeri ise 28,2 olarak elde edilmiştir [36].

KCVTS çalışması hazırlanırken Türkçe literatürde, trafik işaretlerini işlemeye yönelik çalışmaların kısıtlı sayıda olduğu fark edilmiştir. Bu açıdan KCVTS çalışması, Türkçe literatüre de bir katkı olarak değerlendirilebilir. Bu çalışmayla birlikte KCVTS, mobil haritalama amaçlı trafik işaretlerini gerçek-zamanlı işleme sistemleri arasında yerini almıştır. İlerleyen çalışmalarda KCVTS'in sınıflandırma başarısını artırmak ve KCVTS'i, başta ADAS ve ITS gibi çeşitli alanlara entegre etmek üzere çalışılabilir. Bunun yanında; trafik işaretleri dışında kalan çukur, kasis, su birikme noktaları gibi yoldaki anomalilere ilişkin bilgilerin yine KCVTS tarafından toplaması ve işlenmesi de bir araştırma konusu niteliğindedir.

Teşekkür (Acknowledgement)

Bu çalışma Türkiye Bilimsel ve Teknolojik Araştırma Kurumu (TÜBİTAK) 1505 programı 3200751 numaralı proje ve 2244 programı 119C200 numaralı proje ile desteklenmiştir.

Kaynaklar (References)

1. Arcos-García Á., Álvarez-García J. A., Soria-Morillo L. M., Evaluation of deep neural networks for traffic sign detection systems, *Neurocomputing*, 316, 332-344, 2018.
2. Salti S., Petrelli A., Tombari F., Fioraio N., Di Stefano L., Traffic sign detection via interest region extraction, *Pattern Recognit.*, 48 (4), 1039-1049, 2015.

3. Qiu Z., Martínez-Sánchez J., Brea V. M., López P., Arias P., Low-cost mobile mapping system solution for traffic sign segmentation using Azure Kinect, *Int. J. Appl. Earth Obs. Geoinformation*, 112 (102895), 2022.
4. Timofte R., Zimmermann K., Van Gool L., Multi-view traffic sign detection, recognition, and 3D localisation, *Workshop on Applications of Computer Vision (WACV)*, Snowbird, UT, USA: 1-8, 2009.
5. Li R., *Mobile Mapping: An Emerging Technology for Spatial Data Acquisition*, The Map Reader, John Wiley & Sons, Ltd, NJ, ABD, 170-177, 2011.
6. Tao C., *Mobile Mapping Technology for Road Network Data Acquisition*, *Journal of Geospatial Engineering*, 2 (2), 1-13, 2001.
7. Frenzos E., Tournas E., Skarlatos D., *Developing An Image Based Low-Cost Mobile Mapping System for GIS Data Acquisition*, *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.*, XLIII-B1-2020, 235-242, 2020.
8. Kim G.-H., Sohn H.-G., Song Y.-S., *Road Infrastructure Data Acquisition Using a Vehicle-Based Mobile Mapping System*, *Comput.-Aided Civ. Infrastruct. Eng.*, 21 (5), 346-356, 2006.
9. Manandhar D., Shibasaki R., *Vehicle-borne laser mapping system (VLMS) for 3-D GIS, IGARSS 2001. Scanning the Present and Resolving the Future. Proceedings. IEEE 2001 International Geoscience and Remote Sensing Symposium, Sydney, NSW, Australia, 2073-2075, 2001.*
10. El-Halawany S. I., Lichti D. D., *Detecting road poles from mobile terrestrial laser scanning data*, *GIScience Remote Sens.*, 50 (6), 704-722, 2013.
11. Kumar P., McElhinney C. P., Lewis P., McCarthy T., *Automated road markings extraction from mobile laser scanning data*, *Int. J. Appl. Earth Obs. Geoinformation*, 32, 125-137, 2014.
12. D. Barber, J. Mills, ve S. Smith-Voysey, “Geometric validation of a ground-based mobile laser scanning system”, *ISPRS J. Photogramm. Remote Sens.*, 63 (1), 128-141, Oca. 2008.
13. Hammoudi K., Dornaika F., Paparoditis N., *Extracting Building Footprints From 3d Point Clouds Using Terrestrial Laser Scanning at Street Level*, *ISPRS Workshop on City Models, Roads and Traffic (CMRT)*, Paris, Fransa, 65-70, 3-4 Eylül 2009.
14. Yiğit A. Y., Hamal S.N.G., Ulvi A., Yakar M., *Comparative analysis of mobile laser scanning and terrestrial laser scanning for the indoor mapping*, *Build. Res. Inf.*, 1-16, 2023.
15. Yiğit A.Y., Hamal S.N.G., Yakar M., Ulvi A., *Investigation and Implementation of New Technology Wearable Mobile Laser Scanning (WMLS) in Transition to an Intelligent Geospatial Cadastral Information System*, *Sustainability*, 15(9), 7159, 2023.
16. He Z., Nan F., Li X., Lee S.-J., Yang Y., *Traffic sign recognition by combining global and local features based on semi-supervised classification*, *IET Intell. Transp. Syst.*, 14 (5), 323-330, 2020.
17. Zhu Y., Yan W. Q., *Traffic Sign Recognition Based on Deep Learning Technique*, *Multimed Tools Appl*, 81, 17779-17791 2022.
18. De La Escalera A., Armingol J. M., Mata M., *Traffic sign recognition and analysis for intelligent vehicles*, *Image Vis. Comput.*, 21 (3), 247-258, 2003.
19. Seifert C. Paletta L., Jeitler A., Hödl E., Andreu J.P., Luley P., Almer A., *Visual object detection for mobile road sign inventory*, *Lect. Notes Comput. Sci. Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinforma.*, 3160, 491-495, 2004.
20. Arcos-García, J. A. Álvarez-García Á., Soria-Morillo L. M., *Deep neural network for traffic sign recognition systems: An analysis of spatial transformers and stochastic optimisation methods*, *Neural Netw.*, 99 (January), 158-165, 2018.
21. Nuakoh E. B., Roy K., Yuan X., Esterline A., *Deep learning approach for U.S. traffic sign recognition*, *ICDLT '19: Proceedings of the 2019 3rd International Conference on Deep Learning Technologies NY, ABD*, 47-50, 2019.
22. Pei S., Tang F., Ji Y., Fan J., Ning Z., *Localized Traffic Sign Detection with Multi-scale Deconvolution Networks*, *Proc. - Int. Comput. Softw. Appl. Conf.*, 1, 355-360, 2018.
23. Yao Y., Han L., Du C., Xu X., Jiang X., *Traffic sign detection algorithm based on improved YOLOv4-Tiny*, *Signal Process. Image Commun.*, 107 (116783), 2022.
24. Wu X., Cao H., *Traffic Sign Detection Algorithm Based on Improved YOLOv4*, *J. Phys. Conf. Ser.*, 2258 (1), 2022.

25. Velamati A., Gopichand G., Traffic Sign Classification Using Convolutional Neural Networks and Computer Vision, Turk. J. Comput. Math. Educ., 12 (3), 4244-4250, 2021.
26. Zhou K., Zhan Y., Fu D., Learning region-based attention network for traffic sign recognition, Sens. Switz., 21 (3), 1-21, 2021.
27. Batool A., Nisar M. W., Hussain Shah J., Rehman A., Sadad T., IELMNet: An Application for Traffic Sign Recognition using CNN and ELM, 1st Int. Conf. Artif. Intell. Data Anal. CAIDA Riya, Suudi Arabistan, 132-137, 6-7 Nisan 2021.
28. Wan H., Gao L., Su M., You Q., Qu H., Sun Q., A Novel Neural Network Model for Traffic Sign Detection and Recognition under Extreme Conditions, J. Sens., 2021, 2021.
29. T.-Y. Lin vd., Microsoft COCO: Common Objects in Context. arXiv, <http://arxiv.org/abs/1405.0312>, 20 Şubat 2015, 18 Temmuz 2023.
30. ITU Racing Driverless, TTVS (Türkiye Trafik Veri Seti) <https://github.com/ituracingdriverless/TTVS>, 03 Nisan 2023, 18 Temmuz 2023.
31. GitHub, LabelImg Graphical Image Annotation Tool, <https://github.com/heartexlabs/labelImg>, 2018, 18 Ekim 2022.
32. Howard A. G., MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. arXiv, <https://arxiv.org/abs/1704.04861> 16 Nisan 2017, 18 Ekim 2022.
33. Liu W., SSD: Single Shot MultiBox Detector, Computer Vision – ECCV 2016. ECCV 2016. Lecture Notes in Computer Science, 9905, 21-37, 2016.
34. Chen B. vd., MnasFPN: Learning latency-aware pyramid architecture for object detection on mobile devices, Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., 13604-13613, 2020.
35. Sandler M., Howard A., Zhu M., Zhmoginov A., Chen L. C., MobileNetV2: Inverted Residuals and Linear Bottlenecks, Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., 4510-4520, 2018.
36. GitHub, TensorFlow 2 Detection Model Zoo, GitHub, https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/tf2_detection_zoo.md, 2021, 13 Temmuz 2023.
37. Kargah-Ostadi N., Waqar A., Hanif A., Automated Real-Time Roadway Asset Inventory using Artificial Intelligence, Transp. Res. Rec., 2674 (11), 220-234, 2020.

