

T.C.  
İSTANBUL GEDİK ÜNİVERSİTESİ  
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ



**DERİN ÖĞRENME KULLANARAK  
AKILLI CİHAZ GÖRÜNTÜLERİNİN  
KALİTE DEĞERLENDİRMESİ**

**YÜKSEK LİSANS TEZİ**

**Mehmet Emin EKER**

**Yapay Zekâ Mühendisliği Anabilim Dalı**

**Yapay Zekâ Mühendisliği Tezli Yüksek Lisans Programı**

**AĞUSTOS 2025  
İSTANBUL**

T.C.  
İSTANBUL GEDİK ÜNİVERSİTESİ  
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ



**DERİN ÖĞRENME KULLANARAK  
AKILLI CİHAZ GÖRÜNTÜLERİNİN  
KALİTE DEĞERLENDİRMESİ**

**YÜKSEK LİSANS TEZİ**

**Mehmet Emin EKER  
(231239004)**

**Yapay Zekâ Mühendisliği Anabilim Dalı**

**Yapay Zekâ Mühendisliği Tezli Yüksek Lisans Programı**

**Tez Danışmanı: Dr. Öğr. Üyesi Erdoğan BOZKURT**

**İstanbul 2025**

T.C.  
İSTANBUL GEDİK ÜNİVERSİTESİ  
Lisansüstü Eğitim Enstitüsü Müdürlüğü

Jüri Tez Onay Formu

27.08.2025

LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ MÜDÜRLÜĞÜ

Bu çalışma 27.08.2025 tarihinde aşağıdaki jüri tarafından Yapay Zekâ Mühendisliği Anabilim Dalı, Yapay Zekâ Mühendisliği (Tezli Yüksek Lisans) Programı Yüksek Lisans Tezi olarak kabul edilmiştir.

TEZ JÜRİSİ

**Dr. Öğr. Üyesi Erdoğan BOZKURT**

Danışman

İstanbul Gedik Üniversitesi

Üye (İmza)

**Prof. Dr. İlhami Muharrem ORAK**

Karabük Üniversitesi

Üye (İmza)

**Dr. Öğr. Üyesi Ümit BOZOKLU**

İstanbul Gedik Üniversitesi

## YEMİN METNİ

Yüksek Lisans tezi olarak sunduğum “Derin Öğrenme Kullanarak Akıllı Cihaz Görüntülerinin Kalite Değerlendirmesi” başlıklı bu çalışmanın, bilimsel ahlak ve geleneklere uygun şekilde tarafımdan yazıldığını, bu tezdeki bütün bilgileri akademik ve etik kurallar içinde elde ettiğimi, yararlandığım eserlerin tamamının kaynaklarda gösterildiğini ve çalışmamın içinde kullanıldıkları her yerde bunlara atıf yapıldığını, patent ve telif haklarını ihlal edici bir davranışımın olmadığını belirtir ve bunu onurumla doğrularım (27/08/2025).

Mehmet Emin EKER

## ÖNSÖZ

Bu tez çalışmasının hazırlanma sürecinde katkı ve destekleriyle yanımda olan değerli kişilere teşekkür etmeyi bir borç bilirim.

Öncelikle, çalışmamın her aşamasında bilgi ve tecrübesiyle yönlendiren, yapıcı eleştirileri ve yol göstericiliğiyle bana ışık tutan danışmanım Dr. Öğr. Üyesi Erdoğan Bozkurt'a en içten teşekkürlerimi sunarım. Süreç boyunca sabrı, anlayışı ve desteğiyle yanımda olan sevgili eşim Halime Suvay Eker'e minnettarlığımı ifade etmek isterim. Henüz dünyaya gelmemiş olan kızım İlay, bu çalışmanın tamamlanmasında en büyük motivasyon kaynaklarımdan biri olmuştur.

Veri hazırlama aşamasında göstermiş oldukları kıymetli katkılardan dolayı Eylül Dilara Eren ve kardeşim Gül Eker'e teşekkür ederim. Ayrıca, yoğun çalışma dönemlerinde moral kaynağı olan kedim Baybars ve minik kuşum Fıstık'a da şükranlarımı sunarım.

Bu tez, tüm bu değerli desteklerin bir araya gelmesiyle tamamlanabilmiştir.

Ağustos 2025

Mehmet Emin EKER

## İÇİNDEKİLER

	Sayfa No.
<b>ÖNSÖZ</b> .....	<b>iv</b>
<b>İÇİNDEKİLER</b> .....	<b>v</b>
<b>KISALTMALAR</b> .....	<b>vii</b>
<b>ÇİZELGE LİSTESİ</b> .....	<b>viii</b>
<b>ŞEKİL LİSTESİ</b> .....	<b>ix</b>
<b>ÖZET</b> .....	<b>x</b>
<b>ABSTRACT</b> .....	<b>xi</b>
<b>1. GİRİŞ</b> .....	<b>1</b>
1.1 Çalışmanın Konusu .....	1
1.2 Tezin Amacı.....	1
1.3 Literatür Araştırması .....	2
<b>2. YAPAY ZEKÂ</b> .....	<b>9</b>
<b>3. MAKİNE ÖĞRENMESİ</b> .....	<b>11</b>
3.1 Denetimli Öğrenme .....	11
3.2 Denetimsiz Öğrenme.....	12
3.3 Pekiştirmeli Öğrenme.....	12
<b>4. DERİN ÖĞRENME</b> .....	<b>13</b>
4.1 Tarihçe .....	13
4.2 Tanım.....	14
4.3 Evrişimsel Sinir Ağları .....	15
4.3.1 Evrişimli katman (Convolutional layer) .....	17
4.3.2 Havuzlama katmanı (Pooling layer) .....	17
4.3.3 Tam bağlantılı katman (Fully connected layer) .....	18
4.4 YOLO (You Only Look Once).....	19
4.4.1 YOLOv11 .....	21
4.5 ResNet .....	22
4.5.1 ResNet18 .....	22
4.5.2 ResNet34 .....	23

4.5.3 ResNet50 .....	23
4.5.4 ResNet101 .....	24
4.5.5 ResNet152 .....	24
<b>5. YÖNTEM.....</b>	<b>25</b>
5.1 Araştırmanın Modeli .....	25
5.2 Araştırmanın Veri Seti .....	25
5.3 Araştırmada Kullanılan Yöntemler.....	27
5.4 Araştırma için Geliştirilen Etiketleme Uygulaması .....	28
5.5 Modelin Bir Web Uygulamasında Kullanılması .....	32
5.5.1 PyTorch model tipi .....	33
5.5.2 TensorFlow.js model tipi .....	33
5.5.3 ONNX Runtime model tipi .....	34
<b>6. VERİNİN ANALİZİ VE BULGULAR .....</b>	<b>35</b>
6.1 YOLO Modelinden Elde Edilen Bulgular .....	35
6.2 ResNet18 Modelinden Elde Edilen Bulgular .....	41
6.3 YOLO ve ResNet18 Modellerinden Elde Edilen Bulguların Karşılaştırılması	44
<b>7. MODELİN KULLANILMASI .....</b>	<b>46</b>
7.1 Web Arayüzünün Geliştirilmesi .....	46
<b>8. SONUÇ VE ÖNERİLER.....</b>	<b>49</b>
<b>KAYNAKLAR.....</b>	<b>52</b>
<b>ÖZGEÇMİŞ.....</b>	<b>57</b>

## KISALTMALAR

<b>AI</b>	: Artificial Intelligence (Yapay Zekâ)
<b>CNN</b>	: Convolutional Neural Network (Evrifimsel Sinir Ađı)
<b>ESA</b>	: Evrifimsel Sinir Ađı
<b>GPU</b>	: Graphics Processing Unit (Grafik İřlem Birimi)
<b>LSTM</b>	: Long Short-Term Memory (Uzun Kısa Sfireli Hafıza)
<b>ML</b>	: Machine Learning (Makine Öğrenmesi)
<b>NoSQL</b>	: Not Only SQL (SQL Olmayan Veri Tabanı)
<b>ONNX</b>	: Open Neural Network Exchange
<b>ReLU</b>	: Rectified Linear Unit (Düzeltilmiş Doğrusal Birim)
<b>ResNet</b>	: Residual Network (Artık Ađ)
<b>YOLO</b>	: You Only Look Once
<b>YZ</b>	: Yapay Zekâ

## ÇİZELGE LİSTESİ

	<b>Sayfa No.</b>
Çizelge 5.1: Veri Seti Dağılımı .....	26
Çizelge 6.1: YOLO ve ResNet18 Model Metrikleri .....	44



## ŞEKİL LİSTESİ

	Sayfa No.
Şekil 4.1: Yapay Zekâ, Makine Öğrenmesi ve Derin Öğrenme Arasındaki İlişki .....	14
Şekil 4.2: Derin Öğrenme Sinir Ağı Yapısı .....	15
Şekil 4.3: Evrimsel Sinir Ağı (CNN) Mimarisi Akış Diyagramı.....	16
Şekil 4.4: Evrimsel Katman İşlemi.....	17
Şekil 4.5: Maksimum Havuzlama ve Ortalama Havuzlamanın Gösterimi .....	18
Şekil 4.6: Tam Bağlantılı Katman .....	18
Şekil 4.7: YOLO Sürümlerinin Performans Karşılaştırması.....	20
Şekil 5.1: "Kötü" Olarak Etiketlenen Örnek 10 Fotoğraf.....	26
Şekil 5.2: "İyi" Olarak Sınıflandırılan Örnek 10 Fotoğraf.....	27
Şekil 5.3: Etiketleme Uygulaması Giriş Ekranı .....	29
Şekil 5.4: Etiketleme Uygulaması Fotoğraf Etiketleme Ekranı .....	30
Şekil 5.5: Google Firebase Firestore Veritabanı .....	31
Şekil 6.1: Google Colab NVIDIA L4 GPU Kullanımı .....	35
Şekil 6.2: Derin Öğrenme Modeli Katmanları.....	36
Şekil 6.3: YOLO Model Eğitim Parametreleri.....	37
Şekil 6.4: YOLO Modeli Eğitimi Sonrası Elde Edilen Loss Grafikleri.....	38
Şekil 6.5: YOLO Modeli Eğitimi Sonrası Elde Edilen top-1 accuracy Grafiği.....	39
Şekil 6.6: YOLO Modeli Eğitimi Sonrası Elde Edilen Confusion Matrix Grafiği ....	39
Şekil 6.7: "İyi" Olarak Sınıflandırılan 10 Adet Fotoğraf Örneği.....	40
Şekil 6.8: "Kötü" Olarak Sınıflandırılan 10 Adet Fotoğraf Örneği .....	40
Şekil 6.9: ResNet18 Modeli Eğitimi Sonrası Elde Edilen Loss Grafiği .....	42
Şekil 6.10: ResNet18 Modeli Eğitimi Sonrası Elde Edilen Accuracy Grafiği.....	43
Şekil 6.11: ResNet18 Modeli Eğitimi Sonrası Elde Edilen Confusion Matrix Grafiği .....	43
Şekil 7.1: Web Uygulaması Fotoğraf Çekme Ekranı .....	47
Şekil 7.2: Web Uygulamasında Kötü Olarak Etiketlenen Fotoğraflar 1 .....	47
Şekil 7.3: Web Uygulamasında Kötü Olarak Etiketlenen Fotoğraflar 2 .....	48
Şekil 7.4: Web Uygulamasında İyi Olarak Etiketlenen Fotoğraflar.....	48

# DERİN ÖĞRENME KULLANARAK AKILLI CİHAZ GÖRÜNTÜLERİNİN KALİTE DEĞERLENDİRMESİ

## ÖZET

Derin öğrenme, yapay zekânın bir alt dalı olarak son yıllarda insan hayatının pek çok alanında önemli gelişmelere öncülük etmiş, sağlık, otomotiv, görüntü işleme gibi çeşitli disiplinlerde yaygın şekilde kullanılmaya başlanmıştır. Bu yöntemler, karmaşık verilerin analizinde yüksek başarı göstermesi nedeniyle giderek daha fazla uygulama alanı bulmaktadır. Özellikle görsel veri işleme alanında, derin öğrenme modelleri nesne tanıma, sınıflandırma ve kalite değerlendirme gibi kritik görevlerde etkin sonuçlar vermektedir.

Bir teknoloji şirketinin satış sonrası teknik servis departmanında, onarılmak üzere teslim alınan akıllı cihazların fotoğrafları onarım sonrası yaşanacak olası müşteri iddialarını doğrulamak amacıyla düzenli olarak çekilmekte ve en az altı ay boyunca saklanmak koşuluyla depolanmaktadır. Ancak kullanıcıların çekimlerdeki özen eksikliği nedeniyle pek çok fotoğraf kalitesiz olmaktadır. Bu durum, teknik servis süreçlerinde görüntü doğrulama aşamasını zorlaştırmakta ve aynı zamanda depolama alanında gereksiz bir israfa yol açmaktadır. Kalitesiz fotoğrafların büyük oranda sisteme dahil olması hem zaman kaybına hem de maliyet artışına sebep olmaktadır.

Bu soruna çözüm olarak, çekilen fotoğrafların kalitesini otomatik olarak tespit edebilen bir derin öğrenme modeli geliştirilmiştir. Model, görüntülerin aydınlık, bulanıklık, kadraj gibi çeşitli kalite parametrelerini değerlendirerek fotoğrafları “iyi” veya “kötü” olarak sınıflandırmaktadır. Elde edilen deneysel sonuçlar, geliştirilen derin öğrenme tabanlı modelin, teknik serviste çekilen fotoğrafların kalitesinin etkili ve güvenilir bir şekilde belirlenebildiğini göstermiştir. Sonuç olarak, teknik servis süreçlerinde fotoğraf kalitesinin otomatik olarak değerlendirilmesi sayesinde düşük kaliteli görseller elenebilir; böylece depolama alanından tasarruf sağlanarak iş süreçlerinde verimlilik artırılabilir.

**Anahtar Kelimeler:** *Yapay Zekâ, Derin Öğrenme, Görüntü Kalitesi Değerlendirmesi, Görüntü İşleme, Akıllı Cihazlar*

# QUALITY ASSESSMENT OF SMART DEVICE IMAGES USING DEEP LEARNING

## ABSTRACT

Deep learning, a subfield of artificial intelligence, has recently led significant advancements across various aspects of human life and has been increasingly applied in diverse disciplines such as healthcare, automotive, and image processing. Due to its high effectiveness in analyzing complex data, deep learning methods have found expanding application areas. In particular, deep learning models have demonstrated strong performance in critical tasks within visual data processing, including object recognition, classification, and quality assessment.

In the after-sales technical service department of a technology company, photographs of smart devices delivered for repair are routinely taken in order to verify potential customer claims after the repair. These photographs are stored for at least six months. However, due to users' lack of care during photography, many of the images are of poor quality. This situation complicates the image verification stage in technical service processes and leads to unnecessary storage waste. The inclusion of low-quality images in the system results in both time loss and increased costs.

To address this issue, a deep learning-based model was developed to automatically assess the quality of the captured images. The model evaluates various quality parameters such as brightness, blurriness, and framing, classifying the images as "good" or "poor." Experimental results demonstrate that the proposed deep learning model can effectively and reliably determine the quality of images taken in the technical service context.

Consequently, by automating the evaluation of image quality within technical service processes, low-quality images can be filtered out, enabling storage savings and improving operational efficiency.

**Keywords:** *Artificial Intelligence, Deep Learning, Image Quality Assessment, Image Processing, Smart Devices*

# 1. GİRİŞ

Derin öğrenme, yapay zekânın bir alt dalı olarak son yıllarda pek çok alanda önemli gelişmelere öncülük etmiş, çeşitli disiplinlerde yaygın şekilde kullanılmaya başlanmıştır. Bu yöntemler, karmaşık verilerin analizinde yüksek başarı göstermesi nedeniyle giderek daha fazla uygulama alanı bulmaktadır. Özellikle görsel veri işleme alanında nesne tanıma ve sınıflandırma gibi çalışmalarda etkili sonuçlar vermektedir.

## 1.1 Çalışmanın Konusu

Bu çalışmanın konusu, fotoğrafların kalitesini değerlendirebilen bir derin öğrenme modelinin geliştirilmesidir. Günümüzde dijital görüntülerin kalitesi birçok alanda önem taşımaktadır. Bu bağlamda, geliştirilecek model görüntü kalitesini otomatik olarak analiz edebilecektir.

## 1.2 Tezin Amacı

Müşteriden teslim alınan akıllı cihazlar teknik servise gönderilmeden önce belirli açılardan bir webcam aracılığıyla fotoğrafları çekilmektedir. Donanım yetersizliği ve kullanıcıların fotoğraf çekimindeki özensiz davranışı nedeniyle pek çok fotoğraf düşük kaliteli olmaktadır. Aynı zamanda fotoğraf içerisinde bir cihazın bulanmaması da istenilen bir durum olmamaktadır. Bu gibi durumlar, teknik servis süreçlerinde cihaz fotoğrafının kontrol aşamasını zorlaştırmaktadır ve aynı zamanda depolama alanında gereksiz bir israfa yol açmaktadır. Düşük kaliteli fotoğrafların büyük oranda sistemde saklanması hem zaman kaybına hem de maliyet artışına neden olmaktadır.

Bu sorunlara çözüm olarak, çekilen fotoğrafların kalitesini otomatik olarak tespit edebilen bir derin öğrenme modeli geliştirilmesine karar verilmiştir. Derin öğrenme modeli fotoğrafları “iyi” veya “kötü” olarak sınıflandırarak yaşanan problemleri en aza indirmek amaçlanmıştır.

Bu çalışma, teknik servis süreçlerinde akıllı cihaz fotoğraflarının düşük kalitesinden kaynaklanan yaygın sorunları hedef almaktadır. Düşük kaliteli fotoğraflar, gereksiz depolama, zaman kaybı ve maliyet artışı gibi verimsizliklere yol açmaktadır. Bu nedenle, çekilen fotoğrafları “iyi” veya “kötü” olarak otomatik olarak sınıflandırabilen bir derin öğrenme modeli geliştirilmesi hem operasyonel verimliliği artırabilir hem de hizmet kalitesini yükseltebilir. Araştırma bu yönüyle önemlidir.

### 1.3 Literatür Araştırması

Literatür incelendiğinde derin öğrenme yönteminin farklı birçok alanda uygulandığı görülmektedir.

Aslan ve Özupak (2025) beyin tümörlerinin doğru ve zamanında tespitinin başarılı bir tedavi için çok önemli olduğunu vurgulayarak çalışmalarında yapay zekâ tabanlı bir model önermişlerdir. MRG (Manyetik Rezonans Görüntüleme) taramalarından beyin tümörlerini tespit etmek için YOLOv8s-cls modelini kullanan araştırmacılar deneysel çalışmalar sonucunda modelin %98.7'lik yüksek bir başarı gösterdiğini saptamışlardır.

Budak vd. (2025) çalışmalarında baskılı devre kartı (PCB) üretim sürecinde ortaya çıkabilecek kusurların otomatik biçimde saptanarak insan kaynaklı hataların azaltılması ve üretim hızının artırılması amacıyla, görüntü işleme teknikleri ile derin öğrenme tabanlı algoritmaların bir arada kullanıldığı bir tespit ve sınıflandırma yaklaşımı önermişlerdir. Bu kapsamda, farklı kusur tiplerini içeren 1194 görüntüden oluşan açık erişimli DeepPCB veri setini kullanarak, YOLOv4 ve YOLOv8 algoritmaları ile ayrı ayrı model eğitimi gerçekleştirmiştir. Elde edilen sonuçlar karşılaştırıldığında, YOLOv8'in üretim hatlarında daha yüksek doğruluk oranı sağladığı ve hata oranını düşürdüğü belirlenmiştir.

Karakan ve Oğuz (2024) yaptıkları çalışmada bir iş yeri kamerasından alınan görüntüler ile gerçek zamanlı olarak (kask, ayakkabı, yelek gibi) kişisel koruyucu ekipmanlarının tespitini gerçekleştirmişlerdir. Bu doğrultuda iş yeri kamerasından alınan görüntülerden elde edilen veri seti YOLOv5 mimarisinde çalıştırılmıştır. 9 farklı kişisel koruyucu ekipmanın tespiti üzeri yapılan analizin sonuçlarında ortalama doğruluk 97.3%, mAP 93.8% ve recall 91.7% olarak bulunmuştur.

Kılıç vd. (2024) arařtırmalarında derin öğrenme tekniklerini kullanarak mobilya görüntülerini sınıflandırmayı amaçlamıřlardır. Bu amaç doğrultusunda Alexnet, VGGNet-19, DenseNet 201, Squeezenet1.1 ve ResNet-152 olmak üzere beř farklı evriřimli sinir ađı mimarisinden faydalanmıřlardır. Alınan sonuçlarda sırasıyla %94.15, %98.87, %98.65, %97.07 ve %98.87 başarı elde edildiđi saptanmıřtır.

Sofuođlu (2024) çalıřmasında tarım sektörü için bitki yaprak hastalıklarını doğru bir řekilde sınıflandıran yeni derin öğrenme modelleri önermeyi amaçlamıřtır. Bu amaç doğrultusunda yeni evriřimli sinir ađı mimarileri tasarlayarak üzüm ve patates yaprakları ile bitki hastalıklarının tespitine odaklanmıřtır. Gerçek veri setleri üzerinde yapılan deneylerin sonucunda, önerilen modellerde, sınıflandırma doğruluđu açısından en son modellere kıyasla ortalama olarak önemli (yaklařık yüzde 8,6) bir iyileřme sađlandıđı tespit edilmiřtir.

Yanılmaz vd. (2024) yaptıkları çalıřmada güneř enerjisi santrallerindeki, PV modüllerdeki hotspot (sıcak nokta) tespiti için YOLO algoritmalarının ne derecede etkin ve avantajlı olduđunu arařtırmıřlardır. Bu doğrultuda, gerçek güneř panellerinden elde edilen görüntülerden oluřan veri setine YOLOv5, YOLOv6, YOLOv7 ve YOLOv8 algoritmaları uygulanarak performansları kıyaslanmıř ve en iyi sonuç veren model tespit edilmiřtir. Alınan sonuçlarda YOLOv8 algoritmasının %88.7 özgüllük, %80.5 duyarlılık ve %83.8 mAP deđerleri ile diđer modellere göre daha iyi sonuçlar verdiđi saptanmıřtır.

Arabacı Hayaliođlu (2023) çalıřmasında, Aydın ilinde örnek olarak seçilen dođal ve kültürel peyzajların görsel kalite analizinin belirlenmesini amaçlamıřtır. Bu doğrultu seçilen 36 adet fotođraf 8 adet peyzaj özelliđine göre (canlılık, uyum (harmoni), büyüleyicilik, dođallık, ilginçlik, bitki varlıđı, renk çeřitliliđi, görsel tercih) deđerlendirilmek üzere 120 katılımcıya Görsel Kalite Analizi formu ile sunulmuřtur. Görsel kalite deđerlendirme yöntemi kullanılarak deđerlendirilen fotođraflardan görsel tercih puanlarının (GTP) ortalamaları incelenmiř ve her peyzaj tipinden en yüksek puanı alan fotođraf ortaya çıkarılmıřtır. Alınan sonuçlara göre; Dođal Peyzaj tipinden; Dilek Yarımadası-Büyük Menderes Deltası Milli Parkının fotođraflarından 6. fotođrafın (MP6) (GTP=5.5250), Bafa Gölü Tabiat Parkı fotođraflarından 5. fotođrafın (BGTP5) (GTP=5.8167), Kültürel Peyzaj tipinde yer alan, Kentsel Peyzaj fotođraflarından; Aydın Kentine ait 4. fotođraf (AK4) (GTP=3.3250) ve 6. fotođraf (AK6) (GTP=3.3250)'ların, Kırsal Peyzaj tipinde yer

alan Tarımsal Peyzajlara ait fotoğraflardan, 5. fotoğrafın (TP5) (GTP=4.2250), Sanayi Peyzajlarına ait fotoğraflardan, 2. fotoğrafın (SP2) (GTP=3.1917), Geleneksel Yerleşim fotoğraflarından 6. fotoğrafın (GY6) (GTP=6.0667) görsel tercih puanlarının diğer fotoğraflara göre daha yüksek olduğu saptanmıştır.

Jain ve Peddi (2023) araştırmalarında, yüz fotoğraflarından cinsiyetin belirlenmesine yönelik bir model geliştirmişlerdir. Bu amaçla, derin öğrenme tabanlı yaklaşımların biri olan ResNet-152 mimarisine dayalı bir cinsiyet sınıflandırma modeli önermişlerdir. Daha önce eğitilmiş ResNet-152 modeli, geniş bir yüz görüntüsü veri seti üzerinde yeniden ayarlanmış (fine-tuning) ve bireylerin yüz görünümüne göre kadın veya erkek olarak sınıflandırılması sağlanmıştır. Elde edilen bulgulara göre, önerilen model test veri setinde %99 doğruluk oranı ile diğer birçok güncel cinsiyet sınıflandırma modelinden daha yüksek performans göstermiştir.

Özel vd. (2023) çalışmalarında YOLO algoritmasının gastronomide tabaktaki yemek içeriğini belirlemek için kullanımını araştırmışlardır. Bununla birlikte doğru bir veri kümesi seçimi ve eğitim süreci gibi konularda bilgiler vererek YOLO'nun gelecekteki kullanım potansiyelini tartışmışlardır.

Tbaileh (2023) yaptığı çalışmada, e-ticaret dünyasının dinamik yapısında, ürün görsellerinin görsel etkisinin, tüketicilerin algıları ve tercihleri üzerinde önemli bir etkiye sahip olmasından hareket ederek müşterinin bakış açısından ürün görüntüsünü hassas bir şekilde değerlendiren yapay zekâ (YZ) destekli model önermeyi amaçlamıştır. Bu amaç doğrultusunda Farklı birçok veri setinin düzenlenmesiyle, karmaşık bir evrimsel sinir ağı (CNN) mimarisi geliştirmiştir. Alınan sonuçlarda önerilen modelin test veri setinde %98'lik dikkate değer bir doğruluk elde ettiği ve görüntüleri doğru bir şekilde sınıflandırdığı görülmüştür.

Balcıoğlu (2023) yaptığı çalışmada görüntü işleme teknikleri ve derin öğrenme algoritmaları kullanılarak toz metalürjisi ile üretilmiş olan on farklı metal parçanın hatalı olanlarını tespit edebilen, hızlı ve maliyeti düşük bir sistem geliştirmiştir. Metal somun parçalarına ait gerçek zamanlı bir sistem tarafından yakalanan görüntüler, kusurları işlemek ve bulmak için kullanılmıştır.

Akar ve Akgül (2022) çalışmalarında derin öğrenme ile yüz ifadelerinden duygu tanımayı gerçekleştirmeyi amaçlamıştır. Bundan hareketle KDEF ve PICS

veri setlerinden faydalanarak evrimsel sinir ağı ile yeni bir model geliştirmişlerdir. VGGNet temel alınarak geliştirilen modelde KDEF veri setinden %97.44, PICS veri setinden %98.24 doğruluk ile yüksek bir başarı elde edilmiştir.

Ali vd. (2021) yaptıkları çalışmada iç mekân sahnelerinin sınıflandırılmasına yönelik bir sistem geliştirmişlerdir. Çalışmada sınıflandırma işlemi için 18 katmandan oluşan ResNet-18 mimarisi kullanılmıştır. Ayrıca, transfer öğrenme yaklaşımı uygulanmış ve iki farklı yöntem olan ince ayar (fine-tuning) ve katman dondurma (freezing layers) teknikleri ile sınıflandırma performansı karşılaştırılmıştır.

Chen vd. (2021) yüz gibi büyük ölçekli değişikliklere sahip nesnelere çalışırken doğruluğun önemli ölçüde azaldığını belirtmişlerdir. Bundan yola çıkarak değişen yüz ölçeklerinin algılama sorununu çözmek amacıyla, yüz algılama performansını iyileştirmek için YOLOv3'e dayalı YOLO-face adlı bir yüz algılayıcı önermişlerdir. Mevcut yaklaşım, yüz algılama için daha uygun olan bağlantı kutularının kullanılması ve daha hassas bir regresyon kayıp fonksiyonunu içermektedir. İyileştirilmiş algılayıcının, hızlı algılama hızını korurken doğruluğu önemli ölçüde artırdığı gözlemlenmiştir. WIDER FACE ve FDDB veri kümeleri üzerinde yapılan deneyler, geliştirilen algoritmanın YOLO ve çeşitlerinden daha iyi performans gösterdiği tespit edilmiştir.

Dharneeshkar vd. (2020) Hindistan'da yol bakım sürecinin çoğunlukla manuel olarak yapılmasının çok fazla zaman aldığı ve insan gücü gerektirdiğinden hareket ederek yol çukurlarının otomatik olarak tanımlanmasını sağlayan bir yöntem önermişlerdir. Yollara dair herhangi bir çukur veri seti bulunmadığından ilk olarak 1500 görüntülük bir veri seti oluşturmuşlardır. Devamında bu veri seti YOLOv3, YOLOv2, YOLOv3-tiny üzerinde eğitilmiş ve sonuçlar karşılaştırılmıştır. Farklı çukur görüntülerinde test edilen modelin makul bir doğrulukla algılama yaptığı tespit edilmiştir.

Yıldırım ve Çağıl (2020) araştırmalarında kapı, pencere ve kış bahçesi üretimi yapan bir firmada, montaj projelerinde kullanılmakta olan aksesuarların kamera tespiti ile adedini belirlemek amacıyla görüntü işleme ve derin öğrenme tabanlı tekniklerden faydalanmışlardır. Google Colab Platformu, Python Programlama Dili, OpenCV kütüphanesi kullanılarak gerçekleştirilen çalışmada uygun sonuçlara ulaşılmıştır.

Ayyachamy vd. (2019) tıbbi görüntüleme alanında içerik tabanlı görüntü geri getirme (content- based image retrieval) üzerine bir çalışma gerçekleştirmiştir. Yazarlar, yüksek çözünürlüklü görüntü veri tabanlarından benzer patolojilerin geri getirilmesi amacıyla derin öğrenme tabanlı bir model önermiştir. Çalışmada önceden eğitilmiş ResNet-18 modeli kullanılmış ve dört farklı yaklaşım (CT- Computed Tomography, MRI- Magnetic Resonance Imaging, MG- Mammogram ve PET- Positron Emission Tomograph) içeren, yirmi üç sınıftan oluşan çoklu veri kümesi üzerinde yöntem test edilmiştir. Elde edilen sonuçlarda ortalama %92 sınıflandırma doğruluğu ve 0.90 ortalama kesinlik değeri rapor edilmiştir.

Demir vd. (2019) çalışmalarında, cilt kanserinin erken tanısına yönelik bilgisayar destekli bir sınıflandırma yöntemi geliştirmişlerdir. Bu kapsamda 2437 eğitim, 660 test ve 200 doğrulama görüntüsünden oluşan bir veri seti oluşturmuş ve görüntüleri iyi huylu ve kötü huylu olarak sınıflandırmayı amaçlamışlardır. Çalışmada sınıflandırma işlemi için ResNet-101 ve Inception-v3 derin öğrenme mimarileri kullanılmış olup, elde edilen sonuçlar doğrultusunda ResNet-101 ile %84.09, Inception-v3 ile %87.42 doğruluk oranına ulaşılmıştır.

Ghosal vd. (2019) çalışmalarında MRI verilerinden beyin tümörünün sınıflandırılması için otomatik bir araç sunmuşlardır. Bu araçta, örnek görüntüler, Evrişimli Sinir Ağı (CNN) tabanlı bir Squeeze and Excitation ResNet modeline aktarılmıştır. Önerilen CNN modelinin etkinliğini göstermek amacıyla serbest tümör veri tabanında görelî bir çalışma gerçekleştirilmiştir. Deneysel değerlendirmeler, önerilen CNN modelinin veri artırımı uygulanmadan %89.93 genel doğruluk oranına ulaştığını göstermektedir. Veri artırımı eklenmesiyle Gliomas, Meningioma, ve Pituitary tümörleri için doğruluk oranlarının sırasıyla %98.67, %91.81 ve %91.03'e yükseldiği ve genel doğruluk oranının %93.83 olduğu tespit edilmiştir. Ayrıca, bazı güncel yöntemlerle karşılaştırıldığında, duyarlılık (sensitivity) ve özgüllük (specificity) açısından da umut verici iyileşmeler olduğu gözlemlenmiştir.

Rateke vd. (2019) çalışmalarında, sürücülerin konforunu ve güvenliğini doğrudan etkileyen yol yüzeyinin kalitesini tespit etmeyi amaçlamışlardır. Bu amaçla yüzey türleri ve kalite sınıflandırması üzerine yapılacak deneyler için, düşük maliyetli bir kamera ile toplandıkları veriler ile çalışmanın veri setini oluşturmuşlardır. Veri setinde genel olarak iyi ve kötü asfalt örnekleri (çukurlar ve diğer hasarlar ile) diğer kaldırım türleri ve ayrıca birçok asfaltsız yol örneği (çukurlu

ve çukursuz) yer almıştır. Evrişimsel Sinir Ağı aracılığı ile yol türü ve kalitesi için yapılan sınıflandırmada sonuçların umut verici olduğu saptanmıştır.

Toğaçar vd. (2019) çalışmalarında zatürre hastalığının erken teşhisi için derin öğrenme modellerinden evrişimsel sinir ağı (ESA) modelini kullanmışlardır. Bu model ile akciğer görüntülerinin öznitelik çıkarımı elde edildikten sonra farklı sınıflandırıcılar kullanılarak başarılar kıyaslanmıştır. Destek vektör makinelere %95.8'lik yüksek bir başarı oranı ile sonuçlandırıldığı tespit edilmiştir.

Doğan ve Türkoğlu (2018) araştırmalarında bitki yapraklarını sınıflandırma da derin öğrenme algoritmalarının başarısını tespit etmeyi amaçlamışlardır. Bu amaç doğrultusunda 32 sınıftan ve yaklaşık 1900 görüntüden oluşan veri setini kullanarak AlexNet, Vgg16, Vgg19, ResNet18, GoogleNet gibi derin öğrenme algoritmalarını uygulamışlar. ResNet18 algoritmasının %99.15 ile en iyi sonucu verdiği sonucuna varılmıştır.

Liu vd. (2018) çalışmalarında gerçek dünya çekimlerinde gürültü, bulanıklık ve titreme gibi birçok görüntü sorunu bulunduğu hareket ederek trafik işaretlerini örnek almış YOLO ağına dayalı görüntü bozulma modelleri oluşturmuşlardır. Devamında geleneksel görüntü işleme yöntemlerini birleştirerek gerçek dünya çekimlerinde mevcut sorunları simüle etmişlerdir.

Hansen vd. (2017) çalışmalarından ambalajlarda farklı biçimlerde bulunan barkodların algılanmasını amaçlamışlardır. Bu doğrultuda derin öğrenme tabanlı nesne algılama modeli olan YOLO mimarisini hem 1D barkodlar hem de QR kodlarının hızlı ve güvenilir biçimde tespit edilebilmesi için kullanmışlardır. Geliştirilen model Muenster BarcodeDB veri seti üzerinde %99.1'lik algılama oranı ile başarı sağlamıştır.

Pekezou Fouopi vd. (2016) trafikte çarpışmayı önlemek için, otonom araçların trafik durumunu anlaması gerektiğini belirterek nesne algılamayı iyileştirmek için evrişimli sinir ağları (Fast R-CNN) ve bağlam sınıflandırıcıların (Naive Bayes, aspect ratio temelli) bir kombinasyon önermişlerdir. Önerilen model gerçek trafik verileri üzerinde eğitilmiş ve değerlendirilmiştir.

Krizhevsky vd. (2012) yaptıkları çalışmada, ImageNet LSVRC-2010 yarışmasındaki 1.2 milyon yüksek çözünürlüklü görüntüyü 1000 farklı sınıfa sınıflandırmak için bir evrişimli sinir ağını eğitmişlerdir. ImageNet, yaklaşık 22000

kategoriye ait 15 milyondan fazla etiketlenmiş yüksek çözünürlüklü görüntüden oluşan bir veri kümesidir. Görüntüler Web'den toplanmış ve Amazon'un Mechanical Turk kitle kaynak aracı kullanılarak insan etiketleyiciler tarafından etiketlenmiştir, arařtırmacılar deneylerin çoğunu bu sürüm üzerinde gerçekleřtirmiřtir. Alınan sonuçlarda Top-1 ve Top- 5 için sırasıyla %37.5 ve %17'lik hata oranlarına ulařmıřlardır. Bu sonucun önceki teknolojidenden önemli ölçüde daha iyi olduđu belirtilmiřtir.



## 2. YAPAY ZEKÂ

Yapay zekâ (Artificial Intelligence - AI), akıllı makineler, özellikle akıllı bilgisayar programları üretme bilimi ve mühendisliğidir. Bu teknoloji, bilgisayarların insan zekâsını anlayabilmesi amacıyla kullanılan çalışmalarla ilişkili olsa da yalnızca biyolojik süreçlerle sınırlı değildir (McCarthy, 2007). Yapay zekâ, alanının en dikkat çekici yönlerinden biri, konusunun ne olduğuna dair kesin bir yanıt olmaması olarak gösterilebilir. Bu sorun temelde iki boyutludur, çünkü yapay olanın doğasını yeterince kavramak, ancak zekâ kavramını anlamakla mümkün olabilir. Yapay zekânın “yapay” olan yanı, doğal (özellikle biyolojik veya evrimsel) etkilerin bir sonucu olarak değil, insan zekâsı ve yaratıcılığının bir ürünü olarak ortaya çıkmasıyla ilgilidir. Diğer bir deyişle, yapay zekâyâ sahip sistemler, doğal zekâyâ sahip olanlardan farklıdır. Çünkü bunlar, normalde zekâ ile ilişkilendirilmeyen nesnelerin, özel bir süreç sonucunda (tasarlanarak, üretilerek veya yaratılarak) zekâ özelliği kazanmış hâlidir. (Fetzer, 1990).

Genel olarak yapay zekâ kavramı bir bilgisayarın veya bilgisayar kontrolündeki bir makinenin, akıl yürütme, anlamlandırma, genelleme yapma ve geçmiş deneyimlerden öğrenme gibi çoğunlukla insana özgü kabul edilen ileri düzey bilişsel süreçleri gerçekleştirebilme yetisi şeklinde tanımlanmaktadır (Nabiyev, 2012: 25). Yapay zekâ ile ilgili yapılan diğer farklı tanımlamalar ise şu şekildedir: “Slage’ye göre yapay zekâ; sezgisel programlama temelinde olan bir yaklaşımdır. Genesereth ve Nilsson’a göre yapay zekâ, akıllı davranış üzerine bir çalışmadır. Ana hedefi, doğadaki varlıkların akıllı davranışlarını yapay olarak üretmeyi amaçlayan bir kuramın oluşturulmasıdır. Tesler’e göre ise yapay zekâ; şu ana kadar yapılamayanlardır. Axe’e göre ise yapay zekâ; akıllı programları hedefleyen bir bilimdir.” (Akt. Nabiyev, 2012: 25).

Yapay zekâ kavramının tarihsel gelişimine ve temellerine bakıldığında, II. Dünya Savaşı sonrasında birçok araştırmacının birbirinden bağımsız olarak akıllı makineler üzerine çalışmaya başladığı görülmektedir. İngiliz matematikçi Alan Turing, bu alandaki öncü isimlerden biri olarak değerlendirilmektedir. Turing, 1947

yılında yapay zekâ konulu bir konferans vermiş ve yapay zekâ arařtırmalarının, makineler inşa etmek yerine bilgisayarları programlayarak yürütülmesinin daha uygun olacağı düşüncesini ortaya koymuřtur. 1950’li yılların sonlarına gelindiğinde ise, sayıları giderek artan arařtırmacılar yapay zekâ alanında çalışmalar yürütmekte olup, bunların büyük çoğunluğu bilgisayar programlamaya dayalı yöntemleri benimsemiřtir (Fetzer, 1990).

Farklı disiplinlerde ve sektörlerde geniş bir uygulama yelpazesine sahip olan yapay zekânın uygulama alanlarından bazıları řu řekilde açıklanmıřtır (Fetzer, 1990: 8):

**Oyun oynama (Game playing):** Satranç gibi oyunlarda, insanlara karřı ustalık seviyesinde oynayabilen, genellikle yüksek hızda çok sayıda olasılığı deęerlendiren yapay zekâ sistemleridir.

**Konuřma tanıma (Speech recognition):** İnsan konuřmasını algılayıp sayısal verilere dönüřtürerek sınırlı görevlerde pratik kullanım sunan yapay zekâ teknolojisidir.

**Doęal dil anlama (Understanding natural language):** Bilgisayarların yalnızca kelimeleri çözümlemenin ötesine geçerek, metnin bağlamını ve konu alanını anlamasını saęlayan yapay zekâ yaklařımıdır.

**Bilgisayarla görme (Computer vision):** İki boyutlu görüntülerden üç boyutlu nesne ve sahne bilgisi çıkararak görsel veriyi anlamlandıran yapay zekâ alanıdır.

**Uzman sistemler (Expert systems):** Belirli bir alandaki uzman bilgisini kurallara ve çıkarım mekanizmalarına dönüřtürerek karar verme veya problem çözme süreçlerini otomatikleřtiren yapay zekâ programlarıdır.

**Sezgisel sınıflandırma (Heuristic classification):** Belirli kriterlere göre verileri önceden tanımlanmıř kategorilere ayıran ve çeřitli bilgi kaynaklarını bir arada kullanan yapay zekâ yöntemidir.

Anlařıldığı üzere, yapay zekâ kavramsal olarak farklı tanımlamalara konu olmakla birlikte, tarihsel süreçteki gelişimi ve genişleyen uygulama alanlarıyla günümüzde hem akademik hem de endüstriyel bağlamda stratejik öneme sahip, disiplinlerarası bir arařtırma ve teknoloji alanı olarak deęerlendirilebilir.

### 3. MAKİNE ÖĞRENMESİ

Makine öğrenimi, çevredeki ortamdan öğrenerek insan zekâsını taklit etmek üzere tasarlanmış hesaplama algoritmalarının gelişen bir dalıdır (El Naqa and Murphy, 2015). Bununla birlikte araştırmacıların sürekli ilgisini çeken ve bu ilgi sonucunda literatüre pek çok yeni yöntemin kazandırıldığı yapay zekâ alt alanlarından biridir. Herbert Simon'a göre "öğrenme bir sistemin deneyime dayalı olarak başarımını artırdığı bir süreçtir" biçimindeki tanımı esas alınarak, makine öğrenmesi belirli bir görevde dış müdahale olmaksızın, deneyimden elde edilen bilgilerle performansını geliştiren algoritmaların tasarlanması olarak tanımlanabilir. Makine öğrenmesi yaklaşımları, öğrenme sürecinde sistemin erişebileceği "veri" veya "geri bildirim"nin niteliğine göre üç ana kategori altında sınıflandırılmaktadır (Efe ve Cangır, 2022).

#### 3.1 Denetimli Öğrenme

Makine öğrenmesinin temel yaklaşımlarından biri olarak, sınıflandırma ve tahmin gibi görevlerde algoritmaların etiketlenmiş veri kümeleriyle eğitilmesine dayanmaktadır. Bu yöntemde, modelin zaman içerisinde hedeflenen çıktıları üretebilmesi için girdiler ile bunlara karşılık gelen doğru çıktılarından oluşan bir eğitim seti kullanılmaktadır. Algoritmanın performansı, genellikle bir kayıp fonksiyonu aracılığıyla değerlendirilmekte ve hatayı minimize etmek amacıyla model parametreleri iteratif bir biçimde güncellenmektedir. Denetimli öğrenme süreci, algoritmanın doğru ve tutarlı bir şekilde öğrenebilmesi için iyi tanımlanmış ve etiketlenmiş giriş verileriyle beslenmesini gerektirir. Etiketlenmiş veri setleri, hem verinin özelliklerini temsil eden giriş niteliklerini hem de algoritmanın tahmin etmeyi amaçladığı hedef çıktıları içermektedir. Bu yapı, modelin doğruluk oranını artırmak ve genelleme kabiliyetini güçlendirmek açısından kritik bir rol oynamaktadır. Kaliteli ve doğru etiketlenmiş veriler, modelin eğitim aşamasında daha güvenilir sonuçlar üretmesine ve gerçek dünya uygulamalarında tutarlı performans sergilemesine katkıda bulunmaktadır (Sinap, 2024).

### 3.2 Denetimsiz Öğrenme

Denetimsiz öğrenmede, kullanıcı sisteme doğrudan bir müdahalede bulunmaz, yalnızca ham girdi verileri modele sunulur ve bu veriler üzerinde herhangi bir etiketleme işlemi gerçekleştirilmez. Model, bu verilerden yola çıkarak kendi başına desenleri keşfetmeye, veri yapısını analiz etmeye ve değişkenler arasındaki ilişkileri ortaya çıkarmaya çalışmaktadır (Kızılkaya ve Oğuzlar, 2018: 92)

### 3.3 Pekiştirmeli Öğrenme

Pekiştirmeli öğrenmenin temel prensibi, gerçekleşen bir eylemin sonuçlarını ödül veya ceza mantığı çerçevesinde değerlendirerek, hedeflenen sonuca ulaşmayı sağlayacak bir öğrenme süreci oluşturmaktır. Bu yaklaşımda, durum vektörünün ve kontrol sinyalinin alabileceği tüm olası değerler için bir tablo oluşturulur; her bir kombinasyona karşılık gelen ödül değerleri bu tabloda yer alır. Ardından, istenen davranışı ortaya çıkaracak kontrol sinyali, en yüksek ödül değerine sahip durumlar arasından seçilir. Pekiştirmeli öğrenme hem modele dayalı hem de modelden bağımsız yöntemlerle uygulanabilmektedir (İmamoğlu vd., 2009).

Günümüzde makine öğrenmesinin geniş ölçekte benimsenmesinde en belirleyici etken, bilgisayar teknolojilerindeki kayda değer gelişmelerdir. Öyle ki bu teknolojik ilerlemeler sayesinde, ihtiyaç duyulan verilerin yüksek kapasitede ve güvenli bir şekilde depolanması, hızlı biçimde erişilmesi ve analiz süreçlerinde etkin olarak kullanılması mümkün hâle gelmiştir (Kızılkaya ve Oğuzlar, 2018: 92). Böylelikle makine öğrenimine dayalı teknikler, örüntü tanıma, bilgisayar görüşü, uzay aracı mühendisliği (spacecraft engineering), finans, eğlence ve hesaplamalı biyoloji gibi çeşitli alanlardan biyomedikal ve tıbbi uygulamalara kadar geniş bir yelpazede başarıyla uygulanmaktadır (El Naqa and Murphy, 2015).

Bu kapsamda makine öğrenmesinin, veri zenginliği ile teknolojik ilerlemelerin kesişim noktasında konumlanarak, disiplinler arası etkileşimi güçlendiren, bilimsel ve endüstriyel alanlarda dönüşümü hızlandıran önemli bir teknoloji hâline geldiği söylenebilir.

## 4. DERİN ÖĞRENME

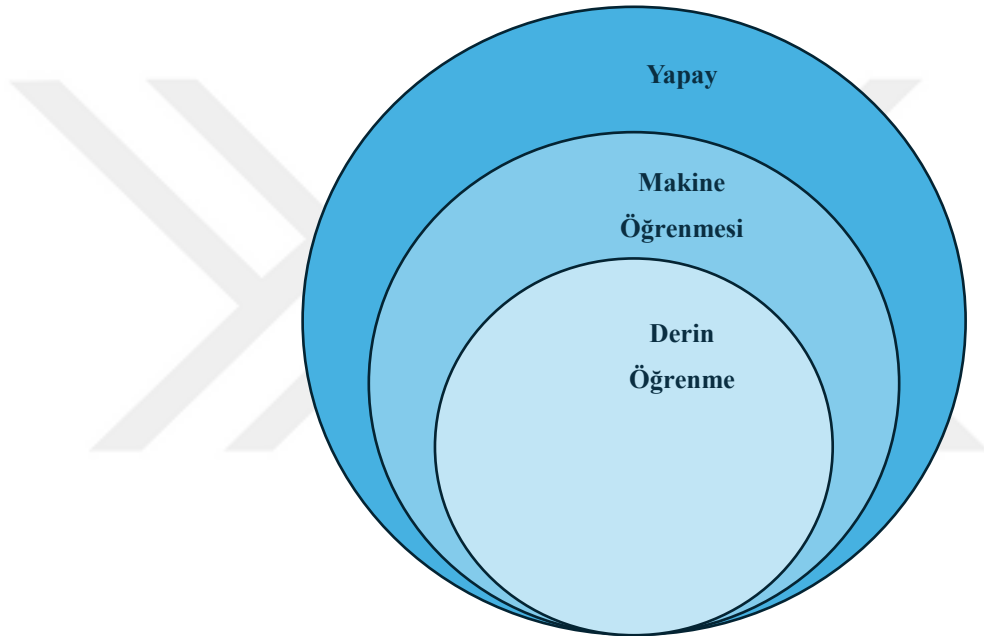
### 4.1 Tarihçe

Derin öğrenmenin temellerini, 1943 yılında Pitts ve McCulloch tarafından geliştirilen McCulloch-Pitts nöronları olarak ifade edilen matematiksel bir sinir ağı oluşturmuştur. 1950'li yıllarda Turing ve Samuel'in çalışmaları makine öğreniminin altyapısını oluştururken, 1959'da Hubel ve Wiesel'in basit ve karmaşık hücre yapılarını keşfederek doğrudan derin öğrenmeye olmasa dahi yapay sinir ağlarının gelişimine katkı sağlamışlardır. 1965'te Ivakhnenko ve Lapa, çok değişkenli veri kümelerini temel alarak bilgisayar destekli matematiksel bir model geliştirmiş ve bu modeli sinir ağı yapısına uyarlayarak, derin öğrenme alanında çalışan ilk ağlardan birini ortaya koymuşlardır. 1980 yıllara gelindiğinde Fukushima tarafından geliştirilen ve Neocognitron olarak adlandırılan yapay sinir ağı modeli, el yazısı tanıma ve diğer desen tanıma problemlerine yönelik olarak tasarlanmış hiyerarşik ve çok katmanlı bir sinir ağına sahiptir. Bu model, günümüzde özellikle görüntü işleme alanında yaygın olarak kullanılan konvolüsyonel sinir ağlarının (CNN) öncüsü olarak kabul edilmektedir. 1997 yılında Hochreiter ve Schmidhuber, uzun vadeli bağımlılık sorunlarını ele almak amacıyla Uzun Kısa Süreli Hafıza (LSTM) yapısını, bir tür tekrarlayan sinir ağı olarak ortaya koymuşlardır. 2000'li yılların ortalarında Hinton ve Salakhutdinov'un çalışmalarıyla birlikte "Derin Öğrenme" kavramı giderek daha fazla ilgi görmeye başlamıştır. Bununla birlikte çok katmanlı bir sinir ağının, bir defada bir katmanının nasıl önceden eğitilebileceği gösterilmiştir. 2014 yılında yüz tanıma sistemi olan DeepFace geliştirilmiş, Facebook, kullanıcılarını fotoğraflarda otomatik olarak tanıyıp etiketlemek amacıyla bu derin öğrenme tabanlı bir yüz tanıma teknolojisini kullanmaya başlamıştır. Derin öğrenme ve yapay sinir ağı yaklaşımları, günümüzde tıptan robotik sistemlere, nesne algılama, görüntü ve ses verilerinin işlenmesi, öngörüselsel analizler, endüstriyel uygulamalar ve finansal modellemelere kadar çok çeşitli alanlarda başarıyla uygulanmaktadır (Akın ve Şahin: 2024; Kayaalp ve Süzen: 2018).

Tüm bu tarihsel gelişmeler incelendiğinde, derin öğrenme ve yapay sinir ağı teknolojilerinin, disiplinler arası katkılarla evrilerek günümüzde son derece geniş bir uygulama yelpazesine ulaştığı görülmektedir.

## 4.2 Tanım

Derin öğrenme, birbirlerini izleyen katmanlarda veriler işlenirken zamanla artan biçimde daha kullanışlı gösterimler kazandırabilen makine öğrenmesinin ve yapay zekânın bir alt kümesi olarak ifade edilebilir.

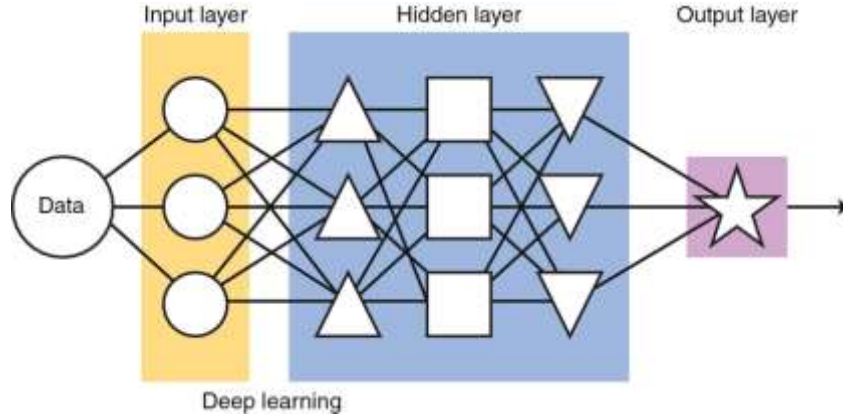


**Şekil 4.1: Yapay Zekâ, Makine Öğrenmesi ve Derin Öğrenme Arasındaki İlişki**

Şekil 4.1’de yapay zekâ, makine öğrenmesi ve derin öğrenme arasındaki ilişki gösterilmiştir. Burada derin öğrenmenin makine öğrenmesinin, makine öğrenmesinin ise yapay zekânın bir alt alanı olduğunu gösterilmektedir.

Derin öğrenme kavramında geçen derin kelimesi, derin öğrenmenin derin bilgiler elde etmesinden ziyade birbirini izleyen katmanları ifade etmektedir. Modelde yer alan katman sayısı ise modelin derinliğini meydana getirmektedir. Modern derin öğrenme modelleri, onlarca hatta yüzlerce katmandan oluşur ki bu katmanlar sinir ağı şeklini almaktadır. Söz konusu sinir ağları her ne kadar insan beyninden etkilenerek nörobiyolojiye atıf yapsa da insan beynine ait bir model değil

veriden faydalı fayda sağlayacak gösterimler öğrenebilen matematiksel bir araçtır (Chollet, 2021: 9).



**Şekil 4.2: Derin Öğrenme Sinir Ağı Yapısı**

**Kaynak:** Okagawa, Y., Abe, S., Yamada, M., Oda, I., & Saito, Y. (2022). Artificial intelligence in endoscopy. *Digestive Diseases and Sciences*, 67(5), 1553-1572.

Yukarıda yer alan Şekil 4.2’de derin öğrenmede kullanılan yapay sinir ağlarının temel yapısını gösterilmektedir. Veri önce girdi katmanına alınır, ardından birden fazla gizli katmandan geçerek işlenir ve sonunda çıktı katmanında tahmin ya da sınıflandırma sonucu üretilir. Bu yapı, derin öğrenmenin çok katmanlı mimarisi sayesinde verilerden karmaşık ilişkilerin öğrenilmesini sağlamaktadır.

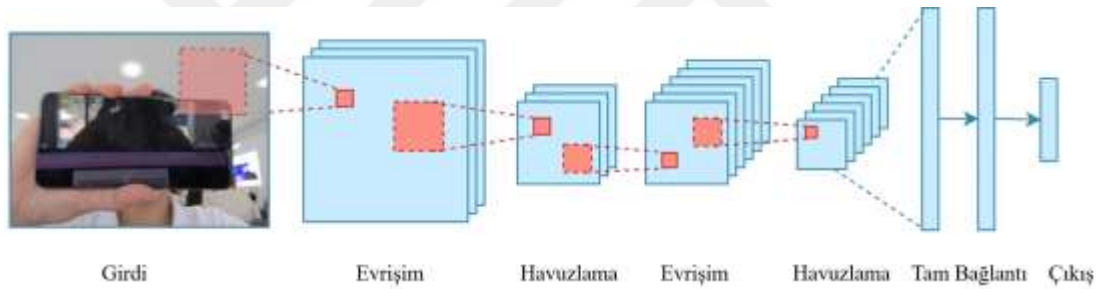
Son yıllarda bilgisayar bilimciler tarafından yoğun bir şekilde kullanılan derin öğrenmenin popüler bir çalışma alanına dönüştüğü görülmektedir. Özellikle derin öğrenme yöntemlerinden biri olan evrişimsel sinir ağları (convolutional neural networks) görüntü sınıflama, nesne tanıma ve tespit yöntemi olarak sıkça tercih edilmektedir. Bu yöntemleri dikkat çekici kılan sınıflama doğruluk oranlarının yüksek olmasıdır (Arı ve Hanbay, 2019: 1398).

### 4.3 Evrişimsel Sinir Ağları

Evrişimsel sinir ağları, verilerin satır ve sütunlar halinde düzenlenmiş, yani ızgara benzeri bir yapıya sahip olduğu durumlarda bu yapıyı işlemek için kullanılan özel bir sinir ağı türüdür. Örnekler, düzenli zaman aralıklarında örnekler alan bir boyutlu ızgara olarak düşünülebilecek zaman dizisi verisini ve iki boyutlu bir piksel ızgarası olarak düşünülebilecek görüntü verisini içermektedir. Evrişimsel ağların pratik uygulamalarda önemli derecede başarılı olduğu bilinmektedir. “Evrişimsel sinir ağı” adı ağıın, evrişim (convolution) adı verilen bir matematiksel işlemi

kullandığını göstermektedir. Evrişim, özel bir tür doğrusal işlemdir. Evrişimsel ağlar, basitçe, katmanlarından en az birinde genel matris çarpımı yerine evrişim kullanan sinir ağlarıdır (Goodfellow vd., 2018:331).

Bir başka tanımda, evrişimsel sinir ağları girdi verisi olarak görüntüleri kullanan bir derin öğrenme algoritması olarak ifade edilmiştir. Bu ağ, görüntülerin özelliklerini (features) kullanarak görüntüleri birbirinden ayırt etmektedir diğer bir deyişle sınıflandırma işlemini gerçekleştirmektedir. Evrişimsel sinir ağlarında (ESA) kullanılacak veriler için gerekli olan ön işleme, diğer sınıflandırma algoritmalarına kıyasla çok daha düşüktür. Bunun sebebi ise ESA'nın diğer sınıflandırma algoritmalarından farklı olarak filtreleri öğrenme yeteneğine sahip olmasıdır. ESA'nın mimarisi temel olarak üç katmandan oluşur. Evrişimli Katman (Convolutional Layer), Havuzlama Katmanı (Pooling Layer) ve Tam Bağlantılı Katman (Fully Connected Layer). Bu katmanlardan geçen görsel, farklı işlemlerden geçirilerek derin öğrenme modeline girecek hale gelmektedir (Sayracı, 2021).



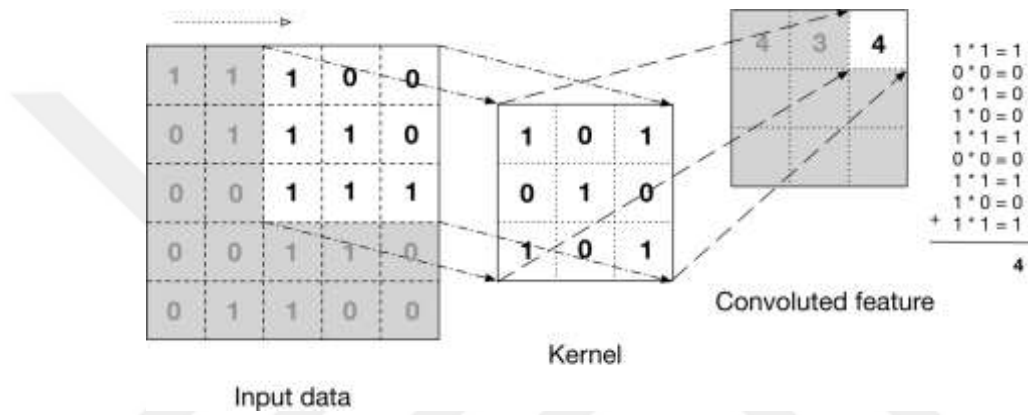
**Şekil 4.3: Evrişimsel Sinir Ağı (CNN) Mimarisi Akış Diyagramı**

Şekil 4.3'te bir girdi görüntüsünden evrişim ve havuzlama katmanları aracılığıyla hiyerarşik olarak özniteliklerin çıkarıldığı ve bu özniteliklerin tam bağlantılı bir katman vasıtasıyla sınıflandırılarak bir çıktıya dönüştürüldüğü temel bir evrişimli sinir ağı mimarisini göstermektedir.

Evrişimsel sinir ağları çokça filtreleme ve öznitelik çıkarım işleminde bulunduğundan görüntüde yer alan farklı tipteki özelliklerin belirlenmesine ve sınıflandırılmasına imkân vermektedir. Görselde yer alan nesnelere hangi özellikleri barındırdığı (insan, hayvan, bitki, araç ve benzeri) kolaylıkla belirlenebilmektedir (Akın ve Şahin, 2024).

### 4.3.1 Evrişimli katman (Convolutional layer)

Evrişim katmanı, filtre ile görüntü arasındaki evrişim sürecini gerçekleştirerek görüntüde yer alan dokuyu veya deseni keşfetmeyi sağlar. ESA'nın yapı taşı olan bu katmana dönüşüm katmanı da denilebilmektedir. Dönüşüm işlemi belirli bir filtrenin bütün görüntüde gezinmesi üzerine kurulmaktadır. Bu filtreler 2x2, 3x3 veya 5x5 gibi boyutlarda olabilmekte ve daha önceki katmandan gelen görüntülere konvolüsyon işlemini yaparak çıkış verisini meydana getirmektedir. Uygulanan bu konvolüsyon işlemi sonucunda aktivasyon haritası belirlenmektedir (Çekiç ve Çavdar, 2023: 158).

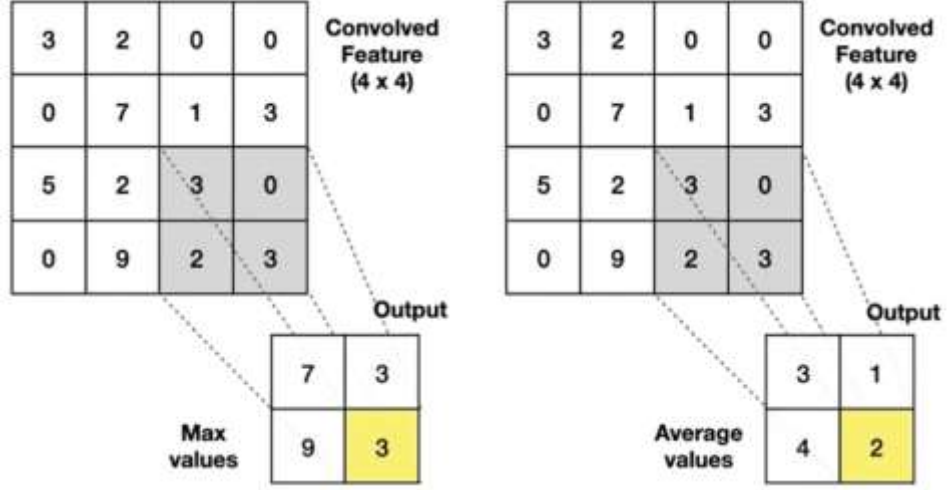


Şekil 4.4: Evrişimli Katman İşlemi

**Kaynak:** Patterson, J., & Gibson, A. (2017). *Deep learning: A practitioner's approach*. " O'Reilly Media, Inc."

### 4.3.2 Havuzlama katmanı (Pooling layer)

Evrişimsel sinir ağlarında havuzlama işlemi, parametre sayısını azaltmak ve aşırı uyumun önüne geçmek için uygulanmaktadır. Uygulamada iyi sonuçlar verdiği için en yaygın kullanılan havuzlama türlerinden biri olan Maksimum (Max) işlemi, bir katmanın 4x4 matrislik bir kısmındaki en yüksek değeri alırken; bir diğer popüler yöntem olan Ortalama (Avg) işlemi ise aynı 4x4'lük alandaki tüm değerlerin ortalamasını bularak çıktı üretmektedir. (Yılmaz ve Kaya, 2021: 19).

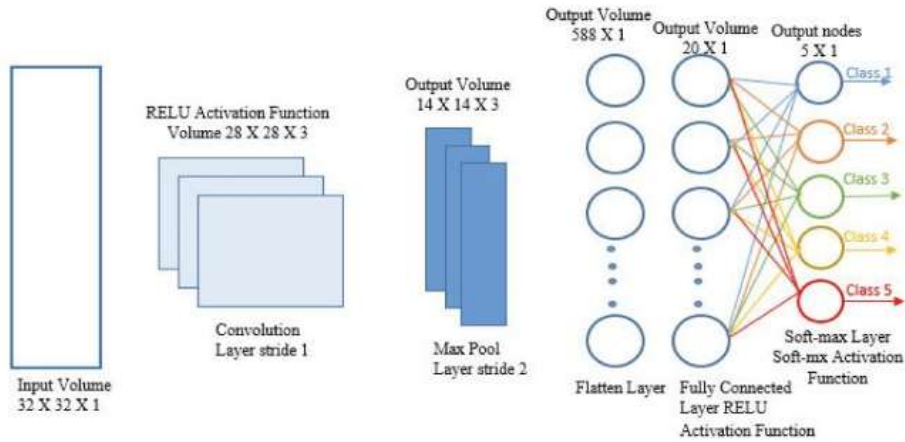


**Şekil 4.5: Maksimum Havuzlama ve Ortalama Havuzlamanın Gösterimi**

**Kaynak:** Dobilas, S. (2022). Convolutional Neural Networks Explained – How to Successfully Classify Images in Python. Towards Data Science.

### 4.3.3 Tam bağlantılı katman (Fully connected layer)

Bu üçüncü katman, genellikle evrişimli çıktı katmanı olarak adlandırılan tam bağlantılı katmandır. Bu katman genellikle ağın en alt katmanında yer almaktadır. Son havuzlama veya evrişimli çıktı katmanından girdi alır ve sonraki katmana gönderilmeden önce düzleştirilmektedir. Çıktının eşit dağılımı, son havuzlama veya evrişim katmanından sonra elde edilen sonucun tüm değerlerinin bir vektöre (3D matris) açılması anlamına gelmektedir. Bu yöntem, evrişimli çıktı katmanı tarafından temsil edilen bir özelliğin yüksek düzeyli doğrusal olmayan kombinasyonlarını incelemek için kullanılan basit bir tekniktir (Purwono vd., 2022: 742).



**Şekil 4.6: Tam Bağlantılı Katman**

**Kaynak:** Bhatt, D., Patel, C., Talsania, H., Patel, J., Vaghela, R., Pandya, S., ... & Ghayvat, H. (2021). CNN variants for computer vision: History, architecture, application, challenges and future scope. Electronics, 10(20), 2470.

#### 4.4 YOLO (You Only Look Once)

YOLO, derin öğrenme tabanlı bir nesne algılama algoritmasıdır. Geleneksel nesne algılama yaklaşımlarından ayrılarak, nesnelerin hem tespitini hem de sınıflandırmasını tek bir derin sinir ağı modeli üzerinden gerçekleştirmektedir. Bu yapısı sayesinde, hız gerektiren gerçek zamanlı uygulamalarda etkin bir şekilde kullanılabilir (Özel vd., 2023:32). YOLO algoritması, regresyon temelli bir yaklaşımı benimsemiş olup, görüntüde sadece belirli bölgelerin seçilmesi yerine, tek bir işlemde tüm görüntü için nesne sınıflarını ve bunlara ait sınırlayıcı kutuları öngörmektedir. Bu algoritmanın çalışma mantığını kavrayabilmek için, öncelikle neyin tahmin edildiğinin netleştirilmesi gerekir. Nesne tespitinde temel amaç, nesnenin ait olduğu sınıfı ve konumunu tanımlayan sınırlayıcı kutuyu doğru bir şekilde belirlemektir (Altay ve Yılmaz, 2023:70).

YOLO algoritması, nesne tespiti sürecinde sınırlayıcı kutular kullanmaktadır. Bir sınırlayıcı kutu içerisinde nesne bulunduğunda, öncelikle kutunun orta noktası belirlenir. Bu kutu, nesnenin boyut bilgilerini, konum koordinatlarını ve güven skorunu içermektedir. Söz konusu değerler, vektör formunda temsil edilmekte olup her sınırlayıcı kutu için ayrı bir tahmin vektörü oluşturulmaktadır. Denklem 4.1’de, bu tahmin sürecinde kullanılan vektör yapısı verilmiştir (Orhan ve Arıcı: 2024: 694).

$$y^T = [pc, bx, by, bw, bh, c] \quad (4.1)$$

*pc*: Nesne tespit edildiğinde 1 edilmediğinde 0 değerini alır.

*bx*: Nesnenin orta noktasının x koordinatını temsil etmektedir.

*by*: Nesnenin orta noktasının y koordinatını temsil etmektedir.

*bw*: Nesnenin genişliğini ifade etmektedir.

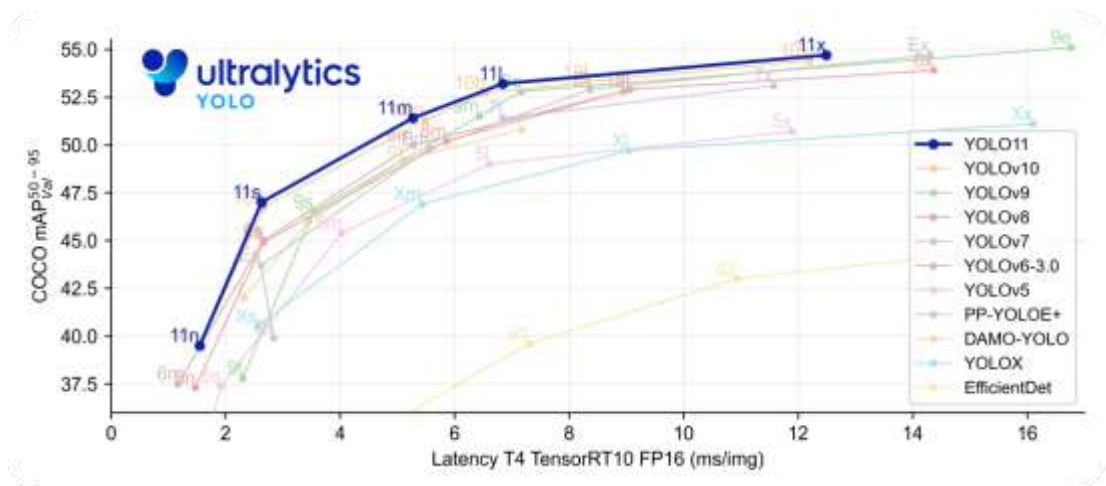
*bh*: Nesnenin yüksekliğini ifade etmektedir.

*c*: Bağlı skoru ifade etmektedir. Sınıf sayısı ne kadar ise o kadar sınıf değeri bulundurur. Örneğin, iki sınıflı bir algoritma için araba:1, yaya:0 olacak biçimde iki değer alır.

YOLO modelleri açık kaynaklıdır ve bu alanda yaygın olarak kullanılmaktadır. Bu modeller 2015 yılından beri bir sürümden diğerine geliştirilerek daha iyi doğruluk, performans ve ek yetenekler sağlayarak günümüze kadar gelmiştir. 2015 yılında YOLOv1 orijinal YOLO modeli olarak ortaya çıkmıştır, 2016 yılında daha hızlı işleme, toplu normalleştirme, bağlantı kutuları ve çok ölçekli

tahminler sunarak YOLOv2/YOLO9000 ortaya çıkmıştır. 2018 yılında YOLO Darknet-53 mimarisi ile doğruluk yeniden tanımlanmıştır. 2020 yılında YOLOv4 ile CSPDarknet53'ü temel alarak önceki versiyona göre doğruluk ve hız açısından iyileştirilmiştir. Aynı yıl YOLOv5 ile hafif ve verimli nesne algılama modeli, geliştirilmiş performans ve daha küçük model boyutları ile tanıtılmıştır. 2021 yılında ankrajsız (Anchor-free) dedektör özelliğine sahip YOLOX, 2022 yılında daha hızlı ve daha verimli gerçek zamanlı nesne algılama özelliğine sahip YOLOv6 ve gelişmiş model ölçeklendirme ve iyileştirilmiş tasarımına sahip YOLOv7 ortaya çıkmıştır. 2023 yılında dinamik ankrajsız algılama ile yeniden tasarlanmış mimari yapısıyla YOLOv8 tanıtılmış ve 2024 yılında geldiğinde dönüşümcü tabanlı özellik çıkarma ve çok ölçekli algılama yapısıyla YOLOv9, AI uygulamaları için niceleme farkındalıklı eğitim ve donanım dostu tasarımıyla YOLOv10 ve son olarak hibrit CNN- dönüştürücü modelleri ile YOLOv11 geliştirilmiş ve kullanıcılara sunulmuştur (Boesch, 2024).

Şekil 4.7’de farklı YOLO sürümlerine ait COCO veri seti üzerindeki doğruluk (mAP – mean Average Precision) ve işlem süresi (Latency) performansları karşılaştırılmıştır. Şekil 7 incelendiğinde YOLOv11x modelinin YOLOv9e modeline göre yaklaşık olarak aynı doğrulukta olmasına karşılık daha düşük işlem süresine sahip olduğu görülmektedir. YOLOv11n modelinin ise diğer modellere göre daha düşük işlem süresi olduğu görülmekte buna karşılık benzer diğer modellere kıyasla daha yüksek doğrulukta olduğu görülmektedir.



**Şekil 4.7: YOLO Sürümlerinin Performans Karşılaştırması**

**Kaynak:** Ultralytics YOLO Docs. (2025). Ultralytics YOLO11 Erişim Adresi (20.03.2025): <https://docs.ultralytics.com/tr/models/yolo11/>

#### 4.4.1 YOLOv11

2024 yılında en son YOLO sürümü olarak tanıtılan YOLOv11 gelişmiş mekanizmaları ile daha verimli bir mimari sunmaktadır. Bu sürüm YOLO'nun bilinen gerçek zamanlı çıkarım hızını korurken, küçük nesnelerin algılanmasını ve doğruluğu artırmak için tasarlanmıştır. YOLOv11'in mimarisi, YOLOv8, YOLOv9 ve YOLOv10 gibi önceki YOLO sürümlerinde tanıtılan gelişmelerden yararlanarak hem hızı hem de doğruluğu optimize etmek üzere tasarlanmıştır. YOLOv11'deki başlıca mimari yenilikler, C3K2 bloğu, SPFF (Spatial Pyramid Pooling Fast) modülü ve C2PSA bloğu etrafında şekillenmektedir. Bu bileşenlerin tümü, yüksek hızlı çıkarım özelliğini korurken uzaysal bilgileri işleme yeteneğini geliştirmektedir (Nikhileswara Rao, 2024).

Ultralytics CEO'su ve kurucusu Glenn Jocher bu sürüm için "YOLOv11 ile, gerçek dünya uygulamaları için hem güç hem de pratiklik sunan bir model geliştirmeye karar verdik. Artan doğruluk ve verimliliği sayesinde, farklı sektörlerin karşılaştığı özel sorunlara uyarlanmış çok yönlü bir araçtır." şeklinde açıklamada bulunmuştur (Boesch, 2024).

Jocher'ın ifadelerinden anlaşıldığı üzere YOLOv11'in geliştirilmesiyle amaçlanan, yalnızca nesne tespitinde yüksek doğruluk ve işlem verimliliği sağlamak değil, aynı zamanda bu yetenekleri farklı sektörlerde karşılaşılan özgün problemlere uyarlanabilir bir sistem haline getirmektir.

Modelin esnekliği, onu otonom sürüşten tıbbi görüntülemeye kadar hem hassasiyetin hem de hızlı çıkarımın çok önemli olduğu çeşitli gerçek dünya uygulamaları için ideal bir aday haline getirmektedir. YOLOv11'in çok ölçekli algılama ve uzaysal dikkat alanındaki gelişmeleri, karmaşık nesne yapılarına sahip ortamlarda üstün performans sergilemesini sağlarken, aynı zamanda kendine özgü hızlı çıkarım yeteneklerini de korumaktadır. Genel olarak, YOLOv11'in sadece bir yineleme değil, bir sıçrama niteliğinde olduğunu ve günümüzde mevcut en verimli ve çok yönlü nesne algılama modellerinden biri haline geldiği söylenebilir. Nitekim gerçek zamanlı algılama görevleri üzerinde çalışan araştırmacılar için YOLOv11, hız ve doğruluk arasında her zaman önemli olan dengeyi sağlayan pratik bir çözüm sunacaktır (Nikhileswara Rao, 2024).

Özetle YOLOv11 önceki sürümlerin güçlü yönlerini yenilikçi mimari unsurlarla birleştirmiştir. Bu yönüyle hem akademik araştırmalarda hem de endüstriyel uygulamalarda kayda değer bir doğruluk, hız ve uyarlanabilirlik açısından günümüz gereksinimlerini karşılayan bir nesne tespit modeli olarak öne çıkmıştır.

## 4.5 ResNet

Derin evrimsel sinir ağları, görüntü sınıflandırma alanında birçok atılım gerçekleştirmiştir. Ancak derin öğrenmenin gelişim sürecinde, daha derin ağların yakınsaması ele alındığında bozulma problemi ortaya çıkmaktadır. Yani, sinir ağının derinliği arttıkça doğruluk oranı önce yükselmekte, ardından doygunluğa ulaşmaktadır. Derinlik daha da artırıldığında ise doğruluk azalmaktadır. Hem eğitim hem de test kümelerinde hataların artması, bu etkinin aşırı uyumdan (overfitting) kaynaklanmadığını göstermektedir. ResNet, katmanlar arası özellik birleşimini (cross-layer feature fusion) kullanarak ağın özellik çıkarma kapasitesini artıran ve böylelikle ağ performansını, derinlik arttıkça kademeli olarak geliştiren bir yöntemdir (Liang, 2020). ResNet (artık sinir ağı), Microsoft Araştırma Enstitüsü'nün Kaiming He ekibi tarafından önerilen ve Çince'de artık ağ (residual network) olarak adlandırılan bir modeldir. ResNet tasarımının temel amacı, sinir ağının bozulma sorununu çözmektir, yani sinir ağı ne kadar derin olursa, eğitim hata oranı o kadar yüksek olur. Bu sorunu çözmek için ekip, bir kalıntı yapısı önermiştir. Ağ katmanının işlevi, her katmanın girdisinin kalıntı işlevi (residual function) olarak yeniden programlanmaktadır. Matematiksel istatistikte, kalıntı kavramı, gerçek gözlem değeri ile tahmin edilen değer (uyum değeri) arasındaki farktır (Zhang vd., 2018).

ResNet, farklı veri setleri, hesaplama kapasiteleri ve uygulama ihtiyaçlarına uyum sağlamak amacıyla zaman içinde çeşitli versiyonlara ayrıldığı görülmektedir. Aşağıda, ResNet'in temel sürümleri verilmiş ve bu sürümlerin genel olarak özellikleri açıklanmıştır.

### 4.5.1 ResNet18

ResNet18, derin sinir ağlarında ortaya çıkan kaybolan gradyan (vanishing gradient) problemini çözmek amacıyla geliştirilen Artık Ağ (Residual Network, ResNet) mimarisinin bir varyantıdır. Bu mimari, ağların daha derin tasarlanmasına

olanak tanıyarak verilerdeki karmaşık örüntülerin öğrenilme kapasitesini artırmayı hedeflemektedir. ResNet18 modeli, konvolüsyon katmanları ve artık (residual) bloklar dahil olmak üzere 18 katmandan oluşmaktadır. Artık bloklar, ResNet mimarilerinin temel bileşenini oluşturmakta olup, bir veya birden fazla katmanı atlayarak bağlantı sağlayan “skip connection” (kısa yol bağlantısı) içerir.

Bir artık blok, girişin bir veya daha fazla katmanı kısa yol bağlantısı aracılığıyla atlmasına olanak tanır, bu durum, kaybolan gradyan (vanishing gradient) probleminin hafifletilmesine yardımcı olur. Bu mekanizma, orijinal haritalamayı öğrenmek yerine artık (residual) haritalamanın öğrenilmesiyle sağlanır. ResNet18’de tipik bir artık blok, batch normalization ve ReLU aktivasyon fonksiyonuna sahip iki evrişimli katmandan oluşur.

Kısa yol bağlantısı, bloğun girişini ikinci evrişimli katmanının çıkışına ekleyerek gradyanların ağ boyunca daha etkin bir şekilde akmasını sağlar. Temel amaç, modelin orijinal referanssız haritalamayı öğrenmek yerine artık haritalamaları öğrenmesine olanak tanımadır (GeeksforGeeks, 2025).

#### **4.5.2 ResNet34**

Sınıflandırma dahil birçok görevi yerine getiren hızlı ve basit bir evrişimli sinir ağıdır. Resnet34, 34 katmanlı bir evrişimli sinir ağı olarak yapılandırılmış ve “Görüntü Tanıma için Derin Kalıntı Öğrenimi”nde tanımlanan, bir görüntü sınıflandırma modelidir. Resnet34, 200 farklı sınıftan 100.000’den fazla görüntü içeren ImageNet veri seti üzerinde önceden eğitilmiştir.

Bununla birlikte, ResNet geleneksel sinir ağlarından farklıdır; çünkü her katmandan elde edilen artık (residual) değerleri alır ve bunları bir sonraki bağlı katmanlarda kullanır. Bu yaklaşım, metin tahmini için kullanılan artık sinir ağlarına benzer bir mekanizmadır (Roboflow, 2015).

#### **4.5.3 ResNet50**

ResNet50, artık değerli (residual) nöral ağların bir varyantıdır ve evrişimli sinir ağlarının geliştirilmiş bir versiyonunu temsil etmektedir. Geleneksel ardışık ağ mimarilerinden (AlexNet, VGGNet gibi) farklı olarak, ResNet50’de bazı katmanlar arasındaki değişimler atlanarak bir sonraki alt katmana geçiş yapılabilmektedir. Bu sayede ağın performansının daha yüksek seviyelere ulaşması sağlanmaktadır.

Modelin temel amacı, derin ağlar yakınsamaya başladığında ortaya çıkan evrişimli sinir ağlarındaki performans düşüşü problemini çözmektir. ResNet50 mimarisi, 177 katmandan oluşan bir ağ yapısına sahip olup, katmanlar arası bağlantıların nasıl kurulacağına dair detaylı bilgi sağlamaktadır. Giriş katmanı,  $224 \times 224 \times 3$  boyutunda yapılandırılmıştır (Gökalp ve Aydın, 2021:37).

#### **4.5.4 ResNet101**

ResNet-101 toplamda 104 evrişim katmanından oluşmaktadır. Ayrıca, toplamda 33 katman bloğu içermekte olup, bu bloklardan 29'u önceki bloğun çıktısını doğrudan kullanmaktadır. Bu durum, artık (residual) bağlantılar olarak tanımlanmakta ve söz konusu artık bağlantılar, her bloğun sonunda kullanılan toplam operatörünün ilk terimi olarak değerlendirilerek, bir sonraki bloğun girdisinin elde edilmesinde kullanılmaktadır. Geriye kalan 4 blok ise, önceki bloğun çıktısını alarak  $1 \times 1$  filtre boyutuna ve 1 adım (stride) uzunluğuna sahip bir evrişim katmanında kullanmakta, ardından normalizasyon işlemini gerçekleştiren bir toplu (batch) normalizasyon katmanı uygulanmaktadır. Elde edilen çıktı, ilgili bloğun çıktısında bulunan toplama operatörüne gönderilmektedir (Demir vd., 2019).

#### **4.5.5 ResNet152**

ResNet-152, devrim niteliğindeki artık (residual) öğrenme çerçevesini uygulayan derin bir evrişimsel sinir ağı modelidir. Microsoft tarafından geliştirilen bu model, özgün ResNet mimarisinin 1.5 sürümünü temsil etmekte olup, 152 katmandan oluşmaktadır ve görüntü sınıflandırma görevleri için ImageNet-1k veri kümesi üzerinde eğitilmiştir (PromptLayer, Tarih Yok).

## 5. YÖNTEM

Bu başlık altında araştırmanın modeli, veri seti, yöntemi, etiketleme uygulaması ve analizleri hakkındaki bilgilere yer verilmiştir.

### 5.1 Araştırmanın Modeli

Bu araştırma, YOLO mimarisi ve ResNet18 temel alınarak geliştirilen bir derin öğrenme modelinin, fotoğraflarda bulunan cihazları (telefon, kulaklık, akıllı saat vb.) tespit edebilmesi ve aynı zamanda fotoğrafların kalite düzeyini (aydınlık-karanlık, bulanıklık durumu, nesne yakınlığı/uzaklığı gibi kriterlere göre) “iyi” veya “kötü” olarak sınıflandırabilmesi temeline dayanmaktadır. Araştırma kapsamında YOLO ve ResNet18 modellerinin başarıları kıyaslanmıştır. Araştırma modeli, iki temel aşamadan oluşmaktadır. İlk aşamada, etiketlenmiş veri setleri kullanılarak eğitilen derin öğrenme modeli ile nesne tanıma ve görüntü kalite sınıflandırma işlemleri yapılmıştır. İkinci aşamada ise geliştirilen modelin kullanıcı dostu bir arayüz ile web tabanlı bir uygulamaya entegre edilmesidir. Bu kapsamda, araştırma modeli hem teknik başarıyı hem de kullanıcı erişilebilirliğini esas alan bütüncül bir yapıda tasarlanmıştır.

### 5.2 Araştırmanın Veri Seti

Çalışma için gerekli olan veri seti ilgili şirketten temin edilen 5000 adet cihaz fotoğrafı ile oluşturulmuştur. Cihaz fotoğraflarının etiketlenmesi yine çalışma kapsamında geliştirilen topluluk bazlı bir etiketleme yazılımı ile yapılmıştır. Etiketleme yazılımına tanımlanan kullanıcılar “Cihaz fotoğrafı ideal/kusursuz bir şekilde çekilmiş mi?” sorusuna “Evet” veya “Hayır” cevabını vererek fotoğrafların etiketlenmesini sağlamışlardır. Bu soruyu cevaplarken aşağıdaki kriterler dikkate alınmıştır:

Alınan 5000 adet fotoğraftan oluşan ham veri seti etiketlendikten sonra “iyi” ve “kötü” sınıflarına ayrılmıştır. Sonrasında veri seti train (eğitim), test ve validation

(doğrulama) olacak şekilde sırasıyla %70, %15 ve %15 olacak şekilde bölünmüştür ve Çizelge 5.1’de sunulmuştur.

**Çizelge 5.1: Veri Seti Dağılımı**

	Eğitim	Test	Doğrulama	Toplam
İyi	891	191	191	1273
Kötü	2609	559	559	3727
İyi ve Kötü	3500	750	750	5000
Dağılım	%70	%15	%15	%100

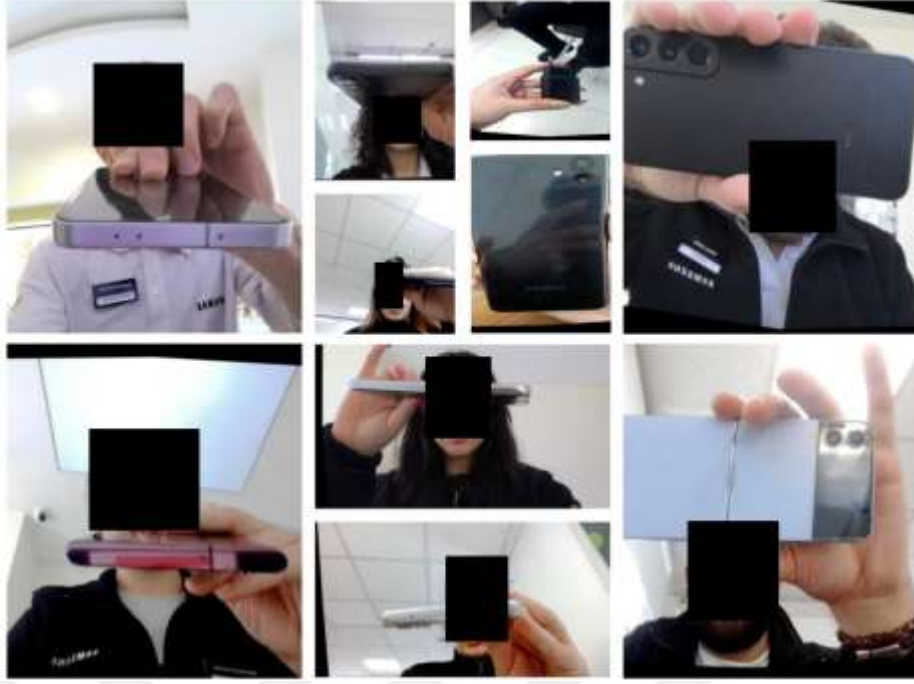
“Kötü” olarak etiketlenen fotoğraflara örnekler Şekil 5.1’de gösterilmiştir.



**Şekil 5.1: “Kötü” Olarak Etiketlenen Örnek 10 Fotoğraf**

Şekil 5.1’de çok aydınlık, çok karanlık, cihazın fotoğraf çerçevesine sığmaması, kameranın kapalı olması ve cihazın ideal mesafede olmaması gibi fotoğrafın “Kötü” kalitede çekildiğini belirleyen durumlar gösterilmektedir.

“İyi” olarak etiketlenen fotoğraflara örnekler Şekil 8’de gösterilmiştir.



**Şekil 5.2: "İyi" Olarak Sınıflandırılan Örnek 10 Fotoğraf**

Şekil 5.2’da fotoğrafta en az bir cihazın olması, ideal mesafede olması, bulanıklık gibi görüntü kalitesini olumsuz etkileyen etmenlerin olmaması, fotoğrafın odak noktasının cihaz olması gibi fotoğrafın “İyi” kalitede çekildiğini belirleyen durumlar gösterilmektedir.

### **5.3 Araştırmada Kullanılan Yöntemler**

Bu çalışmada, derin öğrenme modelini oluşturmak için bilgisayarla görme (computer vision) alanında önemli bir yere sahip olan gerçek zamanlı nesne algılama, sürücüsüz araçlardan güvenlik kameralarına kadar birçok alanda kullanılmakta olan YOLO mimarisi ve ResNet18 temel alınmıştır. Model fotoğrafın kalitesini “iyi” veya “kötü” olarak sınıflandırmayı amaçlamaktadır. YOLO’nun 11.sürümü ve ResNet18 kullanılmıştır. YOLO modelinde modelin hafif ve hızlı olabilmesi adına YOLO’nun n-cls modeli tercih edilmiştir.

Model eğitimi Google Colab üzerinde yapılmıştır. Google Colab, “Colaboratory” kelimesinin kısaltmasıdır ve kullanıcıların web tarayıcılarında doğrudan Python kodu yazıp çalıştırabilmelerini sağlayan ücretli veya ücretsiz bulut tabanlı bir platformdur. Colab notebook adı verilen etkileşimli bir ortam sunar ve bu ortamda çalıştırılabilir kod, zengin metin, görüntüler, HTML, LaTeX ve daha fazlasını tek bir belgede birleştirir (Colab, tarih yok).

Verilerin etiketlenmesi için tezin amacına uygun bir şekilde fotoğraf etiketleme uygulaması geliştirilmiştir. Geliştirilen etiketleme uygulaması ile veri seti birden fazla kullanıcı tarafından etiketlenmiştir. Kullanıcılar bilişim sektöründe aktif olarak çalışan profesyonellerden seçilmiştir. Kullanıcılara etiketleme öncesinde uygulamanın nasıl kullanılacağına ve fotoğrafların neyi ifade ettiğine dair sunum yapılarak etiketleme sürecinin doğru bir şekilde ilerlemesi sağlanmıştır. Bu sayede fotoğraflara verilen etiketlerin tutarlılığını artırmak amaçlanmıştır. Etiketleme uygulaması Vuejs, Nuxtjs ve Firebase teknolojileri kullanılarak geliştirilmiştir.

#### 5.4 Araştırma için Geliştirilen Etiketleme Uygulaması

Araştırmalar neticesinde çalışmanın ihtiyaç duyduğu fotoğraf etiketleme yazılımlarının yetersiz olması veya çok kapsamlı olması nedeniyle, çalışmanın sağlıklı bir şekilde ilerlemesi adına, fotoğrafların etiketlenmesi için topluluk bazlı etiketlemeyi destekleyen bir web uygulaması geliştirilmiştir. Uygulama Google ile giriş yapılarak sisteme tanımlı kullanıcıların etiketlenmesi için tanımlanan fotoğraflara belirlenen sorulara cevap vermesi ve çoğunluğun hem fikir olduğu cevabın doğru kabul edilmesi yöntemi uygulanmıştır. Uygulama Vue.js ve Nuxt ile geliştirilmiştir. Google Firebase'in Authentication, Firestore Database, Storage, Functions ve Hosting bileşenlerinden faydalanılmıştır.

**Vue.js:** Vue, kullanıcı arayüzleri oluşturmak için kullanılan bir JavaScript framework'üdür. Standart HTML, CSS ve JavaScript üzerine kuruludur ve herhangi bir karmaşıklıkta kullanıcı arayüzlerini verimli bir şekilde geliştirmenize yardımcı olan, bildirimsel, bileşen tabanlı bir programlama modeli sunmaktadır (Vue.js, tarih yok).

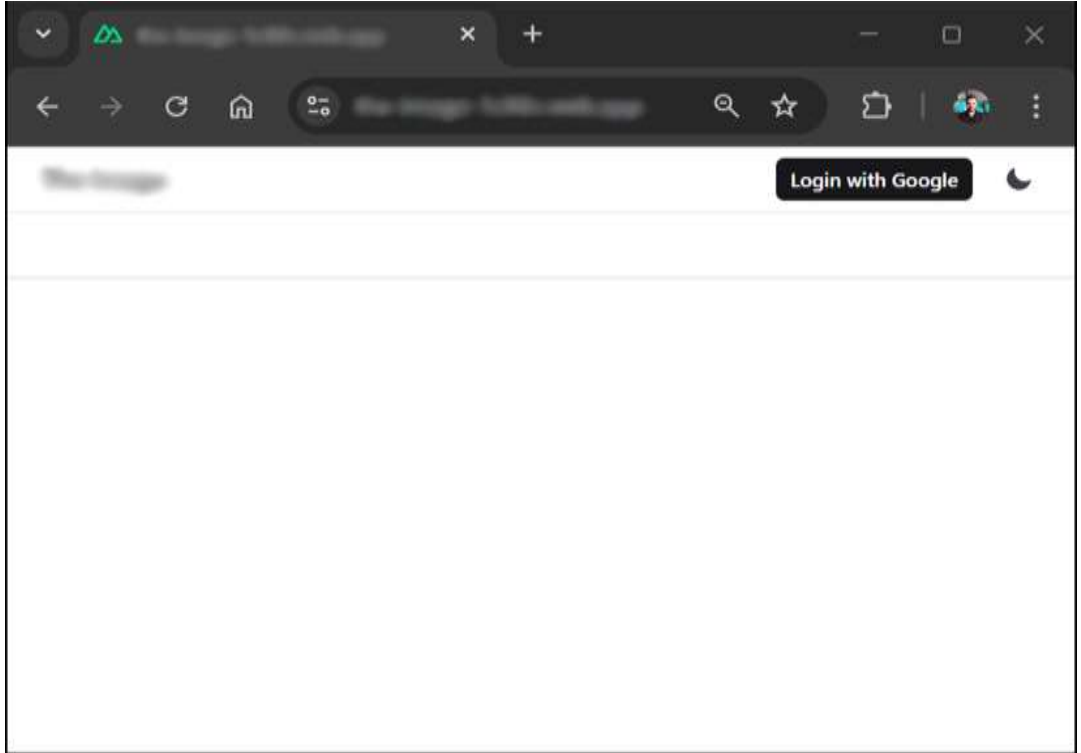
**Nuxt:** Vue.js tabanlı modern full-stack web geliştirme frameworkü olup, geliştiricilere sezgisel, esnek ve yüksek kaliteli uygulamalar oluşturma olanağı sunmaktadır. Açık kaynaklı yapısı sayesinde hem bireysel geliştiriciler hem de büyük ekipler tarafından erişilebilir ve ölçeklenebilir çözümler üretmek amacıyla kullanılabilir. Nuxt, yalnızca basit projeler için değil, aynı zamanda çok sayıda geliştiricinin katkı sağladığı karmaşık yapıli sistemler için de uygundur. Geliştirici deneyimini önceliklendiren bu çerçeve, farklı beceri düzeylerine sahip kullanıcılar için erişilebilirlik sağlarken, esnek yapısı sayesinde çeşitli derleme

araçlarıyla uyumlu çalışmakta, böylece geliştirme sürecinin hem özelleştirilebilirliğini hem de verimliliğini artırmaktadır (Nuxt, tarih yok).

**Firestore:** Google tarafından geliştirilen ve geliştiricilerin uygulamalarını oluşturmalarına, yönetmelerine ve büyütmelerine yardımcı olan çeşitli araçlar ve hizmetler sunan kapsamlı bir platformdur. Firestore'in temel bileşenlerinden biri, Firestore Realtime Database ve Cloud Firestore'u içeren veri tabanı hizmetleridir.

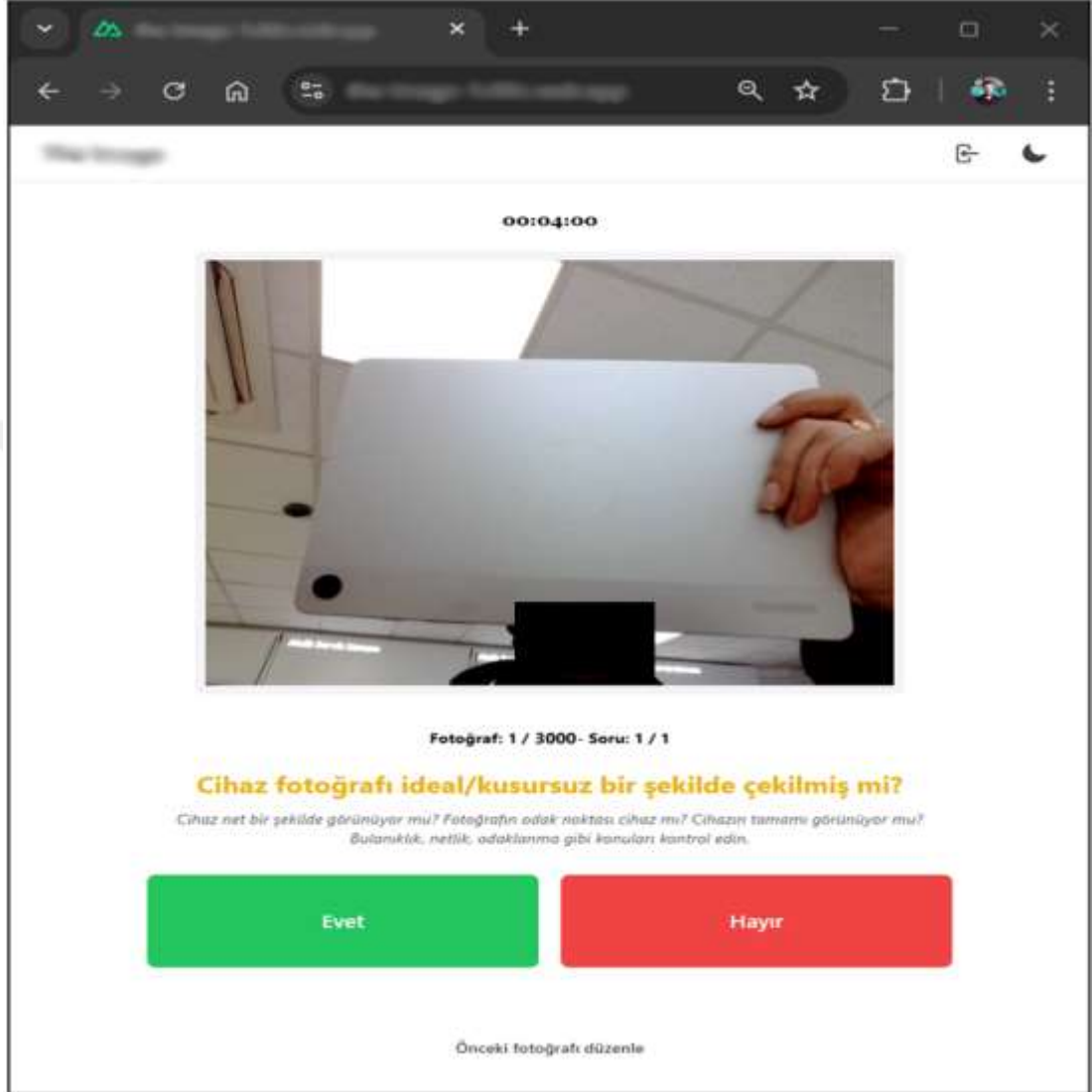
Firestore Realtime Database, verileri JSON olarak depolayan ve tüm bağlı istemcilerde gerçek zamanlı olarak senkronize eden bulut tabanlı bir NoSQL veri tabanıdır. Bu, verilerde yapılan herhangi bir değişikliğin tüm cihazlara anında yansıtılması anlamına gelir ve kesintisiz ve iş birliğine dayalı bir kullanıcı deneyimi sağlamaktadır. Realtime Database, sohbet uygulamaları, iş birliği araçları ve canlı veri akışları gibi gerçek zamanlı güncellemeler gerektiren uygulamalar için özellikle kullanışlıdır (Firestore, tarih yok).

Cloud Firestore, Firestore tarafından sunulan bir başka NoSQL veri tabanıdır. Realtime Database'e kıyasla daha esnek ve ölçeklenebilir bir çözüm sunmaktadır. Cloud Firestore, verileri koleksiyonlar halinde düzenlenmiş belgeler biçiminde depolar. Bu yapı, daha karmaşık sorgular ve büyük veri kümeleri için daha iyi performans sağlar (GeeksforGeeks, 2025).



Şekil 5.3: Etiketleme Uygulaması Giriş Ekranı

Yukarıda yer alan Şekil 5.3'te etiketleme uygulaması giriş ekranı sunulmuştur. Kullanıcılar Login with Google butonuna tıkladıklarında Google hesapları ile giriş yapmaktadır.



**Şekil 5.4: Etiketleme Uygulaması Fotoğraf Etiketleme Ekranı**

Yukarıda yer alan Şekil 5.4'de kullanıcılar etiketleme uygulamasına giriş yaptıktan sonra karşlarına gelen ekran gösterilmiştir. Ekranın ortasında kalitesi belirlenecek olan fotoğraf gösterilmektedir ve kullanıcılardan “Cihaz fotoğrafı ideal/kusursuz bir şekilde çekilmiş mi?” sorusu üzerinden değerlendirmesi istenmektedir. Kullanıcılar şekilde yer alan fotoğrafı genel olarak odak noktasının cihaz olması durumu, cihazın tamamının fotoğraf karesine sığması, netlik, bulanıklık, aydınlık, karanlık ve cihazın ideal uzaklıkta olması gibi açılardan değerlendirerek fotoğrafın kaliteli veya kalitesiz olduğunu “Evet” veya “Hayır” butonlarından birini

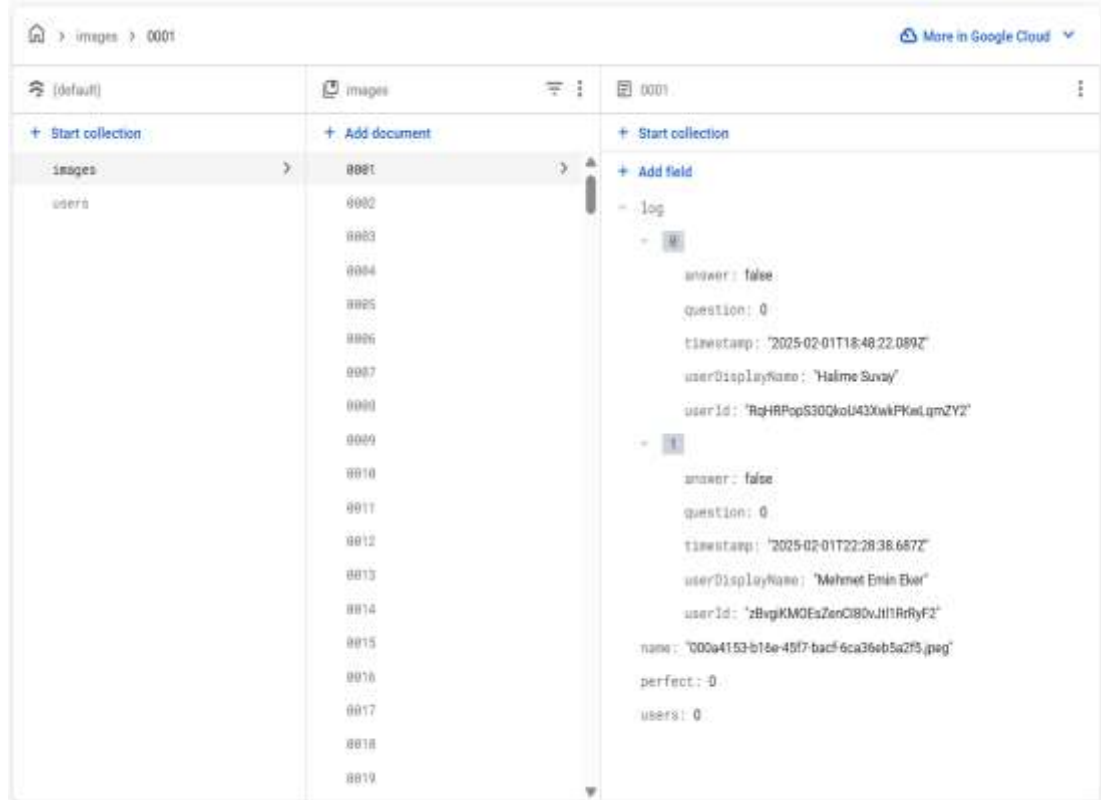
seçerek belirlemektedir. Devamında uygulama ekranında bir sonraki fotoğraf gösterilir.

Eğer kullanıcı fotoğraf ile ilgili hatalı bir değerlendirme yaptığını düşünür veya fark ederse “Önceki fotoğrafı düzenle” butonuna tıklayarak bir önceki fotoğrafı tekrar görüntüleyebilir. Bu ekran üzerinden kullanıcı fotoğrafa dair yeni değerlendirmesini gerçekleştirir. Veritabanında kullanıcının en son değerlendirmesi geçerli olacaktır.

Uygulama ekranında aynı zamanda kullanıcıların etiketlemeyi ne kadar sürede yaptığını göstermek için bir kronometre eklenmiştir. Örneğin, Şekil 5.4’de fotoğrafın üzerinde yer alan 00:04:00 değeri etiketleme süresini ifade etmektedir.

Kullanıcıların kaçınıcı fotoğrafı etiketlediğini göstermek üzere bilgilendirici metin eklenmiştir. Örneğin, Şekil 11’de fotoğrafın hemen altında bulunan “Fotoğraf: 1 / 3000 – Soru: 1 / 1” ifadesi etiketlenmesi gereken 3000 fotoğraf olduğunu ve şu an 1.fotoğrafın etiketlendiğini ve 1.sorunun cevaplandırıldığını ifade etmektedir.

Kullanıcıların verdikleri yanıtlar Google Firebase Firestore Veritabanına eş zamanlı olarak aktarılmaktadır.



Şekil 5.5: Google Firebase Firestore Veritabanı

Yukarıda yer alan Şekil 5.5’de Google Firebase Firestore Veritabanından bir ekran görüntüsü sunulmuştur. Veritabanında “images” ve “users” isimli iki koleksiyon kullanılmıştır. “images” koleksiyonunda kullanıcıların yanıtları fotoğraf bazlı olarak tutulmaktadır. “0001” şeklinde ifade edilen doküman 1.fotoğrafa karşılık gelmektedir. Bu dokümanın içerisinde yer alan “name” alanı fotoğrafın adını, “users” alanı fotoğrafı etiketleyen kullanıcı sayısını, “perfect” alanı kaç kullanıcının ilgili fotoğrafa kaliteli/ideal/kusursuz cevabını verdiğini, “log” alanı ise hangi kullanıcının hangi soruya ne zaman ne cevap verdiği ifade etmektedir. Örneğin, “log” alanına bakıldığında Halime Suvay isimli kullanıcının 01.02.2025 tarihinde saat 18:48’de 1.fotoğrafı kalitesiz olarak etiketlediği anlaşılmaktadır.

### **5.5 Modelin Bir Web Uygulamasında Kullanılması**

YOLO ve ResNet18 modelleri, makine öğrenmesi alanında popüler olan PyTorch adlı bir kütüphane kullanılarak eğitilmektedir. Eğitim sürecinin sonunda, modelin öğrenme çıktısı olan ağırlıklar ve parametreler, PyTorch'un kendi formatı olan .pt veya .pth uzantılı bir dosya içinde saklanmaktadır. Bu dosyası, modelin adeta "beyni" gibidir; yani, modelin nesnelere doğru bir şekilde tanımak için edindiği tüm bilgiyi barındırmaktadır. Bu, bir modelin eğitimini tamamladıktan sonra, aynı eğitimi yeniden yapmak zorunda kalmadan, bu dosyayı kullanarak doğrudan nesne tespiti yapmanıza olanak tanımaktadır. Bu sayede, modelin yetenekleri kolayca saklanabilmekte ve farklı ortamlarda yeniden kullanılabilir.

Bu dosyanın temel yapısı, modelin her bir katmanının sinir hücreleri arasındaki bağlantı güçlerini (ağırlıklarını) ve eğilim değerlerini (bias'ları) içermektedir. Örneğin, modelin bir katmanı belirli bir kenarı veya rengi tanımayı öğrenmişse, bu öğrenme, ilgili ağırlık değerlerinde saklıdır. .pt formatı, bu karmaşık sayısal değerleri sıkıştırarak ve organize bir şekilde tutarak, PyTorch'un bu bilgileri hızlıca belleğe yüklemesini ve çıkarım (inference) işlemini yüksek performansla gerçekleştirmesini sağlamaktadır. Bu özellik, modeli Python tabanlı masaüstü uygulamalarında, yerel sunucularda veya diğer PyTorch destekli ortamlarda kullanmayı oldukça pratik hale getirmektedir. Ancak, bu yapı PyTorch'a özgü olduğu için, bu formatın dışındaki platformlarda doğrudan kullanılması mümkün olmamaktadır.

Web uygulamalarının mimarisi ve çalışma ortamı, genellikle Python veya PyTorch ortamlarından farklı olmaktadır. Web tarayıcıları ve istemci tabanlı uygulamalar, model çalıştırma süreçlerini TensorFlow.js veya ONNX Runtime gibi özel kütüphaneler aracılığıyla yönetmektedir. Bu kütüphaneler, .pt formatındaki dosyaları doğrudan işleyememektedir. Bu nedenle, bir YOLO veya ResNet18 modelinin web tabanlı bir platformda kullanılabilmesi için, modelin web ortamına uyumlu bir formata dönüştürülmesi gerekmektedir.

Bu dönüştürme işlemi, modelin mimarisini ve öğrenilmiş ağırlıklarını, hedef platformun anlayabileceği bir yapıya aktarmayı amaçlamaktadır. Bu bağlamda en sık tercih edilen hedef formatlar TensorFlow.js ve ONNX olmaktadır.

### **5.5.1 PyTorch model tipi**

PyTorch, derin öğrenme alanında yaygın olarak kullanılan açık kaynaklı bir makine öğrenmesi kütüphanesidir. PyTorch ile oluşturulan modellerin kaydedilmiş versiyonları genellikle .pt veya .pth uzantılı dosyalarda saklanmaktadır. Bu dosyalar, modelin eğitilmiş ağırlıklarını, mimarisini ve öğrenilmiş parametrelerini içerir. PyTorch modeli, kütüphanenin kendi ekosistemi içinde, Python tabanlı ortamlarda en yüksek verimlilikle çalışmak üzere optimize edilmiştir. Bu nedenle, bir .pt dosyasını kullanabilmek için genellikle PyTorch kütüphanesinin kurulu olduğu bir Python ortamı gereklidir.

### **5.5.2 TensorFlow.js model tipi**

TensorFlow.js, makine öğrenmesi modellerini doğrudan web tarayıcılarında veya Node.js ortamında çalıştırmak için geliştirilmiş bir JavaScript kütüphanesidir. TensorFlow.js modeli, tarayıcıların anlayabileceği formatta yapılandırılmış dosyalar topluluğundan oluşur. Bu format, genellikle modelin mimarisini tanımlayan bir model.json dosyası ve ağırlıkları içeren bir veya daha fazla ikili (binary) dosyadan meydana gelir. Bu model tipi, makine öğrenmesi çıkarımını kullanıcıların cihazlarına taşıyarak sunucu yükünü azaltır ve internet bağlantısı olmadan bile çalışabilme yeteneği sunmaktadır. TensorFlow.js modelleri, web tabanlı etkileşimli deneyimler ve gerçek zamanlı uygulamalar için ideal bir çözüm olmaktadır.

### 5.5.3 ONNX Runtime model tipi

ONNX (Open Neural Network Exchange), farklı makine öğrenmesi çerçeveleri arasında modellerin taşınabilirliğini sağlamak amacıyla geliştirilmiş açık bir standarttır. ONNX modeli, modelin ağ yapısını ve eğitilmiş ağırlıklarını, .onnx uzantılı tek bir dosyada saklar. Bu formatın temel amacı, bir modelin PyTorch, TensorFlow veya Keras gibi farklı kütüphanelerde eğitilip, başka bir kütüphane veya platformda sorunsuz bir şekilde kullanılabilmesidir. Bu model tipi, platformlar arası uyumluluk ve esneklik açısından büyük avantaj sunmaktadır. ONNX dosyaları, hem sunucu tarafı uygulamalarda ONNX Runtime gibi motorlarla hem de web tarayıcılarında ONNX.js ile kullanılabilir. Bu özelliği sayesinde, modelin deployment (dağıtım) süreçlerini kolaylaştırmaktadır.



## 6. VERİNİN ANALİZİ VE BULGULAR

### 6.1 YOLO Modelinden Elde Edilen Bulgular

Model eğitimi Google Colab NVIDIA L4 GPU mimarisi kullanılarak gerçekleştirilmiştir. Eğitim 105 dakikada tamamlanmıştır. Modelin overfitting (aşırı öğrenme) olmaması adına veri setinde yatay ve dikey çevirme, eğim verme, siyah beyaz görüntü oluşturma, parlaklık değerini değiştirme, bulanıklaştırma gibi çeşitli veri genişletme yöntemleri uygulanmıştır. “İyi” olan fotoğrafların bozulmaması ve modelin daha doğru çalışması amacıyla data augmentation özellikle “kötü” olarak etiketlenen fotoğraflara daha yoğun uygulanmıştır. YOLOv11’in transfer learning (transfer öğrenme) desteği ile ImageNet veri seti üzerine eğitim gerçekleştirilmiştir (Ultralytics YOLO Docs, 2025). Ayrıca VizWiz veri seti ile de ekstra olarak veri çeşitliliği sağlanmıştır (VizWiz, 2020). Model overfitting’e eğilimli olduğu için dropout (bırakma/seyreltme), learning rate (öğrenme oranı) ve early stopping (erken durdurma) yöntemleri uygulanmıştır. Epoch değeri 100, batch boyutu ise 16 olarak belirlenmiştir. Early stopping uygulanması nedeniyle 20 epoch boyunca bir iyileşme olmadığı için 55.epochta otomatik olarak durdurulmuştur. 55 epoch sonucunda 35.epochta elde edilen en iyi model kullanılmıştır.

```
Sat Jul 5 16:59:32 2025
```

NVIDIA-SMI 550.54.15			Driver Version: 550.54.15			CUDA Version: 12.4		
GPU	Name	Persistence-M	Bus-Id	Disp.A	Volatile Uncorr. ECC			
Fan	Temp	Perf	Pwr:Usage/Cap	Memory-Usage	GPU-Util	Compute M.	MIG M.	
0	NVIDIA L4	Off	00000000:00:03.0	Off	0			
N/A	34C	P8	11W / 72W	0MiB / 23034MiB	0%	Default	N/A	

Processes:							
GPU	GI	CI	PID	Type	Process name	GPU Memory	Usage
ID	ID						
No running processes found							

Şekil 6.1: Google Colab NVIDIA L4 GPU Kullanımı

Şekil 6.1’de Google Colab üzerinde NVIDIA L4 GPU mimarisinin aktif edildiği gösterilmektedir.

```
train: /content/images/train... found 18574 images in 2 classes ✓
val: /content/images/val... found 3977 images in 2 classes ✓
test: /content/images/test... found 3977 images in 2 classes ✓
```

	from	n	params	module	arguments
0	-1	1	464	ultralytics.nn.modules.conv.Conv	[3, 16, 3, 2]
1	-1	1	4672	ultralytics.nn.modules.conv.Conv	[16, 32, 3, 2]
2	-1	1	6640	ultralytics.nn.modules.block.C3k2	[32, 64, 1, False, 0.25]
3	-1	1	36992	ultralytics.nn.modules.conv.Conv	[64, 64, 3, 2]
4	-1	1	26080	ultralytics.nn.modules.block.C3k2	[64, 128, 1, False, 0.25]
5	-1	1	147712	ultralytics.nn.modules.conv.Conv	[128, 128, 3, 2]
6	-1	1	87040	ultralytics.nn.modules.block.C3k2	[128, 128, 1, True]
7	-1	1	295424	ultralytics.nn.modules.conv.Conv	[128, 256, 3, 2]
8	-1	1	346112	ultralytics.nn.modules.block.C3k2	[256, 256, 1, True]
9	-1	1	249728	ultralytics.nn.modules.block.C2PSA	[256, 256, 1]
10	-1	1	332802	ultralytics.nn.modules.head.Classify	[256, 2]

YOLO11n-cls summary: 86 layers, 1,533,666 parameters, 1,533,666 gradients, 3.3 GFLOPs

### Şekil 6.2: Derin Öğrenme Modeli Katmanları

Şekil 6.2’de modelin görüntü sınıflandırması yapmak üzere tasarlanmış katmanlar görünmektedir. Model, ilk evrişim katmanlarıyla (ultralytics.nn.modules.conv.Conv) görüntüden temel özellikleri çıkararak başlar ve bu özellikleri, art arda gelen C3 ve C2PSA gibi karmaşık bloklarla (ultralytics.nn.modules.block.C3k2, ultralytics.nn.modules.block.C2PSA) derinlemesine işler. Bu bloklar, özellik haritalarını verimli bir şekilde birleştirerek ağın derinliğini artırırken hesaplama maliyetini düşürmeyi hedefler. Katmanların parametre sayısı, derinleştikçe artarak modelin daha soyut ve karmaşık desenleri öğrenme kapasitesini yansıtır. Son olarak, modelin ultralytics.nn.modules.head.Classify adlı başlık katmanı, tüm bu özelliklerin üzerine ikili sınıflandırma yaparak nihai tahmini gerçekleştirir. Toplam 86 katman, 1.5 milyondan fazla parametreye sahip olduğu görünmektedir.

```
results = model.train(  
    data="/content/images",  
    epochs=100,  
    imgsz=640,  
    erasing=0.0,  
    mosaic=0.0,  
    scale=0.0,  
    translate=0.0,  
    dropout=0.2,  
    hsv_s=0.0,  
    hsv_v=0.0,  
    patience=20,  
    lr0=0.01,  
    lrf=0.001,  
    cos_lr=True,  
    batch=16,  
    weight_decay=0.0001,  
    plots=True,  
    device=[-1, -1],  
)
```

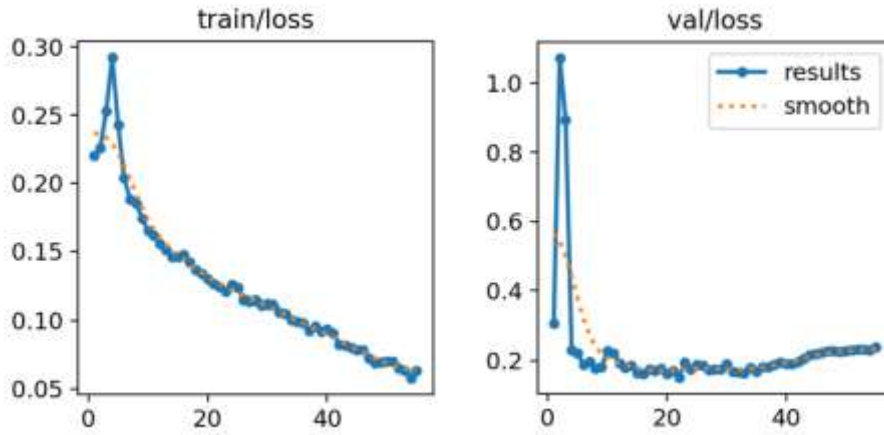
**Şekil 6.3: YOLO Model Eğitim Parametreleri**

Şekil 6.3'te yer alan YOLO modelinin eğitimi için belirlenen parametrelere bakıldığında, model eğitiminde 100 epoch tercih edilmiş olup, bu süre zarfında ağırlıklar sürekli güncellenerek hedeflenen doğruluk seviyesine ulaşması amaçlanmıştır. Giriş görselleri için 640×640 piksel boyut seçilmiş, bu da modelin hem yeterli ayrıntıyı yakalayabilmesi hem de işlem maliyetini dengeleyebilmesi açısından uygun bir çözünürlük sağlamaktadır. Eğitim sürecinde batch size 16 kullanılmıştır. Bu değer, bellek kapasitesi ve öğrenme sürecinin istikrarı arasında bir denge gözetilerek belirlenmiştir.

Veri artırma stratejileri incelendiğinde, mosaic, scale, translate, hsv-saturation ve hsv-value parametrelerinin sıfırlandığı görülmektedir. Bu durum, modelin eğitim sırasında orijinal görüntüler üzerinde herhangi bir yapay çeşitlilik oluşturmadığını, dolayısıyla öğrenme sürecinin yalnızca mevcut veri setinin doğal dağılımı üzerinden gerçekleştiğini göstermektedir. Bununla birlikte, dropout oranı 0.2 olarak belirlenmiştir; bu sayede ağırlıklı nöronları rastgele devre dışı bırakılarak aşırı öğrenmenin (overfitting) önüne geçilmesi hedeflenmiştir. Benzer şekilde erasing parametresinin sıfır olarak bırakılması, modelin yapay olarak görüntü parçalarını silmeden eğitildiğini göstermektedir.

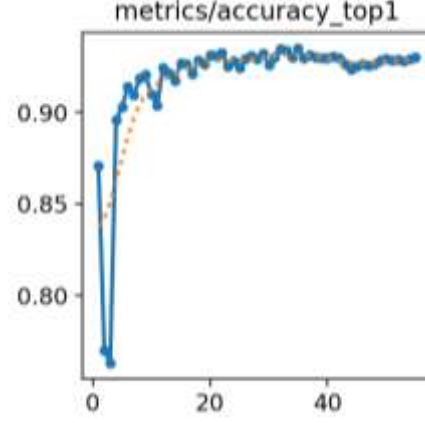
Öğrenme sürecinde kullanılan optimizasyon parametrelerine bakıldığında başlangıç öğrenme oranı (lr0) 0.01 olarak atanmış, bu oran ise eğitim ilerledikçe cosine learning rate scheduling yöntemiyle kademeli olarak 0.001 değerine kadar düşürülmüştür. Bu strateji, modelin başlangıçta daha hızlı öğrenmesini, ilerleyen aşamalarda ise daha küçük adımlarla güncelleme yaparak yerel minimumlara daha kararlı şekilde ulaşmasını sağlamaktadır. Ayrıca weight decay katsayısı 0.0001 olarak belirlenmiş ve böylece modelin ağırlıklarının aşırı büyümesinin önüne geçilerek genelleme yeteneği güçlendirilmiştir.

Eğitim sürecinin etkinliğini artırmak amacıyla early stopping mekanizması devreye alınmış, patience değeri 20 olarak belirlenmiştir. Bu, modelin 20 ardışık epoch boyunca gelişme göstermemesi halinde eğitimin sonlandırılacağı anlamına gelmektedir. Eğitim süreci boyunca elde edilen kayıp fonksiyonları ve başarı metrikleri plots=True parametresi sayesinde görselleştirilmiştir. Son olarak, model eğitimi sırasında device=[-1, -1] ayarı ile eğitim işleminin CPU üzerinde gerçekleştirildiği anlaşılmaktadır.



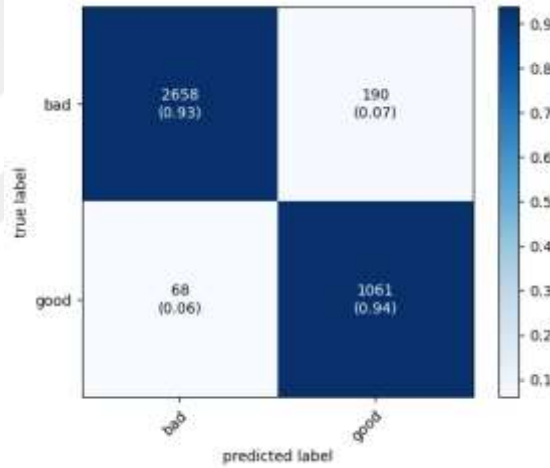
**Şekil 6.4: YOLO Modeli Eğitimi Sonrası Elde Edilen Loss Grafikleri**

Şekil 6.4'e bakıldığında modelin eğitimine 55 epoch boyunca devam edildiği görülmektedir. Train/loss (eğitim kaybı) ve val/loss (doğrulama kaybı) grafiklerine bakıldığında train/loss değerinin beklenen şekilde düzgün azalarak devam ettiği ancak val/loss değerinin ise küçük miktarlarda artış sergilediği gözlemlenmiştir. Val/loss grafiğindeki artışın devam etmesi overfitting'e neden olacağı için early stopping (erken durdurma) uygulanarak, 20 epoch boyunca artış olması durumunda durdurulmuş ve 35.epoch baz alınmıştır.



**Şekil 6.5: YOLO Modeli Eğitimi Sonrası Elde Edilen top-1 accuracy Grafiği**

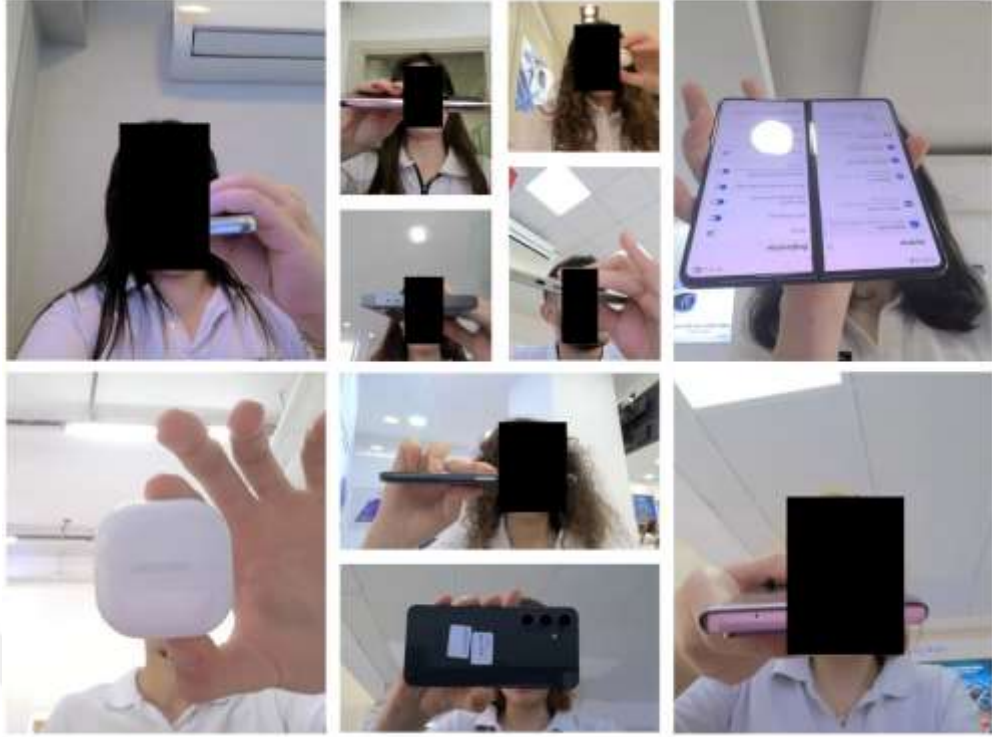
Şekil 6.5'te top-1 accuracy (doğruluk) grafiği gösterilmiştir. Eğitim sonucunda elde edilen doğruluk oranı 35.epochta baz alınan değer olan %93.5 olmuştur.



**Şekil 6.6: YOLO Modeli Eğitimi Sonrası Elde Edilen Confusion Matrix Grafiği**

Şekil 6.6'da yer alan Confusion matrix (karışıklık matrisi) grafiğine bakıldığında, modelin toplam 3977 örnek üzerinde test edildiği görülmektedir. Gerçek sınıfı “bad” olan 2848 örneğin 2658’i doğru, 190’ı yanlış tahmin edilmiştir. Gerçek sınıfı “good” olan 1129 örneğin 1061’i doğru, 68’i yanlış tahmin edilmiştir. “bad” sınıfı için doğruluk oranı %93, “good” sınıfı için ise %94 olduğu gözlemlenmiştir.

Model test edildiğinde kesinlik (precision) değerinin 0.9340, duyarlılık (recall) değerinin 0.9341 ve f1-score değerinin ise 0.9333 olduğu tespit edilmiştir.



**Şekil 6.7: "İyi" Olarak Sınıflandırılan 10 Adet Fotoğraf Örneği**

Şekil 6.7'de modelin "İyi" olarak sınıflandırdığı 10 adet fotoğraf örneği görülmektedir. Fotoğraflarda bir cihazın olduğu ve odak noktasının cihaz olduğu anlaşılmaktadır.



**Şekil 6.8: "Kötü" Olarak Sınıflandırılan 10 Adet Fotoğraf Örneği**

Şekil 6.8’de modelin “Kötü” olarak sınıflandırdığı 10 adet fotoğraf örneği görülmektedir. Fotoğraflara bakıldığında ışık problemlerinin olduğu, kamera lensinin parmak ile kapatıldığı, cihazın net gözükmeyişi veya kadrāja tam olarak alınmadığı anlaşılmaktadır.

## 6.2 ResNet18 Modelinden Elde Edilen Bulgular

YOLO modelinin eğitim sürecinde hazırlanan aynı veri seti kullanılarak ResNet18 modelinin eğitilmesi sağlanmıştır. Model eğitimi yine Google Colab NVIDIA L4 GPU mimarisi kullanılarak gerçekleştirilmiştir. Eğitim 45 dakikada tamamlanmıştır.

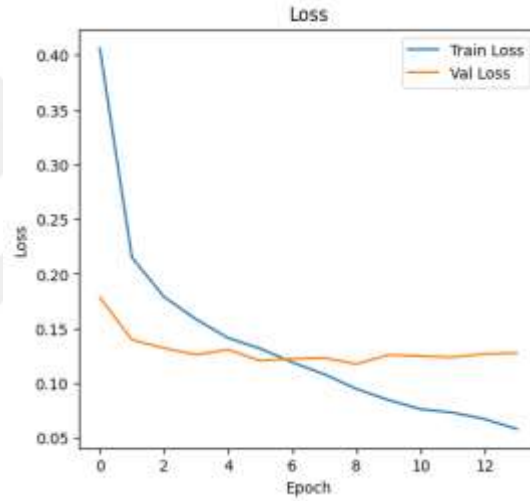
Kullanılan ResNet18 modeli, transfer öğrenme yaklaşımıyla eğitilmiştir. Modelin ilk katmanları önceden ImageNet veri seti üzerinde eğitilmiş ağırlıklarla başlatılmış, böylece modelin düşük seviyeli özellikleri (kenar, doku, renk geçişleri vb.) tanıyabilmesi sağlanmıştır. Son katman ise mevcut problemde iki sınıfın (iyi ve kötü) ayrımını yapacak şekilde yeniden yapılandırılmıştır. Bu kapsamda, modelin tam bağı katmanı (fully connected layer) çıkarılarak yerine %50 dropout içeren ve ardından iki nöronlu doğrusal (linear) bir katman eklenmiştir. Dropout mekanizması, aşırı öğrenmeyi önlemek amacıyla nöronların bir kısmının rastgele devre dışı bırakılması anlamına gelmektedir.

Veri ön işleme sürecinde, eğitim ve doğrulama verilerine farklı dönüşümler uygulanmıştır. Eğitim aşamasında, rastgele kırpma ve yeniden boyutlandırma (RandomResizedCrop) yöntemi kullanılarak modelin farklı ölçeklerde öğrenmesi teşvik edilmiştir. Ayrıca rastgele yatay çevirme ve parlaklık ile kontrast değişiklikleri (ColorJitter) uygulanarak veride çeşitlilik artırılmıştır. Bu yöntemler, modelin gerçek dünya koşullarındaki varyasyonlara karşı daha dayanıklı hale getirilmesini amaçlamaktadır. Tüm görseller, 224×224 piksel boyutuna dönüştürülmüş ve ResNet mimarisinin standart girdi normalizasyon değerleri kullanılarak ölçeklenmiştir. Doğrulama ve test verileri için ise yalnızca yeniden boyutlandırma ve normalizasyon işlemleri yapılmıştır.

Eğitim sürecinde, batch size 48 kullanılmış ve veri kümesi eğitim, doğrulama ve test olacak şekilde üç alt bölüme ayrılmıştır. Eğitim sırasında Adam optimizasyon algoritması tercih edilmiş, başlangıç öğrenme oranı 1e-5 olarak belirlenmiş ve ağırlık

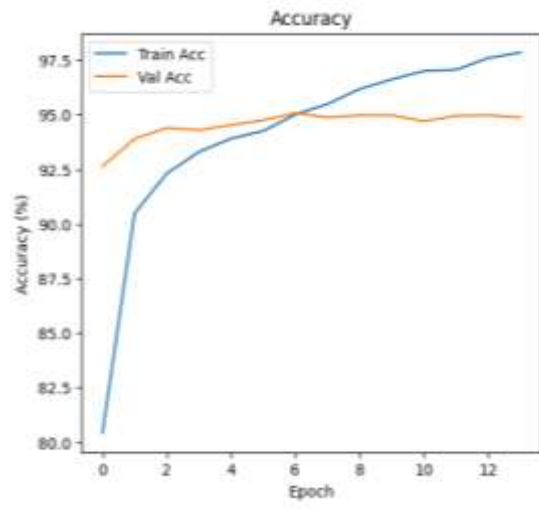
çürümesi (weight decay)  $1e-5$  değeriyle uygulanmıştır. Böylece parametrelerin aşırı büyümesi engellenmiş ve modelin genelleme yeteneği artırılmıştır. Öğrenme oranı zamanla düzenlenmiş, bu amaçla CosineAnnealingLR kullanılmıştır. Bu yaklaşım, öğrenme oranını eğitim boyunca kademeli olarak azaltarak modelin daha istikrarlı bir şekilde minimum noktalara ulaşmasını sağlamaktadır.

Eğitim süreci boyunca 50 epoch için planlama yapılmış, ancak aşırı öğrenmeyi engellemek ve gereksiz hesaplama maliyetlerini azaltmak amacıyla early stopping stratejisi devreye alınmıştır. Eğer doğrulama kaybı (validation loss) üst üste 5 epoch boyunca iyileşme göstermemişse, eğitim sonlandırılmıştır. Modelin eğitimi, GPU hesaplama gücünden faydalanılarak hızlandırılmış ve bu sayede eğitim süresi önemli ölçüde azaltılmıştır.



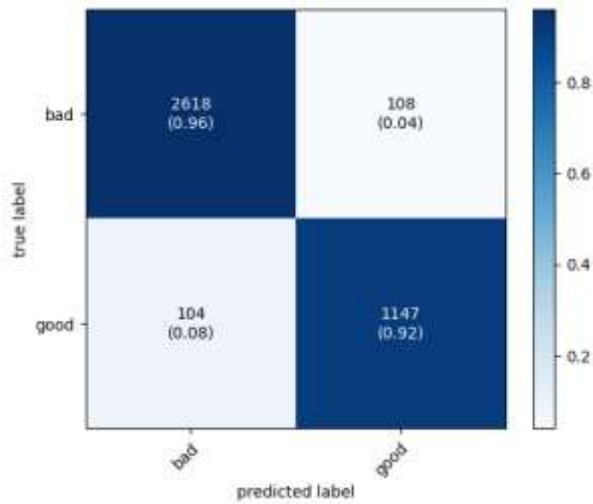
**Şekil 6.9: ResNet18 Modeli Eğitimi Sonrası Elde Edilen Loss Grafiği**

Modelin öğrenme sürecinin etkinliği, Şekil 6.9'da gösterilen kayıp (loss) grafiği ile analiz edilmiştir. Bu grafikte, eğitim kaybının (Train Loss) epoch'lar ilerledikçe istikrarlı bir şekilde azaldığı görülmektedir. Bu durum modelin eğitim veri setindeki örüntüleri başarıyla öğrendiğine işaret etmektedir. Buna karşılık, doğrulama kaybı (Val Loss) ilk birkaç epoch'ta düştükten sonra yaklaşık 8. epoch'tan itibaren düşüşünü durdurmuş ve yatay bir seyir izlemeye başlamıştır. Doğrulama kaybındaki bu durağanlaşma, modelin artık yeni veriler üzerinde genelleme yapma yeteneğinin iyileşmediğini ve aşırı öğrenme riskinin başladığını göstermektedir.



**Şekil 6.10: ResNet18 Modeli Eğitimi Sonrası Elde Edilen Accuracy Grafiği**

Modelin sınıflandırma başarımındaki gelişim ise Şekil 6.10'daki doğruluk (accuracy) grafiği ile izlenmiştir. Eğitim doğruluğu (Train Acc), eğitim kaybındaki düşüşe paralel olarak süreç boyunca sürekli bir artış göstererek %97'nin üzerine çıkmıştır. Doğrulama doğruluğu (Val Acc) ise hızlı bir başlangıcın ardından %95 seviyelerinde durağanlaşmıştır. Doğrulama doğruluğunun daha fazla artmaması, modelin performansının tepe noktasına ulaştığını teyit etmektedir. Bu grafik, early stopping mekanizmasının neden devreye girdiğini ve eğitimi en uygun noktada sonlandırarak aşırı öğrenmeyi engellediğini görsel olarak açıklamaktadır.



**Şekil 6.11: ResNet18 Modeli Eğitimi Sonrası Elde Edilen Confusion Matrix Grafiği**

Eğitim tamamlandıktan sonra, modelin genelleme performansı daha önce hiç görmediği test verileri kullanılarak değerlendirilmiştir. Bu değerlendirmenin sonuçları, Şekil 6.11'de sunulan karmaşıklık matrisi (confusion matrix) ile

özetlenmektedir. Bu matris, modelin her bir sınıf için ne kadar başarılı olduğunu detaylı olarak göstermektedir. Matrise göre model, "kötü" olarak etiketlenmiş örneklerin %96'sını doğru bir şekilde sınıflandırmıştır. Benzer şekilde, "iyi" olarak etiketlenmiş örneklerdeki başarı oranı ise %92'dir. Bu değerler, modelin her iki sınıfta da yüksek bir isabet oranıyla çalıştığını, ancak "kötü" sınıfını tanımada marjinal olarak daha başarılı olduğunu ortaya koymaktadır.

Kullanılan ResNet18 mimarisi toplamda 11.177.538 parametre içermektedir. Bu büyüklük, modelin yüksek temsil kapasitesine sahip olduğunu, ancak aynı zamanda aşırı öğrenme riskine karşı dikkatli bir eğitim süreci gerektirdiğini göstermektedir. Genel olarak, eğitim süreci modelin hızlı bir şekilde yüksek doğruluk seviyelerine ulaştığını, ancak erken dönemden itibaren performansın doygunluğa girdiğini göstermektedir. Bu sonuç, transfer öğrenme sayesinde modelin güçlü bir başlangıç yaptığı ve sınırlı sayıda epoch içinde yüksek başarı elde ettiğini ortaya koymaktadır.

### 6.3 YOLO ve ResNet18 Modellerinden Elde Edilen Bulguların Karşılaştırılması

Aşağıda yer alan Tablo 2'de YOLO ve ResNet18 modellerinden elde edilen sonuçlar sunulmuştur.

**Çizelge 6.1: YOLO ve ResNet18 Model Metrikleri**

	Eğitim Süresi (dk)	Doğruluk (Accuracy)	Kesinlik (Precision)	Duyarlılık (Recall)	F1-Score	Model Boyutu (mb)
<b>YOLO</b>	105	0.9350	0.9340	0.9341	0.9333	5.9
<b>ResNet18</b>	45	0.9497	0.9496	0.9497	0.9496	42.6

Çizelge 6.1'deki verilere bakıldığında, YOLO ve ResNet18 modelleri farklı performans ve kaynak kullanımı profilleri sergilemiştir. Eğitim süresi açısından ResNet18 45 dakika ile YOLO'nun 105 dakikalık süresine kıyasla daha kısa bir sürede eğitilebilmektedir. Bu da ResNet18'in eğitim verimliliğinin daha yüksek olduğunu göstermektedir. Doğruluk (Accuracy) açısından ResNet18 %94.97 ile YOLO'nun %93.50'sinden daha iyi sonuç vermekte ve benzer şekilde kesinlik (Precision) ve duyarlılık (Recall) değerlerinde de ResNet18 hafif bir üstünlük

sağlamaktadır. Sırasıyla %94,96 ve %94.97 ile YOLO'nun %93.40 ve %93.41'inin üzerine çıkarken F1-Score değerleri de bu eğilimi desteklemektedir. ResNet18'in %94.96'lık F1-Score'u, YOLO'nun %93.33'lük değerinin üzerindedir. Model boyutu açısından değerlendirildiğinde, YOLO'nun belirgin bir avantaja sahip olduğu görülmektedir. YOLO 5.9 MB boyutunda olup, ResNet18'in 42.6 MB'lık boyutuna kıyasla daha küçüktür. Bu durum, özellikle sınırlı hafıza kapasitesine veya düşük işlem gücüne sahip cihazlar için önemli bir avantaj teşkil etmektedir. Özetle, ResNet18 performans açısından YOLO'dan daha yüksek değerler sunarken, YOLO eğitim süresi ve model boyutu açısından daha ekonomik ve hızlı bir çözüm sunmaktadır.



## 7. MODELİN KULLANILMASI

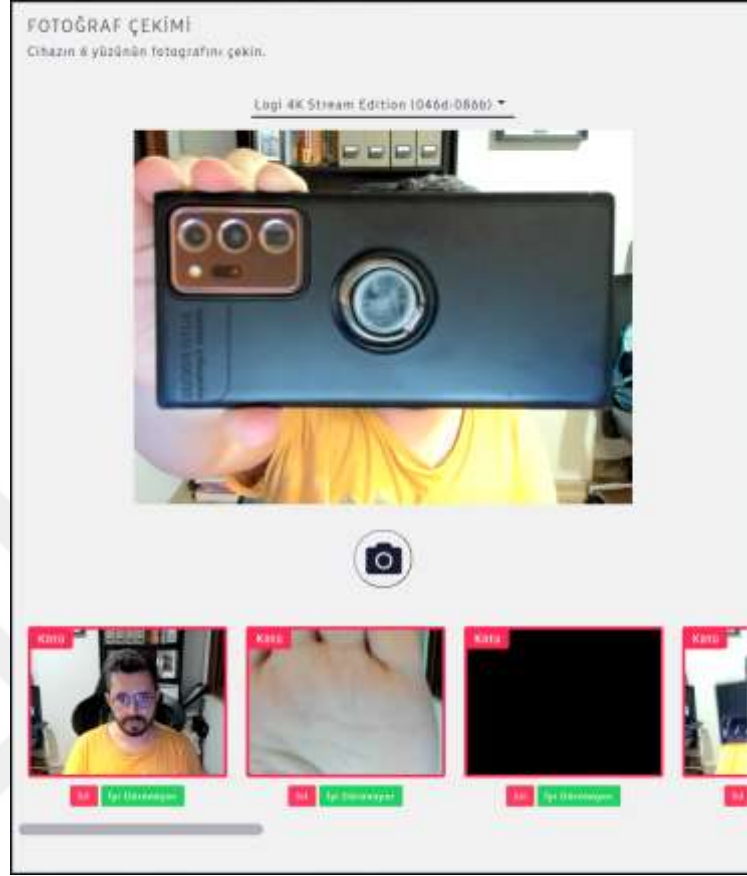
### 7.1 Web Arayüzünün Geliştirilmesi

Geliştirilen derin öğrenme modeli ASP.NET Core ve Vue.js ile geliştirilmiş olan müşteri tarafından aktif olarak kullanılan fotoğraf çekme ekranına entegre edilmiştir. Fotoğraf çekme ekranında webcam bağlantısı ile cihazların fotoğraflar çekilmektedir. Daha önceden çekilmiş olan fotoğraflar incelendiğinde kullanıcıların çoğunluğunun özentsiz düşük kalitede fotoğraf çektiği tespit edilmiştir. Bu kapsama geliştirilen model ile kullanıcıya öneri niteliğinde çekmiş olduğu fotoğrafın kalitesi tahmin edilmektedir. Çeşitli fotoğraflarda hiçbir cihazın olmaması da istenilen bir durum olmamaktadır. Bu bağlamda YOLO veya ResNet18 ile oluşturulan model PyTorch formatında olduğu için web uygulamasında kullanılmak üzere ONNX Runtime formatına dönüştürülmüştür. Bu sayede model ASP.NET Core tarafında kullanılabilir hale getirilmiştir.

Kullanıcıların iş akışını etkilememek adına modelin tahminleri öneri niteliğinde sunulmuştur. Kullanıcı fotoğrafın iyi kalitede olduğunu düşünüyorsa “Kötü” olarak etiketlenen fotoğraf “İyi” olarak manuel olarak kullanıcı tarafından yeniden etiketlenmesi sağlanmıştır. Her fotoğraf için belirlenen etiket veritabanında saklanmaktadır. Buradaki amaç modelin tekrar eğitilmesi için veri seti oluşturarak modelin doğruluk oranını sürekli iyileştirmektir. Bu sayede sürekli öğrenen bir yapı kurulmuştur. Kullanıcıların modelin başarı oranını olumsuz etkilememeleri için etiketi değiştirilen fotoğraflar tekrar gözden geçirilerek model eğitiminde kullanılacaktır.

ASP.NET Core tarafında ONNX Runtime formatındaki YOLO modeli YoloSharp kütüphanesi kullanılarak entegre edilmiştir. YoloSharp YOLO modellerinin ASP.NET Core projelerine kolay bir şekilde entegre edilmesini sağlamaktadır. ResNet18 modeli ise Microsoft.ML.OnnxRuntime kütüphanesi kullanılarak YOLO modelinin entegre edilmesine benzer şekilde entegre edilerek kullanılmıştır. Microsoft.ML.OnnxRuntime kütüphanesi ONNX formatındaki yapay

zekâ modellerini farklı platformlarda (Windows, Linux, macOS) yüksek performansla çalıştırmaya yarayan bir kütüphanedir.



**Şekil 7.1: Web Uygulaması Fotoğraf Çekme Ekranı**

Şekil 7.1’de cihaz fotoğrafının çekildiği ekran yer almaktadır. Kameraya cihaz gösterilerek cihazın fotoğrafının çekilmesi sağlanmaktadır. Çekilen fotoğraflar ekranın alt kısmında yer alan bölümde listelenmektedir. Liste eklenen fotoğraf modele gönderilerek fotoğrafın kalitesi tahmin ettirilmektedir.



**Şekil 7.2: Web Uygulamasında Kötü Olarak Etiketlenen Fotoğraflar 1**

Şekil 7.2’de çekilen fotoğraflara verilen “Kötü” etiketi gösterilmektedir. Kötü olarak etiketlenen fotoğrafların hemen altında yer alan “İyi Görünüyor” butonu ile

kullanıcıların fotoğrafları yeniden etiketlemesi sağlanmaktadır. Burada modelin doğruluk oranını artırmak hedeflenmiştir. Fotoğraflara bakıldığında 1.fotoğrafta kameraya cihaz tutmayan bir kullanıcı görülmektedir. Cihaz olmadığı için model “Kötü” olarak etiketlemiştir. 2. ve 3.fotoğrafa bakıldığında kamera el veya kamera tamamen kapatılmıştır. Bu durumda da model “Kötü” olarak etiketlediği görülmektedir.



**Şekil 7.3: Web Uygulamasında Kötü Olarak Etiketlenen Fotoğraflar 2**

Şekil 7.3'te da çekilen fotoğraflara verilen “Kötü” etiketi gösterilmektedir. Bu fotoğraflarda ise cihazın net bir şekilde görünmediği, bulanık olduğu veya cihazın kameranın ortasında yer almadığı görülmektedir.



**Şekil 7.4: Web Uygulamasında İyi Olarak Etiketlenen Fotoğraflar**

Şekil 7.4'te çekilen fotoğraflara verilen “İyi” etiketleri gösterilmektedir. “Kötü” olarak etiketlenen fotoğrafların aksine burada “İyi Görünüyor” butonu gösterilmemiştir. Modelin tahminine güvenilmiştir. Fotoğraflar incelendiğinde cihazın net bir şekilde kameranın ortasında olacak şekilde çekildiği görülmektedir.

Model sayesinde kullanıcıların çekmiş olduğu fotoğrafların kalitesinin yüksek olması sağlanmıştır.

## 8. SONUÇ VE ÖNERİLER

Bu çalışma kapsamında, teknik servis süreçlerinde kullanılan akıllı cihaz fotoğraflarının kalite değerlendirmesini otomatik olarak gerçekleştirebilen YOLO ve ResNet18 tabanlı iki derin öğrenme modeli geliştirilmiştir. Modeller, fotoğraflardaki cihazın varlığı, netlik durumu, aydınlatma seviyesi ve kadraj uygunluğu gibi temel kriterleri dikkate alarak görselleri “iyi” veya “kötü” olarak sınıflandırmıştır. Gerçek veri seti üzerinde yapılan deneysel çalışmalar sonucunda YOLO modeli %93.5 doğruluk oranına ulaşılmış buna karşılık ResNet18 modeli ise %94.97 doğruluk oranına ulaşmıştır, her iki sınıf için de yüksek tespit başarısı elde edilmiştir.

Eğitim süresi açısından bakıldığında, ResNet18 modeli 45 dakikalık sürede eğitilerek YOLO’ya kıyasla (105 dakika) daha verimli bir süreç sunmuştur. Doğruluk, kesinlik, duyarlılık ve F1-Score değerlerinde de ResNet18’in YOLO’ya göre bir miktar üstünlük sağladığı görülmektedir. Elde edilen sonuçlar, ResNet18’in hem daha kısa sürede eğitilebilmesi hem de doğruluk ve diğer performans metriklerinde sağladığı üstünlüklerle YOLO’ya kıyasla daha verimli ve güvenilir bir model olduğunu işaret etmektedir.

Bununla birlikte, model boyutu açısından değerlendirildiğinde YOLO’nun önemli bir avantaj sunduğu söylenebilir. YOLO modeli yalnızca 5.9 MB büyüklüğe sahipken, ResNet18 modeli 42.6 MB boyutundadır. Bu durum, YOLO’yu özellikle bellek ve işlem kapasitesi sınırlı cihazlarda kullanım için daha uygun hale getirmektedir.

Genel olarak değerlendirildiğinde, ResNet18 modeli doğruluk ve performans metrikleri açısından daha yüksek başarı sunarken, YOLO modeli daha küçük boyutu sayesinde taşınabilirlik ve hafiflik yönünde öne çıkmaktadır. Dolayısıyla uygulama amacına bağlı olarak model tercihi farklılıklar söz konusu olabilir. Yüksek doğruluk ve güvenilirlik öncelikli olduğunda ResNet18 daha uygun bir seçim olurken, sınırlı kaynaklara sahip cihazlarda kullanım için YOLO daha avantajlı bir seçenek olarak öne çıktığı söylenebilir.

Aslan ve Özüpak'ın (2025) beyin tümörü tespitinde YOLOv8s-cls modeli ile elde ettikleri %98.7'lik doğruluk oranı veya Budak ve arkadaşlarının (2025) baskılı devre kartı kusur tespitinde YOLOv8 ile ulaştıkları yüksek doğruluk değerleri, nesne algılama ve sınıflandırma görevlerinde YOLO mimarilerinin güçlü performansını ortaya koymaktadır. Bu bağlamda, söz konusu performansın akıllı cihaz fotoğraf kalitesi değerlendirmesi gibi özgün bir endüstriyel senaryoya uyarlanabilmiş olması, çalışmanın literatüre katkısını artırmaktadır.

Yanılmaz ve arkadaşlarının (2024) güneş panellerindeki hotspot tespitinde YOLOv8 algoritmasının diğer modellere kıyasla daha yüksek özgüllük ve duyarlılık değerleri elde etmesi, modelin farklı veri türlerine uyarlanabilirliğini desteklemektedir. Benzer şekilde, Tbaileh'in (2023) e-ticaret ürün görsellerinin kalite değerlendirmesinde elde ettiği %98'lik doğruluk oranı, görüntü kalitesi analizinde derin öğrenme tabanlı yöntemlerin üstünlüğünü ortaya koymaktadır. Bu bağlamda, mevcut çalışmada ulaşılan sonuçlar, önceki araştırmaların bulgularıyla uyumlu olup, geliştirilen sistemin gerçek dünya koşullarında uygulanabilirliğini teyit etmektedir.

Araştırma bulgularının, şirket açısından önemli operasyonel kazanımlar sağlayacağı düşünülmektedir. Örneğin, düşük kaliteli görsellerin otomatik olarak tespit edilmesi, depolama maliyetlerinin düşmesini sağlayacaktır. Ayrıca, teknik servis çalışanlarının doğrulama süreçleri hızlanacak, iş gücü verimliliği artacaktır. Geliştirilen web arayüzü aracılığıyla toplanacak yeni veri setleri kullanılarak, modelin yeniden eğitilmesi ve böylece doğruluk oranının zamanla artırılması planlanmaktadır.

Toplumsal açıdan bakıldığında geliştirilen model, sağlık, güvenlik, e-ticaret ve üretim gibi farklı alanlarda benzer kalite kontrol süreçlerine uyarlanabilir. Böylece, otomatik kalite değerlendirme sistemleri insan hatasını azaltarak süreç güvenilirliğini artırarak hizmet kalitesinin yükselmesine katkıda bulunacaktır.

Gelecek çalışmalar açısından, modelin yalnızca kalite sınıflandırması yapmakla sınırlı kalmayıp, kaliteyi olumsuz etkileyen spesifik faktörleri (bulanıklık, aşırı aydınlatma, yanlış kadraj vb.) ayrı kategoriler halinde tespit edebilmesi hedeflenebilir. Ayrıca, farklı marka ve modellerin yanı sıra çeşitli ışık koşullarını kapsayan daha geniş bir veri seti ile modelin genelleme kabiliyeti artırılabilir. Kullanıcıya fotoğraf çekimi sırasında gerçek zamanlı geri bildirim sağlayacak etkileşimli bir modül eklenmesi, sistemin kullanıcı deneyimini geliştirebilir. Bunun

yanında, modelin mobil cihazlar ve gömülü sistemler için optimize edilmesi, saha koşullarında yaygın kullanımını mümkün kılabilir. Son olarak, benzer otomatik kalite kontrol yaklaşımlarının sağlık, tarım ve endüstriyel üretim gibi farklı alanlara uyarlanması hem akademik hem de toplumsal fayda alanını genişletecektir.

Sonuç olarak, bu çalışma, derin öğrenme tabanlı bir görüntü kalite değerlendirme sistemi olarak hem akademik literatüre hem de endüstriyel uygulamalara katkı sağlayan, kayda değer bir doğruluk oranına sahip, ölçeklenebilir ve sürekli öğrenebilen bir çözüm ortaya koymuştur. Literatürdeki benzer çalışmalarla uyumlu olarak elde edilen yüksek performans, geliştirilen modelin yalnızca mevcut problem özelinde değil, farklı sektörlerde de etkin biçimde kullanılabileceğini göstermektedir.



## KAYNAKLAR

- Akar, F., & Akgül, İ.** (2022). Derin Öğrenme Modeli ile Yüz İfadelerinden Duygu Tanıma. *Journal of the Institute of Science and Technology*, 12(1), 69-79.
- Ali, H., Kabir, S., & Ullah, G.** (2021). Indoor scene recognition using ResNet-18. *International Journal of Research Publications*, 69(1), 7.
- Akın, E., & Şahin, M. E.** (2024). Derin öğrenme ve yapay sinir ağı modelleri üzerine bir inceleme. *EMO Bilimsel Dergi*, 14(1), 27-38.
- Altay, A., & Yılmaz, S.** (2023). YOLO Algoritması Kullanılarak T Hücrelerinin Sınıflandırılması. *İleri Mühendislik Çalışmaları ve Teknolojileri Dergisi*, 3(2), 66-81.
- Arabacı Hayalioğlu, B.** (2023). “Doğal ve kültürel peyzajlarda görsel kalite analizi: Aydın ili örneği”. Yüksek Lisans Tezi, *Aydın Adnan Menderes Üniversitesi, Fen Bilimleri Enstitüsü*. Aydın.
- Arı, A., & Hanbay, D.** (2019). Bölgesel evrişimsel sinir ağları tabanlı MR görüntülerinde tümör tespiti. *Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi*, 34(3), 1395-1408.
- Aslan, E., & Özüpak, Y.** (2025). Performance Comparison of Deep Learning Models in Brain Tumor Classification. *Balkan Journal of Electrical and Computer Engineering*, 13(2), 203-209.
- Ayyachamy, S., Alex, V., Khened, M., & Krishnamurthi, G.** (2019, March). Medical image retrieval using Resnet-18. *In Medical imaging 2019: imaging informatics for healthcare, research, and applications* (Vol. 10954, pp. 233-241). SPIE.
- Balcıoğlu, Y., S.** (2023). “Derin Öğrenme ile Görüntü İşleme: Endüstriyel Parça Üretiminin Derin Öğrenme Sayesinde Yüzey Hata Tespiti (Görüntüleme Üzerine Uygulama)”. Doktora Tezi, *Gebze Teknik Üniversitesi, Sosyal Bilimler Enstitüsü*. Kocaeli.
- Budak, İ., Bal, S., & Korkmaz, H.** (2025). PCB Üretiminde Çok Sınıflı Kusur Tespiti için YOLO Tabanlı Derin Öğrenme Modeli. *Afyon Kocatepe Üniversitesi Fen ve Mühendislik Bilimleri Dergisi*, 25(4), 816-826.
- Boesch, G.** (2024). YOLO Explained: From v1 to Present. Erişim Adresi (07.06.2025): <https://viso.ai/computer-vision/yolo-explained/>
- Boesch, G.** (2024). YOLO11: A New Iteration of “You Only Look Once”. Erişim Adresi (07.06.2025): <https://viso.ai/computer-vision/yolov11/>
- Cekic, I., & Cavdar, K.** (2022). Detection of the cracks in metal sheets using convolutional neural network (CNN). *Journal of the Faculty of Engineering and Architecture of Gazi University*, 38(1).

- Chen, W., Huang, H., Peng, S., Zhou, C., & Zhang, C.** (2021). YOLO-face: a real-time face detector. *The Visual Computer*, 37(4), 805-813.
- Chollet, F.** (2021). Python ile derin öğrenme. *Buzdağı Yayınevi*, Ankara.
- Colab.** (Tarih Yok). What is Google Colab? Erişim Adresi (12.06.2025) <https://colab.research.google.com/>
- Demir, A., Yilmaz, F., & Kose, O.** (2019, October). Early detection of skin cancer using deep learning architectures: resnet-101 and inception-v3. In *2019 medical technologies congress (TIPTEKNO)* (pp. 1-4). IEEE.
- Dharneeshkar, J., Aniruthan, S. A., & Karthika, R.** (2020, February). Deep Learning based Detection of potholes in Indian roads using YOLO. In *2020 international conference on inventive computation technologies (ICICT)* (pp. 381-385). IEEE.
- Dobilas, S.** (2022). *Convolutional Neural Networks Explained – How to Successfully Classify Images in Python. Towards Data Science*. Erişim Adresi (08.05.2025): <https://towardsdatascience.com/convolutional-neural-networks-explained-how-to-successfully-classify-images-in-python-df829d4ba761>
- Doğan, F., & Türkoğlu, İ.** (2018). Derin öğrenme algoritmalarının yaprak sınıflandırma başarımlarının karşılaştırılması. *Sakarya University Journal of Computer and Information Sciences*, 1(1), 10-21.
- Efe, M., & Cangır, A. K.** (2022). Yapay Zekâ, Makine Öğrenmesi ve Tıp Uygulamaları. *Journal of Ankara University Faculty of Medicine/Ankara Üniversitesi Tıp Fakültesi Mecmuası*, 75.
- El Naqa, I., & Murphy, M. J.** (2015). What is machine learning?. In *Machine learning in radiation oncology: theory and applications* (pp. 3-11). Cham: Springer International Publishing.
- Fetzer, J. H.** (1990). What is artificial intelligence?. In *Artificial intelligence: Its scope and limits* (pp. 3-27). Dordrecht: Springer Netherlands.
- Firestore.** (Tarih Yok). Firestore Realtime Database. Erişim Adresi (11.06.2025): <https://firebase.google.com/docs/database?hl=tr>
- Fukushima, K.** (1980). Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological cybernetics*, 36(4), 193-202.
- GeeksforGeeks.** (2025). Firestore – Introduction. Erişim Adresi (11.06.2025): <https://www.geeksforgeeks.org/firebase-introduction/>
- GeeksforGeeks.** (2025). ResNet18 from Scratch Using PyTorch. Erişim Adresi (30.08.2025): <https://www.geeksforgeeks.org/deep-learning/resnet18-from-scratch-using-pytorch/>
- GeeksforGeeks.** (2025). What is YOLOv11? Erişim Adresi (12.06.2025): <https://www.geeksforgeeks.org/what-is-yolov11/>
- Ghosal, P., Nandanwar, L., Kanchan, S., Bhadra, A., Chakraborty, J., Nandi, D.** (2019, February). Brain tumor classification using ResNet-101 based squeeze and excitation deep neural network. In *2019 Second*

*International Conference on Advanced Computational and Communication Paradigms (ICACCP)* (pp. 1-6). IEEE.

- Goodfellow, I., Bengio, Y., & Courville, A.** (2018). Derin öğrenme. *Basım, Eryaman, Ankara: Buzdağı Yayınevi.*
- Gökalp, S., & Aydın, İ.** (2021). Farklı Derin Sinir Ağı Modellerinin Duygu Tanımadaki Performanslarının Karşılaştırılması. *Muş Alparslan Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi*, 2(1), 35-43.
- Hansen, D. K., Nasrollahi, K., Rasmussen, C. B., & Moeslund, T. B.** (2017). Real-time barcode detection and classification using deep learning. In *International Joint Conference on Computational Intelligence* (pp. 321-327). SciTePress.
- Hinton, G.E. & Salakhutdinov, R.** (2006). Reducing the Dimensionality of Data with Neural Networks, *Science* (80-. ), vol. 313, no. 5786, pp. 504–507.
- Hinton, G.E. & Salakhutdinov, R.,** (2009). Replicated Softmax: an Undirected Topic Model, in *Advances in Neural Information Processing Systems 22*, pp. 1607 1614.
- Hochreiter, S. & Schmidhuber, J.** (1997). Long Short-Term Memory. *Neural Computation*, 9(8), 1735–1780. <https://doi.org/10.1162/neco.1997.9.8.1735>
- Hubel, D. H., & Wiesel, T. N.** (1959). Receptive fields of single neurones in the cat's striate cortex. *J physiol*, 148(3), 574-591
- Ivakhnenko, A. G., & Lapa, V.G.** (1965). *Cybernetic predicting devices*. N.Y. CCM Information Corp.
- İmamoğlu, N., Eresen, A., & Efe, M. Ö.** (2009). Döner Ters Sarkaç Sisteminin Pekıştirmeli Öğrenme Algoritmaları ile Kontrolü. *Otomatik Kontrol Türk Millî Komitesi, Otomatik Kontrol Ulusal Toplantısı TOK09.*
- Jain, N., & Peddi, P.** (2023, October). Gender classification model based on the resnet 152 architecture. In *2023 IEEE International Carnahan Conference on Security Technology (ICCST)* (pp. 1-7). IEEE.
- Karakan, A., & Oğuz, Y.** (2024) Derin Öğrenme Tabanlı İşyeri Kamerası ile Gerçek Zamanlı Kişisel Koruyucu Ekipman ve Depo Güvenliği Tespiti. *Adıyaman Üniversitesi Mühendislik Bilimleri Dergisi*, 11(24), 402-414.
- Kayaalp, K. & Süzen, A. A.** (2018). Derin Öğrenme. *Derin Öğrenme ve Türkiye'deki Uygulamaları, Adıyaman, Türkiye: İKSAD Yayınevi*, 25-28.
- Kılıç, K., Özcan, U., Kılıç, K., & Dogru, İ.** (2024). Using deep learning techniques furniture image classification. *Politeknik Dergisi*, 27(5), 1903-1911.
- Krizhevsky, A., & Hinton, G.** (2010). Convolutional deep belief networks on cifar-10. *Unpublished manuscript*, 40(7), 1-9.
- Liang, J.** (2020, September). Image classification based on RESNET. In *Journal of Physics: Conference Series* (Vol. 1634, No. 1, p. 012110). IOP Publishing.

- Liu, C., Tao, Y., Liang, J., Li, K., & Chen, Y.** (2018, December). Object detection based on YOLO network. In *2018 IEEE 4th information technology and mechatronics engineering conference (ITOEC)* (pp. 799-803). IEEE.
- McCarthy, J.** (2007). What is artificial intelligence.
- McCulloch, W. S., & Pitts, W.** (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5, 115-133.
- Nabiyev, V. V.** (2012). *Yapay Zekâ*. Ankara: Seçkin Yayınları.
- Nikhileswara Rao, S.** (2024). YOLOv11 Architecture Explained: Next-Level Object Detection with Enhanced Speed and Accuracy. Erişim Adresi (07.06.2025): <https://medium.com/@nikhil-rao-20/yolov11>
- Nuxt.** (Tarih Yok). The Progressive Web Framework. Erişim Adresi (11.06.2025): <https://nuxt.com/>
- Okagawa, Y., Abe, S., Yamada, M., Oda, I., & Saito, Y.** (2022). Artificial intelligence in endoscopy. *Digestive Diseases and Sciences*, 67(5), 1553-1572.
- Orhan, H., & Arıcı, F. N.** (2024). Farklı YOLO Algoritmalarının Gerçek Zamanlı İHA Tespitine Etkileri ve Karşılaştırmalı Analizi. *Gazi Üniversitesi Fen Bilimleri Dergisi Part C: Tasarım ve Teknoloji*, 12(2).
- Özel, M. A., Gül, M. Y., & Güneş, E.** (2023). YOLO algoritması ile gastronomide yemek içeriklerini tespit etmede bir makine öğrenimi yaklaşımı. *Necmettin Erbakan Üniversitesi GastroMedia Dergisi (NEU GastroMedia)*, 2(1), 31-38.
- Patterson, J., & Gibson, A.** (2017). *Deep learning: A practitioner's approach*. "O'Reilly Media, Inc."
- Pekezou Fouopi, P., Srinivas, G., Knake-Langhorst, S., & Köster, F.** (2016). Object detection based on deep learning and context information. *Machine Learning Reports*.
- PromptLayer.** (Tarih Yok). ResNet-152. Erişim Adresi (30.08.2025): <https://www.promptlayer.com/models/resnet-152>
- Purwono, P., Ma'arif, A., Rahmian, W., Fathurrahman, H. I. K., Frisky, A. Z. K., & ul Haq, Q. M.** (2022). Understanding of convolutional neural network (cnn): A review. *International Journal of Robotics and Control Systems*, 2(4), 739-748.
- Rateke, T., Justen, K. A., & Von Wangenheim, A.** (2019). Road surface classification with images captured from low-cost camera-road traversing knowledge (rtk) dataset. *Revista de Informática Teórica e Aplicada*, 26(3), 50-64.
- Roboflow.** (2015). ResNet 34. Erişim Adresi (30.08.2025): <https://roboflow.com/model/resnet-34>
- Sayracı, B.** (2021, Ağustos 26). *Derin Öğrenme ve Evrimsel Sinir Ağları*. Retrieved Haziran 2, 2024, from Medium: <https://medium.com/deep-learning-from-deepest/derin-ogrenme-ve-evrimsel-sinir-aglari-446dc8a8d2f>

- Sinap, V.** (2024). Makine Öğrenmesinde Temel Yaklaşımlar ve Gelişen Eğilimler.
- Sofuoğlu, C., İ.** (2024). “Plant leaf disease detection and classification using deep learning”. Yüksek Lisans Tezi, *Dokuz Eylül Üniversitesi, Graduate School of Natural and Applied Sciences*. İzmir.
- Taigman, Y., Yang, M., Ranzato, M. & Wolf, L.** (2014). DeepFace: Closing the Gap to Human-Level Performance in Face Verification. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1701–1708. <https://doi.org/10.1109/CVPR.2014.220>
- Tbaileh, I.** (2023). *Use of a Deep Learning CNN Architecture in Product Image Quality Assessment for Improving E-Commerce Customer Experience*. Yüksek Lisans Tezi, Bahçeşehir Üniversitesi, İstanbul.
- Toğaçar, M., Ergen, B., & Sertkaya, M. E.** (2019). Zatürre Hastalığının Derin Öğrenme Modeli ile Tespiti. *Firat University Journal of Engineering Science*, 31(1).
- Ultralytics YOLO Docs.** (2025). Ultralytics YOLO11 Erişim Adresi (20.03.2025): <https://docs.ultralytics.com/tr/models/yolo11/>
- Ultralytics YOLO Docs.** (2025). Image Classification Erişim Adresi (10.02.2025): <https://docs.ultralytics.com/tasks/classify/>
- Ultralytics YOLO Docs.** (2025). Model Export with Ultralytics YOLO Erişim Adresi (10.08.2025): <https://docs.ultralytics.com/modes/export/>
- Vue.js.** (Tarih Yok). Introduction. Erişim Adresi (11.06.2025): <https://vuejs.org/guide/introduction.html#api-styles>
- Yanılmaz, S., Türkoğlu, M., & Aslan, M.** (2024). Güneş enerjisi santrallerinde YOLO algoritmaları ile hotspot kusurlarının tespiti. *Firat Üniversitesi Mühendislik Bilimleri Dergisi*, 36(1), 121-132.
- Yıldırım, B., & Cagıl, G.** (2020). Bir montaj parçasının derin öğrenme ve görüntü işleme ile tespiti. *Journal of Intelligent Systems: Theory and Applications*, 3(2), 31-37.
- Yılmaz, A., Kaya, U.** (2021). *Derin Öğrenme*, KODLAB yayınevi, Bağcılar/İstanbul, ISBN: 978-605-2118-39-9
- YoloSharp** (Tarih Yok). dme-compunet/YoloSharp: A high performance real-time object detection solution using YOLO11 powered by ONNX-Runtime. Erişim Adresi (25.07.2025): <https://github.com/dme-compunet/YoloSharp>
- Zhang X., Zhou X. and Lin M.** (2018.) Shufflenet: An extremely efficient convolutional neural network for mobile devices. *In Conference on Computer Vision and Pattern Recognition*, 6848-6856.

## ÖZGEÇMİŞ

Mehmet Emin EKER

### ÖĞRENİM DURUMU:

- **Lisans:** 2015, Sivas Cumhuriyet Üniversitesi, İktisadi ve İdari Bilimler Fakültesi, Yönetim Bilişim Sistemleri
- **Yüksek Lisans:** 2025, İstanbul Gedik Üniversitesi, Lisansüstü Eğitim Enstitüsü, Yapay Zekâ Mühendisliği Anabilim Dalı, Yapay Zekâ Mühendisliği Tezli Yüksek Lisans Programı

### MESLEKİ DENEYİM:

- Doğuş Teknoloji, Expert Software Developer

### TEZDEN TÜRETİLEN YAYINLAR:

- **Eker M.E.**, Bozkurt E., 2025: Akıllı Cihaz Görüntülerinin Kalitesini Değerlendirmek İçin Bir Derin Öğrenme Yaklaşımı. *Atlas 15. Uluslararası İleri Bilimsel Çalışmalar ve Disiplinlerarası Araştırmalar Kongresi*

### DİĞER YAYINLAR:

- Bağcı S., **Eker M.E.**, Ulusoy A., Suvay Eker H., Satış Sonrası Müşteri Destek Alanında Ağaç Yapılı Konu Modelleme Yöntemleri: Literatür Araştırması. *5th International Conference on Data Science and Applications (ICONDATA'22)*
- Suvay Eker H., **Eker M.E.**, 2021: Genetik Algoritma ile Çağrı Merkezi Çalışanları İçin Müşteri Atama Probleminin Optimizasyonu. *1. Uluslararası Yapay Zekâ ve Veri Bilimi Kongresi*