



İSKENDERUN TEKNİK
ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

**YÜKSEK
LİSANS
TEZİ**

**DERİNLİK KAMERASI İLE
DERİN ÖĞRENME
TABANLI NESNE TESPİTİ**

Taner GÜNDÜZ

**BİLGİSAYAR MÜHENDİSLİĞİ
ANABİLİM DALI**

AĞUSTOS 2023



**DERİNLİK KAMERASI İLE DERİN ÖĞRENME TABANLI NESNE
TESPİTİ**

Taner GÜNDÜZ

**YÜKSEK LİSANS TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI**

**İSKENDERUN TEKNİK ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ**

AĞUSTOS 2023

Taner GÜNDÜZ tarafından hazırlanan “DERİNLİK KAMERASI İLE DERİN ÖĞRENME TABANLI NESNE TESPİTİ” adlı tez çalışması aşağıdaki jüri tarafından OY BİRLİĞİ ile İskenderun Teknik Üniversitesi Bilgisayar Mühendisliği Anabilim Dalında YÜKSEK LİSANS TEZİ olarak kabul edilmiştir.

Danışman: Doç. Dr. Yakup KUTLU

Bilgisayar Mühendisliği Anabilim Dalı, İskenderun Teknik Üniversitesi

Bu tezin, kapsam ve kalite olarak Yüksek Lisans Tezi olduğunu onaylıyorum.

.....
.....

Başkan: Doç. Dr. Yalçın İŞLER

Bilgisayar Mühendisliği Anabilim Dalı, İskenderun Teknik Üniversitesi

Bu tezin, kapsam ve kalite olarak Yüksek Lisans Tezi olduğunu onaylıyorum.

.....
.....

Üye: Doç. Dr. Yakup KUTLU

Bilgisayar Mühendisliği Anabilim Dalı, İskenderun Teknik Üniversitesi

Bu tezin, kapsam ve kalite olarak Yüksek Lisans Tezi olduğunu onaylıyorum.

.....
.....

Üye: Doç. Dr. Sertan ALKAN

Bilgisayar Mühendisliği Anabilim Dalı, İskenderun Teknik Üniversitesi

Bu tezin, kapsam ve kalite olarak Yüksek Lisans Tezi olduğunu onaylıyorum.

.....
.....

Tez Savunma Tarihi: 18/08/2023

Jüri tarafından kabul edilen bu tezin Yüksek Lisans Tezi olması için gerekli şartları yerine getirdiğini onaylıyorum.

.....

Doç. Dr. Ersin BAHÇECİ
Lisansüstü Eğitim Enstitüsü Müdürü

ETİK BEYAN

İskenderun Teknik Üniversitesi Lisansüstü Eğitim Enstitüsü Tez Yazım Kurallarına uygun olarak hazırladığım bu tez çalışmada;

- Yükseköğretim Kuruluna gönderilen kopya ile tarafından Lisansüstü Eğitim Enstitüsü'ne verilen basılı ve/veya elektronik kopyaların birebir aynı olduğunu,
 - Tez içinde sunduğum verileri, bilgileri ve dokümanları akademik ve etik kurallar çerçevesinde elde ettiğimi,
 - Tüm bilgi, belge, değerlendirme ve sonuçları bilimsel etik ve ahlak kurallarına uygun olarak sunduğumu,
 - Tez çalışmada yararlandığım eserlerin tümüne uygun atıfta bulunarak kaynak gösterdiğimi,
 - Kullanılan verilerde herhangi bir değişiklik yapmadığımı,
 - Bu tezde sunduğum çalışmanın özgün olduğunu,
- bildirir, aksi bir durumda aleyhime doğabilecek tüm hak kayıplarını kabullendiğimi beyan ederim.

İmza
Taner GÜNDÜZ
18/08/2023

DERİNLİK KAMERASI İLE DERİN ÖĞRENME TABANLI NESNE TESPİTİ
(Yüksek Lisans Tezi)

Taner GÜNDÜZ

İSKENDERUN TEKNİK ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ
Ağustos 2023

ÖZET

Dünya nüfusundaki artış, tarımsal üretimin bu artışı karşılaması açısından yeni teknolojilere ihtiyaç oluşturmaktadır. Bu kapsamda tarımsal üretimin hasat kısmında gerekli işçi gücünün bulunması ve verimli hasat işlemleri açısından sıkıntılar yaşanmakta olduğu tüm dünyada aşikâr bir gerçekliktir. Bu konuda otonom hasat robotlarının geliştirilmesi ve aktif olarak sahalarda kullanılması bu sorunun çözümü olarak sunulmaktadır. Dünya genelinde birçok ürün özelinde otonom hasat robotları geliştirme çalışmaları yapılmaktadır. Bu çalışmalar açısından özellikle bilgisayarlı görü çalışmalarında yaşanan gelişmeler, nesne tespiti ve sınıflandırılmasında kullanılan derin öğrenme algoritmalarında alınan başarılı sonuçlar otonom hasat robotlarının tarım sahalarda aktif kullanılmasını mümkün hale getirmiştir. Yazılımsal olarak yaşanan bu gelişmelerin yanı sıra donanımsal olarak bu algoritmaların üzerinde çalışacağı kartların gelişmesi ile otonom hasat robotlarının işlem hızları artmakta ve hasat süreleri azalmaktadır. Otonom hasat robotlarının gelişimi ile tarımsal alanda yaşanan işçi gücü problemi, insan kaynaklı hatalar ve iş kazalarının yaşanması sorunlarına çözüm geliştirilmektedir.

Bu tez çalışmasında, kamkat meyvesi otonom hasadında kullanılmak üzere derin öğrenme tabanlı nesne tespiti çalışması yapılmıştır. Bu kapsamda otonom hasat robotlarının görü sisteminde ihtiyaç duyduğu gerçek zamanlı nesne tespitini yapmak için YOLO algoritmasının dört farklı mimarisi ile eğitimler yapılarak en uygun model tespiti gerçekleştirilmiştir. Nesne tespitinden sonra kamkat meyvesinin hasadı için ihtiyaç duyulan toplama noktasının tespiti işlemi gerçekleştirilmiştir. Toplama noktası tespiti ile otonom hasat işleminde problem oluşturacak yaprak ve dal gibi problemlerin önüne geçilmesi hedeflenmektedir. Yapılan test işlemleri sonucunda en başarılı sonuçlar YOLOv7 modelinde %93 genel doğruluk ve 35 FPS (Saniyedeki Kare Sayısı) oranı ile elde edilmiştir.

Anahtar Kelimeler : Otonom hasat robotu, derin öğrenme, YOLO, kamkat, bilgisayarlı görü

Sayfa Adedi : 54

Danışman : Doç. Dr. Yakup KUTLU

DEEP LEARNING BASED OBJECT DETECTION WITH DEPTH CAMERA
(M. Sc. Thesis)

Taner GÜNDÜZ

ISKENDERUN TECHNICAL UNIVERSITY
INSTITUTE OF GRADUATE STUDIES

August 2023

ABSTRACT

The increase in the world population creates a need for new technologies in order to meet this increase in agricultural production. In this context, it is an obvious reality all over the world that there are problems in the harvesting part of agricultural production in terms of finding the necessary labor force and efficient harvesting processes. In this regard, the development of autonomous harvesting robots and their active use in the fields are presented as a solution to this problem. There are studies to develop autonomous harvesting robots for many products around the world. In terms of these studies, especially the developments in computer vision studies, the successful results obtained in deep learning algorithms used in object detection and classification have made it possible to actively use autonomous harvesting robots in agricultural fields. In addition to these developments in software, with the development of the cards on which these algorithms will work in hardware, the processing speed of autonomous harvesting robots increases and harvest times are reduced. With the development of autonomous harvesting robots, solutions are being developed to the problems of labor power in the agricultural field, human-induced errors and occupational accidents.

In this thesis, a deep learning-based object detection study was carried out to be used in autonomous harvesting of kumquat fruit. In this context, the most suitable model was determined by training with four different architectures of the YOLO algorithm in order to detect real-time objects that autonomous harvesting robots need in the vision system. After the object detection, the collection point needed for the harvest of kumquat fruit was determined. With the collection point detection, it is aimed to prevent problems such as leaves and branches that will cause problems in the autonomous harvesting process. As a result of the test processes, the most successful results were obtained in the YOLOv7 model with 93% overall accuracy and 35 FPS (Frame Per Second).

Key Words : Autonomous harvesting robot, deep learning, YOLO, kumquat, computer vision

Page Number : 54

Supervisor : Assoc. Prof. Dr. Yakup KUTLU

TEŐEKKÜR

Yüksek Lisans tez konusunun belirlenmesi, araştırılması ve yazımı sırasında sahip olduđu bilgi birikimi ve tecrübesi ile çalışmayı yönlendiren ve hiçbir yardımı esirgemeyen saygıdeđer danışman hocam Doç. Dr. Yakup KUTLU'ya sonsuz saygı ve teşekkürlerimi sunarım.

Tez çalışmalarım sırasında hiçbir desteđi esirgemeyen ve sürekli yanımda olan saygıdeđer mesai arkadaşım Yüksek İnşaat Mühendisi Samed GÖÇMEN'e, en büyük destekçim sevgili eşim Remziye GÜNDÜZ'e ve aileme şükranlarımı sunarım.

İÇİNDEKİLER

	Sayfa
ÖZET	iv
ABSTRACT.....	v
TEŞEKKÜR.....	vi
İÇİNDEKİLER	vii
ÇİZELGELERİN LİSTESİ.....	ix
ŞEKİLLERİN LİSTESİ	x
RESİMLERİN LİSTESİ	xii
SİMGELER VE KISALTMALAR.....	xiii
1. GİRİŞ.....	1
2. ÖNCEKİ ÇALIŞMALAR.....	3
2.1. Klasik Yöntemler Kullanılan Çalışmalar	3
2.2. Yapay Zeka Yöntemleri Kullanılan Çalışmalar	5
3. MATERYAL VE YÖNTEM	9
3.1. Deneysel Çalışma.....	9
3.2. Veri Seti	9
3.2.1. Veri artırım yöntemleri	10
3.3. Görüntü İşleme	11
3.3.1. Görüntü işleme uygulamaları	11
3.3.2. Görüntü işleme teknikleri.....	13
3.4. Yapay Zeka	14
3.4.1. Makine öğrenmesi	15

	Sayfa
3.4.2. Yapay sinir ađları.....	17
3.4.3. Derin öğrenme.....	18
3.5. Gerçek Zamanlı Nesne Tespiti ve Takibi.....	20
3.5.1. Evrişimli sinir ađı	22
3.5.2. YOLO algoritması	25
3.6. Performans Ölçümü	29
4. ARAŞTIRMA BULGULARI VE SONUÇLAR.....	31
4.1. Derin Öğrenme Tabanlı Kamkat Meyvesi Otonom Hasat Uygulaması.....	31
4.2. Veri Setinin Oluşturulması.....	31
4.3. YOLO Modellerinin Eğitim ve Test İşlemleri	33
4.3.1. YOLOv4.....	33
4.3.2. YOLOv5.....	36
4.3.3. YOLOv6.....	39
4.3.4. YOLOv7.....	41
4.4. Kamkat Toplama Noktasının Tespiti	45
4.5. YOLO Modelleri Test Sonuçları.....	46
5. TARTIŞMA VE ÖNERİLER.....	48
5.1. Tartışma.....	48
5.2. Öneriler	49
KAYNAKLAR	50
EKLER.....	54

ÇİZELGELERİN LİSTESİ

Çizelge	Sayfa
Çizelge 3.1. Hata (karışıklık) matrisi	29
Çizelge 4.1. YOLO modelleri test sonuçları	47
Çizelge 4.2. YOLO modelleri FPS sonuçları.....	47
Çizelge 5.1. Literatürdeki otonom hasat robotları ile karşılaştırma	48



ŞEKİLLERİN LİSTESİ

Şekil	Sayfa
Şekil 3.1. Yapay zeka, makine öğrenmesi ve derin öğrenme ilişkisi.....	15
Şekil 3.2. Öğrenme yaklaşımına göre makine öğrenmesi modelleri	16
Şekil 3.3. Yapay sinir ağı modeli	18
Şekil 3.4. Makine öğrenmesi ve derin öğrenme genel çalışma prensibi.....	19
Şekil 3.5. Nesne tespiti ve takibi yöntemleri	20
Şekil 3.6. Evrişimli sinir ağının yapısı	22
Şekil 3.7. Aktivasyon fonksiyonları.....	24
Şekil 3.8. Gerçek zamanlı uygulamalarda oransal hız karşılaştırması.....	26
Şekil 3.10. YOLO mimarisi	26
Şekil 4.1. Sistem İlerlemesi Akış Diyagramı	31
Şekil 4.2. DenseNet yoğun blok modelinin yapısı.....	34
Şekil 4.3. DensNet aktarımı ve ağırlık güncelleme denklemi.....	34
Şekil 4.4. CSPDensNet yoğun blok yapısı.....	34
Şekil 4.5. CSPDensNet ağırlık güncelleme denklemleri	35
Şekil 4.6. YOLOv4 mimarisinin genel yapısı.....	35
Şekil 4.7. YOLOv4 kayıp ve ortalama keskinlik grafiği	36
Şekil 4.8. YOLOv5 mimarisinin genel yapısı.....	37
Şekil 4.9. YOLOv5 sonuç grafikleri	38
Şekil 4.10. YOLOv5 karmaşıklık matrisi	38
Şekil 4.11. Rep-PAN boyun yapısı	39

Şekil	Sayfa
Şekil 4.12. YOLOv6 ayrılmış kafa yapısı mimarisi.....	40
Şekil 4.13. YOLOv6 sonuç grafikleri	40
Şekil 4.14. YOLOv6 karmaşıklık matrisi	41
Şekil 4.15. ELAN mimari diyagramı	42
Şekil 4.16. E-ELAN mimari diyagramı	42
Şekil 4.17. YOLOv7 mimarisi bileşik ölçeklendirme yapısı	43
Şekil 4.18. YOLOv7 sonuç grafikleri	44
Şekil 4.19. YOLOv7 karmaşıklık matrisi	44

RESİMLERİN LİSTESİ

Resim	Sayfa
Resim 3.1. Veri seti örnekleri.....	9
Resim 3.2. Giriş resminin ızgaralara bölünmesi	27
Resim 3.3. Sınır kutularının çizdirilmesi.....	28
Resim 3.4. IOU hesabı ve kıyaslanması	28
Resim 4.1. Kamkat veri setinden örnekler	32
Resim 4.2. Gürültü eklenmiş kamkat görüntüleri.....	32
Resim 4.3. Kamkat etiketleme örnekleri	33
Resim 4.4. YOLOv7 nesne tespiti çıktısı.....	45
Resim 4.5. Kamkat toplama noktalarının işaretlenmesi.....	46

SİMGELER VE KISALTMALAR

Bu çalışmada kullanılmış simgeler ve kısaltmalar, açıklamaları ile birlikte aşağıda sunulmuştur.

Kısaltmalar	Açıklamalar
CNN	Evrişimli Sinir Ağı
CUDA	Paralel Bilgi İşlem Platformu
ELAN	Verimli Katman Toplama Ağı
FPN	Özel Piramit Ağı
FPS	Saniyedeki Kare Sayısı
GPU	Grafik İşlem Birimi
HSV	Ton ve Doygunluk Değeri
IOU	Kesişme Oranı
LIDAR	Işık Algılama ile Ölçüm Sensörü
MRI	Manyetik Rezonans Görüntüleme
NMS	Maksimum Olmayanı Bastırma Yöntemi
OTSU	Eşitleme Yöntemi
PAN	Yol Toplama Ağı
RGB	Renkleri Çıkaran Metot
SAM	Uzamsal Dikkat Modülü
SPP	Uzamsal Piramit Havuzlama Katmanı
YOLO	Sadece Bir Kez Bak Algoritması
YZ	Yapay Zeka

1. GİRİŞ

Birleşmiş Milletler Gıda ve Tarım Örgütü raporlarına göre tarım sektörüne karşı olan ihtiyaç önümüzdeki yıllarda üst düzey bir artış gösterecektir. Bu örgütün geçmiş yıllardaki raporlarında, 2050 yılında artan dünya nüfusunun gıda ihtiyacının karşılanması için 2006 yılı itibariyle %70 daha fazla gıda üretilmesi gerektiği belirtilmiştir (Odegard ve Van der Voet, 2014). Bu talebi karşılamak için çiftçiler ve tarım şirketleri, yıllardır daha verimli üretim yöntemleri bulmaya çalışmaktadır. Bu kapsamda öncelikle gübre ve ilaç sanayisi gelişmiştir. Kullanılan bu ürünlerle birim metrekare başına düşen ürün veriminin artırılması sağlanmıştır. Ürünlerin bazı hastalıklardan korunması sağlanmıştır. Fakat bunların uygulanması esnasında ihtiyaç duyulan insan gücü sorunlara neden olmaktadır. Bu sebeple özellikle ilaçlama konusunda son yıllarda dronlar ile büyük tarım arazilerinin ilaçlanması işlemleri gerçekleştirilmektedir. Gelişen dron ve kamera teknolojisi ile tarım arazileri üzerinde uçuşlar yapılarak alınan görüntüler üzerinden analizler yapılmakta, ürüne bölgesel olarak gerekli ilaçların reçetesi çıkarılmaktadır.

Tarımsal üretimin artırılmasındaki en büyük engellerden birisi tarımsal iş gücü ihtiyacıdır. Tarımsal iş gücü, ürün yetiştirme aşmalarında en çok hasat işlemi sırasında yoğun şekilde gerekmektedir. Bu kapsamda yıllar içerisinde makinalı tarıma geçilmiş fakat bu sadece bir kısım ürünler için çözüm olmuştur. Özellikle sebze ve meyve yetiştiriciliğinde hasat işlemi için hala yoğun iş gücü kullanılmaktadır. Bu konuda en önemli çözüm robotik teknolojilerinin gelişmesi ile ortaya çıkan otonom hasat robotlarıdır. Robotik sistemlerin gelişmesinin yanı sıra gelişen yazılım mimarileri ve bunların üzerinde çalışabileceği donanımların ortaya çıkması ile çalışmalar bu alanda yoğunlaşmaya başlamıştır.

Otonom hasat robotları ile ilgili çalışmalar daha çok sera ortamındaki ürünler özelinde yapılmaktadır. Sera ortamındaki ürünlerin hasadında dış ortam koşullarından daha az etkilenildiği için görece başlangıç için daha uygun ortamlardır. Bu kapsamda kullanılan kamera sistemleri ile görüntü işleme yazılımlarının uygulanmasında karşılaşılan hata oranları daha kabul edilebilir seviyelerde olmaktadır. Yapılan çalışmalarda henüz gerçek tarımsal bir sahada yüksek verim ile hasat işlemi yapabilen bir sonuç elde edilememiştir. Bu noktada yapılan çalışmalar incelendiğinde, ilk olarak başarımları olumsuz etkileyen

faktör olarak tarımsal saha ortamında ürünlerin gerçek zamanlı tespitini yaparken değişen ışık miktarı, gölge durumları, ürünlerin yaprakların arkasında kalması ve yapışık meyvelerin ayrı ayrı tespit edilememesi gibi durumların ortaya çıktığı gözlenmiştir. Bunlardan özellikle ürünlerin yaprak ve dalların arkasında kalması durumlarında tespit yapılsa bile robotik sistem tarafından yapılan hasat sırasında başarısızlıklar yaşandığı görülmektedir.

Bu tez çalışmasında otonom hasat robotlarında ihtiyaç duyulan gerçek zamanlı nesne tespiti ve toplama noktasının belirlenmesi üzerine çalışılmıştır. Çalışmada meyvesi küçük ve hassas bir yapıda olduğu için tek tek elle toplanmak zorunda olunan kamkat tercih edilmiştir. Kamkat meyvesi hasat işlemi el ile tek tek yapıldığı için yoğun tarımsal iş gücüne ihtiyaç duymaktadır. Otonom hasat işlemindeki en önemli konu ilk olarak hasat edilecek ürünün nesne tespitinin doğru ve hızlı yapılmasıdır. Bu kapsamda gerçek zamanlı nesne tespitinde ön plana çıkan YOLO algoritmasının dört farklı mimarisinde kamkat meyvesi özelinde nesne tespiti için eğitim ve test işlemleri gerçekleştirilmiştir. Otonom hasat işlemi gerçek zamanlı bir uygulama olacağı için modellerin tespit başarımı kadar tespit hızları da önem arz etmektedir. En uygun model seçilirken genel başarımın yanında FPS oranları da dikkate alınarak karar verilmiştir. Bu aşamadan sonra literatürde yer alan otonom hasatları için yapılan çalışmalar incelendiğinde eğitilen modeller robotik sistemlere uygulandığında yaprak ve dalların hasat başarımını düşürmesi durumuyla karşılaşıldığı görülmüştür. Bu kapsamda nesne tespiti işleminin devamı olarak tespit edilen kamkat meyveleri üzerinde hasat için en uygun toplama noktası tespiti yapılmıştır. Toplama noktası tespiti ile otonom hasat robotları için yaptığımız görüşümüz daha başarılı ve sahada uygulanabilir bir yapıya kavuşmuştur. Bu çalışma ile hem gerçek zamanlı bir hasat işlemi için başarılı bir derin öğrenme modeli önerilmiş hem de hasat işleminde literatürde yer alan çalışmalarda ortaya çıkan yaprak ve dal arkası ürünlerde yaşanan probleme bir çözüm önerisi getirilmiştir.

2. ÖNCEKİ ÇALIŞMALAR

Otonom hasat robotları ile ilgili yapılmış çalışmalar incelediğinde üzerinde durulan en önemli konunun meyvenin tespiti ve konumunun belirlenmesi olduğu görülmektedir. Bu noktadaki başarımlar ne kadar yüksek olursa mekaniksel olarak yapılacak hasat işlemi bir o kadar başarılı olacaktır. Bu kapsamda literatürdeki otonom hasat robotu çalışmaları nesne tespitinde kullanılan yöntemlerine göre klasik yöntemlerin kullanıldığı çalışmalar ve yapay zeka yöntemlerinin kullanıldığı çalışmalar olarak iki grupta incelenmiştir.

2.1. Klasik Yöntemler Kullanılan Çalışmalar

Tanigaki, Fujiura, Akase ve Imagawa tarafından kiraz meyvesi hasadı için yapılan çalışmada hasat robotu için 4 eksenli robot kol seçilmiştir. Robot kol üzerine kırmızı ve kızılötesi lazer diyotlar ile 3 boyutlu bir sensör yerleştirilmiştir. Nesne tespiti için klasik görüntü işleme yöntemleri kullanılmış fakat bir kısmı yaprak arkasında kalan meyvelerin tespiti sağlanamamıştır. Hasat robotunun tutma ucu için vakumlu bir tasarım kullanılmıştır. Tasarlanan hasat sistemi test aşamasında kiraz ağacı üzerinde sadece 12 meyve tespit ederken bu meyvelerden 8 tanesi hasat edilebilmiştir (Tanigaki, Fujiura, Akase ve Imagawa, 2008).

Yin, Chai, Yang ve Mittal tarafından domates hasadı için yapılan çalışmada hasat robotları için nesne tespiti ve konum belirleme üzerine yoğunlaşmıştır. Bu kapsamda olgun domateslerin belirlenmesinde renk tabanlı K-Means kümeleme algoritması kullanılmıştır. Domateslerin şekil ve boyut özellikleri kullanılarak lazer sensörler aracılığı ile hasat için doğru konumu tespiti yapılmaya çalışılmıştır (Yin, Chai, Yang ve Mittal, 2009).

Kurtulmuş, Lee ve Vardar tarafından narenciye hasadı için yapılan çalışmada hasat robotları için nesne tespiti üzerinde durulmuştur. Bu doğrultuda nesne tespiti için klasik görüntü işleme yöntemleri ile renk verisi ve Gabor filtresinden faydalanılmıştır. Test aşamasında toplanan 96 narenciye görüntüsü üzerinde %75,3 nesne tespit başarımları sağlanmıştır (Kurtulmuş, Lee ve Vardar, 2011).

Zhao, Lv, Ji ve Zhang tarafından elma meyvesi hasadı için yapılan çalışmada hasat robotu için 5 eksenli robot kol seçilmiştir. Elma meyvesinin tespiti için radyal tabanlı fonksiyona sahip destek vektör makinası kullanılmıştır. Sonuç olarak yaprak arkası ve yapışık meyvelerin tespiti başarımı düşürmüştü fakat tespit edilen meyvelerin hasat işleminde %77'lik bir başarıma ulaşılmıştır (Zhao, Lv, Ji ve Zhang, 2011).

Qingchun, Xiu, Wengang, Quan ve Kai tarafından çilek meyvesi hasadı için yapılan çalışmada hasat robotu için 6 eksenli robot kol seçilmiştir. Çilek meyvesinin tespiti için HSV parametrelerinden ton ve doyguluk özellikleri kullanılarak klasik görüntü işleme metotları uygulanmıştır. Deneysel ortamında yapılan test işlemlerinde nesne tespiti aşamasında yaprak ve ışık değişimi sorunu yaşanmış sonuç olarak tespit edilen meyvelerin %86 oranında hasadı gerçekleştirilmiştir (Qingchun, Xiu, Wengang, Quan ve Kai, 2012).

Berki, Artan ve Orhan tarafından kiraz meyvesi hasadı için yapılan çalışmada hasat robotu için nesne tespiti konusu üzerine durulmuştur. Bu kapsamda proje için Raspberry Pi 3 kontrolcü kartı ve A4TECH-PK-635-USB kamera kullanılmıştır. Nesne tespiti algoritması için C++ programlama dilinde Opencv kütüphanesi kullanılarak HSV renk uzayı ile toplanması en uygun kirazlar tespit edilmiştir. Deneysel test ortamında yapraklardan arındırılmış ve nominal ışık altında tüm kirazlar tespit edilirken gün içinde gün ışığının yön değişimlerine bağlı olarak başarılı tespitler yapılamadığı belirtilmiştir (Berki, Artan ve Orhan, 2017).

Feng, Zou, Fan, Zhang ve Wang tarafından salkım domates hasadı için yapılan çalışmada hasat robotu için kurulan sera ortamı ray sistemi üzerinde hareket eden 6 eksenli robot kol seçilmiştir. Salkım domatesin rengi kırmızı ve yaprakları yeşil olduğu için klasik görüntü işleme metotları ile renk parametreleri kullanılarak nesne tespiti yapılmıştır. Deneysel olarak en uygun koşulları sağlayarak tasarlanan test ortamında 30 adet salkım domatesten 25 tanesi başarılı bir şekilde toplanarak %83 başarıma ulaşılmıştır (Feng, Zou, Fan, Zhang ve Wang, 2018).

Fu, Tola, Al-Mallahi, Li ve Cui tarafından kivi meyvesi hasadı için yapılan çalışmada hasat robotu için nesne tespiti sistemi üzerine yoğunlaşmıştır. Kivi meyvesinin tespiti için öncelikle renk segmentasyonu kullanılarak istenmeyen arka planlar elimine edilmiştir. Bu

işlem ile elde edilen görüntü üzerinde OTSU metodu ve sonrasında morfolojik görüntü işleme yöntemleri kullanılarak kivi meyvesinin tespiti gerçekleştirilmiştir. Yaprak ve ışık sorunundan arındırılmış uygun deneysel ortamda yapılan testlerde tasarlanan sistem ile %93,7 oranında doğru sonuçlar elde edilmiştir (Fu, Tola, Al-Mallahi, Li ve Cui, 2019).

Yu ve arkadaşları tarafından elma meyvesi hasadı için yapılan çalışmada hasat robotu için iki kollu insan yapısında bir robot kol tasarlanmıştır. Elma meyvesinin tespiti için HSV renk uzayı kullanılmıştır. Canny kenar dedektörü ve morfolojik işlemler kullanılarak klasik görüntü işleme yöntemlerinden yararlanılmıştır. Yapılan işlemler sonucunda kurulan deneysel test ortamında nesne tespiti başarımı %82, hasat başarı ise %72 olarak raporlanmıştır (Yu ve diğerleri, 2021).

2.2. Yapay Zeka Yöntemleri Kullanılan Çalışmalar

Birrell, Hughes, Cai ve Iida tarafından marul hasadı için yapılan çalışmada hasat robotu olarak 6 eksenli bir robot kol kullanılmıştır. Sera ortamında marulların tespiti için 665 görüntü toplanmıştır. Marul görüntüleri kullanılarak YOLOv3 eğitimleri sonucunda %91 başarılı nesne tespiti yapılmıştır. Tutma ucu olarak hem tutma hem kesme yapabilen bir tasarım gerçekleştirilmiştir. Sera deney ortamında yapılan testlerde 69 marul içerisinde 31 tanesi başarılı şekilde hasat edilmiştir (Birrell, Hughes, Cai ve Iida, 2019).

Williams ve arkadaşları tarafından kivi meyvesi hasadı için yapılan çalışmada hasat robotu olarak 4 eksenli bir robot kol içeren sistem tasarlanmıştır. Bu sistem içerisinde ayrıca kivi meyvesinin hassas yapısına uygun bir tutma ucu tasarlanmıştır. Nesne tespiti için Fully-Convolutional Network (semantik bölütleme) modeli kullanılmıştır. Deneysel test ortamında yapılan hasat işlemlerinde meyve kayıp oranı %24,6 olarak belirtilmiştir. Bunun yanı sıra her bir meyve için toplama döngü süresi 5.5 sn olarak ölçülmüştür (Williams ve diğerleri, 2019).

Arad ve arkadaşları tarafından biber hasadı için yapılan çalışmada hasat robotu olarak 6 eksenli bir robot kol kullanılmıştır. Sera ortamında yapılan çalışmada biber fideleri arasına raylar döşenmiş ve robot kol için yükseklik ayarı yapılabilen bir mekanizma yerleştirilmiştir. Biberlerin tespiti için RGB-D kamera ile semantik segmentasyon

algoritması kullanılmıştır. Bu sayede biberler ve sap kısımları ayrı renkler ile işaretlenerek tespit işlemi gerçekleştirilmiştir. Görüntü işleme yöntemleri kullanılarak biber hasadında kesme işlemi yapılacak noktalar belirlenmiştir. Test işlemlerinde başarılı hasat oranı %61 olarak ölçülmüştür. Toplama süresi her bir ürün için 24sn olarak tespit edilmiştir (Arad ve diğerleri, 2019).

Yu, Zhang, Liu, Yang ve Van Zhang tarafından çilek meyvesi hasadı için yapılan çalışmada hasat robotu olarak 6 eksenli bir robot kol kullanılmıştır. Çalışmada hasat işlemi için nesne tespiti ve toplama noktası belirleme üzerine yoğunlaşmıştır. Bu kapsamda gerçek zamanlı nesne tespiti algoritması olan YOLO algoritmasının V3 (versiyon3) mimarisi kullanılmıştır. YOLOv3 algoritmasının öz nitelik çıkarma katmanları yerine MobileNet-V1 algoritmasının ilgili kısımları eklenerek R-YOLO isimli yeni bir derin öğrenme mimarisi oluşturulmuştur. Elde edilen yeni algoritmanın eğitimi için 1900 adet çilek resmi kullanılmıştır. Test işlemleri sonucunda başarımlar %94,43 olarak tespit edilmiştir. Orijinal YOLOv3 test başarımları ile kıyaslandığında başarımlar %1,35 oranında düştüğü fakat nesne tespit hızında 3,6 kat artış olduğu belirlenmiştir. Yeni algoritmanın gerçek zamanlı bir hasat için daha başarılı olacağı belirtilerek özel yapılandırılmış bir sahada test işlemleri yapıldığında %84,35 oranında hasat başarımları tespit edilmiştir (Yu, Zhang, Liu, Yang ve Van Zhang, 2020).

Kuznetsova, Maleva ve Soloviev tarafından elma meyvesi hasadı için yapılan çalışmada nesne tespiti için kullanılacak derin öğrenme algoritmasının eğitimi aşamasında toplanan resimlerde ön işlemlerin nesne tespit başarımlarına etkisi incelenmiştir. Bu kapsamda derin öğrenme algoritmalarından YOLOv3 kullanılmıştır. Algoritma eğitimi için 5142 adet elma resmi toplanmıştır. Ön işlemler olmadan yapılan test işlemleri sonucu %90,9 başarımlar elde edilmiştir. Ön işlemler olarak kontrast artırımı, bordür kalınlaştırılması ve medyan filtre ile hafif bulanıklık uygulamaları kullanılmıştır. Ön işlemler kullanılarak oluşturulan yeni veri seti ile yapılan eğitimler sonucunda başarımlar %92,2 olarak gerçekleşmiştir (Kuznetsova, Maleva ve Soloviev, 2020).

Yin ve arkadaşları tarafından üzüm meyvesi hasadı için bilgisayarlı görü çalışması yapılmıştır. Çalışmadaki ana gaye başarılı bir hasat işlemi için meyvenin tespiti ve duruş açısının en doğru şekilde belirlenmesi üzerinedir. Bu kapsamda Mask R-CNN evrişimli

sinir ağı kullanılarak segmente bir şekilde nesne tespiti yapılmıştır. Ransac algoritması kullanılarak aykırı değerler elimine edilip silindir model oluşturulmuştur. Bir sonraki adımda ise silindir model eksenini kullanılarak üzüm için poz tahmini yapılmaktadır. Bu işlemin eğitimleri için 150 üzüm meyvesi görüntüsü toplanmış ve bu görüntüler veri artırımı yöntemlerinden döndürme, parlaklık ve doygunluk kullanılarak 1260 görüntü elde edilmiştir. 1050 görüntü ile 400 iterasyonluk eğitim sonucunda 210 resim üzerinde test işlemi yapıp kayıp değerinin 0,1'in altında olduğu tespit edilmiştir. Sonuç olarak hassasiyet: %89,53, hatırlatma: %95,33 ve kesişim (IOU): %82 olarak bulunmuştur. Her bir üzüm bölgesinin tespitinin ise toplamda 1,7 sn sürdüğü belirtilmektedir (Yin ve diğerleri, 2021).

Lin, Zhu, Li, Zou ve Van Tang tarafından guava meyvesi otonom hasadı için tasarlanan robotik sisteme, toplama başarımını artıracak bir derin öğrenme tabanlı toplama yolu planlama algoritması önerilmiştir. Öncelikle geçmiş durumları hatırlamak için tekrarlayıcı sinir ağı kullanılmıştır. Daha sonra derin deterministik politika gradyan algoritması ile robot tutma ucunun meyveye ulaşabilmek için engellere çarpmadan izleyeceği yol simüle edilerek belirlenmiştir. Deneysel test işlemleri sonucunda %90,90 başarıma ulaşıldığı belirtilmektedir (Lin, Zhu, Li, Zou ve Van Tang, 2021).

Jun, Kim, Seol ve Son tarafından domates hasadı için yapılan çalışmada hasat robotu olarak 6 eksenli bir robot kol kullanılmıştır. Hasat işlemi başarımını artırmak için vantuzlu ve makaslı özel bir tutma ucu tasarımı yapılmıştır. Nesne tespiti için Yolov3 algoritması kullanılmıştır. Bu kapsamda domates serasından alınan 770 görüntü, 700 eğitim ve 70 test olarak bölünerek 4000 iterasyonda eğitimi gerçekleştirilmiştir. Test işlemi sonucunda başarımlar %90 olarak tespit edilmiştir. Hasat başarımı için sera ortamında yapılan testlerde domatesin tespiti ve dalından kesilmesi tam bir döngü olarak düşünüldüğünde 5,9 sn döngü süresi olarak belirtilmiştir. Bunun yanı sıra her bir salkımdaki domates sayısı arttıkça hasat başarımının azaldığı değerlendirilmiştir (Jun, Kim, Seol ve Son, 2021).

Zhang, Lammers, Chu, Li ve Lu tarafından elma meyvesi hasadı için yapılan çalışmada hasat robotu olarak 3 eksenli bir robot kol tasarlanmıştır. Nesne tespiti için Mask-R-CNN algoritması kullanılmıştır. Eğitim işlemi için 15000 elma görüntüsü kullanılmıştır. Test işlemi sonucunda %90,5 nesne tespit başarımı elde edilmiştir. Tasarlanan sistemde RGB-D

kamera kullanılarak derinlik tespiti yapılmaktadır. Deneysel test ortamında yapılan hasat çalışmasında 64 elmadan 41 tanesi başarılı bir şekilde toplanarak %64,06 oranında başarı elde edilmiştir. Her bir elma için tam döngü hasat süresi ortalama 8,8 sn olarak tespit edilmiştir (Zhang, Lammers, Chu, Li ve Lu, 2021).

Kar, Basgumus ve Namdar tarafından biber hasadı için yapılan çalışmada sera ortamında manuel olarak gezdirilip daha sonra başlangıç noktasına getirilerek hasada başlayan bir sistem geliştirilmiştir. Bu kapsamda nesne tespiti için 5041 adet biber görüntüsü kullanılarak Fast-R-CNN ve SSD MobilnetV2-little modelleri üzerinden eğitimler yapılmıştır. Kullanılan bu derin öğrenme algoritmalarından SSD MobilnetV2-little ağının daha başarılı olduğu belirtilerek test işlemi sonucunda doğruluk oranının %88,66 olduğu tespit edilmiştir (Kar, Basgumus ve Namdar, 2021).

Rong, Wang, Yang ve Huang tarafından istiridye mantarı hasadı için yapılan çalışmada hasat robotu olarak 3 eksenli bir robot kol tasarlanmıştır. Mantar serası içerisine ray sistemi yerleştirilmiş ve hasat robotu bu raylar üzerinde hareket ettirilmiştir. Robot kol sistemi üzerine görüş için RGB-D kamera ve led ışıklar eklenmiştir. Nesne tespiti için 4600 mantar görüntüsü SSD mimarisi üzerinde eğitilmiştir. Nesne tespiti başarımları %95 olarak tespit edilmiştir. Sera ortamında yapılan hasat test işlemlerinde toplama sırasında zarar gören mantarlar göz ardı edilerek başarımları %86 olarak belirtilmiştir. Hasat işleminde tam bir döngü tamamlanma süresi 8,85 sn olarak tespit edilmiştir (Rong, Wang, Yang ve Huang, 2021).

3. MATERYAL VE YÖNTEM

3.1. Deneysel Çalışma

Bu çalışmada kamkat meyvesinin otonom hasadı için ihtiyaç duyulan gerçek zamanlı nesne tespiti ve toplama noktasının belirlenmesi üzerine çalışılmıştır. Otonom hasat işlemindeki ilk ve en önemli aşama hasat edilecek meyvelerin en doğru şekilde tespit edilmesidir. Ürünün tespit edilip konumunun belirlenmesinin yanında yaprak, dal vb. engellerin elimine edilmesi de bir o kadar önem arz etmektedir. Tüm bunlar göz önüne alındığında çalışmada gerçek zamanlı nesne tespiti için YOLO algoritmasının farklı mimarileri ile eğitimler ve testler gerçekleştirilmiştir. Belirlenen en uygun model üzerinden tespit edilen kamkatlar için toplama noktası tespiti yapılmıştır. Sonuç olarak bir otonom hasat robotunun görü sistemi için gerekli olan tüm çalışmalar kamkat meyvesi özelinde gerçekleştirilmiştir.

3.2. Veri Seti

Derin öğrenme algoritmalarından başarılı sonuçlar almak için veri setinin uygun kriterlerde olması oldukça önemlidir. Veri setinin büyüklüğü ve çeşitliliği algoritmanın eğitim başarımını doğrudan etkilemektedir. Veri setinin büyüklüğü eğitim başarımını olumlu yönde etkilerken bir taraftan da öğrenme için gereken zamanı ve sonucunda elde edilecek eğitilmiş modelin büyüklüğünü artırdığı göz ardı edilmemelidir. Bu noktada modelin donanımlar üzerinde çalışma hızı ve sistemin gerçek sahada uygulanabilirliği azalacaktır.



Resim 3.1. Veri seti örnekleri

Çalışmada kullanılacak derin öğrenme modellerinin kamkat meyvesi özelinde eğitim ve test işlemlerinin yapılabilmesi için kamkat fidelerinden 220 adet farklı açılarda ve ışıkta görüntüler toplanmıştır. Kamkat meyvesi veri setinden bazı örnekler Resim 3.1’de gösterilmektedir.

3.2.1. Veri artırım yöntemleri

Derin öğrenme algoritmalarında kullanılacak veri setinin yetersiz olması durumunda veya daha fazla çeşitlilik gerektiğinde veri artırım yöntemleri kullanılmaktadır. Bu kapsamda görüntü verileri için en çok kullanılan veri artırım yöntemleri incelenmiştir (Yılmaz, 2016).

Döndürme: Görüntülerin belirli açılarda döndürülmesiyle sağlanan veri artırım yöntemidir. Kısaca döndürme işlemi piksellerin satır ve sütunlarının tersine çevrilmesi olarak da tanımlanmaktadır.

Ölçeklendirme: Görüntülerin belirli oranlarda boyutlarının değiştirilmesiyle sağlanan veri artırım yöntemidir. Boyut değişimi ile görüntü içinde tespiti yapılacak nesnelerin de farklı büyüklüklerde versiyonları elde edilmektedir.

Yansıtma: Görüntülerin dikey ve yatay olarak ayna yansımaları ile sağlanan veri artırım yöntemidir. Örneğin bir araba görüntüsü yatay olarak yansıtıldığında karşı yönden gelen araba görüntüsü elde edilir.

Kırpma: Görüntülerin belirli kısımlarında rast gele kesme işlemleri yapılarak sağlanan veri artırım yöntemidir. Örneğin bir plaka okuma sistemi için araba görüntülerinin plaka kısımlarına ihtiyaç duyulduğunda veri artırımı için kullanılması ideal olmaktadır. Görüntülerin istenen kısımlarındaki çözünürlüğü artırmak için de tercih edilmektedir. Kırpma metodunu kullanırken görüntülerdeki anlamlı veri kısımlarının kaybedilmemesine dikkat edilmelidir.

Kontrast değişim: Görüntülerin gün içi ışık ve renk değişimine göre sağlanan veri artırım yöntemidir. Kontrast değişimi metodu, aynı zamanda görüntülerdeki gürültüyü azaltmak veya görüntüyü daha net hale getirmek gibi diğer amaçlar için de kullanılabilir. Dikkat

edilmesi gereken nokta kontrastın aşırı artırılması veya azaltılması görüntünün kalitesini bozabilir bu nedenle modelin performansını düşürebilir.

Gürültü ekleme: Görüntülerin üzerine gürültü ekleyerek kalite değişimi ile veri artırımı sağlar. Bu yöntemde gaussian gürültü ekleme, salt and pepper (tuz ve biber) gürültü ekleme ve permütasyon gürültü ekleme gibi metotlar bulunmaktadır. Görüntülerin üzerine gürültü ekleyerek modelin gerçek koşullar altında daha başarılı olması amaçlanmaktadır. Bu metot ayrıca modelin eğitim aşamasında karşılaşılabileceği aşırı öğrenme problemine de çözüm olarak kullanılmaktadır. Gürültü eklerken düzeyini iyi belirlemek gerekmektedir yoksa veri seti kalitesi çok düşeceği için model başarımı azalabilir.

3.3. Görüntü İşleme

Görüntü işleme; bir dijital görüntünün bilgisayar tarafından analiz edilmesi, işlenmesi ve yorumlanması sürecidir. Bu işlemler; görüntüyü daha kullanışlı hale getirmek, insan gözünün göremediği ayrıntıları ortaya çıkarmak, görüntüdeki nesnelere tanımak ve sınıflandırmak gibi birçok amaçla kullanılmaktadır.

Görüntü işleme, birçok farklı disiplinden faydalanmaktadır. Örneğin; matematik, istatistik, bilgisayar bilimleri, fizik, optik ve psikoloji gibi disiplinlerin birleşiminden yararlanmaktadır. Görüntü işleme uygulamaları tıp, otomotiv, savunma, güvenlik, tarım, eğlence, üretim gibi birçok endüstride kullanılmaktadır.

Görüntü işleme algoritmaları kullanarak renk filtreleme, kenar algılama, resim büyütme, nesne takibi, yüz tanıma, karakter tanıma ve 3 boyutlu modelleme gibi teknikler ile görüntü işleme uygulamaları gerçekleştirilmektedir.

3.3.1. Görüntü işleme uygulamaları

Günümüzde yaygınlaşan teknoloji kullanımı ile görüntü işleme uygulamaları birçok alanda kendine yer bulmaktadır. Görüntü işleme uygulamaları birden fazla yöntem kullanarak farklı farklı alanlarda işleri kolaylaştıran, insan gözünden kaçabilecek önemli noktaları yakalayabilen, hata oranlarını düşüren ve bu sayede vazgeçilmez hale gelmeyi başarabilen

bir seviyeye ulaşmıştır. Bu kapsamda Görüntü işleme uygulamalarının yer aldığı alanlar incelenmiştir (Samtaş ve Gülesin, 2011).

Tıp: Bu alanda röntgen, MRI (Manyetik Rezonans Görüntüleme), tomografi gibi görüntüleme teknikleri kullanılmaktadır. Bu şekilde elde edilen görüntülerin analizi ile tıbbi teşhis ve tedavi süreçlerinde daha başarılı sonuçlar alınmaktadır.

Otomotiv: Bu alanda görüntü işleme, sürücü destek sistemleri ve otonom araç teknolojilerinin geliştirilmesinde kullanılmaktadır. Sürücü destek sistemi ile trafik kazalarının sayısında azalma hedeflenmektedir. Bunun yanında gidilecek yolu en kısa şekilde hesaplayan ve çizen çalışmalar da yapılmaktadır (Göçmen, 2018).

Güvenlik: Güvenlik sektörü görüntü işleme uygulamalarının en yaygın kullanıldığı alanların başında gelmektedir. Sektörün en önemli verisi video kayıtları olduğundan üzerinde görüntü işleme uygulamaları ile kişi tespiti, biyometrik kimlik doğrulama, kişi takibi gibi birçok çözüm geliştirilmektedir.

Tarım: Bu sektörde özellikle son yıllarda nitelikli eleman ve işçi bulma sıkıntısı, doğa olaylarının getirdiği olumsuzluklar düşünüldüğünde teknolojiye dayanarak elzem bir duruma dönüşmektedir. Görüntü işleme uygulamaları ile bitki hastalıklarının erken teşhisi, bitki büyüme analizi, verimlilik artırma, hasat tahmini çalışmalarının yanında otonom ilaçlama ve hasat çalışmaları yapılmaktadır.

Eğlence: Film, televizyon ve video oyunları gibi eğlence uygulamalarında, görüntü işleme teknikleri kullanılarak etkileşim artırılmaktadır. Özellikle son zamanlarda birçok alanda etkileşim alan sanal gerçeklik uygulamaları görüntü işleme uygulamalarının örnekleri görülmektedir.

Endüstriyel üretim: Görüntü işleme uygulamaları ile endüstriyel üretimde insanlar tarafından yapılan birçok uygulama artık görüntü işleme aracılığı ile robot kollarla yapılmaktadır. Örneğin üretim bantlarında hatalı ürünlerin tespit edilip ayrıştırılması insan denetimi yerine görüntü işleme uygulamaları gerçekleştirebilen robotik kollarla yapılmaktadır.

3.3.2. Görüntü işleme teknikleri

Görüntü işleme teknikleri, görüntüleri işlemek ve analiz etmek için kullanılan bir dizi yöntemi ifade etmektedir. Bu kapsamda bazı yaygın görüntü işleme teknikleri incelenmiştir (Gonzalez ve Woods), 2002: 221-807).

Sayısal filtreleme: Gürültüyü azaltmak, kenarları belirginleştirmek, görüntüleri düzeltmek veya diğer özelliklerini belirginleştirmek için bir dizi filtre kullanarak görüntüyü işleme yöntemidir.

Morfolojik işlemler: Nesnelerin şekillerini veya boyutlarını değiştirmek için kullanılan bir dizi işlemlerdir. Örnek olarak nesnelere üzerinde yapılan açma, kapanma ve erozyon işlemleri morfolojik işlemlerdir.

Görüntü segmentasyonu: Görüntüyü nesnelere veya bölümlere ayırmak için kullanılan bir dizi yöntemdir. Örnek olarak eşikleme, bölgesel büyütme veya bölgelerin birleştirilmesi segmentasyon işlemleridir.

Şablon eşleme: Görüntüdeki bir nesnenin özelliklerini kullanarak, benzer nesnelere bulmak için kullanılan bir yöntemdir. Örneğin, nesnelerin kenarları veya renkleri gibi özellikleri kullanılarak şablon eşleme ile benzer nesnelere tespit edilmektedir.

Öznitelik çıkarma: Görüntüdeki özellikleri tanımlamak için kullanılan bir yöntemdir. Nesnelerin özniteliklerine örnek olarak kenarlar, köşeler veya renkler gibi özellikler gösterilir.

Yüz tanıma: Görüntüdeki yüzleri tanımak için kullanılan bir yöntemdir. Bu yöntem, güvenlik uygulamaları ve sosyal medya uygulamaları gibi birçok alanda kullanılmaktadır.

Görüntü rekonstrüksiyonu: Görüntüden 3 boyutlu bir model oluşturmak için kullanılan bir dizi yöntemdir. Örneğin; stereo görüntüleme, lazer tarayıcı veya fotogrametri gibi teknikler kullanılarak yapılmaktadır.

Hedef takibi: Görüntüdeki bir nesneyi takip etmek için kullanılan bir yöntemdir. Örneğin, hareket takibi veya optik akış hesaplamaları gibi teknikler kullanılarak yapılmaktadır. Bu yöntem günümüzde en fazla sürücü destek sistemleri ve güvenlik uygulamalarında kullanılmaktadır.

3.4. Yapay Zeka

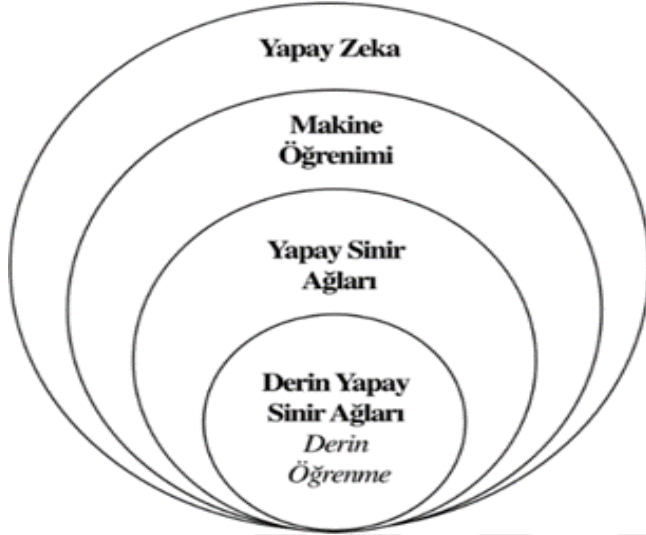
Yapay zeka (YZ) gelişim tarihçesi oldukça uzun bir süreçten geçmiştir. 1950'lerde, Alan Turing'in "Makineler Düşünebilir mi?" makalesi YZ araştırmalarının temelini oluşturmaktadır. İlk YZ çalışmaları, sembolik işlemci yaklaşımı ile yapılmıştır. Bu yaklaşımda, bir problem sembolik olarak ifade edilmiş ve bilgisayar tarafından çözülmeye çalışılmıştır. Fakat sembolik yaklaşımın sınırlamaları ve problemleri nedeniyle yapay zeka gelişmelerinde 1970'lerde bir durgunluk yaşanmıştır. 1980'lerde, YZ araştırmaları uzman sistemler gibi daha spesifik uygulamalarla yeniden canlanmıştır. Uzman sistemler, belirli bir uzmanlık alanında insan tarafından elde edilen bilginin bilgisayar programları aracılığıyla kullanılmasına dayanmaktadır (İnternet, 2022a).

1990'lar, istatistiksel yöntemler ve yapay sinir ağları ile yapılan çalışmaların popüler hale geldiği bir dönemdir. Bu yaklaşımların avantajlarına baktığımızda karşımıza özellikle doğal dil işleme, görüntü işleme ve otomatik öğrenme gibi karmaşık problemlerin çözümünde başarılı sonuçlar vermesi çıkmaktadır. Bu dönemde YZ alanında büyük ilerlemeler kaydedilmiş ve birçok uygulama geliştirilmiştir (İnternet, 2022a).

2000'lerde, büyük veri analizi ve makine öğrenmesi YZ araştırmalarında popüler hale gelmiştir. Bu yaklaşımlar, daha büyük veri kümelerinin işlenmesi ve daha karmaşık örüntülerin tanımlanması için kullanılan yöntemlerdir. YZ teknolojisinin gelişmesi, endüstrilerdeki birçok işlemin otomatik hale gelmesine olanak sağlamıştır. YZ teknolojisi, özellikle sağlık, finans, otomotiv, enerji, güvenlik, eğitim ve oyun endüstrilerinde yoğun bir şekilde kullanılmaktadır (İnternet, 2022a).

Günümüzde, YZ teknolojisi hızla gelişmeye devam etmektedir. Makine öğrenmesi ve derin öğrenme gibi teknikler, özellikle büyük veri analizinde kullanılarak insanların yapamayacağı kadar hızlı ve doğru sonuçlar sağlamaktadır. Günümüzde YZ teknolojisi,

sadece endüstrilerde değil, aynı zamanda kişisel cihazlarda da kullanılmaktadır. Örneğin, sesli asistanlar ve akıllı ev cihazları YZ teknolojisinin birer örneği olarak ortaya çıkmışlardır.



Şekil 3.1. Yapay zeka, makine öğrenmesi ve derin öğrenme ilişkisi

Şekil 3.1’de görüldüğü üzere Yapay zeka kümesi içerisinde sırası ile makine öğrenimi, yapay sinir ağları ve derin öğrenme ağları yer almaktadır (İnternet, 2017). Bu yapıların kullanımı, mimarileri ve aralarındaki farklılıklar ile ilgili detaylar aşağıdaki başlıklarda açıklanmaktadır.

3.4.1. Makine öğrenmesi

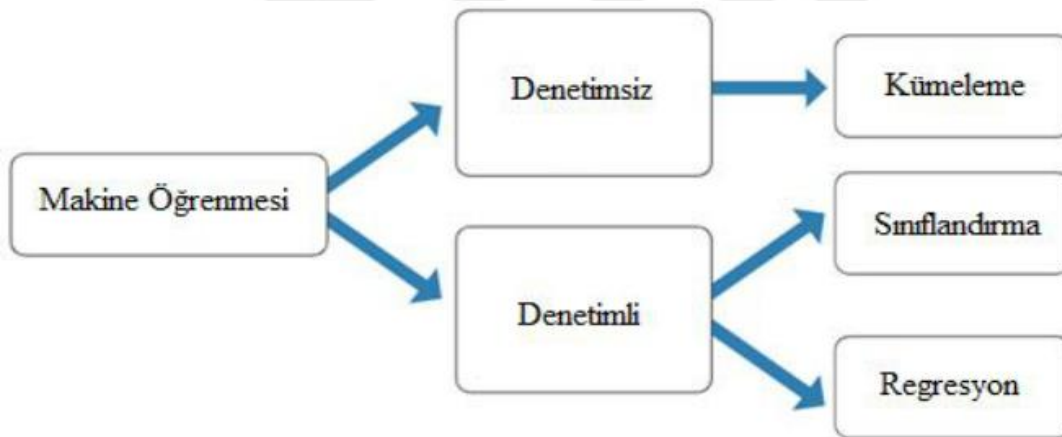
Makine öğrenmesi, bilgisayar sistemlerinin belirli bir görevi gerçekleştirmek için örnek veri veya deneyimler kullanarak kendilerini otomatik olarak geliştirdiği bir yapay zeka alt alanı olarak tanımlanmaktadır. Bu teknoloji; veri madenciliği, istatistiksel modelleme ve yapay zeka alanlarından yararlanarak bilgisayarların öğrenme yeteneklerini geliştirmektedir.

Makine öğrenmesi algoritmaları, verileri analiz ederek örüntüleri ve trendleri tanımlamak için kullanılmaktadır. Bu algoritmalar gelecekteki verileri tahmin etmek için modeller oluştururken deneyimleri, hataları ve başarısızlıkları kullanarak öğrenme işlemi gerçekleştirmektedir. Bahse konu bu tahminler ile müşteri davranışlarının analizi, görüntü

tanıma, ses tanıma, doğal dil işleme, oyun stratejileri geliştirme, arama motorları, araştırma ve öneri sistemleri gibi birçok alanda önemli gelişmeler yaşanmaktadır.

Makine öğrenmesi; büyük veri analizi, veri madenciliği ve istatistik gibi alanlarla yakından ilişkilidir. Yüksek ölçekte verilerin toplanması, işlenmesi ve analizi amacı ile makine öğrenmesi uygulamalarında genellikle paralel işlemciler, GPU'lar ve bulut bilişim hizmetleri gibi yüksek performanslı hesaplama sistemleri kullanılmaktadır.

Makine öğrenmesi teknikleri; veri özelliklerinin keşfi, sınıflandırma, regresyon, kümeleme ve boyut indirgeme gibi farklı görevler için kullanılabilen çeşitli algoritmalar içermektedir. Makine öğrenmesi modelleri arasında öğrenme yaklaşımlarına göre farklı öğrenme çeşitleri bulunmaktadır. Şekil 3.2'de görüldüğü üzere temelde öğrenme modelleri denetimli ve denetimsiz olarak iki grup altında kullanılmaktadır (Gökalp, 2022).



Şekil 3.2. Öğrenme yaklaşımına göre makine öğrenmesi modelleri

Denetimli Öğrenme: Bu teknik belirli bir çıktıya karşılık gelen girdi-veri örnekleri kullanarak bir model oluşturmaktadır. Bu veriler modelin öğrenmesinde kullanıldığı için denetimli öğrenme adını almaktadır. Sınıflandırma ve regresyon problemlerinin çözümünde sıklıkla kullanılmaktadır (Gökalp, 2022).

Denetimsiz Öğrenme: Bu teknik etiketlenmemiş verilerin analizi yoluyla bir model oluşturmaktadır. Bu verilerin doğal özelliklerini ve örüntülerini ortaya çıkarmak için

kullanılmaktadır. Kümeleme ve boyut azaltma gibi görevlerde tercih edilmektedir (Gökalp, 2022).

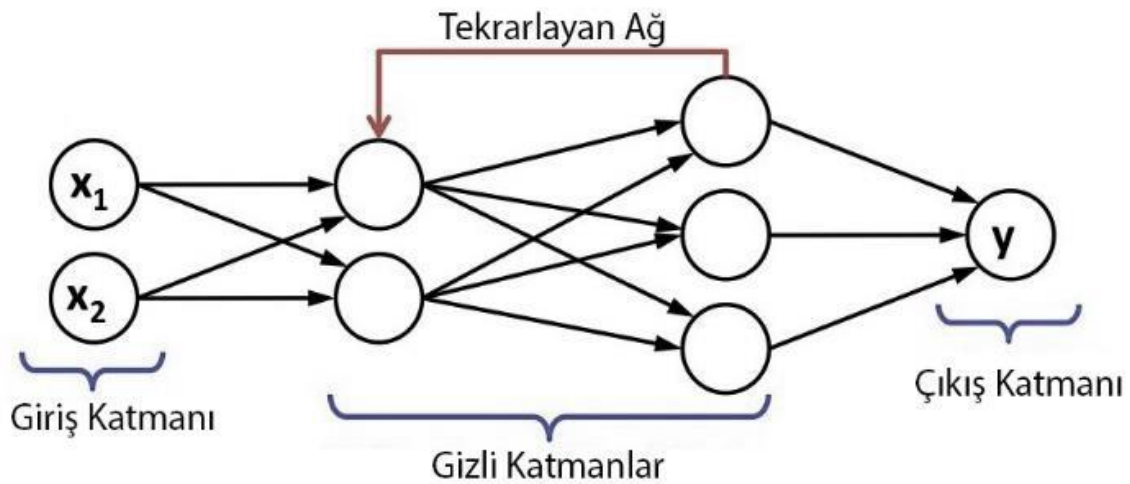
Bu kapsamda denetimli ve denetimsiz öğrenme metotlarının dışında problemlerin niteliklerinin çeşitlilik göstermesiyle farklı öğrenme metotları ortaya çıkmıştır.

Yarı Denetimli Öğrenme: Bu teknik hem etiketli hem de etiketlenmemiş verilerin kullanıldığı bir öğrenme yöntemidir. Modelin doğru bir şekilde öğrenmesini sağlamak için sınıflandırma ve regresyon problemlerinde az sayıda etiketlenmiş veri kullanılırken geniş ölçekte etiketlenmemiş veri kullanılmaktadır (Gökalp, 2022).

Pekiştirmeli Öğrenme: Bu teknikte bir öğrenme ajanı, çevresiyle etkileşim halinde bulunur ve bu etkileşimler sonucunda hangi eylemlerin daha iyi veya daha kötü sonuçlar doğurduğunu öğrenmektedir. Ajan bir durumda bulunur ve bu durumda hangi eylemi gerçekleştireceğine karar verir. Karar verme işlemi, ajanın bir ödül fonksiyonu tarafından belirlenmektedir. Eğer ajan iyi bir sonuç doğuracak bir eylem gerçekleştirirse ödül alır ayrıca ödülün büyüklüğü doğrudan eylemin kalitesiyle ilişkilidir. Eğer ajan kötü bir sonuç doğuracak bir eylem gerçekleştirirse ödül almadığı gibi yanlış bir eylem yaptığı için cezalandırılır. Bu şekilde ajan bir dizi deneyimle öğrenmekte ve deneyimlerini kullanarak gelecekteki kararlarını iyileştirmektedir. Bu teknik, özellikle robotik, oyun ve yapay zeka gibi alanlarda kullanılmaktadır (Gökalp, 2022).

3.4.2. Yapay Sinir Ağları

Yapay sinir ağları Şekil 3.3'te gösterildiği gibi birçok katmandan oluşmaktadır (Doğan ve Türkoğlu, 2019). Ağın ilk katmanı, girdi verilerini kabul etmekte ve bu verileri işlemek için gerekli özelliklerini çıkarmaktadır. İlk katmanda belirlenen bu özellikler bir sonraki katmana iletilmekte ve işlem aynı şekilde devam etmektedir. Her katman, girdileri işlemek için bir matris kullanmaktadır. Bu matrisler, ağın içindeki bağlantıların ağırlıklarını belirlemektedir. Başlangıçta bu ağırlıklar rastgele atanmakta sonrasında ise öğrenme süreci ile optimize edilmektedir.



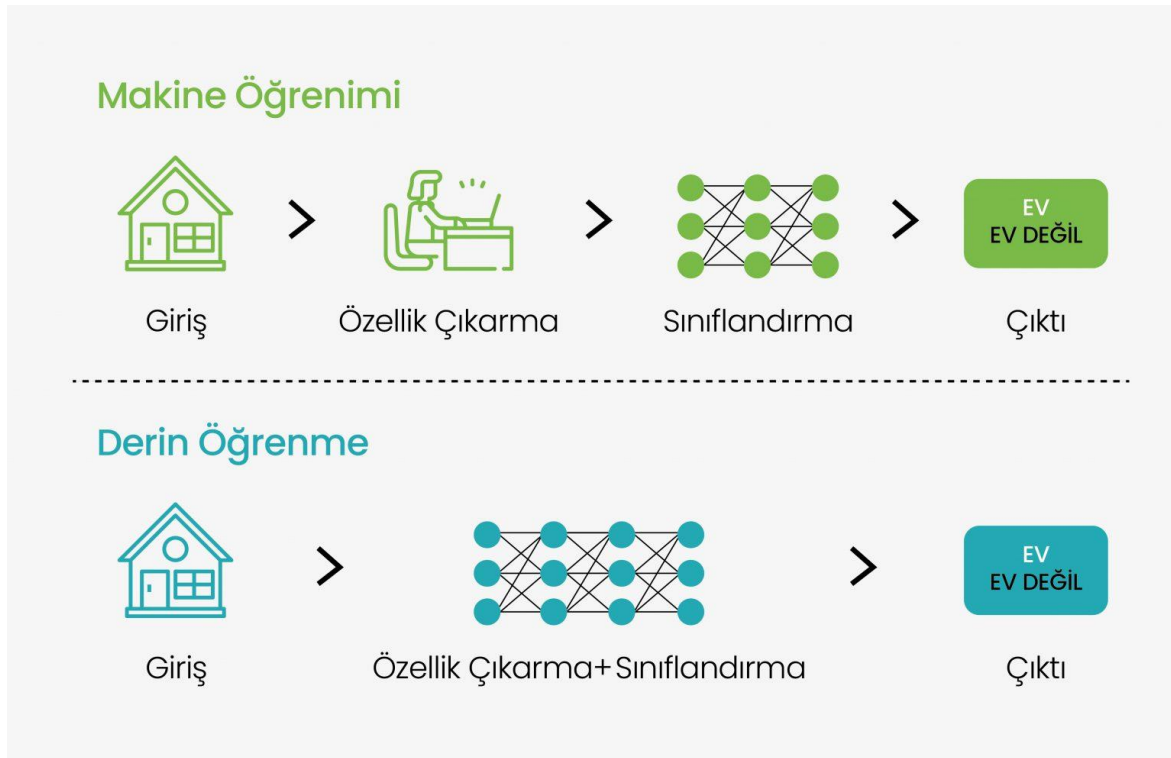
Şekil 3.3. Yapay sinir ağı modeli

Öğrenme süreci, ağın kayıp fonksiyonu aracılığıyla gerçekleştirilmektedir. Kayıp fonksiyonu, ağın tahminlerinin gerçek değerlerden ne kadar uzak olduğunu ölçmekte ve bu hatayı minimize etmek için geriye yayılım algoritması kullanmaktadır. Geriye yayılım işlemi ile hatalar geriye doğru katmanlara dağıtarak her katmanın ağırlıkları ve bağlantıları güncellenmektedir. Bu şekilde ağın çıktıları gerçek değerlere ne kadar yakın olursa, kayıp fonksiyonu o kadar düşük olmaktadır.

3.4.3. Derin Öğrenme

Derin öğrenme algoritmaları, makine öğrenmesinin bir alt dalıdır ve yapay sinir ağları aracılığıyla karmaşık görevleri yerine getiren algoritmaların oluşturulmasına odaklanmaktadır. Bu teknoloji; büyük veri kümelerini analiz etmek, ses tanıma, görüntü işleme, dil çevirisi ve otomatik sürüş gibi birçok uygulama için kullanılmaktadır.

Derin öğrenme mimarileri oluşturulurken kullanılacak verinin türü, miktarı, yapısı ve kullanılacak parametreler büyük önem taşımaktadır. Bu kapsamda oluşturulan derin öğrenme algoritmalarından en sık kullanılanlar evrimsel sinir ağları, tekrarlayan sinir ağları, kısıtlı boltzmann makineleri ve derin inanç ağları olarak karşımıza çıkmaktadır.



Şekil 3.4. Makine öğrenmesi ve derin öğrenme genel çalışma prensibi

Makine öğrenmesi ve derin öğrenme algoritmalarının temel çalışma mantığı Şekil 3.4'te gösterilmektedir (İnternet, 2021a). Bu kapsamda makine öğrenmesi ve derin öğrenme arasındaki teknik farklılıklar, temelde algoritmaların yapıları ve verilerin nasıl işlendiği ile ilgilidir. Makine öğrenmesi algoritmaları, belirli bir sonuca ulaşmak için veriler arasındaki desenleri tanımaya çalışırken derin öğrenme algoritmaları daha karmaşık yapay sinir ağları kullanarak veriler arasındaki daha derin desenleri tanımaya çalışır.

Makine öğrenmesi algoritmaları genellikle sınıflandırma, regresyon, kümeleme, boyut azaltma ve örüntü tanıma gibi temel işlemleri gerçekleştirir. Bu işlemler, veriler arasındaki ilişkileri ve özellikleri tanımlamak için kullanılır. Daha sonra gelecek tahminleri yapmak için bu bilgiyi kullanır. Makine öğrenmesi algoritmaları, doğrusal ve doğrusal olmayan modeller kullanarak verileri işleyerek sonuçları yorumlar.

Derin öğrenme algoritmaları, daha karmaşık ve çok katmanlı yapay sinir ağları kullanarak veriler arasındaki daha derin desenleri tanımaya çalışır. Bu ağlar, verileri çok sayıda katmana ayırarak her katmanda daha yüksek seviyede özellikler çıkartarak işlem yaparlar.

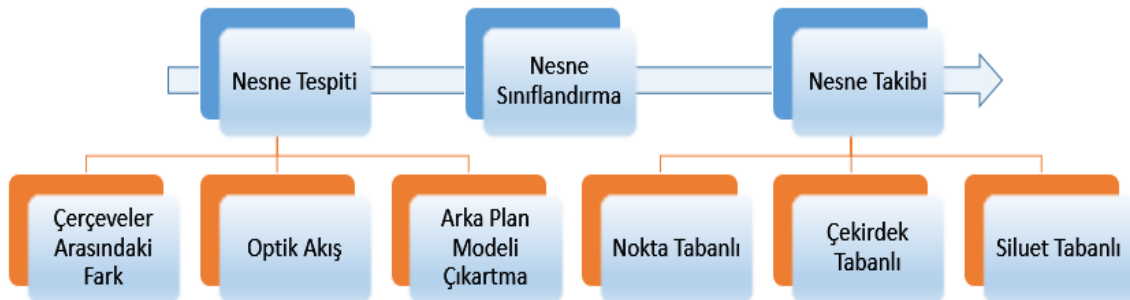
Bu sayede derin öğrenme algoritmaları daha karmaşık işlemleri gerçekleştirmekte ve daha yüksek doğruluk oranları elde etmektedir.

Sonuç olarak makine öğrenmesi algoritmaları daha basit ve doğrusal işlemleri, derin öğrenme algoritmaları ise daha karmaşık işlemleri ve desenleri çözebilir. Bu nedenle derin öğrenme özellikle görüntü işleme, doğal dil işleme ve ses tanıma gibi alanlarda daha yaygın olarak kullanılmaktadır.

3.5. Gerçek Zamanlı Nesne Tespiti ve Takibi

Gerçek zamanlı nesne tespiti ve takibi, bilgisayarlı görü ve yapay zekâ alanlarında son derece önemli bir konudur. Bu teknoloji özellikle otomotiv, güvenlik, endüstriyel otomasyon ve sağlık gibi birçok sektörde yaygın bir şekilde kullanılmaktadır. Gerçek zamanlı nesne tespiti ve takibi 1990'lardan bu yana aktif bir araştırma konusudur.

1990'larda, nesne tespiti ve takibi için çeşitli yaklaşımlar geliştirilmiştir. İlk sistemler sabit bir kamera kullanarak nesne tanıma ve takip etme işlemini gerçekleştirmekteydi. Sabit kameralı sistemler nesnelerin hareketlerini takip etmek için yeterli olmamış ve hareketli kamera sistemlerine geçilmiştir. Bu sayede nesnelerin hareketlerinin takibi daha etkili bir şekilde gerçekleştirilmiştir. 2000'li yıllara gelindiğinde gelişen derin öğrenme teknikleriyle birlikte gerçek zamanlı nesne tespiti ve takibi alanında büyük bir gelişme yaşanmıştır. Yapay sinir ağları, özellikle konvolüsyonel sinir ağları (CNN), nesne tespiti ve takibinde kullanılan en etkili yöntemlerden biri olmuştur. CNN'ler özellikle hızlı ve doğru sonuçlar üretmeleri nedeniyle gerçek zamanlı uygulamalarda yaygın olarak kullanılmaktadır.



Şekil 3.5. Nesne tespiti ve takibi yöntemleri

Nesne tespiti ve takibi yöntemleri Şekil 3.5'te gösterilmektedir (Üzen, 2018). Gerçek zamanlı nesne tespiti ve takibi modelleri için genel bir yol göstermek açısından gerekli aşamalar incelendiğinde karşımıza aşağıdaki başlıklar çıkmaktadır.

Veri Hazırlama: Bu adımı veri toplama ve veri ön işleme olarak iki başlığa ayırabiliriz. Gerçek zamanlı nesne tespiti ve takibi için gerekli veriler genellikle kamera, LIDAR veya radar gibi bir sensörden gelen verilerdir. Toplanan bu veriler veri ön işleme adımında daha yüksek performans ve daha başarılı sonuçlar için işlenebilir hale getirilmektedir. Örnek olarak kameradan alınan verilerin boyutu ayarlanmakta, renk skalası ve kontrastı düzenlenmektedir.

Nesne Tespiti: Nesnelerin sınıfını, konumunu ve boyutunu belirlemeyi içeren bir işlemdir. Gerçek zamanlı nesne tespiti için kullanılacak birçok farklı nesne tespit algoritması bulunmaktadır. Bu noktada yapılacak işe göre en uygun algoritmayı seçmek önem arz etmektedir. Örnek olarak YOLO, SSD, Faster R-CNN ve Mask R-CNN gibi algoritmalar yaygın olarak kullanılmaktadır. Algoritma seçiminden sonra mevcut veri seti ile eğitim aşaması gelmektedir. Eğitim verisinin doğru bir şekilde etiketlenmesi eğitim başarımı için önemlidir. Bu işlemler gerçekleştirildikten sonra elde edilen eğitilmiş model gerçek zamanlı nesne tespiti için kullanılmaktadır.

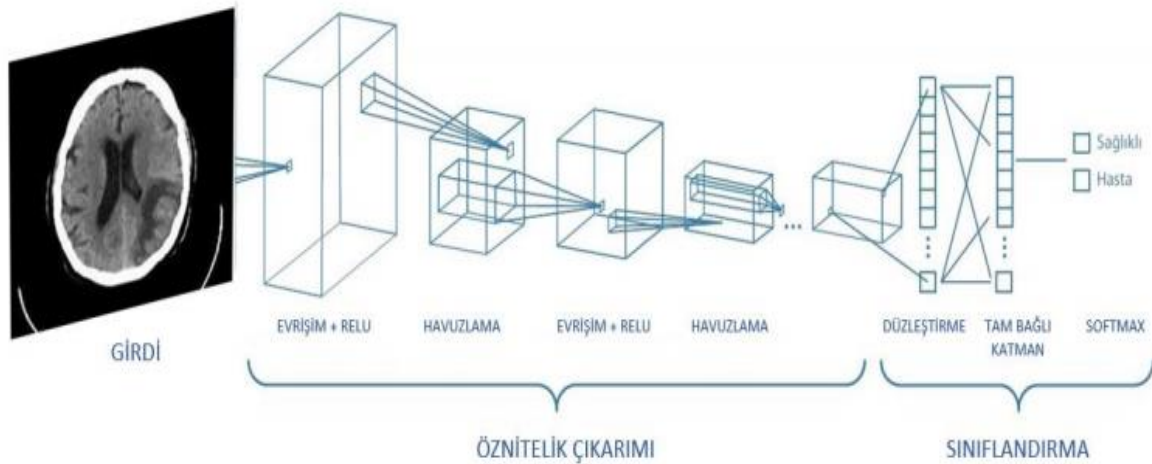
Nesne Takibi: Nesne takibi bir nesnenin hareketini takip etmek için kullanılan bir işlemdir. Takip algoritmaları, önceden tespit edilen nesnelerin konumlarını ve özelliklerini kullanarak yeni konumlarını tahmin etmektedirler. Takip algoritmaları, nesnelerin birbirine benzer olduğu durumlarda bile doğru sonuçlar üretmek için tasarlanmıştır. Nesne takibi için geliştirilen yaklaşımlar nokta tabanlı, çekirdek tabanlı ve siluet tabanlı yaklaşımlar olarak sınıflandırılır.

Nesne Sınıflandırması: Nesne sınıflandırması, belirli nesnelerin sınıflara ayrılmasıdır. Bu sınıflandırma, daha sonra nesnenin davranışının belirlenmesi veya diğer sistemlerle entegrasyonu için kullanılmaktadır. Nesne sınıflandırması, nesne tespitinde kullanılan makine öğrenimi teknikleri ve derin öğrenme algoritmaları ile gerçekleştirilmektedir.

Sonuç Çıkarma: Nesne tespiti ve takibi sonuçları bir çıktı olarak kullanılabilir. Örneğin bir araç takip sistemi için nesne tespiti ve takibi sonuçları; araçların konumunu, hızını ve yönünü tahmin etmek için kullanılabilir. Sonuç çıktısı genellikle diğer sistemlerle entegrasyon için bir arayüzle sunulmaktadır.

3.5.1. Evrişimli sinir ağları

Evrişimli sinir ağları, yapay sinir ağı mimarileri ile özellikle görüntü işleme, ses tanıma ve doğal dil işleme gibi alanlarda kullanılmaktadır. Evrişimli sinir ağları, özellikle görüntü işleme alanında son derece etkili bir şekilde çalışmakta ve birçok endüstriyel uygulama için yaygın bir şekilde kullanılmaktadır. Sinir ağına girdi olarak verilen görüntüler analiz edilir ve görüntüdeki özellikleri tanımlayan filtreler kullanarak özellik haritaları oluşturulur. Bu filtreler kullanılarak görüntülerin kenarları, çizgileri, desenleri ve diğer özellikleri tespit edilmektedir. Daha sonra bir veya daha fazla gizli katmanda, bu özellik haritaları bir araya getirilerek nesne sınıflarının belirlenmesi için sonuçlar oluşturulmaktadır. Bu yapılarda filtreler ve katmanlar genellikle önceden eğitilmiştir. Bu eğitim büyük veri kümeleri kullanarak gerçekleştirilmekte ve genellikle GPU'lar tarafından hızlandırılmaktadır. Eğitim sonucunda filtreler nesnelere tanımlamak için belirli özellikleri tespit etmekte ve model verileri analiz ederek sınıflandırmak için daha kabiliyetli bir yapıya kavuşmaktadır.



Şekil 3.6. Evrişimli sinir ağının yapısı

Evrişimli sinir ağları genellikle birçok evrişimli katman içeren derin öğrenme modelleridir. Evrişimli sinir ağı katmanları Şekil 3.6'da gösterilmektedir (İnternet, 2021b).

Girdi katmanı: Görüntü verileri gibi modelin alacağı girdilerin şekli bu katmanda tanımlanmaktadır.

Evrişimli katmanlar: Bu katmanlar, özellik haritalarının oluşturulduğu kısımlardır. Girdi verileri, filtre adı verilen bir dizi matrisle çarpılarak görüntüdeki belirli özellikler (kenarlar, çizgiler, desenler vb.) vurgulanmaktadır. Bu işlem tüm veriler için tekrarlanarak her bir özellik için ayrı bir özellik haritası oluşturulmaktadır.

Aktivasyon fonksiyonu katmanı: Bu katman, evrişimli katmanlar tarafından elde edilen özellik haritalarını bir aktivasyon fonksiyonu aracılığıyla işlemektedir. Bu sistem nöronların çıkışını sınırlayarak ağı daha fazla öğrenmesini sağlamakta ve modelin daha doğru sonuçlar vermesine katkı sunmaktadır.

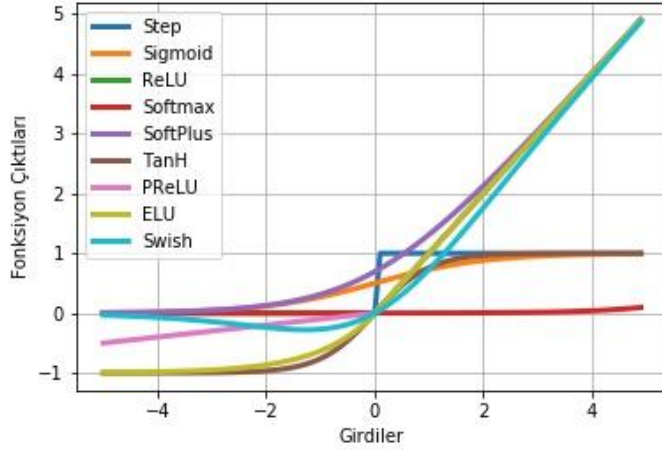
Havuzlama katmanı: Bu katman, özellik haritalarını küçültmek ve daha sonra daha yüksek seviyeli özellikleri tespit etmek için tasarlanmıştır. Özellik haritalarının boyutunu küçültmek için çeşitli matematiksel işlemleri kullanmaktadır.

Düzleştirme Katmanı: Matris yapısında gelen verileri tek boyutlu yapıya dönüştürerek sınıflandırma katmanı için kullanışlı hale getirmektedir. Hemen hemen tüm evrişimli sinir ağı modellerinde bulunmaktadır.

Tam bağımlı katman: Bu katmanlar, ağı sınıflandırma işlemini gerçekleştirdiği yer olarak tanımlanmaktadır. Tüm özellikleri bir araya getirerek, nihai bir sınıflandırma sonucu elde edilmektedir.

Çıkış katmanı: Bu katmanda özellikle sınıflandırma sonuçları belirtilmektedir. Bu işlem için genellikle çok sınıflı yapılarda softmax fonksiyonu kullanılmaktadır. Bu fonksiyon sınıflar arasında olasılık dağılımı oluşturarak her sınıf çıktısı için olasılık değeri döndürmektedir. Bu noktada Şekil 3.7'de görüldüğü üzere Sigmoid, ReLU (Doğrultulmuş

Lineer Birim), Tanh (Hiperbolik Tanjant) gibi farklı fonksiyon seçenekleri de mevcuttur (İnternet, 2018).



Şekil 3.7. Aktivasyon fonksiyonları

Sigmoid: İki sınıflı sınıflandırma problemleri için kullanılır. Sigmoid fonksiyonu, herhangi bir gerçel sayıyı $[0,1]$ aralığındaki bir olasılık değerine dönüştürür. Sigmoid fonksiyonunun dezavantajı gradyanın küçük olması sebebi ile eğitim sırasında gradyanın kaybolması sorunu ortaya çıkabilmektedir (İnternet, 2018).

ReLU: Bu fonksiyon, CNN'lerde en sık kullanılan aktivasyon fonksiyonudur. ReLU, negatif değerleri sıfıra eşitleyerek non-lineerliği sağlar. Sigmoid fonksiyonuna göre gradyan problemi yaşanmamasını ve ağıın daha iyi öğrenmesini sağlamaktadır (İnternet, 2018).

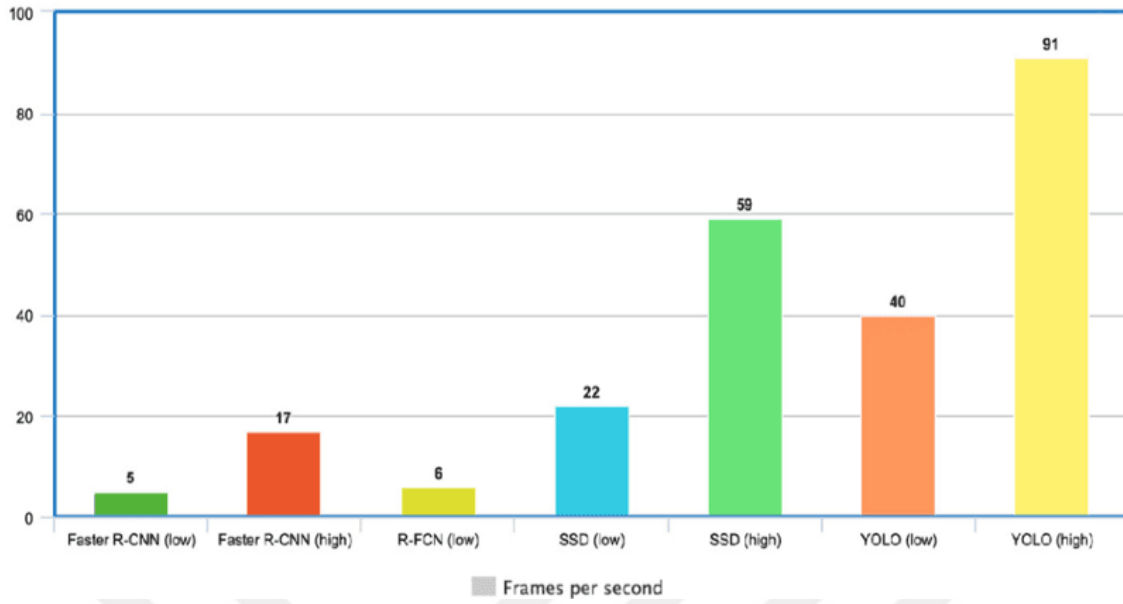
Tanh: Bu fonksiyon sigmoid fonksiyonuna benzemekle beraber farkı çıktı aralığının $[-1, 1]$ arasında olmasıdır. Bu fonksiyon sigmoid fonksiyonuna benzer şekilde gradyanın küçük olması sorununa sahiptir ve bu yüzden gradyan kayıpları yaşanmaktadır (İnternet, 2018).

Softmax: Bu fonksiyon, çoklu sınıflandırma problemleri için kullanılır. Bu fonksiyon, girdileri bir olasılık dağılımına dönüştürür böylece her sınıf için olasılık değerleri hesaplanabilir. Bu işlemde bir sınıfın olasılığı artarken diğer sınıfların olasılıklarının azalması nedeniyle sınıflar arasında rekabet oluşmasına neden olur ve modelin doğruluğunu artırır (İnternet, 2018).

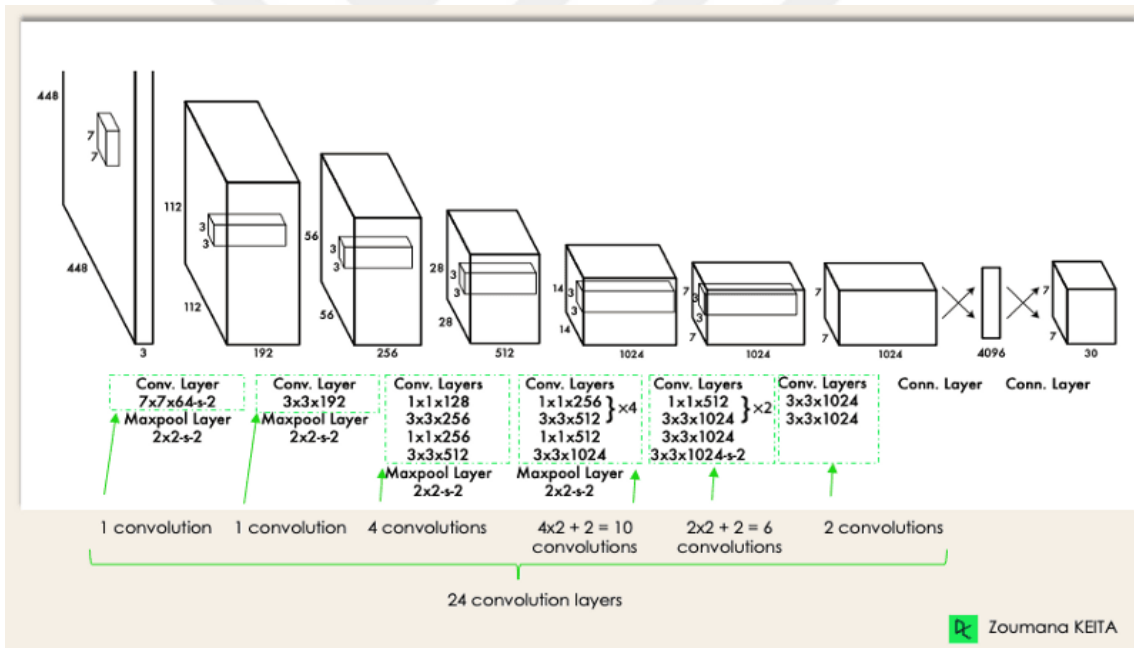
3.5.2. YOLO algoritması

Gerçek zamanlı nesne tespiti için en yaygın kullanılan yöntemler derin öğrenme tabanlı yaklaşımlardır. Bu yaklaşımlara örnekler R-CNN, Fast R-CNN, Mask R-CNN, SSD ve YOLO algoritmaları olarak gösterilebilir. Bu kapsamda bu algoritmaların her biri kendisinden önce tasarlanan algoritmanın yaşadığı bir sıkıntıyı çözmek üstüne tasarlanmıştır. Örneğin R-CNN bölge bazlı nesne tespiti yapmaktadır. Nesnelerin bulunduğu bölgeleri tespit ettikten sonra her bir bölge için ayrı ayrı sınıflandırma işlemi gerçekleştirilmektedir. Bu sebeple her görüntüye iki ayrı işlem uygulamakta ve işlem sayısı artarak hız azalmaktadır. Bunu çözmek için Fast R-CNN algoritması tasarlanmış ve tüm resmi tek seferde sinir ağından geçirmektedir. İşlem sayısını azaltarak daha hızlı sonuçlar üretmiştir. Fast R-CNN gerçek zamanlı uygulamalar için değerlendirildiğinde düşük bir FPS (saniyede kare sayısı) ortalamasına sahip olduğu görülmüş ve başarılı sonuçlar elde edilememiştir.

Gerçek zamanlı nesne tespitinde uygulanabilirlik açısından hem yüksek doğruluk oranı hem yüksek FPS oranına ihtiyaç vardır. Bu noktada 2015 yılında Joseph Redmon, Santosh Divvala, Ross Girshick ve Ali Farhadi tarafından “You Only Look Once: Unified, Real-Time Object Detection” isimli araştırma makalesi yayınlanarak YOLO (Yalnızca Bir Kez Bak) mimarisi ortaya çıkarılmıştır. YOLO mimarisinde nesne algılama problemi bir regresyon problemi olarak değerlendirilmektedir. Diğer algoritmalara göre en önemli özelliği Şekil 3.8’de görüldüğü üzere gerçek zamanlı uygulamalardaki hızıdır (İnternet, 2022b).



Şekil 3.8. Gerçek zamanlı uygulamalarda oransal hız karşılaştırması



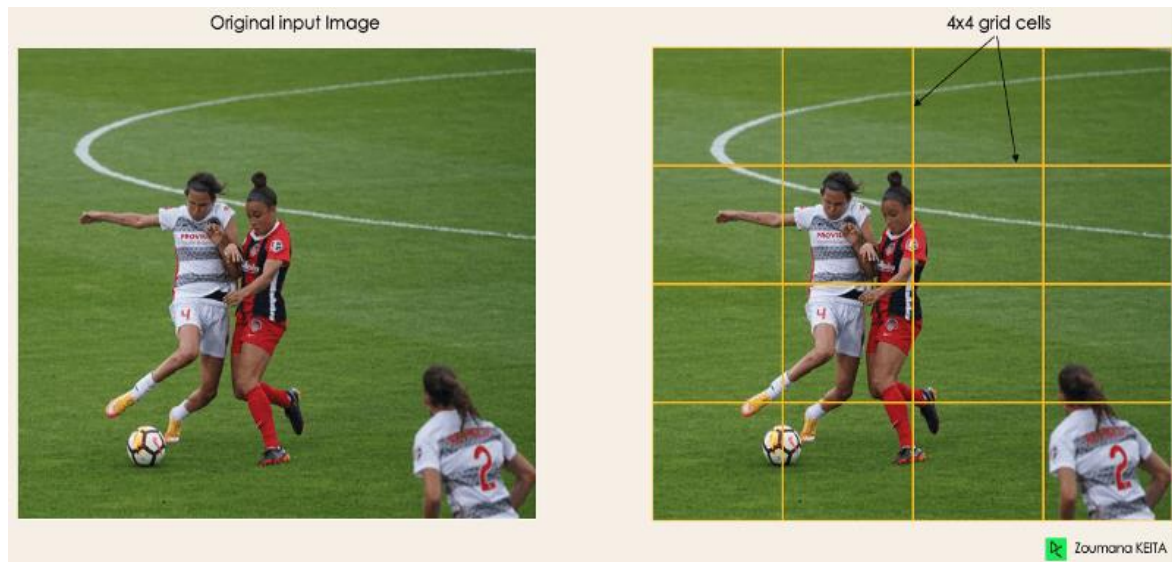
Şekil 3.9. YOLO mimarisi

YOLO algoritmasının mimarisi Şekil 3.9’da gösterilmektedir. Mimaride 24 evrişimli katman, dört adet havuzlama katmanı ve iki adet sınıflandırma katmanı bulunmaktadır. Evrişimli ağdan geçmeden önce giriş görüntüsünü 448x448 olarak yeniden boyutlandırılmaktadır. Kanal sayısını azaltmak için önce 1x1’lik bir evrişim uygulanmaktadır. Daha sonra kübik bir çıktı oluşturmak için 3x3’lük bir evrişim uygulanmaktadır. Aktivasyon fonksiyonu, doğrusal aktivasyon fonksiyonu kullanan son

katman dışında ReLU'dur. Toplu normalleştirme gibi bazı ek teknikler kullanılarak model düzenli hale getirilmekte ve fazla uydurmasının önüne geçilmektedir (İnternet, 2022b).

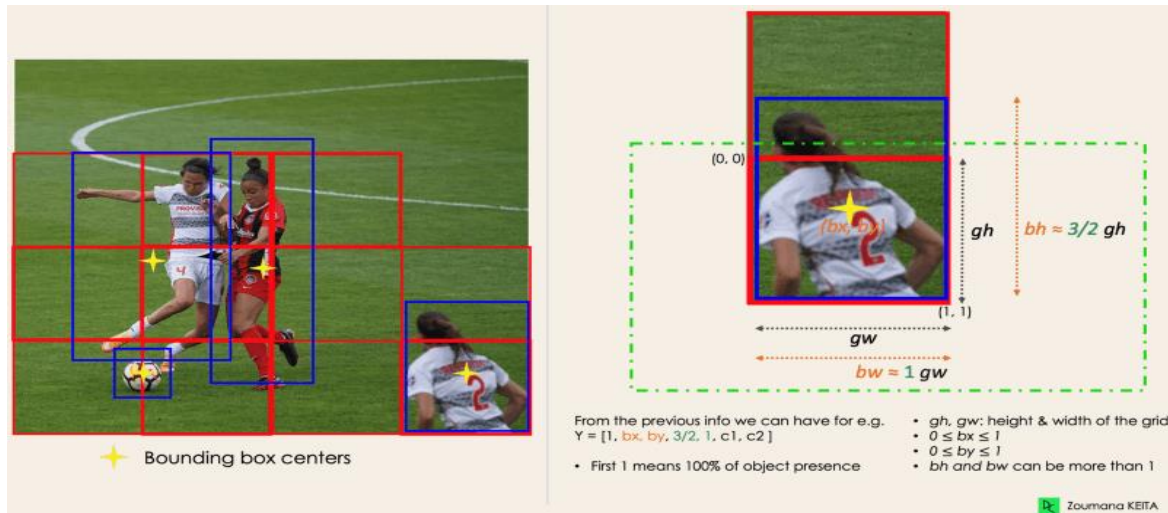
YOLO algoritmasının nesne algılama işlemini nasıl yaptığını incelediğimizde karşımıza dört başlık çıkmaktadır.

Blokların oluşturulması: Nesne algılamanın ilk aşamasında giriş görüntüsü Resim 3.2'de gösterildiği gibi eş büyüklükteki ızgaralara bölünür. Her bir ızgara hücresindeki nesnelerin sınıfından ve güven endeksinden sorumludur (İnternet, 2022b).



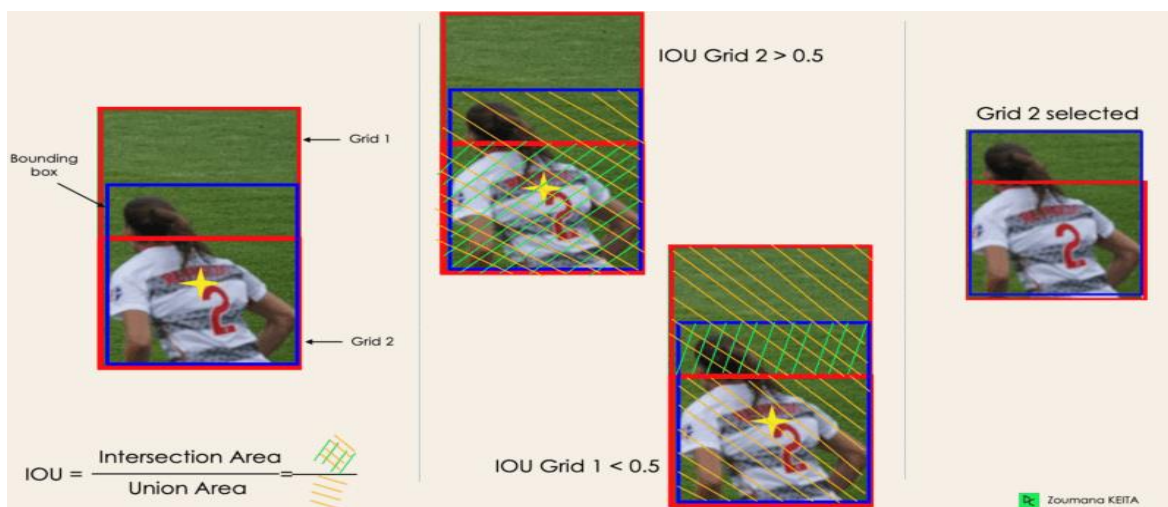
Resim 3.2. Giriş resminin ızgaralara bölünmesi

Sınır kutularının çizilmesi: Izgaraları çizilen giriş görüntüsü üzerindeki her bir nesne için sınır kutuları çizilecektir. Bunun için algoritmada her sınırlayıcı kutu için $Y = [pc, bx, by, bh, bw, c1, c2]$ şeklinde bir vektör tanımlanmaktadır. Bu vektörde pc: ızgara içinde nesne var (1) ve yok (0) şeklinde tanımlanır, bx ve by: sınırlayıcı kutunun merkez koordinatları, bh ve bw: sınırlayıcı kutunun eni ve boyunu, c1 ve c2: resim içerisindeki nesnelerin sınıflarını tanımlamaktadır. Nesne sınıfları kısmı tespit edilen nesne sınıfı kadar uzatılmaktadır. Resim 3.3'te giriş resminin orta noktaları işaretlenmiş ve sınır kutuları çizdirilmiş şekli görülmektedir. Ayrıca pc:0 olan yani ızgara içerisinde nesne bulunmayan ızgaralarında silindiği görülmektedir (İnternet, 2022b).



Resim 3.3. Sınır kutularının çizdirilmesi

Birleşmeler üzerinde kesişme (Intersection over unions - IOU): Görüntüdeki nesne iki veya daha fazla ızgara içerisinde bulunuyorsa bu yöntemden faydalanarak bahse konu nesne için sorumlu ızgara belirlenmektedir. Resim 3.4'te görüldüğü üzere iki numaralı forması olan oyuncu yani giriş görüntüsünün en sağ alt kısmındaki oyuncu için iki ızgarada bulunma söz konusudur. Bu noktada IOU hesabı yapılır. Bu hesap nesnenin bulunduğu her bir ızgara için ayrı ayrı nesne sınır kutusunun ızgara alanı ile kesiştiği alanın ızgara alanına oranı olarak 0-1 arasında bir değer olarak bulunur. Model için önceden atanan örneğin 0,5 gibi bir IOU oranı varsa bu değer altındaki ızgara elenirken üzerinde kalan ızgara o nesneden sorumlu olmaktadır (İnternet, 2022b).



Resim 3.4. IOU hesabı ve kıyaslanması

Maksimum olmayanı bastırma (Non-Max Suppression - NMS): Görüntüdeki nesnelere için birden fazla sınırlar kutusu çizdirilir. Bu sebeple en doğru sınırlar kutusunu elde edebilmek için NMS kullanılmaktadır. Bu yöntemde öncelikle güven skoru hesaplanır. Güven skoru = $P_c \times IOU$ şeklinde tanımlanmaktadır. Bu değerin hesabından sonra belirli bir eşik değerinin altındaki kalanlar otomatik olarak elenirken üzerinde kalanlar arasında en yüksek skora sahip olan ile devam edilmektedir (İnternet, 2022b).

3.6. Performans Ölçümü

Nesne tespiti ve sınıflandırma problemlerinin sonuçlarını değerlendirmek, başarımlar oranlarını tespit edebilmek için eğitilen her bir model için hata (karışıklık) matrisi kullanılır. Hata matrisi Çizelge 3.1'de gösterildiği gibi TP, FP, FN, TN olarak isimlendirilen değerlerden oluşmaktadır (Dersuneli, Gündüz ve Kutlu, 2021).

TP (True Pozitif): Sınıfa ait doğru tahminler

FP (False Pozitif): Sınıfa ait yanlış tahminler

FN (False Negatif): Sınıfa ait yanlış yapılan negatif tahminler

TN (True Negatif): Sınıfa ait doğru yapılan negatif tahminler

Çizelge 3.1. Hata (karışıklık) matrisi

Tahmin Edilen Sınıf			
Pozitif	Negatif		
TP	FN	Pozitif	Gerçek Sınıf
FP	TN	Negatif	

IOU değerinin nasıl hesaplandığından ve standart olarak 0,5 olarak atandığından daha önce bahsetmiştik. Bu kapsamda belirlenen bu standart değer üzerinde kalan kutu tahminlerine TP ataması, altında kalan ve elimine olan kutu tahminlerinde FP ataması yapılacaktır. Bunların dışında nesne için kutu olmasına rağmen sınırlayıcı kutu çizilmiyorsa FN, geriye kalan tüm arka planlar için ise TN ataması yapılmaktadır.

Hata matrisi üzerinden elde edilen deęerler kullanılarak mAP (ortalama kesinlik deęeri), hassasiyet, özgünlük ve genel doęruluk performans ölçütleri hesaplanmaktadır. Hassasiyet Eş. 3.1’de görüldüğü üzere doęru tahminlerin tüm tahminlere oranı olarak belirtilmektedir. Özgünlük Eş. 3.2’de görüldüğü üzere tahmin edilen nesne sınıfının hangi oranda doęru yapıldığını belirtmektedir. Genel doęruluk Eş. 3.3’te görüldüğü üzere doęru yapılan nesne tahmininin dięerlerine oranı olarak belirtilmektedir. Ortalama kesinlik deęeri hesabı ise Eş. 3.4’te görüldüğü üzere ortalama kesinlik deęerlerinin toplamının sınıf sayısına bölünmesi ile elde edilmektedir (Yayık ve Kutlu, 2012).

$$Hassasiyet = \frac{TP}{TP+FP} \quad (3.1)$$

$$Özgünlük = \frac{TP}{TP+FN} \quad (3.2)$$

$$Genel Doęruluk = \frac{TP+TN}{TP+FP+TN+FN} \quad (3.3)$$

$$mAP = \frac{\text{Ortalama kesinlik deęeleri toplamı}}{\text{Toplam sınıf sayısı}} \quad (3.4)$$

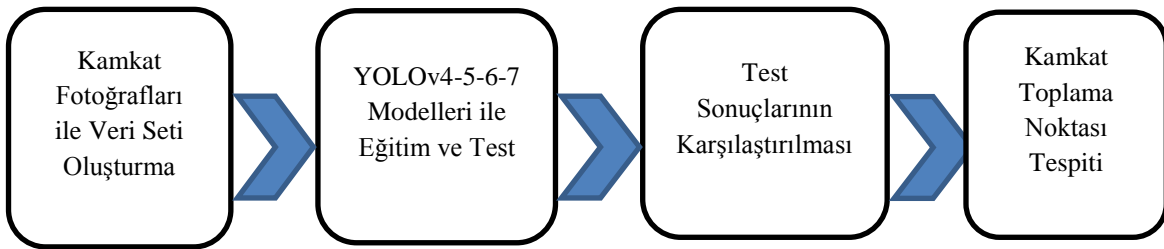
4. ARAŞTIRMA BULGULARI VE SONUÇLAR

4.1. Derin Öğrenme Tabanlı Kamkat Meyvesi Otonom Hasat Uygulaması

Bu kısımda derin öğrenme algoritmalarında kullanacağımız veri setinin oluşturulmasından başlanarak kamkat meyvesinin otonom hasadı için gereken tüm bilgisayarlı görü çalışmalarının dahil olduğu süreç ele alınacaktır. Bu kapsamda gerçek zamanlı nesne tespiti için YOLO algoritmalarında eğitim ve test işlemleri gerçekleştirilecektir. Uygun modelin tespiti ve bu model eğitim sonucu üzerinde Python ile opencv kütüphanesi kullanılarak kamkat meyvesi için en uygun toplama noktasının tespit işlemi gerçekleştirilecektir.

Bu tez çalışmasında gerçek zamanlı nesne tespiti için YOLO algoritmasının YOLOv4, YOLOv5, YOLOv6 ve YOLOv7 modelleri ile kendi veri setimize göre uygun konfigürasyon ayarları yapılarak eğitim ve test işlemleri gerçekleştirilmiştir. Bu sonuçlar kıyaslanarak gerçek zamanlı bir hasat uygulaması için en uygun modele karar verilecektir.

Son aşamada ise belirlenen YOLO modeli ile nesne tespiti yapıp bunun üzerinden hasat için en uygun toplama noktasının tespiti yapılacaktır. Yapılan çalışmalar ile ilgili sistem ilerlemesi akış diyagramı Şekil 4.1’de gösterilmektedir.



Şekil 4.1. Sistem İlerlemesi Akış Diyagramı

4.2. Veri Setinin Oluşturulması

Kamkat fideleri üzerinden farklı açı ve ışık durumları altında 220 adet fotoğraf çekilerek ilk veri seti oluşturulmuştur. Veri setinden bazı örnekler Resim 4.1’de gösterilmektedir.



Resim 4.1. Kamkat veri setinden örnekler

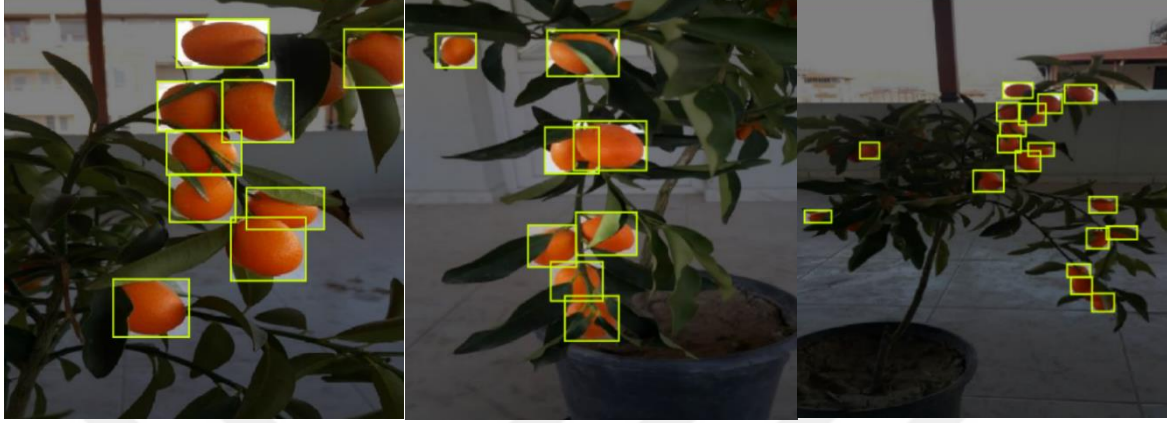
Bahse konu veri seti üzerinden rastgele seçilen bazı fotoğraflar kopyalanıp üzerlerine tuz biber gürültüsü ekleme ve bulanıklaştırma veri artırım yöntemleri uygulanmıştır. Bu uygulamadaki amaç hem veri setini büyütmek hem de gerçek zamanlı bir uygulamada oluşabilecek gürültülü görüntüler üzerinden de tespit yapmaya hazırlıklı olmaktır. Sonuç olarak 312 adet fotoğraf üzerinde 2965 adet kamkat meyvesi içeren bir veri seti oluşturulmuştur. Veri setindeki gürültü eklenmiş bazı görüntü örnekleri Resim 4.2’de yer almaktadır.



Resim 4.2. Gürültü eklenmiş kamkat görüntüleri

Veri setinin oluşturulmasından sonra sıra etiketleme işlemine gelmektedir. Kullanacağımız derin öğrenme algoritma modelleri YOLO mimarileri olduğu için etiketleme işlem sonucunun buna göre oluşturulması gerekmektedir. Bu işlem için “makesense.ai” isimli bir

web sitesi kullanılmıştır. Etiketleme işlemi sonucu oluşan örnek görseller Resim 4.3'te gösterilmektedir.



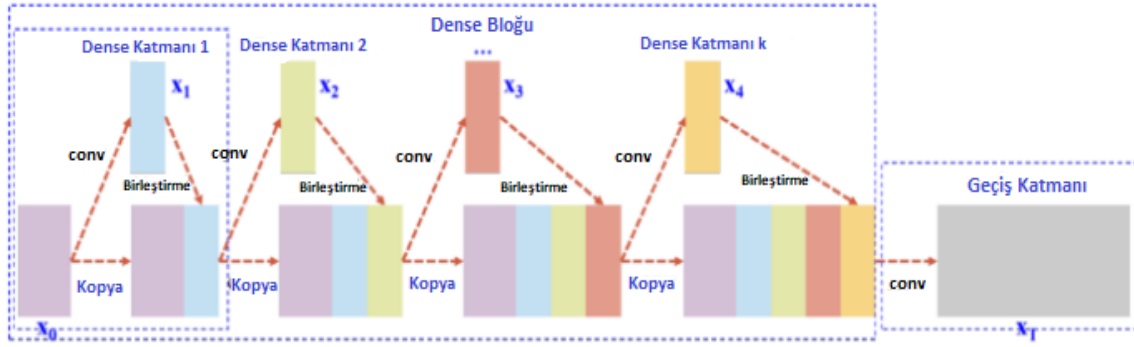
Resim 4.3. Kamkat etiketleme örnekleri

4.3. YOLO Modellerinin Eğitim ve Test İşlemleri

Bu kısımda gerçek zamanlı nesne tespiti için YOLO derin öğrenme algoritmasının çeşitli versiyonlarında kamkat veri seti için eğitim ve test işlemleri gerçekleştirilecektir. Test sonucunda elde edilen veriler karşılaştırılarak otonom hasat robotu için en uygun model tespit edilerek sistem üzerinde kullanılacaktır.

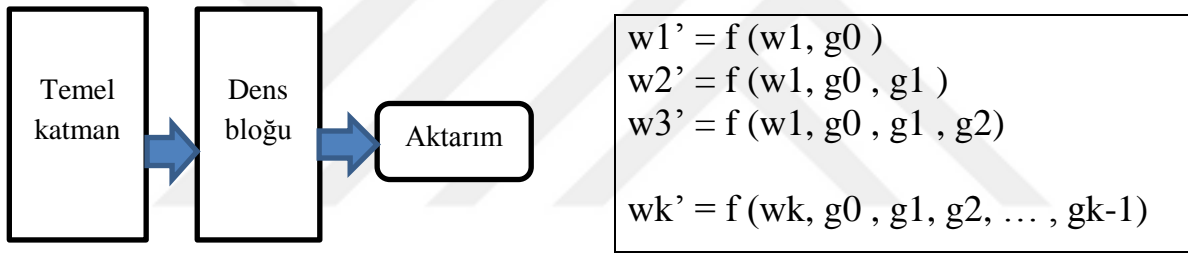
4.3.1. YOLOv4

Yolov4 mimarisi Alexey Bochkovskiy, Chien-Yao Wang ve Hong-Yuan Mark Liao tarafından 2020 yılında kendinden önceki YOLOv3 mimarisinin gelişmiş versiyonu olarak ortaya çıkarılmıştır. YOLOv4 mimarisinde omurga kısmında CSPDarknet53, kafa kısmında ise YOLOv3 mimarisini kullanmaktadır. Bu kapsamda YOLOv4 mimarisinin daha düşük donanım gereksinimleri ile daha iyi sonuçlar verdiği gerçekleştirilen test işlemleri sonucunda belirtilmektedir (Bochkovskiy, Wang ve Liao, 2020).



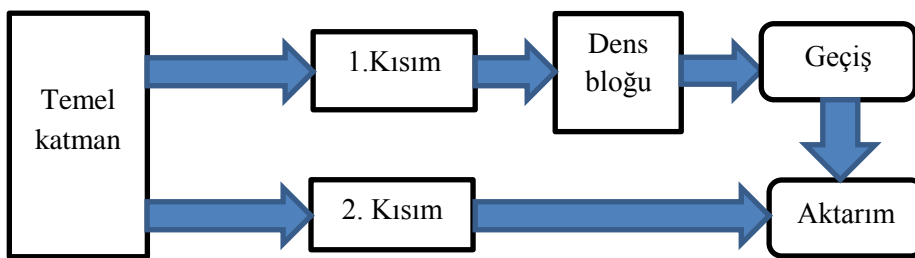
Şekil 4.2. DenseNet yoğun blok modelinin yapısı

Şekil 4.2’de gösterilen DenseNet yoğun blok yapısında her bir katmanın çıktısı bir sonraki katmanın girişi olmaktadır. Yoğun blok yapısı içerisinde ağırlıkların güncellenmesi için kullanılan geri yayılım algoritmasının denklemleri Şekil 4.3’te gösterilmektedir (İnternet, 2021c).



Şekil 4.3. DensNet aktarımı ve ağırlık güncelleme denklemi

CSPDensNet yoğun blok yapısında ise temel katman Şekil 4.4’te gösterildiği gibi iki kısma ayrılmaktadır. İlk kısım doğrudan sonraki adıma aktarılırken ikinci kısım yoğun bloklar ve geçiş katmanından geçerek aktarılmaktadır. Bu şekilde bir sonraki aşamaya aktarılacak özellik haritası oluşturulmaktadır. CSPDensNet ağırlık güncelleme denklemleri Şekil 4.5’te gösterilmektedir (İnternet, 2021c).

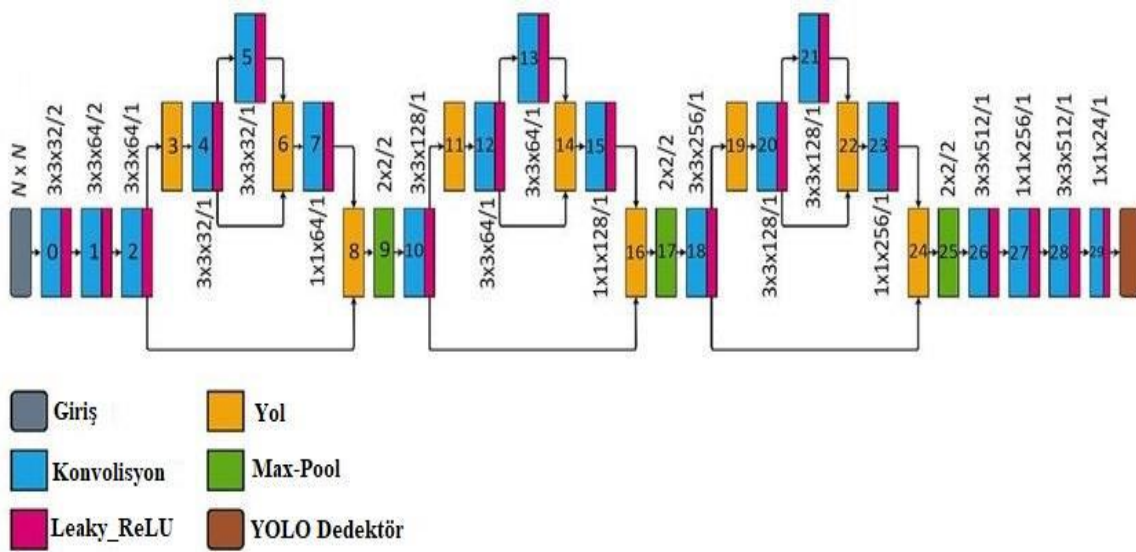


Şekil 4.4. CSPDensNet yoğun blok yapısı

$x_k = w_k * [x_0, x_1, \dots, x_{k-1}]$ $x_T = w_T * [x_0, x_1, \dots, x_k]$ $x_U = w_U * [x_0, \dots, x_T]$	$w_k' = f(w_k, g_0, \dots, g_{k-1})$ $w_T' = f(w_T, g_0, \dots, g_k)$ $w_U' = f(w_U, g_0, \dots, g_T)$
---	--

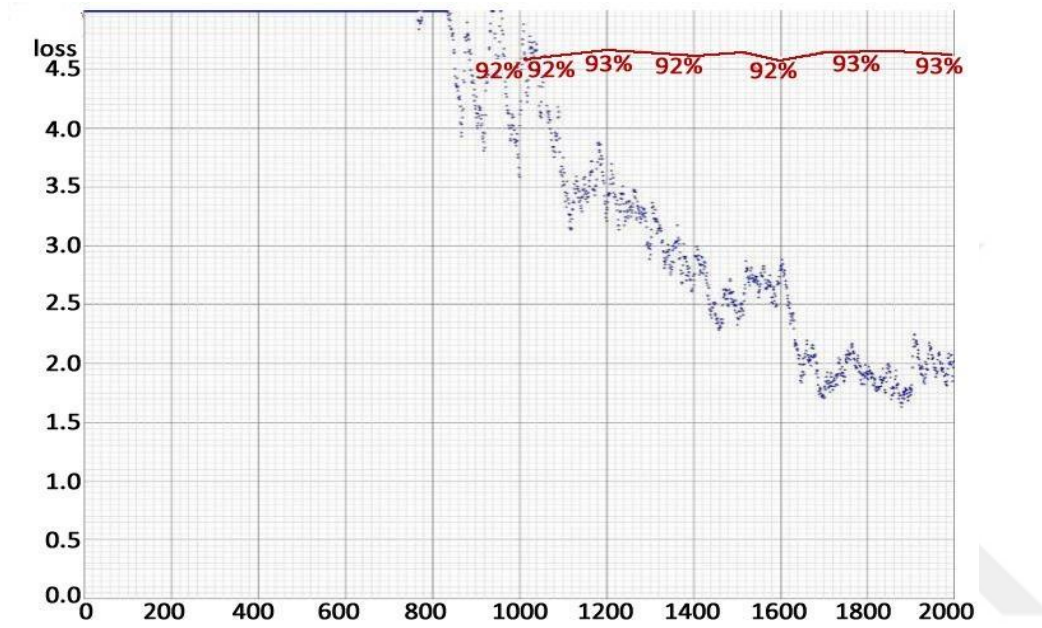
Şekil 4.5. CSPDensNet ağırlık güncelleme denklemleri

YOLOv4 mimarisinin genel yapısı Şekil 4.6'da gösterilmiştir. Bu mimaride uzaysal piramit havuzlama katmanı (SPP) farklı ölçek seviyelerinde algılama tahmini yapmak için kullanılmaktadır. SPP ile 1×1, 5×5, 9×9, 13×13 gibi kayan bir çekirdeğe maksimum havuz uygular fakat uzamsal boyut korunur. Farklı çekirdek boyutlarından özellik haritaları daha sonra çıktı olarak birleştirilmektedir. YOLOv4 mimarisinde kullanılan bir diğer yöntem ise yol toplama ağı (PAN) olarak karşımıza çıkmaktadır. PAN ile tüm katmanlardan gelen bilgiler birleştirilmektedir. Kendinden önceki yöntemlere göre en üst katmana ulaşmak için daha kısa bir yol tanımlamıştır. Bu sayede üst katmanlara geçiş işlemi için ince taneli bilgiler kullanılabilir duruma gelmektedir. Mimaride bulunan bir diğer yöntem ise uzamsal dikkat modülü (SAM) olarak nitelendirilmektedir. Standart SAM mimarisinde özellik haritası oluşturulurken maksimum ve ortalama havuzlama ayrı ayrı uygulanmaktadır. Bu modül YOLOv4'te kullanılırken maksimum ve ortalama havuzlama uygulanmadan modifiye edilmiş mimarisi ile yer almaktadır (İnternet, 2020).



Şekil 4.6. YOLOv4 mimarisinin genel yapısı

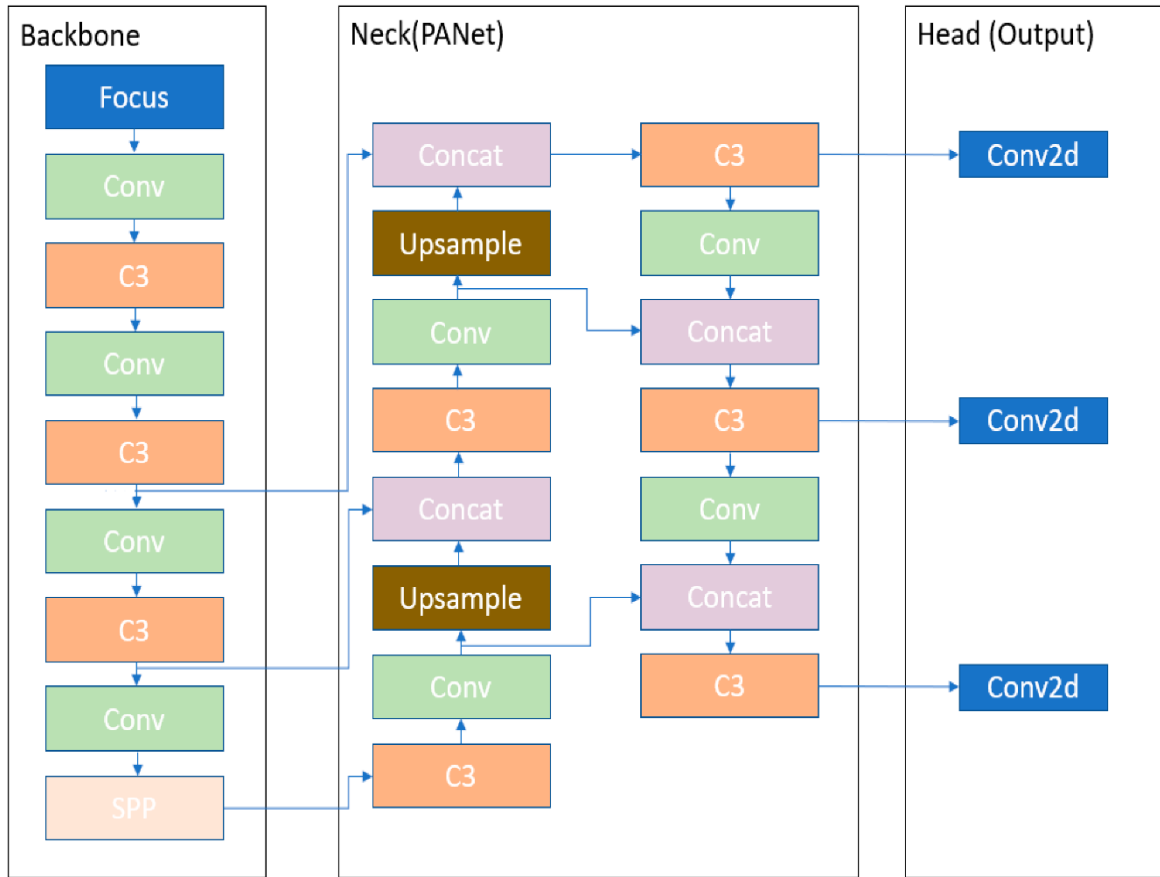
Bu tez çalışmasında kamkat meyvesi özelinde YOLOv4 mimarisi ile eğitimleri gerçekleştirmek için Darknet dedektörü kullanılmıştır. Darknet'te cfg dosyası olarak YOLOv4.cfg dosyası indirilip yapacağımız eğitime göre düzenlenmiştir. YOLOv4 mimarisinde yapılan eğitimlerin kayıp miktarı ve ortalama kesinlik değerleri Şekil 4.7'de gösterilmektedir.



Şekil 4.7. YOLOv4 kayıp ve ortalama kesinlik grafiği

4.3.2. YOLOv5

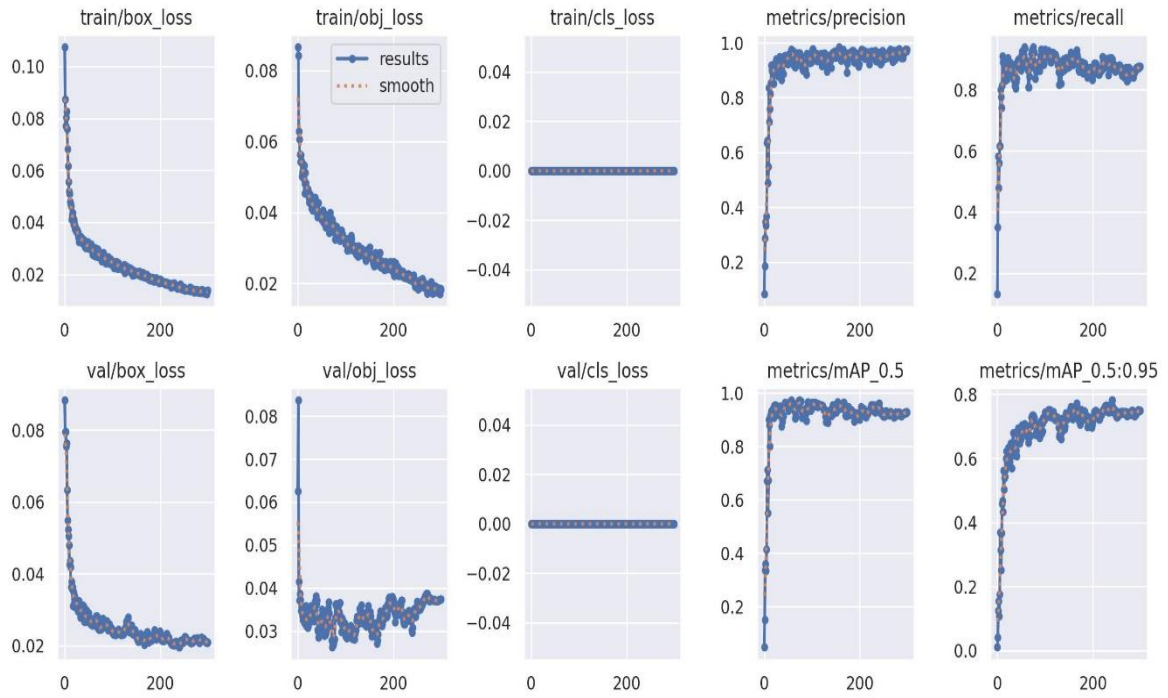
YOLOv5 önceki sürümlerden farklı olarak Darknet yerine PyTorch kullanmaktadır. YOLOv4 gibi omurga kısmında CSPDarknet53 kullanmaktadır. Bu omurga kısmında tekrarlayan gradyan bilgisi çözülürken ayrıca gradyan değişikliği ile doğruluğu artırıp parametreleri azaltmaktadır. Bu sayede model boyutu küçültülerek özellik haritasına entegre edilmektedir. Bilgi akışını artırmak için boyun olarak yol toplama ağını (PAN) kullanır. PAN, birkaç aşağıdan yukarıya ve yukarıdan aşağıya katmanları içeren yeni bir özellik piramit ağını (FPN) benimser. Bu sayede modeldeki düşük seviyeli özelliklerin yayılması iyileştirilir. YOLOv5 mimarisinin kafa kısmı, çok ölçekli tahmin elde etmek için üç farklı özellik haritası çıktısı oluşturan YOLOv4 ve YOLOv3 ile aynıdır. Son olarak YOLO katmanında sonuçlar üretilmektedir. Şekil 4.8'de YOLOv5 mimarisinin genel yapısı gösterilmektedir (Zhu, Geng, Li ve Liu, 2021).



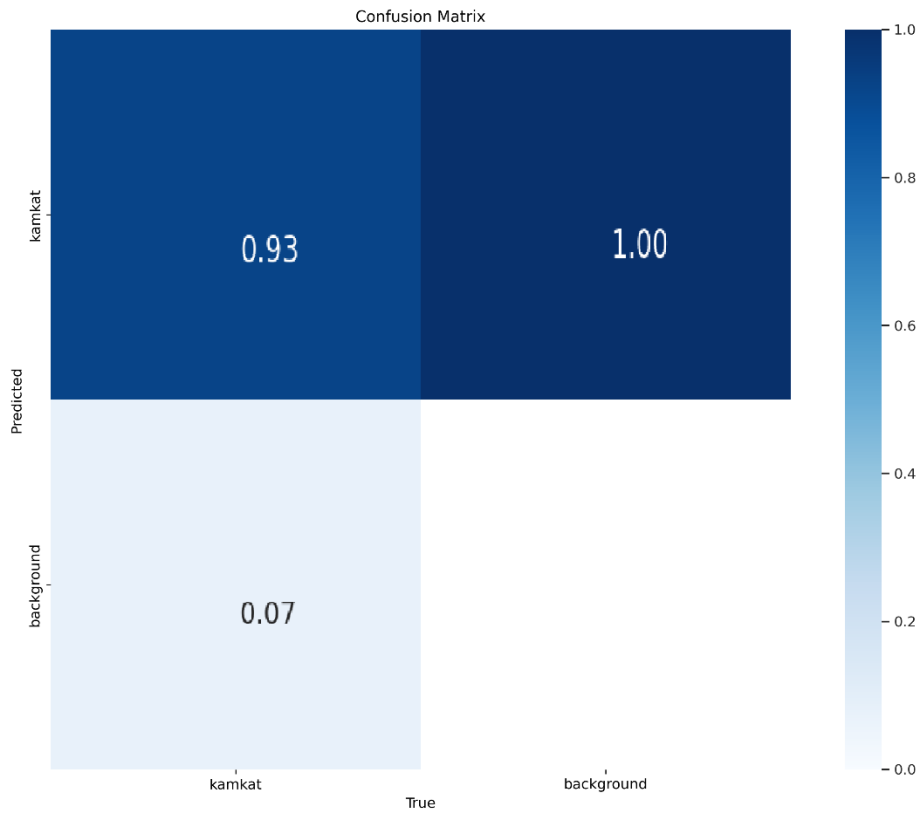
Şekil 4.8. YOLOv5 mimarisinin genel yapısı

YOLOv3, YOLOv4 ve YOLOv5 mimarisi arasındaki temel farka bakarsak YOLOv3'te Darknet53 omurgası, YOLOv4 mimarisinde CSPdarknet53 omurgası, YOLOv5 mimarisinde ise CSPdarknet53 ile Odak katmanı kullanılmaktadır. Odak katmanı ilk olarak YOLOv5 mimarisinde tanıtılmıştır. Odak katmanı, YOLOv3 algoritmasındaki ilk üç katmanın yerini almaktadır. Odak katmanı kullanmanın avantajı gerekli CUDA belleğinin azalması, katmanların azalması, ileri yayılımın ve geri yayılımın artmasıdır (Zhu, Geng, Li ve Liu, 2021).

Tez çalışmasında kamkat meyvesi veri seti özelinde YOLOv5 mimarisinde gerçekleştirilen eğitimler sonucunda elde edilen sonuç grafikleri Şekil 4.9'da, karmaşıklık matrisi ise Şekil 4.10'da gösterilmektedir.



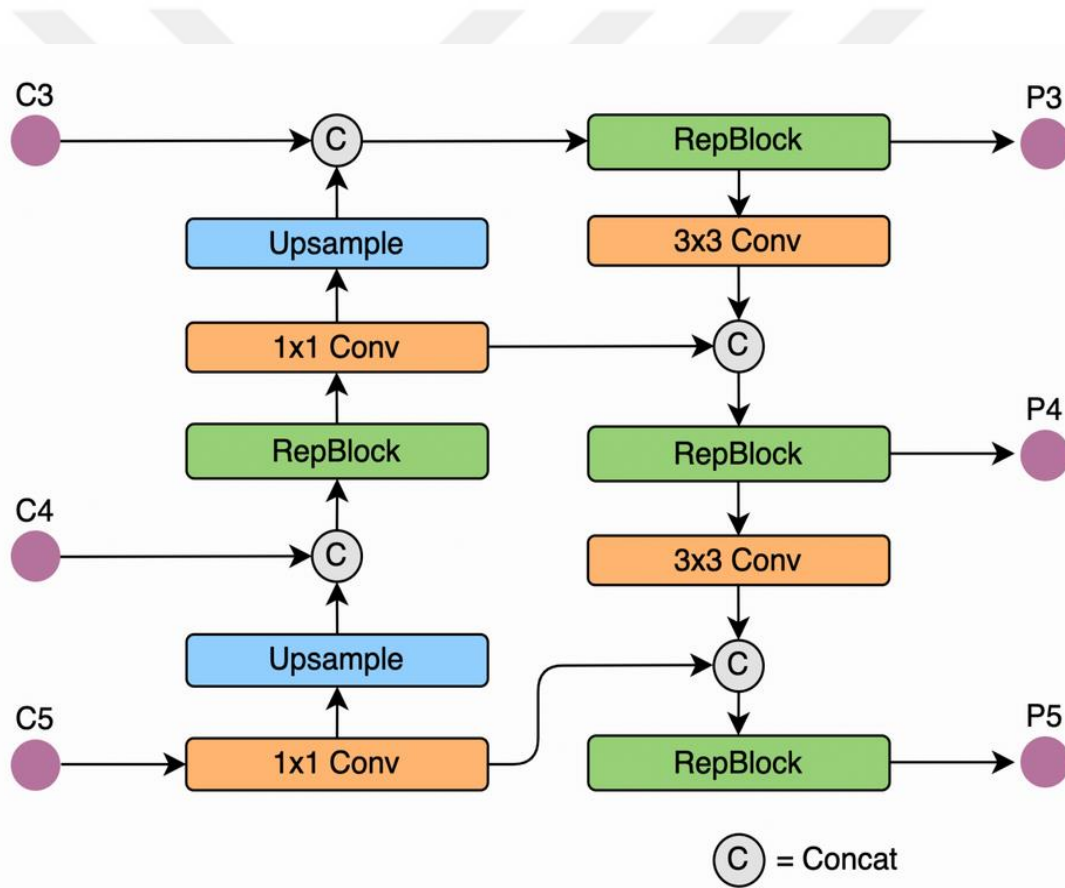
Şekil 4.9. YOLOv5 sonuç grafikleri



Şekil 4.10. YOLOv5 karmaşıklık matrisi

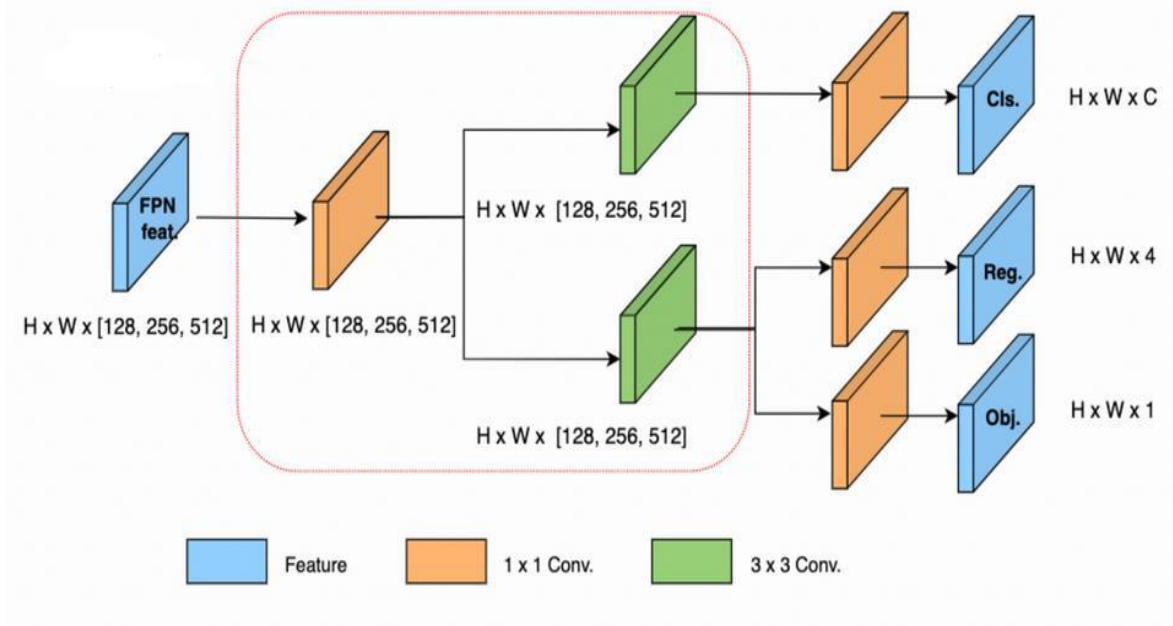
4.3.3. YOLOv6

YOLOv6 mimarisi geliştirilirken ana hedef endüstride kullanılan donanımlar olarak belirlenmiştir. Bu kapsamda model tasarımı daha düşük donanımsal ihtiyaçlarda başarılı sonuçlar almak üzerine tasarlanmıştır. YOLOv6 mimarisinde eğitim işlemleri için giriş görüntüleri 640 x 640 olarak verilmektedir. YOLOv6 mimarisinde model parametreleri diğer modellerden farklı olarak Python'da tanımlanmaktadır. YOLOv6 mimarisinde boyun ve omurga yapısı donanıma uygun şekilde tasarlanarak EfficientRep omurgası ile Rep-PAN boyun yapısı tanıtılmıştır. Rep-PAN boyun yapısı Şekil 4.11'de yer almaktadır (İnternet, 2022c).



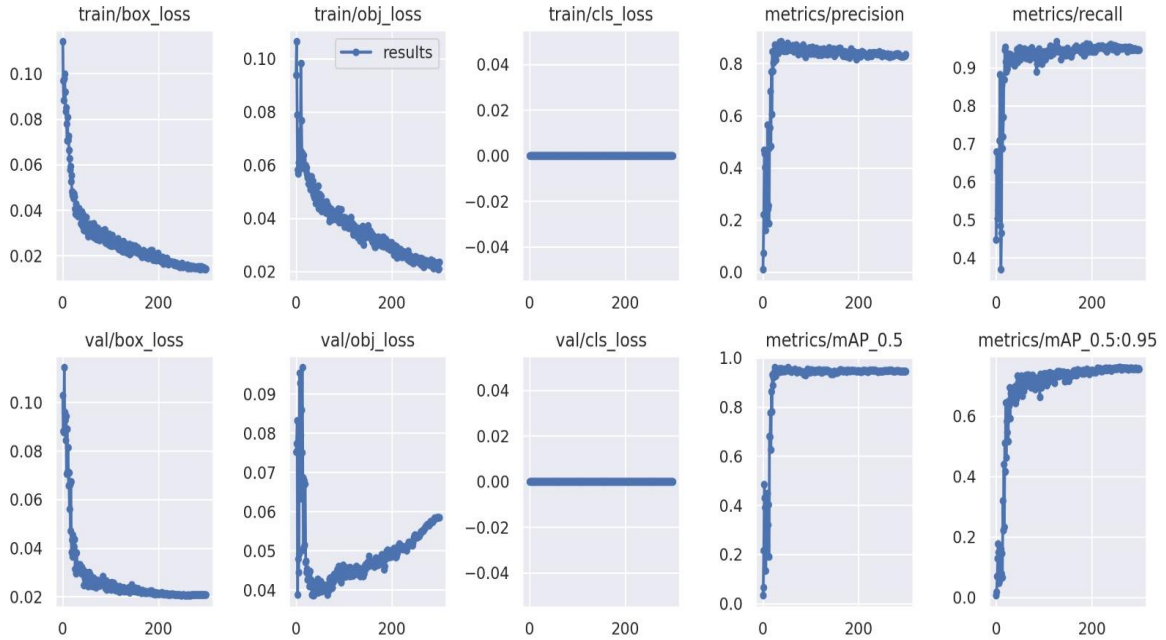
Şekil 4.11. Rep-PAN boyun yapısı

YOLOv6 mimarisinde kafa yapısının tasarımında diğer modellerden farklı olarak ayrılmış yapı şekli kullanılmıştır. Ayrılmış kafa yapısı mimarisi Şekil 4.12'de gösterilmektedir (İnternet, 2022c).

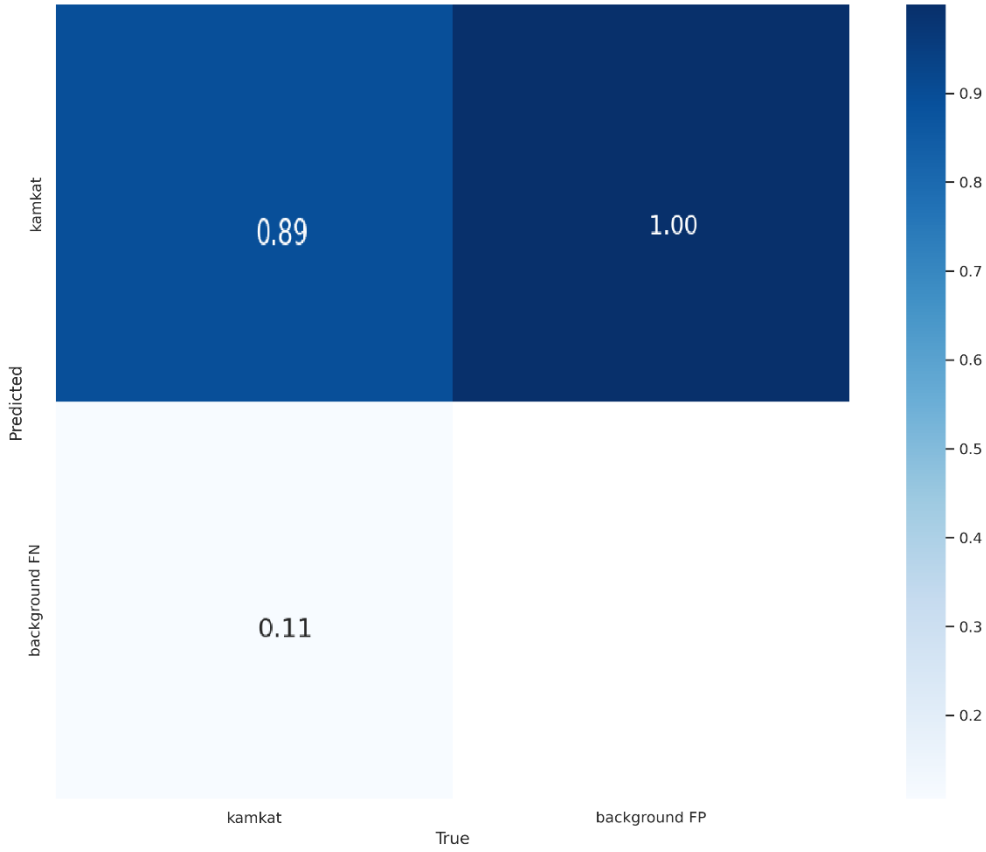


Şekil 4.12. YOLOv6 ayrılmış kafa yapısı mimarisi

Tez çalışması kapsamında kamkat meyvesi veri seti için YOLOv6 mimarisinde eğitim ve test işlemleri gerçekleştirilmiştir. Yapılan işlemlerle ilgili sonuç grafikleri Şekil 4.13'te, karmaşıklık matrisi ise Şekil 4.14'te gösterilmektedir.



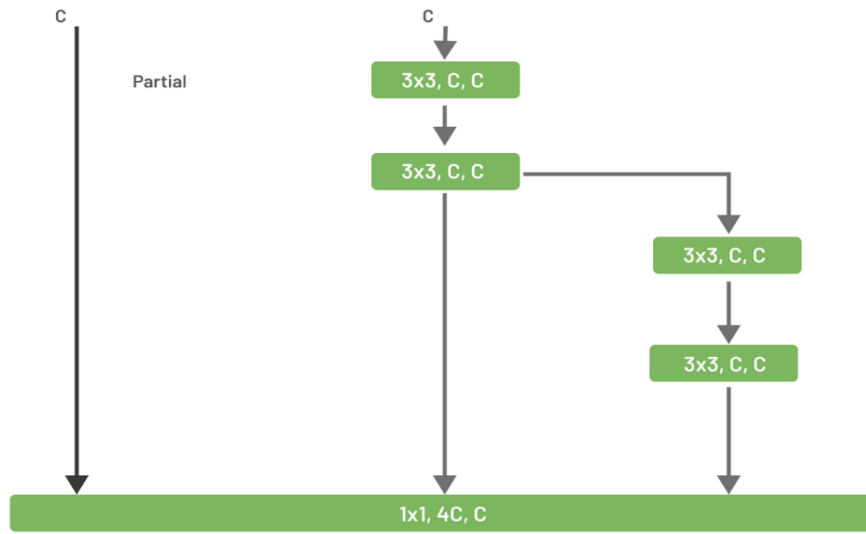
Şekil 4.13. YOLOv6 sonuç grafikleri



Şekil 4.14. YOLOv6 karmaşıklık matrisi

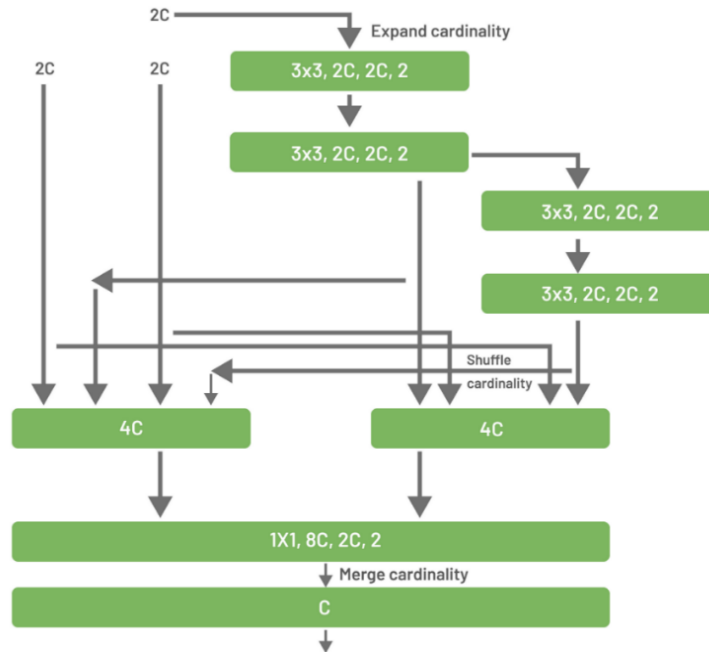
4.3.4. YOLOv7

YOLOv7 mimarisi YOLOv4 mimarisinin bir uzantısı olan Scaled YOLOv4 yazarları tarafından geliştirilmiştir. YOLOv7 mimarisi geliştirilirken kendinden önceki versiyonlardan farklı olarak verimli katman toplama ağı (ELAN) yapısından faydalanılmıştır. ELAN yapısı, daha derin ağların etkili bir şekilde birleşip öğrenilmesi için en kısa ve en uzun gradyan yolunu kontrol ederek verimli bir ağ tasarlamayı düşündür. Gradyan ne kadar kısa olursa, ağları o kadar güçlü bir şekilde öğrenebilecektir. ELAN mimari diyagramı Şekil 4.15'te gösterilmiştir (İnternet, 2022d).



Şekil 4.15. ELAN mimari diyagramı

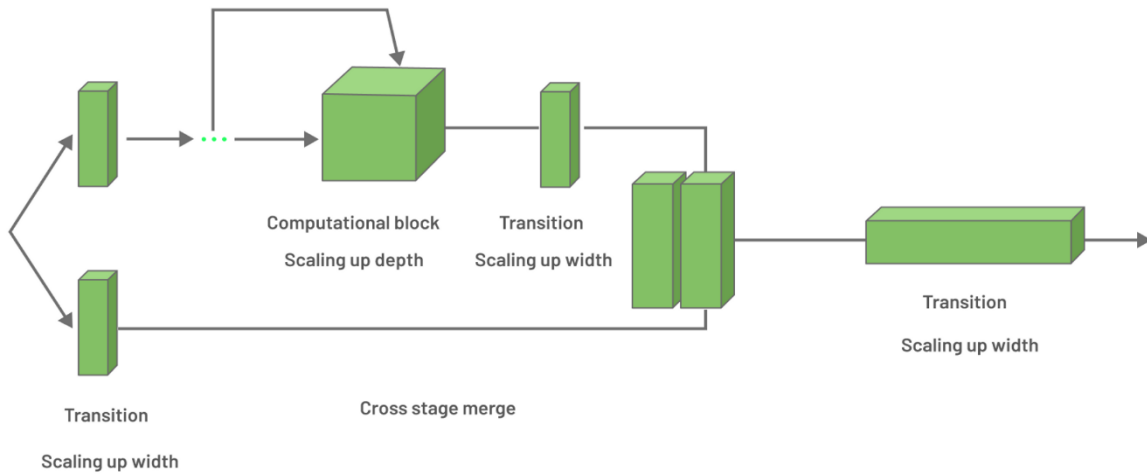
Şekil 4.15'te yer alan diyagram incelendiğinde girişin 1x1 evrişim ile doğrudan alt bloğu beslediği görülmektedir. Alt bloğa yapılan diğer iki bağlantı ise girişin 2 ve 4 blok 3x3 evrişim ile evrilerek elde edilmiştir. Alt blokta tüm özellikler birleştirilmekte ve zengin bilgileri öğrenmek için 1x1 evrişim uygulanmaktadır (İnternet, 2022d).



Şekil 4.16. E-ELAN mimari diyagramı

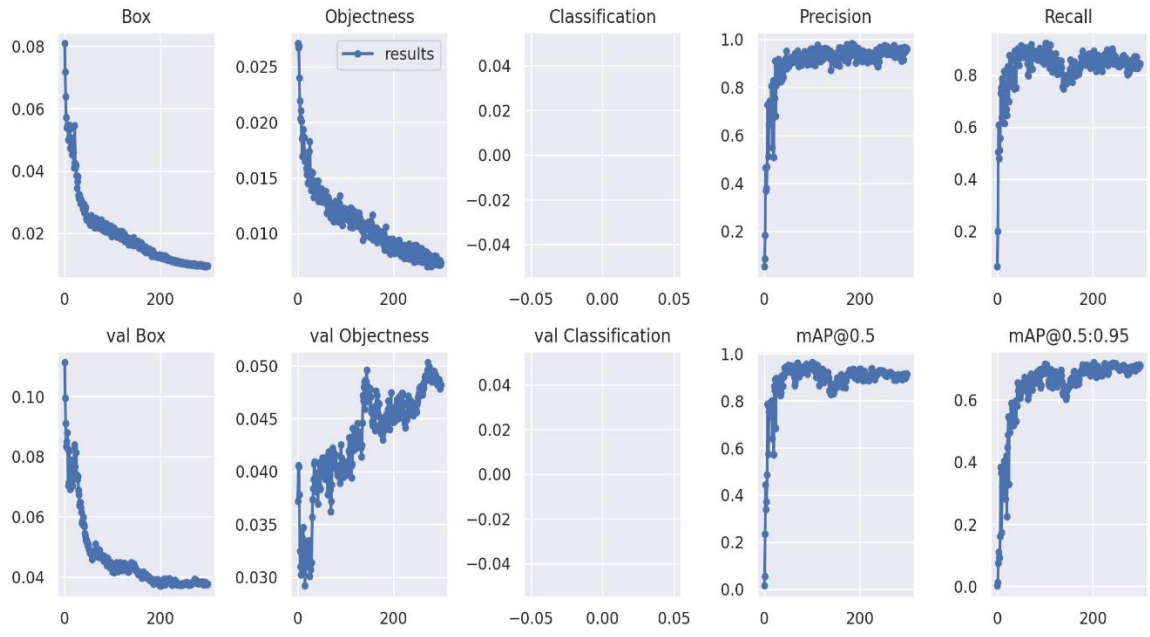
YOLOv7 mimarisi ELAN yapısını kullanırken bazı değişiklikler yaparak bunu genişletilmiş verimli katman toplama ağı (E-ELAN) olarak isimlendirmiştir. Bu kapsamda ELAN gradyan akış yollarını bozmadan model öğrenme yeteneğini geliştirmek için genişletme, karıştırma ve birleştirme kardinalitesini kullanır. Mimari yapıda yalnızca hesaplama bloğundaki mimari değiştirilirken diğer katmanlar ELAN ile aynı şekilde bırakılmıştır. E-ELAN, hesaplama bloğunun kanallarını ve önemliliğini genişletmek için grup evrişimini kullanmaktadır. Hesaplama katmanındaki tüm hesaplama bloklarına aynı kanal çarpanını ve grup parametresini uygulanmaktadır. E-ELAN mimari diyagramı Şekil 4.16'da gösterilmektedir (İnternet, 2022d).

YOLOv7 mimarisinde yer alan bir diğer yapı ise bileşik model ölçekleme yaklaşımı olarak ortaya çıkmaktadır. Bu yapı ile tüm katmanları birleşik yapıya sahip olan YOLOv7 mimarisinde sahip olunan özellikler optimum şekilde korunacaktır. YOLOv7 mimarisi bileşik ölçeklendirme yapısı Şekil 4.17'de yer almaktadır (İnternet, 2022d).

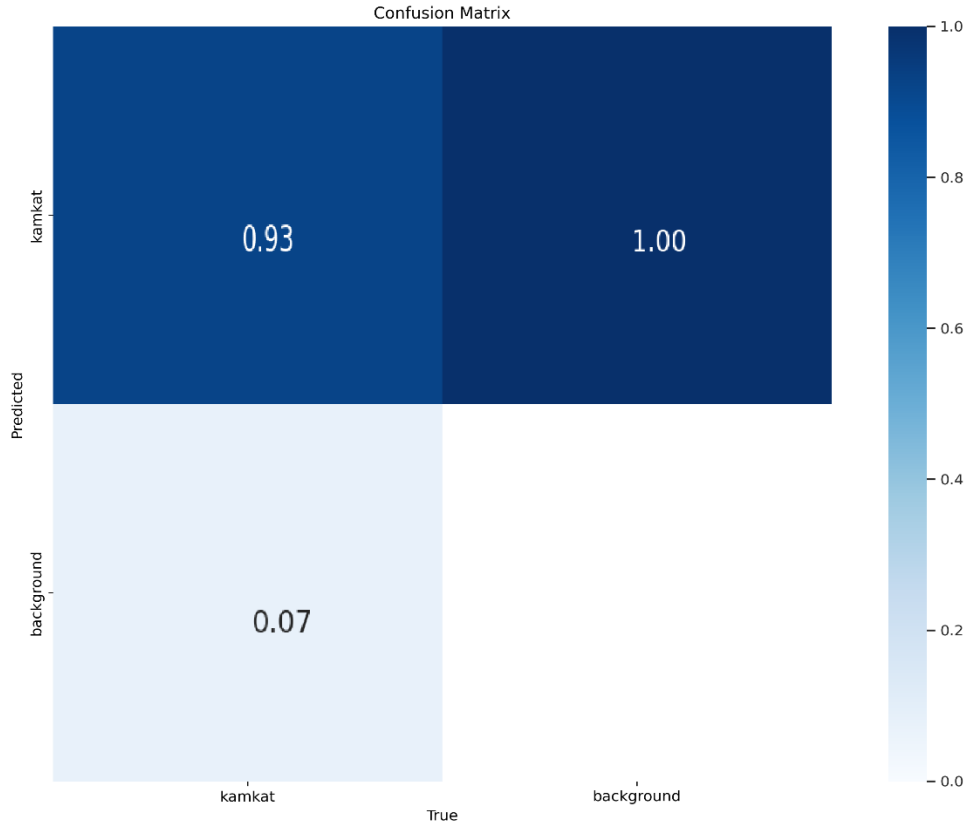


Şekil 4.17. YOLOv7 mimarisi bileşik ölçeklendirme yapısı

Tez çalışması kapsamında kamkat meyvesi veri seti için YOLOv7 mimarisinde eğitim ve test işlemleri gerçekleştirilmiştir. Yapılan işlemlerle ilgili sonuç grafikleri Şekil 4.18'de, karmaşıklık matrisi ise Şekil 4.19'da gösterilmektedir. Ayrıca eğitilmiş model ile yapılan nesne tespiti çıktısının bir örneği Resim 4.4'te yer almaktadır.



Şekil 4.18. YOLOv7 sonuç grafikleri



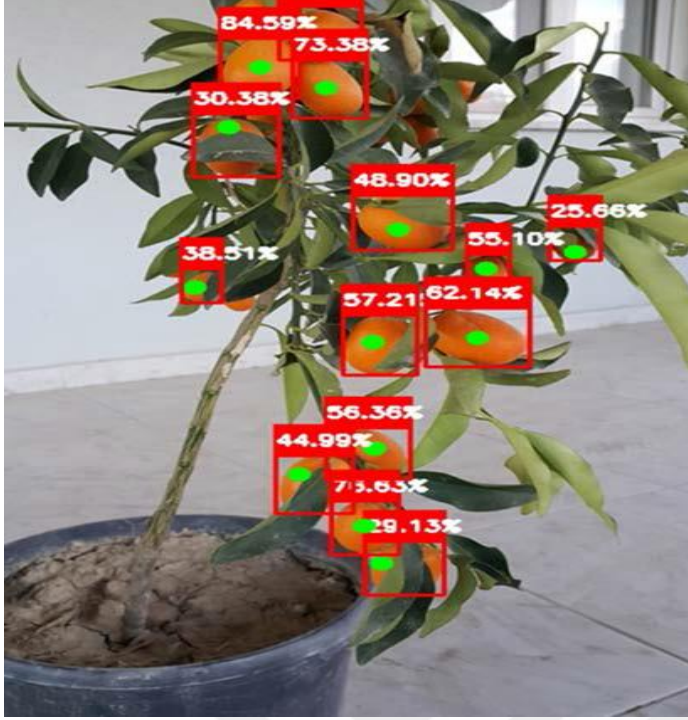
Şekil 4.19. YOLOv7 karmaşıklık matrisi



Resim 4.4. YOLOv7 nesne tespiti çıktısı

4.4. Kamkat Toplama Noktasının Tespiti

Literatürdeki birçok otonom hasat robotu çalışmasında, yaprakların arkasında daha az görünen meyvelerin nesne algılama performansını düşürerek toplama performansını olumsuz yönde etkilediği belirtilmektedir. Bu sebeple otonom hasat robotları görü sisteminde tespit edilen nesne üzerinde yaprakların ve dalların bulunmadığı kısımda toplama için en uygun nokta tespiti yapılmıştır. Bunun için Python dilinde opencv kütüphanesi kullanılarak en uygun yapışma noktasının belirlenmesi amacı ile bir algoritma geliştirilmiştir. Resim 4.5'te kamkat toplama noktalarının bir örneği gösterilmektedir.



Resim 4.5. Kamkat toplama noktalarının işaretlenmesi

Geliştirilen algoritmada ilk önce kullanılan YOLO modeli ile yapılan nesne tespiti sonuçları alınmaktadır. Bu veriler içinde tespit edilen kamkatlar için güven endeksleri yer almaktadır. Bu endeksler kullanılarak güven endeksi %80 altında olanlar elenmiştir. Geriye kalan tespit edilmiş kamkatların çerçeveleri çizdirilmiş ve kamkatın toplanabilir durumda olduğunu belirlemek için bu çerçeveler içinde kalan alandaki turuncu piksel oranlarına bakılmıştır. Çerçeve alan içinde piksel oranı %50 ve üzeri turuncu olanlar toplanabilir olarak belirlenip o kamkatlar ile devam edilmiştir. Daha sonra geriye kalan çerçeve içindeki kamkatların alanlarındaki turuncu yoğunluğu en yüksek olan piksel alanı üzerine toplama noktası işareti etiketlenmiştir. Belirlenen bu toplama noktaları kamkat meyvesinin engellere takılmadan en kolay hasat edileceği kısımlardır.

4.5. YOLO Modelleri Test Sonuçları

Otonom hasat robotu gerçek zamanlı nesne tespiti sistemi için YOLO algoritması mimarilerinden YOLOv4, YOLOv5, YOLOv6 ve YOLOv7 kullanılarak kamkat veri seti üzerinde eğitim ve test işlemleri gerçekleştirilmiştir. Bu işlemler sonucunda elde edilen test sonuçları Çizelge 4.1'de yer almaktadır.

Çizelge 4.1 YOLO modelleri test sonuçları

Model	Sınıf	Hatırlama (Recall)	Keskinlik (Precision)	F1 Puanı	Map50
YOLOv7	Kamkat	0,84	0,95	0,90	0,93
YOLOv6	Kamkat	0,60	0,72	0,81	0,89
YOLOv5	Kamkat	0,87	0,97	0,92	0,93
YOLOv4	Kamkat	0,91	0,74	0,82	0,90

Model başarımlarının tespiti sonrasında gerçek zamanlı otonom hasat işlemi için başarımların yanı sıra FPS karşılaştırması önem arz etmektedir. Bu kapsamda yapılan test işlemleri sonucu elde edilen veriler Çizelge 4.2’de gösterilmektedir.

Çizelge 4.2 YOLO modelleri FPS sonuçları

Model	Genel Başarım (%)	FPS
YOLOv7	93	35
YOLOv6	89	33
YOLOv5	93	29
YOLOv4	90	25

Yapılan çalışmalar sonucunda modellerin hem doğruluk hem de FPS başarımları kıyaslandığında en uygun model YOLOv7 olarak tespit edilmiştir. YOLOv7 mimarisi %93 genel doğruluk ve 35 FPS oranı ile diğer modellerden daha başarılı sonuçlar ortaya koymuştur. Genel doğruluk oranları birçok modelde yakın sonuçlar vermiştir fakat gerçek zamanlı bir hasat uygulamasında kullanılacağı için FPS oranları kıyaslamasında YOLOv7 modeli ön plana çıkmıştır. Genel doğruluk ve FPS oranı bir bütün olarak düşünüldüğünde otonom bir hasat robotunda görü sistemi için YOLOv7 mimarisinin kullanılması en uygundur.

5. TARTIŞMA VE ÖNERİLER

5.1. Tartışma

Çalışmamızda gerçek zamanlı nesne tespiti işlemi için veri seti hazırlanırken tamamen doğada bulunduğu durumda saksı kamkat fidesi üzerinden görüntüler toplanmıştır. Literatürdeki bazı hasat robotları için yapılan bilgisayarlı görü çalışmalarında ağaçlardan yaprak, dal veya yapışık meyve seyreltilmesi gibi modifiye işlemlerinden bahsedilmektedir. Bu noktada başarımlar kıyaslanırken bahse konu durumun da göz önünde bulundurulması gerekmektedir. Tamamen deneysel ortamda ve modifiye edilmiş bir örnek üzerinden yapılan test sonuçları, gerçek sahada yapılacak bir uygulamada aynı sonuçları veremeyecektir.

Çizelge 5.1 Literatürdeki otonom hasat robotları ile karşılaştırma

Çalışma	Ürün	Model	Test başarımları (%)
Birrell ve diğerleri (2019)	Marul	Yolov3	91
Yu ve diğerleri (2020)	Çilek	R-YOLO	94
Kuznetsova ve diğerleri (2020)	Elma	YOLOv3	92
Yin ve diğerleri (2021)	Üzüm	M-RCNN	89
Zhang ve diğerleri (2021)	Elma	M-RCNN	90
Kar ve diğerleri (2021)	Biber	SSD	88
Bu çalışmada	Kamkat	YOLOv7	93

Çizelge 5.1’de görüldüğü üzere literatürde farklı meyve ve sebzeler için yapılmış otonom hasat çalışmaları bulunmaktadır. Bu kapsamda farklı derin öğrenme algoritmaları kullanılarak nesne tespiti çalışmaları yapılmıştır. Bazı çalışmalarda elde edilen nesne tespit

algoritması robotik bir sistem üzerine uygulanarak deneysel ortamda hasat denemeleri yapılmıştır. Geçmiş çalışmalar incelendiğinde farklı ürün türleri üzerinde birçok farklı derin öğrenme algoritması ile otonom hasat için bilgisayarlı görü denemeleri görülmektedir. Genel olarak nesne tespitinde yüksek doğruluklar elde edilmiştir. Fakat nesne tespiti algoritması test edilirken özellikle modifiye edilmiş veri seti kullanılan çalışmaların bu noktada yüksek başarımlar alırken, robotik sistem üzerindeki uygulamalarında başarısız sonuçlar alındığı görülmektedir. Ayrıca gerçek zamanlı hasat işlemi sırasında yaprak ve dal gibi engellerden dolayı toplama başarımının düşük olduğu görülmüştür. Çalışmalarda kullanılan bazı modeller çok fazla katmana ve ağır bir mimariye sahip olduğundan hız olarak gerçek zamanlı bir uygulama için uygun görülmemiştir. Bu kapsamda yapılan bu tez çalışmasında önerilen YOLOv7 derin öğrenme algoritması ile hem yüksek doğruluk hem de yüksek FPS oranı elde edilmiştir. Bu kapsamda otonom bir hasat robotunun görü sisteminde bu modelin kullanılması gerçek sahada yapılacak hasat başarımını da artıracaktır. Çalışmada ayrıca yapılan toplama noktası tespiti ile gerçek sahada yapılacak bir hasat işleminde literatürde karşılaşılan yaprak veya dal gibi engellerin hasat başarımını olumsuz etkilemesinin önüne geçilecektir.

5.2. Öneriler

Çalışmada YOLOv7 modeli test işlemleri sonucunda %93 genel doğruluk ve 35 FPS oranı elde edilmiştir. Bunun yanında hasat işlemi için toplama noktası tespiti yapılmıştır. Yapılan bu çalışma kamkat meyvesi özelinde yapılmıştır fakat birçok ürünün hasadı için yapılacak otonom sistemlerde kullanılabilir. Literatürde yapılan benzer çalışmalarda bahsedilen yaprak ve dalların hasat başarımına olumsuz etkisini azaltacak bir öneri sunulmuştur.

Bir sonraki adım olarak kurulacak bir otonom hasat robotu sistemine elde ettiğimiz veriler entegre edilerek saha testlerinin yapılmasıdır. Bunun sonuçlarına göre iyileştirmeler ve farklı çözümler üretilmelidir.

KAYNAKLAR

- Arad, B., Balendonck, J., Barth, R., Ben- Shahaar, O., Edan, Y., Hellström, T. ve Van Tuijl, B. (2020). Development of A Sweet Pepper Harvesting Robot. *Journal of Field Robotics*, 37(6), 1027-1039.
- Berki, K., Artan, K. ve Orhan, M. (2017). Ağaçtaki Kirazın Tespit Edilmesi. *Soma Meslek Yüksekokulu Teknik Bilimler Dergisi*, 2(24), 93-104.
- Birrell, S., Hughes, J., Cai, J. Y. ve Lida, F. (2020). A Field- Tested Robotic Harvesting System for Iceberg Lettuce. *Journal of Field Robotics*, 37(2), 225-245.
- Bochkovskiy, A., Wang, C. Y. ve Liao, H. Y. M. (2020). Yolov4: Optimal Speed and Accuracy of Object Detection. *ArXiv Preprint ArXiv:2004.10934*.
- Dersuneli, M., Gündüz, T. ve Kutlu, Y. (2021). Bul-Tak Oyuncağı Şekillerinin Klasik Görüntü İşleme ve Derin Öğrenme Yöntemleri ile Tespiti. *Bitlis Eren Üniversitesi Fen Bilimleri Dergisi*, 10(4), 1290-1303.
- Doğan, F. ve Türkoğlu, İ. (2019). Derin Öğrenme Modelleri ve Uygulama Alanlarına İlişkin Bir Derleme. *Dicle Üniversitesi Mühendislik Fakültesi Mühendislik Dergisi*, 10 (2), 409-445.
- Feng, Q., Zou, W., Fan, P., Zhang, C. ve Wang, X. (2018). Design and Test of Robotic Harvesting System for Cherry Tomato. *International Journal of Agricultural and Biological Engineering*, 11(1), 96-100.
- Fu, L., Tola, E., Al-Mallahi, A., Li, R. ve Cui, Y. (2019). A Novel Image Processing Algorithm to Separate Linearly Clustered Kiwifruits. *Biosystems Engineering*, 183, 184-195.
- Gonzalez, R.C. ve Woods, R.E. (2002). *Digital Image Processing, 3rd Edition*, United States: Prentice Hall, 221-807.
- Göçmen, S. (2018). Gezin Satıcı Problemleri Baz Alınarak Multimodal Taşımacılık Güzergahlarının Optimizasyonu ve Akıllı Ulaşım Sistemlerine Entegrasyonu. Yüksek Lisans Tezi, İskenderun Teknik Üniversitesi Mühendislik ve Fen Bilimleri Enstitüsü, Hatay.
- Gökalp, Ö. M. (2022). Makine Öğrenmesi. Gazi Üniversitesi, Gazi Bilişim Enstitüsü, Adli Bilişim Bölümü, 14.
- İnternet: (2017). Yapay Zeka, Makine Öğrenmesi ve Derin Öğrenme İlişkisi. URL: <https://becominghuman.ai/the-artificial-use-of-the-term-intelligence-c82878447dfd>, Son Erişim Tarihi: 10.07.2022

- İnternet: Karakuş, B. A. (2018). Derin Sinir Ağları için Aktivasyon Fonksiyonları. URL: <http://buyukveri.firat.edu.tr/2018/04/17/derin-sinir-aglari-icin-aktivasyon-fonksiyonlari>, Son Erişim tarihi: 07.08.2022.
- İnternet: Hui, J. (2020). YOLOv4. URL: <https://jonathan-hui.medium.com/yolov4-c9901eaa8e61>, Son Erişim tarihi: 14.10.2022.
- İnternet: (2021a). Deep Learning Nedir? URL: <https://blog.turhost.com/deep-learning-nedir>, Son Erişim tarihi: 10.07.2022.
- İnternet: Sayracı, B. (2021b). Derin Öğrenme ve Evrimsel Sinir Ağları. URL: <https://medium.com/@beyzasayraci>, Son Erişim tarihi: 07.08.2022.
- İnternet: Tsang, S. (2021c). Review-CSPNet: A New Backbone That Can Enhance Learning Capability of CNN. URL: <https://sh-tsang.medium.com/review-cspnet-a-new-backbone-that-can-enhance-learning-capability-of-cnn-da7ca51524bf>, Son Erişim tarihi: 14.10.2022.
- İnternet: (2022a). Yapay Zeka Zaman Çizelgesi. URL: <https://turkiye.ai/kaynaklar/yapay-zeka-zaman-cizelgesi>, Son Erişim tarihi: 10.07.2022.
- İnternet: Keita, Z. (2022b). YOLO Object Detection Explained. URL: <https://www.datacamp.com/blog/yolo-object-detection-explained>, Son Erişim tarihi: 06.01.2023.
- İnternet: Solawetz, J. ve Nelson, J. (2022c). What is YOLOv6? A Beginner's Guide URL: <https://blog.roboflow.com/yolov6>, Son Erişim tarihi: 11.01.2023.
- İnternet: Munawar, M. R. (2022d). YOLOv7 Architecture Explanation. URL: <https://www.cameralyze.co/blog/yolov7-architecture-explanation>, Son Erişim tarihi: 11.01.2023.
- Jun, J., Kim, J., Seol, J., Kim, J. ve Son, H. I. (2021). Towards an Efficient Tomato Harvesting Robot: 3D Perception, Manipulation, and End-Effector. IEEE Access, 9, 17631-17640.
- Kar, Y. E., Basgumus, A. ve Namdar, M. (2021). Machine Learning Assisted Autonomous Vehicle Design and Control. In 2021 5th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT) IEEE, 462-466.
- Kurtulmus, F., Lee, W. S. ve Vardar, A. (2011). Green Citrus Detection Using 'Eigenfruit', Color and Circular Gabor Texture Features Under Natural Outdoor Conditions. Computers and Electronics in Agriculture, 78(2), 140-149.
- Kuznetsova, A., Maleva, T. ve Soloviev, V. (2020). Using YOLOv3 Algorithm with Pre-and Post-Processing for Apple Detection in Fruit-Harvesting Robot.

- Agronomy, 10(7), 1016.
- Lin, G., Zhu, L., Li, J., Zou, X. ve Tang, Y. (2021). Collision-Free Path Planning for A Guava-Harvesting Robot Based on Recurrent Deep Reinforcement Learning. *Computers and Electronics in Agriculture*, 188, 106350.
- Odegard I. Y. R. ve Van Der Voet E. (2014). The Future of Food Scenarios and The Effect on Natural Resource Use in Agriculture in 2050. *Ecological Economics*, 97, 51-59.
- Qingchun, F., Xiu, W., Wengang, Z., Quan, Q. ve Kai, J. (2012). New Strawberry Harvesting Robot for Elevated-Trough Culture. *International Journal of Agricultural and Biological Engineering*, 5(2), 1-8.
- Rong, J., Wang, P., Yang, Q. ve Huang, F. (2021). A Field-Tested Harvesting Robot for Oyster Mushroom in Greenhouse. *Agronomy*, 11(6), 1210.
- Samtaş, G. ve Gülesin, M. (2011). Sayısal Görüntü İşleme ve Farklı Alanlardaki Uygulamaları. *Electronic Journal of Vocational Colleges*, 2(1), 85-97.
- Tanigaki, K., Fujiura, T., Akase, A. ve Imagawa, J. (2008). Cherry-Harvesting Robot. *Computers and Electronics in Agriculture*, 63(1), 65-72.
- Üzen, H. (2018). Geleneksel ve Derin Öğrenme Tabanlı Nesne Takip Yöntemlerinin Performans Değerlendirmesi. Yüksek Lisans Tezi, İnönü Üniversitesi Fen Bilimleri Enstitüsü, Malatya.
- Williams, H. A., Jones, M. H., Nejati, M., Seabright, M. J., Bell, J., Penhall, N. D. ve MacDonald, B. A. (2019). Robotic Kiwifruit Harvesting Using Machine Vision, Convolutional Neural Networks, and Robotic Arms. *Biosystems Engineering*, 181, 140-156.
- Yayık A. ve Kutlu Y. (2012). Diagnosis of Congestive Heart Failure Using Poincare Map Plot. 20th Signal Processing and Communications Applications Conference (SIU), 1-4.
- Yin, H., Chai, Y., Yang, S. X. ve Mittal, G. S. (2009). Ripe Tomato Recognition and Localization for A Tomato Harvesting Robotic System. In 2009 International Conference of Soft Computing and Pattern Recognition IEEE, 557-562.
- Yin, W., Wen, H., Ning, Z., Ye, J., Dong, Z. ve Luo, L. (2021). Fruit Detection and Pose Estimation for Grape Cluster-Harvesting Robot Using Binocular Imagery Based on Deep Neural Networks. *Frontiers in Robotics and AI*, 8, 626989.
- Yılmaz, B. (2016). İnsan Tespitinde Yapay Veri Artırımı. Doktora Tezi, Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü, İstanbul.
- Yu, Y., Zhang, K., Liu, H., Yang, L. ve Zhang, D. (2020). Real-Time Visual Localization of The Picking Points for A Ridge-Planting Strawberry Harvesting Robot. *IEEE Access*, 8, 116556-116568.

- Yu, X., Fan, Z., Wang, X., Wan, H., Wang, P., Zeng, X. ve Jia, F. (2021). A Lab-Customized Autonomous Humanoid Apple Harvesting Robot. *Computers & Electrical Engineering*, 96, 107459.
- Zhang, K., Lammers, K., Chu, P., Li, Z. ve Lu, R. (2021). System Design and Control of An Apple Harvesting Robot. *Mechatronics*, 79, 102644.
- Zhao, D., Lv, J., Ji, W. ve Zhang, Y. (2011). Design and Control of An Apple Harvesting Robot. *Biosystems Engineering*, 110(2), 112-122.
- Zhu, L., Geng, X., Li, Z. ve Liu, C. (2021). Improving YOLOv5 with Attention Mechanism for Detecting Boulders from Planetary Images. *Remote Sensing*, 13(18), 3776.



DİZİN

A

Abstract · v

B

Bilgisayarlı görü · iv, 6, 20, 31,
48, 49

Boyun · 36, 39

D

Darknet · 33, 36, 37

DenseNet · 34

Derin öğrenme · iv, 2, 6, 7, 8, 9,
10, 12, 14, 15, 17, 18, 19, 20,
21, 23, 25, 31, 32, 33, 48, 49

Doğruluk · iv, 8, 20, 25, 30, 47,
49

E

Endüstri · 11, 14, 15, 39

Etiketleme · 32, 33

Evrişim · 6, 22, 23, 26, 42, 43

F

Filtre · 3, 7, 11, 13, 22, 23

Fonksiyon · 4, 17, 18, 23, 24, 26

FPS · iv, v, 21, 25, 47, 49

G

Gerçek zamanlı · 2, 6, 9, 20, 21,
25, 26, 31, 33, 46, 47, 48, 49

Görüntü işleme · 1, 3, 4, 5, 11,
12, 13, 14, 17, 20, 22

Güven skoru · 29

H

Hasat · iv, 1, 2, 3, 4, 5, 6, 7, 8, 9,
12, 31, 33, 45, 46, 47, 48, 49

K

Kafa · 33, 36, 39, 40

Koordinat · 27

M

Makine öğrenmesi · 14, 15, 16,
17, 19, 20

N

Nesne takibi · 11, 21

Nesne tespiti · iv, 2, 3, 4, 5, 6, 7,
8, 9, 20, 21, 22, 25, 29, 31, 33,
43, 45, 46, 48, 49

O

Omurga · 33, 36, 37, 39

Opencv · 4, 31, 45

Ö

Ön işlem · 6, 21

Öz nitelik · 6

R

Renk · 3, 4, 5, 11, 13, 21

S

Sınır kutusu · 28, 29

T

Tarım · iv, 1, 2, 11, 12

Tutma ucu · 3, 5, 7

V

Vakum · 3

Veri seti · 6, 9, 10, 11, 21, 31, 32,
33, 37, 40, 43, 46, 48, 49

Y

Yapay zeka · 3, 5, 14, 15, 17



TEKNOVERSITE



teknoversite **AYRICALIĞINDASINIZ**

İSTE

