



T.C.

TOKAT GAZİOSMANPAŞA ÜNİVERSİTESİ

LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

YÜKSEK LİSANS PROGRAMI

**GÖRÜNTÜ İŞLEME VE DERİN ÖĞRENME KULLANARAK
YUMURTA TİPLERİNİN VE DURUMUNUN TESPİTİ**

YÜKSEK LİSANS TEZİ

Aya MOHAMMED HUSSEIN

Danışman: Doç.Dr.: Bülent TURAN

TOKAT- 2025

ETİK SÖZLEŞME

Tokat Gaziosmanpaşa Üniversitesi Lisansüstü Eğitim Enstitüsü tez yazım kılavuzuna göre, Doç.Dr.Bülent Turan danışmanlığında hazırlamış olduğum “Görüntü İşleme Ve Derin Öğrenme Kullanarak Yumurta Tiplerinin Ve Durumunun Tespiti” adlı Yüksek Lisans tezinin bilimsel etik değerlere ve kurallara uygun, özgün bir çalışma olduğunu, aksinin tespit edilmesi halinde her türlü yasal yaptırımını kabul edeceğimi beyan ederim

.../.../...

Aya MOHAMMED HUSSEIN

JÜRİ KABUL VE ONAY

Aya Mohammed Hussein tarafından hazırlanan **Görüntü İşleme Ve Derin Öğrenme Kullanarak Yumurta Tiplerinin Ve Durumunun Tespiti** adlı tez çalışmasının savunma sınavı, 12/08/2025 tarihinde yapılmış olup aşağıda verilen Jüri tarafından Oy Birliği ile Tokat Gaziosmanpaşa Üniversitesi Lisansüstü Eğitim Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı'nda Yüksek Lisans Tezi olarak kabul edilmiştir.

Jüri Üyeleri (Unvanı, Adı Soyadı)

İmzası

Üye (Başkan):

Üye

Üye

ONAY

...../...../.....

Lisansüstü Eğitim Enstitüsü Müdürü

TEŞEKKÜRLER

Öncelikle, bu görevi başarmam için bana verdiği güç için Allah'a hamd olsun. Tezime büyük bir özveriyle emek verdim. Ancak, çok sayıda kişinin cömert yardımını ve rehberliği olmadan bu tezin meyvesini vermesi mümkün olmazdı. Danışmanım Doç. Dr. Bülent Turan'a en içten teşekkürlerimi sunarım.



ÖZET

GÖRÜNTÜ İŞLEME VE DERİN ÖĞRENME KULLANARAK YUMURTA TÜRLERİ VE DURUMUNU TESPİT ETME

Aya, Mohammed Hussein

Yüksek Lisans, Lisansüstü Eğitim Enstitüsü, Bilgisayar Mühendisliği Anabilim Dalı

Tez Danışmanı: Doç.Dr. Bülent Turan

Ağustos 2025, xi+ 68 sayfa

Yumurtaların izlenmesi ve toplanması için bilgisayarla görme ve otomasyonun kullanılması, işgücü verimliliğinin artırılması için çok önemlidir. Derin öğrenme ve bilgisayarla görme teknikleri, araştırmacılar ve geliştiriciler tarafından çeşitli alanlarda yaygın olarak benimsenmiştir. Bu çalışma, YOLO V8 kullanan verimli ve hafif bir ağ modeli önermektedir. Bu tez, çeşitli yumurta türlerini ve bunların farklı koşullarını dikkate alan geliştirilmiş bir model sunmaktadır. Metodolojimiz üç ana aşamadan oluşmaktadır: ilk aşama görüntü kırpma dahil olmak üzere görüntü ön işleme ve büyütme tekniklerini içerir; ikinci aşama YOLOv8 tespit algoritmalarını kullanır; ve son aşama sınıflandırma doğruluğunu artırmak için derin öğrenme yaklaşımlarını uygular. Bunu, algılama ve sınıflandırmadaki algoritma performansının çeşitli kriterlere göre karşılaştırmalı bir analizi takip etmektedir. Literatürü incelediğimizde, bu alandaki araştırmaların çoğunun belirli bir tür veya sınırlı bir yumurta durumu (örneğin, sağlam veya kırık) üzerinde yoğunlaştığını görülmektedir (örneğin, tavuk veya ördek yumurtalarının sınıflandırılması). Bu çalışmada tavuk yumurtası (hem sağlam hem de kırık), ördek yumurtası (hem sağlam hem de kırık) ve bıldırcın yumurtası (hem sağlam hem de kırık) bir arada kullanılmıştır. Bu çalışma, modelin parametre sayısını ve hesaplama yükünü azaltırken yüksek tespit doğruluğunu korur. Bu yöntem, dağıtım masraflarını azaltır ve robotik platformlar için uygunluğunu artırır. Bu çalışmada, altı yumurta türünü kategorize etmek için bir sınıflandırma modeli belirlenmiştir. Böylece CNN modelinin performansı diğer modeller ile karşılaştırılmıştır. Karşılaştırmada kullanılan modeller “Random Forest”, ‘K-Nearest Neighbors (KNN)’, ve ‘ResNet50’ tir. Veri kümesi eğitim, doğrulama ve test kümeleri olarak düzenlenmiştir. Deneysel bulguların analizi, Evrimsel Sinir Ağının (CNN) performans açısından diğer modelleri geride bıraktığını göstermektedir. CNN modeli hafif mimarisi, minimum parametre sayısı ve hızlı performansı ile öne çıkmaktadır. Sistemin yumurta sınıflandırmasını da kapsayan tahmin

yetenekleri, işlendikten sonra yumurtaların sınıflandırma tipleri için %99 ve yumurtaların sınıflandırma durumları için %100 başarı elde edilmiştir.

Anahtar Kelimeler: Yumurta Sınıflandırma, Yolo v8, Derin Öğrenme, Tespit.



ABSTRACT

DETECTION EGG TYPES AND STATE USING IMAGE PROCESSING AND DEEP LEARNING

Aya, Mohammed Hussein

Master's Thesis, Institute of Graduate Studies, Department of Computer Engineering

Advisor: Assoc.Prof.Dr Bülent Turan

August 2025, xi+ 68 pages

The utilization of computer vision and automation for monitoring and collecting eggs is crucial for enhancing labor productivity. Deep learning and computer vision techniques have been extensively adopted by researchers and developers in diverse fields. This study proposes an efficient and lightweight network model utilizing YOLO V8. Rather than integrating many attributes simultaneously, the majority of research concentrate on a singular attribute, such as size or kind. This thesis presents an enhanced model that considers various egg types and their distinct conditions. Our methodology comprises three primary stages: the initial stage involves image preprocessing and augmentation techniques, including image cropping; the second stage employs YOLOv8 detection algorithms; and the last stage applies deep learning approaches to enhance classification accuracy. This is succeeded by a comparative analysis of algorithm performance in detection and classification according to diverse criteria. Upon examining the literature, we found that most research in this domain has concentrated on a particular type or a restricted range of egg states (e.g., intact or broken) (e.g., classification of chicken or duck eggs). This study involved the combination of chicken eggs (both intact and broken), duck eggs(both intact and broken), and quail eggs(both intact and broken). This study preserves elevated detection accuracy while diminishing the model's parameter count and computational burden. This method reduces deployment expenses and improves its suitability for robotic platforms. This study will identify a classification model to categorize six types of eggs and evaluate the performance of the CNN model against other models: “Random Forest”, “K-Nearest Neighbors (KNN)” and “ResNet50”. The dataset was organized into training, validation, and test sets. The analysis of the experimental findings indicates that the Convolutional Neural Network (CNN) surpasses the other models in performance. The CNN model is distinguished by its lightweight architecture, minimal parameter count, and rapid performance. The prediction capabilities of the system, including egg classification, are 99% for the classification types of eggs and 100% for the classification states of eggs after processing.

Keywords: Egg Classification ,Yolo v8 ,Deep Learning ,Detection.

TABLO LİSTESİ

Tablo No	Sayfa
Tablo 3.1: Gerçek ve tahmin edilen model için ikili karışıklık matrisi.....	12
Tablo 4.1: Yumurta veri seti.....	19
Tablo 4.2: Tavuk yumurtası	19
Tablo 4.3: Ördek yumurtası	20
Tablo 4.4: Bildircin yumurta.....	21
Tablo 4.5: Kullanılan CNN modelinin özeti.....	23
Tablo 5.1: İşlemeden önce yolov8 modellerinin eğitilmesi.....	27
Tablo 5.2: İşleme öncesi CNN modelini eğitme	29
Tablo 5.3: İşleme öncesi Rastgele Orman sınıflandırması	34
Tablo 5.4: İşleme öncesi K-En Yakın Komşular (KNN) sınıflandırması	36
Tablo 5.5: İşleme öncesi Resnet50 sınıflandırması	38
Tablo 5.6: Yolov8 modellerinin işlendikten sonra eğitilmesi	39
Tablo 5.7: İşlemden sonra CNN modelinin eğitimi	42
Tablo 5.8: İşleme sonrası Rastgele Orman sınıflandırması	47
Tablo 5.9: İşleme sonrası K-En Yakın Komşular (KNN) sınıflandırması	49
Tablo 5.10: İşleme sonrası Resnet50 sınıflandırması	51
Tablo 5.11: Karşılaştırma sonuçları	52
Tablo 5.12: Önceki çalışmalarla karşılaştırmalar	53

ŞEKİL LİSTESİ

Şekil No	Sayfa
Şekil3.1: Havuzlama katmanı	7
Şekil3.2: ResNet50 mimaris	9
Şekil3.3: Rastgele Ormanın genel mimaris.....	10
Şekil3.4: K-En Yakın Komşular (KNN)	11
Şekil3.5: Sigmoid fonksiyonunun eğrisi	15
Şekil3.6: ReLu fonksiyonunun eğrisi	15
Şekil3.7: YOLOv8 mimarisi	17
Şekil4.1: Veri seti görüntülerinin örneği	18
Şekil4.2: Tavuk yumurtası	20
Şekil4.3: Ördek yumurtası	21
Şekil4.4: Bildircin yumurtası	22
Şekil4.5: CNN mimarisi	22
Şekil4.6: Altı yumurta tipinde üç CNN modelinin eğitilmesi	24
Şekil4.7: Önerilen yumurta tipi ve durum sistemi yapısı	25
Şekil5.1: CNN modelinin karışıklık matrisi.....	28
Şekil 5.2: CNN modelinin çapraz doğrulaması.....	29
Şekil 5.3: CNN model eğitimi doğruluğu, doğrulama doğruluğu ve kayıp grafiği	30
Şekil5.4: Tavuk yumurtası testi	31
Şekil5.5: Bildircin yumurtası testi	32
Şekil5.6: Ördek yumurtası testi	32
Şekil5.7: Karışıklık matrisi - Rastgele Orman	33
Şekil 5.8: Rastgele Orman modelinin çapraz doğrulaması.....	34
Şekil 5.9: Rastgele Orman modelinin eğitimi ve değişimini grafiği.....	35
Şekil5.10: Karışıklık matrisi – KNN	35
Şekil5.11: K-En Yakın Komşular (KNN) modelinin çapraz doğrulaması.....	36
Şekil5.12: Karışıklık matrisi - Resnet50.....	37
Şekil5.13: Resnet50 modelinin çapraz doğrulaması.....	38
Şekil5.14: CNN modelinin karışıklık matrisi	41
Şekil5.15: CNN modelinin çapraz doğrulaması.....	42
Şekil5.16: CNN model eğitimi doğruluğu, doğrulama doğruluğu ve kayıp grafiği	43
Şekil5.17: Tavuk yumurtası testi	44
Şekil5.18: Ördek yumurtası testi	45
Şekil5.19: Bildircin yumurtası testi	45
Şekil5.20: Karışıklık matrisi - Rastgele Orman	46
Şekil5.21: Rastgele Orman modelinin çapraz doğrulaması.....	47
Şekil5.22: Rastgele Orman modelinin eğitimi ve değişimini grafiği.....	48
Şekil5.23: Karışıklık matrisi – KNN	48
Şekil5.24: K-En Yakın Komşular (KNN) modelinin çapraz doğrulaması.....	49
Şekil5.25: Karışıklık matrisi - Resnet50.....	50
Şekil5.26: Resnet50 modelinin çapraz doğrulaması.....	51

KISALTMALAR

Kısaltma	Açıklama
ADAM	Uyarlanabilir Moment Tahmini
AI	Yapay Zeka
CDBN	Evrişimsel Derin İnanç Ağları
CNN	Evrişimli Sinir Ağları
DCNN	Derin Evrişimli Sinir Ağları
DL	Derin Öğrenme
DLA	Derin Öğrenme Algoritmaları
DLN	Derin Öğrenme Ağı
DNN	Derin Sinir Ağı
Epoch	Özellikle Birinci Dönemde Değişim Olur
FC	Tamamen Bağlantılı
FN	Yanlış Negatif
FP	Yanlış Pozitif
HMM	Gizli Markov Modeli
IC	Bilgi Toplayıcı
IMAN	Bilgi Maksimizasyonu Uyum Ağı
K	Çekirdek
ML	Makine Öğrenmesi
MLP	Çok Katmanlı Perceptron
NN	Sinir Ağı
PPV	Pozitif Tahmini Değer
ReLU	Düzeltemeli Doğrusal Ünite
ResNet	Artık Ağ
RGB	Kırmızı Yeşil Mavi
MCC-CNN	Çok Sınıflı Sınıflandırma Evrişimli Sinir Ağı
RNN	Tekrarlayan Sinir Ağı
ROC	Alıcı Çalışma Karakteristik Eğrisi
TN	Gerçek Negatif
TP	Gerçek Pozitif
TPR	Gerçek Pozitif Oranı
VGG	Görsel Geometri Grubu
YOLO	Sadece Bir Kez Bakarsın

İÇİNDEKİLER

	Sayfa
ETİK SÖZLEŞME SAYFASI.....	i
JÜRI KABUL VE ONAY.....	ii
TEŞEKKÜR	iii
ÖZET (TÜRKÇE).....	iv
ABSTRACT(İNGİLİZCE).....	vi
TABLolar LISTESİ.....	vii
ŞEKİLLER LISTESİ.....	viii
KISALTMALAR.....	ix
1. GİRİŞ.....	1
1.1 Genel Bakış	2
1.2 Problem	2
1.3 Tez Amacı.....	2
1.4 Tezin Düzeni.....	2
2. LİTERATÜR ÖZETLERİ.....	3
3. DERİN ÖĞRENME.....	6
3.1 Evrişimsel Sinir Ağlar.....	7
3.1.1 Giriş katmanı.....	7
3.1.2 Havuzlama katmanı.....	7
3.1.3 Aktivasyon katmanı.....	8
3.1.4 Normalizasyon katmanı.....	8
3.1.5 Tam bağlantılı katman.....	8
3.1.6 Çıktı katmanı.....	8
3.2 Transfer öğrenme.....	9
3.3 ResNet 50.....	9
3.4 Rastgele orman.....	10
3.5 K-En Yakın Komşular (KNN).....	10
3.6 Karmaşa matrisi.....	11
3.6.1 Doğruluk.....	12
3.6.2 Yanlış sınıflandırma.....	13
3.6.3 Hatırlama.....	13
3.6.4 Kesinlik.....	13
3.6.5 Yanlış Pozitif Oranı.....	14
3.7 Aktivasyon fonksiyonu.....	14
3.7.1 Sigmoid.....	14
3.7.2 ReLu.....	15
3.7.3 Softmax.....	16
3.8 YOLO.....	16
3.8.1 YOLOv8 genel bakış.....	16
4. KULLANILAN MATERYAL YÖNTEMLER.....	18
4.1 Veri kümeleri.....	19
4.2 CNN yarış modellerinin eğitimi ve test edilmesi.....	22
4.3 CNN sistem yapısı.....	25
5. SONUÇLAR VE TARTIŞMA.....	26

5.1 İşlemeden önce eğitim.....	26
5.1.1 YOLO V8 modellerinin uygulama sonuçları.....	26
5.1.2 CNN modellerinin uygulama sonuçları.....	28
CNN modeli uygulaması.....	38
5.1.3 Rastgele Orman modeli uygulaması.....	33
5.1.4 K-En Yakın Komşular (KNN) modeli uygulaması...	35
5.1.5 ResNet50 model uygulaması.....	37
5.2 İşlemeden sonra eğitim.....	39
5.2.1 YOLO V8 modellerinin uygulama sonuçları.....	39
5.2.2 CNN modellerinin uygulama sonuçları.....	41
CNN modeli uygulaması.....	41
5.2.3 Rastgele Orman modeli uygulaması.....	46
5.2.4 K-En Yakın Komşular (KNN) modeli uygulaması....	48
5.2.5 ResNet50 mode uygulaması.....	50
5.3 Önceki çalışmalarla karşılaştırma.....	53
6. SONUÇ VE ÖNERİLER.....	54
6.1 Sonuç.....	54
6.2 Gelecekteki Çalışmalar İçin Öneriler.....	55
KAYNAKLAR.....	56

1. GİRİŞ

1.1 Genel Bakış

Tavukların yumurta üretimi, özellikle zemin yumurtalarının yaygın olduğu kafessiz sistemlerde, kırık yumurtalar nedeniyle azalır (Yang ve ark., 2023). Tüketici memnuniyetini değerlendirmek ve kalite notu oluşturmak için kritik bir faktör "Yumurta Boyutu"dur. Tek görünüm metrolojisi ve derin öğrenmeyi kullanan araştırma, yumurtaların ana ve yan eksenlerini ölçmeyi amaçlamaktadır (Liu ve ark., 2023). Bireylerin beslenme gereksinimleri, küresel nüfusun büyümesiyle birlikte artmıştır. Son yıllarda, önemli bir gıda bileşeni olan yumurta talebinde artış olmuştur. Sahtecilik, yanlış etiketleme ve ikame gibi olumsuz durumlara hazırlıklı olmak için bireyleri yumurta kalitesi hakkında bilgilendirmek esastır (Şehirli ve Arslan, 2022). Tüketiciler ve modern gıda işletmeleri, yumurtaların güvenliği ve kalitesi konusunda önemli endişeler taşımaktadır (Nasiri ve ark., 2020). Yumurta sınıflandırmasında kalite kontrolü için temel teknik, insanlar tarafından yapılan görsel incelemedir. Ayrıca yumurta kabukları, çatlaklar ve kirlenme gibi dış kusurlara ek olarak kan ve et lekeleri gibi iç düzensizlikler açısından da incelenir. Ancak arkadan aydınlatma altında, yumurtanın yüzeyindeki belirgin siyah alanlar yumurta kabuğundaki çatlakların tespitini engeller. Her şeritte 10.000 yumurta işleyebilen yüksek hızlı elektronik tasnif makinelerinin gelişmesi, insan eliyle yapılan tasnif işlemini önemli bir darboğaz haline getirmiştir. Son araştırmalar, yumurta kabuğu kırığı tespitinin otomasyonu için hem optik hem de mekanik tespit yöntemlerine odaklanmıştır (Nakamura ve Katayama, 2000). Çatlakların benzersiz olması gerekliliği, çatlak tespiti için görüntü tabanlı bir yaklaşımın kullanılmasının bir sınırlamasıdır (Wu ve ark., 2018). Yine de çatlakları tespit etmek için yeni bir makine görüşü tabanlı yaklaşım tanıtılmıştır (Bao ve ark., 2019). Derin öğrenme metodolojileri, geleneksel yöntemlerden farklıdır. Eş zamanlı olarak çıkarım ve sınıflandırma, geri yayılım, hata optimizasyonu ve tekrarlanan ağırlık ayarlamaları yoluyla makine öğrenme tekniklerine sahiptir; bunlar derin öğrenme tekniklerini kullanmanın bir diğer avantajı olan özellik çıkarma kabiliyetleri nedeniyle, sınıflandırıcının bazen insanlar tarafından gözden kaçan temel ve beklenmedik özellikleri içerebilmesidir. Bu prosedüre özellik öğrenmesi denir (Shin ve ark., 2016). Derin öğrenme modellerindeki son gelişmeler, birden fazla alanda başarılı görüntü sınıflandırmasını sağlamıştır. Bu modellerde birkaç hiper parametrenin kesin olarak tanımlanması esastır (Kaya ve ark., 2023). Evrişimli Sinir Ağları (CNN'ler), derin öğrenme modelleri için temel görevi görür. Temelde dört katmandan oluşur. Bunlara "Evrişimli Katman", "ReLU Katmanı", "Havuzlama Katmanı" ve "Tam Bağlantılı Katman" denir. Derin öğrenme modelleri, bu katmanların değişen miktarları kullanılarak oluşturulur. (İnik ve Turan, 2018)

1.2 Problem

Son yıllarda, bilgisayarlı görüş ve derin öğrenme teknolojilerindeki muazzam gelişmeler, hem geleneksel hem de otomatik sistemler dahil olmak üzere yumurta tanıma sistemleri için yolu kolaylaştırdı. Yumurta tanıma teknolojisinin çeşitli uygulamaları arasında, endüstri ve bilimde operasyonel performansı iyileştirme gibi birçok alanda geniş etkisi nedeniyle yumurta durumunu ve türünü belirlemek son derece önemlidir. Kümes hayvanı çiftçilerine üretim verimliliğini artırmak, sınıflandırma hatalarını azaltmak, sıralama sürecini kolaylaştırmak ve insan kaynaklarına olan bağımlılığı azaltmak için yeni bir araç sağlar. Ancak, yumurta türü tanıma için çok sayıda veri kümesi mevcut olsa da, yalnızca birkaç veri kümesi hem yumurta durumunu hem de türünü aynı anda ele alır. Bu modeller, her dönem boyunca çok sayıda parametrenin ağırlığını değerlendirir ve gerçek zamanlı ve çevrimiçi işlemede sıklıkla zorluklarla karşılaşır. Bu nedenle, "Önceden eğitilmiş CNN" modelleri, özellikle hızlı karar almanın kritik olduğu durumlarda, binlerce kategorideki milyonlarca görüntü üzerinde eğitim ve değerlendirmeye tabi tutulmuştur.

1.3 Tez Amacı

1. Farklı yumurta türü veri kümelerinden yeni ve dengeli veri kümesi toplamak.
2. Yolo v8 kullanarak görüntülerden yumurtaları tespit etmek için bir model oluşturmak ve ardından CNN ve derin öğrenme kullanarak bunları kategorilere göre sınıflandırmak. Yumurta sınıflandırması için kullanılan sinir ağı performansı ve zamanı iyileştirmek için minimum sayıda parametreyle altı kategoriyi tanıır: tavuk yumurtaları (sağlam ve kırık), ördek yumurtaları (sağlam ve kırık) ve bıldırcın yumurtaları (sağlam ve kırık).
3. Yumurta türlerini ve koşullarını tanımak için "Random Forest", 'K-Nearest Neighbors (KNN)' ve 'ResNet50', modellerini eğitmek ve test etmek: tavuk yumurtaları (sağlam ve kırık), ördek yumurtaları (sağlam ve kırık)ve bıldırcın yumurtaları(sağlam ve kırık).
- 4.Beş modelin performansını aynı veri kümesinde karşılaştırmak.

1.4 Tezin Düzeni

2. Bölüm: Literatür İncelemesi.
3. Bölüm: Materyal ve Metod bölümünü içerir.
4. Bölüm: Sunulan şemanın ayrıntılı tasarımını açıklar.
5. Bölüm: 3. Bölümde önerilen algoritmalar kullanılarak yürütülen deneylerin sonuçlarını tartışır ve sunar.
6. Bölüm: Sunulan yaklaşımın sonuçlarını ve araştırma önerilerini sunar.

2. LİTERATÜR ÖZETLERİ

Son zamanlarda, derin öğrenme ve bilgisayarlı görme alanındaki araştırma çabaları, özellikle yumurta türü tanıma için teknik yetenekler geliştirmeye odaklanarak dikkate değer bir büyüme göstermiştir. Tüm çalışmalar, tanıma sistemlerinin performansını ve doğruluğunu iyileştiren teknikler geliştirmeyi amaçlamaktadır. Bu bölüm, her alanda kullanılan algoritmalar ve teknikler dahil olmak üzere yumurta durumu ve tür tanıma ile ilgili literatürü ve araştırma çalışmalarını inceler. Bölüm ayrıca yumurta tanıma tekniklerini entegre eden mevcut çalışmaları da kapsar. Çeşitli çalışmalar aşağıda gösterilmiştir:

Bao G ve ark., (2019), bilgisayarlı görüş yöntemlerini kullanarak yumurta kabuklarındaki çatlakları tanımlamak için yeni bir yöntem önerdiler. Bu çalışma, görüntüleme sırasında aydınlatmanın neden olduğu koyu lekeler nedeniyle tanımlanması zor olan yumurta kabuklarındaki küçük çatlakları tanımlama sorununu çözmeyi amaçlamıştır. Yumurta fotoğraflarındaki çatlakların daha iyi görünmesi için araştırmacılar negatif bir LOG (Gaussian Laplasyen) filtresi uyguladılar. Ayrıca gürültünün etkisini azaltmak için ek yöntemler kullandılar. Kolayca görünmeyen koyu lekeler ve minik çatlaklara rağmen deneyler sistemin %92.5 tespit oranına sahip olduğunu ortaya koydu.

Wu L ve ark., (2018), görüntü işleme teknikleri (Soft-Margin SVM) aracılığıyla çatlak yumurtaları tanımlamak için bir çalışma yürüttüler. Bu çalışma, yumurta kabuklarındaki çatlakları tespit etmek için hassas ve etkili bir sistem oluşturmayı amaçladı. İki yüz kahverengi yumurta kullanıldı. Araştırmacılar, çatlakları görsel olarak tanımlamak için özel bir görüntüleme teknolojisi kullandılar. Kenarları tanımlamak için çeşitli algoritmalar kullanıldı. Sınıflandırma prosedürü, yumurtaları kategorize etmek için Soft-Margin SVM algoritmasını kullandı ve eğitim seti için %94 ve test seti için %93 sınıflandırma doğruluğu elde etti.

Jiang ve ark., (2024), bilgisayarlı görüş metodolojileriyle güvercin yumurtalarını tanımlamak için kolaylaştırılmış bir metodoloji önerdiler. Bu çalışma, parametreleri ve hesaplama karmaşıklığını en aza indirirken gerçek ve sahte güvercin yumurtalarını tanımak için etkili bir model oluşturmayı ve dolayısıyla modelin mobil robotik sistemlerde uygulanmasını sağlamayı amaçlamaktadır. YOLOv8n mimarisi, bir örnek modülünün tanıtımıyla birlikte Verimli Çok Ölçekli Dikkat (EMA) mekanizmasını da içerecek şekilde geliştirildi. Kayıp fonksiyonunu geliştirmek için yöntemler kullandılar. Deneysel sonuçlar, geliştirilmiş modelin (YOLOv8-PG) temel YOLOv8n modeline kıyasla üstün performans ölçütlerine ulaştığını gösterdi.

Pasco Sánchez J. M. ve ark., (2023) , kümes hayvanı yumurtalarını boyut ve şekil gibi fiziksel özelliklere göre sınıflandıran bir sistem oluşturarak sıralama işlemlerinin verimliliğini artırdılar ve manuel incelemeye olan bağımlılığı azalttılar. Modeli eğitmek için evrişimli sinir ağlarını kullanarak bilgisayarlı görüş ve derin öğrenme metodolojilerini kullandılar ve bu da yüksek doğrulukla sonuçlandı. VGG19 modeli kullanılarak %97.33'lük bir performans elde edildi.

Sehirli, E ve Arslan, K. (2023), yumurta ağırlığı, yumurta beyazı yüksekliği, yumurta sarısı ağırlığı, yumurta gücü, yumurta kabuğu kalınlığı ve çeşitli endeksler dahil olmak üzere 438 tavuktan alınan yumurtalara ilişkin 20 değişkenden oluşan bir veri setini incelediler. Bu çalışma, tavuk çiftliklerindeki yumurta sıralama prosedürlerini iyileştirdi ve elle incelemeye olan bağımlılığı azalttı. Araştırmacılar, "Karar Ağacı (DT)", "Doğrusal Ayrımcı Analiz (LDA)", "Lojistik Regresyon (LR)", "Naive Bayes (NB)", "Destek Vektör Makineleri (SVM)", "K-En Yakın Komşular (KNN)", "Rastgele Orman (RF)" ve "Yapay Sinir Ağları (ANN)" dahil olmak üzere çeşitli makine öğrenme yöntemlerini kullandılar. Bulgular, "Rastgele Orman (RF)"nin sınıflandırma kalitesi açısından %98.6'lık en büyük sınıflandırma doğruluğuna ulaşmasıyla olumlu sonuçlar gösterdi.

Milovanovic B ve ark., (2021), yumurta kabuklarının, sarılarının ve beyazlarının rengini analiz etmek için klasik Minolta kolorimetresine kıyasla bilgisayarlı görüş sisteminin doğruluğunu değerlendirmek üzere bir çalışma yürüttüler. Beş farklı yumurta türünün tonu ölçüldü: "tavuk", "kaz", "ördek", "bıldırcın" ve "hindi". Bilgisayarlı görüş sistemi (CVS) ile geleneksel Minolta kolorimetresinin etkinliği arasında karşılaştırmalar yapıldı. Yumurta, sarısı ve beyazının ölçümü yapıldı ve bulgular sonuçlarda derlendi. Bilgisayarlı görüş sistemi (CVS) %75 ile %100 arasında bir doğruluk aralığı gösterdi, ancak Minolta kolorimetresinin performansı ölçülen değerler ile numunelerin gerçek renkleri arasında önemli tutarsızlıklar olduğunu gösterdi.

Yang X ve ark., (2023), Yumurta kategorizasyonunu ve hata tanımlamasını geliştirmek için derin öğrenme ve makine öğrenme metodolojilerini entegre eden iki kademeli bir model geliştirdiler. Birincil ve ikincil eksenler dahil olmak üzere alınan özelliklere dayalı olarak yumurta ağırlığını tahmin etmek için Rastgele Orman yöntemleri kullanıldı. %94.8'e ulaşan doğrulukla olumlu sonuçlar elde edildi.

Kaewtapee, C. ve Supratak, A. (2015) yaptıkları çalışmada, insan incelemesine olan bağımlılığı en aza indirmek için bir yumurta sarısı değerlendirme sistemi oluşturmayı amaçlamaktadır. Bu teknik üretim kalitesini artırmayı hedeflemektedir. Dijital bir kamera aracılığıyla yumurta sarılarının fotoğraflarını aldılar ve dönüşümler ve diğerleri de dahil olmak üzere görüntü işleme tekniklerini kullandılar. Tahmini metodolojileri, özellikle yapay sinir ağlarını ve derin öğrenme yaklaşımlarını kullandılar. Doğruluk %97,1'e ulaştı.

Ab Nasir ve ark., (2018), eski yumurta işleme tesislerinde, yumurtalar sıklıkla yumurta işleme makinelerini denetleyen personel tarafından elle sınıflandırılırdı. Üretim hattından geçen yumurtalar, elektromekanik kayıtlar kullanan insanlar tarafından işaretlenir ve daha sonra ayırma makineleri tarafından ayrılırdı. Tavuk yumurtalarının hassas otomatik değerlendirmesi, yumurta üretim sürecini geliştirmek ve hızlandırmak için önemlidir. hijyenik bir üretim ortamını sürdürmek için bu da dikkate değerdir.

Mahmoud (2015), yumurta üretiminde bazı nedenlerden dolayı kalitede düşüş meydana gelmektedir. Çünkü depolama sırasında tavuk yumurtalarında fizikokimyasal değişiklikler meydana gelmektedir. Bu çalışmada yapay sinir ağları (ANN), bayes ağları (BN), karar ağaçları (DT) ve destek vektör makineleri (SVM) dahil olmak üzere tazelik tespiti için çeşitli makine öğrenimi (ML) metodolojileri oluşturulmuştur. Tüm BN çeşitleri olağanüstü sonuçlar göstermiştir.

Caner ve Yüceer (2014), yaptığı çalışmada tüm kaplanmış yumurta kabukları, kaplanmamış yumurtalara kıyasla üstün delme mukavemeti sergilemiştir, bu nedenle yumurta kabuklarının kaplanması depolama sırasında raf ömrünü uzatabilir. Çalışmada elde edilen sonuçlara göre kaplamasız yumurtalar en yüksek albümin pH'ına, en büyük ağırlık kaybına ve en düşük albümin viskozitesine sahipti. Depolama süresi ve kitosan kaplamaları Haugh birimini, sarısı indeksini, ağırlık kaybını, albümin pH'ını, kuru maddeyi, bağıl çırpma kapasitesini (RWC) ve albümin viskozitesini önemli ölçüde etkiledi.

3. DERİN ÖĞRENME

3.1 Evrişimsel Sinir Ağlar

Derin öğrenme yaklaşımları, özellik çıkarma ve sınıflandırmayı eş zamanlı olarak yürüterek sınıflandırıcının insanların gözden kaçırabileceği temel ve beklenmedik özellikleri dahil etmesini sağlayarak geleneksel makine öğrenme algoritmalarından farklılık gösterir. Derin öğrenme tekniklerini kullanmanın avantajı, geri yayılım ve hata optimizasyonu ve yinelemeli ağırlık ayarlamaları yoluyla özellikleri toplamasıdır. Bu prosedüre özellik öğrenmesi denir (Shin ve ark., 2016). Evrişimsel sinir ağı (CNN), özellikle görüntü işleme ve nesne tanıma için geliştirilen derin sinir ağlarının bir alt kümesidir. CNN algoritmaları, internette erişilebilir olan muazzam miktardaki bilgi nedeniyle kolayca kullanılabilir. Giriş verilerini türetmek ve özellikleri ayırt etmek için yalnızca matematiksel bir çerçeveye güvenirlir. Görüntünün önceki özelliklerini belirlemek için çok az gereklilik olmadan kullanılabilirler (Handalage, 2021). Geleneksel CNN tabanlı sınıflandırma modelleri sıralı olarak yapılandırılmaktadır ve modelin tepesinde herhangi bir çıktı dallanması olmadan tekil bir tahmin üretir (Zhu ve Bain, 2017). Bir CNN temel olarak iki bileşenden oluşur: "evrişimsel katmanlar" ve sonraki "tam bağlı katmanlar". Evrişimsel katmanlar kayan pencere yaklaşımı kullanarak işlev görür ve aktivasyonların mekansal yapılandırmasını gösteren özellik haritaları üretir. Evrişimsel katmanlar önceden belirlenmiş bir görüntü boyutu gerektirmez ve değişen boyutlarda özellik haritaları üretebilir. Tersine, tam bağlı katmanlar tanım gereği sabit boyutlu bir girdi gerektirir. Sonuç olarak, sabit boyut sınırlaması yalnızca ağı daha derin bir seviyesinde bulunan tam bağlı katmanlardan kaynaklanır (He ve ark., 2014). CNN'ler özellikle 2B görüntü tabanlı eğitim veri kümelerini işlemede uzmandır (Shin, Kim ve Kwon, 2016). Derin öğrenme modellerindeki son gelişmeler, birden fazla alanda başarılı görüntü sınıflandırmasına yol açmıştır. Bununla birlikte, bu modellerde çeşitli hiper parametrelerin kesin olarak tanımlanması esastır. (Kaya ve ark., 2023). Evrişimsel Sinir Ağları (CNN'ler) derin öğrenme modellerinin temelini oluşturur. Temel olarak dört katmandan oluşur. Bunlar "Evrişimsel Katman", "ReLU Katmanı", "Havuzlama Katmanı" ve "Tam Bağlantılı Katman"dır. Derin öğrenme modelleri, bu katmanların değişen miktarları kullanılarak oluşturulur (İnik ve Turan, 2018). Geleneksel bir CNN modeli, bir "giriş katmanı", "aktivasyon katmanı", "normalizasyon katmanı" ve nihayetinde bir "çıkış katmanı" dahil olmak üzere birden fazla katmandan oluşur (Handalage, 2021; Al-Gaashani ve ark.,2024).

Derin sinir ağları (DNN'ler), çeşitli bilimsel disiplinlerde kullanılan en umut verici metodolojilerden birini temsil eder. Evrişimli sinir ağı (CNN) olarak bilinen derin sinir ağlarının belirli bir çeşidi, her biri bir aktivasyon fonksiyonu ve bir havuzlama katmanı (Mou ve Zhang, 2022) tarafından takip edilen birden fazla evrişimli katmandan oluşur.

CNN katmanları aşağıdaki gibidir (Handalage, 2021):

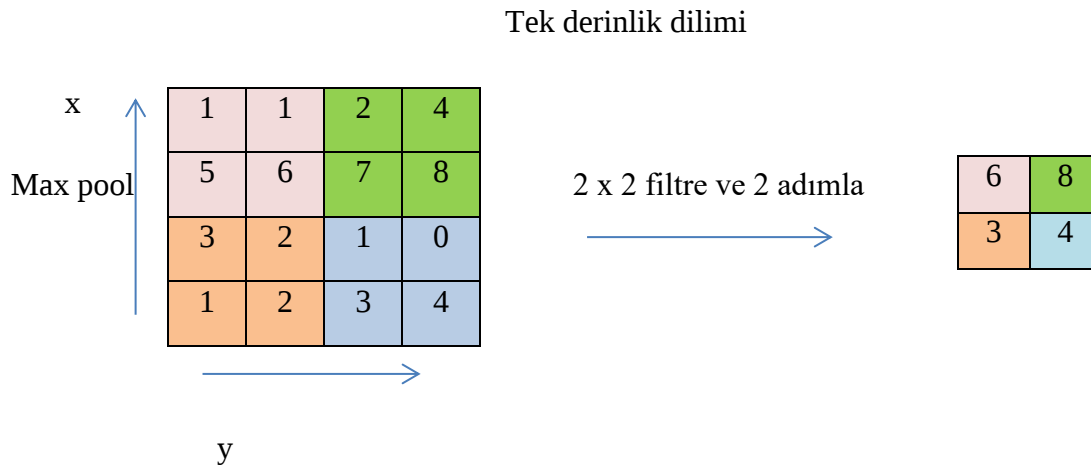
3.1.1 Giriş katmanı:

Giriş katmanı, giriş görüntü verilerini başlatır ve tüm boyutları sıfır civarında normalleştirir. Bu katman, tüm giriş verilerini "0" ile "1" arasındaki bir aralığa normalleştirerek daha hızlı yakınsamayı kolaylaştırır. Bu normalleştirme, verileri aydınlatarak yedekliliği azaltmaya yardımcı olur (Handalage, 2021).

Evrışimsel katman, giriş görüntüsüne göre aynı boyutlara sahip ancak azaltılmış sabit parametre değerine sahip bir çekirdek filtresi kullanarak giriş görüntülerinin evrişimini gerçekleştirir ve böylece önemli özellikleri çıkarır (Khan ve Aslam, 2022; Singh ve Meitei, 2020). Her filtre, belirli bir matrisle hizalanarak giriş görüntüsünde evrişim gerçekleştirir. Bu evrişim, özellik haritaları üretir (Ashar ve ark., 2024).

3.1.2 Havuzlama katmanı

Çeşitli vaka çalışmalarında optimum havuzlama yöntemini belirlemek zordur. Uzmanlar, derin öğrenme ilkelerine uymaları koşuluyla, görüntü işleme senaryolarında en etkili havuzlama türlerini uygulayabilirler. Dinamik sınır ağı tasarımı, katmanlar için uygun bir havuzlama mekanizması belirlemek aslında mümkün değildir. Kısıtlamaları gidermek için, hem maksimum havuzlamayı hem de ortalama havuzlamayı aşan ideal bir havuzlama stratejisi geliştirmek gerekir (Huang ve Wanda, 2020). Bu katman, girdinin mekansal boyutlarını önemli ölçüde azaltarak tutarlı bir şekilde iki temel hedefe ulaşır. Başlangıçta, hesaplama masraflarını azaltmak için parametre veya ağırlık miktarını azaltır. Sonraki aşama, ağı "aşırı uyumunu" azaltmaktır (Mou ve Zhang, 2022). Şekil 3.1, 2x2 filtre kullanan havuzlama katmanını göstermektedir.



Şekil 3.1: Havuzlama katmanı

3.1.3 Aktivasyon katmanı

“Doğrusal olmayan katman” olarak adlandırılan aktivasyon katmanı, evrimsel katmanı takip eder. Doğrusal bir hesaplama sistemine doğrusal olmayanlığı tanıtmak bu katmanın birincil amacıdır (Peng ve ark., 2023). Aktivasyon fonksiyonu, doğrusal olmayanlığı modele dahil ederek sinir ağının ifade kapasitesini artırır. Yaygın aktivasyon fonksiyonları, denklem 3.1.'de gösterildiği gibi, Düzeltmeli Doğrusal Birim “ReLU” fonksiyonu, “Sigmoid” fonksiyonu ve diğerlerini içerir (Wang ve ark., 2024).

$$f(x) = \max(0, x) \quad (3.1)$$

Bu fonksiyon, x'in pozitif değerleri için f(x) ile gösterilen çıktının her zaman sabit olduğunu, ancak x'in negatif değerleri için sıfır olduğunu ima eder.

3.1.4 Normalizasyon katmanı

Katman normalizasyonu (LayerNorm), ara katmanların dağılımlarını standartlaştırmak için bir yöntemdir. Daha sorunsuz eğimler, hızlandırılmış eğitim ve gelişmiş genelleme doğruluğu sağlar (Krichen, 2020). Toplu normalizasyon, her toplu için ortalama, standart sapma, ölçek ve ofset parametrelerinin hesaplanmasını gerektirir de, bir düzenleyici görevi görür ve ağ eğitim hızını "14"aktörüyle artırır, "evrimsel", "toplu normalizasyon" ve "tam bağlı katmanlar" CNN tasarımının hesaplama açısından en maliyetli bileşenleridir (Ioffe ve Szegedy, 2015).

3.1.5 Tam bağlantılı katman

Belirli bir CNN mimarisinde performansı artırmak için gereken nöron sayısı ve bu katmanlar içindeki optimum FC katmanı sayısı incelenmemiştir. Mevcut literatür, her CNN mimarisinin bir veya daha fazla tam bağlı (FC) katman içerdiğini göstermektedir. (Basha ve ark., 2019). İkileştirilmiş CNN'i hızlandırmadaki bir sorun, tam bağlı katmanların ortadan kaldırılmasıdır (Nakahara ve ark., 2017) (Han ve ark., 2016).

3.1.6 Çıktı katmanı

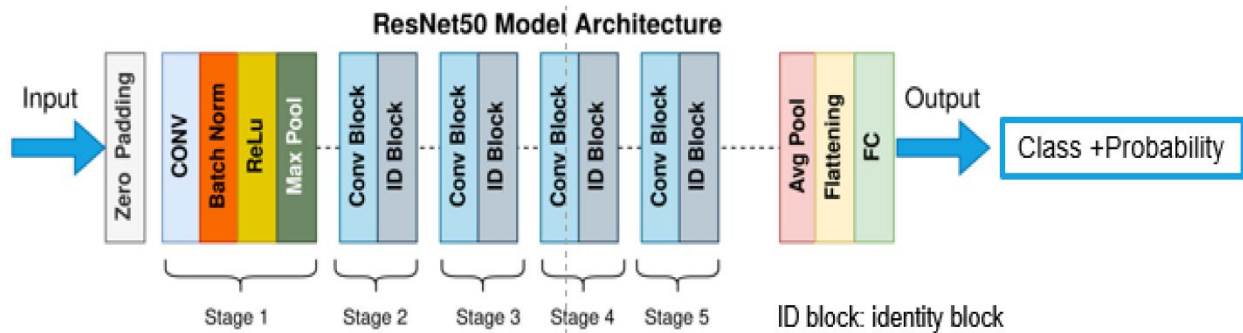
Yinelenen Sinir Ağı, bir katmanın çıktısını koruyarak ve katmanın sonucunu tahmin etmeye yardımcı olmak için girdiye yeniden entegre ederek işlev görür. Sonuçlar Sinir ağının çıktısı 0 ile 1 arasında bir gerçek sayı, bir Boole değeri veya ayrı bir değer (örneğin, bir kategori tanımlayıcısı) olabilir. (Kumar ve Singh, 2019). Tipik olarak, çıktısı 0'dan büyük ve 1'den küçük bir aralıkla sınırlandırılmış tek bir nöron içerir. Ancak, potansiyel olarak birkaç çıktı bulunabilir (Abdelrahman ve Shukla, 2004).

3.2 Transfer Öğrenme

Çok sayıda ön çalışma, denetlenen önceden eğitilmiş modellerin sınıflandırma zorluklarına aktarılmasını ele almıştır. Etkili bir değerlendirme tekniği geliştirmek için, önceden eğitilmiş modellerden toplanan özelliklere dayalı olarak etiket kanıtının maksimum değerini tahmin etmek esastır (You ve ark., 2021). Hafif bir mimariyi korurken, önceden eğitilmiş modellerin bir birlik öğrenme çerçevesine entegre edilmesi bekleyen bir sorundur. (Tan ve ark., 2022). Önceden eğitilmiş bir özellik çıkarıcıyı kullanarak (Hinterstoisser ve ark., 2018) yalnızca sentetik veriler üzerinde son teknoloji nesne dedektörlerini eğitebiliriz. Önceden eğitilmiş bir modeli ince ayarlamak sıklıkla felaketle sonuçlanan bir unutmaya karşılaşır. Yeni görevlerde ince ayar yapıldığında, ön eğitim görevlerindeki performans düşer (Schmied ve ark., 2023). Ön eğitim kavramı transfer öğrenmesiyle ilgilidir (Bahl, Jelinek ve Mercer, 1983). Önceden eğitilmiş bir model ve ince ayar yöntemlerini kullanan transfer öğrenmesi resim kategorizasyonuna uygulanmıştır. (Sharma, A. ve Kaur, P., 2021).

3.3 ResNet50

Model, her biri birden fazla evrimsel katman ve kimlik kısayolları içeren çok sayıda evrimsel bloğa sahiptir (Hassan ve ark., 2020). Bu kısayollar modelin temsilini kolaylaştırır (Alqahtani ve ark., 2023). Bu nedenle nesne tespiti, görüntü segmentasyonu ve görüntü sınıflandırması "ResNet50" çeşitli bilgisayarlı görüş uygulamalarında olağanüstü performans göstermiştir (Geem ve ark., 2020). 2015'te ImageNet Büyük Ölçekli Görsel Tanıma Mücadelesi'nde (ILSVRC) zafer kazanarak çok sayıda önceki sistemi geride bırakmıştır (Karpthy ve ark., 2015). Şekil. 3.3'de ResNet50 Mimarisi:

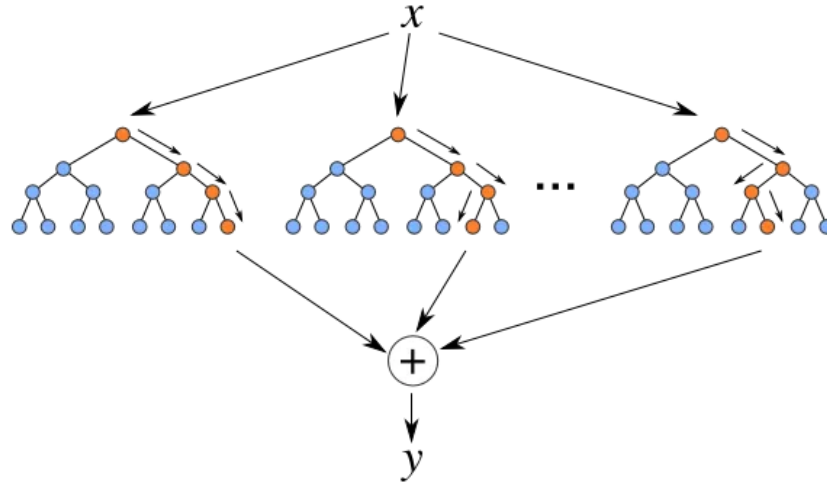


Şekil.3.2 ResNet50 mimaris

(Yu ve Nishio, 2022)

3.4. Rastgele Orman

Rastgele ormanlar bileşik bir makine öğrenme algoritması oluşturur. Sonuçlar, her ağacın en yaygın sınıf için tek bir oy kullandığı ve sonuçta nihai sıralanmış sonucu veren birkaç ağaç sınıflandırıcısından toplanır. Rastgele Orman iyi sınıflandırma doğruluğu gösterir, aykırı değerleri ve gürültüyü etkili bir şekilde barındırır ve aşırı uyuma karşı dirençlidir (Breiman, L. 2001). Rastgele ormanlar gibi topluluk öğrenme teknikleri, özellikle orta ila büyük veri kümeleri için etkilidir. Rastgele karar ağaçları, verilerde bulunan doğrusal olmayanlıkları etkili bir şekilde yönetir ve doğrusal regresyona göre gelişmiş tahmin doğruluğu sağlar. (Genuer ve ark., 2017). Şekil 3.8'de Rastgele Ormanın kapsamlı bir tasarımı:

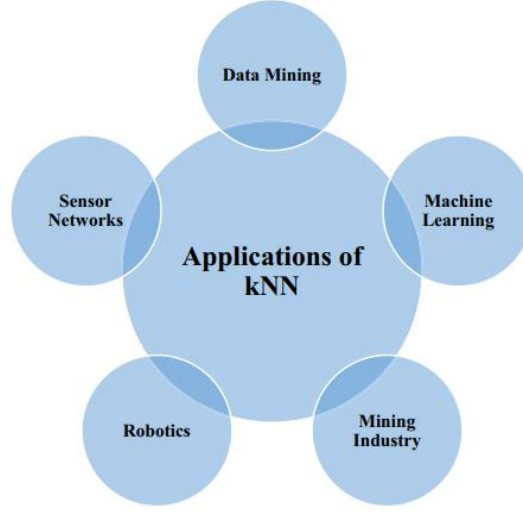


Şekil .3.3 Rastgele Ormanın genel mimaris

(Chaya, 2020)

3.5. K-En Yakın Komşular (KNN)

Bir veri madenciliği yaklaşımı olarak, sınıflandırma ve regresyonda çeşitli kullanımları vardır. Tamamlayıcı kullanımlar arasında bilgisayar sistemlerindeki saldırıları tahmin etmek ve tespit etmek ve kablosuz olarak bağlı PC'ler gibi mobil varlıkların veritabanlarını yönetmek yer alır. (Imandoust ve Bolandraftar (2013). Bu, "k değeri", "mesafe hesaplaması" ve "uygun tahmin edicilerin seçimi" gibi yönlerin model performansını önemli ölçüde etkilediği basit bir makine öğrenme yöntemidir (Zhang, 2016).



Şekil .3.4. K-En Yakın Komşular (KNN)

(Halder ve ark., 2024)

3.6 Karmaşa Matrisi

Karmaşa matrisi (CM), tahmin edilen ve gerçek sınıf etiketlerinin her bir kombinasyonu için örneklerin sıklığını gösteren pozitif tam sayılardan oluşan bir matris olarak tanımlanır. Karmaşa matrisi (CM), tahmin edilen ve gerçek sınıf kümeleri ($\hat{Y}, Y$) kullanılarak hesaplanabilir. Denklem (3.2)'de gösterildiği gibi, M ayrı sınıf etiketine sahip bir sınıflandırma sorununda, ortaya çıkan karışıklık matrisi $M \times M$ boyutundadır ve aşağıdaki gibi tanımlanır:

$$CM = [c_{km}], k, m = 1, 2, \dots, M \quad (3.2)$$

Karışıklık matrisi, sınıflandırma algoritmasının performansını belirten geniş bir gösterge dizisinin tanımının temelini oluşturur. Doğruluk, kesinlik, geri çağırma ve F1 puanı gibi metrikler hem ikili hem de çok sınıflı sınıflandırma görevleri için oluşturulmuştur. (Markoulidakis, I. ve Markoulidakis, G., 2024; Ting, K. M., 2010). Karışıklık matrisi, önerilen mimari tarafından kullanılan kritik bilgileri içerir (Marom ve ark., 2010). Hatalar genellikle her sınıf için tip I ve tip II hataları belirleyen karışıklık matrisleriyle niceliksel olarak belirlenir. Uzman olmayan kullanıcılar karışıklık matrislerini anlama ve kullanma konusunda zorluklar yaşarlar. Hem sütun bazında hem de satır bazında yorumlanmaları gerekir; bu da zahmetli ve hatalara açıktır ve teknik kavramlarının açıklığa kavuşturulması gerekir. (Beauxis-Aussalet, E. ve Hardman, L. (2014). Bir sınıflandırma sistemi tarafından gerçekleştirilen gerçek ve beklenen sınıflandırmalarla ilgili bilgileri kapsayan makine öğrenmesinden gelen bir kavramdır. Karışıklık matrisi iki boyuttan oluşur: biri nesnenin gerçek sınıfı tarafından dizinlenirken, diğeri sınıflandırıcı tarafından tahmin edilen sınıf tarafından dizinlenir. (Deng ve ark., 2016).

Çizelge 3.1. Gerçek ve tahmin edilen model için ikili karışıklık matrisi

		Gerçek Değerler	
		Positive	Negative
Tahmini Değerler	Positive	TP	FP
	Negative	FN	TN

Gerçek Pozitif (TP): Bu, modelin pozitif örnekleri doğru bir şekilde sınıflandırdığı durumları tanımlar. Örneğin, tıbbi bir teşhis durumunda, TP, modelin hastaları belirli bir rahatsızlığa sahip olarak doğru bir şekilde sınıflandırdığı örnekleri gösterebilir.

Yanlış Pozitif (FP): Bu, modelin olumsuz örnekleri yanlış bir şekilde olumlu örnekler olarak yorumlaması durumunda ortaya çıkar. Bu tür bir soruna Tip 1 hatası da denir.

Yanlış Negatif (FN): Bu, modelin pozitif örnekleri yanlış bir şekilde negatif olarak tanımlaması durumunda gerçekleşir. Başka bir deyişle, model aslında pozitif olmasına rağmen negatif bir sonuç öngörür. FN, Tip 2 hatası olarak da bilinir. Örneğin, tıbbi bir teşhiste, model belirli bir rahatsızlığı olan hastaları tanımlayamadığında FN meydana gelir.

Gerçek Negatif (TN): sınıflandırması durumunda gerçekleşir. Aksi takdirde, model pozitif bir sonuç oluştuğunda negatif bir sonuç öngörür. FN için başka bir isim de Tip 2 hatasıdır. Örneğin, model belirli bir rahatsızlığı olan kişileri tanımlayamıyorsa FN tıbbi bir teşhiste görünebilir.

3.6.1 Doğruluk

Doğruluk, doğru tahminlerin toplam tahmin sayısına oranıdır ve doğru tahminlerin sıklığını gösterir. Doğruluk = (Doğru sınıflandırılmış örnek sayısı) / (Toplam örnek sayısı) denkleminde 3.3. gösterildiği gibi

$$\text{Doğruluk} = (TP + TN) / (TP + FP + TN + FN) \quad (3.3)$$

$$\text{Sınıflandırma doğruluğu} = (\text{doğru tahminler} / \text{toplam tahminler}) * 100 \quad (3.4)$$

Hata, sınıflandırma doğruluğunun birden çıkarılmasıyla hesaplanır. Dolayısıyla,

$$\text{Hata oranı} = 1 - \text{Sınıflandırma doğruluğu} \quad (3.5)$$

(Sokolova, M. ve Lapalme, G., 2009; Giannoulidis ve ark., 2024)

3.6.2 Yanlış sınıflandırma

Yanlış Sınıflandırma Oranı: Buna "hata oranı" da denir ve modelin ne kadar yanlış olduğunu gösterir. Bu nedenle, bu durumda yüksek bir geri çağırma önemlidir. Yanlış sınıflandırma, sınıflandırıcının eksikliklerine ve performans geliştirme alanlarına ışık tutar, denklem 3.6.'da gösterildiği gibi (Sokolova, M. ve Lapalme, G., 2009; Zeng, G., 2020)

$$\text{Yanlış Sınıflandırma Oranı} = (FP + FN) / (FP + FN + TP + TN) \quad (3.6)$$

3.6.3 Hatırlama

Bu, doğru tahmin edilen gerçek pozitiflerin oranını niceliksel olarak ifade eder ve modelin pozitif örnekleri tahmin etmedeki doğruluğunu gösterir. Hatırlama, yanlış negatifleri azaltmanın önemli olduğu durumlarda özellikle önemlidir. (Sokolova, M. ve Lapalme, G. (2009). Hatırlama daha yüksekse, çoğu pozitif örneği (TP+FN) pozitif (TP) olarak belirleyeceğiz. Sonuç olarak, (FP) okumaları artacak ve genel doğruluk düşecektir. Hafızanız zayıfsa, pozitif olması gereken ancak negatif olarak kategorize edilen çok sayıda (FN)'niz vardır. Bu, pozitif bir örnek belirlendiğinde, denklemde gösterildiği gibi, bunun gerçek bir pozitif olduğuna dair daha fazla güvenceye sahip olunabileceğini gösterir. 3.7. (Sattar ve ark., 2024).

$$\text{Geri çağırma} = TP / (TP + FN) \quad (3.7)$$

3.6.4 Kesinlik

3.8. denklemde gösterilen pozitif olarak tanımlanan vakalar içindeki doğru tahminlerin oranını görüntüler (Ma ve ark., 2019).

$$\text{Kesinlik} = TP / TP+FP \quad (3.8)$$

Yanlış Pozitiflerin Yanlış Negatiflerden daha önemli olduğu vakalarda değerli bir göstergedir. (Sattar ve ark., 2024).

Modelin kesinliğini artırmaya çalıştığımızda, geri çağırma azalır ve bunun tersi de geçerlidir. F1 Puanı, Kesinlik ve Geri Çağırmanın harmonik ortalamasıdır ve bu iki metriğin birleşik bir ölçüsünü sağlar. Kesinlik geri çağırmaya eşit olduğunda en üst düzeye çıkar. F1 Puanının yorumlanabilirliği yetersizdir ve bu, sınıflandırıcının doğruluğu mu yoksa geri çağırmayı mı optimize ettiği konusunda belirsizliği gösterir. Bu nedenle, denklemde gösterildiği gibi

sonuçların kapsamlı bir genel görünümünü sağlayarak bunu diğer değerlendirme kriterleriyle birlikte kullanırız (Warner ve ark., 2024).

$$F1 - \text{Puan} = 2 \times \text{Kesinlik} \times \text{Geri Çağırma} / (\text{Kesinlik} + \text{Geri Çağırma}) \quad (3.9)$$

3.6.5 Yanlış pozitif oranı

Yanlış pozitif oranı, denklem (3.10)'da gösterildiği gibi, yanlış bir şekilde pozitif olarak tahmin edilen olumsuz durumların oranını niceliksel olarak ifade eder.

$$\text{FPR} = \text{FP} / (\text{TN} + \text{FP}) \quad (3.10)$$

3.7 Aktivasyon Fonksiyonu

Bu, k değeri, mesafe hesaplaması ve uygun tahmin edicilerin seçimi gibi unsurların model performansını önemli ölçüde etkilediği basit bir makine öğrenme. (GeeksforGeeks, 2025). Şekil (3.3), KNN çalışmasını göstermektedir. Yapay Sinir Ağında, aktivasyon fonksiyonları, girdiler ve ilişkili çıktılar arasındaki doğrusal olmayan ve karmaşık eşlemelerin öğrenilmesini ve yorumlanmasını kolaylaştırdıkları için önemlidir. (Sharma ve ark., 2020)

Aktivasyon fonksiyonu, herhangi bir Yapay Sinir Ağına temel doğrusal olmayan özellikler kazandırır ve eğitimleri için çok önemlidir. Karmaşık gösterimlerin öğrenilmesini kolaylaştırmak için modelin gerekli doğrusal olmayanlığını sağlarlar. (Adjailia, F. ve Adjailia, S. 2020), Derin sinir ağlarının önemli başarıları, aktivasyon fonksiyonlarının ilerlemesinin en önemli olduğu çeşitli faktörlere dayanmaktadır (Dubey ve ark., 2021), Bir ağ veya onun belirli düğümleri için bir aktivasyon fonksiyonunun seçimi kritik bir girişimdir; (Szandała, T, 2021):

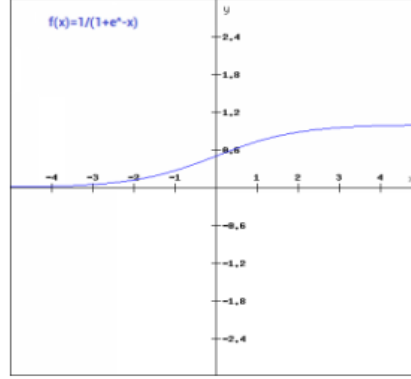
Yaygın aktivasyon fonksiyonları şunları içerir:

3.7.1 Sigmoid

Sigmoid fonksiyonu, geri yayılım eğitim sürecini kolaylaştıran türevlenebilir yapısı nedeniyle tercih edilir. Sıfır olmayan merkezli bir çıktı değeri ağ performansını düşürür. (S. Shanthi. 2012), Sigmoid fonksiyonu, derin öğrenme araştırmalarının erken aşamalarında evrimsel sınıflandırma modellerinde yaygın olarak kullanılmıştır; ancak, genellikle gradyan difüzyonu sorununa yol açar. Sigmoid fonksiyonunun eğrisi, yaygın bir doğrusal olmayan aktivasyon fonksiyonunu temsil eden Şekil (3.9)'da gösterilmiştir. Bu fonksiyonun çıktısı sınırlıdır ve derin öğrenmenin yeni aşamasında derin sinir ağlarında aktivasyon fonksiyonu olarak yaygın olarak kullanılmıştır. Sigmoid fonksiyonunun nöronal sinapslarla hizalanmasına ve türevini elde etmenin kolaylığına rağmen, sınırlamaları nedeniyle günümüzde neredeyse hiç kullanılmamaktadır. Sigmoid fonksiyonu, eğrisinin gösterdiği gibi yumuşak doygunluk özelliğini sergiler. Giriş aşırı derecede büyük veya küçük hale geldikçe grafiğin eğimi sifıra yaklaşır. Fonksiyonun eğimi sifıra yaklaştığında, altta yatan ağa iletilen eğim önemli ölçüde azalır ve ağ parametrelerinin eğitimini etkisiz hale getirir. Aynı anda, bu fonksiyonun çıktısı sürekli olarak pozitif olduğundan ağırlık

güncellemesi tekil bir yönde ilerler ve böylece yakınsama oranını etkiler. Sigmoid fonksiyonu aşağıdaki formülle tanımlanır (Kumar, A. ve Singh, R. 2023; Zhang ve ark., 2020):

$$f'(x) = e^{-x} / (1 + e^{-x})^2 \quad (3.11).$$



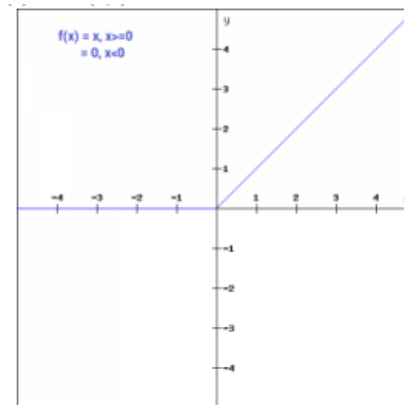
Şekil 3.5 Sigmoid fonksiyonunun eğrisi

(Sharma ve ark., 2020)

3.7.2 ReLu

ReLU fonksiyonu en etkili aktivasyon fonksiyonlarından biridir, ancak belirli sınırlamalara sahiptir. ReLU fonksiyonunun türevi, nöronal nekroza neden olabilecek negatif giriş değerleri için sürekli olarak 0'dır; ancak ReLU fonksiyonu önemli ölçüde geliştirilmiş hesaplama hızı sunar. ReLU'nun doymamış yapısı nedeniyle, sigmoid ve tanh fonksiyonlarının aksine, gradyan difüzyon sorunlarıyla karşılaşmaz. Bu fonksiyonun denklemi aşağıdaki gibi çizilmiştir (Kumar, A. ve Singh, R. 2023):

$$f(x) = \max(0, x) \quad 3.12$$



Şekil 3.6: ReLu fonksiyonunun eğrisi

(Sharma ve ark ., 2020)

ReLU, tüm nöronları aynı anda aktive etmediği için alternatif fonksiyonlardan daha etkili olduğunu kanıtlamaktadır; bunun yerine, herhangi bir anda yalnızca belirli bir nöron alt kümesi aktive olmaktadır (Dooley ve ark., 2024).

3.7.3 Softmax

Softmax aktivasyon fonksiyonu ilk olarak doğruluk gereksinimleri açısından incelenir ve belirli değişiklikler önerilir; son tahminleri oluşturmak için tam bağlantılı katmanlar kullanılır. Bir softmax katmanı genellikle CNN'nin son çıktısından önce gelir. Softmax katmanı olasılık değerlerini matematiksel formülasyonuna göre hesaplar. Çıktı değerlerinin toplamı 1'e eşittir. Değerler, aynı anda denklem (3.13)'te sonuca olan güven düzeyini gösterir. (Bawa, V. S. ve Kumar, V. 2019).

$$(z)_j = e_j / \sum_{k=1}^K e_k \text{ for } j = 1, \dots, K \quad (3.13)$$

3.8 YOLO

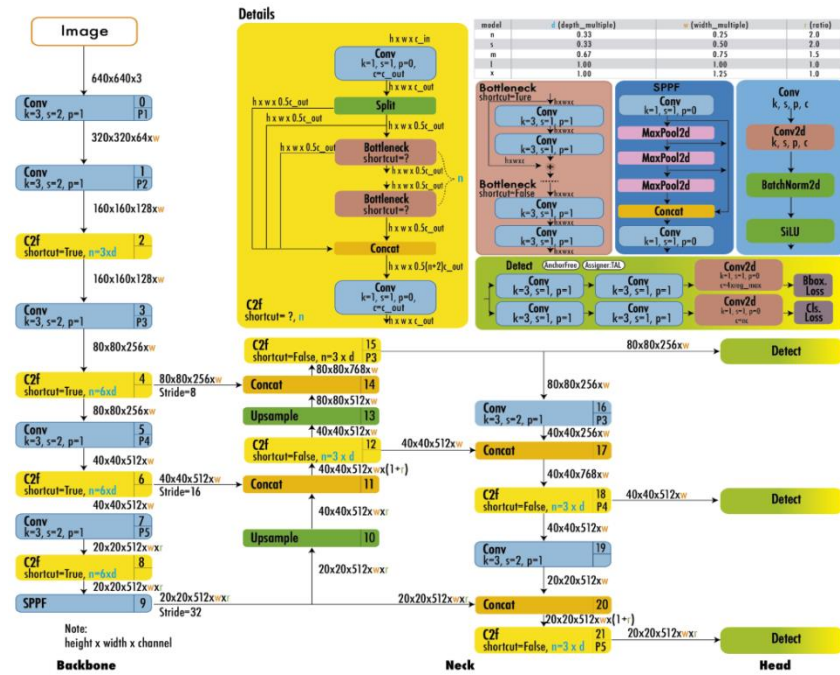
Nesne tanımlama metodolojileri yapay zeka alanının temel taşı oluşturur. Birincil gözlem, You Only Look Once (YOLO) algoritmasındaki geliştirmelerin hala devam ettiğidir. (Wang ve ark., 2022). You Only Look Once (YOLO) ağı, gerçek zamanlı nesne algılama için uygun hızlı tanıma yetenekleri gibi çeşitli nedenlerle ünlüdür. Sonuç olarak, gerçek zamanlı nesne algılama, otomasyonun ilerlemesinde ve insan emeğinin ikamesinde kritik bir alan olarak ortaya çıkmıştır. Bilgisayar görüşü ve nesne algılama, çok yönlü tepki veren robotik sistemlerin geliştirilmesini kolaylaştırması beklenen makine öğrenimi içindeki önemli alanlardır. (Handalage, 2021). YOLO, robotik, otonom araçlar ve video gözetim uygulamaları için görüntülerdeki nesnelerin hızlı ve güvenilir bir şekilde tanımlanmasını kolaylaştıran temel bir gerçek zamanlı nesne algılama sistemi olarak ortaya çıkmıştır. Bu çerçeve, hız ve hassasiyetin olağanüstü dengesiyle öne çıkmaktadır. YOLO ailesinin performans iyileştirmeleri, mimari gelişmelerin, eğitim metodolojilerinin ve veri kümesi artırmanın bir sentezine atfedilmektedir. (Terven ve ark., 2023).

3.8.1 YOLO v8 genel bakış

YOLOv8, Ultralytics tarafından 10 Ocak 2023'te nesne algılama, görüntü sınıflandırma ve nesneler arasında ayırım yapmak için en son teknoloji YOLOv5 modeline bir uzantı olarak tanıtıldı. Ultralytics tarafından YOLOv8'de YOLOv5'ten daha verimli ve kullanıcı dostu hale getirilen birkaç iyileştirme yapıldı (P. Pan ve ark., 2024) (M. Sohan ve ark., 2024). Bu, farklı ihtiyaçlara göre özelleştirilmiş çeşitli boyutlarda mevcut, dağıtım açısından verimli bir modeldir. En küçük model olan YOLOv8n, son derece hız ve verimlilik dostudur ve çok sınırlı hesaplama gücü veya zorlu gerçek zamanlı gereksinimler içeren durumlar için uygundur. YOLOv8s, hızlı algılama gerektiren görevler için hızı ve doğruluğu makul bir hassasiyet seviyesiyle dengeler. YOLOv8m orta yolu sunar: orta düzeyde hesaplama talepleriyle daha yüksek doğruluk elde edilebilir. YOLOv8l, daha yüksek doğruluklara daha fazla vurgu yapması anlamında büyüktür, hızdan ziyade hassasiyete öncelik verilen görevler için oldukça uygundur. En büyük model olan

YOLOv8x, en yüksek doğruluğu üretir ve uygulamaları maksimum performans ve büyük hesaplama kaynaklarının kullanılabilirliği düşünülerek çalıştırmak üzere tasarlanmıştır. Bu modeldeki varyasyonlar, kullanıcıların hız, doğruluk ve kullanılabilir kaynaklar (G. G. Casas ve ark., 2023) gereksinimlerine en uygun modeli seçmelerine olanak sağlamak için yapılmıştır.

YOLOv8'in mimarisi, Şekil 3.7'de gösterildiği gibi üç temel bileşenden oluşur: "Omurga", "Özellik Geliştirme Ağı (boyun)" ve "Algılama Başlığı" (Bolikulov ve ark., 2024). YOLOv8 modelinin omurgası, YOLOv5'te etkili olan ve sağlam özellik çıkarımı için güçlü bir temel sağlayan Çapraz Aşamalı Kısmi (CSP) modül mimarisine dayanmaktadır (P. Pan ve ark., 2024) (B. Xiao ve ark., 2024) YOLOv8, görsel özelliklerin çıkarılmasını geliştirmek için CSP modüllerine sahip bir CSPDarknet53 omurgasıyla özelleştirilmiştir (G. G. Casas. ve ark., 2023). YOLOv8, omurga mimarisini iyileştirerek, yüksek algılama performansını korurken modelin çeşitli veri kümeleri arasında genelleme yapma yeteneğini geliştirir. (Sohan ve ark., 2024). Özellik Geliştirme Ağı olarak işlev gören YOLOv8-Neck, yukarı örnekleme aşamasında belirli evrimsel süreçleri ortadan kaldırarak süreci optimize ederken, Özellik Piramidi Ağı ile Yol Toplama Ağı (PA-FPN) metodolojisini kullanır. Algılama Başlığı, sınıflandırma ve yerleştirme dalları arasında ayırma yapan yeni bir "ayrılmış baş" modülüne sahiptir. Bu mimari, sınıflandırma ve regresyon görevleri arasındaki çatışmaları azaltarak gelişmiş performansla sonuçlanır. Mimari, hesaplama verimliliğini ve algılama doğruluğunu optimize etmeyi amaçlar ve YOLOv8'i gerçek zamanlı nesne tanımlama uygulamaları için özellikle uygun hale getirir. (Yi ve ark., 2024)



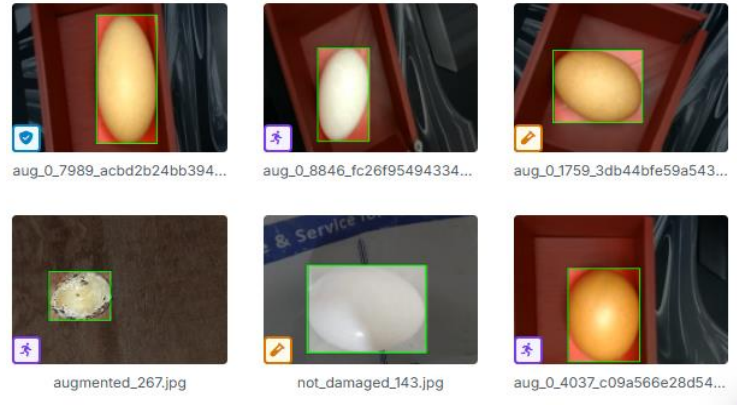
Şekil 3.7. YOLOv8 mimarisi

(Sattar ve ark., 2024)

4. KULLANILAN MATERYAL VE YÖNTEMLER

Bu bölüm, bir yumurta türleri ve durumları tanıma sistemi geliştirmeye odaklanmaktadır. Sistem birkaç aşamadan oluşmaktadır: veri kümesi toplama ve yumurta görüntülerini segmentlere ayırmak için veri ön işleme. Bu bölüm her aşamayı ayrıntılı olarak ele almakta ve önerilen derin sinir ağlarını ve hiper parametre ayarını tanıtmaktadır.

Veri kümesinin ön işlenmesi, derin öğrenmenin iş akışında yaygın bir ilk adımdır. Toplanan veri kümesini, CNN modellerinin "eğitim", "doğrulama" ve "test" için kullanabileceği yöntemleri kullanarak hazırladık. YOLOv8'i "ince ayarlama"nın ilk adımı, veri kümesini hazırlamayı içerir. Bu çalışma için, "roboflow web sitesi" tarafından tavuk yumurtası (hem sağlam hem de kırık), ördek yumurtası (hem sağlam hem de kırık) ve bıldırcın yumurtası (hem sağlam hem de kırık) olmak üzere birden fazla yumurta kategorisi içeren açıklamalı bir veri kümesi oluşturuldu. Veri kümesinin düzgün bir şekilde etiketlendiğinden emin olmak veri kümesindeki çeşitlilik, modelin tespit görevleri sırasında iyi bir şekilde genelleştirilmesini sağlar. Roboflow'un açıklama araçları kullanılarak, Şekil (4.1)'de gösterildiği gibi, görüntülerdeki her nesnenin etrafına sınırlayıcı kutular çizildi. Veri seti, kırpma adımı uygulanmadan önce 1734 görüntü içermektedir ve kırpma adımı sonrasında 1469 görüntü elde edilmiştir. Açıklama işlemi tamamlandıktan sonra, veri seti, sınırlayıcı kutu koordinatlarını ve karşılık gelen sınıf etiketlerini içeren her görüntü için bir metin dosyası sağlayan YOLO biçiminde dışa aktarıldı.



Şekil 4.1: Veri seti görüntülerinin örneği

Bir sonraki adımda, altı veri kümesini bir eğitim (70%), doğrulama veri kümesi (20%) ve bir test veri kümesi (10%) olarak bölmektir. Tüm klasörler Görüntüler "içerir" klasörü ve "labels" klasöründen oluşacak şekilde oluşturuldu. Üçüncü adım yumurtanın görüntüdeki konumuna bağlı olarak görüntülerde bazı değişiklikler yapmak için bazı kodlar kullanmak (kırpma işlemi), şimdi iki veri kümemiz var.

Altı yumurta veri setinde, kesme işleminden önce eğitim görüntülerinin sayısı 1263'tür ve işlemden sonra eğitim görüntülerinin sayısı 842'dir, ve doğrulama görüntülerinin sayısı işlemden önce 388'dir ve işlemden sonra 384'dir ve son sınıf test sınıfının işlemden önce sayısı 245'tür ve işlemden sonra ise her yumurta türünden 243 görüntü sayısıdır, Çizelge 4.1'de gösterildiği gibi.

Çizelge 4.1. Yumurta veri seti

Yumurtalar	İşlemeden <u>önce</u> Örnek veri kümesi	İşlemeden <u>sonra</u> Seçili Veri Kümesi
Eğitim	1263	842
Doğrulama	388	384
Test	245	243

Bu çalışmadaki derin öğrenme modellerinin deneysel yürütülmesi ve eğitimi, aşağıdaki özelliklere sahip bir kişisel bilgisayarda gerçekleştirildi:

2.30GHz'de çalışan bir Intel® Core™ i5-5300U, "CPU", Windows 10 Pro 64-bit, "YOLO" ve "CNN" derin öğrenme modellerinin eğitimi ve değerlendirmesi, yüksek performanslı GPU'lara erişim sağlayan ve daha verimli model eğitimi, deney ve düşük zaman sağlayan "Google Colab Pro" kullanılarak gerçekleştirildi.

4.1 Veri Kümeleri

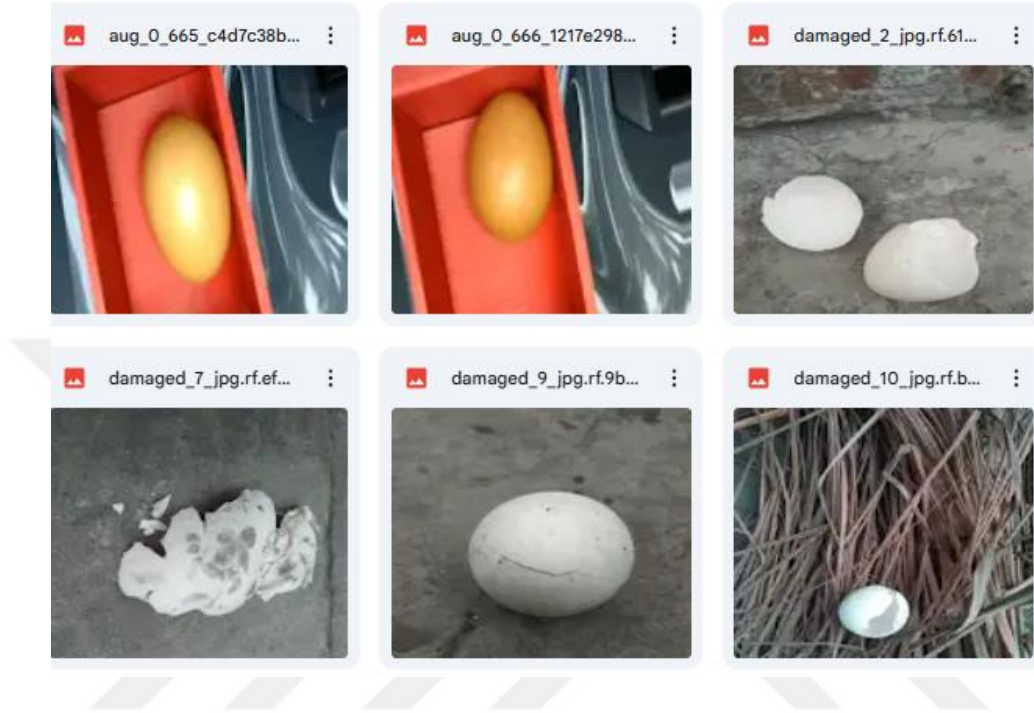
Önerilen modelde, veri kümesi belirtilen tüm yumurtaların görüntülerini içerir. Bu çalışmada, veri kümesini çeşitli kaynaklardan kullandık:

1. Tavuk yumurtası görüntülerinin örnekleri, 7392 görüntü içeren kaggle web sitesinden toplandı. Bu veri kümesinden seçilen örnekler, Çizelge 4.2'de gösterildiği 472'dir.

Çizelge 4.2. Tavuk yumurtası örnekleri

Türü	Veri seti örnekleri	Seçilmiş örnekler	Veri setleri
Tavuk yumurtaları	7392	472	(Gauravduttakiit, n.d.)

Şekil 4.2'de gösterilen tavuk yumurtalarının rastgele örnekleri



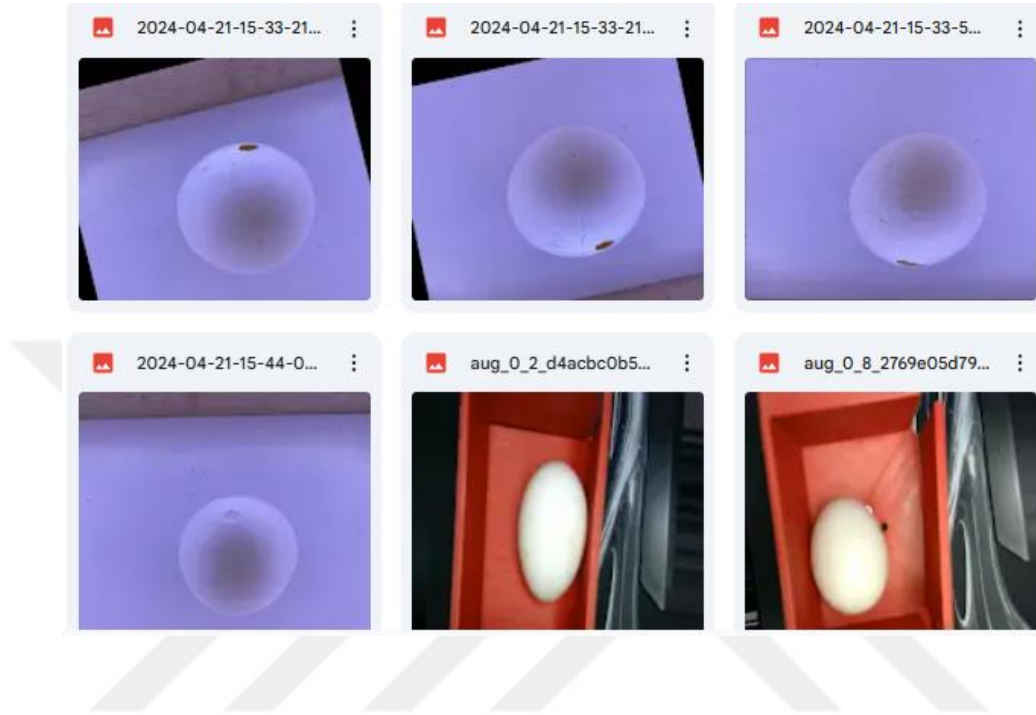
Şekil 4.2. Tavuk yumurtası örnekleri

2. Ördek yumurtası görüntü örnekleri kaggle web sitesinden toplanan 7337 görüntü içeriyor. Bu veri setinden seçilen örnek sayısı Çizelge 4.3'te gösterildiği gibi 746'dır.

Çizelge 4.3. Ördek yumurtası örnekleri

Türü	Veri seti örnekleri	Seçilmiş örnekler	Veri setleri
Ördek yumurtaları	7337	746	(Gauravduttakiit, n.d.)

Şekil 4.3'te gösterilen ördek yumurtalarının rastgele örnekleri



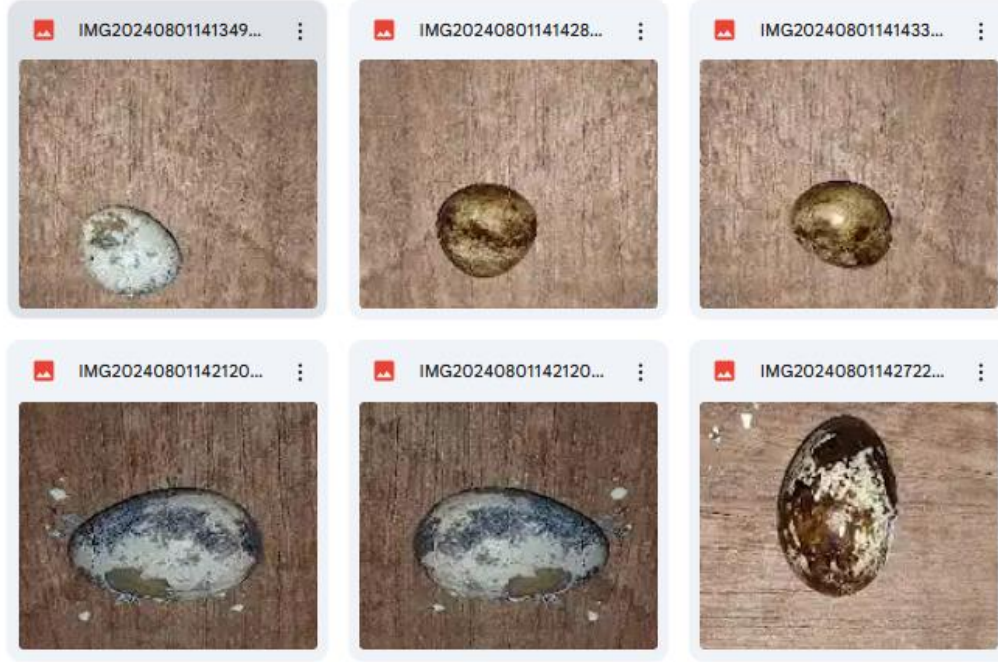
Şekil 4.3: Ördek yumurtası örnekleri

3. Bıldırcın yumurtası görüntülerinin örnekleri, 2071 görüntü içeren roboflow web sitesinden toplandı. Bu veri setinden seçilen örnek sayısı, Çizelge 4.4'te gösterildiği gi 45'dir.

Çizelge 4.4. Bıldırcın Yumurta Örnekleri

Türü	Veri seti örnekleri	Seçilmiş örnekler	Veri setleri
Bıldırcın yumurtası	2071	45	(Roboflow, n.d.)

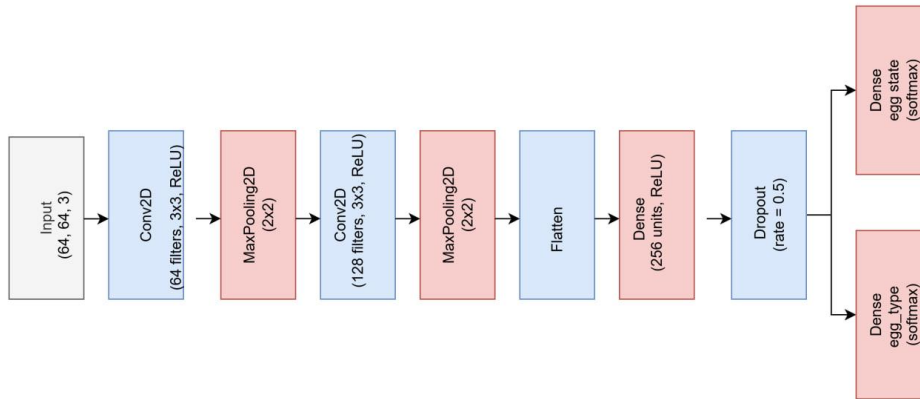
Şekil 4.4'te gösterilen rastgele Bildircin yumurtası örnekleri



Şekil 4.4: Bildircin yumurtası örnekleri

4.2 CNN Yarış Modellerinin Eğitimi Ve Test Edilmesi

Bu model altı tür ve yumurta durumunu tanır: Tavuk yumurtası (hem sağlam hem de kırık), ördek yumurtası (hem sağlam hem de kırık) ve bildircin yumurtası (hem sağlam hem de kırık). Bunu mümkün olan en az parametreyle yapar. Şekil 3.7, CNN modelinin mimarisini özetlemektedir.



Şekil 3.5 CNN mimarisi

Yumurta modeli aşağıdaki adımlardan oluşur:

Adım 1: Veri setinin ön işlenmesi (Tavuk yumurtaları, Ördek yumurtaları, Bildircin yumurtaları)

Adım 2: Veri setini “eğitim” için (%70), “doğrulama” için (%20) ve “test” için (%10) olmak üzere üçe bölün.

Adım 3: Modelleri nesneyi (yumurta) tespit etmek için YOLO V8 ile eğitin ve test edin ve sınıflandırma için farklı bir cnn modeli ("rest net 50") ve makine öğrenmesinde bazı algoritmalar ("knn", "rastgele orman") kullanın.

Adım 4: Modelleri değerlendirin ve kaydedin.

İlk yumurta veri setleri eğitim modeli özeti şunlardır:

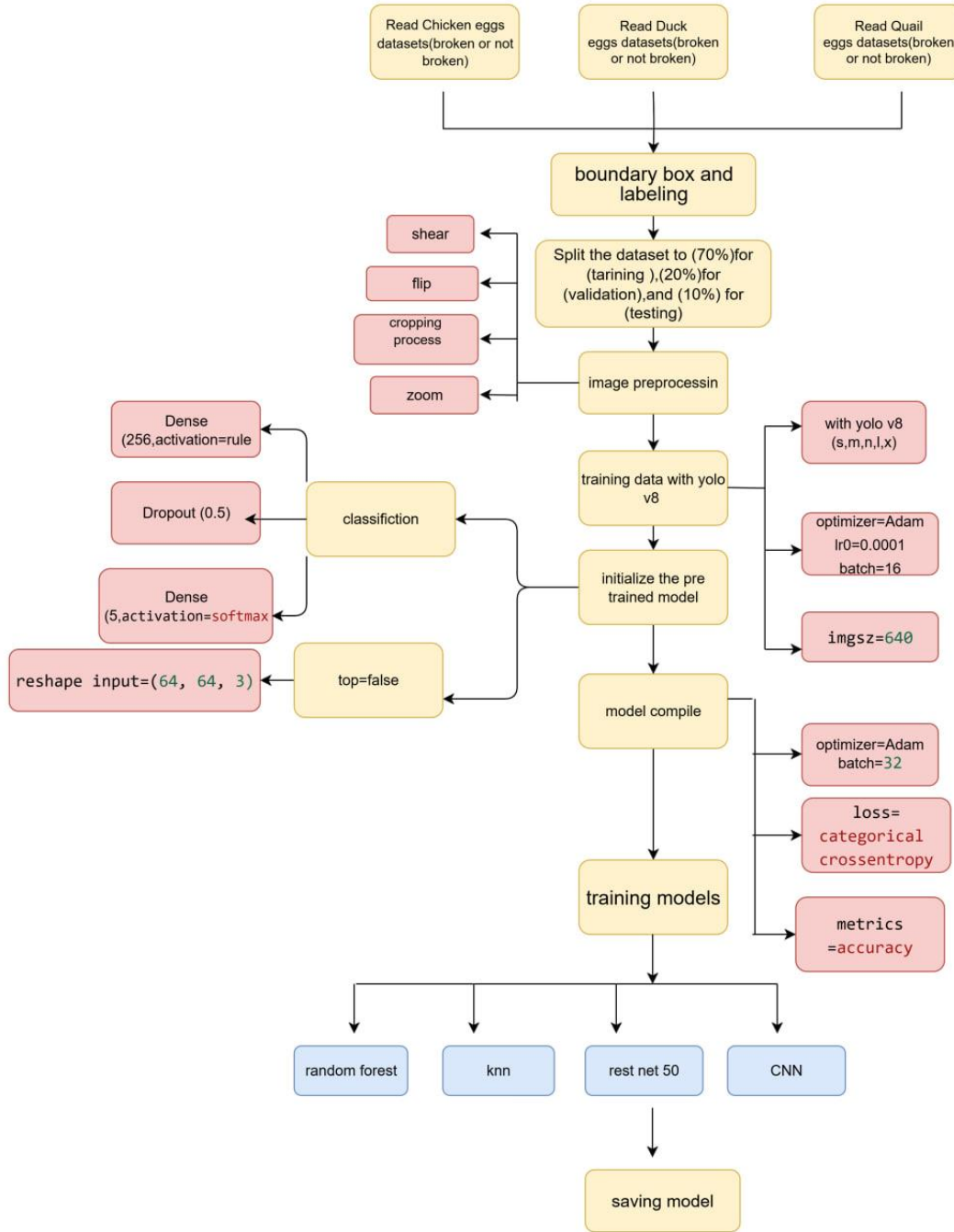
Çizelge 4.5. Kullanılan CNN modelinin özeti

Model: "functional_12"

Layer (type)	Output Shape	Param #	Connected to
input (InputLayer)	(None, 128, 128, 3)	0	-
conv2d_24 (Conv2D)	(None, 128, 128, 64)	1,792	input[0][0]
max_pooling2d_24 (MaxPooling2D)	(None, 64, 64, 64)	0	conv2d_24[0][0]
conv2d_25 (Conv2D)	(None, 64, 64, 128)	73,856	max_pooling2d_24...
max_pooling2d_25 (MaxPooling2D)	(None, 32, 32, 128)	0	conv2d_25[0][0]
flatten_12 (Flatten)	(None, 131072)	0	max_pooling2d_25...
dense_12 (Dense)	(None, 256)	33,554,688	flatten_12[0][0]
dropout_12 (Dropout)	(None, 256)	0	dense_12[0][0]
egg_type (Dense)	(None, 3)	771	dropout_12[0][0]
egg_condition (Dense)	(None, 2)	514	dropout_12[0][0]

Total params: 33,631,621 (128.29 MB)
Trainable params: 33,631,621 (128.29 MB)
Non-trainable params: 0 (0.00 B)

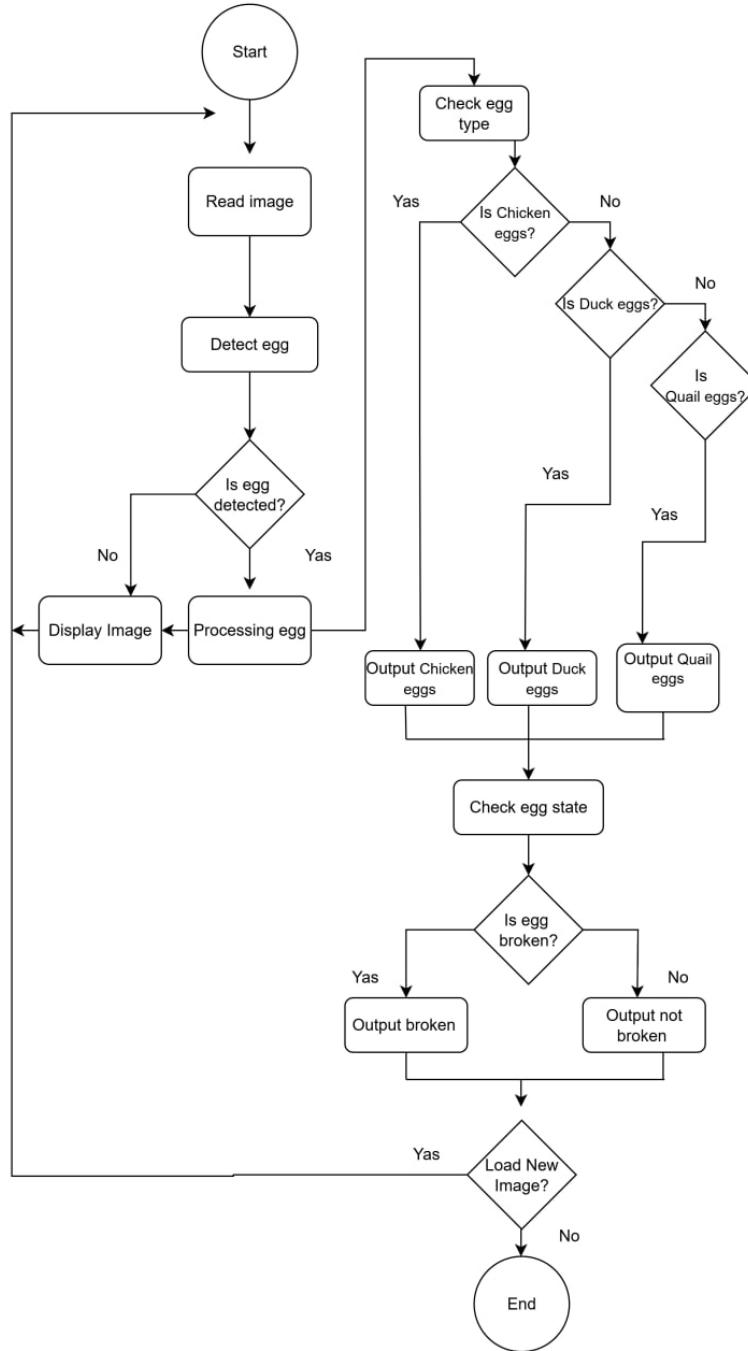
Model, görüntünün yumurta türlerini ve durumlarını belirlemekte ve sonuçları görüntülemektedir. Üç model, Şekil 4.6'de gösterildiği gibi, bir cnn modeli ve makine öğrenmesindeki bazı algoritmalarla metodolojiyi eğitmekte ve test etmektedir.



Şekil 4.6: Altı yumurta tipinde üç CNN modelinin eğitilmesi.

4.3 CNN Sistem Yapısı

Şekil (4.7) yumurta türü ve durum tanıma sistemleri için adımları ana hatlarıyla belirtir. Sistem, işlemi görüntüyü yükleyerek başlatır. Model yumurtayı algılar. Model yumurta görüntüsünü tanımlayamazsa, sistem yumurtayı algılayana kadar görüntü adımını tekrar dener. Model görüntünün türlerini ve durumlarını tanımlamakta ve sonuçları göstermektedir.



Şekil 4.7: Önerilen yumurta tipi ve durum sistemi yapısı

5. SONUÇLAR VE TARTIŞMA

Bu bölüm, Dördüncü Bölüm'de özetlendiği gibi, üç tür (Tavuk yumurtaları, Ördek yumurtaları, Bildircin yumurtaları) (hem sağlam hem de kırık) kullanılarak yumurta türleri ve durum tanımayı hedefleyen önerilen metodolojilerin sonuçlarını sunar. Önerilen modellerin performans değerlendirmesi, doğruluk, geri çağırma, kesinlik, kayıp ve F puanı dahil olmak üzere kişisel kimlik tanıma sistemlerini değerlendirmek için sıklıkla kullanılan standart ölçümler kullanılarak gerçekleştirildi. Bölüm, modellerin (YOLO V8, CNN modeli, ResNet50, Random Forest ve KNN) kullanımından elde edilen sonuçların bir incelemesiyle başlar.

5.1 İşlemeden önce eğitim:

5.1.1 YOLOV8 modellerinin uygulama sonuçları

Başlangıçta, yolov8 görüntülerdeki yumurta tespit etmek ve (sınır kutusu) koymak için kullanıldı. Her görüntü için belirli bir format içeren etiketler oluşturuldu: class_id x_center y_center width height, burada class_id sınıf numarasını temsil eder (sıfır tabanlı, örneğin sağlam için 0, kırık için 1, vb.).

x_center: Nesnenin merkezi (görüntü genişliğine göre, 0 ile 1 arasında normalize edilmiştir).

y_center: Nesnenin merkezi (görüntü yüksekliğine göre, 0 ile 1 arasında normalize edilmiştir).

width: Nesnenin sınırlayıcı kutusunun genişliği (0 ile 1 arasında normalize edilmiştir).

height: Nesnenin sınırlayıcı kutusunun yüksekliği (0 ile 1 arasında normalize edilmiştir).

class_id'den sonraki tüm değerler görüntü boyutlarına göredir (normalize edilmiştir).

Ayrıca, görüntülere giden yollar, etiketler ve sınıf adları hakkında bilgi içeren yaml adlı bir dosya da vardır. Eğitim veya değerlendirme sırasında YOLO'ya geçirilir. Biçimi şöyledir

path: /content/drive/MyDrive/

train: main_train

val: main_val

nc: 6

names: ['chicken_not_broken', 'chicken_broken', 'duck_not_broken', 'duck_broken', 'quail_not_broken', 'quail_broken']

Çizelge 5.1. İşlemeden önce yolov8 modellerinin eğitilmesi

Types of yolo v 8	Precision	Recall	mAP@0.5	mAP@0.5:0.95	Fitness
Yolo v8 n	0.7831	0.715	0.7642	0.6897	0.6972
Yolo v8 s	0.7553	0.9325	0.8570	0.7840	0.7913
Yolo v8 m	0.7953	0.7838	0.8823	0.7917	0.8008
Yolo v8 l	0.8111	0.8256	0.8159	0.7314	0.7398
Yolo v8 x	0.8523	0.7967	0.8763	0.7863	0.7953

En yüksek mAP@0.5 değeri (0.8823), nesne sınıfları arasında üstün yerelleştirme doğruluğunu gösteren YOLOv8m tarafından gösterilmiştir. En iyi uygunluk puanı (0.8008) ve yüksek geri çağırma (0.7838) ile güçlü bir genel performans sergileyerek, hassasiyet ve geri çağırma arasında bir denge gerektiren senaryolar için güçlü bir rakip haline gelmiştir.

En yüksek geri çağırma (0.9325), kaçırılan tespitleri önlemek için önemli olan ve özellikle kırık yumurta tespiti gibi güvenlik açısından kritik işlerde faydalı olan YOLOv8s tarafından elde edilmiştir. Bununla birlikte, bu durum doğrulukta (0.7553) ve mAP@0.5:0.95 (0.7840) değerlerinde küçük bir ödün verilmesine yol açmıştır. En karmaşık model olmasına rağmen, YOLOv8x, YOLOv8m'den belirgin bir şekilde daha iyi performans göstermedi. Kesinlik (0.8523), geri çağırma (0.7967) ve mAP@0,5 (0.8763) değerleri yüksek ölçütlerdi; ancak uygunluk puanı (0.7953), büyük olasılıkla aşırı uyum veya model karmaşıklığından kaynaklanan azalan getirilerin bir sonucu olarak YOLOv8m'nin biraz altındaydı.

0,8159'luk bir mAP@0,5, yüksek hassasiyet (0,8111) ve güçlü geri çağırma (0,8256) ile YOLOv8l dengeli bir profil sergiledi. Tek bir kritere göre en iyisi olmasa da, vasat performansı, hız ve doğruluğun dengelenmesi gerektiğinde en iyi seçenek olabileceğini göstermektedir. Tüm metriklerde, özellikle geri çağırma (0,715) ve mAP@0,5:0,95 (0,6897) olmak üzere en düşük puanlara sahip olan en küçük ve en hızlı varyant olan YOLOv8n, sınırlı donanımda gerçek zamanlı çıkarım en önemli öncelik olmadığı sürece karmaşık tespit görevleri için o kadar kullanışlı olmayabilir.

Sonuç: En yüksek mAP@0,5 ve uygunluk, güçlü geri çağırma ve hassasiyet değerleriyle Yolov8m, işleme başlamadan önce gerçek dünya uygulamaları için en mantıklı seçenektir.

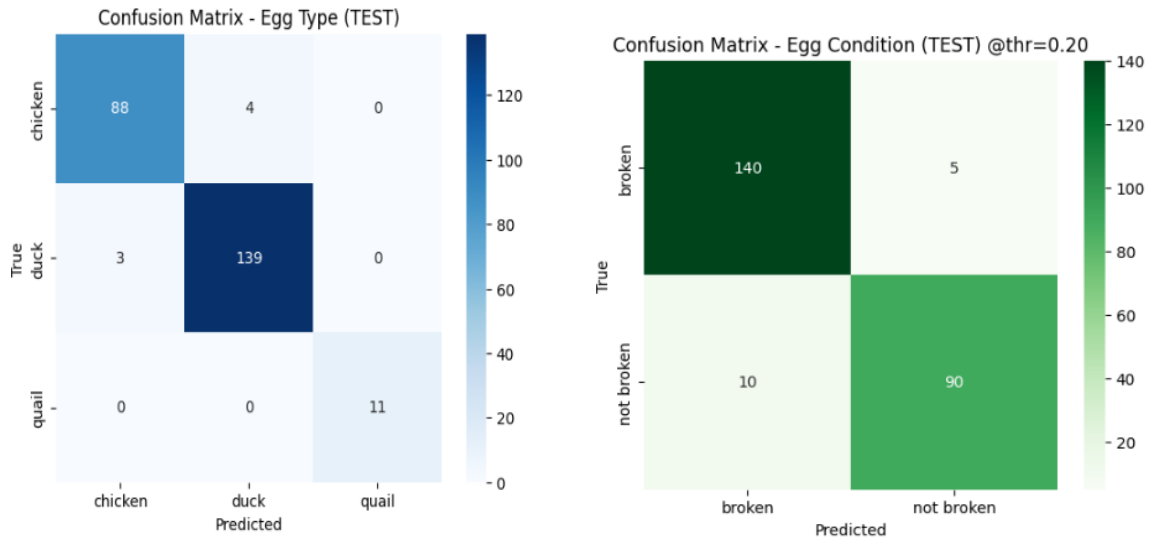
Hata tespiti veya tıbbi görüntüleme gibi geri çağırmanın kritik öneme sahip olduğu uygulamalarda YOLOv8s daha iyi olabilir. Karmaşıklığına rağmen YOLOv8x, YOLOv8m'ye göre ikna edici bir iyileştirme sunmamaktadır.

5.1.2 CNN modellerinin uygulama sonuçları

Görüntüler Yolo v8'e girildikten sonra, veri seti CNN modellerine ve çeşitli makine öğrenimi algoritmalarına beslenmektedir. Her model "makine öğrenimi algoritmalarından" birinde test edilmekte ve daha sonra yumurta türüne ve durumuna özgü eğitim ve test modelleri uygulandıktan sonra doğruluk ve hata çıkarılmaktadır.

CNN modeli uygulaması

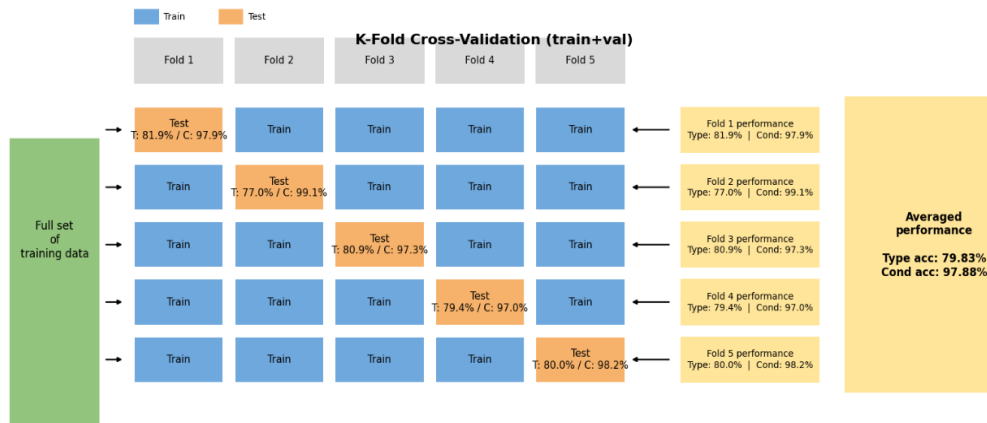
u deney için iki yumurta durumunu (kırık ve sağlam) ve üç farklı yumurta türünü (bıldırcın, ördek ve tavuk) kategorize etmek amacıyla çok çıkışlı bir Konvolüsyonel Sinir Ağı (CNN) modeli oluşturulmuştur. Yumurtanın türünü tahmin etmek ve kırık olup olmadığını belirlemek için model tek bir girdi katmanı, paylaşılan gizli katmanlar ve iki farklı çıktı dalı ile oluşturulmuştur. Model, mimarisi sayesinde yumurtanın türünü ve durumunu aynı anda sınıflandırabilmiştir. Adam optimize edicisi, modeli 32'lik bir parti boyutu ve 0,001'lik bir öğrenme oranı ile 50 epok üzerinde eğitmek için kullanılmıştır. Modelin etkinliğini değerlendirmek için, doğrulama verileri üzerindeki tahminler, iki karışıklık matrisi ve sınıflandırma raporları (kesinlik, geri çağırma ve F1-skoru içeren) kullanılarak her görev için ayrı ayrı incelenmiştir. Şekil 5.2 yumurta durumuna göre (kırık ve sağlam) sınıflandırma performansını gösterirken, Şekil 5.1 altı yumurta kategorisi için (örneğin, tavuk_kırık, ördek_sağlam, vb.) sınıflandırma performansını göstermektedir.



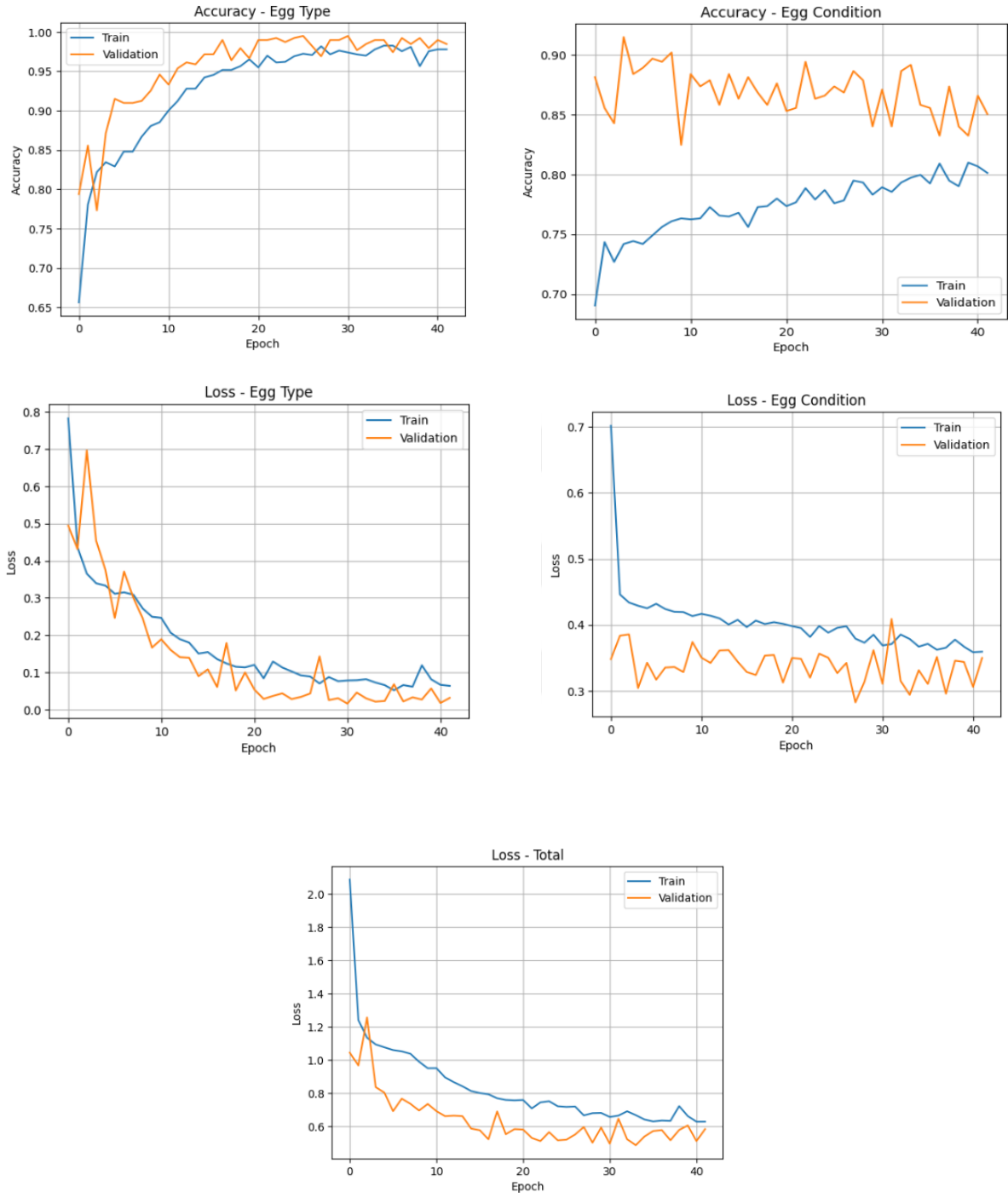
Şekil 5.1. CNN modelinin karışıklık matrisi

Çizelge 5.2 İşleme öncesi CNN modelini eğitme

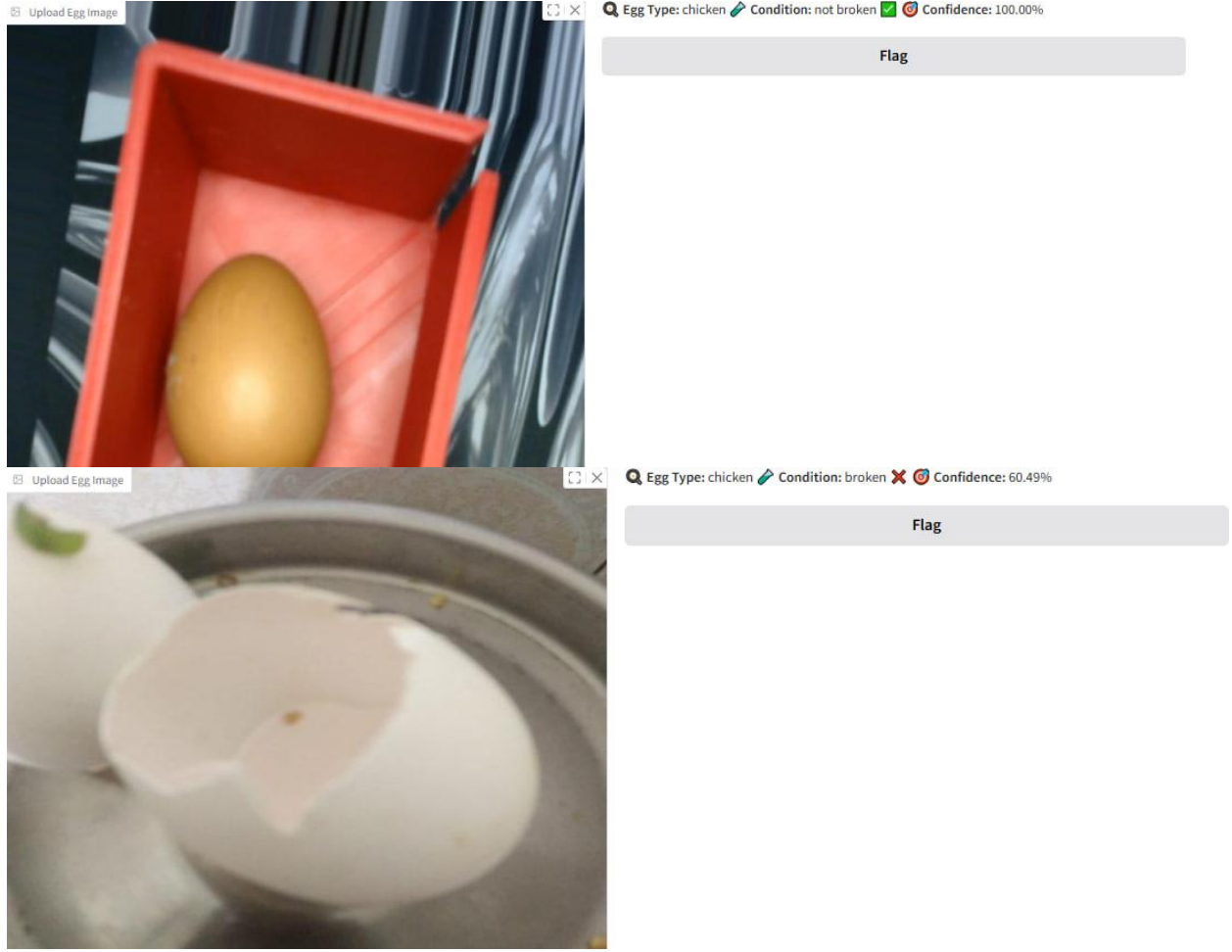
Sınıflandırma Raporu – Tür			
	Precision	Recall	f1-score
Tavuk	0.97	0.96	0.96
Ördek	0.97	0.98	0.98
Bıldırcın	1.00	1.00	1.00
Accuracy	0.97		
Sınıflandırma Raporu – Yumurta Durumu			
	Precision	Recall	f1-score
Kırık	0.93	0.97	0.95
Sağlam	0.95	0.90	0.92
Accuracy	0.94		



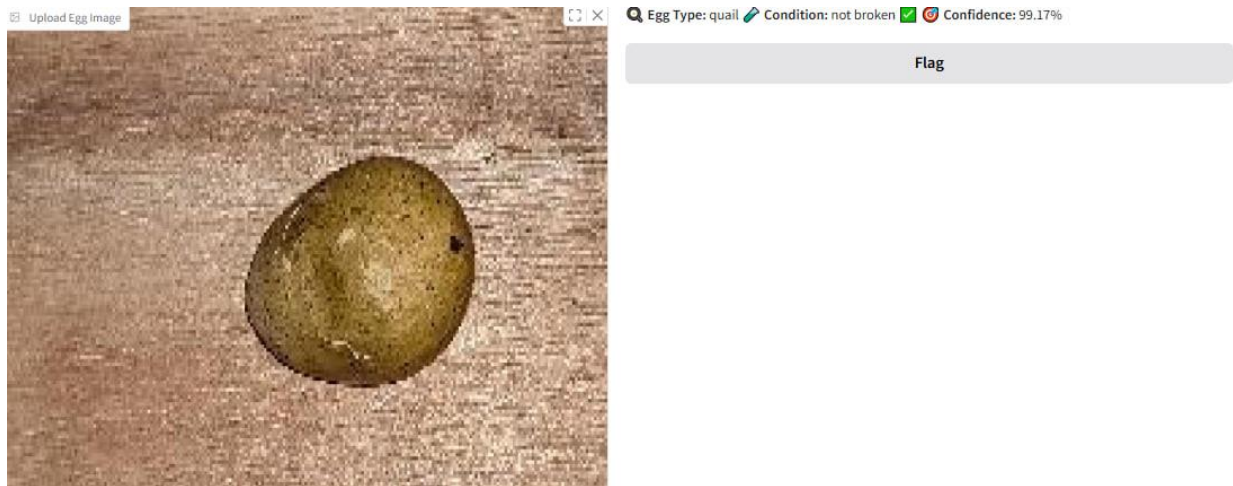
Şekil 5.2. CNN modelinin çapraz doğrulaması



Şekil 5.3. CNN model eğitim doğruluğu, doğrulama doğruluğu ve kayıp grafiği



Şekil 5.4. Tavuk yumurtası testi

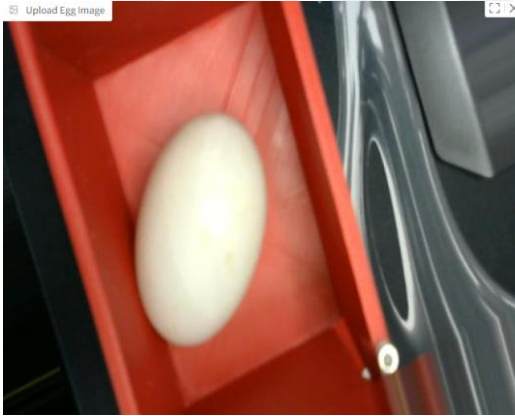




Q Egg Type: quail Condition: broken Confidence: 87.51%

Flag

Şekil 5.5. Bıldırcın yumurtası testi



Q Egg Type: duck Condition: not broken Confidence: 100.00%

Flag



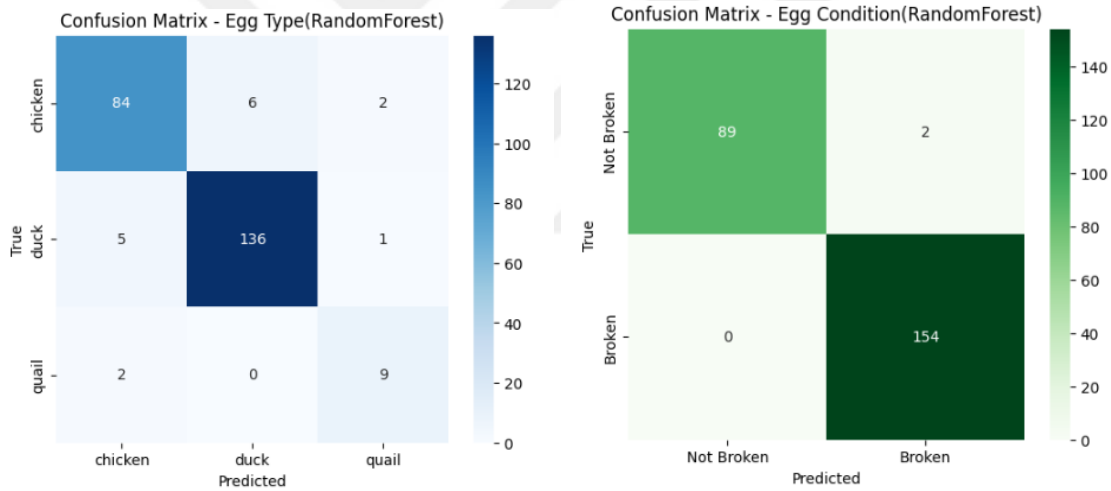
Q Egg Type: duck Condition: broken Confidence: 98.76%

Flag

Şekil 5.6. Ördek yumurtası testi

5.1.3 Rastgele Orman modeli uygulaması

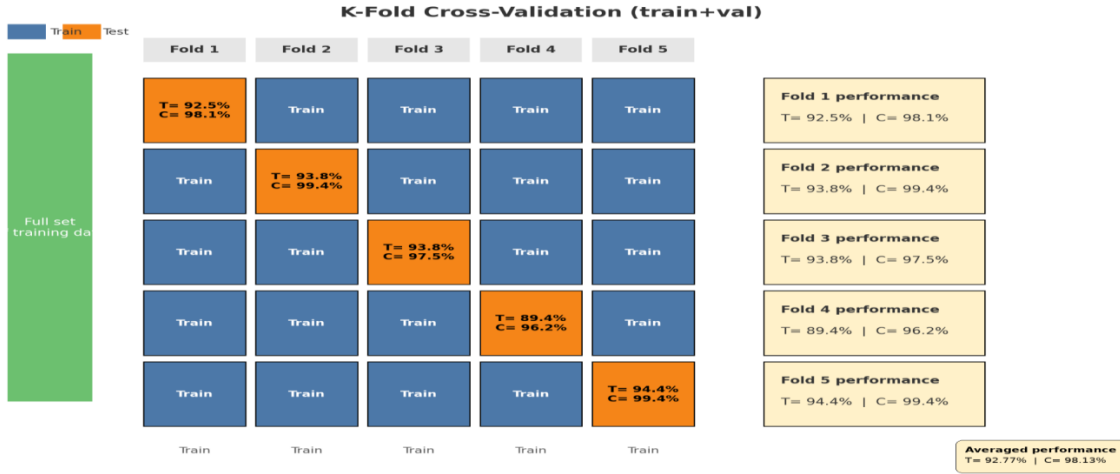
Yumurtaların tipini ve durumunu sınıflandırmak için yumurta görüntülerinden oluşan bir veri kümesi Rastgele Orman modeline tabi tutuldu. Tekrarlanabilirliği garanti altına almak için Rastgele Orman sınıflandırıcıları 100 tahmin edici (ağaç) ve 42 sabit rastgele durum kullanılarak oluşturuldu. İki ayrı sınıflandırıcı eğitildi: biri yumurta tipini (ör. bıldırcın, ördek veya tavuk) belirlemek için, diğeri ise yumurtanın kırık mı yoksa sağlam mı olduğunu belirlemek için. Her iki sınıflandırma görevi için de karışıklık matrislerine dayanarak, hassasiyet, geri çağırma ve F1 puanı gibi sınıflandırma performans ölçütleri Çizelge 5.3 ve Şekil 5.7 gösterilmiştir. Bu bulgular, modelin hasarlı yumurtaları dengeli bir doğruluk seviyesiyle tespit etme ve çeşitli yumurta tipleri arasında ayırım yapma kapasitesini göstermektedir.



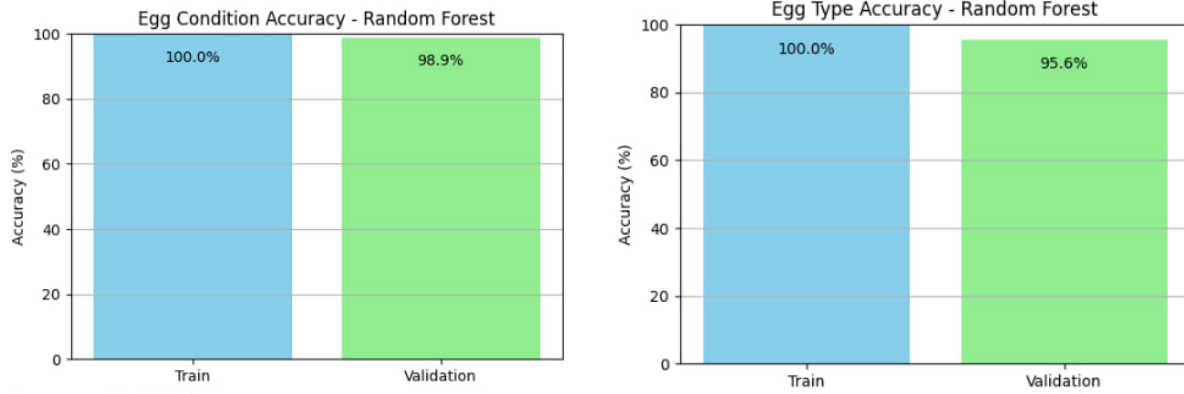
Şekil 5.7. Karışıklık Matrisi - Rastgele Orman

Çizelge 5.3. İşleme Öncesi Rastgele Orman Sınıflandırması

Sınıflandırma Raporu – TÜR			
	Precisio n	Recall	f1- score
Tavuk	0.92	0.91	0.92
Ördek	0.96	0.96	0.96
Bıldırcın	0.75	0.82	0.78
Accuracy	0.93		
Sınıflandırma Raporu – Yumurta Durumu			
		Recall	f1- score
Sağlam	1.00	0.98	0.99
Kırık	0.99	1.00	0.99
Accuracy	0.99		



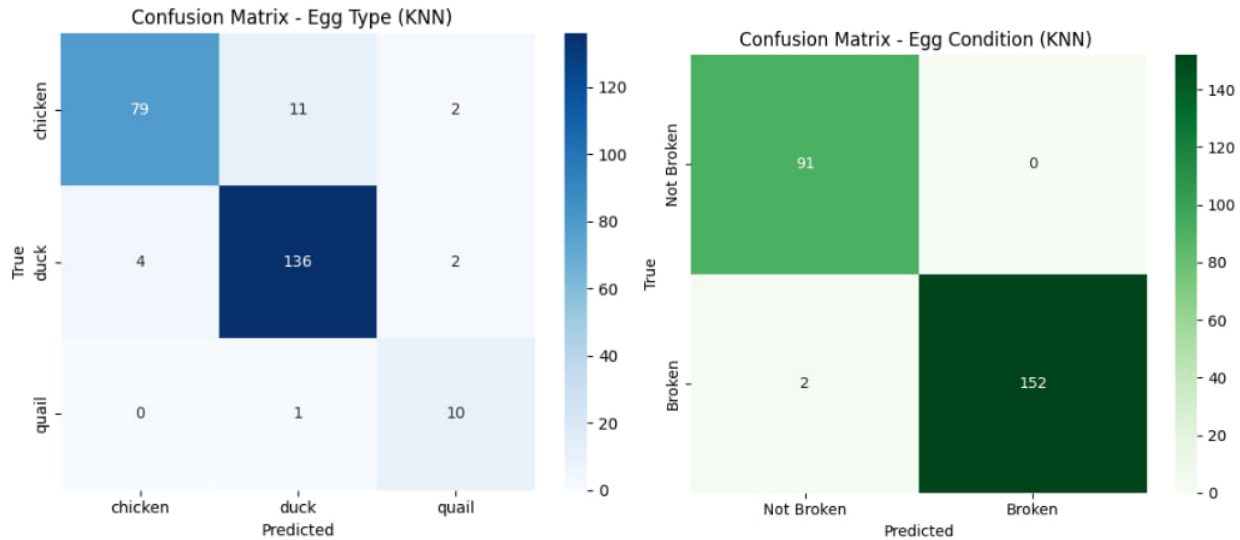
Şekil 5.8. Rastgele Orman modelinin çapraz doğrulaması



Şekil 5.9. Rastgele Orman modelinin eğitimini ve değişimini grafiği

5.1.4 K-En Yakın Komşular (KNN) modeli uygulaması

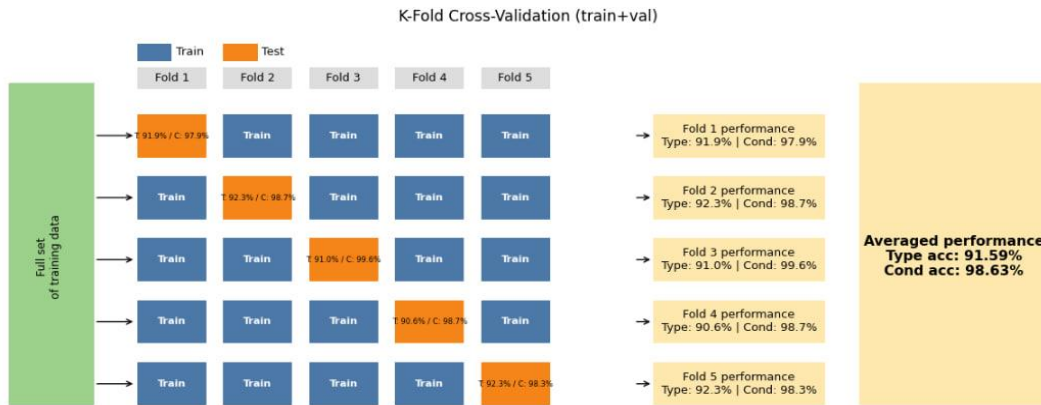
Bu deneyde yumurtaların türünü ve durumunu sınıflandırmak için veri setine K-En Yakın Komşular (KNN) modeli uygulandı. Basit ancak güvenilir bulguları korumak için, KNN modeli $k = 3$ komşu ile kuruldu. İki farklı KNN sınıflandırıcısı eğitildi: biri yumurtanın durumunu (kırık veya kırık değil) belirlemek, diğeri ise türünü (örneğin tavuk, ördek, bıldırcın) tahmin etmek için. Karmaşıklık matrislerine ve hassasiyet, geri çağırma ve F1 puanı gibi ölçütlere dayanarak, dört sınıfın her biri için performans derecelendirmeleri Çizelge 5.4 ve Şekil 5.10'da gösterilmiştir. Sonuçlar, KNN modelinin veri setini basit bir mesafeye dayalı benzerlik yöntemi kullanarak kategorize etmede başarılı olduğunu göstermektedir.



Şekil 5.10. Karmaşa Matrisi – KNN

Çizelge 5.4. İşleme öncesi K-En Yakın Komşular (KNN) sınıflandırması

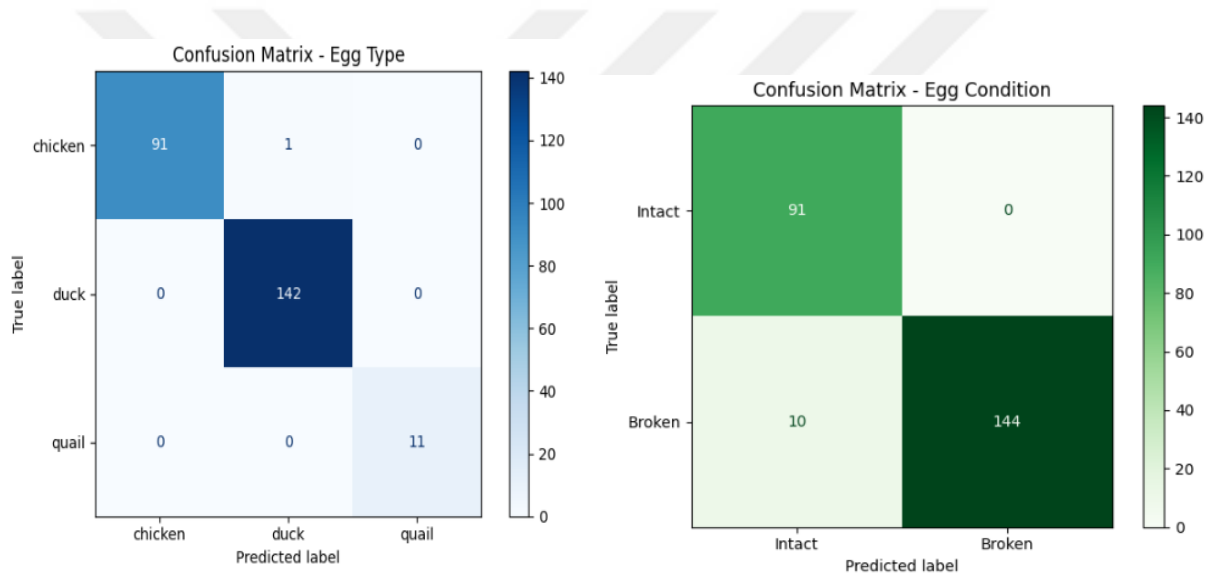
Sınıflandırma Raporu – TÜR			
	Precision	Recall	f1-score
Tavuk	0.95	0.86	0.90
Ördek	0.92	0.96	0.94
Bıldırcın	0.71	0.91	0.80
Accuracy	0.92		
Sınıflandırma Raporu – Yumurta Durumu			
	Precision	Recall	f1-score
Sağlam	0.98	1.00	0.99
Kırık	1.00	0.99	0.99
Accuracy	0.99		



Şekil 5.11. K-En Yakın Komşular (KNN) modelinin çapraz doğrulaması

5.1.5 ResNet50 model uygulaması

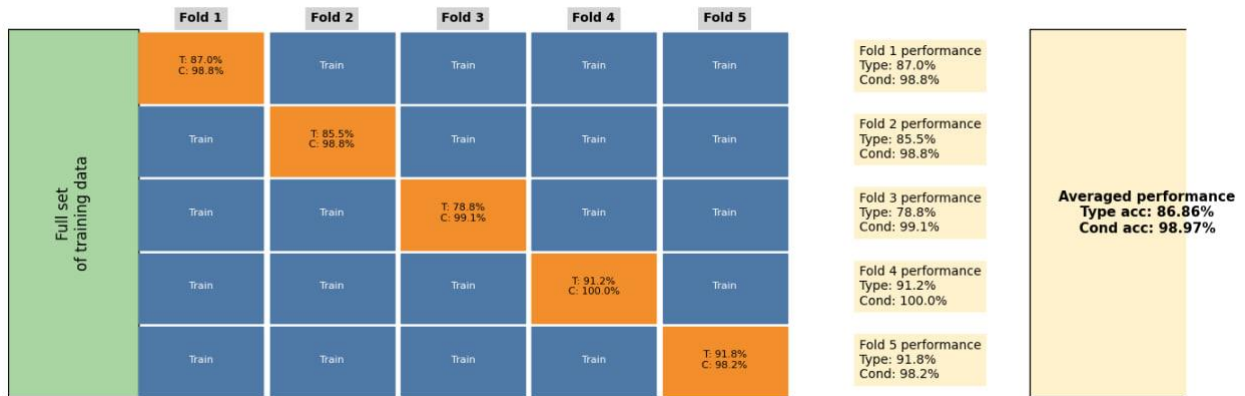
Yumurtanın türünü (bıldırcın, ördek veya tavuk) ve durumunu (kırık veya kırık değil) tahmin etmek için bu deneyde ResNet50 tabanlı ikili bir sınıflandırma oluşturuldu. Modeli eğitmek için 224 x 224 piksele yeniden boyutlandırılmış görüntüler kullanıldı. Ağın bir dalı ikili çapraz entropi kullanarak fiziksel durumu tespit edecek şekilde eğitilirken, diğer dal seyrek kategorik çapraz entropi kullanarak yumurta türlerini ayırt edecek şekilde ince ayarlandı. Her iki modeli de $1e-4$ öğrenme oranıyla optimize etmek için Adam kullanıldı. Karmaşıklık matrislerine ve hassasiyet, geri çağırma ve F1 puanı gibi ölçütlere dayanarak, dört sınıfın her biri için performans derecelendirmeleri Çizelge 5.5 ve Şekil 5.12'de gösterilmiştir.



Şekil 5.12. Karmaşa matrisi - ResNet50

Çizelge 5.5. İşleme öncesi Resnet50 sınıflandırması

Sınıflandırma Raporu – Tür			
	Precision	Recall	f1-score
Tavuk	1.00	0.99	0.99
Ördek	0.99	1.00	1.00
Bıldırcın	1.00	1.00	1.00
Accuracy	0.99		
Sınıflandırma Raporu – Yumurta Durumu			
	Precision	Recall	f1-score
Kırık	1.00	0.94	0.97
Sağlam	0.90	1.00	0.95
Accuracy	0.96		



Şekil 5.13. Resnet50 modelinin çapraz doğrulaması

5.2 İşlemeden sonra eğitim:

5.2.1 YOLO V8 modellerinin uygulama sonuçları

Tüm modeller işlemeden sonra aynı değerlerle uygulandı. görüntülere giden yollar, etiketler ve sınıf adları hakkında bilgi içeren yaml adlı bir dosya da vardır. Eğitim veya değerlendirme sırasında YOLO'ya geçirilir. Biçimi şöyledir

path: /content/drive/MyDrive/main_cropped_v3

train: main_tr

val: main_vl

nc: 6

names: ['chicken_not_broken', 'chicken_broken', 'duck_not_broken', 'duck_broken', 'quail_not_broken', 'quail_broken']

Çizelge (5.6)'de YOLOv8 eğitimi ve işlemeden sonra hangi tiplerin kullanıldığı gösterilmektedir (Kesme işlemi):

Types of yolo v 8	Precision	Recall	mAP@0.5	mAP@0.5:0.95	Fitness
Yolo v8 n	0.7326	0.1567	0.4111	0.3737	0.3774
Yolo v8 s	0.7813	0.5531	0.6618	0.6462	0.6478
Yolo v8 m	0.6201	0.898	0.7677	0.7655	0.7657
Yolo v8 l	0.6385	0.7744	0.8043	0.8042	0.8042
Yolo v8 x	0.671	0.7392	0.8153	0.8153	0.8153

Çizelge 5.6. YOLOv8 modellerinin işlendikten sonra eğitilmesi

Maksimum $mAP@0.5$ ve $mAP@0.5:0.95$ (0.8153) değerleri YOLOv8x tarafından elde edilmiş ve tüm yumurta tipleri ve koşulları için olağanüstü genel tespit doğruluğu göstermiştir. Ancak bazı daha küçük modellerle karşılaştırıldığında, hassasiyeti (0.671) daha düşüktü; bu da genel olarak daha fazla hedef keşfetmesine rağmen daha fazla yanlış pozitif sonuç ürettiğini göstermektedir.

0,8043'lük $mAP@0,5$ ve 0,8042'lik $mAP@0,5:0,95$ değerleriyle YOLOv8l, biraz daha iyi bir hassasiyetle (0,6385) oldukça rekabetçi bir performans sergileyerek ikinci sırada yer aldı. Hesaplama maliyeti bir sınırlama olmadığında, geri çağırma ve doğruluk arasındaki denge nedeniyle güçlü bir seçimdir.

Tüm sürümler arasında en yüksek geri çağırma değerine (0,898) sahip olan YOLOv8m, yumurta kalitesi izleme sistemleri için çok önemli olan kırık yumurtalar da dahil olmak üzere neredeyse tüm gerçek yumurta tespitlerini tespit edebildiği için özellikle dikkat çekiciydi. İyi bir $mAP@0,5:0,95$ (0,7655) değeriyle, dengeleme açısından genel olarak en iyi performansı göstermektedir. %60,9 daha yüksek $mAP@0,5$ (0,6618'e karşı 0,4111) ve önemli ölçüde daha yüksek hassasiyet (0,7813'e karşı 0,7326) ile YOLOv8s, her kategoride daha hafif YOLOv8n'den daha iyi performans gösterdi. Ancak geri çağırması hâlâ sınırlıydı (0,5531), bu nedenle YOLOv8m veya l gibi modellerden daha doğru olsa da daha fazla gerçek vakayı kaçırdı.

Beklendiği gibi, en küçük model olan YOLOv8n, özellikle geri çağırma (0,1567) ve $mAP@0,5$ (0,4111) değerlerinde en kötü performansı gösterdi. Son derece düşük kaynaklı cihazlar veya tespit hızının doğruluktan daha önemli olduğu gerçek zamanlı uygulamalar için, basitliği yine de haklı görülebilir.

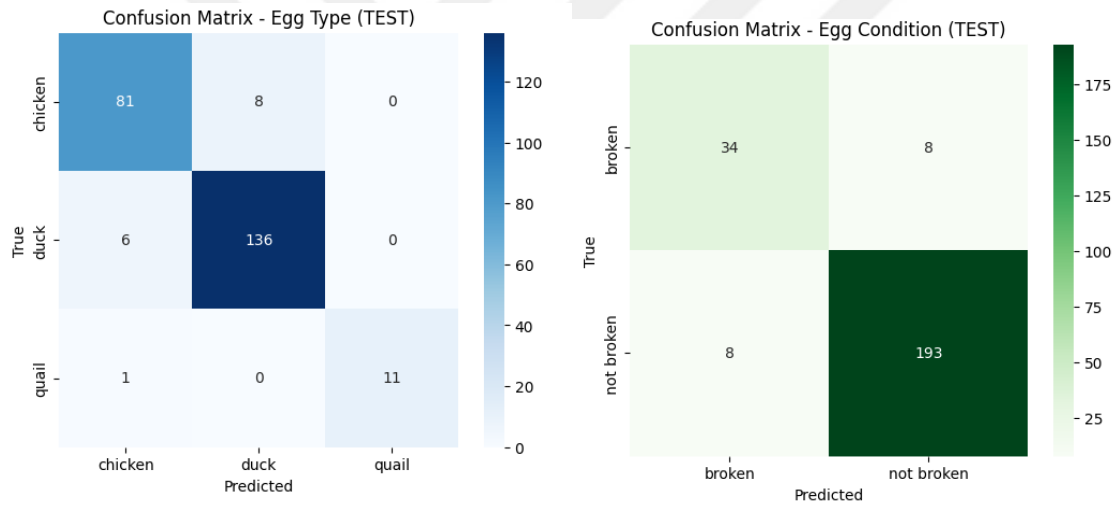
YOLOv8m, değerlendirilen tüm varyasyonlar arasında bu yumurta sınıflandırma ataması için en iyi model gibi görünüyor. Sağlam mAP değerlerini, saygın hassasiyeti ve olağanüstü geri çağırma bir araya getirerek, kaçırılan tespitleri azaltmak için mükemmel. Bu da onu, gömülü kümes hayvanları çiftlikleri veya gerçek zamanlı sınıflandırma sistemleri gibi hassasiyet, hassasiyet ve model boyutunun dengelenmesi gereken gerçek dünya kullanımları için ideal hale getiriyor.

5.2.2 CNN modellerinin uygulama sonuçları

CNN model uygulaması:

Bu deneyde, yumurta türü (örneğin tavuk, ördek, bıldırcın) ve durumu (kırık veya kırık değil), bir Evrişimsel Sinir Ağı (CNN) modeli kullanılarak sınıflandırılmıştır. 0,001 öğrenme oranına ve 32 parti boyutuna sahip bir Adam optimizasyon aracı, modeli 50 dönem boyunca eğitmek için kullanılmıştır. Her iki çıktı dalı için de veri setine beş sınıflandırma hedefi eklenmiştir: iki durum (kırık veya kırık değil) ve üç farklı yumurta türü.

Ağ mimarisi, biri yumurta türünü tahmin etmek, diğeri ise durumunu belirlemek için olmak üzere iki softmax katmanı kullanılarak çoklu çıktı ortamında eğitilmiştir. Evrişimsel katmanları, maksimum havuzlama, yoğun ve bırakma katmanları izlemiştir. Çizelge 5.7 ve Şekil 5.14'de her iki sınıflandırma görevi için sınıflandırma performansını değerlendirmek amacıyla karışıklık matrisleri ve doğruluk, kesinlik, geri çağırma ve F1 puanı gibi önemli metrikler kullanılmıştır.



Şekil 5.14. CNN modelinin Karışıklık Matrisi

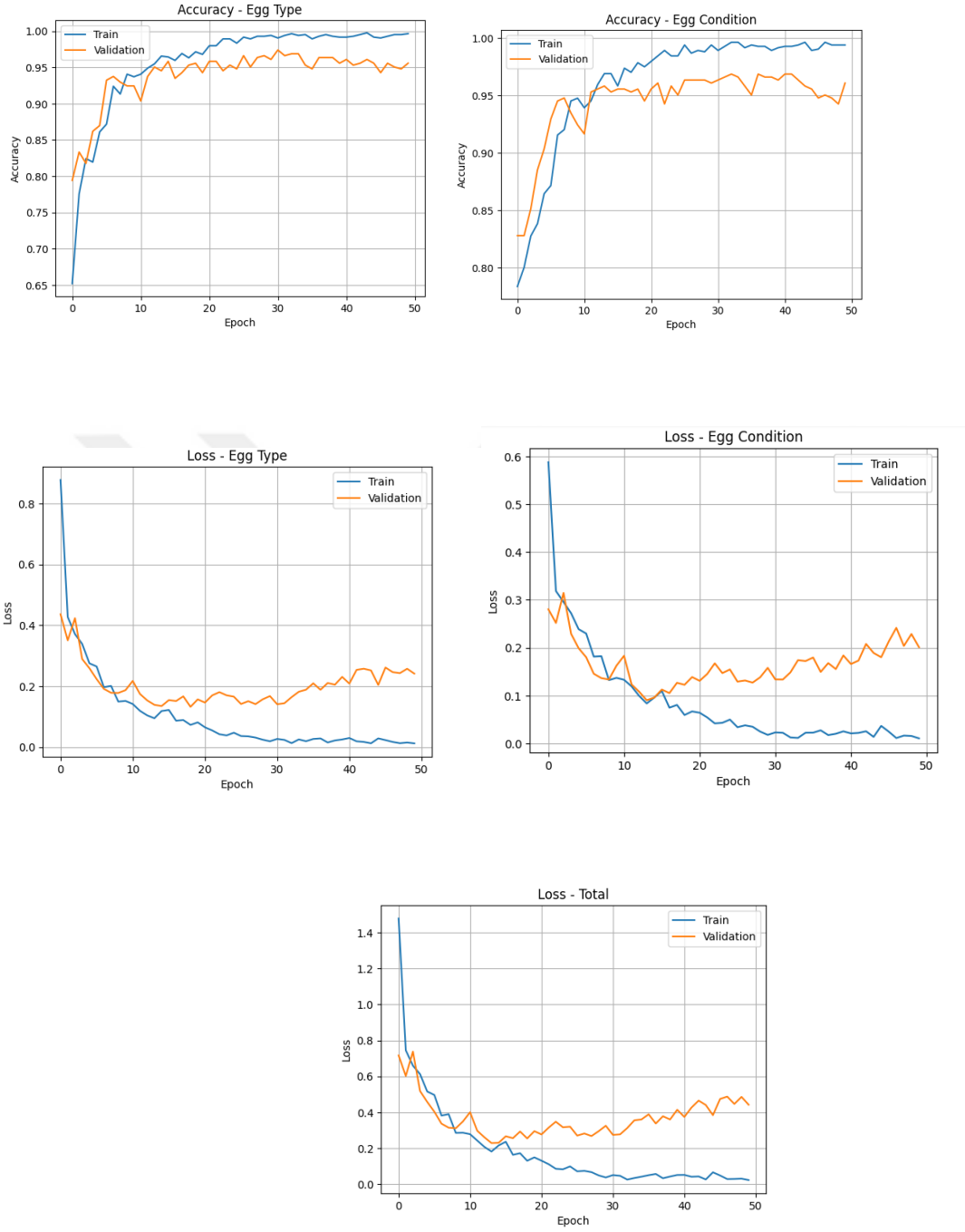
Çizelge 5.7. İşlemden Sonra CNN Modelinin Eğitimi

Sınıflandırma Raporu – TÜR			
	Precision	Recall	f1-score
Tavuk	0.92	0.91	0.92
Ördek	0.94	0.96	0.95
Bıldırcın	1.00	0.92	0.96
Accuracy	0.94		
Sınıflandırma Raporu – Yumurta Durumu			
	Precision	Recall	f1-score
Kırık	0.81	0.81	0.81
Sağlam	0.96	0.96	0.96
Accuracy	0.93		

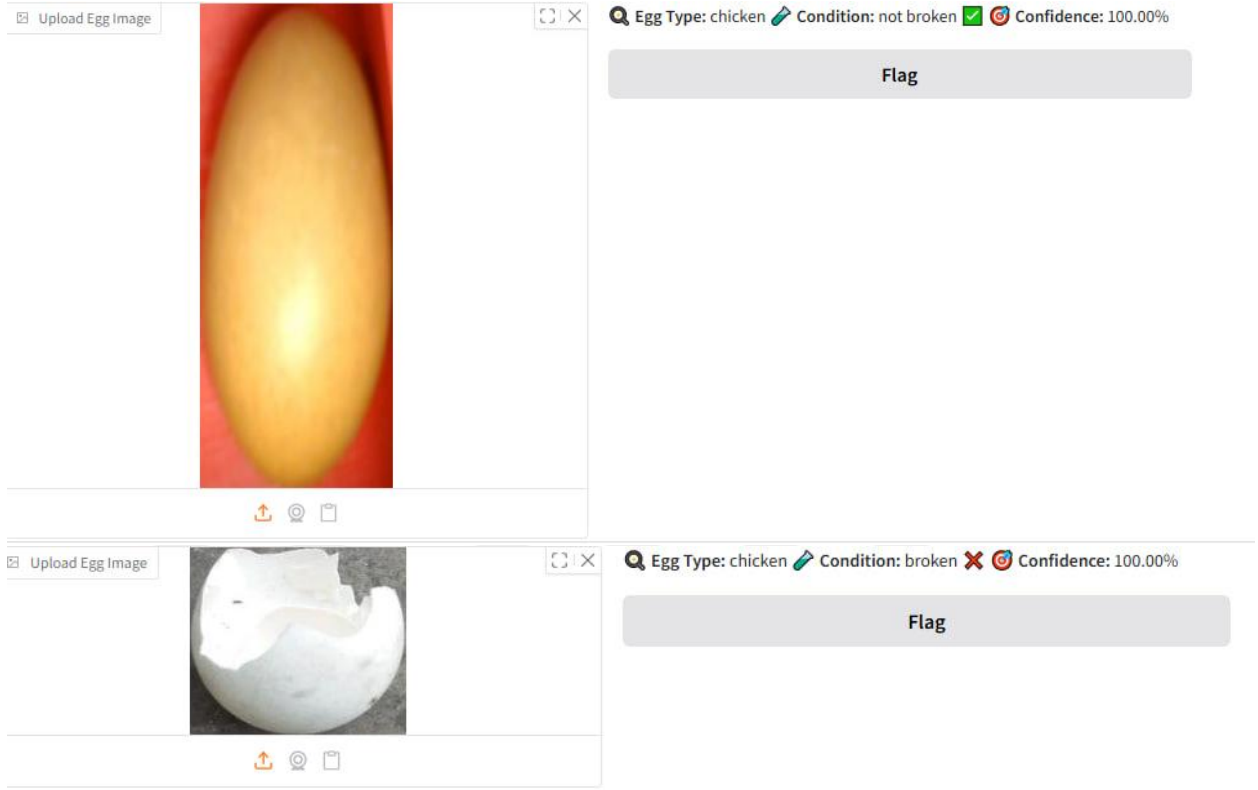
K-Fold Cross-Validation (train+val)					
	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Iteration 1	T: 93.1% / C: 94.3%	Train	Train	Train	Train
Iteration 2	Train	T: 94.7% / C: 93.1%	Train	Train	Train
Iteration 3	Train	Train	T: 96.7% / C: 95.1%	Train	Train
Iteration 4	Train	Train	Train	T: 93.5% / C: 93.5%	Train
Iteration 5	Train	Train	Train	Train	T: 94.3% / C: 94.7%

Averaged performance:
Type acc: 94.46% | Cond acc: 94.14%

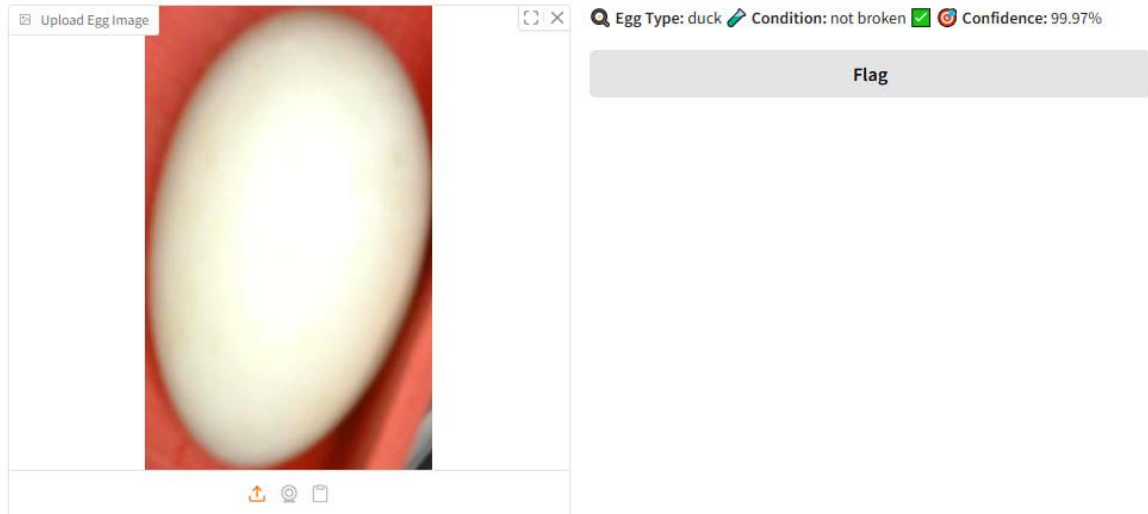
Şekil 5.15. CNN modelinin çapraz doğrulaması

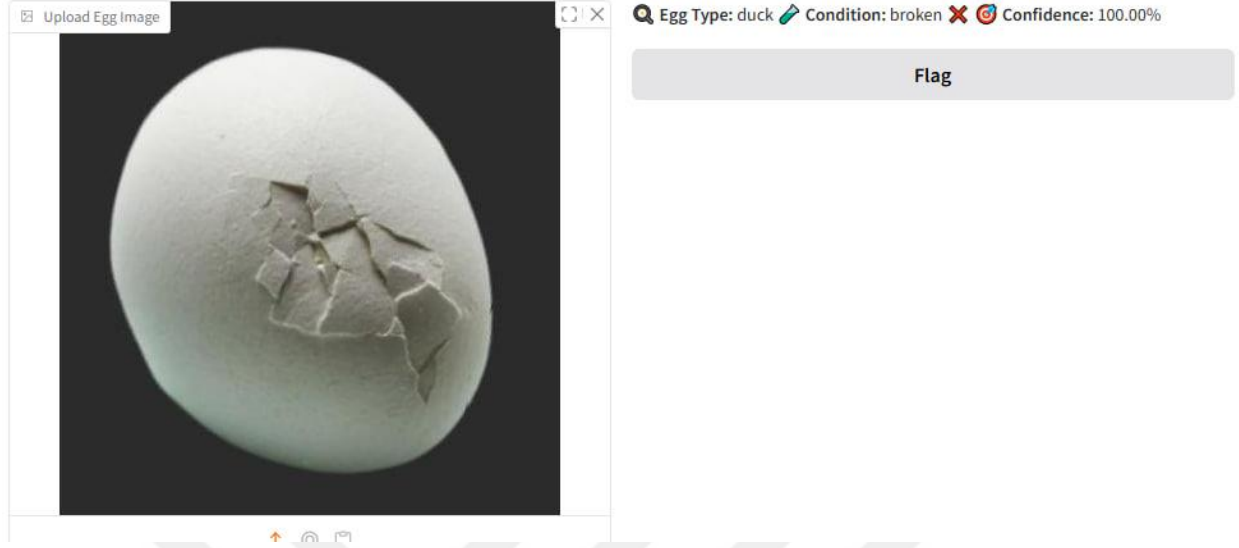


Şekil 5.16. CNN model eğitim doğruluğu, doğrulama doğruluğu ve kayıp grafiği

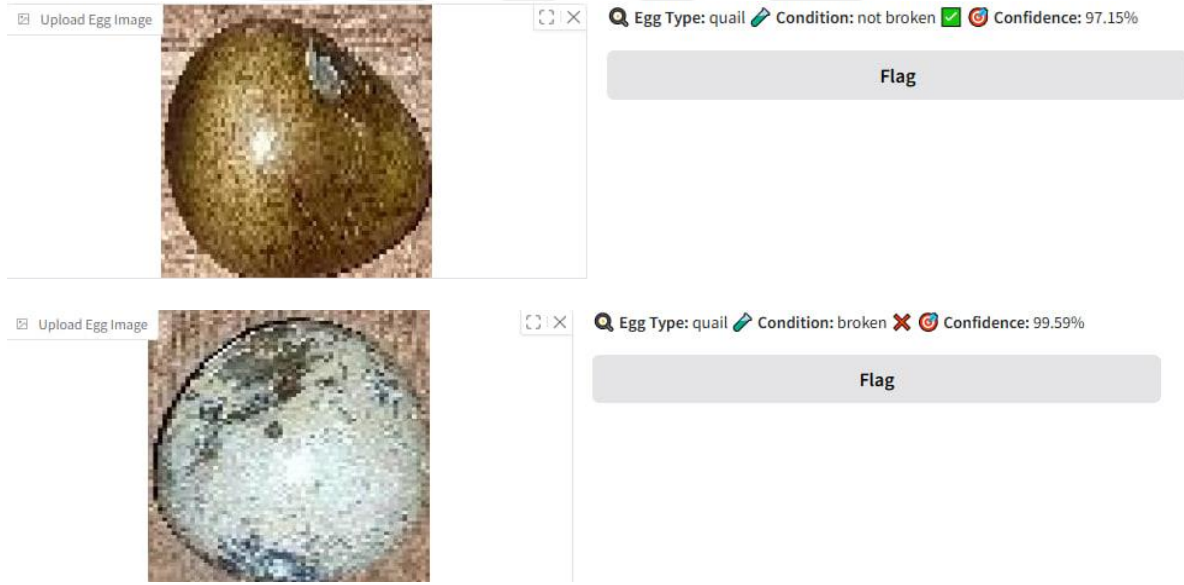


Şekil 5.17. Tavuk yumurtası testi





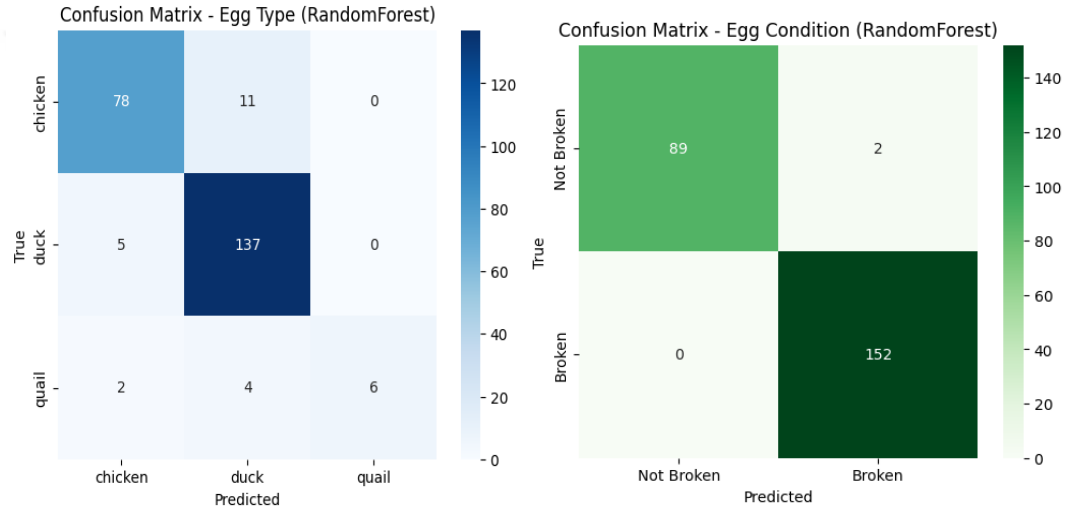
Şekil 5.18. Ördek yumurtası testi



Şekil 5.19. Bıldırcın yumurtası testi

5.2.3 Rastgele Orman modeli uygulaması

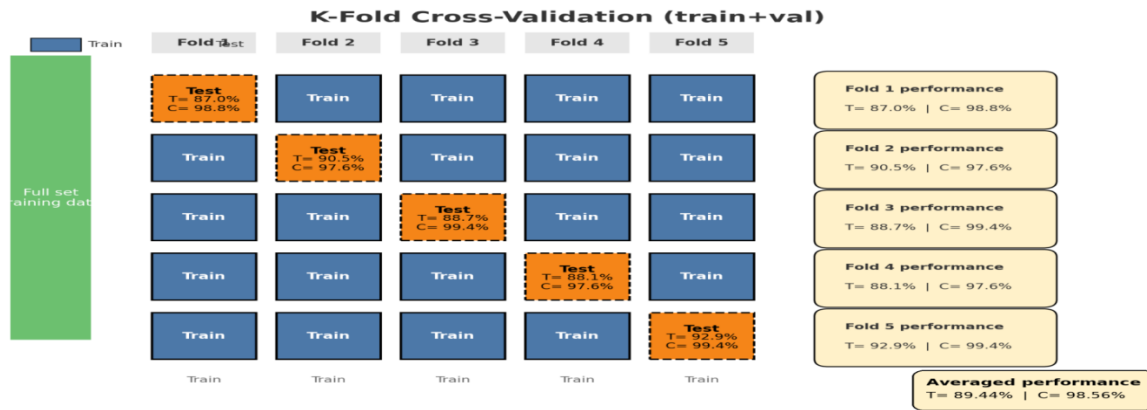
Yumurtaların türünü ve durumunu sınıflandırmak için bu deneyde bir Rastgele Orman sınıflandırıcısı eğitildi ve değerlendirildi. Fotoğraflar özellik vektörlerine düzleştirilip 64x64 piksele küçültüldükten sonar.. İki farklı Rastgele Orman modeli eğitildi: biri durumu (kırık veya kırık değil) belirlemek, diğeri ise yumurta türünü (örneğin tavuk, ördek, bıldırcın) tahmin etmek için. Tutarlılığı garantilemek için, her iki sınıflandırıcı da 100 karar ağacı (tahmin edici) ve 42 sabit rastgele durum kullanılarak oluşturuldu Çizelge 5.8 ve Şekil 5.20 gösterilmiştir



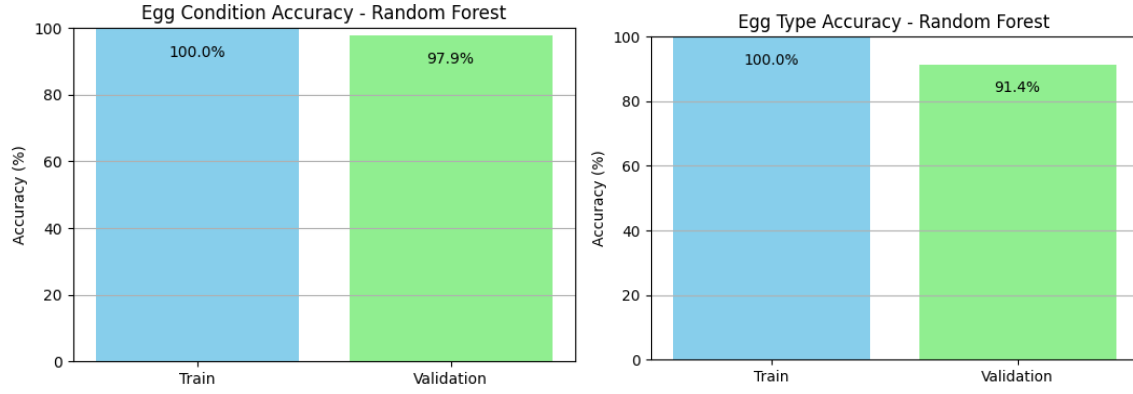
Şekil5.20. Karışıklık matrisi - Rastgele Orman

Çizelge 5.8. İşleme sonrası Rastgele Orman sınıflandırması

Sınıflandırma Raporu – Tür			
	Precision	Recall	f1-score
Tavuk	0.92	0.88	0.90
Ördek	0.90	0.96	0.93
Bıldırcın	1.00	0.50	0.67
Accuracy	0.91		
Sınıflandırma Raporu – Yumurta Durumu			
	Precision	Recall	f1-score
Kırık	1.00	0.98	0.99
Sağlam	0.99	1.00	0.99
Accuracy	0.99		



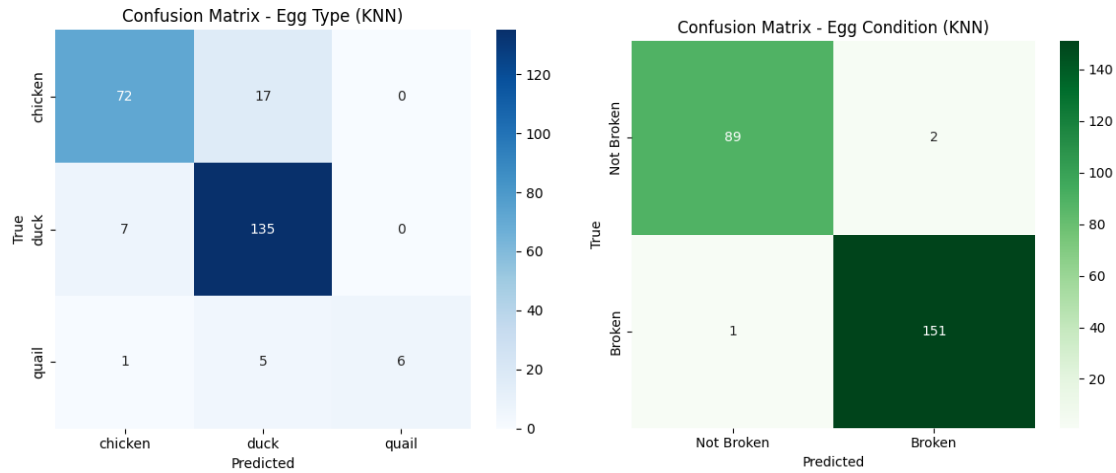
Şekil 5.21. Rastgele Orman modelinin çapraz doğrulaması



Şekil 5.22. Rastgele Orman modelinin eğitimini ve değişimini grafiği

5.2.4 K-En Yakın Komşular (KNN) modeli uygulaması

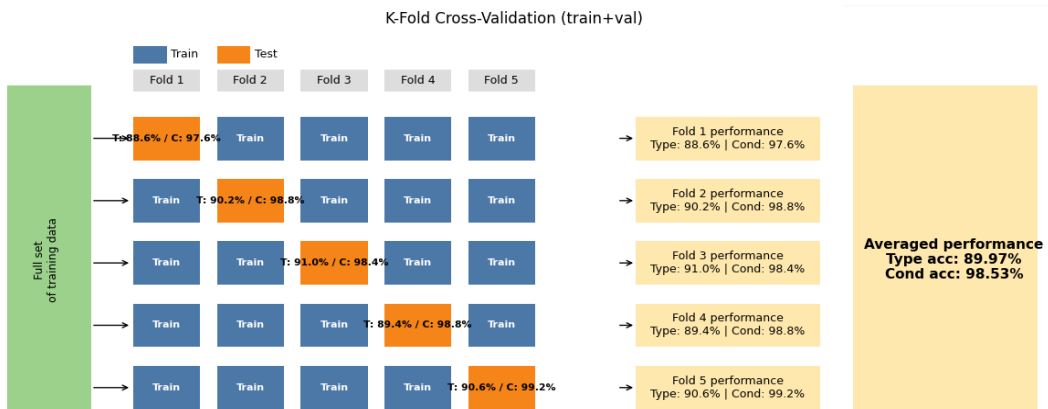
Yumurtaların türleri ve koşulları, $k = 3$ olan bir K-En Yakın Komşular (KNN) modeli kullanılarak sınıflandırıldı. Her görüntü 64x64 piksele küçültüldü. İki farklı KNN sınıflandırıcı kullanıldı: biri koşul (kırık veya kırık değil) ve diğeri yumurta türü için. Çizelge 5.7 ve Şekil 5.23'da belirtildiği gibi, model performansını değerlendirmek için hassasiyet, geri çağırma, F1 puanı ve karışıklık matrisleri gibi değerlendirme kriterleri kullanıldı.



Şekil 5.23. Karışıklık matrisi – KNN

Çizelge 5.9. İşleme sonrası K-En Yakın Komşular (KNN) sınıflandırması

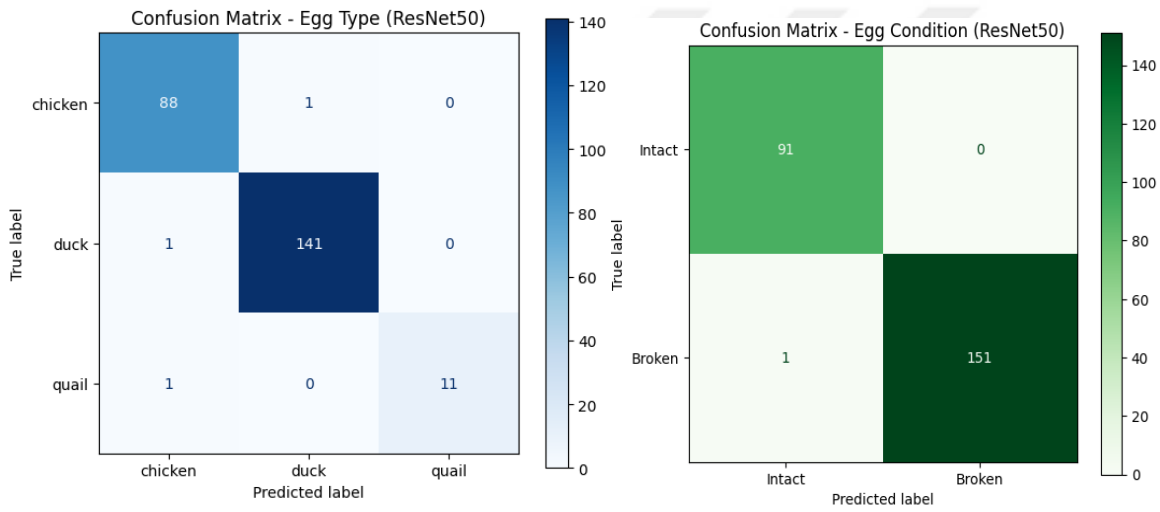
Sınıflandırma Raporu – Tür			
	Precision	Recall	f1-score
Tavuk	0.90	0.81	0.85
Ördek	0.86	0.95	0.90
Bıldırcın	1.00	0.50	0.67
Accuracy	0.88		
Sınıflandırma Raporu – Yumurta Durumu			
	Precision	Recall	f1-score
Sağlam	0.99	0.98	0.98
Kırık	0.99	0.99	0.99
Accuracy	0.99		



Şekil 5.24. K-En Yakın Komşular (KNN) modelinin çapraz doğrulaması

5.1.5 ResNet50 model uygulaması

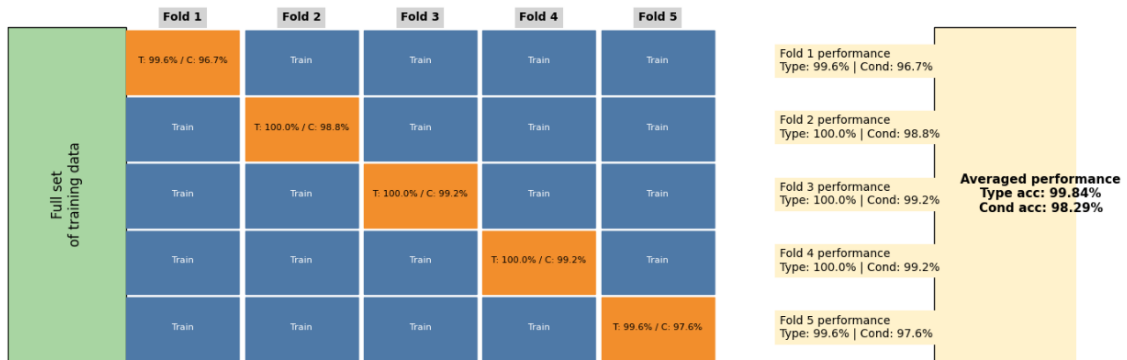
Bu çalışmada, farklı yumurta türlerini (tavuk, ördek ve bıldırcın gibi) ve durumlarını ((kırık veya kırık değil) sınıflandırmak için ResNet50 tabanlı bir evrişimli sinir ağı kullanılmıştır. Model, iki adet tam bağlantılı softmax çıkış başlığı, bir Küresel Ortalama Havuzlama katmanı ve önceden eğitilmiş ImageNet ağırlıkları kullanmıştır. 32'lik bir toplu iş boyutu ve 224x224 piksellik bir giriş görüntü boyutuyla 20 dönem için eğitmek üzere Adam optimizasyon aracı kullanılmıştır. Çizelge 5.8 ve Şekil 5.25'da belirtildiği gibi model performansını değerlendirmek için hassasiyet, geri çağırma, F1 puanı ve karışıklık matrisleri gibi değerlendirme kriterleri kullanıldı.



Şekil 5.25. Karışıklık matrisi - Resnet50

Çizelge 5.10. İşleme Sonrası Resnet50 Sınıflandırması

Sınıflandırma Raporu – TÜR			
	Precision	Recall	f1-score
Tavuk	0.98	0.99	0.98
Ördek	0.99	0.99	0.99
Bıldırcın	1.00	0.92	0.96
Accuracy	0.99		
Sınıflandırma Raporu – Yumurta Durumu			
	Precision	Recall	f1-score
Sağlam	0.99	1.00	0.99
Kırık	1.00	0.99	1.00
Accuracy	1.00		



Şekil 5.26. Resnet50 modelinin çapraz doğrulaması

Çizelge 5.11. Karşılaştırma Sonuçları

Önerilen Yöntem	işlemeden önce (kırpma)	Tavuk yumurtası, ördek yumurtası, bıldırcın yumurtası	Yolo v 8 Cnn model Random Forest The K-Nearest Neighbors (KNN) ResNet50	80% (durum)ve (tür) 94% (durum) / 97% (tür) 99% (durum) 93% (tür) 99% (durum) / 92% (tür) 96%(durum) / 99% (tür)
Önerilen Yöntem	İşlemeden sonra (kırpma)	Tavuk yumurtası, ördek yumurtası, bıldırcın yumurtası	Yolo v 8 Cnn model Random Forest The K-Nearest Neighbors (KNN) ResNet50	81%(durum)ve (tür) 93%(durum) / 94% (tür) 99% (durum) / 91% (tür) 99% (durum) / 88% (tür) 100% (durum) / 99% (tür)

Bunlar, CNN modelinin üç derin öğrenme algoritmasıyla uygulanmasının doğruluk oranlarıdır. “Yolo v 8” uygulamasında ve ResNet50 modelinde “kırpma” işleminin uygulanmasından sonra oranların arttığını not ediyoruz.

5.3 Önceki Çalışmalarla Karşılaştırma

Bu bölüm, önerilen metodolojinin etkinliğini ve sonuçlarını yumurta türü ve durum tanıma üzerine önceki araştırmalarla karşılaştırır. Çizelge 5.12, farklı yıllar ve yazarlardan önerilen metodolojilerin ve alternatif yaklaşımların karşılaştırmalı bir özetini sunar.

Çizelge 5.12. Önceki Çalışmalarla Karşılaştırmalar

Yazarın	Veri Seti	Yumurtalar	Modeller	Doğruluk
(Li et al., 2020)	Bağlamdaki Ortak Nesneler (COCO) veri seti	zemin yumurtaları	faster region-based CNN (faster R-CNN)	91.9%–100%
(Huang et al., 2023)	kamerayla fotoğraf çekildi	kırık yumurta	improved YOLOv5	96.4%
(Datta et al., 2019)	kamerayla fotoğraf çekildi	Çatlak Tavuk yumurtaları	Faster Region based Convolutional Neural Network (R-CNN)	75%
(Sánchez et al., 2023)	yerel bir çiftlikten kullanıldı	kahverengi tavuk yumurtaları	VGG19	97.33%
(Brasil et al., 2021)	yerel çiftliklerden elde edildi	Bıldırcın yumurtasının kalitesi	PLSDA and SVM	80%

Çizelge 5.12'te gösterildiği gibi, birçok araştırmacı yumurta sınıflandırması üzerinde çalışmaktadır. Çeşitli yumurta koşulları ve türleri kullanmışlardır. Bu çalışma, yumurta koşulu ve türü (Tavuk yumurtaları, Ördek yumurtaları, Bıldırcın yumurtaları) için ön ve son işleme olmak üzere iki yeni veri kümesi oluşturur. Derin öğrenme algoritmalarıyla, kısa bir yürütme süresinde mükemmel sonuçlar elde edilir. Önerilen metodolojinin, farklı modeller kullanarak hem yumurta türü hem de koşulu için yüksek tanıma oranları elde etmedeki çok yönlülüğü, sağlam ve doğru biyometrik tanımlama uygulamaları için potansiyelini göstermektedir

6. SONUÇ VE ÖNERİLER

6.1 Sonuç

Bu çalışmayla ilgili en önemli bulgular şunlardır:

1. Çalışmada, yumurtanın toplumdaki önemi keşfedildi ve durumu veya türünü tespit etmek için nasıl birden fazla adım atıldığı. Tezde yer alan tüm çalışmalar, birçok ülkede endüstri ve ticaret gibi toplumdaki birçok alanda yumurtanın önemi için çalışmaktadır. Yumurta üreten kümes hayvanlarının önemi daha az değildir ve bu nedenle ürünler üzerinde etkileri olduğu için uygun gıdalarla beslenmelidir. Yumurtaların insanlar için birçok faydası vardır ve tam bir gıda olarak kabul edilir. Bu temelde, kalitesi bilinmelidir.
2. Yumurta türünü ve durumunu sınıflandırmak için bu türlerden (tavuk yumurtası, ördek yumurtası, bıldırcın yumurtası) 1.896 örnek içeren yeni bir veri kümesi oluşturuldu ve veri kümesi işlendikten (kırıldıktan) sonra 1.469 oldu.
3. Bu çalışmada, işlemeden önce (kırpma), üç farklı derin öğrenme ve makine öğrenmesi modelini eğitmek, doğrulamak ve test etmek için kullanılmıştır: Rastgele Orman, K-En Yakın Komşular (KNN) ve ResNet50. Modellerin yumurta türlerini sınıflandırma doğruluk puanları şu şekildedir: Rastgele Orman %93, KNN %92 ve ResNet50 %99. Modellerin (kırık veya sağlam) yumurtaları sınıflandırma doğruluk puanları şu şekildedir Rastgele Orman %99 ,KNN %99 ve ResNet50 %96.
4. İşlemeden sonra (kırpma), üç farklı modeli eğitmek, doğrulamak ve test etmek için kullanıldı: Rastgele Orman, K-En Yakın Komşular (KNN) ve ResNet50. Modellerin yumurta türlerini sınıflandırma doğruluk puanları şu şekildedir: Rastgele Orman %91, KNN %88 ve ResNet50 %99. Yumurtanın durumunu (kırık veya sağlam) sınıflandırma doğruluk puanları şu şekildedir Rastgele Orman %99,KNN %99 ve ResNet50 %100.
5. Önerilen model, önceden işlenmiş veri kümeleri üzerinde eğitildi, doğrulandı ve test edildi. Yumurta türü sınıflandırması için model, ön işlemeden önce %97, ön işlemeden sonra ise %94 doğruluk elde etti. Yumurta durumu sınıflandırması için doğruluk, ön işlemeden önce %94, ön işlemeden sonra ise %93 idi.
6. Ayrıca bu çalışmada Yolo v8 sınıflandırma yaklaşımı kullanılmıştır. Analiz, tüm Yolo v 8 modellerini kullanarak işlemeden önce altı türü sınıflandırmak için doğruluk oranının %80 işlemeden önce arasında değiştiğini ve işlemeden sonra doğruluğun %81 işlendikten sonra arasında değiştiğini göstermektedir.

6.2 Gelecekteki Çalışmalar İçin Öneriler

Bu tez için bazı potansiyel gelecekteki yollar şunlardır:

1. Yeni modeli (CNN) farklı veri kümelerine uygulanabilir.
2. Yumurta durumlarını ve türünü sınıflandırmak için yumurta veri kümesinde diğer evrimsel sinir ağı modelleri ("Rastgele Orman", "K-En Yakın Komşular (KNN)" ve "ResNet50" hariç) kullanılabilir.
3. Önerilen yöntem daha büyük ve daha çeşitli veri kümelerinde değerlendirmek, geliştirilebilirliğini ve gerçek dünya senaryolarına uygulanabilirliğini doğrulamak için yararlı olacaktır.
4. Yumurta türü veri kümeleri ve diğerleri için CNN modelinin performansını iyileştirmek için yeni parametreler eklenebilir.

KAYNAKLAR

- Ab Nasir, A. F., Sabarudin, S. S., Abdul Majeed, A. P. P., & Abdul Ghani, A. S. (2018). Automated egg grading system using computer vision: Investigation on weight measure versus shape parameters. *IOP Conference Series: Materials Science and Engineering*, 342(1), 012003.
- Abdelrahman, M., & Shukla, M. (2004). Artificial neural networks based steady state security analysis of power systems. *Proceedings of the Thirty-Sixth Southeastern Symposium on System Theory*, 266–269.
- Abdullahkhanuet22. (n.d.). *Eggs images classification: Damaged or not* [Dataset]. Kaggle. <https://www.kaggle.com/datasets/abdullahkhanuet22/eggs-images-classification-damaged-or-not/data>
- Adjailia, F., & Adjailia, S. (2020). A review of activation function for artificial neural network. *International Journal of Computer Applications*, 975, 8887.
- Agarwal, M., & Mascarenhas, S. (2021, November 19). *An analysis of the architecture frameworks for image classification using VGG16, VGG19, and ResNet50*. In *Proceedings of the International Conference on Disruptive Technologies for Multidisciplinary Research and Applications (CENTCON)* (Vol. 1, pp. 96–99). IEEE.
- Albashish, D., Al-Sayyed, R. M. H., Abdullah, A., Ryalat, M. H., & Almansour, N. (2021). Deep CNN model based on VGG16 for breast cancer classification. In *2021 International Conference on Information Technology (ICIT)* (pp. 805–810). IEEE.
- Al-Gaashani, M. S. A. M., Samee, N. A., Alkanhel, R., Atteia, G., Abdallah, H. A., & Alabdulhafith, M. (2024). Deep transfer learning with gravitational search algorithm for enhanced plant disease classification. *Heliyon*, 10(7), e04998.

- Alqahtani, A., Song, Y. X., Lee, C. P., & Lim, K. M. (2023, July 14). Plant-CNN-ViT: Convolutional neural network ensemble and vision transformer used for plant classification. **Plants*, 12*(14), 2642.
- Ashar, N., Raut, G., Trivedi, V., & Kumar, A. (2024). QuantMAC: Enhancing hardware performance in DNNs with quantize enabled multiply-accumulate unit. *IEEE Access*, 12, 43600–43612.
- B. Xiao, M. Nguyen, and W. Q. Yan, “Fruit ripeness identification using YOLOv8 model,” *Multimed. Tools Appl.*, vol. 83, no. 9, pp. 28039–28056, 2024 .
- Bahl, L. R., Jelinek, F., & Mercer, R. L. (1983). A maximum likelihood approach to continuous speech recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-5(2), 179–190.
- Bao, G., Jia, M., Xun, Y., Cai, S., & Yang, Q. (2019). Cracked egg recognition based on machine vision. Zhejiang University of Technology.
- Basha, S. H. S., Dubey, S. R., Pulabaigari, V., & Mukherjee, S. (2019). Impact of fully connected layers on performance of convolutional neural networks for image classification. *Neurocomputing*, 378, 112–119.
- Bawa, V. S., & Kumar, V. (2019). Linearized sigmoidal activation: A novel activation function with tractable non-linear characteristics to boost representation capability. *Expert Systems with Applications*, 120, 346–356.
- B-CNN: Branch Convolutional Neural Network for Hierarchical Classification Zhu, X., & Bain, M. (2017). B-CNN: Branch convolutional neural network for hierarchical classification. *arXiv preprint arXiv:1709.09890*.
- Beauxis-Aussalet, E., & Hardman, L. (2014). Simplifying the visualization of confusion matrix. *Proceedings of the 2014 International Conference on Information Visualization Theory and Applications (IVAPP 2014)*, 197–204.

- Biau, G., & Scornet, E. (2016). A random forest guided tour. *TEST*, 25(2), 197–227.
- Bolikulov, F., Abdusalomov, A., Nasimov, R., Akhmedov, F., & Cho, Y.-I. (2024). Early poplar (Populus) leaf-based disease detection through computer vision, YOLOv8, and contrast stretching technique.
- Brasil, Y. L., Silva, D. F., & Oliveira, M. S. (2021). Fast online estimation of quail eggs freshness using portable NIR spectrometer and machine learning. *Food Control*, 129, 108223.
- Breiman, L. (2001). Random forests. *Machine Learning*, 45(1), 5–32.
- Caner, C., & Yüceer, M. (2014). Antimicrobial lysozyme–chitosan coatings affect functional properties and shelf life of chicken eggs during storage. *Journal of the Science of Food and Agriculture*, 94(1), 153–162.
- Chaya. (2020, June 8). *Random Forest Regression*. Level Up Coding. Retrieved from <https://levelup.gitconnected.com/random-forest-regression-209c0f354c84> (Erişim Tarihi: 29 Temmuz 2025)
- Datta, A. K., Botta, B., & Gattam, S. S. R. (2019). Damage detection on chicken eggshells using Faster R-CNN. *ASABE Annual International Meeting*, 1–9.
- Deng, X., Liu, Q., Deng, Y., & Mahadevan, S. (2016). An improved method to construct basic probability assignment based on the confusion matrix for classification problem. *Information Sciences*, 340–341, 250–261.
- Dooley J, Leeser M, Vaicaitis A, Jiang Y. Efficient Neural Networks on the Edge with FPGAs by Optimizing an Adaptive Activation Function. 2024 Mar 13; 24(6):1829; Sensors
- Dubey, S. R., Singh, S. K., & Chaudhuri, B. B. (2021). Activation functions in deep learning: A comprehensive survey and benchmark. *Neural Computing and Applications*, 33(13), 8969–9000.

- G. G. Casas, Z. H. Ismail, M. M. C. Limeira, A. A. L. da Silva, and H. G. Leite, “Automatic Detection and Counting of Stacked Eucalypt Timber Using the YOLOv8 Model,” *Forests*, vol. 14, no. 12, 2023..
- Gauravduttakiit. (n.d.). *Class dataset: Automated egg classification* [Dataset]. Kaggle. <https://www.kaggle.com/code/gauravduttakiit/class-dataset-automated-egg-classification/input>
- GeeksforGeeks. (2025, February 1). *K-Nearest Neighbor (KNN) Algorithm*.
- Geem ZW, Bokde ND, Hashmi MF, and Murthy CB. A thorough overview of research on object detection in photos and movies utilizing several deep learning methods and embedded systems. *Applied sciences*. 10(9):3280; May 8, 2020.
- Genuer, R., Poggi, J.-M., & Tuleau-Malot, C. (2017). The random forest algorithm for statistical learning. *arXiv preprint arXiv:1708.05070*.
- Giannoulidis, A., Gounaris, A., Naskos, A., Nikolaidis, N., & Caljouw, D. (2024). Engineering and evaluating an unsupervised predictive maintenance solution: A cold-forming press case-study. *Journal of Intelligent Manufacturing*, 36, 2121–2139.
- H. Yi, B. Liu, B. Zhao, and E. Liu, “Small Object Detection Algorithm Based on Improved YOLOv8 for Remote Sensing,” *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 17. pp. 1734–1747, 2024.
- Halder, R. K., Uddin, M. N., Uddin, M. A., Aryal, S., & Khraisat, A. (2024). Enhancing K-nearest neighbor algorithm: A comprehensive review and performance analysis of modifications. *Journal of Big Data, 11*, Article 113.
- Han, S., Mao, H., & Dally, W. J. (2016). Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding. *International Conference on Learning Representations (ICLR)*.

- Han, Z., Wei, B., Zheng, Y., Yin, Y., Li, K., & Li, S. (2017). Breast cancer multi-classification from histopathological images with structured deep learning model. *Scientific Reports*, 7, 4172.
- Handalage, U. H. D. I. (2021). Real-time object detection using YOLO: A review. *International Journal of Research in Engineering and Technology*, 10(5), 1–7.
- Hassan MF, Quach LD, Quoc NP, Thi NH, Tran DC. utilizing surf to enhance the resnet-50 model for the algorithm that detects diseases in poultry. The International Conference on Computational Intelligence (ICCI) will take place on October 8, 2020 (pp. 317–321). IEEE
- Hinterstoisser, S., Lepetit, V., Wohlhart, P., & Konolige, K. (2018). On pre-trained image features and synthetic images for deep learning. *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*.
- Huang, J. J., & Wanda, P. (2020). RunPool: A dynamic pooling layer for convolution neural network. *International Journal of Computational Intelligence Systems*, 13(1), 66–76.
- Huang, Y., Luo, Y., Cao, Y., Lin, X., Wei, H., Wu, M., Yang, X., & Zhao, Z. (2023). Damage detection of unwashed eggs through video and deep learning. *Foods*, 12(11), 2179.
- Imandoust, S. B., & Bolandraftar, M. (2013). Application of K-nearest neighbor (KNN) approach for predicting economic events: Theoretical background. *International Journal of Engineering Research and Applications*, 3(5), 605–610.
- İnik, Ö., & Turan, B. (2018). Classification of different age groups of people by using deep learning. *Journal of New Results in Science*, 7(3), 9–16.
- Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*.
- Jiang, T., Zhou, J., Xie, B., Liu, L., Ji, C., Liu, Y., Liu, B., & Zhang, B. (2024). Improved YOLOv8 model for lightweight pigeon egg detection. *Animals*, 14(8), 1226.

- Kaewtapee, C., & Supratak, A. (2021). Yolk color measurement using image processing and deep learning. In *IOP Conference Series: Earth and Environmental Science* (Vol. 686, No. 1, p. 012054). IOP Publishing.
- Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., Satheesh, S., Ma, S., Huang, Z., Deng, J., Su, H., Krause, J., & Russakovsky, O. (2015). *The large-scale visual recognition task of ImageNet*. *International Journal of Computer Vision*, 115*(3), 211–252.
- Kaya, M., Ulutürk, S., Çetin Kaya, Y., Altıntaş, O., & Turan, B. (2023). Optimization of several deep CNN models for waste classification. *Sakarya University Journal of Computer and Information Sciences*, 6(2), 91–99
- Khan, M. A., & Aslam, M. (2022). Understanding of convolutional neural network (CNN): A review. *International Journal of Robotics and Control Systems*, 2(4), 317–324.
- Khan, M. Y., Qayoom, A., Nizami, M. S., Raazi, K.-U.-R., & Syed, M. (2021). Automated prediction of good dictionary examples (GDEX): A comprehensive experiment with distant supervision, machine learning, and word embedding-based deep learning techniques. *Complexity*, 2021, 1–17.
- Kramer, O. (2013). K-nearest neighbors. In *Dimensionality reduction with unsupervised nearest neighbors* (pp. 13–23). Springer.
- Krichen, M. (2020). Adaptation of convolution and batch normalization layer for CNN implementation on FPGA. *ResearchGate*.
- Kumar, A., & Singh, R. (2023). ErfReLU: Adaptive activation function for deep neural network. *Pattern Analysis and Applications*.
- Kumar, R., & Singh, R. (2019). An overview of neural network. *American Journal of Neural Networks and Applications*, 5(1), 8–15.
- Li, G., Xu, Y., Zhao, Y., Du, Q., & Huang, Y. (2020). Evaluating convolutional neural networks for cage-free floor egg detection. *Sensors*, 20(2), 332.

- Liu, C., Wang, Q., Ma, M., Zhu, Z., Lin, W., Liu, S., & Fan, W. (2023). Single-view measurement method for egg size based on small-batch images. *Foods*, 12(5), 936 .
- Loey, M., Khalifa, N. E., Manogaran, G., & Taha, M. H. (2021). Defending against COVID-19: A new deep learning model for medical face mask identification using ResNet-50 and YOLO-v2. **Sustainable Cities and Society*, 65*, 102600.
- M. Sohan, T. Sai Ram, and C. V. Rami Reddy, “A Review on YOLOv8 and Its Advancements.” pp. 529–545, 2024.
- Ma, J., Ding, Y., & Cheng, J. C. P. (2019). Analyzing the leading causes of traffic fatalities using XGBoost and grid-based analysis: A city management perspective. *IEEE Access*, 7, 148059–148072.
- Mahmoud, O. (2015). Detection of poultry egg freshness by dielectric spectroscopy and machine learning techniques. *LWT - Food Science and Technology*, 62(1), 287–293.
- Markoulidakis, I., & Markoulidakis, G. (2024). Probabilistic confusion matrix: A novel method for machine learning algorithm generalized performance analysis. *Technologies*, 12(7), 113.
- Marom, N. D., Rokach, L., & Shmilovici, A. (2010). Using the confusion matrix for improving ensemble classifiers. *Proceedings of the 2010 IEEE 26th Convention of Electrical and Electronics Engineers in Israel (IEEEI)*, 000555–000559. IEEE.
- Milovanovic, B., Tomovic, V., Djekic, I., Solowiej, B. G., Lorenzo, J. M., Barba, F. J., & Tomasevic, I. (2021). Color assessment of the eggs using computer vision system and Minolta colorimeter. *Journal of Food Measurement and Characterization*, 15(6), 5097–5112.
- Mou, L., & Zhang, Y. (2022). A comparison of pooling methods for convolutional neural networks. *Applied Sciences*, 12(17), 8643.

- Nakahara, H., Fujii, T., & Sato, S. (2017). A fully connected layer elimination for a binarized convolutional neural network on an FPGA. In *Proceedings of the 27th International Conference on Field Programmable Logic and Applications (FPL)* (pp. 1–4). IEEE.
- Nakamura, R., & Katayama, T. (2000). Eggshell crack detection based on acoustic resonance frequency analysis. *Journal of Agricultural Engineering Research*, 76(2), 157–163.
- Narayanan, R. S., Abisharan, B., & Konetigari, L. R. (2023). *Smart Vision: Real-time object detection*. SSRN.
- Nasiri, A., Omid, M., & Taheri-Garavand, A. (2020). An automatic sorting system for unwashed eggs using deep learning. *Journal of Food Engineering*, 283, 110036.
- P. Pan *et al.*, “Lightweight cotton diseases real-time detection model for resource-constrained devices in natural environments,” *Front. Plant Sci.*, vol. 15, no. June, pp. 1–15, 2024.
- Papadopoulos, G., & Soudris, D. (2020). Hardware implementation of a softmax-like function for deep learning. *Technologies*, 8(3), 46.
- Peng, H., Wang, C., Xu, X., Wen, W., Ding, C., Ran, R., Luo, Y., Zhao, J., Huang, S., Thorat, K., Geng, T., Wang, C., & Zhao, J. (2023). LinGCN: A graph convolutional network with structural linearization for homomorphically encrypted inference. *arXiv preprint arXiv:2309.14331*.
- Roboflow. (n.d.). *Quail eggs quality detector*. Roboflow Universe.
https://universe.roboflow.com/quailleggsqualitydetector/quail_eggs_quality_detector
- S. Shanthi. (2012). Significance of activation functions in back propagation neural network: A study on binary categorical data. *Artificial Intelligence and Machine Learning*, 4(7).
- Sánchez, J. M. P., Moreno, L. A. O., Rodríguez, J. L. R., & Corro, I. D. M. (2023). Poultry egg classification system using deep learning. *Scientific Reports*, 13, 12345.

- Sattar, K. A., Ishak, I., Affendey, L. S., & Mohd Rum, S. N. (2024). Road crash injury severity prediction using a graph neural network framework. *IEEE Access*, 12, 37540–37556.
- Sattar, K., Maqsood, U., Hussain, Q., Majeed, S., Kaleem, S., Babar, M., & Qureshi, B. (2024). Soil texture analysis using controlled image processing. *Smart Agricultural Technology*, 9, 100588. [_](#)
- Schmied, T., Hofmarcher, M., Paischer, F., Pascanu, R., & Hochreiter, S. (2023). Learning to modulate pre-trained models in RL. *Advances in Neural Information Processing Systems (NeurIPS)*.
- Sehirli, E., & Arslan, K. (2022). An application for the classification of egg quality and Haugh unit based on characteristic egg features using machine learning models. *Expert Systems with Applications*, 205, 117692.
- Sharma, A., & Kaur, P. (2021). A pre-trained vs fine-tuning methodology in transfer learning. *International Journal of Computer Applications*, 183(22), 1–5.
- Sharma, S., Sharma, S., & Athaiya, A. (2020). Activation functions in neural networks. *International Journal of Engineering Applied Sciences and Technology*, 4(12), 310–316
- Sharma, S., Sharma, S., & Athaiya, A. (2020). Activation functions in neural networks. **International Journal of Engineering Applied Sciences and Technology (IJEAST)**, 4(12), 310–316.
- Shin, M., Kim, M., & Kwon, D.-S. (2016). *Baseline CNN structure analysis for facial expression recognition*. arXiv preprint arXiv:1611.04251.
- Singh, P., & Meitei, M. T. (2020). Short PCG classification based on deep learning. In *Deep Learning Techniques for Biomedical and Health Informatics* (pp. 103–117). Elsevier.
- Sohan, M., Sai Ram, T., & Venkata Rami Reddy, C. (2024). A review on YOLOv8 and its advancements. In *Proceedings of International Conference on Recent Innovations in Computing (ICRIC 2023)* (pp. 529–545). Springer.

- Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, 45(4), 427–437.
- Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition He, K., Zhang, X., Ren, S., & Sun, J. (2014). Spatial pyramid pooling in deep convolutional networks for visual recognition. *Lecture Notes in Computer Science*, 8691, 346–361.
- Szandała, T. (2021). Review and comparison of commonly used activation functions for deep neural networks. In A. Bhoi, P. Mallick, C. M. Liu, & V. Balas (Eds.), *Bio-inspired neurocomputing* (pp. 203–224). Springer.
- Tan, Y., Long, G., Ma, J., Liu, L., Zhou, T., & Jiang, J. (2022). Federated learning from pre-trained models: A contrastive learning approach. *Advances in Neural Information Processing Systems (NeurIPS)*.
- Taş, E., & Gökçe, B. (2017). Yumurtaların çevrimiçi bir destek vektör makinesi kullanılarak sınıflandırılması. *Afyon Kocatepe Üniversitesi Fen ve Mühendislik Bilimleri Dergisi*, 17(3), 914–921.
- Terven, J., Córdova-Esparza, D.-M., & Romero-González, J.-A. (2023). A comprehensive review of YOLO architectures in computer vision: From YOLOv1 to YOLOv8 and YOLO-NAS. *Machine Learning and Knowledge Extraction*, 5(4), 1680–1716.
- Ting, K. M. (2010). Confusion matrix. In C. Sammut & G. I. Webb (Eds.), *Encyclopedia of machine learning and data mining* (pp. 209–209). Springer.
- V7 Labs. (2021). Activation functions in neural networks: 12 types & use cases. *V7 Labs*.
- Wang, Q., Zhang, Y., & Li, H. (2022). A review of YOLO algorithm developments. *Procedia Computer Science*, 199, 1066–1073.

- Wang, Y., Li, X., & Zhang, H. (2024). The prediction of pervious concrete compressive strength based on a convolutional neural network. *Buildings*, 14(4), 907.
- Wang, Y., Pan, Z., Zheng, J., Qian, L., & Li, M. (2019). A hybrid ensemble method for pulsar candidate classification. *Astrophysics and Space Science*, 364(8), 139.
- Warner TA, Maxwell AE, Farhadpour S. Selecting and Interpreting Multiclass Loss and Accuracy Assessment Metrics for Classifications with Class Imbalance: Guidance and Best Practices. *Remote Sensing*. 2024 Jan 30;16(3):533.
- Wu, L., Wang, Q., Dengfei, J., Wang, S., Zhu, Z., & Xiong, L. (2018). Detection of crack eggs by image processing and soft-margin support vector machine. *Journal of Computational Methods in Sciences and Engineering*, 18(1), 21–31.
- Xu, J., Sun, X., Zhang, Z., Zhao, G., & Lin, J. (2019). Understanding and improving layer normalization. *arXiv preprint arXiv:1911.07013*.
- Yang, X., Bist, R. B., Subedi, S., & Chai, L. (2023). A computer vision-based automatic system for egg grading and defect detection. *Animals*, 13(14), 2354.
- You, K., Liu, Y., Wang, J., & Long, M. (2021). LogME: Practical assessment of pre-trained models for transfer learning. *Proceedings of the 38th International Conference on Machine Learning (ICML)*.
- Yu, W., & Nishio, M. (2022). Multilevel structural components detection and segmentation toward computer vision-based bridge inspection. *Sensors*, 22(9), 3502.
- Zeng, G. (2020). On the confusion matrix in credit scoring and its analytical properties. *Communications in Statistics - Theory and Methods*, 49(9), 2080–2093.
- Zhang, Y., Li, Y., Liu, F., & Zhu, A. (2020). The influence of the activation function in a convolution neural network model of facial expression recognition. *Applied Sciences*, 10(5), 1897.

Zhang, Z. (2016). Introduction to machine learning: K-nearest neighbors. *Annals of Translational Medicine, 4*(11), 218.

Zhang, Z. (2016). Introduction to machine learning: K-nearest neighbors. *Annals of Translational Medicine*, 4(11), 218.



ÖZGEÇMİŞ

