

**TC
BANDIRMA ONYEDİ EYLÜL ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

YÜKSEK LİSANS TEZİ

**DERİN ÖĞRENME İLE TRAFİK İŞARETLERİNİ TANIYAN
BİR SÜRÜCÜ DESTEK SİSTEMİ**

MOHAMED TAGHİ

Akıllı Ulaşım Sistemleri ve Teknolojileri Anabilim Dalı

Akıllı Ulaşım Sistemleri ve Teknolojileri Programı

AĞUSTOS 2022

TC
BANDIRMA ONYEDİ EYLÜL ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

YÜKSEK LİSANS TEZİ

DERİN ÖĞRENME İLE TRAFİK İŞARETLERİNİ TANIYAN
BİR SÜRÜCÜ DESTEK SİSTEMİ

MOHAMED TAGHİ

DANIŞMAN
Dr. Öğr. Üyesi İlyas ÖZER

Akıllı Ulaşım Sistemleri ve Teknolojileri Anabilim Dalı

Akıllı Ulaşım Sistemleri ve Teknolojileri Programı

AĞUSTOS 2022

ONAY

Mohamed Taghi tarafından “**Derin Öğrenme İle Trafik İşaretlerini Tanıyan Bir Sürücü Destek Sistemi**” adlı tez çalışması .../.../...tarihinde yapılan sınavla aşağıdaki jüri tarafından oybirliği/oyçokluğu ile Bandırma Onyedli Eylül Üniversitesi Fen Bilimleri Enstitüsü Akıllı Ulaşım Sistemleri ve Teknolojileri Anabilim Dalı’nda YÜKSEK LİSANS TEZİ olarak kabul edilmiştir.

Jüri Üyeleri

Dr. Öğr. Üyesi İlyas ÖZER

(Danışman)

Prof. Dr. Mehmet TEKTAŞ

(Üye)

Doç. Dr. Hasan ŞAHİN

(Üye)

İmza

Bu tezin kabulü, Fen Bilimleri Enstitüsü Yönetim Kurulu'nun .../.../...gün ve .../... sayılı seçim ile onaylanmıştır.

Doç. Dr. Abdullah YEŞİL

Enstitü Müdürü

BANDIRMA ONYEDİ EYLÜL ÜNİVERSİTESİ FEN BİLİMLERİ ENSTİTÜSÜ
YÜKSEK LİSANS TEZ ÇALIŞMASI ETİK BEYANI

Bandırma Onyedi Eylül Üniversitesi Fen Bilimleri Enstitüsü tez kılavuzuna göre hazırlanmış **“Derin Öğrenme ile Trafik İşaretlerini Tanıyan Bir Sürücü Destek Sistemi”** tezimin özgün bir şekilde, tezken tüm seçimlerde kullanımda tüm kullanıma yönelik etik eskiye uygun tanımında, tezde tüm uygulamalarda kullanıma uygun olacak şekilde, tezde kullanımda tüm nesnelere uygun olarak genel kullanıma yönelik başvuruda bulunacaktır./...../2022

Mohamed Taghi

ÖNSÖZ

Akıllı ulaşım alanında popüler bir araştırma yönü olarak, çeşitli bilim adamları trafik işareti tespiti ile geniş çapta ilgilenmişlerdir. Trafik işareti görüntülerinin öznetelik çıkarma şeması, optimum algılama yöntemleri seçimi ve algılama görevlerinin nesnel sınırlamaları gibi çeşitli zorluklara sahiptir. Bu zorlukların üstesinden gelmek amacıyla, bu makale, derin öğrenme yöntemlerini birleştirerek YOLO' ya dayalı hafif bir gerçek zamanlı trafik işareti algılama entegrasyon çerçevesi önermektedir. Çerçeve, ağırlı hesaplama yükünü azaltarak gecikme endişesini optimize eder ve çeşitli düzeylerde bilgi aktarımını ve paylaşımını kolaylaştırır. Algılama verimliliğini artırırken, belirli bir genelleme ve sağlamlık derecesi sağlar ve ölçek ve aydınlatma değişiklikleri gibi nesnel ortamlarda trafik işaretlerinin algılama performansını artırır. Önerilen model, gerçek yol sahnesi veri kümeleri üzerinde test edilmiş ve değerlendirilmiş ve etkinliğini doğrulamak için mevcut ana akım gelişmiş algılama modelleriyle karşılaştırılmıştır.

Çalışmanın hazırlanmasında bilgisi, donanımı ve tecrübeleriyle desteğini esirgemeyen tez danışmanım Dr. Öğr. Üyesi İlyas ÖZER'e, Anabilim dalı başkanı Prof. Dr. Mehmet Tektaş'a, hayatımda bana her zaman örnek olan annem Maryeme ELHASSEN ve babam Ahmed Mohamed TAGHI'ye teşekkür ederim.

ÖZET

YÜKSEK LİSANS TEZİ

DERİN ÖĞRENME İLE TRAFİK İŞARETLERİNİ TANIYAN BİR SÜRÜCÜ DESTEK SİSTEMİ

Mohamed Taghi

Bandırma Onyedli Eylül Üniversitesi

Fen Bilimleri Enstitüsü

Akıllı Ulaşım Sistemleri ve Teknolojileri Anabilim Dalı

Danışman: Dr. Öğr. Üyesi İlyas ÖZER

Ağustos 2022, 50 sayfa

Günümüzde toplumun ve ekonominin hızla gelişmesiyle birlikte otomobiller neredeyse her hane için uygun ulaşım modlarından biri haline gelmiştir. Bu, karayolu trafik ortamını giderek daha karmaşık hale getiriyor ve insanlar, sürücülere trafik işareti bilgileri sağlayan, sürücü işlemlerini düzenleyen veya yol güvenliğini sağlamak için araç kontrolüne yardımcı olan akıllı görüş destekli uygulamalara sahip olmayı önemli hale getiriyor. Daha önemli işlevlerden biri olan trafik işareti algılama ve tanıma, yurtiçinde ve yurtdışında araştırmacıların sıcak araştırma yönü haline geldi. Esas olarak, gerçek zamanlı yol görüntülerini yakalamak ve daha sonra yolda karşılaşılan trafik işaretlerini tespit etmek ve tanımlamak için araç kameralarının kullanılmasıdır, böylece sürüş sistemine doğru bilgi sağlanır. Ancak, gerçek sahnedeki yol koşulları çok karmaşıktır. Uzun yıllar süren yoğun çalışmalardan sonra, araştırmacılar tanıma sistemini henüz pratik hale getiremediler ve daha fazla araştırma ve iyileştirmeye hala ihtiyaç bulunmaktadır. Geleneksel olarak trafik işaretleri, standart bilgisayarlı görme yöntemleri kullanılarak algılanır ve sınıflandırılır, ancak görüntünün önemli özelliklerini manuel olarak işlemek de oldukça zaman alır. Bilim ve teknolojinin gelişmesi ve ilerlemesiyle birlikte, giderek daha fazla bilim insanı bu

sorunu çözmek için derin öğrenme teknolojisini kullanmaktadır. Derin öğrenme algoritmaları, sınıflandırma görevlerinde son teknoloji performanslar göstermiştir. Sonuç olarak, derin öğrenmeye dayalı nesne algılama algoritmaları, bilgisayarla görme görevlerinde popüler hale geldi. İki ana kategoriye ayrılabilirler: İki aşamalı algılama algoritmaları ve tek aşamalı algılama algoritmaları. İki aşamalı algılama algoritmaları, tek aşamalı algılama algoritmalarına kıyasla konum belirleme ve tanıma doğruluğu açısından daha iyi performansa sahiptir. Ancak, tek aşamalı algılama algoritmaları daha hızlı olacak şekilde tasarlanmıştır, bu da onları algılama süresinin çok önemli olduğu gerçek zamanlı uygulamalar için uygun hale getirir. Bu projede, esas olarak son teknoloji tek aşamalı algılama algoritmalarından bir trafik işareti algılama algoritması sunulmaktadır. Yol trafik işareti algılama ve tanıma uygulamasına dayanan, algılama ve tanımanın doğruluğuna ve yüksek verimliliğine odaklanmaktadır. Açık kaynaklı çerçeve olan YOLOv3 ve YOLOv7 kullanarak ile veri tabanı GTSDb (Alman Trafik İşareti Algılama Kıyaslaması) iki model oluşturuldu ve sırasıyla YOLOv7 97.3% ve YOLOv3 99.46% başarımları elde edilmiştir. Trafik işaretlerini sınıflandırabilecek bir model elde etmek ve bu trafik işaretlerinden en kritiklerini öğrenip tanımlamak için bir derin evrişim sinir ağı algoritması önerilmiştir.

Anahtar Kelimeler: YOLO, Trafik, Derin Öğrenme, GTSDb.

ABSTRACT

M.Sc. THESIS

A DRIVER ASSISTANCE SYSTEM THAT RECOGNIZES TRAFFIC SIGNS WITH DEEP LEARNING

Mohamed TAGHİ

Bandırma Onyedi Eylül University

Graduate School of Natural and Applied Sciences

Department of Intelligent Transportation Systems and Technologies

Supervisor: Prof. Dr. Üyesi İlyas Özer

August 2022, 50 pages

Today, with the rapid development of society and economy, automobiles have become one of the appropriate modes of transportation for almost every household. This is making the road traffic environment more and more complex, and people expect to have smart Vision powered apps that provide drivers with traffic sign information, regulate driver operations or assist vehicle control to ensure road safety. Traffic sign detection and recognition [1], one of the more important functions, has become the hot research direction of researchers at home and abroad. It is essentially the use of on-board cameras to capture real-time images of the road and then detect and identify traffic signs encountered on the road, thus providing accurate information to the driving system. However, road conditions in the real scene are very complex. After many years of hard work, researchers have yet to make the recognition system practical, and further research and refinement is still needed. Traditionally, traffic signs are detected and classified using standard computer vision methods, but manually processing important features of the image is also quite time consuming. With the development and advancement of science and technology, more and more scientists are using deep learning technology to solve this problem. Deep learning algorithms have shown cutting-edge performances in classification tasks. As a result, deep learning-based object

detection algorithms have become popular in computer vision tasks. They can be divided into two main categories: Two-stage detection algorithms and one-stage detection algorithms. Two-stage detection algorithms have better performance in terms of positioning and recognition accuracy compared to single-stage detection algorithms. However, single-stage detection algorithms are designed to be faster, making them suitable for real-time applications where detection time is crucial. In this project, a traffic sign detection algorithm, mainly from state-of-the-art single-stage detection algorithms, is presented. Based on the application of road traffic sign detection and recognition, it focuses on the accuracy and high efficiency of detection and recognition. We have built two models of the database GTSDb (German Traffic Sign Detection Benchmark) using the open-source framework YOLOv3 and YOLOv7, and the resulting results are YOLOv7 97.3% and YOLOv3 99.46%. A deep convolutional neural network algorithm is proposed to obtain a model that can classify traffic signs and train traffic sign trainers to learn and identify the most critical of these traffic signs.

Keywords: YOLO, Traffic, Deep Learning, GTSDb.

İçindekiler

ÖZET	I
ABSTRACT.....	III
ŞEKİL LİSTELERİ.....	VII
TABLO LİSTELERİ.....	VIII
1. 1 Motivasyon.....	3
1.2 Sorun Bildirimi	4
1. 3 Amaç	5
1. 4 Araştırma Kapsamı	5
2. LİTERATÜR İNCELEMESİ.....	6
3. BİLGİSAYARLA GÖRMENİN KÖKENİ	10
3.1 Nesne Algılama.....	11
3.1.1 İki Aşamalı Nesne Algılama Mimarisi.....	11
3.1.2 Tek Aşamalı Nesne Algılama Mimarisi.....	13
3.2 Nesne Tanıma	14
3.3 Derin Öğrenme.....	15
4 YAPAY SİNİR AĞLARI	16
4.1 Sinir Ağları.....	16
4.2 Biyolojik Sinir Ağı.....	16
4.3 Evrişimli Sinir Ağları (CNN).....	19
4.3.1 Evrişimsel Katman	20
4.3.2 Havuzlama	21
4.3.3 Tam Bağlantılı Ağlar	22
4.3.4 Seyreltme Katmanı.....	23
4.3.5 Aktivasyon Fonksiyonları	24
5 CNN Mimarisi.....	26
5.1 R-CNN	26
5.2 Fast R-CNN.....	26
5.3 Faster R-CNN	26
5.4 SPPnet	27

5.5 AlexNet	27
5.6 Mobil Net	28
5.7 YOLO.....	28
5.8 YOLOV7.....	31
6 TRAFİK İŞARETLERİ	34
6.1 Trafik ışıklarının doğuşu	34
6.2 Trafik kontrol sistemleri.....	35
7 METODOLOJİ.....	36
7.1 Sadece bir kez bakarsın (YOLO)	36
7.2 Veri Kümesi	36
7.3 Deney Tasarımı	37
7.4 Performans Değerlendirme Ölçütleri	38
8 BULGULAR VE KARŞILAŞTIRMALAR	39
9 SONUÇLAR.....	43
10 KAYNAKLAR	44

ŞEKİL LİSTELERİ

Şekil 1-1Trafik işaretlerin bazı örnekleri	2
Şekil 2-1 Trafik işaretleri tanımada zorlukları.	9
Şekil 3-1 Ortak bir iki aşamalı nesne algılama hattının gösterimi.	12
Şekil 3-2 Ortak bir tek aşamalı nesne algılama hattının gösterimi.	13
Şekil 4-1 Biyolojik nöronun bir taslağı[66].....	17
Şekil 4-2 Yapay nöron şeması.....	18
Şekil 4-3 Evrişimli katman işlemi örneği. Kırmızı katman bir özellik haritası içerir ve mavi katman beş özellik haritası içerir	21
Şekil 4-4 2x2'lik bir adım ve filter boyutunun.....	22
Şekil 4-5 Çok Katmanlı Derin Tam Bağlantılı Ağ, Görüntü Kaynağı	22
Şekil 4-6 seyreltme tekniği	23
Şekil 5-1 YOLOV3 Mimarisi.....	30
Şekil 5-2 YOLOv7 Diğer gerçek zamanlı nesne dedektörleriyle karşılaştırma.	32
Şekil 6-1 Londra'nın ilk trafik sinyali, 1868.....	35
Şekil 8-1 Önerilen model için harita sonuçlar YOLOv7	41

TABLO LİSTELERİ

<u>Tablo</u>	<u>sayfa</u>
Tablo 1-1 Trafic kazası istatistikleri, 2010-2021	4
Tablo 8-1 Modelleme Parametreleri.....	39
Tablo 8-2 Önerilen model için harita sonuçlar YOLOv3.....	40
Tablo 8-3 Önerilen model, mevcut kıyaslama çalışmaları ile karşılaştırılmıştır.....	42



SİMGE VE KISALTMA LİSTESİ

Simgeler	Açıklama
σ :	Sigmoid
Σ :	Toplam Formülü

Kısaltmalar	Açıklama
YSA :	Yapay Sinir Ağları
ReLu :	Rektifiyeli Lineer Ünite
ML :	Makine öğrenme
DL:	Derin Öğrenme
ANN:	Sinir Ağı
CNN:	Evrişimsel Sinir Ağı
LSTM :	Uzun Kısa Vadeli Bellek
ReLU :	Rektifiye Lineer Birimler
mAP :	Ortalama Ortalama Hassasiyet
YOLO :	Sadece Bir kez Bakıyorsun
SVM :	Destek Vektör Makineleri
R-CNN:	Bölge Tabanlı Evrişimli Sinir Ağları
Maske R-CNN:	Maske Bölge Tabanlı Evrişimli Sinir Ağı
SSD:	Tek atış Çoklu Kutu Dedektörü
GPU :	Grafik İşleme birimi
İTS :	İlaç Takip Sistemi
GAN:	Çekişmeli Üretici Ağlar
TSR:	Trafik İşareti Tanıma

CAD: Bilgisayar destekli tasarım

ADAS: Gelişmiş Sürücü Destek Sistemleri



1. GİRİŞ

Modern şehirlerdeki araç sayılarının hızla artması ve trafik kazaları hayatı giderek zorlu hale getirmektedir. Seyahat etkinliğini ve güvenliğini artırmaya yönelik teknolojik çözümler, akıllı sistemlerin ve araç ağlarının kullanılmasını içerir. Trafik işareti algılama ve tanımlama, araç ağlarının veya kendi kendini süren otomobillerin endüstriyel modeli için çok önemlidir. Derin öğrenme (DL), büyük gerçek zamanlı doğruluk gerektiren çok nesnel tanımlama uygulamaları için iyi çalışabilir [1] – [3]. Otonom bir araç, insanları, trafik işaretlerini ve trafik sinyallerini tanımlamak da dahil olmak üzere güvenli olması için çevresinin farkında olmalıdır. Trafik işaretleri algılama sistemleri, bir giriş görüntüsündeki (örneğin, Dur İşaretleri, U-Dönüş İşaretleri Yok, Hız Sınırı İşaretleri) her trafik işaretini bulmayı (bir sınırlayıcı kutu kullanarak) ve tanımayı amaçlar. Halka açık caddelerde seyahat etmek üzere tasarlanmış otonom araçlar için trafik işaretlerinin güvenilir bir şekilde tanınması çok önemlidir; aksi takdirde, bir kaza olasılığı önemli ölçüde artar. Sonuç olarak, bu konuyu ele almak için birçok eser yazılmıştır [4] – [6]. Yol işaretleri, kullanım amaçlarına bağlı olarak, hızlı tanıma için çeşitli simgeler, renkler ve biçimler kullanır. Bu model nedeniyle, erken trafik şarkı tanıma algoritmaları, el yapımı özellik seçimine dayanmaktaydı [7]. Bu özellikler öncelikle renkler ve formlar kullanılarak oluşturulmuştur. Ancak, bu yaklaşımın sağlamlığı sınırlıdır, çünkü el yapımı özellikler fazla sığma eğilimindedir. Sağlamlığı ve genelleştirmeyi geliştirmek için Süpürt vektör Machine (SVM), AdaBoost veya JointBoost gibi öğrenmeye dayalı bir teknik kullanılabilir. Son yıllarda, Derin Sinir Ağları (DNN), geleneksel yöntemleri geride bırakarak ilgi kazanmıştır [8], [9]. CNN'ler, trafik işareti tanımlama ve yaya dahil olmak üzere otonom sürüşte çeşitli zorlukları çözmüştür. Bunun nedenlerinden biri, Faster R-CNN ve YOLO algılama gibi son teknoloji derin dedektörlerin [10] minimum çaba ve gereksinimle çeşitli görevlere uygulanabilmesidir. Bununla birlikte, trafik işareti görüntü örnekleri içeren büyük kamu veri kümelerinin yayınlanmasıyla birlikte, son zamanlarda trafik işaretleri tanımlaması için uygulamaları mümkün hale geldi [11], [12]. DL ile ilgili kritik sorunlardan biri, geniş etiketli veri gereksinimidir. Bir uygulamanın kullanmak istediği hedef etki alanı için verileri elde etmek ve etiketlemek genellikle çok zaman ve para gerektirir. Bu durumda görüntülerin elde edilmesi ve manuel olarak kişilere göre sınıflandırılması uzun çalışma saatlerini almıştır. Bu, görüntüdeki her trafik işaretinin tanınan ve anlamı belirtilen bir sınırlayıcı kutuya sahip olması gerektiğinden, algılama görevi için çok önemlidir. Bazı çabalar, üretken çekişmeli ağ (GAN)

kullanarak gereken veri miktarını azaltır. Ancak bu genellikle bir miktar etiketleme gerektirir [13]. TSR (Trafik İşareti Tanıma) sistemleri, sürücüler, yolcular ve yayalar için daha fazla güvenlik sağlayan başka bir sürücü yardımı derecesi sağlar. Gelişmiş Sürücü Destek Sistemlerinin (ADAS) bir parçası olan TSR, sürücülerin dikkat dağınıklığı, yorgunluk, aydınlatma, hava koşulları ve kısmi tıkanmalar gibi senaryolarda trafik işaretleri konusunda uyararak kazalardan kaçınmasına yardımcı olabilir. Her birinin faydaları ve sakıncaları olan birkaç TSR tekniği önerilmiştir. Bununla birlikte, sağlam bir algoritma geliştirmek hala olağanüstü bir araştırma sorusudur. Gerçek dünyada trafik işaretinin maruz kalması nedeniyle, tanımlama çeşitli faktörler için sorunludur [14]. Fotoğraflar hareket halindeki bir araçtan çekildiği için titreşimlere, bulanık sahnelere ve dalgalanan ışıklara maruz kalmaktadır. Ayrıca bina veya otomobil gibi renk ve şekil bakımından benzer nesnelerin varlığı sistemi etkileyebilir ve işaret tanıma sürecini zorlaştırabilir. Sistem, sis, yağmur, bulut ve güneş ışığı gibi çeşitli hava ve aydınlatma koşullarında trafik işareti görünürlüğü ile başa çıkabilmelidir.



Şekil 1-1Trafik işaretlerin bazı örnekleri

1. 1 Motivasyon

Yol güvenliği için yardımcı bir tesis olarak trafik işaretleri, insanların günlük yaşamında hayati bir rol oynamaktadır. Yol katılımcılarının yol koşullarını zamanında anlamalarına yardımcı olmakla kalmaz, aynı zamanda yolun düzgünlüğü, sürücü davranış kuralları ve trafik kaza oranlarının azaltılmasında da önemli bir rol oynar. Son yıllarda insansız sürüş sistemi ve gelişmiş araç yardımcı sistemi üzerine araştırmalar tüm hızıyla devam etmektedir. Trafik işaretleri tanıma ve algılama, sistemin en önemli parçasıdır. Trafik işaretlerinin etkin bir şekilde tanımlanması, sürücülere daha iyi araç kullanmak için sürücüler tarafından değerlendirilebilecek gerçek zamanlı ve etkili trafik bilgileri sağlayabilir.

Trafik işaretlerinin tespiti ve tanınması, akıllı ulaşım sistemlerinde (ITS) önemli bir konudur. Trafik işaretlerinin etkin ve doğru bir şekilde tanınması, sürüş yardım sistemi için bir garanti sağlayabilir ve aynı zamanda tam olarak gerçekleşmesi için gerekli koşulları sağlayabilir. Trafik işaretleri yolun her iki tarafında yer alır ve rehberlik, yasak, uyarı veya talimat bilgilerini iletmek için metinler veya özel semboller kullanır. Araçların hareketini kontrol etmede ve trafik kurallarını iletmekte önemli bir rol oynarlar. Yolun her iki tarafına yerleştirilmiş trafik işaretleri genellikle çok önemli bilgiler içerir. Yol koşulları hakkında erken uyarı vermek için uyarı işaretleri, sapmalar, dik yokuşlar, vb . Gibi mevcut yol durumu bilgilerini yansıtır. Sollama yasağı, dönüş yasağı gibi yasak işaretleri ve hız sınırı işaretleri, sürücülerin davranışlarını düzenler ve trafiğin düzenli bir şekilde ilerlemesini sağlar. Bu nedenle trafik işaretlerinin anlamının zamanında ve doğru bir şekilde tanınması sürücüler için çok önemlidir. Şehirleşmenin gelişmesiyle birlikte çeşitli araç sayısı yıldan yıla artmış ve bunların yol açtığı bazı trafik sorunları giderek daha ciddi hale gelmiştir. Trafik sorunlarını çözmenin yollarından biri de akıllı ulaşım sistemlerini (AUS) geliştirmektir. Ayrıca, insansız sürüş teknolojisi de son yıllarda sıcak bir konu olmuştur ve sürüş ortamındaki trafik işaretlerini otomatik olarak algılamak ve tanımak için bilgisayarların kullanılması, bu alanlarda çeşitli işlevlerin gerçekleştirilmesinin temelini oluşturmaktadır. Özetle, trafik işaretlerinin tespiti ve tanınması üzerine yapılan çalışmalar büyük değere ve uygulama beklentilerine sahiptir. Trafik işaretleri için geleneksel tespit ve tanıma modellerinde, genellikle trafik işaretlerinin görüntüsünü algılar ve tanırlar. Bu durumda sadece mevcut görüntünün bilgisi kullanılır ve görüntü dizileri arasındaki ilişki dikkate alınmaz. Ayrıca, bu geleneksel modellerin trafik görüntü dizilerini algılamak ve tanımak için temel gereksinimleri karşılayabilmesine rağmen, trafik videosunda algılama ve tanıma doğruluğunu geliştirmek için hala biraz alan olduğunu görebiliriz.

1.2 Sorun Bildirimi

Türkiye'de her yıl trafik kazaları önemli bir ölüm, yaralanma ve maddi hasar nedenidir. 2021 yılında toplam 1 milyon 186 bin 353 trafik kazası meydana gelmiştir. Bu kazaların 998 bin 390'ı maddi hasarlı, 187 bin 963'ü ise ölümlü yaralanmalı trafik kazalarıdır. Ölümlü yaralanmalı trafik kazalarının %78,6'sı yıl içinde, %21,4'ü yerleşim yeri dışında meydana gelmiştir [15].

Tablo 0-1-1 Trafik kazası istatistikleri, 2010-2021

Yıl	Toplam Kaza	Ölümlü Kaza	Maddi Hasarlı	Can Kaybı Sayısı			Yaralı Sayısı
				Toplam	Kaza yerinde	Kaza sonrası ¹	
2010	1106201	116 804	989 397	4 045	4 045	-	211 496
2011	1 228 928	131 845	1 097 083	3 835	3 835	-	238 074
2012	1296634	153 552	1 143 082	3 750	3 750	-	268 079
2013	1 207 354	161 306	1 046 048	3 685	3 685	-	274 829
2014	1 199 010	168 512	1 030 498	3 524	3 524	-	285 059
2015	1 313 359	183 011	1 130 348	7 530	3 831	3 699	304 421
2016	1 182 491	185 128	997 363	7 300	3 493	3 807	303 812
2017	1 202 716	182 669	1 020 047	7 427	3 534	3 893	300 383
2018	1 229 364	186 532	1 042 832	6 675	3 368	3 307	307 071
2019	1 168 144	174 896	993 248	5 473	2 524	2 949	283 234
2020	983 808	150 275	833 533	4 866	2 197	2 669	226 266
2021	1 186 353	187 963	998 390	5 362	2 421	2 941	274 615

Ölümcul olayların tam olarak kaçının hız nedeniyle olduğunu ölçmek zordur. Hız ne kadar yüksek olursa, sürücülerin tepki süresi o kadar az olur ve kazadan kaçınmak o kadar zor olur. Bunu düzeltmenin iki yolu bulunmaktadır. İlk yaklaşım, yol kullanıcılarının sorumlulukları konusunda farkındalığı artırmak, ikinci yol ise arabaları daha güvenli ve hızlı hale getirmektir. Teorik farkındalık oturumları şiddetle teşvik edilirken, sürücüler muhtemelen yolda savaşıacak ve hatalar yapacaktır. Bu nedenle hükümetler ve araştırmacılar, maksimum hızın makul şerit hızına göre düzenlenebileceği güvenli sistemler geliştirmek zorundadır.

Otonom sürüş araçları, son birkaç yılda araştırma camiasında popüler hale gelmiştir. Otonom bir araba, çevresini insan müdahalesi olmadan algılayabilir ve yönlendirebilir. Otonom bir araç için temel unsurlardan biri, yol işaretlerini tanımlamayı ve algılamayı içeren 'vizyon 'dur. Doğal sahnelerde meydana gelen tıkanma, değişen aydınlatma koşulları, kamera açısı ve diğer değişkenler gibi engeller nedeniyle trafik işareti tespiti zorlu bir faaliyettir. Başa çıkmak için başka bir zorluk, tek bir görünümde yer alan birçok işarettir.

1. 3 Amaç

Amaç, trafik işaretlerini etkin bir şekilde algılayabilen ve sınıflandırabilen ve gerçek zamanlı uygulamalarda uygulanabilecek performansa sahip bir sistem geliştirmektir. YOLOv3 ve YOLOv7 modeline dayalı olarak geliştirilen mimariyi eğitmek için GTSDb (Alman Trafik İşareti Algılama Kıyaslaması) veri setleri kullanılmıştır.

1. 4 Araştırma Kapsamı

Bu tez kapsamında trafik işaretlerini otomatik olarak tanıyabilen derin öğrenme tabanlı bir yaklaşımın sunulması planlanmaktadır. Bu kapsamda önerilen model literatürde bulunan veri setleri üzerinde test edilerek sistemin performansı değerlendirilecektir. Bunlara ek olarak önerilen derin öğrenme modelinin gömülü sistemler üzerinde test edilmesi planlanmaktadır.

2. LİTERATÜR İNCELEMESİ

DL, son yıllarda bilgisayarla görme, görüntü işleme ve sinyal işleme dahil olmak üzere çeşitli alanlarda önemli başarılar elde etti. Nesne izleme gibi birçok standart bilgisayarla görme görevinin performansı, DL çerçeveleri kullanılarak iyileştirilir. Nesne tespiti için birçok önde gelen DL çerçevesi, iki ana çalışmadan oluşan yaya tespiti için de kullanılabilir: sınıflandırma ve yerelleştirme [16] – [18]. En yeni ve yaygın olarak kullanılan yaya algılama çerçeveleri için iki ana kategori vardır: bağlantılı ve bağlantısız [19]. Yayaları tanımlamak için çok sayıda önceden programlanmış çapa kullanan çerçeveler. Daha hızlı R-CNN [20] ve YOLO [11], geleneksel çapa tabanlı çerçevelerin örnekleridir, FCOS [21] ve CSP [22] ise yeni çapa içermeyen dedektörlerin örnekleridir. DL çerçeveleri gibi, yaya dedektörünün performansı da birçok etikete bağlıdır. Yapılandırılmamış girdilerle karşı karşıya kaldıklarında, iyi eğitilmiş modellerin genelleme performansı düşer. Son yıllarda dört kategoride sınıflandırılabilir bir kaç uygun seçenek sunulmuştur: Bir kaç araştırma çalışması, sahneye özel algılama için bir çapraz eğitim çerçevesi geliştirdi [23]. Diğerleri, dört adımlı özellik aktarımına dayalı aşamalar oluşturmuştur. Aşama algılama yönteminde [24]. Yukarıda bahsedilen çabalar, hedef alandaki algılama performansını önemli ölçüde artırabilse de sahneye özel bir dedektör oluşturmak her zaman karmaşık süreçler gerektirir, gerçek zamanlı ihtiyaçları ve pratik uygulamaları sınırlar. (2) Denetimsiz özellik uyarlanabilirliği: Bu sorunun üstesinden gelmek için, bir kaç araştırmacı çıktı ve girdi arasındaki boşluğu kapatmak için alan uyarlamasını kullandı ve veri kümesine özgü bir çerçeve yerine tüm çerçevenin genel özelliği öğrenmesine izin verdi [25]. Diğer bilgisayarlı görme görevlerinde kullanılan etki alanı uyarlama stratejilerinden etkilenen denetimsiz özellik çok yönlülüğü izleme ve tanımlamada, bir dizi çalışmada kaybı kullanılmıştır [26]. Diğer araştırmacılar [27], ilerici gizli modellere dayalı, kendi hızında bir alan uyarlama stratejisi önerdiler. Son olarak, etki alanı uyarlanabilirliği, kaynak ve hedef verilere erişme yeteneğidir. Kaynak ve hedef özellik bilgileriyle iş birlikli eğitim, olağanüstü performansa sahip bir transfer öğrenme mimarisi üretir. Bununla birlikte, patlama bölgesine milyonlarca veri ve açıklama getirmek, belirli uygulamalar için uygun değildir. Ayrıca, donanım ilişkili yaratıcı ve nesnel verileri kullanan maliyetli öğrenme yerine, iyi bir genelleme yeteneğine sahip çok iyi bir çerçeve, belirli iletişim sistemleri için tipik olarak daha uygundur. (3) Gürültülü etiketlerden öğrenme: İyi eğitilmiş modellerle sahte etiketler oluşturmak, açıklama içermeyen veri kümeleri için uygun bir seçenektir. Bazıları, otonom bir adaptasyon saptama çerçevesini eğitmek için yüksek güvenilirliğe sahip sahte etiketler kullanmayı

önerdi [28]. Ayrıca, sanal etiketlerin güvenilirliği öğrenme üzerinde önemli bir etkiye sahip olduğu için performans sınırlıdır. Sonuçlar öncelikle diğer etiketli kategorilerdeki algoritmalara bağlıdır. Ek olarak, sınıflandırma doğruluğu ve kilit noktaların yerleştirilmesi, sözde etiketlerin nesneleri tanımlamada ne kadar başarılı olduğunu belirleyen faktörlerdir. Küçük bir kutu kayması, kaynak etki alanındaki test performansı üzerinde hiçbir etkiye sahip değildir, ancak vahşi veri kümelerinde eğitim yönünü değiştirecektir. (4) GAN kullanarak veri artırma: Son yıllarda başlangıcından bu yana, birçok araştırma kaynak ve hedef arasındaki boşluğu kapatmak için GAN'ı kullandı. GAN yöntemlerini kullanarak yeni bir veri kümesi oluşturmak mantıklıdır [29]. Bir fotoğraf düzenlemesi olarak kabul edilebilir ve kaynak verileri ve ayrıca hedef sınıf dağılımları ile açıklamaları içerir. Başka bir strateji, yayaların sayısını artırmak ve bunları orijinal verilere rastgele eklemek, daha iyi [30]genelleme yeteneklerine sahip bir dedektörü eğitmek için ek özellikler sağlamaktır. Bununla birlikte, önceki çabalar, yürütülmesi zor olan ve teknik özellikleri tam olarak karşılamayan eğitim prosedürü boyunca kaynakları ve hedef verilerini gerektiriyordu. Çok sayıda önceki çalışmanın etiketlenmemiş veri vahşi veri kümelerinde denetimsiz algılama performansının nasıl geliştirileceğini tartışmış olmasına rağmen, hiçbir yanlışı etiketlenmiş veri kümelerinde iyi eğitilmiş süreçleri değiştirmek ve kaliteli izlenmeyen verimliliği elde etmeye yardımcı olmak için bir öğrenme sisteminin nasıl oluşturulacağını henüz tartışmamıştır.

Tespit için kullanılan CNN Modelleri

[31]de, tanıma işlemi derin görsel özelliklere dayandığından, algılama işlemi iki adımda gerçekleştirilir: 1) görüntüler toplanır ve işlenir, ardından görüntülerin ana kareleri belirlenir ve 2) evrişimle Sinir Ağı (CNN) görüntülerin derin görsel özelliklerini çıkarmak için uygulanır. CNN tarafından çıkarımın iki faydası vardır: 1) yerel algısal vizyon ve 2) paylaşılan ağırlıklar. Özniteliklerin çıkarılması, bir dizi evrişim katmanı tarafından yapılır. Öznitelikler daha sonra SVM sınıflandırıcısı gibi bir sınıflandırıcıyı tek boyutlu bir vektörde eğitmek için kullanılır. Bu yöntemin doğruluğu %71,42'dir. CNN, derin algısal özellik çıkarıcı olarak kullanılır [32], işlevi insan görsel korteksiyle karşılaştırılır. Daha seçici bir CNN öğrenilmiş özellik setinin önceki TSR modellerinden daha ekonomik ve kesin olduğu iddia edilmektedir. Bu modelin doğruluğu %99,54'tür.

Havuzlama katmanındaki maksimum konumları kullanmayı önerirler ve bu, modelin öğrenme yeteneklerini artırmalarına olanak tanır. Ayrıca sınıflandırma sürecinin hızla arttığını iddia edilmiştir. Bu modeldeki doğruluk %98,86'dır. CNN mimarisi, her sette maksimum havuz katmanları ile 3 set everişimse katmandan oluşur. Sonunda 2 tamamen bağlı katman kullanılır ve çıktıdan önce bir seyreltme katmanı yer almaktadır.

Sınıflandırma için kullanılan CNN modelleri

Görüntülerin sınıflandırılması için kullanılan CNN'lerin varyantları olmuştur [33]. [34], sınıflandırma için iki aşamalı bir everişimse sinir ağı mimarisi kullanılmıştır. Çıkarma için geleneksel öznitelik çıkarıcıları kullandılar. Doğruluk %97'dir ve CNN mimarisi, [34] ReLu ve havuzlama katmanlarını içeren 2 küme everişim katmanı içerir. CNN mimarisi, içinde [33] önerilen LeNet-5'tir [35].

[36]de de benzer bir yapı kullanılmıştır. Bununla birlikte, geleneksel özellik çıkarımından sonra, aşamalı olarak öğrenmek için CNN'yi kullanması bakımından farklıdır. Bu mimarideki ana değişiklik, 1. aşamadaki özelliklerin, 2. aşama öznitelikleriyle birlikte sınıflandırıcıya beslenmesi ve bu, diğer tüm mekanizmalardan daha doğru sonuçlar vermesidir. Burada elde edilen doğruluk, ağın derinliğini artırarak ve renk bilgisini göz ardı ederek %99,17 olmuştur.

Tespit ve Sınıflandırma için kullanılan CNN modelleri

Araştırmalar [37], hem tespit hem de sınıflandırma yapmak için bir CNN modeli kullandı. Bu, daha seçici bir özellik temsiline sahip olmalarını sağladı. Tespit, çok görevli bir CNN ile renk tabanlı bir bölge önerisi kullanılarak yapılır ve sınıflandırma da aynı anda yapılır. CNN mimarisi, havuzlama katmanları ve ardından 2 tam bağlantılı katman içeren 3 küme everişim katmanı içerir. Elde edilen doğruluk %98,83'tür.

Bu yaklaşım [38]diğerlerinden farklıdır çünkü iki CNN kullanarak metin tabanlı trafik işareti algılama kullanır. İlk CNN, işaretleri ve ilgilenilen bölgeleri algılar ve ikinci CNN metni algılar. İkinci CNN, [21]'de önerilen modele dayanmaktadır. İlk CNN modeli, 5 aşamalı everişimler içerir ve bunların sonunda ROI'yi (ilgi bölgeleri) önermek için basamaklanır. Bu yöntemin kesinliği %90'dır ve sınıflandırma için geçen süre 0.15s'dir.

Hızlı Bölgesel CNN

[39]'de araştırmacılar, derin CNN'ye TSRnin test sürecini hızlandıran bir dal-çıkı mekanizması eklemiştir. Biyolojik süreçten ilham alan bu mekanizma, işaretlere ve özelliklerine ilişkin geçmiş bilgileri sunarak hızı ve doğruluğu artırır. Bu nedenle, bir işaretin küçük bir bölümünün veya sadece şeklinin tespiti onu sınıflandırmak için yeterli olabilir, oysa diğer mekanizmalar onu sınıflandırmadan önce tüm görüntüyü işlemelidir. Fast Branch geliştirilmeden önce çoğu mekanizma, sistemin tüm görüntüyü görmesi gerektiği varsayımı altında çalışıyordu ve bu bir dezavantaj olarak ortaya çıktı. Bu modelde kullanılan CNN mimarisi, bir havuzlama katmanı ve bir normalleştirme katmanının izlediği evrişiklerden oluşan 3 küme everişim katmanı içerir. Dallanma, everişimden sonra ve havuzlama katmanından önce gerçekleşir. Teoride bu katmanların n sayıda kümesi olabilir. Doğruluk %98.52'dir. Şekil 2 1 En üst sıradaki görüntülere bakan bir insan bunları nispeten kolaylıkla tanımlayabilirken, geleneksel bir yaklaşım bunu yapamayacaktır [22].



Şekil 2-1 Trafik işaretleri tanımada zorlukları.

3. BİLGİSAYARLA GÖRMENİN KÖKENİ

Görüntüden otomatik olarak bilgi çıkarmaya yönelik ilk girişim, doktora tezinde [40] bir resimden bir küpün 3B yapısını çıkarmayı öneren Laurence Roberts tarafından 1963'e dayanmaktadır. O zamanlar yapay zekanın öncüleri [41], görme problemini [41] çözmenin planlama ve üst düzey akıl yürütme gibi daha zor görevlere doğru kolay bir adım olacağını düşündüler [42]. Basit örtüşmeyen nesnelerle sahne analizine segmentasyona dayanır. Ancak, bu görevleri çözmek on yıllar aldı. Bu arada, araştırmacılar çizgi etiketleme gibi düşük seviyeli işleme görevlerine odaklanır ve ardından [43] görüntülerin görsel içeriğini basitleştirme hedefiyle kenar algılamaya doğru ilerler. Hubel ve diğerlerinin görsel sistem üzerindeki keşiflerinden esinlenmiştir. David Marr, vizyon boru hattı için aşağıdan yukarıya bir yaklaşım önermektedir. İlk aşama, sahnenin ilk taslağına yol açan 2B görüntülere uygulanan düşük seviyeli görüntü işlemenin kullanılmasına karşılık gelir. Daha sonra binoküler stereo kullanılarak 3 boyutlu bir çizim elde edilir. Son olarak, nihai temsil, yapısal analiz kullanılarak 3 boyutlu bir haritada sahnenin bir 3B modelidir. Araştırmacılar [43], görüntü özelliklerine dayalı karşılık gelen bir 3D Bilgisayar Destekli Tasarım (CAD) modelini takarak ya [44]da görünür şeklin bakış açısına göre değişmeyen geometrik özelliklerini kullanarak, görüntülerdeki nesneleri tanımak için bu paradigmadan ilham alırlar [45]. Bununla birlikte, geometri tabanlı yöntemler, aydınlatmalar nedeniyle zorluklar göstermektedir Burns ve ark. (1993) ve yaklaşımlar görünüm özelliklerinin kullanımına doğru ilerlemektedir. Sirovich ve ark. (1987), bir görme görevini çözmek için ilk istatistiksel yaklaşım olan PCA ile yüz görüntülerini düşük boyutlu bir uzaya ayırtırmayı önermiştir. Bu yaklaşım, PCA uzayında [46] bir Öklid mesafesini kullanarak yüzleri tanıyabilen gerçek zamanlıya yakın bir sistem olan 1991'de Öz yüzlerin ayrıştırılmasının tanıtılmasıyla geliştirilmiştir. Görünüm özelliklerinin kullanımı daha sonra bir sonraki bölümde açıklandığı gibi daha fazla göreve başarıyla uygulanır. Bilgisayarlı görmen kökeninin bu kısa özeti, şu anda sahada kullanılan öğrenme paradigmasına doğru giden genel eğilimi göstermektedir.

3.1 Nesne Algılama

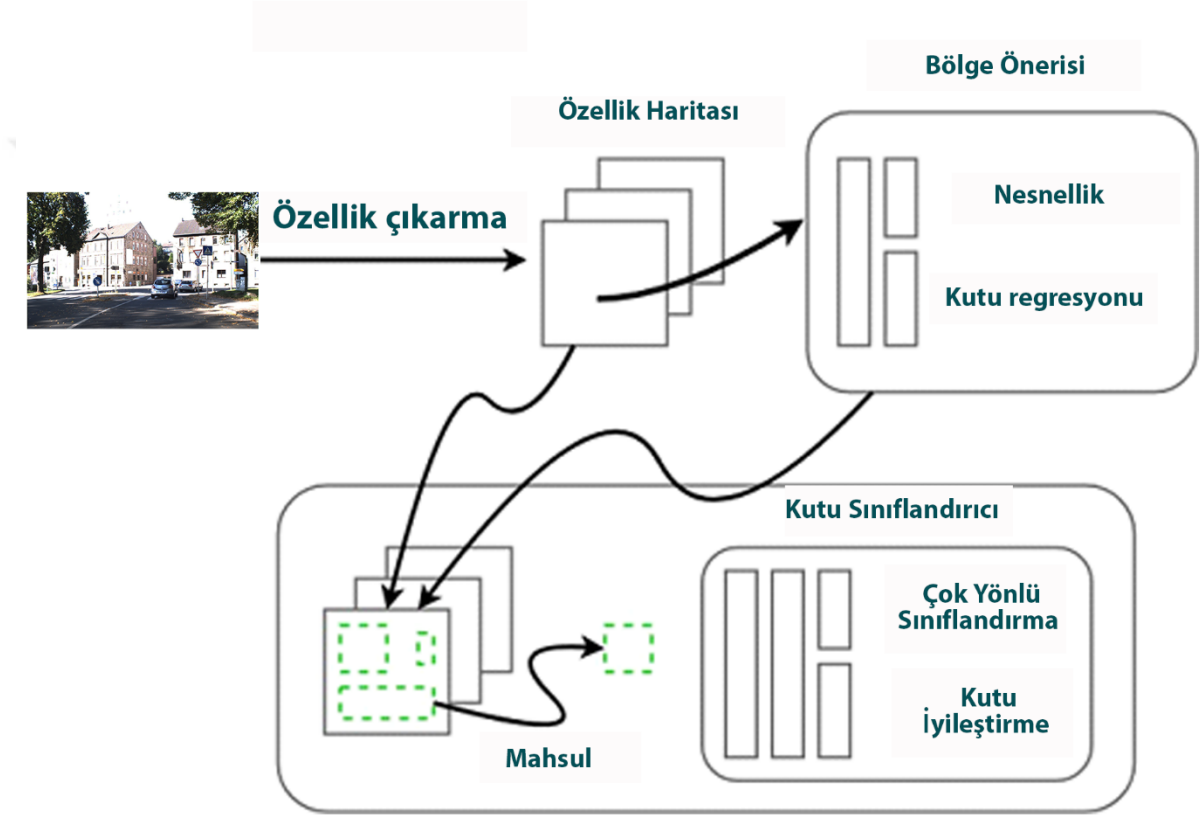
Görüntü sınıflandırma problemleri, görüntüde tek bir nesne veya sahne olduğunu ve uzamsal bir kapsamın olmadığını varsayar. Nesne algılamanın amacı ise, her bir örneği çevreleyen sınırlayıcı kutular kullanarak görüntü içindeki nesnelerin düzgün bir şekilde tanımlanmasını ve yerleştirilmesini gerektirir. Bir görüntü, aynı veya farklı sınıflardan birden çok nesne örneği içerebilir. Modern nesne dedektörleri, tek aşamalı ve iki aşamalı olmak üzere iki tür mimariye sahiptir. İki aşamalı nesne dedektörleri[47], [48] nesnelerin seyrek bir aday konumları kümesi oluşturmak ve konumlarını ikinci aşamasında sınıflandırmak için bir bölge önerme aşaması kullanır, öte yandan aşağıda verilenler gibi tek aşamalı nesne dedektörleri [49], [50] [51] nesnelerin aday konumlarını belirlemez. Genel olarak iki aşamalı dedektörler, düşük hızlarda daha yüksek doğruluk oranlarına sahiptir ve tek aşamalı dedektörler, yüksek hızlarda daha düşük doğruluk oranlarına sahiptir.

3.1.1 İki Aşamalı Nesne Algılama Mimarisi

İki aşamalı veya teklif tabanlı nesne dedektörleri, ilk aşamada bir dizi aday nesne konumu veya ilgi alanı (ROI) oluşturur ve mimarinin ikinci aşamasında RoI'lerde ilgi duyulan nesneleri sınıflandırır ve lokalize eder. Modern iki aşamalı dedektörlerin tümü R-CNN ailesi üzerine inşa edilmiştir[48]. Girshick *et al.* 2014 yılında iki aşamalı nesne dedektörlerinin yapı taşı haline gelen R-CNN'yi tanıttı. Bir everişimse sinir ağına (CNN) beslenmek üzere sabit bir boyuta yeniden boyutlandırılan aday bölge önerilerini elde etmek için seçici arama gibi bir öneri mekanizması kullanır [53]. CNN çıktıları, öngörülen sınıf puanlarını ve sınırlayıcı kutu iyileştirmelerini verir. Puanlara dayalı olarak, daha yüksek puan alan bölge önerileri seçilir ve kesinlik etiketleriyle karşılaştırılır. Karşılaştırma IOU (birlik üzerinden kesişme) metriği ile, temel doğrulukla belirli bir miktarda örtüşen yüksek puanlara sahip bölge önerilerini seçmek için yapılır. Belirli miktara eşik denir ve genellikle 0,50'den büyüktür. Düşük IoU'lu bölge önerilerinin ortadan kaldırılmasından sonra, bir ilgi nesnesinin en doğru tahminle tespit edildiğinden emin olmak için gereksiz tahminleri ortadan kaldıran bir teknik kullanılır. Fazlalık algılamaların ortadan kaldırılması, maksimum olmayan bastırma (NMS) ile yapılır.

R-CNN ile ilgili sorun, belleğin çok geniş olmasıydı. Her bölge önerisini CNN aracılığıyla çalıştırma sorununu azaltmak için Fast R-CNN tanıtıldı [47]. Bireysel çalıştırmalar yerine, tüm

görüntünün özellik haritasını elde etmek için tüm görüntü bir CNN'den geçirilir. Bu şekilde, ana hesaplama omurga ağında gerçekleşir. Daha sonra, RoI kullanarak ilgilenilen bölgeyi (RoI'ler) elde etmek için özellik haritasında bir bölge önerisi yöntemi kullanılır. RoI havuzu, daha sonra sınıflandırma ve sınırlayıcı kutu iyileştirme için CNN'lere beslenen kırpılmış ve yeniden boyutlandırılmış özelliklerle sonuçlanır.



Şekil 3-1 Ortak bir iki aşamalı nesne algılama hattının gösterimi.

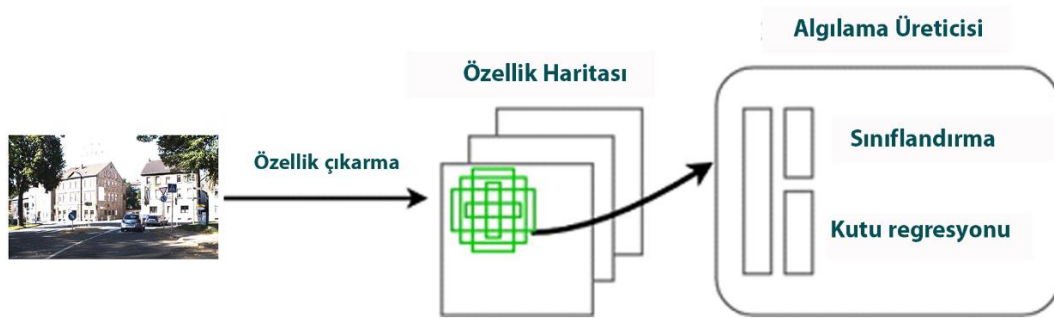
Bir özellik çıkarıcı, bölge öneri ağı ve bir kutu sınıflandırma ve regresyon bölümünden oluşur. Şekil 3-1’de ortak bir iki aşamalı nesne algılama hattının gösterimi yer almaktadır. Bir özellik çıkarıcı, bölge öneri ağı ve bir kutu sınıflandırma ve regresyon bölümünden oluşur.

R-CNN ailesinin son üyesi, [47]Ren ve ark. bölge tekliflerini tahmin etmek için özellik çıkarmadan (omurga) sonra bir bölge teklif ağı (RPN) eklemiştir. Öznitelik haritasını elde etmek için

görüntü, öznetelik çıkarımı için omurga ağından geçirilir. Görüntü düzeyindeki özellikler daha sonra bölge önerilerini almak için RPN'ye geçirilir. RPN, bir bağlantının ilgilenilen bir nesneye (nesnellik) sahip olma olasılığını belirlemek için, aynı zamanda bağlantı kutuları olarak da adlandırılan, farklı ölçek ve en boy oranlarına sahip görüntü üzerine yerleştirilmiş önceden tanımlanmış sınırlayıcı kutular kullanır. RPN, tekliflerin koordinatlarını hassaslaştırmak için çapalarda kutu regresyonunu da kullanır. Daha sonra, görüntü özellik haritasını ve RPN çıktılarını kullanarak, RoI havuzu, son sınıflandırma ve sınırlayıcı kutu regresyonu için kullanılacak bölge önerilerini kırpar. RPN kısmı hariç, Daha Hızlı R-CNN ve Hızlı R-CNN mimarileri aynıdır. Daha Hızlı R-CNN'ye dayanan klasik bir iki aşamalı nesne algılama ardışık düzeni Şekil 3.1'de gösterilmektedir.

3.1.2 Tek Aşamalı Nesne Algılama Mimarisi

Aşırı yetenek ilk tek aşamalı nesne dedektörlerinden biriydi [54]. Sermanet *et al.* [54] bir bölge önerme mekanizması kullanan iki aşamalı nesne dedektörlerinden daha iyi performans göstermek için geleneksel bir kayan pencere yaklaşımı kullandı. Daha sonra [54] tek kademeli nesne dedektörlerinin öncüleri [50] ve [49] popüler hale geldi. Bu mimariler, bu görevleri bölge öneri konumlarından gerçekleştirmek yerine nesneleri sınıflandırmak ve algılamak için görüntü pikselleri üzerinde global bir sınıflandırma ve regresyon gerçekleştirir. Şekil 3-2 Ortak bir tek aşamalı nesne algılama hattının gösterimi. Bir özellik çıkarıcı ve bir kutu sınıflandırma ve regresyon bölümünden oluşur



Şekil 3-2 Ortak bir tek aşamalı nesne algılama hattının gösterimi.

Tek aşamalı nesne dedektörlerinin temel parçalarından biri, farklı konumlarda ve en boy oranlarında görüntü üzerine yerleştirilmiş çeşitli ölçeklere sahip önceden tanımlanmış kutular koleksiyonuna sahip olmaktır. Bunlara çapalar veya öncelikler denir ve olası nesnelerin ideal şekil ve boyutlarına sahiptirler. Farklı uzamsal konumlar, görüntüdeki bir nesne ile bir çapanın örtüşmesini mümkün kılar. Farklı şekillere sahip olarak, görüntüdeki farklı şekilli nesnelerle bir çapanın örtüşmesi mümkündür. Farklı boyutlara sahip olarak, görüntü içindeki küçük, orta veya büyük boyutlu nesnelerle bir bağlantı çakışması elde etmek mümkündür. Nesneleri içermesi muhtemel bölge önerileri yerine, tek aşamalı nesne dedektörleri, bir görüntü boyunca sırayla düzenlenmiş farklı şekil ve boyutlarda geniş bir aday nesne konumları kümesine sahiptir.

Tek aşamalı dedektörlerin yüksek verimliliğe ancak daha az doğruluğa sahip olması ankrajlar nedeniyledir. Nesne algılama problemi bir görüntü sınıflandırma problemi haline geldiğinden modellerin doğruluğunu artırmak mümkündür. Bununla birlikte, yalnızca küçük bir miktar ankraj, kesin bilgi kutuları ile örtüşür, yani aşırı negatif numuneler ve nispeten daha az sayıda pozitif numune vardır. Bu, ön plan ve arka plan sınıfı dengesizlik sorunlarına neden olur ve modellerin doğruluğunu azaltır. Ortak bir tek aşamalı nesne algılama hattının bir gösterimi Şekil 3.2'de verilmiştir.

3.2 Nesne tanıma

İnsanlar, kamera görüş noktası, nesnenin boyutu, ölçeği ve dönüşü ne olursa olsun, görüntülerdeki nesneleri çok az çabayla tanır. Hatta görüntüde kısmen kapalı olan bir nesnenin kategorisini bile çıkarabilirler. Nesne tanıma sistemlerinin amacı, bir görüntüde bulunan nesneyi kategorize etmekten oluşan bu davranışı taklit etmektir. Bunu yapmak için, nesne tanıma sistemlerinin her nesne için değişmez, sağlam ve ayırt edici temsiller oluşturması gerekir. Nesne tanıma, bir giriş görüntüsüne önceden tanımlanmış bir etiket atamaya karşılık gelen bir sınıflandırma görevidir [55]. Bu, özgeçmişte on yıllardır görüntü tanımlayıcıların kalitesini değerlendirmek için kullanılan standart bir görevdir. Tarihsel olarak bilgisayarlı görü sistemleri, görüntü temsillerini çıkarmak için iki tür tanımlayıcı kullanmıştır. Görüntülerdeki ilgi noktaları etrafında düşük seviyeli özellikleri çıkarmayı amaçlayan yerel tanımlayıcılar ve yerel özellikler kümesinden tüm görüntünün sıkıştırılmış bir temsilini kodlayan global tanımlayıcılar. Bu iki tanımlayıcı, piksellerden katı matematiksel modellemeye dayalı bir vektör temsiline kadar elle hazırlanmış bir görüntü hattından oluşan el yapımı yöntemlerle kullanılır. Son zamanlarda, alan, hiyerarşik bir yapı kullanarak aynı zamanda düşük

seviyeden yüksek seviyeye özniteliğin öğrenilmesi olan CNN tabanlı uçtan uca öğrenme sistemlerine doğru ilerlemiştir.

3.3 Derin Öğrenme

Derin öğrenme, yapay zekanın alt alanıdır. Kendi başlarına veriye dayalı kararlar alabilen sinir ağı modelleri oluşturmakla ilgilenir [56]. Derin öğrenme, sistemlerin deneyimlerden öğrenmesini ve dünyayı kavramlar hiyerarşisi açısından anlamasını sağlar [57]. Bilgisayar bilgi toplama konusunda kendi kendine yeterli olduğundan, bilgisayar için gereken bilgiyi belirlemek için herhangi bir insan müdahalesi yoktur. Özellik çıkarma, derin öğrenmeyi özel kılan özelliklerden biridir. Özellik çıkarımı, modele sağlanan girdi verilerinden özelliklerin çıkarıldığı bir kavramdır. Bu şekilde, herhangi bir açık kodlama olmadan görüntü tanıma sağlanabilir. Her görüntü piksellerden oluşur. Bu pikseller komşu piksellerle ilişkilidir. Derin öğrenme teknikleri, daha büyük bir görüntünün küçük bölümlerinden özellikler çıkararak bu fenomenden yararlanır. Bu tür yöntemler üst düzey öznitelikleri öğrenme yeteneğine sahiptir, böylece her yeni problem için bir öznitelik çıkarıcı geliştirme görevi ortadan kalkar. Derin öğrenme algoritması, makine öğrenmesi algoritmalarından daha fazla parametreye sahip olduğundan, bir derin öğrenme modelini eğitmek genellikle daha fazla zaman alır.

4 YAPAY SİNİR AĞLARI

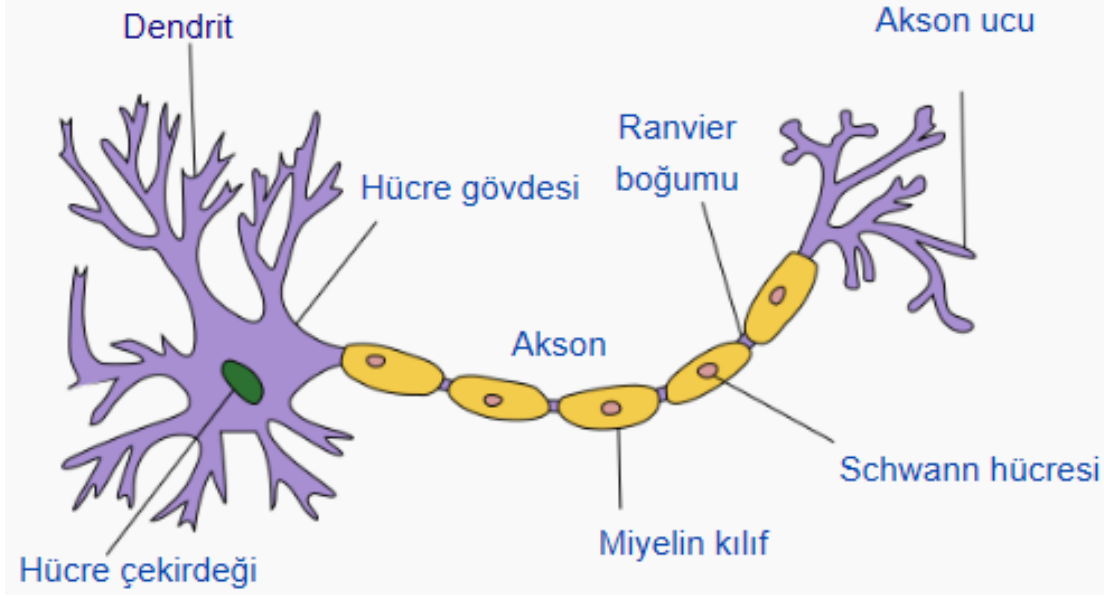
4.1 Sinir Ağları

1943'te Warren McCulloch ve Walter Pitts, insan beynini taklit etmenin bir yolu olarak sinir ağlarını öneren bir makale yayınladılar [58]. 1951'de Minsky ve Dean Edmunds, tarihteki ilk sinir ağı olarak kabul edilen stokastik sinirsel analog güçlendirme hesaplayıcısını geliştirdiler. [59]1955'te Allen Newell ve Herbert Simon, tarihte ilk kez yapay zeka programları geliştirdiler. Logic [60]Theorist adlı program, Whitehead ve Russell'in birlikte yazdığı bir çalışma olan Principe Mathematica'nın ilk 52 aksiyomundan 38'ini kanıtladı. Güçlü yapay zekanın (AI) uygulanmasına yönelik beklentiler belirsiz olsa da, yapay sinir ağı inşası yoluyla zayıf AI'nın evrimi devam etti. 1959'da Arthur Samuel, "makine öğrenimi" terimini tanıtarak zayıf yapay zekanın gelişimini hızlandırdı. [61]1965'te, güçlü yapay zekanın temel sorunlarına işaret etmek için art arda anlamlı araştırmalar yapıldı. Hubert Dreyfus kitabında, insanın zihninde bilgisayarların ulaşamayacağı şekilde çalışan bir alan olduğunu vurguladı. Joseph Weizenbaum [62], Turing'in taklit oyununun yanılmasını ortaya çıkaran ELIZA adlı bir AI programı geliştirdi. ELIZA [63], insanların makinelerle İngilizce iletişim kurmasını sağlayan bir yapay zeka programıydı. Programın çalışmasıyla ilgili deneyler yoluyla, makinenin, iletişimdeki kişi hakkında herhangi bir özel bilinç veya derin anlayış olmaksızın insanlarla yüzeysel bir düzeyde iletişim kurmak için kullanılabileceği gösterildi. Bir taklit oyununun kurulabilmesinin nedeninin, tüm insan zihninin bilgisayarla aynı şekilde gerçekleştirilmesi değil, oyuna katılanların zihinlerinin diyalog halinde makine ile duygusal olarak özümsemesi olduğu öne sürüldü [64].

4.2 Biyolojik Sinir Ağı

Yapay sinir ağı, insan beyni gibi biyolojik nöronların mimarisinden esinlenmiştir. İnsan beyni, birbirine bağlı çok sayıda nörondan oluşur. Her nöron, bir giriş sinyaline yanıt vermek gibi basit bir görevi yerine getiren bir hücredir. Bununla birlikte, bir nöron ağı birbirine bağlandığında, konuşma ve görüntü tanıma gibi karmaşık görevleri inanılmaz bir hız ve doğrulukla gerçekleştirebilirler. Genellikle, bir insanın bir yüzü tanıma gibi bir görevi tamamlaması birkaç yüz milisaniye alırken, beynin bireysel nöronunun hesaplama hızı birkaç milisaniyedir. Bu, bir bilgisayarın benzer bir görevi

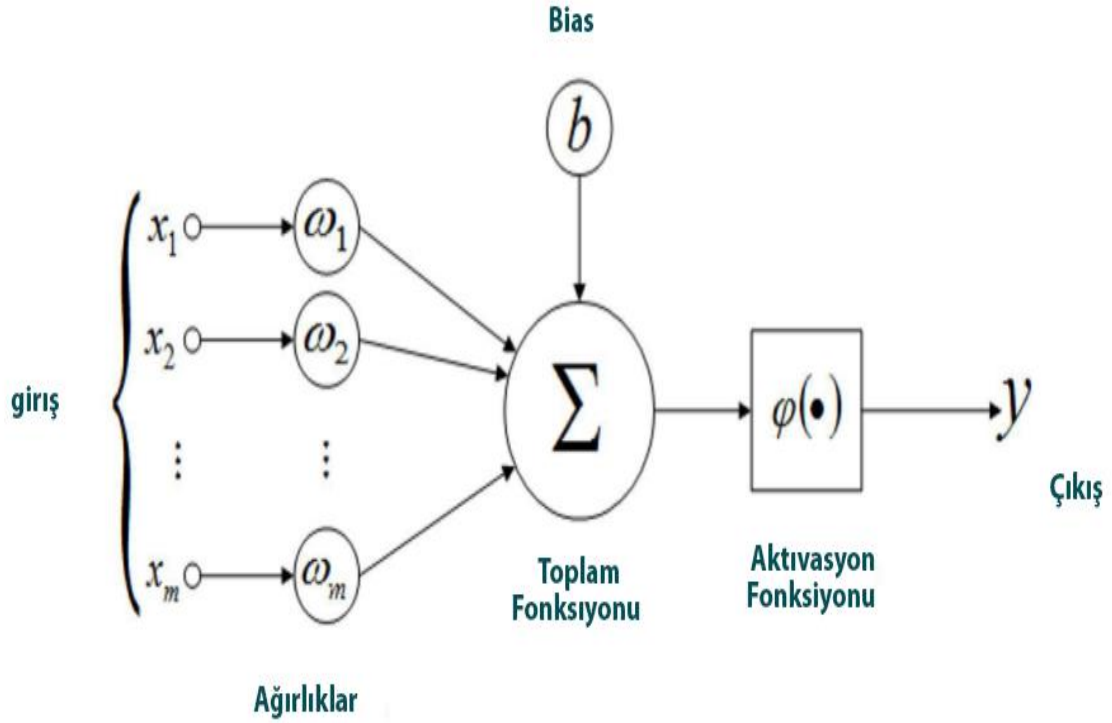
yerine getirmesi için gereken milyonlarca adıma kıyasla, beynin böyle bir görevi gerçekleştirmek için yalnızca yüz hesaplama adımı attığını gösterir [65]. Bu kadar kısa işlem süresi, nöronlar arasında iletilen bilginin çok küçük olması gerektiği anlamına gelir. Geleneksel bilgisayardan farklı olarak, tüm bilgi nörondan nörona geçmez, nöron ağının karmaşık ara bağlantılarında kodlanır. Bu nedenle sinir ağına bağlantılılık da denir. Bir nöron, Şekil 4-1'de gösterildiği gibi dinenilerden, bir hücre gövdesinden ve bir aksondan oluşur. Soma olarak da adlandırılan hücre gövdesi, tıpkı diğer hücreler gibi çekirdeği içerir. Dendiler, hücre gövdesine bağlanan ve diğer nöronlardan sinyal almak için uzayda uzanan dallardır. Akson, nöronun vericisidir. Komşu nöronlara sinyaller gönderir. Bir nöronun aksonunun ucu ile komşu nöronun dendritleri arasındaki bağlantı, iki nöron arasındaki iletişim birimi olan sinaps olarak adlandırılır. Elektrokimyasal sinyaller sinaps boyunca iletilir. Bir nöronun aldığı toplam sinyal sinaps eşliğinden büyük olduğunda nöronun ateşlenmesine yani elektrokimyasal bir sinyal göndermesine neden olur.



Şekil 4-1 Biyolojik nöronun bir taslağı[66]

komşu nöronlara sinyal gönderir. Sinaptik bağlantının gücünün değişmesinin hafızanın temeli olduğu varsayılır.[67]

Bu kavram, bu alandaki en önemli kavramlardan biridir ve bu bölümde bazı temel kavramlardan bahsedilecektir. Sinir ağlarından bahsedebilmek için öncelikle yapay zekada nöronun ne olduğunun anlaşılması gerekir. Bir nöron, yapay bir ağdaki bilgi sürecinin temel bir birliğidir ve biyolojik bir nöronun işleyişine dayanır. Yapay nöronlar ve biyolojik muadilleri, giriş bağlantılarına, çıkış bağlantılarına ve bir giriş sinyaline yanıt olarak bir çıkış sinyali üreten bir dahili işlemciye sahiptir.



Şekil 4-2 Yapay nöron şeması

$$u_k = \sum_{j=1}^n w_{kj} \cdot x_j + b_k \quad (1)$$

Yapay nöronlar şunlardan oluşur:

- Giriş sinyalleri ($x_1, x_2, \dots, x_m$) veya diğer nöronların çıktılarından gelebilecek giriş bilgileri
- Bir dizi ağırlık ($w_{k1}, w_{k2}, \dots, w_{km}$), hangi kuvvetler arasındaki bağlantıyı açıklar. Bu, birliktelikleri temsil eden pozitif veya nöronun aktivasyonunu devre dışı bırakan negatif olabilir. Nöronlar arasında bağlantı olmadığında sinaptik ağırlık sıfırdır.
- Önyargı (b_k), işlevi, boş bir girdi alındığında hataların ortaya çıkmasını önlemek olan, nörona harici bir parametredir.
- Toplam (Σ) veya giriş kendi ağırlıklarıyla çarpımını temsil eden giriş işlevi.
- (0,1) veya [-1,1] arasında normalleştirilmiş bir aralıkta nöronun genliğini veya önemini sınırlayan aktivasyon fonksiyonu ($\phi(\cdot)$)
- Nöron tarafından üretilen sonuç olan çıkış sinyali (y_k). Nöronların giriş sinyalleri, sinaptik ağırlık olarak da adlandırılan ilişkili ağırlıklarını çarpar ve yanlılığa eklenen bu sonuçların toplamı, nörona giriş bilgisini verir

4.3 Evrişimli Sinir Ağları (CNN)

Bir evrişimsel sinir ağı (CNN) [68], çıktıları öznitelik olarak çıkarılan ve düzleştirilen ve bir dizi tam bağlantılı katmana beslenen ilişkili alt örnekleme adımına sahip bir veya daha fazla evrişim katmanından oluşur. CNN'nin varyantları farklı yapılara sahiptir, ancak temel değişmeden kalır.

Evrişim ağları (CNN veya ConvNet olarak da bilinir), verileri bilinen bir ızgara benzeri topoloji ile işleyebilen sinir ağları ailesine aittir. Görüntüler söz konusu olduğunda, bir 2B piksel ızgarası vardır. Bir CNN bir girdi görüntüsü alır ve görüntüdeki çeşitli yönlere/nesnelere ağırlıklar ve önyargılar ekleyerek ve bir görüntüyü diğerinden ayırt ederek önem katar. CNN, katmanlarında matris çarpımı yerine evrişim adı verilen matematiksel bir işlem kullanır. CNN, bir giriş katmanından ve aralarında çoklu evrişimli, havuzlamalı veya tam bağlantılı katmanlara sahip bir çıkış katmanından oluşur [27]. Evrişim işlemi matematiksel olarak şu şekilde ifade edilebilir:

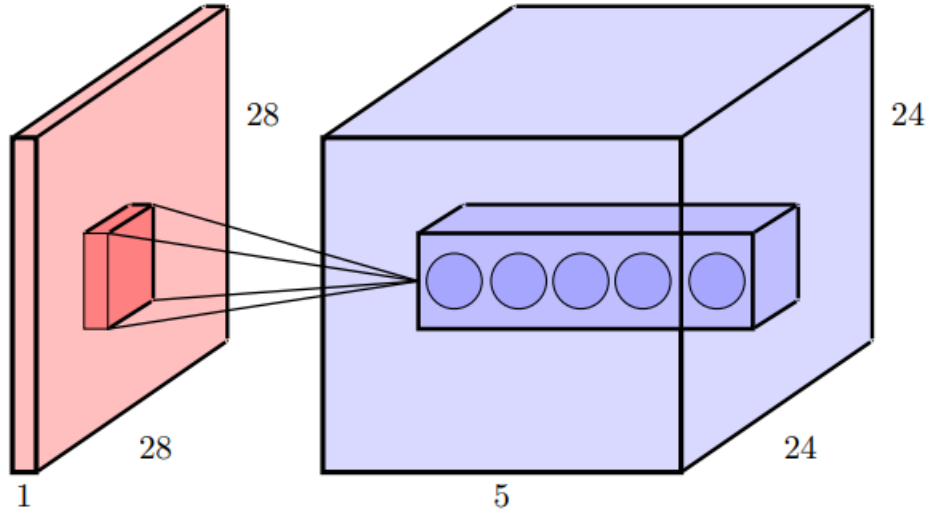
$$s(t) = \sum x(a) w(t - a) \quad (2)$$

$$s(t) = (x * w)(t) \quad (3)$$

Burada x girdi değişkeni olarak kabul edilir, yani görüntü pikselleri ve w çekirdek veya piksellere uygulanan filtre olarak kabul edilir. Çekirdek, görüntüden daha küçük boyutlu bir matristir. Bu işlem, özellik haritası adı verilen bir çıktı elde etmek için çekirdeğe karşılık gelen görüntü pikselleri mahallesini uygular. Bir katmanın çıktısı diğer katmana girdi olarak gönderilir. Bu şekilde, görüntü pikselleri için everişimler gerçekleşir ve modelin öznitelikleri çıkardığı karşılık gelen öznitelik haritaları oluşturulur.

4.3.1 Evrişimsel Katman

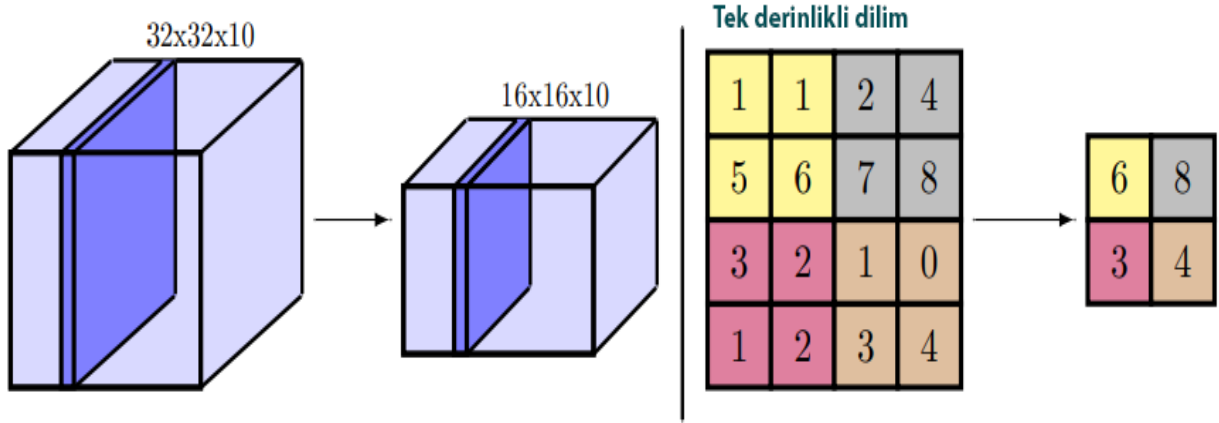
Görüntüleri analiz etmek için ConvLayers'ın iyi sonuçlar elde etmede çok başarılı olduğu kanıtlanmıştır. Bir görüntünün, birlikte görüntüyü oluşturan daha küçük özelliklerden oluşan yüksek boyutlu bir girdi olduğu gerçeğini kullanırlar. Daha küçük özellikleri çıkarmak için bir 'alıcı alan' kullanılır. Alıcı alan, aynı ağırlıklar kullanılarak önceki katmanın tamamına uygulanan boyutları $[n * n \text{ derinliği}]$ olan küçük bir matristir. Her hareketten sonra alıcı alan bir lojistik regresyon işlemi gerçekleştirir ve ağırlık derinlik yönünde önceden tanımlanmış miktarda nöron çıktısı verir. Bu nöronlar, alıcı alan çıkışına bağlıdır, ancak görüntünün başka herhangi bir alanına bağlı değildir. Şekil 4.3da, kırmızı katmandaki $5 \times 5 \times 1$ 'lik bir alıcı alan, mavi katmana beş nöron çıkışı veren daha küçük koyu kırmızı bir alan olarak gösterilmektedir. Alıcı alan daha sonra tüm kırmızı alana uygulanır ve daha sonra mavi katmanı oluşturan her noktada nöronlar üretir. Alıcı alanın her işlem arasında kaç piksel hareket ettiğine "adım" adı verilir. Bir adımda mavi katman, kırmızı katmanla yaklaşık olarak aynı yüksekliğe ve genişliğe sahip olacaktır. Gerçek boyut, kenarlar boyunca ne tür bir dolgunun kullanıldığına bağlı olduğundan, boyutla ilgili olarak yukarıda 'yaklaşık olarak' kullanılmıştır. Kenarların etrafındaki alanı sıfırlarla doldurmak anlamına gelen 'sıfır doldurma', bir sonraki katman için tam olarak aynı genişlik ve yükseklikle sonuçlanırken, 'geçerli doldurma' hem genişliğin hem de yüksekliğin azalmasına neden olur.



Şekil 4-3 Evrişimli katman işlemi örneği. Kırmızı katman bir özellik haritası içerir ve mavi katman beş özellik haritası içerir

4.3.2 Havuzlama

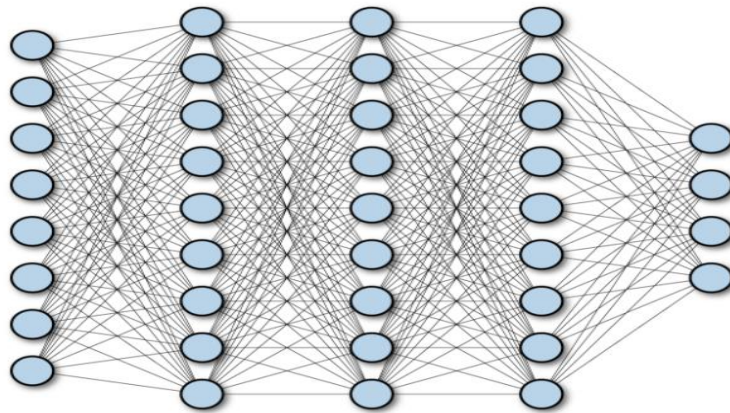
Havuzlama katmanının giriş alanına filtre katmanı denir ve havuzlama katmanının ConvLayer'ın alıcı alanına eşdeğerdir. Adım (stride) parametresi havuz oluşturmada da mevcuttur ve ConvLayer ile aynı şekilde çalışır, ancak aynı değeri alması gerekmez. Farklı havuzlama türleri vardır, ancak en başarılı olduğu gösterilen varyant maksimum havuzlamadır ve bu nedenle bu projede kullanılacaktır. Maksimum havuzlamanın nasıl kullanılabileceğine dair bir örnek, 2x2'lik bir adım ve filtre boyutunun kullanıldığı Şekil 4.4'da gösterilmektedir.



Şekil 4-4 2x2'lik bir adım ve filter boyutunun

4.3.3 Tam Bağlantılı Ağlar

Tam bağlantılı bir sinir ağı, bir katmandaki her nöronu diğer katmandaki her nörona bağlayan bir dizi tam bağlantılı katmandan oluşur. Tam bağlantılı ağların en büyük avantajı, "yapıdan bağımsız" olmalarıdır, yani girdi hakkında yapılması gereken özel varsayımlar yoktur. Yapıdan bağımsız olmaları, tam bağlantılı ağları çok geniş bir şekilde uygulanabilir kılarken, bu tür ağlar, bir sorun alanının yapısına ayarlanmış özel amaçlı ağlardan daha düşük performansa sahip olma eğilimindedir.

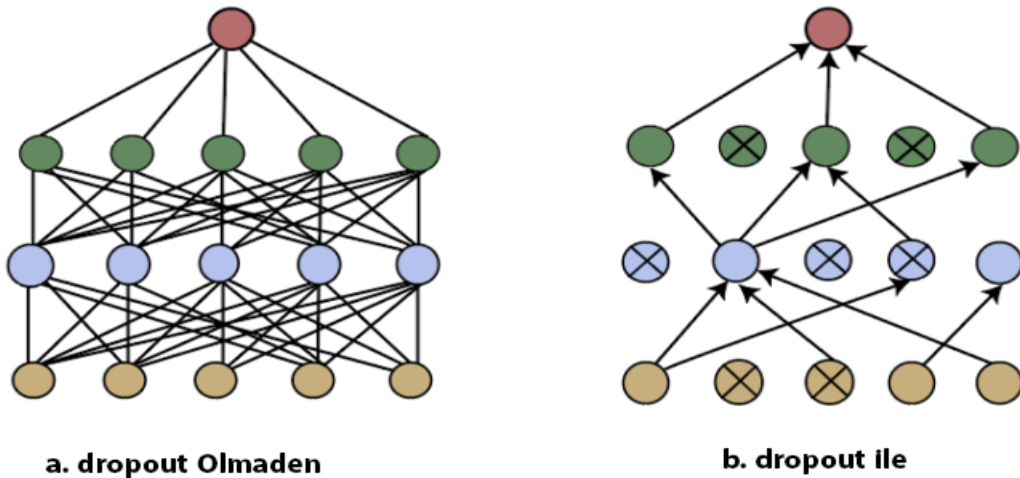


Şekil 4-0-5 Çok Katmanlı Derin Tam Bağlantılı Ağ, Görüntü Kaynağı

4.3.4 Seyreltme Katmanı

Az sayıda örnek içeren bir eğitim veri kümesinin, derin öğrenme sinir ağlarına hızlı bir şekilde uyması muhtemeldir. Alternatif model kombinasyonlarına sahip sinir ağları toplulukları tarafından fazla uydurmanın azaltıldığı bilinmektedir, ancak birçok modelin eğitimi ve bakımı, hesaplama maliyetine ve zamanına katkıda bulunur. Eğitim sırasında düğümleri rastgele bırakarak, çok çeşitli farklı ağ modellerine sahip olmayı simüle etmek için tek bir model kullanılabilir. Seyreltme, her türden derin sinir ağlarında fazla uydurmayı azaltan ve genelleme hatasını iyileştiren bir düzenleme stratejisidir. Hesaplama açısından ucuz ve şaşırtıcı derecede etkilidir [69].

Sınırlı veri kümeleri üzerinde eğitilmiş devasa sinir ağları, eğitim verilerine fazla uyma eğilimindedir. Mimari, test veri kümesi gibi yeni veriler üzerinde test edildiğinde, model eğitim verilerindeki rastgele varyasyonu öğrenir ve bu da düşük performansa neden olur. Fazla takma tahmin hatasını artırır. Tüm uygulanabilir farklı sinir ağlarını aynı veri kümesine yerleştirmek ve her mimariden tahminlerin ortalamasını almak, fazla uydurmayı azaltmanın bir yoludur. Gerçekte, bu mümkün değildir, ancak topluluk olarak bilinen küçük bir farklı model grubu kullanılarak yaklaşık olarak tahmin edilebilir. Bir sinir ağında, katman başına Seyreltme uygulanabilir. Yoğun tam bağlı katmanlar, erişimli katmanlar ve uzun kısa süreli bellek ağ katmanı gibi tekrarlayan katmanlar dahil olmak üzere çeşitli katman türleri onunla kullanılabilir. Seyreltme, ağıın tüm veya bazı gizli katmanlarında ve ayrıca görünür veya giriş katmanında kullanılabilir. Ancak seyreltme çıkış katmanında kullanılamaz, Seyreltme tekniği aşağıda Şekil 4-6'da gösterilmiştir.



Şekil 4-6 seyreltme tekniği

4.3.5 Aktivasyon Fonksiyonları

Aktivasyon fonksiyonları, girdi verisini bir sonraki katmana girdi olarak gönderilen çıktı verisine dönüştürmek için YSA'da kullanılır. Bir sinir ağı modelinin tahmin doğruluğu, kullanılan aktivasyon fonksiyonunun tipine bağlıdır [70]. Aktivasyon fonksiyonları kullanılmazsa, çıktı verileri basit bir lineer polinom fonksiyonu olacaktır. Model, verilerden karmaşık eşlemeleri öğrenemez ve tanıyamaz. Bu nedenle sinir ağı görüntü, konuşma, metin, video, ses vb. karmaşık veri türlerinin modellenmesinde etkili olmayacaktır. ReLU ve Seyreltme bu tezde kullandığımız iki aktivasyon fonksiyonudur.

- ReLU

ReLU sinir ağlarında yaygın olarak kullanılan doğrusal olmayan aktivasyon fonksiyonuna ait düzeltilmiş doğrusal birim anlamına gelir . ReLU etkinleştirme işlevi, ağırlık başlatma ve kaybolan gradyanların yükünü azaltarak derin sinir ağlarının eğitimi kolaylaştırdı [71]. ReLU, tüm nöronları aktive etmek yerine aynı anda sadece birkaç nöron aktive edildiğinden diğer aktivasyon fonksiyonlarından daha verimlidir. ReLU, pozitif kısmı bir kimlik fonksiyonu (x) olan ve negatif kısmı sıfır olan parçalı doğrusal bir fonksiyondur [72]. ReLU işlevi matematiksel olarak şu şekilde temsil edilebilir:

$$f(x) = \frac{1}{1 + e^{-x}} \quad (4)$$

- Seyreltme

Seyreltme aktivasyon fonksiyonu, çoklu sigmoid aktivasyon fonksiyonlarının bir kombinasyonudur. Bu işlevin çıktı değerleri, belirli sınıf veri noktaları olarak değerlendirilebilecek 0 ile 1 aralığında olacaktır. İkili sınıflandırma için kullanılan sigmoid işlevinden farklı olarak Seyreltme işlevi, çoklu sınıf sınıflandırması için kullanılabilir [70]. Seyreltme etkinleştirme işlevi kullanılarak oluşturulan modeller, sınıf sayısı ile aynı sayıda nörona sahip bir çıktı katmanına sahiptir. Seyreltme işlevi matematiksel olarak şu şekilde ifade edilebilir:

$$\delta(z)_j = \frac{e^{z_j}}{\sum_{k=1}^K e^{z_k}} \text{ for } j = 1, \dots, K. \quad (5)$$

Burada, bir giriş vektörü z 'nin her z_j elemanına bir üstel fonksiyon uygulanır ve bu değerleri tüm üstlerin toplamına bölerek normalize eder. $\delta(z)$ çıktı vektöründeki bileşenlerin toplamıdır.



5 CNN Mimarisi

5.1 R-CNN

R-CNN (Bölgesel Evrişimli Sinir Ağı) [73], anlamsal görüntü bölgelerinin hedeflere katkısı için PASCAL VOC veri seti yarışmasında yüksek doğruluğu kanıtlanmış yeni bir CNN yapısıdır. Bu önerinin avantajı, CNN özelliklerine dayanmaktadır, ancak CNN özellikleri rastgele seçilmez özelliklerinin görüntü içeriğini daha iyi yansıtabilmesi için başlangıçta semantik segmentasyonlarla oluşturulur. R-CNN için bölge önerisi seçici arama ile uygulanmaktadır [74]. Seçici arama, çeşitli seçim stratejileri kullanan ve sonuçları birleştiren bir nesne algılama algoritmasıdır. Dikkate alınan dört temel koşul vardır: Doku, renk, boyut ve örtüşme [74]. Nihai konum hipotezleri, nitelik ve nicelik arasında bir değiş tokuştan sonra önerilmiştir. Konum hipotezleri sıralanır ve seçilen konum sınırlama kutularının bir nesneyi içermesi açısından yüksek güvenilirliğe sahip olmasını sağlamak için düşük dereceli teklifler kaldırılır.

Konum hipotezleri derin öğrenme ağlarına beslenir. [74]'de önerilen modelde, son sınıflandırma problemi için 4096 öznitelik ve SVM çıkarabilen büyük bir CNN düşünülmüştür.

5.2 Fast R-CNN

Fast R-CNN [47], hesaplama basitleştirmesi yoluyla daha hızlı algılama hızıyla R-CNN'nin bir ileri adımıdır. Bu ağ, orijinal R-CNN teklifinden 9 kat daha hızlı olan VGG16 ağını kullanır. Bu algoritma, bölge önerilerini beslemek yerine, sınıflandırmalara özellik haritaları uygular. İlgi Bölgesi (RoI) havuzda toplanır ve sabit boyutlu bir özellik bölgesine dönüştürülür. Sınıflandırma için tam bağlantılı katmana sahip Seyreltme kullanılır.

5.3 Faster R-CNN

Son 10 yılda birkaç CNN mimarisi sunuldu [21, 26]. Model mimarisi , farklı uygulamaların performansını iyileştirmede kritik bir faktördür. 1989'dan günümüze CNN mimarisinde çeşitli modifikasyonlar yapılmıştır. Bu tür modifikasyonlar arasında yapısal reformülasyon, düzenlileştirme, parametre optimizasyonları vb. yer alır. Tersine, CNN performansındaki kilit yükseltmenin, büyük ölçüde işlem biriminin yeniden düzenlenmesi ve yeni blokların geliştirilmesi nedeniyle gerçekleştiği

belirtilmelidir. Özellikle, CNN mimarilerindeki en yeni gelişmeler ağ derinliğinin kullanımı konusunda gerçekleştirilmiştir. Bu bölümde, 2012'de Alex Net modelinden başlayıp 2020'de Yüksek Çözünürlüklü (HR) modele kadar uzanan en popüler CNN mimarilerini incelenmiştir. Bu mimarilerin özelliklerini (giriş boyutu, derinlik ve sağlamlık gibi) incelemek, araştırmacıların hedef görevleri için uygun mimariyi seçmelerine yardımcı olmak amaçlanmıştır.

Daha Hızlı R-CNN [73] yüksek doğruluk ve hızlı hesaplama hızı için oluşturulmuştur. Hızlı R-CNN'nin bir ilerlemesidir. Hem R-CNN hem de Hızlı RCNN, zaman alıcı olan seçici aramayı kullanır. Bu algoritma bize kaynak görüntülerden öğrenmek ve özellik haritaları üretmek için erişimli bir sinir ağı önerir. Oluşturulan özellikler, Hızlı R-CNN ile aynı CNN'ye beslenir.

R-CNN ailesi klasiktir ancak hesaplama hızını hızlandıran yeni mekanizmalara rağmen diğer yaklaşımların aksine yavaş kalır. Hesaplama yoğun olan bölge çıkarımları gerektirir. Daha Hızlı R-CNN, Bölge Teklif Ağı'nı (RPN) tanıtarak bu konuyu ele alsa da, model oluşturma karmaşıklığını ekler.

5.4 SPPnet

Mekansal piramit havuzlama ağı (SPPnet) [75], hesaplama hızını hızlandırır ve paylaşılan hesaplama ile doğruluğu artırır. Havuzlama tekniği ile seçilen özellikler, çeşitli filtre boyutları ile işlenir ve daha sonra tamamen bağlı bir ağa giriş olarak birleştirilir.

Bu yapı, küresel ve yerel görsel özellikleri yansıtır. Karşılaştırılabilir performans ölçütleriyle R-CNN'den daha hızlıdır, ancak diğer yaklaşımlara kıyasla yeterince hızlı değildir.

5.5 AlexNet

Alex Net, CNN'nin çeşitli görüntü tanıma yarışmalarında gösterdiği performans nedeniyle ödüllendirilen belirli bir yapısının adıdır [76]. Bu çalışma, eğitim sırasında (GPU aracılığıyla) hesaplama verimliliği üzerinde muazzam etkileri olan sinir ağı derinliğinin önemini ortaya koymaktadır.

AlexNet , beş everişimsel katmana ve ardından üç tam bağlantılı katmana sahiptir. Ağ, tahmin için son çıktı katmanına kadar her biri izole edilmiş bir GPU üzerinde eğitilerek yarıyı kullanılarak ayrılır.

5.6 Mobil Net

Sandler, Howard (2018) , verimlilik ve hesaplama maliyeti arasında iyi bir uzlaşma sağlayan Mobile Net modelini önerdi [77]. Mobile Net, yüksek sınıflandırma verimliliğini korurken, evrişimli katman parametrelerini önemli ölçüde düşürmek için ayrılabilir evrişim ve derinlikli evrişim kullandı. Özellik haritalarının yeniden kullanılması için Mobile Net'te artık darboğaz blokları önerildi. Sonuç olarak, Mobile Net önemli ölçüde daha kısa eğitim süresi ile yüksek sınıflandırma doğruluğu sağlayabilir. Mobile Net mimarisi, standart bir evrişimi derinlik esaslı bir evrişim ve bir 1x1 noktasal evrişim olarak bölen bir tür çarpanlara ayrılmış evrişim olan, derinlemesine ayrılabilir evrişimler üzerine inşa edilmiştir. Mobile Net tarafından kullanılan derinlik bilgili evrişim, her giriş kanalı için ayrı bir filtre sunar. Derinlemesine evrişimin çıktıları daha sonra noktasal evrişim yoluyla bile 1x1 evrişim kullanılarak birleştirilir. Bir aşamada, geleneksel bir evrişim, yeni bir sonuç kümesi geliştirmek için girdileri filtreler ve birleştirir. Derinlemesine ayrılabilir evrişim, onu iki katmana böler: biri filtrasyon için, diğeri kombinasyon için. Bu çarpanlara ayırma, hesaplama süresini ve model boyutunu önemli ölçüde azaltır. Tam bir evrişim olan ilk katmanının yanı sıra, Mobile Net yapısı derinlemesine ayrılabilir evrişimler üzerine kuruludur. Ağı bu temel terimlerle belirterek uygun bir ağı belirlemek için ağ yapılarını hızla keşfedebiliriz. Doğrusallığı olmayan ve bunu sınıflandırma için Seyreltme katmanına besleyen son tam bağlı katman dışında, modelin tüm katmanlarını toplu normalleştirme ve ReLU doğrusalsızlık takip eder. Derinlik bazında evrişim, 1x1 noktasal evrişim ve toplu normalleştirme ve her evrişim katmanından sonra ReLU ile, toplu normalleştirme ve ReLU doğrusal olmama durumu faktör eşitilmiş katmana eklenir. Derinlemesine erişimlerde ve ayrıca ilk katmanda, aşağı örnek adımlı evrişim yoluyla gerçekleştirilir. Tamamen bağlı katmandan önce bile, son bir havuzlama katmanı uzamsal çözünürlüğü 1'e düşürür. Derinlik ve noktasal konvolüsyonlar bağımsız katmanlar olarak sayılır.

5.7 YOLO

Nesne algılama, neyi ve nerede olduğunu, özellikle de belirli bir görüntünün içinde hangi nesnelerin olduğunu ve ayrıca görüntüde nerede olduklarını tanımaya çalıştığınız bilgisayarla görme alanındaki klasik sorunlardan biridir. Nesne algılama sorunu, nesneleri tanıyabilen ancak nesnenin görüntüde nerede olduğunu göstermeyen sınıflandırmadan daha karmaşıktır. Ayrıca sınıflandırma, birden fazla nesne içeren görüntülerde çalışmaz.

YOLO tamamen farklı bir yaklaşım kullanır. YOLO, gerçek zamanlı olarak nesne algılama yapmak için akıllı bir evrişimli sinir ağıdır (CNN). Algoritma, tam görüntüye tek bir sinir ağı uygular ve ardından görüntüyü bölgelere ayırır ve her bölge için sınırlayıcı kutuları ve olasılıkları tahmin eder. Bu sınırlayıcı kutular, tahmin edilen olasılıklarla ağırlıklandırılır. YOLO, gerçek zamanlı olarak da çalışabilmesinin yanı sıra yüksek doğruluk sağladığı için popülerdir. Algoritma, tahminlerde bulunmak için sinir ağından yalnızca bir ileri yayılım geçişi gerektirmesi anlamında görüntüye "sadece bir kez bakar". Maksimum olmayan bastırmadan sonra (bu, nesne algılama algoritmasının her nesneyi yalnızca bir kez algılamasını sağlar), ardından tanınan nesneleri sınırlayıcı kutularla birlikte çıkarır. YOLO ile, tek bir CNN aynı anda birden fazla sınırlayıcı kutu ve bu kutular için sınıf olasılıklarını tahmin eder. YOLO, tam görüntüler üzerinde eğitim alır ve algılama performansını doğrudan optimize eder. Bu modelin diğer nesne algılama yöntemlerine göre birtakım avantajları vardır:

- YOLO son derece hızlı
- YOLO, eğitim ve test süresi boyunca görüntünün tamamını görür, böylece sınıflar ve görünüşleri hakkında bağlamsal bilgileri örtük olarak kodlar.
- YOLO, nesnelerin genelleştirilebilir temsillerini öğrenir, böylece doğal görüntüler üzerinde eğitildiğinde ve sanat eserleri üzerinde test edildiğinde, algoritma diğer en iyi algılama yöntemlerinden daha iyi performans gösterir.

YOLO, Darknet sınıflandırıcısının üzerine inşa edilmiş bir evrişimsel sinir ağıdır. Hem YOLO hem de Darknet başlangıçta Joseph Redmon tarafından C-yazılı açık kaynaklı sinir ağı çerçevesi Darknet üzerinde çalışacak şekilde tasarlanmış ve geliştirilmiştir. YOLO'yu rakibiyle karşılaştırıldığında öne çıkaran şey hızıdır: eşit doğrulukta YOLO, oyun katmanı GPU kullanarak hareket eden nesneleri 30 fps videolarda gerçek zamanlı olarak izleme yeteneği elde ederek, zamanın yarısından daha kısa sürede çıkarım yapabilir.

Bu çalışmada YOLOV3 ve YOLOV7 sürümü kullanılmıştır. İlk olarak, sınırlayıcı kutular, sınırlayıcı kutular yapmak için çapalar kullanılarak önceki sürümlerle aynıdır. RP içinde nesne bulunup bulunmadığına her nesne için bir sınırlayıcı kutu atanmasına bağlı olarak karar verilir ve atanmamışsa tahakkuk ettirilir [73]. Sınıf tahmininde, önceki sürümlerden daha hızlı olmasını sağlayan Seyreltme tahmini yerine bağımsız lojistik sınıflandırma kullanmışlardır [73]. Aşağıdaki Şekil 5-1, Darknet-53 ile YOLOV3'ün mimarisini göstermektedir:

	Type	Filters	Size	Output
	Convolutional	32	3×3	256×256
	Convolutional	64	$3 \times 3 / 2$	128×128
1x	Convolutional	32	1×1	
	Convolutional	64	3×3	
	Residual			128×128
	Convolutional	128	$3 \times 3 / 2$	64×64
2x	Convolutional	64	1×1	
	Convolutional	128	3×3	
	Residual			64×64
	Convolutional	256	$3 \times 3 / 2$	32×32
8x	Convolutional	128	1×1	
	Convolutional	256	3×3	
	Residual			32×32
	Convolutional	512	$3 \times 3 / 2$	16×16
8x	Convolutional	256	1×1	
	Convolutional	512	3×3	
	Residual			16×16
	Convolutional	1024	$3 \times 3 / 2$	8×8
4x	Convolutional	512	1×1	
	Convolutional	1024	3×3	
	Residual			8×8
	Avgpool		Global	
	Connected		1000	
	Softmax			

Şekil 5-1 YOLOV3 Mimarisi

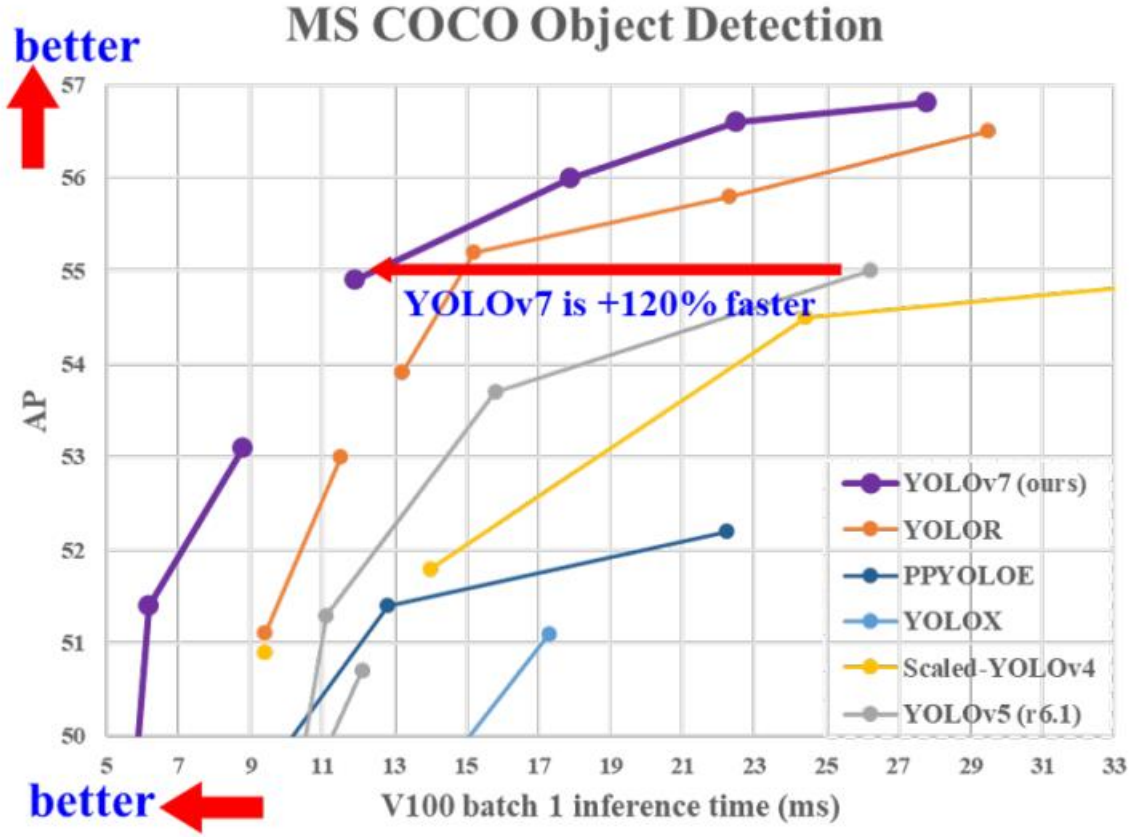
5.8 YOLOV7

Gerçek zamanlı nesne tanımlama, çoklu nesne takibi, otonom sürüş, robotik ve tıbbi resim analizi dahil olmak üzere bilgisayarlı görü sistemlerinde sıklıkla gerekli olduğu için bilgisayarlı görüde kritik bir konudur. Gerçek zamanlı nesne tanımlama, genellikle bir mobil CPU veya GPU ve önemli üreticiler tarafından sağlanan çok sayıda sinirsel işlem birimi (NPU) tarafından gerçekleştirilir. NPU'lar arasında Apple sinir motoru (Apple), Intel sinir bilgi işlem çubuğu, Jetson AI uç cihazları (Nvidia), Google uç TPU, Qualcomm sinir işleme motoru, MediaTek AI işlem birimi ve Keron AI SoC'leri bulunur.

Daha önce belirtilen bazı uç cihazlar, vanilya evrişim, derinlik bilgili evrişim veya MLP işlemleri gibi belirli süreçleri hızlandırmaya odaklanmıştır. Son yıllarda, çeşitli uç cihazlar için gerçek zamanlı nesne dedektörleri oluşturulmuştur. Örneğin MCUNet ve NanoDet, düşük güçlü tek çip üretmeye ve uç CPU'larda çıkarım performansını artırmaya odaklandı. YOLOX ve YOLOR gibi yöntemler, çeşitli GPU'ların çıkarım hızını artırmaya odaklanır.

Son zamanlarda, gerçek zamanlı nesne dedektörü geliştirme, verimli mimari oluşturmaya odaklandı. Bir CPU'da kullanılan gerçek zamanlı nesne dedektörleri genellikle Mobile Net, ShuffleNet veya GhostNet'e dayanır. GPU için bir başka popüler gerçek zamanlı nesne dedektörü ResNet, DarkNet veya DLA'dır ve ardından mimariyi geliştirmek için CSPNet tekniği kullanılır. YOLOv7 araştırmada önerilen yaklaşımlar, geliştirme yollarında mevcut ana akım gerçek zamanlı nesne dedektörlerinden farklıdır. Mimari optimizasyona ek olarak, önerilen yaklaşımlar eğitim süreci optimizasyonuna odaklanacaktır.

YOLO7 Makalesi, nesne algılama doğruluğunu iyileştirmek için çıkarım maliyetini düşürürken eğitim maliyetini artırabilecek birkaç geliştirilmiş modül ve optimizasyon stratejisine odaklanacaktır. Önerilen modüller ve optimizasyon yaklaşımları, eğitilebilir ücretsiz paketler olarak adlandırılır. Model yeniden parametrelendirme ve dinamik etiket atama, ağ eğitimi ve nesne tanımlamada son zamanlarda kritik konular olarak ortaya çıkmıştır. Nesne dedektörlerinin eğitimi, özellikle yukarıda belirtilen yeni kavramlar tanıtıldığında, birkaç yeni zorluk geliştirir



Şekil 5-2 YOLOv7 Diğer gerçek zamanlı nesne dedektörleriyle karşılaştırma.

YOLOv7 makalesi, ortaya çıkardıkları bazı yeni zorlukları açıklayacak ve pratik çözümler sağlayacaktır. Araştırmacılar, gradyan yayılma yolu fikrini kullanarak çeşitli ağlardaki katmanlara uygulanabilir model yeniden parametre eştirme tekniklerini inceler ve planlı bir yeniden parametre eştirilmiş model sunar. Ayrıca, birçok çıktı katmanına sahip bir modeli eğitmenin, dinamik etiket atama teknolojisini kullanırken ek zorluklar yarattığını keşfettiler. “Farklı branşların çıktılarına dinamik hedefler nasıl atanabilir?”

Kalın-ince kurşun kılavuzlu etiket ataması olarak adlandırılan yaklaşım için yeni bir etiket ataması önerilmiştir.

YOLOv7 makalenin katkıları aşağıdaki gibi özetlenmiştir:

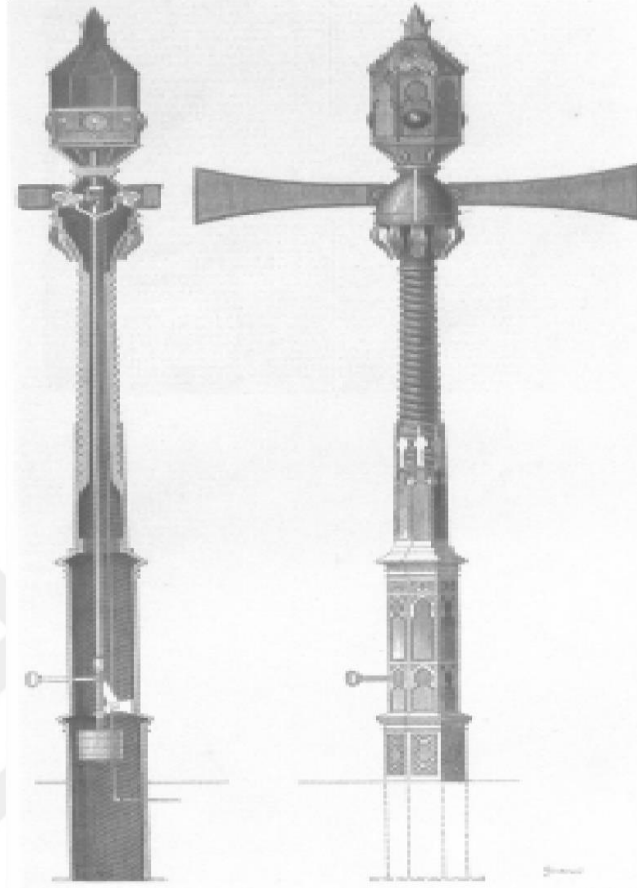
- ❖ Gerçek zamanlı nesne algılamanın, çıkarım maliyetini artırmadan algılama doğruluğunu önemli ölçüde artırabilmesi için eğitilebilir birkaç ücretsiz paket yöntemi tasarlanır.
- ❖ Nesne algılama yöntemlerinin gelişimi için iki yeni konu keşfedildi: yeniden parametrelendirilmiş bir modülün orijinal modülü nasıl değiştirdiği ve dinamik etiket atama stratejisinin farklı çıktı katmanlarına atamayla nasıl ilgilendiği
- ❖ (3) Parametreleri ve hesaplamayı etkin bir şekilde kullanabilen gerçek zamanlı nesne dedektörleri için “genişletme” ve “birleşik ölçekleme” yöntemleri önermek
- ❖ Önerilen yöntem, son teknoloji bir gerçek zamanlı nesne dedektörünün parametrelerinin yaklaşık %40'ını ve hesaplamasının %50'sini etkili bir şekilde azaltabilirken, aynı zamanda daha hızlı çıkarım hızına ve daha yüksek algılama doğruluğuna sahiptir.

6 TRAFİK İŞARETLERİ

Uluslararası Aydınlatma Komisyonu'na (CIE) göre QA işareti, durumu, şekli, rengi veya deseni nedeniyle ve bazen semboller veya alfa sayısal karakterler kullanılarak görsel bir mesaj sağlayan bir cihazdır. Bu kısa mesaj bilgi aktarmak için kullanılır. Böyle bir iletişimin amacı, alıcının gönderenin ne anlama geldiğini anlamasını sağlamaktır. Gönderici, işaret tasarımcısıdır ve sürücünün ayırt etme, yorumlama ve hatırlama becerilerini ve çevre koşullarını dikkate almalıdır. Trafik için uyarılar ve talimatlar, hız sınırları, yönler ve diğer bilgiler, ayrıca yol işaretleri, trafik ışığı sinyalleri, otoyol matris sinyalleri, yol çalışmalarında kullanılan koniler ve silindirler[78]

6.1 Trafik ışıklarının doğuşu

Şehir trafiğini yönlendirmeye yönelik ilk cihazın Londra'da tanıtıldığı söyleniyor. 1868'de, bir parlamento seçim komitesinin emriyle, bir demiryolu sinyal üreticisi, Parlamento Evleri'nin önündeki bir kavşakta tepesinde bir ışık bulunan bir semafor inşa etti. Parlamento üyelerine olay yerindeki yoğun araç trafiğinde güvenli bir şekilde rehberlik etmesi gerekiyordu. Semaforun kolları ya yatay konumda ya da 45° açıyla indirilerek trafik yönlendirildi. Makinenin kolları yatay konumdayken trafiğin durması gerekiyordu, ikinci konum sürücülerini 'dikkatli' ilerlemeye çağırırdı. Geceleri direğin tepesindeki lamba yanıyordu. Kollar yatay pozisyondayken kırmızı ışık yanıyordu. Kollar indirildiğinde yeşil ışık yanıyordu. Gazla çalışan lamba, gaz sızıntısı nedeniyle birkaç ay çalıştıktan sonra patlasa da sinyalin kendisi başarılı olarak kabul edildi. Avam Kamarası, halkın semafora uymaya istekli olduğu ve böylece trafik güvenliğinin sağlandığı sonucuna varmıştır. Bu bulgulara yanıt olarak, demiryolu sinyal şirketi geliştirilmiş bir tasarım geliştirdi. Birkaç yıl sonra Amsterdam, kavşaklarda trafik düzenlemesi de getirdi. 1870'lerin ortalarında Amsterdam'da ilk atlı tramvayların tanıtılmasıyla birlikte, yerel yetkililer, şehrin dar sokaklarındaki trafiği korumak için düzenlemenin gerekli olduğunu hissettiler. 1876'da şehir merkezinin doğusundaki bir tramvay güzergahında iki dar sokağın kavşağına bir polis yerleştirdiler. Memur el gesi kullandı trafik akışını yönlendirmek için. 1880'lerde tramvay ağının şehir içinde yaygınlaşmasıyla birlikte polislerin yerini alacak yeni sinyalizasyon türleri geliştirildi. Amsterdam tramvay şirketi, çift hat için yer bulunmayan dar sokaklarda iki yönlü tramvay trafiğini düzenlemek için [79].



Şekil 6-1 Londra'nın ilk trafik sinyali, 1868

6.2 Trafik kontrol sistemleri

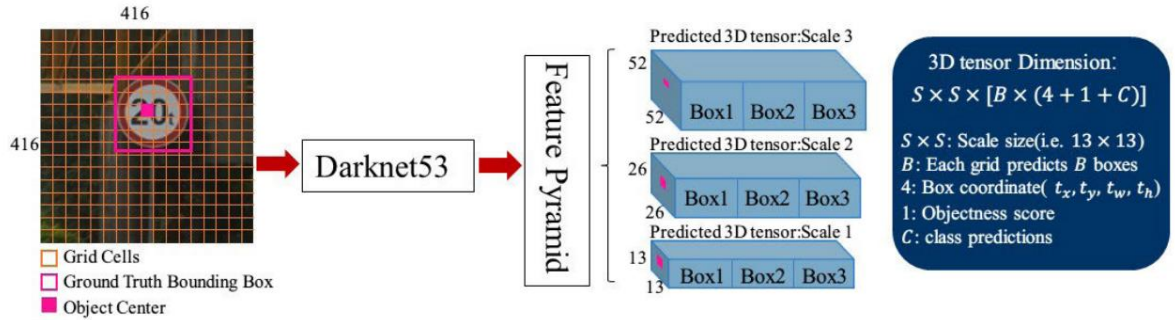
Trafik kontrol sistemleri, kentsel altyapının en görünür ögesidir. Teknolojik unsurları, trafik ışıkları, tabelaları ve boyalı kaldırımları bu tanıma uysa da bunlar sadece telefon kanalizasyonları veya sokaklar gibi fiziksel sistemler değildir. Aksine, hareket etmek ya da hareketsiz kalmak gibi en temel insan davranışları üzerinde güçlü bir sosyal kontrol empoze etmeye çalışan sistemlerdir. Trafik mühendisleri polisi, sürücüleri ve yayaları kontrol etmelidir. Kent altyapısının diğer birçok unsuru için, kullanıcıların davranışlarını kontrol etmek tasarımcıların birincil amacını oluşturmadı. Trafik mühendisleri için, sürücülerin ve yayaların davranış kalıplarını anlamak ve manipüle etmek (sadece yayaları değil, aynı zamanda araç kullanan insanları da içeren bir grup). Oyun, sosyal toplantılar ve ticaret için sokak) kontrol mekanizmalarının kendisinden daha önemli bir sorun olduğunu kanıtladı.[80]

7 METODOLOJİ

Bu bölüm, tez çalışmasının aşamalarını, deney düzeneğini, simülasyon aracını ve ölçülecek metrikleri içeren bu çalışma için önerilen araştırma çerçevesini açıklamaktadır.

7.1 Sadece bir kez bakarsın (YOLO)

Şekil 7.1'de gösterildiği gibi, YOLOv3 tanıma boru hatlarını kullanan tek bir sinir ağı, tüm görüntüden tüm sınırlayıcı kutuları ve kategori olasılıklarını üretir. Ağ, her resmi eğitim kümesine S (örneğin, $S = 13$) ızgaralarına böler. Her ızgara, farklı boyutlarda potansiyel kaplar sunar. Nesnenin temel gerçeği merkezi ızgara sınırları içindeyse, ağ nesne algılamaya yardımcı olur. Kıvrımlı katman daha sonra özellikleri elde etmek için kullanılır (Darknet53). Son olarak, YOLO katmanı kullanılarak çeşitli ölçekler tahmin edilir. Sınırlayıcı kutular, güven puanları ve sınıf koşullu olasılıkların tümü, her bir ızgara tarafından tahmin edilir.



Şekil 7-1 YOLOv3 Tespit Mekanizması.[81]

7.2 Veri Kümesi

Trafik işareti tespiti için en popüler veri setlerinden biri, Alman Trafik İşareti Tespit Kıyaslamasıdır (GTSDb) [14]. GTSDb trafik göstergelerinin kabaca üç kategoriye ayrılır (yani, zorunlu işaretler, tehlike işaretleri ve yasaklayıcı işaretler). Veri kümesi biçimlendirilmelidir, böylece YOLOv3 beslenmeden önce anlar. GTSDb veri tabanındaki fotoğraflar ppm formatını kullanır. YOLOv 3 için GTSDb kullanmak için ppm formatındaki fotoğrafları jpg dosyalarına dönüştürülmesi

gerekir. Fotoğraflara isim vermeli ve her biri için metin dosyalarının oluşturması gerekir. Genellikle dört dosya oluşturulur: obj.names , obj.data , train.txt ve test.txt. Nesne adları dosyası tüm veri kümesinin sınıflarını listelerken, train.txt ve test.txt dosyaları sırasıyla eğitim ve test fotoğraflarının yolunu içerir.

GTSDDB ve GTSRB veri kümeleri, sırasıyla trafik işareti algılama ve tanıma modellerini eğitmek ve değerlendirmek için çok popüler veri kümeleridir.

GTSDDB veri seti, Bochum, Almanya yakınlarında kaydedilen video dizilerinden oluşturulmuştur. Farklı hava ve aydınlatma koşulları altında kentsel, kırsal ve otoyol senaryolarından görüntülerden oluşur ve bu onu oldukça temsili ve zorlu hale getirir. Toplam veri seti, 600 eğitim görüntüsü ve 300 test görüntüsüne bölünmüş, ham PPM formatında 1360×800 piksellik 900 görüntüden oluşur. Eğitim görüntüleri 846 trafik işareti içerir ve test görüntüleri 360 trafik işareti içerir. Görsellerdeki trafik işareti boyutları 16 ile 128 piksel arasında değişmektedir.

7.3 Deney Tasarımı

Modeli eğitmek için Google Colab Notebook kullanıldı. Google Colab , hesaplama gücüne ve GPU'lara ücretsiz, kurulum gerektirmeyen erişim sağlayan, barındırılan bir Jupyter dizüstü bilgisayar hizmetidir. Tarayıcı tabanlı yardımcı program Colab , kullanıcıların herhangi bir Python kodunu yazmasına ve yürütmesine olanak tanır. Öğretim, veri analizi ve makine öğrenimi için iyi çalışır. Colab'da 12 GB'lık bir GPU mevcuttur. Müşteriler, Colab'da farklı zamanlarda farklı türde GPU'lar kullanabilir .

❖ Python

Python, yorumlanmış üst düzey, genel amaçlı, kolay okunabilir bir programlama dilidir. Dinamik olarak yazılır ve çöp toplanır. Prosedürel, işlevsel ve nesne yönelimli programlamayı destekler. Python, birçok göreve uygun araçlar sağlayan büyük bir kitaplık koleksiyonuna sahiptir. Python, Veri Bilimi, Makine Öğrenimi ve Yapay Zeka için popülerdir. Flask, Django vb. iyi çerçeveler içerdiğinden web geliştirme için de popülerdir. Bu tezde kullanılan python kitaplıkları aşağıda açıklanmıştır.

❖ OpenCV

Bu kütüphane, derin bir sinir ağı öğrenme modülüne sahip, bilgisayarla görü için başka bir açık kaynak kütüphanedir. YOLOv3'ü desteklediği için test etmek için kullanıldı. Darknet'in yalnızca CPU üzerinde çalışması için yeniden derlenmesi gerekeceğinden Darknet yerine OpenCV kullanıldı, diğer modeller bu şekilde test edildi.

7.4 Performans Değerlendirme Ölçütleri

Deneyisel sonuçları değerlendirmek için üç ölçüm kullanılır. Duyarlılık, doğruluk ve kesinliği içeren bu ölçümler aşağıdaki gibi hesaplanır:

Ortalamaların Ortalaması Hassasiyet (map)

Terimi tanımlamak için, Ortalamaların Ortalaması Hassasiyet (veya mAP), Nesne Algılama algoritmalarını değerlendirmek için tasarlanmış bir Makine Öğrenimi metriğidir. Açıklığa kavuşturmak için, günümüzde mAP'yi Örnek ve Semantik Segmentasyon modellerini değerlendirmek için de kullanabilirsiniz. FP ve FN, yanlış pozitif ve negatiflerin sayılarıdır. TP ve TN, gerçek pozitif ve negatiflerin sayılarıdır.

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)} \quad (6)$$

$$Recall \text{ (Sensitivity)} = \frac{TP}{(TP + FN)} \quad (7)$$

$$Precision = \frac{TP}{(TP + FP)} \quad (8)$$

$$Fscore = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (9)$$

$$mAP = \frac{1}{N} \sum_{k=1}^{k=n} AP_k \quad (10)$$

8 BULGULAR VE KARŞILAŞTIRMALAR

Modeli eğitmek için Darknet çerçevesini kullanıldı. Eğitim için kullanılan parametreler Tablo 8-1' de açıklanmıştır. Tablo 8-1' den de görülebileceği gibi, ağ boyutu (girdi) boyut 608×608 . Ağa besleme yapmadan önce, girdi görüntüleri bu uzamsal boyuta göre yeniden boyutlandırıldı. Algılama doğruluğunu artırmak için varsayılan 416×416 giriş boyutu yerine 608×608 giriş boyutu seçildi. Görüntüler ayrıca her biri 64 adetlik gruplar halinde toplanmıştır. Bir dizi alt bölüm değeri olarak 16 olarak belirlendi. Toplu iş parametresi, bir yineleme sırasında işlenen görüntülerin sayısını temsil eder. Alt bölüm parametresi, GPU'nun bir kerede işlediği bir toplu iş içindeki mini toplu işlerin sayısını temsil eder. Ağırlıklar, bu tür her yinelemeden sonra güncellendi.

Tablo 8-1 Modelleme Parametreleri

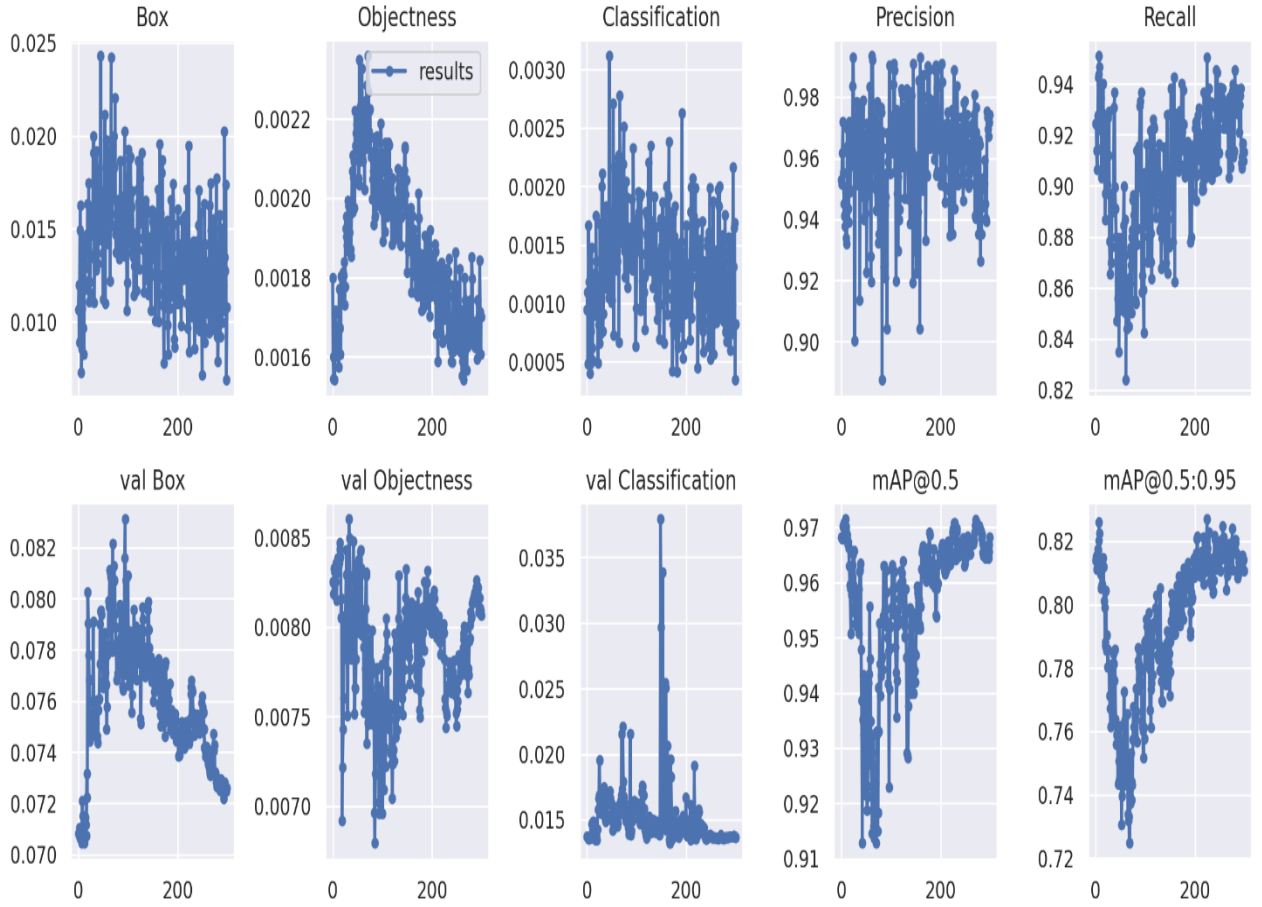
Parametre	Değer
Giriş boyutu	608 *608
Grup	64
Alt bölümler	16
İtme	0.9
Bozulma oranı	0.0005
Öğrenme oranı	0.001
Doyma	1.5
Maruziyet	1.5

Tablo 8-2 Önerilen model için harita sonuçlar YOLOv3

yinelemeler	Algılamalar	Ortalama hassasiyet				harita
		Yasaklayıcı	Tehlike	Zorunlu	Diğer	
1000	1095	%74.28 (TP = 94, AP = 43)	60.16 % (TP = 14, AP = 8)	%71,3 (TP = 5 , FP = 2)	64.00 % (TP = 15, AP = 2)	%67,44 (TP = 128, FP = 55)
2000	836	%96.38 (TP = 118, AP = 3)	%94.82 (TP = 31, FP = 3)	%98.82 (TP = 8, FP = 5)	%87.08 (TP = 29, AP = 6)	%94.22 (TP = 186, FP = 17)
3000	988	%98.01 (TP = 124, AP = 7)	%96.09 (TP = 35, FP = 3)	%100,00 (TP = 8, FP = 22)	%92.13 (TP = 33, AP = 7)	%96.56 (TP = 200, FP = 36)
4000	247	%99.86 (TP = 125, FP = 4)	%97.22 (TP = 35, FP = 0)	%92.76 (TP = 7, FP = 0)	%88.72 (TP = 30, FP = 0)	%94.64 (TP = 197, AP = 4)
5000	291	%99.84 (TP = 125, FP = 8)	%100,00 (TP = 35, FP = 0)	%87.50 (TP = 7, FP = 0)	%96.22 (TP = 31, AP = 1)	%95.89 (TP = 198, AP = 9)
6000	258	%99.98 (TP = 125, FP = 2)	%100,00 (TP = 36, AP = 0)	%93.06 (TP = 7, FP = 0)	%93.13 (TP = 31, AP = 1)	%96.54 (TP = 199, FP = 3)
7000	236	%100,00 (TP = 126, AP = 0)	%100,00 (TP = 36, AP = 0)	%100,00 (TP = 7, FP = 0)	%94.07 (TP = 34, AP = 1)	%98.52 (TP = 203, FP = 1)
8000	255	%99,99 (TP = 126, FP = 3)	%100,00 (TP = 36, AP = 0)	%100,00 (TP = 8, FP = 0)	%97.84 (TP = 34, AP = 2)	%99.46 (TP = 204, FP = 5)

Tablo 8-1, toplam yineleme sayısı olarak 8000, her 1000 yinelemede doğrulama görüntülerinde hesaplanan mAP sonuçları ile eğitim sırasındaki kaybı ve mAP'yi gösterir. Ayrıca, 8000 olan belirli yineleme noktasında eğitim sırasında bulunan en yüksek haritayı gösterir 99.46%.

Tablo 8.2, önerilen modelin tüm kriterlerde rakip modellerden nasıl daha iyi performans gösterdiğini ve %99,46'lık bir genel doğruluğa ulaştığını göstermektedir.



Şekil 8-1 Önerilen model için harita sonuçlar YOLOv7

Tablo 8-3 Önerilen model, mevcut kıyaslama çalışmaları ile karşılaştırılmıştır.

Yazarlar	veri kümesi	sınıflar	Algılama yöntemi	Ortalama ortalama hassasiyet (%)
Mannan ve ark. (2019)	GTSDb	3	Gauss karışımı	85.00
Gámez ve Ruichek (2020)	GTSDb/GTSRB	4	Maske R-CNN	97.35
Balado ve al. (2020)	GTSDb	5	etinaNet	89.7
Zhongqin ve Fuqiang (2022)	GTSDb	4	YOLOv4	85.59%
Riadh ve Mouna (2022)	GTSD	4	SqueezeNet	93.4
Önerilen model YOLOv3	GTSDb	4	YOLOv3	99.46
Önerilen model YOLOv7	GTSDb	4	YOLOv7	97.3

Bu çalışmada önerilen modelin sonuçları daha önce yapılan çalışmalarla karşılaştırdı. Tablo 8-3'te önerilen modeller son modellerden daha yüksek sonuçlar elde edildiğini göstermektedir.

9 SONUÇLAR

Bu çalışmada, bir dizi trafik işareti sınıfı için trafik işareti tanıma sorununu incelemektedir. Tespit açısından, GTSDDB, dört kategoride trafik işaretlerinin konumlarını tahmin etmek için derin bir sinirsel YOLOv3 ve YOLOv7 modelini eğitmek için kullandık. YOLO algoritması, otomotiv trafik işareti tanıma sistemlerinin kalitesini iyileştirebilir ve güvenilirliğini artırabilir ve bir kareyi işlemek için gereken süreyi azaltabilir, bu da video dizisinden Full HD 1920x1080 görüntülerde işaretlerin algılanmasını ve tanınmasını gerçekleştirme olasılığını getirir. Özellikle hız trafik araştırması için çıkarımlar, tanıma alanını ve sürücülerin sürüşü basitleştirmesine ve daha güvenli hale getirmesine yardımcı olmak için uygulanabilecek uygulamaları gösterir.

Deneyler, trafik işaretlerini tanımak ve sınıflandırmak için yalnızca dört sınıflı bir derin ağı tanımlamasının, birçok türden önceki yaklaşımlardan daha yüksek, %99.46'lık YOLOv3 ve %97.3'lık YOLOv7 yüksek bir mAP@0.5 doğruluğu ile sonuçlandığını göstermiştir.

Gelecekteki çalışmalarda, farklı hava ve aydınlatma koşullarını kapsayan daha büyük ve daha çeşitli veri kümeleri üzerinde eğitim ile algoritmanın hassasiyeti geliştirilebilir. Ayrıca, bu algoritma, yanlış algılamaların sayısını azaltmak için diğer sensör okumalarıyla füzyonda kullanılabilir.

Bölgelerin genellikle herhangi bir trafik işareti örneği içermediği gerçeği göz önünde bulundurularak çalışmamız daha da geliştirilebilir. Bir bölgede trafik işareti örneklerinin bulunmama olasılığını değerlendirmek faydalı olacaktır; boş bölgeler atılabilir, böylece zamandan ve donanım kaynaklarından tasarruf sağlanır.

Son olarak, algılama modelimizin performansını ve kararlılığını daha da optimize edeceğiz ve algılama görevini tamamlamak için onu diğer araştırma senaryolarına aktarmaya çalışacağız. Bu projenin başarısı umut vericidir ve kendi kendine sürüş araştırmalarında daha fazla ilerleme için etkileri vardır.

10 KAYNAKLAR

- [1] T. Yang, X. Long, A. K. Sangaiah, Z. Zheng, and C. Tong, “Deep detection network for real-life traffic sign in vehicular networks,” *Computer Networks*, vol. 136, pp. 95–104, 2018, doi: 10.1016/j.comnet.2018.02.026.
- [2] X. Zhang, Y. Zheng, Z. Zhao, Y. Liu, M. Blumenstein, and J. Li, “Deep learning detection of anomalous patterns from bus trajectories for traffic insight analysis,” *Knowl Based Syst*, vol. 217, p. 106833, 2021, doi: 10.1016/j.knosys.2021.106833.
- [3] Á. Arcos-García, J. A. Álvarez-García, and L. M. Soria-Morillo, “Deep neural network for traffic sign recognition systems: An analysis of spatial transformers and stochastic optimisation methods,” *Neural Networks*, vol. 99, pp. 158–165, 2018, doi: 10.1016/j.neunet.2018.01.005.
- [4] J. P. Vieira de Mello *et al.*, “Deep traffic light detection by overlaying synthetic context on arbitrary natural images,” *Computers and Graphics (Pergamon)*, vol. 94, pp. 76–86, 2021, doi: 10.1016/j.cag.2020.09.012.
- [5] J. G. Park and K. J. Kim, “Design of a visual perception model with edge-adaptive Gabor filter and support vector machine for traffic sign detection,” *Expert Syst Appl*, vol. 40, no. 9, pp. 3679–3687, 2013, doi: 10.1016/j.eswa.2012.12.072.
- [6] A. Campbell, A. Both, and Q. (Chayn) Sun, “Detecting and mapping traffic signs from Google Street View images using deep learning and GIS,” *Comput Environ Urban Syst*, vol. 77, no. February, p. 101350, 2019, doi: 10.1016/j.compenvurbysys.2019.101350.
- [7] E. Lee and D. Kim, “Accurate traffic light detection using deep neural network with focal regression loss,” *Image Vis Comput*, vol. 87, pp. 24–36, 2019, doi: 10.1016/j.imavis.2019.04.003.
- [8] C. Jang, C. Kim, D. Kim, M. Lee, and M. Sunwoo, “Multiple exposure images based traffic light recognition,” in *2014 IEEE Intelligent Vehicles Symposium Proceedings*, 2014, pp. 1313–1318. doi: 10.1109/IVS.2014.6856541.
- [9] S. Salti, A. Petrelli, F. Tombari, N. Fioraio, and L. Di Stefano, “Traffic sign detection via interest region extraction,” *Pattern Recognit*, vol. 48, no. 4, pp. 1039–1049, 2015, doi: 10.1016/j.patcog.2014.05.017.
- [10] X. Li, H. Ma, X. Wang, and X. Zhang, “Traffic Light Recognition for Complex Scene With Fusion Detections,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 1, pp. 199–208, 2018, doi: 10.1109/TITS.2017.2749971.
- [11] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You Only Look Once: Unified, Real-Time Object Detection,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 779–788. doi: 10.1109/CVPR.2016.91.
- [12] A. Fregin, J. Muller, U. Krebel, and K. Dietmayer, “The DriveU Traffic Light Dataset: Introduction and Comparison with Existing Datasets,” in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 3376–3383. doi: 10.1109/ICRA.2018.8460737.
- [13] Z. L. Sun, H. Wang, W. S. Lau, G. Seet, and D. Wang, “Application of BW-ELM model on traffic sign recognition,” *Neurocomputing*, vol. 128, pp. 153–159, 2014, doi: 10.1016/j.neucom.2012.11.057.

- [14] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, “Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition,” *Neural Networks*, vol. 32, pp. 323–332, 2012, doi: 10.1016/j.neunet.2012.02.016.
- [15] 2021 Karayolu Trafik Kaza İstatistikler, “Türkiye İstatistik kurum”, Accessed: Jul. 14, 2022. [Online]. Available: <https://data.tuik.gov.tr/Bulten/Index?p=Karayolu-Trafik-Kaza-Istatistikleri-2021-45658>
- [16] J. Zhang, Q. Su, C. Wang, and H. Gu, “Monocular 3D vehicle detection with multi-instance depth and geometry reasoning for autonomous driving,” *Neurocomputing*, vol. 403, pp. 182–192, 2020, doi: 10.1016/j.neucom.2020.03.076.
- [17] Y. Xue, Z. Ju, Y. Li, and W. Zhang, “MAF-YOLO: Multi-modal attention fusion based YOLO for pedestrian detection,” *Infrared Phys Technol*, vol. 118, no. November 2020, p. 103906, 2021, doi: 10.1016/j.infrared.2021.103906.
- [18] C. B. Rasmussen, K. Kirk, and T. B. Moeslund, “Anchor tuning in Faster R-CNN for measuring corn silage physical characteristics,” *Comput Electron Agric*, vol. 188, no. July, 2021, doi: 10.1016/j.compag.2021.106344.
- [19] S. Zhang, C. Chi, Y. Yao, Z. Lei, and S. Z. Li, “Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 9759–9768.
- [20] S. Ren, K. He, R. Girshick, and J. Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks,” *Adv Neural Inf Process Syst*, vol. 28, 2015.
- [21] Z. Tian, C. Shen, H. Chen, and T. He, “Fcos: Fully convolutional one-stage object detection,” in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 9627–9636.
- [22] W. Liu, S. Liao, W. Ren, W. Hu, and Y. Yu, “High-level semantic feature detection: A new perspective for pedestrian detection,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5187–5196.
- [23] X. Zeng, W. Ouyang, M. Wang, and X. Wang, “Deep learning of scene-specific classifier for pedestrian detection,” in *European Conference on Computer Vision*, 2014, pp. 472–487.
- [24] X. Wang, M. Wang, and W. Li, “Scene-specific pedestrian detection for static video surveillance,” *IEEE Trans Pattern Anal Mach Intell*, vol. 36, no. 2, pp. 361–374, 2013.
- [25] D. Vazquez, A. M. Lopez, J. Marin, D. Ponsa, and D. Geronimo, “Virtual and real world adaptation for pedestrian detection,” *IEEE Trans Pattern Anal Mach Intell*, vol. 36, no. 4, pp. 797–809, 2013.
- [26] L. Liu, W. Lin, L. Wu, Y. Yu, and M. Y. Yang, “Unsupervised deep domain adaptation for pedestrian detection,” in *European Conference on Computer Vision*, 2016, pp. 676–691.
- [27] Q. Ye *et al.*, “Self-learning scene-specific pedestrian detectors using a progressive latent model,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 509–518.
- [28] A. RoyChowdhury *et al.*, “Automatic adaptation of object detectors to new domains using self-training,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 780–790.

- [29] A. Khosravian, A. Amirkhani, H. Kashiani, and M. Masih-Tehrani, "Generalizing state-of-the-art object detectors for autonomous vehicles in unseen environments," *Expert Syst Appl*, vol. 183, no. May, p. 115417, 2021, doi: 10.1016/j.eswa.2021.115417.
- [30] S. Liu *et al.*, "A novel data augmentation scheme for pedestrian detection with attribute preserving GAN," *Neurocomputing*, vol. 401, pp. 123–132, 2020.
- [31] F. Lin, Y. Lai, L. Lin, and Y. Yuan, "A traffic sign recognition method based on deep visual feature," in *2016 Progress In Electromagnetics Research Symposium, PIERS 2016 - Proceedings*, Nov. 2016, pp. 2247–2250. doi: 10.1109/PIERS.2016.7734920.
- [32] Y. Zeng, X. Xu, D. Shen, Y. Fang, and Z. Xiao, "Traffic Sign Recognition Using Kernel Extreme Learning Machines with Deep Perceptual Features," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 6, pp. 1647–1653, Jun. 2017, doi: 10.1109/TITS.2016.2614916.
- [33] S. Jung, U. Lee, J. Jung, and D. H. Shim, "Real-time Traffic Sign Recognition system with deep convolutional neural network," in *2016 13th International Conference on Ubiquitous Robots and Ambient Intelligence, URAI 2016*, Oct. 2016, pp. 31–34. doi: 10.1109/URAI.2016.7734014.
- [34] Bangladesh University of Engineering and Technology, Bangladesh University of Engineering and Technology. Department of Electrical and Electronic Engineering, IEEE Region 10, Institute of Electrical and Electronics Engineers. Bangladesh Section, and Institute of Electrical and Electronics Engineers, *5th IEEE Region 10 Humanitarian Technology Conference 2017: Smart Technologies for Humanity: 21-23 December, 2017, BUET, Dhaka, Bangladesh*.
- [35] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2323, 1998, doi: 10.1109/5.726791.
- [36] P. Sermanet and Y. Lecun, "Traffic sign recognition with multi-scale convolutional networks," in *Proceedings of the International Joint Conference on Neural Networks*, 2011, pp. 2809–2813. doi: 10.1109/IJCNN.2011.6033589.
- [37] R. Qian, B. Zhang, Y. Yue, Z. Wang, and F. Coenen, "Robust Chinese traffic sign detection and recognition with deep convolutional neural network," in *Proceedings - International Conference on Natural Computation*, Jan. 2016, vol. 2016-January, pp. 791–796. doi: 10.1109/ICNC.2015.7378092.
- [38] Y. Zhu, M. Liao, M. Yang, and W. Liu, "Cascaded Segmentation-Detection Networks for Text-Based Traffic Sign Detection," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 1, pp. 209–219, Jan. 2018, doi: 10.1109/TITS.2017.2768827.
- [39] W. Hu, Q. Zhuo, C. Zhang, and J. Li, "Fast Branch Convolutional Neural Network for Traffic Sign Recognition," *IEEE Intelligent Transportation Systems Magazine*, vol. 9, no. 3, pp. 114–126, Sep. 2017, doi: 10.1109/MITS.2017.2709780.
- [40] L. G. M. perception of three-dimensional solids. Diss. M. I. of T. Roberts, "Machine perception of three-dimensional solids. Diss. Massachusetts Institute of Technology," *Diss. Massachusetts Institute of Technology*, 1961, Accessed: Jul. 23, 2022. [Online]. Available: <https://dspace.mit.edu/bitstream/handle/1721.1/11589/33959125-MIT.pdf>
- [41] S. A. Papert, "The Summer Vision Project," <https://dspace.mit.edu/handle/1721.1/6125>, 1966. Accessed: Jul. 23, 2022. [Online]. Available: <https://dspace.mit.edu/handle/1721.1/6125>
- [42] Richard Szeliski, *Computer Vision: Algorithms and Applications*. 2010.

- [43] D. H.-M. intelligence and undefined 1971, “Impossible object as nonsense sentences,” *ci.nii.ac.jp*, Accessed: Jul. 23, 2022. [Online]. Available: <https://ci.nii.ac.jp/naid/20000376094/>
- [44] D. Huttenlocher, S. U.-P. of the D. Image, and undefined 1987, “Object recognition using alignment,” *books.google.com*, Accessed: Jul. 23, 2022. [Online]. Available: [https://books.google.com/books?hl=tr&lr=&id=0fVRAAAAMAAJ&oi=fnd&pg=PA370&dq=Huttenlocher,+Daniel+P+\(1987\).+%E2%80%9CObject+recognition+using+alignment%E2%80%9D.+In:+ICCV+\(cit.+on+p.+28\).&ots=inapo0b4q2&sig=VbbQzhcBc9akY58yxxiJRdqfgEU](https://books.google.com/books?hl=tr&lr=&id=0fVRAAAAMAAJ&oi=fnd&pg=PA370&dq=Huttenlocher,+Daniel+P+(1987).+%E2%80%9CObject+recognition+using+alignment%E2%80%9D.+In:+ICCV+(cit.+on+p.+28).&ots=inapo0b4q2&sig=VbbQzhcBc9akY58yxxiJRdqfgEU)
- [45] I. W.-P. C. T. C. Society and undefined 1988, “Projective invariants of shapes,” *store.computer.org*, Accessed: Jul. 23, 2022. [Online]. Available: <https://store.computer.org/csdl/proceedings-article/cvpr/1988/00196251/12OmNxdDFEL>
- [46] M. Turk, A. P.-J. of cognitive neuroscience, and undefined 1991, “Eigenfaces for recognition,” *direct.mit.edu*, Accessed: Jul. 23, 2022. [Online]. Available: <https://direct.mit.edu/jocn/article-abstract/3/1/71/3025>
- [47] Ross Girshick and Microsoft Research, “Fast R-CNN,” 2015. <https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=7410526> (accessed Jul. 16, 2022).
- [48] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”, Accessed: Jul. 16, 2022. [Online]. Available: <https://github.com/>
- [49] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You Only Look Once: Unified, Real-Time Object Detection”, Accessed: Jul. 16, 2022. [Online]. Available: <http://pjreddie.com/yolo/>
- [50] W. Liu *et al.*, “SSD: Single shot multibox detector,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2016, vol. 9905 LNCS, pp. 21–37. doi: 10.1007/978-3-319-46448-0_2.
- [51] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, “Focal Loss for Dense Object Detection”.
- [52] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation”, Accessed: Jul. 16, 2022. [Online]. Available: <http://arxiv>.
- [53] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders, “Selective search for object recognition,” *Int J Comput Vis*, vol. 104, no. 2, pp. 154–171, Sep. 2013, doi: 10.1007/S11263-013-0620-5/FIGURES/9.
- [54] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. Lecun, “OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks,” 2014.
- [55] M. Riesenhuber and T. Poggio, “Models of object recognition,” *Nature Neuroscience* 2000 3:11, vol. 3, no. 11, pp. 1199–1204, 2000, doi: 10.1038/81479.
- [56] I. Goodfellow, Y. Bengio, and A. Courville, “Deep learning,” 2016, doi: 10.4258/hir.2016.22.4.351.
- [57] J. D. Kelleher, “Deep learning,” 2019, Accessed: Jul. 26, 2022. [Online]. Available: https://books.google.com/books?hl=tr&lr=&id=b06qDwAAQBAJ&oi=fnd&pg=PP9&dq=J.+D.+Kelleher,+Deep+learning,+MIT+press,+2019&ots=_oE_XLoXWM&sig=dYxo-UUc4kaWQUPi7-ZYFjAw0bQ

- [58] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *Bull Math Biophys*, vol. 5, no. 4, pp. 115–133, Dec. 1943, doi: 10.1007/BF02478259.
- [59] M. M. Poulton, "Computational neural networks for geophysical data processing," 2001, Accessed: Jul. 23, 2022. [Online]. Available: https://books.google.com/books?hl=tr&lr=&id=-6F9a5WV6dQC&oi=fnd&pg=PP1&dq=Poulton+MM.+A+Brief+History.+Oxford:+Elsevier+Science+%3B+2001&ots=XD0nKRcb69&sig=pWjmhR7Jsg_Qrf-30esPR0rkbgs
- [60] A. Newel, H. S.-C. of the ACM, and undefined 1976, "Computer science as empirical inquiry: Symbols and search," *contrib.andrew.cmu.edu*, Accessed: Jul. 23, 2022. [Online]. Available: <http://www.contrib.andrew.cmu.edu/~kk3n/80-300/newell-simon1976.pdf>
- [61] M. Bowling, J. Fürnkranz, T. Graepel, R. M.-M. learning, and undefined 2006, "Machine learning and games," *researchgate.net*, doi: 10.1007/s10994-006-8919-x.
- [62] H. Dreyfus, "Alchemy and artificial intelligence," 1965, Accessed: Jul. 23, 2022. [Online]. Available: <https://apps.dtic.mil/sti/citations/AD0625719>
- [63] J. Weizenbaum, "Computer power and human reason: From judgment to calculation.,," 1976, Accessed: Jul. 23, 2022. [Online]. Available: <https://psycnet.apa.org/record/1976-11270-000>
- [64] J. Weizenbaum, "ELIZA-A computer program for the study of natural language communication between man and machine," *Commun ACM*, vol. 9, no. 1, pp. 36–45, Jan. 1966, doi: 10.1145/365153.365168.
- [65] J. Feldman, M. Fanty, N. G.- Computer, and undefined 1988, "Computing with structured neural networks," *ieeexplore.ieee.org*, Accessed: Jul. 23, 2022. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/34/>
- [66] "Nöron ve Glia Hücreleri - Doktor Fizik." <https://www.doktorfizik.com/sinir-sistemi/norobilim/noron-ve-glia-hucreleri/> (accessed Sep. 02, 2022).
- [67] D. O. Hebb, "The Organization of Behavior," *The Organization of Behavior*, Apr. 2005, doi: 10.4324/9781410612403/ORGANIZATION-BEHAVIOR-HEBB.
- [68] J. Wu, "Introduction to Convolutional Neural Networks," 2017.
- [69] P. Rani, *Emotion Detection of Autistic Children Using Image Processing*.
- [70] S. Sharma, S. Sharma, A. A. data science, and undefined 2017, "Activation functions in neural networks," *ijeast.com*, vol. 4, pp. 310–316, 2020, Accessed: Jul. 26, 2022. [Online]. Available: <https://www.ijeast.com/papers/310-316,Tesma412,IJEAST.pdf>
- [71] F. Agostinelli, M. Hoffman, ... P. S. preprint arXiv, and undefined 2014, "Learning activation functions to improve deep neural networks," *arxiv.org*, Accessed: Jul. 26, 2022. [Online]. Available: <https://arxiv.org/abs/1412.6830>
- [72] S. Qian, H. Liu, C. Liu, S. Wu, H. S. W.- Neurocomputing, and undefined 2018, "Adaptive activation functions in convolutional neural networks," *Elsevier*, Accessed: Jul. 26, 2022. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0925231217311980>
- [73] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation", Accessed: Jul. 16, 2022. [Online]. Available: <http://arxiv>.

- [74] J. R. R. Uijlings, · K E A van de Sande, · T Gevers, and · A W M Smeulders, “Selective Search for Object Recognition,” *Int J Comput Vis*, vol. 104, pp. 154–171, 2013, doi: 10.1007/s11263-013-0620-5.
- [75] X. Z. Kaiming He, “Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition,” 2015. <https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=7005506> (accessed Jul. 16, 2022).
- [76] “ImageNet Classification with Deep Convolutional Neural Networks.” <https://proceedings.neurips.cc/paper/2012/hash/c399862d3b9d6b76c8436e924a68c45b-Abstract.html> (accessed Jul. 16, 2022).
- [77] F. Chollet, “Xception: Deep learning with depthwise separable convolutions,” *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, vol. 2017-January, pp. 1800–1807, Nov. 2017, doi: 10.1109/CVPR.2017.195.
- [78] Candida Castro and Time Horberry, “The Human Factors of Transport Signs,” 2005.
- [79] H. Buiter and P.-E. Staal, “City lights: Regulated streets and the evolution of traffic lights in the Netherlands, 1920-1940.”
- [80] C. Mcshane, “THE ORIGINS AND GLOBALIZATION OF TRAFFIC CONTROL SIGNALS,” 1993. [Online]. Available: <http://www.whc.neu.edu/his->
- [81] “YOLOv3 Algorithm with additional convolutional neural network trained for traffic sign recognition; YOLOv3 Algorithm with additional convolutional neural network trained for traffic sign recognition,” 2020.

ÖZGEÇMİŞ

kişisel Bilgiler		
Adı soyadı	Mohamed Taghi	
doğum Yeri		
doğum tarihi		
Uyruğu		
telefon		
E-Posta Adresi		
Web Adresi		

	Eğitim Bilgileri
	lisans
üniversite	
Fakülte	
Bölümü	
Mezuniyet Yılı	

Yüksek Lisans	
üniversite	Bandırma Onyedü Eylül
Enstitü Adı	Fen Bilimleri Enstitüsü
Anabilim Dalı	Akıllı Ulaşım Sistemleri ve Teknolojileri Anabilim Dalı
programı	Akıllı Ulaşım Sistemleri ve Teknolojileri Tezli Yüksek Lisans Programı

Makale ve Bildiriler
Taghi M, Özer İ. (2022) An Enhanced Real-Time Detection And Classification Of Traffic Signs Using The YOLO Version 3 Framework. International Joint Conference on Engineering, Science and Artificial Intelligence (ICESAI), 194-200