

T.C.  
BİLECİK ŐEYH EDEBALI ÜNİVERSİTESİ  
LİSANSÜSTÜ EĐİTİM ENSTİTÜSÜ  
BİLGİSAYAR MÜHENDİSLİĐİ ANABİLİM DALI

**DERİN ÖĐRENME YÖNTEMLERİ KULLANILARAK BEYAZ LAHANA YAPRAK  
HASARLARININ TESPİTİ**

YÜKSEK LİSANS TEZİ

SERKAN SÖKMEN

TEZ DANIŐMANI

DR. ÖĐR. ÜYESİ VEDAT MARTTİN

BİLECİK, 2025

10602372

T.C.  
BİLECİK ŐEYH EDEBALI ÜNİVERSİTESİ  
LİSANSÜSTÜ EĐİTİM ENSTİTÜSÜ  
BİLGİSAYAR MÜHENDİSLİĐİ ANABİLİM DALI

**DERİN ÖĐRENME YÖNTEMLERİ KULLANILARAK BEYAZ LAHANA YAPRAK  
HASARLARININ TESPİTİ**

YÜKSEK LİSANS TEZİ

SERKAN SÖKMEN

TEZ DANIŐMANI

DR. ÖĐR. ÜYESİ VEDAT MARTİN

BİLECİK, 2025

10602372

## BEYAN

Derin Öğrenme Yöntemleri Kullanılarak Beyaz Lahana Yaprak Hasarlarının Tespiti adlı yüksek lisans tezinin hazırlık ve yazımı sırasında bilimsel araştırma ve etik kurallarına uyduğumu, başkalarının eserlerinden yararlandığım bölümlerde bilimsel kurallara uygun olarak atıfta bulunduğumu, kullandığım verilerde herhangi bir tahrifat yapmadığımı, tezin herhangi bir kısmının Bilecik Şeyh Edebali Üniversitesi veya başka bir üniversitede başka bir tez çalışması olarak sunulmadığını, aksinin tespit edileceği muhtemel durumlarda doğabilecek her türlü hukuki sorumluluğu kabul ettiğimi ve vermiş olduğum bilgilerin doğru olduğunu beyan ederim.

|   |                          |                            |
|---|--------------------------|----------------------------|
| Bu çalışmanın,<br>Bilimsel Araştırma Projeleri (BAP), TÜBİTAK veya benzeri kuruluşlarca desteklenmesi durumunda; projenin ve destekleyen kurumun adı proje numarası ile birlikte, ETİK KURUL onayı alınması durumunda ise ETİK KURUL tarih karar ve sayı bilgilerinin beyan edilmesi gerekmektedir. |                          |                            |
| <b>DESTEK ALINMIŞTIR</b>  | <input type="checkbox"/> | <b>DESTEK ALINMAMIŞTIR</b> |
|   |                          | <b>X</b>                   |
| <b>Destek alındı ise;</b>   |                          |                            |
| <b>Destekleyen kurum;</b>   |                          |                            |
| <b>Desteğin Türü</b>  | <b>Proje Numarası</b>    |                            |
| <b>1- BAP (Bilimsel Araştırma Projesi)</b>  |                          |                            |
| <b>2- TÜBİTAK</b>   |                          |                            |
| <b>Diğer;.....</b>  |                          |                            |
| <b>ETİK KURUL onayı var ise;</b>  |                          |                            |
| <b>ETİK KURUL karar tarih/sayı:</b>   | ...../....               |                            |

Serkan Sökmen

.../.../2025

İmza

## ÖN SÖZ

Yüksek lisans tez sürecimdeki tüm aşamalarda bana yol gösteren ve rehberlik eden, sabır ve gayretle çalışmama destek olan akademik danışmanım Sayın Dr. Öğr. Üyesi Vedat MARTTİN' e değerli katkılarından dolayı teşekkür ederim.

Yüksek lisans tez çalışmamda yardımını esirgemeyen, sorduğum sorulara sabırla cevap veren Sayın Öğr. Gör. Musa TURKAN'a, tarım alanındaki danışmalığı ile desteğini esirgemeyen Yüksek Ziraat Mühendisi Sayın Öğr. Gör. İrem YAZICI' ya ve bu akademik süreçte emeği geçen herkese teşekkür ederim.

Tüm hayatım boyunca benden maddi ve manevi hiçbir konuda emeğini esirgemeyen ve her zaman yanımda olan, başta babam Selahattin SÖKMEN, annem Saliha SÖKMEN ve eşim Nalan SÖKMEN olmak üzere tüm aile bireylerime teşekkür ederim.

**Serkan Sökmen**

**2025**

## ÖZET

### DERİN ÖĞRENME YÖNTEMLERİ KULLANILARAK BEYAZ LAHANA YAPRAK HASARLARININ TESPİTİ

Dünya nüfusunun giderek artması, iklim değişikliği gibi nedenlerle gıdaya olan ihtiyacın artmasıyla çiftçilerin daha verimli ve kaliteli ürünler yetiştirmesi gerekmektedir. Yetiştirilen ürünlerde bitkilerdeki hastalık ve zararlıların tespiti büyük önem taşımaktadır. Günümüzde genellikle hastalık ve zararlılar çiftçiler tarafından tespiti gözle kontrol edilerek yapılmaktadır. Bu tespit süreci büyük alanlarda üretim yapan çiftçiler için ciddi zaman almakla birlikte zararlının geç tespit edilmesine de sebep olabilmektedir.

Günümüzde pek çok alanda yapay zekanın kullanımı yaygınlaşmaktadır. Tarım sektöründe yapay zekanın kullanımı, son yıllarda önemli ölçüde artış göstermiştir. Hastalıklı yaprakların tespiti, bitki büyüme dönemlerinde bitkinin sağlığı için kritik bir öneme sahiptir. Yapay zekanın kullanımında bitkiye ait görüntü işleme teknikleriyle bitki sağlığı takip edilebilmektedir.

Bu çalışmada tarımsal üretimde hastalık tespiti sorununa yönelik olarak görüntü işleme ve derin öğrenme tekniklerini kullanarak beyaz lahana bitkisine ait elde edilen görüntüler işlenerek You Only Look Once (YOLO) tekniği yardımıyla, hastalık belirtilerinin doğru bir şekilde sınıflandırılması ve lokalize edilmesi hedeflenmiştir. Çalışmada kullanılan veri seti için Bilecik/Pazaryeri bölgesine ait yetiştirilen 3960 adet beyaz lahana fotoğrafları çekilmiş ve veri etiketleme yapılmıştır. Toplam etiket sayısı 75000 civarındadır. Etiketlenen veriler Colab platformu yardımıyla Python dili kullanılarak YOLO tekniğiyle eğitip sınıflandırılmıştır. Ağ modeli 500 devir (epoch) eğitilmiştir. Model etiket alanları kullanılarak hasarlı bölge oranı yüzde olarak hesaplanmıştır. Eğitilen model hasarlı bölgeleri %95 oranında doğru şekilde tespit ettiği görülmüştür. Önerilen yöntemin, tarımsal verimliliği artırma, hastalıkların yayılmasını önleme ve genel olarak tarım teknolojilerinin ilerlemesine katkı sağlama potansiyeline sahip olduğu düşünülmektedir.

**Anahtar Kelimeler:** Nesne Tespiti, Derin Öğrenme, YOLO, Bitki Hasar Tespiti, Beyaz Lahana.

## ABSTRACT

### DETECTION OF WHITE CABBAGE LEAF DAMAGES USING DEEP LEARNING METHODS

As the world population increases and the need for food increases due to climate change, farmers need to grow more efficient and higher quality products. Detection of diseases and pests in the products grown is of great importance. Today, farmers usually detect diseases and pests by visual inspection. This detection process takes a lot of time for farmers who produce in large areas and can also cause late detection of the pest.

Nowadays, the use of artificial intelligence is becoming widespread in many areas. The use of artificial intelligence in the agricultural sector has increased significantly in recent years. Detection of diseased leaves is of critical importance for the health of the plant during plant growth periods. In the use of artificial intelligence, plant health can be monitored with image processing techniques of the plant.

In this study, it is aimed to accurately classify and localize disease symptoms by processing images obtained from white cabbage plants using image processing and deep learning techniques for the problem of disease detection in agricultural production with the help of You Only Look Once (YOLO) technique. For the dataset used in the study, 3960 white cabbages grown in Bilecik/Pazaryeri region were photographed and data labeling was done. The total number of labels is around 75000. The labeled data was trained and classified with the YOLO technique using the Python language with the help of the Colab platform. The network model was trained for 500 epochs. The damaged area ratio was calculated as a percentage using the model label fields. It was observed that the trained model correctly detected the damaged areas by 95%. It is thought that the proposed method has the potential to increase agricultural productivity, prevent the spread of diseases and contribute to the advancement of agricultural technologies in general.

**Keywords:** Object Detection, Deep Learning, YOLO, Plant Damage Detection, White Cabbage.

# İÇİNDEKİLER

|   | Sayfa |
|---|-------|
| ÖN SÖZ.....   | i     |
| ÖZET.....   | ii    |
| ABSTRACT.....   | iii   |
| İÇİNDEKİLER.....                                      | iv    |
| ŞEKİLLER LİSTESİ.....                                 | vi    |
| TABLolar LİSTESİ.....                                 | viii  |
| SİMGELER VE KISALTMALAR LİSTESİ.....                  | ix    |
| 1. GİRİŞ.....   | 1     |
| 1.1. Literatür Taraması.....                          | 1     |
| 1.2. Tezin Amacı ve Motivasyonu .....                 | 6     |
| 1.3. Tezin organizasyonu.....                         | 7     |
| 2. MATERYAL VE YÖNTEM.....                            | 8     |
| 2.1. Yapay Zeka .....                                 | 8     |
| 2.1.1. Yapay Zekanın Zorlukları ve Etik Sorunlar..... | 10    |
| 2.2. Makine Öğrenmesi .....                           | 10    |
| 2.3. Derin Öğrenme .....                              | 14    |
| 2.4. Yapay Sinir Ağları.....                          | 16    |
| 2.4.1. Nöronların Çalışma Prensipleri.....            | 18    |
| 2.5. Evrişimli Sinir Ağları (CNN).....                | 19    |
| 2.5.1. CNN'lerin Avantajları.....                     | 20    |
| 2.5.2. CNN'lerin Uygulama Alanları.....               | 21    |
| 2.6. YOLO v11 .....                                   | 21    |
| 3. VERİ SETİ HAZIRLANMASI VE LAHANA ZARARLISI.....    | 25    |
| 3.1. Veri Etiketlemesi .....                          | 26    |

|  |           |
|--|-----------|
| <b>3.2. Görsel Etiketleme Araçları.....</b>                  | <b>26</b> |
| <b>3.3. Veri Seti.....</b>                                   | <b>29</b> |
| <b>3.4. Lahana Zararlıları.....</b>                          | <b>30</b> |
| <b>4. DENEYSEL ÇALIŞMALAR VE BULGULAR.....</b>               | <b>33</b> |
| <b>4.1. YOLO Ağı ile Lahanelerin Sınıflandırılması .....</b> | <b>33</b> |
| <b>4.2. Hasarlı Bölgenin Hesaplanması .....</b>              | <b>39</b> |
| <b>5. SONUÇLAR VE ÖNERİLER.....</b>                          | <b>48</b> |
| <b>KAYNAKÇA .....</b>  | <b>50</b> |

## ŞEKİLLER LİSTESİ

|  | Sayfa |
|--|-------|
| Şekil 1.1. Lahana yaprak zararlısı uygulamalarının bibliyometrik ağ görseli .....                                | 5     |
| Şekil 1.2. Lahana üzerine çalışmaların ilgili alanlarına ait ağ görseli .....                                    | 5     |
| Şekil 1.3. Lahana ve derin öğrenme üzerine çalışmaların ilgili alanlarına ait ağ görseli .....                   | 6     |
| Şekil 2.1. Yapay zeka'nın temel bileşenleri.....   | 8     |
| Şekil 2.2. Makine öğrenmesine ait temel bileşenler ve kullanım alanları .....                                    | 11    |
| Şekil 2.3. Denetimli öğrenme .....   | 11    |
| Şekil 2.4. Denetimsiz öğrenme .....  | 12    |
| Şekil 2.5. Pekiştirmeli öğrenme .....  | 13    |
| Şekil 2.6. Makine öğrenmesi ve Derin öğrenme yöntemlerinin farkı.....  | 14    |
| Şekil 2.7. Yapay zeka, Makine öğrenimi ve Derin öğrenmenin gösterimi .....                                       | 15    |
| Şekil 2.8. Sinir hücresi ve yapay sinir ağı.....   | 17    |
| Şekil 2.9. Yapay sinir ağlarının çalışma prensibi.....   | 18    |
| Şekil 2.10. YOLO Sürüm tarihçesi .....   | 22    |
| Şekil 2.11. YOLO v11' in Mimari yapısı .....   | 23    |
| Şekil 3.1. Bilecik/Pazaryeri lahana görsellerinin toplandığı alan .....  | 25    |
| Şekil 3.2. LabelImg programı kullanılarak etiketlenmiş bir fotoğraf .....  | 27    |
| Şekil 3.3. Veri setindeki görüntülerden örnekler .....   | 29    |
| Şekil 3.4. Etiketlenen veri sayıları .....   | 30    |
| Şekil 3.5. Lahana yaprak güvesi zararı bulunan lahana bahçesi (A ve B) ve larvanın yapraktaki görünümü (C) ..... | 31    |
| Şekil 3.6. Lahana yaprak güvesinin Dünya üzerindeki yayılış alanları .....                                       | 32    |
| Şekil 4.1. YOLO algoritması sürüm farklılıkları .....  | 33    |
| Şekil 4.2. Ağ eğitimi süresi boyunca doğrulama ve başarımleri metrikleri değişimi .....                          | 34    |
| Şekil 4.3. Precision ve güven eşiği değişimi.....  | 35    |

|   |           |
|---|-----------|
| <b>Şekil 4.4.</b> Duyarlılık ve güven eşiği değişimi .....  | <b>36</b> |
| <b>Şekil 4.5.</b> F1 skoru ve güven eşiği değişimi .....  | <b>37</b> |
| <b>Şekil 4.6.</b> Karışıklık matrisi .....  | <b>38</b> |
| <b>Şekil 4.7.</b> Doğrulama veri seti üzerinde yapılan sınıflandırmalar.....                        | <b>38</b> |
| <b>Şekil 4.8.</b> Etiketlenmiş lahana görseli.....  | <b>39</b> |
| <b>Şekil 4.9.</b> Test verisi üzerinde sınıflandırmalar .....                                       | <b>40</b> |
| <b>Şekil 4.10.</b> %26,9 Hasar oranlı test resmi etiketleri.....                                    | <b>42</b> |
| <b>Şekil 4.11.</b> %11,2 Hasar oranlı test resmi etiketleri.....                                    | <b>43</b> |
| <b>Şekil 4.12.</b> %26,9 Hasar oranlı bir görüntünün Conf değeri 0,01 ile hasırlı bölge gösterimi.. | <b>44</b> |
| <b>Şekil 4.13.</b> %11,2 Hasar oranlı bir görüntünün Conf değeri 0,05 ile hasırlı bölge gösterimi.. | <b>45</b> |

## TABLULAR LİSTESİ

|  | Sayfa     |
|--|-----------|
| <b>Tablo 2.1.</b> YOLO Sürümlerinin performans karşılaştırması .....   | <b>24</b> |
| <b>Tablo 3.1.</b> Etiketlenmiş bir fotoğrafın verileri.....            | <b>28</b> |
| <b>Tablo 4.1.</b> Ağın eğitiminde kullanılan model parametreleri ..... | <b>46</b> |
| <b>Tablo 4.2.</b> Eğitilen modelin başarımlar değerleri .....          | <b>47</b> |

## SİMGELER VE KISALTMALAR LİSTESİ

**AI:** Yapay Zeka (Artificial Intelligence)

**ANN:** Yapay Sinir Ağı (Artificial Neural Network)

**BiLSTM:** İki Yönlü Uzun Kısa Süreli Bellek (Bidirectional Long Short-term Memory)

**CNN:** Evrişimsel Sinir Ağları (Convolution Neural Network)

**DL:** Derin Öğrenme (Deep Learning)

**KNN:** K en yakın komşu Algoritması (K-nearest neighbour)

**LSTM:** Uzun Kısa Süreli Bellek (Long Short-term Memory)

**mAP:** Ortalama Hassasiyet (Mean Average Precision)

**ML:** Makine Öğrenmesi (Machine Learning)

**NLP:** Doğal Dil İşleme (Natural Language Processing)

**ReLU:** Düzeltilmiş Doğrusal Birim (Rectified Linear Unit)

**ResNet:** Artık Ağlar (Residual Network)

**RGB:** Kırmızı, Yeşil, Mavi (Red, Green, Blue)

**RNN:** Tekrar Eden Sinir Ağları (Recurrent Neural Network)

**SVM:** Destek Vektör Makineleri (Support Vector Machine)

**TÜİK:** Türkiye İstatistik Kurumu

**UNFPA:** Birleşmiş Milletler Nüfus Fonu (United Nations Fund for Population Activities)

**YOLO:** You Only Look Once

## 1. GİRİŞ

Birleşmiş Milletler Nüfus Fonu (UNFPA, 2022) tarafından açıklanan raporda 2050 yılında dünya nüfusunun 9,1 milyara ulaşması tahmin edilmektedir. Nüfus artışının gelişmekte olan ülkelerde devam edeceği ve dünya nüfusunun yaklaşık olarak %70' inin şehir merkezlerinde yaşayacağı tahmin edilmektedir. Dünya nüfusunun artan gıda talebini karşılamak için tarımsal üretimin daha verimli ve sürdürülebilir yöntemler kullanılarak gerçekleştirilmesi gerektiği vurgulanmaktadır. Türkiye İstatistik Kurumu (TÜİK, 2024) tarafından yapılan araştırmalara göre, ülkemizin 2023 yılında 85,3 milyon olan nüfusunun %93' ü il ve ilçe merkezlerinde, %7' si de belde ve köylerde yaşamaktadır. Dünya ve ülke nüfusunun artması, iklim değişikliği gibi olumsuz etkilerin olması, çiftçilerin daha verimli ve kaliteli ürünler yetiştirmesi gerekliliğini ortaya koymaktadır. Verim ve kaliteyi artırmak için bitkilerdeki hastalık ve zararlıların erken teşhisi büyük önem arz etmektedir. Günümüzde bu hastalık ve zararlılar çiftçiler tarafından gözle kontrol edilerek tespit edilmekte ve bu tespit işlemi özellikle büyük alanlarda üretim yapan çiftçiler için ciddi zaman almakta, ayrıca üretim alanlarının bir kısmında oluşabilecek hastalık ve zararlıların geç tespit edilmesine sebep olabilmektedir. Ayrıca doğru bir şekilde hastalık ve zararlı tespiti yapılmadan uygulanacak tarım ilaçları bitkide biriken ilaç kalıntıları arttırıp, tüketici için ciddi sağlık sorunları oluşturabilmektedir. Üretici açısından iş gücü, zaman kaybı, verim düşüklüğü ve tarım ilaçlarına verilen ilave paralar ile üretim maliyetleri artmakta, karlılıkları ise düşmektedir.

Tarım sektöründeki ürün yetiştirme aşamalarının bir çoğunda yapay zekanın kullanımı ve önemi gün geçtikçe artış göstermektedir. Bitkinin büyüme sürecinde, hastalıklı yaprakların ve zararlıların hızlı tespiti, bitkinin sağlığı ve ürün kalitesi için kritik bir öneme sahiptir.

### 1.1. Literatür Taraması

Literatürde yapılan bitki hastalık ya da yaprak hastalığı tespiti çalışmaları dikkat çekmektedir. Yapılan derleme çalışmasında Saleem ve arkadaşları, bitki hastalık tespitinde doğruluğun artırılmasını derin öğrenme modelleriyle açıklamıştır (Saleem vd., 2019).

Tez çalışmaları kapsamında yapılan araştırmalarda Duman, elma yaprağı hastalıklarının tespiti amacıyla Convolutional Neural Network (CNN) mimarileri arasında ResNet50V2 ve MobileNet kullanılarak başarı oranları karşılaştırılmıştır. Eğitim süreçlerinde Adam optimizasyon algoritması tercih edilmiştir. Genel değerlendirmeler, ResNet50V2 ve MobileNet mimarileri üzerinden gerçekleştirilmiş olup, çalışma süresi açısından yapılan analizler sonucunda en verimli modelin ResNet50V2 olduğu belirlenmiştir (Duman, 2022).

Hamad'ın tez çalışmasında, internetten elde edilen Plant Village veri seti görüntüleri ilave olarak domates görselleri ile veri artırma yöntemiyle genişletilerek seçilen örnek hastalık türüne göre sağlıklı yaprakların tespiti gerçekleştirilmiştir. Veri seti önceden eğitilmiş birkaç ImageNet modelinde transfer öğrenimi kullanılmış ve InceptionV3, eğitim setinde %98,41, doğrulama setinde %97,5 elde edilmiştir (Hamad, 2022).

Bozcu'nun tez çalışmasında, Plant Village veri seti ile Eskişehir Kozlu Köyü'ndeki kiraz tarlalarındaki görüntülerden oluşturulan Kozlu veri seti kullanılmıştır. PlantVillage veri setinde AlexNet, VGG-16, Inception-V3, MobileNet-V2 ve CNN modelleri, Kozlu veri setinde ise Inception-V3, MobileNet-V2, CNN, CNN + Uzun Kısa Süreli Bellek (LSTM) ve CNN + İki Yönlü Uzun Kısa Süreli Bellek (BiLSTM) modelleri kullanılarak yapraklardaki hastalıkların teşhisi gerçekleştirilmiş ve yöntemlerin performansları karşılaştırılmıştır (Bozcu, 2024).

Bülgen tez çalışmasında, mısır için Kaggle veri seti, buğday için ise Mendeley veri seti kullanılarak yapay zeka tabanlı sınıflandırma modelleri geliştirilmiştir. Mısır sınıflandırmasında ResNet Artık Ağlar (Residual Network) modelleri, buğday sınıflandırmasında ise "You Only Look Once (YOLO)" modelleri tercih edilmiştir. Çalışmada ayrıca ResNet-50/101/152 ve YOLO-v5s/v8 modellerinin performansları karşılaştırmalı olarak analiz edilmiş ve mısır sınıflandırmasında ResNet101 modelinin, buğday sınıflandırmasında ise YOLOv8 modelinin benzer veri setleriyle kıyaslandığında en yüksek başarı oranlarını sunduğu belirlenmiştir (Bülgen, 2024).

Sibiya ve arkadaşlarının çalışmasında, mısır hastalıklarının sınıflandırılması amacıyla evrimsel sinir ağları (CNN) tabanlı bir model geliştirilmiş ve bu model, mısır yapraklarında bulunan üç farklı hastalığı başarıyla tespit edebilmiştir. Bilgisayarlı görme tekniklerinden biri olan görüntülerden renk özelliği çıkarımı yöntemi, bitki yapraklarındaki hastalıkların tanınmasında yaygın olarak kullanılan tekniklerden biridir (Sibiya ve Sumbwanyambe, 2019).

Gülmez çalışmasında, gül yaprakları üzerinde yapay zeka modellerinin çeşitli türleri incelenmiştir. Çalışmada CNN mimarisinin üç temel katman tipi açıklanmıştır: evrimsel katmanlar, girdiden özelliklerin çıkarılmasını sağlayan desenleri içerir; havuzlama katmanları, verinin boyutunu küçülterek işlem sürecini basitleştirir; ve tam bağlantılı yapay sinir ağı katmanı, veriyi düzleştirerek sınıflandırma işlemini gerçekleştirir. RoseNet veri kümesinin kullanıldığı belirtilmiş olup, bu veri kümesinin üç farklı hastalık türünü içerdiği ifade edilmiştir.

Yapay zeka sistemlerinin eğitimi, bu hastalıklara ait örnek resimlerin kullanılmasıyla gerçekleştirilmiştir (Gülmez, 2023).

Tejaswini ve arkadaşları çalışmasında, Hindistan'da pirinç yaprakları üzerinde başlangıç olarak VGG-16 modeli kullanılmış ve %58 başarı oranı elde edilmiştir. Bu çalışmada, başarı oranını artırmak amacıyla VGG-16 modeline ek olarak MobileNet-V2, Inception-V3, LSTM ve BiLSTM yöntemleri de kullanılmıştır. Ayrıca, belirli veri artırma teknikleri ve renk uzayları uygulanarak modelin performansı iyileştirilmiştir (Tejaswini vd., 2022). Literatürde, bu tür iyileştirme yöntemlerine dair çeşitli örnekler de bulunmaktadır.

Tian ve arkadaşları çalışmasında YOLOv8n'i geliştiren hasat dönemi lahana başı tanıma algoritması önermiştir. Geliştirilen modelde lahana başının farklı ışık koşulları ve karmaşık arka planlar altında lahana başı tanımayı uyarlayabileceğini göstermektedir. Modelde, %91 hassasiyet, %87,2 geri çağırma ve %94,5'lik bir mAP@50 (Mean Average Precision – Nesne algılamadaki ortalama hassasiyet) elde edilmiştir (Tian vd., 2024).

Gao ve arkadaşları çalışmasında, hızlı R-CNN mimarisinden türetilen ağ başını ayıran yeni bir yaklaşımla YOLOv7 modelini iyileştirerek bu zorlukları ele almayı amaçlamıştır. Çalışmada üst üste binen Çin lahanalarının merkezlerini doğru bir şekilde belirlemek için bir BiFormer dikkat mekanizması tanıtılmıştır. Modelde %94,2 doğruluk ve %91,3 Çin lahanası tanımlama oranına ulaşmıştır. Çalışmada üst üste binen sebze tanımlamasının doğruluğunu iyileştirmekte ve hassas tarımı geliştirmek için bir çerçeve sağlamaktadır (Gao vd., 2024).

Yuan ve çalışma arkadaşları, Çin çiçekli lahana tomurcuklarını doğru bir şekilde tanımlama zorluğunu ele almak için, YOLOv5 algoritmasının özellik birleştirme yaklaşımında iyileştirmeler yapılmış ve bunun sonucunda dinamik olarak ayarlanabilir bir algılama başlığına sahip FPNDyH-YOLOv5 adında yeni bir algoritma önermiştir. Deneysel sonuçlar, geliştirilmiş modelin %93,9'luk ortalama bir hassasiyete (mAP@0,5) ulaştığını ve temel modele kıyasla %2,5'lik bir iyileştirme sağladığını göstermiştir. Farklı olgunluk seviyelerindeki tomurcuklar için ortalama hassasiyet (AP) sırasıyla %96,1, %86,9 ve %98,7 dir. Yöntem sayesinde doğal ortamlarda Çin çiçekli lahana tomurcuklarının nispeten doğru bir şekilde sınıflandırılması ve sayılması kolaylaşmıştır (Yuan vd., 2023).

Fu ve arkadaşları çalışmasında, derin öğrenme ve çevrimiçi tanımlama yöntemleri uygulanarak hedef odaklı püskürtme için kontrol teknolojisi, araştırma nesnesi olarak lahanalarla incelemiştir. Püskürtücülerin çalışması sırasında güçlü ışık koşullarında hareket bulanıklığı ve düşük ortalama hassasiyetin üstesinden gelmek için, karmaşık ortamlarda lahana

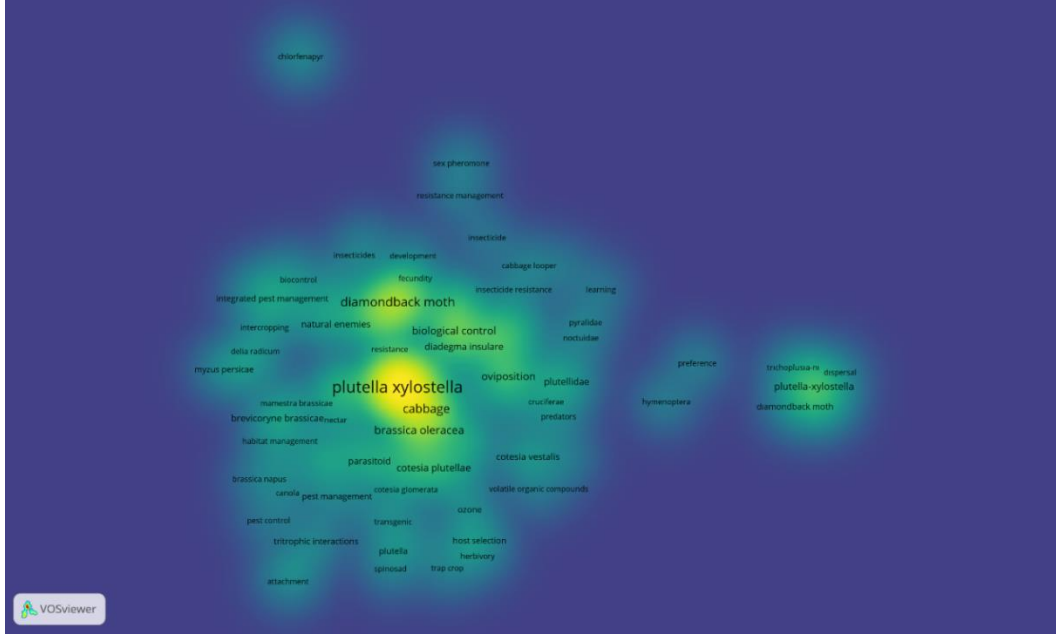
tarlaları için doğru çevrimiçi tanımlama elde etmek üzere bir transformatör modülü yerleştirilmiş yenilikçi bir YOLOV5 modeli kullanılmıştır. Kapalı alan test sonuçları, ortalama hassasiyetin %96,14 olduğu hareket bulanıklığı oluştuğunda, hedef için ortalama hassasiyet %90,31 olarak bulunmuştur. Ortalama hassasiyet %98,65'tir ve tasarruf oranı %54,04'e ulaşmaktadır. Çalışmada genel olarak tasarlanan hedef odaklı püskürtme sistemi beklenen deneysel sonuçları elde etmekte ve saha hedef püskürtme için teknik destek sağlayabilmektedir (Zhao vd., 2022).

Wu ve arkadaşları, Çin çiçekli lahanasının sayımı ve zamanında hasadı için Cabbage-YOLO modelini önermiştir. Modelin ortalama tespit doğruluğu %86,4'e ulaşmıştır. Orijinal model YOLOv8-n ile karşılaştırıldığında Cabbage-YOLO modeli daha yüksek doğruluk, daha küçük boyut ve hafif dağıtımda belirgin bir avantaja sahip olduğu gözlenmiştir (Wu vd., 2024).

Mamat ve arkadaşları, derleme çalışmasında tarım endüstrisinde görüntü açıklamasının çeşitli derin öğrenme yaklaşımlarını kullanarak uygulanmasını tartışmıştır. Görüntüleri eğitmek için kullanılan çeşitli ek açıklama türleri sunulmuştur. Son yayınlar, derin öğrenmenin mevcut ilerleme teknolojisiyle uygulanması temelinde incelenmiştir. Bitki tanıma, hastalık tespiti, sayma, sınıflandırma ve verim tahmini, tarımda birçok uygulamada kullanılan derin öğrenme mimarisini kapsamlı bir şekilde incelenmiştir (Mamat vd., 2022).

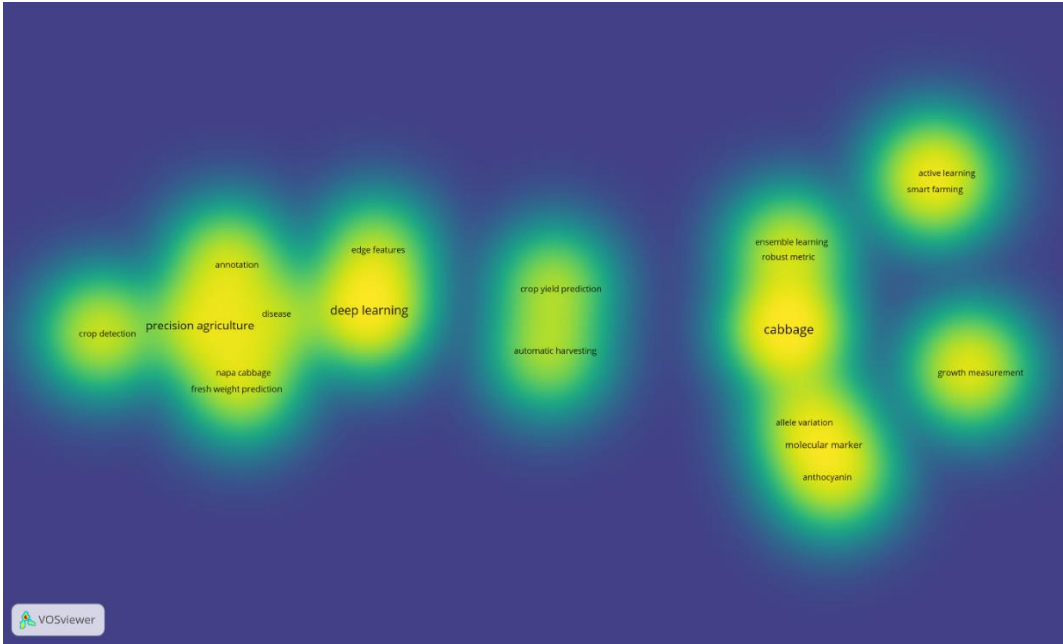
Literatür çalışmalarında daha çok çeşitli yaprak hastalıkları ve bunun görselleri üzerinde yapılan araştırmaların olduğu görülmüştür. Yaprak hasarı ya da yaprak hasarına sebep olan zararlı hakkında yapılan araştırmalara pek rastlanamamıştır.

Şekil 1.1.'de Web of Science(WoS) veritabanında anahtar kelime olarak “plutella xylostalla” ve ”cabbage” taratıldığında VOSViewer programıyla elde edilen bibliyometrik ağ görseli verilmektedir.



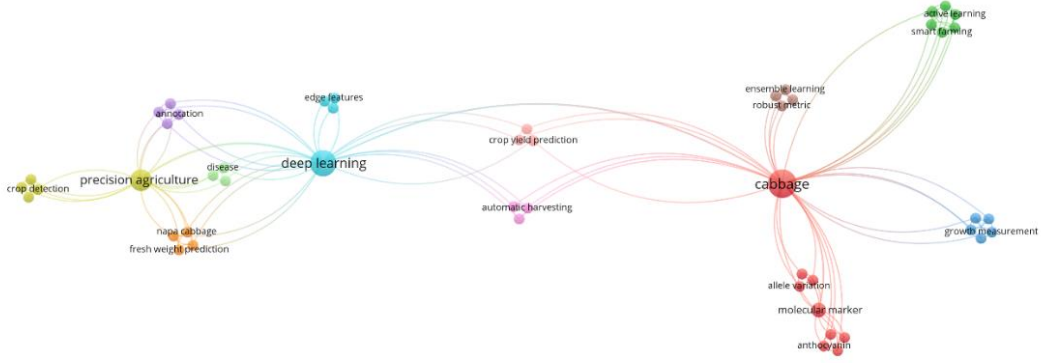
**Şekil 1.1.** Lahana yaprak zararlısı uygulamalarının bibliyometrik ağ görseli

Şekil 1.2.'de lahana ile ilgili yapılan çalışmaların daha çok ürün tespiti, tarımda taze ürün ağırlığı ve tahmini, derin öğrenme ile özellik çıkarımı, otomatik hasat, ürün-hasat tahmini, akıllı tarım ve ürün geliştirme ölçümleri üzerine olduğu görülmektedir.



**Şekil 1.2.** Lahana üzerine çalışmaların ilgili alanlarına ait ağ görseli

Şekil 1.3.'de Web of Science(WoS) veritabanında anahtar kelime olarak “deep learning” ve “cabbage” taratıldığında VOSViewer programıyla elde edilen ağ görseli verilmektedir.



**Şekil 1.3.** Lahana ve derin öğrenme üzerine çalışmaların ilgili alanlarına ait ağ görseli

## 1.2. Tezin Amacı ve Motivasyonu

Bu tezin amacı, görüntü işleme ve derin öğrenme teknolojileriyle geliştirilen YOLO tekniği kullanılarak hastalık belirtilerinin doğru, hızlı bir şekilde belirlenmesi ve lokalize edilmesi hedeflenmektedir. Önerilen yöntemin, tarımsal verimliliği artırma, hastalıkların yayılmasını önleme ve genel olarak tarım teknolojilerinin ilerlemesine katkı sağlama potansiyeline sahip olduğu düşünülmektedir. Çalışmada lahana veri seti oluşturulmuştur. Lahana yapraklarında hasar yüzdesi hesaplanmıştır.

Çalışmanın öne çıkan yanları şu şekilde sıralanabilir:

- Çalışmada, Bilecik ili Pazaryeri ilçesinde beyaz lahana ekimi yapılan bir alandan 2023 yılı Eylül ayında fotoğraf çekimi yapılarak lahana görselleri tek tek toplanmış ve orijinal bir veri seti oluşturulmuştur.
- YOLO eğitimine uygun hale getirmek için lahana görsellerine ait etiketlemeler Ziraat Yüksek Mühendisi eşliğinde yapılmıştır.
- Literatürde çeşitli bitkiler ve yaprak hastalıklarına yönelik tez çalışması ve araştırmalar yapılmıştır. Beyaz lahana özelinde bir teze rastlanamamıştır. Bu bakımından tez literatüre katkı sağlayacaktır.
- Literatürde yaprak hasarı ve yaprak hasarı zararlısı hakkında bir çalışmaya rastlanamamıştır. Tez çalışması bu alanda da literatüre katkı sağlayacaktır.

### **1.3. Tezin organizasyonu**

Birinci bölümde yapay zekanın tarımda kullanım çalışmaları incelenmiştir. Literatürde daha önce yapılmış benzer çalışmalar sunulmuştur. İkinci bölümünde yapay zeka kavramları incelenmiş, çalışmada kullanılan YOLO modeli tanıtılmıştır. Üçüncü bölümde tezde kullanılan veri setinin hazırlanması ile lahana zararlısı tanıtılmış ve bitki üzerindeki etkileri anlatılmıştır. Dördüncü bölümde YOLO modeli kullanılarak yapılan çalışmanın detayları sunulmuştur. Beşinci bölümde ise sonuçlar değerlendirilmiş ve yapılabilecek yeni çalışma önerileri sunulmuştur.

## 2. MATERYAL VE YÖNTEM

Bu kısımda yapay zeka, makine öğrenmesi, derin öğrenme, yapay sinir ağları, CNN ve YOLO hakkında bilgiler verilmektedir.

### 2.1. Yapay Zeka

Yapay Zeka (Artificial Intelligence - AI), makinelerin, özellikle bilgisayarların, insan zekasını taklit edebilen sistemler geliştirmeyi amaçlayan bir teknoloji alanıdır. Yapay zeka, insan zekasına özgü düşünme, öğrenme, problem çözme, plan yapma, dil anlama ve hatta yaratıcı düşünme gibi yetenekleri bilgisayarlara kazandırmayı hedefler. Yapay zeka, geniş bir yelpazede uygulamalarla insan yaşamını dönüştüren yenilikçi bir teknoloji olarak öne çıkmaktadır. (Pirim, 2006)



**Şekil 2.1.** Yapay zeka'nın temel bileşenleri

Şekil 2.1.'de Yapay zeka'nın temel bileşenleri gösterilmektedir. Yapay zeka, insan zekasına benzer işlevleri gerçekleştiren sistemler geliştirmek için birçok farklı teknolojiyi ve yöntemleri bir araya getirir. Bunlar arasında makine öğrenmesi, doğal dil işleme, görüntü işleme-bilgisayarlı görü, derin öğrenme ve robotik gibi alanları sayılabilir. Ayrıca endüstri, otomasyon, tıp, finans, hukuk ve öneri sistemleri gibi uygulama alanlarında da kullanılmaktadır.

Yapay zekanın önemli bir dalı olan makine öğrenmesi, bilgisayarların açıkça programlanmadan verilerden öğrenmesini sağlar. Algoritmalar, büyük veri setlerinden kalıpları ve eğilimleri öğrenerek yeni verilerle doğru tahminler yapar. Örneğin, e-posta sistemlerinde spam e-postaları otomatik olarak tanımak için makine öğrenmesi algoritmaları kullanılır.

Doğal Dil İşleme (Natural Language Processing - NLP), makinelerin insan dilini anlaması, işlemesi ve üretmesi ile ilgilenir. Sesli asistanlar, çeviri uygulamaları ve metin analiz sistemleri, doğal dil işlemenin yaygın uygulamalarıdır. NLP, metin anlama, dil çevirisi ve duygu analizi gibi birçok görevi yerine getirebilir (Adalı, 2012).

Bilgisayarla görü, makinelerin görsel verileri analiz edip anlamasını sağlar. Bu teknoloji, görüntü işleme ve nesne tanıma gibi görevlerde kullanılır. Otonom araçlar, güvenlik kameraları ve yüz tanıma sistemleri, bilgisayarla görü tekniklerini kullanır.

Derin öğrenme, yapay sinir ağlarını kullanarak büyük ve karmaşık veri setlerinden öğrenmeyi sağlayan bir makine öğrenmesi türüdür. Özellikle görüntü, ses ve dil işleme gibi karmaşık veriler üzerinde başarılıdır. Derin öğrenme, çok katmanlı sinir ağları (yapay sinir ağları) kullanarak daha derin ve karmaşık ilişkileri öğrenebilir.

Yapay zeka, robotik sistemlere entegre edilerek otonom ve akıllı makinelerin geliştirilmesine olanak tanır. Otonom robotlar, çevrelerini algılayıp bağımsız kararlar alabilirler. Endüstriyel robotlar, otonom araçlar ve insansız hava araçları, yapay zeka temelli robotik sistemlere örnektir.

Yapay zeka, tıbbi görüntü analizi, hastalık teşhisi ve ilaç geliştirme süreçlerinde yaygın olarak kullanılmaktadır. Özellikle kanser teşhisi ve kişiselleştirilmiş tedavi planları gibi alanlarda başarılı sonuçlar elde edilmiştir.

Finansal analiz, ticaret, risk yönetimi ve dolandırıcılık tespiti gibi alanlarda yapay zeka algoritmaları kullanılmaktadır. Makine öğrenmesi, yatırım stratejilerini optimize etme ve büyük veri analizi yapma konusunda önemli bir rol oynar.

Ürün öneri sistemleri, kişiselleştirilmiş reklamcılık ve müşteri analizi gibi uygulamalarda yapay zeka, kullanıcıların davranışlarını analiz ederek daha doğru ve hedefe yönelik öneriler sunar.

Otonom sürüş teknolojisi, yapay zeka ile geliştirilmiştir. Otonom araçlar, çevresel verileri analiz ederek trafik, yaya ve yol bilgilerini anında işleyip sürüş kararları alabilirler.

Yapay zeka, eğitim teknolojilerinde kişiselleştirilmiş öğrenme deneyimleri sunmak için kullanılmaktadır. Öğrenci performansını analiz edip uygun öğrenme materyalleri öneren yapay zeka sistemleri, eğitimde devrim yaratmaktadır.

Yapay zeka, hukuki belgelerin analizi, sözleşme oluşturma ve güvenlik tehditlerinin tespiti gibi görevlerde kullanılmaktadır.

Yapay zeka, insanlara göre çok daha hızlı ve doğru bir şekilde veri işleyebilir. Özellikle büyük veri analizi gerektiren işlerde yapay zeka sistemleri, verimliliği artırarak daha kısa sürede sonuç elde edebilir. Yapay zeka, kullanıcı davranışlarını analiz ederek kişiye özel öneriler ve deneyimler sunabilir. İnsan hatalarını minimuma indirerek daha güvenilir sonuçlar üretir. Özellikle tekrarlayan ve karmaşık görevlerde yapay zeka, hata oranlarını düşürür. Zaman alıcı ve rutin görevleri otomatikleştirerek, çalışanların daha yaratıcı ve stratejik işlere odaklanmasına olanak tanınması avantaj olarak sayılabilir.

### **2.1.1. Yapay Zekanın Zorlukları ve Etik Sorunlar**

Yapay zeka teknolojisinin hızla gelişmesi, bazı zorlukları ve etik sorunları da beraberinde getirmektedir.

Yapay zeka temelli otomasyon, bazı işlerin makineler tarafından devralınmasına yol açabilir. Bu durum, iş kayıpları ve işsizlik oranlarının artması gibi sorunları gündeme getirebilir. Yapay zeka sistemleri büyük miktarda veri kullanır. Bu, kişisel verilerin gizliliği ve güvenliği konularında endişelere yol açabilir. Yapay zeka algoritmaları, veri setlerindeki önyargıları yansıtabilir. Yanlıştır veya taraflı verilerle eğitilen yapay zeka sistemleri, adil olmayan kararlar alabilir. Derin öğrenme gibi bazı yapay zeka modelleri, "kara kutu" olarak adlandırılan, son derece karmaşık yapılar içerir. Bu modellerin nasıl karar verdiklerini anlamak zor olabilir ve bu da güven sorunlarına yol açabilir olması yapay zekanın dezavantajlarını ve etik sorunlarını oluşturmaktadır (Yeşilkaya, 2022).

### **2.2. Makine Öğrenmesi**

Makine Öğrenmesi (Machine Learning - ML), bilgisayarların açıkça programlanmadan öğrenebilmesini sağlayan bir yapay zeka dalıdır. Klasik yazılım yaklaşımlarında, bir sistemin nasıl çalışacağını sistemi programlayan insanlar belirlerken, makine öğrenmesi modelleri büyük veri setlerinden öğrenerek kendilerini geliştirme yeteneğine sahiptir (González vd., 2019). Bu süreç, verilerdeki kalıpların, ilişkilerin ve eğilimlerin keşfi ile gerçekleşir. Makine öğrenmesi, otomatik karar alma sistemlerinden, kişiselleştirilmiş önerilere kadar birçok alanda kullanılmaktadır (Atalay ve Çelik, 2017) .

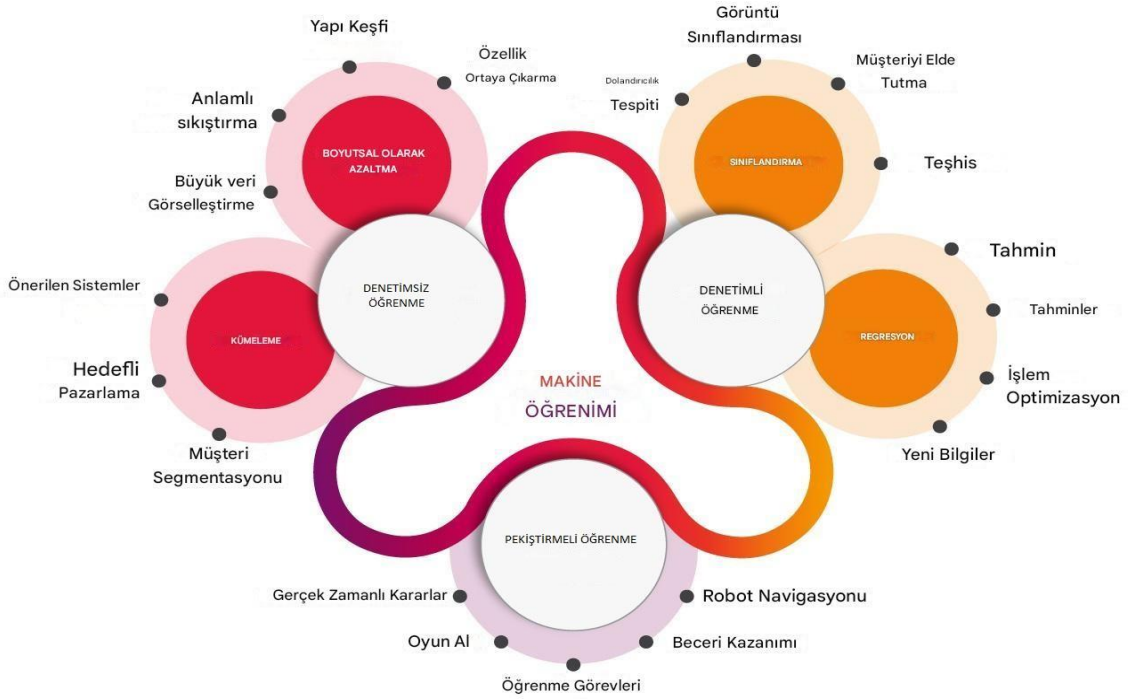
Makine öğrenmesi modelleri, verilerden öğrenerek bir problemi çözmek için veriler, model geliştirme ve öğrenme algoritmaları olmak üzere üç temel bileşene dayanmaktadır.

Veriler, makine öğrenmesinin temelini oluşturur. Model, ham veriler üzerinde öğrenir. Bu veriler, sayısal veya kategorik olabilir. Veri kalitesi ve miktarı, modelin performansını doğrudan etkiler. Büyük ve doğru veri setleri, daha iyi sonuçlar elde etmek için önemlidir.

Model, verilerden öğrenen matematiksel bir yapıdır. Modelin amacı, girdi verisi ile çıktı arasındaki ilişkiyi öğrenmek ve tahmin yapmaktır. Probleme uygun farklı makine öğrenmesi algoritmaları ve farklı türde modeller kullanılabilir.

Algoritmalar, verilerden öğrenmeyi sağlayan kurallardır. Modelin nasıl eğitileceğini ve nasıl optimize edileceğini belirler. Her algoritma, verilerdeki kalıpları bulmak ve bu kalıplara dayanarak kararlar almak için farklı yaklaşımlar kullanır.

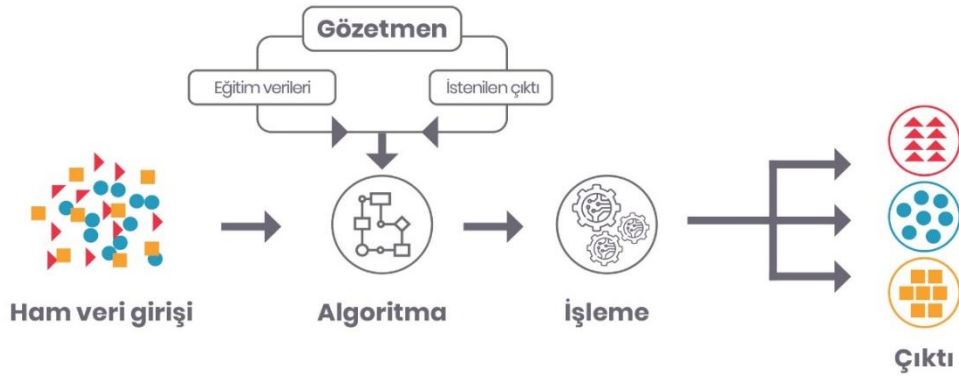
Şekil 2.2.'de Makine öğrenmesinin denetimli, denetimsiz ve pekiştirmeli olmak üzere üç ana türe ayrıldığı ve bu türlere ilişkin bileşenlerin kullanım alanları gösterilmektedir.



Şekil 2.2. Makine öğrenmesine ait temel bileşenler ve kullanım alanları

**Kaynak:** (Efe, 2021)

Denetimli öğrenmede, model belirli bir giriş ve ona karşılık gelen doğru çıkış verileri ile eğitilir. Model, girdiler ve çıktılar arasındaki ilişkiyi öğrenir ve yeni verilerle karşılaştığında doğru tahminler yapmayı amaçlar (Saba Raof ve Durai, 2022). Denetimli öğrenme sınıflandırma ve regresyon problemlerin çözümünde yaygın olarak kullanılmaktadır. Şekil 2.3.'de Denetimli makine öğrenme modeli gösterilmiştir.



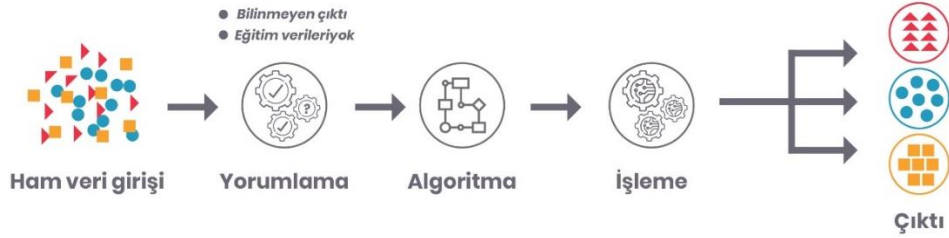
Şekil 2.3. Denetimli öğrenme

**Kaynak:** (Yurtsever, 2024)

Sınıflandırma (Classification), Model, veriyi belirli sınıflara ayırmayı öğrenir. Örneğin, bir e-posta sisteminde model, gelen e-postaları "spam" veya "spam değil" olarak sınıflandırabilir.

Regresyon (Regression), Regresyon problemlerinde model, sürekli bir değer tahmini yapar. Örneğin, bir evin fiyatını, konum, büyüklük ve yaş gibi faktörlere dayalı olarak tahmin edebilir.

Denetimsiz öğrenmede, model sadece giriş verileri ile eğitilir. Veri üzerinde herhangi bir etiket veya doğru çıktı yoktur. Model, veriler arasındaki gizli kalıpları ve ilişkileri keşfetmeyi amaçlar (Ahmad vd., 2022). Denetimsiz öğrenme kümeleme ve boyut indirgeme problemlerin çözümünde yaygın olarak kullanılmaktadır. Şekil 2.4.'de Denetimsiz makine öğrenme modeli gösterilmiştir.



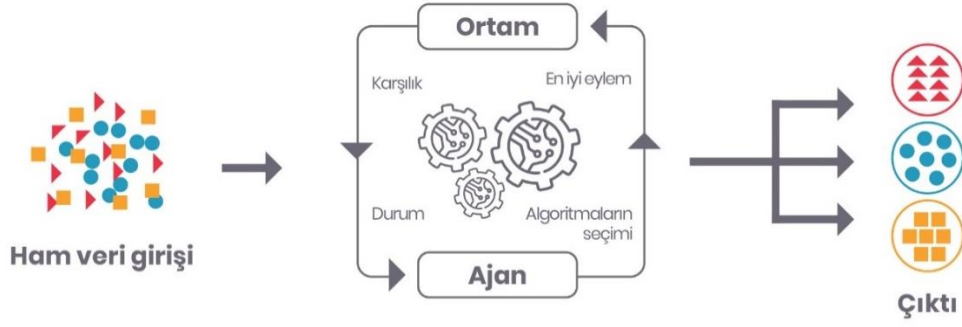
**Şekil 2.4.** Denetimsiz öğrenme

**Kaynak:** (Yurtsever, 2024)

Kümeleme (Clustering), model verileri gruplara ayırır. Örneğin, müşteri verileri üzerinde yapılan bir kümeleme analizinde, benzer alışveriş davranışlarına sahip müşteri grupları bulunabilir.

Boyut İndirgeme (Dimensionality Reduction), yüksek boyutlu verilerde önemli özellikleri belirleyip, veri boyutunu azaltarak, daha az bilgi kaybıyla veriyi sadeleştiren bir tekniktir.

Şekil 2.5.'de Pekiştirmeli öğrenme, bir modelin ödül ve ceza sistemi üzerinden öğrenmesini sağlar. Model, bir ortamda kararlar alır ve bu kararların sonucunda ödül veya ceza alarak kendini optimize eder. Bu öğrenme türü, özellikle oyun oynayan yapay zekalarda ve otonom sistemlerde yaygındır (Gökalp, 2022).



**Şekil 2.5.** Pekiştirmeli öğrenme

**Kaynak:** (Yurtsever, 2024)

Makine öğrenmesi, birçok farklı algoritma kullanılmaktadır. En yaygın kullanılan makine öğrenmesi algoritmalarına doğrusal regresyon, karar ağaçları, destek vektör makineleri, k-en yakın komşu algoritması ve yapay sinir ağları örnek olarak verilebilir.

Doğrusal regresyon, bağımsız değişkenlerle bağımlı değişken arasındaki doğrusal ilişkiyi modelleyen bir algoritmadır. Bu yöntem, sürekli bir hedef değişkenin tahmini için kullanılır.

Karar ağaçları, veriyi sınıflandırma ve regresyon görevlerinde kullanılan bir algoritmadır. Her bir dal, belirli bir özelliğe dayalı karar noktalarını temsil eder ve model, veriyi bu ağaç yapısına göre sınıflandırır.

Destek Vektör Makineleri (Support Vector Machines - SVM), iki sınıf arasındaki en iyi ayırma çizgisini bulmaya çalışan bir sınıflandırma algoritmasıdır. Bu algoritma, özellikle yüksek boyutlu verilerde başarılı sonuçlar verir.

K-En Yakın Komşu (K-Nearest Neighbors - KNN), sınıflandırma ve regresyon problemlerinde kullanılan basit bir algoritmadır. Veriler arasındaki mesafeleri kullanarak, yeni bir veri noktasının en yakın komşularına bakar ve onların sınıfına göre tahmin yapar.

K-Ortalamalar (K-Means), verileri belirli sayıda kümeye ayırarak, her kümeye bir merkez belirler ve verileri bu merkezlere göre gruplandırır (Erdoğan vd., 2016).

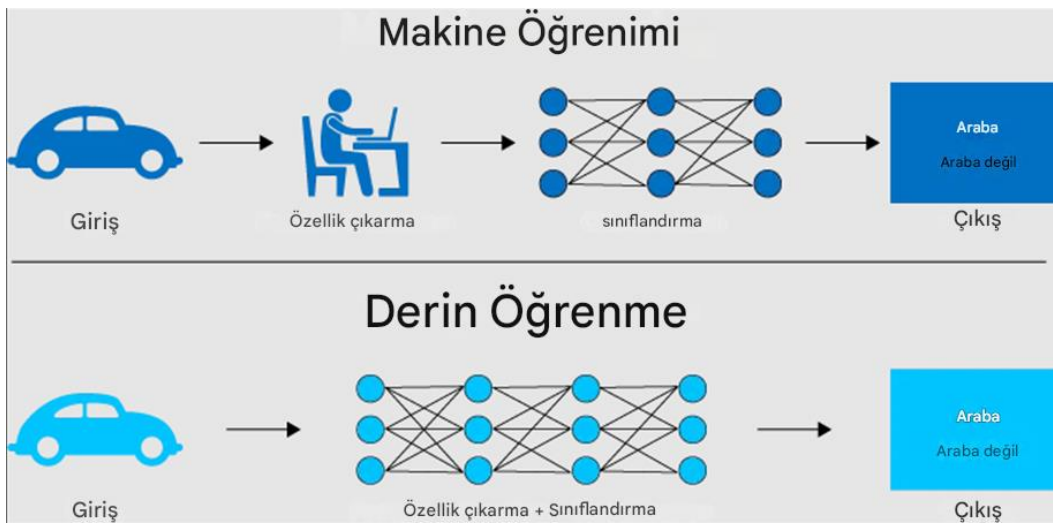
Hiyerarşik Kümeleme (Hierarchical Clustering), verileri hiyerarşik olarak gruplara ayırarak, bir ağaç yapısında kümeler oluşturur.

Yapay Sinir Ağları (Artificial Neural Networks - ANN), insan beynindeki sinir ağlarından esinlenilerek geliştirilmiş, çok katmanlı yapılara sahip bir modeldir. Özellikle derin

öğrenme, sinir ağları üzerinden gerçekleştirilen makine öğrenmesi algoritmalarının bir alt alanıdır.

### 2.3. Derin Öğrenme

Derin Öğrenme (Deep Learning - DL), yapay zekanın ve makine öğrenmesinin bir alt alanı olarak tanımlanabilir. DL, özellikle büyük veri setlerinden otomatik olarak öğrenebilen, çok katmanlı ANN'yi temel alır. Bu katmanlı yapı sayesinde, verilerdeki karmaşık ve yüksek seviyeli özellikleri öğrenerek, tahminler, sınıflandırmalar ve keşifler yapabilir. Derin öğrenmenin temel amacı, ham verilerden anlamlı bilgiler çıkarmaktır (Doğan & Türkoğlu, 2019). Şekil 2.6.'de ML ve DL yöntemleri arasındaki farkı gösterilmektedir.



Şekil 2.6. Makine öğrenmesi ve Derin öğrenme yöntemlerinin farkı

**Kaynak:** (Wiguna vd., 2022)

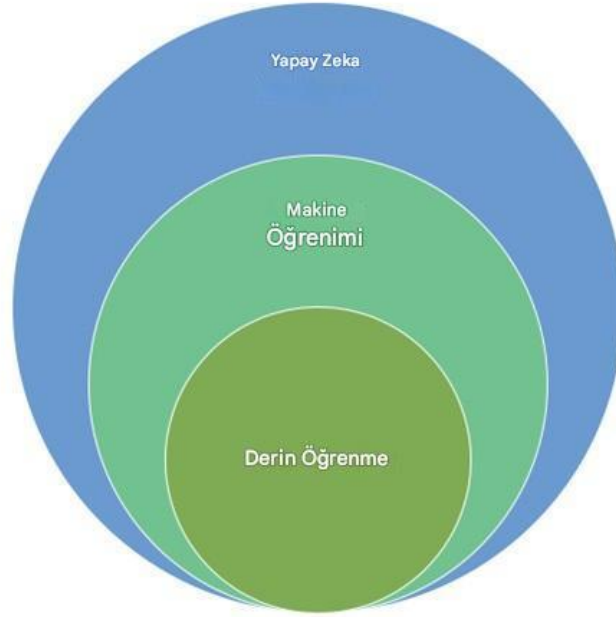
DL, çok katmanlı sinir ağları kullanarak öğrenme gerçekleştirir. Bu ağlar, katman sayısına bağlı olarak "derin" hale gelir. Genellikle bir derin öğrenme modelinde şunlar bulunur:

Girdi Katmanı (Input Layer), bu katman, verinin ağa verildiği yerdir. Görüntü, ses veya metin gibi çeşitli veri tipleri olabilir.

Gizli Katmanlar (Hidden Layers), bu katmanlar, girdilerden özellikler çıkaran çok sayıda nörondan oluşur. Derin öğrenmenin gücü, bu gizli katmanlarda yatar. Gizli katmanlar arasında veriyi işleyerek yüksek seviyeli temsiller üretir.

Çıktı Katmanı (Output Layer), bu katman, son tahminin veya sınıflandırmanın yapıldığı yerdir. Modelin amacına bağlı olarak, bir sınıflandırma, regresyon veya başka bir görev gerçekleştirilir (Şeker vd., 2017).

Şekil 2.7.' de Yapay zeka, Makine öğrenimi ve Derin öğrenmenin kavramsal ilişkileri gösterilmektedir.



**Şekil 2.7.** Yapay zeka, Makine öğrenimi ve Derin öğrenmenin gösterimi

DL farklı yapılarıdaki sinir ağlarını kullanarak çok çeşitli problemleri çözebilir. En yaygın kullanılan derin öğrenme algoritmaları ve modelleri arasında ANN, CNN, RNN ve alt modelleri sayılabilir.

ANN, en basit sinir ağı türüdür. Çok katmanlı yapılarla veriden öğrenme gerçekleştirir.

CNN, genellikle görüntü işleme, nesne tanıma ve görüntü sınıflandırmada kullanılır. Evrişimli katmanlar, verinin yerel özelliklerini öğrenmek için filtreler kullanır. Özellikle iki boyutlu verilerde (örneğin görüntüler) başarılıdır (Wiguna vd., 2022).

RNN, zaman serisi verileri ve sıralı verilerde kullanılır. RNN'ler, önceki adımlardan gelen bilgiyi hafızasında tutarak zaman içindeki bağımlılıkları öğrenir. Doğal dil işleme ve konuşma tanıma gibi alanlarda kullanılır.

LSTM, RNN'lerin bir çeşididir ve uzun süreli bağımlılıkları daha etkili bir şekilde öğrenmek için geliştirilmiştir. Zaman serisi analizlerinde ve sıralı veri işleme görevlerinde yaygındır.

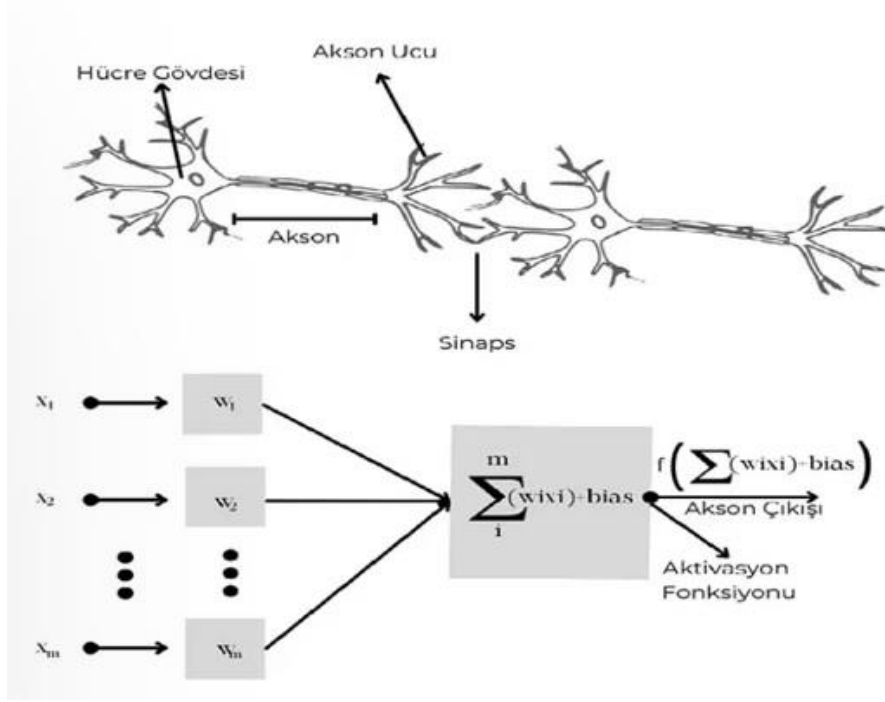
Oto-Kodlayıcılar, veri sıkıştırma ve özellik çıkarımında kullanılan bir sinir ağı türüdür. Girdi verisini daha düşük boyutlu bir temsile indirger, sonra tekrar orijinaline yakın bir veri üretir.

Derin öğrenme modelleri, büyük veri setleri üzerinde çok yüksek performans gösterir. Özellikle görüntü, ses ve metin gibi karmaşık verilerde başarılı sonuçlar verir. Derin öğrenme modelleri, verilerden otomatik olarak özellik çıkarabilir. Bu, geleneksel makine öğrenmesi modellerinin aksine, manuel özellik mühendisliği ihtiyacını ortadan kaldırır. Derin öğrenme, farklı veri türleri üzerinde uygulanabilir. Görüntü, ses, metin, zaman serisi gibi çok çeşitli veri türleri üzerinde çalışabilir olması avantajlarını oluşturmaktadır.(Pouyanfar vd., 2018)

Derin öğrenme, genellikle büyük miktarda veriye ihtiyaç duyar. Küçük veri setlerinde iyi performans göstermeyebilir. Derin öğrenme modelleri, eğitilmesi ve çalıştırılması için güçlü donanım gerektirir. Bu, özellikle büyük ve karmaşık modellerde daha belirgindir. Derin ağlar, çok fazla parametreye sahip oldukları için aşırı öğrenme riski taşıyabilir. Bu, modelin eğitim verilerine çok iyi uyum sağladığı ama genel verilerde kötü performans gösterdiği anlamına gelmesi de dezavantaj olarak sayılabilir.(Guo vd., 2016)

#### **2.4. Yapay Sinir Ağları**

ANN, Şekil 2.8.' de gösterildiği gibi insan beynindeki sinir hücrelerinin (nöronların) çalışma biçiminden esinlenerek geliştirilmiş bir makine öğrenme modelidir. ANN, karmaşık problemleri çözmek, verilerden öğrenmek ve bu öğrenilen bilgiyi gelecekteki durumlara uygulamak için kullanılan güçlü bir algoritmadır. Yapay zeka ve makine öğrenmesinin temel taşlarından biri olan yapay sinir ağları, birçok farklı alanda başarılı uygulamalara sahiptir (Arı ve Berberler, 2017).



**Şekil 2.8.** Sinir hücresi ve yapay sinir ağı

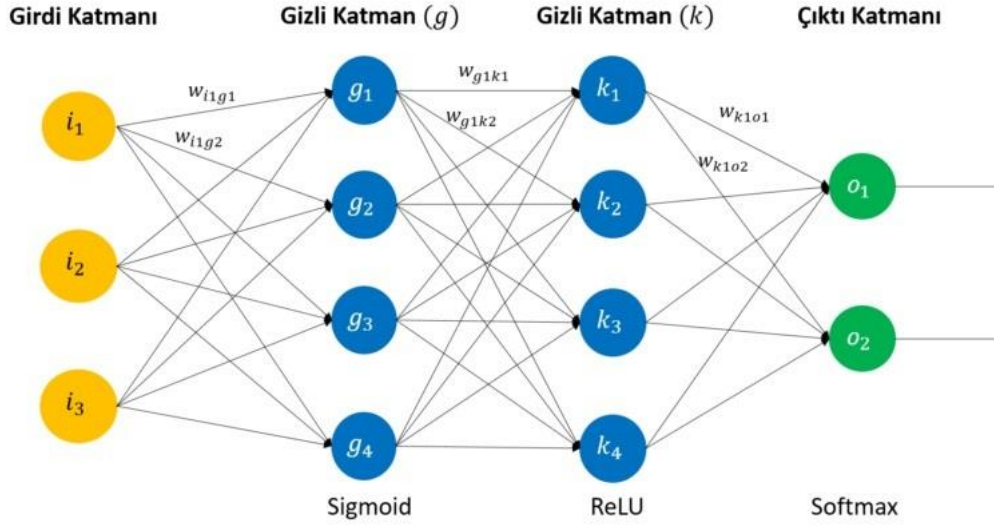
**Kaynak:** (Diri, 2012)

ANN, nöronlar (birimler) ve bu nöronları birbirine bağlayan ağırlıklı bağlantılardan oluşur. Her bir nöron, aldığı girdiyi işler ve bir çıktı üretir. Ağlar Şekil 2.9.'da gösterildiği gibi genellikle girdi, gizli katman ve çıktı katmanından meydana gelmektedir.

Girdi katmanı, dış dünyadan gelen verilerin alındığı katmandır. Bu veriler, ağıdaki her nörona bir sinyal olarak iletilir. Örneğin, bir görüntü verisi ağın girdi katmanına piksel değerleri olarak sunulabilir.

Gizli katmanlar, girdiyi işler ve içyapısındaki özellikleri çıkarır. Gizli katmanlar, ağın derinliğini belirler. Bu katmanlarda, nöronlar arasında bağlantılar vardır ve her bir bağlantı ağırlıklarla çarpılarak işlenir. Çok katmanlı yapılar, daha karmaşık veri ilişkilerini öğrenmeyi sağlar.

Çıktı katmanı, ağın son katmanıdır ve problemin çözümüyle ilgili nihai sonuçları üretir. Örneğin, bir sınıflandırma probleminde çıktı katmanında farklı sınıflar arasından en olası sonuç belirlenir.



**Şekil 2.9.** Yapay sinir ağlarının çalışma prensibi

**Kaynak:** (Özbay, 2022)

#### 2.4.1. Nöronların Çalışma Prensibi

ANN’lerde her bir nöron, kendisine gelen girdilerle ağırlıklı toplam ve aktivasyon fonksiyonu sürecinden geçerek sonuç üretmektedir (Şekil 2.9.).

Ağırlıklı toplam, girdiler her bir nörona belirli ağırlıklarla gelir. Her bir nöron, girdileri kendi ağırlıklarıyla çarpar ve toplar. Bu süreç, nöronun karar alması için ön koşuldur.

Aktivasyon fonksiyonu, ağırlıklı toplam sonucu bir aktivasyon fonksiyonundan geçirilir. Bu fonksiyon, girdiyi bir sınıra getirip çıkışın ne olacağını belirler. Aktivasyon fonksiyonu, nöronların karmaşık ilişkileri öğrenmesine ve doğrusal olmayan ilişkileri modellemesine yardımcı olur.

ANN’nin eğitimi, ağırlıkların veriden öğrenilmesi sürecidir. Bu süreç genellikle ileri yayılım (forward propagation) ve geri yayılım (backpropagation) algoritmalarıyla gerçekleştirilir (Şalvarcı ve Ayten, 2019).

İleri Yayılım, girdi verisi ağ boyunca ilerletilir. Her bir katmandaki nöronlar, önceki katmandan gelen verileri işler ve sonuçları bir sonraki katmana iletir. Çıktı katmanında nihai sonuç elde edilir.

Geri Yayılım (Backpropagation), hata hesaplandıktan sonra, bu hata ağa geri iletilir. Geri yayılım algoritması, her bir ağırlık için türev hesaplar ve bu türevleri kullanarak ağırlıkları

günceller. Bu sayede model, her yinelemede hatayı azaltarak daha doğru sonuçlar üretmeyi öğrenir.

## 2.5. Evrişimli Sinir Ağları (CNN)

Evrişimli Sinir Ağları, özellikle görüntü işleme ve bilgisayarla görü gibi alanlarda kullanılan derin öğrenme modellerinden biridir. CNN' ler, büyük veri setlerinden öğrenerek görüntülerdeki özellikleri otomatik olarak çıkarır ve bu sayede görüntü sınıflandırma, nesne tanıma ve yüz tanıma gibi görevlerde yüksek performans sergiler. CNN' lerin başarısı, görüntülerdeki mekânsal ilişkileri ve özellikleri öğrenme yeteneklerine dayanır (Kızrak ve Bolat, 2018).

CNN, geleneksel yapay sinir ağlarından farklı olarak, görüntü verilerinin özel yapısını ve mekânsal ilişkilerini dikkate alacak şekilde tasarlanmıştır. CNN' ler, genellikle birkaç temel katmandan oluşur (Gu vd., 2018).

CNN' lere genellikle görüntü verileri giriş olarak verilir. Görüntü, piksel değerlerinden oluşan bir matrise dönüştürülür. Örneğin, bir renkli görüntü 3 kanallı (kırmızı, yeşil, mavi - RGB) bir veri matrisi olarak temsil edilir.

CNN' lerin en önemli yapı taşlarından biri olan evrişim katmanları, girdi verisi üzerinde filtreler kullanarak görüntüdeki özellikleri çıkarmayı sağlar. Her filtre, görüntü üzerinde belirli bir pencere boyunca kayarak yerel özellikleri tespit eder. Örneğin, bir filtre kenar, köşe veya renk değişiklikleri gibi düşük seviyeli özellikleri tanıyabilir. Evrişim katmanları, görüntülerdeki yerel bağlantıları öğrenerek uzaysal bilgiyi korur ve her filtre görüntünün farklı bir özelliğini çıkarır.

Filtreler, küçük boyutlu matrislerdir (örneğin 3x3 veya 5x5). Bu filtreler, görüntünün üzerinde kayarak evrişim işlemi gerçekleştirir. Her evrişim işlemi, yeni bir özellik haritası (feature map) oluşturur.

Evrişim katmanının çıktısı genellikle doğrusal olmayan bir aktivasyon fonksiyonundan (çoğunlukla ReLU - Rectified Linear Unit) geçirilir. ReLU, negatif değerleri sıfırlar ve pozitif değerleri aynen aktarır, bu da modelin doğrusal olmayan özellikleri öğrenmesine yardımcı olur.

Havuzlama katmanları, veri boyutunu küçültmek ve hesaplama maliyetlerini azaltmak için kullanılır. Bu katman, genellikle evrişim katmanlarından sonra yer alır ve özellik haritasındaki önemli bilgileri koruyarak boyutları küçültür. En yaygın kullanılan havuzlama türü maksimum havuzlamadır. Maksimum havuzlama, belirli bir penceredeki (örneğin 2x2)

maksimum değeri seçerek bilgiyi yoğunlaştırır. Bu süreç, modelin daha az parametre ile öğrenmesini sağlar ve aşırı öğrenmeyi azaltır.

Evrişim ve havuzlama katmanlarının ardından gelen tam bağlantılı katman, sinir ağlarının klasik bileşenidir. Bu katman, görüntüden çıkarılan özelliklerin sınıflandırılmasını sağlar. Tam bağlantılı katmanda, her bir nöron bir önceki katmandaki tüm nöronlarla bağlantılıdır. Bu katman, görüntüdeki özellikleri kullanarak sınıflandırma yapar. Örneğin, bir CNN modelinde tam bağlantılı katman, bir görüntünün "kedi" veya "köpek" olup olmadığını belirleyebilir (Li vd., 2021).

CNN'nin son katmanı, sınıflandırma yapılacak sınıf sayısına göre nöronlardan oluşur. Son katmanda genellikle softmax gibi bir aktivasyon fonksiyonu kullanılır. Softmax, her sınıf için bir olasılık değeri üretir ve bu sayede model, bir görüntünün hangi sınıfa ait olduğunu belirler.

CNN için öğrenme sürecinde ilk olarak, görüntü verisi piksel değerleriyle temsil edilen bir giriş matrisi olarak CNN'ye verilir.

Evrişim ve havuzlama katmanları, görüntüdeki önemli yerel özellikleri (kenarlar, dokular vb.) tespit eder ve bu özellikleri öğrenir. İlk evrişim katmanları düşük seviyeli özellikleri (kenarlar, köşeler), daha derin katmanlar ise daha yüksek seviyeli özellikleri (şekiller, nesnelere) öğrenir.

Çıkarılan özellikler tam bağlantılı katmana iletilir. Bu katman, sınıflandırma yaparak görüntünün hangi kategoriye ait olduğunu belirler.

Modelin çıktısı gerçek sonuçlarla karşılaştırılır ve hatalar hesaplanır. Bu hatalar, ağırlıkları güncellemek için geriye yayılım algoritması ile modelin parametreleri üzerinde ayarlama yapar.

### **2.5.1. CNN'lerin Avantajları**

CNN'ler, görüntüdeki özellikleri otomatik olarak öğrenir ve çıkarır. Bu, manuel özellik mühendisliği gereksinimini ortadan kaldırır. Evrişim katmanları, görüntüdeki yerel bağlantıları ve mekansal bilgiyi koruyarak daha etkili bir öğrenme sağlar. CNN'ler, havuzlama katmanları sayesinde veri boyutunu küçülterek daha az parametreyle daha verimli öğrenir. Bu, hesaplama maliyetlerini düşürür ve aşırı öğrenmeyi engeller. CNN'ler, görüntü işleme ve sınıflandırma gibi alanlarda üstün performans sergiler ve günümüzde birçok bilgisayarla görü uygulamasında kullanılır olması CNN'lerin avantajlarını oluşturur.

### **2.5.2. CNN'lerin Uygulama Alanları**

CNN'ler, özellikle görüntü sınıflandırma, nesne tanıma, yüz tanıma, tıbbi görüntü analizi ve doğal dil işleme alanlarında büyük başarı göstermiştir:

Görüntü Sınıflandırma, CNN'ler, görüntüleri farklı sınıflara ayırmada mükemmel performans sergiler. Örneğin, bir model, kedileri köpeklerden ayırt edebilir.

Nesne Tanıma, CNN'ler, görüntülerdeki nesnelere tespit edebilir. Otonom araçlar, trafik işaretlerini veya yayaları tanımak için CNN'leri kullanır.

Yüz Tanıma, güvenlik sistemlerinde kullanılan yüz tanıma teknolojileri, CNN'ler sayesinde yüzleri yüksek doğrulukla tanıyabilir.

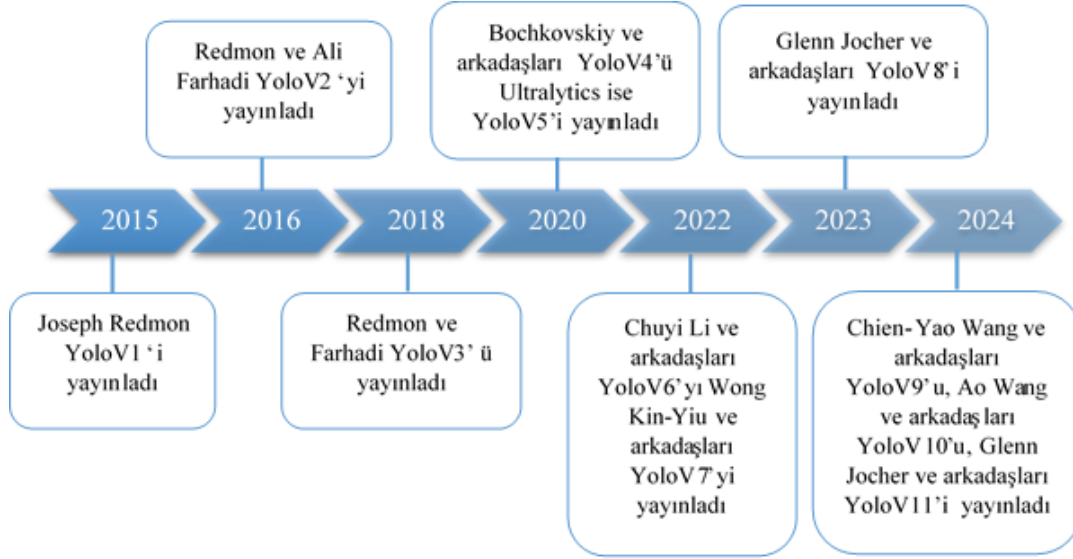
Tıbbi Görüntü Analizi, CNN'ler, tıbbi görüntülerde (röntgen, MR, tomografi) anormallikleri tespit etmek için kullanılır. Kanser gibi hastalıkların erken teşhisinde etkili bir yöntem olarak kullanılır.

Doğal Dil İşleme, CNN'ler, metin verileri üzerinde de kullanılabilir. Örneğin, metin sınıflandırma ve duygu analizi gibi görevlerde başarılıdır (Wang vd., 2024).

### **2.6. YOLO v11**

Bilgisayarla görme, yapay zekâ alanında hızla gelişen bir dal olup, nesne tespiti bu alanın en kritik bileşenlerinden biridir. YOLO algoritması, ilk olarak Redmon ve arkadaşları tarafından 2016 yılında geliştirilmiş ve nesne tespitinde çığır açan bir yaklaşım sunmuştur (Redmon vd., 2016). Şekil 2.10.'da yıllara göre YOLO algoritmasının farklı sürümlerinin ortaya çıkışı gösterilmiştir. YOLO ailesi, her yeni sürümde doğruluk, hız ve verimlilik açısından gelişerek günümüzde nesne tespitinin en yaygın kullanılan yöntemlerinden biri hâline gelmiştir. 2024 yılında tanıtılan YOLO v11, özellikle daha verimli bir mimari, gelişmiş veri işleme teknikleri ve çok yönlü kullanım alanlarıyla dikkat çekmektedir.

YOLO v11, önceki sürümlere kıyasla daha hızlı ve hassas nesne tespiti yapabilen bir modeldir. Modelin geliştirilmesinde, veri işleme süreçleri optimize edilerek daha az hesaplama gücü gerektiren bir mimari oluşturulmuştur. Derin öğrenme tabanlı bu nesne tespit modeli, gerçek zamanlı analizlerde en iyi performans sağlayan algoritmalarından biri olarak kabul edilmektedir (Sapkota vd., 2024).



**Şekil 2.10.** YOLO Sürüm tarihçesi

YOLO v11'in mimari yapısı, önceki sürümlere kıyasla önemli mimari geliştirmeler içermektedir. Şekil 2.11.' de bu mimari yapı gösterilmektedir.

Gelişmiş Omurga(Backbone); YOLO v11, önceki sürümlerde kullanılan CSP Darknet mimarisini daha da optimize ederek, daha yüksek verimlilik sağlayan yeni nesil bir özellik çıkarma ağı kullanmaktadır. Bu backbone, geniş ölçekli konvolüsyon katmanları ve derinlemesine ayrı konvolüsyonlar içererek daha fazla bağlamsal bilgi yakalar.

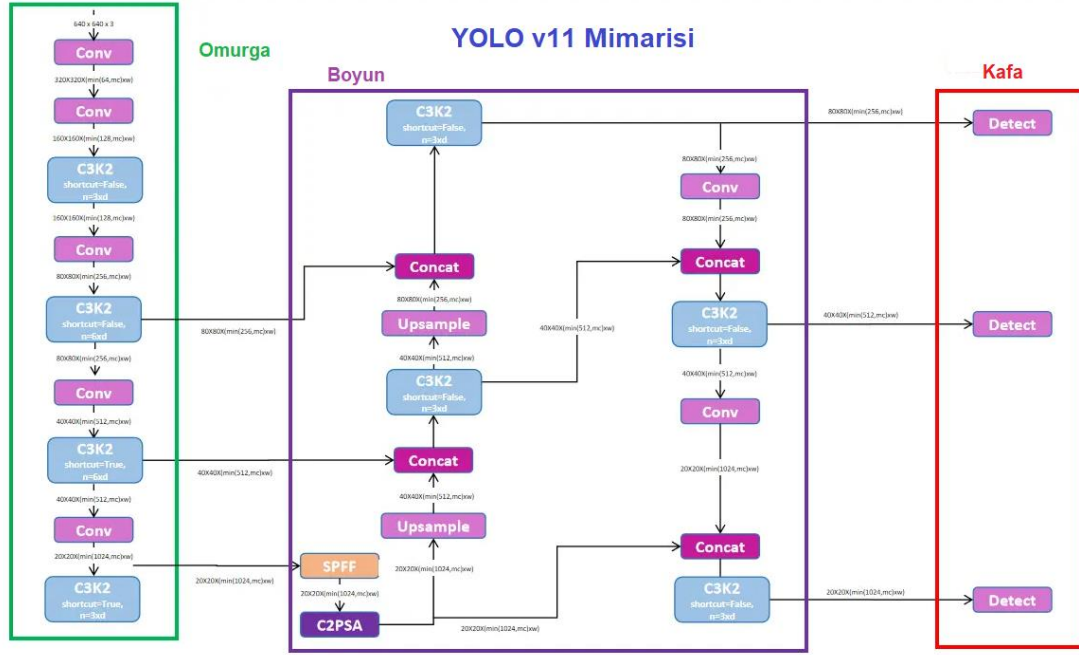
Transformer Entegrasyonu; YOLO v11, geleneksel CNN'e ek olarak, transformer tabanlı dikkat mekanizmalarını içermektedir. Bu entegrasyon, uzun mesafeli bağımlılıkları modelleyerek nesne sınırlarının daha iyi belirlenmesini sağlamaktadır. Görüntüdeki farklı bölgeler arasındaki ilişkileri daha iyi anlamak için self-attention mekanizmaları kullanılmaktadır (Khanam ve Hussain, 2024).

NMS' siz Eğitim; YOLO v11, geleneksel Non-Maximum Suppression (NMS) algoritmasını ortadan kaldırarak, yerine daha akıllı bir çerçeve filtreleme yöntemi getirmiştir. Bu yöntem, yinelenen tahminleri daha doğru bir şekilde değerlendiren ve doğruluk oranını artıran bir sıralama algoritması içerir. Böylece yanlış pozitif oranı azalırken, nesne tespit doğruluğu önemli ölçüde artmaktadır.

Büyük Çekirdekli Konvolüsyonlar; modelde kullanılan büyük çekirdekli konvolüsyon katmanları, daha geniş ölçekli özelliklerin çıkarılmasını sağlarken, düşük çözünürlüklü girişlerde bile ayrıntı kaybını minimuma indirir. Özellikle karmaşık sahnelerde tespit performansını artırarak, daha net nesne ayırımı sağlar.

Özelleştirilmiş Aktivasyon Fonksiyonları; geleneksel ReLU ve Leaky ReLU yerine, daha stabil gradyan akışı sağlayan ve küçük değerlerde bilgi kaybını önleyen Swish ve Mish aktivasyon fonksiyonları kullanılmıştır (Jocher, 2024).

Gelişmiş Çıkış Katmanı; YOLO v11, çıkış katmanında daha optimize edilmiş ankraj kutuları (anchor boxes) kullanarak farklı ölçeklerdeki nesnelere daha doğru bir şekilde sınıflandırmaktadır. Aynı zamanda, çok ölçekli özellik haritalarını kullanarak küçük nesnelere tespit etme başarısını artırmaktadır.



Şekil 2.11. YOLO v11' in Mimari yapısı

**Kaynak:** (D. Wang vd., 2025)

Daha Hafif Model Yapısı; Modelin parametre sayısı optimize edilerek hem hesaplama maliyeti düşürülmüş hem de gömülü sistemler ve mobil cihazlarda verimli çalışması sağlanmıştır. Yeni nesil kuantizasyon ve sıkıştırma teknikleri sayesinde, modelin bellek kullanımı da optimize edilmiştir.

Öğrenme ve Adaptasyon Yeteneği; YOLO v11, transfer öğrenme teknikleri ile farklı veri kümelerine hızla adapte olabilir ve yeni nesne türlerini öğrenebilir. Kendi kendine süpervizyonlu öğrenme teknikleri sayesinde etiketlenmemiş verilerden de faydalanabilir.

YOLO v11'in performansı, önceki YOLO sürümleri ve diğer nesne tespit algoritmalarıyla karşılaştırıldığında dikkat çekici iyileştirmeler sunmaktadır.

YOLO v11, hem ortalama doğruluk (mAP) değerlerinde hem de hız açısından önemli gelişmeler göstermektedir. Modelin parametre sayısındaki azalmaya rağmen, performansının artmış olması dikkat çekicidir.

**Tablo 2.1.** YOLO Sürümlerinin performans karşılaştırması

| <b>Model</b>    | <b>mAP</b>   | <b>FPS (Frame Per Second)</b> | <b>Parametre Sayısı</b> |
|-----------------|--------------|-------------------------------|-------------------------|
| YOLO v9         | 50.3%        | 230                           | 56M                     |
| YOLO v10        | 54.6%        | 275                           | 48M                     |
| <b>YOLO v11</b> | <b>58.2%</b> | <b>310</b>                    | <b>42M</b>              |

**Kaynak:** (Jocher, 2024)

YOLO v11'in sunduğu gelişmiş tespit yetenekleri, birçok farklı sektörde kullanılmaktadır:

Sağlık Teknolojileri; YOLO v11, tıbbi görüntülerde anormalliklerin tespitinde yüksek başarı göstermektedir (Örneğin, tümör tespiti ve X-ray analizleri).

Akıllı Şehirler; YOLO v11, trafik analizi, otomatik plaka tanıma ve yaya güvenliği sistemlerinde kullanılmaktadır.

Otonom Araçlar; otonom sürüş teknolojilerinde, çevresel nesnelerin doğru şekilde algılanması için YOLO v11 kullanılmaktadır.

Tarım Teknolojileri; drone görüntülerinin analizi yoluyla bitki sağlığı takibi ve zararlı tespiti gibi uygulamalarda da kullanılmaktadır.

Güvenlik ve Savunma; YOLO v11, video gözetim sistemlerinde şüpheli hareketleri tespit etmek ve güvenlik tehditlerini analiz etmek için kullanılmaktadır.

Üretim hatlarında kalite kontrol süreçlerinde, müşteri analizleri ve raf yönetimi için de kullanılmaktadır.

### 3. VERİ SETİ HAZIRLANMASI VE LAHANA ZARARLISI

Model eğitimi, önceden hazırlanmış veri setleri veya sonradan oluşturulan veri setleri kullanılarak gerçekleştirilebilmektedir. Eğitim amacıyla kullanılacak açık kaynaklı birçok veri setine, internet üzerinden ücretsiz olarak erişim sağlanabilmektedir. Ancak bu veri setlerinin çoğu genellikle genel kullanım için hazırlandığından, belirli projeler için gereken özel veri setleri her zaman bulunamayabilir. Bu tür durumlarda, proje gereksinimlerine uygun bir veri setinin özel olarak oluşturulması gereklidir (Ök, 2022) . İhtiyaç duyulan veri setini oluşturmak amacıyla, web kazıma teknikleriyle internette görsel toplama veya fotoğraf çekimi gibi yöntemler kullanılabilir. Bu çalışmada, Bilecik ili Pazaryeri ilçesinde Şekil 3.1.'de koordinatları verilen alandan fotoğraf çekimi yapılarak lahana görselleri tek tek toplanmış ve orijinal bir veri seti oluşturulmuştur. Ancak, özgün bir veri seti oluşturma süreci oldukça zaman alıcı olmuştur. Çalışmada kullanılan veri seti, toplamda 3960 görsel içermektedir.



Şekil 3.1. Bilecik/Pazaryeri lahana görsellerinin toplandığı alan

### 3.1. Veri Etiketlemesi

Yapay zekâ modellerinin eğitimi sürecinde, veri seti etiketleme işlemi kritik bir öneme sahiptir. Etiketleme işlemi, modele verilen bir girdiye karşılık beklenen çıktının tanımlanmasını sağlamaktadır. Bu yöntem sayesinde model, bir girdiye karşılık nasıl bir çıktı üretmesi gerektiğini öğrenir. Veri seti etiketlemesi, hem nesne tanıma hem de nesne tespiti modellerinde farklı yaklaşımlar gerektirir. Nesne tanıma modellerinde, yalnızca nesnenin ne olduğunun modele öğretilmesi hedeflenirken, nesne tespiti modellerinde ise nesnenin türü ile birlikte görsel üzerindeki konumunun da öğretilmesi amaçlanır. Bu nedenle, veri seti etiketleme yöntemleri proje türüne göre değişiklik göstermektedir (Ök, 2022).

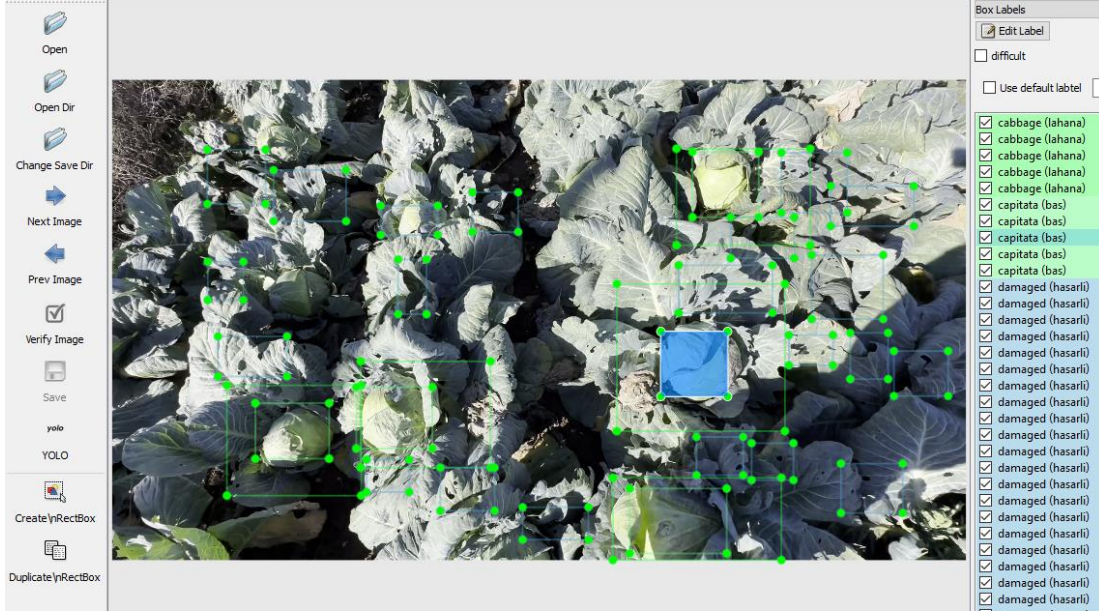
Nesne tanıma için veri seti etiketleme işlemi genellikle hazır yazılımlar kullanılarak otomatikleştirilebilir. Ancak, nesne tespiti işlemlerinde böyle bir yazılım bulunmadığından, tüm etiketleme sürecinin el yordamıyla gerçekleştirilmesi gerekmektedir. Bu süreçte, bir görseldeki her nesne tek tek belirlenip bir kutu içine alınarak her birine bir etiket adı atanır. Bu işlem hem masaüstü yazılımları hem de çevrimiçi platformlar aracılığıyla gerçekleştirilebilmektedir.

### 3.2. Görsel Etiketleme Araçları

Veri etiketleme yazılımları pek çok farklı özellikleriyle birbirinden ayrılmaktadır. Bazı yazılımlar ücretli olarak kullanılırken, ücretsiz olarak açık kaynaklı sunulan yazılımlar da bulunmaktadır. Etiketlenen verinin türüne göre nesne sınıflandırma, segmentasyon gibi görevler için etiketleme yapılmaktadır. Bazı araçlar yapay zeka desteklidir ve etiketleme işlemini hızlandırabilmektedir. Etiketlemede kullanılacak araçlardan bazıları şu şekildedir:

- MakeSense.AI
- LabelImg
- VGG image annotator
- LabelMe
- Scalable
- RectLabel
- Roboflow

Bu araçlardan ücretsiz olması ve kolay kullanımı sebebiyle LabelImg seçilerek etiketleme yapılmıştır.



**Şekil 3.2.** LabelImg programı kullanılarak etiketlenmiş bir fotoğraf

Yapılan bu çalışmada etiketleme işlemi için LabelImg programı tercih edilmiştir. Şekil 3.2.' de LabelImg programı kullanılarak her bir fotoğraf cabbage (lahana), capitata (baş), damaged (hasarlı) olmak üzere 3 farklı etiket türünde işaretlenmiştir.

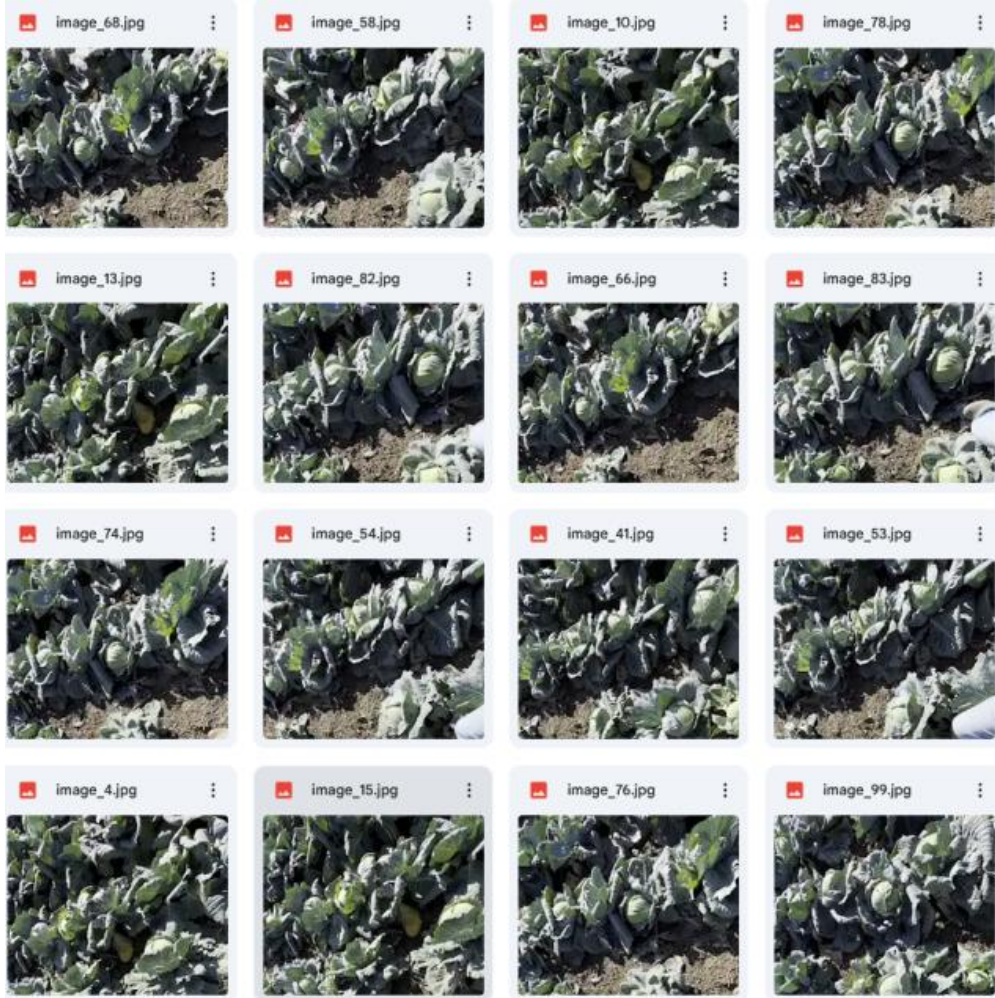
Her bir etiketin sahip olduğu koordinat değerleri Tablo 3.1.' de gösterilmektedir. Bu değerler sayısal formatta ifade edilmekte ve her bir görsel için otomatik olarak ayrı bir metin dosyası oluşturulmaktadır. İşaretlenen alandaki değerler, belirli bir sıraya göre bu metin dosyalarının içerisine kaydedilmektedir. Her bir metin dosyasında birden fazla etiket bulunabilmektedir. Tablo 3.1.' de etiketleme aşamasında elde edilen verilerden A sütünü etiket türünü göstermektedir. A sütündeki 0 rakamı lahana etiketini, 1 rakamı baş etiketini, 2 rakamı hasarlı alanların etiketini ifade etmektedir. B sütünü nesnenin dikdörtgen sınır kutusunun merkezinin, görüntü genişliğine göre normalleştirilmiş X koordinatını, C sütünü nesnenin dikdörtgen sınır kutusunun merkezinin, görüntü yüksekliğine göre normalleştirilmiş Y koordinatını, D sütünü nesnenin dikdörtgen sınır kutusunun genişliği, görüntü genişliğine göre normalleştirilmiş değerini, E sütünü ise Nesnenin dikdörtgen sınır kutusunun yüksekliği, görüntü yüksekliğine göre normalleştirilmiş değerini göstermektedir. B, C, D ve E sütununun değerleri 0 ile 1 arasında oluşmaktadır.

**Tablo 3.1.** Etiketlenmiş bir fotoğrafın verileri

| <b>A</b> | <b>B</b> | <b>C</b> | <b>D</b> | <b>E</b> |
|----------|----------|----------|----------|----------|
| 0        | 0.690104 | 0.578241 | 0.196875 | 0.306481 |
| 0        | 0.213021 | 0.751389 | 0.157292 | 0.228704 |
| 0        | 0.367708 | 0.697685 | 0.152083 | 0.221296 |
| 0        | 0.739323 | 0.244907 | 0.156771 | 0.200926 |
| 0        | 0.685677 | 0.914352 | 0.197396 | 0.171296 |
| 1        | 0.210938 | 0.731019 | 0.086458 | 0.115741 |
| 1        | 0.330729 | 0.703704 | 0.089583 | 0.127778 |
| 1        | 0.681771 | 0.591667 | 0.078125 | 0.135185 |
| 1        | 0.719010 | 0.218056 | 0.077604 | 0.136111 |
| 1        | 0.663021 | 0.919444 | 0.113542 | 0.135185 |
| 2        | 0.448698 | 0.275926 | 0.053646 | 0.083333 |
| 2        | 0.347917 | 0.292593 | 0.066667 | 0.061111 |
| 2        | 0.352083 | 0.431019 | 0.033333 | 0.113889 |
| 2        | 0.323437 | 0.825000 | 0.050000 | 0.066667 |
| 2        | 0.416406 | 0.852315 | 0.064062 | 0.089815 |
| 2        | 0.519271 | 0.924074 | 0.077083 | 0.066667 |
| 2        | 0.164323 | 0.576389 | 0.080729 | 0.084259 |
| 2        | 0.133333 | 0.418519 | 0.041667 | 0.079630 |
| 2        | 0.232031 | 0.241204 | 0.084896 | 0.106481 |
| 2        | 0.145573 | 0.201852 | 0.068229 | 0.112963 |
| 2        | 0.822396 | 0.214352 | 0.077083 | 0.125000 |
| 2        | 0.890365 | 0.263426 | 0.097396 | 0.084259 |
| 2        | 0.761719 | 0.328704 | 0.074479 | 0.085185 |
| 2        | 0.719010 | 0.436574 | 0.109896 | 0.099074 |
| 2        | 0.861198 | 0.432407 | 0.083854 | 0.133333 |
| 2        | 0.819010 | 0.562963 | 0.051562 | 0.062963 |
| 2        | 0.886719 | 0.575463 | 0.044271 | 0.097222 |
| 2        | 0.947656 | 0.612037 | 0.064062 | 0.092593 |
| 2        | 0.711719 | 0.784259 | 0.054688 | 0.079630 |
| 2        | 0.773438 | 0.795833 | 0.048958 | 0.076852 |
| 2        | 0.889844 | 0.850926 | 0.072396 | 0.101852 |

### 3.3. Veri Seti

Veri seti oluşturmak için Şekil 3.1.'deki alandan çekilmiş görüntülerden elde edilen kareler kullanılmıştır. Şekil 3.3.'de çekilmiş fotoğraflardan örnekler verilmiştir. Elde edilen karede bulunan tüm lahanalar üç farklı sınıfa ayrılacak şekilde etiketlenmiştir. Lahananın genel gövdesini tanımak amacıyla lahana gövdesi, hasarlı bölgelerin ayrı ayrı işaretlenmesi ve lahananın orta bölgesinin etiketlenmesi sağlanmıştır. Görüntüde bulunan lahana sayısının tespit edilebilmesi, hasarlı bölge oranının tespit edilebilmesi için 3 farklı etiketleme yapılmıştır. Lahana merkezi, kenar yapraklarıyla tüm lahana ve hasarlı bölge olarak etiketleme yapılmıştır. Etiketleme işlemi YOLO formatında yapılmıştır. Etiketlenen görüntüler ve etiket bilgilerinin bulunduğu metin dosyaları daha sonra ağın eğitilmesi için kullanılmıştır. Şekil 3.4.'de etiketlenmiş görüntülerin, etiket türlerine göre ve etiketlerin fotoğraflardaki konumuna göre oluşturulmuş grafikler yer almaktadır.

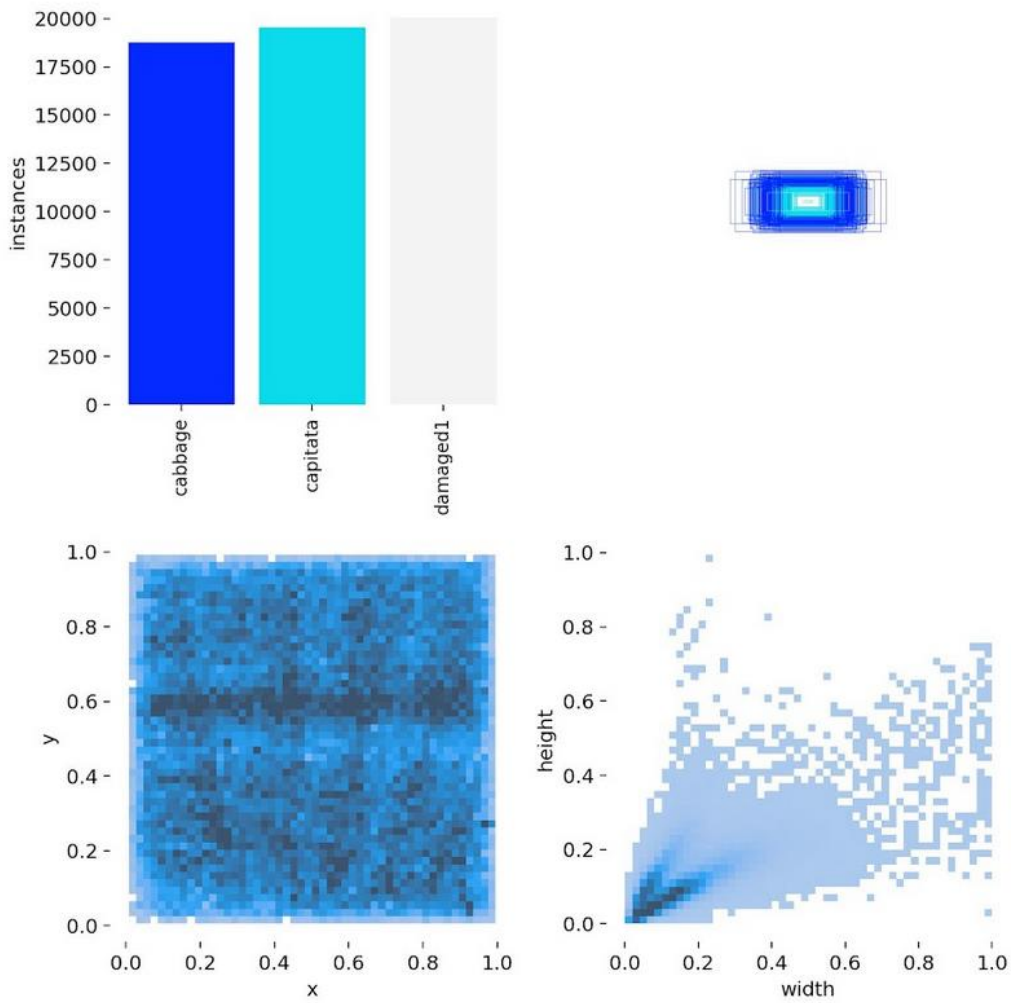


Şekil 3.3. Veri setindeki görüntülerden örnekler

Toplam 3960 farklı görüntü üzerinde 75.000 farklı etiket bölgesi işaretlenmiştir. Böylece hasarlı lahana bölgelerinin tespiti ve görüntüdeki lahana sayısının tespitinin yapılabilmesi amaçlanmıştır.

YOLO formatında etiketleme etiketin sınıf numarası, görüntüdeki konumunun yüzde olarak değeri ve görüntünün yüzdelik olarak boyutları belirtilerek yapılmaktadır. Her bir görüntüdeki tüm lahanalar etiketlenmiştir.

Verilerin %30'u test amacıyla ayrılmıştır. Geriye kalanlar eğitim amacıyla kullanılmıştır. Etiketlenen verilerin sayıları aşağıdaki şekilde sunulmuştur.



Şekil 3.4. Etiketlenen veri sayıları

### 3.4. Lahana Zararlıları

Birleşmiş Milletler Gıda ve Tarım Örgütü (FAO) verilerine göre 2020 yılında Dünya'da 70.862.165 ton lahana ve diğer Brassicaceae bitkilerin üretimi gerçekleştirilmiştir. Türkiye ise bu miktarın %1,20'sini oluşturarak, 2020 yılında 851.648 ton lahana ve diğer Brassicaceae

bitkilerin üretildiğini rapor etmiştir. Ülkemizde lahana, karnabahar ve brokoli üretiminin en fazla yapıldığı iller, Samsun, Niğde, İzmir, Bursa, Antalya, Mersin olarak sıralanmaktadır (TÜİK, 2021). Türkiye'de 2021 yılı toplam üretim beyaz baş lahana 597.910 ton, üretim alanı toplam 132.115 dekadır (Saran, 2022).

Lahana yaprak güvesi, sadece Brassicaceae familyasına ait bitkiler ile beslenmektedir. Larvalar, beslendikleri yapraklar üzerinde zarar oluşturarak dökülmesine ve sonuç olarak lahanada %91,2 oranında verim kaybına neden olmaktadır (Saran, 2022). Şekil 3.5.'de lahana yaprak güvesinin bitkiye vermiş olduğu zararı göstermektedir.

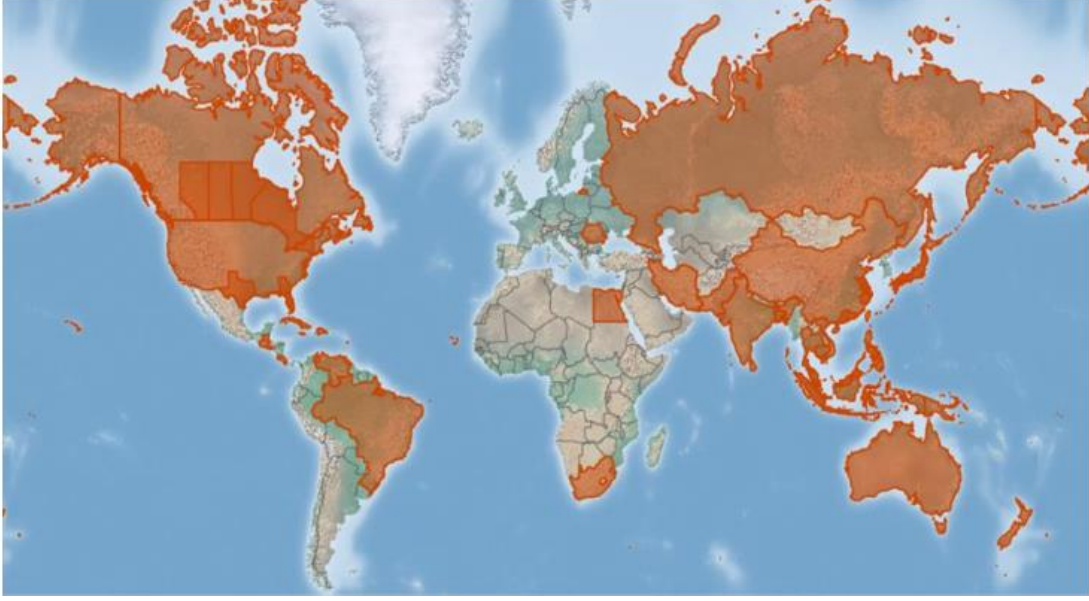
Lahana zararlısının tespiti ve yoğunluğu yapraklardaki delikleri saymak, yaprak üzerindeki larvaları saymak yoluyla yapılmaktadır (Oral, 2022). Bu tez çalışması bu işlemin yapay zeka yardımıyla daha hızlı tespiti hedeflemektedir.



**Şekil 3.5.** Lahana yaprak güvesi zararı bulunan lahana bahçesi (A ve B) ve larvanın yapraktaki görünümü (C)

**Kaynak:** (Saran, 2022)

Lahana zararlısının dünya üzerinde pek çok bölgede (Şekil 3.6.) yoğun olarak yaşadığı görülmektedir (Saran, 2022). Bu zararlıya karşı alınabilecek daha etkin ve zamanında önlemler önemli ekonomik kazançlar sağlayabilir.



**Şekil 3.6.** Lahana yaprak güvesinin Dünya üzerindeki yayılış alanları

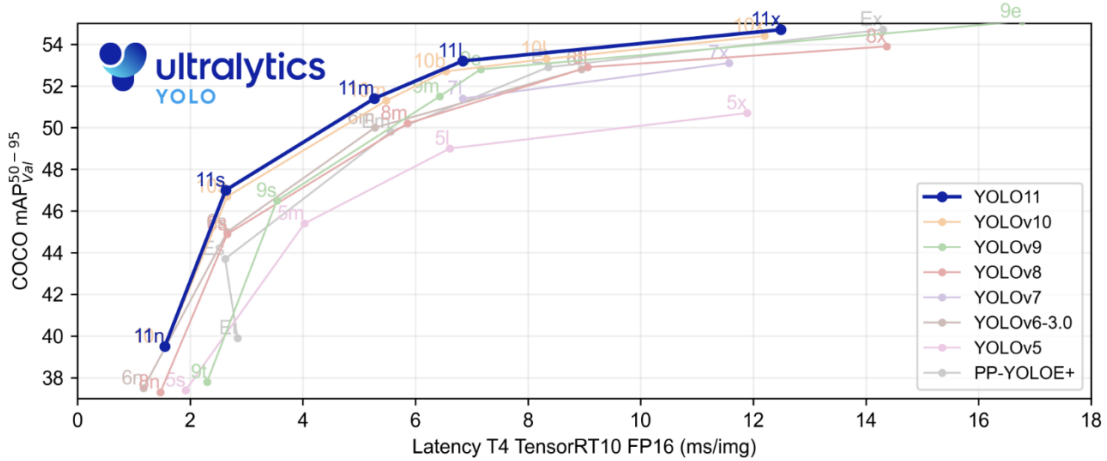
**Kaynak:** (Mason, 2022)

## 4. DENEYSEL ÇALIŞMALAR VE BULGULAR

Bu bölümde zarar görmüş lahanaların tespitini yapan yapay zeka uygulaması anlatılmaktadır. Veri setinin oluşturulması sonrası ağı eğitimi ve eğitim sonuçlarının değerlendirilmesi sunulmaktadır.

### 4.1. YOLO Ağı ile Lahanaların Sınıflandırılması

YOLO ağı güncel olarak geliştirilmeye devam eden tek aşamalı bir yapay sinir ağıdır. Tek aşamalı ağı olması sebebiyle görüntü sınıflandırma işlemlerinin hızlı şekilde yapılabilmesini sağlar. Güncel olması, görüntüler üzerinde hızlı çalışması sebebiyle YOLO modeli kullanılmıştır. YOLO algoritması her ortaya çıkan sürümde iyileştirmeler yapılarak ilerlemiştir. YOLO algoritmalarının farklı sürümleri arasındaki performans ve parametre farklılıkları Şekil 4.1.'de görülmektedir. YOLOV9 %10 daha az parametre ile %5-15 daha az hesaplama yükü sağlamaktadır. Buna rağmen ağı başarımı %0.4 ile %0.6 arasında artış göstermiştir. YOLOV9 kendinden önceki modellerle karşılaştırıldığında image-net üzerindeki görüntülerde daha başarılı sonuçlar vermiştir. YOLO v11, hem ortalama doğruluk (mAP) değerlerinde hem de hız açısından önemli gelişmeler göstermektedir. Modelin parametre sayısındaki azalmaya rağmen, performansının artmış olması dikkat çekicidir (Khanam & Hussain, 2024; C.-Y. Wang et al., 2024).



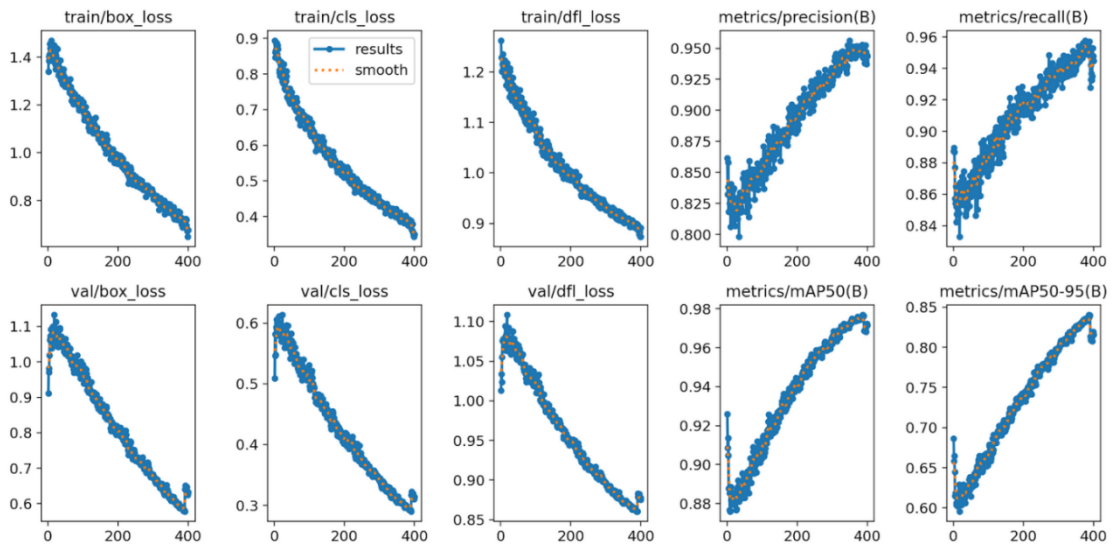
Şekil 4.1. YOLO algoritması sürüm farklılıkları

**Kaynak:** (Khanam ve Hussain, 2024)

YOLO algoritması farklı parametre sayılarına sahip birkaç farklı öğrenme modelinin eğitilmesine imkan tanır. Bu çalışma kapsamında YOLOV11-x modeli kullanılarak ağı eğitilmiştir. Parametre sayısı olarak diğer YOLO modellerine göre daha az sayıda parametre ile

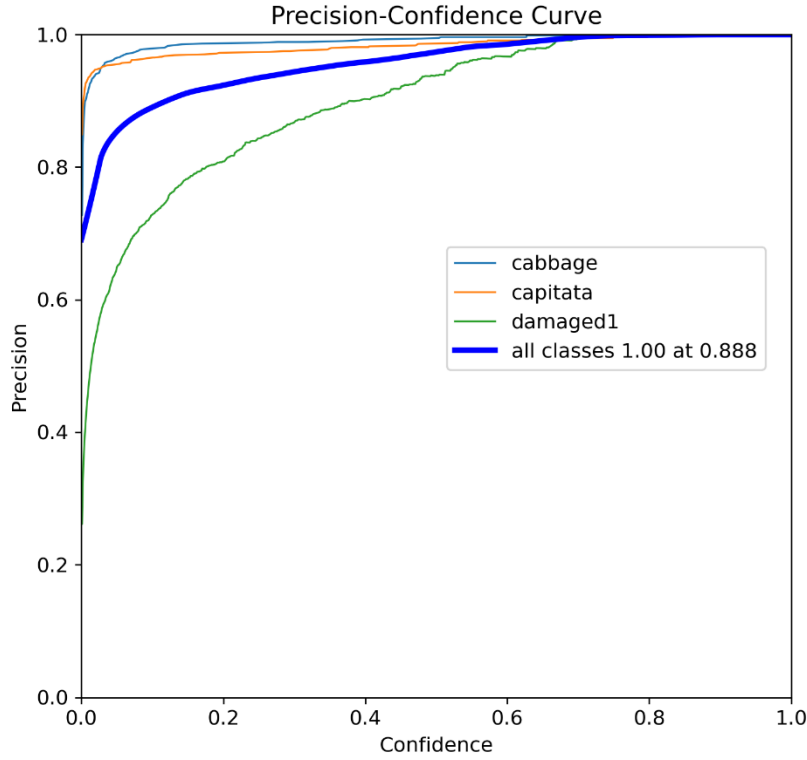
ağ eğitildiğinden daha küçük donanım yapılarında da çalıştırılması kolay olacaktır. Yapılan etiketleme işlemlerinde de yeterli seviyede sınıflandırma başarımı gösterdiği için tercih edilmiştir. YOLOv11-x resmi reposu kullanılmıştır. Eğitim parametreleri olarak batch size 4 seçilmiş, epoch 500, görüntü boyutu 640 x 640 olarak seçilmiştir. Ağa uygulanan görüntüler 640 x 640 olarak biçimlendirilerek uygulanmaktadır. Ağ tahmin yaparken conf skoru 0.5 olarak belirlenmiştir. Güven eşiği ağın başarım durumuna göre tahmin sırasında uygun oranda belirlenebilir. Varsayılan olarak 0.5 olarak alınır. Ağ eğitildiğinde güven eşiği değiştirilse bile ağ başarımının uzun bir aralıkta değişmediği gözlenmiştir. Ağ optimizasyon fonksiyonu olarak Adam optimizer kullanılmıştır. YOLO11x 631 katmandan oluşur ve 56.877.225 parametreye sahiptir.

Ağ eğitimi süresince oluşan kayıp ve başarıım grafikleri Şekil 4.2.'de gösterilmektedir. Eğitim grafikleri incelendiğinde yaklaşık 500 devir sonrasında eğitim verisi üzerinde kayıp azalmaya devam ederken doğrulama veri kümesinde ise kayıp değerinin artmaya başladığı gözlenmiştir. Bu durum ağın bu devirden sonra ezberlediği olarak yorumlanabilir. Eğitim süresi boyunca en yüksek başarımın olduğu ağırlık değerleri kayıt edilerek hasar tespiti işlemlerinde bu model tahmin modeli olarak kullanılmıştır. Precision, recall grafiklerinin de güven eşiği değişimine bağlı olarak uzun süre sabit kaldıkları gözlenmiştir. Özellikle küçük olan hasarlı bölgelerin etiketlenmesi ve bu küçük alanların tanınması için ağın daha uzun süre eğitilmesi gerekmiştir. Büyük boyutlu olanak sınıflara göre daha küçük yer kaplayan hasarlı bölge etiketlerinin sınıflar arasında en düşük skorla tanınan bölgeler olduğu gözlenmiştir.



Şekil 4.2. Ağ eğitimi süresi boyunca doğrulama ve başarıım metrikleri değişimi

Precision skorunun eğitim süresince değişimi Şekil 4.3.'de gösterilmektedir.



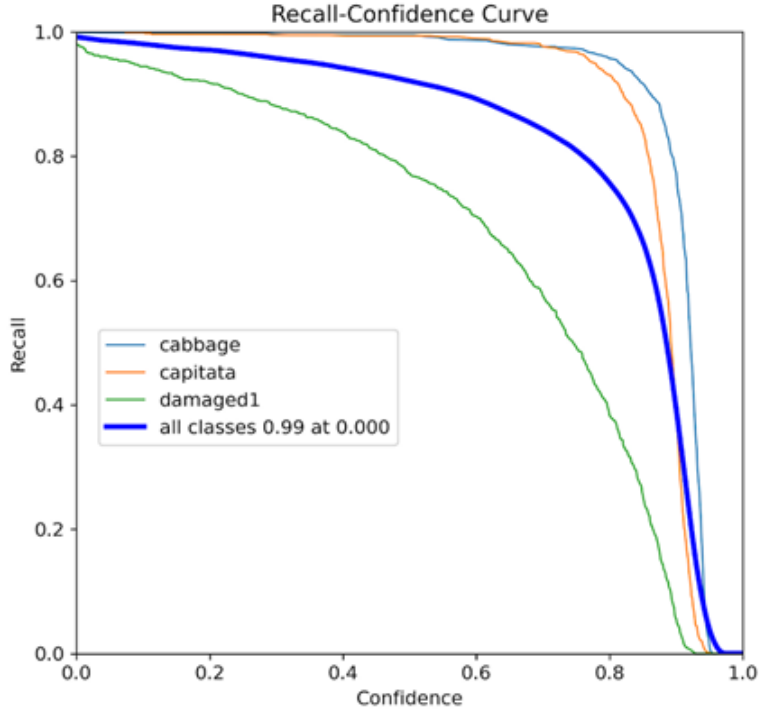
Şekil 4.3. Precision ve güven eşiği değişimi

Kesinlik skorunun formülü Denklem 4.1'de verilmektedir. Kesinlik değeri ağın sınıfları ayırt edebilirliğini göstere yaygın kullanılan bir skordur. Eğitim süresince bu değer ideale yaklaştığı gözlenmektedir.

$$Kesinlik(Precision) = \frac{Doğru\_Pozitif(TP)}{Doğru\_Pozitif(TP)+Yanlış\_Pozitif(FP)} \quad (4.1)$$

Duyarlılık (Recall), pozitif olarak tahmin edilmesi gereken etiketlerin ne kadarının pozitif olarak tahmin edildiği göstere bir metriktir. Formülü Denklem 4.2'de verilmektedir. Duyarlılık değeri false negatif olan tahminlerin maliyetinin yüksek olduğu durumlar için bize bir fikir verir. Duyarlılık değeri de güven eşiği değeri ile çok fazla etkilenmeden uzun süre aynı seviyelerde kalmaktadır. Duyarlılık ve güven eşiği değişimi Şekil 4.4.'de gösterilmektedir. Güven eşiğine bağlı olarak ağın başarımının çok fazla etkilenmemiş kararlı bir ağın varlığını gösteren önemli bir parametredir.

$$Duyarlılık(Recall) = \frac{TPDoğru\_Pozitif(TP)}{Doğru\_Pozitif(TP)+Yanlış\_Negatif(FN)} \quad (4.2)$$

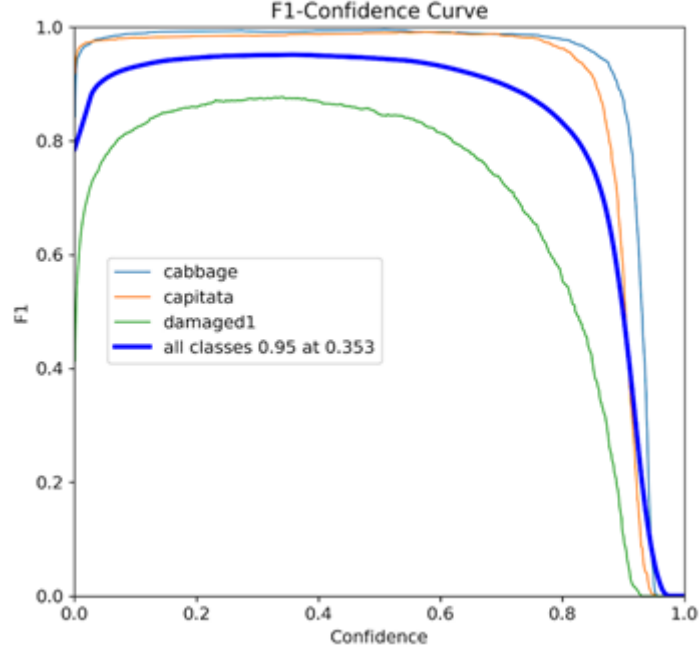


**Şekil 4.4.** Duyarlılık ve güven eşiği değişimi

F1 skoru değişimi incelendiğinde ağ eğitimi sonucunda yüksek başarımlı gösterdiği gözlemlenebilir. F1 skoru bize kesinlik ve duyarlılık skorunun harmonik ortalamasını veren bir skordur. Uç durumları göz ardı edebilmek için harmonik ortalama kullanılarak hesaplanır. F1 skorunun hesaplama formülü Denklem 4.3’de gösterilmektedir.

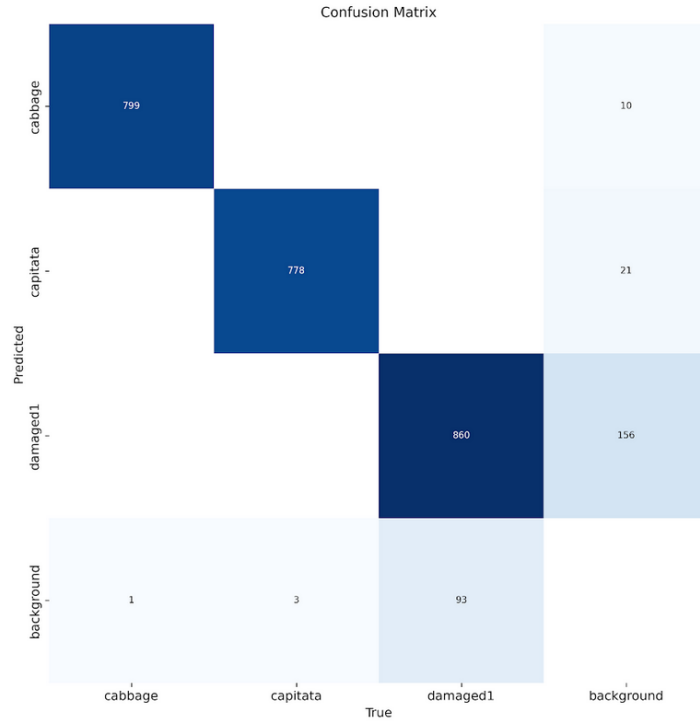
$$F1 = \frac{Kesinlik(Precision) * Duyarlılık(Recall)}{Kesinlik(Precision) + Duyarlılık(Recall)} \quad (4.3)$$

Özellikle eşit dağılmayan verilerin olduğu veri kümelerinde sınıflandırma başarımlarını doğru şekilde değerlendirmek için başarımlar yerine F1 skoru kullanılması tercih edilir. Yaklaşık 0.1 ve 0.8 güven eşiği arasındaki değerlerde F1 skorunun aynı seviyelerde kaldığı gözlemlenmiştir (Şekil 4.5.). Bütün sınıflar için ortalama F1 skoru değeri 0.95 olarak gözlemlenmiştir.



**Şekil 4.5.** F1 skoru ve güven eşiği değişimi

Test verileri üzerinde oluşan karışıklık matrisi Şekil 4.6.'da verilmiştir. Karışıklık matrisinden görüldüğü üzere ağ sınıfları yüksek başarıyla tanımaktadır. Hasarlı bölgeleri sınıflandırırken düşük oranda da olsa hatalı sınıflandırdığı bölgeler olabilmektedir. Etiketli verilerin bazılarında çok küçük hasarlı bölgeler etiketle belirtilmemesine rağmen ağ bu bölgeleri de tanıyabilmektedir. Bu durum karışıklık matrisinde hatalı sınıflandırma gibi görülse de hatalı bir durum değildir. Buna rağmen karışıklık matrisi ağın başarımının iyi durumda olduğunu göstermektedir.

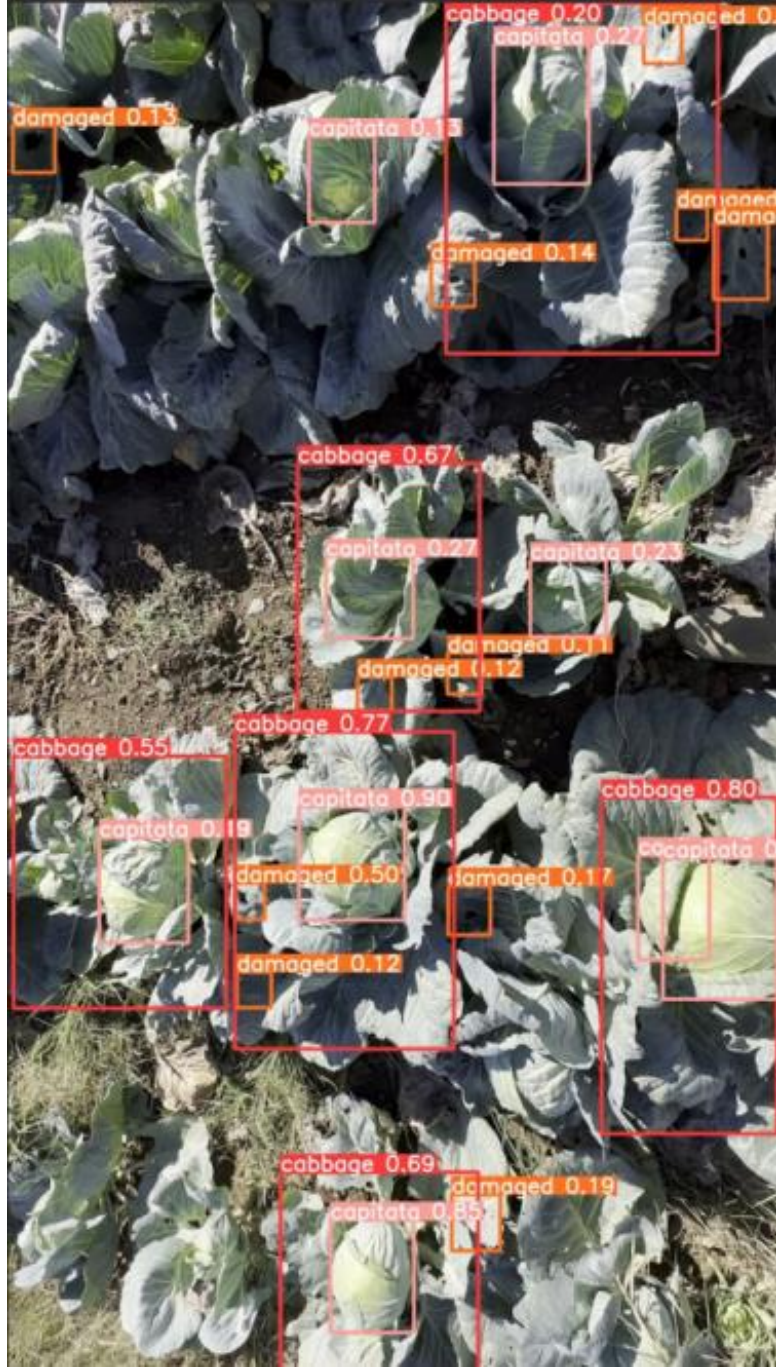


Şekil 4.6. Karışıklık matrisi

Ağ eğitildiğinde test verileri üzerinde yapılan örnek bir sınıflandırma Şekil 4.7.'de ve Şekil 4.8'de görülmektedir.



Şekil 4.7. Doğrulama veri seti üzerinde yapılan sınıflandırmalar



Şekil 4.8. Etiketlenmiş lahana görseli.

#### 4.2. Hasarlı Bölgenin Hesaplanması

Eğitim sonucunda doğrulama veri seti üzerinde yapılan sınıflandırmalar incelendiğinde büyük oranda başarılı olarak etiketleme yapıldığı görülmektedir (Şekil 4.9.). Örnek görüntüler üzerinde model çalıştırıldığında ortaya çıkan sonuç incelendiğinde başarılı sonuçlar oluştuğu gözlenmiştir. Çok küçük boyutlu hasarlı bölgeleri de model ayırt edebilmiştir. Çok az sayıda hasarlı olmayan gölgelerinde hatalı sınıflandırıldığı gözlenmiştir. Modelin tespit ettiği hasarlı bölge oranı görüntünün üzerinde yüzdelik olarak ne kadar yer kapladığına bakılarak lahana

bahçesinde zararlı için ilaçlama ihtiyacı olup olmadığı tahmin edilebilir. Bu amaçla görüntüde modelin tespit ettiği hasarlı bölge alanlarının yüzdesi hesaplanır. Görüntü üzerindeki hasarlı bölge yüzdesi %20'nin üzerinde ise zararlı böcek için ilaçlama ihtiyacı olduğu kabul edilmiştir. YOLO modelinin verdiği sonuçlardaki hasarlı bölge etiketine sahip alanların yüzdesi hesaplanarak bu işlem yapılmaktadır. Hasarlı bölge olarak işaretlenmiş kutuların hepsinin toplam alanı toplanarak hesaba dahil edilmektedir.



**Şekil 4.9.** Test verisi üzerinde sınıflandırmalar

Modelin sınıflandırma başarımı hasarlı bölge oranını hesaplamak için yeterli seviyededir. Az da olsa hatalı sınıflandırma örnekleri olsa da bu durum görseldeki hasar oranını belirlenmesini büyük oranda etkilememektedir.

Tespit edilen etiketli bölgeler kullanılarak görüntüdeki hasar oranı Denklem 4.4'deki formülle hesaplanmaktadır. Hesaplama yapılırken tespit edilen lahana kutularının alanı ve hasarlı bölge kutularının toplam alanı piksel olarak hesaplanmaktadır. Hasarlı bölge yüzdesi Denklem 4.5'de verilen formülle hesaplanmaktadır. Alan yüzdesi tüm fotoğraf yerine görüntüde tanımlanmış lahana kutularına göre oranlanmaktadır.

$$ToplamLahanaAlanı = \sum w * h \quad (4.4)$$

W:Tespit edilen lahana kutusu genişliği

H:Tespit edilen lahana kutusu yüksekliği

$$Hasar Yüzdesi \% = \frac{Toplam Lahana Alanı}{Hasarlı Bölgelerin Alanı} \quad (4.5)$$

Hasarlı kutular ve normal lahana kutularının alanı piksek olarak hesaplanmıştır. Modelin tespit ettiği kutuların en boy değerleri kullanılarak hesaplama yapılmıştır. Isı haritası görselleştirme işlemi amacıyla hesaplanan hasarlı kutu koordinatları npy dosyası olarak kayıt edilmektedir.

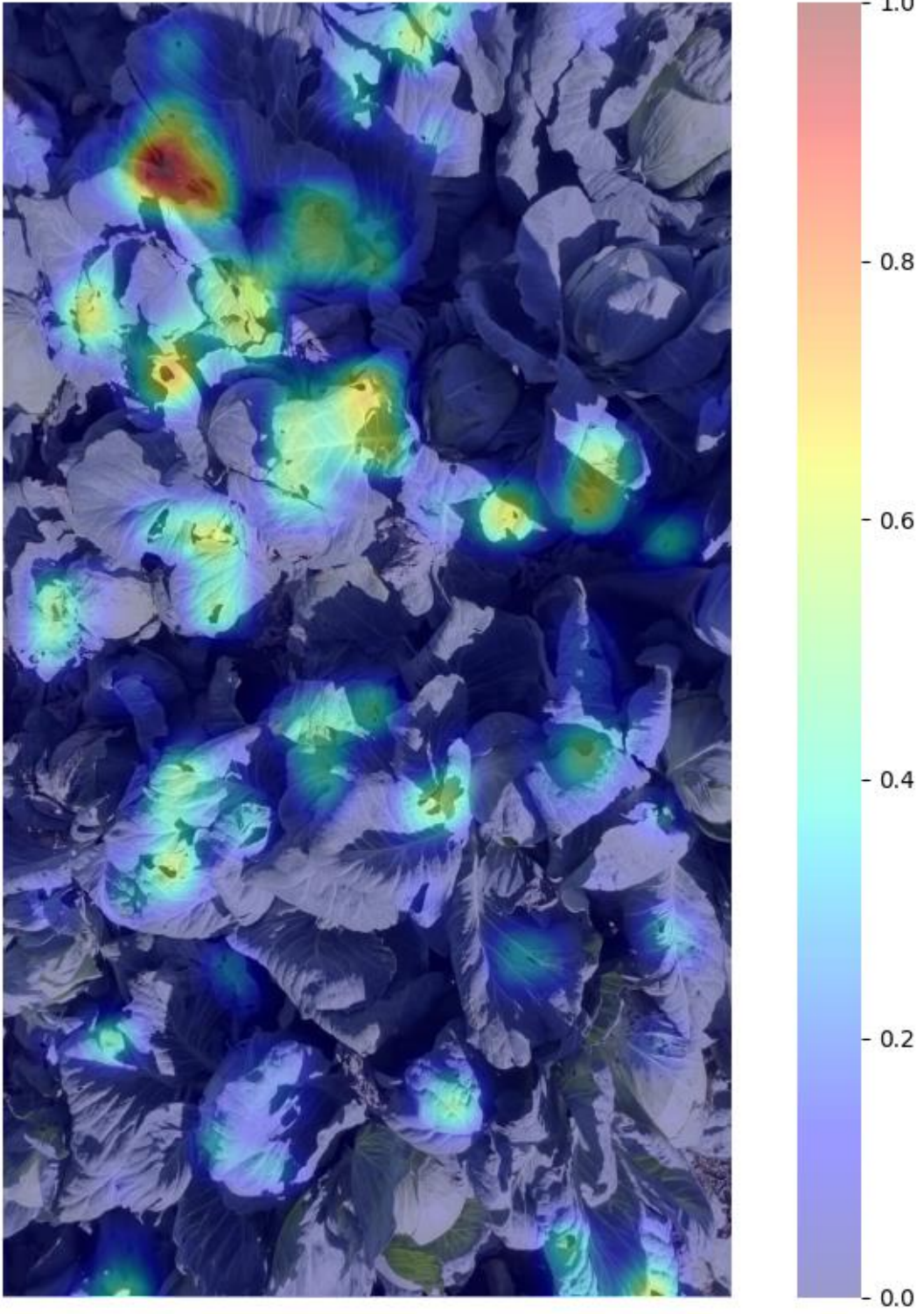
YOLO modeli kullanılarak yapılan tahmin verisindeki hasarlı bölge koordinatları kullanılarak hasarlı bölgelerin ısı haritası şeklinde görüntülenmesi sağlanmıştır. Conf değerinin değiştirilmesi ile model farklı sayılarda hasarlı bölge tespiti yapabilir. Ancak eğitilen model çok geniş bir eşik aralığında kararlı sonuçlar vermektedir. Muhtemel hasarlı bölgelerin gözden kaçırılmaması için düşük Conf değeri tercih edilebilir. Şekil 4.10. ve Şekil 4.11.'de test grubuna ait aynı görsele conf değeri farklı alınarak oluşturulmuştur. Şekil 4.10.'da Conf değeri 0.01 ile 148 tane hasarlı bölge tespit edilmiş olup bu sayı %26,9 oranına karşılık gelmektedir. Şekil 4.11.'de Conf değeri 0.05 ile 44 tane hasarlı bölge tespit edilmiş olup bu sayı %11,2 oranına karşılık gelmektedir. Şekil 4.12. ve Şekil 4.13. de ise aynı görselin hasarlı bölgeleri ısı haritası ile gösterilmektedir.



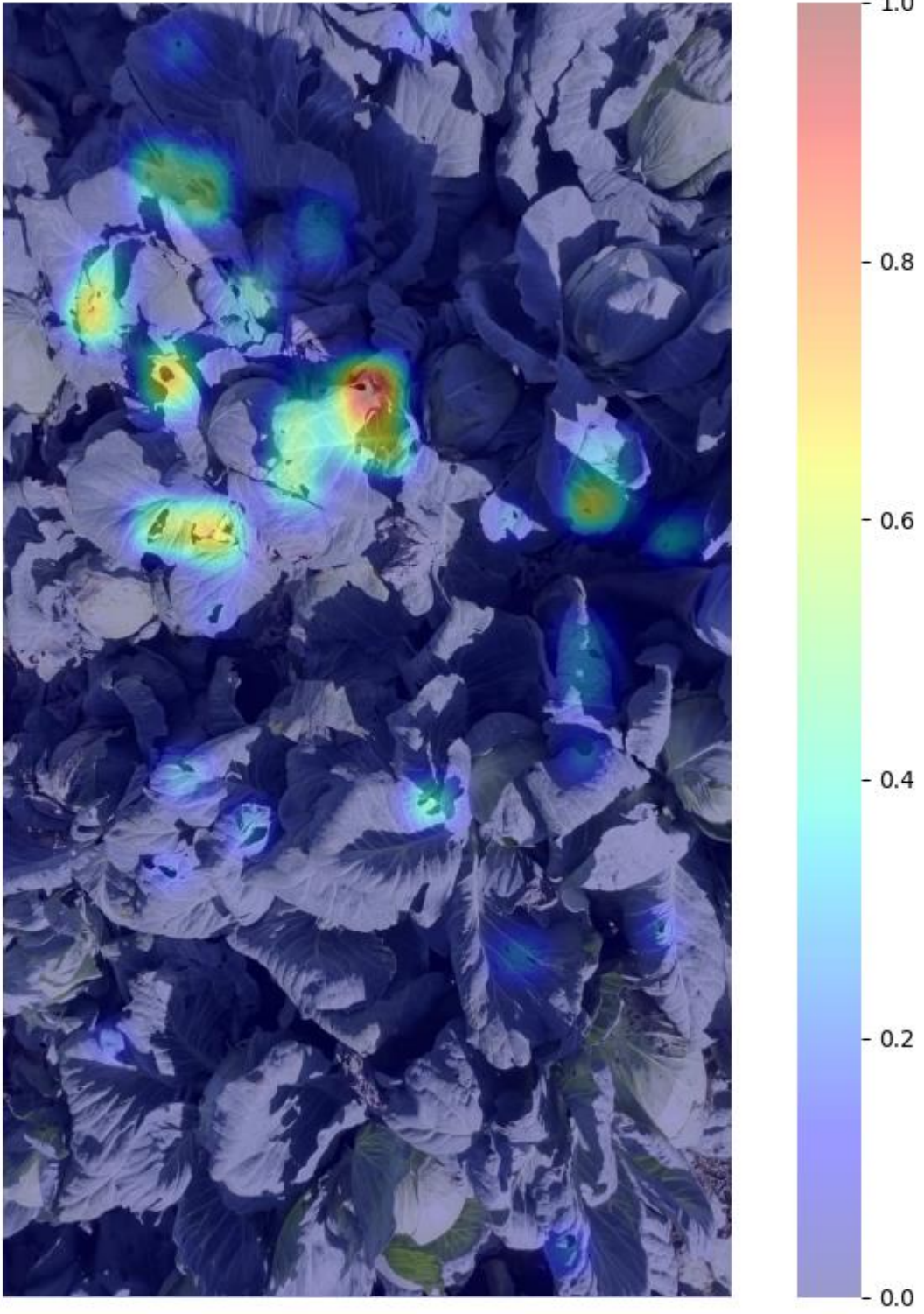
Şekil 4.10. %26,9 Hasar oranlı test resmi etiketleri



Şekil 4.11. %11,2 Hasar oranlı test resmi etiketleri



Şekil 4.12. %26,9 Hasar oranlı bir görüntünün Conf değeri 0,01 ile hasırlı bölge gösterimi



Şekil 4.13. %11,2 Hasar oranlı bir görüntünün Conf değeri 0,05 ile hasırlı bölge gösterimi

Eđitim ve test ařamalarında elde edilen sonuları incelendiđinde, modelin hasarlı blgeleri tespit etme konusunda olduka yksek bir hassasiyete sahip olduđu gzlemlenmiřtir. Modelin performansını deđerlendirirken kullanılan bařlıca metrikler řunlardır:

Dođruluk (Accuracy); modelin genel bařarı oranını gstermektedir. Test verileri zerinde yapılan analizlerde modelin dođruluk oranının olduka yksek olduđu tespit edilmiřtir.

Kesinlik (Precision); Modelin tespit ettiđi hasarlı blgelerin ne kadarının gerekten dođru olduđunu gstermektedir. Hassasiyetin yksek olması, modelin yanlış pozitif oranının dřk olduđunu gstermektedir.

Duyarlılık (Recall); Modelin var olan tm hasarlı blgeleri ne kadar dođru tespit ettiđini ifade etmektedir. Yapılan testlerde, modelin kk lekli hasarlı blgeleri de bařarılı bir řekilde algıladıđı belirlenmiřtir.

F1-Skoru; kesinlik ve duyarlılık deđerlerinin dengeli bir ltdr ve modelin genel performansını belirlemek iin kullanılır. Modelin F1-skora gre dengeli ve bařarılı sonular verdiđi grlmřtr.

Elde edilen veriler ıřıđında, modelin farklı ıřık kořullarında ve eřitli aılardan ekilmiş grntlerde dahi bařarılı sonular verdiđi belirlenmiřtir. Ancak, modelin zaman zaman etiketleme srecinde gzden kaan ok kk hasarlı blgeleri de hasarlı olarak tanımladıđı grlmřtr. Bu durum, modelin fazla duyarlı olmasından kaynaklanmaktadır.

**Tablo 4.1.** Ađın eđitiminde kullanılan model parametreleri

| <b>Model Parametreleri</b> |            |
|----------------------------|------------|
| rnek Sayısı               | 3960       |
| Etiketleme Sayısı          | 75000      |
| Batch Size                 | 4          |
| Devir(Epoch) Size          | 500        |
| Grnt Boyutu             | 640x640    |
| Optimizer                  | Adam       |
| Katman Sayısı              | 631        |
| Parametre Sayısı           | 56.877.241 |

**Tablo 4.2.** Eğitilen modelin başarıml değerleri

| <b>Model Başarıml Parametreleri</b> |        |
|-------------------------------------|--------|
| Ortalama Hassasiyet (mAP)           | 98.00% |
| Çerçeve Hızı (FPS)                  | 310    |
| Kesinlik (Precision)                | 0.97   |
| Duyarlılık (Recall)                 | 0.95   |
| F1 Skoru                            | 0.95   |
| Tahmin Güven Skoru (Confidence)     | 0.5    |

Tablo 4.1.'de Yolo v11 ađında eğitilen modelin, eğitimde kullanılan parametreleri gösterilmiştir. Tablo 4.2.'de ise bu modelin eğitiminden elde edilen sonuçlar gösterilmektedir.

## 5. SONUÇLAR VE ÖNERİLER

Bu tez çalışmasında, derin öğrenme tabanlı güncel YOLO modeli kullanılarak beyaz lahana yapraklarında zararlı kaynaklı hasarın tespiti gerçekleştirilmiştir. Çalışmanın temel amacı, yapay zeka destekli görüntü işleme yöntemleriyle bitki yapraklarında oluşan hasarların erken tespit edilmesini sağlayarak tarımsal verimliliği artırmak ve çiftçilerin müdahale süreçlerini optimize etmektir. Yapılan çalışmada ağır başlangıçta etiketlenmemiş olan çok küçük hasarlı bölgeleri de tespit edebildiği gözlenmiştir. Veri sayısının artırılmış olması ağır performansını arttırmaktadır. Karışıklık matrisi incelendiğinde etiketlerde arka plan olarak herhangi bir etiketleme yapılmayan bölgelerin hasarlı bölge olarak işaretlendiği gözlenmektedir. Bu durum incelendiğinde etiket yapılırken çok küçük hasarlı alanların ihmal edilmesi sebebiyle modelin bu ihmal edilen alanları da tanıyarak etiketlenmesinden kaynaklandığı gözlenmiştir. Bu durum modelin performansının çok küçük hasarlı bölgeleri algılamada da başarılı olduğunu göstermektedir. Bahçenin tüm görüntüsü yukarıdan yüksek çözünürlüklü olarak elde edildiğinde lahana rekoltesi tahmin edilebilir, bahçedeki hasar oranı tahmin edilebilir. Lahana zararlısı ile mücadelede bu veri çiftçiye kolaylık sağlayabilir. Çok büyük boyutlu tarlalarda hasarın yoğun olduğu bölgeler tespit edilerek ilaçlamanın tüm bahçe yerine sadece hasar oluşan bölgeye yapılması sağlanabilir.

Gerçekleştirilen testler ve değerlendirmeler sonucunda, modelin yüksek doğruluk oranına sahip olduğu, hasarlı bölgeleri belirleme konusunda başarılı sonuçlar verdiği tespit edilmiştir. Özellikle modelin, insan gözüyle fark edilmesi zor olan küçük hasarlı bölgeleri de tanımlayabilmesi, yapay zekanın tarım sektöründeki kullanım potansiyelini gözler önüne sermektedir.

Bu çalışma, geleneksel yöntemlerle yapılan hasar tespit süreçlerine kıyasla avantaj sağlamaktadır. Geleneksel olarak, bitkilerdeki hastalıklar ve zararlılar görsel muayene ile tespit edilmektedir. Ancak bu yöntem, büyük ölçekli tarım arazilerinde zaman alıcı ve hataya açık bir süreçtir. YOLO modelinin kullanımı, bu süreci otomatikleştirerek zaman ve maliyet tasarrufu sağlayabilir.

Tarım uygulamaları kapsamında; modelin sağladığı yüksek doğruluk sayesinde, çiftçiler tarladaki hasar yoğunluğunu hızlıca belirleyebilir. Hasarın yoğun olduğu bölgelere yönelik noktasal ilaçlama yapılabilir, böylece gereksiz kimyasal kullanımının önüne geçilebilir. Tarımsal verimliliğin artırılmasına katkıda bulunarak çiftçilerin ekonomik kayıplarını en aza indirebilir. Konumsal çalışmalar geniş bölgelere yayıldığında zararlıların yoğun olduğu bölgeler

tespit edilebilir ve önlem alınabilir. Tespit edilen hasar bölgeleri haritalandırılarak bölgesel ilaçlama programları oluşturulabilir. Lahana yapraklarında meydana gelen hasar oranı, ilerleyen dönemlerde rekolte tahmini yapmak için kullanılabilir. Büyük ölçekli tarım işletmelerinde ürün kayıpları önceden belirlenerek gerekli önlemler alınabilir. Çevre dostu tarım uygulamaları teşvik edilerek gereksiz kimyasal kullanımının önüne geçilebilir. Model, drone veya robotik sistemlerle entegre edilerek geniş tarım alanlarının düzenli olarak analiz edilmesini sağlayabilir. Gerçek zamanlı görüntü işleme ile tarım arazilerinin sağlık durumu belirli aralıklarla takip edilebilir.

Bu çalışmada elde edilen sonuçlar oldukça başarılı olmakla birlikte, modelin bazı sınırlamaları ve iyileştirme gerektiren noktaları bulunmaktadır;

Veri seti çeşitliliği; model, belirli bir bölgeden alınan verilerle eğitildiği için farklı iklim ve toprak koşullarında test edilememiştir. Çeşitli bölgelerden alınan daha geniş veri setleri ile modelin genelleme yeteneği artırılabilir. Farklı büyüme dönemlerinde alınacak görüntülerle veri seti genişletilebilir.

Zararlı çeşitliliği; model, belirli bir zararlı türü üzerine eğitildiğinden, farklı böcek türleri veya mantar hastalıkları gibi diğer risk faktörleri için yeniden eğitilmesi gerekmektedir.

Gerçek zamanlı kullanım; modelin gerçek zamanlı drone veya robot sistemlerine entegre edilerek geniş ölçekli analizler yapması için optimizasyon süreçlerine ihtiyaç duyulabilir.

Bu çalışma kapsamında tarım alanında daha geniş bir kullanım alanına sahip olabilmesi için araştırmacılara bazı öneriler sunulmaktadır;

Farklı coğrafi bölgelerden, farklı ışık koşullarında çekilmiş ve farklı zararlıları içeren görüntülerle modelin eğitilmesi gerekmektedir. Model, farklı hasar tiplerini tespit edebilmesi için genişletilmiş bir veri kümesiyle tekrar eğitilmelidir. Model, drone ve tarımsal robot sistemleriyle entegre edilerek gerçek zamanlı analiz yapabilecek hale getirilebilir. Hasarlı bölgelerin otomatik haritalandırılması için konumsal veri işleme sistemleri ile birlikte kullanılabilir. Bu şekilde zararlıın yoğunlaştığı bölgeler tespit edilebilir. YOLO modeli dışında Faster R-CNN, EfficientDet veya Transformers tabanlı görsel tanıma modelleri ile karşılaştırmalı analizler yapılabilir. Modelin hız ve doğruluk oranlarının farklı yöntemlerle kıyaslanarak en verimli algoritma belirlenebilir. Modelin hasar tespiti ile birlikte zaman içinde hastalıkların nasıl yayıldığını analiz edebilecek tahmin modelleri geliştirilebilir.

## KAYNAKÇA

- Adalı, E.** (2012). Doğal dil işleme. *Türkiye Bilişim Vakfı Bilgisayar Bilimleri ve Mühendisliği Dergisi*, 5(2).
- Ahmad, R., Wazirali, R., & Abu-Ain, T.** (2022). Machine learning for wireless sensor networks security: An overview of challenges and issues. *Sensors*, 22(13), 4730.
- Arı, A., & Berberler, M. E.** (2017). Yapay sinir ağları ile tahmin ve sınıflandırma problemlerinin çözümü için arayüz tasarımı. *Acta Infologica*, 1(2), 55-73.
- Atalay, M., & Çelik, E.** (2017). Büyük veri analizinde yapay zekâ ve makine öğrenmesi uygulamaları-artificial intelligence and machine learning applications in big data analysis. *Mehmet Akif Ersoy Üniversitesi Sosyal Bilimler Enstitüsü Dergisi*, 9(22), 155-172.
- Bozcu, H.** (2024). *Derin Öğrenme Tabanlı Kiraz Yapraklarında Hasar Tespiti*. Bilecik Şeyh Edebali Üniversitesi, Lisansüstü Eğitim Enstitüsü, Bilecik.
- Bülgen, A.** (2024). *Görüntü işleme ve derin öğrenme tabanlı bitkisel tarım ürünlerinde hastalık tespiti*. Erzincan Binali Yıldırım Üniversitesi, Fen Bilimleri Enstitüsü, Erzincan.
- Diri, B.** (2012). Makine Öğrenmesine Giriş (Machine Learning–ML).
- Doğan, F., & Türkoğlu, İ.** (2019). Derin öğrenme modelleri ve uygulama alanlarına ilişkin bir derleme. *Dicle Üniversitesi Mühendislik Fakültesi Mühendislik Dergisi*, 10(2), 409-445.
- Duman, B.** (2022). Derin öğrenme yöntemleri ile bitki sınıflandırma.
- Efe, A.** (2021). Yenilikçi Endüstri 4.0 paradigması kapsamında kurumsal kaynak planlaması ve yönetim bilişim sistemlerinde yapay zekâ. *Pamukkale Üniversitesi İşletme Araştırmaları Dergisi*, 8(1), 186-214.
- Erdoğan, P., Çolak, B., & Durdağ, Z.** (2016). K-Means algoritması ile otomatik kümeleme. *El-Cezeri*, 3(2).
- Gao, X., Wang, G., Qi, J., Wang, Q., Xiang, M., Song, K., & Zhou, Z.** (2024). Improved YOLO v7 for Sustainable Agriculture Significantly Improves Precision Rate for Chinese Cabbage (*Brassica pekinensis* Rupr.) Seedling Belt (CCSB) Detection. *Sustainability*, 16(11), 4759.
- González García, C., Núñez Valdéz, E. R., García Díaz, V., Pelayo García-Bustelo, B. C., & Cueva Lovelle, J. M.** (2019). A review of artificial intelligence in the internet of things. *International Journal Of Interactive Multimedia And Artificial Intelligence*, 5.

**Gökalp, Ö. M.** (2022). Makine öğrenmesi. *Gazi Üniversitesi, Gazi BilişimEnstitüsü, Adli Bilişim Bölümü.*

**Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., Liu, T., Wang, X., Wang, G., & Cai, J.** (2018). Recent advances in convolutional neural networks. *Pattern recognition*, 77, 354-377.

**Guo, Y., Liu, Y., Oerlemans, A., Lao, S., Wu, S., & Lew, M. S.** (2016). Deep learning for visual understanding: A review. *Neurocomputing*, 187, 27-48.

**Gülmez, B.** (2023). Parçacık sürü optimizasyonu destekli derin öğrenme ile gül yaprağı hastalık tespiti. *International Conference on Modern and Advanced Research*, 348-352. doi:10.59287/icmar.1309

**Hamad, Y. N. H.** (2022). *Detection of plant leaf diseases via deep learning.* Altınbaş Üniversitesi, Lisansüstü Eğitim Enstitüsü, İstanbul.

**Jocher, G.** (2024). Ultralytics YOLO11. [Erişim:15.02.2025 <https://docs.ultralytics.com/tr/models/yolo11/> ]

**Khanam, R., & Hussain, M.** (2024). Yolov11: An overview of the key architectural enhancements. *arXiv preprint arXiv:2410.17725.*

**Kızrak, M. A., & Bolat, B.** (2018). Derin öğrenme ile kalabalık analizi üzerine detaylı bir araştırma. *Bilişim Teknolojileri Dergisi*, 11(3), 263-286.

**Li, Z., Liu, F., Yang, W., Peng, S., & Zhou, J.** (2021). A survey of convolutional neural networks: analysis, applications, and prospects. *IEEE transactions on neural networks and learning systems*, 33(12), 6999-7019.

**Mamat, N., Othman, M. F., Abdoulghafor, R., Belhaouari, S. B., Mamat, N., & Mohd Hussein, S. F.** (2022). Advanced technology in agriculture industry by implementing image annotation technique and deep learning approach: A review. *Agriculture*, 12(7), 1033.

**Mason, P.** (2022). *Plutella xylostella* (elmas sırtlı güve). [Erişim:16.02.2025 <https://www.cabidigitallibrary.org/doi/full/10.1079/cabicompndium.42318> ]

**Oral, B.** (2022). *Çanakkale ili kışlık sebze alanlarında lahana yaprak güvesi (Plutella xylostella L.)(Lepidoptera: Plutellidae)'nin popülasyon gelişimi ve zarar durumunun belirlenmesi.* Çanakkale Onsekiz Mart Üniversitesi, Lisansüstü Eğitim Enstitüsü,

**Ök, B.** (2022). *Bitki yapraklarındaki hastalıkları yapay zeka ile tespit edebilen ve noktasal ilaçlama yapabilen otonom mobil manipülatör sisteminin geliştirilmesi*. Karabük Üniversitesi, Lisansüstü Eğitim Enstitüsü, Karabük.

**Özbay, E.** (2022). Use Of Deep Learning Models In Different Application Areas. In *International Research in Engineering Sciences*: Eğitim Yayınevi.

**Pirim, A. G. H.** (2006). Yapay zeka. *Yaşar Üniversitesi E-Dergisi*, 1(1), 81-93.

**Pouyanfar, S., Sadiq, S., Yan, Y., Tian, H., Tao, Y., Reyes, M. P., Shyu, M.-L., Chen, S.-C., & Iyengar, S. S.** (2018). A survey on deep learning: Algorithms, techniques, and applications. *ACM computing surveys (CSUR)*, 51(5), 1-36.

**Redmon, J., Divvala, S., Girshick, R., & Farhadi, A.** (2016). *You only look once: Unified, real-time object detection*. Paper presented at the Proceedings of the IEEE conference on computer vision and pattern recognition.

**Saba Raouf, S., & Durai, M. S.** (2022). A comprehensive review on smart health care: applications, paradigms, and challenges with case studies. *Contrast Media & Molecular Imaging*, 2022(1), 4822235.

**Saleem, M. H., Potgieter, J., & Arif, K. M.** (2019). Plant disease detection and classification by deep learning. *Plants*, 8(11), 468.

**Sapkota, R., Meng, Z., Churuvija, M., Du, X., Ma, Z., & Karkee, M.** (2024). Comprehensive performance evaluation of yolo11, yolov10, yolov9 and yolov8 on detecting and counting fruitlet in complex orchard environments. *arXiv preprint arXiv:2407.12040*.

**Saran, C.** (2022). *Çanakkale ve çevresinden elde edilen lahana yaprak güvesi Plutella xylostella L.(Lepidoptera: Plutellidae) popülasyonlarının biyolojik özelliklerinin ve genetik çeşitliliğinin araştırılması*. Çanakkale Onsekiz Mart Üniversitesi, Lisansüstü Eğitim Enstitüsü,

**Sibiya, M., & Sumbwanyambe, M.** (2019). A computational procedure for the recognition and classification of maize leaf diseases out of healthy leaves using convolutional neural networks. *AgriEngineering*, 1(1), 119-131.

**Şalvarcı, Ü. B., & Ayten, U. E.** (2019). Yapay sinir ağları ile görüntü işlemeye dayalı uzaklıktan bağımsız ağırlık tahmin sistemi: yumurta ve portakal örnekleri. *Jeodezi ve Jeoinformasyon Dergisi*, 6(1), 26-46.

- Şeker, A., Diri, B., & Balık, H. H.** (2017). Derin öğrenme yöntemleri ve uygulamaları hakkında bir inceleme. *Gazi Mühendislik Bilimleri Dergisi*, 3(3), 47-64.
- Tejaswini, P., Singh, P., Ramchandani, M., Rathore, Y. K., & Janghel, R. R.** (2022). *Rice leaf disease classification using CNN*. Paper presented at the IOP Conference Series: Earth and Environmental Science.
- Tian, Y., Zhao, C., Zhang, T., Wu, H., & Zhao, Y.** (2024). Recognition Method of Cabbage Heads at Harvest Stage under Complex Background Based on Improved YOLOv8n. *Agriculture*, 14(7), 1125.
- Wang, C.-Y., Yeh, I.-H., & Liao, H.-Y. M.** (2024). Yolov9: Learning what you want to learn using programmable gradient information. *arXiv preprint arXiv:2402.13616*.
- Wang, D., Tan, J., Wang, H., Kong, L., Zhang, C., Pan, D., Li, T., & Liu, J.** (2025). SDS-YOLO: An improved vibratory position detection algorithm based on YOLOv11. *Measurement*, 244, 116518.
- Wiguna, L. H. S., Bimantoro, F., & Nugraha, G. S.** (2022). KLASIFIKASI MASALAH PADA KULIT WAJAH MENGGUNAKAN METODE CNN.
- Wu, M., Yuan, K., Shui, Y., Wang, Q., & Zhao, Z.** (2024). A Lightweight Method for Ripeness Detection and Counting of Chinese Flowering Cabbage in the Natural Environment. *Agronomy*, 14(8), 1835.
- Yeşilkaya, N.** (2022). Yapay zekâya dair etik sorunlar. *Şarkiyat*, 14(3), 948-963.
- Yuan, K., Wang, Q., Mi, Y., Luo, Y., & Zhao, Z.** (2023). Improved Feature Fusion in YOLOv5 for Accurate Detection and Counting of Chinese Flowering Cabbage (*Brassica campestris* L. ssp. *chinensis* var. *utilis* Tsen et Lee) Buds. *Agronomy*, 14(1), 42.
- Yurtsever, S. B.** (2024). Sosyal medya istihbaratının makine öğrenmesi çerçevesinde incelenmesi: Terörizm çalışmaları. *Savunma ve Güvenlik Araştırmaları Dergisi*, 1(1), 97-119.
- Zhao, X., Wang, X., Li, C., Fu, H., Yang, S., & Zhai, C.** (2022). Cabbage and weed identification based on machine learning and target spraying system design. *Frontiers in Plant Science*, 13, 924973.