MARMARA ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

DERİN ÖĞRENME YÖNTEMLERİ İLE TRAFİK İŞARETİ TANIMA

OKAN YILDIRAN

YÜKSEK LİSANS TEZİ

Bilgisayar Mühendisliği Anabilim Dalı

Bilgisayar Mühendisliği Yüksek Lisans Programı

DANIŞMAN

Dr. Öğr. Üyesi Kazım YILDIZ

İSTANBUL, 2019

MARMARA ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

DERİN ÖĞRENME YÖNTEMLERİ İLE TRAFİK İŞARETİ TANIMA

OKAN YILDIRAN

523616037

YÜKSEK LİSANS TEZİ

Bilgisayar Mühendisliği Anabilim Dalı

Bilgisayar Mühendisliği Yüksek Lisans Programı

DANIŞMAN

Dr. Öğr. Üyesi Kazım YILDIZ

İSTANBUL, 2019

MARMARA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

Marmara Üniversitesi Fen Bilimleri Enstitüsü Yüksek Lisans Öğrencisi Okan YILDIRAN'ın "**Derin Öğrenme Yöntemleri ile Trafik İşareti Tanıma**" başlıklı tez çalışması, 25 Eylül 2019 tarihinde savunulmuş ve jüri üyeleri tarafından başarılı bulunmuştur.

Jüri Üyeleri

Dr. Öğr. Üyesi Kazım YILDIZ (Danışman)
Marmara Üniversitesi Bilgisayar Mühendisliği Bölümü .....

Doç. Dr. Ali BULDU (Üye)
Marmara Üniversitesi Bilgisayar Mühendisliği Bölümü .....

Doç. Dr. Turgay Tugay BİLGİN (Üye)
Bursa Teknik Üniversitesi Bilgisayar Mühendisliği Bölümü .....

ONAY

Marmara Üniversitesi Fen Bilimleri Enstitüsü Yönetim Kurulu'nun 02/10/2019 tarih ve 2019/20-02 sayılı kararı ile Okan YILDIRAN'ın Bilgisayar Mühendisliği Anabilim Dalı Bilgisayar Mühendisliği Programında Yüksek Lisans derecesi alması onanmıştır.

Fen Bilimleri Enstitüsü Müdürü
Prof. Dr. Bülent EKİCİ

ÖNSÖZ

Bu tez çalışmasında otonom araçlar için büyük öneme sahip trafik işareti tespiti ve sınıflandırması üzerine farklı yöntemler incelenmiş, farklı verisetleri üzerinde uygulanarak karşılaştırılmıştır. Bu alanda kullanılmak üzere Türkiye'de toplanan görüntüler ile farklı hava koşullarını içeren bir veriseti oluşturulmuştur.

Tez çalışmamın araştırılmasında ve yürütülmesinde desteğini esirgemeyen, bana her fırsatta yardımcı olan değerli danışmanım Dr. Öğr. Üyesi Kazım YILDIZ'a teşekkürlerimi sunarım.

Eylül, 2019 **Okan YILDIRAN**

İÇİNDEKİLER

	SAYFA
ÖNSÖZ	**i**
İÇİNDEKİLER	**ii**
ÖZET	**v**
ABSTRACT	**vi**
SEMBOLLER	**vii**
KISALTMALAR	**viii**
ŞEKİL LİSTESİ	**ix**
TABLO LİSTESİ	**xi**
1. GİRİŞ	**1**
1.1 Trafik İşaretleri	2
1.2 Araçların Dijitalleşmesi	**3**
1.3 Verinin İşlenmesi	**4**
1.4 Society of Automotive Engineers (SAE) Seviyeleri	**5**
1.5 Görüntü İşleme Yöntemlerine Dayalı Yaklaşım	**6**
1.6 Derin Öğrenmeye Dayalı Yaklaşım	**7**
1.7 Veri Setleri	**7**
1.8 Amaç ve Önem	**9**
1.9 Literatür Araştırması	**9**
1.9.1 Tespit Aşaması için Yaklaşımlar	10
1.9.1.1 Renk Filtreleme	11
1.9.1.2 Seçici Arama	12
1.9.2 Sınıflandırma Aşaması için Yaklaşımlar	12
1.9.2.1 SVM Kullanarak Sınıflandırma	13

1.9.2.2 Yapay Sinir Ağı Kullanarak Sınıflandırma 13

1.9.3 Derin Öğrenme Tabanlı Yaklaşımlar 15

2. MATERYAL VE YÖNTEM 18

2.1 Kullanılan Veri Setleri 18

2.1.1 Veri Toplama 21

2.1.2 İşaretleme 21

2.2 Yapay Veri Üretimi 22

2.3 Kullanılan Araçlar 23

2.4 Görüntü İşleme Tabanlı Yaklaşım Uygulanması 23

2.4.1 RGB - HSV Dönüşümü 24

2.4.2 HSV Renk Filtresi 24

2.4.3 Medyan Filtre 26

2.4.4 Morfolojik Açma İşlemi 26

2.4.5 Piksel Kümeleri Tespiti ve Filtresi 27

2.5 Derin Öğrenme Yaklaşımı Uygulanması 29

2.5.1 Sınırlayıcı Kutu Bulucu 29

2.5.1.1 Kayıp Fonksiyonu 31

2.5.2 Görüntü Çoğullama 34

2.5.2.1 En İyileme 35

2.5.2.2 Azami Baskılama (Non-Maximum Supression) 36

2.5.2.3 Yapay Sinir Ağı Modeli 36

2.5.2.4 Modelin Eğitimi 38

2.5.3 Sınıflandırma Modeli 38

3. BULGULAR VE TARTIŞMA 39

3.1 Görüntü İşleme ile Sınırlayıcı Kutu Tespiti Performansı 39

3.2 Derin Öğrenme ile Sınırlayıcı Kutu Tespiti Performansı **40**

3.3 Derin Öğrenme ile Sınıflandırma Modeli Performansı **43**

3.4 YOLO Modeli Performansı **44**

4. SONUÇLAR **45**

KAYNAKLAR **47**

ÖZGEÇMİŞ **52**

ÖZET

DERİN ÖĞRENME YÖNTEMLERİ İLE TRAFİK İŞARETİ TANIMA

Gelişen araç teknolojileri ile birlikte sürücüye yardımcı olacak sistemler ön plana çıkmaktadır. Bu sistemler sürüş emniyetini ve konforunu artırmak için gün geçtikçe daha fazla araca entegre edilmektedir. Sürücüye yardımcı olan bu sistemler bir yandan da otonom araçlar için önemli bir altyapıyı oluşturmaktadır. Otonom araçların emniyetli bir şekilde kullanılabilir olması için etrafındaki tüm bilgiyi elde edebilen, bu bilgileri doğru bir şekilde yorumlayan ve karar alabilen sistemler olması gerekmektedir. Otonom araçların kullanabileceği en önemli bilgilerden biri ise sürüş sırasında karşılaşılan trafik işaretlerinin içeriğidir.

Trafik işaretlerinin tespit edilebilmesi için araç üzerinde bulunan bir kamera ile çevrenin görüntüsünün alınması gerekmektedir. Bu işaretlerin görüntü üzerinden tespit edilmesi, görüntü işleme ve makine öğrenmesi alanları için zorlayıcı bir problemdir. En büyük problemlerden biri ise veri setlerinin yetersiz olmasıdır. Bu noktada Türkiye'deki yollardan elde edilen bir görüntü ve video veriseti bulunmamaktadır. Bu tezde, Türkiye'de toplanan görüntü ve videolar ile trafik işareti veriseti oluşturulmuştur. Bu verisetinde farklı hava koşullarında günün farklı zamanlarında (gündüz-gece) elde edilen görüntüler bulunmaktadır.

Bu tezde sürücü yardım sistemlerinde ve otonom araçlarda kullanılmak üzere, derin öğrenme ve görüntü işleme yöntemleri ile eğitilmiş, verilen görüntü üzerinden trafik işaretlerinin konumunu ve içeriğini gerçek zamanlı olarak tespit edebilen bir sistem gerçeklenmiştir.

Eylül, 2019 **Okan YILDIRAN**

ABSTRACT

TRAFFIC SIGN DETECTION USING DEEP LEARNING

Advancing vehicle technology increases the interest of the systems which help to drivers. These systems are being integrated to more vehicles everyday to make better safety and comfort while driving. Such systems also make an important infrastructure for autonomous vehicles. To make safer autonomous vehicles; these systems have to collect all information around itself, interpret these information correctly and make decisions according to them. One of the most important information which autonomous vehicle can use is traffic sign information while driving.

For detecting traffic signs, images of the environment are need to be taken from a camera on the vehicle. Acquiring these signs from the images is a challenging problem for image processing and machine learning fields. One of the most challenging problem is the lack of datasets that containts various traffic signs. At this point, there is no dataset of images and videos that is collected in Turkey's roads. In this project it is a dataset that is collected in Turkey is built. In this dataset, there are images from different times (day-night) and different weather conditions.

In this project developed a system that trained with deep learning and image processing methods, detects traffic signs in real time and to be used by driver assistant systems and autonomous systems.

September, 2019 **Okan YILDIRAN**

SEMBOLLER

μ	:	Ortalama
λ	:	Sabit katsayı
$\sum$	:	Toplam
1^{obj}	:	Nesne bulunma durumu
1^{noobj}	:	Nesne bulunmama durumu
p	:	Olasılık

KISALTMALAR

CNN	**:**	Convolutional Neural Networks
GTSDB	**:**	German Traffic Sign Detection Benchmark
HOG	**:**	Histogram of Oriented Gradients
HSV	**:**	Hue, Saturation, Value
IOU	**:**	Intersection over Union
LISA	**:**	Laboratory for Intelligent & Safe Automobiles
mAP	**:**	Mean Average Precision
RCNN	**:**	Region - Convolutional Neural Networks
RGB	**:**	Red, Green, Blue
ROI	**:**	Region of Interest (İlgi alanı)
RTSD	**:**	Russian Traffic Sign Dataset
SSD	**:**	Single Shot Detector
SVM	**:**	Support Vector Machine
TRTSD	**:**	Turkish Traffic Sign Dataset
TS	**:**	Traffic Sign
YOLO	**:**	You Only Look Once

ŞEKİL LİSTESİ

SAYFA

Şekil 1.1. Çeşitli Avrupa ülkelerindeki trafik işaretleri örnekleri. 2
Şekil 1.2. Trafik uyarı işaretlerine örnekler. 3
Şekil 1.3. Trafik tanzim işaretlerine örnekler. 3
Şekil 1.4. Trafik bilgi işaretlerine örnekler. 3
Şekil 1.5. Trafik işaretinin tespitini zorlaştıran etkenlere ait örnekler. 4
Şekil 1.6. Trafik işaretinin kameradan elde edilen görüntü ile algılanması. 5
Şekil 1.7. Görüntü işleme yöntemleri uygulanmış trafik işaretleri. 6
Şekil 1.8. Özellik çıkarımı ve sınıflandırma yapan derin öğrenme modeli diyagramı. 7
Şekil 1.9. BelgiumTS verisetinden örnek görüntüler. 8
Şekil 1.10. GTSDB verisetinden örnek görüntüler. 8
Şekil 1.11. RTSD verisetinden örnek görüntüler. 8
Şekil 1.12. Traffic Signs verisetinden örnek görüntüler. 9
Şekil 1.13. Trafik işareti tanıma problemi için literatürde var olan yöntemler. 10
Şekil 1.14. Renk filtreleme ve görüntü işleme yöntemleri ile trafik işareti tespiti 11
Şekil 1.15. Seçici arama yönteminin trafik işareti bulunan görüntü üzerindeki sonucu. 12
Şekil 1.16. Trafik işareti görsellerinden çıkarılan HOG özellikleri. 13
Şekil 1.17. R-CNN mimarisi 14
Şekil 1.18. Fast R-CNN mimarisi. 15
Şekil 1.19. Faster R-CNN mimarisi 16
Şekil 1.20. YOLO v1 mimarisi. 17
Şekil 2.1. TRTSD verisetinden örnek gece görüntüleri. 19
Şekil 2.2. TRTSD verisetinden örnek gündüz görüntüleri. 19
Şekil 2.3. Veriseti işaretlemede kullanılan VoTT yazılımı. 22
Şekil 2.4. 3 boyutlu ortamda yapay veri üretimi. 22
Şekil 2.5. Yapay veri üretimine ait örnek görüntüler. 23
Şekil 2.6. Görüntü işleme yaklaşımı aşamaları. 23
Şekil 2.7. R,G,B ve H,S,V kanalları 24
Şekil 2.8. Renk Filtreleme yöntemi için örnek girdi görüntüsü. 25
Şekil 2.9. Renk filtreleme işleminden sonra elde edilen görüntü. 25
Şekil 2.10. HSV renk filtresi görüntüsü üzerinde medyan filtre uygulanması sonucu. . 26
Şekil 2.11. Medyan filtre sonrası açma işleminin uygulanması ile elde edilen görüntü. 27

Şekil 2.12. Tespit edilen piksel kümeleri. 27
Şekil 2.13. Piksel kümelerine filtre uygulanması. 28
Şekil 2.14. Görüntü işleme tabanlı yaklaşım uygulaması sonucu tespit edilen ilgi alanı adayı bölgelerin orjinal görüntüdeki karşılıkları. 29
Şekil 2.15. Trafik işareti boyutları kullanılarak K-means uygulanması. 30
Şekil 2.16. Veri setlerine K-means uygulandıktan sonra çıkan merkezlerin çapa kutusu karşılıkları 31
Şekil 2.17. Yapay veri üretimi akış diyagramı. 34
Şekil 2.18. Yapay veri üretimi aşamalarının örnek çıktıları. 35
Şekil 2.19. En iyileme metodlarının kayıp değeri ve iterasyon karşılaştırması. 36
Şekil 3.1. Görüntü işleme yönteminin GTSDB veriseti üzerindeki tespit sonuçları. 39
Şekil 3.2. Derin Öğrenme ile sınırlayıcı kutu tespiti için kayıp ve mAP grafiği 41
Şekil 3.3. Derin Öğrenme ile sınırlayıcı kutu tespiti için BelgiumTS veriseti sonuçları. 42
Şekil 3.4. Derin Öğrenme ile sınırlayıcı kutu tespiti için TRTSD gündüz veriseti sonuçları 42
Şekil 3.5. Derin Öğrenme ile sınırlayıcı kutu tespiti için GTSDB veriseti sonuçları 42
Şekil 3.6. Derin Öğrenme ile sınırlayıcı kutu tespiti için RTSD veriseti sonuçları 43
Şekil 3.7. Derin öğrenme ile sınıflandırma kayıp ve başarı grafiği. 43

TABLO LİSTESİ

SAYFA

Tablo 1.1. Literatürde bulunan verisetleri. 8
Tablo 2.1. TRTSD verisetinde işaretlenen trafik işaretlerinin dağılımı. 20
Tablo 2.2. Eşik seviyesi filtresi için kullanılan değerler. 25
Tablo 2.3. Küme filtresi için değerler. 28
Tablo 2.4. Derin öğrenme modelinde kullanılan katmanlar. 37
Tablo 2.5. Sınıflandırma modelinde kullanılan katmanlar. 38
Tablo 3.1. YOLO Modelinin BelgiumTS sonuçları. 44

1. GİRİŞ

1930'lu yıllarda gelişmeye başlayan dijital bilgisayarlar ile birlikte, uzun süredir mekanik sistemlere dayanan otomobiller de hızlı bir dijitalleşme sürecine girmiştir. Bu süreçte otonom olarak paralel park edebilen, trafiği takip edebilen, şerit takibi yapabilen internete bağlı araçlara dönüşmektedirler. Bu sürecin en büyük basamaklarından biri ise sürücüsüz kullanımı amaçlayan tamamen otonom araçlardır. Bu hedefe ulaşmanın ilk adımlarını sürücüye eşlik eden yardımcı sistemler olarak görülmektedir. Bu yardımcı sistemler sürücü ve yolcuların daha güvenli ve ekonomik ulaşımına imkan sağlamaktadır. Yardımcı sistemlere örnek olarak adaptif hız sabitleyici, adaptif far kontrolü, otomatik frenleme, çarpışma önleme, sürücü denetleme ve şerit uyarısı gibi sistemler verilebilir. Yardımcı sistemlerin önemli bir parçası ise trafik işareti tanıma sistemidir. Bu işaretlerin, yol ve çevre hakkında büyük miktarda bilgi sağlamasından dolayı otonom araçlar için olmazsa olmaz sistemlerden biridir.

Görüntü üzerinden nesne tespiti problemi, bilgisayar bilimleri için her zaman ilgi çekici ve zorlayıcı bir alan olmuştur. Bu problemin zorlayıcı olmasının nedenleri arasında görüntü ve ışık farklılıkları, nesnelerin çok fazla çeşitliliğe sahip olması, görüntü üzerindeki hareket ve hava durumu kaynaklı bozulmalar, fiziksel bozukluklar ve yüksek hesaplama maliyeti gösterilebilir. Günümüzde nesne tespiti problemi için uygulanan en başarılı yöntemler yapay sinir ağları ve derin öğrenme yaklaşımını kullanmaktadır [1]. Derin öğrenme; büyük miktarda yapay sinir ağı katmanı içeren bir modeli, büyük miktarda veri ile eğitmeye dayanır. Görüntü işleme yöntemlerine dayalı yaklaşımlarda, görüntüden bilgi çıkarımı aşamaları uzman kişilerce belirlenip programlanır. Derin öğrenmede ise bu aşamalara gerek kalmadan sistemin kendi kendine gerekli çıkarımları elde etmesi sağlanır. Bu nedenle görüntü üzerinden trafik işaretlerinin tespiti için en uygun yöntem olarak ön plana çıkmaktadır.

Derin öğrenmenin başarılı olabilmesi için büyük miktarda veriye ihtiyacı vardır. Trafik işaretlerinin tespiti için derin öğrenme kullanımının da en önemli ihtiyacı budur. Günümüzde trafik işaretlerini küresel standartlaşmayı sağlayamamaktadır. Her ülke kendine özgü trafik işareti standardına sahiptir. Bu nedenle yapılan çalışmalarda kullanılan veri setleri, başka ülkeler için kullanılamamaktadır. Bu noktada Türkiye'de elde edilen görüntülerden oluşan bir veri seti bulunmadığı için bu tezin kapsamında veri

seti oluşturulmuştur. Geleneksel görüntü işleme yöntemleri ile derin öğrenme yaklaşımı, elde edilen veri seti ile birlikte farklı ülkelerden elde edilmiş veri setleri kullanılarak karşılaştırılmıştır.

1.1 Trafik İşaretleri

Trafik işaretleri tarih boyunca yollarda bilgi verme için kullanılmıştır. Örneğin Roma döneminde taştan yapılmış direkler, Roma'ya olan mesafeyi ve yönü göstermekteydi. Orta çağda şehirlere ve kasabalara olan mesafeleri ve yönleri göstermek için kullanılmaktaydı. Günümüze doğru yaklaştıkça şehir içlerinde de kullanmaya başlayan işaretler, araçların ortaya çıkması ve yayılması ile birlikte; yön, bilgi ve kuralları göstermek için kullanılmaktadır. Bu işaretler küresel olarak ortak amaçlar dahilinde kullanılsa da tam anlamıyla şekil ve içerik olarak küresel bir standartlaşma henüz sağlanamamıştır. 1968'de yapılan "Vienna Convention on Road Signs and Signals" [2] adlı konferans ile bu amaçlanmış, önemli bir aşama kat edilmiştir. Ancak ülkeler ekonomik nedenlerden dolayı, üzerinde anlaşılan tüm maddeleri yerine getirememiş, ülkeler arası farklılıklar devam etmiştir. Şekil 1.1'de çeşitli Avrupa ülkelerindeki trafik işaretleri arasındaki farklar görülmektedir.



Şekil 1.1. Çeşitli Avrupa ülkelerindeki trafik işaretleri örnekleri.

Türk trafik işaretleri 3 kısma ayrılmaktadır. Bunlar Şekil 1.2, 1.3 ve 1.4'te sırası ile gösterilmektedir. Şekil 1.2'de tehlike uyarı işaretleri, Şekil 1.3'te trafik tanzim işaretleri ve Şekil 1.4'te bilgi işaretleri görülmektedir. Bütün trafik işaretleri 4 parçadan oluşmaktadır. Bunlar; çerçeve, iç dolgu, işaret şekli ve piktogramdır. Türk trafik işaretlerinde çerçeve kırmız renk, iç dolgu beyaz ya da mavi renk, piktogramlar siyah, beyaz, turuncu, kırmızı ya da sarı renklerinin kullanımı ile oluşmaktadır.



Şekil 1.2. Trafik uyarı işaretlerine örnekler.



Şekil 1.3. Trafik tanzim işaretlerine örnekler.



Şekil 1.4. Trafik bilgi işaretlerine örnekler.

1.2 Araçların Dijitalleşmesi

Elektronik ve bilgisayar alanların yaşanan gelişmeler ile birlikte, çevreden veri elde edilmesi ve işlenmesi de paralel olarak gelişmektedir. Araca içine veya dışına yerleştirilecek bir veya birden fazla kamerayla renkli görüntü elde edilmesi ile kızıl ötesi verici ve kızıl ötesi algılayıcılarla derinlik görüntüsü elde edilmesi, görüntü verisinin elde edilmesine örnek verilebilir. Gelişen elektronik sistemler ile daha yüksek çözünürlükte, farklı ışık koşullarında daha iyi verim alınabilen kameralar geliştirilmektedir. Bu çözünürlük artışına paralel olarak bilgisayar alanındaki gelişmeler ile birlikte hesaplama kapasitesi de artmaktadır. Kullanılan kameraların geniş açılı olması ve farklı ışık koşullarında gerekli görüntü kalitesini sağlayabilmesi en önemli kriterlerdendir. Bir diğer önemli kriter ise aracın hızı arttıkça yüksek netliği koruyabilmek için düşük pozlama süresine sahip olmasıdır. Yapılan çalışmalarda radar ve kızılötesi verileri ile birlikte görüntü verisi beraber kullanılarak, derinlik bilgisine de

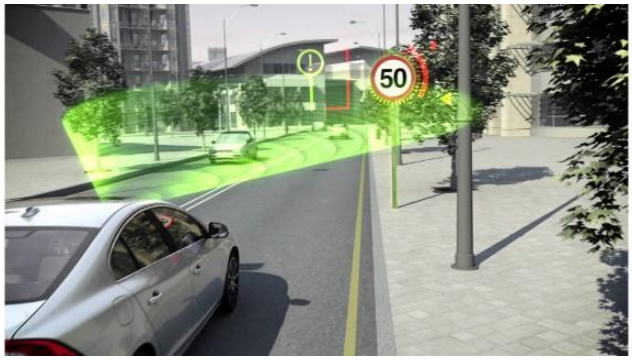
ulaşılabilmektedir [3]. Bu derinlik bilgisini kullanarak yanlış tespit edilen sonuçların azaltılması sağlanabilmektedir.



Şekil 1.5. Trafik işaretinin tespitini zorlaştıran etkenlere ait örnekler [4].

1.3 Verinin İşlenmesi

Elde edilen veriyi anlamlı bir hale getirebilmek için verinin bir takım yöntemler kullanarak işlenmesi gerekmektedir. Bu yöntemler bilgisayar bilimlerinin temel araştırma konularını oluşturmaktadır. Bu nedenle görüntü işleme alanı günümüzde popülerliğini daha da artırmaktadır. Görüntü üzerinden bilgi çıkarımı yapabilmek için bir çok farklı yaklaşım geliştirilmiştir. Kısıtlı tanıma ve çeşitliliğe sahip problemlerde görüntü işleme yöntemleri işe yaramakla beraber, yüksek karmaşıklıkta ve çeşitlilikteki problemler için derin öğrenme yöntemlerinin daha uygun olduğu akademik çalışmalar ile görülmüştür [1]. Araçlardan elde edilecek görüntülerden yapılacak bilgi çıkarımı otonom araçlar için en önemli basamağı oluşturmaktadır. Kullanım alanlarına örnek olarak yaya tespiti, araç tespiti, yol ve yol çizgisi tespiti ve trafik işareti tespiti verilebilir. Şekil 1.6'da aracın kameradan elde ettiği görüntü ile trafik işaretini algılaması tasvir edilmiştir. Otonom araçlar için en önemli kriter insan güvenliğidir. Bunu sağlamak için aracın etrafındaki tüm veriyi mümkün olan en yüksek biçimde kullanıp yorumlaması ve karar alması gerekmektedir. Şekil 1.5'te görüldüğü üzere, aracın tespit etmesi gereken bu bilgiler yüksek çeşitliliğe, zorlayıcı unsurlara ve kavramsal ilişkilere sahiptir. Bu nedenle yalnızca görüntü işleme yöntemleri kullanılması bu problem için yetersiz kalmaktadır.



Şekil 1.6. Trafik işaretinin kameradan elde edilen görüntü ile algılanması.

1.4 Society of Automotive Engineers (SAE) Seviyeleri

SAE [5] seviyeleri, otomasyonsuz araçlar ile tamamen otonom araçlar arasındaki seviyeleri göstermektedir. Bu seviyeler aracın otonomluk seviyelerindeki artışa göre artmakta ve 6 seviyeden oluşmaktadır. Seviyelerin ilkinden itibaren görülebileceği üzere, bilgi veren sistemler otonom araçların temelini oluşturmaktadır. Seviye 0'dan itibaren en önemli yeteneklerden biri trafik işareti tespitidir. SAE seviyeleri şu şekilde özetlenmektedir:

Otomasyonsuz (SAE Level 0): Sistem çevre hakkında uyarı ve bilgi verir, hiçbir zaman aracın kontrolünü almaz

Sürücülü Otomasyon (Hands on - SAE Level 1): Sürücü ve sistem aracın kontrolünü paylaşır. Örneğin sistem aracın hızını ayarlarken, sürücü direksiyonu kontrol eder.

Kısmi Otomasyon (Hands off - SAE Level 2): Sistem aracın tüm kontrolünü alır. Fakat sürücü sistemi gözetlemek ve ani bir durum karşısında müdahale için hazır beklemek zorundadır. Tesla Autopilot [6] sistemi bu seviye için örnektir.

Koşullu Otomasyon (Eyes off - SAE Level 3): Sürücü güvenli bir şekilde dikkatini film izlemek ya da telefonla ilgilenmek gibi şeylere verebilir. Sistem ani durumlara müdahale etme yeteneğine sahiptir, ancak yine de sürücü hazır olmalıdır.

Yüksek Otomasyon (Mind off - SAE Level 4): Bu seviyede sürücü dikkatine ihtiyaç duyulmamaktadır, örneğin sürücü uyuyabilir ya da sürücü koltuğunu bırakabilir.

Tam Otomasyon (Steering wheel optional - SAE Level 5): Tamamen kendi başına karar alabilen sistemler bu seviyeye girmektedir, örneğin robotik taksiler bu otomasyona örnek verilebilir.

1.5 Görüntü İşleme Yöntemlerine Dayalı Yaklaşım

Günümüzde en yoğun kullanılan renk modeli RGB(Red, Green, Blue)'dir. Bu modelde 3 farklı renk kanalı kırmızı, yeşil ve mavi değerlerini taşıyarak bir piksel için rengi belirler. Bu piksellerin bütünü ise görüntüyü oluşturur. Görüntü işlemede bilgi çıkarımı, bir takım işlemlerin bu değerleri kullanarak sonuç elde etmesi ile olur. Klasik yöntemlerde bu işlemler özellik çıkarımı, örüntü tespiti, kenar tespiti, gürültü azaltma, bulanıklaştırma-netleştirme gibi problemlerin çözümünde kullanılmaktadır. Şekil 1.7'de grileştirme, bulanıklaştırma, değer filtresi, genişletme, aşındırma ve kenar tespiti için örnek sonuçlar verilmiştir. Bu yöntemlerin probleme özgü olarak, uzman kişilerce seçilip uygulanması gerekmektedir. Nesne tanıma gibi bağlam tabanlı problemlerde ise bu yöntemler yetersiz kalmaktadır. Bu nedenlerden dolayı son dönemlerde yapay sinir ağları ve derin öğrenme yöntemleri ön plana çıkmıştır. Görüntü işleme yöntemleri ise derin öğrenme aşamasından önce ve öğrenme aşamasında verinin çoğaltılması için kullanılmaktadır [7].

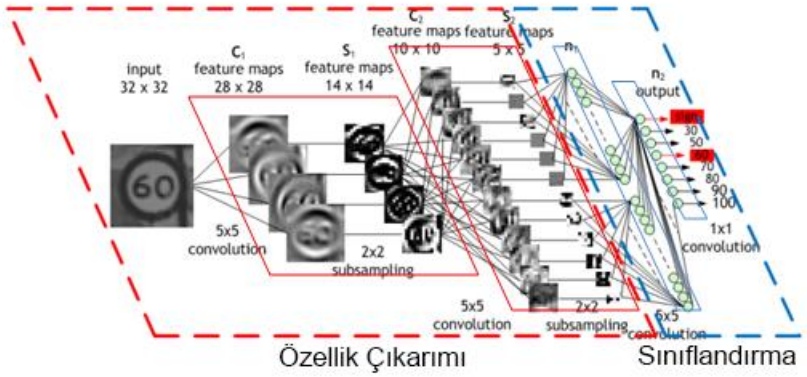


(a) (b) (c) (d) (e) (f) (g)

Şekil 1.7. Görüntü işleme yöntemleri uygulanmış trafik işaretleri. a) Orjinal görüntü, b) Grileştirilmiş görüntü, c) Bulanıklaştırma, d) Değer filtresi, e) Genişletme, f) Aşındırma, g) Kenar tespiti

1.6 Derin Öğrenmeye Dayalı Yaklaşım

Yapay sinir ağları, biyolojiden esinlenerek oluşturulan hesaplama yapılarıdır. Bu yapılar nöron ve sinapsların çalışmasını matematiksel olarak gerçekleştirmektedir. Problem bazlı ön bilgiye ihtiyaç duymadan makine öğrenmesini gerçekleştirebilen bu yapılar akademik alanda oldukça popüler hale gelmiştir. Bu popülerliğin en önemli kaynağı derin öğrenmenin birçok problemde oldukça başarılı sonuçlar üretmiş olmasıdır [1]. Derin öğrenme, büyük miktarda yapay sinir ağlarından oluşan katmanların oluşturduğu makine öğrenmesi modelidir. Şekil 1.8'deki örnek derin öğrenme modelinde görüldüğü üzere, görüntüden özellik çıkarımı ve sınıflandırma tek model ile yapılabilmektedir. Bu modelin büyük miktarda veri ile yapının problemdeki ilişkileri bularak çözüme ulaşması, klasik yöntemlerdeki uzman ihtiyacını ortadan kaldırması ve bu yöntemlerden daha başarılı olması, tercih edilmesinde en büyük nedendir. Oluşturulan yapay sinir ağı, kenar tespiti gibi görüntü işleme ile çıkarılabilecek özellikleri kendi kendine öğrenerek, bu özellikleri yorumlamayı öğrenebilmektedir.



Şekil 1.8. Özellik çıkarımı ve sınıflandırma yapabilen derin öğrenme modeli diyagramı.

1.7 Veri Setleri

Ülkeler arasında standartlaşma sağlanamaması sonucunda, akademik çalışmalarda kullanılan veri setleri de farklılık göstermektedir. Örnek olarak Belçika [4], Almanya [8], İsveç [9], ABD [10], Rusya [11] ve Çin [12]'de elde edilmiş veri setleri verilebilir. Trafik işaretlerindeki farklılaşma, ülkelerdeki mimari ve peyzaj farklılıkları, trafik işaretleri ve reklam tabelaları ile ilgili yönetmelikleri gibi nedenlerden dolayı, bu verisetleri büyük farklılıklar göstermektedir. Bu tez ile birlikte ülkemizden toplanan

görüntüler ile veriseti oluşturulmuştur. Tablo 1.1'de verisetlerinin çözünürlükleri, içerdikleri görüntü ve sınıf adedi verilmiştir.

Tablo 1.1. Literatürde bulunan verisetleri.

Veriseti	Ülke	Çözünürlük	Oran	Trafik işareti bulunan görüntü sayısı	Sınıf adedi
BelgiumTS	Belçika	1628x1236	~4:3	9003	62
GTSDB	Almanya	1360x800	~16:9	741	43
Traffic Signs Dataset	İsveç	1280x960	4:3	4000	7
LISA	ABD	640x480 – 1024x522	4:3 - ~2:1	6610	47
RTSD	Rusya	1280x720 – 1920x1080	16:9	101033	156
TT100K	Çin	2048x2048	1:1	30000	221
TRTSD	Türkiye	1280x720	16:9	4503	31



Şekil 1.9. BelgiumTS verisetinden örnek görüntüler [4].



Şekil 1.10. GTSDB verisetinden örnek görüntüler [8].



Şekil 1.11. RTSD verisetinden örnek görüntüler [11].



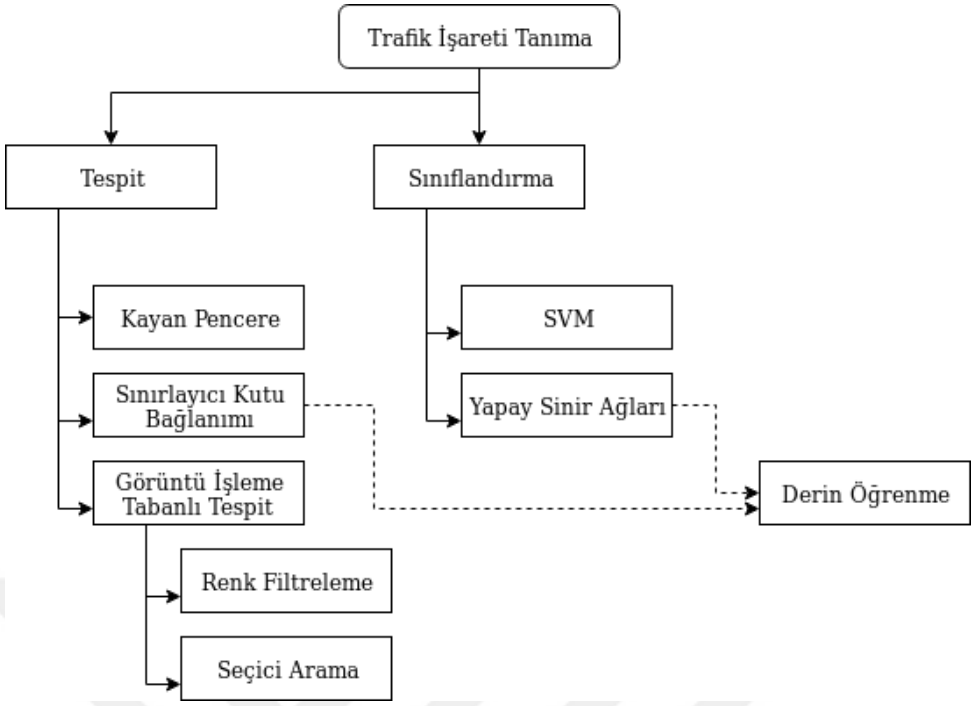
Şekil 1.12. Traffic Signs verisetinden örnek görüntüler [9].

1.8 Amaç ve Önem

Bu tez çalışmasında trafik işareti tespitini görüntü işleme ve derin öğrenme yöntemlerini gerçekleştirilmesi, karşılaştırılması ve farklı ülkelerden elde edilen veri setleri ile uygulanması amaçlanmıştır. Bu kapsamda Türkiye'den toplanan görüntüler ile birlikte derin öğrenme için yeterli olacak seviyede bir veri seti oluşturulması hedeflenmiştir.

1.9 Literatür Araştırması

Trafik işareti tanıma problemi, diğer nesne tanıma problemleri gibi iki aşamaya ayrılmaktadır [13]. Birinci aşama görüntüde nesnenin bulunduğu bölgenin *tespit* edilmesidir. Bu aşamadaki ana yöntemler; kayan pencere (rolling window), sınırlayıcı kutu bağlanımı (bounding box regressor) ve görüntü işleme tabanlı yaklaşımlar olarak ayrılmaktadır. Görüntü işleme tabanlı yaklaşımlar, renk filtreleme (color thresholding) ve seçici arama (selective search) olarak ayrılmaktadır. İkinci aşama ise tespit edilen bu alandaki nesnenin içeriğinin *sınıflandırılmasıdır*. Bu aşamada ise SVM (Support Vector Machine – Destek Yöney Makinesi) ve yapay sinir ağları en yaygın yöntemlerdir. Son dönemde yapılan derin öğrenme çalışmaları ile bu iki aşama tek bir sisteme indirgenebilmiştir. Yapılan çalışmalar; tespit aşaması için yaklaşımlar, sınıflandırma aşaması için yaklaşımlar ve derin öğrenme tabanlı yaklaşımlar olarak üçe ayrılarak incelenmiştir. Şekil 1.13'te bu aşamalar için farklı yaklaşımların ve bunların derin öğrenme ile ilişkisi verilmiştir.



Şekil 1.13. Trafik işareti tanıma problemi için literatürde var olan yöntemler.

1.9.1 Tespit Aşaması için Yaklaşımlar

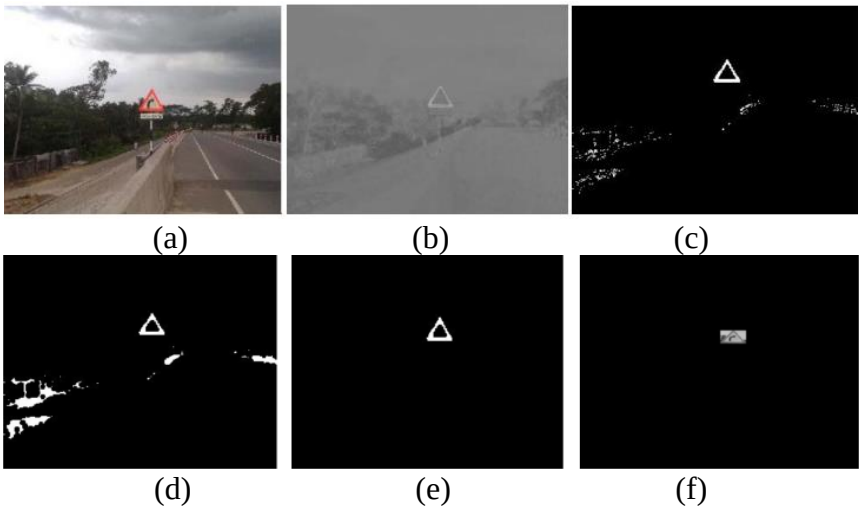
Tespit aşamasındaki en önemli unsur nesne bulunma ihtimali olan bölgeleri belirten sınırlayıcı kutuları (bounding box) bulmaktır. Bu kutuların nesneyi oluşturacak en küçük şekilde olması beklenmektedir. Tespit edilen bu kutular daha sonra belirlenen sınıflandırma yöntemine verilerek sonuca ulaşılmaktadır. Bu noktada akla ilk gelebilecek çözüm kayan pencere (Sliding Window) uygulamasıdır. Kayan pencere yönteminde farklı boyuttaki dikdörtgenlerin görüntü üzerinde kaydırılması esas alınır. Ancak farkedilebileceği gibi görüntü üzerinde bu yöntem ile sonsuz sayıda sınırlayıcı kutu elde edilebilir. Bu yüksek maliyeti azaltabilmek için farklı çözüm yöntemleri geliştirilmiştir. Bu aşamadaki çözümleri görüntü işleme tabanlı ve derin öğrenme tabanlı olarak ikiye ayrılabilir.

Nesne tanıma problemi için geçmişte yapılan çözümler çoğunlukla görüntü işleme adımlarının birleşimi olarak uygulanmıştır. Bu adımların başarılı olması için görüntüdeki çeşitliliğin sınırlandırılmış olması önemlidir. Bu nedenle yüksek çeşitliliğe sahip problemlerde yetersiz kalmaktadırlar.

1.9.1.1 Renk Filtreleme

Bu yöntemde ilk adım belirli değer aralığında olmayan renk değerlerinin çıkarılmasıdır. Bu işlemdeki değer aralığı, probleme göre uzman kişilerce belirlenmektedir. Bu işlem sonunda gürültü azaltmak için bir takım görüntü işlemleri de uygulanabilmektedir. Örneğin medyan filtre uygulayarak bulanıklaştırma, morfolojik açma işlemi ile küçük bağlantıları koparma gibi işlemler verilebilir. Bu adımlardan sonra kümelenmiş piksellerin tespiti yapılıp, kümenin boyutuna göre filtreleme yapılmaktadır. En son noktada; kalan kümeler nesne bulunma ihtimali olan bölge, sınırlayıcı kutuları gösterir. Bu yöntem görüntüdeki çeşitliliğin az olması ve uzmanlarca girilen değerlerin sınırlı ortamda çalışmasına bağlıdır. Örneğin gündüz ışığı koşullarına göre belirlenmiş değerler, gün batımı ışığındaki görüntüler için geçersiz kalacaktır. Bu yöntemin avantajı ise derin öğrenme yöntemleri gibi yüksek miktarda veriye ihtiyaç duymamasıdır.

M. Abedin [14] çalışmasında görüntüye renk filtresi uyguladıktan sonra morfolojik işlemlerden geçirerek ilgi alanlarını (ROI-Region of Interest) elde etmiştir. Şekil 1.14'te bu yöntemin aşamaları görülmektedir. A. Adam [15] çalışmasında, M. Abedin [14] çalışmasına temel oluşturacak şekilde görüntü renk filtresinden geçirilmiştir. C. Souani [16] çalışmasında A. Adam [15] çalışmasına benzer olarak renk filtresi uygulayarak ilgi alanlarını elde etmiştir.



Şekil 1.14. Renk filtreleme ve görüntü işleme yöntemleri ile trafik işareti tespiti [14]. a) Orjinal görüntü, b) Normalleştirilmiş görüntü, c) Değer filtresi sonucu, d) Morfolojik işlemlerin sonucu, e) Blob filtresi sonucu, f) Elde edilen ilgi alanı.

1.9.1.2 Seçici Arama(Selective Search)

Görüntü üzerinde bölütlendirme (segmentation) renk bilgisine göre yapılabilir. Bu bölütlendirme kavramsal bilgiden uzaktır. Örneğin aynı renkte iki nesne tek bir bölütlemeye dahil olabilir. “Selective Search” [17] yönteminde görüntü küçük bölütlere ayrılır. Ardından her bir bölüt birbirlerine olan yakınlığına göre birleştirilir. Bu işlem bir bölüt kalana kadar devam eder. Bu süreçteki tüm bölütler nesne içerebilecek ilgi alanı olarak ele alınır. Böylece yüksek miktarda ilgi alanı sınıflandırma aşamasına geçer ve nesne içermeyenler elenir. Literatürde bu yöntemi kullanan çalışmalarda, olabilecek tüm ilgi alanları bir üst limit girilerek kısıtlandırılmıştır. Örneğin Fast R-CNN (Region with Convolutional Neural Network features) [18] çalışmasında bu limit 2000 olarak belirlenmiştir. Şekil 1.15’te bu yöntemin örnek sonucu verilmiştir. Elde edilen ilgi alanları büyüklük, küçüklük, en ve boy gibi kriterlere göre filtrelenerek ilgi alanı sayısı azaltılabilmektedir. Seçici arama yöntemi, yüksek miktarda işlem gücü gerektirdiği için gerçek zamanlı uygulamalara uygun değildir.



(a) (b)

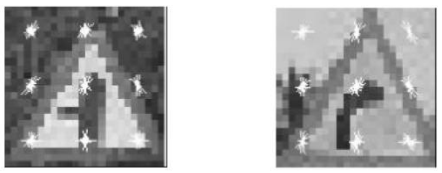
Şekil 1.15. a) Seçici arama yönteminin trafik işareti bulunan görüntü üzerindeki sonucu, b) Boyutlarına göre filtrelenmiş sonuçlar.

1.9.2 Sınıflandırma Aşaması için Yaklaşımlar

Tespit edilen bölgelerin içeriğinin belirlenmesi için sınıflandırma yapılması gerekmektedir. Bu nedenle makine öğrenmesi yöntemleri bu aşamada uygulanmaktadır. En yaygın kullanım yöntemleri SVM ve yapay sinir ağları olarak gösterilebilir.

1.9.2.1 SVM Kullanarak Sınıflandırma

Makine öğrenmesinde en temel görevlerden biri sınıflandırmadır. Sınıflandırıcı model verilen örneğin hangi sınıfa ait olduğunu söyler. Bu nedenle verilen örnekten özellik çıkarımı yapılması ve bu özelliklerin modele verilmesi gerekmektedir. Model bu özelliklere göre veriyi en uygun şekilde sınıflara göre bölmeye çalışır. Görüntü işleme problemlerinde SVM modellerine verilebilecek özelliklerden biri HOG (Histogram of Oriented Gradients) [19] özellikleridir. HOG özelliklerinin çıkarılması için görüntü belirli sayıda kareye bölünerek her bir kare için 8 yönde eğilim çıkarılmaktadır. Bu eğilimler HOG özelliklerini oluşturur. Şekil 1.16'da örnek HOG özellikleri görülebilir. HOG özellikleri çıkarıldıktan sonra, SVM [20] modeli bu özellikleri kullanarak sınıflandırma yapar [21] [22]. Bu yöntemin dezavantajı tüm veristine tek seferde ihtiyaç duymasıdır. Büyük veri setleri için büyük miktarda hafıza gerekmektedir. Buna karşın modelin yapısı gereği sadece sınıra denk gelen veriyi kullanması, geriye kalan verinin işlevsiz olması demektir. Dolayısıyla bu veri yapılabilecek bir ön işleme ile filtrelenebilir ve veri setinin ihtiyaç duyacağı hafızayı azaltılabilir.



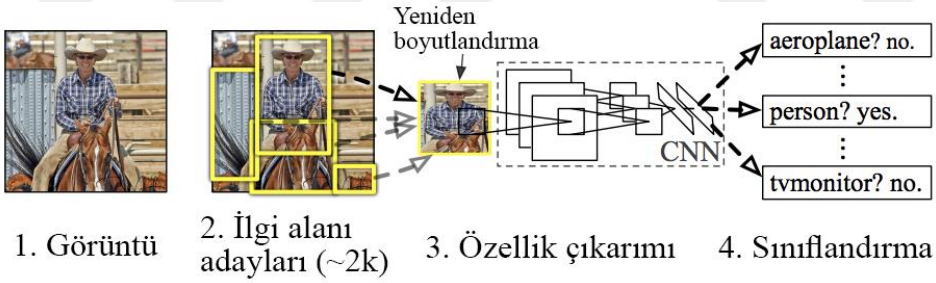
Şekil 1.16. Trafik işareti görsellerinden çıkarılan HOG özellikleri [14].

1.9.2.2 Yapay Sinir Ağı Kullanarak Sınıflandırma

Yapay sinir ağları biyolojiden esinlenerek oluşturulmuş bir modeldir. Bu modelde birbirine bağlı çok sayıda nöron bulunmaktadır. Yapay sinir ağı modeli nöron katmanlarından oluşur. Bunlar girdi katmanı, gizli katmanlar ve çıktı katmanıdır. Gizli katmanların sayısı arttıkça model derinleşmekte, dolayısıyla daha karmaşık problemleri çözebilmektedir. Yapay sinir ağları geri yayılma algoritmasına dayanır. Bu algoritmada verilen girdinin gerçek sonucu ile hesaplanan sonucu arasındaki hata hesaplanarak nöronların parametreleri ayarlanır. Bu işlem defalarca tekrar edilebilir, eğitilmiş bir model daha sonra tekrar eğitilmeye devam edebilir. SVM modelindeki büyük miktarda hafıza ihtiyacına karşın, yapay sinir ağlarında veri kümeleri ile eğitim yapılabilmekte dolayısıyla hafıza kısıtlamasından etkilenmemektedir. Normal yapay sinir ağları tüm

nöronların birbirine bağlı olduğu bir modeldir. Bu nedenle görüntü işleme problemlerine çok uygun değildir. Bunun yerine görüntü işleme problemlerinde yaygın olarak evrişimli (convolutional) [18] [23] [24] yapay sinir ağı modeli kullanılmaktadır. Bu modelde görüntü işleme yöntemlerindeki filtrelere benzer olarak çekirdek (kernel) kullanılmaktadır. Görüntüde gezdirilen bu çekirdek, evrişimli değerleri çıktı olarak üreten bir yapay sinir ağı katmanı oluşturur (2 boyutlu evrişimli katman). Bu katmanların çoğullanması ile 3 boyutlu evrişimli katmanlar elde edilir. 3 boyutlu evrişimli katmanların ard arda farklı boyutlarda sıralanması ile elde edilen modeller derin öğrenme modeli olarak adlandırılır.

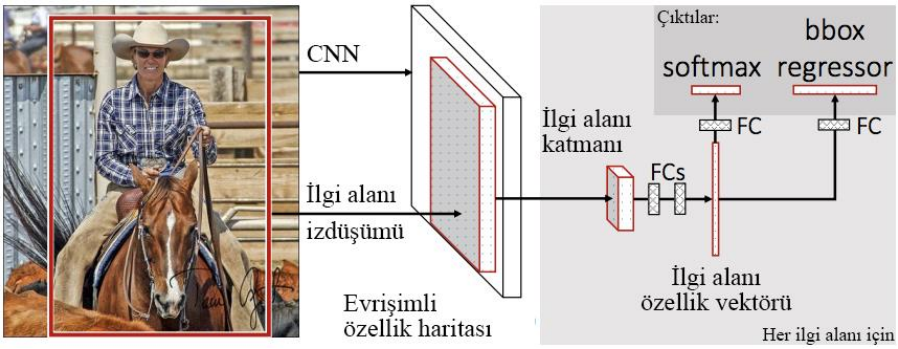
R-CNN [25] mimarisinde seçici arama yöntemi ile elde edilen çok sayıda bölgenin CNN modeline verilip görüntüden özellik çıkarılması sağlanmıştır. Bu özellikler daha sonra SVM modeline verilerek sınıflandırma yapılmıştır. Aynı zamanda CNN modeli seçici arama ile çıkarılan bölgenin regresyonunu yaparak, bölgenin kaydırma miktarını belirleyip nesneye daha iyi oturmasını sağlamıştır. Şekil 1.17'de R-CNN mimarisinin genel akışı görülmektedir.



Şekil 1.17. R-CNN mimarisi [25].

Fast R-CNN [18] çalışmasında R-CNN mimarisinin yavaş olma nedenlerinin önüne geçilmeye çalışılmıştır. Seçici arama ile elde edilen büyük miktarda bölgenin her biri için CNN modelini beslemek yerine, bölgelerin tamamı tek seferde CNN modeline verilmektedir. Ayrıca görüntüden tek seferde elde edilen özellik haritası ile bu bölgelerin izdüşümleri alınarak CNN modelini beslemektedir. Şekil 1.18'de görülebileceği gibi, bu özellik haritasına seçici arama ile elde edilen bölgeler tek seferde

verilerek daha hızlı sonuç alınmıştır. R-CNN mimarisinde CNN, SVM ve bölge regresyonu, Fast R-CNN mimarisinde birleştirilerek tek CNN modeli ile hesaplanmıştır.

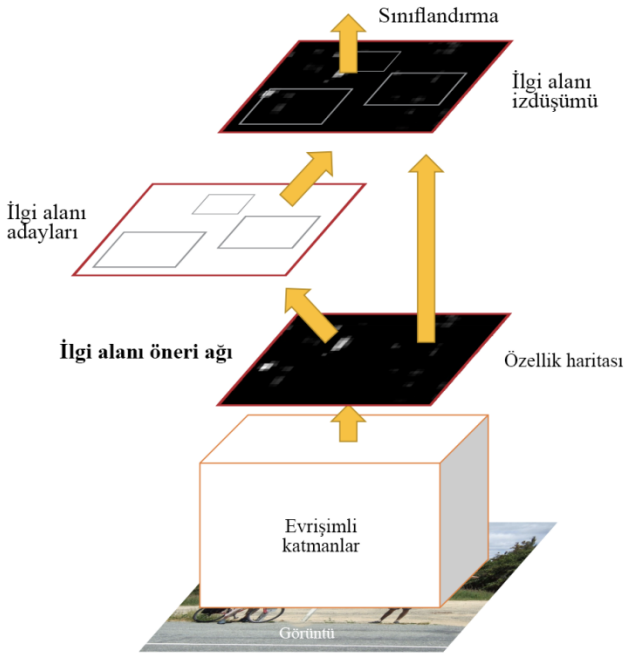


Şekil 1.18. Fast R-CNN mimarisi [18].

1.9.3 Derin Öğrenme Tabanlı Yaklaşımlar

Evrişimli sinir ağları modellerinin başarısı kanıtlanmaya devam ettikçe görüntü işleme tabanlı yaklaşımların yerini bu modeller almaya başlamıştır. Fast R-CNN mimarisinde görüldüğü üzere farklı yapılar tek bir CNN modelinde toplanmıştır. Daha sonra yapılan çalışmalarda ise tüm mimari tek bir CNN modeline aktarılarak, görüntüye bir kez bakıp tamamen CNN modeli ile yorumlayan mimariler ortaya çıkmıştır. Bu modellerin en önemli üstünlüğü hızlı olması ve gerçek zamanlı problemlere uygulanabilir olmasıdır.

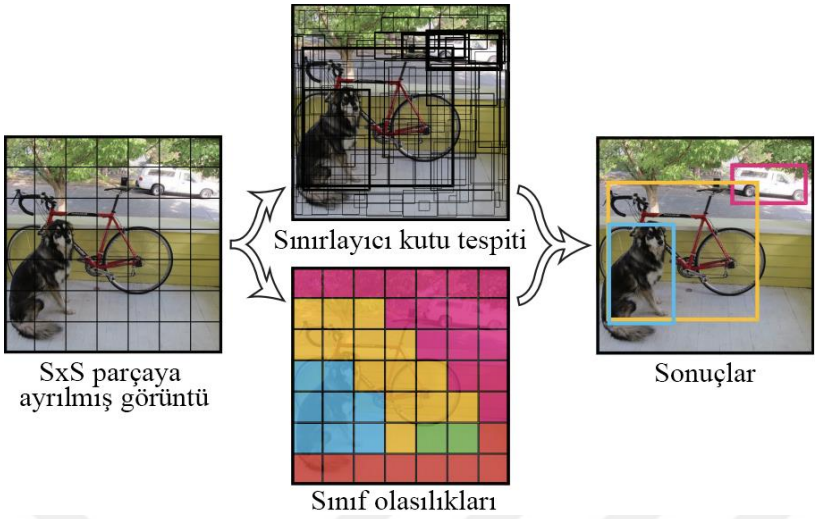
Faster R-CNN [26] çalışmasında, Fast R-CNN mimarisindeki özellik haritasından, bölge çıkarımı yapılması sağlanarak seçici arama modeli mimariden çıkarılmıştır. Bunun yerine evrişimli katmanlardan oluşan bir ağ kullanılarak, ilgi alanı adaylarını tespit etmesi sağlanmıştır. Bu sayede büyük bir hız kazanımı sağlanmıştır. Bu yöntemde farklı boyuttaki nesneleri daha başarılı tespit edebilmek için çapa kutuları (Anchor Box) yöntemi kullanılmıştır. Elde edilen ilgi alanı adayları, Fast R-CNN mimarisinde olduğu gibi özellik haritasındaki izdüşümleri ile modeli beslemektedir. Şekil 1.19'da Faster R-CNN mimarisinin genel akışı görülmektedir.



Şekil 1.19. Faster R-CNN mimarisi [26].

YOLO v1 (You Only Look Once) [23] mimarisinin temel özelliği, CNN modelinin tek seferde görüntüye bakması ve bu nedenle çok hızlı çalışabilmesidir. Bunun karşılığında diğer modellere göre başarısı biraz daha düşüktür. Bu mimaride görüntü yeniden boyutlandırılıp, YOLO modeline verilir. Model görüntüyü belirli sayıda hücreye ayrırır. Her hücre belirlenen sayıda tahmin yapar. Tahminler sınırlayıcı kutu ve sınıflandırma bilgisini üretir. Şekil 1.20'de özellik haritası çıkarma ve ilgi alanı adaylarının oluşturulması akışı görülmektedir. Son aşamada, birbirine yakın olan tahminlerin elenmesi için; tüm tahminler azami baskılamadan (Non Maximum Supression) geçer ve kalan tahminler sonuç olarak alınır. Azami baskılama yöntemi, birbirine benzer ve üst üste gelen tahminlerin, tek tahmine indirgenmesi için kullanılır.

SSD [24] çalışmasında YOLO mimarisinden farklı olarak Faster R-CNN'de kullanılan çapa kutuları yöntemi kullanılmıştır. Bu yöntemde sınıflandırma için, CNN modelinde tam bağlı katman yerine evrişimli katmanlar kullanmışlardır. Modelin sonunda bulunan katman, farklı boyutlandırmalardan üretilen özellik haritalarını kullanarak daha başarılı sonuçlar alınmıştır.



Şekil 1.20. YOLO v1 mimarisi [23].

YOLO v2 [27] çalışmasında, YOLO v1 çalışmasına göre hem başarı oranı hem de hız artırılmaya çalışılmıştır. Rakip yöntem olarak bahsedebileceğimiz SSD çalışmasında başarısı kanıtlanan yöntemler buraya da uygulanmıştır. Faster R-CNN ve SSD'de olduğu gibi çapa kutuları kullanmışlardır. Çapa kutuları elle belirlemek yerine data üzerinde K-means yaparak belirlemişlerdir. Bir önceki versiyonda bulunan tam bağlı katman kaldırılmıştır.

YOLO v3 [28] çalışmasında YOLO v2 mimarisi geliştirilmeye çalışılmıştır. En önemli geliştirme, SSD'de bulunan farklı boyutlardan elde edilen özellik haritası kullanımının uygulanmasıdır. Model, 3 farklı görüntü boyutu üzerinden tahmin yapmaktadır. Bunun en önemli yansıması, küçük boyutlu nesneleri daha iyi tespit edebilmesi olmuştur. Öte yandan üç farklı boyut için tahmin yapması, çıktı boyutunu artırdığından; bir önceki mimariye göre daha yavaş performansa sahip olmasına neden olmuştur.

2. MATERYAL VE YÖNTEM

Bu tezde farklı verisetleri kullanarak, görüntü işleme ve derin öğrenme yöntemleri uygulanmıştır. Aynı zamanda Türkiye'den elde edilen görüntüler ile veri seti oluşturulmuştur.

2.1 Kullanılan Veri Setleri

Uygulanan yöntemlerin sonuçlarını gözlemlemek ve çeşitlilik sağlamak adına farklı verisetleri kullanılmıştır. Bu kapsamda BelguimTS (Belgium Traffic Sign) [4], GTSDB (German Traffic Sign Detection Benchmark) [8], RTSD (Russian Traffic Sign Dataset) [11] veristelerinin yanında Türkiye'deki görüntüler ile oluşturulan TRTSD (Turkish Traffic Sign Dataset) veriseti kullanılmıştır.

BelguimTS veriseti, Belçika'da araç üzerine yerleştirilmiş 8 farklı kameradan elde edilen görüntüler ile oluşturulmuştur. 9000 işaretlenmiş görüntü içermektedir. Bu görüntüler 1628x1236 boyutuna sahip olup 4:3 orandadır. Belçika mimarisi ve şehir yapısı gereği trafik işaretlerine benzer, yanıltıcı görseller çok azdır. Görüntüler gündüz, normal hava koşullarında çekilmiştir. Şekil 1.9'da örnek görüntüler verilmiştir.

Almanya'dan elde edilen görüntüler ile oluşturulan GTSDB veriseti, BelgiumTS verisetine içerik açısından oldukça benzemektedir. 900 adet işaretlemiş görüntü bulunmaktadır. Bu nedenle tek başına derin öğrenme uygulamaları için uygun değildir. Görüntüler 1360x800 boyutunda yani 17:10 oranındadır. Görüntüler gündüz, normal hava koşullarında; bir kısmı sisli hava koşullarını içerecek şekilde çekilmiştir. Şekil 1.10'da örnek görüntüler verilmiştir

En yüksek görüntü sayısına sahip RTSD (Russian Traffic Sign Dataset) [11] verisetinde, Rusya'dan elde edilmiş 179.000 görüntü bulunmaktadır. Bu görüntüler gündüz görüntülerinin yanında, karlı hava ve gece görüntülerini de içermektedir. BelgiumTS ve GTSDB verisetlerine göre daha karmaşık içeriğe ve yanıltıcı görsellere sahiptir. Görüntülerin bir kısmı 1280x720, bir kısmı ise 1920x1080 çözünürlüktedir, 16:9 oranındadır. Miktarının fazla olmasından dolayı derin öğrenme yöntemleri için uygundur, ancak farklı hava koşullarında görüntüler içerdiği için görüntü işleme yöntemleri için çok uygun değildir (Şekil 1.11).

Türkiye’den elde edilen veri seti bulunmadığı için, bu tez kapsamında TRTSD veriseti oluşturulmuştur. Toplamda 2105 işaretli görüntü bulunmaktadır. Gündüz ve gece görüntüleri normal ve yağışlı hava olarak bulunmaktadır. Şekil 2.1’de örnek gece görüntüleri, Şekil 2.2’te örnek gündüz görüntüleri verilmiştir. Veriseti 1920x1080 ve 1280x720 çözünürlüklerinde görüntülerden oluşmakta, 16:9 oranındadır. Kullanılan verisetleri içerisinde en fazla karmaşıklığa sahip verisetidir. Trafik işaretlerini yanıltıcı bir çok unsur içermektedir. Tablo 2.1’de işaretlenen trafik işaretlerinin dağılımı verilmiştir.



Şekil 2.1. TRTSD verisetinden örnek gece görüntüleri.



Şekil 2.2. TRTSD verisetinden örnek gündüz görüntüleri.

Tablo 2.1. TRTSD verisetinde işaretlenen trafik işaretlerinin dağılımı.

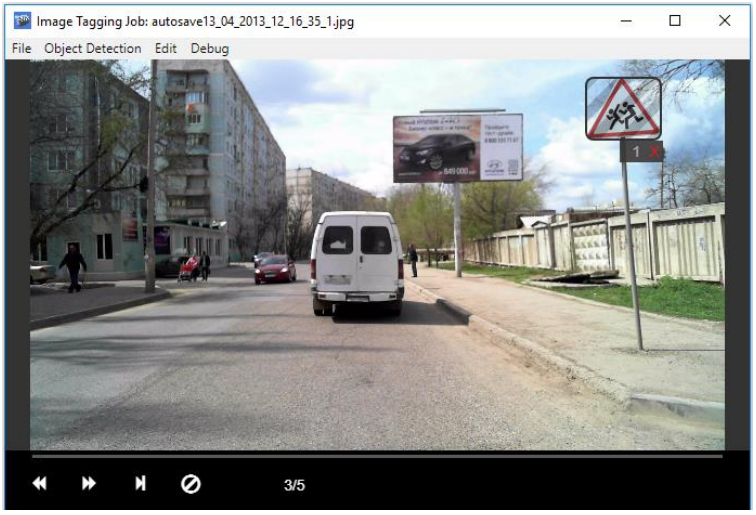
Etiket	İsim	Gündüz (adet)	Gündüz (yüzde)	Gece (adet)	Gece (yüzde)	Gündüz+Gece (adet)	Gündüz+Gece (yüzde)
b14-a	Yaya Geçidi	42	0,89%	18	1,62%	60	1,03%
b22	Durak	74	1,56%	-	-	74	1,27%
b38	Anayol	545	11,52%	14	1,26%	559	9,57%
b52-a	İki Yönlü Yol-1	226	4,78%	-	-	226	3,87%
p1	Park Yasağı	43	0,91%	-	-	43	0,74%
p2	Duraklama ve Park Yasağı	786	16,62%	258	23,20%	1044	17,87%
t1-a	Sağa Tehlikeli Viraj	-	-	7	0,63%	7	0,12%
t1-b	Sola Tehlikeli Viraj	58	1,23%	-	-	58	0,99%
t11	Yaya Geçidi	185	3,91%	177	15,92%	362	6,20%
t12	Okul Geçidi	122	2,58%	14	1,26%	136	2,33%
t16	Işıklı İşaret Cihazı	437	9,24%	136	12,23%	573	9,81%
t23-a	Sağdan Ana Yola Giriş	-	-	16	1,44%	16	0,27%
t7	Kasisli Yol	168	3,55%	13	1,17%	181	3,10%
tt1	Yol Ver	-	-	103	9,26%	103	1,76%
tt2	Dur	61	1,29%	-	-	61	1,04%
tt20	Genişliği Fazla Olan Taşıt Giremez	-	-	63	5,67%	63	1,08%
tt26-b	Sola Dönülmez	654	13,83%	11	0,99%	665	11,38%
tt26-c	U Dönüşü Yapılmaz	35	0,74%	17	1,53%	52	0,89%
tt27	Öndeki Taşıtı Geçmek Yasaktır	72	1,52%	-	-	72	1,23%
tt29-a	Azami Hız Sınırlaması	85	1,80%	-	-	85	1,45%
tt35-a	Sağa Mecburi Yön	-	-	23	2,07%	23	0,39%
tt35-c	İleri Mecburi Yön	126	2,66%	111	9,98%	237	4,06%
tt35-d	İleri ve Sağa Mecburi Yön	99	2,09%	23	2,07%	122	2,09%
tt36-a	Sağdan Gidiniz	527	11,14%	42	3,78%	569	9,74%
tt36-b	Soldan Gidiniz	-	-	31	2,79%	31	0,53%
tt37	Ada Etrafında Dönünüz	-	-	35	3,15%	35	0,60%
tt4	Girişi Olmayan Yol	317	6,70%	-	-	317	5,43%
tt42-a	Zincir Takmak Mecburi	68	1,44%	-	-	68	1,16%
		Toplam 4730		**Toplam 1112**		**Toplam 5842**	

2.1.1 Veri Toplama

Veri toplama aşamasında aracın iç kısmında öne yerleştirmiş kamera ile görüntüler elde edilmiştir. Bu aşamada otoyol, cadde ve sokak araları gezilerek çeşitlilik sağlanmıştır. Gece görüntüleri düşük çözünürlükte, bozulmalara ve ışık kırılmalarına sahiptir. Gündüz görüntüleri genellikle kapalı havada çekilmiştir. Bu nedenle ışığın az olması yöntemler için zorlayıcı unsurdur.

2.1.2 İşaretleme

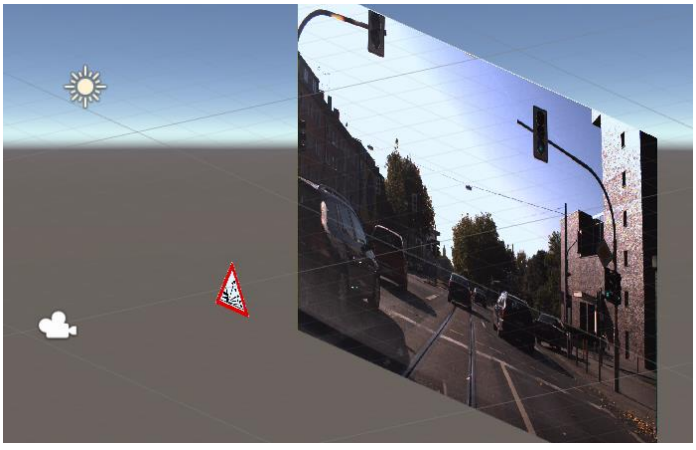
Elde edilen görüntüler işaretlenirken, VoTT (Visual Object Tagging Tool) [29] isimli program kullanılmıştır. Bu programda görüntüler tek tek incelenip trafik işareti ve sınırlayıcı kutular işaretlenmiştir. Şekil 2.3'te görüntü üzerinde işaretleme süreci örneği verilmiştir. Toplanan görüntülerin işaretlenme aşaması, veri setinin oluşturulmasında en fazla zamanı gerektiren aşama olmuştur. Türkiye'de 209 farklı trafik işareti bulunmaktadır. İşaretleme sırasında sınırlayıcı kutunun en uygun şekilde işaretlenmesi ve etiketlenmesi, oldukça fazla zaman gerektiren bir işlemdir. Veri toplama aşamasında elde edilen görüntüler video olarak toplanmıştır. Videolar saniyede 30 kareden oluşmaktadır. Bir görüntüde birden fazla trafik işareti olabilmektedir. Birbirine yakın görüntü karelerindeki işaretlenmiş veriler, derin öğrenme aşamasında fayda sağlamamaktadır. Derin öğrenme uygulamasında görüntüler çoğullama aşamasından geçtiği için benzer görüntüler elde edilebilmektedir. Bu nedenle saniyede 30 kare işaretlemek yerine her saniyede bir frame işaretlenmiştir.



Şekil 2.3. Veriseti işaretlemede kullanılan VoTT yazılımı.

2.2 Yapay Veri Üretimi

Trafik işaretinin sınıflandırılması aşamasında eğitilecek model için de büyük miktarda veriye ihtiyaç vardır. Bu nedenle veriyi yapay olarak oluşturmak adına bir program geliştirilmiştir. Bu program Unity3D [30] oyun motorunu kullanarak 3 boyutlu ortamda arkaplan, trafik işareti ve ışık olmak üzere üç kısımdan oluşmaktadır. Şekil 2.4'te 3 boyutlu ortamdaki dizilim görülmektedir. Program tüm trafik işaretlerini, seçilen arkaplanları ve ışığı rastgele belirleyerek; konum ve yönleri de rastgele değiştirerek çıktı üretmektedir. Bu sayede büyük miktarda ilgi alanı (ROI) verisi elde edilebilmektedir. Şekil 2.5'de yapay olarak üretilen ilgi alanları görülmektedir.



Şekil 2.4. 3 boyutlu ortamda yapay veri üretimi.



Şekil 2.5. Yapay veri üretimine ait örnek görüntüler.

2.3 Kullanılan Araçlar

Görüntü işleme tabanlı yaklaşım ve derin öğrenme tabanlı yaklaşım Python 3.5 [31] kullanılarak hazırlanmıştır. Matris işlemleri için Numpy [32], görüntü işleme adımları için OpenCV [33], yapay sinir ağları için TensorFlow [34] ve Keras [35], veri çoğaltma için ImgAug [36] ve SVM [20] için Scikit [37] kütüphaneleri kullanılmıştır. ROI çoğaltma işlemi Unity3D [30] oyun motoru ve C# kullanılarak hazırlanmıştır.

2.4 Görüntü İşleme Tabanlı Yaklaşım Uygulanması

Görüntü işleme tabanlı yaklaşımda, A. Adam and C. Ioannidis'in çalışmasındaki [15] yöntemler uygulanmıştır. Bu yöntemde ilk olarak renk dönüşümü ve renk filtresi uygulanmaktadır. Sonrasında gürültü azaltmak için medyan filtre ve morfolojik işlemler uygulanmaktadır. En son aşamada kalan parçalar (blob) boyutlarına göre filtrelenip sonuç olarak alınmaktadır. Görüntü işleme tabanlı yaklaşım aşamalarının örnek çıktıları Şekil 2.6'da görülmektedir.

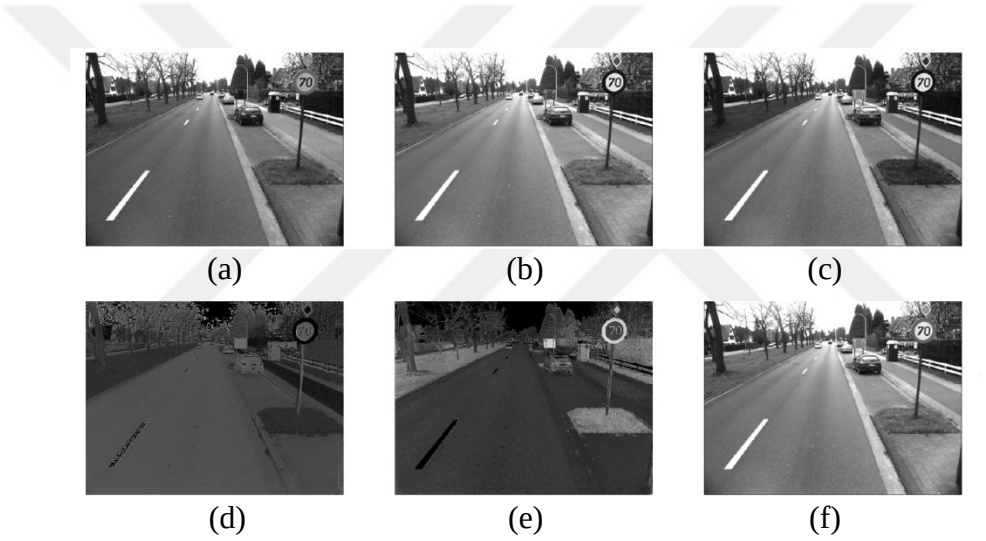


(a) (b) (c) (d) (e)

Şekil 2.6. Görüntü işleme yaklaşımı aşamaları. a) Orjinal görüntü. b) Renk filtresi sonucu. c) Medyan filtre uygulanması. d) Morfolojik işlemlerin uygulanması. e) Parça filtresi uygulanması.

2.4.1 RGB - HSV Dönüşümü

Günümüzde en yaygın kullanılan renk modeli RGB'dir bu modelde renk kırmız, yeşil ve mavi değerlerinin birleşimi olarak saklanır. Ancak insanları renk algılaması bu şekilde işlemediği için bu modele alternatif olarak HSV (Hue, Saturation, Value) modeli geliştirilmiştir. Bu modelde ışık bilgisi ayrı bir kanalda tutulmakta, diğer kanallar ışık bilgisinden çok daha az etkilenmektedir. Bu nedenle görüntü işleme problemlerinde farklı koşullar için daha uygun olmaktadır. Dolayısıyla ilk adım olarak RGB değerleri HSV değerlerine dönüştürülmüştür. Şekil 2.7'de RGB ve HSV için örnek görüntü üzerinden renk kanalları ayrı ayrı verilmiştir.



Şekil 2.7. (a,b,c) R,G,B kanalları, (d,e,f) H,S,V kanalları

2.4.2 HSV Renk Filtresi

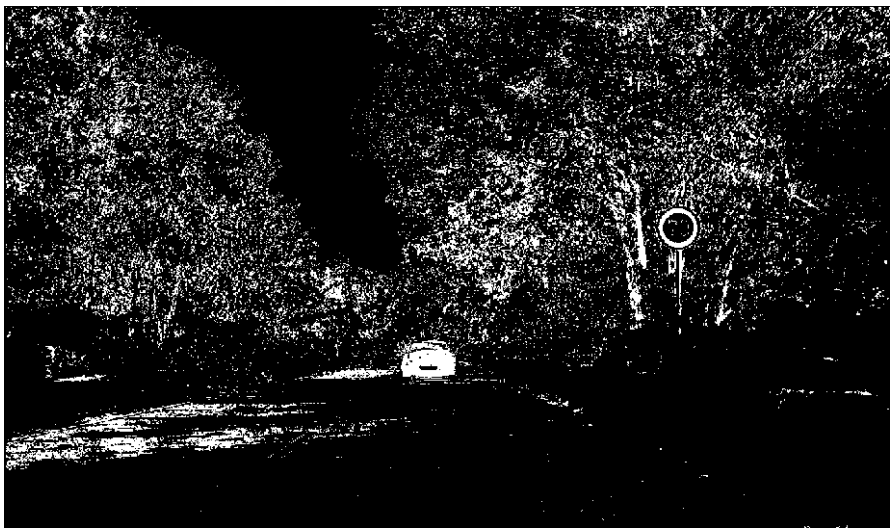
Bu aşamada sadece Hue (H) ve Saturation (S) renk bilgileri kullanarak eşik seviyesi filtresi uygulanmıştır. Filtrenin ile elenen pikseller siyah (sıfır), kalanlar beyaz (bir) olarak işaretlenerek ikili değere sahip görüntü elde edilmiştir. Eşik filtresi için belirlenen değerler Tablo 2.2'te verilmiştir. Bu değerler trafik işaretlerinin ana renklerini tespit edebilmek için belirlenmiştir. Şekil 2.8'da yöntem uygulanmadan önceki orjinal görüntü yer almaktadır. Renk filtre uygulandıktan sonra elde edilen görüntü Şekil 2.9'da verilmiştir.

Tablo 2.2. Eşik seviyesi filtresi için kullanılan değerler.

	Mavi	**Kırmızı**	**Sarı**
Hue (H)	180 < H < 290	H < 22 ya da H > 306	18 < H < 72
Saturation (S)	S > 102	S > 38	0.5 < S < 204



Şekil 2.8. Renk filtreleme yöntemi için örnek girdi görüntüsü.



Şekil 2.9. Renk filtreleme işleminden sonra elde edilen görüntü.

2.4.3 Medyan Filtre

Elde edilen ikili değerli görüntüde gürültü miktarı fazladır. Görüntü işlemede gürültü azaltma amaçlı kullanılan temel yöntem medyan filtre uygulamaktır. Bu aşamada medyan filtre uygulanarak gürültü azaltılmıştır. Şekil 2.10'da HSV renk filtresinden sonra uygulanan medyan filtrenin sonucu görülmektedir. Medyan filtre belirlenen büyüklükte filtreyi resimde gezdirerek, filtrenin medyan değerini piksele yazması olarak çalışmaktadır. Bu filtre için 3x3 filtre büyüklüğü seçilmiştir.



Şekil 2.10. HSV renk filtresi görüntüsü üzerinde medyan filtre uygulanması sonucu.

2.4.4 Morfolojik Açma İşlemi

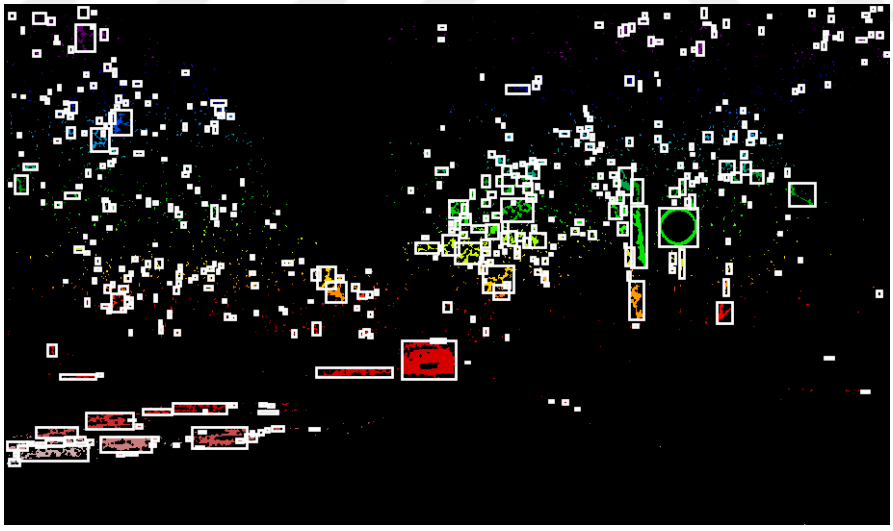
Bu aşamaya kadar elde edilen görüntüde ağaç ve benzeri nesnelerden dolayı yanıltıcı alanlar bulunmaya devam etmektedir. Morfolojik işlemlerden olan açma işlemi ile bu yanıltıcı alanların azaltılması sağlanmıştır. Morfolojik işlemlerin 2 temel işlemden oluşur. Bunlar aşındırma (Erosion) ve genişletmedir (Dilation). Aşındırma işleminde belirlenen büyüklükteki filtreye denk gelen piksellerin tamamı dolu ise filtreden geçmiş sayılır, yani nesneler küçülmüş olur. Genişletme işleminde ise tam tersi olarak, herhangi bir piksel dolu ise filtreden geçmiş kabul edilir. Bu da görüntüdeki nesnenin büyümesine neden olur. Açma (Opening) işleminde ise aşındırma uygulandıktan sonra genişletme uygulanır. Bu sayede nesneler arasındaki ince ve küçük bağlantılar

giderilmiş olur. Bu aşamada filtre büyüklüğü 3x3 olarak seçilmiştir. Şekil 2.11'de medyan filtresi sonrasında uygulanan açma işleminin sonucu görülmektedir.



Şekil 2.11. Medyan filtre sonrası açma işleminin uygulanması ile elde edilen görüntü.

2.4.5 Piksel Kümeleri Tespiti ve Filtresi

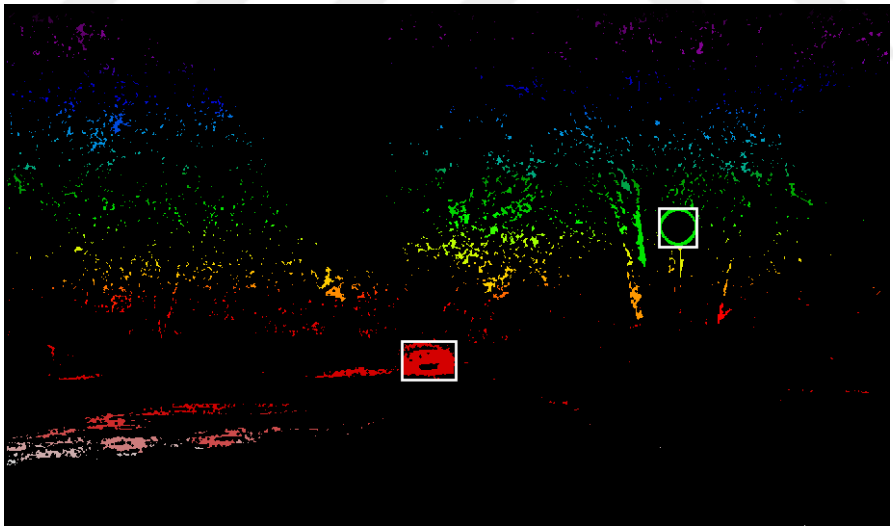


Şekil 2.12. Tespit edilen piksel kümeleri.

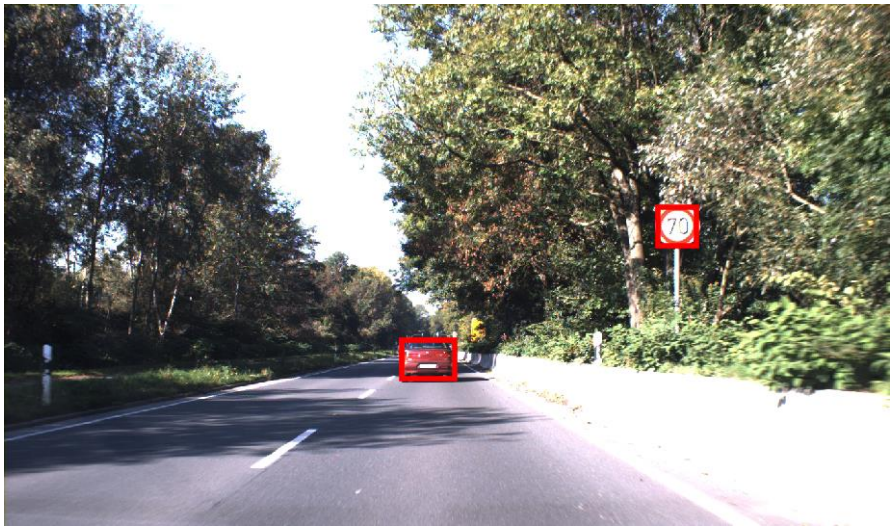
Elde edilen görüntüde birbirine bağlı piksel kümeleri bulunmaktadır. Bu kümelerin belirlenmesi ile birlikte, boyutlarına ve oranlarına bakarak filtreleme uygulanmıştır. Bu sayede gereksiz kümeler elenmiştir. Filtrelenerek elde edilen piksel kümeleri ilgi alanlarını oluşturmaktadır. Örnek görüntü için elde edilen piksel kümeleri Şekil 2.12'te görülebilir. Tablo 2.3'te belirtilen değerlere göre filtre uygulandıktan sonra kalan piksel kümeleri Şekil 2.13'te ve bu kümelerin orjinal görüntüdeki karşılığı Şekil 2.14'te görülmektedir.

Tablo 2.3. Küme filtresi için değerler.

	En az	**En fazla**
Alan (piksel)	500	5000
En/Boy oranı	0.4	1.6



Şekil 2.13. Piksel kümelerine filtre uygulandıktan sonra kalan ilgi alanı adayı bölgeler.



Şekil 2.14. Görüntü işleme tabanlı yaklaşım uygulaması sonucu tespit edilen ilgi alanı adayı bölgelerin orjinal görüntüdeki karşılıkları.

2.5 Derin Öğrenme Yaklaşımı Uygulanması

Derin öğrenme yaklaşımında sınırlayıcı kutu tespiti ve sınıflandırma için farklı yapay sinir ağları eğitilmiştir. Buradaki amaç farklı verisetleri ile sınırlayıcı kutu bulucu eğitip veri seti yetersiz olan ülkelere ait görüntülerde deneyebilmektir. Bu amaçla YOLO mimarisindeki sınırlayıcı kutu bulucu yapısı uygulanmıştır.

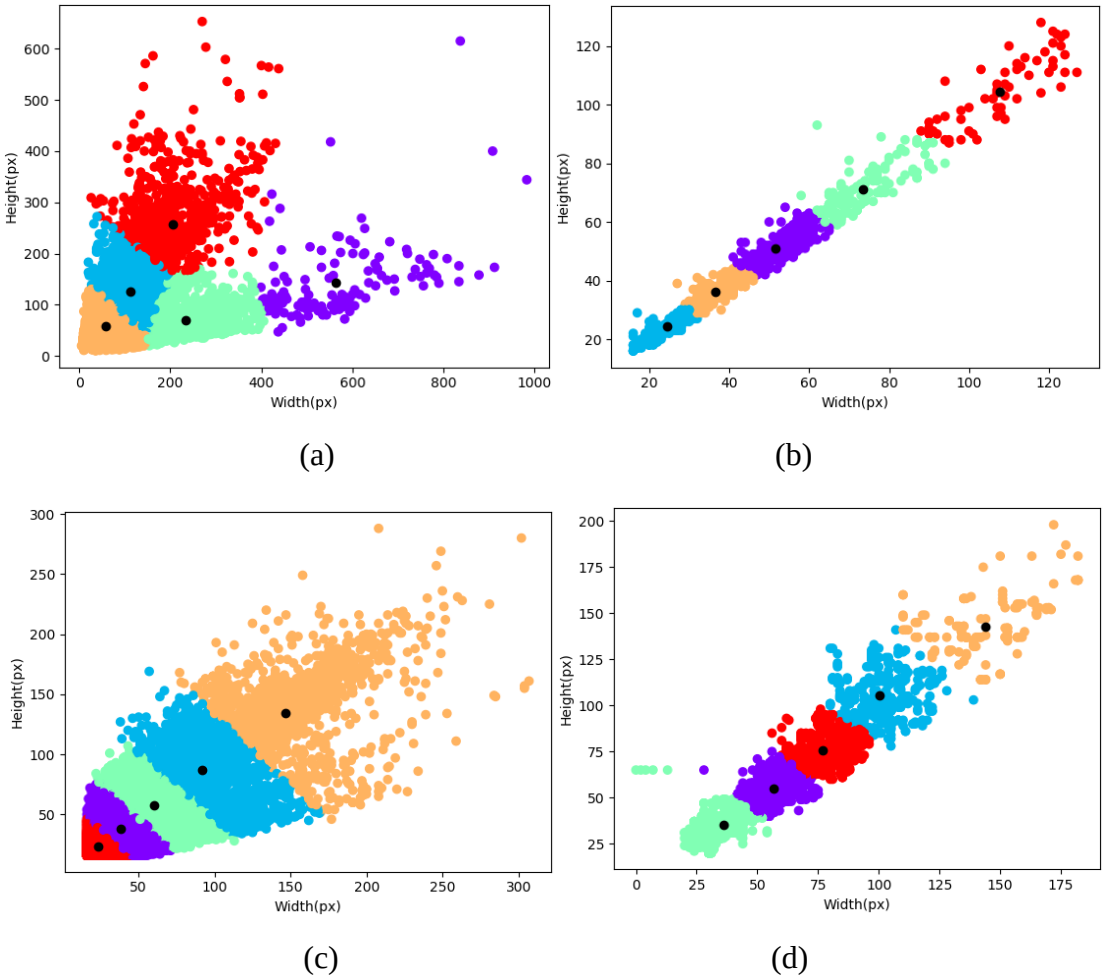
2.5.1 Sınırlayıcı Kutu Bulucu

YOLO mimarisinde görüntü hücrelere ayrılır. Model, her bir hücre için N adet tahminde bulunur. Bu tahminlerde güvenilirlik skoru, koordinatlar ve sınıflandırma sonucu bulunur. Bu tezde yalnızca sınırlayıcı kutu tespiti amaçlı kullanıldığı için sınıflandırma çıktısı kullanılmamıştır. Bu mimari için gerekli olan girdi için görüntü hazırlık sürecinden geçirilmiştir. Bu aşamada sınırlayıcı kutunun denk geldiği hücre ve hücrenin merkezine olan uzaklık kesin doğru olarak girilmiştir. Mimari çıktı üretip kesin doğru ile olan farka bakarak kendini eğitmektedir. Bunlara ek olarak YOLO v2 mimarisinde kullanılan çapa kutuları yöntemi kullanılmıştır. Bu yöntemde her hücre için belirli sayıda ön tanımlı kutular belirlenmiştir. Hazırlık aşamasında gerçek sınırlayıcı kutunun denk geldiği hücreye en yakın olan çapa kutusuna göre kesin doğru olarak girilmiştir.

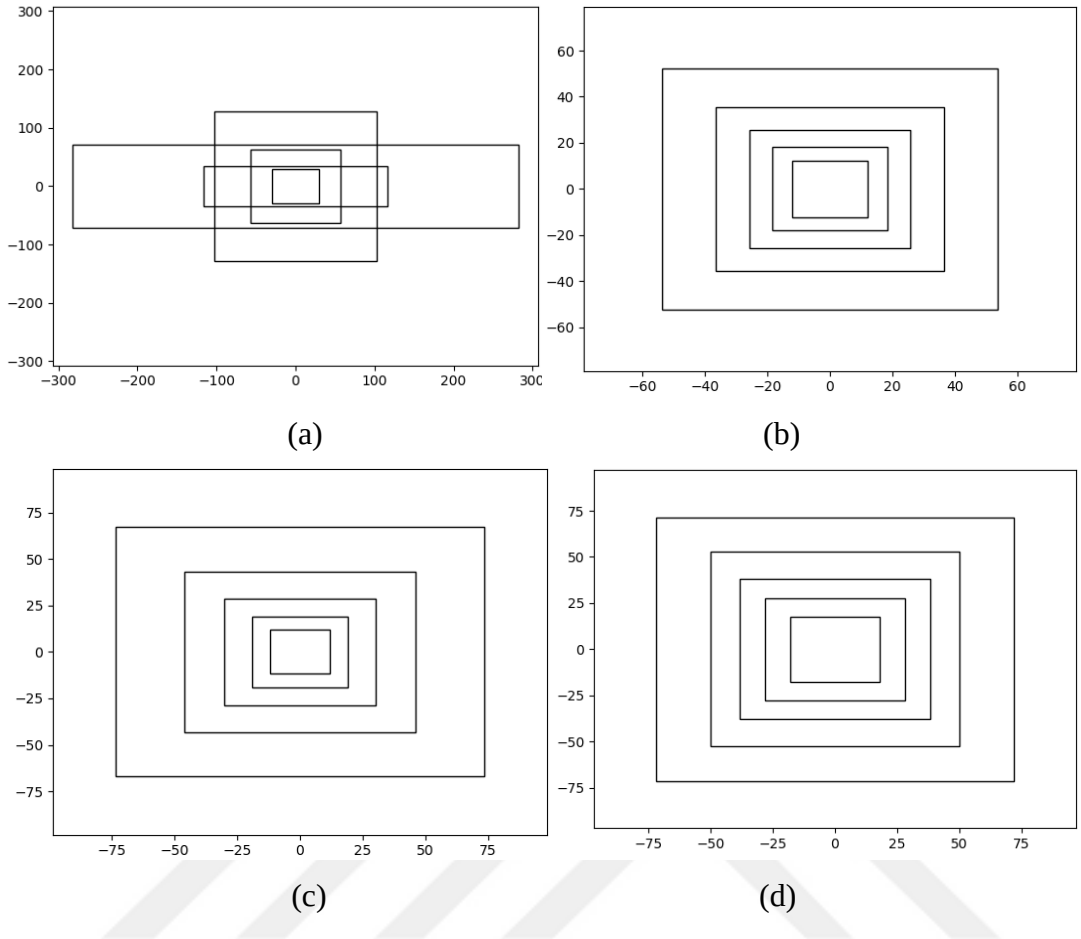
Görüntü 6x8 adet hücreye bölünmüştür. Her hücre için 5 çapa kutusu tanımlanmıştır. Her çapa kutusu; güvenilirlik skoru, tahmin edilen kutunun merkezinin hücreye olan x

ve y eksen farkları ve genişlik ve yüksekliği görüntü genişlik ve yükseliğine oranlarını tahmin eder. Bu nedenle bir görüntü için toplamda 1200 değerden oluşan bir vektör üretmektedir.

Çapa kutularını belirmek için yine YOLO v2 çalışmasında olduğu gibi eğitim verisindeki sınırlayıcı kutuların boyutları ile K-means kümeleme işlemi yapılmıştır. Şekil 2.15'te farklı verisetleri üzerinde yapılan K-means işleminin sonuçları görülmektedir. Şekil 2.16'da ise bu uygulama sonucu merkezlerin çapa kutusu karşılıkları görülmektedir.



Şekil 2.15. Veri setlerindeki trafik işareti boyutları kullanılarak K-means uygulanması. a) BelgiumTS, b) GTSDB, c) RTSD, d) TRTSD.



Şekil 2.16. Veri setlerine K-means uygulandıktan sonra çıkan merkezlerin çapa kutusu karşılıkları. a) BelgiumTS, b) GTSDB, c) RTSD, d) TRTSD.

2.5.1.1 Kayıp Fonksiyonu

Yapay sinir ağlarının temel prensibi; yaptığı tahmin ile verisetindeki kesin doğru arasındaki farkı gözeterek, katmanların parametrelerini ayarlamasıdır. Bu süreç geri yayılma algoritması (*backpropagation*) olarak adlandırılmaktadır. Yapay sinir ağları optimizasyon problemlerini çözmek için oluşturulmuş yapılardır. Yapay sinir ağı oluşturulduğu ilk anda, nöronların ağırlık değerleri rastgele değerler olarak atanır. Verisetinden bir miktar veri seçilir (*batch*) ve yapay sinir ağının ürettiği sonuç hesaplanır. Verisetinde işaretlenmiş kesin değerler ile yapılan tahmin arasındaki fark, belirlenen kayıp fonksiyonu (*loss function*) ile hesaplanır. Bu fark geri yayılma algoritması tarafından yapay sinir ağındaki nöronların ağırlıklarının optimize edilmesi

için kullanılır. Bu optimizasyonun çok küçük oranda yapılmaktadır. Bu nedenle veri tekrar tekrar aynı sürece (*epoch*) girmektedir. Epoch sayısının fazla olması yapay sinir ağının sonuca yakınsaması için gereklidir. Bu da eğitim süresinin artmasına neden olmaktadır. Kayıp fonksiyonunun hatayı iyi şekilde temsil edebilmesi; yapay sinir ağının sonuca daha hızlı yakınsamasını, dolayısıyla eğitim süresinin azalmasını sağlayabilmektedir.

YOLO modelinde tahmin, SxS hücreye bölünmüştür. Her bir hücre için B adet sınırlayıcı kutu tahmini ile sonuç temsil edilir. Kutular, hücrenin konumuna olan x ve y kayma miktarını, hücreye olan genişlik (w) ve yükseklik (h) oranını ve sınıf tahminini (C) içermektedir. YOLO modelinde kayıp fonksiyonu 5 kısma ayrılmıştır. Bu 5 kısmın sonuçları toplanarak kayıp değeri elde edilir. İlk kısım

$$\boldsymbol{\lambda_{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbf{1}_{ij}^{\boldsymbol{obj}} [(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2] \tag{2.1}$$

şeklinde tanımlanmıştır. Nesne bulunan ($\mathbf{1}_{ij}^{\boldsymbol{obj}}$) her hücre için x ve y konum tahminlerinin işaretlenmiş verideki kesin doğru ile farkları elde edildikten sonra, karelerinin toplamı alınarak $\boldsymbol{\lambda_{coord}}$ katsayısı ile çarpılır. Bu katsayının amacı probleme göre daha uygun sonuç alabilmek için ayarlanabilir bir fonksiyon oluşturmaktır. Model, bu kısım sayesinde tahminlerinin konumunu verisetine yakınsayarak öğrenmektedir. İkinci kısım benzer şekilde

$$+ \boldsymbol{\lambda_{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbf{1}_{ij}^{\boldsymbol{obj}} \left[\left(\sqrt{w_i} - \sqrt{\widehat{w}_i} \right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right] \tag{2.2}$$

olarak tanımlanmıştır. Genişlik (w) ve yükseklik (h) tahminlerinin gerçek veri ile olan farkı alınmaktadır. Ancak buradaki farklar büyük olabileceği için etkisini azaltmak adına karekökleri alınmaktadır. Model bu kısım ile nesnelerin genişlik ve yüksekliğini öğrenmektedir. Üçüncü kısım sınıf tahminindeki hatayı tespit etmek için

$$+ \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbf{1}_{ij}^{obj} \left(C_i - \hat{C}_i\right)^2 \tag{2.3}$$

şeklinde tanımlanmıştır. Bu kısımda her işaretli nesne içeren kutu için sınıf tahmini ile gerçek sonucun farkının karesi hesaplanır. Bu kısımda model içeriğin sınıflandırılmasını öğrenmektedir. Dördüncü kısım ise

$$+ \boldsymbol{\lambda_{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbf{1}_{ij}^{noobj} \left(C_i - \hat{C}_i\right)^2 \tag{2.4}$$

şeklinde tanımlanmıştır. İşaretli nesne içermeyen ($\mathbf{1}_{ij}^{noobj}$) hücreler için tahmin ile gerçek değer arasındaki fark hesaplanmaktadır. Bu kısımda $\boldsymbol{\lambda_{noobj}}$ katsayısının amacı, nesne içermeyen hücrelerin kayıp fonksiyonuna etkisini azaltmaktır. Görüntü üzerinde çoğu hücre nesne içermediği için düşük katsayı ile bu etki en aza indirilmektedir. Bu kısım modelin arkaplan ile nesne ayrımını yapabilmesini sağlamaktadır. Son kısım

$$+ \sum_{i=0}^{S^2} \mathbf{1}_{i}^{obj} \sum_{c \in classes} \left(p_i(c) - \hat{p}_i(c)\right)^2 \tag{2.5}$$

nesne içeren hücreler için güvenilirlik skoru ($p_i(c)$) farklarının kareleri hesaplamaktadır. Böylece model yaptığı tahminin ne kadar güvenilir olduğunu da belirtmektedir. Modelin tahminleri bu değere göre filtrelenmektedir. Güvenilirlik skoru,

$$IOU = \frac{B_1 \cap B_2}{B_1 \cup B_2} \tag{2.6}$$

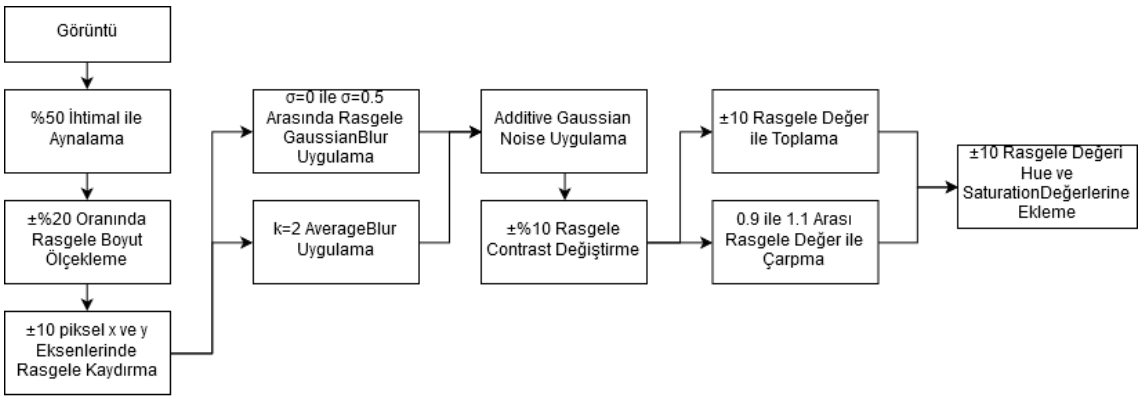
denklemi ile elde edilmektedir. Skor sonucu, tahmin edilen bölge ile işaretlenmiş kesin doğru bölgenin kesişim oranı ile toplam alanına bölünmesi ile elde edilmektedir (*Intersection Over Union - IOU*).

Kayıp fonksiyonunun ürettiği sonuç, SxS hücre için B adet (5 + sınıf miktarı) büyüklüğünde bir vektördür. Bu vektörün değerlerinin 0 ile 1 arasında kalmasını sağlamak adına x, y, w ve h değerleri görüntünün boyutuna oranlanarak girilmiştir. C sınıf tahmini, her sınıf için 0 ile 1 arasında elde edilmektedir. Aynı şekilde güvenilirlik skoru($p_i(c)$)) 0 ile 1 arasındadır.

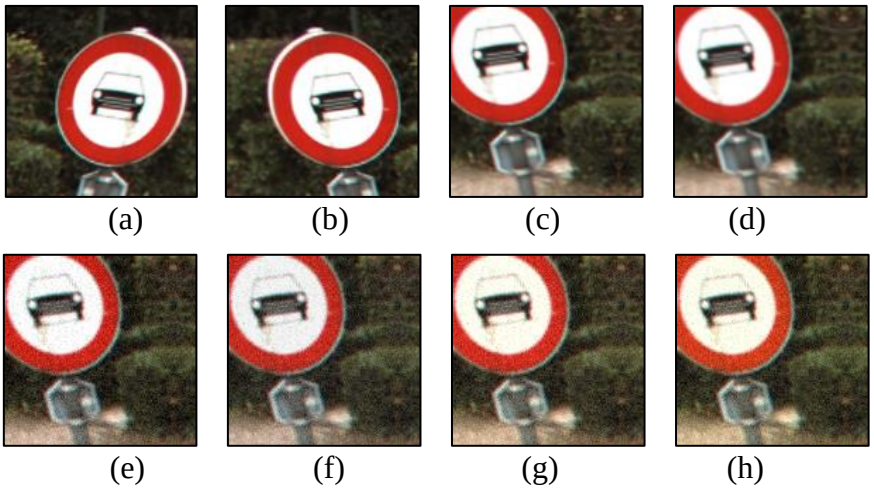
Sınırlayıcı kutu bulucu uygulamasında sınıf adedi 1 kabul edilmiştir. Böylece model sadece trafik işareti varlığını öğrenmektedir. Derin öğrenme uygulamasında ise kullanılan verisetindeki sınıf miktarı kullanılmıştır. Bu sayede model hem sınırlayıcı kutu tespitini hem de sınıflandırmayı, işaretlenmiş verisetine yakınsayarak öğrenebilmiştir.

2.5.2 Görüntü Çoğullama

Yapay sinir ağlar yığın (batch) eğitimini desteklemektedir. Bu nedenle aynı veri değişiklik yapılarak tekrar kullanılabilmektedir. Miktarı az olan veri setleri için bu yapay veriler oluşturabilme anlamına gelmektedir. Aynı zamanda modelin başarısını da artırmaktadır [7]. Bu amaçla eğitim setindeki görüntüler modele verilmeden önce konum kaydırma, renk değiştirme, aynalama ve yeniden boyutlandırma gibi adımlar rastgele uygulanarak yapay çoğaltma sağlanmıştır. Şekil 2.17'de çoğullama adımları akış diyagramı olarak verilmiştir. Şekil 2.18'de her adım için örnek çıktılar gösterilmiştir.



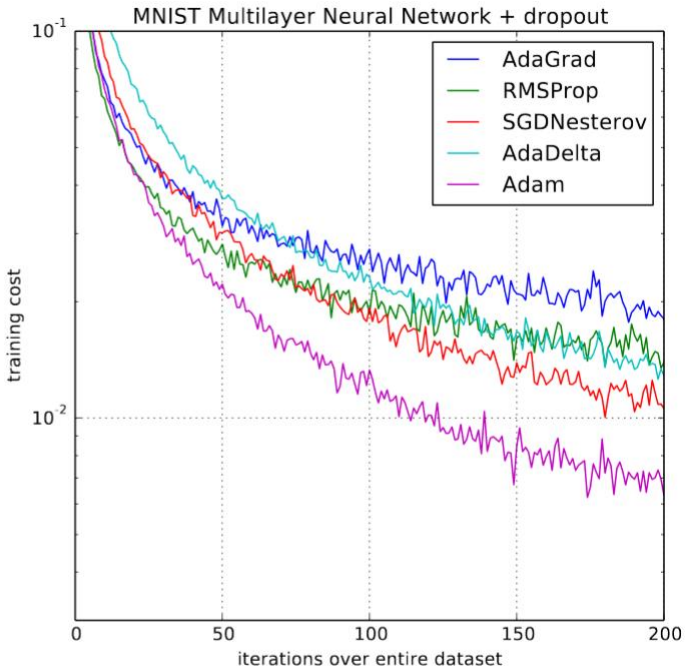
Şekil 2.17. Yapay veri üretimi akış diyagramı.



Şekil 2.18. Yapay veri üretimi aşamalarının örnek çıktıları. a) Orijinal görüntü, b) Aynalama, c) Yeniden boyutlandırma ve kaydırma, d) Bulanıklaştırma, e) Gürültü ekleme, f) Kontrast değiştirme, g) Rastgele değer ile toplama, h) Hue ve Saturation ekleme.

2.5.2.1 En İyileme

Hesaplanan kayıp değeri ile modelin parametrelerini hangi yönde değiştireceğini belirlemek adına farklı en iyileme (optimizer) yöntemleri uygulanabilmektedir. YOLO mimarisinde kullanılan ADAM (Adaptive Moment Estimation) [38] en iyileme yöntemi bu tezde kullanılmıştır. ADAM, RMSPROP (Root Mean Square Propagation) [39] ve ADAGRAD (Adaptive Gradient Descent) [40] en iyileme yöntemlerinin avantajlarını kullanan bir yöntemdir. Bu en iyileme yöntemi ile birlikte model daha hızlı ve başarılı eğitim gerçekleştirmektedir. Şekil 2.19'da, ADAM [38] çalışmasında karşılaştırılan farklı iyileme yöntemlerinin performansı görülmektedir. En iyileştirme uygulamalarında patlayan/sönen gradyanlar problemi görülebilmektedir. En iyileştirme yönteminin öğrenme oranı yüksek ise modeldeki nöron ağırlıkları, geri yayılım sürecinde sonsuza yakınsayarak eğitim sırasındaki hesaplamaları bozabilmektedir. Aynı şekilde çok düşük öğrenme oranı, ağırlıkların sıfıra yakınsamasına ve hesaplamaların yapılamamasına neden olabilmektedir. Bu nedenle en iyileme yöntemlerinde, yöntem parametrelerin el ile ayarlanması gerekmektedir.



Şekil 2.19. En iyileme metodlarının kayıp değeri ve iterasyon karşılaştırması [38].

2.5.2.2 Azami Baskılama (Non-Maximum Supression)

YOLO mimarisinde görüntü hücrelere bölünüp, her hücre farklı tahmin yaptığı için; farklı hücreler birbirine yakın sonuçlar üretebilmektedir. Son aşamada bu tekrarları kaldırmak için azami baskılama yöntemi uygulanmaktadır. Bu yöntemde %50'den fazla kesişim gösteren sınırlayıcı kutular elenmektedir. Kesişim miktarının tespiti için IOU (Intersection over Union) değeri kullanılmaktadır.

2.5.2.3 Yapay Sinir Ağı Modeli

Yapay sinir ağı modeli YOLO mimarisinde olduğu üzere evrişim katmanlarından oluşmaktadır. Girdi görüntü 416x416 boyutuna küçültülmüştür. Toplamda 23 CNN katmanı bulunmaktadır.

Tablo 2.4'te katmanlar, çekirdek boyutları,derinlik ve çıktı boyutları listelenmiştir. Bu katmanlar giderek derinleşmekte ve boyutları küçülmektedir. Overfit önlemek adına her katmanda LeakyRelu aktivasyonu ve Batch Normalization kullanılmıştır. Model toplamda 60 milyon parametre içermektedir.

Tablo 2.4. Derin öğrenme modelinde kullanılan katmanlar.

Katman	Kernel	Derinlik	Stride	Çıktı Boyutu
Giriş	-	-	-	416x416x3
Convolution 1	3x3	32	1	416x416x32
Max Pooling	2x2	-	2	208x208x32
Convolution 2	3x3	64	1	208x208x64
Max Pooling	2x2	-	2	104x104x64
Convolution 3	3x3	128	1	104x104x128
Convolution 4	1x1	64	1	104x104x64
Convolution 5	3x3	128	1	104x104x128
Max Pooling	2x2	-	1	52x52x128
Convolution 6	3x3	256	1	52x52x256
Convolution 7	1x1	128	1	52x52x128
Convolution 8	3x3	256	1	52x52x256
Max Pooling	2x2	-	1	26x26x256
Convolution 9	3x3	512	1	26x26x512
Convolution 10	1x1	256	1	26x26x256
Convolution 11	3x3	512	1	26x26x512
Convolution 12	1x1	256	1	26x26x256
Convolution 13	3x3	512	1	26x26x512
Max Pooling	2x2	-	2	13x13x512
Convolution 14	3x3	1024	1	13x13x1024
Convolution 15	1x1	512	1	13x13x512
Convolution 16	3x3	1024	1	13x13x1024
Convolution 17	1x1	512	1	13x13x512
Convolution 18	3x3	1024	1	13x13x1024
Convolution 19	3x3	1024	1	13x13x1024
Convolution 20	3x3	1024	1	13x13x1024
Convolution 21	1x1	1024	1	13x13x1024
Convolution 22	3x3	1024	1	13x13x1024
Convolution 23	1x1	Anchor Box * (5 + Sınıf adedi)	1	13x13x(Anchor Box * (5 + Sınıf adedi))

2.5.2.4 Modelin Eğitimi

Hazırlanan model farklı veri setleri ile eğitilmiştir. Eğitim Nvidia GTX1080 ekran kartı üzerinde çalıştırılmıştır. Her bir batchte 32 görsel olmak üzere, maksimum 150 epoch olarak belirlenmiştir. Her bir veri seti %80 eğitim, %10 test ve %10 validasyon seti olarak bölünmüştür. Verilen her eğitim görseli görüntü çoğaltma aşamasından geçirilerek verilmiştir. Validasyon seti orjinal şekli ile kullanılmıştır.

2.5.3 Sınıflandırma Modeli

Sınıflandırma modeli için YOLO benzeri bir yapay sinir ağı modeli oluşturulmuştur. Yine aynı şekilde ADAM en iyileme yöntemi kullanarak eğitilmiştir. Tablo 2.5'de bu modelin katmanları ve detayları verilmiştir.

Tablo 2.5. Sınıflandırma modelinde kullanılan katmanlar.

Katman	Kernel	Derinlik	Stride	Çıktı Boyutu
Giriş	-	-	-	32x32x3
Convolution 1	7x7	64	1	32x32x64
Max Pooling	2x2	-	2	16x16x64
Convolution 2	3x3	128	1	16x16x128
Max Pooling	2x2	-	2	8x8x128
Convolution 3	1x1	128	1	8x8x128
Convolution 4	3x3	196	1	8x8x196
Max Pooling	2x2	-	2	4x4x196
Convolution 5	1x1	196	1	4x4x196
Convolution 6	3x3	256	1	4x4x256
Convolution 7	1x1	256	1	4x4x256
Dense	-	-	-	Sınıf adedi

3. BULGULAR VE TARTIŞMA

3.1 Görüntü İşleme ile Sınırlayıcı Kutu Tespiti Performansı

Bu yöntem, GTSDB veriseti ile denenmiş, örnek sonuçlar Şekil 3.1'de verilmiştir. Yöntem sadece renk bilgisine bakıp, kavramsal bilgiden yoksun olduğu için başarı oranı düşük çıkmıştır. Öte yandan seçici arama yöntemine göre daha hızlı ve daha az sayıda ilgi alanı adayı çıkartmaktadır. İşlemleri her bir görüntü için yapmak gerektiğinden, gerçek zamanlı uygulamalar için yeterli performansı sağlayamamaktadır. Ayrıca renk eşik değerlerinin elle girilmesi gerektiğinden, farklı koşullarda çekilmiş görüntülere karşı toleransı düşüktür.



Şekil 3.1. Görüntü işleme yönteminin GTSDB veriseti üzerindeki tespit sonuçları.

3.2 Derin Öğrenme ile Sınırlayıcı Kutu Tespiti Performansı

Bu yöntem, görüntü işleme yöntemine ek olarak arkaplan ayrımını da yapabildiği için bulduğu ilgi alanı adaylarının sayısı az ancak başarı oranı yüksektir. Model, YOLO modelinin sınıf tespitini içermeyen halidir. Modeli eğitme aşaması uzun sürmesine rağmen; gerçek zamanlı çalışma performansı, görüntü işleme yöntemine göre yüksektir. Model, parametreleri verisetinden kendisi öğrenmektedir. Görüntü işleme yönteminde ise uzman tarafından parametrelerin ayarlanması gerekmektedir. Bu yöntem, görüntüde küçük yer kaplayan nesnelerin tespitinde düşük başarıya sahiptir. Modelin eğitimi BelgiumTS veriseti ile 254 epoch olmak üzere yaklaşık 18 saat sürmüştür. TRTSD verisetinde ise gündüz görüntüleri kullanılarak 50 epoch ile eğitim tamamlanmıştır. Gece görüntüleri ise yeterli miktarda olmaması ve görüntülerdeki zorluklar nedeniyle başarılı bir şekilde eğitilememiş, model overfit (eğitim setinin ezberlenmesi) olmuştur. Modelin başarısı mAP (Mean Average Precision) skoru ile hesaplanmıştır. mAP skoru her bir sınıf için bulunan AP (Average Precision) skorlarının ortalaması alınarak hesaplanmaktadır. AP skoru şu şekilde tanımlanmaktadır:

$$Precision = \frac{TP}{TP + FP} \tag{3.1}$$

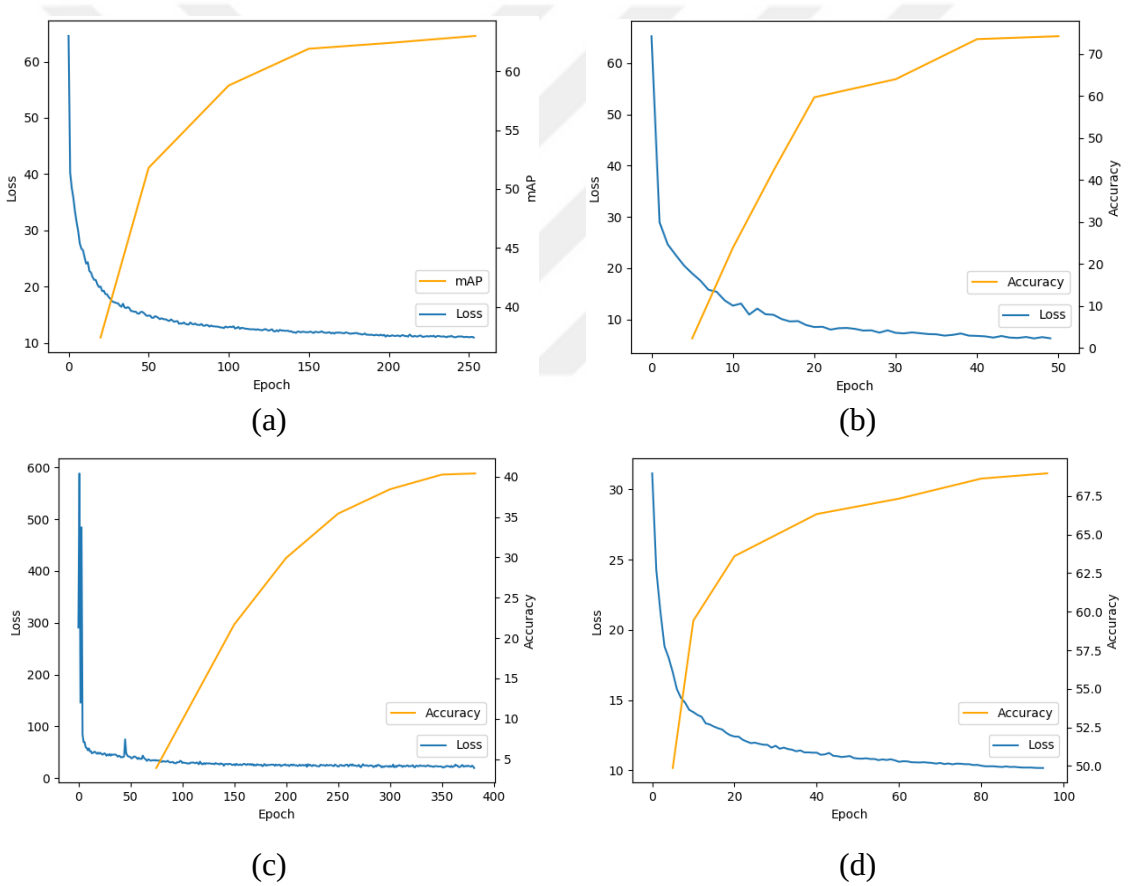
$$Recall = \frac{TP}{TP + FN} \tag{3.2}$$

Bu denklemlerde TP true positive, FP false positive ve FN false negative'dir. AP skoru hesabını kolaylaştırmak adına 0 ile 1 arasında 0.1 birimlik artışlar ile değerler alınarak alan hesaplanmaktadır. Bu yöntemde sınıf bilgisi trafik işareti bulunup bulunmamasını göstermektedir.

$$AP = \frac{1}{11}\sum_{i=0}^{10} Precision(Recall_i) \tag{3.3}$$

Eğitimin sonucunda BelgiumTS %63.33 mAP skoruna ulaşılmıştır. TRTSD gündüz veriseti ise %74.26 mAP skoruna ulaşmıştır. GTSDB veriseti %39.07, RTSD veriseti %68.97 mAP skoruna ulaşmıştır. Şekil 3.2'de her bir epoch için zamanla kayıp miktarının azaldığı ve mAP skorunun arttığı görülmektedir. Bu eğilim modelin öğrenme

performansının başarılı olduğunu ve veriseti miktarının yeterli koşulları sağladığını göstermektedir. Şekil 3.3, Şekil 3.4, Şekil 3.5 ve Şekil 3.6'da örnek tespit sonuçları görülmektedir. Veriseti; görüntüde trafik işareti bulunmasına rağmen, etiketlenmiş veride bulunmayan görüntüler içermektedir. Bu tip görüntüler, modelin yaptığı tahminin kayıp fonksiyonunda yanlış olarak değerlendirmesine neden olmaktadır. Bu nedenle başarı oranını azaltan nedenlerden biri olmuştur. Başarı oranının artmasını engelleyen bir diğer neden, YOLO mimarisinde her bir hücre için tek tahmin yapılabiliyor olmasıdır. Bu nedenle birbirine yakın ve görüntüde uzakta durduğu için küçük olan nesneler tahmin edilememektedir.



Şekil 3.2. Derin Öğrenme ile sınırlayıcı kutu tespiti için kayıp ve mAP grafiği, a) BelgiumTS, b) TRTSD, c) GTSDB, d) RTSD.



Şekil 3.3. Derin Öğrenme ile sınırlayıcı kutu tespiti için BelgiumTS veriseti sonuçları.



Şekil 3.4. Derin Öğrenme ile sınırlayıcı tespiti için TRTSD gündüz veriseti sonuçları



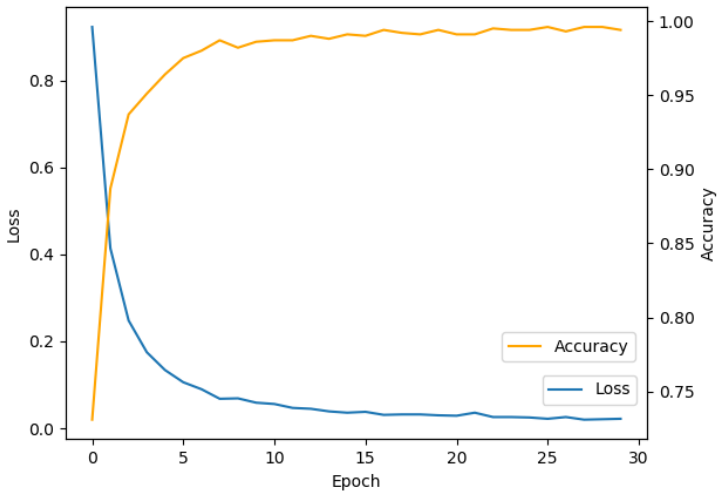
Şekil 3.5. Derin Öğrenme ile sınırlayıcı kutu tespiti için GTSDB veriseti sonuçları



Şekil 3.6. Derin Öğrenme ile sınırlayıcı kutu tespiti için RTSD veriseti sonuçları

3.3 Derin Öğrenme ile Sınıflandırma Modeli Performansı

Bu yöntemdeki avantaj, sınıflandırma verisetinin farklı olabilmesidir. Bu sayede mevcut veriseti, yapay üretilmiş ilgi alanı veriseti ile desteklenip başarı oranı artırılabilmiştir. Eğitim için GTSRB veriseti kullanılmıştır. Eğitim 30 epoch sürmüştür. Eğitimin sonucunda %99.1 başarı oranına ulaşılmıştır. Şekil 3.7'te her bir epoch için kayıp değerinin azaldığı, başarı oranının arttığı görülmektedir.



Şekil 3.7. Derin öğrenme ile sınıflandırma kayıp ve başarı grafiği.

3.4 YOLO Modeli Performansı

YOLO modeli görüntüyü bir yapay sinir ağı ile hem ilgi alanı hem de sınıflandırma sonucunu tek seferde üretebilmiştir. Bu özelliği, diğer yöntemlere göre oldukça yüksek bir performans başarısı elde edebilmesini sağlamıştır. Bu nedenle gerçek zamanlı uygulamalara oldukça uygundur. Ancak bu hız karşılığında küçük nesnelerin tespitinde düşük başarıya sahiptir. Eğitim için BelgiumTS veriseti kullanılmıştır. Bu verisetinde 13 adet üst sınıf bilgisi bulunmaktadır. Tablo 3.1'de görülebileceği üzere, her bir sınıf için farklı başarı oranları alınmıştır. Bunun nedeni verisetinde bulunan miktarların farklı olmasıdır. mAP skoru için ağırlıklı ortalama %39.80 olmuştur. Sınırlayıcı kutu tespiti ile elde edilen %63 mAP skoruna göre düşük olma nedeni, yanlış tahmin edilen sınıflar için sınırlayıcı kutunun yanlış kabul edilmesidir.

Tablo 3.1. YOLO Modelinin BelgiumTS sonuçları.

Sınıf	**Adet**	**mAP**
1	297	39.94
2	288	31.26
3	111	38.47
4	169	47.17
5	160	63.43
6	62	10.35
7	50	10.17
8	43	47.90
9	8	40.00
10	42	52.83
11	47	49.76
12	78	31.24
13	6	66.67
Ağırlıklı Ortalama		**39.80**

4. SONUÇLAR

Görüntü işleme yöntemleri ile sınırlayıcı kutu tespiti en düşük başarı oranına sahiptir. Bu yöntemleri sınırlayıcı kutu tespiti için kullanmak yerine, veri setini çoğullama amacıyla kullanmak çok daha etkili bir çözüm olmuştur. Sınırlayıcı kutu tespiti için yapay sinir ağları kullanımı daha yüksek başarı oranlarına ulaşılabilmesini sağlamıştır. Aynı zamanda gerçek zamanlı uygulamalar için gerekli performansı da sağlayabilmektedir. Sınıflandırma için ise ayrı bir yapay sinir ağı modeli eğitmek, farklı veri seti ve yapay veri kullanımını mümkün kıldığı için yüksek başarı oranına ulaşabilmiştir. Öte yandan sınırlayıcı kutu tespiti ve sınıflandırmayı tek model ile yapan YOLO yönteminde başarı daha düşük olmasına rağmen, en yüksek performanslı yöntem olmuştur. Derin öğrenme yöntemleri için en önemli kriter, veri setinin yeteri kadar büyük olmasını sağlayabilmektir. Bu amaçla veri çoğullama ve yapay veri üretimi uygulanmıştır.

Eğitimlerin sonucunda BelgiumTS veriseti ile %63.33 mAP skoruna ulaşılmıştır. Görüntülerin sade, benzerliklerinin az ve veriseti büyüklüğünün yeterli olması nedeniyle başarı oranı yüksek çıkmıştır. BeligumTS ile benzer seviyede görüntüler içeren GTSDB veriseti, küçük olması nedeniyle %39.07 mAP skoruna ulaşmıştır. TRTSD gündüz veriseti, %74.26 mAP skoruna ulaşmıştır ancak verisetinde benzer görüntülerin fazla olması ve verisetinin küçük olması nedeniyle model veriyi ezberlemiştir. Aynı şekilde benzer görüntüler içeren ancak yüksek miktarda veri içeren RTSD veriseti ise %68.97 mAP skoruna ulaşmıştır.

Hali hazırda bulunan yabancı veri setleri ve hazırlanan TRTSD veri seti derin öğrenme için yeteri kadar büyük değildir. Veri seti oluşturma, yüksek maliyetli bir iş olduğundan; yapay veri üretimi ve veri çoğullama yöntemleri, başarı oranını belirleyen faktörler olmuştur. Daha büyük verisetleri elde edilebilmesi için aracın üzerine birden fazla kameranın farklı açılarla yerleştirilmesi, en kolay uygulanabilir yöntem olarak değerlendirilebilir. Verisetinin büyüklüğünün yanında verinin çeşitliliği de belirleyici faktördür. TRTSD verisetinin yetersiz kalma nedenlerinden biri, görüntü toplanan ortamların birbirine çok yakın olmasıdır. Çeşitliliği sağlamak için şehrin farklı bölgelerinde otoyol, ana caddeler ve sokaklarda veri toplanması gerekmektedir. Otoyollar farklı verisetlerinde birbirine yakın görüntüler içermektedir. Bu görüntüler

trafik işaretleri için yanıltıcı unsurların en az bulunduğu görüntülerdir. Bu nedenle verisetinde ana cadde ve sokaklara ağırlık verilmesi daha uygun olacaktır. Türkiye için görüntülerde en fazla karşılaşılan yanıltıcı unsur reklam ve işyeri tabelaları olmuştur.

Görüntülerin toplandığı İstanbul için zorluklardan biri ise yoğun trafik olmuştur. Görüntüler video olarak toplanmış olup, aracın sabit durduğu anları fazlaca içermektedir. Bu nedenle hem işaretleme maliyeti artmış hem de veriseti çeşitliliği düşmüştür. Farklı ülkelerden elde edilmiş verisetleri karşılaştırıldığında, Belçika gibi trafiğin ve yanıltıcı unsurun az olduğu, standartlaşmanın yüksek olduğu yerlerde toplanmış verisetleri ile daha yüksek başarı elde edilebildiği görülmektedir. Türkiye'de toplanan TRTSD veriseti ise en yüksek karmaşıklığa ve yanıltıcı unsura sahiptir. Bu nedenle verisetinin büyüklüğünü ve çeşitliliğini artırmak en önemli kriter olmuştur. Veri işaretleme maliyetlerini düşürmek adına; farklı verisetleri ile eğitilmiş sınırlayıcı kutu tespiti yapan modeller, veri işaretleme sırasında öneri sistemleri olarak kullanılabilir. Trafik işareti çeşitlerinin fazla olması da işaretleme maliyetini yükselten bir diğer unsur olmuştur. Yapay olarak üretilen ilgi alanları ile eğitilmiş bir sınıflandırıcı model; veri işaretleme sırasında öneri sistemlerinin bir parçası olarak kullanılıp, maliyetlerin azaltılmasını sağlayabilir.

KAYNAKLAR

[1] Z. Q. Zhao, P. Zheng, S. T. Xu ve X. Wu, «Object Detection With Deep Learning: A Review,» *IEEE Transactions on Neural Networks and Learning Systems,* pp. 1-21, 2019.

[2] Inland transport committee, «Convention on road signs and signals. Economic commission for Europe,» Vienna, 1968.

[3] R. Timofte, K. Zimmermann ve L. Van Gool, «Multi-view traffic sign detection, recognition, and 3D localisation,» *Machine Vision and Applications,* cilt 25, no. 3, pp. 633-647, 2014.

[4] M. Mathias, R. Timofte, R. Benenson ve L. Van Gool, «Traffic sign recognition—How far are we from the solution?,» *Neural Networks (IJCNN), The 2013 International Joint Conference on,* pp. 1-8, 2013.

[5] SAE International, *Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles,* 2018.

[6] Tesla, *Autopilot,* https://www.tesla.com/autopilot, 2019.

[7] L. Perez ve J. Wang, «The Effectiveness of Data Augmentation in Image Classification using Deep Learning,» 2017.

[8] S. Houben, J. Stallkamp, J. Salmen, M. Schlipsing ve C. Igel, «Detection of traffic signs in real-world images: The German traffic sign detection benchmark,» *Proceedings of the International Joint Conference on Neural Networks,* no. May 2014, 2013.

[9] F. Larsson ve M. Felsberg, «Using Fourier Descriptors and Spatial Models for Traffic Sign Recognition,» *SCIA konferens,* no. 9783319087795, pp. 238-249, 2011.

[10] A. Møgelmose, M. M. Trivedi ve T. B. Moeslund, «Vision-based traffic sign detection and analysis for intelligent driver assistance systems: Perspectives and survey,» *IEEE Transactions on Intelligent Transportation Systems,* cilt 13, no. 4, pp. 1484-1497, 2012.

[11] V. I. Shakhuro ve A. S. Konushin, «Russian traffic sign images dataset,» *Computer Optics,* cilt 40, no. 2, pp. 294-300, 2016.

[12] Z. Zhu, D. Liang, S. Zhang, X. Huang, B. Li ve S. Hu, «Traffic-Sign Detection and Classification in the Wild,» *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* pp. 2110-2118, 2016.

[13] P. Dewan, «An Overview of Traffic Signs Recognition Methods,» *International Journal of Computer Applications,* cilt 168, no. 11, pp. 7-11, 2017.

[14] M. Z. Abedin, P. Dhar ve K. Deb, «Traffic Sign Recognition Using SURF : Speeded Up Robust Feature Descriptor and Artificial Neural Network Classifier,» *19th International Conference on Computer and Information Technology, December 18-20, 2016, North South University, Dhaka, Bangladesh,* pp. 198-201, 2016.

[15] A. Adam ve C. Ioannidis, «Automatic road sign detecion and classification based on support vector machines and HOG descriptos,» *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences,* Cilt %1 / %2II-5, no. June, pp. 1-7, 2014.

[16] C. Souani, H. Faiedh ve K. Besbes, «Efficient algorithm for automatic road sign recognition and its hardware implementation,» *Journal of Real-Time Image Processing,* cilt 9, no. 1, pp. 79-93, 2014.

[17] J. R. R. Uijlings, K. E. A. Van De Sande, T. Gevers ve A. W. M. Smeulders, «Selective Search for Object Recognition,» 2012.

[18] R. Girshick, «Fast R-CNN,» *Proceedings of the IEEE International Conference on Computer Vision*, cilt 2015 Inter, pp. 1440-1448, 2015.

[19] N. Dalal ve B. Triggs, «Histograms of Oriented Gradients for Human Detection».

[20] C. Cortes ve V. Vapnik, «SUPPORT-VECTOR NETWORKS,» *Machine Learning*, cilt 20, no. 973, pp. 273-297, 1995.

[21] Y. Yang, H. Luo, H. Xu ve F. Wu, «Towards Real-Time Traffic Sign Detection and Classification,» *IEEE Transactions on Intelligent Transportation Systems*, pp. 1--10, 2015.

[22] Y. Yang, H. Luo, H. Xu ve F. Wu, «Traffic Sign Recognition Using a Multi-Task Convolutional Neural Network,» *IEEE Transactions on Intelligent Transportation Systems*, pp. 1--12, 2017.

[23] J. Redmon, S. Divvala, R. Girshick ve A. Farhadi, «You Only Look Once: Unified, Real-Time Object Detection,» 2015.

[24] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu ve A. C. Berg, «SSD: Single shot multibox detector,» *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, cilt 9905 LNCS, pp. 21-37, 2016.

[25] R. Girshick, J. Donahue, T. Darrell ve J. Malik, «Rich feature hierarchies for accurate object detection and semantic segmentation,» *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 580-587, 2014.

[26] S. Ren, K. He, R. Girshick ve J. Sun, «Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks,» *IEEE Transactions on Pattern Analysis and Machine Intelligence*, cilt 39, no. 6, pp. 1137-1149, 2017.

[27] J. Redmon ve A. Farhadi, «YOLO9000: Better, faster, stronger,» *Proceedings* -

30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Cilt %1 / %22017-Janua, pp. 6517-6525, 2017.

[28] J. Redmon ve A. Farhadi, «YOLOv3: An Incremental Improvement,» 2018.

[29] Microsoft, *VoTT - Visual Object Tagging Tool*, https://github.com/microsoft/VoTT, 2019.

[30] Unity Technologies, «Unity,» https://unity.com/, 2019.

[31] V. Rossum, G. Jr, D. Jr ve F. L, Python tutorial, Centrum voor Wiskunde en Informatica Amsterdam, The Netherlands, 1995.

[32] T. E. Oliphant, A guide to NumPy, Trelgol Publishing USA, 2006.

[33] G. Bradski, «The OpenCV Library,» *Dr. Dobb's Journal of Software Tools*, 2000.

[34] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, R. Jozefowicz, Y. Jia, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, M. Schuster, R. Monga, S. Moore, D. Murray, C. Olah, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu ve X. Zheng, *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*, 2015.

[35] F. a. o. Chollet, *Keras*, https://keras.io, 2015.

[36] A. B. Jung, *Imgaug*, 2018.

[37] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondei, A. Müller, J. Northman, G. Louppe, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot ve É. Duchesnay, «Scikit-learn: Machine learning in Python,» *Journal of machine learning research*, cilt 12, pp. 2825--2830, 2011.

[38] D. P. Kingma ve J. Ba, «Adam: A Method for Stochastic Optimization,» pp. 1-15, 2014.

[39] T. Tieleman ve G. Hinton, «Lecture 6.5---RmsProp: Divide the gradient by a running average of its recent magnitude,» *COURSERA: Neural Networks for Machine Learning*, 2012.

[40] J. Duchi, E. Hazan ve Y. Singer, «Adaptive Subgradient Methods for Online Learning and Stochastic Optimization,» *Jmlr*, cilt 12, pp. 1-40, 2011.

ÖZGEÇMİŞ

Adı Soyadı : Okan YILDIRAN

Doğum Yeri ve Tarihi : Diyarbakır – 1991

Yabancı Dili : İngilizce

E-Posta : okan.yldrn@gmail.com

Öğrenim Durumu

Derece	Bölüm/Program	Üniversite/Lise	Mezuniyet Yılı
Lise	Anadolu Lisesi	Diyarbakır Anadolu Lisesi	2009
Lisans	Bilgisayar Mühendisliği	Yıldız Teknik Üniversitesi	2014

İş Deneyimi

Yıl	Firm/Kurum	Görevi
2014	Huawei Turkey R&D Center	Assistant Research Engineer
2016	Gamegos Teknoloji A.Ş.	Game Developer