

T.C.
BİLECİK ŞEYH EDEBALI ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
ELEKTRONİK VE BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

**DERİN ÖĞRENME METOTLARI İLE DEMİRYOLU ARIZALARININ TEŞHİSİ VE
SINIFLANDIRILMASI**

DOKTORA TEZİ

RİDVAN ÖZDEMİR

TEZ DANIŞMANI
DOÇ. DR. MEHMET KOÇ

BİLECİK, 2024

10686990

T.C.
BİLECİK ŞEYH EDEBALI ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
ELEKTRONİK VE BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

**DERİN ÖĞRENME METOTLARI İLE DEMİRYOLU ARIZALARININ TEŞHİSİ VE
SINIFLANDIRILMASI**

DOKTORA TEZİ

RİDVAN ÖZDEMİR

TEZ DANIŞMANI
DOÇ. DR. MEHMET KOÇ

BİLECİK, 2024

10686990

BEYAN

‘Derin Öğrenme Metotları ile Demiryolu Arızalarının Teşhisi ve Sınıflandırılması’ adlı doktora tezinin hazırlık ve yazımı sırasında bilimsel araştırma ve etik kurallarına uyduğumu, başkalarının eserlerinden yararlandığım bölümlerde bilimsel kurallara uygun olarak atıfta bulunduğumu, kullandığım verilerde herhangi bir tahrifat yapmadığımı, tezin herhangi bir kısmının Bilecik Şeyh Edebali Üniversitesi veya başka bir üniversitede başka bir tez çalışması olarak sunulmadığını, aksinin tespit edileceği muhtemel durumlarda doğabilecek her türlü hukuki sorumluluğu kabul ettiğimi ve vermiş olduğum bilgilerin doğru olduğunu beyan ederim.

Bu çalışmanın, Bilimsel Araştırma Projeleri (BAP), TÜBİTAK veya benzeri kuruluşlarca desteklenmesi durumunda; projenin ve destekleyen kurumun adı proje numarası ile birlikte, ETİK KURUL onayı alınması durumunda ise ETİK KURUL tarih karar ve sayı bilgilerinin beyan edilmesi gerekmektedir.			
DESTEK ALINMIŞTIR	X	DESTEK ALINMAMIŞTIR	
Destek alındı ise;			
Destekleyen kurum; Eskişehir Teknik Üniversitesi			
Destegin Türü		Proje Numarası	
1- BAP (Bilimsel Araştırma Projesi)	X	21GAP081	
2- TÜBİTAK			
Diğer;.....			
ETİK KURUL onayı var ise;			
ETİK KURUL karar tarih/sayı:		/.....	

Rıdvan ÖZDEMİR

Tarih

.....

İmza

.....

ÖN SÖZ

Lisansüstü eğitimim boyunca katkılarını esirgemeyen danışman hocam Doç. Dr. Mehmet KOÇ' a, tez çalışmam boyunca deneyimlerini paylaşarak çalışmalarına yön veren hocalarım Doç. Dr. Emre DANDIL, Prof. Dr. Ömer Nezih GEREK'e ve desteğini her zaman hissettiğim değerli arkadaşım Doç. Dr. Fehmi Burçin ÖZSOYDAN'a katkı ve emekleri için teşekkürlerimi sunarım. Demiryolları hakkındaki uzman görüşü ile katkılarını esirgemeyen arkadaşım Yüksek Mekatronik Müh. Hakkı KÜÇÜKKESKİN'e ve kullandığımız görüntüleri sağlayan TCDD DATEM işletme müdürlüğüne teşekkürlerimi sunarım.

Çalışma hayatım ile birlikte bu lisansüstü eğitim sürecinde de yanımda olan ETİ Makine ailesine ve değerli mesai arkadaşlarıma da teşekkürlerimi sunarım.

Ayrıca bu zorlu yolculuğun her anında yanımda olan, desteklerini hiçbir zaman esirgemeyen dostlarıma ve aileme, bir dönem ilkokul öğretmenliğimi de yapmış olan babam Veli ÖZDEMİR'e, canım annem Pakize ÖZDEMİR'e, özellikle de maddi-manevi her zorlukta desteğini hep yanımda hissettiğim sevgili dedem Arif ÖZDEMİR ve babaannem Ümmühan ÖZDEMİR'e en içten dileklerimle şükranlarımı sunarım.

Rıdvan ÖZDEMİR

2024

ÖZET

DERİN ÖĞRENME METOTLARI İLE DEMİRYOLU ARIZALARININ TEŞHİSİ VE SINIFLANDIRILMASI

Derin öğrenme metotları üzerine çalışmalar, bilgisayar donanımındaki, özellikle de grafik işlemci kartındaki gelişmelerden sonra büyük ivme kazanmıştır. Bu gelişmeler makine görüşü destekli uygulamaların hem daha başarılı sonuçlar vermesini hem de kullanım alanlarının genişlemesini sağlamıştır. Nesne sınıflandırma alanında insan başarımları seviyesi geride bırakılmış ve nesne tespit algoritmaları gerçek zamanlı çalışabilir hale gelmiştir. Derin öğrenme algoritmaları veriye aç sistemler olduğu için bunların eğitilmesi ve test edilmesinde veri kümesi kritik öneme sahiptir. Demiryolu arıza teşhisi ve sınıflandırması üzerine internette halka açık olarak erişimi mümkün olan bir veri kümesi bulunmamaktadır. Bu çalışmada Türkiye Cumhuriyeti Devlet Demiryolları'nın (TCDD) demiryolu hat bakım ve izleme çalışmalarında elde edilen görüntüler derlenip, literatürde en sık karşılaşılan ray kusurları göz önünde bulundurularak, demiryolu arıza teşhisi ve sınıflandırması uygulaması geliştirebilmek için bir veri kümesi oluşturulmuştur. Oluşturulan bu veri kümesi kullanılarak YOLOv4 tabanlı bir kusur tespit sistemi geliştirilmiştir. Önerilen bu kusur tespit sistemi geliştirilirken literatürdeki uygulamaların aksine tüm görüntülerin elle etiketlendiği bir tam denetimli öğrenme yöntemi yerine yarı denetimli öğrenci-öğretmen modeli geliştirilmiş ve görüntülerin otomatik olarak etiketlenmesi sağlanmıştır. Bu yöntem, etiketleme sürecindeki insan müdahalesini azaltarak yüksek maliyetlerin düşürülmesini sağlamış, sözde etiketleme yöntemi ile genişletilen veri kümesi sayesinde de arıza tespit modelinin mAP değeri; %87.94'ten %94.15'e, F₁-skor değeri de 0.83'ten 0.91'e arttırılmıştır. Tez kapsamındaki diğer bir çalışma ise demiryolu bağlantı elemanları özelinde yapılmıştır. Bu çalışmada aktivasyon fonksiyonu seçiminin arıza tespit modelinin başarımları üzerine etkisi incelenmiş ve leaky, swish, x-swish ve mish fonksiyonlarının performansları kıyaslanmıştır. Yapılan optimizasyon çalışmaları sonucunda, TCDD'den alınan veri kümesinin elde edildiği yöntemle karşılaştırıldığında, Gerçek Pozitif (TP) sayısı artırılmış ve Yanlış Pozitif (FP) sayısı azaltılmıştır. Son çalışmada ise demiryolu arızalarının daha yüksek oranda tespiti için dinamik alfa parametrelili odaklanmış kayıp (focal loss) fonksiyonu geliştirilmiştir. Böylece demiryolu arızalarında daha kritik öneme sahip arıza sınıfının tespitine ağırlık verilmiş ve bu sınıftaki arıza teşhis başarımları arttırılmıştır. Deney sonuçları, önerilen yöntemlerin mevcut yöntemlere göre daha yüksek doğruluk ve verimlilik sağladığını göstermektedir.

Anahtar Kelimeler: Demiryolu arıza tespiti, Söзде etiketleme, Yarı denetimli öğrenme, Demiryolu arıza veri kümesi, Öğrenci-öğretmen modeli



ABSTRACT

DIAGNOSIS AND CLASSIFICATION OF RAILWAY FAULTS USING DEEP LEARNING METHODS

The study of deep learning methods has gained great momentum following developments in computer hardware, especially in graphics processor cards. These developments have enabled machine vision-based applications to achieve more successful results and expand their areas of application. In the field of object classification, the level of human performance has been surpassed and object recognition algorithms have become capable of working in real time. As deep learning algorithms are data hungry systems, the dataset is critical for training and testing. There is no publicly available railway fault diagnosis and classification dataset on the internet. In this study, a railway fault diagnosis and classification dataset has been created by compiling the images obtained from the TCDD (Turkish Republic State Railways) railway line maintenance and monitoring works and considering the most common railway faults found in the literature. With this data set, a YOLOv4 based fault detection system has been developed. During the development of the proposed fault detection system, a semi-supervised student-teacher model was developed instead of a fully supervised learning method in which all images are manually labelled and the images are automatically labelled, unlike the applications in the literature. This method reduced the human intervention in the labelling process and the high costs, and the performance of the error detection model was gradually improved thanks to the data set extended by the pseudo-labelling method. Another study within the thesis was carried out on railway fasteners. This study analysed the effect of activation function selection on the performance of the fault detection model and compared the performance of leaky, swish, x-swish and mish functions. As a result of the optimization studies, the number of True Positives (TP) was increased and the number of False Positives (FP) was reduced compared to the method used to obtain the dataset from TCDD. In the last study, a dynamic alpha-parametric focussed loss function was developed to increase the detection rate of railway faults. This emphasises the detection of the more critical fault class in railway faults and increases the diagnostic performance in this class. Experimental results show that the proposed methods provide higher accuracy and efficiency than the existing methods.

Keywords: Rail defect detection, Pseudo-labeling, Semi-supervise learning, Rail defect dataset, Student–teacher model

İÇİNDEKİLER

Sayfa

ÖN SÖZ.....	i
ÖZET.....	ii
ABSTRACT	iv
İÇİNDEKİLER.....	v
TABLolar LİSTESİ.....	vii
ŞEKİLLER LİSTESİ.....	ix
KISALTMALAR VE SİMGELER LİSTESİ	xi
1. GİRİŞ.....	1
1.1. Demiryolunda Meydana Gelebilecek Arızalar	2
1.1.1. Ray Aşınmaları	2
1.1.2. Ray Kırılmaları.....	2
1.1.3. Ray Ezilmeleri.....	3
1.1.4. Ondülasyon	4
1.1.5. Mantar Yırtılmaları	5
1.1.6. Hasarlı Bağlantı Elemanı.....	6
1.2. Amaç ve Katkılar	6
1.3. Tez Organizasyonu.....	7
2. MATERYAL VE METOT	8
2.1. Veri Kümesinin Oluşturulması.....	10
2.1.1. Cıvata.....	11
2.1.2. Bağlantı Elemanı	12
2.1.3. Ray Kırığı.....	12
2.1.4. Hasarlı Bağlantı Elemanı.....	13
2.1.5. Travers.....	13
2.1.6. Yüzey kusuru	14
2.2. Elle Etiketleme ile Oluşturulan Veri Kümeleri	14
2.3. Nesne Tespit Algoritmaları	16
2.4. YOLO Versiyonları.....	18

2.5. Veri Etiketleme Aracı	22
2.6. Değerlendirme Metrikleri.....	22
3. SÖZDE ETİKETLER KULLANARAK YARI DENETİMLİ DERİN ÖĞRENME TABANLI DEMİRYOLU KUSUR TESPİTİNİN İYİLEŞTİRİLMESİ	24
3.1. Giriş	24
3.2. Malzeme ve Yöntemler	29
3.2.1. Veri Kümesi	29
3.2.2. Test Veri Kümesi	31
3.2.3. Öğrenci-Öğretmen Modeli.....	31
3.3. Deneyler.....	34
4. AKTİVASYON FONKSİYONU ADAPTASYONU İLE DEMİRYOLU KUSUR TESPİT ALGORİTMASININ PERFORMANS İYİLEŞTİRMESİ.....	39
4.1. Giriş	39
4.2. Malzeme ve Yöntemler	41
4.2.1. Veri Kümesi	41
4.2.2. Aktivasyon Fonksiyonları	44
4.3. Deneyler.....	46
5. DİNAMİK PARAMETRE KULLANIMI İLE RAY ARIZALARINDA SINIF TESPİT DOĞRULUĞUNUN İYİLEŞTİRİLMESİ	50
5.1. Ray Kusur Dereceleri	50
5.2. Deneyler.....	51
5.3. Özelleştirilmiş Kayıp Fonksiyonunun Modele Uygulanması	53
6. SONUÇLAR	60
KAYNAKÇA	62
EKLER.....	71

TABLÖLER LİSTESİ

	Sayfa
Tablo 2.1. Düzenlemeler sonunda elde edilen sınıflar	10
Tablo 2.2. İlk eğitim veri kümesi nesne sayısı dağılımı	15
Tablo 2.3. İkinci eğitim veri kümesi nesne sayısı dağılımı	16
Tablo 2.4. Test veri kümesi nesne sayısı dağılımı	16
Tablo 2.5. Değerlendirme ölçütleri için formüller	23
Tablo 3.1. Eğitim ve Test Veri Kümesi Görüntü dağılımı	31
Tablo 3.2. Öğrenci-Öğretmen modelin hiper parametreleri	33
Tablo 3.3. Sözde etiketlenmiş görüntülerin başarıma etkileri	34
Tablo 3.4. Sözde etiketli görüntüler eklendikten sonra genişletilmiş veri kümesindeki sınıfların nihai dağılımı	35
Tablo 3.5. İkinci deneyde sözde etiketli görüntülerin etkisi	37
Tablo 3.6. İkinci deneyde sözde etiketli görüntülerle genişletildikten sonra eğitim setinin nihai örnek dağılımı	38
Tablo 4.1. Eğitim, Doğrulama ve Test Veri Kümelerindeki Görüntü Sayısı	43
Tablo 4.2. Aktivasyon fonksiyonu performanslarının karşılaştırılması	46
Tablo 4.3. Aktivasyon fonksiyonlarına göre modelin kusur sınıfı tespit performansı	48
Tablo 4.4. Genişletilmiş veri kümesi ile eğitilen modellerin, aktivasyon fonksiyonu özelinde performanslarının karşılaştırılması	49
Tablo 4.5. Aktivasyon fonksiyonlarına göre modelin kusurlu sınıf özelinde performansı	49
Tablo 5.1. Sınıfların kusur dereceleri	50
Tablo 5.2. Modelin test veri kümesi üzerindeki performansı	51
Tablo 5.3. Modelin kusur sınıfları özelindeki performansı	51
Tablo 5.4. Odaklanmış kayıp fonksiyonu uygulandıktan modelin ezilme sınıfı özelinde başarımının α katsayısı değerine göre değişimi	56

Tablo 5.5. Odaklanmış kayıp fonksiyonu uygulandıktan modelin genel performansının α katsayısı değerine göre değişimi	57
--	----



ŞEKİLLER LİSTESİ

	Sayfa
Şekil 1.1. Ray aşınması	2
Şekil 1.2. Ray kırılması arızası	3
Şekil 1.3. Ray kırığı arızası	3
Şekil 1.4. Kısa dalga boylu ondülasyon kusurları.....	4
Şekil 1.5. Uzun dalga boylu ondülasyon kusurları	5
Şekil 1.6. Mantar yırtılmaları arızası	6
Şekil 2.1. Demiryolu hattından alınan örnek görüntü	8
Şekil 2.2. Demiryolu hattından alınan örnek görüntüler.....	9
Şekil 2.3. Cıvata sınıfı genel görüntüsü	11
Şekil 2.4. Cıvata sınıfından türetilen nesnelerin görüntüde işaretlenmesi	11
Şekil 2.5. Bağlantı elemanı nesnesinin görüntüde işaretlenmesi	12
Şekil 2.6. Ray kırığı ve mantar yırtılmaları nesnelerinin görüntüde işaretlenmesi.....	12
Şekil 2.7. Hasarlı bağlantı elemanı nesnesi ve görüntüde işaretlenmesi.....	13
Şekil 2.8. Travers ve travers çatlağı nesnesinin görüntüde tespiti	13
Şekil 2.9. Ondülasyon, yüzey kusuru ve ezilme nesnelerinin görüntüde tespiti.....	14
Şekil 2.10. YOLOv4 mimarisinin seviye katmanlarının özet gösterimi.....	21
Şekil 2.11. LabelImg’de etiketleme işlemi	22
Şekil 3.1. Test setinden örnek görüntüler. (a) mantar yırtılmaları, (b) ray kırığı, (c) hasarlı bağlantı elemanı, (d) ezilme ve (e) yüzey hatası.	31
Şekil 3.2. Öğrenci-öğretmen modeli kullanılarak sözde etiketleme yöntemi. İlk olarak manuel olarak etiketlenmiş görüntüler eğitim için kullanılır, daha sonra sözde etiketlenmiş görüntüler bir sonraki eğitim adımında eğitim veri kümesine eklenir.	33
Şekil 3.3. Sözde etiketli görüntülerin eklenmesinin etkisi AP değerleri üzerinde etkisi	35
Şekil 3.4. Temel gerçek ve öğrenci-öğretmen model etiketleme karşılaştırma örnekleri.....	36

Şekil 3.5. (a) Ezilme ve (b) ray kırığı sınıfları için etiketleme sürecinin iyileştirilmesi.....	37
Şekil 3.6. İkinci deneyde sözde etiketli görüntülerin eklenmesinin AP değerleri üzerindeki etkisi	38
Şekil 4.1. Travers, Bağlantı elemanı OK ve Bağlantı elemanı NOK Sınıflarına Örnekler	42
Şekil 4.2. Kusurlu olarak etiketlenen normal görüntüler	42
Şekil 4.3. Kusurlu olarak etiketlenen görüntülerin kontrol edilmesi	43
Şekil 4.4. Veri artırma tekniklerinin uygulanması örneği.....	43
Şekil 4.5. Aktivasyon fonksiyonlarının karşılaştırılması	44
Şekil 5.1. Farklı γ değerleri için kayıp ve olasılık değişimi.....	53
Şekil 5.2 AP değerlerinin α katsayısı ile değişimi	58
Şekil 5.3 α katsayısı ile TP ve FP sayıları değişimi	59

KISALTMALAR VE SİMGELER LİSTESİ

$\mathcal{L}_{cls}$	: Sınıf Kaybı (Class Loss)
$\mathcal{L}_{conf}$	: Güven Kaybı (Confidence Loss)
$\mathcal{L}_{loc}$	: Konumlandırma Kaybı (Localization Loss)
AP	: Ortalama Kesinlik (Average Precision)
ARTC	: Australian Rail Track Corporation
AWL	: Uyarlanabilir Ağırlıklı Kayıp (Adaptive Weighted Loss)
CE	: Çapraz Entropi (Cross Entropy)
CNN	: Evrişimli Sinir Ağı
CSP	: Aşamalar Arası Kısmi Paylaşım (Cross Stage Partial)
DATEM	: Demiryolu Araştırma ve Teknoloji Merkezi
F1	: Kritik Kusurlar
F2	: Büyük Kusurlar
F3	: Küçük Kusurlar
FL	: Odaklanmış Kayıp (Focal Loss)
FP	: Yanlış Pozitif (False Positive)
FN	: Yanlış Negatif (False Negative)
fps	: Frame Per Second
GPU	: Grafik İşleme Ünitesi (Graphical Processing Unit)
IoU	: Intersection over Union
mAP	: Sınıfların Ortalama Kesinliği (Mean Average Precision)
MSE	: Ortalama Karesel Hata (Mean Square Error)
PAN	: Yol Birleştirme Ağı (Path Aggregation Network)
R-CNN	: Bölgesel Evrişimli Sinir Ağları (Region-based Convolutional Neural Networks)
RAM	: Rastgele Erişimli Bellek (Random Access Memory)
RGB	: Kırmızı Yeşil Mavi (Red Green Blue)
RODD	: Rail Obstacle Detection Dataset
SAM	: Uzaysal Dikkat Modülü (Spatial Attention Module)
SE	: Sıkma ve Uyarma (Squeeze and Excitation)
SPP	: Uzaysal Piramit Havuzlama (Spatial Pyramid Pooling)

SSD	: Tek Atışta Nesne Tespiti (Single Shot MultiBox Detector)
SSL	: Yarı Denetimli Öğrenme (Semi Supervised Learning)
SVM	: Destek Vektör Makinesi (Support Vector Machine)
TCDD	: Türkiye Cumhuriyeti Devlet Demiryolları
TP	: Doğru Pozitif (True Positive)
VCUBE	: Hat Denetimi ve Ölçüm Sistemi
YOLO	: You Only Look Once



1. GİRİŞ

Demiryolu hatlarının elle ve gözle kontrol edilip, arıza teşhisinin yapılması artık demode bir yöntem olmuştur. Çünkü bu yöntem hem çok zahmetli hem de çok vakit alan bir işlemdir. Ayrıca kontrol eden kişinin bilgi ve yeteneği ile sınırlı öznel bir uygulamadır. Bu yüzden büyük tehlikelere neden olabilirler (Wilson & Norris, 2005).

Demiryolu arıza teşhisi için manuel kontrol haricinde, ultrasonik cihazlar kullanılarak yapılan kontrol uygulamaları da mevcuttur. Ultrasonik cihazlar yardımı ile temaslı olarak demiryolu hatlarının profilleri hakkında birtakım veriler elde edilir. Daha sonra bu veriler değerlendirilerek demiryolu arıza teşhis işlemi yapılır. Buradaki demiryolu arıza teşhis işleminde demiryoluna temas edildiği için bu yöntem, temaslı demiryolu arıza teşhis sistemi kategorisine girmektedir. Bu yöntem demiryolu hattı üst yüzeyinde meydana gelen aşınma, çatlak ve kırılma gibi hataların tespitinde çok etkilidir. Fakat arıza teşhis işleminin temaslı olarak yapılması bazen demiryolu yüzeyindeki hataların daha da artmasına bazen de yeni arızalar meydana gelmesine sebep olabilmektedir. Ayrıca sistem üst yüzeydeki arızaları teşhis edebilirken demiryolu hattının yan yüzeyindeki hataları belirleyememektedir (Jie vd., 2009).

Manuel kontrol ile yapılan uygulamaların sahip oldukları dezavantajlardan dolayı demiryollarında arıza teşhisi artık daha gelişmiş olan bilgisayarlı görü uygulamaları ile gerçek zamanlı olarak yapılmaktadır. Bilgisayarlı görüden faydalanan demiryolu arıza teşhis sistemlerinde yüksek çözünürlüklü alan veya çizgi kamerası, ışık kaynağı, enkoder gibi donanımlar kullanılmaktadır. Kullanılan bu yöntemlerde demiryolu yüzeyine temas olmadığı için bu yöntemler temassız kontrol şeklinde ifade edilir. Temassız yöntemler temaslı yöntemlerde meydana gelebilecek dezavantajları ortadan kaldırırken aynı zamanda da daha hızlı ve daha başarılı sonuçlar vermektedirler (Z. Liu vd., 2011).

Demiryollarındaki arızaların makine öğrenmesi ile teşhisi eğitilmiş model üzerinde, demiryolundan alınan anlık video görüntüsünün kare kare incelenerek, teste tabi tutulması ile olur. Fakat demiryolu üzerinde çevresel faktörlerden dolayı meydana gelen yağ ve toz lekeleri gibi izler algoritma tarafından arıza olarak teşhis edilebilmektedir. Bu sistemlerin en büyük dezavantajı budur (Feng vd., 2014).

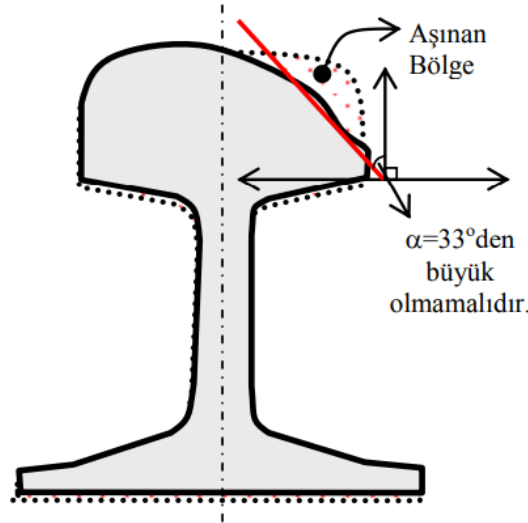
Demiryolu arıza teşhisi için kullanılan temassız yöntemlerden diğeri de üç boyutlu kameralar ile yapılan bilgisayar görü uygulamalarıdır. Fakat bu yöntem diğer yöntemlere göre daha maliyetli bir temassız arıza teşhisi yöntemidir (Min vd., 2018).

1.1. Demiryolunda Meydana Gelebilecek Arızalar

Bilgisayarlı görü temelli demiryolu arıza teşhisi uygulamalarında genel olarak arızaların tümünün aynı algoritma tarafından tespit edilmesi amaçlanmamaktadır. Nesne tespit algoritmaları ya yüzey kusurlarının ya da hasarlı bağlantı elemanının tespiti üzerine özelleştirilmiştir. Bu demiryolu arızalarını genel olarak şu başlıklar altında toplayabiliriz: ray aşınmaları, ray kırılmaları, ray ezilmeleri, ondülasyon, mantar yırtılmaları ve hasarlı bağlantı elemanı.

1.1.1. Ray Aşınmaları

Demiryolunda bulunan raylar düşey ve yatay yüzeylerinde tekerleğin teması ile zamanla aşınmaya maruz kalırlar. Meydana gelen bu aşınmalar zamanla büyüyerek ulaşım güvenliğini etkileyecek boyutlara ulaşabilirler. Bunun için rayların aşınmaya karşı yılda en az bir kez ilgili cihazlar ile kontrolünün yapılması gerekmektedir. Yapılan kontrollerde bu aşınmanın 33° 'den fazla olması durumunda bu rayların değiştirilmesi gerekmektedir (Santur, 2018). Şekil 1.1'de bu duruma örnek bir ray görseli çizimi ve ilgili açının gösterimi verilmiştir.



Şekil 1.1. Ray aşınması

Kaynak: (Santur, 2018)

1.1.2. Ray Kırılmaları

Rayın herhangi bir kısmında, malzemede süreksizlikler bulunması belirli bir zaman sonra kırılmaya neden olabilir. Gözle görünebilir veya görünemez olması fark etmeksizin bu süreksizlikler çatlak olarak kabul edilir. Bu durum görüldüğünde rayın hemen değiştirilmesi gerekir. Rayların ikiden fazla parçaya ayrılması, 50 mm'den daha fazla veya 10 mm'den daha

derin boşluklar oluşması durumunda ray kırılma arızası oluşmaktadır (TCDD, 2013b). Şekil 1.2’de ray kırılması arızasına bir örnek gösterilmektedir.



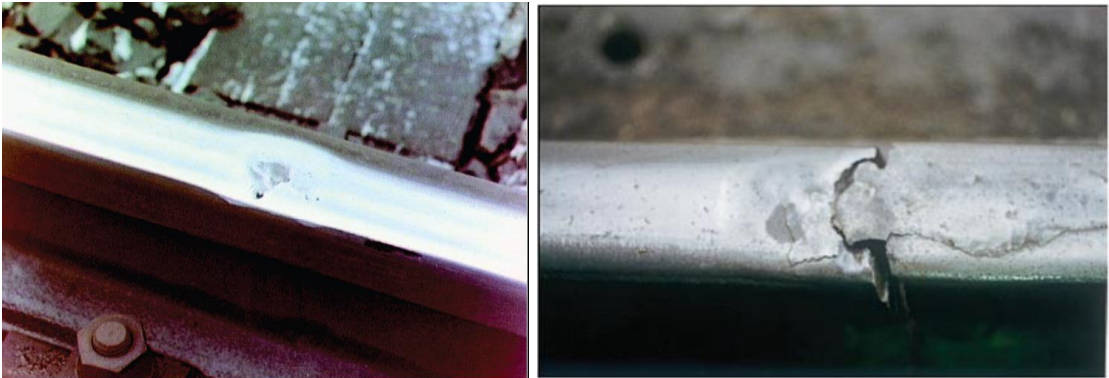
Şekil 1.2. Ray kırılması arızası

Kaynak: (Broken Rail, 2011)

1.1.3. Ray Ezilmeleri

Demiryolu yüzeyinde, tren tekerlekleriyle temas eden alanlarda oluşan yük nedeniyle genişlemeler meydana gelebilir. Bu yapılar, ray ezilme kusuru olarak adlandırılır. Yüzeydeki çatlaklar, rayın mantar kısmına doğru ilerleyerek 3-5 mm derinliğe ulaştığında, çatlaklar aşağıya doğru enine gelişir ve rayın kırılmasına neden olabilir. Bu kusur, genellikle ondülasyonlu bölgelerde ve termit kaynaklı alanlarda görülür (TCDD, 2013b).

Ray ezilmesi, tren tekerleğinin temas ettiği yüzeyde bölgesel olarak meydana gelen göçüklerle ortaya çıkar ve yüzeyde genişleme olarak gözlemlenir. Çatlaklar, mantarın iç kısmına doğru ilerledikçe rayın yapısal bütünlüğünü zayıflatır ve ciddi arızalara yol açabilir. Özellikle ondülasyonlu ve kaynaklı bölgelerde bu kusur tipine sıkça rastlanır. Şekil 1.3’te bu duruma örnek bir ray yüzeyi gösterilmiştir.



Şekil 1.3. Ray kırığı arızası

Kaynak: (TCDD, 2013b)

1.1.4. Ondülasyon

Demiryolu hattının raylarının üst yüzeyinde oluşan dalgalı aşınmalar ondülasyon olarak adlandırılır. Ondülasyon kusuru, üretim sırasında meydana gelebileceği gibi, tren tekerleğinin yuvarlanması sonucu oluşan aşınma, yorulma veya metal yığılması gibi etkenlerin bir araya gelmesiyle de ortaya çıkar.

Bu kusur doğrudan ray kırılmalarına yol açmasa da bağlantı elemanlarının gevşemesi, balastın yayılması, ses ve titreşim seviyesinin artması gibi sorunlara neden olur. Bu durum, ulaşım konforunun düşmesine sebep olabilir. Ondülasyon kusuru erken teşhis edilirse, ray yüzeyine yapılan taşlama işlemi ile giderilebilir. Ondülasyon kusuru ikiye ayrılır:

Kısa Dalga Boylu Ondülasyon: Dalga boyu 3 ile 8 cm arasında değişen bu tür, genellikle kusurlu tekerlekler, teker takımları veya ağır yük altında çalışma nedeniyle oluşur. Şekil 1.4'te kısa dalga boylu ondülasyona örnek görseller verilmiştir.

Uzun Dalga Boylu Ondülasyon: Dalga boyu 8 ile 30 cm arasında olan bu tip ondülasyon, genellikle rayın haddelenmesi sırasında oluşan üretim kusuru, yoğun trafik veya yüksek hızdan kaynaklanır. Şekil 1.5'te uzun dalga boylu ondülasyona örnek görseller verilmiştir (TCDD, 2013b).



Şekil 1.4. Kısa dalga boylu ondülasyon kusurları

Kaynak: (ARTC, 2006)



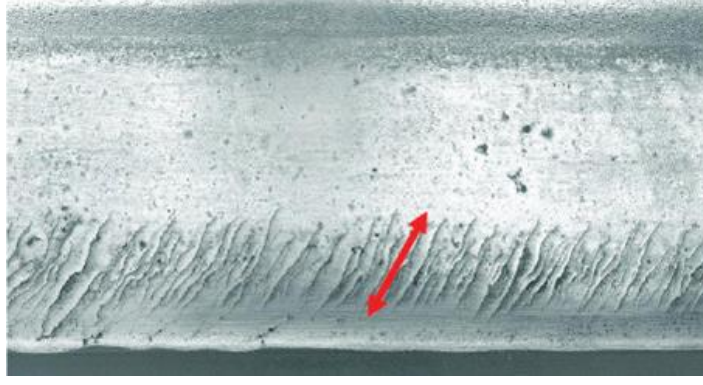
Şekil 1.5. Uzun dalga boylu ondülasyon kusurları

Kaynak: (ARTC, 2006)

1.1.5. Mantar Yırtılmaları

Mantar yırtılmaları, demiryolu raylarında çeşitli mekanik etkiler sonucu oluşan kılcal çatlaklar olarak tanımlanır. Bu çatlaklar, demiryolu aracının kalkışı sırasında raya yapışma, kumlama gibi nedenlerle ray yüzeyinde meydana gelen gerilmelerle başlar. Genellikle demiryolunun yuvarlanma temas yüzeyinden ekartman yönüne doğru, paralel eksenine 30 ile 60 derece arasında açıyla oluşurlar. Başlangıçta ince yüzey çatlakları olarak görülen bu kusurlar, önlem alınmadığı takdirde büyüyerek ray kırılmalarına yol açabilir. Erken tespit edildiklerinde ise taşlama işlemiyle giderilmeleri mümkündür (TCDD, 2013b).

Mantar yırtılmalarının aralarındaki mesafe genellikle 0,5 mm ile 10 mm arasında değişirken, R260 (900A) kalitesindeki raylarda daha kısa çatlaklar olarak karşımıza çıkar. Bu kusurlar dingil yükü, hız, patinaj, frenleme ve trenin dinamik kuvvetlerinden kaynaklanır. Farklı şekillerde gelişebilen ve her zaman açıkça gözlemlenemeyen mantar yırtılmaları, rayın performansını ve güvenliğini tehlikeye atabilecek ciddi arızalara dönüşebilir. Erken teşhis edilmediğinde ise, demiryolu altyapısında büyük hasarlara ve dolayısıyla yüksek onarım maliyetlerine yol açabilir. Mantar yırtılmaları arızasına ait örnek bir görsel Şekil 1.6'da verilmiştir.



Şekil 1.6. Mantar yırtılmaları arızası

Kaynak: (TCDD, 2013b)

1.1.6. Hasarlı Bağlantı Elemanı

Hasarlı bağlantı elemanı demiryolu altyapısında kritik bir kusur türüdür. Bağlantı elemanları, rayların traverslere sabitlenmesini ve düzgün bir şekilde hizalanmasını sağlayarak demiryolu hattının güvenliğini ve stabilitesini korur. Ancak uzun süreli kullanım, aşırı yükler, çevresel faktörler veya malzeme yorgunluğu nedeniyle bu elemanlar zamanla deforme olabilir. Bu deformasyonlar, ray ile travers arasındaki bağlantının zayıflamasına, rayların yerinden oynamasına ve hat üzerindeki titreşimlerin artmasına neden olabilir. Erken tespit edilmediğinde, bu durum raylarda hizalama bozuklukları, balastın aşınması ve hatta ray kırılmaları gibi ciddi arızalara yol açabilir (Lakušić vd., 2016).

Hasarlı bağlantı elemanları, demiryolu altyapısında güvenlik açısından büyük bir risk teşkil eder ve düzenli bakım süreçlerinin bir parçası olarak dikkatle izlenmelidir. Özellikle yüksek hızlı tren hatlarında, bu tür arızalar trenin dinamik yükleriyle birleştiğinde hat güvenliğini ciddi şekilde tehlikeye atabilir. Bağlantı elemanlarının deformasyonu erken evrede tespit edilirse, hasarın boyutuna bağlı olarak parça değişimi veya sıkılaştırma işlemleri ile arızanın önüne geçilebilir. Bu tür arızaların etkili bir şekilde yönetilebilmesi, demiryolu hattının uzun vadeli performansını ve güvenliğini artırmak için çok önemlidir (Chandran vd., 2021).

1.2. Amaç ve Katkılar

Bu tez, demiryolu arızalarının tespisi ve sınıflandırılması üzerine üç temel çalışmayı ele almaktadır. İlk çalışmada, Türkiye Cumhuriyeti Devlet Demiryolları'nın (TCDD) bakım ve izleme süreçleri sırasında toplanan görüntüler kullanılarak demiryolu arızalarının tespisi ve sınıflandırılması için kapsamlı ve özgün bir veri kümesi oluşturulmuştur. Literatürde demiryolu arızalarına ilişkin bir veri kümesi bulunmadığından, bu veri kümesi literatürdeki

önemli bir boşluğu doldurmuştur. Yarı-denetimli öğrenci-öğretmen modeli kullanılarak, veri etiketleme sürecindeki insan müdahalesi ve maliyetler büyük ölçüde azaltılmış, böylece modelin daha geniş bir veri yelpazesıyla eğitilmesi sağlanmıştır. Bu veri kümesi YOLOv4 tabanlı derin öğrenme modeliyle entegre edilerek arıza tespitinde yüksek doğruluk elde edilmiş ve tespit oranı önemli ölçüde artırılmıştır. Böylece bu çalışma sonraki adımlar için güçlü bir temel oluşturmuştur.

Tezin ikinci aşamasında, ilk çalışmada elde edilen veri seti ve model yapısı daha da geliştirilerek demiryolu bağlantı elemanlarının tespiti ve sınıflandırılmasına odaklanılmıştır. Bağlantı elemanları demiryolu hattının güvenliği için kritik öneme sahip olduğundan, bu elemanların arızalanması ciddi kazalara yol açabilmektedir. Bu çalışmada aktivasyon fonksiyonlarının model performansı üzerindeki etkileri araştırılmış ve leaky, swish, mish gibi fonksiyonlar karşılaştırılarak en uygun aktivasyon fonksiyonu seçilmiştir. Aktivasyon fonksiyonları ile sistem performansı optimize edilerek doğru pozitif (TP) oranları artırılmış ve yanlış pozitiflerin (FP) sayısı azaltılmıştır. Bu iyileştirmeler, veri kümesini ve modeli bağlantı elemanları gibi belirli hatalara karşı daha hassas hale getirilmiştir. Sonuç olarak, sistemin hassasiyeti ve doğruluğu artırılmış ve sistem arıza tespitinde daha verimli olmuştur.

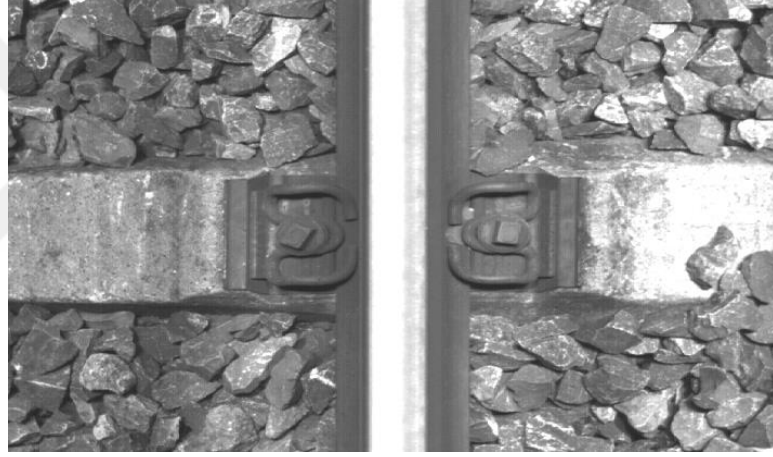
Tezin üçüncü aşamasında, daha kritik demiryolu hatalarını tespit etmek için dinamik alfa parametre odaklı bir kayıp fonksiyonu geliştirilmiştir. Bu fonksiyon, arıza tespit sisteminin özellikle kritik arıza sınıflarına daha fazla ağırlık vermesini sağlamış ve modelin bu sınıflardaki performansını artırmıştır. Veri seti ve model optimizasyonları üzerine inşa edilen bu yeni kayıp fonksiyonu, kritik arızaların tespit oranını iyileştirerek genel sistem performansını önemli ölçüde artırmıştır. Dolayısıyla, bu çalışma sadece daha büyük veri kümesi ve aktivasyon fonksiyonlarının değil, aynı zamanda kayıp fonksiyonunun da model performansında kritik bir rol oynadığını göstermiştir.

1.3. Tez Organizasyonu

Tezin geri kalanı şu şekilde düzenlenmiştir: Bölüm 2’de kullanılan materyal ve metotlardan bahsedilmiştir. Bölüm 3’te ise sözde etiketler kullanarak yarı denetimli derin öğrenme tabanlı demiryolu kusur tespitinin iyileştirilmesinden bahsedilmiştir. Bölüm 4’te demiryolu bağlantı elemanları kusur tespit algoritmasının aktivasyon fonksiyonlarının adaptasyonu ile nasıl optimize edildiği anlatılmaktadır, Bölüm 5’te dinamik alfa parametreleriyle raylı sistem arızalarının sınıf bazında tespiti ele alınmıştır ve tezin genel bir değerlendirmesi Bölüm 6’da yapılmıştır.

2. MATERYAL VE METOT

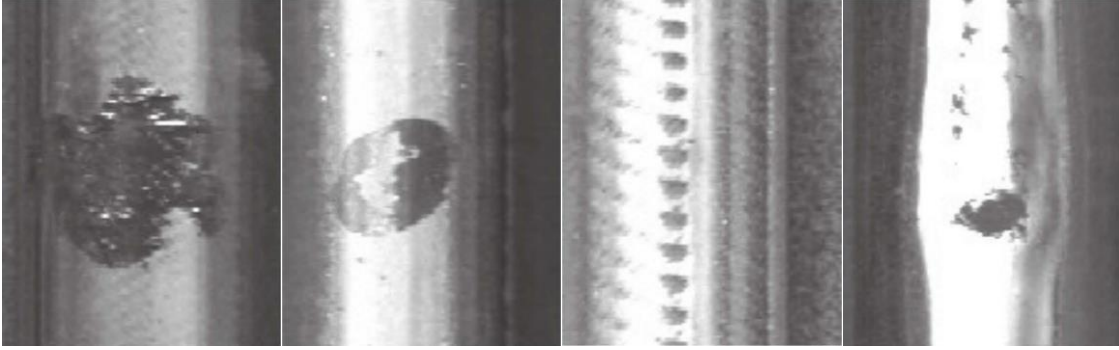
Demiryolu arızalarının tespitinde bilgisayarlı görü destekli sistemlerin kullanımı yaygınlaşmaktadır. TCDD demiryolu arıza tespitinde MERMEC şirketi tarafından geliştirilen VCUBE adında bilgisayarlı görü tabanlı bir ‘Hat Denetimi ve Ölçüm Sistemi’ kullanılmaktadır. VCUBE, hat boyunca gözle denetim gerçekleştiren devriye denetçilerinin yerini almak için, otomatik olarak demiryolu analiz, denetim ve ölçümünü gerçekleştirmek üzere tasarlanmış bir bilgisayarlı görü sistemidir. VCUBE sisteminin uygulanmasıyla, özellikle normalde kapalı olması gereken ve bakım için çok sınırlı vakit bulunan hızlı tren hatlarında, demiryolu hatları boyunca devriye işleri çok daha kolay ve daha etkin olmaktadır (Santur, 2018). Hat Denetim Sistemi hat durumunun otomatik kontrolü için, rayı ve çevresini görüntüleyen iki dijital kamera bulundurur. Şekil 2.1’de VCUBE sisteminin demiryolu hattından aldığı örnek bir görüntü verilmiştir (TCDD, 2013a).



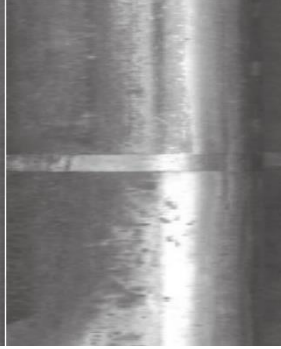
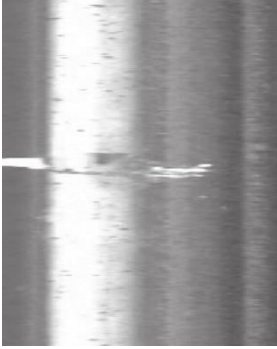
Şekil 2.1. Demiryolu hattından alınan örnek görüntü

Sistem rayı ve çevresini görüntüleyen sayısal kameranın dışında nesnenin kararlı ve dengeli aydınlatılmasını sağlayacak bir ışık sistemi de bulunmaktadır. Sistem tarafından yakalanan bazı arızalara ait bazı örnekler aşağıda Şekil 2.2’de verilmiştir.

Ray yüzey kusurları



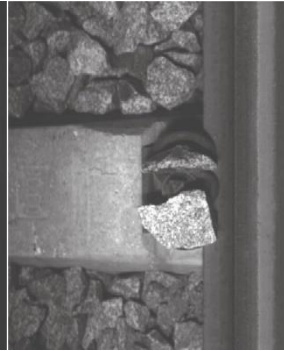
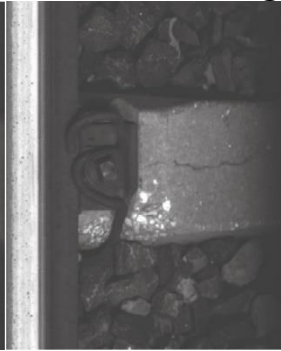
Ray kırıkları



Kayıp bağlantı elemanı



Hasarlı bağlantı elemanı



Şekil 2.2. Demiryolu hattından alınan örnek görüntüler

2.1. Veri Kümesinin Oluşturulması

Demiryolu hatları belli aralıklar ile görsel olarak kontrol edilir. Bu kontrol hızlı tren hatlarında 2 ayda bir, konvansiyonel hatlarda ise 6 ayda bir olarak planlanmıştır. Bu kontrol sürecinde bir kusur tespit edilirse önem derecesine bağlı olarak; küçük kusurlar, büyük kusurlar ve kritik kusurlar olmak üzere 3 grupta toplanır. Bu kontroller esnasında alınan görüntülerden, veri kümesini oluşturmak için öncelikle ham halde bulunan veri işlenebilir hale getirilmiştir. 14,8 GB boyutunda 747 farklı klasör içinde bulunan 289.488 adet dosyadan oluşan bu veri yığını içerisinde; sıkıştırılarak kaydedilmiş, mükerrer, kullanmaya elverişli olmayan (boyut olarak) ve gri ölçekte olmayan (lazer görüntüsü) görüntüler de bulunmaktaydı. Bunun için öncelikle sistemde bulunan bütün sınıfların listesi çıkartılmış ve mükerrer olan görseller elenmiştir. Veri kümesinin düzenlenmesine ait detaylı bilgi EK-1 ve EK-2 bölümlerinde verilmiştir.

Yeni oluşturulan veri kümesi 19831 adet görüntüden meydana gelmekte olup, 7 adet sınıf içermektedir. Literatürdeki arıza tipleri ve veri kümesindeki görüntüler incelendiğinde ise bazı sınıfların alt sınıflar barındırdığı ve bunların da ayrı birer nesne olarak ele alınabileceği görüldü. Örneğin ‘civata’ nesnesini içeren görüntüler içinde ‘Yabancı nesne’ nesnesini içiren yeterli sayıda görsel bulunmaktadır. Aynı şekilde ‘Ray kırığı’, ‘Travers’ ve ‘Yüzey kusuru’ sınıflarından da alt sınıflar türetilerek 11 adet sınıf elde edilmiştir. Tablo 2.1’de düzenlemeler elde edilen sınıf listesi verilmiştir. Çalışmanın devamında bu sınıflar tek tek ele alınıp detaylı bilgi verilmiştir.

Tablo 2.1. Düzenlemeler sonunda elde edilen sınıflar

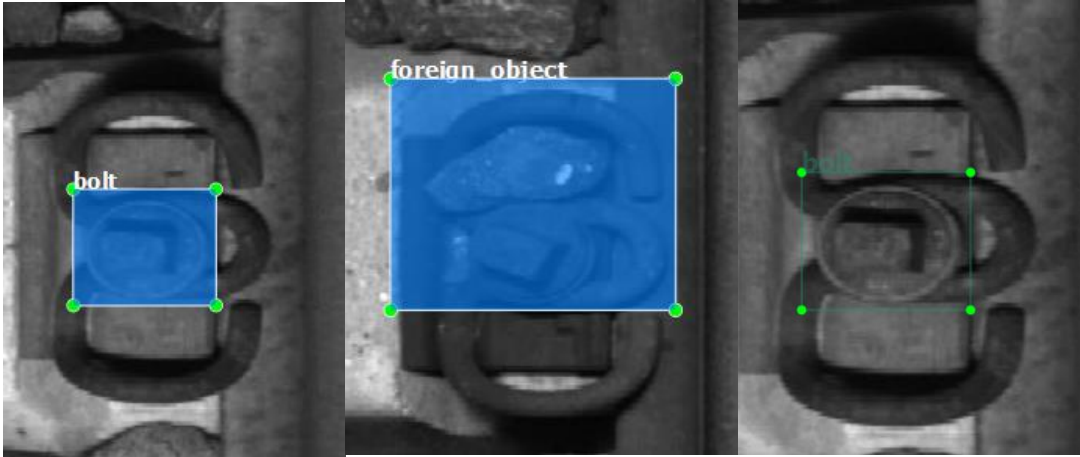
No	Sınıf
1	Travers
2	Yabancı nesne
3	Civata
4	Mantar yırtılmaları
5	Bağlantı elemanı
6	Travers çatlağı
7	Ray kırığı
8	Hasarlı bağlantı elemanı
9	Ondülasyon
10	Ezilme
11	Yüzey kusuru

2.1.1. Cıvata

Cıvata sınıfındaki verilerden ‘**cıvata**’ ve ‘**yabancı nesne**’ alt sınıfları türetilmiştir. Şekil 2.3’te cıvata sınıfına ait genel görüntü verilirken Şekil 2.4’te ise cıvata sınıfına ait alt sınıf nesnelerin konumları işaretlenmiş ve etiketlenmiştir. Burada cıvata sıkma elemanı manasında olup bağlantı elemanı aracılığı ile traversin rayı tutmasını sağlar. Bu sıkma elemanının kayıp olması bir arıza türüdür. Yabancı nesne ile ise bağlantı elemanı üzerindeki yabancı maddeler kastedilmektedir. Bu maddeler genelde balast kısmından gelen ufak taşlar olabilmektedir.



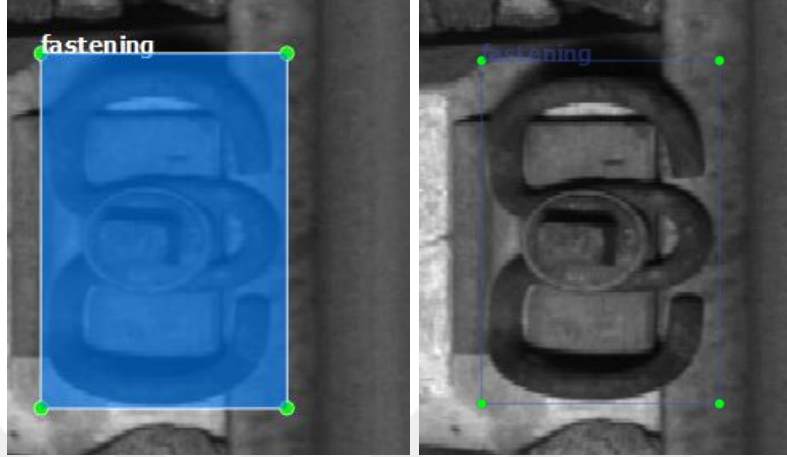
Şekil 2.3. Cıvata sınıfı genel görüntüsü



Şekil 2.4. Cıvata sınıfından türetilen nesnelerin görüntüde işaretlenmesi

2.1.2. Bağlantı Elemanı

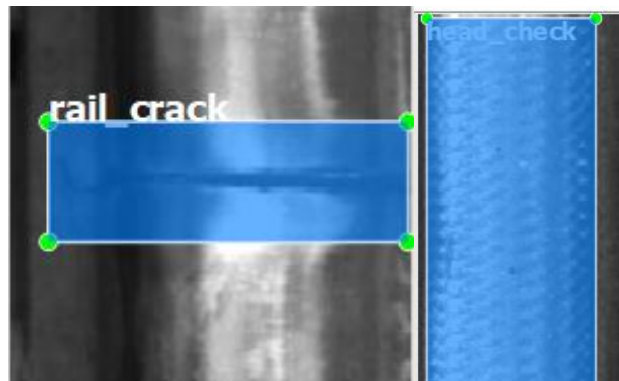
Bağlantı elemanı rayın traverse tutunarak sabit kalabilmesini sağlar. Bu kısıcının kayıp olması bir arıza türüdür. Şekil 2.5'te ise Bağlantı elemanı nesnenin görüntüdeki seçimi ve işaretlenmesine ait bir örnek verilmiştir.



Şekil 2.5. Bağlantı elemanı nesnesinin görüntüde işaretlenmesi

2.1.3. Ray Kırığı

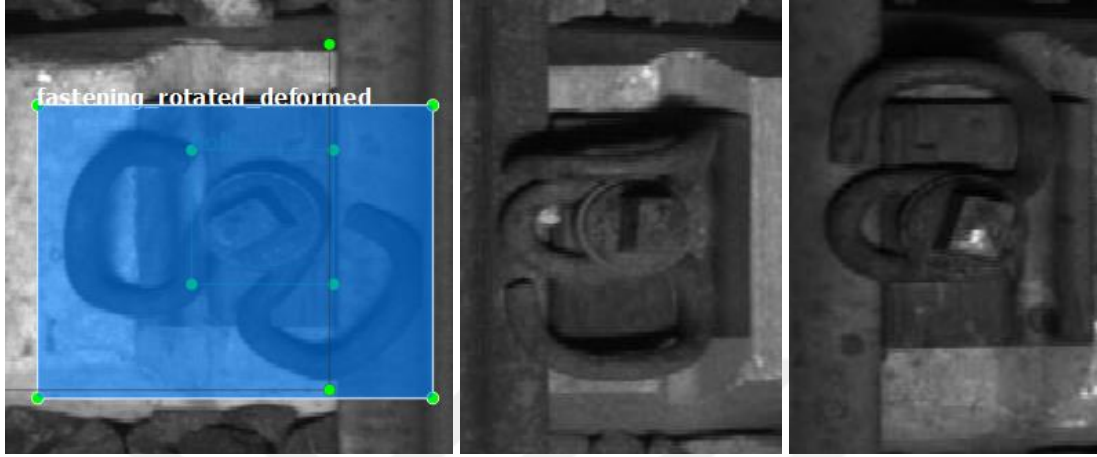
Ray kırığı genel literatürde 'rail crack' olarak adlandırılır. Bu sınıfa ait görüntüler incelendiğinde içerisinde 'Mantar yırtılmaları' hatalarını da barındıran görüntülerin olduğu gözlemlenmiştir. Bunlar ray yüzeyi hatalarından sadece ikisidir. Ray kırılmaları en tehlikeli arızalardan biri olup acilen müdahale gerektirmektedir. Mantar yırtılmaları arızaları ise trenin duruş ve kalkışlarda ray yüzeyinde meydana getirdiği deformasyonlardır. Şekil 2.6'da ray kırığı ve mantar yırtılmaları arızalarının görüntüdeki tespiti ve etiketlenmesine örnek verilmiştir.



Şekil 2.6. Ray kırığı ve mantar yırtılmaları nesnelerinin görüntüde işaretlenmesi

2.1.4. Hasarlı Bağlantı Elemanı

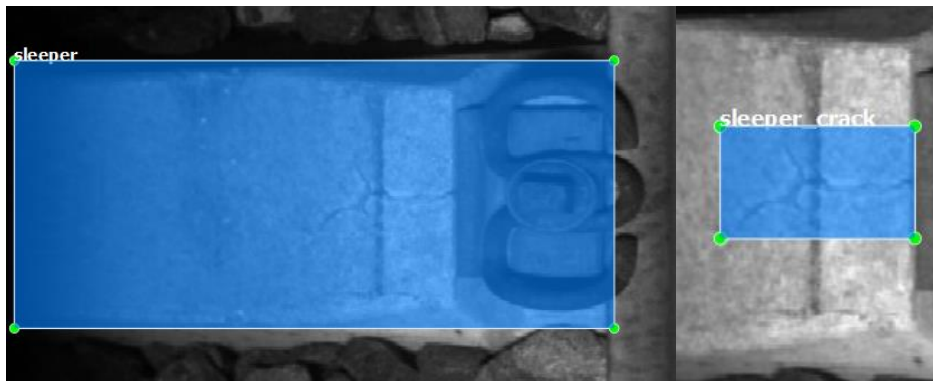
Hasarlı bağlantı elemanı rayın traverse tutuşunu azaltarak arızalara neden olabilmektedir. Şekil 2.7’de hasarlı bağlantı elemanı nesnesine ait dönme, deforme olma ve kısmen kopma örnekleri verilmiştir.



Şekil 2.7. Hasarlı bağlantı elemanı nesnesi ve görüntüde işaretlenmesi

2.1.5. Travers

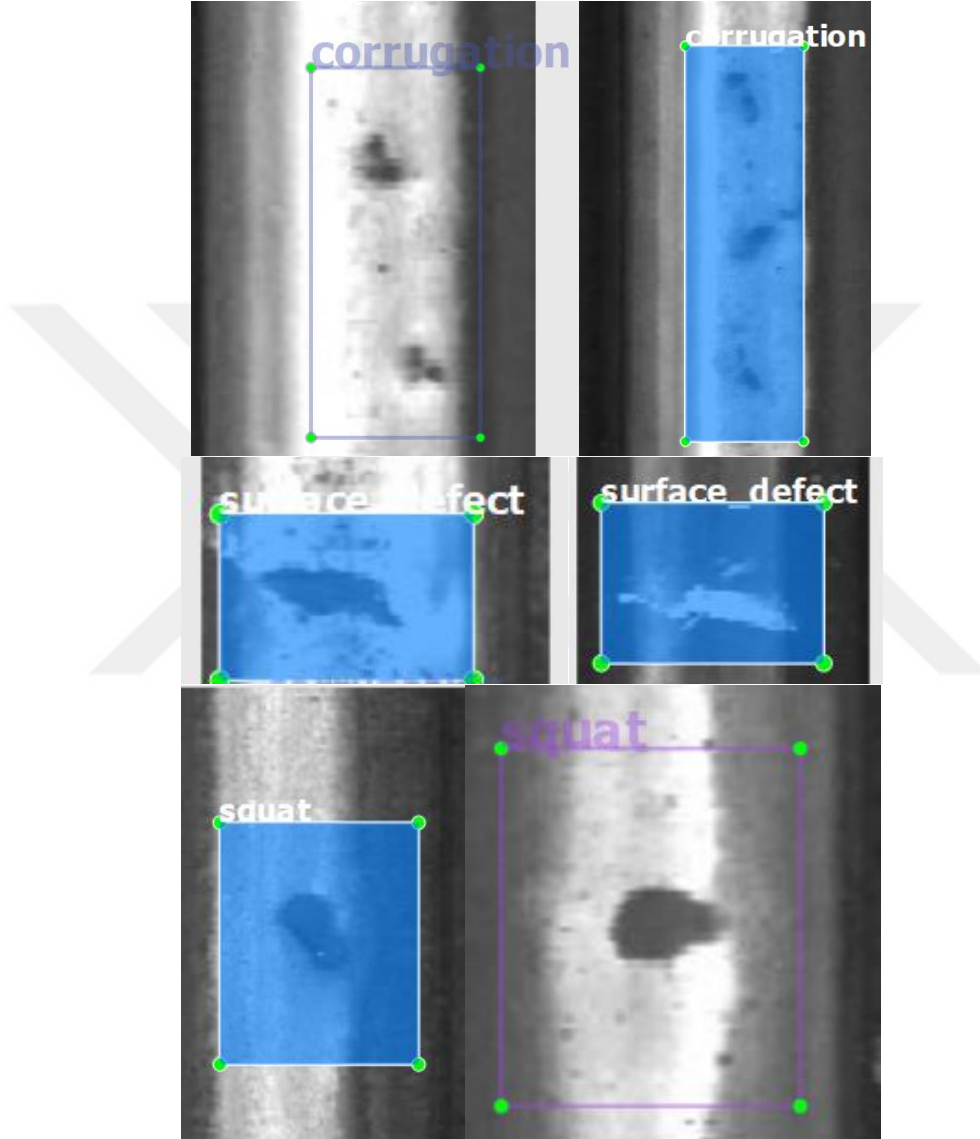
Travers balastın üstünde rayın altında bulunan rayın düzgün ve sabit durmasını sağlayan demiryolu elemanıdır. Bu sınıf içerisindeki görüntüler incelendiğinde ‘Travers çatlağı’ olarak tabir edilen travers çatlakları arızaları da görülmüş ve bunlar için de bir nesne sınıfı oluşturulmuştur. Ayrıca ‘kayıp cıvata’ ve ‘kayıp bağlantı elemanı’ arızalarının tespiti için de travers nesnesinin tespiti önemlidir çünkü travers üzerinde cıvata veya bağlantı elemanı nesnelerinin bulunmaması durumunda bu arızalardan birinin olduğu söylenebilir. Şekil 2.8’de travers ve travers çatlağı nesnelerinin tespitine ait görüntüler verilmiştir.



Şekil 2.8. Travers ve travers çatlağı nesnesinin görüntüde tespiti

2.1.6. Yüzey kusuru

Yüzey kusuru yani yüzey hasarlarına ait görüntüler incelendiğinde bunlar arasında ezilme ve ondülasyon arızalarının da olduğu saptanmış ve bunlar için de birer nesne sınıfı oluşturulmuştur. Şekil 2.9’da ondülasyon, yüzey kusuru ve ezilme nesnelerinin görüntüde tespitine örnekler verilmiştir.



Şekil 2.9. Ondülasyon, yüzey kusuru ve ezilme nesnelerinin görüntüde tespiti

Veri kümeleri elle ve otomatik olarak etiketlenenler olmak üzere iki ana gruba alınmış olup sıradaki bölümlerde detaylı olarak ele alınmışlardır.

2.2. Elle Etiketleme ile Oluşturulan Veri Kümeleri

Bir uzman görüşü doğrultusunda, VCUBE sistemi tarafından demiryolu hattından alınan görüntülerin, etiketleme aracı kullanılarak işaretlenmesi ile oluşturulan veri kümeleri

‘elle etiketleme ile oluşturulan veri kümeleri’ olarak adlandırıldı. Toplam 1195 adet etiketlenmiş görüntü ile 3 adet veri kümesi oluşturuldu. Bu veri kümeleri sırasıyla 374, 995 ve 200 görüntüden oluşmaktadır. Bu veri kümelerine ait detaylar ve içerdikleri nesne sınıfı sayıları şu şekildedir:

374 görüntü ve 897 adet nesneden oluşan veri kümesi ilk eğitim veri kümesi olarak oluşturulmuştur. Görüntü sayısı ve nesne sayısının farklı olmasının nedeni bazı görüntülerde birden fazla nesne veya kusur olmasıdır. İlk eğitim veri kümesinin nesne sayısına ait detaylar Tablo 2.2’de verilmiştir.

Tablo 2.2. İlk eğitim veri kümesi nesne sayısı dağılımı

Sınıf	Adet
travers	215
yabancı nesne	58
cıvata	221
mantar yırtılmaları	46
bağlantı elemanı	125
travers çatlağı	42
ray kırığı	28
hasarlı bağlantı elemanı	47
ondülasyon	35
ezilme	33
yüzey kusuru	47

İlk eğitim veri kümesine yeni görüntüler eklenerek genişletilmesi ile ikinci eğitim veri kümesi elde edilmiştir. Bu eğitim veri kümesinin oluşturulma amacı; etiketlenmiş örnek görüntü sayısındaki artışın nesne tespit modelinin performansı üzerindeki etkisini inceleyebilmektir. Oluşturulan bu ikinci eğitim veri kümesinde toplam 995 görüntü ve 2651 adet nesne bulunmaktadır. İkinci eğitim veri kümesindeki nesne sayısı dağılımına ait detaylar Tablo 2.3’te verilmiştir.

Tablo 2.3. İkinci eğitim veri kümesi nesne sayısı dağılımı

Sınıf	Adet
Travers	643
yabancı nesne	177
cıvata	752
mantar yırtılmaları	77
bağlantı elemanı	407
Travers çatlağı	89
ray kırığı	49
hasarlı bağlantı elemanı	159
ondülasyon	103
ezilme	77
yüzey kusuru	118

Son olarak oluşturulan veri kümesi ise test veri kümesi olarak kullanılmıştır. Test veri kümesinde 200 adet görüntü ve 520 adet nesne bulunmaktadır. Test veri kümesindeki nesne sayısına ait detaylar Tablo 2.4’te verilmiştir.

Tablo 2.4. Test veri kümesi nesne sayısı dağılımı

Sınıf	Adet
Travers	125
cıvata	145
mantar yırtılmaları	10
bağlantı elemanı	77
ray kırığı	9
hasarlı bağlantı elemanı	35
ondülasyon	18
ezilme	20
yüzey kusuru	34

2.3. Nesne Tespit Algoritmaları

Nesne tespit algoritmaları, görüntülerdeki nesneleri sınıflandırmak ve konumlandırmak için kullanılan yapay zekâ teknikleridir.

Haar sınıflandırıcıları, ilk gerçek zamanlı yüz dedektöründe kullanılan sınıflandırıcılardır. Haar sınıflandırıcı veya Haar Cascade sınıflandırıcı, görüntü ve videodaki nesneleri tanımlayan bir makine öğrenimi nesne algılama algoritmasıdır (Viola & Jones, 2001). Algoritma dört aşamada açıklanabilir: haar özelliklerinin hesaplanması, integral görüntüler oluşturma, adaboost'u kullanma, basamaklı sınıflandırıcıların uygulanması (A. Mittal, 2020).

Nesne algılama üzerine yapılan diğer çalışmalar, algılamayı gerçekleştirmek için sınıflandırıcıları yeniden kullanmaktadır. Fakat YOLO algoritması, nesne tespitini uzamsal olarak ayrılmış sınırlayıcı kutulara ve ilişkili sınıf olasılıklarına bağlı bir regresyon problemi olarak ele almaktadır. Tek bir yapay sinir ağı, tek bir değerlendirmede sınırlayıcı kutuları ve sınıf olasılıklarını doğrudan görüntüyü parçalara ayırmadan tahmin eder. Tüm algılama hattı tek bir yapay sinir ağı olduğundan, doğrudan algılama performansı üzerinde uçtan uca optimize edilebilir.

YOLO'nun bütünleşik mimarisi son derece hızlıdır. Temel YOLO modeli görüntüleri gerçek zamanlı olarak saniyede 45 kare hızında işleyebilir. Ağın daha küçük bir versiyonu olan Fast YOLO, saniyede 155 kare hızda işlem yapabilir. Bu performans o zaman için diğer gerçek zamanlı dedektörlerin iki katı mAP (ortalama hassasiyet) değerine ulaşmıştır. Son geliştirilen tespit sistemleriyle karşılaştırıldığında, YOLO daha fazla lokalizasyon hatası yapmasına karşın, nesnenin var olmadığı durumlarda, yanlış tespit tahminlerinin sayısı da diğer nesne tespit algoritmalarına göre çok daha düşüktür. Son olarak, YOLO nesnelerin çok genel temsillerini öğrenir. Hem Picasso Veri Kümesi hem de People-Art Veri Kümesi üzerinde doğal görüntülerden sanat eserlerine genelleme yaparken DPM ve R-CNN dahil olmak üzere diğer tüm algılama yöntemlerinden çok daha iyi performans göstermiştir (Redmon vd., 2016).

SSD (Single Shot MultiBox Detector), tek bir derin sinir ağı kullanarak görüntülerdeki nesneleri tespit eden bir algoritmadır (W. Liu vd., 2016). Nesne öneri adımlarını ortadan kaldırarak, farklı en-boy oranları ve ölçeklere sahip varsayılan kutular kullanır ve bu kutuları nesne şekillerine uyacak şekilde ayarlar. Farklı çözünürlüklü özellik haritalarından gelen tahminleri birleştirerek, çeşitli boyutlardaki nesneleri etkili bir şekilde algılar. SSD, PASCAL VOC ve MS COCO gibi veri kümelerinde, nesne önerisi gerektiren yöntemlerle benzer doğrulukta ancak çok daha hızlı sonuçlar elde etmiştir.

R-CNN (Region-based Convolutional Neural Networks) ise görüntüdeki olası nesne bölgelerini seçip her bölgeyi ayrı ayrı konvolüsyonel sinir ağları ile işler. İlk adımda bölge önerileri üretir, ardından bu bölgeleri sınıflandırır. Bölge önerileri nedeniyle daha yavaş ama daha yüksek doğruluk sağlar (Girshick vd., 2013).

Daha sonra geliştirilen Fast R-CNN ise R-CNN'den farklı olarak, tüm görüntüyü bir kerede işler ve ardından seçilen bölgeleri tek bir ağ ile sınıflandırır ve sınır kutularını iyileştirir. R-CNN'den daha hızlıdır, ancak hala bölge öneri süreci gerektirir. Fast R-CNN, derin evrişimli sinir ağlarını ve önceki çalışmalardan edinilen bilgileri de kullanarak, nesneleri

verimli bir şekilde sınıflandırabilmektedir. Önceki çalışmalarla karşılaştırıldığında, Fast R-CNN eğitim ve test hızını iyileştirmek ve aynı zamanda tespit doğruluğunu artırmak için çeşitli yenilikler kullanır. Fast R-CNN, VGG-16 ağını R-CNN'den 9 kat daha hızlı eğitir, test zamanında 213 kat daha hızlıdır ve PASCAL VOC 2012'de daha yüksek bir mAP elde eder. SPPnet (Uzaysal Piramit Havuzlama Ağı) ile karşılaştırıldığında, Fast R-CNN VGG-16'yı 3 kat daha hızlı eğitir, 10 kat daha hızlı test eder ve daha doğru sonuçlar verir (Girshick, 2015).

Fast R-CNN modelinin geliştirilmesi ile elde edilen Faster R-CNN, Fast R-CNN modeline entegre edilmiş bir 'Bölge Öneri Ağı' literatürdeki adıyla, Region Proposal Network (RPN) kullanır. RPN, görüntüyü tarayarak nesne adaylarını daha verimli bir şekilde önerir. Hem hızlı hem de yüksek doğruluk sağlar, ancak hesaplama maliyeti yüksektir (Ren vd., 2016).

Mask R-CNN ise, Faster R-CNN'nin bir uzantısı olarak çalışır ve tespit edilen her nesne için bir piksel seviyesinde maske oluşturur. Hem nesne tespiti hem de segmentasyonu gerçekleştirir. Hem sınırlayıcı kutu hem de piksel seviyesinde segmentasyon sağlar (K. He vd., 2018).

Diğer bir algoritma olan RetinaNet, odaklanmış kayıp fonksiyonunu kullanarak küçük ve nadir nesneleri tespit etmeyi iyileştirir. Nesne tespiti için tek bir konvolüsyonel ağ kullanır. Küçük ve zor tespit edilen nesneler için optimize edilmiştir, YOLO ve SSD'ye kıyasla daha dengeli bir performans sağlar (T.-Y. Lin vd., 2018).

CenterNet algoritması, nesnelerin merkez noktalarını tespit eder ve bu noktaları kullanarak sınır kutularını oluşturur. Hem nokta tespiti hem de sınırlayıcı kutu oluşturma için aynı ağı kullanır. Özellikle insan ve trafik analizi gibi uygulamalarda etkili olan basit ve etkili bir yöntemdir (Duan vd., 2019).

Bir diğer nesne tespit algoritması olan EfficientDet ise EfficientNet tabanlı bir mimari kullanır ve EfficientNet'in verimliliğini nesne tespiti için uygular. Ölçeklenebilirlik ve verimlilik için optimize edilmiştir. Daha az hesaplama gücü gerektirir ve yüksek doğruluk sağlar, özellikle mobil ve gömülü cihazlar için uygundur (Tan vd., 2020).

2.4. YOLO Versiyonları

YOLO algoritmasının farklı sürümleri, nesne tespitinde hız ve doğruluğu artırmak için çeşitli iyileştirmeler ve yenilikler sunar. İşte YOLO'nun sürümlerinin detayları ve aralarındaki farklar:

YOLO'nun ilk sürümü, görüntüyü ızgaralara böler ve her bir ızgara için sınıflandırma ve sınırlayıcı kutu tahmini yapar. Sinir ağı yapısı olarak DarkNet-19'u kullanarak, tespit ve sınıflandırma işlemlerini aynı anda gerçekleştirir. Çok hızlıdır, ancak küçük nesneler ve yoğun alanlar için doğruluğu düşüktür (Redmon vd., 2016).

YOLOv2 veya YOLO9000, Darknet-19 ağ yapısını Darknet-19 olarak günceller ve Batch Normalization ile eğitimi stabilize eder. Anchor kutuları eklenir, bu da daha iyi sınır kutusu tahmini sağlar. Özellik piramidi ve daha yüksek çözünürlükteki girdiler ile daha iyi performans elde edilir. WordTree ile birlikte çalışarak 9000 sınıfı tespit edebilir. YOLOv1'e göre hem hız hem de doğrulukta önemli iyileştirmeler sağlar (Redmon & Farhadi, 2016).

YOLOv3, Darknet-53 ağ yapısını kullanır, bu da daha derin ve daha geniş bir ağ yapısıdır. Üç farklı ölçekle nesne tespiti yaparak daha iyi küçük nesne tespiti sağlar. Yol-3 algoritması Reziidüel (Residual) bloklar içerir, bu da daha iyi öğrenme ve performans sağlar. Lojistik regresyon kullanarak sınıflandırma yapılır. YOLOv2'ye göre daha yüksek doğruluk sağlar, özellikle küçük nesneler için daha iyidir, ancak hesaplama maliyeti fazladır (Redmon & Farhadi, 2018).

YOLOv4, CSPDarknet53 tabanlı bir omurga (backbone) kullanır. Mozaik veri artırma, DropBlock ve Mish aktivasyon fonksiyonu gibi çeşitli tekniklerle eğitimi ve genel performansı iyileştirir. YOLOv3'ün avantajlarını korur ve hem hız hem de doğrulukta önemli iyileştirmeler sunar. Bag of Freebies (BoF) ve Bag of Specials (BoS) teknikleri kullanılarak daha iyi genel performans sağlanır. YOLOv3'e göre daha hızlı ve daha doğrudur, veri artırma ve optimizasyon teknikleri ile eğitim performansı iyileştirilmiştir (Bochkovskiy vd., 2020).

YOLOv4, tespit ve sınıflandırmayı aynı anda gerçekleştiren tek aşamalı bir özellik dedektörüdür. YOLOv4 modelinin amacı, YOLOv3'ün gerçek zamanlı nesne algılama hızını ve doğruluğunu optimize etmektir. YOLOv4 temel olarak üç ana bölümden oluşmaktadır: Omurga olarak CSPDarknet53 (C.-Y. Wang vd., 2019), boyun olarak uzaysal piramit havuzlama ek modülü (SPP) (K. He vd., 2014) ve yol birleştirme (PAN) (S. Liu vd., 2018) ve baş olarak YOLOv3 (Redmon & Farhadi, 2018). CSPDarkNet53, giriş görüntülerini işleyen ve etkili nesne tespiti için üst düzey özellikler çıkaran bir özellik çıkarıcıdır. DensNet (Huang vd., 2017) tabanlı ağ CSPDarknet53, temel katmanın özellik haritasını Cross Stage Partial connections (CSP) bloklarını kullanarak iki parçaya böler ve ardından iki bölümü birleştirmek için bir çapraz aşama hiyerarşisi kullanır. Bu strateji gradyan yayılımını iyileştirir, özelliklerin daha etkili bir şekilde yeniden kullanılmasını sağlar ve gradyanın kaybolması sorununu ele alarak modelin öğrenme kapasitesini artırır. Mimarının boyun kısmında, SPP ve PAN

modülleri farklı aşamalardan gelen özellik haritalarını bir araya getirmek için kullanılır. SPP, özellik haritalarını çeşitli boyutlardaki kutulara ayırır ve her birine maksimum havuzlama uygular. Böylece, çeşitli ölçeklerde nesne tespiti için uygun sabit uzunlukta bir temsil oluşturulur. PAN, düşük seviyeli bilgileri daha yüksek katmanlara verimli bir şekilde iletmek için kısa yol yolları kullanarak katmanlar arasında bilgi entegrasyonunu sağlar YOLOv4 ağında üç algılama başlığı mevcuttur. Nihai tahminler, her bir algılama kafasının içinde bir YOLOv3 ağı kullanılarak hesaplanır. YOLOv4 ağı, sınırlayıcı kutuları, sınıflandırma puanlarını ve nesnellik puanlarını tahmin etmek için 19×19 , 38×38 ve 76×76 boyutlarında özellik haritaları oluşturur. YOLOv4'ün kayıp fonksiyonunun üç ana bileşeni vardır:

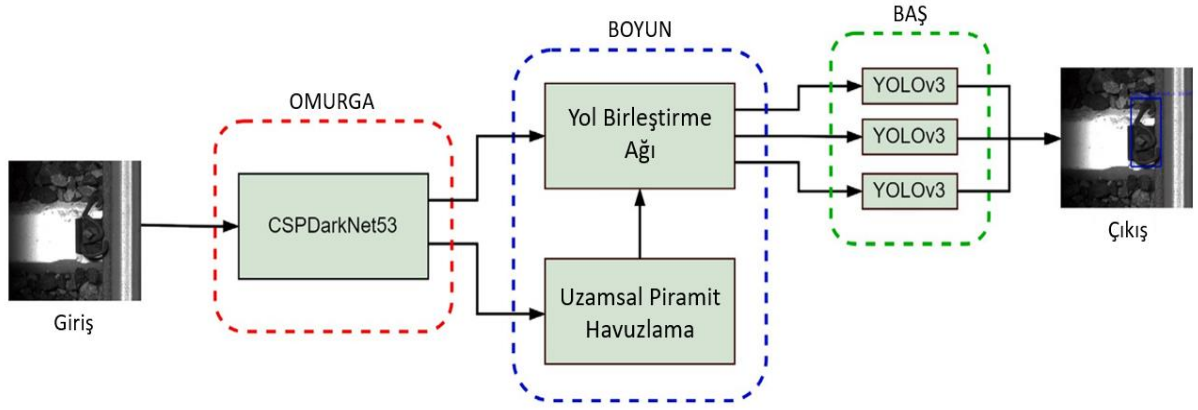
- (i) Konumlandırma kaybı (Localization loss - $\mathcal{L}_{loc}$): Konumlandırma kaybı, tahmin edilen ve gerçek sınırlayıcı kutu konumları arasındaki farkları cezalandırarak modeli nesneleri doğru şekilde konumlandırmaya teşvik eder. Ortalama karesel hata (MSE), tahmin edilen sınırlayıcı kutu koordinatlarını yer gerçeği koordinatlarıyla karşılaştırmak için kullanılır.
- (ii) Güven Kaybı (Confidence loss - $\mathcal{L}_{conf}$): Güven kaybı, sınırlayıcı kutu içinde bir nesnenin varlığını gösteren beklenen ve temel gerçek güven puanları arasındaki farktır. YOLOv4 şunları uygular bu bileşen için ikili çapraz entropi kaybı.
- (iii) Sınıf kaybı (Class loss - $\mathcal{L}_{cls}$): YOLOv4, tahmin edilen sınıf olasılıklarını gerçek sınıf etiketleriyle karşılaştırmak için kategorik çapraz entropi kaybını (cross-entropy loss) kullanır ve kayıp fonksiyonunun sınıf tahmini bileşenini oluşturur.

YOLOv4'ün kayıp fonksiyonu, aşağıdaki gibi bu üç kaybın doğrusal kombinasyonudur:

$$\mathcal{L} = \lambda_{loc} \mathcal{L}_{loc} + \lambda_{conf} \mathcal{L}_{conf} + \lambda_{cls} \mathcal{L}_{cls} \quad (2.1)$$

burada λ_{loc} , λ_{conf} ve λ_{cls} lokalizasyon, güven ve sınıflandırma kayıplarına karşılık gelen ağırlık katsayılarıdır.

YOLOv4 mimarisinin seviye katmanlarının özet gösterimi Şekil 2.10'da verilmiştir.



Şekil 2.10. YOLOv4 mimarisinin seviye katmanlarının özet gösterimi

Kaynak: (Ozdemir & Koc, 2024)

YOLOv5, PyTorch ile yazılmıştır ve birçok önceden eğitilmiş model sunar. YOLOv4'ün optimizasyonlarına ek olarak daha iyi bir kullanıcı deneyimi sunar, örneğin, daha kolay eğitim ve uygulama süreçleri. Model boyutu, hız ve doğruluk arasında çeşitli denge sağlayan farklı versiyonlar sunar (YOLOv5s, YOLOv5m, YOLOv5l, YOLOv5x). Kullanım kolaylığı ve esneklik açısından öne çıkar, çeşitli modellerle farklı gereksinimlere uyum sağlar (Jocher, 2020).

YOLOv6, YOLOv5'in geliştirilmiş bir versiyonudur, daha hızlı ve daha doğru nesne tespiti sağlar. Yeni optimizasyonlar ve veri artırma teknikleri ile eğitim ve genel performans iyileştirilmiştir. YOLOv5'e göre daha yeni optimizasyonlar ve iyileştirmeler içerir, ancak henüz yaygın olarak benimsenmemiştir (C. Li vd., 2022).

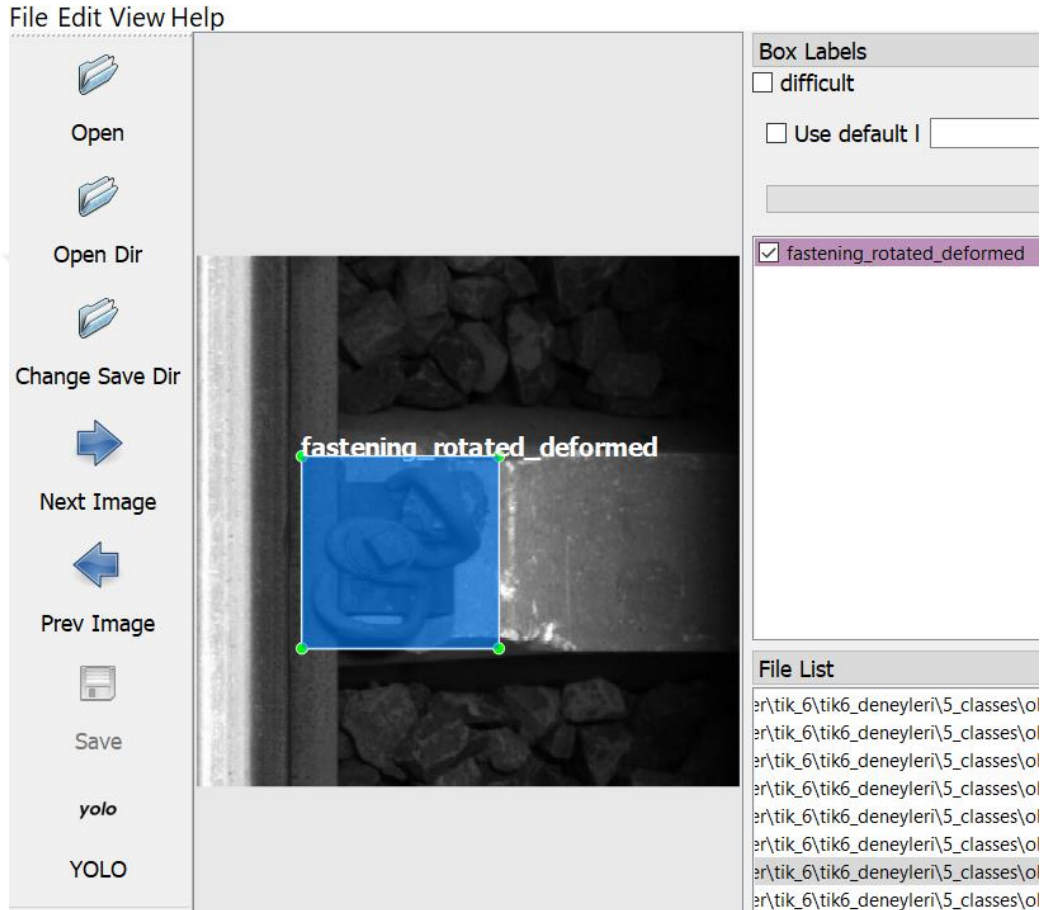
YOLOv7, serinin nispeten yeni sürümlerinden biridir ve daha iyi genel performans için çeşitli yeni teknikler ve optimizasyonlar içerir. YOLOv6'ya göre daha fazla iyileştirme ve optimizasyon sunar. En son teknikler ve optimizasyonlarla geliştirilmiştir, bu da daha iyi doğruluk ve hız sağlar (C.-Y. Wang vd., 2022).

YOLOv8, YOLO ailesinin en yeni sürümlerinden biri olarak, önceki versiyonların tüm avantajlarını içerir ve daha fazla optimizasyon ve yenilik sunar. Model mimarisi ve eğitim süreçlerinde daha fazla iyileştirme yapılmıştır. En son yenilikler ve optimizasyonlarla donatılmıştır, dolayısıyla en yüksek verimlilik hedeflenmiştir.

Bu sürümler, her biri kendi dönemlerinde çeşitli iyileştirmeler ve yenilikler sunarak nesne tespitinde daha iyi doğruluk ve hız sağlamak için geliştirilmiştir. Her yeni sürüm, önceki sürümlerin eksikliklerini gidererek ve yeni teknikler ekleyerek daha üstün performans sunar (Torres, 2024).

2.5. Veri Etiketleme Aracı

Seçilen görüntülerin nesne tespit algoritmalarının eğitiminde kullanılabilmesi için belirli formatlarda etiketlenmesi gerekmektedir. Görüntülerin etiketlenmesi için geliştirilmiş bazı araçlar bulunmaktadır. Veri kümesindeki imgeler etiketlenirken LabelImg aracı kullanılmıştır (labelImg, 2015). Örnek olarak YOLO formatındaki bir görüntünün labelImg aracı kullanılarak etiketlenmesi Şekil 2.11'de gösterilmiştir.



Şekil 2.11. labelImg'de etiketleme işlemi

Kaynak: (Ozdemir & Koc, 2024)

2.6. Değerlendirme Metrikleri

Deneylerin etkinliği ortalama kesinlik, geri çağırma, F_1 -skor, Intersection over Union (IoU) ve mAP gibi çeşitli değerlendirme ölçütleri kullanılarak doğrulanmıştır. Değerlendirme metriklerinin formülleri Tablo 2.5'te özetlenmiştir. Burada C_i kategori sayısını, AP_i , i kategorisinin hassasiyetini ifade ederken, gerçek pozitif, yanlış pozitif ve yanlış negatif ise sırasıyla TP , FP , FN ile temsil edilmektedir. Ayrıca IoU ise, tahmin edilen ile gerçek alanın kesişiminin; tahmin edilen ile gerçek alanın birleşimine olan oranını ifade eder.

Tablo 2.5. Değerlendirme ölçütleri için formüller

$\text{Ortalama kesinlik} = \frac{TP}{TP + FP}$
$\text{Geri çağırma} = \frac{TP}{TP + FN}$
$F_1 - \text{skor} = 2 \times \frac{\text{Ortalama kesinlik} \times \text{Geri çağırma}}{\text{Ortalama kesinlik} + \text{Geri çağırma}}$
$\text{IoU} = \frac{TP}{TP + FP + FN}$
$\text{mAP} = \frac{1}{C} \sum_{i=0}^C AP_i$

3. SÖZDE ETİKETLER KULLANARAK YARI DENETİMLİ DERİN ÖĞRENME TABANLI DEMİRYOLU KUSUR TESPİTİNİN İYİLEŞTİRİLMESİ

Son yıllarda, derin öğrenme yöntemlerinin demiryolu arıza tespitinde kullanımı, teknolojinin arıza teşhisinde doğruluğu ve verimliliği artırma kabiliyeti nedeniyle hızla yaygınlaşmıştır. Bununla birlikte, derin öğrenme uygulamalarının modellenmesi büyük miktarda veri gerektirir ve bu tür veri kümeleri için etiketleme işlemi önemli ölçüde insan gücü gerektirir. Bu boşluğu gidermek amacıyla, görüntüleri otomatik olarak etiketleyerek görüntü sınıflandırmada gereken manuel çabayı azaltmayı hedefleyen ve YOLOv4 tabanlı öğrenci-öğretmen modeli kullanan yarı denetimli bir öğrenme yaklaşımı sunulmaktadır. Bu bölümde ayrıca, TCDD tarafından sağlanan görüntülerden oluşturulan ve beş farklı demiryolu kusurunu içeren yeni bir veri kümesi oluşturulmuştur. Aynı test veri kümesi üzerinde gerçekleştirilen deney sonuçları, önerilen öğrenci-öğretmen modelinin yalnızca YOLOv4'ün tespit performansını çeşitli karar ölçütlerine göre iyileştirmekle kalmayıp, aynı zamanda eğitim veri kümesini yüksek güvenilirliğe sahip sözde etiketli görüntülerle genişlettiğini de ortaya koymaktadır.

3.1. Giriş

Demiryolu hatlarının arıza tespit uzmanı yardımıyla manuel ve görsel olarak denetlenmesi artık modası geçmiş bir yöntem haline gelmiştir. Bunun en önemli nedeni, sürecin çok sıkıcı ve zaman alıcı olmasıdır. Ayrıca, kontrol işlemini gerçekleştiren kişinin bilgi ve becerisiyle sınırlı, tehlikeli sonuçlara yol açabilecek öznel bir süreçtir (Wilson & Norris, 2005). Temas halinde arızaları teşhis etme süreci bazen demiryolu yüzeyindeki kusurların artmasına veya yeni arızaların ortaya çıkmasına neden olabilir. Ayrıca, sistem üst yüzeydeki arızaları tespit edebilmekte ancak demiryolu raylarının yan yüzeyindeki arızaları tespit edememektedir (Jie vd., 2009). Geleneksel arıza tespit yöntemlerinin sınırlamaları göz önüne alındığında, gerçek zamanlı demiryolu arıza tespiti artık gelişmiş bilgisayarla görme uygulamaları kullanılarak gerçekleştirilmektedir. Temassız yöntemler olarak adlandırılan bu yöntemler, teşhisi gerçekleştirmek için dijital kameralar ve ışık kaynakları gibi özel ekipmanlar gerektirir. Bu ekipmanlar demiryoluna fiziksel olarak temas etmediğinden, geleneksel arıza teşhisine göre daha etkili ve verimli bir alternatif sunmaktadır (Z. Liu vd., 2011). Temassız yöntemler, temaslı yöntemlerde oluşabilecek dezavantajları ortadan kaldırmakla kalmayıp daha hızlı ve başarılı sonuçlar vermektedir (Y. Santur vd., 2016 t.y.; Taştımur vd., 2015). Bununla birlikte, demiryolu üzerinde meydana gelen yağ ve toz lekeleri

çevresel faktörler algoritma tarafından hata olarak teşhis edilebilir. Bu, bu sistemlerin önemli bir dezavantajıdır (Feng vd., 2014).

Basit görüntü işleme (Q. Li & Ren, 2012), geleneksel makine öğrenimi (Z. Liu vd., 2011; Min vd., 2018), derin öğrenme (Faghih-Roohi vd., 2016; F. Guo vd., 2021; Hsieh vd., 2020; James vd., 2018; Y.-W. Lin vd., 2019; S. Mittal & Rao, 2018; Qi vd., 2020; Rampriya vd., 2022; Santur vd., 2017; Şener vd., 2022; Xu vd., 2021; Zheng vd., 2021) veya bu yöntemlerin herhangi bir kombinasyonunu (Aydin vd., 2022; Giben vd., 2015a; Gibert vd., 2017; Güçlü, Aydin, vd., 2021; Santur, 2018; Wei vd., 2019) kullanan çeşitli çalışmalar literatüre sunulmuştur.

Ping ve arkadaşları, deformasyon ve çatlakları tespit eden bir sistem tasarladı, demiryolu yüzeyinde tespit yöntemleri (Z. Liu vd., 2011). Mittal ve Rao, derin öğrenme yöntemiyle demiryollarında görüntü tabanlı hata takibi ve nesne tespiti gerçekleştirmiştir (S. Mittal & Rao, 2018). Malzeme sınıflandırması için uçtan uca eğitilmiş Evrişimsel Sinir Ağının (CNN), elle tasarlanmış özelliklere sahip sığ öğrenme yöntemlerinden daha başarılı ve gürültüye karşı daha dayanıklı olduğunu göstermişlerdir. Demiryolu yüzey kusurlarını tespit etmeyi amaçlayan Faghih-Roohi ve arkadaşları, CNN'lerin performansları farklı aktivasyon fonksiyonları kullanılarak farklı katman sayıları oluşturulmuştur (Faghih-Roohi vd., 2016). Li ve Ren tarafından yapılan çalışmada ise ray yüzeyinde oluşabilecek ezilme, ondülasyon ve ray kırığı tespit edilmiştir (Q. Li & Ren, 2012). Çalışma için sistemden gerçek zamanlı olarak alınan görüntüler bazı görüntü işleme yöntemleri uygulandıktan sonra sınıflandırılmıştır. Gibert ve arkadaşları çoklu kombinasyon dedektörleri için çoklu görev öğrenme çerçevesinde bir konsept önermişlerdir (Gibert vd., 2017). Birden fazla görevi öğrenen bu sistem, iz materyallerini 10 kategoride sınıflandırmak için bir dedektörde birleştirir. Destek Vektör Makinesi (SVM) sınıflandırıcılarını kullanarak bağlantı elemanlarındaki kusurları tanımlar. Bunlar bir CNN aracılığıyla tek bir sistemde birleştirilir. Santur ve arkadaşları, ray hattının üst ve yan yüzeylerinde oluşabilecek çatlak, mantar yırtılmaları, ondülasyon ve ezilme gibi arıza sınıflarını tespit etmektedir (Santur vd., 2017). Test görüntüleri sağlıklı ve hatalı olarak kategorize edilmiştir. Min ve arkadaşları, yerleşik bir led ek ışık kaynağına sahip görüntü toplama sistemi geliştirmiş ve makine öğrenimi ve görüntü işleme algoritmaları kullanarak demiryollarında gerçek zamanlı yüzey hatası tespiti için bir teknik önermiştir (Min vd., 2018). James ve arkadaşları yaptıkları çalışmada, birden fazla aşamadan oluşan derin öğrenme tabanlı bir demiryolu yüzey kusuru tespit yöntemi önermiştir (James vd., 2018). İlk adımda segmentasyon yapılmakta, ardından ilgilenilen bölgeye odaklanmak için görüntü

kırılmaktadır. Santur, lazer kameralarla demiryolu kusurlarının gerçek zamanlı teşhisi için görüntü işleme ve makine öğrenimi tabanlı yöntemler önermiştir. Ayrıca, doktora tezinde derin öğrenme tabanlı bir demiryolu arıza teşhis yöntemi önerilmiştir (Santur, 2018). Wei ve arkadaşları, görüntü işleme teknikleri, VGG-16 ve Faster R-CNN arasında bağlantı elemanları için kusurları tespit etme kapasitesinin performansını karşılaştırmıştır. Faster R-CNN'nin geri çağırma ve ortalama kesinlik oranlarıyla değerlendirildiğinde en iyi performansa sahip olduğunu bulmuşlardır (Wei vd., 2019). Şener ve arkadaşları, tasarladıkları sığ CNN modeli ile kusurlu ve kusurlu olmayan demiryolu görüntülerini sınıflandırmıştır (Şener vd., 2022). Bu çalışma ile az sayıda etiketli veri kullanımı ile başarılı sonuçlar elde edilebileceği gösterilmiştir. Sıfırdan bir ağ eğitmek yerine ResNet ile transfer öğrenme uygulamışlardır. Giben ve arkadaşları ise, demiryolu görüntülerinde görsel kusur tespiti için CNN ile malzeme sınıflandırması ve semantik segmentasyon kullanan bir teknik önerilmiştir (Giben vd., 2015b). Aydın ve arkadaşları, görüntü ön işleme teknikleri ve sığ bir CNN kullanarak bağlantı elemanlarındaki hataları kategorize etmek için bir çalışma sunmaktadır (Aydın vd., 2022). Sığ CNN'in iki tam bağlı katmanının çıktılarının birleştirilmesiyle elde edilen özellikler, bir dizi makine öğrenimi algoritması aracılığıyla analiz edilmiş ve en etkili sınıflandırıcı belirlenmiştir.

Popüler derin öğrenme tabanlı nesne tespit algoritmalarından YOLO ve türevleri demiryolu kusur tespit problemleri için sıklıkla kullanılmaktadır. Lin ve arkadaşları YOLOv3 algoritması ile ray bağlantı elemanlarının kondisyonunu belirlemişlerdir. Bağlantı elemanı tespitinde geri çağırma ve ortalama kesinlik oranlarını sırasıyla %89 ve %95 olarak bulmuşlardır (Y.-W. Lin vd., 2019). Qi ve arkadaşları, demiryollarındaki bağlantı elemanlarının gerçek zamanlı tespiti için MYOLOv3-Tiny adlı yeni bir model sunmuştur. Bu model, YOLOv3-Tiny mimarisinden daha yüksek hassasiyet elde etmiş ve bellek kullanımı %43 oranında azaltılmıştır (Qi vd., 2020). Hsieh ve arkadaşları, sıradan bir RGB kamera ve LED aydınlatma ile bir veri kümesi oluşturarak kusur teşhisi için YOLOv3 mimarisini kullanmıştır (Hsieh vd., 2020). Wang ve arkadaşları, çalışmalarında gerçek zamanlı olarak ray, civata ve bağlantı tespiti için YOLOv2 modelini kullanmıştır. Bu çalışmada, 35 kare/saniye'de %93 mAP değeri elde edilmiştir (C.-Y. Wang vd., 2019). Guo ve arkadaşları, demiryolu bileşen tespiti için YOLOv4-hibrit modelini önermektedir. Hem doğruluk hem de işlem hızı açısından karşılaştırdıkları tek aşamalı modellerden daha iyi sonuçlar elde etmişlerdir (F. Guo vd., 2021). Güçlü ve arkadaşları, YOLOv4 ve bulanık mantığa dayalı hibrit bir yöntem geliştirmiştir. Bağlantı elemanları 6 farklı bölgeye ayrılarak YOLOv4 ile

belirlenmiş, ardından veriler bulanık mantık yapısı için girdi olarak kullanılarak bağlantı elemanlarının sağlık durumu kontrol edilmiştir (Güçlü, Aydın, vd., 2021). Zheng ve diğerleri, geliştirilmiş YOLOv5 ve Maskelenmiş R-CNN'nin sıralı kullanımını kullanarak çoklu nesne tespiti için bir yöntem önermektedir (Zheng vd., 2021). Geliştirilmiş YOLOv5, nesnelerin konumlarının ilk tespiti için ve raylar üzerindeki yüzey kusurlarını belirlemek için kullanılan maskeli R-CNN tabanlı bir kusur tespiti için kullanılır. Son olarak, ResNet çerçeve tabanlı model daha sonra konektörlerin durumunu kategorize etmek için kullanılır.

Nesne algılama algoritmaları için bir eğitim veri kümesi oluşturmak, çok sayıda görüntünün manuel olarak etiketlenmesi gerektiğinden zaman alıcı ve maliyetli bir iştir. Nesne tespit algoritmasının performansı da eğitim veri kümesindeki görüntü sayısı ile orantılıdır. Burada, manuel çalışmayı çok fazla artırmadan etiketli görüntü sayısını artırma ihtiyacı ortaya çıkmaktadır. Yarı denetimli öğrenme (SSL) yöntemleri sayesinde etiketsiz görüntülerin yapay etiketleri bulunarak etiketli veri ihtiyacı azaltılabilir. Popüler SSL yöntemlerinden biri, etiketsiz görüntüler için yapay etiketlerin üretildiği ve modelin yapay olarak etiketlenmiş görüntüler kullanılarak eğitildiği sözde etiketlemedir. Sözde etiketlemede, modelin sınıf tahmini eğitmek için bir etiket olarak kullanılır (Lee, 2013). Sözde etiketleme yöntemleri kullanarak nesne tespit algoritmalarının performansının artırıldığı gözlemlenmiştir (Kaul vd., 2022; Pham vd., 2021; Riloff, 1996; Sohn vd., 2020; S. Wang vd., 2022; Xie vd., 2020; Xu vd., 2021).

Pham ve arkadaşları, öğretmenin öğrencinin etiketli veri kümesindeki performansına bağlı olarak güncellendiği bir öğretmen-öğrenci modeli önermiştir (Pham vd., 2021). Riloff, etiketlenmemiş metinden otomatik olarak bilgi çıkartma kalıpları (extraction patterns) oluşturmak için bir dizi sezgisel kuralı kullanan bir algoritma önerilmiştir (Riloff, 1996). Xie ve arkadaşları, ImageNet'in etiketli görüntülerini kullanarak bir EfficientNet modeli eğitmiş ve bunu çok sayıda etiketsiz görüntüyü etiketlemek için bir öğretmen olarak kullanmıştır (Xie vd., 2020). Etiketlenmemiş verileri kullanarak ImageNet'in son teknoloji modellerinin sağlamlığını ve doğruluğunu artırmanın mümkün olduğunu göstermişlerdir. Xu ve arkadaşlarının çalışmasında, yarı denetimli öğrenme stratejisi, eğitim için bir öğrenci modeli ve sözde etiketleme için öğrenci modelinden sürekli güncellemeler alan bir öğretmen modeli kullanarak dedektörü ve sözde etiketleri eşzamanlı olarak iyileştirmektedir (Xu vd., 2021). Kaul ve arkadaşları bir doğrulama yaklaşımı kullanarak yanlış sınıf etiketlerine sahip aday tespitleri eleyebilmekte ve eğitim setindeki her yeni nesne sınıfı için üstün sözde açıklamalar üretebilmektedir (Kaul vd., 2022). Bir başka yenilikçi çalışmada, Google Cloud AI ve Google

Brain çalışanları Kihyuk Sohn ve arkadaşları, denetimli bir öğrenme yöntemi yerine Kendi Kendini Eğitime ve Artırmaya Dayalı Uyum Düzenlemesi (STAC) adı verilen özel bir yarı denetimli öğrenme yöntemi kullanarak mAP değerini %76,30'dan %79,08'e çıkarmıştır (Sohn vd., 2020). Wang ve arkadaşları, demiryolu ray yüzey kusurlarını incelemek için yarı denetimli bir öğrenme kusur sınıflandırma yaklaşımı önermiştir (S. Wang vd., 2022). Peng ve arkadaşları, yalnızca anormallik içermeyen örneklerden ve ek açıklamalardan öğrenen yeni bir yarı denetimli anormallik tespit algoritması önermiştir (Peng vd., 2023). Guo ve arkadaşları, uç cihazlardaki demiryolu raylarının hassas, gerçek zamanlı denetimi için tasarlanmış bir bilgisayarla görme mimarisi olan AWL-NanoDet'i önermiştir (J. Guo vd., 2023). Bu model, eğitim aşamasında öğretmen-öğrenci rehberlik sistemini güncelleyen bir Uyarlanabilir Ağırlıklı Kayıp (Adaptive Weighted Loss) yaklaşımı kullanmaktadır. Jiang ve arkadaşları, demiryolu ray yüzey kusurlarının yüksek hızlı ve hassas tespiti için geliştirilen RTDS-Net adlı bir derin öğrenme mimarisi önermiştir (Jiang vd., 2023). Mimari ile %95.2 algılama doğruluğu ve 128 kare/saniye (fps) hıza ulaşılmıştır.

Demiryolu kusurları için derin öğrenmeyi uygulamak birçok zorluğu beraberinde getirmektedir. Bunlardan biri, çeşitli demiryolu kusurlarını ve çeşitli çevresel koşulları içeren kapsamlı, yüksek kaliteli, etiketli, genel erişime açık bir veri setinin olmamasıdır. Güvenlik ve bakım iş akışlarını etkilediği için, çok fazla yanlış pozitif tahmin veya gözden kaçan kusur olmaması adına, veri kümesindeki örnek görüntülerin çeşitliliğini dengelemek de çok önemlidir. Literatürde demiryollarında kusur tespiti için bazı araştırmacıların kendi özel veri kümelerini kullandığı görülmektedir. Rampriya ve arkadaşları Rail Obstacle Detection Dataset'i (RODD) oluşturmuş ve bu veri kümesini çeşitli nesne algılama modellerini eğitmek için kullanmıştır. Modeller eğitildikten sonra performans ölçütlerine göre karşılaştırmalar yapılmıştır (Rampriya vd., 2022). Hsieh ve arkadaşları, sıradan bir RGB kamera ve LED aydınlatma kullanarak altı normal ve dört hasarlı bağlantı elemanı sınıfı içeren bir veri kümesi oluşturmuştur (Hsieh vd., 2020). Feng ve arkadaşları 110 km'lik bir demiryolu hattından elde edilen 57.124 bağlantı elemanı görüntüsünü içeren bir veri kümesi kullanmıştır (Feng vd., 2014). Kısmen aşınmış ve tamamen eksik bağlantı elemanlarını tespit etmek için olasılıksal bir model kullanan otonom bir görsel denetim yöntemi önermektedirler. Faghieh-Roohi ve arkadaşları deneylerinde 350 kilometrelik demiryolu hattından toplanan ve bir normal ve altı farklı seviyede ezilme sınıfı içeren bir veri kümesi kullanmışlardır (Faghieh-Roohi vd., 2016). Yukarıdaki çalışmalarda kullanılan veri kümelerinin hiçbiri kamuya açık değildir ve hiçbiri bilinen beş tür demiryolu kusurunu içermemektedir. Gan ve arkadaşları, otomatik bir görsel

denetim sistemini değerlendirmek için sırasıyla 67 Tip-I ve 128 Tip-II ray yüzeyi görüntüsü içeren bir veri kümesi kullanmıştır. Bu görüntüler ekspres ve genel/ağır yük demiryollarındaki çeşitli kusurları yakalamaktadır (Gan vd., 2017).

Tezin bu bölümünde, demiryolu kusur tespiti için derin öğrenme modellerinde gerekli olan büyük veri kümelerini etiketleme zorluğunu gidermek amacıyla, yarı denetimli bir sözde etiketleme algoritması kullanarak görüntüleri etiketleyen bir öğrenci-öğretmen modeli önerilmiştir. Yöntemde ilk olarak sınırlı sayıda görüntü manuel olarak etiketlenerek öğretmen modeli için bir eğitim veri kümesi oluşturulur. Nesne algılama algoritması, bir öğretmen modeli oluşturmak için bu veri kümesi kullanılarak eğitilir. Daha sonra veri kümesi, öğretmen modeli sayesinde otomatik olarak elde edilen sözde etiketli görüntülerle genişletilir. Bu süreç, eğitim veri kümesinin oluşturulmasında yer alan bazı manuel adımları otomatikleştirerek zamandan tasarruf sağlar ve maliyetleri azaltır. Bu bölümde ayrıca TCDD'nin bir parçası olan Demiryolu Araştırma ve Teknoloji Merkezi (DATEM) tarafından sağlanan görüntüleri kullanarak bir veri kümesi oluşturulmuştur. Veri kümesi, 5 temel demiryolu kusurunun, yani mantar yırtılmaları, ray kırığı, hasarlı bağlantı elemanı, ezilme ve yüzey kusurunun not edildiği görüntülerden oluşmaktadır. Bildiğimiz kadarıyla, çeşitli demiryolu kusurlarını içeren kamuya açık dengeli bir veri kümesi bulunmamaktadır.

Bu bölümdeki çalışmanın katkıları aşağıdaki gibi özetlenebilir:

- YOLOv4'ün tespit performansı, eğitim aşamasında öğrenci-öğretmen modeli tarafından otomatik olarak etiketlenen görüntüler kullanılarak geliştirilmiştir.
- Bir veri kümesi oluşturmak zorlu ve zaman alıcı bir iştir. Beş yaygın demiryolu kusurundan oluşan bir veri kümesi oluşturulmuştur. Aynı veri kümesini kullanan farklı yaklaşımların, adil bir şekilde karşılaştırılmasına olanak sağlamak için hem etiketli hem de etiketsiz veri kümelerinin kamuya açık hale getirilmesi amaçlanmıştır.

3.2. Malzeme ve Yöntemler

3.2.1. Veri Kümesi

Demiryolu arızaları üzerine bir makine öğrenmesi çalışması yapabilmek için ihtiyaç duyulan ilk gereksinim arızalı görüntülerden derlenmiş bir veri kümesidir. İlgili alandaki veri kümesi içerisindeki bir görüntüyü makine öğrenmesi çalışmasında kullanabilmek için gerekli işaretleme ve sınıflandırmaların yapılmış olması gerekmektedir. Bu yüzden TCDD'den elde edilen demiryolu görüntülerine, makine öğrenmesi algoritması eğitiminde kullanılabilmesi için işaretleme ve etiketleme işlemleri uygulanması gerekmektedir. Ayrıca veri kümesinde

kullanmak için yeterli kalitede olmayan bazı görüntüler de vardı. Yapılan sadeleştirmeler sonucunda makine öğrenmesi modeli eğitiminde kullanılabilecek bir veri kümesi elde edilmiştir.

Bu bölümdeki çalışmanın amacı demiryolu kusurlarını saptayacak bir nesne tespit modeli geliştirmek olduğu için veri kümesinin kusurlu görüntü içermeyen kısımları kullanılmamıştır. Kusurların tespitine katkısı olmayacak, cıvata, sağlam bağlantı elemanı, yanlış hizalanmış traversler ve traversler gibi sınıflar elenerek bu bölümdeki çalışma için oluşturulacak veri kümesinden çıkartılmıştır. Bunun yerin veri kümesinde kalan; yüzey kusurları, ray kırığı, hasarlı bağlantı elemanı, ezilme ve mantar yırtılmaları içeren görüntüler hazırlanan veri kümesinde kullanılmıştır.

Mantar yırtılmaları, ezilme ve yüzey kusurları, yüzey kusuru sınıfını oluşturan üç alt kategoridir. Hasarlı bağlantı elemanı, ezilme, yüzey kusurları ve raylardaki çatlaklarla ilgili görüntüleri inceledikten sonra, önemli miktarda yanlış pozitif bulunmuştur. Bu hatalar, bağlantı elemanı üzerine balast sıçraması, çevrede ot ve diğer yabancı cisimlerin büyümesi, kablo bağlantıları, kaplumbağa gibi hayvanların varlığı, ray yüzeyinde kuş pisliği ve yağ lekeleri gibi çevresel faktörlerden kaynaklanmaktadır.

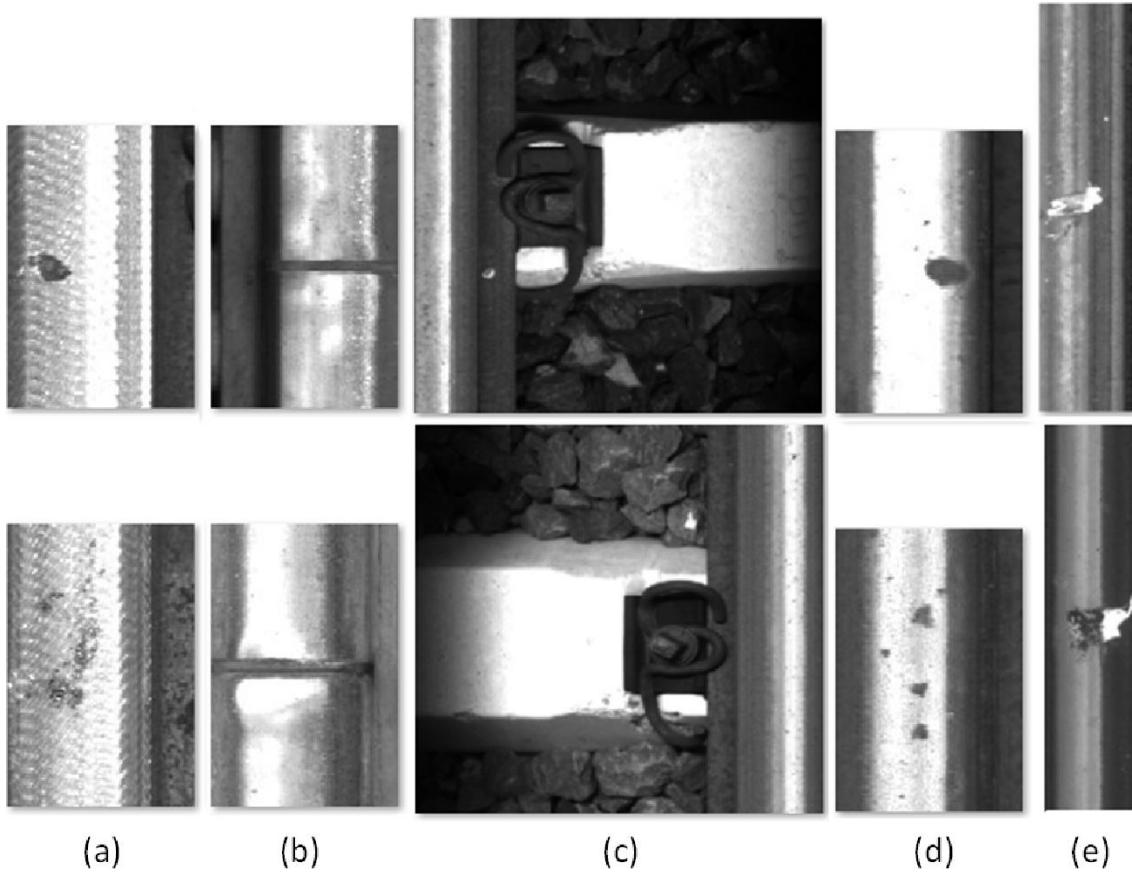
Nesne tespit algoritmasının eğitiminde performansı etkileyen en önemli faktörlerden biri veri kümesidir (Lu & Weng, 2007). Eğitim veri kümesindeki etiketlemenin tutarlılığı ve doğruluğu, modelin test aşamasındaki performansını doğrudan etkiler. Sınıflardaki görüntü sayısının dengesi de doğruluğu etkileyen bir diğer faktördür. Eğitim veri setindeki farklı nesne sınıflarına ait örnek sayısındaki dengesizlik, nesne tespit algoritmasının tespitlerinde örnek sayısı daha fazla olan sınıflara doğru yanlılığa neden olabilir (Fanny & Cenggoro, 2018; Johnson & Khoshgoftaar, 2019). Ayrıca dengeli veri seti, model üzerinde yapılan iyileştirmelerin sonuçlarının daha iyi incelenmesinde önemli bir rol oynamaktadır. Ray kırığı hataları çok nadir görüldüğünden, ray kırığı sınıfına ait görüntü sayısı diğer kusur sınıflarına kıyasla çok azdır. Bu eksikliği gidermek ve dengeli bir veri seti oluşturmak amacıyla, bu sınıfa özel bir dikkat gösterilmiştir. Her sınıftan en az 40 örnek seçilerek, 5 sınıf için toplam 208 görüntü içeren bir eğitim veri kümesi hazırlanmıştır. Görüntülerin bazıları birden fazla sınıfa ait nesneler içerdiğinden 208 görüntüdeki toplam nesne sayısı 240'tır. Her sınıftan örnek sayısı Tablo 3.1'de verilmiştir. LabelImg etiketleme programı kullanılarak veriler etiketlenmiştir.

Tablo 3.1. Eğitim ve Test Veri Kümesi Görüntü dağılımı

	Mantar yırtılmaları	Ray kırığı	Hasarlı bağlantı elemanı	Ezilme	Yüzey kusurlar
Eğitim seti	40	40	40	80	40
Test seti	20	23	20	37	20

3.2.2. Test Veri Kümesi

Manuel olarak etiketlenen görüntülerden 108 tanesi test veri kümesini oluşturmak için ayrılmıştır. Bu test veri kümesinde 5 farklı sınıfa ait toplam 120 nesne bulunmaktadır. Test kümesinden örnek görüntüler Şekil 3.1'de verilmiştir. Tablo 3.1'de test veri kümesindeki her bir nesne için örnek sayısı verilmiştir.



Şekil 3.1. Test setinden örnek görüntüler. (a) mantar yırtılmaları, (b) ray kırığı, (c) hasarlı bağlantı elemanı, (d) ezilme ve (e) yüzey hatası.

Kaynak: (Ozdemir & Koc, 2024)

3.2.3. Öğrenci-Öğretmen Modeli

Nesne tanıma algoritmaları için bir veri kümesi oluşturma süreci, görüntülerin manuel olarak etiketlenmesini içerdiğinden maliyetli ve zaman alıcı olabilir. Etiketleme sürecinin bazı kısımları otomatikleştirilerek bu maliyetler azaltılabilir. Etiketleme sürecinin bir kısmını

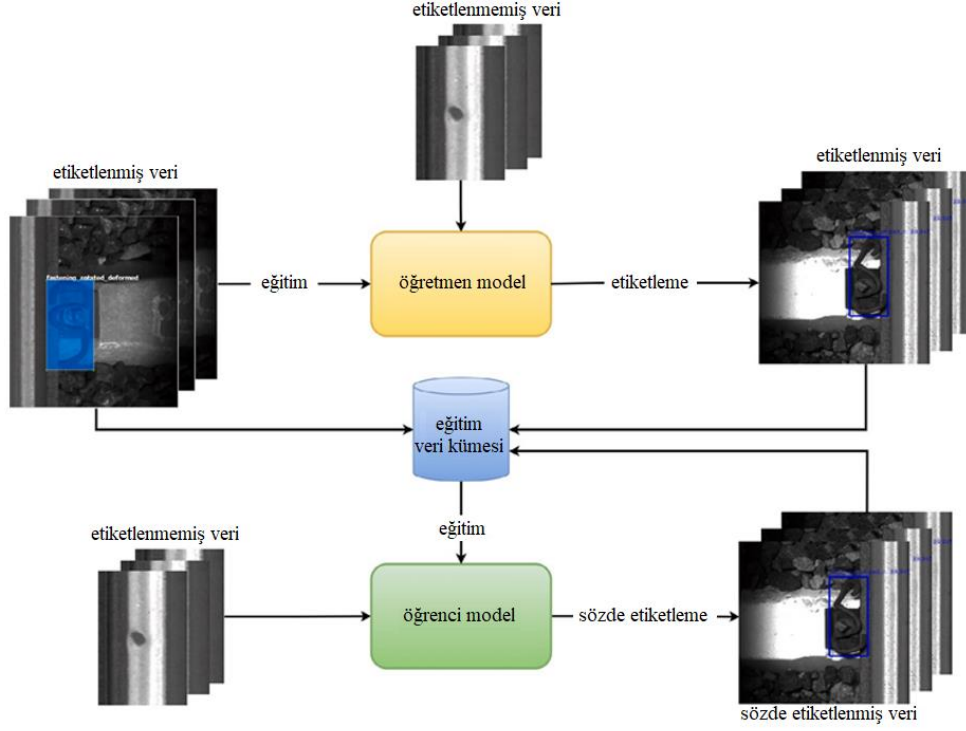
otomatikleştirmek ve maliyetleri azaltmak için sınırlı sayıda manuel olarak etiketlenmiş görüntü üzerinde eğitilmiş bir öğrenci-öğretmen modeli önerilmiştir. Öğretmen modelinde nesne algılama algoritması olarak YOLOv4'ü kullandık. Öğretmen modeli, yine öğretmen modelinin kendisi tarafından otomatik olarak etiketlenen yeni sözde etiketli görüntüleri girdi olarak alır. Öğrenci-öğretmen modeli iki aşamada elde edilir. İlk aşama, sınırlı sayıda manuel olarak etiketlenmiş görüntü kullanarak nesne algılama algoritmasının eğitilmesini içerir. Ardından, eğitilen nesne algılama algoritması kullanılarak etiketsiz görüntülerdeki nesnelerin konumları ve sınıfları tahmin edilir. Yüksek tahmin güven puanlarına sahip olanlar seçilerek sözde etiketli görüntülerden bir veri kümesi oluşturulur. İkinci aşamada, manuel olarak etiketlenmiş görüntülerin yanına sözde etiketlenmiş görüntüler de eklenerek eğitim veri kümesi genişletilir.

Öğrenci-öğretmen modeli algoritmasının genel bir bakışı Şekil 3.2'de verilmiştir.

$\mathcal{X}_L = \{(\mathbf{x}_i, y_i), i = 1, \dots, M\}$ manuel olarak etiketlenmiş veri kümesi, N örnek görüntü ve M etiketli nesne sayısını gösterirken $N \leq M$ ve $y_i \in \{1, \dots, C\}$ de $\mathbf{x}_i$ 'nin sınıfı olsun. Eğer etiketlenmemiş nesnelerden oluşan veri kümesini $\mathcal{X}_U = \{\mathbf{x}_j\}$ ile ifade edersek ve modelin tahminleri ile gerçek değer etiketleri arasındaki kayıp $\mathcal{L}_L(\mathcal{X}_L) = \sum_{i=1}^M \mathcal{L}(\mathbf{x}_i, y_i; y'_i)$ ifadesiyle gösterilirse, önerilen öğrenci-öğretmen modelin kayıp fonksiyonu şu şekilde tanımlanabilir:

$$\mathcal{L}(\mathcal{X}_L, \mathcal{X}_U) = \mathcal{L}_L(\mathcal{X}_L) + \sum_{l=1}^k \sum_{i_l=1}^{M_{l_l}} \gamma_{i_l} \mathcal{L}_l(\mathbf{x}_{i_l}, y_{i_l}; y'_{i_l}) \quad (3.1)$$

Burada M_{l_l} öğrenci-öğretmen modeli tarafından bulunan nesne sayısı, y_{i_l} ve y'_{i_l} sırasıyla birinci iterasyondaki gerçek ve sözde etiketlerdir. Burada, γ_{i_l} nesnenin görüntü içinde bulunup bulunmadığını belirler; nesne görüntü içinde varsa γ_{i_l} değeri 1, yoksa 0 değerini alır. Eğer $\mathbf{x}_{i_l}$ nesnesi için güven puanı önceden tanımlanmış bir seviyeden büyük ise γ_{i_l} 1 değerini alır. Önerilen öğrenci-öğretmen modelinin detayları Algoritma 1'de özetlenmiştir. Deneylerde n_{max} değeri 3 olarak belirlenmiştir.



Şekil 3.2. Öğrenci-öğretmen modeli kullanılarak sözde etiketleme yöntemi. İlk olarak manuel olarak etiketlenmiş görüntüler eğitim için kullanılır, daha sonra sözde etiketlenmiş görüntüler bir sonraki eğitim adımında eğitim veri kümesine eklenir.

Kaynak: (Ozdemir & Koc, 2024)

Algoritma 1: Önerilen öğrenci-öğretmen modelin algoritması

Başlangıç:

$\mathcal{X}_L$ eğitim veri kümesi, $\mathcal{X}_U$ etiketlenmemiş veri ve n_{max} döngü sayısı için:

1. $\mathcal{X}_L$ etiketlenmiş veri kümesini kullanarak YOLOv4 modelini eğit.
 2. **for** $n = 1$ to n_{max} **do**
 3. $\mathcal{X}_U$ 'dan M_l tane etiketlenmemiş veri içeren $\mathcal{X}_{M_l}$ kümesini seç,
 4. $\mathcal{X}_{M_l}$ 'deki kusurları tespit etmek için YOLOv4'ü çalıştır,
 5. $\mathcal{X}_L$ ve $\mathcal{X}_{M_l}$ birleştirilerek yeni eğitim veri kümesini oluştur.
 6. **end for**
 7. **return** Genişletilmiş Eğitim Veri Kümesi.
-

Eğitim veri setinin sözde etiketli verilerle genişletilmesi hem genel olarak hem de nesne sınıfı bazında tüm performans ölçüm kriterlerinde iyileştirmelere yol açabilir. Özellikle, eğitim veri setinin ek sözde etiketli görüntülerle genişletilmesi, nesne algılama performansını belirli bir seviyeye getirir.

Tablo 3.2. Öğrenci-Öğretmen modelin hiper parametreleri

Giriş Boyutu	Yığın	Mini yığın	Öğrenme Katsayısı	Momentum	Optimizasyoncu	Tekrar
352×352	64	2	0,001	0,949	Adam	12000

3.3. Deneyler

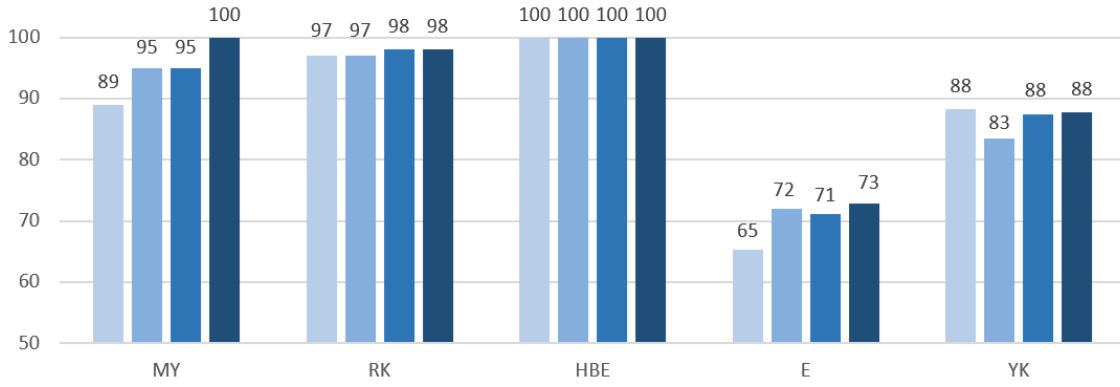
Önerilen öğrenci-öğretmen modelinin performansı, YOLOv4 algoritması kullanılarak demiryolu kusur veri kümesi üzerinde test edilmiştir. Deneyler, 2,50 GHz hızında çalışan Intel i7- 11700 CPU, RTX3060 Ti GPU ve 16 GB RAM'a sahip bir masaüstü bilgisayarda gerçekleştirilmiştir. Hiperparametreler, çeşitli deneyler sonucunda belirlenmiş ve elde edilen değerler Tablo 3.2'de sunulmuştur.

Sözde etiketli görüntüler kullanılarak genişletilmiş veri kümesi üzerinde eğitim yapılarak yüksek performanslı bir nesne algılama modeli geliştirilmiştir.

Nesne algılama modelinin gelişimi, veri kümesindeki sözde etiketli görüntülerin sayısı kademeli olarak artırılarak incelenmiştir. Önerilen öğrenci-öğretmen modelinin performansını test etmek için iki deney gerçekleştirilmiştir. İlk deneyde başlangıçta, toplam 240 nesne içeren 208 manuel etiketli görüntü eğitim için kullanılmıştır. Bir sonraki adımda, 60 görüntü önerilen öğretmen modeli kullanılarak etiketlenmiştir. Eğitim seti, sözde etiketli görüntülerin ilk eğitim setine eklenmesiyle genişletilmiştir. Daha sonra eğitim veri kümesinin genişletilmesi ile 307 nesne ile 268 görüntü sayısına ulaşılmıştır. Deneylerde güven puanı 0,9 olarak ayarlanmıştır. Bir sonraki adımda, genişletilmiş veri kümesi eğitim için kullanılmış ve 60 ek görüntü etiketlenmiştir. Bu işlem 3 kez tekrarlanmıştır. Nesne algılama performansının değerlendirilmesi, her deneyde test veri kümesi üzerinde mAP, F₁-skoru, birleşme üzerinde kesişim (IoU), geri çağırma ve ortalama kesinlik puanlarının hesaplanmasıyla gerçekleştirilmiştir. Deneylerin sonuçları Tablo 3.3'te özetlenmiştir. Tablo, sözde etiketli görüntülerin eğitim setine dahil edilmesiyle veri seti genişledikçe, görüntülerin 208'den 388'e ve nesnelerin 240'tan 427'ye çıkmasıyla tüm değerlendirme metriklerinin iyileştiğini göstermektedir. mAP değeri % 87,94'ten % 91,74'e yükselmiş olup, bu durum hataların daha kesin bir şekilde tanındığını göstermektedir. Aynı zamanda, ortalama kesinlik ve geri çağırmaı dengeleyen F₁-skoru 0,83'ten 0,88'e yükselerek sistemin genel tespit doğruluğunun arttığını göstermektedir.

Tablo 3.3. Sözde etiketlenmiş görüntülerin başarıma etkileri

Görüntü sayısı	Nesne Sayısı	mAP	F ₁ -skor	IoU	Geri çağırma	Ortalama kesinlik
208	240	87,94	0,83	66,08	0,84	0,81
268	307	89,52	0,84	68,33	0,85	0,84
328	367	90,40	0,87	71,55	0,87	0,88
388	427	91,74	0,88	70,76	0,88	0,88



Şekil 3.3. Sözde etiketli görüntülerin eklenmesinin etkisi AP değerleri üzerinde etkisi

Ortalama kesinliğin 0,81'den 0,88'e yükselmesi, sisteme daha fazla sözde etiketli veri beslendikçe gerçek kusurları tanımada daha tutarlı hale geldiğini ve böylece yanlış pozitiflerin sayısının azaldığını göstermektedir. Geri çağırma değeri de 0,84'ten 0,88'e yükselmiştir; bu da önerilen modelin tüm ilgili arıza oluşumlarını tanımada giderek daha etkili hale geldiğini göstermektedir. Sözde etiketli görüntülerin dahil edilmesi ile IoU'yu 66,08'den 71,55'e yükselterek önerilen yarı denetimli tekniğin etkinliğini göstermektedir. Sözde etiketli görüntüler eklendikten sonra genişletilmiş veri kümesindeki sınıfların nihai dağılımı Tablo 3.4'te verilmiştir.

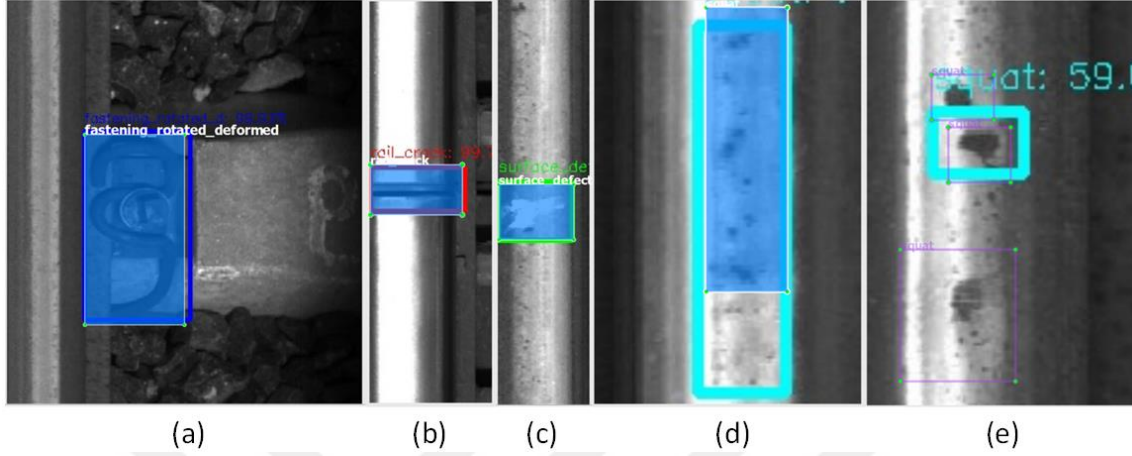
Tablo 3.4. Sözde etiketli görüntüler eklendikten sonra genişletilmiş veri kümesindeki sınıfların nihai dağılımı

	Mantar yırtılmaları	Ray kırığı	Hasarlı bağlantı elemanı	Ezilme	Yüzey kusuru
Örnek sayısı	72	70	70	144	71

Modellerin AP performansları Şekil 3.3'teki grafikte verilmiştir. AP değerleri incelendiğinde mantar yırtılmaları, ezilme ve ray kırığı sınıflarındaki iyileşmeler sırasıyla %12,3, %12,3 ve %1'dir. Hasarlı bağlantı elemanı sınıfının AP değeri ilk deney için %100'dür ve bu değerde herhangi bir azalma görülmemektedir. Yüzey kusuru sınıfının AP değeri ikinci deneyde azalsa da son iki deneyde %88 artmıştır.

Deneyler ve AP değerleri, eğitim veri kümesinin öğretmen modeli tarafından üretilen sözde etiketli görüntülerle genişletilmesinin nesne algılama modelinin performansını artırabileceğini göstermektedir. Şekil 3.3'teki grafiğin incelenmesi, modellerin sınıf tabanlı tespitinde birinci eğitimden dördüncü eğitime kadar belirgin bir iyileşme olduğunu ortaya koymaktadır. Burada açık renkten koyu tona doğru giden sütunlar 1., 2., 3. ve 4. eğitimleri simgelemektedir. Sınıflara ait kısaltmalar ise şu şekildedir; MY (mantar yırtılmaları, RK (ray kırılmaları), HBE (hasarlı bağlantı elemanı), E (ezilme) ve YK (yüzey kusuru). Mantar yırtılmaları, ray kırığı ve ezilme kusur sınıflarının tespit performanslarının iyileştirildiği,

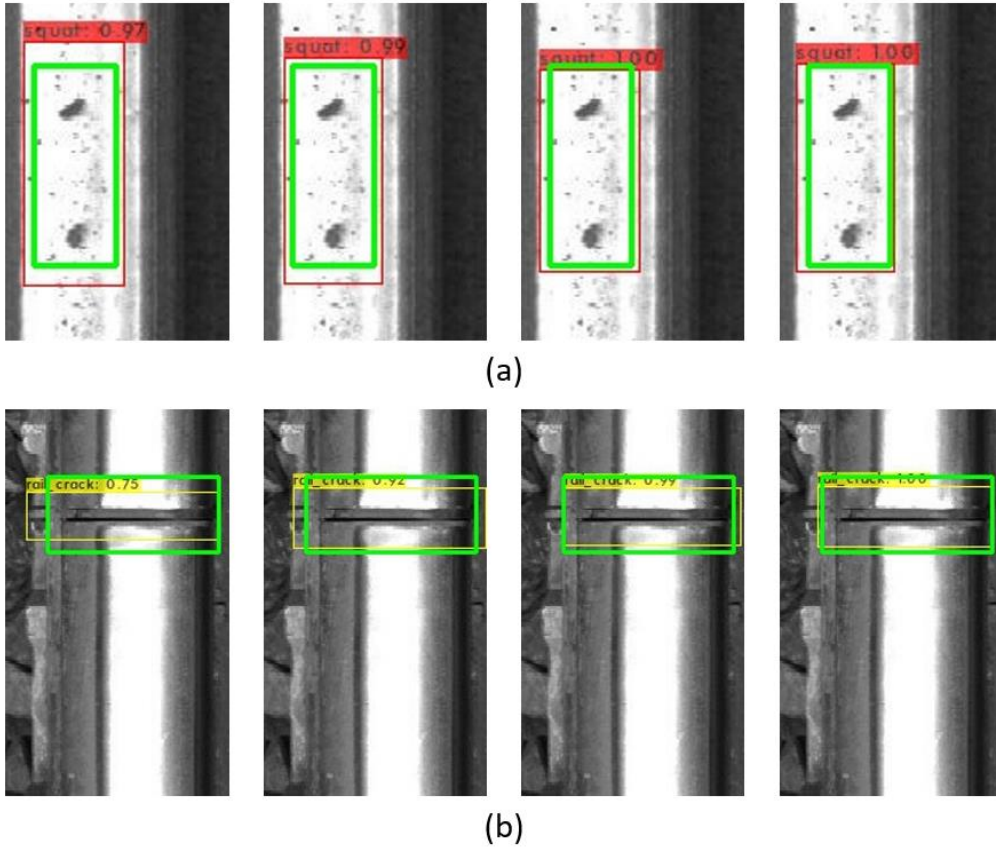
hasarlı bağlantı elemanı ve yüzey kusuru sınıflarının ise aynı kaldığı görülmektedir. Şekil 3.4'te kusur tespitine ilişkin örnekler görselleştirilmiştir. a, b ve c'deki görüntülerde kusurlar doğru bir şekilde tespit edilmiş ve tanımlanmıştır, ancak d ve e'deki görüntülerde kusurlar diğer örnekler kadar doğru tespit edilememiştir.



Şekil 3.4. Temel gerçek ve öğrenci-öğretmen model etiketleme karşılaştırma örnekleri

Kaynak: (Ozdemir & Koc, 2024)

Öğretmen modelinin tespit yeteneklerindeki gelişmeleri daha iyi görmek için, tespit süreci her eğitim döngüsünden sonra test veri kümesindeki 'ezilme' ve 'ray kırığı' sınıflarından örnek görüntülerle tekrarlanmıştır. Gerçek değer bölgeleri bu örnekler üzerinde yeşil bir çerçeveye ile gösterilmiştir. Şekil 3.5 incelendiğinde, öğretmen modelinde kademeli iyileşmeler olduğu görülmektedir. Sözde etiketlemenin etkisiyle, öğretmen modelinin mAP değeri %91,74'e ve F_1 -skoru 0,88'e yükselmiştir.



Şekil 3.5. (a) Ezilme ve (b) ray kırığı sınıfları için etiketleme sürecinin iyileştirilmesi

Kaynak: (Ozdemir & Koc, 2024)

Bu önemli dönüm noktasının ardından öğretmen modeli eğitim sürecinin ikinci aşamasına geçilmiştir. Bu aşamada, eğitim veri setindeki görüntü sayısı ardışık eğitim çalışmalarıyla önemli ölçüde artırılmış; sırasıyla 593, 776, 1164 ve son olarak 1.552'ye çıkarılmıştır. Eğitim veri setindeki bu önemli artışa paralel olarak öğretmen modelinin performansı da iyileşmiştir. Örneğin, mAP değeri % 94,15'e, F₁-skoru ise 0,91'e yükselmiştir. Benzer şekilde, IoU değeri de % 73'ü aşmıştır. Sözde etiketleme modelinin diğer hesaplama metriklerine ait sonuçlar Tablo 3.5'te sunulmuştur. Veri kümesi eğitiminin tamamlanmasının ardından, her sınıf için örnek sayısı Tablo 3.6'da ayrıntılı olarak verilmiştir.

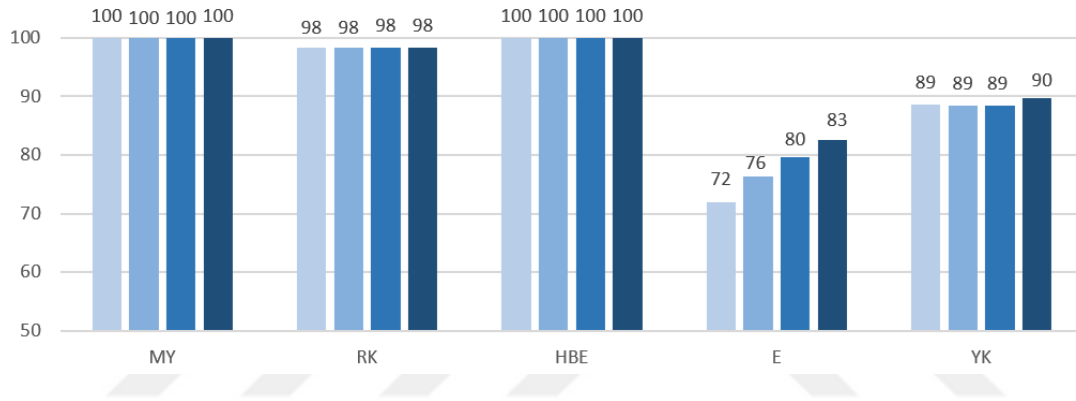
Tablo 3.5. İkinci deneyde sözde etiketli görüntülerin etkisi

Görüntü sayısı	Nesne sayısı	mAP	F1-skor	IoU	Geri çağırma	Ortalama kesinlik
593	660	91,86	0,88	72,67	0,89	0,88
776	854	92,64	0,89	73,21	0,89	0,89
1164	1281	93,31	0,89	73,81	0,90	0,89
1552	1708	94,15	0,91	73,39	0,91	0,90

Tablo 3.6. İkinci deneyde sözde etiketli görüntülerle genişletildikten sonra eğitim setinin nihai örnek dağılımı

	Mantar yırtılmaları	Ray kırığı	Hasarlı bağlantı elemanı	Ezilme	Yüzey kusuru
Örnek sayısı	288	280	280	576	284

Bu deneyler modelin genel performans ölçümleri, sınıf bazlı performansta iyileşmeler olduğunu ortaya koymuştur. Özellikle, mantar yırtılmaları, ray kırığı ve hasarlı bağlantı elemanı sınıfları en yüksek AP değerlerine ulaşmıştır. Ezilme sınıfında %10'luk kayda değer bir iyileşme yaşanırken, yüzey kusuru sınıfı %2'lik bir artış göstermiştir. Süreç boyunca sınıf bazlı AP değerlerinin ilerleyişini gösteren grafik Şekil 3.6'de gösterilmektedir.



Şekil 3.6. İkinci deneyde sözde etiketli görüntülerin eklenmesinin AP değerleri üzerindeki etkisi

4. AKTİVASYON FONKSİYONU ADAPTASYONU İLE DEMİRYOLU KUSUR TESPİT ALGORİTMASININ PERFORMANS İYİLEŞTİRMESİ

Ray kusurlarının tespitinin manuel kontrolü yavaş ve maliyetlidir. Derin öğrenme yöntemleri bu kusurların bazılarını belirli bir dereceye kadar tespit edebilmektedir. Bunun yanı sıra bilgisayarlı görü temelli kusur algılama sistemlerinde, nesne tespit modeli eğitiminde kullanılan veri kümesindeki görüntüler ile sistemin çalışması esnasında gerçek dünyadan alınan görüntüler arasında çevresel faktörler nedeniyle farklılıklar oluşabilmektedir. Bu durumdan kaynaklı olarak algoritma, çok fazla yanlış pozitif üretebilmekte, bu da iş gücü ve maliyet kayıplarına neden olmaktadır (TCDD, 2017). Demiryolu sistemlerindeki en önemli bileşenlerden biri bağlantı elemanlarıdır ve bunların arızalanması ciddi kazalara yol açabilir. Bu bölümdeki çalışmada, yabancı cisimlere ve çevresel koşullara karşı dayanıklı bir nesne tespit algoritması eğitilerek, bağlantı elemanlarının kusur tespiti gerçekleştirilmiştir. Farklı aktivasyon fonksiyonları seçimi ve veri artırma teknikleriyle eğitim veri kümesinin genişletilmesi sonucunda, kusur tespit algoritmasının yanlış pozitif tespitleri büyük ölçüde azalmıştır. Elde edilen algoritma ile test veri kümesi üzerinde yapılan testlerde, en iyi performansta 0,99 F₁-skoru ve %100 mAP değerine ulaşılmıştır.

4.1. Giriş

Demiryolu taşımacılığı, ekonomik ve güvenli olması nedeniyle hem yolcu hem de yük taşımacılığında kritik bir role sahiptir. Yüksek hızlı trenlerin yolcu taşımacılığındaki önemi, sağladıkları konfor ve hız nedeniyle her geçen gün artmaktadır. Demiryolu taşımacılığının artan kullanımı, demiryolu arızalarının tespiti ve bakımını da daha kritik hale getirmektedir. Geçmişte manuel olarak yapılan arıza tespit ve kontrol işlemleri hem maliyetli hem de yavaş olmaktadır. Teknolojinin gelişmesiyle birlikte, daha önce manuel olarak yapılan bu işlemler artık bilgisayarlı görü ve yapay zekâ tabanlı algoritmalar kullanılarak çok daha hızlı bir şekilde gerçekleştirilebilmektedir. Ancak kullanılan arıza tespit modelleri henüz istenilen seviyeye ulaşmış değildir. Tek Atışta Nesne Tespiti (SSD), nesne tespiti ve sınıflandırmasını tek bir sinir ağı mimarisi içerisinde eş zamanlı olarak gerçekleştiren ve gerçek zamanlı uygulamalarda etkili olan nesne tespit yöntemlerinin genel adıdır. SSD'nin başlıca avantajları basitliği, hızı ve farklı boyutlardaki nesnelerin tespitine uygunluğudur. Bu bölümdeki çalışmada demiryolu güvenliği için kritik olan bağlantı elemanlarındaki kusurları tespit etmek için nesne tespit algoritması olarak YOLOv4 modeli kullanılmıştır. YOLOv4 modelinin performansı farklı aktivasyon fonksiyonları kullanılarak analiz edilmiştir. Aktivasyon fonksiyonları, modele ekledikleri doğrusal olmama özelliği nedeniyle modelin performansını

doğrudan etkilemektedir (Szandała, 2021). Bu özellik onları basit doğrusal modellerden ayırmakta ve zorlu gerçek dünya problemleri için başarılı sonuçlar vermelerini sağlamaktadır.

2019 yılında Lin ve arkadaşları, demiryolu bağlantı elemanlarının demiryolunun en önemli bileşenlerinden biri olduğunu ve hasar kontrollerinin manuel olarak yapıldığını belirttikleri bir makale yayınladılar. YOLOv3 modelini 20 km GoPro görüntüsü ile eğittiler ve sırasıyla %89 ve %95 ortalama kesinlik ve geri çağırma elde ettiler (Y.-W. Lin vd., 2019). 2020 yılında Qi ve arkadaşları makalelerinde demiryolu bağlantı elemanlarındaki kusurların gerçek zamanlı olarak tespit edilmesinin bu alanda önemli bir gelişme olacağını belirtmişlerdir. Ayrıca, gömülü bakım sistemlerinin bellek ve işlemci sınırlamaları nedeniyle bu görevin zorlu olduğunu belirtmişlerdir. Geliştirdikleri MYOLOv3-Tiny modeli ile YOLOv3-Tiny modeline kıyasla doğruluk değeri %99,33'e ulaşmış ve bellek tüketimi %43 oranında azaltılmıştır (Qi vd., 2020). Güçlü ve diğerleri çalışmalarında bulanık mantık ve YOLOv4 kullanan yeni bir yöntem sunmuşlardır. Test veri kümesi üzerinde hatalı bağlantı elemanlarını sınıflandırmada %99,25'lik bir başarı oranı elde etmişlerdir (Güçlü, Aydın, vd., 2021). 2021 yılında Liao ve arkadaşları, aktivasyon fonksiyonlarının öğrenme süreci üzerindeki etkilerini kıyaslamalarla araştırmış ve geliştirdikleri hibrit model ile yüzey kusurlarını başarılı bir şekilde tespit ederek çalışmalarında %98,64 mAP değerine ulaşmışlardır (Liao vd., 2021). Şener ve arkadaşları, demiryolu hatlarındaki sorunları %92,2 genel doğrulukla tespit etmek için Tensorflow kütüphanesini ve derin öğrenme yaklaşımlarını kullanan yapay zeka tabanlı bir model önermektedir (Şener vd., 2022). Özdemir ve Koç, demiryollarında karşılaşılan çeşitli kusurları tespit etmek için bir öğrenci-öğretmen modeli ve YOLOv4 ile yarı denetimli bir derin öğrenme stratejisi kullandılar (Ozdemir & Koc, 2024). Türkiye Cumhuriyeti Devlet Demiryolları'ndan (TCDD) bağlantı elemanı hataları da dahil olmak üzere beş farklı kusur içeren yeni bir veri kümesi kullandılar. Yöntemleri, eğitim veri kümesine uygun şekilde sözde etiketlenmiş görüntüler ekleme işlemini otomatikleştirerek model performansını artırmakta ve yoğun emek gerektiren manuel etiketleme görevini azaltmaktadır. Sevi ve arkadaşları, sağlıklı ray bağlantı elemanlarının görüntülerinden kusurlu veriler üreterek demiryolu bağlantı elemanlarındaki kusurları sınıflandırmak için derin öğrenme tabanlı bir yaklaşım önermiştir. Bağlantı elemanı kusurlarını sınıflandırmak için CNN, VGG16 ve ResNet50 modellerini kullanmışlar ve önerilen yöntemle %100 doğruluk oranına ulaşmışlardır (Sevi vd., 2022). He ve arkadaşları bir sıkma ve uyarma (SE) dikkat mekanizması eklemiş, omurgayı hafif bir MobileNet-V2 ağı ile değiştirmiş ve ray bağlantı elemanı hizmeti durum tespiti için Yolov4 modelini geliştirmek üzere görüntü çiftlerini üst

üste bindirme yöntemi (Mixup) ile veri artırımını dahil etmiştir (J. He vd., 2024). Geleneksel Yolov4 modeliyle karşılaştırıldığında, elde ettikleri sonuçlar işlem hızında %67,39'luk bir artış ve tespit doğruluğunda %83,2'lik bir MAP artışı olduğunu göstermiştir.

Yılmaz ve arkadaşları, eksik bağlantı elemanları da dahil olmak üzere üç farklı hata sınıfını ve sağlıklı traverslerin durumunu tespit etmeyi amaçlamış ve yapılan testlerde modelin doğruluğu %95 olarak belirlenmiştir (Yılmaz vd., 2022). Zengzhen Mi ve arkadaşları, sundukları geliştirilmiş YOLOv4 modelinin, karşılaştırdıkları YOLOv3, YOLOv4, YOLOv5, YOLOv6, Faster RCNN ve SSD modellerinden daha iyi sonuçlar verdiğini tespit etmişlerdir. F_1 skoru 0,914 ile en iyi dereceye sahip olan YOLOv6 modelinin performansını 0,925 skorla geçmişlerdir (Mi vd., 2023).

TCDD veri kümesi oluşturulurken, VCUBE tarafından yapılan tespitlerde çok sayıda yanlış pozitif ortaya çıkmıştır. Ortalama kesinlik değeri yaklaşık %11'dir; bu da 2.730 tespitten sadece 301'inin doğru olduğu anlamına gelmektedir. Yanlış pozitifler, bağlantı elemanları üzerindeki balastlar ve diğer çevresel koşullar nedeniyle ortaya çıkabilir ve bu da çok düşük performansa neden olmaktadır. Bağlantı elemanı kusurlarının tespitine odaklanan bu çalışmada, aktivasyon fonksiyonu seçimi ve veri artırmanın modelin performansı üzerindeki etkileri analiz edilmiştir.

4.2. Malzeme ve Yöntemler

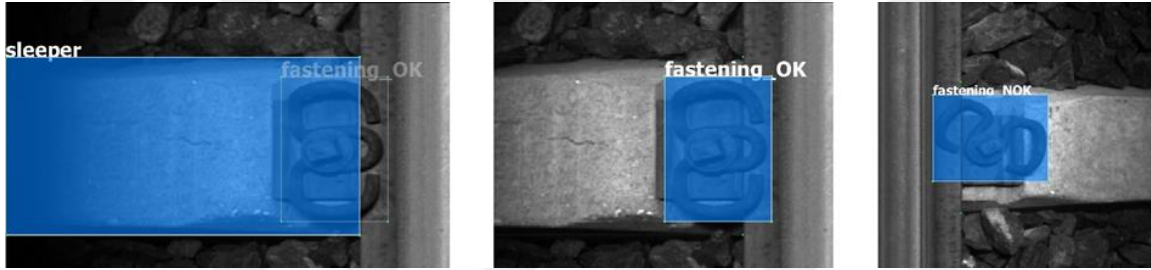
4.2.1. Veri Kümesi

Derin öğrenme modellerinin başarısındaki en kritik faktörlerden biri veri kümesidir (Bochkovskiy vd., 2020). Nesne tespit algoritmasını eğitmek için kullanılan veri kümesindeki sınırlayıcı kutular ve etiketlerin doğru ve tutarlı bir şekilde oluşturulması modelin başarımı açısından çok önemlidir. Ayrıca aynı sınıf içerisindeki görüntü çeşitliliği ne kadar fazla olursa, modelin yeni görüntüler üzerinde başarılı tahmin yapma ihtimali de o kadar yüksek olacaktır. TCDD'nin DATEM biriminden sağlanan demiryolu görüntüleri kullanılarak oluşturulan veri kümesinden, demiryolu bağlantı elemanı kusur tespitinde kullanılabilecek olan sınıflara ait görüntüler alınarak, bu bölümdeki çalışmaya özel bir veri kümesi derlenmiştir. Derlenen bu veri kümesindeki görüntüler analiz edilerek işaretleme ve etiketlemeler kontrol edilmiştir. Tüm görüntüler incelendikten sonra aşağıdaki üç sınıfın kullanılmasına karar verilmiştir:

Travers: Rayların altına yerleştirilir ve rayları desteklemek için kullanılır. Yüğü rayların altına dağıtarak rayları stabilize eder ve rayların zeminle temasını sağlar.

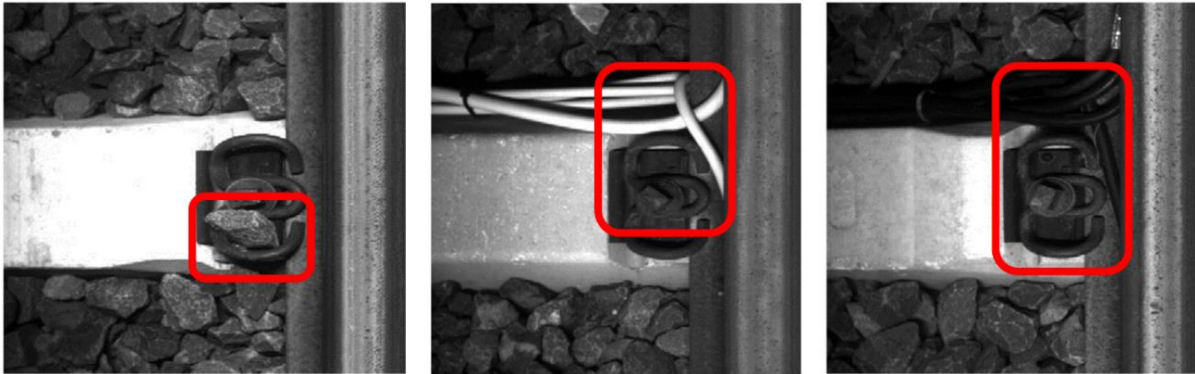
Bağlantı elemanı OK: Bağlantı elemanları rayları ve ray bileşenlerini bağlamak için kullanılır. Bu parçalar rayları yataklara ve alt yapıya bağlar ve rayın güvenli ve dengeli olmasını sağlar. Rayların yerinde tutulmasını ve sabitlenmesini sağlamak için çeşitli bağlantı elemanları kullanılabilir. Bu sistemler titreşimleri emerek ve rayların yüklerini taşıyarak rayın ömrünü uzatır. Bağlantı elemanı sistemleri tipik olarak ray klipsleri, vidalar, kamalar ve diğer donanımlardan oluşur.

Bağlantı elemanı NOK: Bağlantı elemanı herhangi bir nedenle deforme olursa veya dönerek ray ile temasını kaybederse, ray hattının güvenliği tehlikeye girer. Bu durumdaki bağlantı elemanlarının görüntüleri Bağlantı elemanı NOK olarak etiketlenir. Travers, Bağlantı elemanı OK ve Bağlantı elemanı NOK sınıflarının etiketlenmiş örnek görüntüleri Şekil 4.1’de gösterilmektedir.

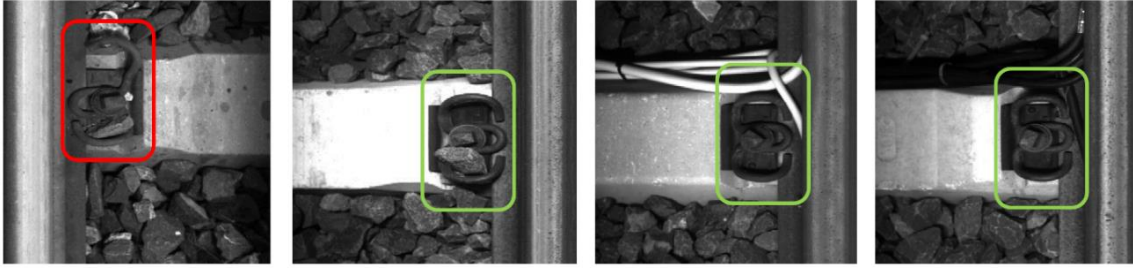


Şekil 4.1. Travers, Bağlantı elemanı OK ve Bağlantı elemanı NOK Sınıflarına Örnekler

DATEM'den alınan demiryolu görüntülerinin, incelenmesinin ardından, sınıflandırmaların yaklaşık %89’unun hatalı olduğu tespit edilmiştir. Örneğin, Şekil 4.2 veri kümesinde başlangıçta hatalı olarak sınıflandırılan ancak aslında doğru olan görüntüleri göstermektedir. Tüm görüntüler analiz edilerek, gerçekten kusurlu olanlar belirlenmiş ve aşağıda gösterildiği gibi kusurlu olmayanlardan ayrılmıştır. Bu duruma Şekil 4.3’te örnek verilmiştir.



Şekil 4.2. Kusurlu olarak etiketlenen normal görüntüler

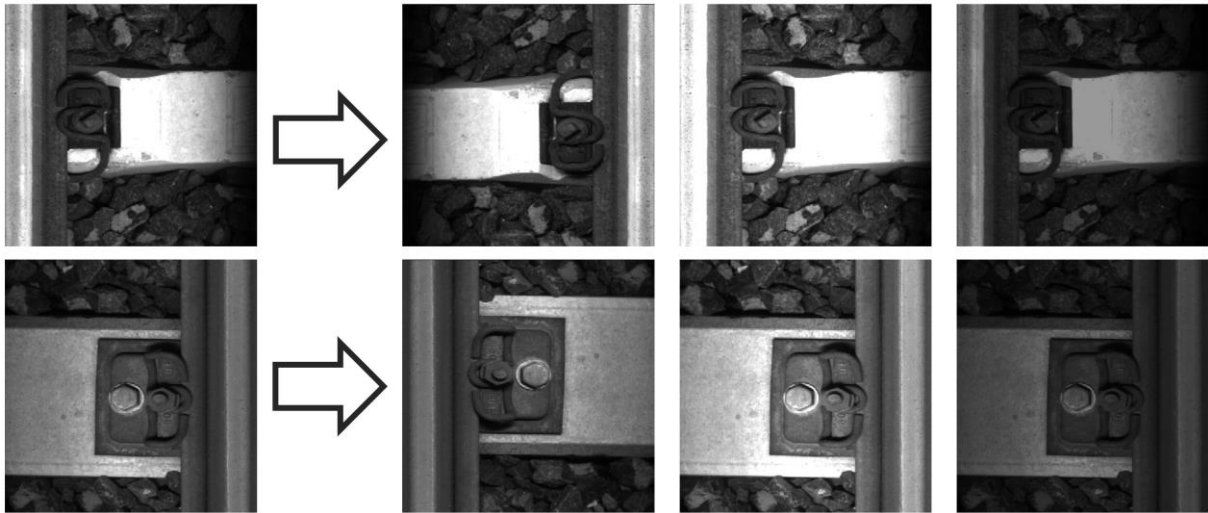


Şekil 4.3. Kusurlu olarak etiketlenen görüntülerin kontrol edilmesi

Daha sonra elde edilen görüntülerden eğitim, doğrulama ve test veri kümeleri oluşturulmuştur. Bağlantı elemanlarının etiketlenmesine ek olarak, görüntülerdeki traversler de etiketlenerek, gerektiğinde eksik bağlantı elemanı kusurlarının tespit edilmesini kolaylaştıran bir veri kümesi oluşturulmuştur. Eğitim, doğrulama ve test veri kümelerindeki görüntü sayıları ve bu görüntülerdeki sınıf sayıları Tablo 4.1.'de verilmiştir.

Tablo 4.1. Eğitim, Doğrulama ve Test Veri Kümelerindeki Görüntü Sayısı

Veri Seti	Görüntü	Travers	Bağlantı elemanı OK	Bağlantı elemanı NOK
Eğitim	812	812	571	241
Doğrulama	100	100	68	32
Test	100	100	72	28



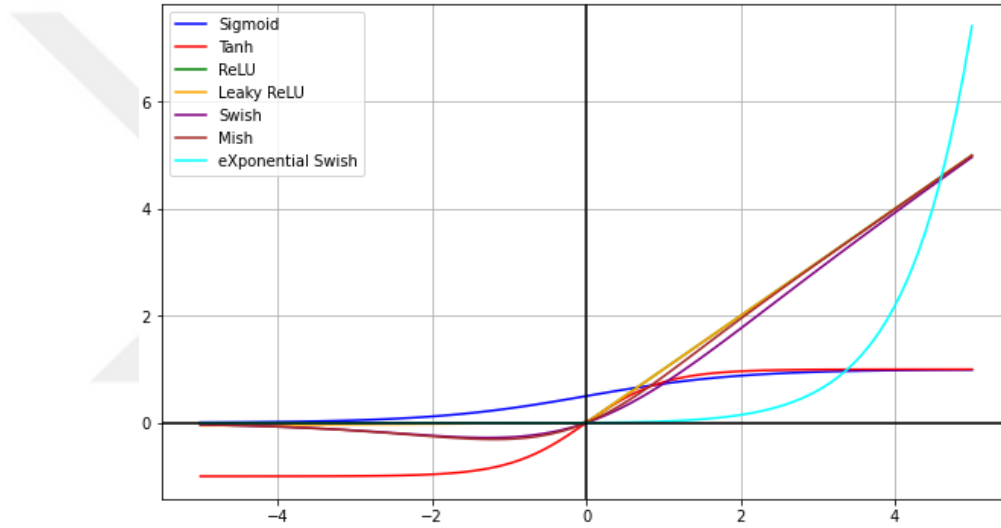
Şekil 4.4. Veri artırma tekniklerinin uygulanması örneği

Veri artırma tekniklerinin kullanımı, özellikle kontrast ve oryantasyon ayarlamaları, görüntü sayısını artırarak eğitim veri setini orijinal boyutunun dört katına çıkarılmıştır. Şekil 4.4 eğitim veri kümesindeki bir görüntüye 180 derece oryantasyon, yüksek ve alçak kontrast uygulanması sonucu elde edilen görüntülerin örnekleri verilmiştir. Artırılmış eğitim veri

kümesinde 3248 Travers, 2284 bağlantı elemanı_OK ve 964 bağlantı elemanı_NOK örneği içeren 3248 görüntü bulunmaktadır.

4.2.2. Aktivasyon Fonksiyonları

YOLOv4 nesne tespit algoritması modelinin öğrenme performansını araştırmak için literatürde yaygın olarak kullanılan bazı aktivasyon fonksiyonları seçilmiş ve performansları test veri kümesi üzerinde karşılaştırılmıştır. Aktivasyon fonksiyonları, modelin eğitimini optimize etmek, gradyan kaybolma/çıkma problemleriyle başa çıkmak ve öğrenme sürecini iyileştirmek için tasarlanmıştır. İdeal aktivasyon fonksiyonu modele ve uygulamaya göre farklılık gösterebilir. Şekil 4.5 derin öğrenme modellerini eğitmek için kullanılan en sık aktivasyon fonksiyonlarından bazılarını göstermektedir.



Şekil 4.5. Aktivasyon fonksiyonlarının karşılaştırılması

Bu çalışmada, model eğitim sürecinde Leaky ReLU (Rectified Linear Unit), Swish, x-Swish (Exponential Swish) ve Mish aktivasyon fonksiyonları kullanılmıştır. Bu aktivasyon fonksiyonlarının her biri aşağıda ayrıntılı olarak açıklanmaktadır:

$$f(x) = \begin{cases} x & x \geq 0 \\ \alpha x & x < 0 \end{cases} \quad (4.1)$$

Swish: Swish fonksiyonu, giriş değerlerini sigmoid fonksiyonu ile çarparak tanımlanır. Matematiksel ifadesi şu şekildedir:

$$f(x) = x \sigma(x) = \frac{x}{1+e^{-x}} \quad (4.2)$$

Burada $\sigma(x)$, sigmoid fonksiyonunu temsil etmektedir. Swish, ReLU ve sigmoid fonksiyonlarının birleşiminden türetilmiş olup, hem lineer olmayan hem de doğrusal bileşenlere sahiptir. ReLU'nun aksine, Swish negatif girişlerde değerleri sıfıra indirmeyip

küçük ama doğrusal olmayan sonuçlar üretmektedir. Bu da Swish'in özellikle negatif girdilerde ReLU'ya göre daha fazla bilgi taşımasına olanak tanır.

x-Swish (eXponential Swish): Derin öğrenme modellerinde kullanılan aktivasyon fonksiyonları, ağı doğrusal olmayan karar sınırlarını öğrenmesi açısından büyük bir öneme sahiptir. ReLU ve Swish gibi fonksiyonlar, son yıllarda popülerlik kazanmış olsa da bu fonksiyonların daha gelişmiş ve uyarlanmış versiyonları da literatürde yer almaktadır. Bu bağlamda x-Swish (eXponential Swish), Swish fonksiyonunun bir türevi olarak geliştirilmiştir ve model performansını artırmak amacıyla önerilmiştir.

$$f(x) = \beta x \sigma(x) = \frac{\beta x}{1 + e^{-x}} \quad (4.3)$$

Bu fonksiyon, giriş verilerine daha dinamik ve yumuşak bir aktivasyon uygular. Swish'in standart yapısına göre daha yüksek doğruluk ve verimlilik elde etmek amacıyla, x-Swish fonksiyonu giriş değerleri üzerinde üstel bir değişiklik yaparak daha esnek bir aktivasyon fonksiyonu sunar.

Mish: Mish, diğer aktivasyon fonksiyonlarından, özellikle Swish ve ReLU'dan daha yumuşak bir eğriye sahiptir.

$$f(x) = x \tanh(\ln(1 + e^x)) \quad (4.4)$$

Burada; $\ln(1 + e^x)$ kısmı softplus fonksiyonunun bir türevidir. Pozitif değerlerde girişin üstel büyümesini kontrol ederken, $\tanh()$, bu üstel büyümeyi yumuşatarak doğrusal olmayan bir geçiş sağlar. Mish, negatif girişler için sıfıra yaklaşan, ancak tamamen sıfır olmayan bir değer üretir. Bu, ReLU'nun negatif değerleri sıfıra dönüştürdüğü durumlardan farklıdır ve negatif değerlerden de bilgi öğrenilmesini mümkün kılar.

Mish*: Mish* aktivasyon fonksiyonu hem Mish aktivasyon fonksiyonu hem de SAM (Spatial Attention Module) modülü ile birleştirilmiştir. Bu, modelin özellik haritasındaki önemli bölgelere odaklanmasına yardımcı olabilir. Bu özel fonksiyonun matematiksel formülasyonu uygulama bağlamına bağlı olarak değişebilir, ancak genel olarak Mish aktivasyon fonksiyonu ve uzamsal dikkat özelliklerini içerir. Mish*, uzamsal dikkat özelliğini Mish fonksiyonu ile birleştirerek nesne tespiti gibi görevlerde daha iyi performans elde etmeyi amaçlamaktadır. Mish aktivasyon fonksiyonundan farklı olarak, eklenti modülleri sadece SPP'ye değil aynı zamanda SAM'e de sahiptir ve baş kısmı lojistik yerine doğrusal aktivasyon fonksiyonunu kullanır.

4.3. Deneyler

Özel demiryolu bağlantı elemanı veri kümesi üzerinde eğitilen YOLOv4 nesne tespit algoritması ile geliştirilen hata algılama modelinin performansını karşılaştırmak için çeşitli aktivasyon fonksiyonları kullanarak 10 farklı deney gerçekleştirilmiştir. Yapılan deneylerde, YOLOv4 modeli, 352×352 piksel girdi boyutu, 64 yığın ve 32 mini yığın ile belirlenen hiperparametreler kullanılarak eğitilmiştir. Öğrenme oranı 0,001 ve momentum 0,949 olarak ayarlanmış, eğitim ise Adam optimizier ile 12.000 iterasyon üzerinden gerçekleştirilmiştir. Deneyler, 2.50GHz hızında Intel i7-11700 CPU, RTX3060 Ti GPU ve 16GB RAM ile donatılmış, Windows 11 işletim sistemli bir masaüstü bilgisayarda gerçekleştirilmiştir.

Tablo 4.2. Aktivasyon fonksiyonu performanslarının karşılaştırılması

Aktivasyon	F₁-skor	Geri çağırma	Ortalama kesinlik	IoU	mAP
Leaky	0,99	0,99	0,98	93,07	98,62
Swish	0,86	0,93	0,80	75,80	99,20
x-swish	0,92	0,95	0,90	87,13	97,97
Mish	0,93	1,00	0,87	81,83	99,24
Mish*	0,98	0,99	0,98	92,57	98,70

Aktivasyon fonksiyonlarının modelin performansı üzerindeki etkilerini araştırmak için Leaky ReLU yerine Swish, x-Swish, Mish ve Mish* aktivasyon fonksiyonları teker teker seçilerek model eğitim veri kümesi üzerinde eğitilmiştir. Bu eğitim sonucunda farklı aktivasyon fonksiyonları kullanılarak elde edilen modeller ile test veri kümesi üzerinde performans karşılaştırmaları yapılmıştır. Bu testlerin sonuçları, genel performans metrikleri üzerinden karşılaştırılmıştır ve Tablo 4.2’de verilmiştir. Ayrıca eğitim sürecinde anlık olarak oluşan ortalama kayıp (Average Loss) değerinin iterasyona göre değişimini gösteren grafikler de ekler bölümünde EK-3’te verilmiştir. Bu grafiklerde aynı zamanda, 100 iterasyonda bir tekrarlanan, doğrulama kümesine ait mAP değerinin sonuçları da görülebilir. Bu grafiklerde ortalama kayıp değerleri mavi, mAP değerleri ise kırmızı noktalar ile gösterilmiştir.

Leaky aktivasyon fonksiyonu seçilen eğitime ait grafik incelendiğinde ilk iterasyonlarda kayıp değeri hızlı bir düşüş gösterdiği ve daha sonra stabil hale gelip 0,2891 değerinde noktalandığı görülmektedir. F₁-skor, geri çağırma ve ortalama kesinlik değeri oldukça yüksek olduğu için modelin tahmin yeteneği de yüksek denilebilir. 93,07 IoU değeri nesnelerin doğru şekilde kapsandığını ve doğru sınıflandırıldığını göstermektedir. Genel olarak baktığımızda ise yüksek doğruluk oranı ve düşük kayıp değerleriyle en iyi performansı sunan aktivasyon fonksiyonlarından biri olarak öne çıktığı görülmektedir.

Swish aktivasyon fonksiyonu seçilen eğitime ait grafik incelendiğinde başlangıçta kayıp değerinin çok hızlı biçimde düştüğü, diğer aktivasyon fonksiyonlarına göre daha az olsa

da x-swish aktivasyon fonksiyonuna göre daha yüksek bir kayıp değerinde stabil hale geldiği görülmektedir. F₁-skor ve geri çağırma değerlerinin performansının ise diğer fonksiyonların gerisinde kaldığı görülmektedir. Ayrıca 0,80 ortalama kesinlik oranı da yanlış pozitif oranının yüksek olduğunu göstermektedir. 75,80 IoU değeriyle de diğer aktivasyon fonksiyonlarının gerisinde kaldığı, doğru tahmin ettiği nesnelerin sınırlayıcı kutularını çizmede diğer aktivasyon fonksiyonları kadar başarılı olmadığı söylenebilir. Genel olarak baktığımızda ise özellikle ortalama kesinlik ve IoU değerleri açısından diğerlerinin gerisinde kaldığı görülmektedir.

X-swish aktivasyon fonksiyonu, model eğitim grafikleri baz alındığında en başarılı sonuç alınmasını sağlayan aktivasyon fonksiyonu olarak dikkat çekmektedir. F₁-skor ve geri çağırma değerlerinin performansı sırasıyla 0,92 ve 0,95 olarak dikkat çekicidir. 0,90 ortalama kesinlik değeri ile swish ve mish aktivasyon fonksiyonlarına göre yanlış pozitif oranını daha iyi kontrol ettiği söylenebilir. IoU değerinde de nispeten başarılı olduğu söylenebilir.

Mish aktivasyon fonksiyonu model eğitim grafiği ele alındığında kayıp değerinin diğer aktivasyon fonksiyonlarına göre çok daha yüksek olduğu ve stabil hale gelmesinin diğer aktivasyon fonksiyonlarına kıyasla çok daha ileriki iterasyonlarda gerçekleştiği görülmektedir. Bu durum birkaç farklı kategoride şu şekilde yorumlanabilir:

- Mish, aktivasyon fonksiyonu olarak genelde daha düzgün ve sürekli bir gradyan sağlar, bu da daha iyi öğrenme performansı sağlar. Ancak, kompleks bir fonksiyon olduğu için, model bazen daha fazla ince ayar yaparak optimum noktayı bulmaya çalışır. Bu durum, eğitim sürecinde dalgalanmalara sebep olabilir.
- Eğer öğrenme oranı (learning rate) çok yüksekse, model kaybı azaltma sırasında daha büyük adımlar atar ve bu da salınımına neden olabilir. Düşük öğrenme oranı ise daha küçük adımlar atmaya yol açarak daha istikrarlı bir düşüş sağlayabilir. Salınımlar bazen öğrenme oranı zamanla küçültülerek azaltılabilir.
- Eğitimde kullanılan batch boyutu ve veri setindeki çeşitlilik de salınımları etkiler. Eğer batch boyutu küçükse, model her iterasyonda farklı örneklerle eğitildiği için kayıp değeri düzensiz bir şekilde değişebilir.
- Kullanılan optimizasyon algoritması da dalgalanmaları etkileyebilir. Bazı optimizasyon algoritmaları, özellikle momentumlu olanlar, kayıp fonksiyonundaki dalgalanmalara karşı hassas olabilir.

0,93 F_1 -skoru ve 1,0 geri çağırma değeri ile dengeli bir performans sağlamıştır. IoU değeri olarak diğerlerine kıyasla orta seviyededir. 99,24 mAP değeri ile de bu kategoride en başarılı aktivasyon fonksiyonu olmuştur. Genel olarak bakıldığında ise en iyi sonuçlardan birine sahip olduğu söylenebilir.

Mish* aktivasyon fonksiyonu modelin eğitim grafiğine bakıldığında düşük kayıp değeri ile hızlı bir stabilizasyon sağladığı söylenebilir. 0,98 F_1 -skoru, 0,99 geri çağırma ve 0,98 ortalama kesinlik değeri ile dengeli ve başarılı bir performans sergilemiştir. IoU ve mAP değerleri de başarılı sonuçlar elde etmiştir. Genel olarak baktığımızda Leaky ile benzer şekilde yüksek doğruluk ve düşük kayıp değeriyle en iyi performansı sunmuştur.

Leaky ve Mish* aktivasyon fonksiyonları, en yüksek F_1 -skor, IoU ve ortalama kesinlik değerlerine ulaşmışlardır. Bu nedenle bu fonksiyonlar, özellikle yüksek doğruluk ve düşük kayıp değeri aranan görevler için ideal olabilirler. Swish fonksiyonu, diğerlerine göre düşük doğruluk ve IoU değeriyle daha zayıf bir performans sergilemiştir.

Nesne tespit modelinin, test veri kümesi üzerinde, kusur sınıfları özelindeki performans sonuçları ise Tablo 4.3'te gösterilmiştir. Test veri kümesi üzerinde Mish aktivasyon fonksiyonu, 28 kusurlu bağlantı elemanından 27'sini tespit ederek %3,57'lik bir iyileşme ve 0,93'lük F_1 -skoru ile en iyi sonucu elde etmiştir. Hatalı bağlantı elemanlarının tespitinde en iyi sonuç elde edilmesine rağmen, 'travers' ve 'bağlantı elemanı_OK' sınıflarındaki yüksek sayıda yanlış pozitif nedeniyle genel performans düşmüştür.

Tablo 4.3. Aktivasyon fonksiyonlarına göre modelin kusur sınıfı tespit performansı

	Travers			Bağlantı elemanı_OK			Bağlantı elemanı_NOK		
AF	AP	TP	FP	AP	TP	FP	AP	TP	FP
Leaky	100,00	100	0	99,67	72	4	96,18	26	0
swish	100,00	89	36	99,12	70	7	98,49	26	2
x-swish	100,00	94	11	99,01	71	6	94,89	26	5
mish	100,00	100	24	100,00	72	6	97,73	27	1
mish*	100,00	100	0	99,67	72	5	96,43	26	0

Hata tespit modelinin performansını artırmak için eğitim veri setindeki görüntü sayısı, çevirme, kontrast azaltma ve kontrast geliştirme gibi veri artırma yöntemleri kullanılarak artırılmıştır. Bu sayede günün farklı saatlerinde, havanın bulutlu veya güneşli olduğu zamanlarda değişen ışık koşullarına daha uyumlu bir eğitim veri seti elde edilmiştir. Bu işlemler sonucunda görüntü sayısı artırılmış ve eğitim veri kümesi orijinal boyutunun dört katına çıkarılmıştır. Genişletilmiş eğitim veri kümesi ile leaky ReLU, Swish, X-Swish, Mish ve Mish* aktivasyon fonksiyonlarına sahip modeller yeniden eğitilmiştir. Eğitim sürecinde anlık olarak oluşan ortalama kayıp değerinin iterasyona göre değişimini gösteren grafikler ise

EK-4'te verilmiştir. Grafikler incelendiğinde ortalama kayıp değerinin en düşük olduğu eğitimler, aktivasyon fonksiyonu olarak, Swish ve x-swish seçildiğinde gerçekleşen eğitimlerdir. Ayrıca bu aktivasyon fonksiyonları kullanıldığında ortalama kayıp değeri hızla düşüp stabil hale gelmiştir. Yeni modellerin performansı test veri kümesi üzerinde karşılaştırılmıştır. Sonuçlar Tablo 4.4'te listelenen genel performans endeksleri ile Tablo 4.5'te ise sınıf bazında karşılaştırılmıştır. Tabloların analizi, Swish ve Mish* aktivasyon fonksiyonlarıyla eğitilen modellerin tüm kusurlu görüntüleri başarıyla tespit ettiğini ortaya koymaktadır. Bununla birlikte, Mish* aktivasyon fonksiyonu ile eğitilen model, F_1 -skoru 0,93 olan Swish aktivasyon fonksiyonlu modele göre 0,99 değerini elde ederek daha iyi performans göstermiştir. Bu üstünlük Swish modelinin daha fazla yanlış pozitif üretmesine ve 'travers' ve 'bağlantı elemanı_OK' sınıflarındaki bazı gerçek pozitifleri tespit edememesine bağlanmaktadır. Sonuç olarak F_1 -skoru, geri çağırma, ortalama kesinlik ve IoU kategorilerinde en yüksek skorları elde eden Mish* aktivasyon fonksiyonunun en başarılı fonksiyon olduğu görülmektedir.

Tablo 4.4. Genişletilmiş veri kümesi ile eğitilen modellerin, aktivasyon fonksiyonu özelinde performanslarının karşılaştırılması

AF	F_1 -skor	Geri çağırma	Ortalama kesinlik	IoU	mAP
Leaky	0,99	0,99	0,98	92,93	100,00
swish	0,93	0,95	0,90	84,48	100,00
x-swish	0,93	0,93	0,93	87,16	99,96
Mish	0,97	0,98	0,96	90,51	99,94
mish*	0,99	1,00	0,99	93,51	100,00

Tablo 4.5. Aktivasyon fonksiyonlarına göre modelin kusurlu sınıf özelinde performansı

AF	Travers			Bağlantı elemanı_OK			Bağlantı elemanı_NOK		
	ap	TP	FP	ap	TP	FP	ap	TP	FP
Leaky	100,00	100	0	100,00	72	4	100,00	26	0
swish	100,00	94	14	100,00	68	5	100,00	28	1
x-swish	100,00	87	13	100,00	72	2	99,88	27	0
Mish	100,00	96	5	99,94	72	4	99,88	27	0
mish*	100,00	100	0	100,00	72	3	100,00	28	0

5. DİNAMİK PARAMETRE KULLANIMI İLE RAY ARIZALARINDA SINIF TESPİT DOĞRULUĞUNUN İYİLEŞTİRİLMESİ

Bu bölümde, dinamik alfa parametrelili odaklanmış kayıp fonksiyonu kullanarak raylı sistemlerde sınıf tabanlı arıza tespit modelinin performansının iyileştirilmesi hedeflenmiştir. Arızaların sahip olduğu önem derecesi ve müdahale edilmesi gereken sürelerin aciliyeti gibi farklılıklar, arıza tespit modelinin başarımının sınıf bazlı olarak özelleştirilebilmesi gereksinimini ortaya çıkarmıştır. Bu doğrultuda, kritik öneme sahip arızaların tespitine daha fazla ağırlık verilmiş ve diğer arızaların tespitinden öncelikli tutulması sağlanmıştır. Nesne tespit algoritmalarının istenilen sınıf üzerine hassasiyet gösterebilmesi için, dengesiz veri kümeleri ile yapılan eğitimlerde başarılı sonuçlar almaya yarayan, odaklanmış kayıp fonksiyonu kullanılabilir. Odaklanmış kayıp fonksiyonu çapraz entropi fonksiyonuna uygulandığında istenilen arıza sınıfının tespiti üzerine hassasiyet geliştirildiği ve test veri kümesinde tespit edilen gerçek pozitif sayısının arttığı gözlemlenmiştir. Ayrıca, odaklanmış kayıp fonksiyonundaki alfa parametresinin sabit bir değer yerine dinamik olarak tanımlanması ve bu yaklaşımın nesne tespit algoritmasında kullanılması, modelin performansını daha da iyileştirmiştir.

5.1. Ray Kusur Dereceleri

Bu bölümde ray kusurlarının ait oldukları kusur dereceleri incelenmiştir (TCDD, 2013b). Aşağıdaki Tablo 5.1’de nesne tespit algoritması ile tespit edilebilen kusurlar ve bu kusurların dereceleri belirtilmiştir.

Tablo 5.1. Sınıfların kusur dereceleri

Kusur Sınıfı	Kusur Derecesi
Mantar yırtılmaları	F3
Ray kırığı	F1
Hasarlı bağlantı elemanı	F3
Ezilme	F3/F2
Yüzey kusuru	F3

Demiryolu yüzeyinde ve içerisinde imalat, işletme veya yola serme işlemi sırasında meydana gelen ve işletme güvenliğini etkileyen kusurlara ray kusurları denir. Bu kusurlar önem derecesine göre küçük, büyük ve kritik olmak üzere üç grup altında toplanmıştır. Aşağıda bu grupların açıklamaları verilmiştir.

Küçük Kusurlar (F3); Demiryolunun kullanılmasına engel teşkil etmeyen, bu konuda herhangi bir olumsuzluk yaratmayan veya küçük bir olumsuzluk yaratabilecek olan kusurlardır.

Büyük Kusurlar (F2); Demiryolunun kullanılmasına uygunluk yönüyle bir güvence vermeyen ve demiryolu işletmesini tehlikeye atabilecek kusurlardır. Mümkün olan en kısa zamanda taşlama yapılarak kusurun giderilmesi ya da rayın değiştirilmesi yöntemiyle giderilmesi gereken kusurlardır.

Kritik Kusurlar (F1); Demiryolu işletmesi güvenliği adına büyük bir öneme sahip ve taşımacılık adına ciddi tehlikeler meydana getirebilecek kusurlardır. Kritik kusur olan rayın değişimi yapılarak demiryolu işletmesinin kullanımı güvenli değildir.

5.2. Deneyler

Bu bölümde; mantar yırtılmaları, ray kırığı, hasarlı bağlantı elemanı, ezilme ve yüzey kusurları hata sınıflarından oluşan bir veri kümesi kullanılmıştır. Bu veri kümesi ile eğitilen modelin test veri kümesi üzerindeki başarımı Tablo 5.2'de sunulmuştur. Tablo 5.2 incelendiğinde ortalama kesinlik ve geri çağırma değerlerinin aynı olduğu görülmektedir, bu modelin doğruluk değeri yüksek tahminler yaptığını, ve yanlış pozitif sayısının düşük olduğunu göstermektedir. IoU değeri incelendiğinde 70,76 oranı elde edildiği görülmektedir, bu modelin sadece sınıf tahmin yeteneğinin değil, aynı zamanda lokalizasyon yeteneğinin de iyi olduğunu göstermektedir. 91,74 oranındaki mAP değeri de modelin genel olarak tahmin yeteneğinin iyi olduğu göstermektedir fakat burada ele alınan problem modelin kritik sınıflar üzerindeki başarımı ve kusurların tespiti olduğu için genel ortalamaya değil kusur sınıfları üzerindeki ortalama kesinlik değerlerini incelemek gerekmektedir. Modelin kusur sınıfları özelindeki performansı ise Tablo 5.3'te verilmiştir.

Tablo 5.2. Modelin test veri kümesi üzerindeki performansı

mAP	Ortalama kesinlik	Geri çağırma	F ₁ -skor	IoU
91,74	0,88	0,88	0,88	70,76

Tablo 5.3. Modelin kusur sınıfları özelindeki performansı

Mantar yırtılmaları			Ray kırığı			Hasarlı bağlantı elemanı			Ezilme			Yüzey kusurları		
AP	TP	FP	AP	TP	FP	AP	TP	FP	AP	TP	FP	AP	TP	FP
100	20	0	98,2	23	3	100	20	0	72,8	26	5	87,7	17	6

Tablo 3.1 verildiği üzere test veri kümesinde ‘Ezilme’ sınıfından 37, örnek diğer sınıflardan ise 23 ve 20’şer adet örnek bulunmaktadır.

Tablo 5.3 incelendiğinde ezilme kusuru hariç, tüm kusur sınıflarının ortalama kesinlik değerinin %87,7 ve üzerinde olduğu görülmektedir. Tablo 5.1’deki sınıfların kusur derecelerine bakıldığında önem sırasının ‘ray kırığı’, ‘ezilme’ ve diğerleri şeklinde sıralandığı görülmektedir.

Tablo 5.3’te gözlemlendiği üzere test veri kümesindeki 23 adet ‘ray kırığı’ örneğinin tamamı tespit edilebilmiştir. Sonraki en önemli sınıf olan ‘ezilme’ kusurları kritik arızalara neden olabileceği için tespit edilmesi önemlidir. Bunun için de bu sınıfa ait hataların mümkün olduğunca yüksek oranda tahmin edilmesi gerekmektedir. Yani test veri kümesi ile yapılan değerlendirmede 26 olan TP sayısı arttırılmalıdır.

İstenilen sınıfta daha yüksek başarımlar sağlanması için odaklanmış kayıp fonksiyonu kullanılabilir. Bunun için kayıp fonksiyonun özelleştirilmesi gerekmektedir. YOLO algoritmasının kayıp fonksiyonu bölüm 2’de bahsedildiği gibi üç ana kısımdan oluşmaktadır (Redmon vd., 2016). Bu kısımların her birine ait formüller denklem 5.1’de verilmiştir.

$$\begin{aligned}
& \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{obj}} [(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2] \\
& + \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{obj}} \left[(\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2 \right] \\
& + \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{obj}} (C_i - \hat{C}_i)^2 \\
& + \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{noobj}} (C_i - \hat{C}_i)^2 \\
& + \sum_{i=0}^{S^2} \mathbb{1}_i^{\text{obj}} \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2
\end{aligned} \tag{5.1}$$

(burada $\mathbb{1}_i^{\text{obj}}$ ilgili hücrede nesnenin görünüp görünmediğini belirtir.)

Hata sınıfına özel ağırlıklandırma yapılacağından, yani sınıflandırma üzerinde özelleştirme yapılacağı için fonksiyonun son kısmında değişiklik yapılmıştır (GitHub, 2018). Kayıp çapraz entropi (CE) fonksiyonu ile değil de odaklanmış kayıp fonksiyonu (FL) ile hesaplanmıştır (T.-Y. Lin vd., 2018).

$$CE(p, y) = \begin{cases} -\log(p) & y = 1 \text{ ise} \\ -\log(1-p) & \text{aksi halde} \end{cases} \quad (5.2)$$

$$p_t = \begin{cases} p & y = 1 \text{ ise} \\ 1-p & \text{aksi halde} \end{cases} \quad (5.3)$$

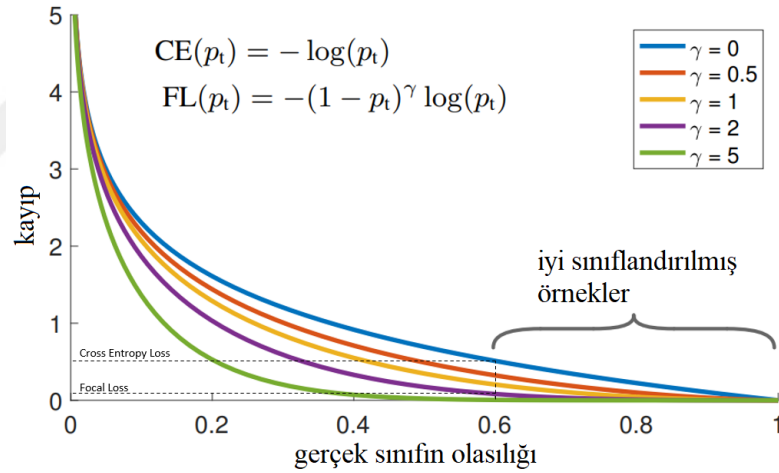
Denklem 5.3, denklem 5.2’de yerine konulduğunda çapraz entropi kaybı aşağıdaki gibi bulunur:

$$CE(p_t) = -\log(p_t) \quad (5.4)$$

Odaklanmış kayıp fonksiyonu ise 5.5’teki denklemde verilmiştir.

$$FL(p_t) = -\alpha_t(1-p_t)^\gamma \log(p_t) \quad (5.5)$$

Şekil 5.1.’teki grafikte odaklanmış kayıp fonksiyonu ve çapraz entropi kaybı karşılaştırması verilmiştir (T.-Y. Lin vd., 2018). FL fonksiyonunda, γ değeri 0’a eşit olduğunda, denklem CE fonksiyonu ile aynı sonucu verir. Bu durum, γ parametresinin eşitlik üzerindeki etkisini açıkça göstermektedir.



Şekil 5.1. Farklı γ değerleri için kayıp ve olasılık değişimi

Kaynak: (T.-Y. Lin vd., 2018)

Denklem 5.5’deki α katsayısına önceliklendirmek istediğimiz sınıflarda daha yüksek bir değer atanarak, ilgili sınıfta yapılan yanlış sınıflandırmalarda uygulanacak hata değeri arttırılmış olur. Böylece model ilgili sınıfa daha duyarlı hale gelir ve diğer sınıflarda başarımda ufak kayıplar olmasına karşın istenilen sınıfta başarımda arttırılmış olur. Şekil 5.1’te Farklı γ değerleri için kayıp ve olasılık değişimi verilmiştir.

5.3. Özelleştirilmiş Kayıp Fonksiyonunun Modele Uygulanması

YOLOv4 algoritmasında özelleştirilmiş kayıp fonksiyonunu uygulayabilmek için öncelikle model yapılandırma dosyasında (cfg) gerekli düzenlemelerin yapılması gerekmektedir (GitHub,

2019b). Yapılan düzenlemelerin ardından, modelin kayıp fonksiyonu, odaklanmış kayıp fonksiyonu ile değiştirilir (GitHub, 2019a). Bu süreçte, kayıp fonksiyonuna ilişkin formül ve hesaplamalar, modelin hata oranlarını azaltmak ve performansı artırmak amacıyla aşağıdaki gibi güncellenmiştir (CSDN, 2017).

$$p_i = \frac{e^{x_i}}{\sum e^k} \quad (5.6)$$

Denklem 5.6 softmax fonksiyonudur p_i ve her bir sınıfa ait olasılığı hesaplar.

FL'nin x_i 'ye göre türevi:

$$\frac{dFL}{dx_i} = \frac{dFL}{dp_i} * \frac{dp_i}{dx_i} \quad (5.7)$$

Yeniden,

$$\begin{aligned} \frac{dFL}{dp_t} &= -\alpha \left(\frac{d(1-p_t)^\gamma}{dp_t} \log(p_t) + (1-p_t)^\gamma \frac{d \log(p_t)}{dp_t} \right) \\ &= -\alpha \left(-\gamma(1-p_t)^{\gamma-1} \log(p_t) + (1-p_t)^\gamma \frac{1}{p_t} \right) \\ &= -\alpha \left(-\gamma(1-p_i)^{\gamma-1} \log(p_i) + (1-p_i)^\gamma \frac{1}{p_i} \right) \end{aligned} \quad (5.8)$$

$y = 1$ olduğundan softmax formülünün x_i tarafından türetilmesi iki durumda ele alınabilir:

if $i == j$:

$$\begin{aligned} \frac{dp_i}{dx_i} &= \frac{e^{x_i} * \sum e^{x_k} - e^{x_i} * e^{x_i}}{\sum e^{x_k^2}} = \frac{e^{x_i}}{\sum e^{x_k}} - \frac{e^{x_i}}{\sum e^{x_k}} * \frac{e^{x_i}}{\sum e^{x_k}} \\ &= p_i - p_i * p_i = p_i(1 - p_i) \end{aligned}$$

if $i \neq j$:

$$\begin{aligned} \frac{dp_i}{dx_i} &= \frac{0 - e^{x_i} * e^{x_j}}{\sum e^{x_k^2}} \\ &= -p_i * p_j \end{aligned} \quad (5.9)$$

Buradan da aşağıdaki eşitliklere ulaşabiliriz:

if $i == j$:

$$\frac{dFL}{dx_i} = \alpha(-\gamma(1-p_i)^{\gamma-1} \log(p_i)p_i + (1-p_i)^\gamma) * (p_i - 1)$$

if $i \neq j$:

$$\frac{dFL}{dx_i} = \alpha(-\gamma(1 - p_i)^{\gamma-1} \log(p_i)p_i + (1 - p_i)^\gamma) * p_j \quad (5.10)$$

5.10' daki denklemde $\alpha = 0,5$ ve $\gamma = 2$ değerleri yerine yazıldığında elde edilen eşitlik YOLOv4 algoritmasının kayıp fonksiyonunun hesaplandığı yolo_layer.c dosyasında aşağıda algoritma 2'de olduğu gibi verilmiştir. İlgili kod satırına kaynakta belirtilen adresten de ulaşılabilir (GitHub, 2021).

Algoritma 2: YOLO algoritmasında odaklanmış kayıp fonksiyonu

```
1. grad = -(1 - pt) * (2 * pt*logf(pt) + pt - 1);
2. delta[index + stride * n] *= alpha * grad;
```

Tablo 5.3'te modelin FL fonksiyonu öncesi sınıf bazındaki başarımları verilmişti. Bölümün başında belirtildiği gibi yapılan düzenleme ile 'ezilme' sınıfındaki TP adetinin artırılması hedeflenmektedir. Bu doğrultuda kullanılacak ağırlıklandırılmış kayıp fonksiyonunun kod üzerindeki örneği Algoritma 3'de verilmiştir. Örnek durumda kat sayısı olarak 0,375 verilmiştir.

Algoritma 3: Dinamik alfa parametresinin odaklanmış kayıp fonksiyonu koduna uygulanması

```
1. if (n == 3)
2.     {
3.         float alpha = 0,375;
4.         delta[index + stride * n] *= alpha * grad;
5.     }
6.     else
7.     {
8.         float alpha = 0,5;
9.         delta[index + stride * n] *= alpha * grad;
10.    }
```

Sınıf numarası 3 olan 'ezilme' kusuruna karşı algoritmada hassasiyet oluşturmak için bu sınıf üzerinde yapılan hatalar α katsayısı ile çarpılmıştır. Yapılan deneysel çalışmalar ile test veri kümesindeki gerçek pozitif adetlerine bakılarak en iyi katsayı bulunmaya çalışılmıştır.

Bunun için öncelikle α katsayısına "1", "1,5", "2", "3", "3,5", "4", "5" ve "6" gibi 1'den büyük değerler verilmiştir. Daha sonra tatmin edici sonuçlar alınamadığı görülüp α katsayısı için 1'den küçük değerler seçilip modelin eğitimi ve akabinde test veri kümesi

üzerindeki performans testlerine devam edilmiştir. Bu aşamada daha başarılı sonuçlar alındığı görülmüştür.

Tablo 5.4. Odaklanmış kayıp fonksiyonu uygulandıktan modelin ezilme sınıfı özelinde başarımının α katsayısı değerine göre değişimi

α	AP	TP	FP	FN
6	67,01	25	8	12
5	69,6	25	9	12
4	74,36	29	9	8
3,5	67,78	23	11	14
3	73,13	26	6	11
2	78,06	28	6	9
1,5	71,33	25	8	12
1	79,02	26	8	11
0,9	83,53	31	6	6
0,8	88,63	29	3	8
0,7	87,09	31	5	6
0,6	77,33	28	7	9
0,5	87,76	31	5	6
0,475	88,04	30	4	7
0,45	86,85	32	5	5
0,425	76,5	30	9	7
0,4	86,52	30	2	7
0,375	91,16	32	3	5
0,35	88,43	31	3	6
0,325	87,5	31	6	6
0,3	78,09	29	8	8
0,25	77,32	30	6	7
0,2	79,69	27	7	10

Tablo 5.4'te kayıp fonksiyonu olarak odaklanmış kayıp fonksiyonu belirlendikten sonra α değerlerine karşılık modelin 'ezilme' kusur sınıfı üzerindeki başarımları verilmiştir. α değerleri 1'den büyük olduğu durumda, veri kümesinde görüntü sayısı ortalamanın altında olan sınıfların öğrenmesine ağırlık verirken, α değerlerinin 1'den küçük olduğu durumlarda, görüntü sayısı, sınıf bazında ortalama görüntü sayısından fazla olan sınıfların öğrenilmesine daha fazla ağırlık verilir. Bu bilgilerin ışığında, veri kümesinde ezilme sınıfının örnek sayısı çoğunlukta olduğu için α değeri 1'in altına indiğinde modelin bu sınıf üzerindeki başarımı artmıştır. Sonuç olarak, başlangıçta ezilme sınıfına ait test kümesindeki 37 örneğin 26'ı tespit edilebilmişken, odaklanmış kayıp fonksiyonu ve α katsayısı optimizasyonu kullanılarak bu sayı 32'ye yükseltilmiştir. AP değeri %26,4 artışla 0,72'den 0,91'a çıkartılmıştır. Farklı α değerlerinin model performansı üzerinde belirgin bir etkisinin olduğu performans sonuçlarından görülmektedir. α değeri, modelin yanlış sınıflandırılmış örnekleri ne kadar

vurguladığını kontrol eden bir parametre işlevi görmektedir. Bu bağlamda, düşük α değerleri doğru sınıflandırmaları ödüllendirirken, yüksek α değerleri yanlış sınıflandırmaların etkisini azaltmaya odaklanmaktadır. α değeri 0.375 seviyesine geldiğinde en yüksek AP (91.16) değeri elde edilmiştir. Bu değerde TP sayısının artması ve FP sayısının azalması ile modelin daha dengeli bir hale geldiği görülmektedir. Bu da α değerinin model başarımını dengeleyici bir etki yarattığını gösterir.

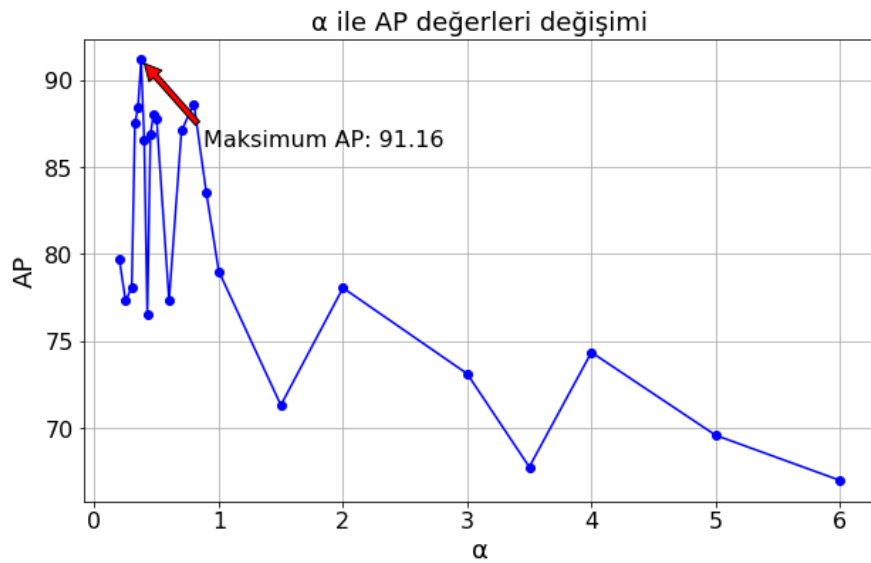
Tablo 5.5. Odaklanmış kayıp fonksiyonu uygulandıktan modelin genel performansının α katsayısı değerine göre değişimi

α	mAP	Ortalama kesinlik	Geri çağırma	F ₁ -skor	IoU
6	87,78	0,66	0,88	0,76	53,52
5	87,58	0,66	0,88	0,75	52,26
4	88,16	0,66	0,91	0,75	51,9
3,5	85,05	0,68	0,85	0,76	54,46
3	89,79	0,75	0,89	0,81	59,52
2	88,07	0,74	0,89	0,81	58,91
1,5	87,75	0,72	0,87	0,79	58,12
1	90,04	0,77	0,88	0,82	61,07
0,9	89,89	0,76	0,93	0,83	58,37
0,8	90,29	0,77	0,90	0,83	60,83
0,7	91,16	0,79	0,93	0,85	62,56
0,6	88,37	0,76	0,88	0,82	59,59
0,5	90,60	0,74	0,93	0,82	58,03
0,475	88,42	0,77	0,86	0,83	60,96
0,45	91,89	0,77	0,93	0,84	60,43
0,425	89,01	0,75	0,92	0,82	58,54
0,4	92,15	0,81	0,92	0,86	63,34
0,375	89,33	0,80	0,92	0,85	62,63
0,35	89,82	0,79	0,91	0,84	61,85
0,325	90,19	0,77	0,92	0,84	60,79
0,3	89,65	0,75	0,91	0,82	60,06
0,25	87,45	0,76	0,90	0,82	58,49
0,2	88,90	0,79	0,88	0,83	61,91

Tablo 5.5'te ise odaklanmış kayıp fonksiyonu uygulandıktan sonra α parametresinin değişimine göre modelin genel performans metrikleri verilmiştir. Yapılan deneyler sonucunda, odaklanmış kayıp fonksiyonundaki α değerinin model performansı üzerindeki etkisi kapsamlı bir şekilde değerlendirilmiştir. Modelin genel değerlendirmede en yüksek performansa, $\alpha=0,4$ değerinde ulaştığı görülmüştür. Bu durumda model, 92,15 mAP, 0,86 F₁-skor, ve 63,34 IoU gibi üstün metrik değerler sunmuştur. Bu sonuçlar, α değerinin doğru seçilmesinin nesne tespiti modellerinde kritik bir öneme sahip olduğunu göstermektedir.

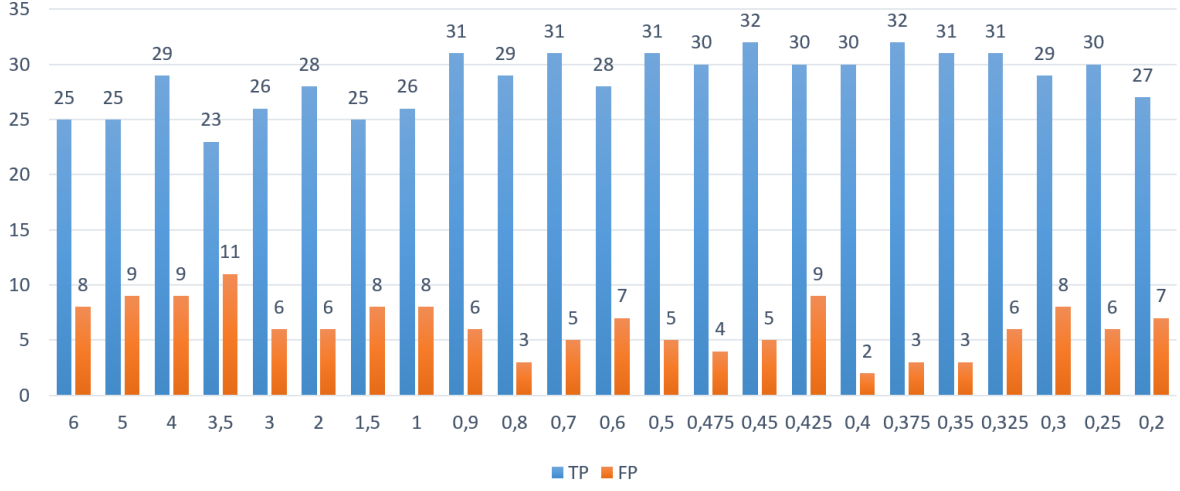
Modelin eğitim süreçlerinde anlık olarak oluşan ortalama kayıp değerinin, iterasyona göre değişimini gösteren grafikler de ekler bölümünde EK-5'te verilmiştir. Ayrıca bu grafiklerde, 100 iterasyonda bir tekrarlanan, doğrulama kümesine ait mAP değerinin sonuçları da kırmızı noktalar şeklinde gösterilmiştir. Bütün eğitimlere ait grafikleri vermek yerine modelin gelişimi göstermek adına $\alpha=1$ değerinde ve en iyi sonuçların elde edildiği $\alpha=0,375$ ve $\alpha=0,4$ değerlerindeki eğitimlere ait grafikler verilmiştir. Grafiklerde gözlemlenen kayıp değerinin hızlı bir düşüş gösterdiği ve düşük seviyelerde stabil hale geldiğidir. Bu durum, modelin odaklanmış kayıp fonksiyonu ile iyi bir öğrenme gerçekleştirdiğini ve eğitim sürecinde doğru öğrenmeyi başardığını göstermektedir. mAP değerleri, α değeri değiştikçe belirli bir seviyeye kadar artış göstermekte, ancak belirli bir seviyeden sonra düşüş trendine girmektedir. Bu durum, optimal bir α değerinin belirli bir aralıkta olduğunu ve bu aralığın dışına çıkıldığında modelin doğruluk oranının azaldığını ifade etmektedir.

Şekil 5.2 ve Şekil 5.3'te verilen grafiklerde α katsayısı ile AP, TP ve FP değerlerinin değişim grafikleri verilmiştir. Şekil 5.2, α katsayısındaki bir artışın, belirli sınıflar için ortalama kesinlik (AP) değerlerinde nasıl bir artışa yol açtığını ve dolayısıyla modelin nadir sınıflar üzerindeki tespit performansını nasıl artırdığını göstermektedir. Şekil 5.3, α arttıkça doğru pozitiflerin (TP) miktarı artarken, yanlış pozitiflerin miktarında da küçük bir artış olduğunu göstermektedir. Her ne kadar α katsayısı sınıf duyarlılığını etkin bir şekilde kontrol etse de bu sonuçlar katsayının TP ve FP arasındaki dengeyi dikkate alarak seçilmesinin önemini işaret etmektedir.



Şekil 5.2 AP değerlerinin α katsayısı ile değişimi

α ile TP ve FP değerleri değişimi



Şekil 5.3 α katsayısı ile TP ve FP sayıları değişimi

Bu bölümde, sınıf tabanlı arıza tespitini iyileştirmek için odak kayıp fonksiyonuna dinamik alfa parametresi eklenmiştir. Deneysel bulgulara göre, dinamik alfa parametreleri model kalitesi ve tespit yeteneği açısından sabit alfa değerlerinden daha iyi performans göstermektedir. Modelin duyarlılığı, özellikle dengesiz veri kümeleri üzerindeki testlerde, önemli arıza sınıfları için çok daha yüksektir. Ancak, dinamik bir alfa değeri kullanıldığında, modelin belirli sınıflara aşırı uyum sağlaması veya aşırı duyarlı hale gelmesi olasılığı vardır. Bu durum, modelin tüm sınıflarda eşit performans göstermesini sağlamak için hiperparametre yapısının ne kadar önemli olduğunu vurgulamaktadır.

6. SONUÇLAR

Bu tez çalışmasında, demiryolu arızalarının tespitine yönelik derin öğrenme tabanlı yöntemlerin etkinliğini artırmak amacıyla yarı denetimli ve aktivasyon fonksiyonu optimizasyonu gibi çeşitli stratejiler uygulanmıştır. Çalışmanın temel hedeflerinden biri, büyük miktarda etiketlenmemiş veriyi daha verimli kullanarak elle etiketleme sürecini azaltmak ve daha yüksek doğruluk oranlarına ulaşmaktır. Yarı denetimli öğrenme yöntemiyle geliştirilen öğrenci-öğretmen modeli, demiryolu kusurlarının tespiti konusunda önemli bir iyileşme sağlamıştır. Yapılan deneylerde mAP skoru %87,94'ten %91,74'e ve ikinci deney serisinde ise %91,86'dan %94,15'e yükselmiştir. Aynı zamanda F1-skorunda da belirgin bir artış görülmüş ve bu artış modelin başarımını ortaya koymuştur. Bu sonuçlar, demiryolu kusur tespitinde öğrenci-öğretmen modelinin etkinliğini ve pratik uygulanabilirliğini göstermektedir.

Yapılan diğer çalışmada ise, TCDD'nin mevcut veri setindeki çevresel etkilerden kaynaklanan yanlış pozitiflerin minimize edilmesi hedeflenmiştir. YOLOv4 modeli, Leaky ReLU, Swish, Mish ve diğer aktivasyon fonksiyonları kullanılarak optimize edilmiştir. Çalışmalar sonucunda geri çağırma, ortalama kesinlik ve mAP metriklerinde %1 ve %1,38 oranında iyileşme sağlanmıştır. Bununla birlikte, özellikle bağlantı elemanları tespitinde veri artırma tekniklerinin uygulanması, başlangıçta gözden kaçan hatalı elemanların başarıyla tespit edilmesine olanak tanımıştır. Bu optimizasyon, mevcut sistemin verimliliğini artırmış ve manuel kontrol ihtiyacını büyük ölçüde azaltmıştır.

Sınıf tabanlı ray sistemi hata tespitinin performansı, dinamik alfa değerlerine sahip bir odak kayıp fonksiyonu kullanılarak artırılmıştır. Kritik arıza sınıflarına atanan ağırlığın artırılmasının, modelin az veriye sahip sınıf tespitine olan duyarlılığını artırdığı gözlemlenmiştir. Yapılan deneyler, dinamik bir alfa parametresinin sabit bir alfa değerine kıyasla daha iyi sonuçlar verdiği, daha yüksek TP tespit oranı ulaştığı ve genel sistem verimliliğinin iyileştirilmesini sağladığı görülmüştür.

Tezde geliştirilen bu yenilikçi yöntemler, demiryolu arıza tespit çalışmalarına önemli katkılar sağlamıştır. İlk olarak, yarı denetimli öğrenme tabanlı model, veri etiketleme sürecindeki iş gücü maliyetini düşürerek derin öğrenme tabanlı sistemlerin daha verimli bir şekilde kullanılmasına olanak tanımaktadır. Bunun yanında, açık erişimli bir veri kümesi oluşturulması, demiryolu alanında çalışan diğer araştırmacılar için kıyaslama yapabilecekleri bir kaynak sağlamaktadır. Son olarak, dinamik alfa parametrelerine dayalı odaklanmış kayıp fonksiyonu ile kritik önem taşıyan arızaların daha doğru bir şekilde tespiti mümkün olmuştur.

Bu yöntem, özellikle insan hayatını tehlikeye atabilecek veya büyük maddi zararlara yol açabilecek arızaların tespitinde kullanılabilir.

Gelecekte yapılacak çalışmalarda, odaklanmış kayıp fonksiyonunun genel performans metriklerine olan etkisi üzerinde durulabilir. Özellikle yanlış pozitif tahminlerin azaltılmasına yönelik iyileştirmeler, sistemin genel başarımını daha da artırabilir.



KAYNAKÇA

ARTC. (2006). *Rail Defects Handbook: Some Rail Defects, their Characteristics, Causes and Control*. Australian Rail Track Corporation.

Aydin, İl., Sevi, M., Salur, M. U., & Akin, E. (2022). Defect classification of railway fasteners using image preprocessing and alightweight convolutional neural network. *Turkish Journal of Electrical Engineering and Computer Sciences*, 30(3), 891-907. <https://doi.org/10.55730/1300-0632.3817>

Bochkovski, A., Wang, C.-Y., & Liao, H.-Y. M. (2020). *YOLOv4: Optimal Speed and Accuracy of Object Detection* (arXiv:2004.10934). arXiv. <http://arxiv.org/abs/2004.10934>

Broken rail. (2011). Wikidata. <https://www.wikidata.org/wiki/Q10658429>

Chandran, P., Asber, J., Thiery, F., Odelius, J., & Rantatalo, M. (2021). An Investigation of Railway Fastener Detection Using Image Processing and Augmented Deep Learning. *Sustainability*, 13(21), Article 21. <https://doi.org/10.3390/su132112051>

CSDN. (2017, Eylül 7). *Derin öğrenme [17] Nesne algılama: Odak Kaybının ters türetilmesi ve darknet_focal kaybı türetme üzerinde uygulanması-CSDN Blogu*. <https://blog.csdn.net/linmingan/article/details/77885832>

Duan, K., Bai, S., Xie, L., Qi, H., Huang, Q., & Tian, Q. (2019). *CenterNet: Keypoint Triplets for Object Detection* (arXiv:1904.08189). arXiv. <https://doi.org/10.48550/arXiv.1904.08189>

Faghih-Roohi, S., Hajizadeh, S., Nunez, A., Babuska, R., & De Schutter, B. (2016). Deep convolutional neural networks for detection of rail surface defects. *2016 International Joint Conference on Neural Networks (IJCNN)*, 2584-2589. <https://doi.org/10.1109/IJCNN.2016.7727522>

Fanny, & Cenggoro, T. W. (2018). Deep Learning for Imbalance Data Classification using Class Expert Generative Adversarial Network. *Procedia Computer Science*, 135, 60-67. <https://doi.org/10.1016/j.procs.2018.08.150>

Feng, H., Jiang, Z., Xie, F., Yang, P., Shi, J., & Chen, L. (2014). Automatic Fastener Classification and Defect Detection in Vision-Based Railway Inspection Systems. *IEEE Transactions on Instrumentation and Measurement*, 63(4), 877-888. *IEEE Transactions on Instrumentation and Measurement*. <https://doi.org/10.1109/TIM.2013.2283741>

- Gan, J., Li, Q., Wang, J., & Yu, H.** (2017). A Hierarchical Extractor-Based Visual Rail Surface Inspection System. *IEEE Sensors Journal*, 17(23), 7935-7944. IEEE Sensors Journal. <https://doi.org/10.1109/JSEN.2017.2761858>
- Giben, X., Patel, V. M., & Chellappa, R.** (2015a). Material classification and semantic segmentation of railway track images with deep convolutional neural networks. *2015 IEEE International Conference on Image Processing (ICIP)*, 621-625. <https://doi.org/10.1109/ICIP.2015.7350873>
- Giben, X., Patel, V. M., & Chellappa, R.** (2015b). Material classification and semantic segmentation of railway track images with deep convolutional neural networks. *2015 IEEE International Conference on Image Processing (ICIP)*, 621-625. <https://doi.org/10.1109/ICIP.2015.7350873>
- Gibert, X., Patel, V. M., & Chellappa, R.** (2017). Deep Multitask Learning for Railway Track Inspection. *IEEE Transactions on Intelligent Transportation Systems*, 18(1), 153-164. <https://doi.org/10.1109/TITS.2016.2568758>
- Girshick, R.** (2015). *Fast R-CNN* (arXiv:1504.08083). arXiv. <https://doi.org/10.48550/arXiv.1504.08083>
- Girshick, R., Donahue, J., Darrell, T., & Malik, J.** (2013, Kasım 11). Rich feature hierarchies for accurate object detection and semantic segmentation. *arXiv.Org*. <https://arxiv.org/abs/1311.2524v5>
- GitHub.** (2018). *Loss function formula · Issue #821 · AlexeyAB/darknet*. GitHub. <https://github.com/AlexeyAB/darknet/issues/821>
- GitHub.** (2019a). *How to implement focal loss in DARKNET · Issue #2594 · AlexeyAB/darknet*. GitHub. <https://github.com/AlexeyAB/darknet/issues/2594>
- GitHub.** (2019b). *Multi class training · Issue #2441 · AlexeyAB/darknet*. GitHub. <https://github.com/AlexeyAB/darknet/issues/2441>
- GitHub.** (2021). *Darknet/src/yolo_layer.c at master · AlexeyAB/darknet*. GitHub. https://github.com/AlexeyAB/darknet/blob/master/src/yolo_layer.c
- Guo, F., Qian, Y., & Shi, Y.** (2021). Real-time railroad track components inspection based on the improved YOLOv4 framework. *Automation in Construction*, 125, 103596. <https://doi.org/10.1016/j.autcon.2021.103596>

- Guo, J., Zhang, S., Qian, Y., & Wang, Y.** (2023). An adaptively weighted loss-enabled lightweight teacher–student model for real-time railroad inspection on edge devices. *Neural Computing and Applications*, 35(34), 24455-24472. <https://doi.org/10.1007/s00521-023-09038-2>
- Güçlü, E., Aydın, İ., Şahbaz, K., Akın, E., & Karaköse, M.** (2021). Demiryolu Bağlantı Elemanlarında Bulunan Kusurların YOLOv4 ve Bulanık Mantık Kullanarak Tespiti. *Demiryolu Mühendisliği*, 14, Article 14. <https://doi.org/10.47072/demiryolu.939830>
- Güçlü, E., Aydın, İ., Şahbaz, K., Akın, E., & Karaköse, M.** (2021). Demiryolu Bağlantı Elemanlarında Bulunan Kusurların YOLOv4 ve Bulanık Mantık Kullanarak Tespiti. *Demiryolu Mühendisliği*, 14, 249-262. <https://doi.org/10.47072/demiryolu.939830>
- He, J., Wang, W., & Yang, N.** (2024). Research on Track Fastener Service Status Detection Based on Improved YOLOv4 Model. *Journal of Transportation Technologies*, 14(2), Article 2. <https://doi.org/10.4236/jtts.2024.142013>
- He, K., Gkioxari, G., Dollár, P., & Girshick, R.** (2018). *Mask R-CNN* (arXiv:1703.06870). arXiv. <https://doi.org/10.48550/arXiv.1703.06870>
- He, K., Zhang, X., Ren, S., & Sun, J.** (2014). *Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition* (C. 8691, ss. 346-361). https://doi.org/10.1007/978-3-319-10578-9_23
- Hsieh, C.-C., Lin, Y.-W., Tsai, L.-H., Huang, W.-H., Hsieh, S.-L., & Hung, W.-H.** (2020). Offline Deep-learning-based Defective Track Fastener Detection and Inspection System. *Sensors and Materials*, 32(10), 3429. <https://doi.org/10.18494/SAM.2020.2921>
- Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q.** (2017). Densely Connected Convolutional Networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2261-2269. <https://doi.org/10.1109/CVPR.2017.243>
- James, A., Jie, W., Xulei, Y., Chenghao, Y., Ngan, N. B., Yuxin, L., Yi, S., Chandrasekhar, V., & Zeng, Z.** (2018). TrackNet—A Deep Learning Based Fault Detection for Railway Track Inspection. *2018 International Conference on Intelligent Rail Transportation (ICIRT)*, 1-5. <https://doi.org/10.1109/ICIRT.2018.8641608>
- Jiang, H., Li, Y., Li, C., & Li, Y.** (2023). Semi-Supervised Framework for Rail Track Surface Damage Detection: Rail track surface damage detected by RTDS-Net. *2023 4th*

International Symposium on Computer Engineering and Intelligent Communications (ISCEIC), 510-514. <https://doi.org/10.1109/ISCEIC59030.2023.10271217>

Jie, L., Siwei, L., Qingyong, L., Hanqing, Z., & Shengwei, R. (2009). Real-time rail head surface defect detection: A geometrical approach. *2009 IEEE International Symposium on Industrial Electronics*, 769-774. <https://doi.org/10.1109/ISIE.2009.5214088>

Jocher, G. (2020). *YOLOv5 by Ultralytics*. PyTorch. <https://github.com/ultralytics/yolov5>

Johnson, J. M., & Khoshgoftaar, T. M. (2019). Survey on deep learning with class imbalance. *Journal of Big Data*, 6(1), 27. <https://doi.org/10.1186/s40537-019-0192-5>

Kaul, P., Xie, W., & Zisserman, A. (2022). *Label, Verify, Correct: A Simple Few Shot Object Detection Method* (arXiv:2112.05749). arXiv. <https://doi.org/10.48550/arXiv.2112.05749>

labelImg. (2015). [Python]. HumanSignal. <https://github.com/HumanSignal/labelImg> (Original work published 2015)

Lakušić, S., Haladin, I., & Ahac, M. (2016). The Effect of Rail Fastening System Modifications on Tram Traffic Noise and Vibration. *Shock and Vibration*, 2016(1), 4671302. <https://doi.org/10.1155/2016/4671302>

Lee, D.-H. (2013). *Pseudo-Label: The Simple and Efficient Semi-Supervised Learning Method for Deep Neural Networks*.

Li, C., Li, L., Jiang, H., Weng, K., Geng, Y., Li, L., Ke, Z., Li, Q., Cheng, M., Nie, W., Li, Y., Zhang, B., Liang, Y., Zhou, L., Xu, X., Chu, X., Wei, X., & Wei, X. (2022). *YOLOv6: A Single-Stage Object Detection Framework for Industrial Applications* (arXiv:2209.02976). arXiv. <https://doi.org/10.48550/arXiv.2209.02976>

Li, Q., & Ren, S. (2012). A Real-Time Visual Inspection System for Discrete Surface Defects of Rail Heads. *IEEE Transactions on Instrumentation and Measurement*, 61(8), 2189-2199. *IEEE Transactions on Instrumentation and Measurement*. <https://doi.org/10.1109/TIM.2012.2184959>

Liao, X., Lv, S., Li, D., Luo, Y., Zhu, Z., & Jiang, C. (2021). YOLOv4-MN3 for PCB Surface Defect Detection. *Applied Sciences*, 11(24), 11701. <https://doi.org/10.3390/app112411701>

- Lin, T.-Y., Goyal, P., Girshick, R., He, K., & Dollár, P.** (2018). *Focal Loss for Dense Object Detection* (arXiv:1708.02002). arXiv. <https://doi.org/10.48550/arXiv.1708.02002>
- Lin, Y.-W., Hsieh, C.-C., Huang, W.-H., Hsieh, S.-L., & Hung, W.-H.** (2019). Railway Track Fasteners Fault Detection using Deep Learning. *2019 IEEE Eurasia Conference on IOT, Communication and Engineering (ECICE)*, 187-190. <https://doi.org/10.1109/ECICE47484.2019.8942769>
- Liu, S., Qi, L., Qin, H., Shi, J., & Jia, J.** (2018). *Path Aggregation Network for Instance Segmentation* (arXiv:1803.01534). arXiv. <https://doi.org/10.48550/arXiv.1803.01534>
- Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., & Berg, A. C.** (2016). *SSD: Single Shot MultiBox Detector* (C. 9905, ss. 21-37). https://doi.org/10.1007/978-3-319-46448-0_2
- Liu, Z., Sun, J., Wang, H., & Zhang, G.** (2011). Simple and fast rail wear measurement method based on structured light. *Optics and Lasers in Engineering*, 49(11), 1343-1351. <https://doi.org/10.1016/j.optlaseng.2011.05.014>
- Lu, D., & Weng, Q.** (2007). A survey of image classification methods and techniques for improving classification performance. *International Journal of Remote Sensing*, 28(5), 823-870. <https://doi.org/10.1080/01431160600746456>
- Mi, Z., Chen, R., & Zhao, S.** (2023). Research on steel rail surface defects detection based on improved YOLOv4 network. *Frontiers in Neurorobotics*, 17, 1119896. <https://doi.org/10.3389/fnbot.2023.1119896>
- Min, Y., Xiao, B., Dang, J., Yue, B., & Cheng, T.** (2018). Real time detection system for rail surface defects based on machine vision. *EURASIP Journal on Image and Video Processing*, 2018(1), 3. <https://doi.org/10.1186/s13640-017-0241-y>
- Mittal, A.** (2020). Haar Cascades, Explained. *Analytics Vidhya*. <https://medium.com/analytics-vidhya/haar-cascades-explained-38210e57970d>
- Mittal, S., & Rao, D.** (2018). *Vision Based Railway Track Monitoring using Deep Learning* (arXiv:1711.06423). arXiv. <https://doi.org/10.48550/arXiv.1711.06423>
- Ozdemir, R., & Koc, M.** (2024). On the enhancement of semi-supervised deep learning-based railway defect detection using pseudo-labels. *Expert Systems with Applications*, 251, 124105. <https://doi.org/10.1016/j.eswa.2024.124105>

- Peng, Y., Liu, C., Yan, Y., Ma, N., Wang, D., Liu, C., & Chen, Q.** (2023). Semi-Supervised Bolt Anomaly Detection Based on Local Feature Reconstruction. *IEEE Transactions on Instrumentation and Measurement*, 72, 1-12. IEEE Transactions on Instrumentation and Measurement. <https://doi.org/10.1109/TIM.2023.3278293>
- Pham, H., Dai, Z., Xie, Q., Luong, M.-T., & Le, Q. V.** (2021). *Meta Pseudo Labels* (arXiv:2003.10580). arXiv. <https://doi.org/10.48550/arXiv.2003.10580>
- Qi, H., Xu, T., Wang, G., Cheng, Y., & Chen, C.** (2020a). MYOLOv3-Tiny: A new convolutional neural network architecture for real-time detection of track fasteners. *Computers in Industry*, 123, 103303. <https://doi.org/10.1016/j.compind.2020.103303>
- Qi, H., Xu, T., Wang, G., Cheng, Y., & Chen, C.** (2020b). MYOLOv3-Tiny: A new convolutional neural network architecture for real-time detection of track fasteners. *Computers in Industry*, 123, 103303. <https://doi.org/10.1016/j.compind.2020.103303>
- Rampriya, R. S., Suganya, R., Nathan, S., & Perumal, P. S.** (2022). A Comparative Assessment of Deep Neural Network Models for Detecting Obstacles in the Real Time Aerial Railway Track Images. *Applied Artificial Intelligence*, 36(1), 2018184. <https://doi.org/10.1080/08839514.2021.2018184>
- Redmon, J., Divvala, S., Girshick, R., & Farhadi, A.** (2016). You Only Look Once: Unified, Real-Time Object Detection. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 779-788. <https://doi.org/10.1109/CVPR.2016.91>
- Redmon, J., & Farhadi, A.** (2016). *YOLO9000: Better, Faster, Stronger* (arXiv:1612.08242). arXiv. <https://doi.org/10.48550/arXiv.1612.08242>
- Redmon, J., & Farhadi, A.** (2018). YOLOv3: An Incremental Improvement. *arXiv:1804.02767 [Cs]*. <http://arxiv.org/abs/1804.02767>
- Ren, S., He, K., Girshick, R., & Sun, J.** (2016). *Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks* (arXiv:1506.01497). arXiv. <https://doi.org/10.48550/arXiv.1506.01497>
- Riloff, E.** (1996). *Automatically Generating Extraction Patterns from Untagged Text*.
- Santur, Y.** (2018). *Lazer kameralar kullanılarak bilgisayar görmesi ile üç boyutlu ray profili arızalarının gerçek zamanlı teşhisi*. <https://platform.almanhal.com/Details/Thesis/2000038467>

Santur, Y., Karaköse, M., & Akin, E. (2017). A new rail inspection method based on deep learning using laser cameras. *2017 International Artificial Intelligence and Data Processing Symposium (IDAP)*, 1-6. <https://doi.org/10.1109/IDAP.2017.8090245>

Santur, Y., Karaköse, M., Aydın, İ., & Akin, E. (t.y.). (2016). *IMU based adaptive blur removal approach using image processing for railway inspection.*, gönderen <https://ieeexplore.ieee.org/abstract/document/7502729>

Sevi, M., Aydın, İ., & Karaköse, M. (2022). Derin Öğrenme Yöntemleri ile Demiryolu Bağlantı Elemanlarının Sınıflandırılması. *European Journal of Science and Technology*. <https://doi.org/10.31590/ejosat.1029905>

Sohn, K., Zhang, Z., Li, C.-L., Zhang, H., Lee, C.-Y., & Pfister, T. (2020). A Simple Semi-Supervised Learning Framework for Object Detection. *arXiv:2005.04757 [Cs]*. <http://arxiv.org/abs/2005.04757>

Szandala, T. (2021). Review and Comparison of Commonly Used Activation Functions for Deep Neural Networks. İçinde A. K. Bhoi, P. K. Mallick, C.-M. Liu, & V. E. Balas (Ed.), *Bio-inspired Neurocomputing* (ss. 203-224). Springer. https://doi.org/10.1007/978-981-15-5495-7_11

Şener, A., Ergen, B., & Toğaçar, M. (2022a). Fault Detection from Images of Railroad Lines Using the Deep Learning Model Built with the Tensorflow Library. *Turkish Journal of Science and Technology*, 17(1), 47-53. <https://doi.org/10.55525/tjst.1056283>

Şener, A., Ergen, B., & Toğaçar, M. (2022b). Fault Detection from Images of Railroad Lines Using the Deep Learning Model Built with the Tensorflow Library. *Turkish Journal of Science and Technology*, 17(1), 47-53. <https://doi.org/10.55525/tjst.1056283>

Tan, M., Pang, R., & Le, Q. V. (2020). *EfficientDet: Scalable and Efficient Object Detection* (arXiv:1911.09070). arXiv. <http://arxiv.org/abs/1911.09070>

Taştimur, C., Akin, E., Karaköse, M., & Aydın, İ. (2015). Detection of rail faults using morphological feature extraction based image processing. *2015 23rd Signal Processing and Communications Applications Conference (SIU)*, 1244-1247. <https://doi.org/10.1109/SIU.2015.7130063>

TCDD. (2013a). *Hızlı Tren Setinde Yol ve Katener Ölçüm Cihazları Hat Denetimi Ve Ölçüm Sistemi (V-CUBE) İşletim ve Bakım Kılavuzu.*

TCDD. (2013b). *TCDD HAT BAKIM EL KİTABI.*

TCDD. (2017). *Piri Reis Yüksek Hızlı Test Treni Raporu*. TCDD.

Torres, J. (2024). *YOLOv8 Architecture Explained: Exploring the YOLOv8 Architecture - YOLOv8*. <https://yolov8.org/yolov8-architecture-explained/>

Viola, P., & Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. CVPR 2001, 1, I-511-I-518. <https://doi.org/10.1109/CVPR.2001.990517>

Wang, C.-Y., Bochkovskiy, A., & Liao, H.-Y. M. (2022). *YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors* (arXiv:2207.02696). arXiv. <https://doi.org/10.48550/arXiv.2207.02696>

Wang, C.-Y., Liao, H.-Y. M., Yeh, I.-H., Wu, Y.-H., Chen, P.-Y., & Hsieh, J.-W. (2019). *CSPNet: A New Backbone that can Enhance Learning Capability of CNN* (arXiv:1911.11929). arXiv. <https://doi.org/10.48550/arXiv.1911.11929>

Wang, S., Liu, F., & Liu, B. (2022). Semi-Supervised Deep Learning in High-Speed Railway Track Detection Based on Distributed Fiber Acoustic Sensing. *Sensors (Basel, Switzerland)*, 22(2), 413. <https://doi.org/10.3390/s22020413>

Wei, X., Yang, Z., Liu, Y., Wei, D., Jia, L., & Li, Y. (2019). Railway track fastener defect detection based on image processing and deep learning techniques: A comparative study. *Engineering Applications of Artificial Intelligence*, 80, 66-81. <https://doi.org/10.1016/j.engappai.2019.01.008>

Wilson, J. R., & Norris, B. J. (2005). Rail human factors: Past, present and future. *Applied Ergonomics*, 36(6), 649-660. <https://doi.org/10.1016/j.apergo.2005.07.001>

Xie, Q., Luong, M.-T., Hovy, E., & Le, Q. V. (2020). *Self-training with Noisy Student improves ImageNet classification* (arXiv:1911.04252). arXiv. <https://doi.org/10.48550/arXiv.1911.04252>

Xu, M., Zhang, Z., Hu, H., Wang, J., Wang, L., Wei, F., Bai, X., & Liu, Z. (2021). *End-to-End Semi-Supervised Object Detection with Soft Teacher* (arXiv:2106.09018). arXiv. <https://doi.org/10.48550/arXiv.2106.09018>

Yilmazer, M., Karaköse, M., Aydın, İ., & Akin, E. (2022). Mask R-CNN Derin Sinir Ağı ile Demiryolu Hattı Bileşenlerinde Çoklu Hata Tespiti. *Çukurova Üniversitesi Mühendislik Fakültesi Dergisi*, 37(4), 1103-1112. <https://doi.org/10.21605/cukurovaumfd.1230955>

Zheng, D., Li, L., Zheng, S., Chai, X., Zhao, S., Tong, Q., Wang, J., & Guo, L. (2021). A Defect Detection Method for Rail Surface and Fasteners Based on Deep Convolutional Neural Network. *Computational Intelligence and Neuroscience*, 2021(1), 2565500. <https://doi.org/10.1155/2021/2565500>





EKLER

EK-1: İLK DÜZENLEME SONUCU ELDE EDİLEN SINIFLAR

İlk düzenleme sonucu elde edilen toplam sınıf sayısı ve bu sınıflara ait adet ve boyut bilgisi aşağıda gösterilmiştir. Belirtilen sınıflara ait veri sayılarının büyük çoğunluğunun 241.761 adet görüntü ile ‘Travers’ sınıflarına ait olduğu görülmektedir. Bu iki sınıfın sahip olduğu görüntü sayısı toplam görüntü sayısının %85’ini oluşturmaktadır. Bu da oldukça büyük görünen veri kümesinin aslında dengesiz olduğunu göstermektedir ayrıca bazı sınıflardaki görüntü sayısı 25 adetin altındadır.

No	Sınıf	Adet	Boyut	Lazer görüntüsü
1	Dingil	15	137x1024	1
2	Eksik_balast_Sol	5766	540x1024	1
3	Eksik_balast_Sağ	8976	540x1024	1
4	Balast_Sol	704	62x1024	1
5	Balast_Sağ	1655	352x1024	1
6	Cıvata_Sol	891	480x480	0
7	Cıvata_Sağ	552	388x480	0
8	Ray_kontrolü	23	220x1024	1
9	Hemzenin_geçit	198	71x1024	1
10	Bağlantı_elemanı_Model_Sol	1586	400x480	0
11	Bağlantı_elemanı_Model_Sağ	1837	388x480	0
12	Yabancı_madde	561	527x1024	1
13	Kalkmış_travers_Sol	1474	806x1024	1
14	Kalkmış_travers_Sağ	1438	715x1024	1
15	Yanlış_hizalanmış_travers_Sol	197	529x1024	0
16	Yanlış_hizalanmış_travers_Sağ	183	529x1024	0
17	Dar_kavrama_boşluğu	1	120x169	0
18	Eksik_kenar_küpeşesi	311	140x1024	1
19	Ray_kırığı	901	96x391	0
20	Hasarlı_bağlantı_elemanı_Sol	2634	448x450	0
21	Hasarlı_bağlantı_elemanı_Sağ	3573	404x450	0
22	Signum	4123	768x1024	1
23	Travers_Sol	129520	500x300	0
24	Travers_Sağ	112241	500x300	0
25	Yüzey_kusuru	6477	112x169	0
26	Makas	237	109x1024	1

EK-2: İKİNCİ VE ÜÇÜNCÜ DÜZENLEME SONUCU ELDE EDİLEN SINIFLAR

Sonraki aşamada lazer görüntüleri ve diğer elenen sınıflar çıkartıldığında 12 sınıftan oluşan yeni veri kümesi elde edildi. Bu veri kümesine ait oluşan yeni dağılım aşağıda gösterilmiştir.

İkinci düzenleme sonucu elde edilen sınıflar

No	Sınıf	Adet	Boyut
1	Cıvata_Sol	891	480×480
2	Cıvata_Sağ	552	388×480
3	Bağlantı elemanı_Model_Sol	1586	400×480
4	Bağlantı elemanı_Model_Sağ	1837	388×480
5	Yanlış hizalanmış travers_Sol	197	529×1024
6	Yanlış hizalanmış travers_Sağ	183	529×1024
7	Ray kırığı	901	96×391
8	Hasarlı bağlantı elemanı_Sol	2634	448×450
9	Hasarlı bağlantı elemanı_Sağ	3573	404×450
10	Travers_Sol	129520	500×300
11	Travers_Sağ	112241	500×300
12	Yüzey kusuru	6477	112×169

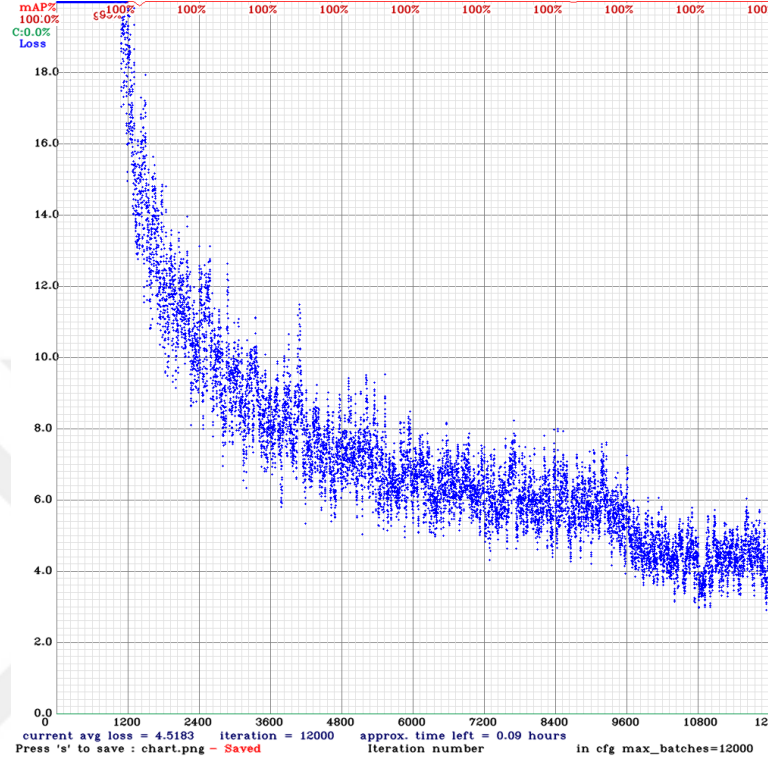
Bir sonraki aşamada ise aynı nesneler sol ve sağ şeklinde iki ayrı sınıfa ayrılmak yerine tek bir sınıf altında toplandı, veri kümesini daha dengeli hale getirmek için ‘Travers’ kümesinin görüntü sayısı 1000 ile sınırlandırıldı ve sonuç olarak üçüncü düzenlemede sınıf sayısı 7’ye düşürüldü. üçüncü düzenleme sonucu elde edilen toplam sınıf sayısı ve bu sınıflara ait adet ve boyut bilgisi gösterilmiştir.

Üçüncü düzenleme sonucu elde edilen sınıflar

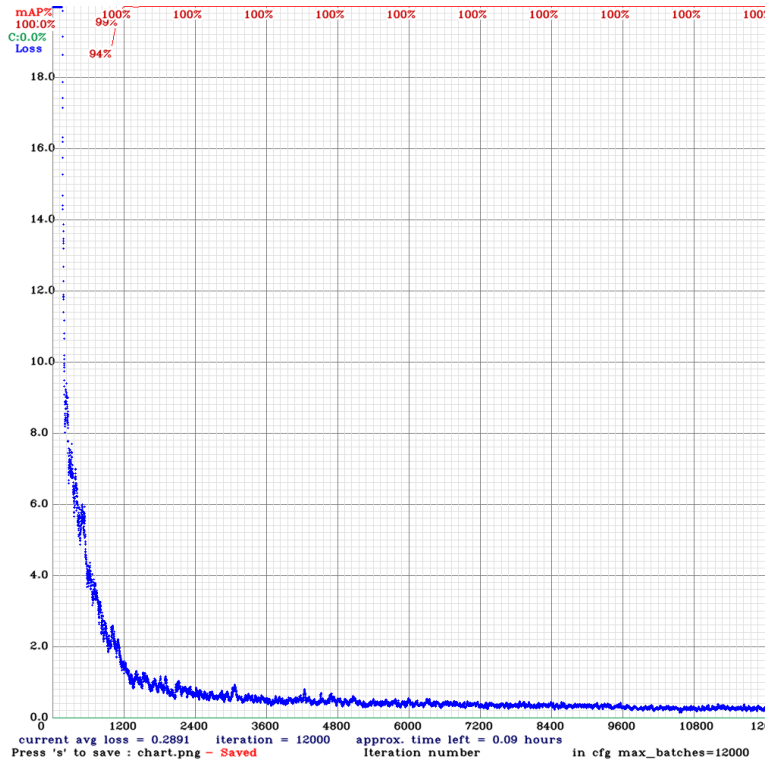
No	Sınıf	Adet	Boyut
1	Cıvata	1443	480×480
2	Bağlantı elemanı_Model	3423	400×480
3	Yanlış hizalanmış travers	380	529×1024
4	Ray kırığı	901	96×391
5	Hasarlı bağlantı elemanı	6207	448×450
6	Travers	1000	500×300
7	Yüzey kusuru	6477	112×169

EK-3: BÖLÜM 4'TE ELE ALINAN DEMİRYOLU KUSUR TESPİT ALGORİTMASININ MISH AKTİVASYON FONKSİYONU KULLANILAN EĞİTİM GRAFİĞİ

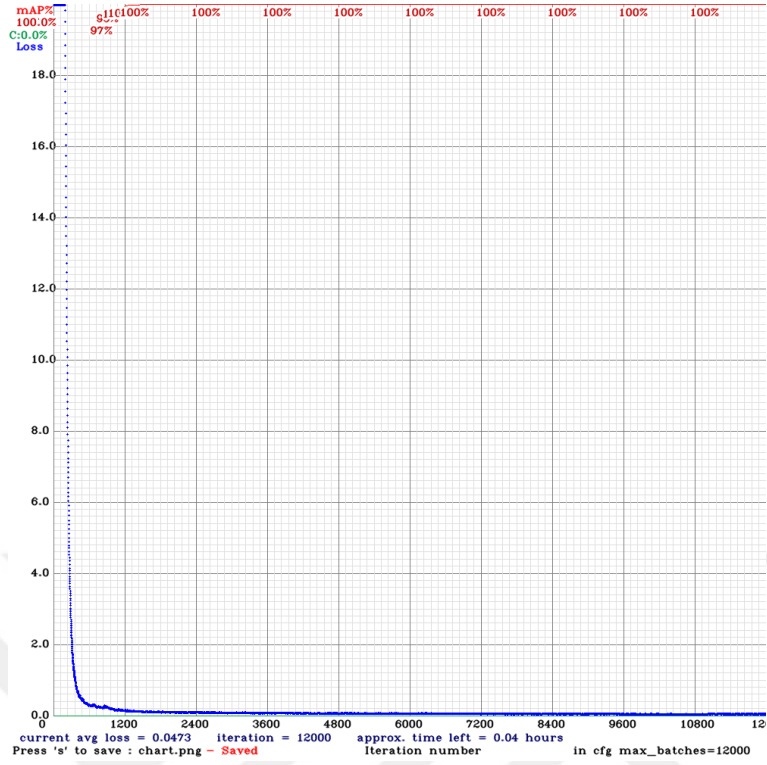
Mish aktivasyonu fonksiyonu kullanılan eğitim grafiği



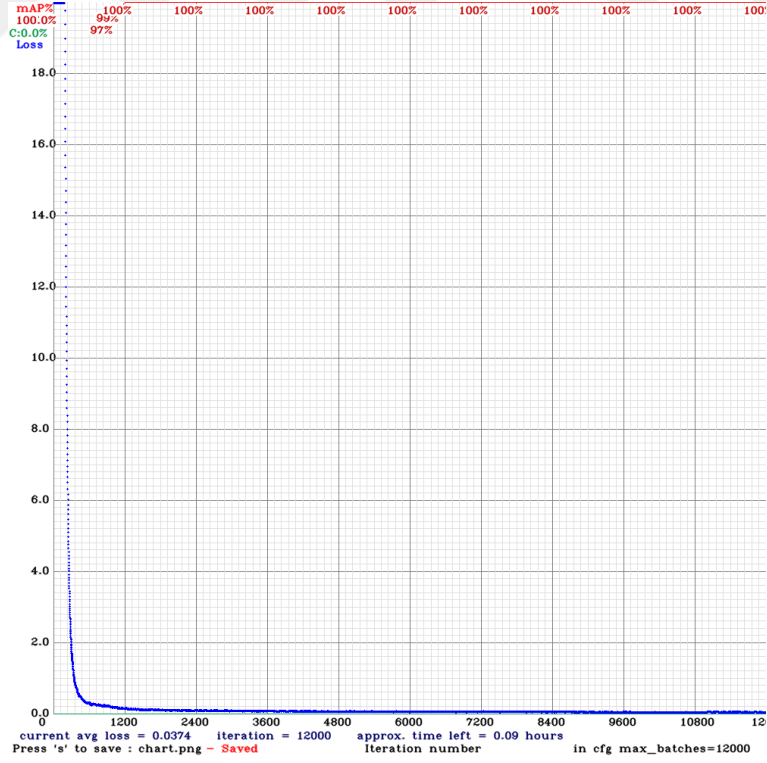
Leaky aktivasyonu fonksiyonu kullanılan eğitim grafiği



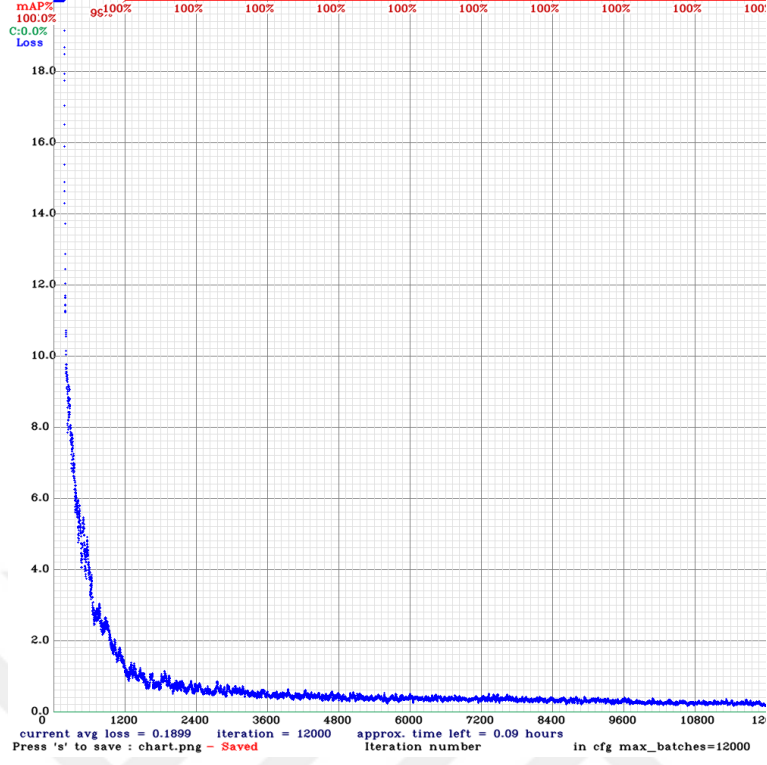
Swish aktivasyonu fonksiyonu kullanılan eğitim grafiği



X- swish aktivasyonu fonksiyonu kullanılan eğitim grafiği

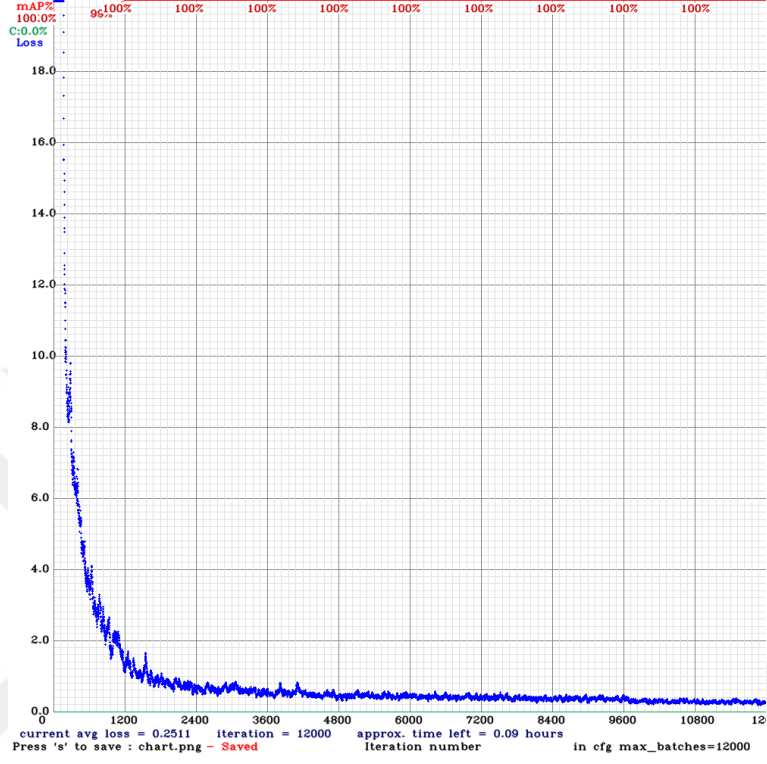


Mish* aktivasyonu fonksiyonu kullanılan eğitim grafiği

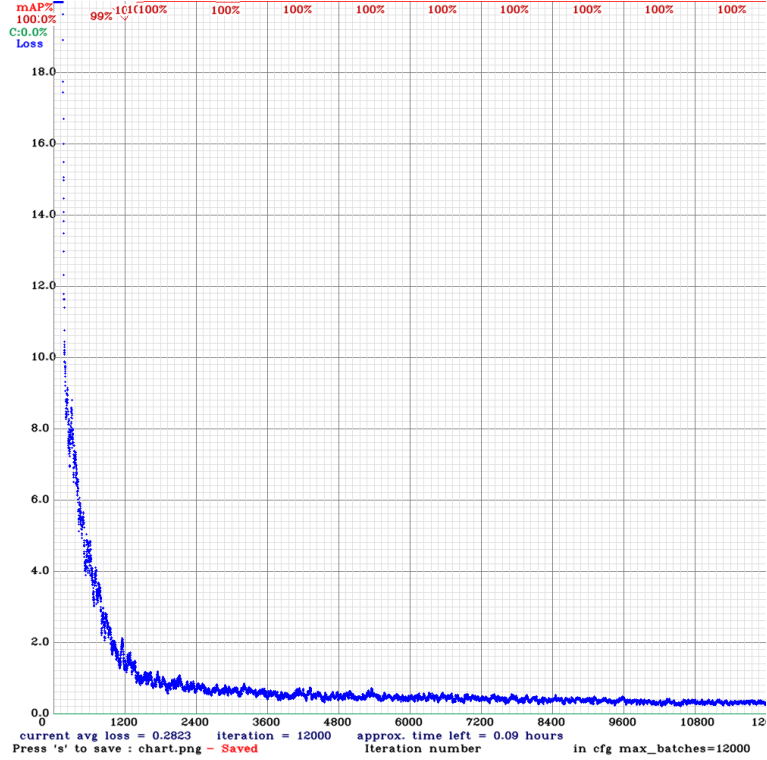


EK-4: BÖLÜM 4'TE ELE ALINAN DEMİRYOLU KUSUR TESPİT ALGORİTMASININ MISH* AKTİVASYON FONKSİYONU KULLANILAN EĞİTİM GRAFİĞİ

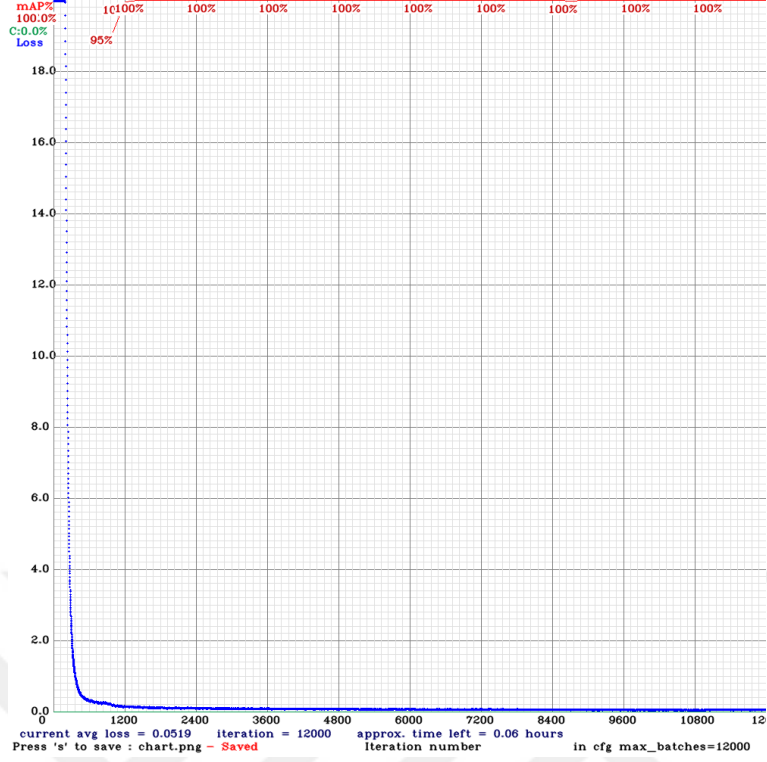
Mish* aktivasyonu fonksiyonu kullanılan eğitim grafiği



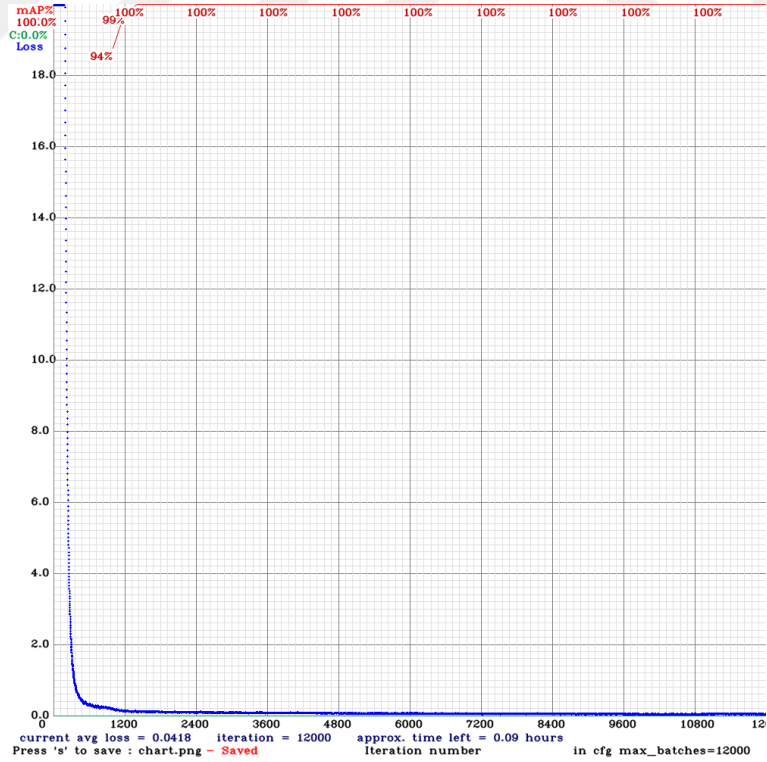
Leaky aktivasyonu fonksiyonu kullanılan eğitim grafiği



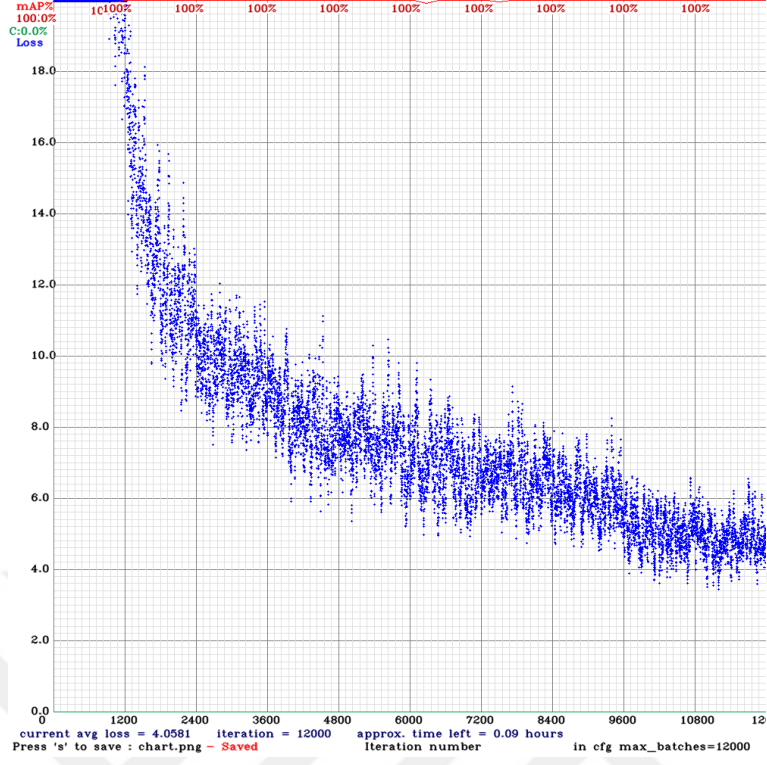
Swish aktivasyonu fonksiyonu kullanılan eğitim grafiği



X- swish aktivasyonu fonksiyonu kullanılan eğitim grafiği

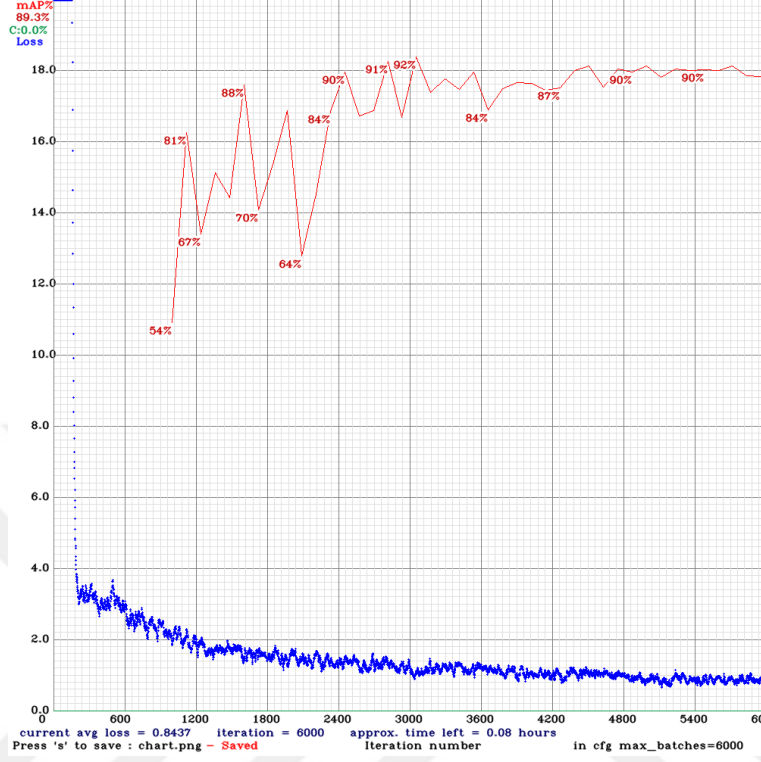


Mish aktivasyonu fonksiyonu kullanılan eğitim grafiği

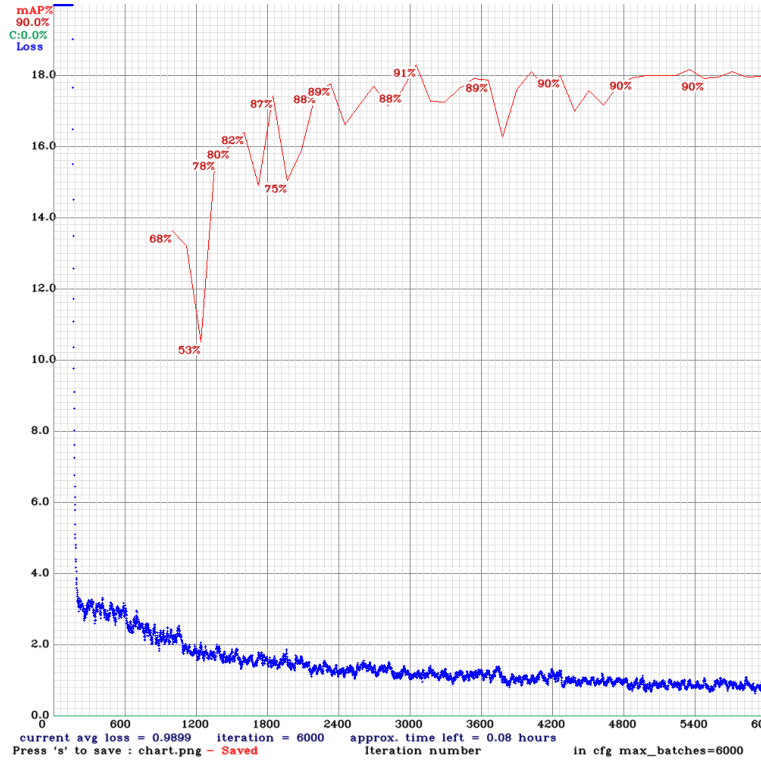


EK-5: BÖLÜM 5'TE ELE ALINAN DEMİRYOLU KUSUR TESPİT ALGORİTMASININ EĞİTİM GRAFİĞİ

$\alpha = 0.375$ parametresinin seçildiği eğitim grafiği



$\alpha = 1$ parametresinin seçildiği eğitim grafiği



$\alpha = 0,4$ parametresinin seçildiği eğitim grafiği

