

**T.C.
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**



**GÖRÜNTÜ İŞLEME VE DERİN ÖĞRENME YÖNTEMLERİ İLE
ÇELİK YÜZEY KUSURLARININ TESPİTİ VE
SINIFLANDIRILMASI**

Ahmet AYDIN

Yüksek Lisans Tezi

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

YAZILIM Bilim Dalı

HAZİRAN 2022

T.C.
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

Bilgisayar Mühendisliği Anabilim Dalı

Yüksek Lisans Tezi

GÖRÜNTÜ İŞLEME VE DERİN ÖĞRENME YÖNTEMLERİ İLE ÇELİK YÜZEY KUSURLARININ TESPİTİ VE SINIFLANDIRILMASI

Tez Yazarı

Ahmet AYDIN

Danışman

Doç. Dr. İlhan AYDIN

HAZİRAN 2022

ELAZIĞ

T.C.
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

Bilgisayar Mühendisliği Anabilim Dalı

Yüksek Lisans Tezi

Başlığı:	Görüntü İşleme ve Derin Öğrenme Yöntemleri ile Çelik Yüzey Kusurlarının Tespiti ve Sınıflandırılması
Yazarı:	Ahmet AYDIN
İlk Teslim Tarihi:	30.05.2022
Savunma Tarihi:	30.06.2022

TEZ ONAYI

Fırat Üniversitesi Fen Bilimleri Enstitüsü tez yazım kurallarına göre hazırlanan bu tez aşağıda imzaları bulunan jüri üyeleri tarafından değerlendirilmiş ve akademik dinleyicilere açık yapılan savunma sonucunda OYBİRLİĞİ ile kabul edilmiştir.

Danışman:	Doç. Dr. İlhan AYDIN Fırat Üniversitesi Mühendislik Fakültesi	İmza Onayladım
Başkan:	Dr. Öğr. Üyesi Muhammed YILDIRIM Turgut Özal Üniversitesi, Mühendislik ve Doğa Bilimleri Fakültesi	Onayladım
Üye:	Dr. Öğr. Üyesi Gülşah KARADUMAN Fırat Üniversitesi Mühendislik Fakültesi	Onayladım

Bu tez, Enstitü Yönetim Kurulunun/...../20..... tarihli toplantısında tescillenmiştir.

İmza

Prof. Dr. Kürşat Esat ALYAMAÇ
Enstitü Müdürü

BEYAN

Fırat Üniversitesi Fen Bilimleri Enstitüsü tez yazım kurallarına uygun olarak hazırladığım “Görüntü İşleme ve Derin Öğrenme Yöntemleri ile Çelik Yüzey Kusurlarının Tespiti ve Sınıflandırılması” Başlıklı Yüksek Lisans Tezimin içindeki bütün bilgilerin doğru olduğunu, bilgilerin üretilmesi ve sunulmasında bilimsel etik kurallarına uygun davrandığımı, kullandığım bütün kaynakları atıf yaparak belirttiğimi, maddi ve manevi desteği olan tüm kurum/kuruluş ve kişileri belirttiğimi, burada sunduğum veri ve bilgileri unvan almak amacıyla daha önce hiçbir şekilde kullanmadığımı beyan ederim.

30.06.2022

Ahmet AYDIN



ÖNSÖZ

Metal yüzeylerdeki kusurların otomatik olarak denetlenmesi ve tespiti, ürün üreticileri ve bu ürünlerin kullanıcıları için önemlidir. Endüstriyel ürünler ulaşım, enerji üretimi, gıda üretimi gibi çeşitli alanlarda kullanıldığı için günümüzde bu ürünlerin denetlenmesi zor bir problemdir. Bu problem özellikle ağır sanayi sanayii düşünüldüğünde büyük ölçekte malzemelerin üretiminde veya onarılmasında geleneksel yöntemler kullanılmaktadır. Gelişen bir takım yöntemler sayesinde lazer sistemler ve görüntü işleme ile arıza tespitler gerçekleştirilirken yine manuel olarak kontrol edilebilme ihtiyacı gerekmektedir.

Bazı sektörler kamuya hizmet verdiğinden ötürü arıza tespit sistemlerinin gelişimi bir noktada güvenlik anlamıyla da kesişmektedir. Bu alanda yapılacak çalışmalarda ise haliyle güvenilecek çıktılara ihtiyaç duyulmaktadır. Günümüzde yapay zekâ ile birlikte bazı alanlarda oldukça gelişim göstermektedir. Bu alanların başında Makine öğrenmesi ve Derin öğrenme gelmektedir. Bu iki alanla birlikte sınıflandırma ve tespit sistemleri de oldukça iyi bir konuma geldiği gibi gelecekte de hayli iyi bir noktaya doğru gideceği yapılan araştırmalarla desteklenmektedir. Bu çalışma arıza tespit sistemlerine derin öğrenme yaklaşımı ile gerek ağır sanayi gerek kamu güvenliği açısından üretilecek olan çıktıların güven altına alınması bakımından önemli olmakla birlikte arıza tespit sistemlerine yeni bir perspektiften bakmayı hedeflemektedir. Tespit amacıyla endüstriyel bir ürün olan yassı çelik yüzey kusurlar ele alınmaktadır. Kusurlar üzerinde sınıflandırma ve kusur tespitinin türünün belirlenmesini görüntü tabanlı bir metot geliştirmek amacıyla bir takım görüntü işleme teknikleri de kullanılarak tespiti ayrıca popüler olarak kullanılan derin öğrenme modellerini de kullanarak sistemin eğitilmesini ve arıza tespit sistemlerine otonom bir yapının güven verecek sonuçları üretebilmesi hedeflenmektedir.

Çalışmalarım süresince zaman gözetmeksizin vakit ayırıp sorularımı cevaplayan, tüm sürece karşı bilimsel bakış açımı genişleten danışman hocam Sayın Doç. Dr. İlhan AYDIN'a çok teşekkür ederim.

Son olarak, hiçbir zaman hakkını ödeyemeyeceğim her zaman ve her koşulda yanımda olan beni büyüten ve bugünlere gelmemde en büyük emeğe sahip olan çok değerli aileme sonsuz şükranlarımı sunarım.

Bu tezimi, aileme ithaf ediyorum.

Bu tez çalışması, TUBİTAK tarafından 120E097 protokol numaralı proje ile desteklenmiştir.

Ahmet AYDIN

ELAZIĞ, 2022

İÇİNDEKİLER

Sayfa

ÖNSÖZ.....	iv
İÇİNDEKİLER	v
ÖZET	vii
ABSTRACT	viii
ŞEKİLLER LİSTESİ	ix
TABLolar LİSTESİ	xi
EKLER LİSTESİ	HATA! YER İŞARETİ TANIMLANMAMIŞ.
SİMGELER	xii
KISALTMALAR	xiii
1. GİRİŞ	1
2. LİTERATÜR TARAMASI	3
2.1. Sınıflandırma Tabanlı Yüzey Kusuru Çalışmaları.....	3
2.2. Tespit Tabanlı Yüzey Kusuru Çalışmaları.....	5
2.3. Segmentasyon Tabanlı Yüzey Kusuru Çalışmaları	7
3. DERİN ÖĞRENME	9
3.1. Evrişimsel Sinir Ağı (ESA)	9
3.1.1. Giriş Katmanı	10
3.1.2. Konvolüsyon Katmanı.....	11
3.1.3. ReLU katmanı	11
3.1.4. Havuzlama Katmanı	12
3.1.5. Tam Bağlı Katman	13
3.1.6. Dropout Katmanı.....	14
3.1.7. Normalizasyon (Ölçeklendirme) Katmanı	15
3.1.8. Sınıflandırma (Softmax) Katmanı	15
3.2. Derin Öğrenme Parametreleri.....	15
3.2.1. Veri Kümesi	16
3.2.2. Aktivasyon Fonksiyonları	16
3.2.3. ReLU Fonksiyonu	16
3.2.4. Softmax Fonksiyonu	17
3.2.5. Tanh Fonksiyonu.....	17
3.2.6. Sigmoid fonksiyonu	18
3.2.7. Leaky ReLU fonksiyonu	18
3.2.8. Öğrenme Katsayısı	19
3.2.9. Parti Boyu.....	20
3.2.10.Katman Sayısı ve Gizli Katmanlar	21
3.2.11.Eğitimde Tur Sayısı.....	21
3.2.12.Kayıp Fonksiyonu	21
3.3. Derin Öğrenmede Transfer Öğrenme	22
3.4. Derin Öğrenmede Kullanılan Kütüphaneler	23
3.4.1. TensorFlow	23
3.4.2. Keras	24
3.4.3. Caffe.....	24

3.4.4. Torch	24
3.4.5. Theano.....	24
3.4.6. MatConvNet.....	24
3.5. Derin Öğrenme ile Nesne Sınıflandırma	25
3.5.1. LeNet.....	25
3.5.2. MobileNet	25
3.5.3. GoogleNet	28
3.5.4. ResNet.....	28
3.6. Derin Öğrenme ile Nesne Tespiti	29
3.6.1. Regions with Convolution Neural Network	30
3.6.2. Single Shot Detector.....	31
3.6.3. You Only Look Once	32
4. MATERYAL METOT	34
4.1. Veri Seti.....	34
4.1.1. NEU Veri Kümesi	34
4.1.2. Sınıflandırma İçin Tarafımızca Hazırlanmış Veri Seti (FUB).....	36
4.1.3. Veri Seti Ön İşleme	37
4.2. Nesne Sınıflandırma	40
4.2.1. VGG-16.....	40
4.2.2. Transfer Öğrenme Tabanlı VGG-16	41
4.2.3. İnce Ayarlama Tabanlı VGG-16	42
4.3. Nesne Tespiti	44
4.3.1. YOLO –V4.....	44
4.3.2. YOLO –V5.....	45
4.3.3. YOLOX.....	46
4.3.4. YOLOR.....	47
5. BULGULAR VE TARTIŞMA	49
5.1. Sınıflandırma İçin Elde Edilen Bulgular	49
5.1.1. NEU Veri Setinden Elde Edilen Bulgular	49
5.1.2. FUB Veri Setinden Elde Edilen Bulgular	50
5.1.3. Sınıflandırma Çalışması İçin Genel Bakış	50
5.2. Nesne Tespiti İçin Elde Edilen Bulgular	51
6. SONUÇLAR.....	56
KAYNAKLAR.....	58
ÖZGEÇMİŞ	

ÖZET

Görüntü İşleme ve Derin Öğrenme Yöntemleri ile Çelik Yüzey Kusurlarının Tespiti ve Sınıflandırılması

Ahmet AYDIN

Yüksek Lisans Tezi

FIRAT ÜNİVERSİTESİ
Fen Bilimleri Enstitüsü

Bilgisayar Mühendisliği Anabilim Dalı

Haziran 2022, Sayfa: xiii + 63

Metaller hayatımızda vazgeçilmez konuma gelmiş unsurlardır. Metaller gündelik hayatta sanayi, ulaşım, teknoloji, inşaat sektörü, enerji sektörü gibi birçok alanda kullanılmaktadır. Metallerin bu kadar çok kullanılmasının sebebi ise dayanıklı ve uzun ömürlü olmasıdır. Ancak, metallerde belirli bir müddet kullanıldıktan sonra kısmi olarak veya tamamen kullanılmaz hale gelebilir. Bu durumda, hemen hemen her alanda kullanılan bu maddenin büyük sorunlara sebep olacağı gerçeğini meydana çıkarmaktadır. Oluşabilecek sorunların önüne geçmek için insanlar bu metalleri yüzyıllarca kendileri kontrol ederek gerekli gördükleri yerlere müdahale ettiler. Bu işlem hem zaman alıcı, pahalı hem de insandan insana değişen bakış açısı yönünden tehlike arz etmekteydi. Bu konuyu ele alan bilim insanları görsel denetim başlığı altında metal yüzeylerde oluşabilecek sorunlara bilimsel yaklaşımlarla hızlı, ucuz ve değişmeyen bir bakış açısıyla çözümler üretmeye başladılar. Görsel denetim için bilim insanları görüntü işleme yöntemlerine başvurmuşlardır. Burada bilim insanları girdap yöntemi, lazerle teşhis gibi yöntemleri kullanmışlardır. Gelişen yöntemlerle derin öğrenme yöntemleri ortaya koyulmuştur. Bu çalışmada derin öğrenme yöntemleri kullanılarak metal yüzeyler üzerindeki arızaları teşhis edebilmek için arıza sınıflandırma ve arıza tespit çalışmaları yapılmıştır. Sınıflandırma için VGG-16 modeli temel alınarak yapılan ince ayarlamalar sonucunda yüksek başarı oranlarıyla arıza sınıflandırılması sağlanmıştır. Nesne tespit çalışmasında ise farklı YOLO modellerin metal yüzeyler karşısındaki karşılaştırılması sağlanmış olup en doğru ve en hızlı sonuç veren modelin YOLO-V5 olduğu tespit edilmiştir.

Anahtar Kelimeler: Derin öğrenme, Metal yüzeyler, Kusur Tespit, YOLO, İnce Ayarlama, VGG-16

ABSTRACT

Detection and Classification of Steel Surface Defects by Image Processing and Deep Learning Methods

Ahmet AYDIN

Master's Thesis

FIRAT UNIVERSITY
Graduate School of Natural and Applied Sciences
Department of Computer Engineering

June 2022, Pages: xiii + 63

Metals are elements that have become indispensable in our lives. Metals are used in many areas such as industry, transportation, technology, construction sector, energy sector in daily life. The reason why metals are used so much is that they are durable and long-lasting. However, after a certain period of use on metals, it may become partially or completely unusable. In this case, it reveals the fact that this substance, which is used in almost every field, will cause great problems. In order to prevent possible problems, people have controlled these metals for centuries and intervened where they deem necessary. This process was both time-consuming, expensive and dangerous in terms of perspective that differs from person to person. Scientists dealing with this issue started to produce solutions to the problems that may occur on metal surfaces under the title of visual inspection, with scientific approaches, with a fast, cheap and unchanging point of view. For visual inspection, scientists have applied to image processing methods. Here, scientists used methods such as vortex method and laser diagnosis. Deep learning methods have been put forward with developing methods. In this study, fault classification and fault detection studies were carried out in order to diagnose faults on metal surfaces using deep learning methods. As a result of fine adjustments made on the basis of the VGG-16 model for classification, fault classification has been achieved with high success rates. In the object detection study, different YOLO models were compared against metal surfaces and it was determined that the model that gave the most accurate and fastest results was the YOLO-V5.

Keywords: Deep learning, YOLO, Metal surfaces, Defect detection, Fine Tuning, VGG-16

ŞEKİLLER LİSTESİ

	Sayfa
Şekil 3.1. Klasik Evrişimsel Sinir Ağı.....	10
Şekil 3.2. Evrişim işlemi	11
Şekil 3.3. ReLU davranış fonksiyonu.....	12
Şekil 3.4. Maksimum ve ortalama havuzlama işlemi	13
Şekil 3.5. Tam bağlı katman yapısı	14
Şekil 3.6. Normal sinir ağı ve dropout uygulanmış sinir ağı	15
Şekil 3.7. Softmax aktivasyon fonksiyonu	17
Şekil 3.8. Tanh fonksiyonuna ait davranış grafiği.....	18
Şekil 3.9. Standart sigmoid aktivasyon grafiği.....	18
Şekil 3.10. Leaky ReLU aktivasyon fonksiyonuna ait davranış grafiği	19
Şekil 3.11. SGD algoritmasının çalışma mantığı.....	20
Şekil 3.12. a) ince ayarlamalı transfer öğrenme b) Standart özellik çıkarıcı olarak önceden eğitilmiş modeller	23
Şekil 3.13. LeNet Ağ Mimarisi	25
Şekil 3.14. MobileNet modeline ait Mantıksal Yaklaşım.....	27
Şekil 3.15. GoogleNet Ağ Mimarisi.....	28
Şekil 3.16. ResNet Ağ Mimarisi.....	29
Şekil 3.17. R-CNN çalışma modeli	31
Şekil 3.18. SSD Algoritması	32
Şekil 3.19. YOLO ağ mimarisi.....	33
Şekil 4.1. Sınıflara göre etiket dağılımları.....	34
Şekil 4.2. Veri kümesinden etiketlenmiş veri örnekleri.....	35
Şekil 4.3. Oluşturulan veri kümelerine bazı örnekler	36
Şekil 4.4. FUB veri kümesine ait hasar sınıfları	37
Şekil 4.5. OTSU ile odak görüntüsünün elde edilmesi.....	38
Şekil 4.6. Orijinal ve sıkıştırılmış görüntü	38
Şekil 4.7. Derin Öğrenme Modellerinde Sıkça Kullanılan Görsel Boyutları	39
Şekil 4.8. Görüntü artırımı örneği	39
Şekil 4.9. VGG-16 Modeli	41
Şekil 4.10. Transfer Öğrenme Tabanlı VGG-16	42
Şekil 4.11. VGG-16 Modeli üzerinde İnce Ayarlama	43

Şekil 4.13. YOLOv4 modelinin COCO veri setindeki diğer modellerle karşılaştırılması.....	44
Şekil 4.14. YOLOv5 mimarisi.....	46
Şekil 4.15. YOLOX mimarisi.....	47
Şekil 4.16. COCO veri setinde YOLOR modelinin karşılaştırılması	48
Şekil 5.1. COCO veri setinde YOLOR modelinin karşılaştırılması.	52
Şekil 5.2. Yolov5'in kusur tespit sonuçları	53



TABLÖLAR LİSTESİ

	Sayfa
Tablo 4.1 Önerilen Yönteme Ait Ağ Yapısı	43
Tablo 5.1. Parti Boyutunun NEU Veri Seti İçin Modellerin Doğruluğu Üzerine Etkisi	49
Tablo 5.2. Parti Boyutunun FUB Veri Seti İçin Modellerin Doğruluğu Üzerine Etkisi	50
Tablo 5.3. Veri Kümelerinin Sınıflandırma Sonuçları	51
Tablo 5.4. En iyi sonucu algılanan veri kümesi	54
Tablo 5.5. Karşılaştırma Tablosu	55



SİMGELER

$b_x b_y$	: Çerçevenin merkez koordinatları
$b_h b_w$	: Çerçevenin yükseklik ve genişliği
c	: Tespit yapılmak istenen sınıf
e^{z_i}	: Giriş vektörü için standart üstel fonksiyon
e^{z_j}	: Çıkış vektörü
e	: Euler Sayısı
e^{-x}	: Çıkış fonksiyonu için standart üstel fonksiyonu
$f_{i,j,k}$	: 3 boyutlu (i, j, k indis) özellik matrisi
K	: Sınıf sayısı
p_c	: Sınıf olasılığı
$x_{i,j}$	: Görüntü piksel ve (i, j) indisleri
y	: Sınırlayıcı Kutu
$\vec{z}$	: Giriş vektörü
w_k^T	: Konvolüsyonel ağırlık
σ	: Softmax

KISALTMALAR

ASFF	: Uyarlamalı uzamsal özellik birleştirme
ATSS	: Uyarlanabilir eğitim örnek seçimi
BVLC	: Berkeley Vision and Learning Center
CAE	: Konvolüsyonel otomatik kodlayıcı
CBL	: Konvolüsyon + BatchNormalization + ReLU
ConCNN	: Eşzamanlı evrişimli sinir ağı
CPU:	: Merkezi işlemci birimi
CVAE	: Evrişimsel değişken otomatik kodlayıcı
DAGM	: Örüntü tanıma için alman çalışma grubu
DESA (DCNN)	: Derin konvolüsyonel sinir ağı
DFEM	: Dilate feature enhanced model
EDDN	: Uçtan uça hata tespit ağı
ESA	: Evrişimsel sinir ağı
EMRA	: Çok ölçekli ters dikkat ağı
FC	: Tam bağlantılı katmanlar
FUB	: Fırat Üniversitesi karşılaştırma veri kümesi
HDC	: Hiyerarşik ayrılmış evrişim ağı
ILSRVRC	: Imagenet Large Scale Visual Recognition Challenge
IOU	: Intersection over Union
MMGCN	: Çoklu mikrogram grafik evrişim ağı
MSF	: Çok ölçekli füzyon
NEU	: Northeast Üniversitesi veri kümesi
NWDAN	: Çok ölçekli çekişmeli ve ağırlık gradyan etki alanı uyarlamalı ağ
R-CNN	: Alan tabanlı evrişimsel sinir ağı
ReLU	: Düzleştirilmiş doğrusal birim
RGB	: Kırmızı-Yeşil-Mavi
SCRA	: Uzaysal kanal alanı ters dikkat ağı
SegNet	: Segmentasyon ağı
SGAN	: Yarı denetimli üretken düşman ağı
SSD	: Tek atış çoklu kutu detektörü
SPP	: Mekânsal piramit havuzu
SVM	: Destek vektör makineleri
PSN	: Parametrik sigmoid norm
VGG	: Visual geometric group
YOLO	: You only look once
YOLOR	: You Only Learn One Representation
YOLOX	: YOLO'yu aşmak
YSA	: Yapay sinir ağı

1. GİRİŞ

Son zamanlarda bilgisayarlı görme ve sınıflandırma alanlarında birçok probleme uygulanan Derin sinir ağlarını ile özellik öğrenme problemlerin çözümünde başarılı olduğunu kanıtlanmıştır. Derin öğrenme tekniklerinin sağlamış olduğu en büyük avantaj manuel olarak özellik çıkarımını kendisi yapmasıdır. Derin öğrenme modeli elindeki verilerden doğrudan olarak probleme özgü nitelikleri otomatik olarak çıkarır ve bunları kendini eğitmek için kullanır [1]. Derin öğrenmenin temelinde bulunan evrimsel sinir ağları da bu şekilde çalışmaktadır.

Evrimsel sinir ağları görüntülerin sınıflandırılması, görüntülerin içerisinde bulunan nesnelerin tespiti ve görüntülerdeki nesnelerin segmentasyonu gibi çeşitli alanlarda kullanılan modellerin temelini oluşturmaktadır. Derin öğrenmenin hızla yaygınlaşan kullanım alanlarını göz önüne alındığında endüstriden gıda sektörüne, tekstil sektöründen güvenlik alanına kadar her alanda mutlak varlığından bahsedilebilir [2].

Endüstrinin gelişmesi ve seri üretime geçilmesi ile birlikte üretimde malzemenin özelliklerine, dayanıklılığına, işlenme yöntemlerine veya kullanıma bağlı faktörlerden dolayı metalik yüzeyler üzerinde kusurlar meydana gelebilir. Yüzey kusurları, metal ürünlerin kalitesini etkileyen en önemli faktörlerden biridir [3]. Ayrıca endüstriyel ürünler yüksek maliyetli ürünlerdir. Oluşacak kusurlar zamanında tespit edilip onarılması durumunda can ve mal kaybına sebebiyet verebilir. Bu yüzden endüstriyel ürünlerin sürekli denetlenerek oluşabilecek kusurların erken aşamada belirlenmesi kaliteli ürünlerin ortaya çıkmasını sağlar.

Görsel hasar tespiti ya da bir başka deyişle görsel arıza tespiti, sağlam bir nesne üzerinde olağanın dışında gelişmiş kusurları tespit etmek amacıyla kullanılan bilgisayarlı bir yöntemdir [4]. Görsel kontrolün başlaması insanların seri üretime geçmeleri ile birlikte bir ihtiyaç haline geldi. Makinelerin, insanların yerini aldığı bu alanda, daha hızlı ve doğru değerlendirmeleri etkili sebepler arasındadır. İnsanların kalite ve kontrol işlemlerini yapması üretilen her ürünü gözden geçirmesi zahmetli ve zaman alan bir işlemdir. Görsel kontrol, hem üretim görevlerinde hem de üretim dışı görevlerde yaygın olarak kullanılır. Üretim görevlerinde ki amaç, bir sonraki montajda veya müşteriye teslimatı esnasındaki kurulumunda bir ürünün hatasız olduğunu doğrulamaktır. Üretim dışı görevlerde ise amaç, hedefte belirtilen özelliklerin mevcut olup olmadığını belirlemektir. Görsel denetimin kullanıldığı birçok alan, potansiyel olarak yüksek denetim hataları ve bu hatadan kaynaklanan maliyetlerin (yaralanma, ölüm, pahalı donanım kaybı, hurdaya ayrılmış eşyalar) fazla olduğu alanlardır [5].

Görsel kontrol ve sürekli yapılması gereken, dikkat gerektiren zorlu bir süreçtir. Denetim için yorgunluk, hastalık, iş gerginliği ve ruh hali gibi insansal faktörlerin etkisi düşünüldüğünde insanların sürekli olarak tekrarlayan bu işleri yapması sağlıklı bir işlem değildir [6]. Ayrıca insandan insana göre değişen nitel değerlendirmelerde söz konusudur. Bir denetmene göre bakım

gerektiren endüstriyel ürünler, bir başka denetmene göre bakım gerektirmeyebilir. Bu sebepten dolayı denetlemeler nicel yani ölçülebilir ifadelerle gerçekleştirilmelidir.

Derin öğrenme modelleri geliştirmeden önce araştırmacılar görüntü işleme kullanarak bu nicel ifadeleri manuel olarak arıza ve hata tespit veya arıza ve hata sınıflandırma algoritmalarına uyguluyorlardı [7]. Bir veri kümesi için geliştirilen geleneksel görüntü işleme tekniklerini aynı türden farklı bir veri kümesinde kullanırken aynı başarı elde edilmeyebilir çünkü bu yöntemler çevresel etkenlerden çok fazla etkilenirler. Görüntünün kontrastı, ışıklandırması, gürültüsü ve arka planı gibi birçok faktörün tekrar gözden geçirilerek eşik ayarlarının yapılması ve algoritmanın geliştirilmesi gerekmektedir. Bu nedenle geleneksel modeller yeterli başarıyı elde edemez ve gerçek zamanlı tespitler için kullanılması uygun değildir. Geleneksel yöntemlerin aksine derin öğrenmenin gelişmesiyle kusur tespitinde yeni bir döneme girilmiştir.

Derin öğrenme kullanılarak yapılan denetlemelerde minimum ölçüde insan gücüne ihtiyaç duyulmaktadır ve iyi tasarlanmış bir derin öğrenme modeli yeterli kaynakla eğitilirse benzeri birçok amaca hizmet edebilir. Derin öğrenme modelleri, kendilerine sunulan veriler ışığında bir eğitim gerçekleştirmektedir. Bu eğitimin sonucunda kaynak olarak verilen verileri en iyi şekilde temsil eden özellikleri otonom olarak çıkarmaktadır. Diğer algoritmalara göre daha hızlı ve maliyet açısından daha tasarruflu olan derin öğrenme modelleri ayrıca daha başarılı sonuçlar ortaya koymaktadır [8].

Bu çalışmada metal yüzeylerin incelenmesinde hatalarının sınıflandırılması ve hataların tespiti ele alınmıştır. Çalışma gerçekleştirilirken son zamanlarda oldukça popüler olan derin öğrenme modelleri kullanılmış ve geliştirilmiştir. Hangi şartlar altında iyi sonuçlar verdikleri de ayrıca tartışılmıştır. Değişik parametreler ve görüntü iyileştirme teknikleri karşısında nasıl tepki verdikleri bu çalışma kapsamında incelenmiş olup sonuçlar paylaşılmıştır. Çalışma metal yüzeyler üzerinde yapılacak denetimlerin hızlandırılması ve olumsuz sonuçlar meydana getirmemesi amacıyla gerçekleştirilmiştir.

Tezin yapısal olarak, ilk bölümde metal yüzeylerin önemini vurgulayan girişle başlamaktadır. İkinci bölümde bu alanda yapılan çalışmaları içeren literatür taraması sunulmuştur. Üçüncü bölümde tezde kullanılan yöntemlerle ilgili ön bilgi verilmek amacıyla derin öğrenmeden bahsedilmiştir. Dördüncü bölümde tezde kullanılan ve geliştirilen yöntemler hakkında bilgiler paylaşılmaktadır. Beşinci bölümde kullanılan yöntemlerden elde edilen bulgular paylaşılmış olup, ilgili çalışmalarla karşılaştırılması sağlanmıştır. Altıncı ve son bölümde sonuçlar paylaşılmaktadır.

2. LİTERATÜR TARAMASI

Derin öğrenme geliştirildiğinden bu yana çok hızlı gelişme göstermektedir. Bu gelişmeler derin öğrenmenin birçok alana uygulanabilmesinden kaynaklanmaktadır. Uygulama alanlarının fazla olması derin öğrenme algoritmalarının daha çok gelişmesine buda derin öğrenmenin popülaritesinin daha da artmasına katkı sağlamıştır. Bu sistem derin öğrenme için bir ekosistem haline gelmiştir. Aşağıda yüzey hasar tespit ile ilgili çalışmalardan bahsedilecektir.

2.1. Sınıflandırma Tabanlı Yüzey Kusuru Çalışmaları

Liu ve ark. [9], 2020 yılında yaptıkları çalışmada gerçek zamanlı hata sınıflandırma uygulamaları için hafif ağırlıklı ve kurulumu kolay, farklı görüntü ölçeklerine sahip yeni bir Eşzamanlı Evrişimli Sinir Ağı (ConCNN) önermişlerdir. ConCNN'nin performansını değerlendirmek için deneylerinde NEU-CLS (Northeast Üniversitesi Sınıflandırma) veri seti kullanmışlardır. Sonuçlarına göre, ConCNN'nin çelik yüzey kusur sınıflandırması için doğruluk ve gecikme süresi dikkate alındığında çok iyi performans gösterdiğini göstermektedir. %98,89 sınıflandırma başarı oranı ve yaklaşık 5,58 ms eğitim gecikme maliyeti bulunmaktadır.

Faghih-Roohi ve ark. [10], öz nitelik öğrenimi için uygun bir teknik olarak gördükleri evrişimli sinir ağlarını kullanmayı önermişlerdir. Yaptıkları Derin evrişimli sinir ağı modelini, farklı boyutlar ve aktivasyon fonksiyonları ile karakterize edilen farklı ağ mimarilerinin sonuçları ile karşılaştırmışlardır. DCNN (Deep Convolutional Neural Network) uygulamalarından kendi oluşturdukları veri tabanı üzerinde yaklaşık %92 sonuç elde etmişlerdir.

Cheon ve ark. [11], çelik yüzey kusur tespiti için transfer öğrenme tabanlı U-Net (TLU-Net) çerçevesini kullanmayı önermişlerdir. Temel olarak bir U-Net mimarisi kullanan araştırmacılar, ResNet ve DenseNet kodlayıcıları üzerinde çalışmışlardır. Ağların performansını karşılaştıran araştırmacılar transfer öğrenme tabanlı yaklaşımın %26 daha iyi sonuç verdiğini öne sürmüşlerdir.

Di ve ark. [12], metal yüzeylerin sınıflandırılması için yarı denetimli yeni bir yöntem önermişlerdir. Önerdikleri yöntem, evrişimsel otomatik kodlayıcı (Convolutional Autoencoder - CAE) ve yarı-denetimli çekişmeli üretici ağlar (Semi Generative Adversarial Networks - SGAN) tabanlı olduğu için CAE-SGAN olarak adlandırmışlardır. Araştırmacılar yaptıkları bu çalışma ile %16'lık geliştirme yakalamışlardır.

Ye ve ark. [13], 2018 yılında kusur özelliklerini ayırt etmek amacıyla evrişimsel sinir ağı tabanlı bir derin öğrenme modeli önermişlerdir. Eğitimin temel amacıyla kusurların karakteristik özelliklerinin referans alınması vardır. Araştırmacıların bulgularına göre, çok büyük ölçekteki verilerin hata sınıflandırması %96 doğruluk oranıyla kısa zamanlarda bulunabiliyordu.

Fu ve ark. [14], 2019 yılında önerdikleri yöntemde, omurga mimarisi olarak önceden eğitilmiş SqueezeNet'i kullanmışlardır. Çeşitli çevresel etkilere maruz kalan, çelik yüzey kusurlarının görüntü artırımı kullanarak oluşturulan test veri kümesin oluşturmuşlardır. Ayrıca, önerilen hafif ESA model, NVIDIA TITAN X Grafik İşlem Birimi ile çalışan bir bilgisayarda 100 fps'nin üzerinde çalışan gerçek zamanlı çevrimiçi denetim işlevini gerçekleştireceğini savunmuşlardır.

Wang ve ark.[15], 2020 yılında gerçekleştirdikleri çalışmada, yüzey kusur sınıflandırması için çoklu mikroskopik grafik evrişim ağı (MMGCN) olarak adlandırılan yeni bir grafik tabanlı yarı denetimli yöntem önermişlerdir. Çalışmalarını test etmek için NEU veri kümesini kullanan araştırmacılar, yarı denetimli deneylerden aldıkları sonuçlara göre 98,06 başarı oranı elde etmişlerdir.

Konovalenko ve ark. [16], 2020 yılında çelik yüzeylerin sınıflandırılması için Resnet50 sinir ağına dayalı bir mimari geliştirmişlerdir. Üç çeşit sınıflandırma verisi bulunan veri kümelerinden %96,91 sonuç elde etmişlerdir. Ayrıca Resnet50'nin çelik yüzeylerin sınıflandırılması için mükemmel bir tanımlayıcı, yüksek doğrulukla ve hızlı bir sinir ağı olduğunu ileri sürmüşlerdir.

Yun ve ark.[17], 2020 yılında yayımlanan çalışmalarında, metal yüzeylerin kusurlarını sınıflandırmak için görüntü tabanlı derin sinir ağı modeli önermişlerdir. Modellerinde evrimsel değişken otomatik kodlayıcı (CVAE) kullanarak, modeller üzerindeki etkilerini incelemişlerdir. CVAE tabanlı veri oluşturma teknolojisi, sınıflandırma modelini eğitmek için yeterli hata verisi üretir. Bir bakıma CVAE kullanarak görüntü artırımı yapmışlardır. Yaptıkları bu çalışma sonucunda modellerin, CVAE ile kullanımı ile CVAE'siz kullanımlarını karşılaştırmışlardır. Deneysel çalışmalarındaki elde ettikleri sonuçlar CVAE ile geliştirdikleri modellerin çok daha yüksek başarı oranı ile çalıştığını sergilemektedir.

Kim ve ark. [18], çelik yüzeylerin sınıflandırılması için yaptıkları çalışmada çok fazla görüntüye ihtiyaç duyulmadan sınıflandırma görevini başarılı bir şekilde nasıl yapacaklarını araştırmışlardır. Yaptıkları çalışmada Siamese Neural Network (Siyam Sinir Ağı) kullanarak birkaç adımlı bir sınıflandırma yöntemi önermişlerdir.

Wang ve ark. [19], NEU veri seti üzerinde yaptıkları Derin öğrenme ESA modelinde teorik ve uygulamalı bir çalıştırma gerçekleştirmiş olup %98,6 sonuç elde etmişlerdir. Çalışmada kullanılan model bir örneği 60 ms sürede sınıflandırmaktadır.

He ve ark. [20], çalışmalarında, genel endüstriyel kusur tespiti için regresyon ve sınıflandırma tabanlı bir çerçeve önermişlerdir. Dört parçadan oluşan bu çerçeve: derin regresyon tabanlı algılama modeli, piksel düzeyinde yanlış pozitif indirgeme, bağlantılı bileşen analizi ve kusur tipi sınıflandırması için derin ağ. Algılama modelini eğitmek için, yüksek performanslı bir derin ağ yapısı ve veri ek açıklamalarından kusur şiddeti bilgisini yakalamak için etiket verileri oluşturmak için bir algoritma oluşturup, önermişlerdir. Yöntemi, AigleRN ve DAGM2007 olmak

üzere iki genel kıyaslama veri kümesinde test etmişlerdir. Yöntemin algılama doğruluğu ve verimliliği açısından performanslı olabileceği ileri sürülmüştür.

Neuhaser ve ark. [21], yüzey kusur sınıflandırması ve tespiti için bir yaklaşım önermektedirler; bu sayede, basit bir kamera, üretim sırasında ihraç edilen edilmiş profilleri kaydeder ve bir sinir ağı mimarisi, kusursuz yüzeyler ile çeşitli yaygın kusurlar içeren yüzeyler sınıflandırmaktadır. Çalışmalarında, yapay zekâdan elde edilen yöntemlerin, bir sinir ağını eğitmek için çok sınırlı veri seti boyutları gibi yaygın endüstri kısıtlamaları altında bile kalite kontrol gibi endüstriyel uygulamalarla oldukça uyumlu olduğunu göstermeye çalışmışlardır. Veri büyütme ve transfer öğrenimi kullanan araştırmacılar, Sınıflandırmalarında 0,98 doğruluk ve algılama ayarında 0,47 ortalama hassasiyet elde etmişlerdir.

Aydin ve ark. [22], 2021 yılında yaptıkları çalışmada ray kusurlarını sınıflandırmak için iki derin öğrenme modelinin özellikleri birleştirilerek bir model ortaya koymuşlardır. Kullandıkları derin öğrenme modelleri olan SqueezeNet ve MobileNetV2'yi seçme nedenleri ise diğer modellerden küçük ve daha hızlı olmalarıdır. Hassasiyeti az olan iki modeli birleştirerek, yüksek hassaslıkta hızlı güvenilir ve küçük bir model yakalamayı hedefleyen araştırmacılar, Bir takım ön işlemden sonra özellik seçme yöntemini kullanarak son katmanda da Destek Vektör Makinelerini (SVM) kullanarak bu modeli geliştirmeyi başarmışlardır. Deneysel sonuçlarında düşük kontrast altında iyi sonuçlar elde ettiklerini ileri sürmüşlerdir.

Aydin ve ark. [23], 2021 yılında yaptıkları çalışmada NEU veri kümesini ve transfer öğrenme kullanarak metal yüzeyleri VGG-16 modelinde bazı katmanlarını dondurarak sınıflandırmışlardır. Yaptıkları çalışmada ayrıca farklı parametreler altında modellerinin başarılarını test ederek parametrelerinin ağırları üzerine etkisini ölçmüşlerdir. Kendi veri tabanları üzerinde daha sonra modellerini çalıştırarak karşılaştırmalar paylaşmışlardır. Modellerinde her iki veri kümesi üzerinde yüksek başarı elde ettiklerini sunmuşlardır.

2.2. Tespit Tabanlı Yüzey Kusuru Çalışmaları

Yanan ve ark. [24], Tren rayları üzerindeki hasarları tespit etmek amacıyla YOLOv3 algoritmasını kullanan bir sinir ağı modeli önermişlerdir. Yaptıkları test sonuçları yaklaşık 0.15 s içinde sonuç vermekte ve tanıma oranının %97'nin üzerine çıkabileceğini ileri sürmüştür. Araştırmacılar bu çalışmayı beş farklı sınıf için uygulamışlardır.

Lv ve ark. [25], 2020 yılındaki çalışmalarında yeni bir karşılaştırma veri kümesi ortaya koymuşlardır. Ayrıca Single Shot MultiBox Detector (SSD) algoritması tabanında geliştirdikleri uçtan uca tespit ağı (end-to-end defect detection network - EDDN) modelini önermişlerdir. Bu modeli kullanırken veri artırımı yöntemine başvuran araştırmacılar, veri kümelerinin ve oluşturdukları ağlarının metalik kusur tespiti için doğruluk gereksinimlerini karşılayabildiğini ileri sürmüşlerdir.

Xu ve ark. [26], Metal yüzey hata tespitinde Sadece YOLO modeline dayalı geliştirilmiş bir metal yüzey kusur tespit teknolojisi sunmuşlardır. Darknet-53'teki 11. katmanın sığ özellikleri, YOLOv3'ün ağ yapısı temelinde yeni bir ölçek özellik katmanı oluşturmak için sinir ağının derin özellikleri ile birleştirmişlerdir. Bu işlemi yapma amaçları küçük kusurların daha fazla özelliklerini ortaya çıkarmaktır. Ayrıca modellerinde K-Means algoritmasını da kullanarak ilk küme merkezine olan duyarlılığı azaltmayı hedeflemişlerdir. Araştırmalarında modifiye edilmiş YOLO modelinin, YOLOv3 modelinden daha iyi sonuç verdiğini ileri sürmüşlerdir.

Li ve ark. [27], çelik yüzeylerin ata tespiti için YOLO modelini geliştirerek bir model önermişlerdir. Bu modele göre evrimsel katmanlarla çalışmışlardır. 27 evrimsel katman kullanan araştırmacılar, yüksek oranlarda başarılı sonuçlar elde etmişlerdir.

Chen ve ark. [28], derin öğrenmeye dayalı hedef tespit yöntemi, elektrik güç ekipmanının çatlak, pas ve yağ sızıntısı gibi yüzey kusurlarını elde etmeye çalışmışlardır. Çalışmalarında, YOLO-V4 omurga ağı geliştirilmiş olup, pozitif ve negatif numune arasındaki dengesizlik nedeniyle düşük algılama doğruluğunu desteklemek için odak kaybı işlevi indüklemişlerdir. Çalışmalarının ön işlemeli ve ön işlemez olarak değerlerinden araştırmacılar iki yöntemde de uyarlanmış YOLOv4 modelinin normal YOLOv4 modeline ve Faster R-CNN modeline göre daha iyi sonuçlar verdiğini ileri sürmüşlerdir.

Li ve ark. [29], bir derin öğrenme yöntemi kullanarak yüzey kusurlarının gerçek zamanlı ve doğru tespitini sağlamayı amaçlamışlardır. Bu amaçla SSD tabanlı olmak üzere, ESA tabanlı MobileNet ile MobileNetSSD'yi birleştirdiler. Daha sonra ortaya çıkan modelin parametrelerini optimize eden araştırmacılar, çalışmalarının geleneksel makine öğrenme yöntemlerinden daha doğru ve daha hızlı olduğunu ortaya koyduğunu ileri sürmüşlerdir.

Lin ve ark. [30], çelik yüzeyindeki kusurları otomatik olarak tespit etmek için bir derin öğrenme yöntemi önermişlerdir. Önerilen sistemi iki kısımda inceleyen araştırmacılar, ilk kısımda olası kusurları SSD modelini kullanmışlardır. Ardından SSD'den elde ettikleri sonuçları ResNet mimarisinde kullandılar. Elde ettikleri deneysel sonuçlarda çelik yüzey kusurlarında önerilen model ile daha yüksek hassasiyet ve geri çağırma elde edebileceklerini ortaya sürmüşlerdir.

Zhao ve ark. [31], NEU veri kümesi üzerinde gerçekleştirdikleri çalışmada geleneksel Faster R-CNN mimarisini yeniden ele alıp yani bu modeli omurga olarak kullanarak kendi modellerini geliştirmişlerdir. Hedef görsellerin küçük özelliklerine dayanarak ağı çok ölçekli füzyonla eğitmişlerdir. Elde ettikleri deneysel sonuçta ortalama hassasiyeti 0.752 olarak elde etmişlerdir. Normal Faster R-CNN modeline 0.128 oranında yapmayı başaran araştırmacılar çelik yüzeylerin otomatik hasar tespiti için bir referans modeli olabileceğini ileri sürmüşlerdir.

Urbonas ve ark. [32], yüzey hasarlarını tespit etmek amacıyla daha hızlı bölge tabanlı bir evrimsel sinir ağı (Faster R-CNN) benimsediler. Sonuçları iyileştirmek için transfer öğrenimi için önceden eğitilmiş AlexNet, VGG16, BNInception ve ResNet152 sinir ağı modellerini

kullanılmıştır. En iyi geliştirmeyi ve ortalama hassasiyet değerini, %80,6'lık değerle ResNet152 sinir ağı modelinde yakalamışlardır.

Cheng ve ark. [33], çalışmalarında çelik yüzey kusur tespiti için yeni bir derin sinir ağı (DNN), farklı kanal dikkati ve uyarlanabilir uzamsal özellik füzyonu (DEA_RetinaNet) ile RetinaNet önermişlerdir. İlk olarak, DEA_RetinaNet'in algılama doğruluğunu geliştirmek için bir diferansiyel evrim arama tabanlı bağlantı optimizasyonu gerçekleştiren araştırmacılar, ikinci olarak bilgi kaybını minimize etmek amacıyla DEA_RetinaNet'e yeni bir kanal dikkat mekanizması entegre etmişlerdir. Son olarak, uyarlamalı uzamsal özellik birleştirme (ASFF) modülü, evrişimli çekirdekler tarafından çıkarılan sık ve derin özelliklerin etkin bir biçimde birleştirilmesi amacıyla kullanılır. Araştırmacılar yaptıkları bu çalışmada %78.25 başarı elde etmişlerdir.

2.3. Segmentasyon Tabanlı Yüzey Kusuru Çalışmaları

Tsai ve ark. [34], yüzey segmentasyonu için iki aşamalı bir derin öğrenme şeması önermişlerdir. Derin öğrenme şemasının ilk aşaması, bir görüntüdeki kusurlu pikselleri otomatik olarak sentezlemek ve açıklama eklemek için iki döngü tutarlı rakip ağ (CycleGAN) modeli kullandılar. Sentezledikleri kusur görüntüleri ve CycleGAN modellerinden bunlara karşılık gelen açıklamalı sonuçları, daha sonra U-Net semantik ağını eğitmek için giriş-çıkış çiftleri olarak kullandılar. Deneysel sonuçlarında, önerilen derin öğrenme şemasının çeşitli dokulu ve desenli yüzeylerde kusur tespiti için uygulanabileceğini ve yüksek tespit doğruluğu ile sonuçlandığını ileri sürdüler.

Cao ve ark. [35], kusur tespitinin doğruluğunu artırmak için, yüzey kusur tespiti için derin özellik füzyonuna dayalı piksel düzeyinde bir segmentasyon ağı önermişlerdir. Araştırmacılar, ResNet50 algoritmasını kullanarak düşük, orta ve yüksek seviyeli özellikleri çıkarır. Uygulamanın ikinci aşamasında tüm seviyelerde ki özellikler birleştirilerek çok seviyeli özellik toplama modülü oluşturulur. Bu sayede araştırmacıların hedeflemiş olduğu gibi tüm seviyelerde detaylı özellik içeriği oluşmaktadır. Araştırmacılar modellerini üç farklı veri kümesine uygulamışlardır.

Usamentiaga ve ark. [36], yaptıkları çalışmada segmentasyon yapmak için var olan modelleri veri kümeslerine uygulayarak bir sonuç elde etmişlerdir. Sonuçlara göre en iyi modeller U-Net ve YOLOv5 olarak belirlemişler ancak YOLOv5 modeli U-Net modeline göre hem daha hızlı hem daha başarılı olduğu sonucunu ileri sürmüşlerdir.

Yang ve ark. [37], kodlayıcı-kod çözücü ağ yapısıyla, yeni bir tahribatsız kusur tespit ağı, yani NDD-Net, piksel tabanlı olan bu yaklaşımlarını uçtan uca bir tahribatsız kusur bölümleme şeması oluşturmak için önermişlerdir.

Liang ve ark. [38], makalelerinde demiryolunun yüzey kusurlarını tespit etmek için SegNet mimarisinin derin bir evrişim sinir ağlarını önermişlerdir. Ray yüzey kusur görüntüleri sistem tarafından elde edilir ve ray yüzey kusurlarını tespit etmek için 120 ray eğitim görüntüsünü,

tasarlanan 59 katmanlı bir eğitim ağında eğitmişlerdir. Yazarlar çalışmanın yüksek verimlilikte çalıştığını iddia etmişlerdir.

Kang ve ark. [39], demiryollarındaki metal yüzeylerin sorununa çözüm bulabilmek için makalelerinde, derin bir evrişimli sinir ağı (DESA) kullanan yeni bir yalıtkan yüzey kusur tespit sistemi önermişlerdir. Araştırmacıların önerdikleri sistem iki aşamadan oluşmaktadır. Sistemin ilk aşamasın, ana bileşenlerini sınırlamak için daha hızlı bir R-CNN ağı benimsemişlerdir. İkinci aşamasında, sınıflandırma puanı ve anomali puanı, derin bir sınıflandırıcı belirleme işlemini gerçekleştirmişlerdir. Kusur durumu, sınıflandırma puanı ve anormallik puanı analiz ederek modellerini oluşturmuşlardır.

Zhou ve ark. [40], ellerindeki büyük ölçekli veri setiyle, U-Net tarafından kullanılan yapıya dayalı olarak, ağ tasarımını yeniden tasarlamışlardır. PSD U-Net için model, alıcı alanı genişletme öncülü altında resim ayrıntılarını en büyük ölçüde korumak için orijinal evrişim katmanı yerine delik evrişim ve HDC tasarımını kullanmaktadır. Tanıma oranını daha da iyileştirmek için parametrik sigmoid norm (PSN) katmanı, özellik çıkarma ağının üzerine eklemişlerdir. Makalelerinde son olarak, U-Net, Mask R-CNN ve PSD U-Net olmak üzere üç ağı dayanmaktadır. Araştırmacılar, PSD U-Net mimarisinin iyi sonuçlara sahip olduğunu ileri sürmektedirler.

Lin ve ark. [41], geliştirdikleri model ile yüzeyler üzerindeki hataları bölütlemek için uç ve çok ölçekli ters dikkat ağını (EMRA-Net) geliştirmişlerdir. EMRA-Net içerisinde özellik çıkarıcı ve özellik birleştirme modülerini barındırmaktadır. Öznitelik çıkarma sürecinde, ön eğitim omurgası ResNet 34, yeni MSF modülü ve yenilikçi piramit kenar modülü aracılığıyla küresel dinamik evrişim öznitelikleri, küresel dinamik çok ölçekli füzyon (MSF) özelliği ve yerel piramit kenar özelliği elde etmektedirler. Özellik birleştirme sürecinde, bu özellikler yeni bir kendi kendine öğrenen ölçek modülü ve yeni bir uzaysal kanal alanı ters dikkat (SCRA) modülü ile adım adım harmanlanmaktadır. Araştırmacılar, bu modelden yaklaşık %95 başarı elde etmişlerdir.

3. DERİN ÖĞRENME

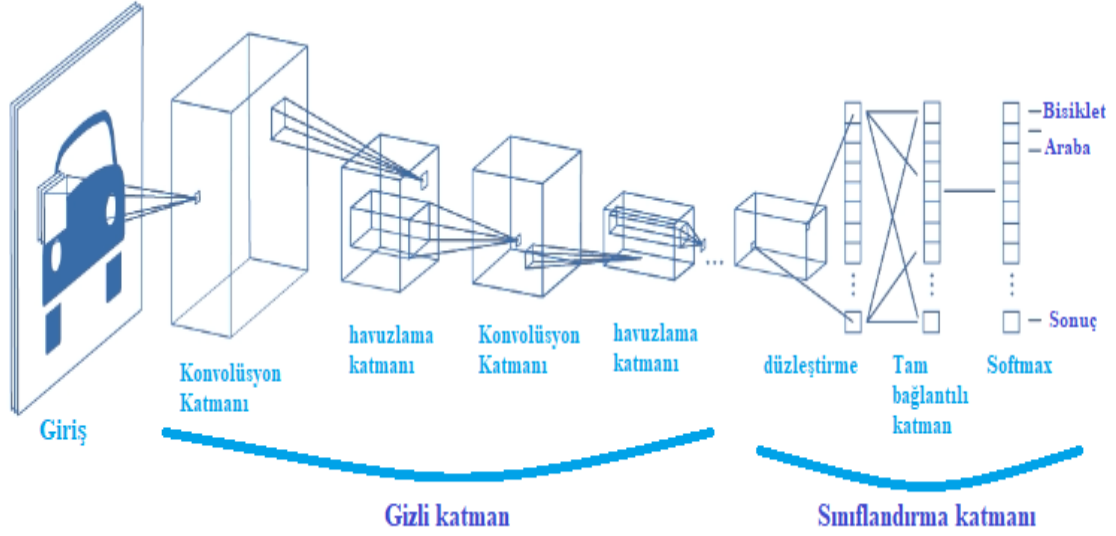
Bilgisayarlı görü alanında desteklemek olarak son yıllarda geliştirilen en önemli teknolojilerden biri derin öğrenmedir. Makine öğrenmesine gelişimine destek olmak adına tespit sistemlerinin gelişimi ve tahminine katkıda önemli ölçüde katkı sunmaktadır [42]. Derin öğrenme, özellikleri daha önceden hazırlanmış denklemlerle çalışmak yerine, verilerle ilgili temel parametreler üzerinde çalışır, birçok işleme katmanından bu verileri işleyerek kalıpları tanımlar bilgisayarın kendi kendisine eğitmesine olanak sağlar. Derin öğrenme teknikleri günümüzde, tanımlama, sınıflandırma, tek kelime ile sonuç çıkarma, algılama ve segmentasyon yeteneklerine sahiptir. Derin öğrenme algoritmalarının özyinelemeli doğası vardır [43]. Katman sayısına bağlı olarak karmaşıklığı artan ve ağırları eğitmek için gereken büyük miktarlardaki veri kümeleri nedeniyle daha fazla hesaplama gücüne ihtiyaç duyarlar. Derin öğrenmenin en büyük avantajlarından biri özellik haritaları çıkarabilmeleridir [44]. Derin öğrenme modelleri evrışimsel sinir ağıları (ESA) üzerine kuruludurlar. Derin öğrenmeyi daha iyi anlamlandırabilmek için ESA ağlarına değinilecektir.

3.1. Evrişimsel Sinir Ağı (ESA)

Klasik ESA art arda gelen evrişimsel tabakalardan ve ortalama tabakalarından oluşur [45]. Konvolüsyonel katmanlar, Relu, sigmoid, tanh vb. doğrusal konvolüsyon fonksiyonlarının uygulanması ile özellik haritası oluşturur. Özellik haritası denklem 3.1'deki formül kullanılarak hesaplanabilir.

$$f_{i,j,k} = \max(w_k^T x_{i,j}, 0) \quad (3.1)$$

Evrışimsel sinir ağıları sınıflandırma ve tespit işlemlerinde oldukça faydalı bir yaklaşım türü olarak neredeyse haricinde bir yöntem kullanılmamaktadır. Bilgisayar gücünden faydalanan evrişimsel sinir ağıları yüksek olasılıkları kolayca ve bilgisayar hızlarına bağlı olmakla birlikte hızlı bir şekilde hesaplayabildiklerinden dolayı insani özelliklere göre avantaj sağlamaktadırlar. Desteklenen çalışmalar görüntüler üzerine yapılmakta iken günümüzde birçok alanda kullanılmaktadır. Denklem 3.1' de görüldüğü gibi konvolüsyon denklemi sunulmaktadır. Denklem 3.1'de özellik çıkarmak üzere w ağırlıkları, $x_{i,j}$ filtre matrisi ve indislerini, 0 ise başlangıç parametresini göstermektedir. Aşağıda Şekil 3.1 üzerinde klasik ESA modeli görülmektedir.



Şekil 3.1. Klasik Evrışimsel Sinir Ağı

Şekil 3.1’de gösterildiği gibi pikseller kenar kombinasyonundan oluşan desenleri, bu desenler bir araya gelerek nesne parçalarını ve bunlarda bir araya gelerek nesneleri oluşturmaktadır [46]. Bir girişe dayalı olan konvolüsyon işlemi, görüntüler için giriş verisini matris şeklinde işlemektedir, bu matris üzerinde belirli filtreler uygulanmasıyla nesneye ait özelliklerin belirlenmesi ve bu özelliklere göre kategorize edilmesiyle evrışimsel sinir ağı işlemi tamamlanmış olmaktadır. Giriş verisi üzerine uygulanan filtre her adımda indisleri eşleşen değerlerin çarpılır ve tüm değerlerin toplamı çıkış matrisinin değeri olarak kaydedilir. Evrışimsel sinir ağları bu işlemleri içeriğinde bulunan katmanlar aracılığıyla gerçekleştirmektedir. ESA katmanları alt başlıklar halinde değinilerek açıklanacaktır.

3.1.1. Giriş Katmanı

Bu katman ESA mimarisinin ilk katmanını oluşturmaktadır. Giriş verisi modelin başarımı üzerinde önemli bir etkidir. Görüntü boyutu veya çözünürlüğünün yüksek seçilmesi test ve eğitim süresini uzatabilir, bu durum bellek açlığına da sebep olabilir. Buna bağlı olarak modelin başarı oranını da yükseltebilir. Giriş verisinin boyutunu düşürmek ise süreyi azaltabileceği gibi modelin başarısını da olumsuz etkileyebilir. İdeal giriş verisinin boyutunun belirlenmesi model için faydalı olacaktır.

3.1.2. Konvolüsyon Katmanı

Bu katman ESA'nın temelini oluşturmaktadır. Evrişim katmanı aynı zamanda, dönüşüm katmanı olarak tanımlanır. Dönüşümün amacı belirlenen filtrenin görüntü üzerinde dolaştırılarak yeni bir matris oluşturmaktır [47]. Her evrişim işlemi bir önceki katmandan gelen çıktı verisine uygulanmaktadır. Özellik haritası da bu aşamada oluşur. Eğitim sırasında belirlenen filtrelerin katsayıları her adımda yinelenir. Örneğin, 224x224 renkli bir görüntü düşünecek olursak RGB kanalları ile gerçek boyutumuz 224x224x3 olacaktır. 5x5 evrişim filtresi giriş görüntüsü üzerinde piksel, piksel kaydırarak tüm pikselleri gezecek şekilde, filtre katsayıları ile piksellerde ki renk değerleri çarpılıp bunların toplamı hesaplandığında aktivasyon haritası oluşur. Şekil 3.2 de evrişim işlemi gösterilmektedir.

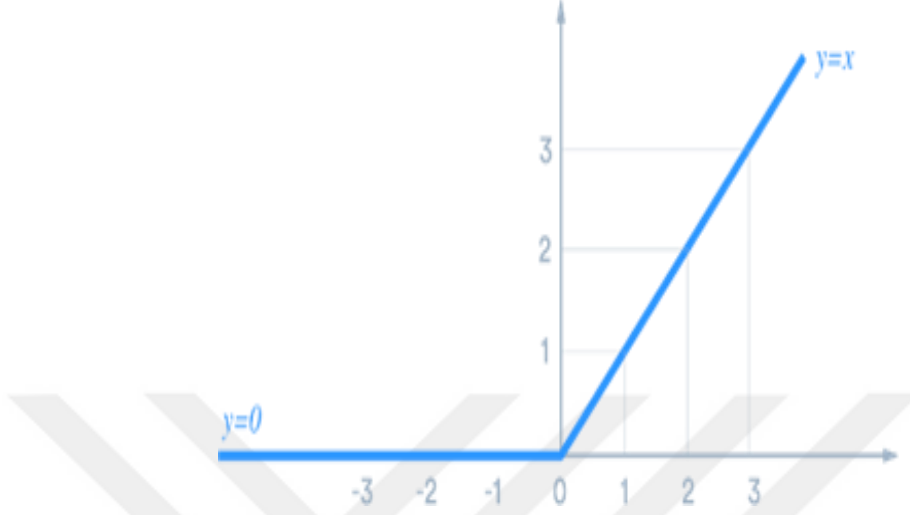


Şekil 3.2. Evrişim işlemi

3.1.3. ReLU katmanı

Bu katman aktivasyon katmanı olarak bilinmektedir. Düzleştirilmiş doğrusal birim katmanı (ReLU) ESA'dan farklı bir bileşen değildir. Konvolüsyon ve ReLU adımlarını ayrı ayrı değerlendiren birçok yazarlar mevcuttur, fakat şu an için kabul gören düşünce ikisinin birlikte tek bir adım oluşturduğudur. ReLU fonksiyonu bir görüntüye baktınız da doğrusal olmayan bölgelerin (kenar, piksel ve renk geçişleri) doğrultulmasında kullanılmaktadır [48]. Negatif görüntü değerlerini sıfıra eşitler ve pozitif görüntü değerlerini doğrusal olarak değerlendirir. Denklem 3.2 üzerinde ReLU katmanının matematiksel ifadesi gösterilmektedir. Ayrıca derin öğrenme ağlarında en fazla kullanılan aktivasyon çeşidi ReLU'dur [49]. ReLU'ya ait aktivasyon grafiği Şekil 3.3'te gösterilmektedir.

$$f(x) = \begin{cases} 0 & \text{eğer } x < 0 \\ x & \text{eğer } x > 0 \end{cases} \quad (3.2)$$

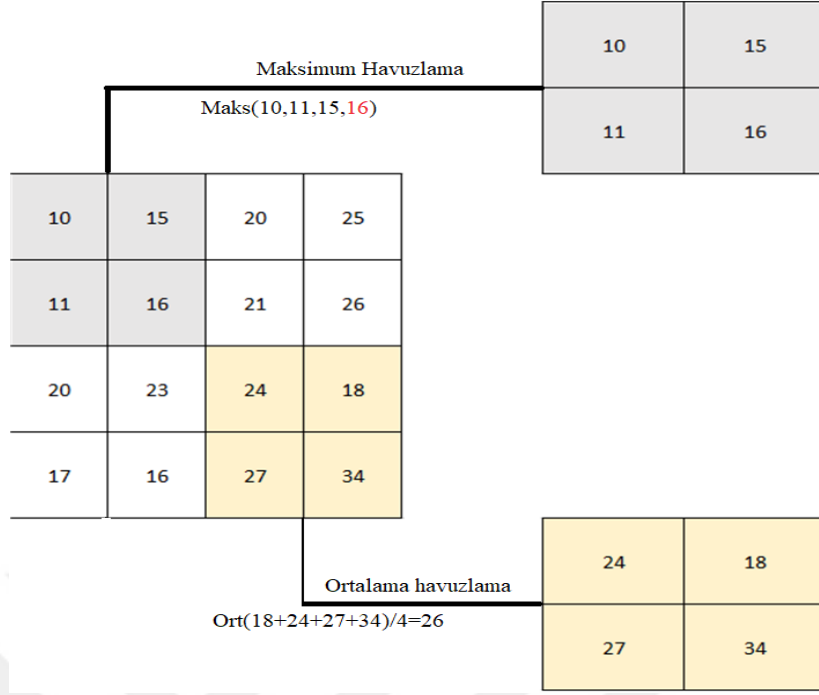


Şekil 3.3. ReLU davranış fonksiyonu

3.1.4. Havuzlama Katmanı

Havuzlama katmanı genellikle konvolüsyon katmanından sonra gelmektedir [50]. Havuzlama katmanındaki amaç kendinden sonraki katman için iş yükünü azaltmaktır. Bu katmanda yapılan işlem ile bilgi kaybı oluşur, bu bilgi kaybının amacı ise bir işlem gerektiren adımda hesaplama maliyetini azaltmaktır. Bu adımın uygulanmasında ki bir diğer amaç ise veri kaybı sağlayarak sistemin ezberlenmesine engel olmaktır.

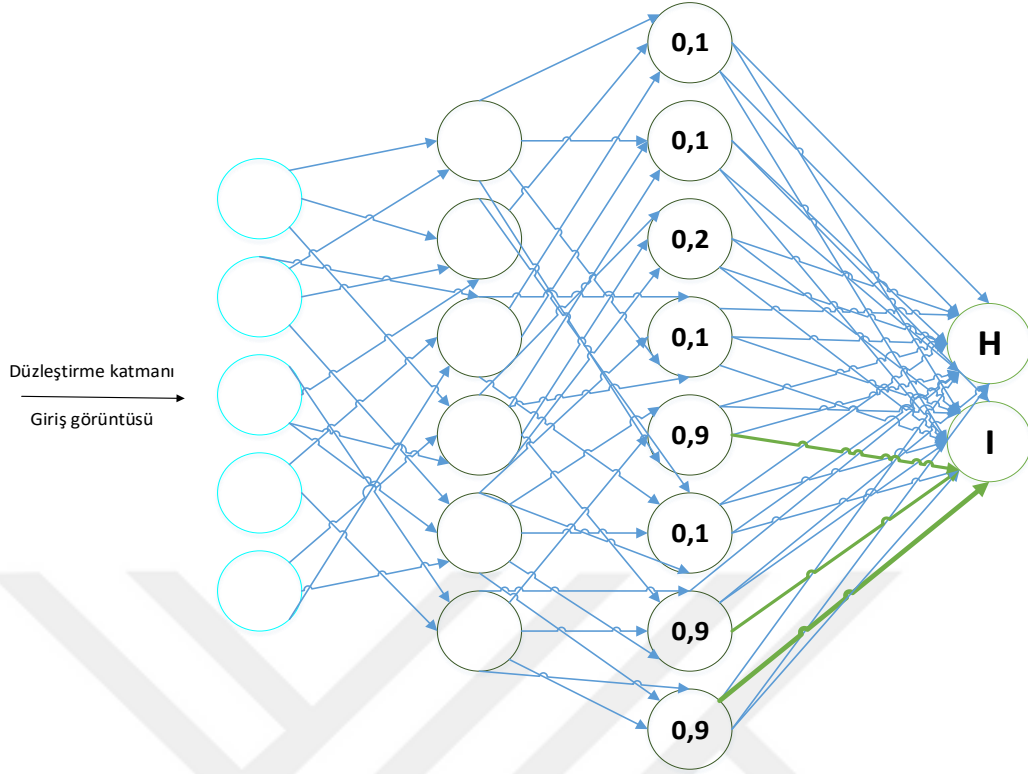
İki çeşit havuzlama katmanı filtresi vardır. Bunlardan ilki ve genellikle tercih edilen maksimum değere göre yapılan filtrelemedir. Bir diğer ise ortalama değere göre yapılan filtrelemedir. Aşağıda Şekil 3.4'te her iki havuzlama filtresine örnek bulunmaktadır.



Şekil 4.4. Maksimum ve ortalama havuzlama işlemi

3.1.5. Tam Bağlı Katman

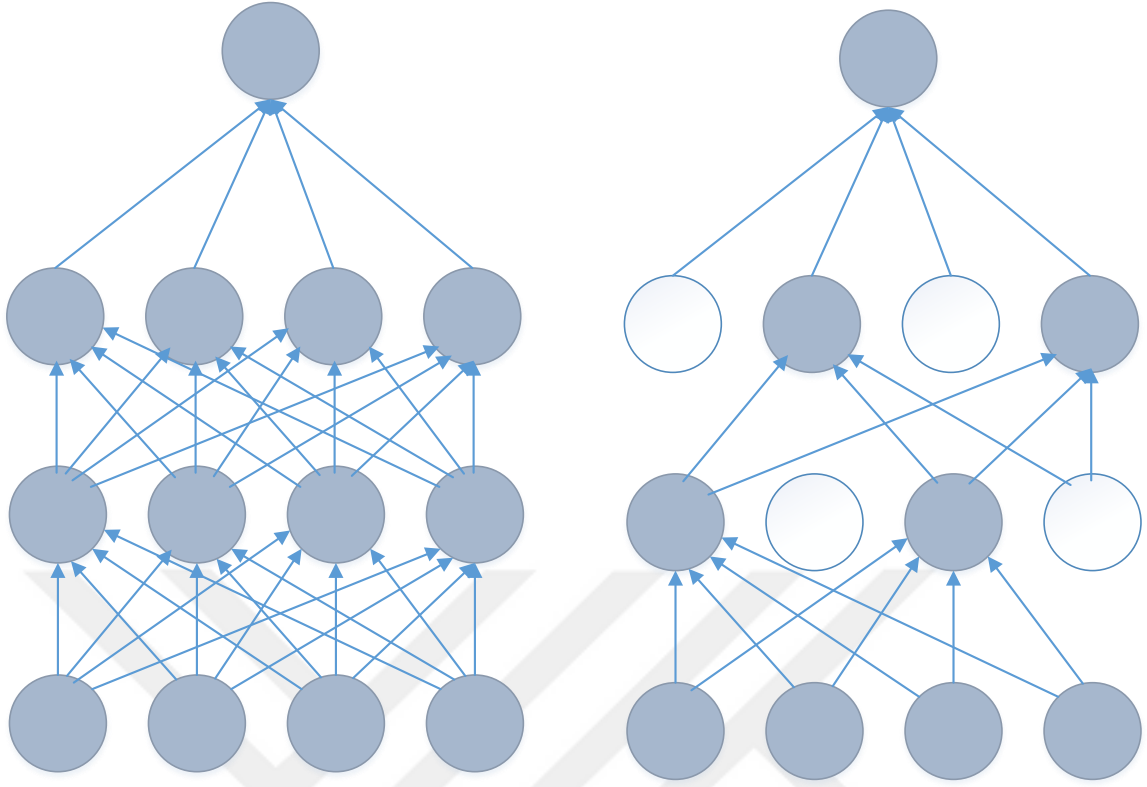
Evrişimsel sinir ağları katmanları arasında her katman kendi sonucunu bir sonraki katmana aktararak işlemlerini gerçekleştirir. Tam bağlı katmana gelindiğinde ise birden fazla matris yapısında olan veriler tek boyutlu matris şekline dönüştürülür bu işlemin amacı hesaplama ve zaman maliyetini düşürmektir [51]. Aşağıda bulunan Şekil 3.5 üzerinde tam bağlı katman yapısı sunulmaktadır.



Şekil 5.5. Tam bağlı katman yapısı

3.1.6. Dropout Katmanı

Dropout katmanı tam bağlantılı katmanda oluşturulan tüm bağlantıları en iyi ifade edecek şekilde sadeleştirir. Bu işlem aşırı öğrenme bir başka ifadeyle ağız ezberlemesine engel olmak amacıyla uygulanan bir işlemdir. Fazla bağlantılar ortadan kalkacağı için ağız performansı da artmış olacaktır [52-53]. Şekil 3.6 üzerinde normal bir sinir ağı ile aynı sinir ağına dropout katmanı uygulanmış ağ yapısı gösterilmektedir.



Şekil 6.6. Normal sinir ağı ve dropout uygulanmış sinir ağı

3.1.7. Normalizasyon (Ölçeklendirme) Katmanı

Evrişimsel sinir ağları için ağı normalize edilmesi, sistem performansı için önemlidir. Ağ normalize edilirse performans artışı sağlanır ayrıca normalizasyon işlemi esnasında katmanlardan gelen verilen düzenli bir hal almaktadır. Katmanlara giren verilerin çok büyük ya da çok küçük olması yerine belirli değerler arasında değerlendirmesi sinir ağı için daha sağlıklı bir işlemdir. Bu sayede ağ performansı ve verilerin girdileri belirli bir ölçeklendirilme ile ifade edilebilir [54].

3.1.8. Sınıflandırma (Softmax) Katmanı

Kendisinden önce gelen katmanlardan elde edinilen bilgileri alarak sınıflandırma işlemi için olasılık değerleri üreten katmandır. Bu hesaplama yapılırken her bir sınıf için ayrı ayrı değerler üretilir [55].

3.2. Derin Öğrenme Parametreleri

ESA mimarilerinin eğitimlerinde katmanların öneminin olduğu kadar parametrelerde önemli rol üstlenirler. Parametrelerin iyi bir şekilde ayarlanmamasının bazı olumsuz etkileri olabilir. Bu etkilerin başında ESA ağının eğitim yapmaması, uzun süren eğitimler sonucunda başarısız sonuçlar

üretmesi veya gereğinden fazla tanımlanan eğitimlerin aynı başarı oranıyla sürekli bir şekilde devam etmesi gibi olumsuzluklara neden olabilmektedir. İlerleyen konularda alt başlıklar önemli parametrelere değinilecektir.

3.2.1. Veri Kümesi

Veri kümesi boyutu sinir ağlarının eğitimi için en önemli faktörlerden birisidir. Veri kümesinde bulunan verilerin sayısı kadar, veri türlerinde ki çeşitlilik de önemli faktörler arasındadır. Veri kümesi içeriği ne kadar büyük olursa ve farklı sınıflar barındırırsa başarıya doğru orantıda etkileri olacaktır. Başarı oranı gibi eğitim süresinde veri kümesinden etkilenecek durumlar arasındadır. Veri kümesi boyutunun aşırı derecede fazla olması bellek açlığı da oluşturabilir. Veri kümesinin oluşturulacak modelin başından sonuna kadarki süreçte doğrudan etkisinin olduğunu söylemek mümkündür.

Veri kümesinde ki örneklerin fazla olması başarıyı olumlu etkilerken, örnek sayısının eksik olması başarıyı olumsuz etkileyecektir. Bu yüzden araştırmacılar ellerinde var olan verileri görüntü işleme yöntemlerini kullanarak görüntü artırımı yöntemine başvurumaktadırlar.

3.2.2. Aktivasyon Fonksiyonları

Aktivasyon fonksiyonları sinir ağı yapılarında önemli bir rolü üstlenmektedirler, bu sebeple ideal sonuçları elde edebilmek için seçilen aktivasyon fonksiyonu oldukça önemlidir. Model geliştirirken doğru aktivasyon fonksiyonunun hangisi olduğuna karar verebilmek için aktivasyon fonksiyonlarının özelliklerinin, avantajlarının ve dezavantajlarının bilinmesi gerekmektedir. Aktivasyon fonksiyonlarının hiç kullanılmaması ise çıkış sinyalini birinci dereceden basit bir grafik olmasına neden olacaktır, bu durumda da sinir ağı sınırlı öğrenme gücüne sahip olacaktır. Buda hata payının artmasına neden olacaktır.

Modellerde ki ağırlıklarda bulunan yüksek hata sapmalarının önüne geçmek için, geriye yayılım tekniği kullanılmaktadır. Bu teknik daha önceki katmanlara dönerek ağırlıkların güncellenerek bir sonraki adıma yeniden taşınmasını sağlar. Bu sayede daha doğru sonuçlar elde edilmektedir. Genel bir ifade kullanılacak olursa aktivasyon fonksiyonlarının en başarılı sonuç vereni ReLU fonksiyonudur [56].

3.2.3. ReLU Fonksiyonu

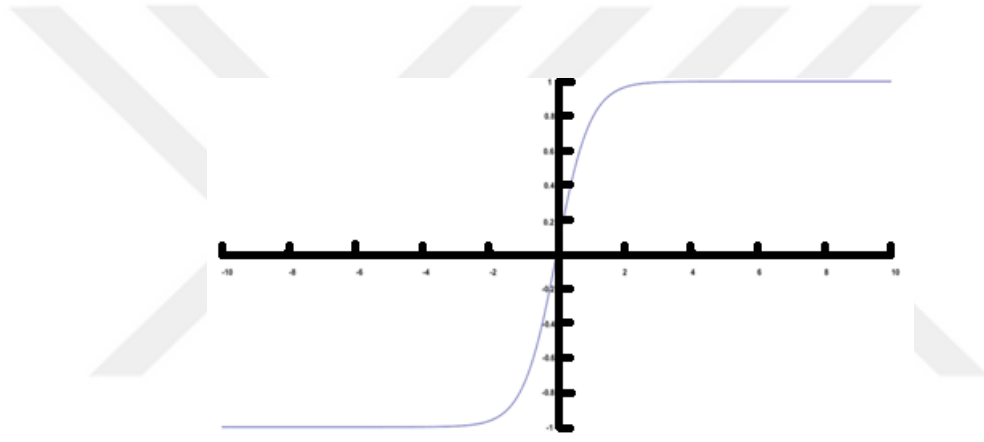
Rectified Linear Unit yani doğrultulmuş lineer birim anlamına gelen ReLU fonksiyonu, kendisine girdi olarak gelen negatif değerlerin yerine sıfır değerini döndürmektedir. Hesaplaması ve zaman maliyeti açısından oldukça avantajlıdır. ReLU fonksiyonuna ait davranış grafiği Şekil 3.3'te gösterilmiştir.

3.2.4. Softmax Fonksiyonu

Softmax fonksiyonu 0 ile 1 arasında deęer döndürmektedir. Sınıflandırma yapılması gereken durumlarda oldukça etkilidir [57]. Aşağıda denklem 3.3'te softmax fonksiyonuna ait matematiksel ifade bulunmaktadır. Şekil 3.7 ise bu matematiksel ifadeye ait grafięi yani softmax fonksiyonu grafięini içermektedir.

$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}} \quad (3.3)$$

Bu formülde σ operatörü softmaxi, $\vec{z}$ giriş vektörünü, e^{z_i} giriş vektörü için standart üstel fonksiyon, K sayısı çok sınıflandırma kullanılacaksa sınıf sayısını, e^{z_j} ise çıkış vektörü için standart üstel fonksiyonu temsil etmektedir.



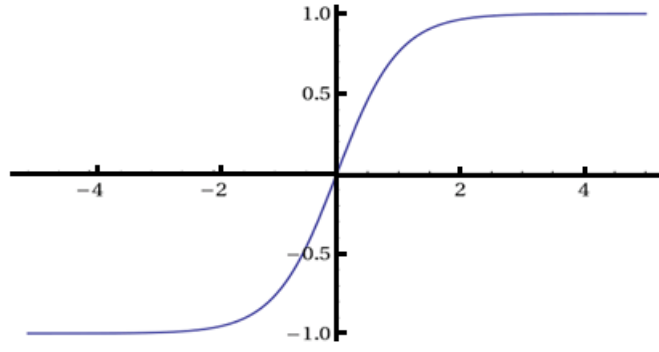
Şekil 7.7. Softmax aktivasyon fonksiyonu

3.2.5. Tanh Fonksiyonu

Tanh fonksiyonu ile softmax fonksiyonu benzerlikler içermektedir ancak tanh fonksiyonu sınıflarının -1 ile 1 arasında olmasından ötürü softmax fonksiyonundan ayrılmaktadır. Denklem 3.4 üzerinde tanh fonksiyonuna ait matematiksel ifade gösterilmektedir. Şekil 3.8'de ise bu matematiksel ifade ait yani tanh fonksiyonunun davranış grafięi bulunmaktadır.

$$\tanh(a) = \frac{e^{2a} - 1}{e^{2a} + 1} \quad (3.1)$$

Tanh fonksiyonu aynı zamanda hiperbolik tanjant olarak da bilinmektedir.



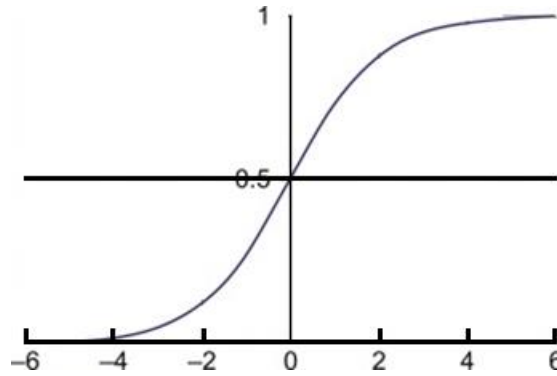
Şekil 8.8. Tanh fonksiyonuna ait davranış grafiği

3.2.6. Sigmoid fonksiyonu

Sigmoid fonksiyonu 0 ile 1 arasında değer almaktadır. Düşük değerleri sıfır olarak kabul ederken yüksek değerleri bir ile ifade etmektedir. Denklem 3.5 üzerinde sigmoid fonksiyonuna ait matematiksel ifade bulunmaktadır. Şekil 3.9 üzerinde ise sigmoid fonksiyona ait davranış grafiği bulunmaktadır.

$$f(x) = \frac{1}{1 + e^{-x}} \quad (3.2)$$

Denklemden bulunan e^{-x} ifadesi çıkış fonksiyonu için standart üstel fonksiyonu temsil etmektedir.



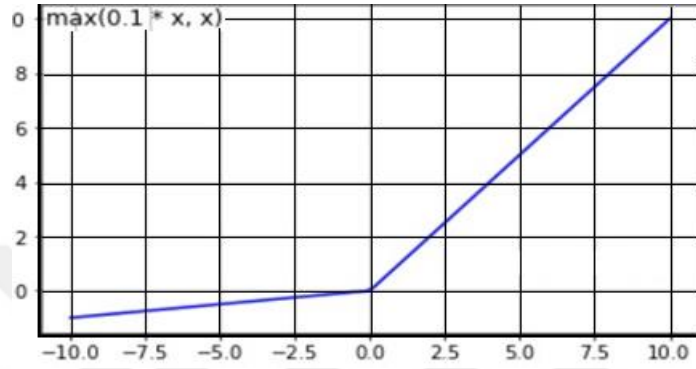
Şekil 9.9. Standart sigmoid aktivasyon grafiği

3.2.7. Leaky ReLU fonksiyonu

Leaky ReLU fonksiyonu ölü nöronlar için geliştirilmiş bir aktivasyon fonksiyonudur. ReLU'dan farklı bir yapıda çalışmaktadır. ReLU fonksiyonu negatif değerler döndürülebilirken, Leaky ReLU fonksiyonu negatif değerlerin yerine çok küçük değerler döndürmektedir. Hesaplama

sayısı fazla olmasına rağmen hızlı çalışmaktadır [58]. Aşağıda denklem 3.6 üzerinde Leaky ReLU aktivasyon fonksiyonuna ait denklem ve Şekil 3.10 üzerinde Leaky ReLU aktivasyon fonksiyonuna ait grafik sunulmaktadır.

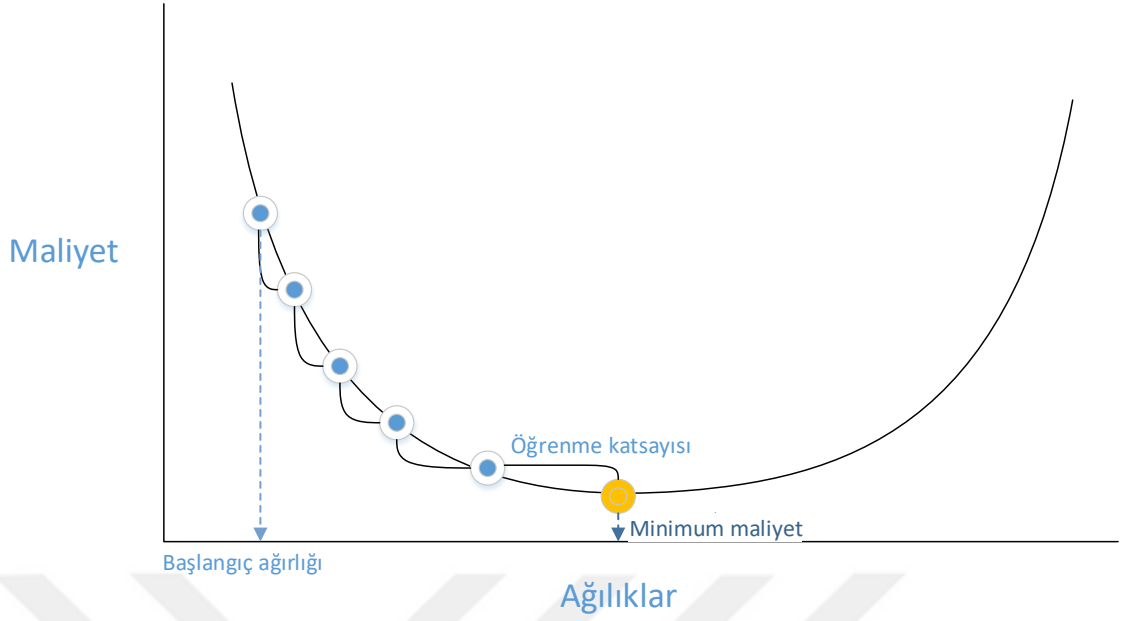
$$f(x) = (0.1x, x) \quad (3.3)$$



Şekil 10.10. Leaky ReLU aktivasyon fonksiyonuna ait davranış grafiği

3.2.8. Öğrenme Katsayısı

Derin öğrenme yapılırken öğrenme katsayısı model üzerinde büyük etkileri olan bir parametredir. Bu parametre derin öğrenme eğitim hızını oldukça etkilemektedir aynı zamanda başarı oranı üzerinde aynı doğrultuda bir etkiye sahiptir. Öğrenme katsayısının literatürdeki bir diğer telaffuzu ise hata düzeltme katsayısıdır. Derin öğrenme modelleri hesaplanırken stokastik eğim inişi (SGD) hesaplanmaktadır. SGD hesaplanırken öğrenme katsayısı burada parametre olarak kullanılır ve öğrenme katsayısı ne kadar büyük olursa adım sayısı azalacağından algoritma daha hızlı çalışacaktır tam tersi durumda ise adım sayısı artacağından ve her adımda hesaplama yapılması gerekeceğinden modelin hızı yavaşlayacak ancak daha doğru bir yaklaşım eğilimine girecektir. SGD algoritmasının temel amacı evrensel en küçük değeri saptamaktır [59]. Bunun için karşılaştırmalar ve hesaplamalar yapar. Bu sebepten dolayı adım sayısının ideal bir değerde ayarlanması gerekmektedir çünkü adım sayısının az olması yani öğrenme katsayısının büyük olması aranan değerin hesaplanmayarak yakınında bulunan bir değerin kabul edilmesiyle sonuçlanabilir. Öğrenme katsayısı için birden fazla sinir ağı bulunan modellerde öngörülen uygun değer 1 ile 10^{-6} aralığındadır [60].



Şekil 11.11. SGD algoritmasının çalışma mantığı

Şekil 3.11 üzerinde SGD algoritmasının öğrenme katsayısına göre minimum maliyetli model ağırlıklarını nasıl bulduğunu temsil eden bir grafik mevcuttur.

3.2.9. Parti Boyu

Parti boyutu veya batchsize eğitim hızını etkileyen faktörlerdendir. Büyük çaptaki veri kümelerinin tek seferde belleğe alınıp işlenmesi günümüz şartları ile mümkün değildir. Bu sebeple veri kümelerinde bulunan veriler bölünerek işlenmelidir. Parti boyutu bu verilerden kaç tanesinin bir araya gelerek işleneceğini göstermektedir. Parti boyutu büyük seçildiğinde tek seferde işlenmek üzere hafızaya alınan eleman sayısı artacağından, veriler üzerinde detaylı bir özellik çıkarımı yapılmayacaktır. Sekiz veriden yapılacak olan özellik çıkarımı ile 64 veriden yapılacak veriden yapılacak özellik çıkarımı arasında farklılıklar bulunmaktadır. 64 veri ile yapılan çalışmada tüm resimlerde olan ortak özellik sayısı azalacağından algoritma daha hızlı işler ancak sekiz veriye göre yapacağı daha detaylı çıkarımda bulunamaz. Yine de model daha çok öğrensin diye parti boyutunun çok düşük seçilmesi önerilmez, çünkü parti boyutu düşük seçildiğinde veriler üzerinde bulunan gürültülerde öğrenmeye katılacaktır. Ayrıca parti boyutunun merkezi işlem birimi olan CPU ve grafik işlemci birimi olan GPU üzerinde daha etkili çalışabilmesi için iki ve katları seçilmesi daha uygun olacaktır [61].

Literatürde yapılan bazı çalışmalar sonucunda 1,2,4,8,16 gibi değerlerin parti boyutu seçiminde kullanılmasının gürültülerin de işlenebileceğine ayrıca bunun zaman maliyetinin artıracığına neden olacağı belirtilirken; 512,1024,2048 gibi değerlerin ise parti boyutu

kullanımında modelin doğrulunu ve başarı performansını düşüreceğini sunulmuştur. Ancak 32,64,128,256 gibi değerleri en iyi sonuçları verdiği belirtilmektedir [62-63].

3.2.10. Katman Sayısı ve Gizli Katmanlar

Katman sayısı derin öğrenme algoritmaları diğer makine öğrenme algoritmalarından ayıran en temel unsurlardan birisidir. Derin öğrenme modellerinin hızını ve öğrenme oranını etkileyen faktörlerden birisi olan katman sayılarının çok olması geri yayılım işleminin en arkada kalan katmanlardan daha az beslenebilmesine neden olabilir. Buda modelin başarısızlığına neden verecek unsurlardan birisidir. Verimli bir model için katman sayısının doğru saptanması önemlidir [64].

3.2.11. Eğitimde Tur Sayısı

Eğitimde tur sayısı ya da epoch sayısı eğitim için kullanılacak verilerin bir kere işlenmesiyle ortaya çıkan işlemdir. Verilerin her bir sefer işlenmesi bir epoch olarak da tanımlanabilir. Model eğitim için ilk turu tamamlandıktan sonra model geri yayılım yapmaya başlar ve ağırlıklar geri yayılım ile güncellenir. Bir ileri bir geri işlemin gerçekleşmesi model için bir turun tamamlanmasına denktir. İyi tanımlanmış bir model için epoch sayısı arttıkça modelin başarı oranı artmaktadır. Yani bu iki değer arasında doğru orantı söz konusudur. Model başarı açısından doygunluğa geldiğinde ise epoch sayısının bir önemi kalmaz ve başarı oranı sabitlenir. Başarı oranı sabitlenen modellerde ağ maksimum başarıya ulaştığından bu noktaya gelindiğinde eğitim sonlandırılır. Ayrıca epoch sayısı eğitim süresini belirleyen faktörlerdendir [65-66].

3.2.12. Kayıp Fonksiyonu

Kayıp fonksiyonu ya da loss fonksiyonu modelin hata oranını ve hata oranına dayanarak modelin ne kadar başarılı olduğunu gösteren bir parametredir. Kayıp fonksiyonları derin sinir ağlarının son katmanında tanımlanmaktadır [67]. Kayıp fonksiyonları aynı zamanda maliyetleri temsil ettiğinden dolayı maliyet fonksiyonları olarak da bilinmektedir. Anlaşılacağı üzere kayıp fonksiyonları yapılan tahminlerin asıl değerlerden ne kadar uzakta olduğunu hesaplamakla yükümlüdürler. İyi sonuç üreten bir derin öğrenme sinir ağı modelinde loss fonksiyonu sıfıra yaklaşacaktır. Loss fonksiyonun sıfıra yakın olması doğruluk payından çok sapmadığı anlamına gelmektedir. Ancak tersi durumda yani loss fonksiyonunun bire yaklaştığı durumlarda modelin başarı oranının oldukça zayıf hatta işlevsiz bir model elde edildiği söylenebilir.

Loss fonksiyonu için kullanılan birçok doğrusal fonksiyon bulunmaktadır. Literatürde en sık kullanılan loss fonksiyonlarının başında softmax, cross entropy, sigmoid ve Likelihood gibi fonksiyonlar bulunmaktadır [68].

3.3. Derin Öğrenmede Transfer Öğrenme

Derin öğrenme yapılırken sık başvuru alan bir yöntem olan transfer öğrenme, literatürde yaygın bir alana sahiptir. İnsanların bilgileri kulaktan kulağa, kitaplar aracılığıyla ya da çeşitli iletişim araçlarını kullanarak nesilden nesile aktarılır [69]. Her geçen zaman artarak gelen bilginin birikimini sağlamaktadır. Bu sayede insanlar geliştikçe kullandıkları bilgi hazneleri ve aktaracakları bilgiler artmaktadır. Her ne kadar bilgi birikimleri artış gösterse de bu birikimler güncel hayatlarda çıkan bazı problemleri çözmek için yeterli olmayabilir. Birey yeni karşılaştığı bu problem karşısında geçmiş bilgilerine başvurur. Geçmiş bilgilerini doğrudan probleme uygulayan birey, problemi çözemez ise çözüm yöntemini geliştirerek birkaç küçük değişiklik ile yeni probleme yeni çözümler üretmeye başlar. Hayatında motosiklet görmeyen bir bireyin, motosikleti bisiklete benzetmesi ve bisikletin hızlı hali diye geliştirmelerde bulunması bu konuya örnek olarak verilebilir.

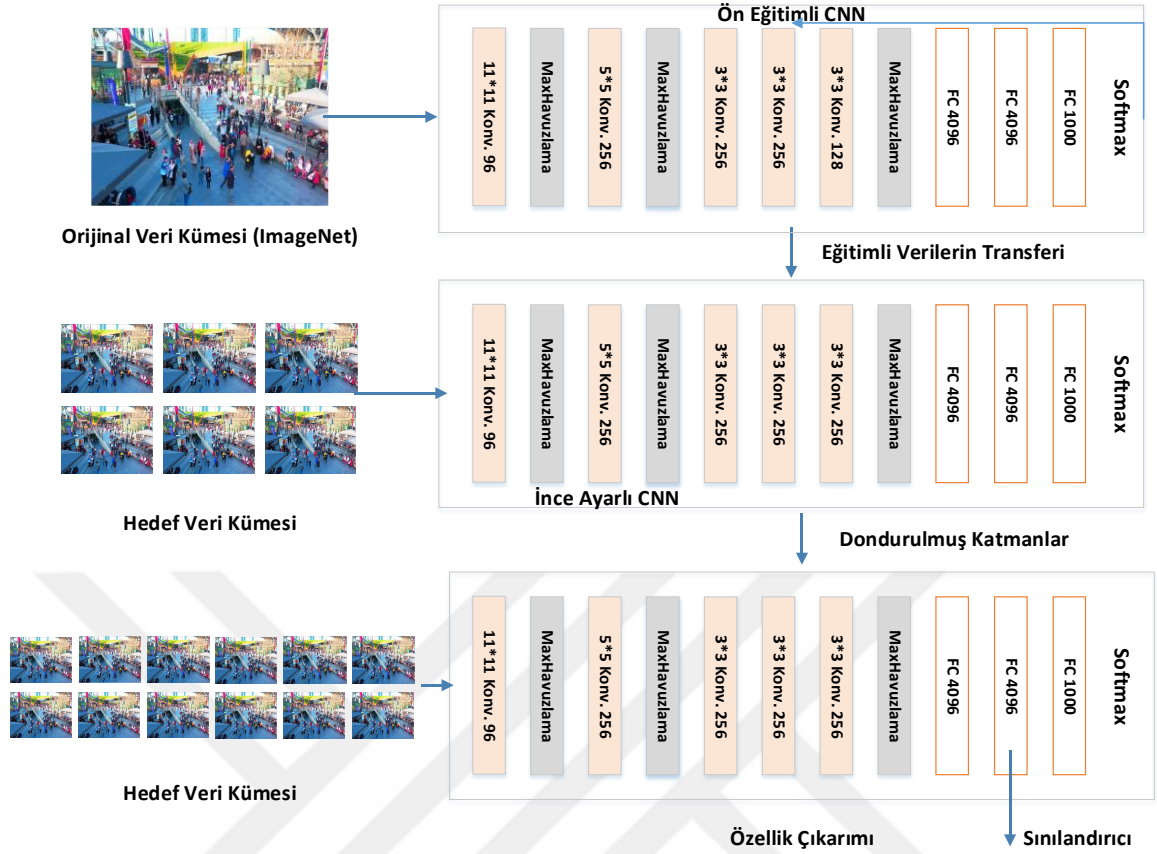
Son yıllara kadar derin öğrenme ve makine öğrenmesi modelleri yalnızca bir problemin çözümüne odaklanmaktaydı. Problemden oluşan küçük gelişmeler veya değişiklikler tasarlanan bu modelin tamamen rafa kaldırılmasına neden olmaktaydı. Bu noktada geliştirilen transfer öğrenme yöntemi geçmişte bulunan bilgilerin yeni problemlerin çözümünde kullanılabilmesi için kullanılmaya başlanmıştır [70].

Transfer öğrenme derin öğrenmede olduğu gibi makine öğrenmesinde de kullanılan bir yöntemdir. Ancak makine öğrenmesinde kullanılan transfer öğrenme yöntemi kendi tasarlandığı sınıf üzerinde başarılı olabilmekteydi. Derin öğrenme de ise bu durum değişmiştir. Derin öğrenmede transfer öğrenme yapıldığında bir modelden diğerine veri aktarılabilir yani ağırlıklar güncellenerek yeni problemin çözümüne göre kendi elden geçirebilmektedir [71]. Buda transfer öğrenmenin daha az veri ile başarılı modeller üretmesini sağlamaktadır.

Derin öğrenmede popüler olarak kullanılan iki çeşit transfer öğrenme yöntemi bulunmaktadır [72]. Bunlardan ilki; Standart özellik çıkarıcı olarak önceden eğitilmiş modeller, ikincisi ise ince ayarlı olarak önceden eğitilmiş modeldir.

İlk yöntemde birbirine bağlı olan tüm katmanların bağlandığı sınıflandırıcı katman kullanılmaktadır. Diğer geri kalan tüm katmanların yerine ise standart özellik çıkarıcı bir katman kullanılmaktadır.

İkinci yöntem ise ince ayarlamalı transfer öğrenme, yalnızca son katmanın kullanımı değil önceki katmanlardan da seçilen katmanların yeniden eğitime katılması sonucu oluşan ileri geri besleme aracılığıyla oluşturulan eğitim yöntemidir. Birçok katman dondurulabilir ancak aktif olan katmanlarda gerçek verilerin ağırlığıyla gerçekleşen ağırlık güncellemesi daha sağlıklı sonuçlar üretmekte oldukça başarılıdır. Transfer öğrenmede kullanılan bazı popüler modeller bulunmaktadır. Bu modeller başlıca; VGG-16, VGG-19, ResNet-50 gibi modellerdir. Aşağıda bulunan Şekil 3.12 üzerinde transfer öğrenmenin iki yaklaşımı sunulmaktadır.



Şekil 12.12. a) ince ayarlamalı transfer öğrenme b) Standart özellik çıkarıcı olarak önceden eğitilmiş modeller

3.4. Derin Öğrenmede Kullanılan Kütüphaneler

Derin öğrenme algoritmalarının eğitimlerini sağlayabilmesi için birçok açık kaynaklı kütüphane geliştirilmiştir. Açık kaynaklı bu kütüphaneler birçok yazılımcı tarafından geliştirilmeye devam edilmektedir. Aşağıda bazı derin öğrenme kütüphanelerine değinilmiştir.

3.4.1. TensorFlow

Google ekipleri tarafından geliştirilmiş olan TensorFlow Kütüphanesi, açık kaynaklı bir derin öğrenme kütüphanesidir. Tek bir API üzerine kurulu olan yapısı ve esnek mimari olanağı ile uzaktan veya yerel cihazlarla CPU ya da GPU dağıtımına izin verir. TensorFlow kütüphanesi ilk başlarda yalnızca makine öğrenmesi ve yapay sinir ağları gibi sistemlerde kullanılmaktayken, günümüzde birçok farklı alana uygulanabilmektedir [73].

3.4.2. Keras

Sinir ağı kütüphanesi olan Keras, diğer derin öğrenme kütüphanelerinin tabanında kullanılmaktadır. Python programlama dili kullanılarak oluşturulan keras kütüphanesi, sonuçların hızlı olarak üretilmesine odaklanarak hazırlanmıştır. Kullanımı kolay ve prototip olarak basit olan kerasın CPU ve GPU desteği mevcuttur. Sınıflandırmada sabit ve yüksek sonuçlar üretir[74-75].

3.4.3. Caffe

Berkeley Vision and Learning Center (BVLC) adlı topluluk tarafından geliştirilmiş olan Caffe kütüphanesi, diğer kütüphaneler gibi CPU ve GPU desteği sunmaktadır. C++ diliyle geliştirilen Caffe, çıkarım gücü yüksek bir kütüphanedir [76].

3.4.4. Torch

Torch kütüphanesinin temelinde Lua programlama dili bulunmaktadır [77]. Torch temel mantık olarak yapılacak işleri basitleştirmek amacıyla, modüler bir yapıda oluşturularak, birleştirilmesi kolay ve esnek olarak sunulmuştur. Modüler olması CPU ve GPU üzerinde çalıştırılacak katmanlara olanak sağlamaktadır. İçerisinde bilgisayarlı görme ve sinyal işleme gibi birçok modül bulunmaktadır. Ayrıca birçok önceden eğitilmiş derin öğrenme modelini de içinde barındırır [78-79].

3.4.5. Theano

Theano [80], Python programlama diliyle geliştirilmiş türev alma işlemlerinde çok etki etkin bir biçimde kullanılan ve derin öğrenme matematiksel işlevlerinde çok kullanışlı olan bir derleyicidir. Theano CPU ve GPU ile birlikte kullanıldığında diğer kütüphanelerden çok daha hızlı sonuçlar üretebilmektedir [81].

3.4.6. MatConvNet

MatConvNet, matlab için geliştirilmiş derin öğrenme modellerinin eğitilebilmesi için kullanılan bir kütüphanedir [82]. Görsel bir ara yüzü bulunan bu kütüphane, Matlab da modellerin bloklar halinde oluşturulmasına izin verir. Blokların okunması ve modüler olması kullanıcıların rahatlıkla model tasarlamasına olanak sağlar. GPU ve CPU üzerinde çalışabilen kütüphane, büyük veri setleri üzerinde de etkili çalışabilmektedir [83].

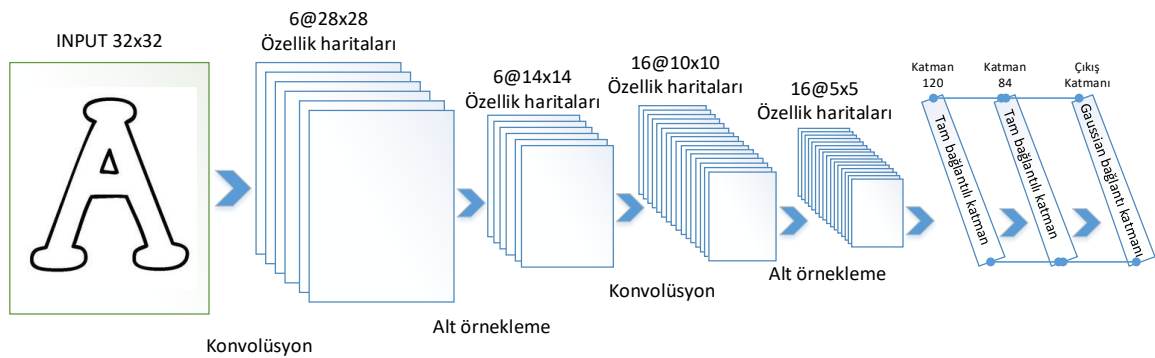
3.5. Derin Öğrenme ile Nesne Sınıflandırma

Genel olarak sınıflandırma varlıkların ortak özellikleri, akrabalık dereceleri ya da benzerliklerine göre yapılan gruplandırma işlemidir. Günlük hayatta da çok kullanılan sınıflandırma işlemi teknolojinin gelişmesi ile daha kullanışlı bir hal almaya başlamıştır. Günümüzde birçok alana uygulanmış örnekleri mevcuttur. Örneğin bir okuldaki öğrencilerin başarı durumlarına göre kategorize edilmesinden, endüstride hata türlerinin ayırt edilmesine kadar kullanılmaktadır.

Derin öğrenme ile geliştirilen sınıflandırma modelleri, son yıllarda oldukça yaygın bir hal almaya başladı. Mail servislerimizde spam maillerin ayırt edilmesi derin öğrenme yöntemleri ile rahatlıkla yapılabilmektedir. Derin öğrenme modellerinin hızla gelişmesi, hızlarının katlanarak artması ve başarı oranlarının günden güne iyileşmesi bu alana olan ilgiyi arttırmıştır. Aşağıda derin öğrenme sınıflandırma modellerine ait bazı modellerden bahsedilecektir.

3.5.1. LeNet

LeNet, LeCun tarafından hazırlanmış ilk Konvolüsyonel sinir ağı olma özelliğine sahiptir [84]. LeCun ve arkadaşlarının yaptığı bu sinir ağı ilk olarak banka çek ve senetleri üzerinde ki sayıları sınıflandırmak amacıyla kullanılmıştır. Sıfır ile dokuz arasındaki sayıları sınıflandıran sinir ağında havuzlama katmanına da yer verilmiştir. LeNet mimarisi içerisinde full connected layer (tambağlı katman), ReLU (aktivasyon) katmanı, havuzlama ve softmax (yumuşak bağı) katmanlarını bulundurmaktadır. Şekil 3.13 üzerinde LeNet mimarisi gösterilmektedir.



Şekil 13.13. LeNet Ağ Mimarisi

3.5.2. MobileNet

MobileNet, Google ekibi tarafından önerildi ve CVPR-2017'de MobileNets, Efficient Convolutional Neural Networks for Mobile Vision Applications başlıklı makalenin başlığıyla

yayınlandı. MobileNet'in adlandırılması, uygulama senaryolarına dayanmaktadır. Adından da anlaşılacağı gibi mobil tarafta kullanılabilen bir ağ modelidir.

Derinlik açısından evrişim ve grup evrişim benzerdir. Derinlik açısından evrişim, özellik haritasının bir kısmından sorumlu bir evrişim çekirdeğidir. Grup evrişim ise, grup özellik haritasından sorumlu bir grup evrişim çekirdeğidir. Derinliksel evrişim, özel bir grup evrişimi olarak da kabul edilebilir, yani bir grup olarak değerlendirilmektedir.

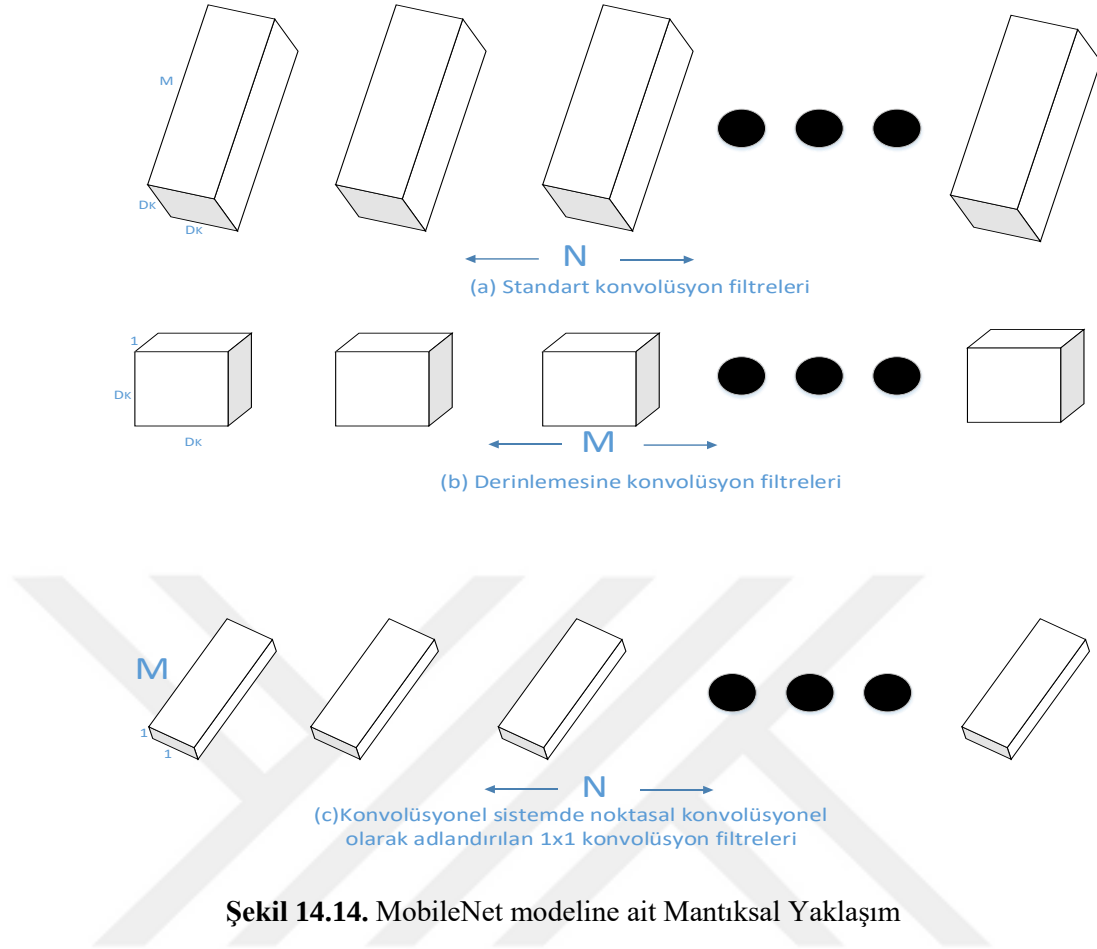
MobileNets'in özü, evrişim yönteminde derinlemesine-akıllı ayrılabilir evrişimde yatar; Derinlemesine ayrılabilir evrişim kullanımı iki hiper parametre içerir: Genişlik Çarpanı ve Çözünürlük Çarpanı. Bu iki hiperparametre, ağıın ne kadar küçük tasarlanacağını ayarlamak için uygundur.

MobileNet, standart birimi iki adıma entegre eder:

- İlk adım, kanal evrişim olan derinlik bazında evrişimdir. Bir evrişim çekirdeği bir kanaldan sorumludur ve bir kanal yalnızca bir evrişim çekirdeği tarafından filtrelendir;
- İkinci adım, Noktasal evrişim, derinlik bilgili evrişimden elde edilen özellik haritasını sıralamaktır.

Başka bir bakış açısından, aslında, her bir özellik haritası çıktısı, giriş katmanının tüm özellik haritası bilgisini içermelidir. Ancak, yalnızca derinlemesine evrişim kullanmak bunu başaramaz, bu nedenle noktasal evrişim gereklidir [85].

Her bir özellik haritası çıktısı, giriş katmanındaki tüm özellik haritalarının bilgilerini içermelidir. Bu, derinlemesine evrişim işlemlerini kullanan tüm ağların çözmesi gereken bir problemdir. Detaylar için Şekil 3.14 Standart evrişim, derinlik yönünde evrişim ve noktasal evrişimin şematik diyagramları aşağıdaki gibidir:



Şekil 14.14. MobileNet modeline ait Mantıksal Yaklaşım

Giriş için M özellik haritası ve çıkış için N özellik haritası vardır. Derinlik açısından evrişim: Bir evrişim çekirdeği bir kanaldan sorumludur ve bir kanal yalnızca bir evrişim çekirdeği tarafından sarılır; o zaman $M \times DK \times DK$ evrişim çekirdekleri vardır;

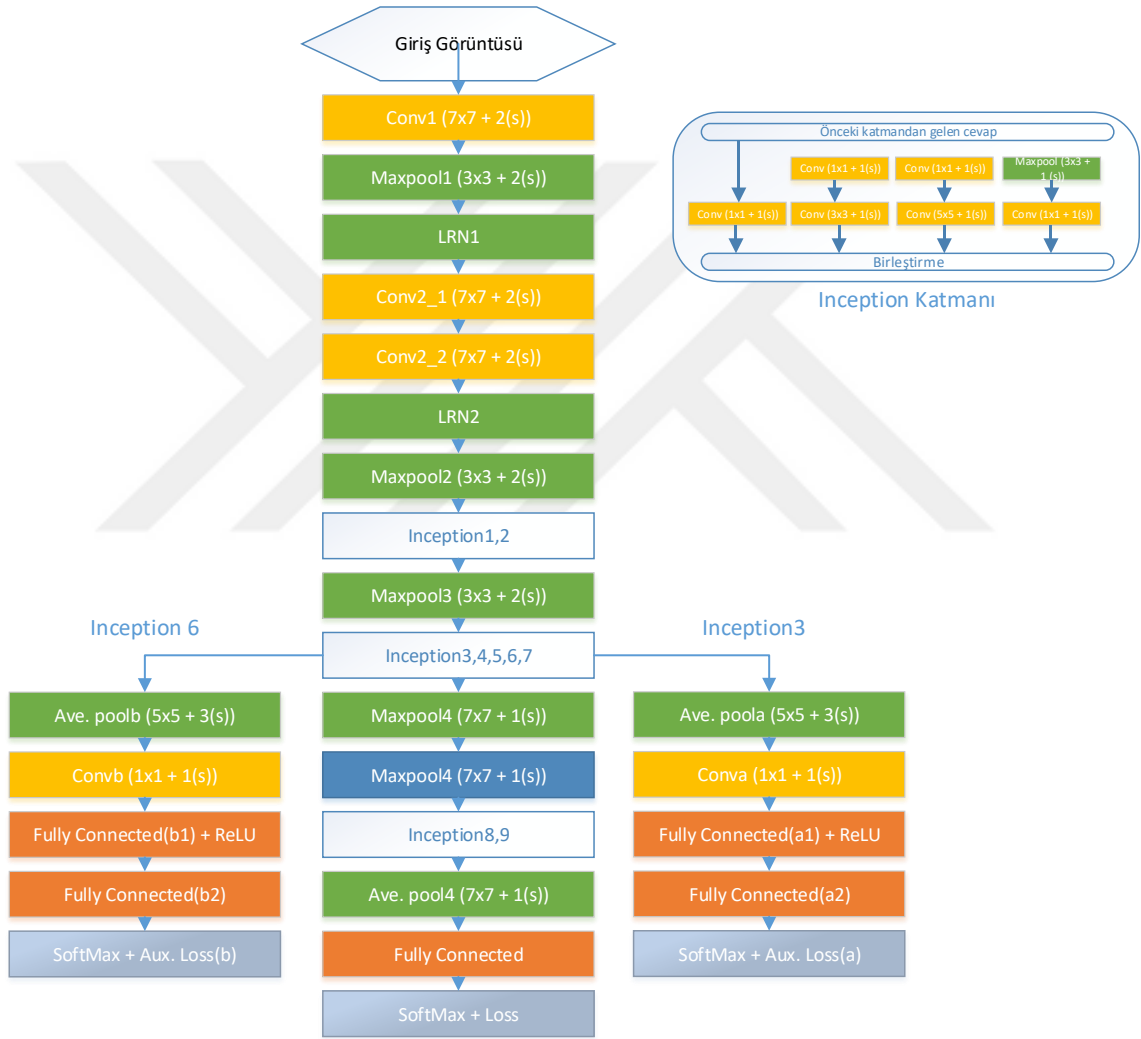
Noktasal evrişim: N adet özellik haritasının çıktısını alma işlemini başarmak için, evrişim için N 1x1 evrişim çekirdeği kullanılır. Buradaki evrişim yöntemi, 1x1 kullanılması dışında geleneksel evrişim yöntemiyle aynıdır. Evrişim çekirdeği; amacı, her yeni özellik haritasının önceki katmanın her bir özellik haritasının bilgilerini içermesini sağlamaktır! Burada, derinliğe dayalı evrişimin çıktısını "dizlemek" anlaşılmaktadır. Kısaca MobileNet'in temel iki özelliği vardır.

- Ana fikir, derinliğe dayalı evrişim işlemini kullanmaktır. Standart evrişim işlemine kıyasla aynı sayıda ağırlık parametresi olması durumunda, ağ çalışma hızını iyileştirme amacına ulaşmak için hesaplama miktarını birkaç kat azaltabilir.

- Derinlik evrişim kullanımının bir sorunu olacaktır, yani bilgi akışı düzgün gerçekleştiremeyecektir. Çıktı özellik haritası, girdi özellik haritasının yalnızca bir bölümünü içerir. Burada, MobileNet bu sorunu çözmek için noktasal evrişim kullanır. Daha sonra ShuffleNet, ağı geliştirmek için aynı fikri kullandı, ancak noktasal evrişimi shuffle ile değiştirdi ve ardından ağa ShuffleNet adını verdi.

3.5.3. GoogleNet

2014 yılında sunulan GoogleNet ağ mimarisi, aynı yıl yapılan ILSVRC galibi olmuştur. Farklı modüllerin bir araya getirerek oluşturmuş olduğu GoogleNet mimarisi karmaşık bir yapıya sahiptir. Toplamda 144 katmanı bulunan GoogleNet ağ mimarisi, 22 katman derinliğe sahiptir. GoogleNet mimarisinde kullanılan önceden eğitilmiş ağlar 224x224 görüntü girişine sahiptirler. Aşağıda GoogleNet ağ mimarisine ait Şekil 3.15'ü görebilirsiniz.

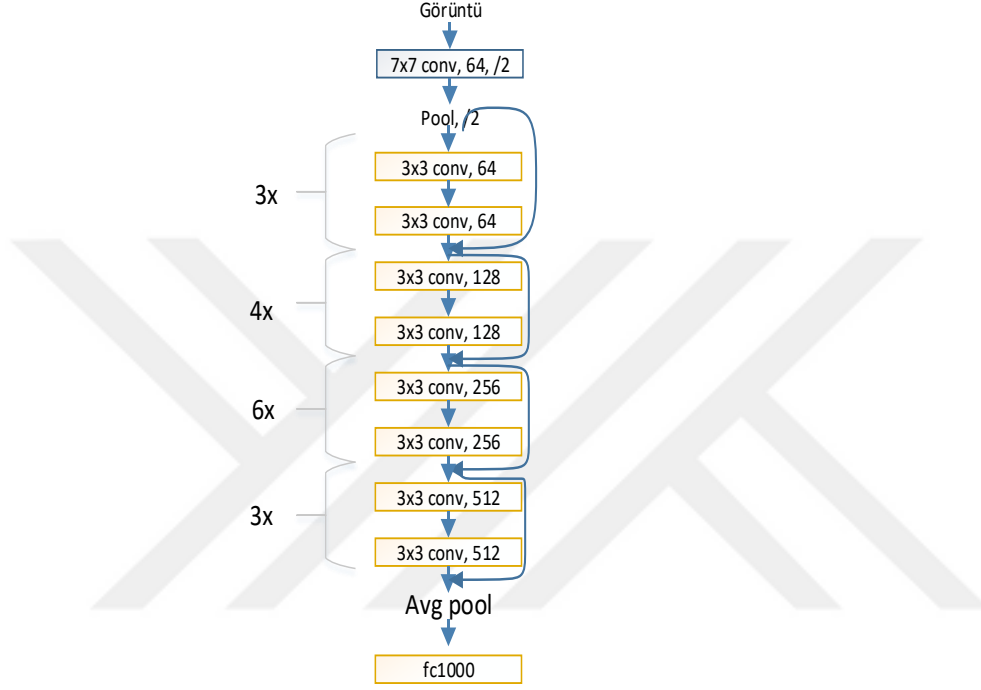


Şekil 15.15. GoogleNet Ağ Mimarisi

3.5.4. ResNet

yılında ImageNet Large Scale Visual Recognition Challenge (ILSVRC) yarışmasını kazanan ResNet, yarışmada ilgili veri seti üzerinde %3.6 oranında hata oranı ile çalışmıştır. Görüntüyü sınıflandırırken insanlar ortalama %5-%10 hata oranı yapmaktadırlar ancak ResNet

sınıflandırma alanında daha iyi olduğunu göstermiştir. Kendisinden daha önce üretilen mimarilerin katman sayısından daha çok katman sayısına sahip olan ResNet ağı ayrıca daha derin bir yapı içermektedir [86]. Aşağıda bu mimariye ait Şekil 3.16 gösterilmektedir.



Şekil 16.16. ResNet Ağ Mimarisi

3.6. Derin Öğrenme ile Nesne Tespiti

Bilgisayarlı görmenin en temel ve zorlu problemlerinden biri olan nesne tespitinin amacı, bilgisayarlı görü uygulamalarının ihtiyaç duyduğu en temel bilgi parçalarından birini sağlayan hesaplamalı modeller ve teknikler geliştirmektir. Derin öğrenme teknikleri, özellik temsillerini doğrudan verilerden öğrenmek için güçlü bir strateji olarak ortaya çıkmış ve genel nesne tespit alanında kayda değer gelişmelere yol açmıştır. Bazı durumlarda nesne algılama ile nesne sınıflandırma algoritmaları karıştırılabilmektedir. Bu iki algoritma arasındaki temel farklılık algılama yapılırken nesneyi sınırlandırıcı kutular içerisine almaya çalışmaktır. Bir görsel içerisinde birden fazla nesne tanımlanabileceğinden yalnızca bir kutu değil de birden fazla kutu çizilmesi gerekebilir. Bu durum nesne algılamanın başlıca zorlukları arasındadır.

Uzun yıllar üzerinde çalışılmış bir konu olmasına karşın ilk devrim niteliğinde atılan adım Paul Viola ve Micheal Jones tarafından bir makale ile duyurulan Viola Jones algoritması oldu [87].

Bu algoritma ile hızla gelişme kazanan nesne tespit algoritmaları çalışmalarda daha çok yer edinmeye başlamıştır. Nesne tespiti o zamandan bu yana birçok alanda kullanılmaya başlanmıştır.

Başlıca bu alanlar şunlardır:

- Yüz tespiti,
- Araç tespiti,
- Yaya tespiti,
- Kusurlu yüzey tespiti,
- Çeşitli ürünlerin ayrıştırılması,
- Depremden sonra uydu görüntüleri üzerinden zemin hasar tespiti.

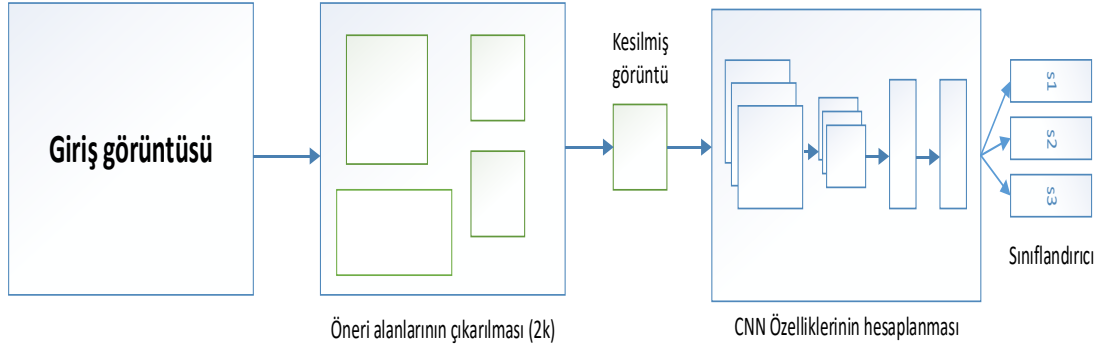
Yukarıda verilen birçok alanda derin öğrenme üzerinden yapılan nesne detektör algoritmalarından faydalanılmıştır. Bu algoritmalar R-CNN(Regions with Convulation Neural Network), SSD(Single Shot Detector), YOLO(You Only Look Once)'dır.

3.6.1. Regions with Convulation Neural Network

Nesne tespit modellerinden birisi olan Regions with Convulation Neural Network (R-CNN), görsel üzerindeki bölge seçimine farklı bir yaklaşım getirmiştir. Ross Girshick ve ark. [88] görüntüden sadece 2000 bölgeyi çıkarmak için seçici aramayı kullanan bir yöntem önerdi ve bunlara bölge önerileri adını verdi. Bu yöntem sayesinde çok sayıda bölgeyi sınıflandırmaktansa yalnızca 2000 bölge üzerinde, seçici arama algoritması kullanılmıştır. Algoritma üç temel prensibe dayanmaktadır. Bu prensipler aşağıda sıralanmıştır.

- İlk alt segmentasyonu oluşturularak, bu segmentasyon ile birçok aday bölge oluşturulur.
- Benzer olan bölgelerin daha büyük benzer bölgelerle yineleme bir şekilde birleştirmek için açgözlü algoritmayı kullanılır.
- Nihai aday bölge teklifleri üretmek için oluşturulan bölgeler kullanılır.

Yukarıdaki seçici algoritma prensibi kullanılarak oluşturulan 2000 aday bölge önerisi bir kare matris olarak 4096 boyutlu bir özellik vektörü olarak bir evrimsel sinir ağı tarafından işlenmektedir. Burada CNN öznitelik çıkarma görevini üstlenmektedir. Bu görüntüden çıkarılan öznitelikler o aday bölge içerisinde destek vektör makineleri (SVM) tarafından sınıflandırılır. Aşağıda Şekil 3.17 üzerinde R-CNN modeline ait gösterim modeli gösterilmektedir.

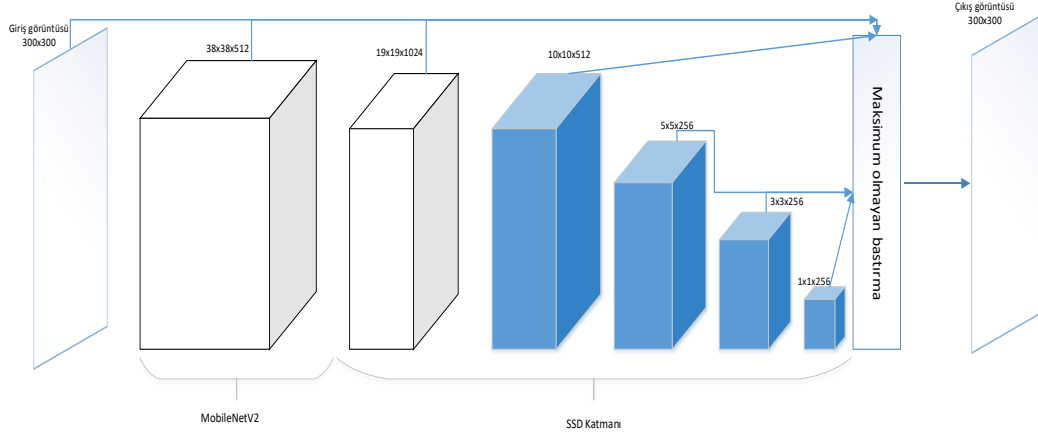


Şekil 17.17. R-CNN çalışma modeli

R-CNN modeli avantajlı olmasına rağmen bazı dezavantajları da bulunmaktadır. Aday bölge önerisi 2000 bölgeden oluşan bir kare matristen bile oluşsa, aday bölge sayılarının çokluğu eğitim süresini etkilemektedir. Yaklaşık olarak her test görüntüsüne bir dakikaya yakın vakit harcayan R-CNN modeli gerçek zamanlı tespitler için uygun değildir. R-CNN modeli sabit bir model olduğundan dolayı, farklı veri kümeleri üzerinde kötü aday bölge önerilerine sebep olabilir.

3.6.2. Single Shot Detector

Tek bir derin sinir ağı kullanarak görüntülerdeki nesneleri algılamak için sunulan bu yöntem nesne tespiti konusunda çok önemli bir yaklaşımdır. SSD adlı yaklaşım, sınırlandırıcı kutuların çıktı alanını, özellik haritası konumu açısından bir dizi varsayılan kutulara bölütler. Bu sayede teklif oluşturma, yeniden örneklendirme aşamalarını tamamen ortadan kaldıran bu yöntem sayesinde algoritma daha hızlı çalışmaktadır. Single shot detector(SSD)'in temel amacı küçük evrimsel katmanlar sayesinde hızlı tespit yapabilmeyi sağlamaktır [89]. SSD aynı zaman da farklı özellik haritalarından farklı ölçeklerin tahminlerini üreterek yapılacak tahminleri en-boy oranına göre ayırmaktadır. SSD yaklaşımı, sabit boyutlu bir sınırlayıcı kutu koleksiyonu üreten ve bu kutularda nesne sınıfı örneklerinin varlığı için puanlar üreten, ileri beslemeli evrimsel ağı dayanmaktadır. Aşağıda Şekil 3.18 üzerinde SSD modelini gösterilmektedir.



Şekil 18.18. SSD Algoritması

3.6.3. You Only Look Once

Derin öğrenme algoritmalarının gelişimi ile birlikte gelişen YOLO(You Only Look Once) algoritması da nesne tespit algoritmaları arasında etkin kullanılan modellerden biridir. Özellikle gerçek zamanlı nesne tanıma algoritmaları arasında yine oldukça yaygın kullanıma sahiptir. Gerçek zamanlı nesne tespiti yapabilen algoritmalar olmasına rağmen YOLO hız faktörü ile popülerlik kazanmıştır. İlk olarak YOLO takip eden süreçte ise YOLO'nun geliştirilmiş versiyonları üretilerek YOLO algoritmaları geliştirilmiştir. YOLO hızının temel sebebi ise görüntüyü bir kez tarayarak görüntü üzerindeki nesnelerin sınıflarını ve ağırlıklarını tahmin edebilmektedir. YOLO nesne tanımak için sınıflandırıcıları kullanarak incelemesi yapılan görüntüde yeterli miktarda konum ve ölçekte çerçeveler inceleyerek sınırlayıcı çerçeveler oluşturup bu çerçevelere olasılık tahmini oluşturarak bu olasılıklardan ağırlıklar oluşturmaktadır. Sonuç çıktısı olarak Genel Ortalama Kesinlik(mAP) değeri oluşturmaktadır. Görüntü üzerinde $X \times X$ (3x3 veya 5x5 istenilen ölçekte) şeklinde maske matrisleri oluşturarak görüntünün taraması gerçekleştirilir. Ardından bu maskeler incelenerek görüntü üzerindeki nesnenin hangi görüntü matrisine denk geldiği tespit edilir. Nesnenin kesiştiği matris nesnenin orta noktasını hesaplamaya çalışarak nesneyi bir çerçeve içine almaktadır. Çerçeveye alınan görüntünün daha sonra ağırlıklara göre ait oldukları sınıfa göre etiketleme işlemi gerçekleşir [90].

Bir sınırlayıcı çerçeve(kutu) belirlenmek üzere yükseklik, genişlik, sınıf, çerçeve merkezleri oluşturulur. Her sınırlayıcı kutu belirli parametreler ile oluşmaktadır. Her kutu kendi tahmin skorunu da oluşturmaktadır.

$$y = (pc, bx, by, bh, bw, c) \quad (5.1)$$

Eşitlikte verilen b_x , b_y çerçevenin merkezini göstermektedir. C tespit yapılmak istenen sınıfları göstermektedir. b_h , b_w ise çerçevenin yükseklik ve genişliğini göstermektedir. Denklem ile

4. MATERYAL METOT

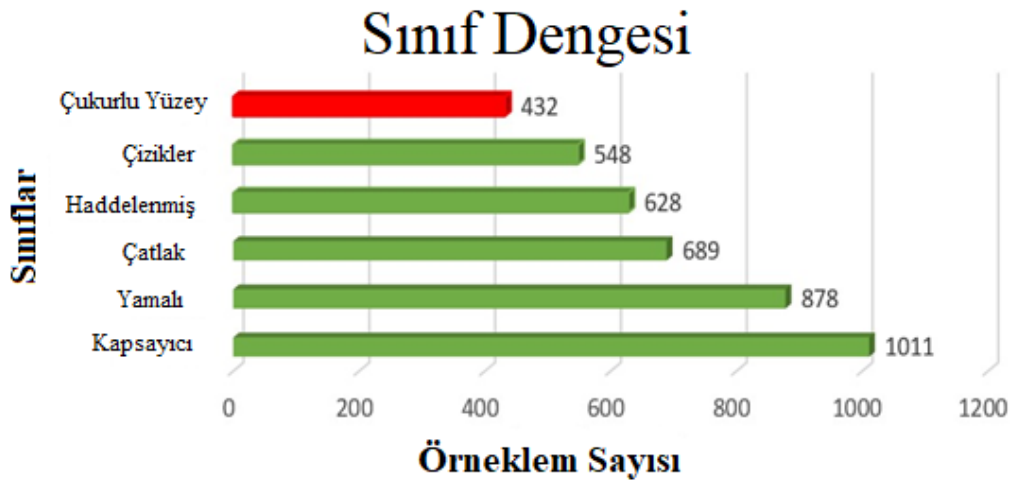
Çalışmanın bu bölümünde yapılan çalışmalarda kullanılan veri setleri, sınıflandırma için kullanılan modeller ve YOLO modellerinin nesne tespitinde karşılaştırılmasında kullanılan modeller açıklanacaktır.

4.1. Veri Seti

Bu bölümde çalışmada kullanılan veri setlerinden bahsedilecektir. Uygulama amacıyla iki veri setinden faydalanılmıştır. Veri setlerinden biri ray yüzey kusur tiplerinin sınıflandırılmasına dayalıdır. Diğer veri seti ise metal yüzey kusurları ile ilgilidir.

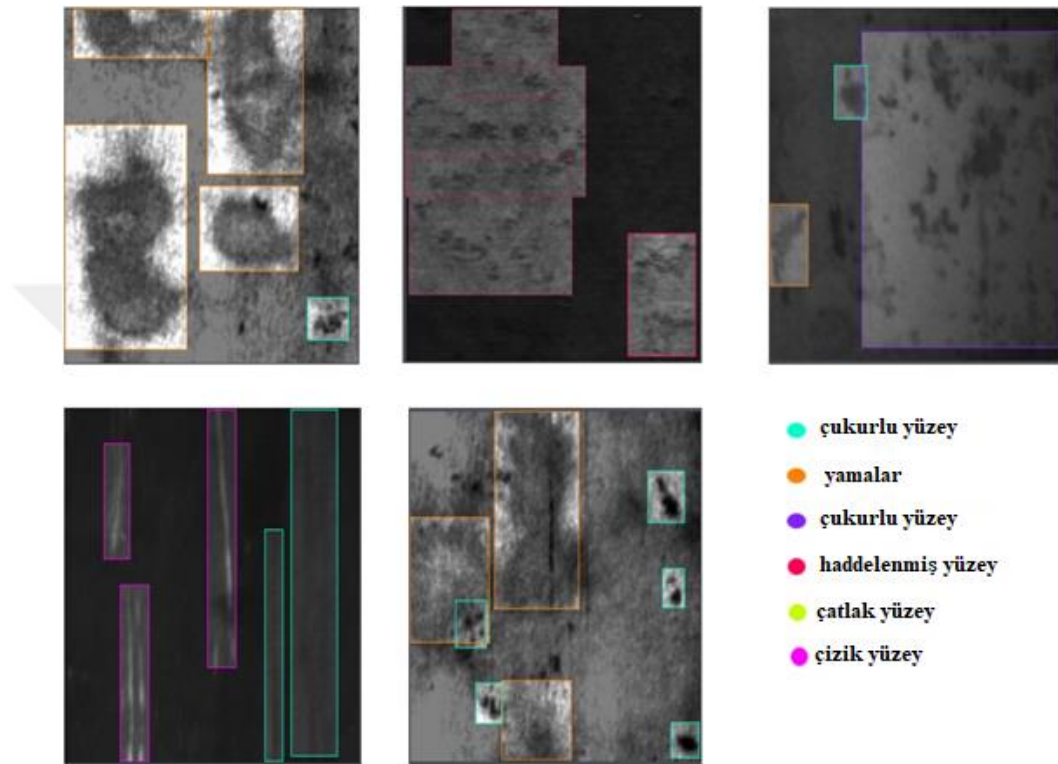
4.1.1. NEU Veri Kümesi

Çalışmada Northeastern Üniversitesi tarafından düz metal yüzey kusurları üzerine hazırlanan veri seti kullanılmıştır. Üniversitenin adını kullanan veri seti aynı zamanda Northeastern University Dataset (NEU Dataset) [91] olarak da bilinmektedir. Bu veri setinde altı farklı sınıf bulunmaktadır. Bu sınıflar; kapsayıcılar, yamalar, çatlama, haddelenmiş ölçek, çizikler ve çukurlu yüzeydir. Veri setinde her sınıfta 300 adet görsel bulunmakta olup toplamda 1800 adet görsel bulunmaktadır. Her resim 200x200 boyutunda ve yaklaşık 40KB'dir. Bu görüntüler arasında tek bir görüntü üzerinde farklı türde kusurlara sahip görüntüler de bulunmaktadır. Bu, arıza tespitini zorlaştıran bir faktördür. 1800 görüntüde 4186 adet arıza etiketi bulunmaktadır. Bu etiketlerin dağılımları Şekil 4.1'de gösterilmiştir.



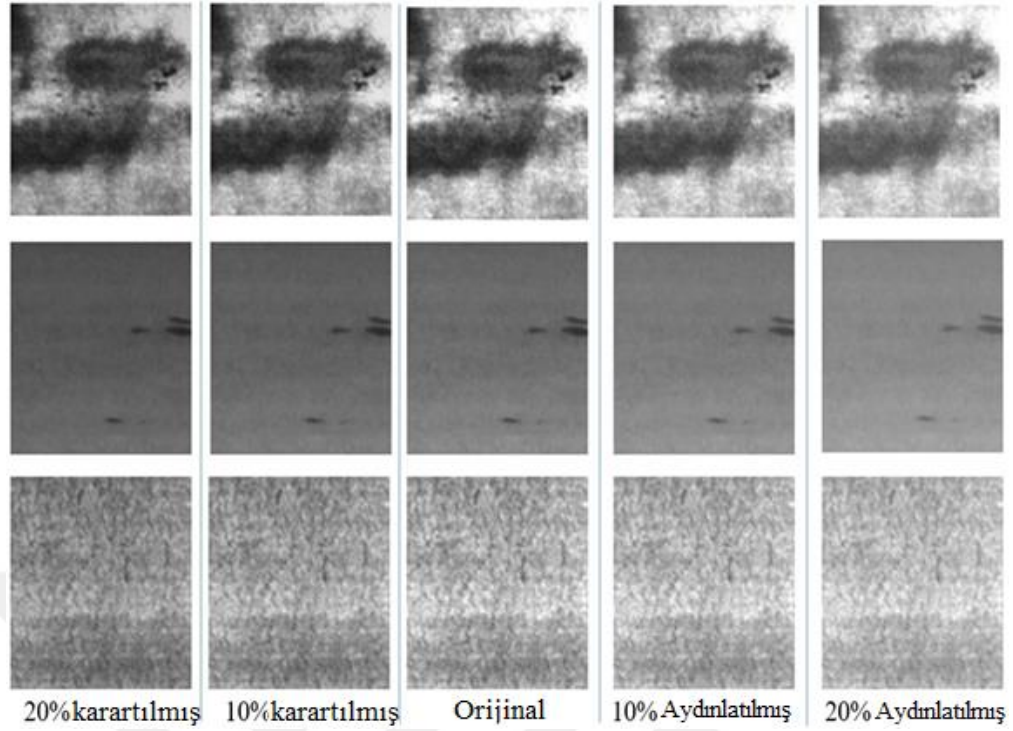
Şekil 20.1. Sınıflara göre etiket dağılımları

Görüldüğü gibi veri setinde dengeli bir dağılım göstermeyen etiketli hata sayıları eğitim için oldukça zorlayıcı bir faktördür. Şekil 4.2 de her sınıf için etiketlenmiş bir veri seti örneği görülmektedir.



Şekil 21.2. Veri kümesinden etiketlenmiş veri örnekleri

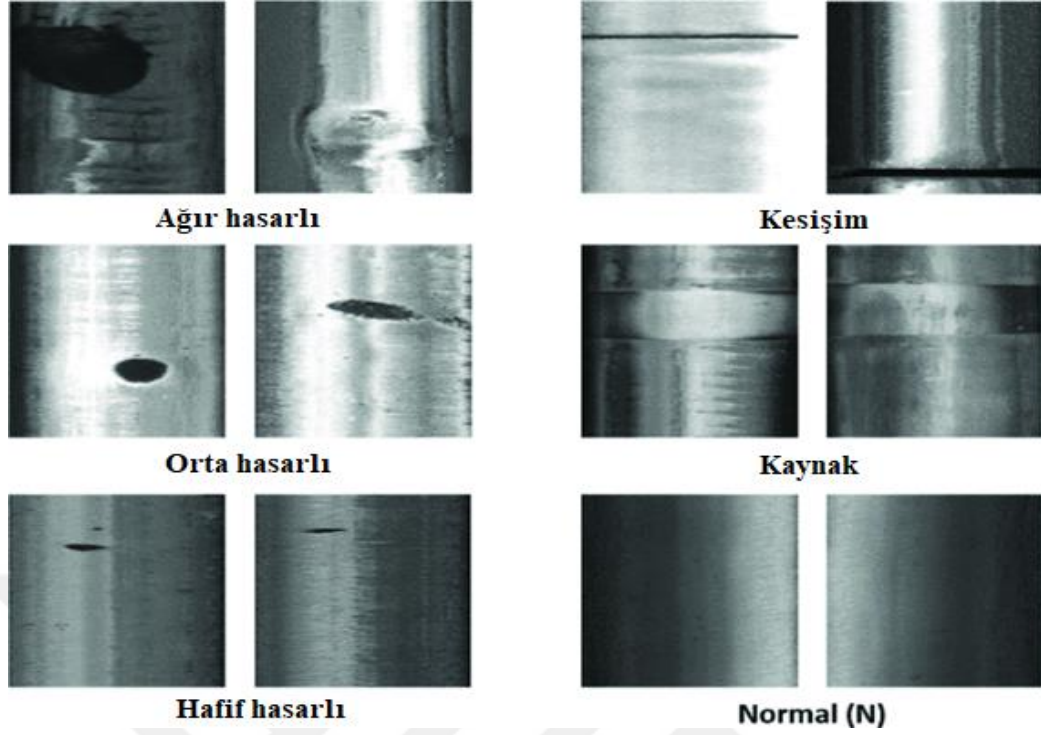
Nesne tespit çalışmasının amacı aydınlatma ve karartmanın eğitim başarısına etkisini ölçmek olduğundan, eğitim, test ve doğrulama verilerinde herhangi bir değişiklik yapılmadan sadece parlaklık ayarları değiştirilerek yukarıda belirtilen modellerde aynı eğitimler tekrarlanmıştır. Orijinal veri seti ve karşılaştırmalar yapılmıştır. Veri seti üzerinde %10 ve %20 aydınlatma ve karartma sağlanarak yeni veri setleri oluşturulmuştur. Bu işlemler sonucunda aydınlatılmış veri, karartılmış veri ve orijinal veri olmak üzere toplam beş veri seti elde edilmiştir. Oluşturulan veri kümelerinin örnek bir gösterimi aşağıda Şekil 4.3 üzerinde gösterilmektedir.



Şekil 22.3. Oluşturulan veri kümelerine bazı örnekler

4.1.2. Sınıflandırma İçin Tarafımızca Hazırlanmış Veri Seti (FUB)

Veri setimizde kendimiz tarafından manuel olarak etiketlenen altı sınıf bulunmaktadır. Bu sınıflar hafif hasar, orta hasar, ağır hasar, normal, eklem ve kaynak olarak sınıflandırılır. Bu sınıflarda eğitim, test ve doğrulama verileri bulunur. Her sınıf için eğitim verisi 750, her sınıf için test verisi 60 ve her sınıf için doğrulama verisi 100'dür. Toplam 5460 görselden oluşmaktadır. Veri setinin her koşulda oldukça doğal bir şekilde elde edilmesine özen gösterilmiştir. Görseller 224x224 boyutunda kullanılmıştır. Aşağıda Şekil 4.4. üzerinde veri setine ait sınıfların örnekleri bulunmaktadır.



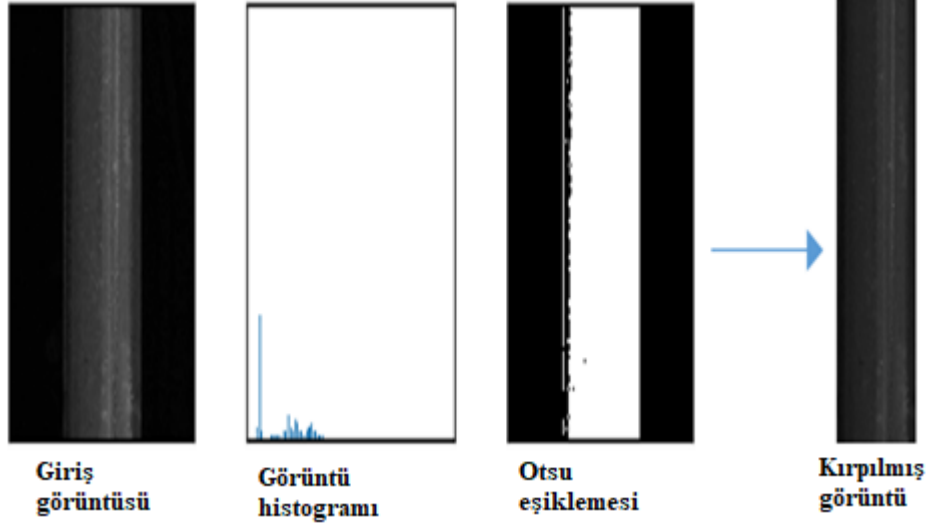
Şekil 23.4. FUB veri kümesine ait hasar sınıfları

FUB veri kümesinde bulunan örnekler NEU veri kümesi ile birlikte yalnızca sınıflandırma amacı ile kullanılmış olup, nesne tespitinde NEU veri kümesi kullanılmıştır.

4.1.3. Veri Seti Ön İşleme

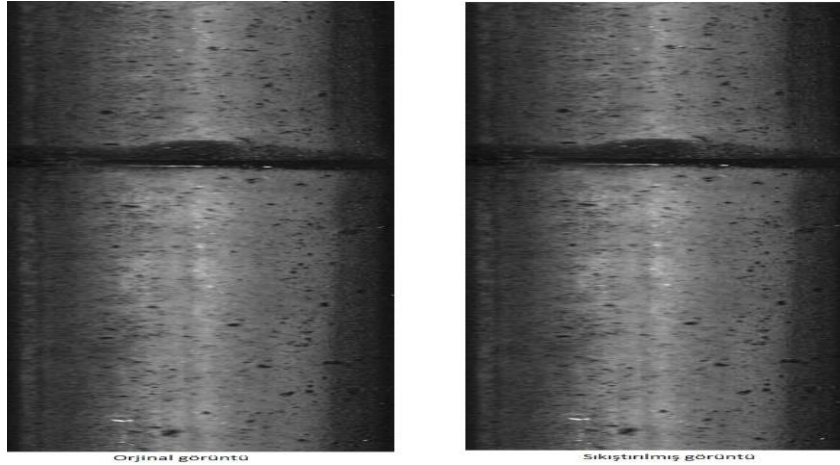
Bazı modeller belli kalıplardaki görseller üzerinde çalışabildiğinden dolayı, bazı ön işlem süreçlerinden geçirilmesi gerekebilir. VGG-16 modeli içinde elimizde bulunan veri kümesi için bazı ön işleme adımları uygulanmış olup, bu adımlara aşağıda sırasıyla değinilmektedir.

Odak Görüntünün Elde edilmesi: Metal yüzeyler parlak metaller olduğu için çevrelerinde nesnelerden farklı piksel aralıkları oluştururlar. Odak görüntüsünü elde etmek için OTSU eşikleme yöntemi kullanıldı. OTSU [92] yöntemi ile bir filtre elde edilmekte, ardından bu filtre üzerinde lineer olarak ilerleyen uç noktalardan görüntü düzlemi boyunca kırpmaya sağlanmaktadır. Bu sayede gürültülerden kurtulmuş tren rayı görselleri elde edilmiş olunur. Odak görüntüsünün elde edilmesi Şekil 4.5 üzerinde adım adım gösterilmektedir. Görüntü histogramı üzerinden çalışan bu algoritma sayesinde görseller çok hızlı bir şekilde gürültülerden kurtulmuş olur.



Şekil 24.5. OTSU ile odak görüntüsünün elde edilmesi

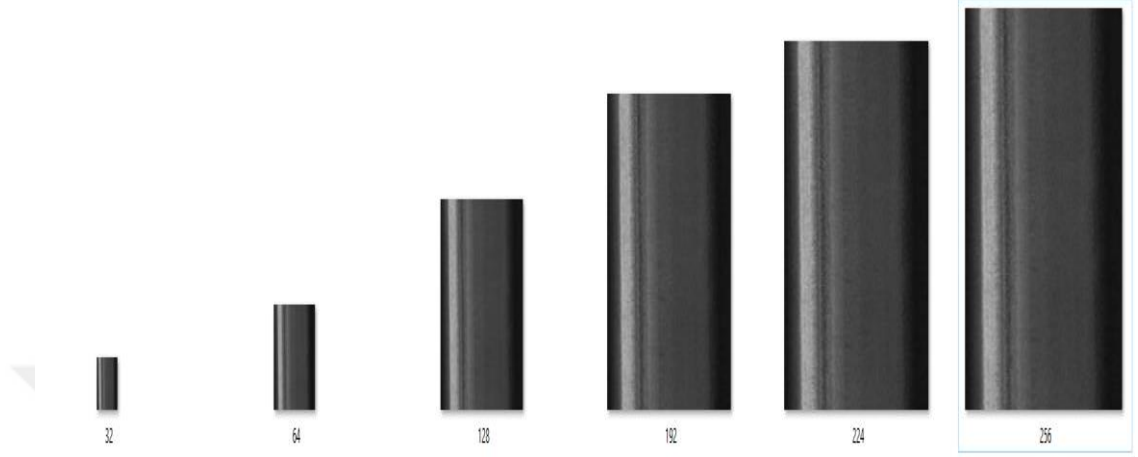
Veri Sıkıştırma Adımı: Veri sıkıştırma adımı; çok fazla yer kaplayan veri kümesi belirli bir yüzde ile sıkıştırılarak gerçekleştirilen derin öğrenme adımlarının daha hızlı gerçekleşmesine olanak sağlamaktadır. Aşağıda bulunan Şekil 4.6 orijinal ve sıkıştırılmış iki görsel bulunmaktadır. Gözle ayırt edilebilecek farklılık olmasa da boyut olarak küçülme söz konusudur.



Şekil 25.6. Orijinal ve sıkıştırılmış görüntü

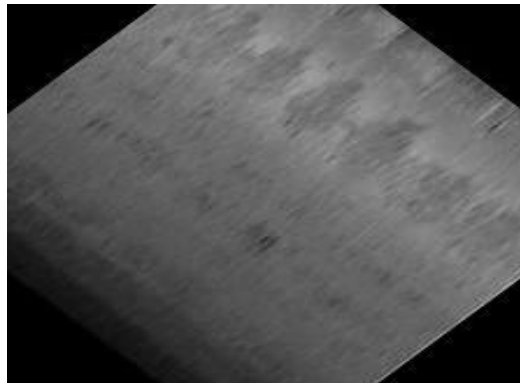
Görüntü boyutunu ayarlama: Etkili bir Derin öğrenme çalışması için önemli iki ölçüt bulunmaktadır. Bunlardan ilki, Görüntülerin en ve boy oranlarının eşit olması hem hızlı çalışma açısından hem de verimli sonuç açısından önem arz etmektedir. İkincisi ise, veri kümesi üzerindeki

sınıflarda bulunan örnek sayısının eşit olmasıdır. Aşağıdaki Şekil 4.7 üzerinde derin öğrenmede en sık kullanılan boyutlandırmaların, bir örnek üzerinde işlenmiş hali mevcuttur.



Şekil 26.7. Derin Öğrenme Modellerinde Sıkça Kullanılan Görsel Boyutları

Görüntü Artırımı: Demir yolu veri seti sağlam verilerin, hasarlı verilere oranının çok yüksek oranlarda olduğu verilerdir. Çünkü Malzeme taşımanın yanında insan da taşıyan trenler bu yolları kullanılmakta olup güvenilir taşıma için gereklidir. Durum bu şekilde olunca her sınıfta bulunan veriler yeterli gelmeyebilir. Bunun için görüntü artırımı yöntemlerinden faydalanılır. Bu yöntemler aynalama, görüntü döndürme gibi çeşitli alternatiflerden birisi olabilir. Ancak, her görüntü artırımı yöntemi yapılacak olan derin öğrenme ağını olumlu yönde etkilemeyebilir. Bazı görüntüler artırım esnasında başka sınıflara benzeşebilir veya özelliğini yitirip kendi sınıfından çıkabilir. Aşağıda bulunan Şekil 4.8 üzerinde görüntü artırımına ait örnek bir görsel paylaşılmaktadır.



Şekil 27.8. Görüntü artırım örneği

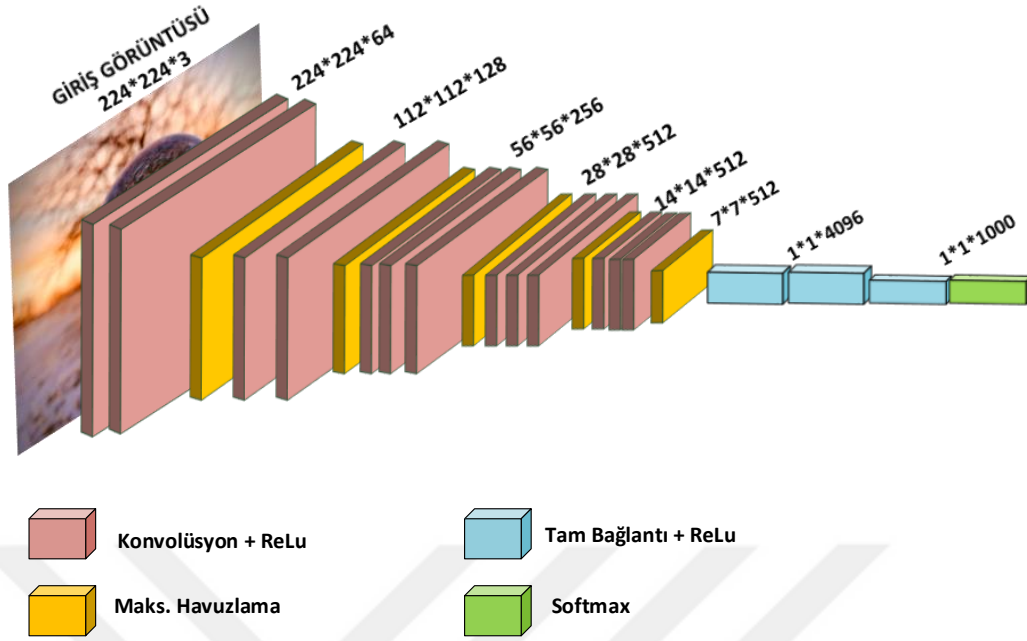
Test ve Doğrulama verisi oluşturma ön işlemenin son adımı olan doğrulama ve test verilerini oluşturma kısmında eğitilecek veri içerisinde %20 oranında veri ayrılır. Bu verinin içerisinde %5 oranında veri test verisi olarak ayrılır. Geri kalan %15 oranında ki veri ise doğrulama verisi olarak kullanılmaktadır. Bu veri setinin ön işleme adımının sonuncusudur. Buranın çıktısı olarak elde edilen görüntüler derin öğrenme ağına kullanılmaya hazırdır.

4.2. Nesne Sınıflandırma

Çalışmanın bu aşamasında mevcut olan iki veri kümesi ile sınıflandırma çalışmaları yapılmıştır. Sınıflandırma için VGG-16 modeli kullanılmış olup, bu model üzerinde birleşik toplu sinir ağı, ince ayarlamalı VGG-16 ve transfer öğrenme yani daha önceden eğitilmiş bir model üzerinde deneysel çalışmalar yapılmıştır. Aşağıda alt başlıklar halinde VGG-16 modeline, toplu sinir ağlarına ve fine-tuning (ince ayarlama) hakkında bilgiler paylaşılacaktır.

4.2.1. VGG-16

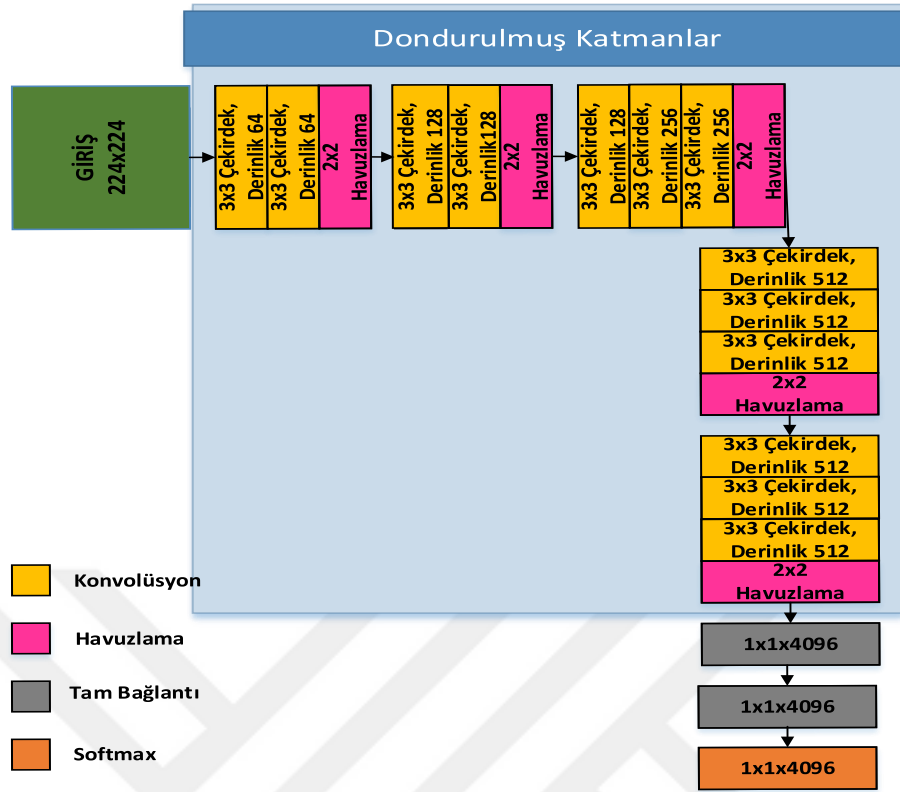
Visual Geometry Group 16(VGG-16), ImageNet yarışmasında 2014 yılında büyük bir başarı göstererek birinci olan bir modeldir. Bugüne kadar mükemmel görüntü modeli mimarisinden biri olarak kabul edilmektedir. Burada kullanılan 16 sayılı 16 katmandan oluştuğundan dolayıdır. Aşağıda Şekil 4.9 üzerinde VGG-16 mimarisi sunulmaktadır. VGG-16'nın temel farkı ise karmaşık parametrelerden kaçınıp tamamen saf bir şekilde evrişim katmanlarında 3*3 filtreler, maksimum havuzlamada 2*2 filtreler kullanmalarıdır. Tam bağlantı katmanında 2 katmandan oluşmaktadır ve bir adet softmax(yumuşak) katman kullanmaktadır. 138 Milyona yaklaşan parametresi ile büyük bir ağ örneğidir. Bu sebepten ötürü büyük ama saf belirli bir kalıpta ilerleyen mimariye sahiptir. Şekil 4.9'da VGG-16 mimarisi sunulmaktadır.



Şekil 28.9. VGG-16 Modeli

4.2.2. Transfer Öğrenme Tabanlı VGG-16

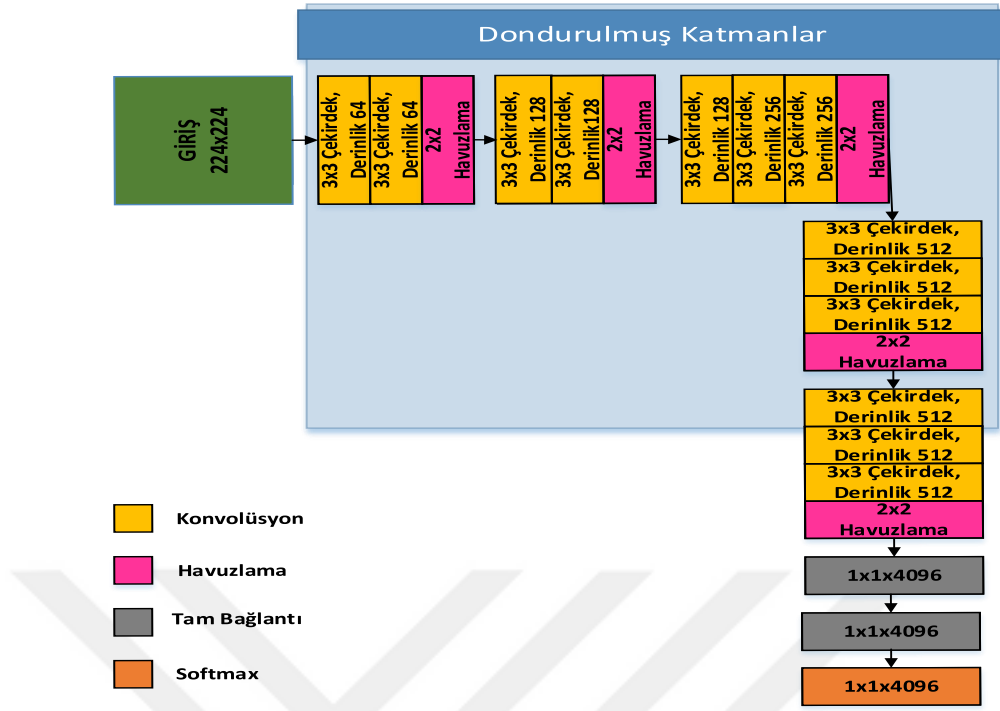
Transfer öğrenme VGG-16 tabanlı, ImageNet ağırlıkları üzerinde eğitilmiş VGG-16 önceden eğitilmiş modeli, özellikleri elde etmek ve bunları yeni veri setimize uygulamak için kullanılır. Kullanılan modelin mimarisi Şekil 4.10'deki blok diyagramda gösterilmektedir. Girdi olarak alınan veriler, transfer öğrenme yoluyla önceden eğitilmiş bir modelde ağırlıklanır. Girilen veriler ağırlık değerlerine göre sınıflandırılır.



Şekil 29.10. Transfer Öğrenme Tabanlı VGG-16

4.2.3. İnce Ayarlama Tabanlı VGG-16

VGG-16'da ince ayar yapılması önerilen modeldir. Bu modelde transfer öğrenmede olduğu gibi tüm katmanlar dondurulmamıştır. Şekil 4.11'da görüldüğü gibi 7x7x512 model katmanından giriş görüntüleri sağlanarak transfer öğrenme sağlanır. VGG-16 ile Transfer öğrenme arasında hibrit bir yapı oluşturulduğu da söylenebilir. Bu hibrit yapı tüm öznetelik çıkarımlarını yapmadığından maliyet açısından zaman kazandırırken transfer öğrenmede olduğu gibi sınıflandırma doğrudan ağırlıklı değildir. Veri setini son katmanlarda eğiterek ağırlıklar yeni değerlerine güncellenir. Böylece daha sağlıklı bir sınıflandırma sağlanmıştır. Bu şekilde alınan bir giriş görüntüsünün katmanları dondurulmuş transfer öğrenme eğitimine alınmadan önce giriş görüntüsünün boyutları yeniden düzenlenerek ara besleme katmanına yani 14x14x512 katmanına sisteme girilir. Bu katmandan transfer, öğrenmeden aldığı değeri güncelleyerek sınıflandırma katmanına geçer. Böylece daha sağlıklı bir sınıflandırma sağlanmaktadır.



Şekil 30.11. VGG-16 Modeli üzerinde İnce Ayarlama

Ayrıca modelin sinir ağı yapısı Tablo 4.1. üzerinde paylaşılmıştır.

Tablo 4.1. Önerilen Yönteme Ait Ağ Yapısı

Önerilen model yapısı	Şekil	Parametre
Layer	Output shape	Param
input_4 (InputLayer)	(None, 14, 14, 512)	0
block5_conv1 (Conv2D)	multiple	2359808
block5_conv2 (Conv2D)	multiple	2359808
block5_conv3 (Conv2D)	multiple	2359808
block5_pool (MaxPooling2D)	multiple	0
conv2d_1 (Conv2D)	(None, 5, 5, 64)	294976
max_pooling2d_1 (MaxPooling2D)	(None, 2, 2, 64)	0
flatten_1 (Flatten)	(None, 256)	0
dense_1 (Dense)	(None, 100)	25700
dense_2 (Dense)	(None, 6)	606
Total params: 7,400,706		
Trainable params: 7,400,706		
Non-trainable params: 0		

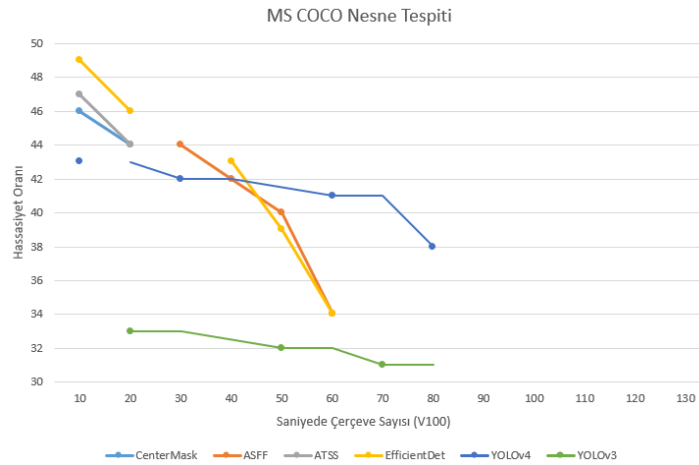
4.3. Nesne Tespiti

Bu bölümde derin öğrenme tabanlı nesne algılama yöntemleri için kullanılan yöntemlerin temel çalışma prensibi anlatılacaktır. Bu amaçla YOLO'nun farklı versiyonlarının arka planı verilerek çalışma prensipleri detaylandırılmıştır.

4.3.1. YOLO –V4

ESA tabanlı bir nesne detektörü olan YOLO-v4, ilk olarak 2020'de Bochkovskiy ve diğerleri [94] tarafından önerildi. YOLOv4 modelinin temel geliştirme amacı, gerçek hayata uygulanabilir bir modelin başarısını daha da artırmaktır. Nesne detektörlerinin omurgaları, GPU ve CPU platformunda farklılık gösterir. GPU platformunda detektörler için VGG [95], DenseNet [96] veya ResNet [97] gibi modeller kullanılabilir. MobileNet [98], SqueezeNet [99] veya ShuffleNet [100], CPU platformunda çalışan detektörler için kullanılabilir. YOLO-v4, tek aşamalı bir nesne detektörüdür ve çalışma stratejisi SSD [101] ve RetinaNet'e benzer. YOLO-v4 mimarisinin omurgası, CSPDarknet53, ek modül olarak SPP, yol toplama boynu olarak PaNet ve YOLOv3 çapa tabanlı kafadan oluşur. Şekil 4.12, YOLOv4'ün mimarisini temsil etmektedir.

YOLOv4 mimarisinin COCO veri setindeki diğer modellerle karşılaştırılması Şekil 4.13'de gösterilmektedir.



Şekil 31.13. YOLOv4 modelinin COCO veri setindeki diğer modellerle karşılaştırılması

Altı farklı model üzerinde yapılan karşılaştırmalar Şekil 4.13'de sunulmaktadır. Adaptive Training Sample Selection yani uyarlanabilir eğitim örnek seçimi (ATSS), nesnenin istatistiksel özelliklerine dayalı olarak pozitif ve negatif örneklerin otomatik olarak seçilmesi yöntemidir.

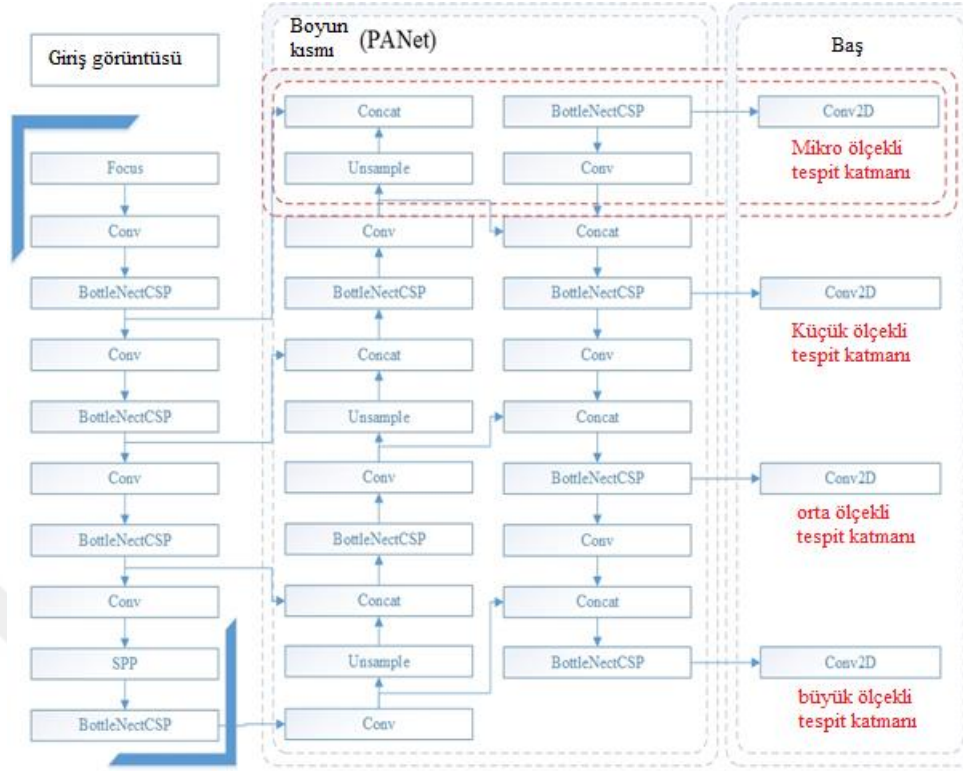
ATSS, YOLOv4 kadar başarılı olmasına rağmen, işletme maliyeti açısından YOLOv4'ün neredeyse iki katı pahalıdır. YOLOv3, işletme maliyeti açısından YOLOv4 ile aynı değerlere sahip olmasına rağmen, nesneleri algılamadaki başarısının YOLOv4'ten daha az olduğu görülmektedir.

EfficientDet için yapılan karşılaştırmalarda artan nesne algılama oranının işletme maliyetini de arttırdığı söylenebilir. Bu başarı açısından YOLOv4 modeline en yakın olan EfficientDet versiyonunun kıyaslandığında oldukça maliyetli olduğu gözlemlendi. Adaptive Spatial Feature Fusion (ASFF) ve Centermask modelleri ise, YOLOv4'e kıyasla yüksek işlem maliyetlerine sahiptir, bu nedenle Şekil 4.13'de kullanılacak ideal modelin YOLOv4 olduğu önerilmektedir.

4.3.2. YOLO –V5

YOLOv5 modeli, Ultralytics'in kurucusu Glenn Jocher tarafından yayınlandı [102]. Modelin en önemli özelliklerinden biri de ilk defa bir YOLO modelinin Pytorch üzerinde çalıştırılabilmesidir. YOLOv5 mimarisi, YOLOv4'teki gibi üç bölümden oluşur. Omurgada giriş görüntüleri 640x640x3 çözünürlüğe odaklanır, dilimleme ile çözünürlük 320x320x12 olur ve ardından 32 konvolüsyondan sonra 320x320x32 özellik haritası oluşturulur.

CBL (Convolution, Batch Normalization, Leaky—ReLU) modülü evrişim, normalleştirme ve LeakyReLU'yu temsil eder. BottleneckCSP modülü ise özellik haritaları üzerinden özellik çıkarımı yaparak görüntülerden bilgi çıkarır. SPP (Spatial Pyramid Pooling) modülü esas olarak ağırlıklı alanı artırır ve farklı ölçeklerde özellikler kazanır. YOLOv5 mimarisi Şekil 4.14'te gösterilmektedir.

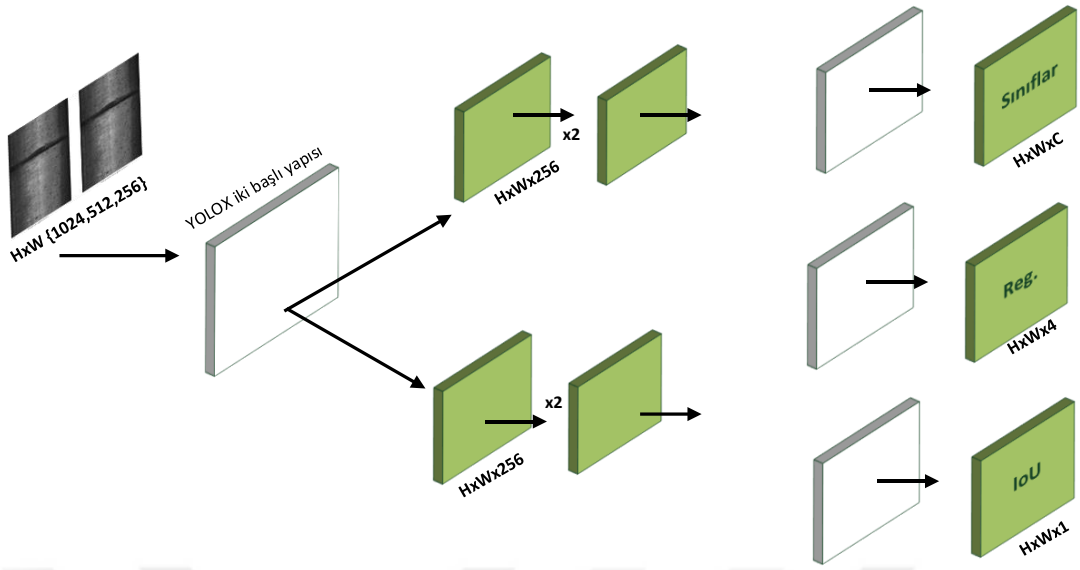


Şekil 32.14. YOLOv5 mimarisi

YOLOv5 mimarisi 4 farklı kafa yapısına sahiptir. Bu yapılar, algılanacak nesnenin toplam görüntüye oranını temsil eder. Görüntülerde algılanacak nesne genel görüntüye göre çok küçük bir alanı kaplıyorsa buna uygun kafa yapısı mikro ölçeklidir. Ancak tespit edilecek nesne geniş bir alanı kaplıyorsa Büyük kafa yapısı kullanılır. NEU veri setinde algılanacak nesneler görüntüye göre çok az yer kapladığı için mikro ölçek kafa yapısı kullanılmıştır.

4.3.3. YOLOX

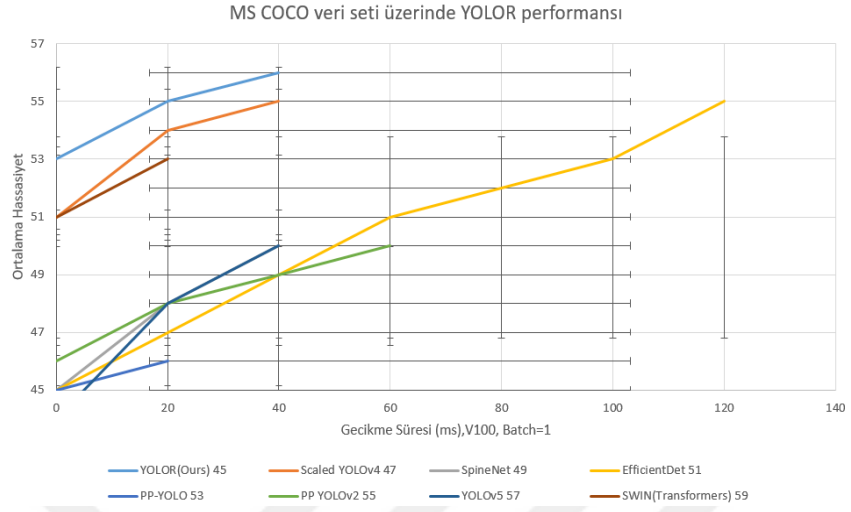
Ge ve diğ. [103] tarafından önerilen Exceeding YOLO (YOLOX), YOLO modellerinin en son versiyonudur. Diğer YOLO modellerinden farklı olarak YOLOX modeli çalışırken bir çapaya ihtiyaç duymaz. YOLOX'un diğer tespit modellerinden en büyük farklarından biri gelişmiş tespit tekniği kullanmasıdır. Şekil 4.15, YOLOX modeli için geliştirilmiş bölünmüş kafa yapısını göstermektedir.



Şekil 33.15. YOLOX mimarisi

4.3.4. YOLOR

Wang ve diğerleri [104] tarafından önerilen You Only Learn One Representation (YOLOR) mimarisi, çeşitli görevleri gerçekleştirmek için birleşik bir ağ yapısından oluşur. Modelin performansını etkin bir şekilde iyileştirmek için eğitim maliyetinin yaklaşık 1/10000'i kadar ek bir maliyet gerekir. Araştırmacıların YOLOR modeli üzerinde yaptıkları çalışmalardan elde ettikleri sonuçlara göre, YOLOR modeli COCO veri setinde YOLOv4'ten yüzde 88 daha hızlı ve YOLOv5'ten daha başarılı. Modelin COCO veri kümesindeki karşılaştırmaları Şekil 4.16'da sunulmuştur.



Şekil 34.16. COCO veri setinde YOLOR modelinin karşılaştırılması

MSCOCO veri seti üzerinde yapılan testlerde yazarlar [105], YOLOR modelinin ürettiği için avantajlı olduğunu savunmuşlardır. Diğer modellere göre daha hızlı ve daha başarılı sonuçlar ürettiği görüşünü ileri sürdüler. Şekil 4.16 YOLOR modelinin hem başarı oranı açısından daha performanslı olduğu ileri sürülmüştür. Karşılaştırıldığında diğer modellere kıyasla hızlıdır.

5. BULGULAR VE TARTIŞMA

5.1. Sınıflandırma İçin Elde Edilen Bulgular

Bu bölümde önerilen ince ayar VGG-16 modelinin sonuçları, çalışmada bahsedilen iki farklı veri seti için analiz edilmiştir. Ayrıca çalışmada uygulanan diğer modellerle karşılaştırmalı olarak ele alınacaktır.

5.1.1. NEU Veri Setinden Elde Edilen Bulgular

NEU veri setindeki işlemler ve sonuçlar tartışılacak ve analiz edilecektir. NEU karşılaştırma veri setinde yer alan görüntülerde herhangi bir gürültü olmadığı için sadece veri büyütme işlemi gerçekleştirilmiştir. Bu veri seti 3 farklı model ile eğitilmiştir. Bunların en düşük sonucu ESA ile elde edilmiştir. Eğitim sonucunda parti büyüklüğü parametresine bağlı olarak elde edilen en yüksek başarı %85,94'tür.

Parti büyüklüğünün model üzerindeki etkisini daha iyi anlamak için yapılan bir diğer çalışma ise aynı eğitimin farklı parti büyüklüğü parametre değerleri ile tekrarlanmasıdır. Bu parametrenin model başarısı üzerindeki etkileri Tablo 5.1'de sunulmuştur.

Tablo 5.1. Parti Boyutunun NEU Veri Seti İçin Modellerin Doğruluğu Üzerine Etkisi

Parti Boyutu	ESA Doğruluk (%)	Transfer Öğrenme Tabanlı VGG-16 Accuracy (%)	Önerilen Yöntem Doğruluk (%)
8	85,94	96,94	99,27
16	83,22	94,82	99,27
32	82,18	93,21	99,22
64	80,17	90,67	99,22

Tablo 5.1'den de anlaşılacağı gibi, ESA modeli artan parti boyutu ile ters orantılı bir doğruluk sunmaktadır. %85,94 gibi yüksek bir sonuç olsa da demiryolları insanlar için önemli bir konumda olduğu için yetersizdir. ESA yerine VGG-16 modeline dayalı Transfer öğrenmemizi kullandığımızda daha etkili olduğu gözlemlenmiştir. Modelin aynı veri seti üzerindeki en iyi başarı oranı %96,94'tür. Transfer öğrenme tabanlı VGG-16'da parti büyüklüğü başarı ile ters orantılı bir etki göstermekte ve başarıyı yüksek oranda etkilemektedir. Tablo 5.1'de elde edilen sonuçlar sunulmuştur.

Sonucu daha da iyileştirmek için transfer öğrenme tabanlı VGG-16 modelinde ince ayarlamalar yapılmıştır ve sonuçlarda iyi derecede gelişmeler gözlemlenmiştir. Transfer öğrenmede, girdi verilerinin donmuş 7x7x512 katmanına beslenmesi ile modelimizin başarısı

%99,27'ye kadar yükseltilmiş ve parti boyutunun bu model üzerindeki etkisi önceki modellere göre azaltılmış hatta yok sayılmıştır.

5.1.2. FUB Veri Setinden Elde Edilen Bulgular

Bu bölümde, tarafımızca oluşturulan veri setinin analizinden bahsedilmiştir. Genel olarak incelendiğinde, doğal koşullarda çekilen ve sınıflandırma kategorileri NEU veri setinden daha zor olan bu resimlerin oluşturduğu veri setinden elde edilen sonuç gerçekten başarılıdır. Veri setindeki görüntülerin bir kısmı gece ortamında, bir kısmı ise aydınlık ortamda elde edilmiştir. Buradaki amaç, modelimizin başarı oranını her koşulda mevcut olan görüntü kümesine karşı zorlamaktır. Veri setimize uygulanan ESA'nın %82,27'sidir. NEU veri kümesine kıyasla düşük sonuç, doğal koşullar altında elde edilen görüntülerin kullanılmasına ve veri görüntü kümemizdeki parti boyutunun daha düşük olmasına bağlanabilir. NEU veri setinde olduğu gibi, parti büyüklüğü ve 3 ayrı model karşılaştırması sunulacaktır. Modelimiz veri setindeki en yüksek başarı oranını %98,33 ile elde etmiştir. Veri setinin detayları Tablo 5.2'de gösterilmektedir.

Tablo 5.2. Parti Boyutunun FUB Veri Seti İçin Modellerin Doğruluğu Üzerine Etkisi

Parti Boyutu	ESA Doğruluk (%)	Transfer Öğrenme Tabanlı VGG-16 Accuracy (%)	Önerilen Yöntem Doğruluk (%)
8	82,17	93,14	98,33
16	81,50	92,33	98,22
32	80,33	91,89	98,22
64	80,17	90,67	98,22

Tablo 5.2'de de görülebileceği gibi, modelimiz yeni bir veri kümesine karşı iyi sonuçlar üretebilir. Parti büyüklüğü parametresine bakıldığında diğer modellerde başarıda düşüş olurken bizim modelimizde fark edilmeyecek bir düşüş mevcuttur.

Sınıflandırma çalışmalarının tamamı i7-6700HQ işlemci, 8 GB Ram ve NVIDIA 950M ekran kartına sahip bireysel bir bilgisayar aracılığı ile yapılmıştır.

5.1.3. Sınıflandırma Çalışması İçin Genel Bakış

Aşağıda bulunan Tablo 5.3 üzerinde veri kümelerin modellerine göre parametre değerleriyle performansını paylaşmaktadır.

Tablo 5.3. Veri Kümelerinin Sınıflandırma Sonuçları

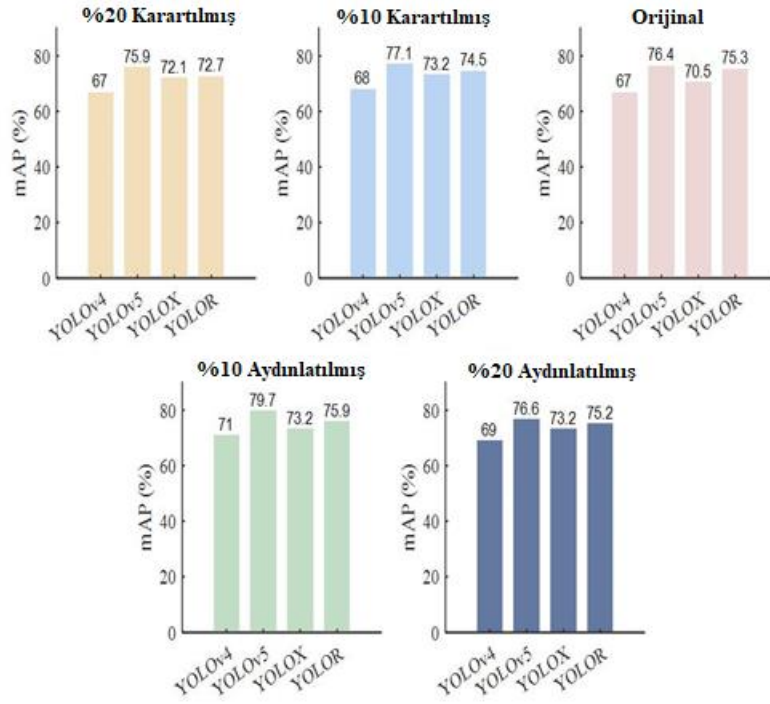
Modeller	Performans Ölçütü	Modeller	Performans Ölçütü	Modeller	Performans Ölçütü	Modeller
	Toplam Epoch	Parti Boyutu	Herbir Epoch Eğitim Süresi	Doğruluk (%)		Toplam Epoch
	FUB	NEU	FUB	NEU		FUB
ESA	50	8	24 s	55 s	82,17	85,94
		16	21 s	38 s	81,50	83,22
		32	18	24 s	80,33	82,18
		64	17 s	17 s	80,17	80,17
TLVGG-16		8	2 s	6 s	93,14	96,47
		16	2 s	3 s	92,33	94,82
		32	1 s	2 s	91,89	93,21
		64	1 s	1 s	90,67	90,67
Önerilen		8	43	98 s	98,33	99,27
		16	25 s	76 s	98,22	99,27
		32	19 s	51 s	98,22	99,22
		64	18 s	38 s	98,22	99,22
ESA	50	8	24 s	55 s	82,17	85,94
		16	21 s	38 s	81,50	83,22

Tablo 5.3, değişken parti büyüklüğü değerlerine göre aynı koşullarda 3 modelin başarı oranlarını göstermektedir. İlk iki modelde, parti büyüklüğü değeri arttıkça doğruluk azalır ve zaman performansındaki iyileşme tatmin edici değildir, ancak son modelde, parti büyüklüğü değerine bağlı olarak doğruluktaki küçük sapmalar ve buna karşılık gelen yüksek performans değerleri, bu modelin çelik yüzeylere ve tren raylarına verdiği zarar. Tanımlama ve sınıflandırmada etkili bir model olduğunu göstermektedir. Modelin en büyük avantajı optimizasyondur. Kesinlikte neredeyse hiç kayıp olmayan bu model, büyük başarı elde etmenin yanı sıra eğitim süresi açısından da büyük avantajlar sağladı.

5.2. Nesne Tespiti İçin Elde Edilen Bulgular

Çalışmada ışığın veri seti üzerindeki etkisini anlamak için dört farklı model kullanılmıştır. Bu modeller sırasıyla tartışılacak ve her birinin hangi veri setinde en iyi sonucu verdiği analiz edilecektir. Ayrıca her modelin en iyi başarı oranı bölümün sonunda sunulacaktır. Her modelin eğitimi sırasında iterasyon sayısı, öğrenme oranı, parti büyüklüğü ve alt bölüm değerleri gibi

parametreler sırasıyla 12000, 0,001, 64, 24 olarak seçilmiştir. Her modelin performansı Şekil 5.1’’de verilmiştir.

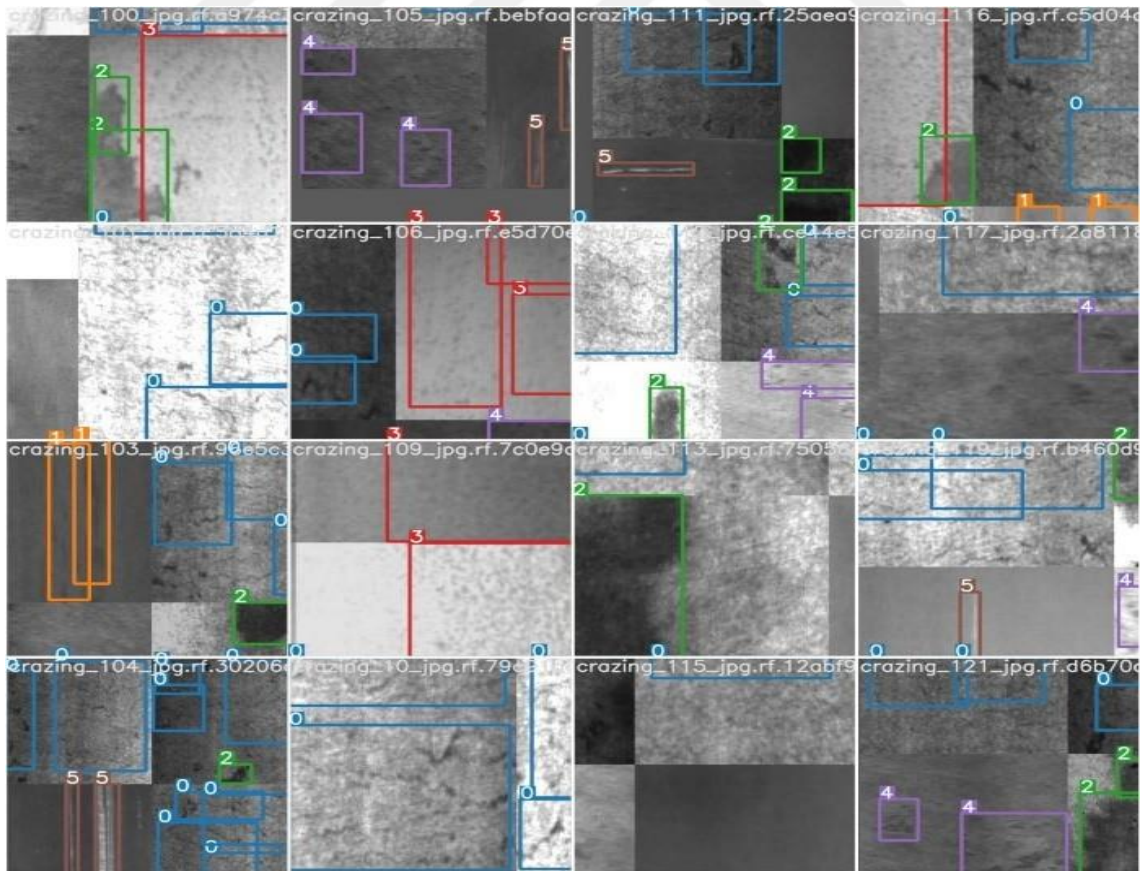


Şekil 35.1. COCO veri setinde YOLOR modelinin karşılaştırılması.

Şekil 5.1’de görüldüğü gibi veri setlerindeki parlaklık oranları modellerin başarı oranlarına etki etmektedir. Parlaklığın modeli olumlu yönde geliştirdiği deneysel çalışmada %10 aydınlatılmış veri setlerinde tüm modellerin daha iyi sonuç verdiği gözlemlenmiştir. YOLOv4 modeli 416x416 görüntü alır, ancak veri seti 200x200 görüntüden oluşur.

Görüntüler 416x416 boyutuna dönüştürülür, eğitim aşamasına geçer Işık değerleri değişen beş farklı veri kümesinden %20 karartma ile veri kümesinin en iyi başarı oranı %67, karartma oranı ile veri kümesinin en iyi başarı oranı %67’dir. %10, %68’dir. YOLOv4 için veri setinde yapılacak karartmanın neredeyse modelin başarısını etkilemediği gözlemlenmiştir. Orijinal veri setinin başarısı %67’dir. Aydınlatılan veri setlerinde başarı oranının olumlu yönde değiştiği gözlenmektedir. %10 aydınlatılmış veri seti, %71 mAP ile en iyi sonuçları verir. %20 aydınlatmalı veri seti, karartma ile orijinal veri setinden daha iyi sonuçlar vermesine rağmen %10 aydınlatmalı veri setinden daha kötü sonuçlar vermiştir. En başarılı veri seti %10 uygulanan veri setidir. YOLOv4 algoritmasından sonra ortaya çıkan YOLOv5 modeli daha gelişmiş bir modeldir. Aynı parametreler kullanılarak veri setlerine beş farklı eğitim uygulanmıştır. Bu eğitimlerde %20

karartma ile veri setinin en iyi başarı oranı %75.9, %10 karartma ile veri setinin en iyi başarı oranı %77,1'dir. YOLOv5 için veri setinde koyulaşmanın YOLOv4'te olduğu gibi modelin başarısını olumsuz etkilediği gözlemlenmiştir. Orijinal veri setinin başarıları %76.4'tür. Aydınlatılmış veri setlerinde başarı oranının YOLOv4'te olduğu gibi olumlu yönde arttığı gözlenmektedir. %10 aydınlatılmış veri seti, %79.7 mAP ile en iyi sonuçları verir. %20 aydınlatmalı veri seti, karartma ile orijinal veri setinden daha iyi sonuçlar vermesine rağmen %10 aydınlatmalı veri setinden daha kötü sonuçlar vermektedir. En başarılı veri seti %10 uygulanan veri setidir. Çalışmada kullanılan YOLOX modeli, [106]'te belirtilen varsayılan parametreler olarak çalıştırılmıştır. Diğer modellere göre daha kısa sürede daha yüksek başarı oranları elde etmesine rağmen YOLOv5 kadar başarı elde edemedi. Çalışmadaki son model YOLOR, 500 yinelemede çalıştırıldı. 500 iterasyondan sonra, tüm veri setlerinde iyileştirme sağlanamadığı için yinelemeler geliştirilmemiştir. Diğer modellerde olduğu gibi parametreler eşdeğer olarak ayarlanmış ancak YOLOv5 kadar başarılı sonuçlar vermemiştir. Aydınlatma elemanı ile değerlendirildiğinde %20 karartmalı veri seti diğer modellerde olduğu gibi %72.7 başarı elde etmiş ve en başarısız eğitim sonucunu vermiştir. Şekil 5.2, en iyi performans gösteren YoloV5'in algılama sonuçlarını gösterir.



Şekil 36.2. Yolov5'in kusur tespit sonuçları

Şekil 5.1'de gösterildiği gibi, aynı görüntü üzerinde farklı kusurların başarıyla tespit edildiği görülebilir. Orijinal veri seti ve %10 aydınlatma sonrası başarı oranı Tablo 5.4'de verilmiştir. Tablo 5.4'de modellerin performans oranları hakkında da bilgi verilmiştir.

Tablo 5.4. En iyi sonucu algılanan veri kümesi

Model Adı	Boyut	Parametre(M)	Katmanlar	GFLOPS's	Gecikme (ms)	FPS	En iyi Veri Sonuç(mAP%)	Geliştirme Oranı
YOLOv4	608x608	27.6	137	52	44	22.7	71.00%	4%
YOLOX	640x640	8.94	286	26.65	33	30.3	73.20%	2.7%
YOLOR	416x416	36.8	665	80.44	22	46	75.90%	0.60%
YOLOv5	416x416	7.3	232	16.8	17	58.8	79.70%	3.30%

Tablo 5.4'de gösterildiği gibi, modeller arasında performans ve başarı oranı farklılıkları vardır. Bu çalışma Tesla P100 ekran kartına ve 16 GB RAM'e sahip bir sunucu üzerinde gerçekleştirilmiştir. Bu tablodan elde edilen sonuçlar, YOLOv5 mimarisinin daha önce metal yüzeyler alanında kullanılmayan en yeni YOLOR ve YOLOX mimarilerinden daha iyi performans gösterdiğini ve daha başarılı olduğunu göstermektedir. YOLOX ve YOLOR mimarileri YOLOv4'ten daha iyi sonuçlar verdiğinde, deneysel sonuçların bir çıktısı olarak görülmektedir. Aydınlatılmış veri setinde en az iyileştirme oranını YOLOR modeli gösterirken, en yüksek farkı YOLOv4 modeli göstermiştir, ancak YOLO modelleri arasında çelik yüzeylerdeki kusurları tespit etmek için en uygun model YOLOv5 modelidir. NEU veri seti üzerinde son zamanlarda yapılan hata tespit çalışmaları incelendiğinde, daha önce YOLOX, YOLOR ve YOLOv5 modellerinin kullanıldığı ve aydınlatmanın başarıya etkisini inceleyen herhangi bir çalışmaya rastlanmamıştır. Ayrıca çalışma, YOLOX ve YOLOR modellerini kullanan ilk metal yüzey kusur tespiti çalışmasının benzersizliğine sahiptir. 2022'de Tian ve ark. [107] Metal yüzeylerde hızlı ve doğru sonuçlar almak için DCC-CenterNet modelini önerdiler. Modelin omurga yapısı olarak ResNet50 tercih edilmiş ve DFEM (Dilate feature Enhanced model) kafa yapısı kullanılmıştır. 2021'de Kou ve ark. [108] NEU veri setinde hasar tespiti için YOLOv3 tabanlı bir model geliştirdiler. Modeli geliştirmek için öznetelik seçimi uygulayan çalışmalar %72.2 mAP elde etti. 2021'de Cheng ve Yu, [109] DEA_RetinaNet modelini metal yüzeylerde kullandı. Temel olarak beş bölümden oluşan bu model, öznetelik çıkarma ağı, DE-blok, FPN, uyarlamalı uzaysal özellik birleştirme (ASFF) modülü ve tahmin ağı modüllerinden oluşmaktadır. Yazarlar çalışma sonucunda %78,25 mAP elde etmişlerdir. 2021 yılında Song ve ark.[110] metal yüzeylerin kusurlarını tespit etmek için çok ölçekli çekişmeli ve ağırlıklı gradyan etki alanı uyarlamalı ağ (MWDAN) modelini kullandı. Bu model, HRNet'in [111] daha hızlı R-CNN modeline omurga olarak uygulanmasıyla elde edilir.

Araştırmacılar bu çalışmada %76,2'lik bir başarı oranı elde ettiler. Karşılaştırma sonuçları Tablo 5.5'de verilmiştir.

Tablo 5.5. Karşılaştırma Tablosu

Referans	Method	FPS	Parametre (M)	mAP(%)
[41]	DCC-CenterNet	71.37	32.81	79.41
[42]	YOLO-V3	64,5	60,7	72,2
[43]	DEA_RetinaNet+ASFF	12	28,5	78,25
Bu Çalışma	MWDAN			
	YOLOv4 Darknet + (10% aydınlatılmış)	22,7	27,6	71,00%
	YOLOR + (10% aydınlatılmış)	46	36,8 M	75,90%
	YOLOX + (10% aydınlatılmış)	30,3	8,94 M	73,20%
	YOLOv5 Darknet + (10% aydınlatılmış)	58,8	7,3	79,70%

Tablo 5.5'de bahsedilen YOLO modeline dayalı nesne detektörlerinin NEU veri setinde YOLOv5 modeline göre daha az başarılı olduğu ve daha fazla parametreye sahip olduğu görülmektedir. Ayrıca bu çalışmalarda aydınlatma ve karartmanın etkileri incelenmemiştir. Çalışma, modeller üzerindeki görüntü iyileştirme tekniklerinin modelin başarısına mutlak bir katkı sağladığını açıkça göstermektedir.

6. SONUÇLAR

Endüstrinin gelişmesi ve seri üretime geçişle birlikte malzemenin özellikleri, dayanıklılığı, işleme yöntemleri gibi faktörlerden dolayı metalik yüzeylerde kusurlar oluşabilmektedir. Bu kusurlar üretimdeki kaliteyi etkiler. Kaliteyi olumsuz etkileyen bu kusurlar endüstriyel malzemelerin kullanılacağı uygulama ve projeleri de etkilemektedir. Ayrıca endüstriyel ürünler yüksek maliyetli ürünlerdir. Arızalar zamanında tespit edilip onarılmadığında can kaybına neden olabilir. Bu nedenle endüstriyel ürünleri sürekli denetleyerek oluşabilecek kusurları erken aşamada tespit etmek kaliteli ürünlerin ortaya çıkmasını sağlamakla birlikte can ve mal güvenliğini de sağlayacaktır.

Bu çalışmada metal yüzeyler ile ilgili olarak iki farklı çalışma yapılmıştır. Çalışmalardan ilki sınıflandırma çalışmasıdır. Sınıflandırma çalışmasında VGG16 mimarisi üzerinde yalnızca son iki katmanın eğitime yeniden dâhil edilmesi ile transfer öğrenmedeki ağırlıklar yeni verilere göre güncellenmesi sağlanmıştır. VGG16 mimarisi üzerinde yapılan ince ayarlamalar aracılığıyla transfer öğrenme VGG-16 modelinden başarı olarak daha yüksek sonuçlar elde edilmiştir. İnce ayarlamalı modelde, müdahale edilmeden yapılan transfer öğrenmenin önüne geçtiğinde ise rasgele olarak oluşturulan 10 adet transfer öğrenmenin (grup sinir ağları), en iyi özelliklerinin birleştirilmesi ile oluşturulan yeni bir modelle karşılaştırılması sağlanmıştır. İnce ayarlamalı model bu karşılaştırmalardan daha iyi sonuçlar elde etmiştir. Sınıflandırma çalışmasında altı sınıf içeren iki farklı veri seti kullanılmıştır. Bu sayede değişken veri kümelerine karşı ve diğer modellere kıyasla başarıyı ölçülmüştür.

Sınıflandırma çalışmasında normal CNN modeline karşın çok daha yüksek sonuçlar elde edilmiş olup, performans açısından da daha verimli bir model ortaya koyulmuştur. Modelin NEU veri kümesi için başarı oranı %99'un üzerindeyken, tarafımızca oluşturulan FUB veri kümesinde başarı oranı %98'in üzerindedir. Sonuçlar değerlendirildiğinde oluşturulan modelin oldukça etkili olduğu anlaşılmaktadır. Ayrıca model farklı parametreler altında çalıştırılarak ideal çalışma değerleri belirlenmiştir. Parti boyutunun modele etkisi incelenmiş olup ideal değer 8 olarak belirlenmiştir.

İkinci yapılan çalışma ise arızaların lokalize edilmesine yöneliktir yani metal yüzey üzerinde bulunan arızanın bir sınırlayıcı kutu tarafından çevrelenerek, tespit edilmesini sağlamaktır. Bu çalışma sağlanırken çevresel bir etmen olan ışığın derin öğrenme nesne algılama modelleri üzerindeki etkisi incelenmiştir ve literatürdeki ışığın söz parametre olarak değerlendirmeyen diğer çalışmalarla farkı ortaya koyulmaya çalışılmıştır.

Çalışma esnasında model tercihi olarak dört farklı YOLO modeli ele alınmıştır. Bu modeller sırasıyla YOLOv4, YOLOv5, YOLOR, YOLOX modelleridir. Yaygın olarak kullanılan YOLOv4 ve YOLOv5 modellerinin yanında güncel olarak eklenen YOLOR ve YOLOX

modellerinin kıyaslanması daha önce literatürde gerçekleştirilmediği gibi ışık etkisi altında metal yüzeyler üzerinde başarılarının ölçülmesi ilk kez tarafımızca ele alınmaktadır.

Çalışma kapsamında farklı aydınlatma koşulları görüntü geliştirme algoritmaları ile sağlanmış olup %20 karartma, %10 karartma, orijinal, %10 aydınlatma, %20 aydınlatma olacak şekilde toplamda beş farklı veri kümesi oluşturulmuş olup bu dört modelde çalıştırılması sağlanmıştır. 20 eğitimin sonucunda NEU veri kümesi için %10 aydınlatma yapılan tüm veri kümelerinin sonuçlarının daha iyi çıktığı ayrı ayrı tüm YOLO modelleri üzerinde kanıtlanmıştır. Bu modellerin kendi içerisinde ayrılan başarılarına göre ise YOLOv5 mimarisinin diğer üç YOLO modeline göre hem daha hızlı, masrafsız ve başarılı sonuçlar ürettiği deneysel sonuçlar neticesinde görülmüştür. ikinci çalışma için deneysel verilerden elde edilen verilere göre, yeterli ışık altında elde edilen veri kümeleri daha iyi sonuçlar üretmektedir.

Sonuç olarak bu tez kapsamında yapılan iki çalışma sonucunda metal yüzeylerde bulunan arızaların nasıl daha iyi sınıflandırılacağı ve bu arızaların nasıl daha iyi tespit edileceği yönünde bilgiler sunulmuştur.

KAYNAKLAR

- [1] Yan, L. C., Yoshua, B., & Geoffrey, H. (2015). Deep learning. *nature*, 521(7553), 436-444.
- [2] Doğan, F., & Türkoğlu, İ. (2019). Derin öğrenme modelleri ve uygulama alanlarına ilişkin bir derleme. *Dicle Üniversitesi Mühendislik Fakültesi Mühendislik Dergisi*, 10(2), 409-445.
- [3] S. R. avikumar, K.I. Ramachandran, V. Sugumaran Machine learning approach for automated visual inspection of machine components, *Expert Syst Appl*, 38 (4) (2011), pp. 3260-3266.
- [4] Cha, Y. J., Choi, W., Suh, G., Mahmoudkhani, S., & Büyüköztürk, O. (2018). Autonomous structural visual inspection using region-based deep learning for detecting multiple damage types. *Computer-Aided Civil and Infrastructure Engineering*, 33(9), 731-747.
- [5] Luckow, A., Cook, M., Ashcraft, N., Weill, E., Djerekarov, E., & Vorster, B. (2016, December). Deep learning in the automotive industry: Applications and tools. In *2016 IEEE International Conference on Big Data (Big Data)* (pp. 3759-3768). IEEE.
- [6] Q. Li and S. Ren, (2012), A visual detection system for rail surface defects, *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 42, no. 6, pp. 1531–1542.
- [7] Zhang, X., Feng, N., Wang, Y., Shen, Y., (2015), Acoustic emission detection of rail defect based on wavelet transform and Shannon entropy. *J. Sound Vib.* 339, 419–432.
- [8] Bruzelius, K.; Mba, D., (2004), An initial investigation on the potential applicability of Acoustic Emission to rail track fault detection. *NDT E Int.*, 37, 507–516.
- [9] Cheon, S., Lee, H., Kim, C. O., & Lee, S. H. (2019). Convolutional neural network for wafer surface defect classification and the detection of unknown defect class. *IEEE Transactions on Semiconductor Manufacturing*, 32(2), 163-170.
- [10] Di, H., Ke, X., Peng, Z., & Dongdong, Z. (2019). Surface defect classification of steels with a new semi-supervised learning method. *Optics and Lasers in Engineering*, 117, 40-48.
- [11] Ye, R., Pan, C. S., Chang, M., & Yu, Q. (2018). Intelligent defect classification system based on deep learning. *Advances in Mechanical Engineering*, 10(3), 1687814018766682.
- [12] Fu, G., Sun, P., Zhu, W., Yang, J., Cao, Y., Yang, M. Y., & Cao, Y. (2019). A deep-learning-based approach for fast and robust steel surface defects classification. *Optics and Lasers in Engineering*, 121, 397-405.
- [13] Wang, Y., Gao, L., Gao, Y., & Li, X. (2021). A new graph-based semi-supervised method for surface defect classification. *Robotics and Computer-Integrated Manufacturing*, 68, 102083.
- [14] Yun, J. P., Shin, W. C., Koo, G., Kim, M. S., Lee, C., & Lee, S. J. (2020). Automated defect inspection system for metal surfaces based on deep learning and data augmentation. *Journal of Manufacturing Systems*, 55, 317-324.
- [15] Kim, M. S., Park, T., & Park, P. (2019, June). Classification of steel surface defect using convolutional neural network with few images. In *2019 12th Asian Control Conference (ASCC)* (pp. 1398-1401). IEEE.
- [16] Chao, W. A. N. G., Liu, Y. T., Yang, Y. N., Xu, X. Y., & Zhang, T. (2019). Research on Classification of Surface Defects of Hot-rolled Steel Strip Based on Deep Learning. *DEStech Trans. Comput. Sci. Eng.*
- [17] Li, Y., Huang, H., Xie, Q., Yao, L., & Chen, Q. (2018). Research on a surface defect detection algorithm based on MobileNet-SSD. *Applied Sciences*, 8(9), 1678.
- [18] Neuhauser, F. M., Bachmann, G., & Hora, P. (2020). Surface defect classification and detection on extruded aluminum profiles using convolutional neural networks. *International Journal of Material Forming*, 13(4), 591-603.
- [19] Aydin, I., Akin, E., & Karakose, M. (2021). Defect classification based on deep features for railway tracks in sustainable transportation. *Applied Soft Computing*, 111, 107706..

- [20] Aydin, A., Salur, M. U., & Aydin, İ. (2021, July). Fine-Tuning Convolutional Neural Network Based Railway Damage Detection. In *IEEE EUROCON 2021-19th International Conference on Smart Technologies* (pp. 216-221). IEEE.
- [21] Yanan, S., Hui, Z., Li, L., & Hang, Z. (2018, November). Rail surface defect detection method based on YOLOv3 deep learning networks. In *2018 Chinese Automation Congress (CAC)* (pp. 1563-1568). IEEE..
- [22] Lv, X., Duan, F., Jiang, J. J., Fu, X., & Gan, L. (2020). Deep metallic surface defect detection: The new benchmark and detection network. *Sensors*, 20(6), 1562.
- [23] Xu, Y., Zhang, K., & Wang, L. (2021). Metal Surface Defect Detection Using Modified YOLO. *Algorithms*, 14(9), 257.
- [24] Li, J., Su, Z., Geng, J., & Yin, Y. (2018). Real-time detection of steel strip surface defects based on improved yolo detection network. *IFAC-PapersOnLine*, 51(21), 76-81.
- [25] Chen, X., An, Z., Huang, L., He, S., Zhang, X., & Lin, S. (2020, December). Surface Defect Detection of Electric Power Equipment in Substation Based on Improved YOLOV4 Algorithm. In *2020 10th International Conference on Power and Energy Systems (ICPES)* (pp. 256-261). IEEE.
- [26] Li, Y., Huang, H., Xie, Q., Yao, L., & Chen, Q. (2018). Research on a surface defect detection algorithm based on MobileNet-SSD. *Applied Sciences*, 8(9), 1678.
- [27] Lin, C. Y., Chen, C. H., Yang, C. Y., Akhyar, F., Hsu, C. Y., & Ng, H. F. (2019, July). Cascading convolutional neural network for steel surface defect detection. In *International Conference on Applied Human Factors and Ergonomics* (pp. 202-212). Springer, Cham.
- [28] Zhao, W., Chen, F., Huang, H., Li, D., & Cheng, W. (2021). A new steel defect detection algorithm based on deep learning. *Computational Intelligence and Neuroscience*, 2021..
- [29] Urbonas, A., Raudonis, V., Maskeliūnas, R., & Damaševičius, R. (2019). Automated identification of wood veneer surface defects using faster region-based convolutional neural network with data augmentation and transfer learning. *Applied Sciences*, 9(22), 4898.
- [30] Cheng, X., & Yu, J. (2020). RetinaNet with difference channel attention and adaptively spatial feature fusion for steel surface defect detection. *IEEE Transactions on Instrumentation and Measurement*, 70, 1-11..
- [31] Tsai, D. M., Fan, S. K. S., & Chou, Y. H. (2021). Auto-annotated deep segmentation for surface defect detection. *IEEE Transactions on Instrumentation and Measurement*, 70, 1-10..
- [32] Cao, J., Yang, G., & Yang, X. (2020). A pixel-level segmentation convolutional neural network based on deep feature fusion for surface defect detection. *IEEE Transactions on Instrumentation and Measurement*, 70, 1-12.
- [33] Usamentiaga, R., Lema, D. G., Pedrayes, O. D., & Daniel, G. (2022). Automated surface defect detection in metals: a comparative review of object detection and semantic segmentation using deep learning. *IEEE Transactions on Industry Applications*.
- [34] Yang, L., Fan, J., Huo, B., Li, E., & Liu, Y. (2022). A nondestructive automatic defect detection method with pixelwise segmentation. *Knowledge-Based Systems*, 242, 108338.
- [35] Liang, Z., Zhang, H., Liu, L., He, Z., & Zheng, K. (2018, July). Defect detection of rail surface with deep convolutional neural networks. In *2018 13th World Congress on Intelligent Control and Automation (WCICA)* (pp. 1317-1322). IEEE.
- [36] Kang, G., Gao, S., Yu, L., & Zhang, D. (2018). Deep architecture for high-speed railway insulator surface defect detection: Denoising autoencoder with multitask learning. *IEEE Transactions on Instrumentation and Measurement*, 68(8), 2679-2690..
- [37] Zhou, G., & Sun, H. (2020, June). Defect detection method for steel based on semantic segmentation. In *2020 IEEE 5th Information Technology and Mechatronics Engineering Conference (ITOEC)* (pp. 975-979). IEEE.

- [38] Lin, Q., Zhou, J., Ma, Q., Ma, Y., Kang, L., & Wang, J. (2022). EMRA-Net: A Pixel-Wise Network Fusing Local and Global Features for Tiny and Low-Contrast Surface Defect Detection. *IEEE Transactions on Instrumentation and Measurement*, 71, 1-14.
- [39] Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., ... & Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*.
- [40] Puarungroj, W., & Boonsirisumpun, N. (2018). Thai license plate recognition based on deep learning. *Procedia Computer Science*, 135, 214-221.
- [41] LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278-2324.
- [42] Lauzon, F. Q. (2012). An introduction to deep learning. In *2012 11th International Conference on Information Science, Signal Processing and their Applications (ISSPA)* (pp. 1438-1439). IEEE.
- [43] Socher, R., Huval, B., Bath, B., Manning, C. D., & Ng, A. (2012). Convolutional-recursive deep learning for 3d object classification. *Advances in neural information processing systems*, 25.
- [44] Q. V. Le, J. Ngiam, A. Coates, A. Lahiri, B. Prochnow ve A. Y. Ng, On optimization methods for deep learning, In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, pp. 265-272, 2011.
- [45] LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278-2324.
- [46] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... & Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1-9).
- [47] Liu, L., Shen, C., & Van den Hengel, A. (2015). The treasure beneath convolutional layers: Cross-convolutional-layer pooling for image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 4749-4757).
- [48] DOĞAN, F., & TÜRKOĞLU, İ. (2018). Derin öğrenme algoritmalarının yaprak sınıflandırma başarımlarının karşılaştırılması. *Sakarya University Journal of Computer and Information Sciences*, 1(1), 10-21.
- [49] Yang, J., Nguyen, M. N., San, P. P., Li, X. L., & Krishnaswamy, S. (2015, June). Deep convolutional neural networks on multichannel time series for human activity recognition. In *Twenty-fourth international joint conference on artificial intelligence*.
- [50] Castelluccio, M., Poggi, G., Sansone, C., & Verdoliva, L. (2015). Land use classification in remote sensing images by convolutional neural networks. *arXiv preprint arXiv:1508.00092*.
- [51] Adler, A., Elad, M., & Zibulevsky, M. (2016). Compressed Learning: A Deep Neural Network Approach. *arXiv preprint arXiv:1610.09615*.
- [52] Xiao, T., Li, H., Ouyang, W., & Wang, X. (2016). Learning deep feature representations with domain guided dropout for person re-identification. In *Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on IEEE*, (pp. 1249-1258)..
- [53] Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. R. (2012). Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*.
- [54] Ioffe, S., & Szegedy, C. (2015, June). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning* (pp. 448-456).
- [55] Tang, Y. (2013). Deep learning using linear support vector machines. *arXiv preprint arXiv:1306.0239*
- [56] Ramachandran, P., Zoph, B., & Le, Q. V. (2017). Searching for activation functions. *arXiv preprint arXiv:1710.05941*.

- [57] Memisevic, R., Zach, C., Pollefeys, M., & Hinton, G. E. (2010). Gated softmax classification. *Advances in neural information processing systems*, 23.
- [58] Xu, B., Wang, N., Chen, T., & Li, M. (2015). Empirical evaluation of rectified activations in convolutional network. *arXiv preprint arXiv:1505.00853*.
- [59] B. A. KARAKUŞ, *Derin öğrenme ve büyük veri yaklaşımları ile metin analizi*, Elazığ: Fırat Üniversitesi Doktora Tezi, 2018.
- [60] Ghosh, B., Dutta, I. K., Carlson, A., Totaro, M., & Bayoumi, M. (2020, October). An Empirical Analysis of Generative Adversarial Network Training Times with Varying Batch Sizes. In *2020 11th IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)* (pp. 0643-0648). IEEE.
- [61] Smith, S. L., Kindermans, P. J., Ying, C., & Le, Q. V. (2017). Don't decay the learning rate, increase the batch size. *arXiv preprint arXiv:1711.00489*.
- [62] D. Mishkin, N. Sergievskiy ve J. Matas, «Systematic evaluation of ESA advances, ArXiv:1606.02228., 2016.
- [63] Wilson, D. R., & Martinez, T. R. (2003). The general inefficiency of batch training for gradient descent learning. *Neural networks*, 16(10), 1429-1451. Stathakis, D. (2009). How many hidden layers and nodes. *International Journal of Remote Sensing*, 30(8), 2133-2147.
- [64] Apaydin, H., Feizi, H., Sattari, M. T., Colak, M. S., Shamshirband, S., & Chau, K. W. (2020). Comparative analysis of recurrent neural network architectures for reservoir inflow forecasting. *Water*, 12(5), 1500.
- [65] Liu, H., Huang, Y., Wang, Z., Liu, K., Hu, X., & Wang, W. (2019). Personality or value: A comparative study of psychographic segmentation based on an online review enhanced recommender system. *Applied sciences*, 9(10), 1992.
- [66] Ding, S., Su, C., & Yu, J. (2011). An optimizing BP neural network algorithm based on genetic algorithm. *Artificial intelligence review*, 36(2), 153-162.
- [67] Wang, Q., Ma, Y., Zhao, K., & Tian, Y. (2022). A comprehensive survey of loss functions in machine learning. *Annals of Data Science*, 9(2), 187-212.
- [68] Leistner, C., Saffari, A., Roth, P. M., & Bischof, H. (2009, September). On robustness of on-line boosting-a competitive study. In *2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops* (pp. 1362-1369). IEEE.
- [69] Leistner, C., Saffari, A., Roth, P. M., & Bischof, H. (2009, September). On robustness of on-line boosting-a competitive study. In *2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops* (pp. 1362-1369). IEEE.
- [70] Pan, S. J., & Yang, Q. (2009). A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10), 1345-1359.
- [71] Weiss, K., Khoshgoftaar, T. M., & Wang, D. (2016). A survey of transfer learning. *Journal of Big data*, 3(1), 1-40.
- [72] Al Hadhrami, E., Al Mufti, M., Taha, B., & Werghi, N. (2018, May). Transfer learning with convolutional neural networks for moving target classification with micro-Doppler radar spectrograms. In *2018 International Conference on Artificial Intelligence and Big Data (ICAIBD)* (pp. 148-154). IEEE.
- [73] Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., ... & Zheng, X. (2016). {TensorFlow}: A System for {Large-Scale} Machine Learning. In *12th USENIX symposium on operating systems design and implementation (OSDI 16)* (pp. 265-283).
- [74] Ketkar, N. (2017). Introduction to keras. In *Deep learning with Python* (pp. 97-111). Apress, Berkeley, CA.

- [75] Manaswi, N. K. (2018). Understanding and working with Keras. In *Deep Learning with Applications Using Python* (pp. 31-43). Apress, Berkeley, CA.
- [76] Collobert, R., Kavukcuoglu, K., & Farabet, C. (2011). Torch7: A matlab-like environment for machine learning. In *BigLearn, NIPS workshop* (No. CONF).
- [77] Bahrampour, S., Ramakrishnan, N., Schott, L., & Shah, M. (2016). Comparative study of caffe, neon, theano, and torch for deep learning.
- [78] Rush, A. M. (2020). Torch-struct: Deep structured prediction library. *arXiv preprint arXiv:2002.00876*.
- [79] Bergstra, J., Bastien, F., Breuleux, O., Lamblin, P., Pascanu, R., Delalleau, O., ... & Kaelbling, P. Theano: deep learning on gpus with python. NIPS 2011. In *BigLearning Workshop* (pp. 1-48).
- [80] Bergstra, J., Breuleux, O., Bastien, F., Lamblin, P., Pascanu, R., Desjardins, G., ... & Bengio, Y. (2010, June). Theano: A CPU and GPU math compiler in Python. In *Proc. 9th python in science conf* (Vol. 1, pp. 3-10).
- [81] Bergstra, J., Breuleux, O., Bastien, F., Lamblin, P., Pascanu, R., Desjardins, G., ... & Bengio, Y. (2010, June). Theano: A CPU and GPU math compiler in Python. In *Proc. 9th python in science conf* (Vol. 1, pp. 3-10).
- [82] Vedaldi, A., Lux, M., & Bertini, M. (2018). MatConvNet: CNNs are also for MATLAB users. *ACM SIGMultimedia Records*, 10(1), 9-9..
- [83] Liu, Y., Yuan, Y., Balta, C., & Liu, J. (2020). A light-weight deep-learning model with multi-scale features for steel surface defect classification. *Materials*, 13(20), 4629.
- [84] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., ... & Fei-Fei, L. (2015). Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3), 211-252.
- [85] Hernavs, J., Ficko, M., Berus, L., Rudolf, R., & Klančnik, S. (2018). Deep learning in industry 4.0—brief overview. *J Prod Eng*, 21(2), 1-5.
- [86] Girshick, R. (2015). Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision* (pp. 1440-1448).
- [87] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 779-788).
- [88] Song, K., & Yan, Y. (2013). A noise robust method based on completed local binary patterns for hot-rolled steel strip surface defects. *Applied Surface Science*, 285, 858-864.
- [89] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016, October). Ssd: Single shot multibox detector. In *European conference on computer vision* (pp. 21-37). Springer, Cham.
- [90] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 779-788).
- [91] Song, K., & Yan, Y. (2013). A noise robust method based on completed local binary patterns for hot-rolled steel strip surface defects. *Applied Surface Science*, 285, 858-864.
- [92] Otsu, N. (1979). A threshold selection method from gray-level histograms. *IEEE transactions on systems, man, and cybernetics*, 9(1), 62-66.
- [93] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.
- [94] Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. M. (2020). Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*.
- [95] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

- [96] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).
- [97] Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4700-4708).
- [98] Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., ... & Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*.
- [99] Iandola, F. N., Han, S., Moskewicz, M. W., Ashraf, K., Dally, W. J., & Keutzer, K. (2016). SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and < 0.5 MB model size. *arXiv preprint arXiv:1602.07360*.
- [100] Ma, N., Zhang, X., Zheng, H. T., & Sun, J. (2018). Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *Proceedings of the European conference on computer vision (ECCV)* (pp. 116-131).
- [101] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016, October). Ssd: Single shot multibox detector. In *European conference on computer vision* (pp. 21-37). Springer, Cham.
- [102] Fang, Y., Guo, X., Chen, K., Zhou, Z., & Ye, Q. (2021). Accurate and Automated Detection of Surface Knots on Sawn Timbers Using YOLO-V5 Model. *BioResources*, 16(3).
- [103] Ge, Z., Liu, S., Wang, F., Li, Z., & Sun, J. (2021). Yolox: Exceeding yolo series in 2021. *arXiv preprint arXiv:2107.08430*.
- [104] Wang, C. Y., Yeh, I. H., & Liao, H. Y. M. (2021). You only learn one representation: Unified network for multiple tasks. *arXiv preprint arXiv:2105.04206*.
- [105] Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., ... & Zitnick, C. L. (2014, September). Microsoft coco: Common objects in context. In *European conference on computer vision* (pp. 740-755). Springer, Cham.
- [106] Yolor, Github: <https://github.com/WongKinYiu/yolor>, Erişim:21.05.2022
- [107] Tian, R., & Jia, M. (2022). DCC-CenterNet: A rapid detection method for steel surface defects. *Measurement*, 187, 110211.
- [108] Kou, X., Liu, S., Cheng, K., & Qian, Y. (2021). Development of a YOLO-V3-based model for detecting defects on steel strip surface. *Measurement*, 182, 109454.
- [109] Cheng, X., & Yu, J. (2020). RetinaNet with difference channel attention and adaptively spatial feature fusion for steel surface defect detection. *IEEE Transactions on Instrumentation and Measurement*, 70, 1-11.
- [110] Song, Y., Liu, Z., Wang, J., Tang, R., Duan, G., & Tan, J. (2021). Multiscale Adversarial and Weighted Gradient Domain Adaptive Network for Data Scarcity Surface Defect Detection. *IEEE Transactions on Instrumentation and Measurement*, 70, 1-10..
- [111] Sun, K., Xiao, B., Liu, D., & Wang, J. (2019). Deep high-resolution representation learning for human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 5693-5703).

ÖZGEÇMİŞ

Ahmet AYDIN

ARAŞTIRMA DENEYİMİ

- ✓ Python, java, C#, C++, (C-C++, MATLAB, LABVIEW, vb.)
- ✓ Anaconda, PyCharm, Jupyter, Spyder, Google Colab, Unity (CAT/CAM, PROTEUS, CALCULUS, vb.)

İŞ DENEYİMİ

2021 – ... :Gaziantep İslam Bilim ve Teknoloji Üniversitesi- Araştırma Görevlisi
2019 – 2021 : ARVAS BİLİŞİM 2019-2021

AKADEMİK FAALİYETLER

Bildiriler:

1. Fine-Tuning Convolutional Neural Network Based Railway Damage Detection
(IEEE EUROCON 2021-19th International Conference on Smart Technologies)

Makaleler:

1. Comparative Analysis of Deep Learning Based Steel Surface Defect Detection under Different Lighting Conditions (Revizyon Aşamasında)

Projeler:

1. Görüntü İşleme ve Derin Öğrenme Teknikleri ile Drone Tespiti (Araştırmacı)
2. Makine Öğrenmesi Yöntemleriyle Elektrik Kesintisi Tahmini ve Bakım Planlama(Yardımcı)