



**İSKENDERUN TEKNİK**  
ÜNİVERSİTESİ  
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

**YÜKSEK  
LİSANS  
TEZİ**

**DERİN ÖĞRENME  
TABANLI MEYVE AYRIŞTIRMA  
SİSTEMİ**

**Muhammed TELÇEKEN**

**BİLGİSAYAR MÜHENDİSLİĞİ  
ANABİLİM DALI**

**AĞUSTOS 2022**



**İSKENDERUN TEKNİK ÜNİVERSİTESİ**

**YÜKSEK LİSANS TEZİ**

Versiyon 01.01



# DERİN ÖĞRENME TABANLI MEYVE AYRIŞTIRMA SİSTEMİ

**Muhammed TELÇEKEN**

**YÜKSEK LİSANS TEZİ  
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI**

**İSKENDERUN TEKNİK ÜNİVERSİTESİ  
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ**

**Ağustos 2022**

Muhammed TELÇEKEN tarafından hazırlanan “DERİN ÖĞRENME TABANLI MEYVE AYRIŞTIRMA SİSTEMİ” 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. Yakup Kutlu

Bilgisayar Mühendisliği Anabilim Dalı, İskenderun Teknik Üniversitesi

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

.....  
.....

**Üye:** Dr. Öğr. Üyesi Selman HIZAL

Bilgisayar Mühendisliği Anabilim Dalı, Sakarya Uygulamalı Bilimler Üniversitesi

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

.....  
.....

**Üye:** Dr. Öğr. Üyesi Gökhan ALTAN

Bilgisayar Mühendisliği Anabilim Dalı, İskenderun Teknik Üniversitesi

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

.....  
.....

Tez Savunma Tarihi: 03/08/2022

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;

- Tez üzerinde Yükseköğretim Kurulu tarafından hiçbir değişiklik yapılamayacağı için tezin bilgisayar ekranında görüntülendiğinde asıl nüsha ile aynı olması sorumluluğunun tarafıma ait 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

Muhammed TELÇEKEN

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

DERİN ÖĞRENME TABANLI MEYVE AYRIŞTIRMA SİSTEMİ  
(Yüksek Lisans Tezi)

Muhammed TELÇEKEN

İSKENDERUN TEKNİK ÜNİVERSİTESİ  
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

Haziran 2022

ÖZET

Teknolojideki gelişmeler ile birlikte bilgisayarlı görü sistemleri birçok alanda kullanılmaya başlamıştır. Gelişen bilgisayarlı görü sistemlerinden tarım alanında da çeşitli şekillerde yararlanılmaktadır. Meyve ve sebzeler market ve pazarların raflarında yerini almadan birçok aşamadan geçmektedir. Özellikle hasat sonrasında meyve ve sebzelerin şekilsel bozuklukları, çürük olup olmadıklarının belirlenmesi bu aşamaların en önemlilerinden bir tanesidir. Günümüzde hasat sonrasında meyve ve sebzeler el yordamıyla çürük, bozuk veya sağlam olarak ayrıştırılmaktadır. El yordamıyla meyve ve sebzeleri sınıflandırmak uzun zaman almakta ve maliyetleri yükseltmektedir. Bilgisayarlı görü sistemlerindeki gelişmelerden meyve ve sebzelerin sınıflandırılması aşamasında yararlanmak mümkündür. Özellikle derin öğrenme modellerinin görüntü işlemedeki yüksek doğruluk oranları meyve ve sebzelerin doğru şekilde otomatik şekilde kusurlu, çürük veya sağlam olarak sınıflandırılması ve ayrıştırılmasında büyük bir fayda sağlayacaktır. Bilgisayarlı görü sistemlerinden meyve ve sebzelerin otomatik olarak ayrıştırılmasından faydalanarak hem zaman tasarrufu hem de maliyetin düşmesi sağlanabilir.

Bu tez çalışmasında, derin öğrenme tabanlı ayrıştırma sistemi tasarlanmıştır. Meyve ve sebzelerin otomatik olarak ayrıştırabilecek bantlı bir mekanik sistem tasarlanmıştır. Bantların üzerinden ilerleyen meyve ve sebzeler kusurlu, çürük ve sağlam oluşlarına göre 3 sınıfa ayrıştırarak şekilde tasarlanmıştır. Sistem derin öğrenme modelleri ile gerçek zamanlı nesne tespiti ve mekanik kısım olarak iki kısımdan oluşmaktadır. İlk olarak gerçek zamanlı nesne tespiti için farklı yaklaşımlar olmasına rağmen YOLO algoritması gerçek zamanlı nesne tespiti için daha hızlı olduğundan bu çalışmada YOLO algoritmasının 6 farklı mimarisi incelenmiştir. Sistemin kurulması ve test edilmesi için portakal meyvesinin kusurlu, çürük ve sağlam olarak sınıflandırılması gerçekleştirilmiştir. Üç sınıf için toplam 11238 portakal görüntüsünden oluşan yeni bir veri seti oluşturulmuştur. İkinci kısımda sistemin minimum donanım gereksinimleri belirlenmiştir. Yapılan prototip üzerinde gerçekleştirilen testlerde 210 portakal için **YOLOv5 modelinde %89,5** e kadar doğruluk elde edilmiştir.

Anahtar Kelimeler : Meyve, Bilgisayarlı Görü Sistemleri, Derin Öğrenme, YOLOv4-tiny, YOLOv4, YOLOv4-P6, YOLOv5, YOLOv6, YOLOv7, Portakal

Sayfa Adedi : 63

Danışman : Doç. Dr. Yakup KUTLU

DEEP LEARNING BASED FRUIT SEPARATION SYSTEM  
(M. Sc. Thesis)

Muhammed TELÇEKEN

ISKENDERUN TECHNICAL UNIVERSITY  
GRADUATE EDUCATION INSTITUTE

June 2022

ABSTRACT

With the developments in technology, computer vision systems have started to be used in many areas. Developing computer vision systems are also used in various ways in the field of agriculture. Fruits and vegetables go through many stages before they take their place on the shelves of markets and markets. One of the most important of these stages is to determine whether the fruits and vegetables are deformed and rotten, especially after the harvest. Today, after harvest, fruits and vegetables are separated by hand as rotten, damaged or intact. Manually sorting fruits and vegetables takes a long time and increases costs. It is possible to benefit from the developments in computer vision systems at the stage of classification of fruits and vegetables. Especially the high accuracy rates of deep learning models in image processing will be of great benefit in correctly automatically classifying and sorting fruits and vegetables as defective, rotten or healthy. By utilizing the automatic sorting of fruits and vegetables from computer vision systems, both time saving and cost reduction can be achieved.

In this thesis, a deep learning-based parsing system was designed. A belted mechanical system that can automatically sort fruits and vegetables has been designed. It is designed to separate the fruits and vegetables moving over the belts into 3 classes according to their defective, rotten and sound condition. The system consists of two parts as real-time object detection and mechanical part with deep learning models. First of all, although there are different approaches for real-time object detection, since the YOLO algorithm is faster for real-time object detection, 6 different architectures of the YOLO algorithm are examined in this study. Classification of orange fruit as defective, rotten and intact was carried out to establish and test the system. A new dataset consisting of a total of 11238 orange images was created for the three classes. In the second part, the minimum hardware requirements of the system are determined. In the tests carried out on the prototype, up to 89.5% accuracy was obtained in the YOLOv5 model for 210 oranges.

Key Words : Fruit, Computer Vision Systems, Deep Learning, YOLOv4-tiny, YOLOV4, YOLOv4-P6, YOLOv5, YOLOv6, YOLOv7, Orange

Page Number : 63

Supervisor : Doç. Dr. Yakup KUTLU

## TEŐEKKÜR

Engin tecrübe ve bilgisiyle her zaman en iyisini başarabilmem için desteęini benden hiçbir zaman esirgemeyen danıřman hocam Doç. Dr. Yakup KUTLU' ya, yıllardır tarifi mümkün olmayan destek ve rehberlikleri ile kişisel gelişimimde ve kariyer planlarımı başarmamda yardımcı olmakla kalmayıp inancımı diri tutmamı sağlayan başta Annem ve Babam olmak üzere aileme şükranlarımı sunuyorum.

Son olarak da tez çalışmamda sistem tasarımının çizim aşamasında desteęini esirgemeyen Ömer Faruk ALTIN kardeşime teşekkürü bir borç biliyorum.





**İÇİNDEKİLER**

	<b>Sayfa</b>
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. LİTERATÜR İNCELENMESİ.....	3
2.1. Görüntü İşleme Teknikleri.....	3
2.2. Derin Öğrenme Tabanlı Yaklaşımlar.....	5
2.3. Bilgisayarlı Görü Sistemli Yaklaşımlar.....	6
3. METERYAL VE METODLAR .....	9
3.1. Deneysel Çalışma.....	9
3.2. Veri Seti.....	9
3.3. Sistem Tasarımı .....	10
3.3.1. Sistem gereksinimleri .....	13
3.4. Yapay Zekâ.....	17
3.4.1. Makine Öğrenmesi.....	19
3.4.2. Derin öğrenme.....	20
3.5. Gerçek Zamanlı Nesne Tespiti .....	23
3.5.1. YOLO algoritması .....	26

	<b>Sayfa</b>
3.5.2. Evrişimli sinir ağı.....	30
3.6. Performans Ölçümü.....	31
<b>4. ARAŞTIRMA BULGULARI VE SONUÇLAR.....</b>	<b>34</b>
4.1. Derin Öğrenme Tabanlı Meyve Ayrıştırma Sistemi Portakal Uygulaması .....	34
4.2. Veri Setinin oluşturulması .....	35
4.3. Nesne Tespiti .....	36
4.4. Uygun Model Tespiti .....	36
4.4.1. YOLOv4.....	36
4.4.2. YOLOv4-tiny .....	39
4.4.3. YOLOv4-p6 .....	41
4.4.4. YOLOv5 .....	42
4.4.5. YOLOv6.....	45
4.4.6. YOLOv7 .....	48
4.5. Gerçek Zamanlı Sistem Denemeleri .....	51
<b>5. TARTIŞMA VE ÖNERİLER .....</b>	<b>56</b>
5.1. Tartışma.....	56
5.2. Öneriler .....	57
<b>KAYNAKLAR.....</b>	<b>59</b>
<b>EKLER.....</b>	<b>65</b>

## ÇİZELGELERİN LİSTESİ

<b>Çizelge</b>	<b>Sayfa</b>
Çizelge 3.1. Veri setinin detaylı gösterimi.....	10
Çizelge 3.2. Karmaşıklık matrisi .....	32
Çizelge 4.1. Gerçek zamanlı testlerde elde edilen fps değerleri.....	54
Çizelge 4.2. Prototip üzerinde 210 portakalın sınıflandırma doğruluk çizelgesi .....	55
Çizelge 5.1. Literatürdeki farklı çalışmalar ile bu çalışmadaki farklı modellerin sınıflandırma başarımları karşılaştırılması .....	57

## ŞEKİLLERİN LİSTESİ

Şekil	Sayfa
Şekil 3.1. Programlanabilir mantık denetleyici yapısı .....	17
Şekil 3.2. Yapay zekâ ve alt dallarını gösteren bir görsel.....	18
Şekil 3.3. Klasik programlama ve makine öğrenmesi genel yapısı .....	19
Şekil 3.4. İki gizli katmanlı bir sinir ağının nöronlar arası bağlarının bir görseli.....	21
Şekil 3.5. Makine öğrenmesi ile derin öğrenme algoritmalarının çalışma yapıları.....	22
Şekil 3.6. Nesne tespitinin genel yapısındaki başlıklar.....	23
Şekil 3.7. Nesne takip yöntemlerinin sınıflandırma yapıları .....	25
Şekil 3.8. YOLO algoritması genel yapısı .....	30
Şekil 3.9. Özgünlük ve hassasiyet değerleri ile özgünlük, hassasiyet eğrisi .....	33
Şekil 4.1. Çalışma akış diyagramı.....	34
Şekil 4.2. DenseNet yoğun blok yapısı.....	36
Şekil 4.3. DenseNet yoğun blok katman bağlantı yapısı ve ağırlık güncelleme denklemi .	37
Şekil 4.4. CSP DenseNet ileri beslemeli geçiş ve ağırlık besleme denklemleri.....	37
Şekil 4.5. Farklı türde özellik birleştirme stratejileri .....	38
Şekil 4.6. YOLOv4 mimarisinin genel yapısı.....	38
Şekil 4.7. YOLOv4 kayıp miktarı ve ortalama kesinlik değerleri grafiği .....	39
Şekil 4.8. YOLOv4-tiny mimarisinin genel ağ yapısı .....	40
Şekil 4.9. YOLOv4-tiny kayıp miktarı ve ortalama kesinlik değerleri grafiği.....	41
Şekil 4.10. YOLOv-p6 mimarisi genel yapısı .....	41
Şekil 4.11. YOLOv4-p6 kayıp miktarı ve ortalama kesinlik değerleri grafiği .....	42
Şekil 4.12. YOLOv5 mimarisi genel ağ yapısı .....	43
Şekil 4.13. YOLOv5 ortalama kesinlik ve kayıp değerleri grafiği .....	44

<b>Şekil</b>	<b>Sayfa</b>
Şekil 4.14. YOLOv5 karmaşıklık matrisi .....	44
Şekil 4.15. YOLOv6 mimarisi Rep-PAN boyun yapısı.....	46
Şekil 4.16. YOLOv6 mimarisi ayrılmış kafa yapısı .....	46
Şekil 4.17. YOLOv6 mimarisi eğitim ortalama kesinlik ve kayıp miktarı sonuçları ...	47
Şekil 4.18. YOLOv6 mimarisi karmaşıklık matrisi .....	47
Şekil 4.19. E-ELAN mimarisinin yapısını gösteren görseller .....	49
Şekil 4.20. YOLOv7 mimarisi bileştirme yapısının görseli .....	49
Şekil 4.21. YOLOv7 mimarisi ortalama kesinlik değerleri ve kayıp miktarının Grafikleri .....	50
Şekil 4.22. YOLOv7 mimarisi karmaşıklık matrisi .....	50
Şekil 4.23. Kurulan sistemin çalışma prensibi akış diyagramı .....	52

**RESİMLERİN LİSTESİ**

<b>Resim</b>	<b>Sayfa</b>
Resim 3.1. Oluşturulan veri setinden örnek resimler .....	9
Resim 3.2. Otomatik ayrıştırma sistemi tasarım görseli .....	11
Resim 3.3. Eğimli konveyör bant sisteminin görselleri .....	12
Resim 3.4. Sistemdeki kameraların örnek görseli .....	12
Resim 3.5. Ayrıştırma noktasının görseli.....	13
Resim 3.6. Otomatik ayrıştırma sisteminin örnek bir görseli .....	13
Resim 3.7. Sevo motor çalışma yapısının bir görseli .....	15
Resim 3.8. Girdi resmi üzerinde ızgaraların uygulanması ile ilgili görsel.....	27
Resim 3.9. Evrişimli sinir ağının yapısı .....	30
Resim 4.1. Veri seti oluşturulurken çekilmiş portakal görüntüleri.....	35
Resim 4.2. Etiketlenmiş portakal görüntüleri .....	35
Resim 4.3. YOLOv5 modelin test verileri tahmin sonuçlarından görseller.....	45
Resim 4.4. YOLOv6 modelin test verileri tahmin sonuçlarından görseller.....	48
Resim 4.5. YOLOv7 modelin test verileri tahmin sonuçlarından görseller.....	51
Resim 4.6. Sistem prototipinin bir görseli .....	53

## SİMGELER VE KISALTMALAR

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

<b>Kısaltmalar</b>	<b>Açıklamalar</b>
<b>BBDC</b>	Geri Yayılım Tabanlı Ayırt Edici Sınıflandırıcı
<b>CPU</b>	Merkez İşlem Birimi
<b>DC</b>	Doğru Akım
<b>DHD</b>	Dairesel Hough Dönüşümü
<b>DÖ</b>	Derin Öğrenme
<b>DVM</b>	Destek Vektör Makinası
<b>GKT</b>	Görsel Kelime Torbası
<b>GYSA</b>	Geri Yayılımlı Sinir Ağı
<b>GZNT</b>	Gerçek Zamanlı Nesne Tespiti
<b>HSİ</b>	Ton,doygunluk ve Yoğunluk anlamına gelir.
<b>KDA</b>	Kuadratik Diskriminant Analizi
<b>KND</b>	Kademeli Nesne Dedektörü
<b>K-NN</b>	K En Yakın Komşu Algoritması
<b>LDA</b>	Liner Diskriminant Analizi
<b>LR</b>	Lojistik Regresyon
<b>MCPCA</b>	Maksimal İlişkili Temel Bileşen Analizi
<b>MOB</b>	Maksimum Olmayanı Bastırma
<b>MÖ</b>	Makine Öğrenmesi
<b>OTSU</b>	Nobuyuki Otsu adını taşıyan eşitleme yöntemidir.
<b>ÖDÖD</b>	Ölçek Değişmeyen Özellik Dönüşümü
<b>PLC</b>	Programlanabilir Lojik Kontrolörler
<b>RD</b>	Renk Dedektörü
<b>RGB</b>	Renkleri ortaya çıkarmaya yarayan bir metodudur.
<b>ROI</b>	İstenilen bölgeye piksellerle erişip işlem yapmaktır.
<b>RTOSA</b>	Radyal Tabanlı Olasılıksal Sinir Ağı
<b>SBK</b>	Saniye Başına Kare
<b>SBKB</b>	Sadece Bir Kez Bakarsın

<b>SPSOB</b>	Silinebilir Programlanabilir Salt Okunur Bellek
<b>YGH</b>	Yönlendirilmiş Gradyanları Histogramı
<b>YSA</b>	Yapay Sinir Ağları
<b>YZ</b>	Yapay Zekâ





## 1. GİRİŞ

Endüstri devrimleri, her zaman Dünyayı küresel ölçekli olarak etkilemiştir. Tüm meslek gruplarının gelişimini incelediğinizde gelişimlerinin temelinde sanayi devrimlerini görmemiz mümkündür. Son olarak Endüstri 4.0 olarak adlandırılan sanayi devrimi tanımlanmıştır. “Endüstri 4.0, üretim sürecindeki tüm aktörlerin birbirleriyle haberleşmesine, verilere eş zamanlı olarak erişmesine ve bu veriler sayesinde yüksek katma değerler üretilmesine zemin hazırlamıştır.” (Özsoylu,2017). Endüstri 4.0 ile bilişim teknolojileri ile sanayinin bir araya gelmesi amaçlanmıştır. Bugün gelinen noktada yeni nesil yazılım ve donanımlar bilişimin sanayide kullanılması için büyük imkanlar sunmaktadır. Endüstri 4.0 ile birlikte bilgisayarlı görü sistemleri birçok endüstriyel alanda olduğu gibi tarımsal alanda da kullanılmaya başlamıştır. Tarım alanında özellikle taze meyve ve sebzeler tüketiciye sunulmadan önce birçok aşamadan geçmektedir. Bu aşamaların en önemlilerinden bir tanesi ürünlerin hasat edildikten sonra doğru şekilde sınıflandırılması ve ayrıştırılmasıdır. Günümüzde hasat sonrasında taze meyve ve sebzeler klasik bir şekilde el ile ayrıştırılmaktadır. El ile meyve ve sebzelerin kusurlarından ayrıştırılması uzun zaman almakta ve maliyetleri belli bir ölçüde yükseltmektedir. Tarımsal Araştırmalar ve Politikalar Genel Müdürlüğü 2019 yılındaki yayınlamış olduğu çalıştay raporunda Dünyadaki toplam meyve üretiminde büyük oranda artış olmasına rağmen hasattan sonra yanlış uygulamalar neticesinde toplam üretimin %30-40 oranında israf edildiğini belirtmiştir. Bu durum hem üreticilerin gelirlerini etkilemekte hem de market ve pazar fiyatlarında artışa sebep olmaktadır. (Gerdan,2020). Zamandan tasarruf etmek ve maliyetleri düşürmek için meyve ve sebzelerin sınıflandırma ve ayrıştırmasında bilgisayarlı görü sistemlerinden yararlanmak mümkündür. Bilgisayarlı görü sistemlerinden yararlanarak otomatik ayrıştırma sistemi kurmak için öncelikle ayrıştırılacak meyve ve sebzelerin gerçek zamanlı nesne tespit yöntemleri ile kamera görüntülerinden tespit edilmesi gerekmektedir. Tespit edilen meyve ve sebze görüntüleri daha sonra doğru şekilde sınıflandırılmalıdır. Gerçek zamanlı nesne tespiti yapmak ve tespit edilen nesnelere sınıflandırmak için klasik yöntemlerle görüntülerden öznitelik çıkartma ve makine öğrenmesi yaklaşımıyla sınıflandırmak mümkündür. Nesneyi tespit ve takip etmek için en çok kullanılan öznitelikler renk, kenar, doku, derinlik ve hareket olarak söylenebilir. Makine öğrenmesi ile çıkartılan öznitelikleri sınıflandırmak için birçok algoritma mevcuttur. Derin öğrenme yöntemlerinin sunulması ile gerçek zamanlı nesne tespiti ve takibi yapılan yöntemlerde sunulmuştur. Derin öğrenme modelleri ile birlikte görüntülerden öznitelik çıkartmak zorunluluğu ortadan kalkmıştır.

Derin öğrenme modelleri içinde bulundurduğu farklı katmalar sayesinde görüntülerin öznitelikleri kendisi çıkartmakta ve sonrasında sınıflandırma yapabilmektedir. Gerçek zamanlı nesne tespiti için makina öğrenmesi yöntemleri yerine derin öğrenme modelleri kullanmak nesnelere tespiti ve takibini hızlandırmıştır. Bu çalışmada meyve ve sebzelerin sınıflandırma ve ayrıştırılmasını en hızlı ve en doğru şekilde yapabilmek için gerçek zamanlı nesne takip algoritması olan YOLO algoritması tercih edilmiştir. Sistemin en başarılı şekilde oluşturmak için YOLO algoritmasının **YOLOv4-tiny, YOLOv4, YOLOv4-p6, YOLOv5, YOLOv6 ve YOLOv7** mimarilerinde eğitimler gerçekleştirilecektir. Nesne tespit, sınıflandırma ve takip konusunda YOLO mimarileri karşılaştırılacaktır. En iyi ortalama kesinlik değeri elde edilen YOLO mimarisinde eğitilmiş model sisteme entegre edilmiştir. Kurulacak bant sistemine iki adet kamera yerleştirilecektir. Kamera görüntülerinden önceden eğitilmiş YOLO modeli ile meyve ve sebzeler tespit edilecek ve sınıflandırılacaktır. Sınıflandırılması yapılmış olan meyve veya sebze bant sistemindeki tetikleyiciler sayesinde buldukları sınıf için belirlenen sepetlerin hizasına geldiğinde bant sisteminin dışına itileceklerdir. Toplam üç sınıf için tasarlanan bu sistemde kusurlu ve çürük sınıflar için tetikleyiciler bant sistemine yerleştirilmiştir. Kusurlu ve çürük olarak sınıflandırılmayan her meyve veya sebze bant sisteminin sonuna yerleştirilen sepete sağlam meyve veya sebze olarak düşecek ve bant sisteminin dışında kalacaktır. Sistem tasarlanırken iki şeye dikkat edilmiştir. İlk olarak doğru şekilde meyve veya sebzelerin sınıflandırılması ikinci olarak ta zaman açısından en hızlı şekilde işlemlerin gerçekleştirilmesi amaçlanmıştır.

Literatür incelendiğinde yapay zekâ üzerine yapılan çalışmaların endüstriyel makinalara dönüştürülmesi yapılan çalışmalara oranla düşük kalmaktadır. Bu tez çalışmasında hedeflenen en önemli nokta yapılacak olan yapay zekâ çalışmasının endüstriye yönelik bir çalışma olmasıdır. Endüstriye dönüşecek her yaklaşım ile katma değerli ürünler ortaya çıkacak, bu ürünler ülke ekonomisine fayda sağlayacaktır. Çalışmada fayda maliyet analizi göz önünde bulundurularak yapay zekâ çalışmaları gerçekleştirilecektir.

## 2. LİTEATÜR İNCELEMESİ

Meyve veya sebzelerin sınıflandırılması için yapılmış olan çalışmaları incelediğimizde iki farklı yaklaşımın benimsendiği görülmektedir. Meyve veya sebzeleri sınıflandırmak için ilk olarak klasik görüntü işleme teknikleri kullanılarak meyve veya sebze resimlerinin öznitelikleri çıkartılarak farklı sınıflandırma algoritma performansları incelenmiştir. İkinci yaklaşım olarak derin öğrenme modellerine dayalı sınıflandırma yöntemleri kullanılmıştır. Derin öğrenme modelleri ile sınıflandırma yaparken meyve ve sebze görüntülerinin iyileştirilmesi için ön işlem olarak görüntü işleme teknikleri kullanılıyor olsa da resimlerin özniteliklerinin çıkartılmasına gerek duyulmamaktadır. Derin öğrenme modelleri tarafından verilerin özellikleri çıkartılmakta ve sınıflandırılmaktadır.

### 2.1. Görüntü İşleme Teknikleri

Meyve ve sebzelerin sınıflandırılması yapılırken sınıflandırma algoritmalarının çoğu resimlerin özniteliklerinin çıkartılmasına ihtiyaç duymaktadır. Görüntülerden özniteliklerin çıkartılması için ön işlem olarak uygulanan yöntemler mevcuttur. Sınıflandırma algoritmalarından önce görüntü işleme teknikleri ile özniteliklerin çıkartıldığı çalışmalarla yapılan meyve veya sebze sınıflandırma çalışmalarına örnekler verecek olursak, (Surya, Kumar, 2015) yapmış oldukları çalışmada, muz meyvesinin olgunluk aşamalarının incelenmesinde muz görüntülerinden öznitelik çıkartmak için arka plan temizleme, nesnenin bölgesinin belirlenmesi, renk ve boyut değerlerinin belirlenmesi gibi görüntü işleme yöntemleri kullanmış ve çıkartılan özniteliklerin sınıflandırılma performanslarını geliştirmiş oldukları renk ortalama yoğunluk değerleri ve alan değerleri algoritmaları ile incelemişlerdir. Mohammadi, Kheiralipour ve Varnamkhasti (2015) Hurma meyvesinin olgunluk seviyesini incelemek için yapmış oldukları çalışmada, Hurma meyvesinin görüntüleri elde ettikten sonra öncelikle görüntülere arka plan temizle işlemi uygulamış daha sonra görüntülerin özelliklerini çıkarmak için bölütleme işlemi uygulayıp dört temel özellik çıkarmış, sınıflandırma performansını incelemek için Liner Diskriminant Analizi ve Kuadratik Diskriminant Analizi algoritmalarını incelemişlerdir. Capizzi, G., Lo Sciuto, Tramontana ve Wonzniak (2016) portakalların kusurlu bölgelerinin belirlenmesi için yapmış oldukları çalışmada Radyal Tabanlı Olasılıksal Sinir Ağı sınıflandırma algoritmasını önermişler, kusurlu bölgelerin tespitinde %88'e kadar doğruluk elde etmişlerdir. Sahu, Demeshwari ve Dewangan (2017) mango meyvelerini şekil, renk ve boyut olarak otomatik

sınıflandıracak bir sistem geliştirmek için yapmış oldukları çalışmada mango meyvelerinin görüntülerinin özellikleri çıkartırken ön işlem olarak görüntülerin arka planlarını temizlemiş sonrada görüntülerin renklerini griye dönüştürmüşlerdir, görüntülerdeki mango meyvelerinin köşelerini bulduktan sonra toplam beyaz alan piksel sayısını hesaplayabilmek için bwarea fonksiyonunu kullanarak mango meyvelerini sınıflandırmışlardır. Hussain Hassan ve Nashat (2019) zeytin meyvesinin kusurlarını otomatik olarak tespit edebilmek için önermiş oldukları yöntemde, zeytin görüntülerine ön işlem uygulayarak resimlerin arka planlarını temizlemiş sonra resimleri bölütlemişlerdir. Bölütledikleri zeytin görüntülerinin özelliklerini çıkartmada son aşama olarak kusurlu pikselleri çıkartmak için bir eşik değeri ataması yapmışlardır. Atamış oldukları eşik değeri değişkendir ve kusurlu alan ile nesnenin normal olan bölgesi arasındaki farkın derecesine göre değişmektedir. Eşik değerin altında kalan görüntülerdeki zeytinleri kusurlu diğer görüntülerdeki zeytinleri de kusursuz olarak sınıflandırmışlardır. Chithra ve Henila (2019) muz ve elma meyvelerinin tespiti için yapmış oldukları çalışmada RGB olarak yükledikleri görüntüleri HSI ye dönüştürmüşler sonra resimlerin arka planlarını çıkarmışlardır. OTSU yöntemini kullanarak 'T' eşiğini elde etmişlerdir. Aynı yöntemi kullanarak ROİ çıkartmış ve delikleri doldurmuşlardır. ROİ görüntülerini yükledikten sonra özellikleri çıkarmak için dikey, yatay ve diyagonal değerleri çıkarmak için Haar filtresi kullanarak istatistiksel ve dokusal özellikleri çıkarmışlardır. Çıkardıkları özellikleri DVM ile sınıflandırmışlardır. Carolina ve David (2014) ananas meyvelerini olgunluk aşamalarının belirlenmesi için yapmış oldukları çalışmada meyvelerin renklerinin oluşumunu anlamak için HSV renk modeli uyguladılar, daha sonra ilgili bölgeyi çıkartmak için görüntüye bölütleme işlemi uygulamış ve renk tonu ve doygunluğunu hesaplamışlardır. Kaur, Sawhney ve Jawandha (2018) erik meyvesinin olgunluk seviyelerini tespit etmek için yapmış oldukları çalışmada ortalama RGB değerlerini kullanarak renk özelliklerini çıkarmış, doğrusal özellikleri çıkarmak için entropi, yerel ikili model ve ayırık kosinüs dönüşümü kullanmışlardır. Sınıflandırma performansını incelemek için çok nitelikli karar verme teorisini kullanmışlardır. Mohammadi, Khojastehnazhand ve Minaei (2019) kayısı meyvesinin olgunluk ve hacminin tespiti için yapmış oldukları çalışmada kayısı meyvesinin görüntülerinin özellikleri çıkartırken görüntüyü kırpma, RGB ve  $L^*a^*b$  değerleri ile renk özellikleri çıkartılmış ve kayısı meyvesinin hacmini belirlemede piksel bazında sıyırma yöntemi kullanmışlardır, daha sonra LDA ile sınıflandırmışlardır. Azarmdel, Jahanbakhshi, Mohtasebi ve Muñoz (2020) dut meyvelerini olgunluklarına göre sınıflandırmak için yapmış oldukları çalışmada dut görüntülerinden geometrik özellikleri, renk ve doku özelliklerini çıkartmak için korelasyona dayalı özellik çıkarma ve tutarlılık alt

kümesi özellik indirgeme yöntemlerini kullanarak özellikleri çıkarttılar ve sınıflandırmak için DVM ve YSA da sınıflandırma başarılarını incelediler. Alam Siddiquee ve diğerleri (2020) domateslerin kusurlu, kusursuz ve olgunlaşmış, olgunlaşmamış olarak tespit edilmesi için yapmış oldukları çalışmada KND ile RD yaklaşımlarını birleştirerek DHD yöntemini domateslerin sınıflandırılmasında incelemişlerdir. Kumari ve Belwal (2021) mango meyvesinin sınıflandırılması için yapmış oldukları çalışmada görüntü özelliklerini çıkarmada geometrik, doku ve renk temelli özellikleri tercih etmişler sınıflandırma doğruluğunu geliştirmek için K-ortalamar kümeleme algoritmasını bölütleme verimliliğini artırmak için kullanmışlar ve özellik seçmek için MCPA yaklaşımını tercih etmişlerdir. Sınıflandırma başarımları için BBDC ile Naive Bayes sınıflandırıcıları karşılaştırmışlardır. Bhargava ve Bansal (2021) farklı elma türlerinin sınıflandırılmasını incelemek için ilk olarak görüntüleri tut-kes yöntemi ve bulanık K-ortalamar kümelemesi ile bölütlemişler sonra görüntülerden istatistiksel, dokusal, geometrik özellikler ayırık dalgacık dönüşümü, yönlendirilmiş gradyan ve kanunların doku enerjisi yöntemleri ile özellik uzayından temel bileşen analizi gerçekleştirerek çıkarmışlar ve sınıflandırma başarımlarını incelemek için K-NN, LR, LDA ve DVM algoritmalarının performanslarını karşılaştırdılar. Zhang, Satapathy ve Wang (2022) meyvelerin sınıflandırılması ve paketlenmesi için yapmış oldukları çalışmada ilk olarak öznelikleri çıkartmak için dönüş açısı vektör ızgarası ile 2- Boyutlu fraksiyonel fourier entropisi kullandılar ve sınıflandırmak için beş katmanlı bir yığın seyrek otomatik kodlayıcı önerdiler.

## 2.2. Derin Öğrenme Tabanlı Yaklaşımlar

Taze meyve ve sebzelerin sınıflandırılmasında DÖ tabanlı çalışmalarda literatürde mevcuttur. Hossain, Al-Hammadi ve Muhammad (2018) meyvelerin sınıflandırılması için yapmış oldukları çalışmada iki farklı veri seti için iki farklı derin öğrenme mimarisi önerdiler, birinci mimarileri altı katmanlı ESA modeli, ikincisi ise on altı katmanlı önceden eğitilmiş bir ESA modelidir. Haggag ve diğerleri (2019) otomatik olarak çürük, olgunlaşmamış ve olgunlaşmış domateslerin sınıflandırılması için yapmış oldukları çalışmada DÖ ve YZ yöntemlerini kullandılar ve ESA ve YSA tabanlı melez bir algoritma önerdiler. Jahanbakhshi, Momeny, Mahmoudi ve Zhang (2020) limon meyvesinin kusurlu olanlarını tespit etmek için yapmış oldukları çalışmada limon görüntülerine birkaç ön işlem uyguladıktan sonra ESA modeliyle sınıflandırdılar ve sonuçlarını MÖ yöntemlerinden birkaç tanesini ile karşılaştırdılar. En iyi sınıflandırma doğruluğunu ESA modeli ile elde

ettiler. Li Z., Li F., Zhu ve Yue (2020) sebzelerin otomatik olarak tanınması için yapmış oldukları çalışmada ESA modellerinden geliştirilmiş VGG ağ modelini kullandılar. Elsharif, Dheir, Mettleq ve Abu-Naser (2020) dört patates türünü belirlenmesi için ESA modeli eğitip sınıflandırma performansını incelediler. Chakraborty ve diğerleri (2021) meyvelerdeki çürüklerin belirlenip ayrıştırılması için elma, muz ve portakal meyve görüntüleri üzerinde yapmış oldukları çalışmada ESA modellerinden max havuzlama, ortalama havuzlama ve MobileNetV2 mimarisi ile sınıflandırma sonuçlarını incelediler. Roy, Shaudhuri ve Pramanik (2021) çürük ve taze elmaların tespit edilmesi için yapmış oldukları çalışmada ESA modellerinden bölümlenme için geliştirilmiş U-Net mimarisini önerdiler. Chougule, Pawar, Kamble, Mujawar ve Bhide (2021) meyve ayıklama sistemi tasarlamak için yapmış oldukları çalışmada dokuz çeşit taze meyve ve beş çeşit çürük meyve için ESA modeliyle meyve tahmini gerçekleştirdiler. Chen, An, Gao, Li ve Kang (2021) kusurlu portakal meyvelerini otomatik olarak ayrıştırılması için tasarladıkları sistemde konveyör bant üzerinde rastgele dönen portakalların kusurlu olanlarını tespit etmek için ESA tabanlı bir dedektör geliştirdiler ve bu sınıflandırma bilgilerini kaydetmek için sıralama algoritması tabanlı bir izleyici tasarladılar. Leelavathy, B., Sri Datta, Y.S.S. ve Rachana (2021) portakal meyvesinin çürük ve sağlam olarak sınıflandırılması için yapmış oldukları çalışmada ESA modeli kullanmış, ESA modelinde kayıp fonksiyonu olarak ikili çapraz entropi kullanarak portakal meyvesini %78,57 doğrulukla sınıflandırmışlardır. Abu-Jamie, Abu-Naser, Alkahlout ve Aish (2022) altı çeşit meyvenin sınıflandırılması ve tanınması için yapmış oldukları çalışmada ESA modellerinden VGG16 mimarisini kullandılar. Aish, Abu-Naser ve Abu-Jamie (2022) dolmalık biberlerin olgunluk seviyelerini tespit etmek için yapmış oldukları çalışmada ESA da eğitim için transfer öğreniminde ağırlık değeri olarak ImageNet değerlerini kullanarak VGG16 mimarisinde eğitimlerini gerçekleştirdiler.

### **2.3. Bilgisayarlı Görü Sistemli Yaklaşımlar**

Meyve veya sebzelerin ayrıştırılmasında, kusurlu ve çürük olanlarının tespit edilmesinde bilgisayarlı görü sistem yaklaşımı çalışmalarda mevcuttur. Sofu, Er, Kayacan ve Cetişli (2016) üç elma türünü renk, ağırlık, boyut ve kusurlu kusursuz olarak sıralamak için bilgisayarlı görüden faydalanılarak otomatik elma sıralama sistemi tasarlamışlardır. Yapmış oldukları sistem makine görüşüne sahip bir kabin, yük hücresi ve kontrol merkezi ile entegre edilmiş makaralı ve sınıf konveyör taşıyıcıdan oluşmaktadır. Kapalı kabin içerisinde elmaların görüntülerini almak için iki tane endüstriyel kamera kullanılmışlardır. Elmaları

tanımlamak ve sınıflandırmak için iki farklı konveyör bant kullanmışlardır. Birinci konveyör bant elmanın her yönden görüntüsünü almak için elmayı döndürerek kapalı kabin içerisinden ileri doğru taşımaktadır. Kabinde görüntüsünü alınan elmaların konumlarını kaybetmemek için ikinci konveyör bant üzerinde kaseler bulunmaktadır. Kaselerin içerisinde ki elmalar tetikleyiciler tarafından sınıflarına ait bölgelerde kaselerden boşaltılmaktadır. Elmaları renklerine göre sınıflandırmak için RGB renk modelini kullanmışlardır. Elmaların boyutlarını belirlemek için elma görüntüsünün orta noktasından enine ve boyuna uzunluklarının piksel değerleri santimetre değerine dönüştürülmüş, bu uzunluklar elmanın çapı olarak kabul edilmiştir. Elmaların belirlenen çapına göre elmalar daha sonra üç farklı boyutta sınıflandırılmıştır. Elmaların kusurlu veya kusursuz olmalarını belirlemek için pencere tabanlı bir yaklaşım kullanmışlardır. Pencere tabanlı yaklaşımda elma görüntüleri 5x5 piksellik parçalara bölünerek ortalamaları ve kalan parçaların değerlerinin hesaplanmasında da istatistiksel standart sapma değeri kullanılmaktadır. Kurmuş oldukları otomatik elma sıralama sisteminde renk tonları birbirine yakın olan Granny Smith ve Golden Delicious türü elmaların sıralamasında birbirine karıştığını gözlemlemişlerdir. Önerdikleri makine ile bir günde 8 saat çalışarak 432.000 elma/gün ve %79 sıralama doğruluğu elde etmişlerdir. Korchagin ve diğerleri (2021) konveyör bant üzerinde hareket eden patateslerin hasarlı ve hastalıklı olanlarını tespit etmek için bilgisayarlı görü sistemi kullandılar. Yapmış oldukları çalışmada ilk olarak aksiyon kamerası ile alınan görüntülerde patates yumrularını algılamak için Viola-Jones algoritması uyguladılar, daha sonra hastalıklı patatesleri ayırtmak için ÖDÖD ve DVM algoritmalarını kullanmışlardır. Zayıf aydınlatma durumları için YGH, GKT ve GYSA yöntemlerini kullandılar. Bant hızına bağlı optimum ayarlarda %80'den %97'e kadar yükselen doğruluk elde etmişlerdir. Mukhtar, Jamil, Arif, Razzaq ve Wasif (2021) limonları sıralamak ve derecelendirmek için yapmış oldukları çalışmada olgun, yarı olgun ve kusurlu olgunlaşmamış limonların ayırtılması için kurdukları bilgisayarlı görü sisteminde limon görüntülerini sistemdeki kameradan alarak Raspberry Pi de görüntülerin analizini gerçekleştirdiler. Limon görüntülerini standart bir şekilde almak için kamera ile görüntüleri alacakları düzlemi ışık kaynağı ile aydınlattılar ve görüntüleri bu şekilde kaydettiler. Kamera ile alınan limon görüntülerine ön işlem olarak kırpma, arka plan temizleme ve gürültü gidermek için Gauss filtresi uyguladılar, daha sonra özellikleri çıkartmak için renk özelliği için mavi, kırmızı ve yeşil kanalların ortalama değerlerini kullandılar, görüntüdeki limonun altındaki piksel sayısını sayarak limonun boyutunu belirlediler. Kusurlu meyvelerin yüzey yapısını belirlemek için global standart sapma, yerel standart sapma ve bağlamdan bağımsız olan uzaysal filtre uyguladılar.

Özellikleri çıkartılan limon görüntülerini sinir ağı kullanarak sınıflandırıp sonuçları incelemiştir. Chopra ve diğerleri (2021) otomatik olarak meyvelerin ayrıştırılması için yapmış oldukları çalışmada yapay zekâ tabanlı bilgisayar görü sistemi tanıttılar. Kurulan sistemde meyvelerin görüntüleri almak için Raspberry Pi kamera kullanılmış, kamera 360 derece döner bir sisteme yerleştirilmiştir. Döner bir sistemden meyvenin görüntülerini alarak meyvenin her yüzeyinin görüntülerini almayı amaçlamışlardır. Sisteme entegre edecekleri ESA modellerini eğitip test etmek için fruit-360 veri setini kullandılar, önerdikleri model ile meyveleri tanıma konusunda %97,97 doğruluk elde ettiler. Meyve ayrıştırma sistemlerinin testi için elma meyvesi üzerinde testlerini gerçekleştirdiler. Sistem üzerinde alınan elma görüntülerine ön işlem olarak kırpma ve yeniden ölçeklendirme işlemi uygulamışlardır. Elma meyvelerini tanımlamak için CustomVision.AI web sitesinden yararlanarak sistem üzerinde elma tanımlamada %82 doğruluk elde etmişlerdir.



### 3. MATERYAL VE METODLAR

#### 3.1. Deneysel Çalışma

Bu çalışmada tasarlanan derin öğrenme tabanlı meyve ayrıştırma sisteminin mekanik olarak çalışmasında kontrolcüye gelecek sinyaller ile ayrıştırma sistemi gerçekleştirileceğinden sistemin mekanik olarak hatasız çalışması mümkün olsa da sistemin asıl zorluğu derin öğrenme yani yapay zekâ olacaktır. Sistem için belirlenecek yapay zekâ yaklaşımı sistem performansının hız ve doğruluk konusunda belirleyicisi olacağından, tasarımı yapılan ayrıştırma sisteminin deneysel çalışmalarını gerçekleştirip en verimli derin öğrenme modelinin belirlenmesi için elde var olan mekanik malzemelerle bir prototip yapılması düşünülerek, sisteme en uygun derin öğrenme modelinin belirlenmesi için portakal meyvesinin kusurlu, çürük ve sağlam olanlarından görüntüler toplanarak veri seti oluşturulmuş ve YOLO algoritmasının 6 farklı mimarisinde eğitimler gerçekleştirilmiştir.

#### 3.2. Veri Seti

Derin öğrenme modellerinin eğitilmesi ve sistemin test edilmesi için kullanılacak veri setini oluşturmak için pazar, market ve manavlarda portakal meyvesinin görüntüleri belirlenen ölçeklerde fotoğraflanarak yeni bir veri seti elde edilmiştir. Elde edilen yeni veri setinde görüntülerin arka planları sabit değildir. Derin öğrenme tabanlı bir sistem tasarlanacağı için kameralardan alınacak görüntülerde ki gürültülü görüntülerini de tanınması ve gürültü olarak algılayabilmesi için görüntülerin arka planları temizlenmemiş ve eğitimler gürültülü görüntüler üzerinden gerçekleştirilmiştir. Oluşturulan yeni veri setinden örnek görüntüler çürük, kusurlu ve sağlam olarak sıralısı ile Resim 3.1’de gösterilmektedir.



Resim 3.1. Oluşturulan veri setinden örnek resimler

Elde edilen veri setinde bulunan portakal görüntüleri üç sınıftan oluşmaktadır. Veri setinde ki üç sınıf için var olan portakal görüntülerinin dağılımı dengeli değildir. Veri setindeki portakal görüntülerinin sınıflara göre görüntü sayıları Çizelge 3.1’de detaylı olarak gösterilmiştir.

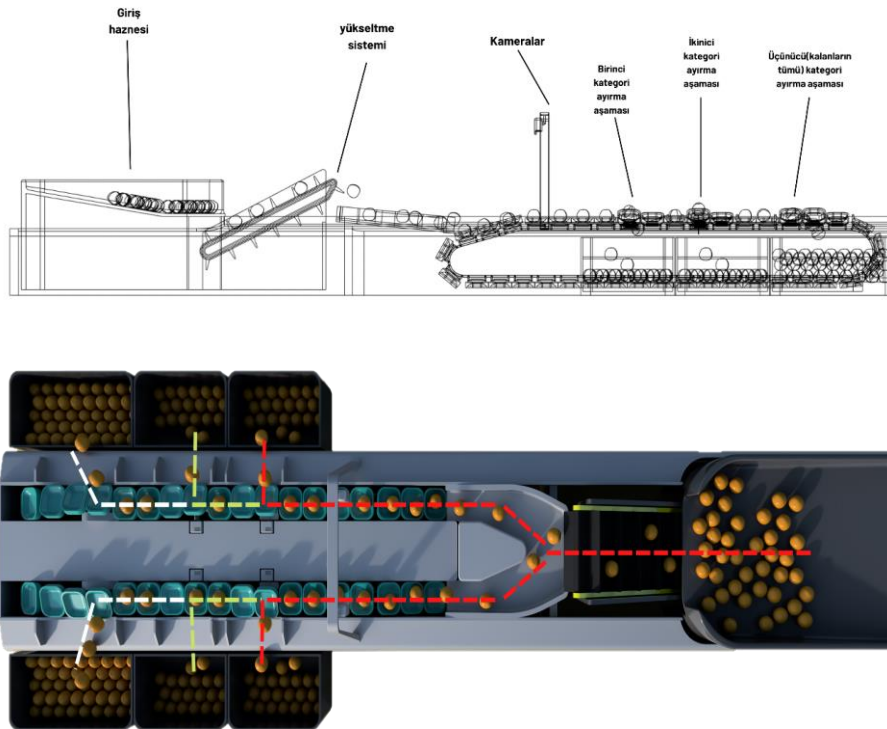
Çizelge 3.1. Veri setinin detaylı gösterimi

Tür Adı	Görüntü Sayısı
Sağlam	3891
Çürük	3652
Kusurlu	3695
<b>Toplam</b>	<b>11238</b>

### 3.3. Sistem Tasarımı

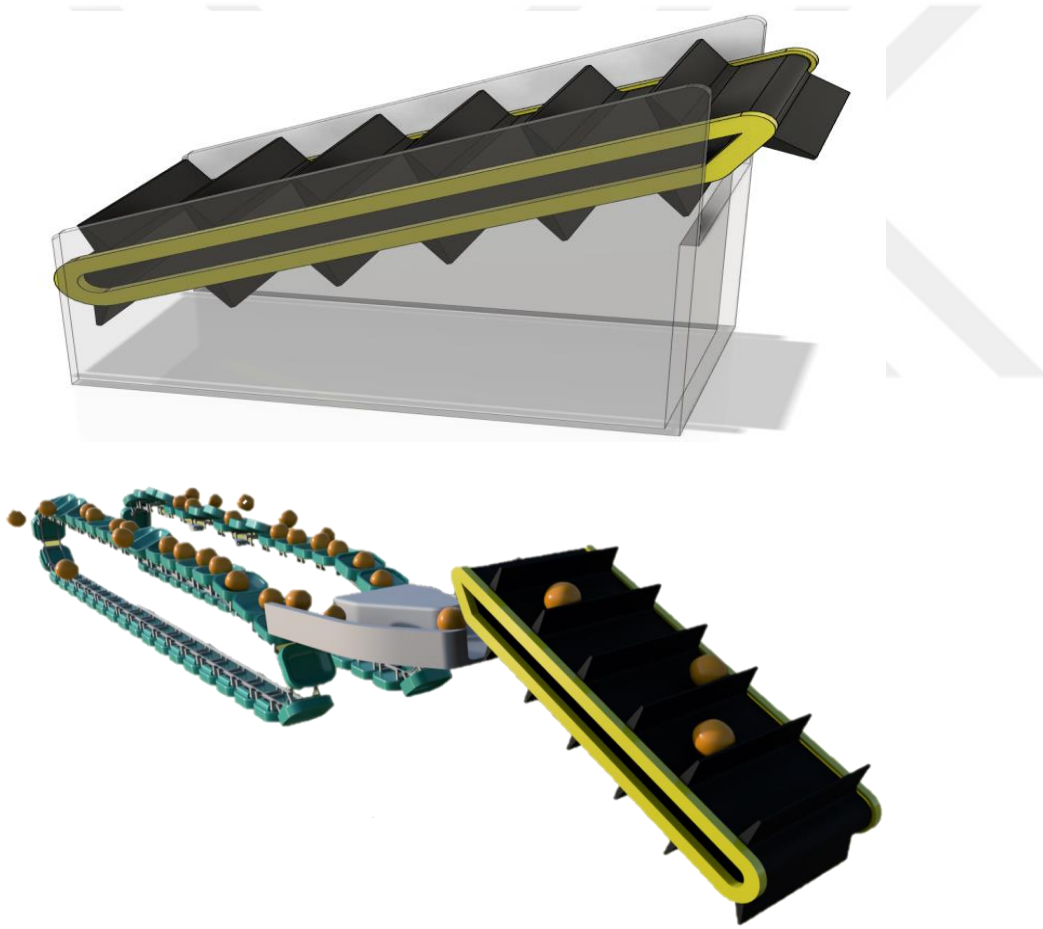
İki sıra şeklinde tasarlanan sistemde meyvelerin taşınması için konveyör bant üzerine taşıma kapları yerleştirilmesi düşünülmektedir. Meyveler eğimli bir şekilde yerleştirilen konveyör bantla başlangıçta bulunan hasat sonrası ürünlerin içerisine boşaltıldığı hazneden ayrıştırılmak için taşınacakları kaplarının içerisine aktarılacaktır. Hasat edilen ürünlerin eğimli bir konveyör bant sistemi ile başlangıç haznesinden ayrıştırma kaplarına taşınmasının sebebi ürünlerin ayrıştırma kaplarının içerisine teker teker düşmesini sağlamaktır. Ayrıştırılacakları kaplar iki sıra halinde konveyör bantlar üzerinde yerleştirilecek ve her kap, içinde bulunan ürünü kendi etrafında hareket ettirecek hareketli bilye sistemi olacak şekilde tasarlanacaktır. Tasarlanan taşıma kabı yapısı ile ürünlerin kusurlu veya çürük kısımlarının

kabın iç kısımlarında kalmasıyla kameraların görüş alanı dışında kalmasının önüne geçilerek daha doğru sınıflandırma başarımı elde etmek amaçlanmıştır. Sistem de iki adet 1080p çözünürlükte 60 FPS kamera kullanılması planlanmaktadır. Kameralar bant sisteminin üzerinde kapların içindeki meyvelerin görüntülerini en iyi alabilecek şekilde yerleştirilecektir. Kameralar ile alınan görüntüler sisteme bağlı olan 4GB'lık Jetson Nano Developer Kit'e gönderilecek, Jetson içerisinde kaydedilmiş olan önceden eğitilmiş model ile görüntüsü alınan ürün sınıflandırılarak, sınıf bilgisi PLC'ye gönderilecektir. Otomasyon sistemin kontrolünün sağlandığı PLC alınan bilgiyi ayrıştırma kaplarına yönlendirerek ürünün ait olduğu sınıfın bulunduğu sepet hizasına geldiğinde sisteme bağlı servo motorlar yardımıyla sepetlere düşmesi sağlanacaktır. Sistem tasarımında meyvelerin en doğru şekilde ayrıştırılmasını sağlayacak model belirlenmeye çalışılırken aynı zamanda sistemin ayrıştırma hızının da en iyi seviyede olabileceği şekilde donanımlar tercih edilmesine önem verilecektir. Konveyör bantların sistemdeki dönüş hızını ayarlayarak görüntünün en iyi şekilde alınması, doğru bir şekilde sınıflandırılıp ayrıştırılması ve ayrıştırma noktalarındaki servo motorların devreye girme zamanlarının doğru şekilde belirlenmesi için konveyör bantların hareketi step motorlarla gerçekleştirilmesi düşünülmektedir. Tasarladığımız otomatik ayrıştırma sisteminin yapısını gösteren görsel Resim 3.2'de gösterilmektedir.



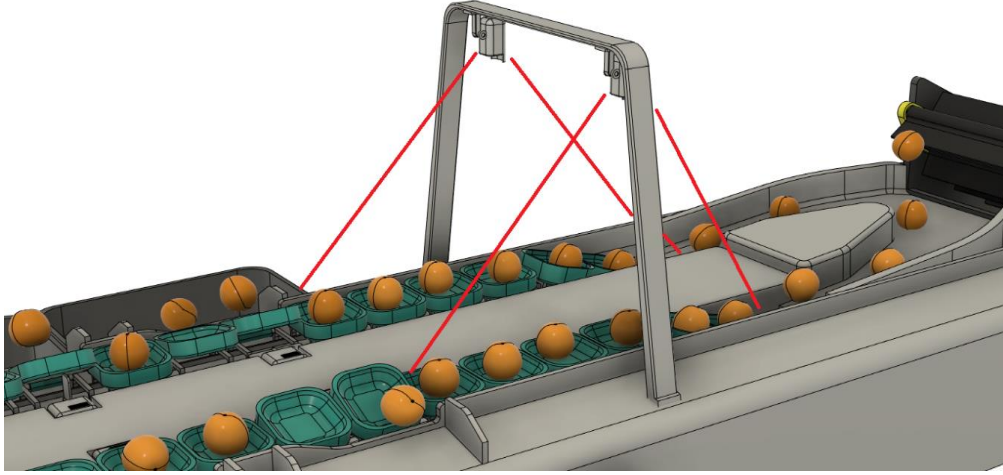
Resim 3.2. Otomatik ayrıştırma sistemi tasarım görseli

Sistemin en doğru şekilde çalışması için ilk olarak taşıma kaplarının içerisine meyvelerin teker teker düşmesini önemlidir. Taşıma kaplarının içerisine birden fazla meyvenin düşmesi, hem taşıma kaplarından meyvelerin taşması gibi bir probleme hem de farklı sınıflara ait meyvelerin aynı kap içerisine düşmesi ile ayrıştırma sırasında farklı sınıfların aynı noktadan sistem dışına itilerek yanlış ayrıştırılmasına neden olabilir. Oluşabilecek bu problemlerin önüne geçmek için başlangıç haznesiyle taşıma kaplarının olduğu bant sistemi arasında hafif eğimli bir konveyör bant sistemi yerleştirilmesi tasarlanmıştır. Bu taşıma bandı sayesinde başlangıç haznesinden kaplara meyveler istenilen hız ve aralıklarla taşınarak taşıma ve yanlış ayrıştırma gibi oluşabilecek problemlerin önlenmesi amaçlanmaktadır. Eğimli konveyör bant sisteminin yapısı ile ilgili görseller Resim 3.3'te görülmektedir.



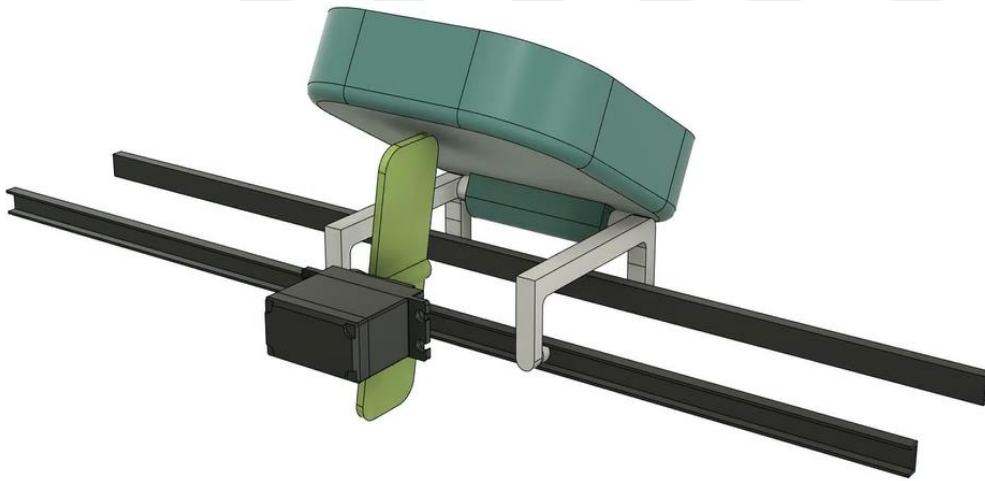
Resim 3.3. Eğimli konveyör bant sisteminin görselleri

Meyvelerin taşıma kaplarından görüntülerinin en iyi şekilde alınabilmesi için iki bantlı olarak tasarlanan sistemde her bant üzerine görüntüleri almak için kamera yerleştirilecektir. Tasarlanan sisteme entegre edilecek kameraların yapısının Resim 3.4'te görüldüğü gibi olacaktır.



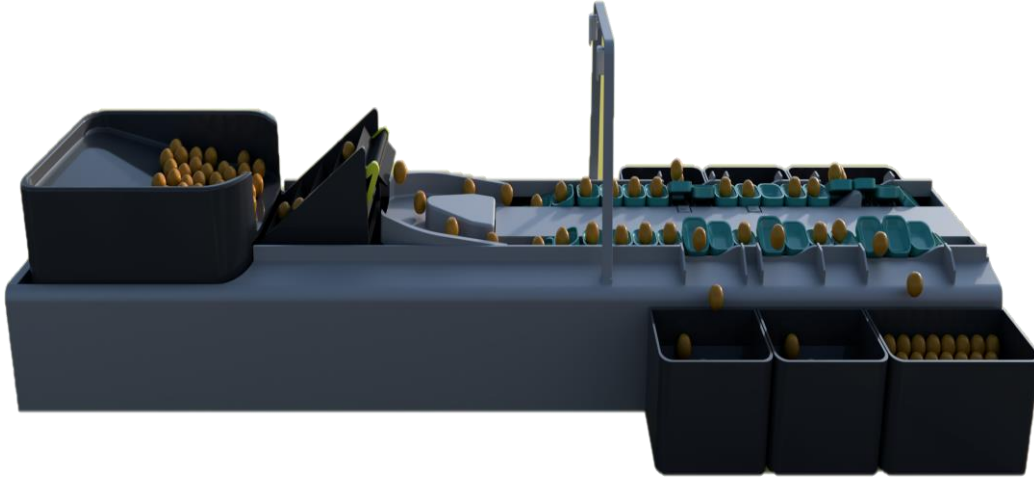
Resim 3.4. Sistemdeki kameraların örnek görseli

Görüntüler alındıktan sonra ait olduğu sınıfı belirlenen meyveler taşıma kaplarının içerisinden ait oldukları sınıfların sepetleri hizasına geldiğinde servo motorlarla kontrol edilen ayırıştırma noktalarındaki döner kanatlar sayesinde kaplardan sepetlerin içerisine boşaltılacaktır. Ayırıştırma noktasının tasarımı Resim 3.5’te görülmektedir.



Resim 3.5. Ayırıştırma noktasının bir görseli

Derin öğrenme tabanlı meyve ayırıştırma sisteminin mekanik tasarımının yapısı Resim 3.6’da görülmektedir.



Resim 3.6. Otomatik ayırıştırma sisteminin örnek bir görseli

### 3.3.1. Sistem gereksinimleri

Bu tez çalışmasında tasarlanan meyve ayırıştırma sistemin donanımlarını belirlerken derin öğrenme tabanlı olacağı için donanımların öncelikle gerçek zamanlı nesne tespiti konusunda uyumlu olmasına ve oluşabilecek problemlere kolay çözümler bulabilecek donamlar olmasına dikkat edilmiştir. Derin öğrenme tabanlı sistemlerde sınıflandırmayı yapacak mimarilerin başarısı kadar tercih edilen donanımların bilgisayarlı görü alanında görüntüleri alırken çözünürlüğü ve hızlı iletişimi önemli konulardır. Aynı zamanda donanımların maliyeleri de sistem gereksinimlerini tercih ederken önemli bir nokta olmuştur.

Konveyör bant: Bant sistemlerinin gelişmesi, kayış malzemesinin globalleşmesinden, kaliteli ve kullanılabilir olmasından kaynaklanmıştır. İlk konveyör bantlar mukavemetinden dolayı derilerden yapılmıştır. 18. Yüzyılda gemilere ürünlerin yüklenmesi için insan gücünden yararlanılarak ahşap kanallar üzerine bant sistemleri kurulmuştur. Diğer alanlarda olduğu gibi sanayi devrimleri ile beraber bu alanda da ilerlemeler kaydedilmiştir. Gelişmeler ile birlikte bant sistemlerinde insan gücü kullanılması yerine motor gücü kullanılmaya başlandı. İlk olarak deriden üretilen bantlar zamanla çelik ve plastik olarak da üretilmeye başlanmıştır. Kapalı devre olan konveyör bantların üretilmiş oldukları malzeme ve yapısına göre seçilmesi taşınacak ürün ile temas halinde olacağından oldukça önemlidir. Aynı zamanda bandın taşıyacağı malzemenin ağırlığını da tartabilmesi önemli bir husustur. Bu tez çalışmasında taşıma için konveyör bantlar üzerinde meyveleri taşınması için özel kaplar tasarlanmıştır. Kaplar, içindeki ürünleri istenilen alanda boşaltabilmek için 60 derece dışı doğru kalkabilecek şekilde tasarlanmıştır.

Servo motor: Sistemlerdeki doğrusal ve açısal pozisyon ve hız kontrolünü hatasız şekilde yapmak için tasarlanmış tetikleme sistemine Servo denir. Sistemdeki hareketleri kontrol etmek üzerine tasarlanmış bir motor türüdür. Günümüzde robot teknolojilerinde en çok kullanılan motor türü servo motorlardır. Uzaktan kumanda kontrol sistemlerinde de kullanılmaktadır. Servo motorlar sistemden yeni bir komut alana kadar bulunduğu pozisyonu korumak üzerine tasarlanmıştır. Servo motorların hareketini içlerinde bulunan DC motorlar sağlamaktadır. Servo motorların içerisinde DC motorlardan başka dişli mekanizması, dışarıdan fiziksel olarak sinyallerle değerleri değiştirilebilen direnç ve motor sürücü devresi vardır (Lui, Hwang, Chen ve Fu, 2009). Genelde 180 derece açı ile çalışan servo motorlar vardır ama özel amaçla üretilen 360 derece çalışma açısı olan servo motorlar da vardır. Servo motorlar kendilerine dışarıdan güç uygulandığında buldukları konumu korumak için direnç gösterirler. Servo motorların AC servo motorlar ve DC servo motorlar olarak iki türü bulunmaktadır. DC motorlar içinde bulundurdukları mekanizmaya göre farklılık gösterirler. Çekirdeksiz DC motorlar mekanizmalarında mıknatıs yerine kablolar ile manyetik alan oluşturlar, bu yüzden hem hafif hem de tepki süreleri daha kısadır. Bu tez çalışmasında gelen sinyal sonrası tepki verme hızının iyi olması ve pozisyon geri bildirimli olması sebebi ile Feetech marka FS5103-FB model standart servo motor tercih edilecektir. Resim 3.7’de servo motor yapısı görülmektedir.



Resim 3.7. Servo motor çalışma yapısının bir görseli

Step motor: Adım adım hareket eden fırçasız bir DC motor türüdür. Geri beslemek için sensör barındırmadan konumlandırılabilmesi için çok kullanışlı bir motor türüdür. Step motorların stator sargıları belirlenmiş sıra ile harekete geçtiğinden içinden bir akım geçtiğinde statorun mıknatıslanması ile sıralı bir şekilde elektromanyetik itmeler oluşturacaktır. Oluşan bu elektromanyetik itmelere step denmektedir (Meng, Yu, 2009). Step motorun tam bir dönme sağlanması için belirli sayıda step gerekmektedir. Farklı adım

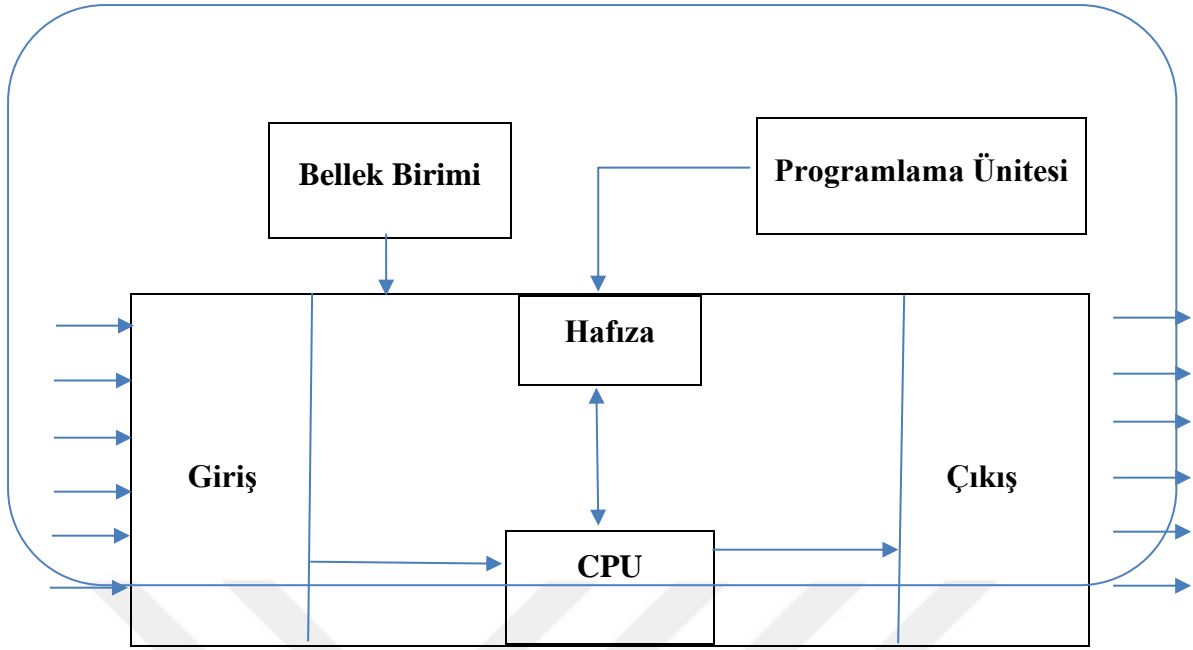
sayılarına sahip step motorlar mevcuttur. Step motorların dönme hızlarının ayarlanabilir olması sebebiyle bu tez çalışmasında tasarlanan otomatik ayrıştırma sisteminin konveyör bantlarını hareket ettirmek için step motor tercih edilecektir.

Jetson nano geliştirici kartı: NVIDIA şirketinin geliştirmiş olduğu yapay zekâ uygulamalarında kullanılmak üzere tasarlanmış güçlü bir bilgisayardır. Küçük yapısına rağmen sağlamış olduğu geniş kullanım yönüyle bilgisayarlı görme, grafikler, multimedya ve daha fazla alanda rahatlıkla kullanılmaktadır. Bu tez çalışmasında Jetson Nano geliştirici kartı olarak 4 GB tercih edilmiştir. Tercih edilen Jetson Nano HDMI 2.0 çıkışı ve 4 adet USB 3.0 çıkışı ile sistemde kullanmaya elverişli olması, 128-core NVIDIA Maxwell GPU desteği ve 4 GB'lık hafızası ile sistem üzerinde kameralardan görüntülerin 1080p çözünürlükte alınmasına ve önceden eğitilerek içine kaydedilmiş modelde görüntülerin sınıflandırması için gerekli imkânı sağlayacaktır. Maliyet analizi olarak incelendiğinde yapay zekâ bilgisayarları içerisinde sistem minimum gereksinimlerini karşılayacak en uygun yapay zekâ bilgisayarı olması sebebi ile Jetson Nano Geliştirici kartının 4 GB'lık modelinin tercih edilecektir.

Kamera: Konveyör bant sistemi üzerindeki kapların içerisinde bulunan meyvelerin görüntülerinin alınması için kullanılmaktadır. Gerçek zamanlı nesne tespiti ve sınıflandırılması yapılacağı için kamera seçiminde 60 FPS 1080p Logitech marka kamera tercih edilecektir. Kullanılacak kamera belirlenirken sistemin minimum gereksinimlerini karşılayacak en uygun kameranın belirlenmesine dikkat edilmiştir.

Programlanabilir mantık denetleyicisi: Otomasyon sistemlerinde PLC'ler temel bir noktayı oluşturmaktadır. PLC'ler endüstrinin farklı alanlarında çok yaygın bir şekilde kullanılmaktadır. PLC fabrikalarda üretim kısımlarının veya makinaların kontrollerinin denetimlerinde kullanılmakta olan bir otomasyon cihazıdır. PLC'lerin otomasyon sistemlerinde en çok tercih edilen donanım olmalarının sebebi elektriksel sorunlara, sıcaklığa ve darbelere karşı dayanıklı bir yapıya sahip olmalarıdır. PLC'ler de birçok giriş ve çıkış kapısı bulunmaktadır (Aydoğan, 2019). PLC girdi bilgilerini tarayarak çıkış bilgilerini gerçek zamana çok yakın bir şekilde cevap olarak iletebilir. Kısa zamanda kaliteli ve çok sayıda ürün üretmek ve hata oranlarını düşürmek için otomasyon sistemlerinde PLC tercih edilmektedir. PLC dört ana bölümden meydana gelmektedir (Oluwaleye, Oguntosin, Idachaba, 2021). (Bkz. Şekil 3.1.) PLC cihazlarının genel yapısı görülmektedir.





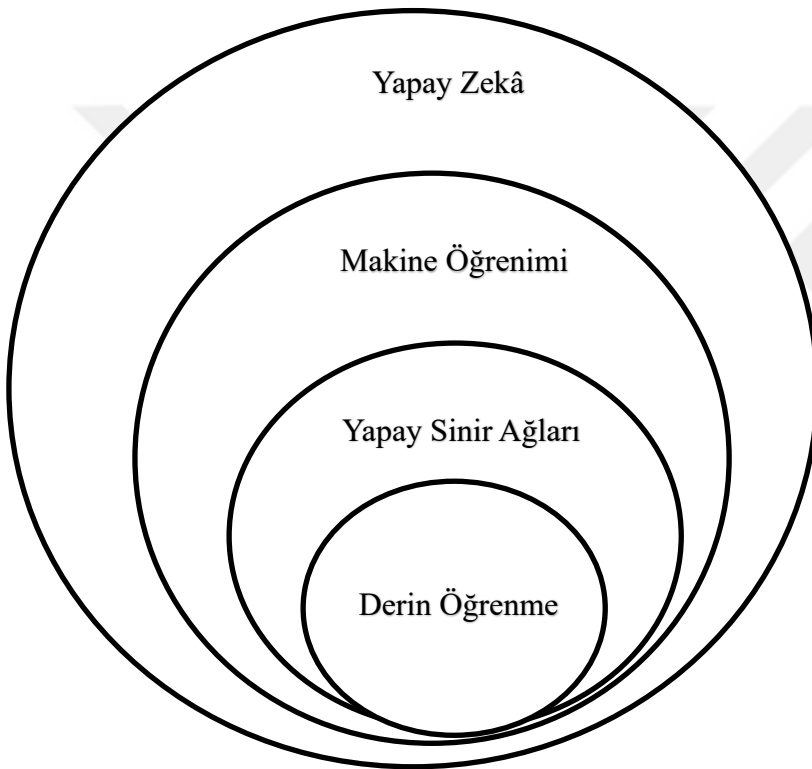
Şekil 3.1. Programlanabilir mantık denetleyicisinin yapısı

CPU, aritmetik ve mantık işlemlerinin yapıldığı, zamanlama ve sayma gibi görevleri üstlenen PLC içinde ki en önemli birimdir. Farklı PLC'ler aynı CPU kullanabilmektedir. Bellek birimde giriş, veri ve program belleği gibi kısımlar bulunmaktadır. Her bellek kısmının farklı işlevleri vardır. PLC'lerde genel olarak Silinebilir Programlanabilir Salt Okunur Bellek kullanılmaktadır. Giriş birimi oluşturulan otomasyon sisteminde sinyal gelen elemanlardan gelen giriş bilgilerini PLC'in anlayacağı mantık gerilim seviyesine dönüştüren birimdir. Kullanılan sensör çeşidine göre basınç, seviye, sıcaklık, kumanda ve yakınlık gibi elemanlardan yönlendirilen ikili değerler giriş birimden alınmaktadır (Aydoğan, 2019). Çıkış birimi PLC içerisinde hesaplanan çıkış elemanlarının mantık gerilim voltajını otomasyon sisteminde kumanda elemanlarına iletmeye uygun elektriksel işaretlere çeviren birimdir. PLC seçiminde maliyet, giriş-çıkış nokta sayısı, bellek kapasitesi, komutların işleme hızı, kesme işletim yeteneği ve çıkışların dielektrik dayanma seviyesi gibi hususlara dikkat edilmektedir.

### 3.4. Yapay Zekâ

Sanayi devriminden itibaren teknolojide büyük gelişmeler yaşanmıştır. Teknolojideki bu gelişmeler ile birlikte insanların yaparken zorlandığı birçok işi artık makinalar yapmaya başlamıştır. Makinalarında yaptıkları işleri öğrenebileceği düşüncesi yapay zekâ alanında çalışmalar yapılmasına sebep olmuştur. Üretilen makinalar ve programların ilham kaynağı

insan zekâsı olmuştur. Daha sonrasında insanların gündelik kullanım ihtiyaçlarını görecektir programlara ve makinalara yönelik çalışmalar yaygınlaşmıştır. 1970’lerde büyük bilgisayar firmaları kişisel bilgisayarları üretmeleri ile bilgisayar popüler hale gelirken yapay zekâ alanındaki çalışmalar daha dar bir alanda devam etmiştir. 2015 yılında Google derin öğrenme denilen yapay sinir ağını kullanan AlphaGo’yu tanıtmaması, yapay zekâ’nın popülerliğini artırmıştır. Yapay zekâ kavramı genel bir yaklaşımdır. Yapay zekâ; makine öğrenmesi, yapay sinir ağları ve derin öğrenme gibi kavramları kapsayan üst bir kavramdır. (Bkz. Şekil 3.2.) Yapay zekâ ve alt dalları detaylı olarak görülmektedir.



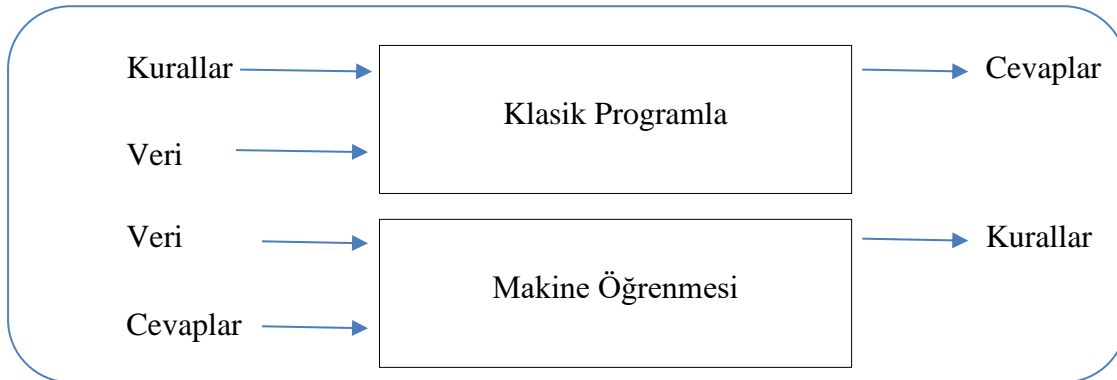
Şekil 3.2. Yapay zekâ ve alt dallarını gösteren bir görsel

Günlük yaşamda yapay zekâ karşımıza yazdıklarımızı tahmin edip otomatik olarak tamamlayan veya sosyal medyada genel olarak takip ettiğimiz içeriklere uygun içerikleri bize öneren bir yapıda çıkabilmektedir. YZ kullanılarak çözülen problemler her zaman çok karmaşık yapıda olmaya bilirler. Karışık bir sayı dizisini sıralamada veya iki şehir arasındaki en kısa yolu bulma konusunda da yapay zekâ kullanılmaktadır. Genel olarak yapay zekânın babaları olarak kabul edilen Minsky ve McCarthy yapay zekâ’yı “Daha önce insan zekâsı gerektirdiği düşünülen bir görevin bir makine tarafından gerçekleştirilmesi” olarak tanımlamışlardır. Bu tanım çok genel bir tanım olarak kabul edilmektedir. Google’da yapay zekâ araştırmacısı olarak çalışan ve makine öğrenmesi yazılım kütüphanesi Keras’ın

yaratıcısı olan Francois yapay zekâ'yı “Bir sistemin yeni bir ortama uyum sağlama ve doğaçlama yapma, bilgisini genelleştirme ve onu alışılmadık senaryolara uygulama becerisine bağlı” olarak nitelemiştir.

### 3.4.1. Makine öğrenmesi

Makine öğrenmesi temel olarak “Bilgisayarların herhangi bir şeyi bizim nasıl yaptığımızın dışına çıkarak kendi kendine yapabilmeyi öğrenmesi mümkün müdür?” veya “Bilgisayarın kullanıcıya sürpriz yaprak bir şeyler ortaya koyması mümkün müdür?” gibi sorulardan ortaya çıkmıştır (Chollet, 2021). MÖ insanların problemlerin çözümü için kurallar üretmesinin yerine makinaların problem çözümü için kuralları oluşturması yaklaşımıdır. YZ'nın klasik yaklaşımlarında programlama yapılırken programa girdi olarak veriler ve kurallar verilir, bu veri ve kurallara göre bir cevap elde edilmektedir. MÖ de programcı hem verileri hem de veriden elde edilecek cevapları girer ve buradan kurallar elde edilir (Chollet, 2021). Elde edilen kurallar yeni veriler üzerine uygulanarak cevaplar elde edilir. Makine öğrenmesinde program yazmak yerine eğitimler gerçekleştirilir. (Bkz. Şekil 3.3) Klasik programlama yöntemi ile MÖ'nin yapıları görülmektedir.



Şekil 3.3. Klasik programlama ve makine öğrenmesinin genel yapısı

Klasik programlama ile çözüm bulmakta zorlanılan karmaşık veriler üzerine çalışmada ve çözüm bulunamayacak problemlerde makine öğrenmesi yaklaşımı problemlerin çözümlerine en yakın sonuçları elde etme noktasında çok önemli bir yaklaşımdır. İstatiksel yaklaşımlarla sınıflandırma veya tahminler yapmak üzere makineler eğitilir. Makine öğrenmesi sadece öğrenen bir yaklaşım değil aynı zamanda performans iyileştirme içinde kullanılan bir yaklaşımdır. MÖ'de temel olarak denetimli ve denetimsiz makine öğrenmesi

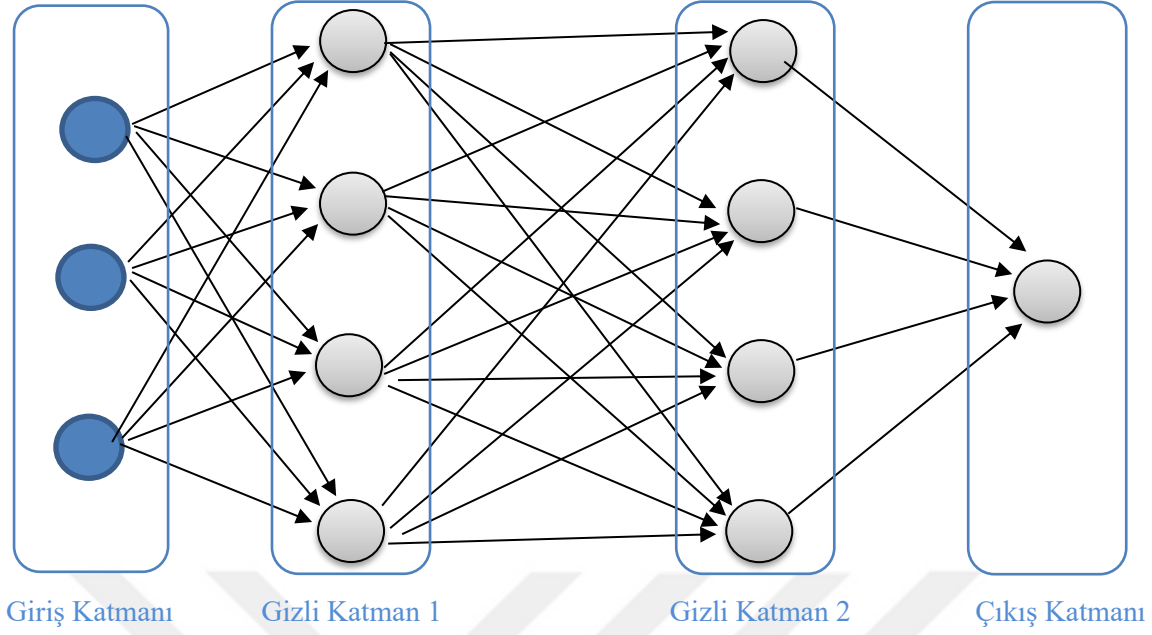
olarak iki öğrenme yaklaşımı bulunmaktadır. İki öğrenme yaklaşımı arasındaki temel fark tahminde bulunmak için verileri ne şekilde öğrendikleridir.

Denetimli makine öğrenmesi: Öğrenme sırasında uzman kişiler tarafından etiketlenmiş veriler üzerinden eğitimler gerçekleştirilerek makinanın öğrenmesi sağlanacaktır. Örnek olarak hiçbir şeyi henüz bilmeyen bir çocuğa meyve çeşitlerini öğretmek için meyveleri devamlı göstererek bu meyvenin ne olduğunu söylemek ve bu şekilde çocuğun meyvelerin türlerini öğrenmesini sağlamak gibi makinaya öğreneği verinin ne olduğunu devamlı tekrar ederek bildirip veriyi öğrenmesini sağlamaktır.

Denetimsiz makine öğrenmesi: Öğrenmenin gerçekleşmesi sırasında hiçbir uzman tarafından etiketlenmemiş veriler üzerinde gerçekleştirilen eğitimlerle makinanın öğrenmesine dayalı bir yaklaşımdır. Örnek olarak bir çocuğa meyveleri öğrenmesi için meyvelerin türü hakkında bilgi vermeden kendi gözlemleri ile meyveleri renk ve şekillerine göre gruplayarak öğrenmesidir. Makinaya veriler etiketsiz olarak verilir ve makine verileri benzerliklerine göre gruplayarak bu verilerin neler olduğunu öğrenir.

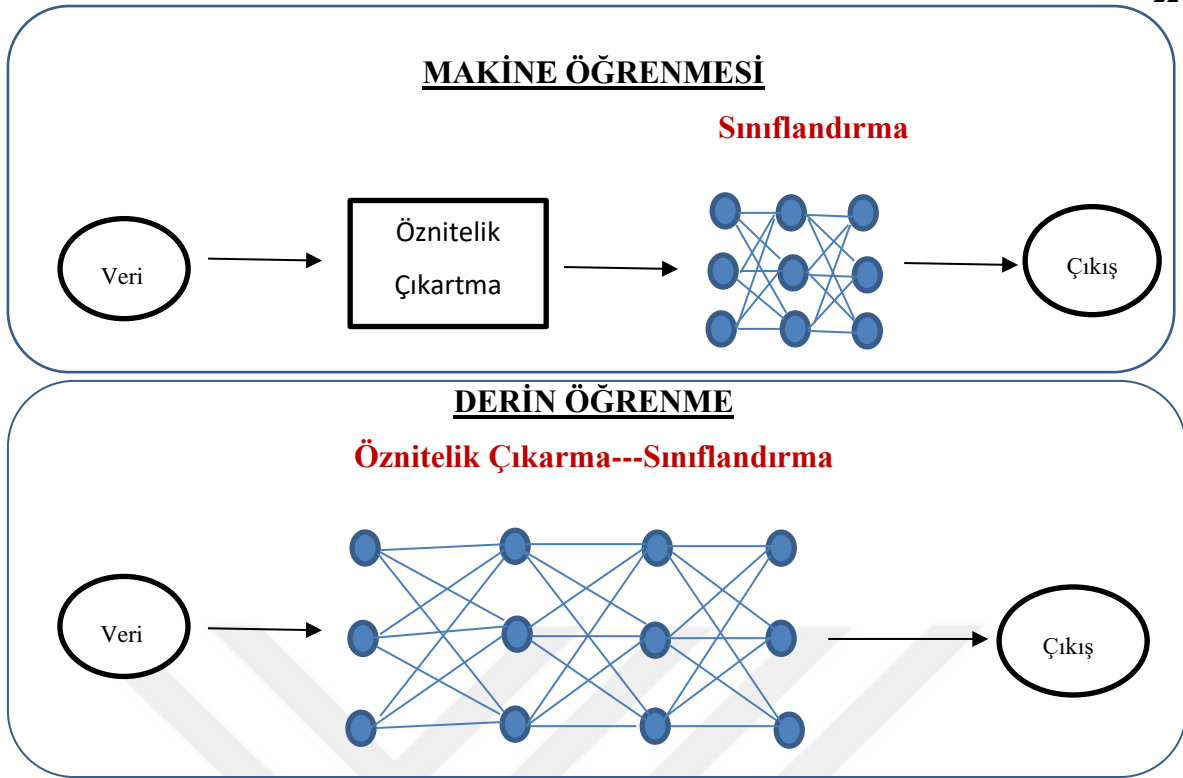
### 3.4.2. Derin öğrenme

Makine öğrenmesi yöntemi olan DÖ veri kümeleri üzerinde çıktıları tahmin edecek bir sinir ağını eğitmeye olanak sağlayan bir yaklaşımdır. DÖ yaklaşımındaki derin kelimesi birden çok gizli katmanın var olduğunu ifade etmektedir. DÖ de katmanlar üç farklı gruba ayrılmaktadır. İlk olarak giriş katmanı gelir onu gizli katmanlar olarak adlandırılan ara katmanlar takip eder ve son olarak çıkış katmanı bulunmaktadır. Giriş katmanında girdi verileri alınmaktadır. Gizli katmanlarda matematiksel hesaplamalar yapılır. Çıktı katmanında üretilen çıktı verileri döndürülecektir. DÖ de nöronlar arasındaki her bir bağlantı bir ağırlıkla ifade edilmektedir. İlk başlangıç olarak ağırlıklar rastgele olarak atanmaktadır. Bağlantılar arasındaki bu ağırlıklar girdi değerlerinin önemini belirten sayısal ifadelerdir. Her nöron için bir aktivasyon fonksiyonu atanarak çıktılar standartlaştırılır. Derin öğrenmenin önemli iki noktası bulunmaktadır. Birincisi derin öğrenmede sinir ağının eğitilmesi ile daha iyi sonuçlar elde etmek için eğitimde kullanmak üzere büyük bir veri kümesine ihtiyaç duyulmaktadır. İkincisi de sinir ağını eğitebileceğimiz çok iyi bir işlem gücüne sahip donanıma ihtiyaç duyulmaktadır. (Bkz. Şekil 3.4.) te sinir ağlarının genel yapısı gösterilmiştir.



Şekil 3.4. İki gizli katmanlı bir sinir ağının nöronları arasındaki bağlantıların bir görseli

MÖ yaklaşımları ile veri kümelerinin sınıflandırılması için verilerin temsil edilmesi için öncelikle verilerden en iyi öznelik matrisleri çıkartılmalıdır. Çıkartılan öznelik matrisleri MÖ algoritmaları ile sınıflandırılır. DÖ yaklaşımlarında veri kümelerinin sınıflandırılması için verilerin öznelik matrislerinin çıkartılmasına gerek duyulmamaktadır. DÖ algoritmalarına veriler ham olarak verilir, eğitim yapılacak model öncelikle verilerin en belirgin özneliklerini çıkartır ve sonra çıkartmış olduğu özneliklere dayalı bir sınıflandırma yapar. (Bkz. Şekil 3.5.) da MÖ yaklaşımı ile DÖ yaklaşımlarının sınıflandırma yapıları detaylı olarak gösterilmiştir. DÖ yaklaşımları sınıflandırma yapmak için daha kullanışlı bir yaklaşım olarak görünmüş olsa da veri sayısı DÖ algoritmalarının performanslarında çok önemli bir kriterdir. MÖ yaklaşımları ile sınıflandırmada çok iyi sonuç elde edilebilecek bir veri kümesinde veri sayısının dengesizliği veya az sayıda olması DÖ yaklaşımında iyi sonuçlar elde etmek için yeterli olmayabilir.



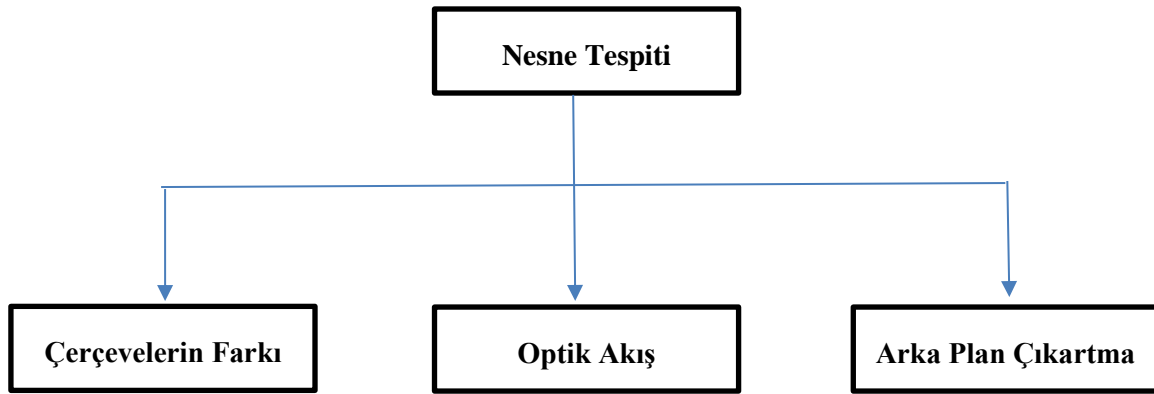
Şekil 3.5. Makine öğrenmesi ile derin öğrenme algoritmalarının çalışma yapısının gösterimi

Denetimli öğrenmede DÖ çok güçlü bir yaklaşımdır. DÖ katmanlarını artırarak veya katmanların içerisindeki birimlerin sayılarını artırarak çok karmaşık fonksiyonların görünmesini sağlayabiliriz (Goodfellow, Bengio ve Courville, 2016). Birçok fonksiyonun bir araya gelmesiyle ağ yapısı oluşur. Örneğin birbirine bağlanmış dört fonksiyonumuz olsun,  $f^1$ ,  $f^2$ ,  $f^3$  ve  $f^4$  fonksiyonları bir araya gelerek  $f(x) = f^4(f^3(f^2(f^1(x))))$  fonksiyonunu oluşturabilecektir. İç içe sıralanmış bu fonksiyonlarda  $f^1$  ağı oluşturan birinci katman,  $f^2$  ağı oluşturan ikinci katman olarak devam etmektedir. Bu iç içe geçmiş fonksiyonların uzunluğu bize modelin derinliğini göstermektedir. Son katman olan  $f^4$  fonksiyonu çıktı katmanıdır. Ağların eğitiminde  $f(x)$  fonksiyonunu  $f^*(x)$  'e doğru yakınsaması için iteriz. Eğitim kümesindeki veriler farklı eğitim noktalarından hesaplanmış şekilde  $f^*(x)$  yaklaşımları sağlayacaktır. Her  $x$  örneği ile beraber  $y \approx f^*(x)$  etiketi bulunmaktadır (Goodfellow ve diğerleri, 2016). Eğitimde kullanılan örnekler çıktı katmanında ki her  $x$  için ne yapılması gerektiğini belirleyerek  $y$  değerine en yakın değeri üretmeye çalışır. Algoritma her bir katmanın nasıl kullanılacağına kendisi karar verecektir. İstenilen çıktı bu katmanlarda gösterilmeyeceğinden dolayı bu katmanlara gizli katman denmektedir (Goodfellow ve diğerleri, 2016). Modellerin genişliği katmanların boyutları ile belirlenmektedir. Her gizli katman vektörel yapıda değildir (Goodfellow ve diğerleri, 2016). Katmanların vektörden vektöre tek bir fonksiyon ile gösterildiğini düşünmek doğru olmayacaktır. Katmanlar birçok

birimden oluşmaktadır ve bu birimler vektörden skalere birer fonksiyonu göstermektedir. Her birim birbirine paralel çalışırken bir birime birçok birimden girdi sağlanır ve bu şekilde her birim kendi aktivasyon fonksiyonunu hesaplar. Her modelin derlenmesi aşamasında bir kayıp fonksiyonu belirlenerek eğitimin her iterasyonunda modelin kayıp miktarı hesaplanacaktır. Kayıp fonksiyonları sınıflandırılacak eğitim kümesinin sınıf sayısına ve etiket yapısına bağlıdır.

### **3.5. Gerçek Zamanlı Nesne Tespiti ve Takibi**

Görüntü işleme yöntemleri ve derin öğrenme algoritmaları ile görüntüler üzerinde ki nesnelere tespit etmek ve sınıflandırmak artık çok daha kolay bir hale gelmiştir. Buna karşın video görüntüleri üzerinde tespit edilen nesnelere takip edilmesi daha karmaşık bir problemdir. Nesnelere tespit edilmesi ve takip edilmesi sıralı gelen resimler veya video görüntülerinden daha önceden sınıflandırılmış veya sınıflandırılmamış nesnelere konum, doğrultu ve hız gibi bilgilerinin bulunmasıdır. Günümüzde nesnelere tespit ve takibi konusunda birçok çalışma gerçekleştirilmektedir. Sıralı gelen iki görüntünün birbirinden çıkartılması ile basit bir şekilde nesne takibi yapmak için başlayan çalışmalar bugün gelişen derin öğrenme yöntemleri ile çok daha hızlı ve başarılı bir şekilde devam etmektedir (Hanbay, Üzen, 2017). Akıllı telefonların kameralarındaki yüz tanıma sistemleri ve araçların plaka tanıma ve takip sistemleri GZNT çalışmalarına örnek olarak gösterilebilir. Nesne takibi dört aşamada gerçekleştirilmektedir. Verilerin ön işlenmesi, nesnenin tespit edilmesi, nesnenin sınıflandırılması ve son olarak nesnenin takip edilmesi. Bu aşamalardan nesne tespiti çok önemli bir yere sahiptir. Çünkü sonraki aşamalarda yapılacak işlemlerinin başarı durumları nesne tespiti aşamasındaki doğru sonuçların elde edilmesine bağlıdır. Nesne tespitini videolarda nesnelere arka plan görüntülerinden arındırmak olarak tanımlayabiliriz (Karasulu, 2013). Literatürde nesne tespiti konusunda birden çok yöntemle karşılaşmak mümkündür. (Bkz. Şekil 3.6.) Ama genel olarak nesne tespitini üç adımda gruplandırabiliriz.



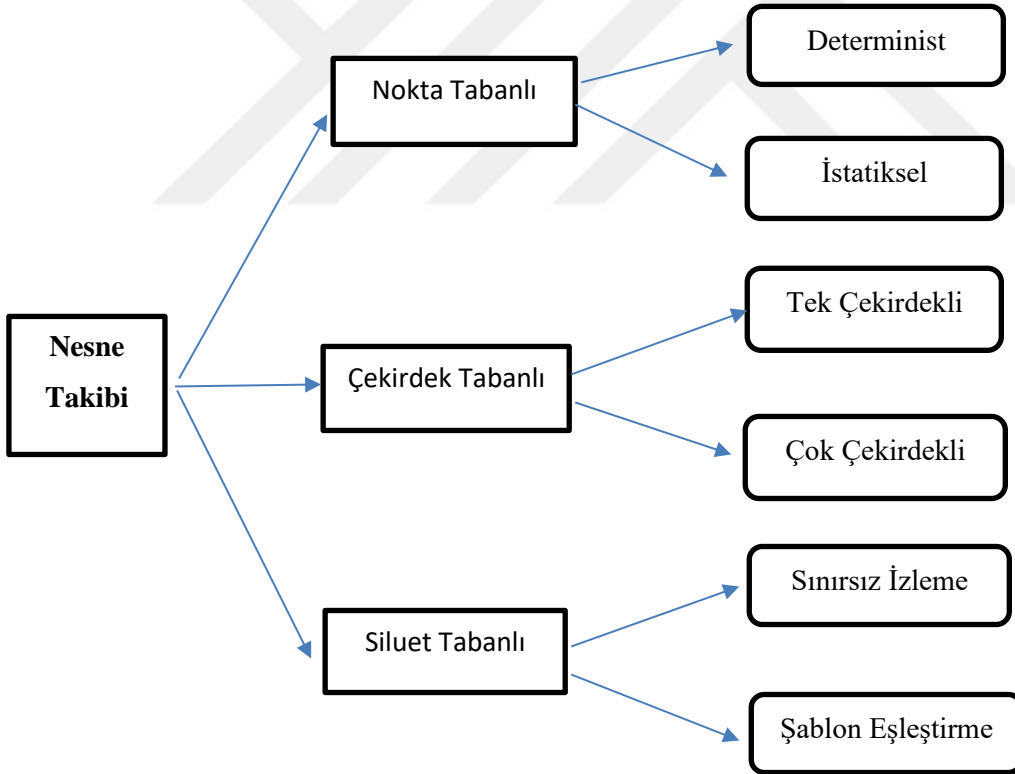
Şekil 3.6. Nesne tespitinin genel yapısındaki başlıklar

Görüntülerdeki veya videolardaki nesnelerin tespitinde temel olarak iki bilgi kullanılmaktadır, bu bilgiler nesnenin renk, şekil, doku gibi öznitelikleri ve hareket bilgisidir. Bu bilgileri görüntülerden elde etmek için belli bir işlem hacmi ve zamanı gerekmektedir. Görüntüden elde edilen bu bilgileri nesne tespiti konusunda ayrı ayrı kullanan yöntemler mevcut olması ile birlikte nesne tespitinin ve takibinin zor olduğu yapılarda farklı öznitelik bilgilerini ve zaman bilgisini birlikte kullanan yöntemlerde bulunmaktadır. Nesne tespiti için farklı bilgilerin birlikte kullanılması nesnelerin doğru şekilde tespit edilmesi konusunda başarıyı artırıyor olsa da işlem zamanını da artırmaktadır. Bu durumda GZNT için önemli bir problemdir. Bu problemin minimize edilmesinde görüntüyü sinir ağından bir seferde geçirerek nesne tespiti yapmak için tasarlanan SBKB algoritmaları çok başarılı bir çözüm ortaya koymaktadır.

GZNT yaklaşımlarının ikinci aşamasında tespit edilen nesnenin sınıflandırılması gelmektedir. Nesnelerin sınıflandırılmasını veri kümesi içerisindeki verilerin belirli özelliklerine göre gruplandırması olarak tanımlayabiliriz. Belirlenen nesnelerin öznitelikleri sayısal değerlerden meydana gelmektedir. Bu sayısal değerler bir piksel değeri veya ortalama piksel değerleri gibi anlamlı şeylerden oluşmaktadır. Nesnelerin sınıflandırılması nesne takip sistemleri için son derece önemlidir. Özellikle çoklu nesne takibi yapılacak çalışmalarda nesnelerin birbirine karıştırılmaması için en doğru şekilde sınıflandırılması oldukça önemlidir. Sınıflandırma algoritmaları hareket tabanlı, doku tabanlı, şekil ve renk tabanlı olarak dört ana başlık altında toplanmıştır (Balaji, Karthikeyan, 2017). Hareket tabanlı yöntemlerde nesnelerin hareket bilgilerini kullanarak sınıflandırma yapılmaktadır. Hareket özneliğini kullanarak video çerçeveleri arasında bir sonraki video çerçevesinde



nesnenin varmış olacağı nokta tahmin edilecektir. Bu tahmin yaklaşımı sayesinde nesnenin tespiti için taranacak alanda daralmış olacaktır. Özellikle belirli bir şekle sahip olmayan nesnelere takip şekil, kenar ve renge bağımlı olmadığından hareket tabanlı sınıflandırmalarla başarılı sonuçlar elde edilmektedir. Doku tabanlı sınıflandırma yönteminde nesnelere öznelikleri çıkartılır ve buna dayalı olarak sınıflandırma yapılmaktadır. Öncelikle özneliklerinin çıkartılması ve sonra sınıflandırmanın yapılması işlem yükünü artırmaktadır. Ama doku tabanlı sınıflandırma yaklaşımları ile çok başarılı sonuçlar elde edilmektedir. Farklı ışık ortamlarında nesnelere renklerinin belirginliğini yitirmesi doku tabanlı sınıflandırma yöntemleri için önemli bir dezavantaj oluşturmaktadır. Görüntülerde tespit edilmiş ve sınıflandırılmış nesnelere takip edilmesi üç kategoride incelenmektedir. Bu kategoriler nokta tabanlı, silüet tabanlı ve çekirdek tabanlı olarak tanımlanabilir. (Bkz. Şekil 3.7.) de nesne takip yöntemlerinin sınıflandırma yapıları detaylı olarak görülmektedir.



Şekil 3.7. Nesne takip yöntemlerinin sınıflandırma yapıları

Nokta tabanlı takip sisteminde her bir nesne tek bir nokta ile temsil edilmektedir. Mevcut görseldeki takip edilecek nesne bir noktaya aktarıldıktan sonra bir önceki görseldeki tespit edilen noktalar arasında bağlantıların doğru şekilde ilişkilendirilmesi gerekmektedir. Görseldeki nesnenin yakalanıp nokta ile temsil edilmesinden sonra tespiti yapılmış noktalar

arasında ki benzerlik değeri hesaplanır. Benzerliklerin hesaplanmasında nesnelerin kaybolması veya yanlış nesnenin yakalanmış olması gibi karmaşık durumların karmaşıklığını minimize etmek için deterministik yöntemler veya istatistiksel yöntemler kullanılmaktadır. Deterministik yöntemlerde yakınlık, maksimum hız, küçük hız değişimleri, genel hareket gibi yaklaşımlarla nesnelerin arasında ki bağ maliyet tanımlaması yapılır. Video görüntüleri her zaman kısmi gürültüye sahip olacağından ve nesnenin hareketleri istenmeyen etkenlere maruz kalabileceğinden istatistiksel yöntemler, nesnenin hız, ivme ve pozisyonu durum tahmini sırasında modellerin belirsizliğini ve ölçüm değerlerini benimseyerek nesne takibi problemlerine çözüm bulmaya çalışmaktadır. Problemin tanımı ve çözümü için Kalman filtresi ve parçacık filtreleme yöntemleri kullanılmaktadır. Görüntüdeki hareketli nesnenin istatistiksel durumunu tahmin edecek problemin denklem (3.1.) de ki gibidir (Talu, 2010). Takip edilmek istenen nesnenin pozisyon bilgisi  $X^t$ :  $t = 1, 2, \dots$  şeklinde tanımlandığında zamana göre pozisyon değişimi Denklem 3.1. de gösterilmektedir.

$$X^t = f(X^{t-1}) + W^t \quad (3.1)$$

$W^t$ :  $t = 1, 2, \dots$  beyaz gürültülerdir. Ölçüm verisi ile pozisyon değişimi arasındaki bağlantıda denklem (3.2) de gösterilmektedir.

$$Z^t = h(X^t) + N^t \quad (3.2)$$

$N^t$  beyaz gürültülerdir ve  $W^t$  den bağımsızdır. İstatistiksel yöntemlere dayalı durum tespiti yapmak,  $t$  anına kadar olan tüm değerlerin göz önüne alınması ile  $X^t$  pozisyon değişimini tahmin etmeye çalışmaktır (Talu, 2010). Bu şekilde sonsal olasılık yoğunluk fonksiyonunu yani  $p(X^t | Z^1, \dots, Z^t)$  elde edilecektir.

Siluet tabanlı nesne takip yöntemleri takip edilmek istenen nesnenin belirgin özelliklerinin olmaması durumunda, kenar ve şekil bilgilerini çıkartarak buna göre bir sonraki görselde bu nesneyi aramaya dayanan bir takip yöntemidir. Şekilsel değişiklik durumlarına çok hassas bir takip yöntemidir.

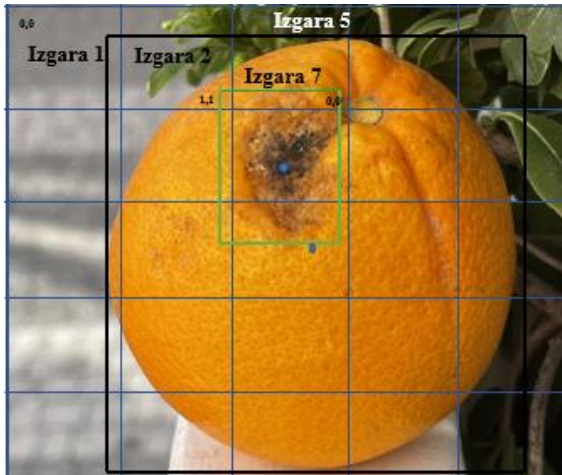
Çekirdek tabanlı nesne takip yönteminde nesnelere bir çerçeve içine alınacaktır. Çerçeve içine alınan nesnenin en belirgin özellikleri hesaplanır ve başlangıçta bulunan şekil yardımıyla nesne takip edilmeye çalışılır. Bu takip yönteminde nesnenin özelliklerinin hesaplanmasından ziyade çizilen geometrik çerçevenin içinde bulunan yapının özelliklerinin

çıkartılması önemlidir. Siluet tabanlı takip yöntemlerine göre iş zaman hızı ve başarı oranı olarak daha başarılı sonuçlar elde edilebilmektedir.

### 3.5.1. YOLO algoritması

Son zamanlarda sıklıkla gerçek zamanlı nesne tespiti yapmak için SBKB algoritması kullanılmaktadır. SBKB algoritmalarının çok popüler hale gelmesinin temelinde GZNT yapma konusunda kendinden önceki algoritmalara göre çok daha hızlı ve doğru şekilde nesnelere tespit etmesi yatmaktadır. SBKB algoritmalarından öncede GZNT 'i gerçekleştiren algoritmalar mevcuttu ama bunların en büyük sorunu yavaş olmalarıydı. SBKB algoritmalarından önce GZNT algoritmalarının ortalama kesinlik değerleri de yeterince iyi değildi. SBKB algoritmasının hem nesne tespiti doğruluk değerinde yeterli olmasını sağlayan hem de çok daha hızlı şekilde çalışmasını sağlayan özelliği resmi tek seferde ESA yapısından geçirerek resim üzerindeki nesnelere koordinatlarını tahmin edebiliyor olmasıdır (Telçeken, Kutlu, 2022). Daha önce kullanılan nesne tespit algoritmalarında resim üzerinde öncelikle nesne bulunması muhtemelen noktalar belirleniyor daha sonra bu noktaların her birisi ayrı ayrı ESA sınıflandırıcılarında yürütülüyordu. Bu durum ne kadar doğru sonuçlar veriyor olsa da hız konusunda sorun yaşanıyordu. Resim ayrı iki işleme tabi tutulduğu için düşük SBK alınıyordu.

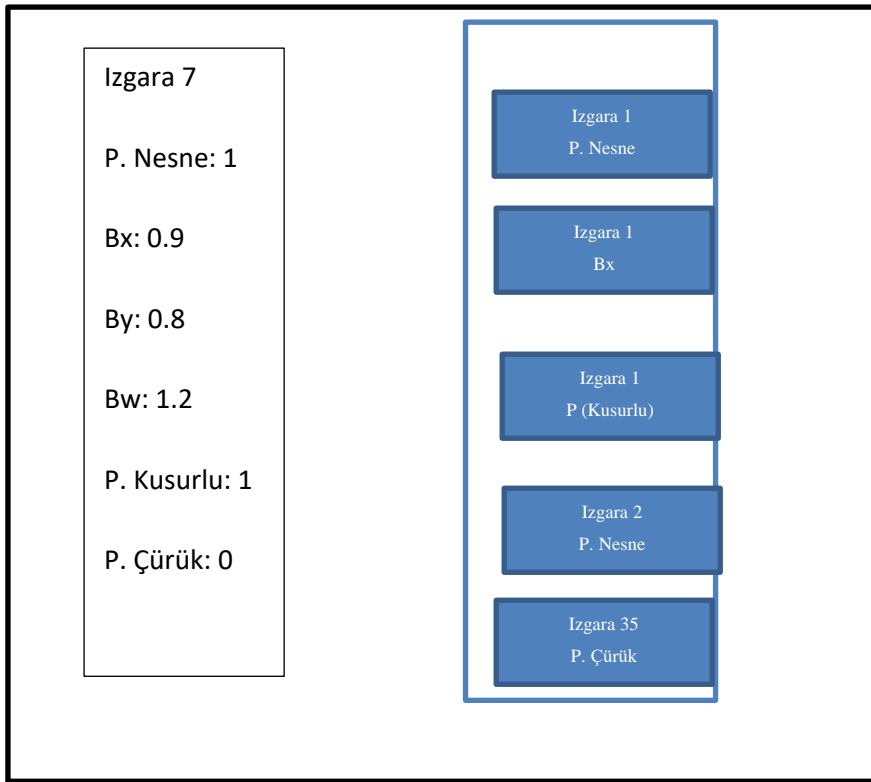
SBKB algoritması nesne tespitini tek bir regresyon problemi olarak ele alıyor. Bu durumu gerçekleştirmek için girdi olarak verilen resmi öncelikle SxS 'ilik ızgaralara ayırıyor (Redmon, Angelova, 2015). Resim 3.8'de SBKB algoritmasının resim üzerine uygulanırken nesnelere tespit etmek için oluşturduğu ızgara yapısı gösterilmektedir.



Resim 3.8. Girdi resmi üzerinde ızgaraların uygulanması ile ilgili görsel

Izgaralara ayrılmış resimde, her ızgara kendi içerisinde nesne olup olmadığı, nesne var ise orta noktasının ızgara içinde olup olmadığı kontrolünü, nesnenin orta noktası ızgara içindeyse eğer uzunluğunu, yüksekliğini ve sınıfını bulmaktan sorumludur. Örnek olarak Resim 3.7’de portakal ızgara 5’e denk geldiği için portakalın tespit edilmesinden ve etrafına kutucuk çizilmesinden o ızgara sorumlu olacaktır. Aynı durum ızgara 7 içinde geçerlidir. Kusurlu bölge ızgara 7’e denk geldiğinden kusurlu noktanın tespit edilmesi etrafına kutu çizilmesinden ızgara 7 sorumlu olacaktır. Girdi olarak Resim 3.7’deki resim SBKB algoritmasına verildiğinde çıktı örneği aşağıdaki gibi olacaktır.

Çıktı



SBKB algoritmasında her ızgara için bir tahmin vektörü oluşturulmaktadır. Tahmin vektörünün her birisi içinde;

**Güven skoru:** Izgara içerisinde nesne olup olmadığı konusunda ne kadar eminlik olduğunu gösteren değerdir. Güven skoru 0 ise ızgara içerisinde nesne yoktur. Güven skoru 1 ise ızgara içinde kesinlikle nesne bulunmaktadır. Güven skorunun hesaplaması denklem 3.3’te gösterilmektedir.

**Bx:** Izgara içindeki nesnenin orta noktasının x koordinatı

**By:** Izgara içindeki nesnenin orta noktasının y koordinatı

**Bw:** Izgara içindeki nesnenin genişliği

**Bh:** Izgara içindeki nesnenin yüksekliği

**Bağlı Sınıf Olasılığı:** Modelde kaç sınıf var ise o kadar tahmin değeri. Izgara 7 ye bakıldığında Kusurlu nokta kesin olarak var olduğundan Kusurlu:1, Çürük:0 olacaktır.

$$\text{Güven Skoru} = \text{Kutu Güven Skoru} \times \text{Bağlı Sınıf olasılığı} \quad (3.3)$$

Kutu güven skoru nesnelere tespit etmek için çizilen kutucukların skorudur. Denklem 3.4'te kutu güven skorunun hesaplaması görülmektedir.

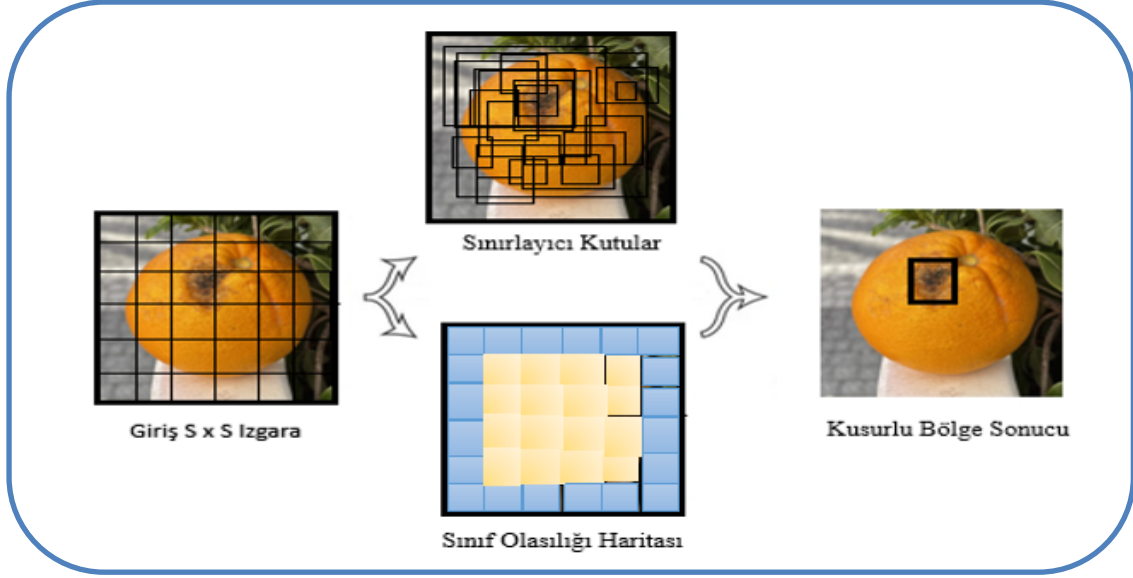
$$\text{Kuru Güven Skoru} = P(\text{Nesne}) \cdot \text{Iou} \quad (3.4)$$

**Iou:** Zemin gerçeği ile tahmin edilen kutu arasındaki benzerlik oranıdır.

İçinde hiçbir nesne olmayan ızgaralar için bağlı sınıf olasılığı 0 olacağından güven skoru da 0 olacaktır. SBKB algoritmalarında her bir ızgara bir tane nesne algılayabilir. Örnek olarak 5x5 'lik bir ızgara kullanıldığında 25 tane tahmin yapılır. Bir ızgara içerisinde birden fazla nesne olduğunda SBKB algoritmalarına YOLOv2 algoritması ile beraber algoritmaya eklenen Anchor Box (Çapa Kutuları) ile çözülmüştür (Zhang, Huang, Jin, Li, 2017). Çapa kutuları metodunun mantığı belirli kalıpların yardımı ile nesnelere etrafına çizilecek kutuların tahmin edilmesidir. Çapa kutularının eklenmesi ile beraber çıktı vektörü denklem 3.5'te olduğu gibi olacaktır.

$$S \times S \times (3 \times (5 + (\text{Sınıf Sayısı}))) \quad (3.5)$$

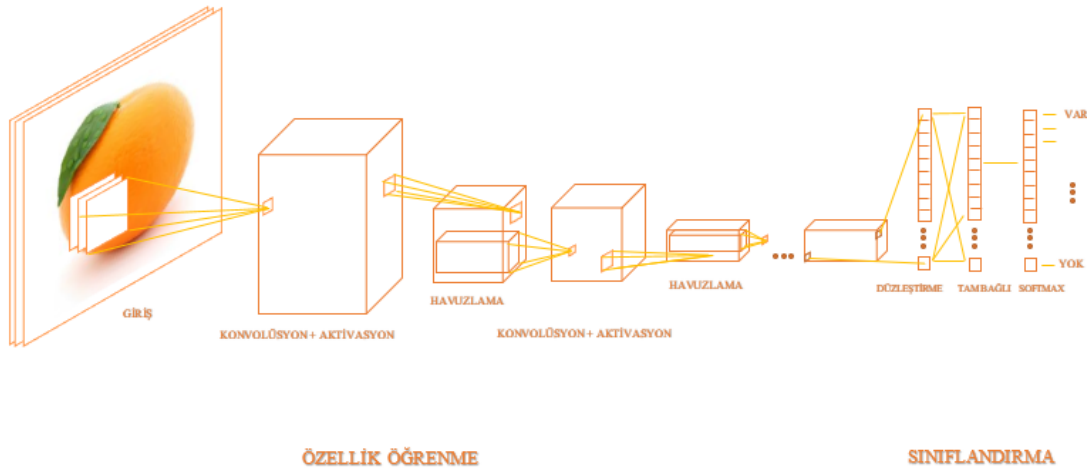
Her bir ızgarada Çapa kutusu sayısı kadar Güven, x, y, w, h ve diğer sınıfların olasılığını hesaplanır. SBKB algoritması çalıştığında gerekli gereksiz birçok kutu çıkabilmektedir. Gereksiz olan kutuların kaldırılması için kutu içerisinde nesne olup olmadığını tahmin eden bir değer var ama ızgara içerisinde nesne var olduğunda, aynı nesne için birden fazla kutu çizilmişse o zaman maksimum olmayan bastırma algoritması ile bu sorun çözülmüştür (Zhang, Huang, Jin, Li, 2017). MOB algoritmasında güven skoru belirli bir seviyeden düşük olan kutular atılır. Henüz gereksiz kutular bulunuyorsa güven skoru en yüksek olan kutu seçilecek ve o çıktı olarak belirlenir. Güven skoru en yüksek olan kutu X kutusu olursa X ile Iou değeri 0.5 ten büyük olan tüm kutular atılır. SBKB algoritmalarının sınıflandırma temelinde ESA yapıları bulunmaktadır. YOLO nesne algılama algoritmasının genel ağ yapısı Şekil 3.8'da görülmektedir.



Şekil 3.8. YOLO algoritması genel yapısı

### 3.5.2. Evrişimli sinir ağı

Evrişimli sinir ağı, girdi verilerinden katmanlarında bulunan filtreler yardımıyla farklı ağırlık değerlerini çıkartıp sinir ağında sınıflandırmaktadır. ESA algoritması evrişim, havuzlama, düzleştirme ve tam bağılı katmanlarından oluşmaktadır. Katmanların sayısı ve katmanlarda kullanılacak filtre sayısını ve yapısını belirlemek sınıflandırılacak veriye ve işlem hacmine göre değişiklik göstermektedir. Resim 3.9’de ESA modelinin katmaları detaylı bir şekilde gösterilmiştir. ESA modelleri görüldüğü gibi iki yapıdan oluşmaktadır. Birinci yapı evrişim ve havuzlama katmanlarından oluşan verilerin özniteliklerinin çıkartılacağı katmanlardır. İkinci bölümde birinci bölümde çıkartılan özniteliklere göre sınıflandırma yapılmaktadır.



Resim 3.9. Evrişimli sinir ağının yapısı

**Evrişim katmanı:** Özellikleri çıkartmak için kullanılan katmandır. Evrişim katmanı uygulandığında uygulanan filtre türüne göre ilk evrelerde düşük seviyeli özellikler, sonraki aşamalarda orta ve yüksek seviyeli özellikler çıkarılmaktadır (Altan, G., 2019). Bu sayede ilk başta düşük seviyeli özellikleri öğrenmeye başlayan model sonraki katmanlara düşük seviyeli özelliklerin baskın olan piksellerini aktarır. Bu süreçlere özellik öğrenme aşamaları da denilmektedir.

**Doğrusal Olmayan Katman:** doğrusal olmayan yapıları sisteme tanıtılması için kullanılan katmandır.

**Havuzlama Katmanı:** ağırlık sayılarını azaltmak ve uygunluğu kontrol etmek için kullanılan katmandır. Evrişim işleminden elde edilen her görüntü matrisini girişte görüntüye uygulanmış farklı filtrelerle elde edilmiş tensöre belirlenen havuzlama büyüklük ve adım aralığında tensör üzerinde hareket ettirerek tüm derinlikler havuzlama için belirlenen filtre yapısına göre hesaplanır (Altan, G., 2019).

**Düzleştirme Katmanı:** Verilerden matris yapısında gelen yapıları tek boyutlu bir yapıya dönüştürerek verilerin sınıflandırmaya hazırlanması için kullanılan katmandır. Bu katman sayesinde giriş görüntüsünde belirlenmiş özellikler yapay sinir ağına verilebilecek şekilde dönüştürülmektedir. Her ESA modelinde bir düzleştirme katmanı görülmektedir.

**Tamamen Bağlı Katman:** Verileri sınıflandırmak için kullanılan sinir ağı katmanıdır.

ESA temelinde sınıflandırma yapmak için sinir ağı kullanılmaktadır, fakat verilerin özellik bilgilerini çıkartmak için diğer katmaları kullanılmaktadır. ESA algoritmalarında verilerin özelliklerini çıkartmaya ihtiyaç duyulmadan sinir ağına vererek sınıflandırma yapmak mümkün hale gelmektedir. Derin öğrenme modellerinden görüntü işleme süreçlerini yapısında bulundurmasından dolayı en geniş kullanılan model ESA'lardır.

### **3.6. Performans Ölçümü**

Çoklu sınıflandırma problemlerinin sonuçları değerlendirilirken eğitilen modeller için hata matrisi kullanılır.

Model tarafından tahmin edilen ve doğru değerlerle 4 farklı şekilde karmaşıklık matrisi oluşturulur. Karmaşıklık matrisi yapısı Çizelge 3.2'de gösterilmektedir. TP, FP, FN, TN olarak isimlendirilen değerler ile oluşturulan karmaşıklık matrisinde;

TP: True Pozitif, model tarafından tahmini yapılan sınıfa ait doğru tahminleri

FP: False Pozitif, model tarafından tahmini yapılan sınıfa ait yanlış tahminler

FN: False Negatif, model tarafından yanlış yapılan negatif tahminler

TN: True Negatif, model tarafından doğru yapılan negatif tahminler

Çizelge 3.2. Karmaşıklık Matrisi

Karmaşıklık Matrisi		Gerçek Değerler		
		Sağlam	Çürük	Kusurlu
Tahmin Edilen Sonuçlar	Sağlam	True Pozitif (TP)	Hata	Hata
	Çürük	Hata	True Pozitif (TP)	Hata
	Kusurlu	Hata	False Negatif (FP)	Hata

IoU değeri belirlenen sınırlayıcı kutu ile modelin tahmin ettiği sınırlama kutusu kesişimleri bileşmesi oranı olarak tanımlanır. Iou değeri standart olarak 0,5 olarak belirlenmektedir. Belirlenen bu değer üzerinde kalan kutu tahminleri TP olarak, altında kalan kutu tahminleri FP olarak alınmaktadır. Görüntü üzerinde nesne için belirlenen kutu olmasına rağmen model tarafından nesne için sınırlayıcı kutu belirlenmemesi FN olarak tanımlanır. Görüntü üzerinde belirlenen nesnelere dışında kalan diğer tüm arka plan görüntüleri TN oluşturur. Performans hesaplamada TN kullanılmamaktadır.

Karmaşıklık matrisi kullanılarak mAP (ortalama kesinlik değeri) ile beraber 4 farklı performans ölçümü gerçekleştirilir. Hassasiyet her sınıf için yapılacak doğru tahminlerin tüm tahminlere oranı olarak hesaplanır. Özgünlük nesne bulunduğu anda hangi oranda doğru bulunduğunun oranıdır. Genel doğruluk ise yapılan doğru nesne tahminlerinin belirlenen nesnelere oranı olarak hesaplanır. Performans ölçümü ile ilgili denklemler aşağıda gösterilmektedir.

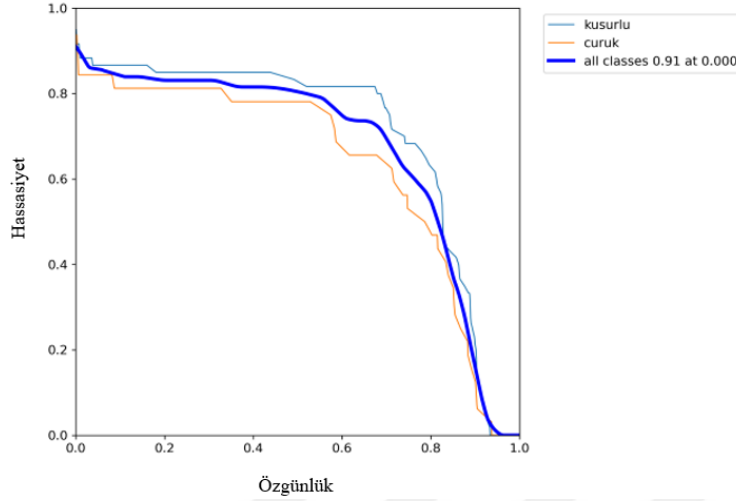
$$Hassasiyet = \frac{TP}{TP+FP} \quad (3.6)$$



$$\text{Özgünlük} = \frac{TP}{TP+FN} \quad (3.7)$$

$$\text{Genel Doğruluk} = \frac{TP+TN}{TP+FP+TN+FN} \quad (3.8)$$

Özgünlük ve hassasiyet değerleri ile özgünlük, hassasiyet eğrisi Şekil 3.9'da gösterilmektedir.



Şekil 3.9. Özgünlük ve hassasiyet değerleri ile özgünlük, hassasiyet eğrisi

Ortalama hassasiyet-özgünlük (AP), eğriler altında kalan olarak tanımlanır. Eğrinin sırası ile sol üst ve sağ altta 1 değerine yaklaşması tahminin doğruluk oranının arttığını göstermektedir. AP hesaplama denklemi aşağıda gösterilmektedir.

$$AP = \int_0^1 p(r) dr \quad (3.9)$$

$p(r)$ : özgünlük-hassasiyet fonksiyonu.

Sınıflara ait AP'lerin toplamının tüm sınıf sayılarına bölünmesi ile ortalama kesinlik değeri mAP değeri elde edilmektedir. mAP değerinin hesaplanma denklemi aşağıda gösterilmektedir.

$$mAP = \frac{\text{Tüm sınıflara ait ortalama kesinlik değerleri toplamı}}{\text{Toplam sınıf sayısı}} \quad (3.10)$$

YOLO algoritmasında eğitimler gerçekleştirilirken iterasyon sayısı tüm sayısına göre belirlenmektedir. İterasyon sayısı sınıf sayısının 2000 ile çarpılması ile elde edilmektedir. Daha önce yapılan çalışmalar en iyi sonuçların elde edilmesi için iterasyon sayısı

belirlenmesi konusunda incelenerek belirlenmiştir. YOLO algoritmasında en iyi başarımlar elde etmek için minimum iterasyon sayısı 6000 olarak belirlenmektedir.

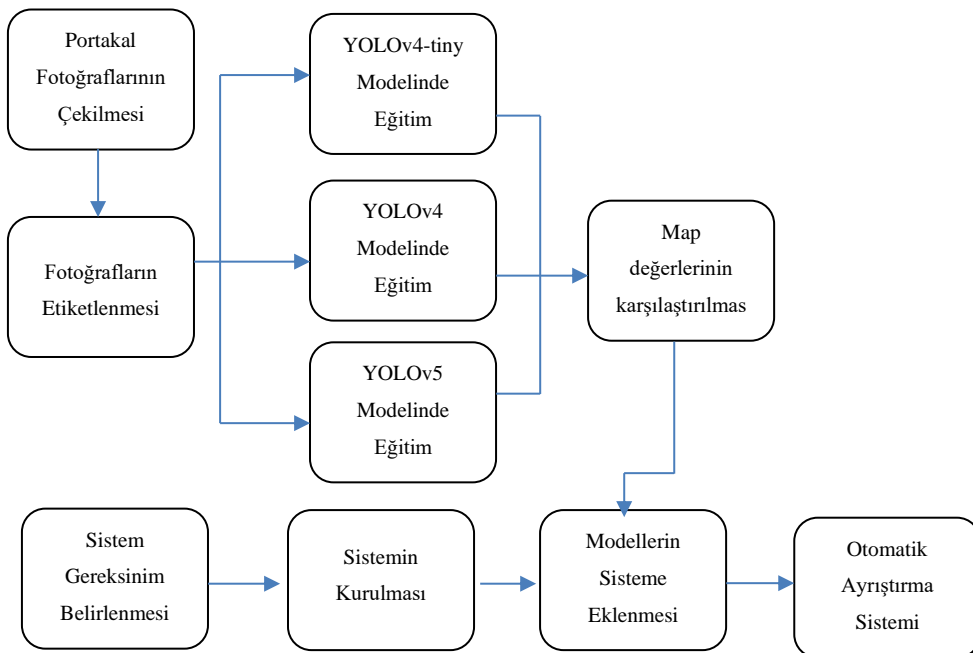


## 4. ARAŞTIRMA BULGULARI VE SONUÇLAR

### 4.1. Derin Öğrenme Tabanlı Meyve Ayırıştırma Sistemi Portakal Uygulaması

Bu bölümde derin öğrenmede kullanılacak yeni veri seti için yapılan işlemlerden, gerçek zamanlı nesne tespiti için algoritma belirlenmesinden ve sistemin kurulmasından bahsedilecektir. Yapılacak olan YOLO algoritması eğitimleri için açık kaynak kodlu tensorflow kütüphanesi kullanılmıştır. Eğitimler İntel 12. Nesil i9 işlemci, Nvidia RTX 3090 Ti 24 gigabayt ekran kartı ve 64 gigabayt DDR5 sistem belleğine sahip bilgisayarla gerçekleştirilmiştir.

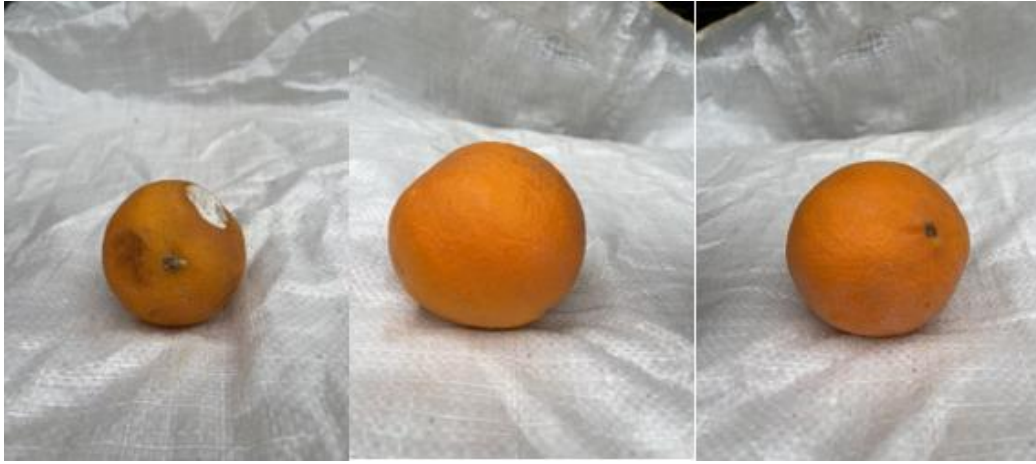
Bu tez çalışmasında derin öğrenme modellerinin eğitim ve test aşamalarında kullanmak için kusurlu, çürük ve sağlam olarak 3 sınıftan oluşan yeni bir veri seti oluşturulmuştur. Oluşturulan veri setinde portakal görüntülerini etiketlemek için Python programla dilinde yazılmış açık kaynak labelImg.exe programı kullanılmıştır. Sistemde GZNT algoritması YOLO algoritmasının farklı versiyonları olan YOLOv4-tiny, YOLOv4, YOLOv4-p6, YOLOv5, YOLOv6 ve YOLOv7 mimarileri ile ayrı ayrı eğitimler gerçekleştirilmiş, ortalama kesinlik değerleri ve oluşturulan prototip üzerinde saniyede işlenen portakal sayısı için performans değerlendirmeleri karşılaştırılmıştır. Sistemin kurulması için minimum gereksinimleri karşılayacak şekilde donanımlar belirlenmiş, sistem prototipi kurulmuştur. Bu tez çalışmasının akış diyagramı Şekil 4.1' de detaylı olarak gösterilmektedir.



Şekil 4.1. Çalışma akış diyagramı

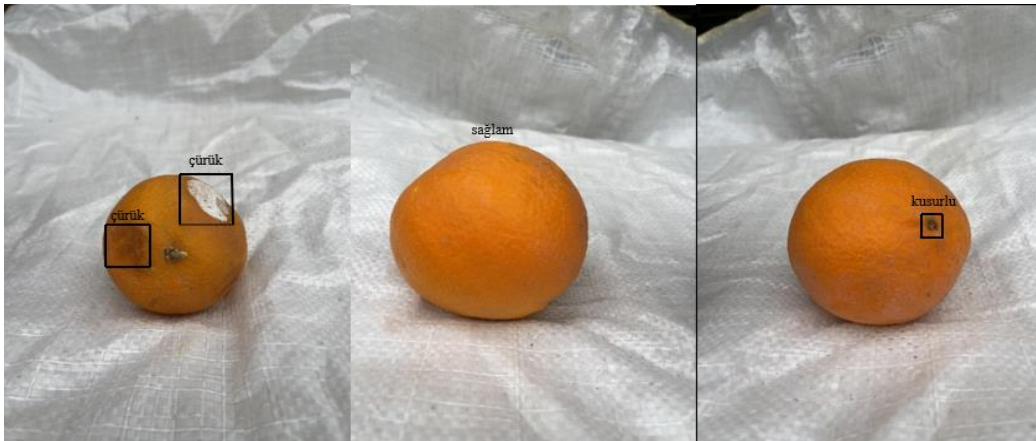
## 4.2. Veri Setinin Oluřturulması

Market, manav ve pazar yerlerinden portakal grntleri ekilmiřtir. ekilen grntler labelImg.exe programında YOLO algoritması iin kullanılabilir şekilde etiketlenmiřtir. Resim 4.1 de ekilen portakal grntlerinden rnekler gsterilmektedir. Derin ğrenme yaklařımlarında etiketlenmiř ham grseller zerinde eđitim yapmak mmkn olduđundan grsellere herhangi bir n iřlem uygulanmamıřtır. Etiketleme sırasında bazı portakal grntlerinin hem kusurlu hem de rk olduđu grlmř, buna gre bir grsel zerine birden fazla etiketleme yapılmıřtır.



Resim 4.1. Veri seti oluřturulurken ekilmiř portakal grntleri

Portakal grselleri etiketlemesinde her grntnn dođru řekilde etiketlenmesi iin grntler 10 kiři tarafından incelenerek rk, sađlam ve kusurlu olarak etiketlenmiřtir. Resim 4.2' de etiketlenmiř portakal grselleri gsterilmektedir.



Resim 4.2. Etiketlenmiř portakal grntleri

### 4.3. Nesne Tespiti

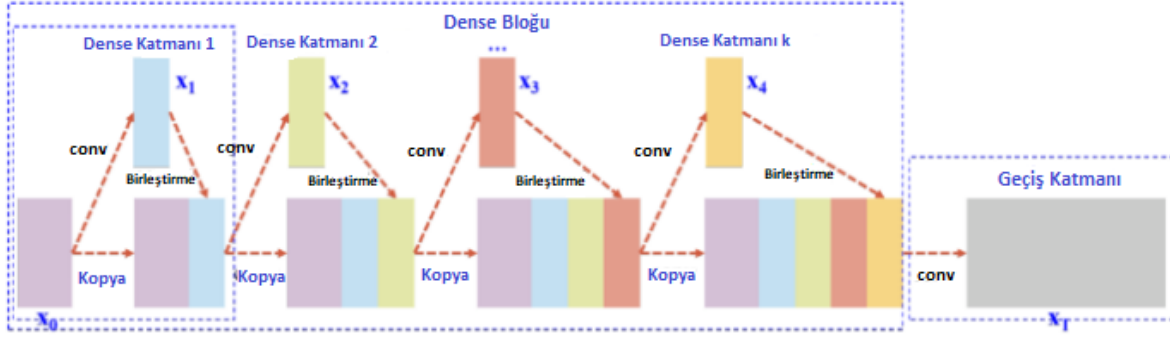
Bu tez çalışmasında portakal görüntüleri toplanıp etiketlendikten sonra, kurulacak konveyör bantlı sistem üzerinde portakalların kusurlu veya çürük olması tespit edileceği için gerçek zamanlı nesne tespit algoritması olan YOLO algoritmasının YOLOv4-tiny, YOLOv4 ve YOLOv5 versiyonlarında eğitimler gerçekleştirilmiştir.

### 4.4. Uygun Model Tespiti

Bu tez çalışmasında çürük ve kusurlu portakalların doğru sınıflandırılması ve sistemin işleme kapasitesinin maksimum seviyede olması için YOLO algoritmasının farklı versiyonları ile nesne tespiti ve sınıflandırması gerçekleştirilmiştir. Endüstriye yönelik bir çalışma yapıldığı için ürünlerin doğru şekilde ayrıştırılması kadar günlük işlenebilecek ürün miktarı da oldukça önemlidir. İncelenen 3 YOLO mimarisinin her biri farklı katman sayı ve yapısına sahip olduğu için her biri ile yapılan eğitim süreleri ve test süreleri farklılık göstermektedir. Bu tez çalışmasında amaçlanan en hızlı şekilde en doğru ayrıştırmayı yapabilecek sistemi kurmaktır.

#### 4.4.1. YOLOv4

YOLOv4, YOLOv3 modelinin geliştirilmiş bir evrimi olan nesne tespit algoritmasıdır. YOLOv4 modeli Alexey Bochkovskiy, Chien-Yao Wang ve Hong-Yuan Mark Liao tarafından geliştirilmiştir. Ortalama hassasiyet ve FPS (Saniyede Kare Sayısı) bakımından sırasıyla %10 ve %12 YOLOv3'ten daha iyidir (Bochkovskiy, Wang, Liao, 2020). YOLOv4 mimarisinde omurgayı CSPDarknet53, kafa yapısını YOLOv3 oluşturmaktadır (Bochkovskiy, Wang, Liao, 2020). YOLOv4 mimarisi düşük donanımsal ihtiyaçlarla, daha hızlı ve doğru eğitimler gerçekleştirmek için tasarlanmıştır. DenseNet yoğun blok yapısı Şekil 4.2 de gösterilmektedir. DenseNet' te,  $x$ ' inci yoğun katman çıktısı  $x$ ' inci yoğun katman girişi ile birleşmektedir birleşen yapı  $(x + 1)$ ' inci katmanın girdisi olmaktadır.



Şekil 4.2. DenseNet yoğun blok yapısı

Ağırlıkların güncellenmesi için geri yayılım algoritması kullanıldığında, ağırlık güncelleme denklemi ve DenseNet yoğun katman bağlantıları Şekil 4.3 de gösterilmektedir.

$X_1 = W_k * X_0$ $X_2 = W_2 * [X_0, X_1]$ $\vdots$ $X_k = W_k * [X_0, X_1, \dots, X_{k-1}]$	$w_{1'} = f(w_1, g_0)$ $w_{2'} = f(w_1, g_0, g_1)$ $w_{3'} = f(w_1, g_0, g_1, g_2)$ $w_{k'} = f(w_k, g_0, g_1, g_2, \dots, g_{k-1})$
--	--

Şekil 4.3. DenseNet yoğun katman bağlantı yapısı ve ağırlık güncelleme denklemi

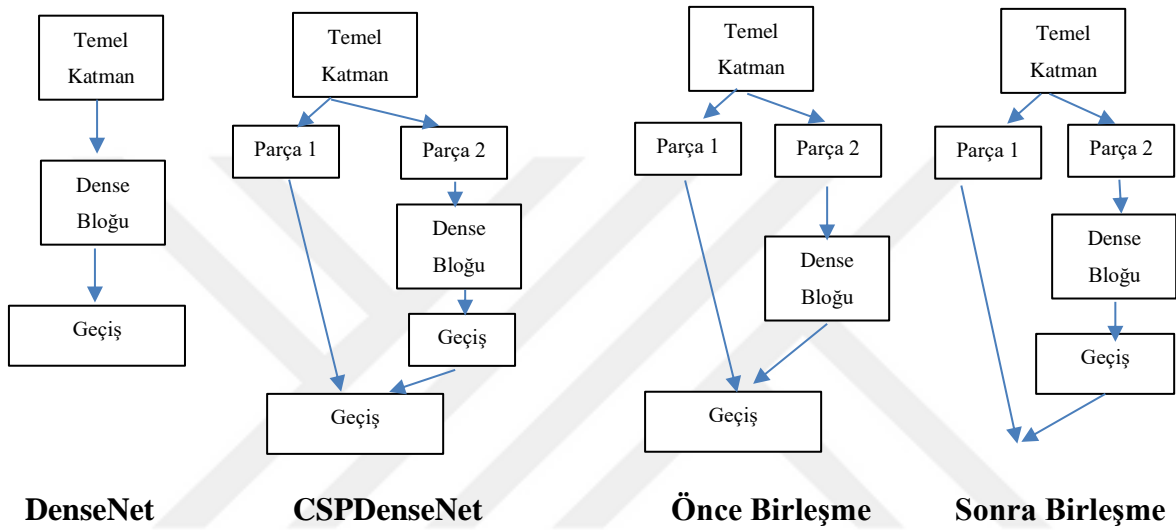
Farklı yoğun katman ağırlıklarının güncellenmesi için yüksek miktarda gradyan bilgisi yeniden kullanılmaktadır. Bu yoğun katmanlar tekrarlı kopyalanan indirgeme bilgilerini öğrenmesine sebep olacaktır.

Aşamalar arası kısmi DenseNet (CSPDenseNet) temel katman haritasını iki parçaya ayırmaktadır. Bir parçası yoğun bloklar ve geniş geçiş katmanından geçmektedir, diğer parça ise daha sonra bir sonraki aşamaya aktarılacak özellik haritası ile birleştirilmektedir. CSPDenseNet ileri besleme geçiş ve ağırlık güncellemeleri denklemleri Şekil 4.4'te gösterilmektedir.

$X_k = W_k * [X_0, X_1, \dots, X_{k-1}]$ $X_T = W_T * [X_0, X_1, \dots, X_k]$ $X_U = W_U * [X_0, X_T]$	$w_{k'} = f(w_k, g_0, g_1, \dots, g_{k-1})$ $w_{T'} = f(w_T, g_0, g_1, \dots, g_k)$ $w_{U'} = f(w_U, g_0, g_T)$
--	---

Şekil 4.4. CSPDenseNet ileri beslemeli geçiş ve ağırlık besleme denklemleri

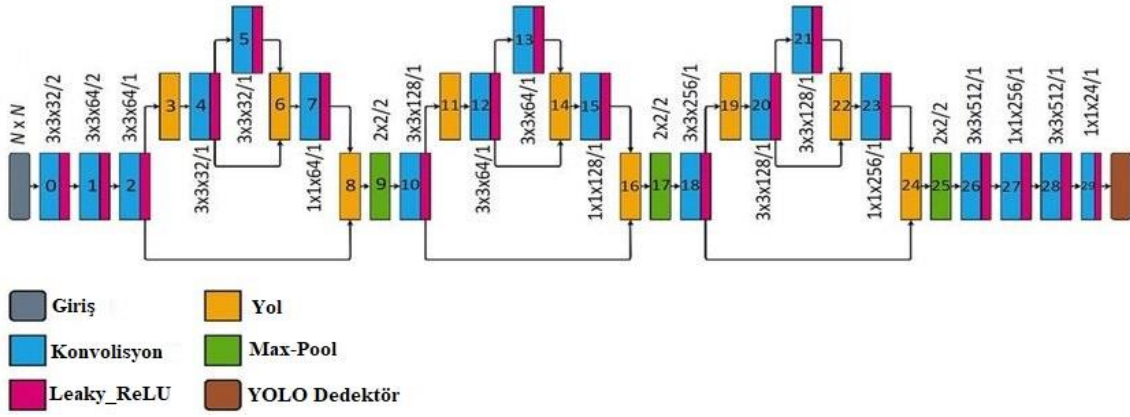
Yoğun katmanlardan gelen gradyanlar ayrı ayrı entegre edilmektedir. Yoğun katmanlardan geçmeyecek özellik haritaları da ayrı ayrı entegre edilmektedir. Ağırlık güncellemede iki tarafta da diğer tarafın tekrar eden gradyan bilgileri bulunmamaktadır. CSPDenseNet, DenseNet özelliklerinin avantajlarını korurken gradyan akışını keserek aşırı miktarda tekrar eden gradyan bilgisini önlemektedir. Kısmi yoğun blok türlerinin aynı anda tasarlanması gradyan kombinasyonunun farkını maksimize etmektedir. Farklı türde özellik birleştirme stratejilerine ait yapılar Şekil 4.5 de gösterilmektedir.



Şekil 4.5. Farklı türde özellik birleştirme stratejileri

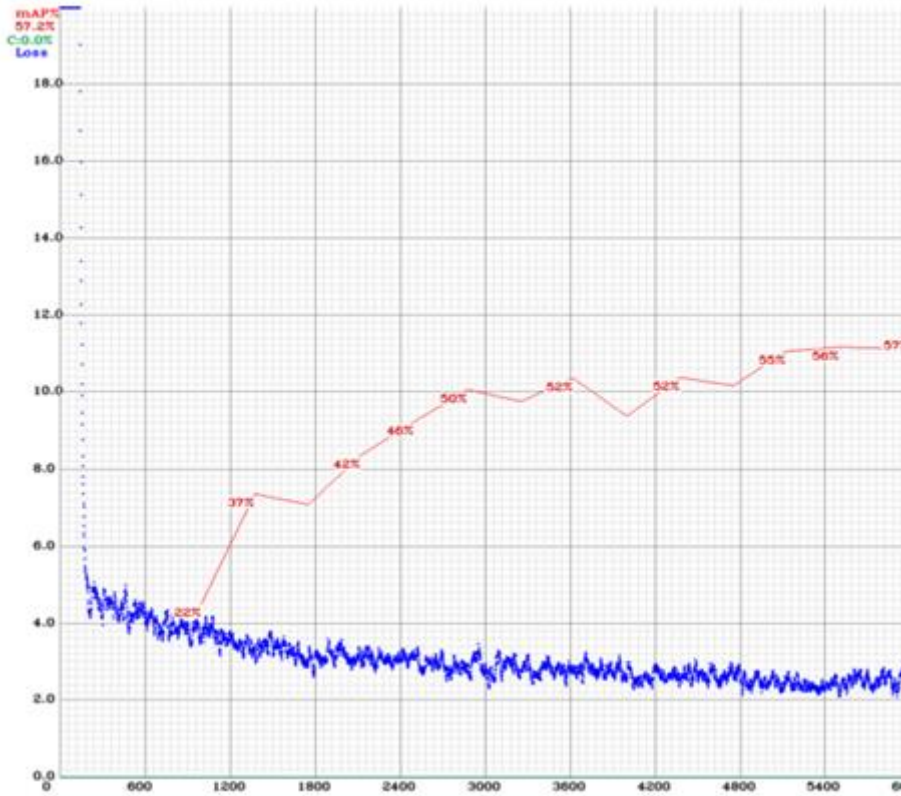
İki birleşme türü tasarlanmıştır. Birincisi önce birleşme, iki parçada oluşturulan özellik haritalarını birleştirip sonra geçiş yapılmaktadır. İkincisinde ise önce geçiş yapılır sonra iki parçada oluşturulan özellik haritalarını birleştirilir. Önce geçiş yapılırsa yüksek miktarda gradyan bilgisi tekrar kullanılacaktır.

YOLOv4 mimarisinde mekânsal piramit havuzlama (SPP), yol toplama ağı (PAN) ve mozaik veri büyütme yöntemi tanıtılmıştır (Bochkovskiy, Wang, Liao, 2020). Aktivasyon fonksiyonu olarak Leaky (Sızan) Relu fonksiyonu kullanılır. Leaky Relu'da sızıntı değeri 0,01 olarak verilmektedir. Tanım aralığı eksi sonsuza doğru deva etmektedir. 0'a yakın değer ile Relu aktivasyon fonksiyonunda kaybolan gradyanlar tekrar kazandırılarak negatif bölgedeki değerlerde öğretiliyor. YOLOv4 mimarisinin genel yapısı Şekil 4.6'te detaylı olarak görülmektedir.



Şekil 4.6. YOLOv4 mimarisinin genel yapısı

Bu tez çalışmasında YOLOv4 mimarisinde 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.

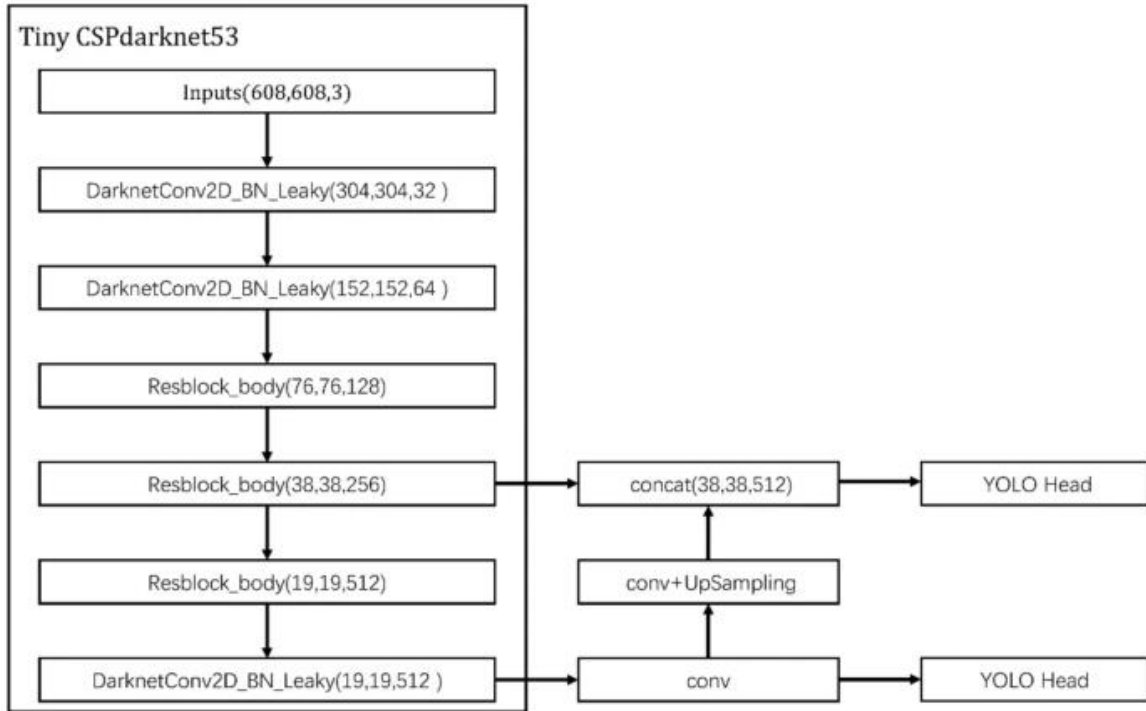


Şekil 4.7. YOLOv4 kayıp miktarı ve ortalama kesinlik değeri grafiği



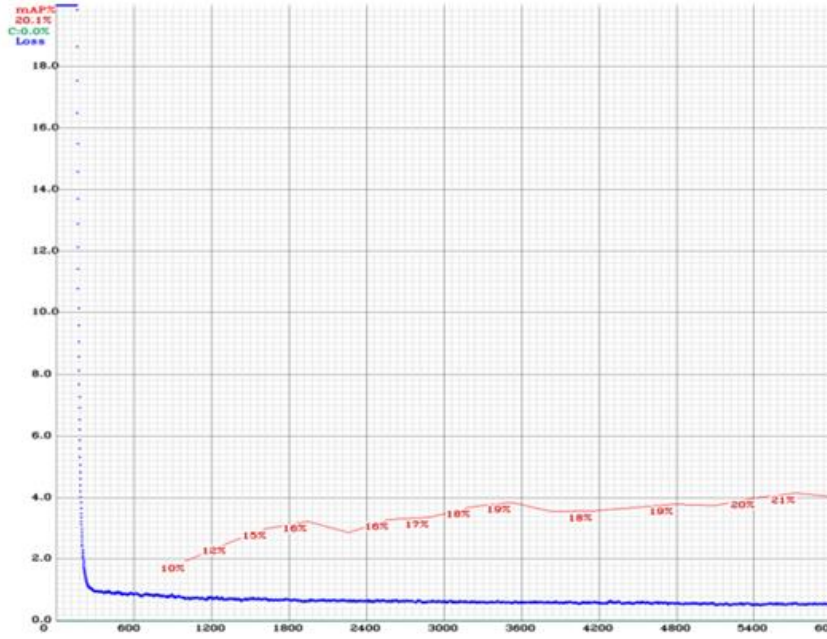
#### 4.4.2. YOLOv4-tiny

YOLOv4 mimarisinin bir versiyonu olan YOLOv4-tiny mimarisi daha düşük donanımlarla ve daha hızlı bir yapı için YOLOv4 mimarisinin sadeleştirilmiş hali olarak önerilmiştir. YOLOv4-tiny de optimizasyon elde edilmek için parametreler tekrarlamalı regrasyon yöntemi ile kayıp işlevi en aza indirgenmektedir. Güven kaybı, birleşim üzerinde tam kesişim kaybı ve sınıflandırma kaybı olarak kayıp fonksiyonu 3 bölüme ayrılmaktadır. Hedeflenen noktanın eklenip eklenmediğini kontrol etmek için güven kaybı eklenmektedir. Nesnelerin sınıf bilgilerine uymaları için sınıflandırma kaybı uygulanmaktadır. Merkez noktanın uzaklığı, kesişme alanı en boy oranısından oluşur ve konumlandırma bilgisi için kullanılmaktadır. Şekil 4.8’de YOLOv4-tiny mimarisinin yapısı görülmektedir.



Şekil 4.8. YOLOv4-tiny mimarisinin genel ağ yapısı

Yüksek seviyeli öznitelik matrisleri 19x19 ölçeğiyle, orta seviye öznitelik matrisleri 38x38’lik birleştirilmek için öznitelik piramidi ağına (FPN) ve sonra hedef tespit edilsin diye tahmin modülüne gönderilmektedir. FPN bilgi bütünlüğü sağlayabilmek için iki seviyeyi bütünlüştürmektedir, fakat bu durum bilgi fazlalığı oluşturabilir. Öznitelik çıkarılan katmanların çıkarım zamanında kısmi hata bilgisi, sonuçlarda daha büyük etkiye sahip olabilmektedir. Şekil 4.9’ da YOLOv4-tiny modeliyle gerçekleştirilen eğitimlerin loss ve mAP değerleri grafiği görülmektedir.



Şekil 4.9. YOLOv4-tiny kayıp miktarı ve ortalama kesinli değeri grafiği

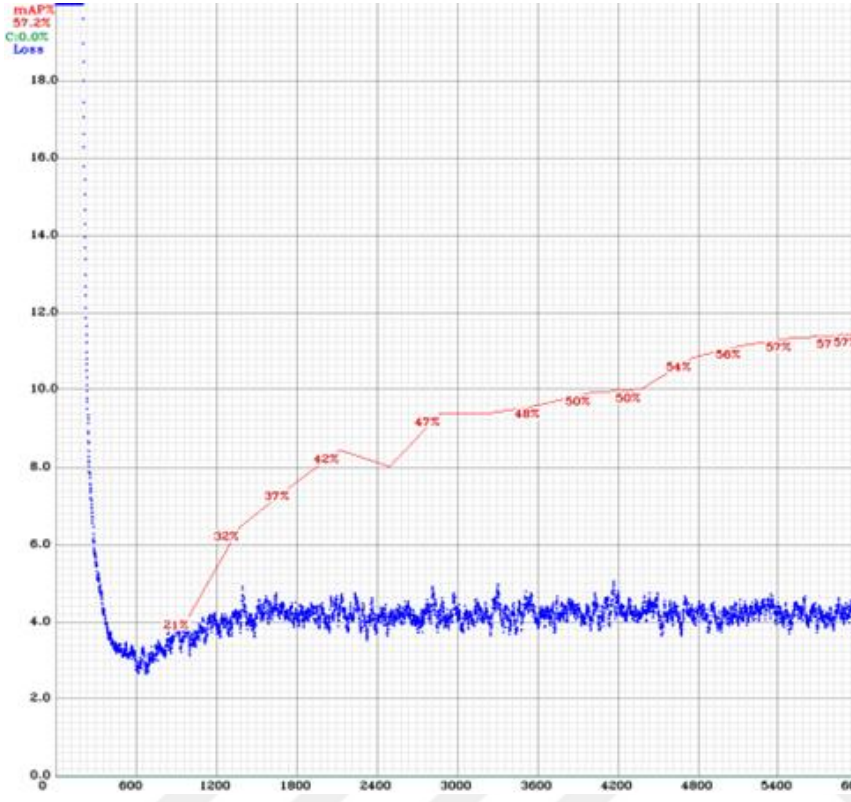
#### 4.4.3. YOLOv4-p6

Daha karmaşık ve uzak görüntülerde nesne tespitinin doğruluğunu artırmak için modifiye edilmiş YOLOv4 mimarisidir. Klasik YOLOv4 mimarisine göre daha fazla katmana sahiptir. Eğitim süresi görüntüleri daha fazla katmanda işlemesinden dolayı diğer YOLOv4 mimarilerine göre daha uzun sürmektedir. Görüntüleri eğitime 1280 x 1280 olarak aldığı için daha az piksel kaybı yaşanmakta, bu durumda ortalama kesinlik değerini artırmaktadır. Şekil 4.10'da YOLOv4-p6 mimarisinin yapısı detaylı olarak görülmektedir.



Şekil 4.10. YOLOv4-p6 mimarisi genel yapısı

Model ile yapılan eğitimlerde elde edilen kayıp miktarı ve ortalama kesinlik değerleri Şekil 4.11’de detaylı olarak görülmektedir.

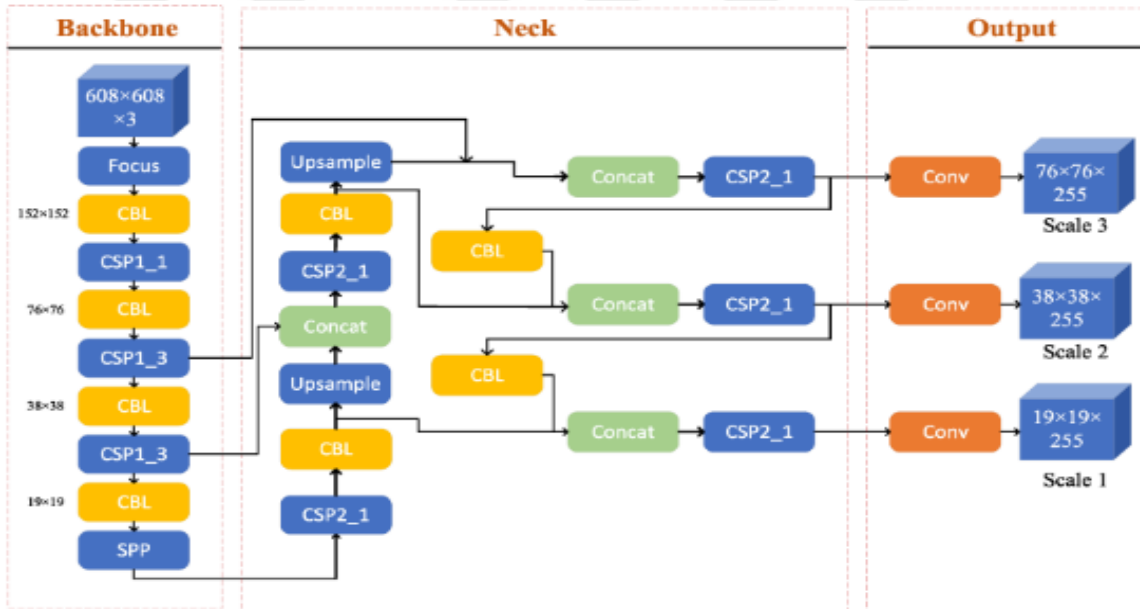


Şekil 4.11. YOLOv4-p6 kayıp miktarı ve ortalama kesinlik değerlerinin grafiği

#### 4.4.4. YOLOv5

YOLOv5 ile farklı parametre ve konfigürasyonlarla farklı modeller sunulmuştur. Her modelin kendi içerisinde artı ve eksi yönleri bulunmaktadır. Ama en önemli farklılıkları genel doğruluk ve performanslarındadır. YOLOv5, öznetelikleri çıkartmak için omurga bölümü, özneteliklerin birleştirilmesi için boyun bölümü ve nesnelere algılamak için çıkış bölümü olarak 3 bölümden meydana gelmektedir. Omurga ağı, çok sayıda evrişim ve havuzlama katmanıyla giriş görüntülerinden farklı boyutlarda özellik haritaları çıkaran bir ESA yapısıdır. Omurga ağını dört katman oluşturmaktadır. Boyun ağı, omurgada çıkartılan özellik haritalarından daha fazla bağlamsal bilgileri elde edebilmek ve bilgi kaybını en aza indirmek için farklı düzelerdeki özellik haritalarını birleştirmektedir. Birleştirme işlemlerinde FPN ve PAN yapılarının özellik piramitleri kullanılmaktadır. FPN yapısı ile güçlü semantik özellikler üst haritalardan alt haritalara aktarılmaktadır. PAN yapısı ile düşük özellik haritalarından daha yüksek özellik haritalarına güçlü lokalizasyon özellikleri

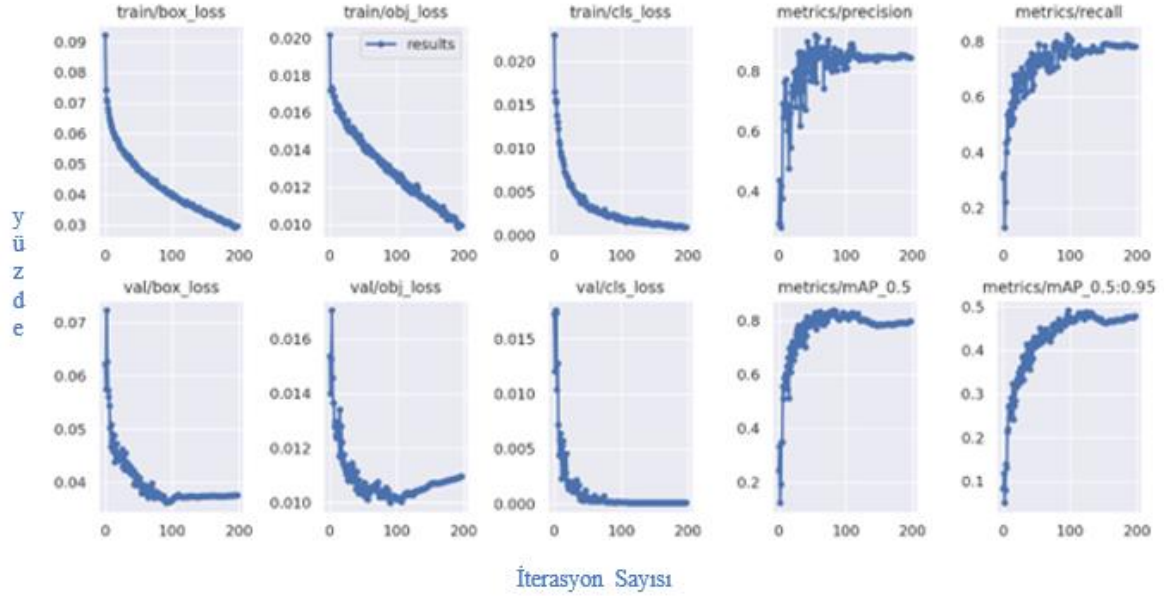
aktarılır. İki yapının boyun bölümünde kullanılmış olması özellik birleştirmeyi güçlendirmektedir (Zhu, Geng, Li, 2021). YOLOv5 modelinde özellikle  $19 \times 19 \times 255$ ,  $38 \times 38 \times 255$  ve  $76 \times 76 \times 255$  boyutlarında yeni özellik haritalarının 3 özellik haritaları birleştirme katmanı bulunmaktadır. Bu yeni özellik haritalarından çıkış bölümünde nesne algılama ve sınıflandırma gerçekleştirilir. YOLOv5 mimarisinde odaklanma modülü görüntüleri dilimlere ayırır ve birleştirir, bu durumun amacı alt örnekleme sırasında özellikleri daha iyi çıkartabilmektir. Mimarinin yapısında iki tür CSP bulunmaktadır. Bunlardan birisi omurgada diğeri boyun bölümünde kullanılır. Mimaride ön ve arka ağ katmanları birbirine çapraz katman bağlantısı ile bağlayan CSP ağı, model boyutunu küçültürken kesinliği koruyarak çıkarım hızının artırılması amaçlanmıştır. Mimarinin genel yapısı Şekil 4.12’de görülmektedir.



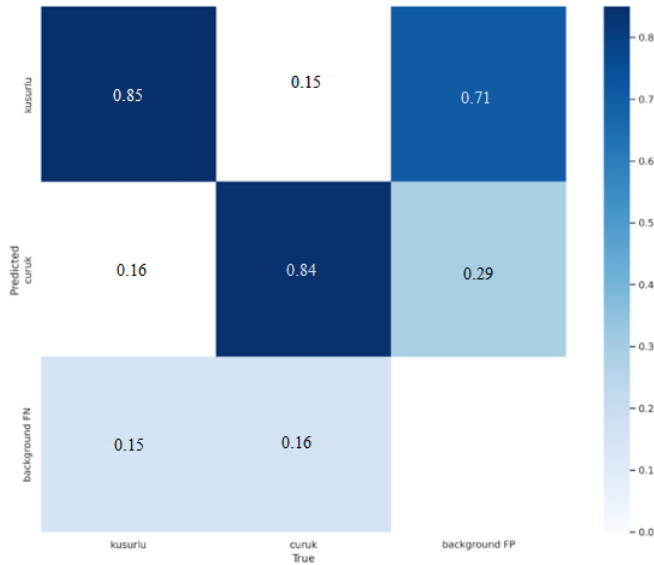
Şekil 4.12. YOLOv5 mimarisinin genel ağ yapısının görseli

YOLOv5 mimarisinin karmaşık ağ yapısı nedeni ile eğitim süresi bakımından YOLOv4 mimarisine göre daha uzun eğitim süresine sahiptir. En hızlı eğitim YOLOv4-tiny mimarisi ile gerçekleştirilmiştir. Katman sayısının sırasıyla arttığı YOLOv4, YOLOv4-P6, YOLOv5 mimarilerinde eğitim süreleri de katman sayısı artıkça artmıştır. Eğitim süresi artmış olsa da katman sayısının artması ortalama kesinlik değerini yükseltmiştir. YOLOv5 mimarisinde gerçekleştirilen eğitim grafikleri Şekil 4.13’de görülmektedir. Resim 4.3’te YOLOv5 mimarisi ile gerçekleştirilen eğitimler sonucunda test verileri üzerinde eğitilmiş modelin sınırlayıcı kutu çizimlerinden örnekler görülmektedir. YOLOv5 mimarisinde, diğer mimarilerde en çok sınırlayıcı kutu çiziminde hata oluşan çürük etiketi için çok daha verimli

sonuçlar elde edildiği gözlemlenmiştir. Sistem üzerinde saniyede yakalanan görüntü sayısı YOLOv5 mimarisinde YOLOv4 mimarisi ve türevlerine göre düşük olsa da sistemin çalışma hızı ve doğruluğu bakımından daha verimli olduğu görülmüştür. YOLOv5 için karmaşıklık grafiği Şekil 4.14’de görülmektedir.



Şekil 4.13. YOLOv5 mimarisi eğitimi ortalama kesinlik, kayıp oranı, hassasiyet, özgünlük sonuç grafikleri



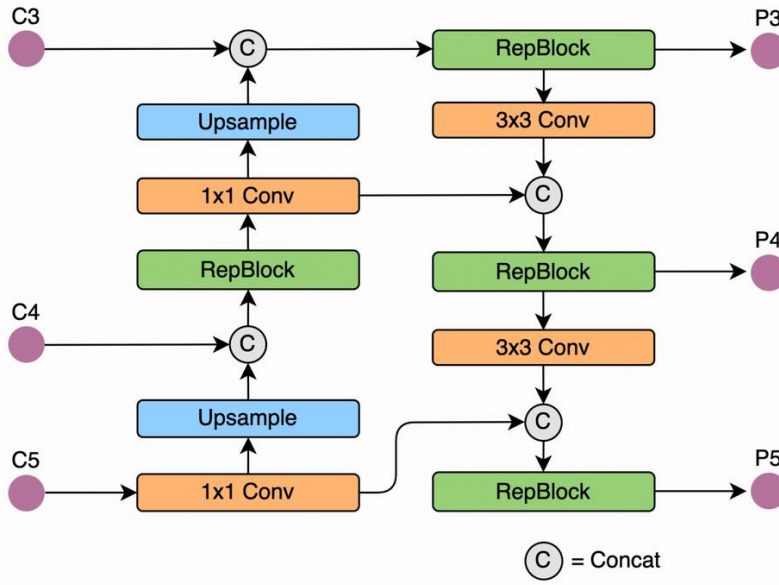
Şekil 4.14. YOLOv5 mimarisi karmaşıklık matrisi



Resim 4.3. YOLOv5 modelin test sonuçlarından bazı görseller

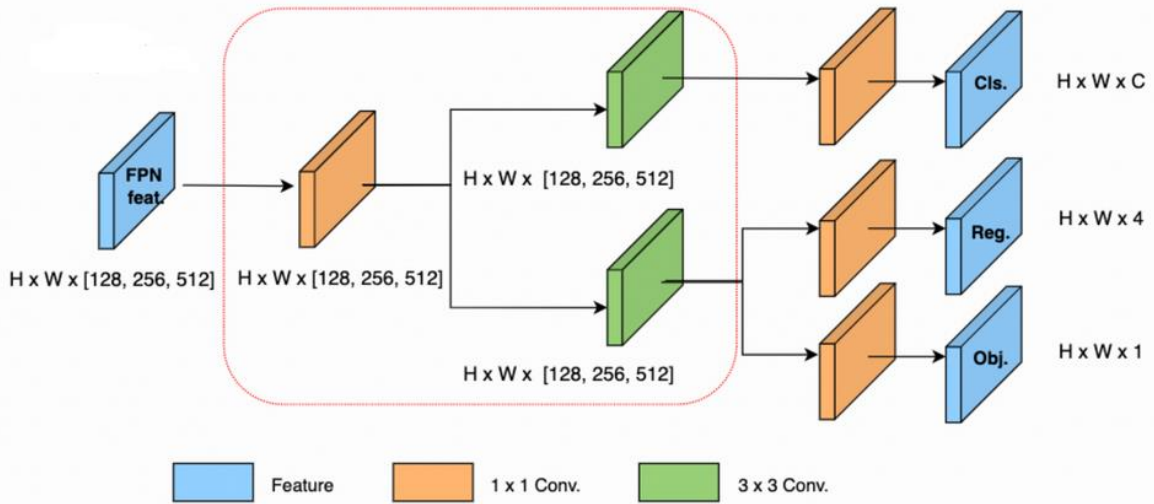
#### 4.4.5. YOLOv6

YOLOv6 mimarisi YOLO mimarisinden esinlenerek geliştirilmiş bir gerçek zamanlı nesne tespit algoritmasıdır. YOLOv6'nin tasarımı daha düşük donanımsal ihtiyaçlarla daha başarılı sonuçlar elde etmek üzerine oluşturulur. YOLOv6 mimarisi 640 pikselden büyük görüntüler ile eğitim yapmak için henüz hazır değildir. Eğitime girdi görüntüleri 640 x 640 olarak verilmektedir. YOLOv5 mimarisi YAML kullanmasına karşılık YOLOv6 mimarisinde model parametreleri Python'da tanımlanır. YOLOv6 mimarisi için ilerleyen zamanlarda daha esnek yapıda olması sayesinde daha farklı sürümlerinin geliştirilmesi ile farklı parametrelerinin özelleştirilmesi ile daha iyi sonuçlar elde edeceği düşünülmektedir. YOLOv6 mimarisi, YOLO boyun ve omurga yapısını donanıma göre yeniden tasarlayarak EfficientRep omurgası ve Rep-PAN boyun yapısını sunmaktadır. Rep-PAN yapısı Şekil 4.15'te görülmektedir.



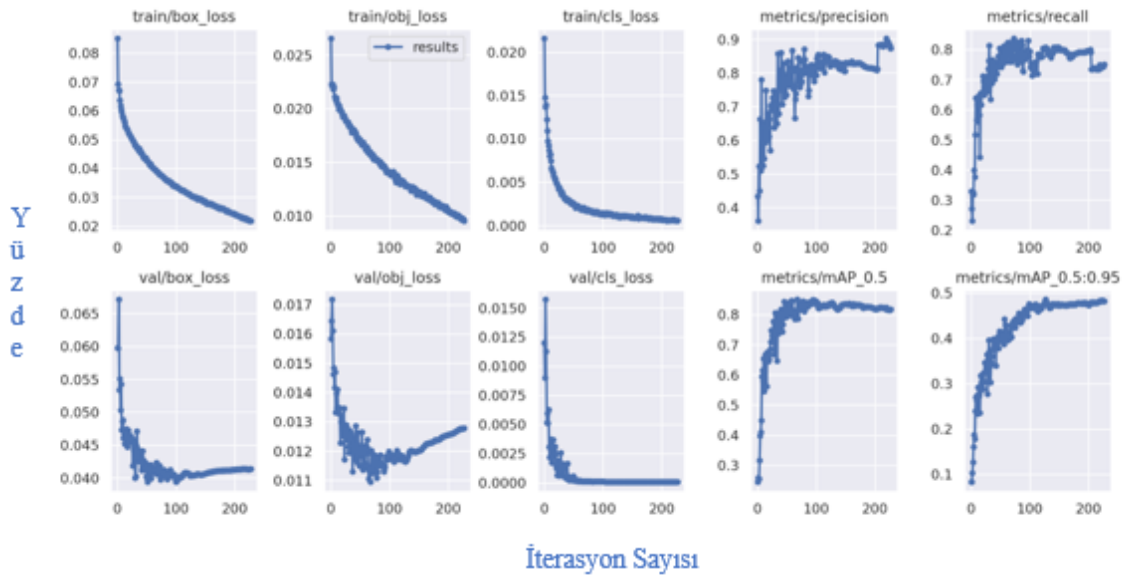
Şekil 4.15. YOLOv6 mimarisini Rep-PAN boyun yapısı

YOLOv6 mimarisinde diğer YOLO mimarilerine göre farklı olan ayrılmış kafa yapısı Şekil 4.16'da görülmektedir.

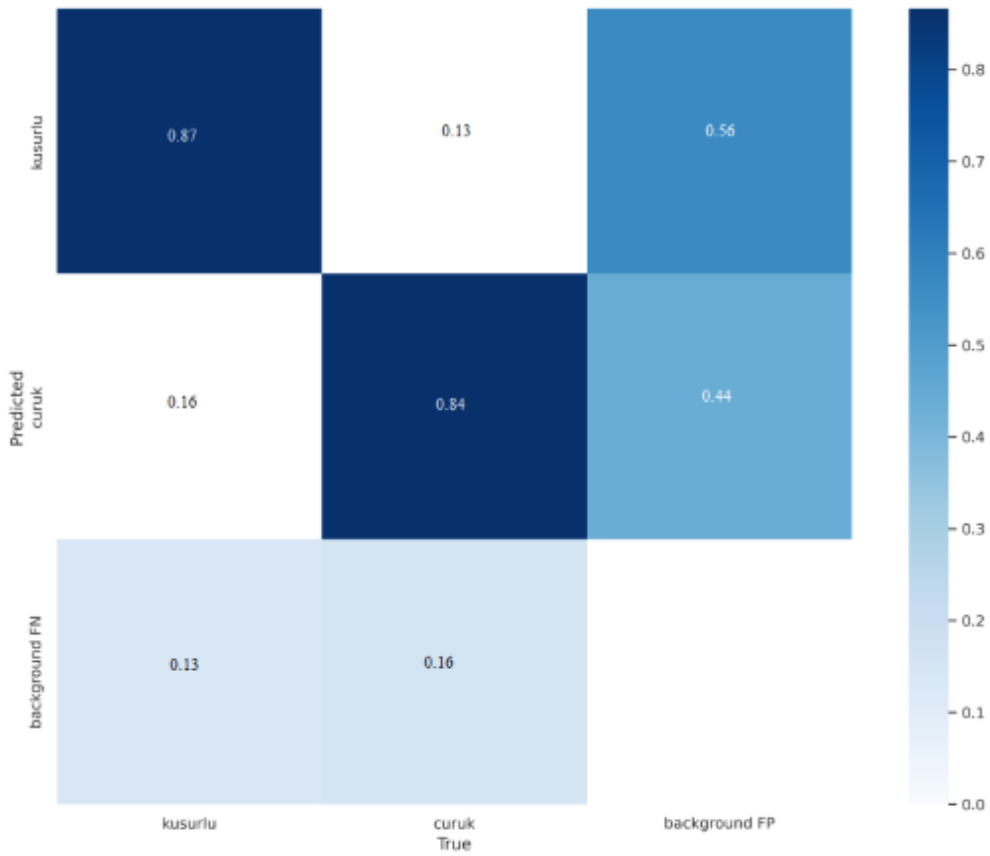


Şekil 4.16. Yolov6 mimarisini ayrılmış kafa yapısı

YOLOv6 mimarisini ile gerçekleştirilen eğitimlerin ortalama kesinlik ve kayıp miktarı ile ilgili görseller Şekil 4.17'de görülmektedir. YOLOv6 mimarisini eğitim sonuçları karmaşıklık matrisi Şekil 4.18'de görülmektedir. Test verileri üzerindeki modelin tahmin sonuçları ile ilgili görseller Resim 4.4'te gösterilmektedir.



Şekil 4.17. YOLOv6 mimarisi eğitim ortalama kesinlik ve kayıp miktarı sonuçları



Şekil 4.18. YOLOv6 mimarisi karmaşıklık matrisi

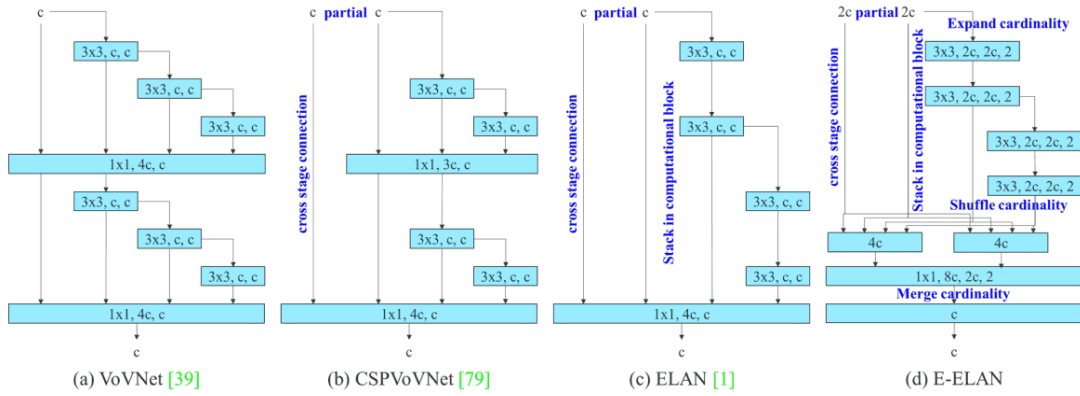




Resim 4.4. YOLOv6 mimarisi test verileri tahmin sonuçlarından görseller

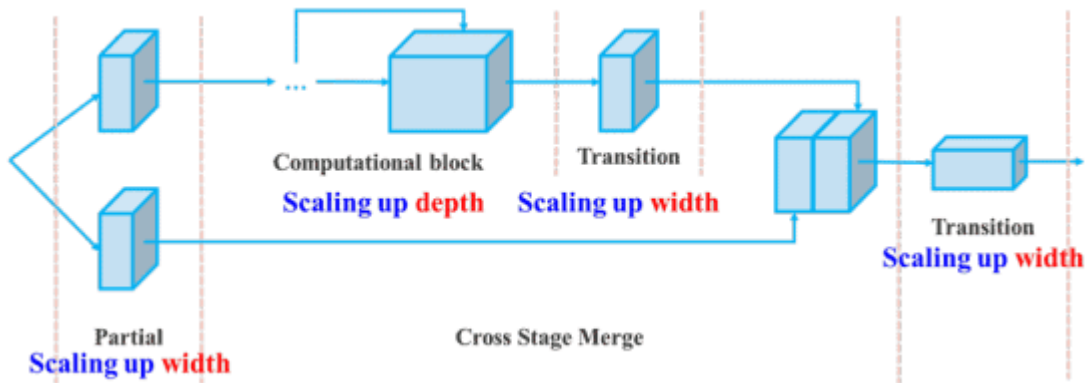
#### 4.4.6. YOLOv7

YOLOv7 mimarisi Scaled YOLOv4, YOLOv4 ve YOLO-R dan türetilmiş bir mimaridir. Ağ verimliliğini artırmak için önceki mimarilerden ilham alarak, doğruluk ve hızı etkileyen G/Ç kanal oranı, aktivasyon fonksiyonları, gradyan yolu ve bellek erişim maliyeti gibi faktörleri hesaplayarak geliştirilmiştir. E-ELAN yapısı YOLOv7 mimarisinde hesaplama bloğudur. E-ELAN var olan gradyan yolunu bozmadan ağın öğrenme yeteneğini sürekli bir şekilde geliştirebilmek için karıştırma, genişletme ve birleştirme kardinalitesini kullanır. E-ELAN sadece hesaplama bloğundaki mimariyi değiştirirken, geçiş katmanlarındaki yapı tamamen aynı kalmaktadır. Grup evrişimini kullanarak blokların kanalını ve kardinalitesini genişletir. Bu sayede mimari daha iyi şekilde öğrenir. E-ELAN yapısı Şekil 4.19'da gösterilmektedir.



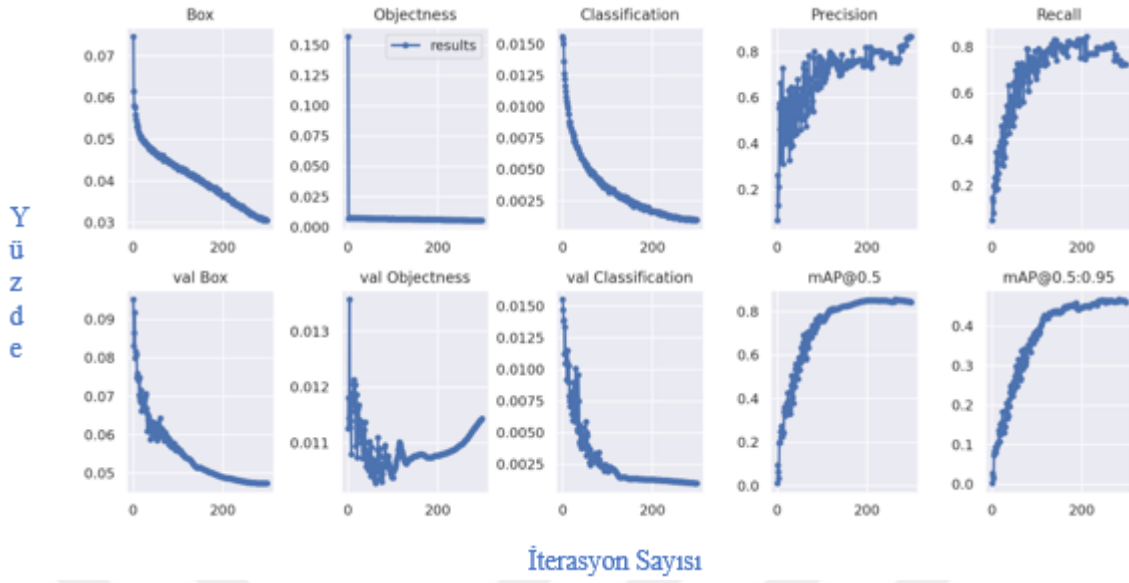
Şekil 4.19. E-ELAN mimarisinin yapısını gösteren görseller

YOLOv7 mimarisinde bileşik model ölçekleme yaklaşımı edilmiş şekilde sunuluyor. Mimaride derinlik ve genişlik, birleştirme tabanlı modellerde kullanmak için tutarlı bir şekilde ölçeklenir. YOLOv7 mimarisi bileşik ölçeklendirme yapısı Şekil 4.20'de gösterilmektedir.

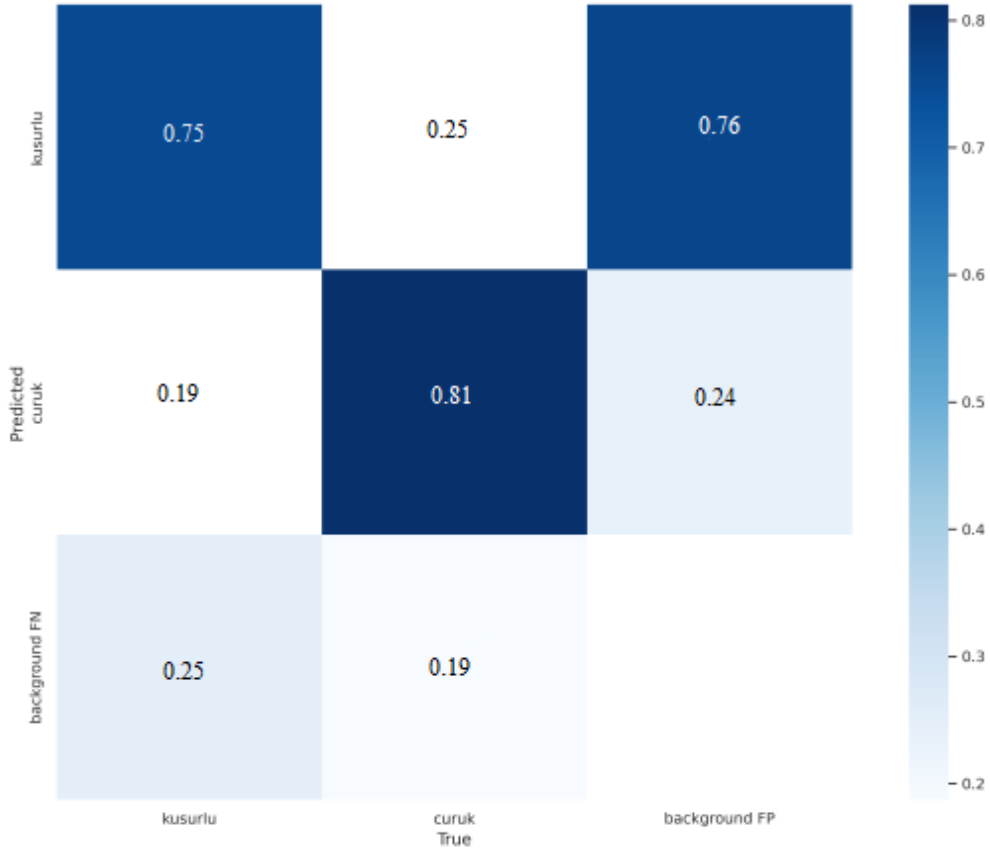


Şekil 4.20. YOLOv7 mimarisi bileştirme yapısının görseli

YOLOv7 mimarisinde parametreler yeniden yapılandırılarak modelin maliyetini artırmadan hızını artıracak bir yapı tanıtıldı. YOLOv7 mimarisi ile yapılmış olan eğitimlerin ortalama kesinlik değerleri ve kayıp miktarlarının grafikleri Şekil 4.21'de gösterilmektedir. Yapılan eğitimlerin karmaşıklık matrisi Şekil 4.22'de gösterilmektedir. Resim 4.5'te YOLOv7 mimarisi eğitim sonucu test verileri tahmin sonuçlarından görseller görülmektedir.



Şekil 4.21. YOLOv7 mimarisi ile yapılan eğitimlerdeki ortalama kesinlik değerleri ve kayıp miktarının grafikleri



Şekil 4.22. YOLOv7 mimarisi karmaşıklık matrisi



Resim 4.5. YOLOv7 mimarisi test verileri tahmin sonuçlarından görseller

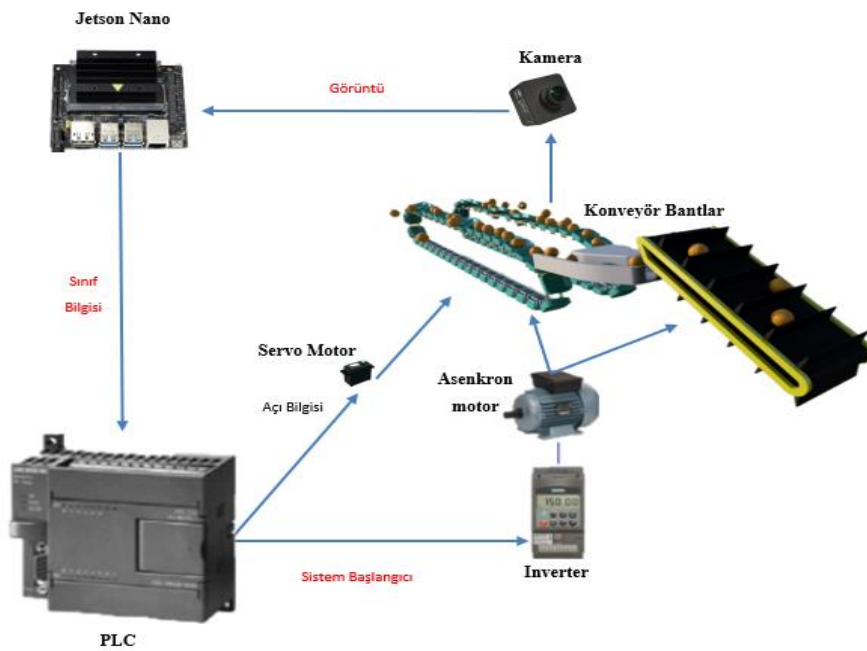
#### 4.5. Gerçek Zamanlı Sistem Denemeleri

Bu çalışmada kurulacak sistem iki aşamadan oluşmaktadır. İlk olarak veriler toplanıp etiketlenmiş, daha sonra YOLO algoritmasının farklı versiyonlarında eğitimler gerçekleştirilmiştir. İkinci aşamada sistemin donanımlarının belirlenmesi ve kurulmasıdır. Sistem donanımları fayda maliyet açısından ayrı ayrı değerlendirilmiş endüstriye en uygun sistemin tasarlanmasına önem verilmiştir. Tasarlanan sistemin ana gövdesini konveyör bantlar oluşturmaktadır. Konveyör bantlar iki kısımda kullanılacaktır. Birinci kısım ürünlerin dolun haznesinden ikinci konveyör bantına taşınması için kullanılacak, ikinci konveyör bant üzerinde taşıma kapları olacak şekilde tasarlanmıştır. Konveyör bantların hareketini, kamera ve modelin saniyede yakalayacakları görüntü sayısına ve ürünlerin modeldeki sınıflandırılma sürelerine uygun şekilde ayarlayabilmek için step motor tercih edilecektir. Step motor tercih edilmesinin başlıca sebepleri endüstriye uygun, frekans durumları değiştirilerek istenilen devir sayısının elde edilebiliyor olması ve maliyet olarak

uygun olmasıdır. Konveyör bantlar üzerinde ki taşıma kapları 60 derece dönebilecek şekilde ayarlanacak şekilde tasarlanmıştır. Ürünlerin görüntülerini almak için 2 adet 1080p çözünürlükte 60 FPS kamera kullanılacaktır. Taşıma kaplarındaki ürünlerin sitem dışına itilmesi için bant kenarlarına her sıra için iki adet servo motor yerleştirilecektir. Modelin kaydedilerek görüntülerdeki ürünlerin tespit ve sınıflandırılması için yapay zekâ bilgisayarı olan Jetson Nano sisteme entegre edilecek, kameralardan yüksek FPS alabilmek için jetson nano içerisinde kaydedilen model TensorRt ile dönüştürülecektir.

Tasarlanan derin öğrenme tabanlı meyve ayırıştırma sisteminin yapısını test etmek için maliyette düşünülerek var olan malzemeler ile bir prototip geliştirilmiş, geliştirilen prototipte testleri gerçekleştirmek için portakal meyvesi için yeni bir veri seti oluşturulmuş, YOLO algoritmasının farklı mimarilerinde eğitimler gerçekleştirilmiştir. YOLO algoritmasının farklı mimarilerinin doğruluk ve sistem hızı açısından testleri yapılan prototip üzerinde gerçekleştirilmiştir. Modeller TensorRt ile dönüştürülmeden prototip çalıştırıldığında gerçek zamanlı alınan görüntülerde maksimum 1080p çözünürlükte maksimum 16 FPS görüntü alınabilirken, model TensorRt ile dönüştürüldükten sonra 1080 p çözünürlükte maksimum 30 FPS'e kadar görüntü alınmıştır. Geliştirilen prototip üzerinde servo motorlar ve sistem Ardinio üzerinden kontrol edilmektedir.

Tasarlanan derin öğrenme tabanlı meyve ayırıştırma sistemin çalışma prensipleri Şekil 4.23'te gösterilmektedir.



Şekil 4.23. Kurulan sistemin çalışma prensibi akış diyagramı

Tasarlanan sistemin gerçek zamanlı olarak testlerini gerçekleştirmek için sistem prototipi geliştirilmiştir. Prototipte tek kanal üzerinde yuvarlanarak portakal ilerlemektedir. Kanalda ilerleyen portakal kendi etrafında dönerek ilerleyeceğinden kanal üzerine yerleştirilen kamera ile portakalın tüm yüzeylerinin görüntüsü alınacaktır. Kanalda 2 adet servo motor ile açılıp kapatılan kanatlar ve kanal sonu çıkışı bulunmaktadır. Python programlama dilinde yazılmış gerçek zamanlı nesne tespit algoritması ile eğitilmiş model jetson nano içerisine kaydedilmiştir. Kameradan yakalanan portakal görüntüleri jetson nano da model tarafından sınıflandırılarak sınıfına göre belirlenen kanatları kontrol eden servo motorlara bilgi verilecektir. Prototip olarak kurulacak düzeneğin haberleşme ve kontrolü programlanabilir mantık denetleyicisi tarafından gerçekleştirilmiştir. Aynı zamanda test için üretilen prototip küçük işletmelerde ürünleri sınıflandırmak için kullanılabilir. Prototipin üzerine yapılan testler sırasında alınan örnek görsel Resim 4.6’te görülmektedir.



Resim 4.6. Sistem prototipinin bir görseli

Gerçek zamanlı nesne takibi için saniyede yakalanan görüntü sayısı nesne takip hızı ve doğruluğu için çok önemlidir. Modeller gerçek zamanlı sistem denemeleriyle ayrı ayrı test edilip en uygun model belirlenmiştir. Modeller TensorRt ile dönüştürülmeden prototip üzerinde test edilmiş, saniyede yakalanan görüntü sayısı düşük kaldığı gözlemlenmiştir. FPS değerinin düşük olması sistem hızını oldukça yavaşlatacağı ve tespiti zorlaştırdığı gözlemlenmiştir. TensorRt ile dönüştürülen modeller ile çok daha yüksek FPS değerleri elde

edilmiştir. En yüksek FPS değerine YOLOv4-tiny ile ulaşılmıştır. (Bkz. Çizelge 4.1) Prototip üzerinde modeller ile elde edilen FPS değerleri görülmektedir.

Çizelge 4.1. Gerçek zamanlı testlerde elde edilen FPS değerleri

	Model	FPS (avg)	TensorRT FPS
720 p	YOLOv4-tiny	26	55
	YOLOVv4	14	33
	YOLOv4-p6	11	24
	YOLOv5	7	11
	YOLOv6	3,52	7,2
	YOLOv7	5	11
1080 p	YOLOv4-tiny	12	25
	YOLOVv4	8	15
	YOLOv4-p6	7	13
	YOLOv5	4	8,5
	YOLOv6	1,9	3,7
	YOLOv7	3	7

Modellerin katman sayıları elde edilen FPS değerlerinde önemli bir noktadır, ancak modellerin katman sayıları azaldıkça FPS değerleri yükseldikçe elde edilen sınıflandırma başarımları düşmüştür. En iyi sınıflandırma başarımları eğitimde en yüksek ortalama kesinlik değeri elde edilen YOLOv6 mimarisinde sağlanmıştır. Sistem hızı endüstri de çok önemli olmasına rağmen sistemin sınıflandırma doğruluğu göz önüne alındığında daha yavaş olan YOLOv5 modeli ayrıştırma sistemine entegre edilmiştir. Modeller ile 210 portakalın gerçek zamanlı ayrıştırma testinde elde edilen sınıflandırma doğruluk sonuçları Çizelge 4.2’de gösterilmektedir.

Çizelge 4.2. Prototip üzerinde 210 portakalın sınıflandırma doğruluk çizelgesi

Model	Sınıf	Örnek Sayısı	Özgünlük (%)	Hassasiyet (%)	G. Doğruluk (%)
YOLOv4-tiny	Kusurlu	70	67,1	58	59,5
	Çürük	82	34,1	82,3	
	Sağlam	58	100	53,2	
YOLOv4	Kusurlu	70	78,5	77,4	69,5
	Çürük	82	40,2	86,8	
	Sağlam	58	100	57,4	
YOLOv4-P6	Kusurlu	70	84,2	92	73,8
	Çürük	82	46,3	92,6	
	Sağlam	58	100	61,7	
YOLOv5	Kusurlu	70	97,1	93,1	89,5
	Çürük	82	75,6	100	
	Sağlam	58	100	77,3	
YOLOv6	<b>Kusurlu</b>	<b>70</b>	<b>92,8</b>	<b>97</b>	<b>94,2</b>
	<b>Çürük</b>	<b>82</b>	<b>91,4</b>	<b>97,4</b>	
	<b>Sağlam</b>	<b>58</b>	<b>100</b>	<b>87,9</b>	
YOLOv7	Kusurlu	70	82,8	89,2	84,7
	Çürük	82	75,6	93,9	
	Sağlam	58	100	73,4	



## 5. TARTIŞMA VE ÖNERİLER

### 5.1. Tartışma

Bu tez çalışmasında, tarım alanında kullanmak için derin öğrenme tabanlı otomatik meyve ayrıştırma sistemi tasarlanmıştır. Sistem tasarımı iki aşamada gerçekleştirilmiştir. Birinci aşama sistemde kullanılacak gerçek zamanlı derin öğrenme modellerinin belirlenmesidir. Sistemin uygunluğunun test edilmesi için portakal meyvesi için sağlam, kusurlu ve çürük portakal görüntüleri toplanarak etiketlenmiştir. Oluşturulan veri seti için gerçek zamanlı nesne tespit algoritması olan YOLO algoritmasının 6 farklı mimarisinde eğitimler gerçekleştirilip sonuçları değerlendirilerek, sisteme en uygun model belirlenmiştir. İkinci aşamada tasarlanan sistemin prototipi oluşturularak, yazılım sisteme entegre edilmiş ve testler gerçekleştirilmiştir. Literatürde meyvelerin sınıflandırılması için birçok çalışma bulunmasına karşı bu çalışmaların endüstriye dönüştürülme oranı oldukça düşüktür. YOLO modellerinin gerçek zamanlı deneysel sonuçları (Bkz. Çizelge 4.1) gösterilmiştir.

Bilgisayarlı görü sistemleri ile meyve sınıflandırma ve sıralama için literatür incelendiğinde Sofu ve diğerleri (2016) üç elma türünü renk, ağırlık, boyut ve kusurlu kusursuz olarak sıralamak için yapay görmeden faydalanarak otomatik elma sıralama sistemi tasarlamışlardır. Elmaları renklerine göre sınıflandırmak için RGB renk modelini kullanırken, elmaların boyutlarını belirlemek için elmanın merkezinden enine ve boyuna piksel değerlerini alarak santimetreye çevirdiler, bu uzunlukları elmanın çapı olarak kabul edip elmaları boyutlandırdılar. Pencere tabanlı bir yaklaşımla elmaları kusurlu kusursuz olarak ayırtmışlardır. Önerdikleri bu yaklaşımla elma sınıflandırma ve sıralamada sistem hızının optimum ayarlarında %79 doğruluk elde etmişlerdir. Mukhtar ve diğerleri (2021) limonları olgun, yarı olgun ve kusurlu olgunlaşmamış olarak sıralamak için raspberry pi kameradan aldıkları görüntüleri sinir ağında sınıflandırmış 99 adet limon verisi için yapmış oldukları testlerde %94 doğruluk elde ettiler. Bu çalışmada portakalların kusurlu, çürük ve sağlam olarak sınıflandırılmasında YOLOv5 modelinde %89,5 genel doğruluk elde edilmiştir. Literatür ile bu çalışmadaki farklı modellerin doğruluk karşılaştırması Çizelge 5.1'de gösterilmektedir.

Çizelge 5.1. Literatürdeki farklı çalışmalar ile bu çalışmadaki farklı modellerin sınıflandırma başarımı karşılaştırılması

Çalışmayı yapan	Adet	Ürün	Çalışma Hızı (ürün/sn)	Doğruluk (%)
Capizzi ve diğerleri (2016)		Portakal		88
Sofu ve diğerleri (2016)		Elma	15	79
Leelavathy ve diğerleri (2021)		Portakal		78,57
Chen ve diğerleri (2021)		Portakal		93,6
Mukhtar ve diğerleri (2021)	99	Limon		94
<b>Bu çalışmada YOLOv4-tiny</b>	210	Portakal	25	59,5
<b>Bu çalışmada YOLOv4</b>	210	Portakal	17	69,5
<b>Bu çalışmada YOLOv4-P6</b>	210	Portakal	13	73,8
<b>Bu çalışmada YOLOv5</b>	<b>210</b>	<b>Portakal</b>	<b>8</b>	<b>89,5</b>
<b>Bu çalışmada YOLOv6</b>	<b>210</b>	<b>Portakal</b>	<b>2</b>	<b>94,2</b>
<b>Bu çalışmada YOLOv7</b>	210	Portakal	4	84,7

Cappizzi ve diğerleri (2016) portakal meyvesinin kusurlu olanlarının belirlenmesi için yapmış oldukları çalışmada sağlam ve kusurlu portakalların sınıflandırılmasında %88 doğruluk elde etmişlerdir. Leelavathy ve diğerleri (2021) portakal meyvesini çürük ve sağlam olarak sınıflandırmak için yapmış oldukları çalışmada %78,57 doğruluk elde etmişlerdir. Chen ve diğerleri (2021) kusurlu portakalların tespiti için kurmuş oldukları sistemde %93,6 hassasiyet elde etmişlerdir. Chen ve diğerleri (2021) kurmuş oldukları sistem sadece kusurlu portakalları tespit etmek üzerine yapılmış bir çalışma olduğundan, Literatürdeki diğer portakal meyvesi için yapılmış ve Çizelge 5.1’de sonuçları verilmiş olan çalışmaların portakalların ikili sınıflandırması üzerine yapılmış çalışmalar olması, ama bu tez çalışmasında yapılan çalışmanın portakal meyvesinin sınıflandırması ve ayrıştırılması üzerine yapılmış olması ve kusurlu, çürük ve sağlam olarak 3 sınıfta ayrıştırılma yapılmış olması sebebiyle literatürdeki bu çalışmalarla karşılaştırılması çok sağlıklı olmayacaktır.

## 5.2. Öneriler

Yapılan çalışmanın veri seti artırılarak farklı yaklaşımlarla incelenmesi gerekmektedir. Ayrıca farklı meyve ve sebze türleri içinde önerilen yöntemler incelenmeli farklı türler için sonuçlar incelenmelidir. Sistem hızını artıracak yeni yaklaşımlar ve donanım yapıları incelenmesi bundan sonra yapılacak çalışmalara büyük oranda yol gösterici olacaktır.



## KAYNAKLAR

- ÖZSOYLU, Ahmet Fazıl. Endüstri 4.0. *Çukurova Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi*, 2017, 21.1: 41-64.
- GERDAN, Dilara. *Görüntü işleme teknikleri kullanılarak bazı meyvelerin sınıflandırılması*. 2020. Master's Thesis. Fen Bilimleri Enstitüsü.
- SURYA PRABHA, D.; SATHEESH KUMAR, J. Assessment of banana fruit maturity by image processing technique. *Journal of food science and technology*, 2015, 52.3: 1316-1327.
- Mohammadi, Vahid, Kamran Kheiralipour, and Mahdi Ghasemi-Varnamkhasti. "Detecting Maturity of persimmon fruit based on image processing technique." *Scientia Horticulturae* 184 (2015): 123-128.
- Capizzi, G., LO SCIUTO, G. R. A. Z. I. A., Napoli, C., Tramontana, E., & WOŹNIAK, M. (2016). A Novel Neural Networks-Based Texture Image Processing Algorithm for Orange Defects Classification. *International Journal of Computer Science & Applications*, 13(2).
- SAHU, Dameshwari; DEWANGAN, Chitesh. Identification and classification of mango fruits using image processing. *Int. J. Sci. Res. Comput. Sci. Eng. Inf. Technol*, 2017, 2.2:203-210.
- HUSSAIN HASSAN, Nashat M.; NASHAT, Ahmed A. New effective techniques for automatic detection and classification of external olive fruits defects based on image processing techniques. *Multidimensional Systems and Signal Processing*, 2019, 30.2: 571-589.
- Chithra, P. L., & Henila, M. (2019). Fruits classification using image processing techniques. *International Journal of Computer Sciences and Engineering*, 7(5).
- CAROLINA, Caro Prieto Diana; DAVID, Nieto Tapias Deivis. Classification of oranges by maturity, using image processing techniques. In: *2014 III international congress of engineering mechatronics and automation (CIIMA)*. IEEE, 2014. p. 1-5.
- KAUR, Harpuneet; SAWHNEY, B. K.; JAWANDHA, S. K. Evaluation of plum fruit maturity by image processing techniques. *Journal of Food Science and Technology*, 2018, 55.8: 3008-3015.
- KHOJASTEHNZHAND, Mostafa; MOHAMMADI, Vahid; MINAEI, Saeid. Maturity detection and volume estimation of apricot using image processing technique. *Scientia Horticulturae*, 2019, 251: 247-251.
- Azarmdel, H., Jahanbakhshi, A., Mohtasebi, S. S., & Muñoz, A. R. (2020). Evaluation of image processing technique as an expert system in mulberry fruit grading based on ripeness level using artificial neural networks (ANNs) and support vector machine (SVM). *Postharvest Biology and Technology*, 166, 111201.
- Alam Siddiquee, K. N. E., Islam, M. S., Dowla, M. Y. U., Rezaul, K. M., & Grout, V. (2020). Detection, quantification and classification of ripened tomatoes: a comparative

- analysis of image processing and machine learning. *IET Image Processing*, 14(11), 2442-2456.
- Kumari, N., & Belwal, R. (2021). Hybridized approach of image segmentation in classification of fruit mango using BPNN and discriminant analyzer. *Multimedia Tools and Applications*, 80(4), 4943-4973.
- BHARGAVA, Anuja; BANSAL, Atul. Classification and grading of multiple varieties of apple fruit. *Food Analytical Methods*, 2021, 14.7: 1359-1368.
- ZHANG, Yu-Dong; SATAPATHY, Suresh Chandra; WANG, Shui-Hua. Fruit category classification by fractional Fourier entropy with rotation angle vector grid and stacked sparse autoencoder. *Expert Systems*, 2022, 39.3: e12701.
- HOSSAIN, M. Shamim; AL-HAMMADI, Muneer; MUHAMMAD, Ghulam. Automatic fruit classification using deep learning for industrial applications. *IEEE Transactions on Industrial Informatics*, 2018, 15.2: 1027-1034.
- Haggag, M., Abdelhay, S., Mecheter, A., Gowid, S., Musharavati, F., & Ghani, S. (2019). An intelligent hybrid experimental-based deep learning algorithm for tomato-sorting controllers. *IEEE Access*, 7, 106890-106898.
- Jahanbakhshi, A., Momeny, M., Mahmoudi, M., & Zhang, Y. D. (2020). Classification of sour lemons based on apparent defects using stochastic pooling mechanism in deep convolutional neural networks. *Scientia Horticulturae*, 263, 109133.
- Li, Z., Li, F., Zhu, L., & Yue, J. (2020). Vegetable recognition and classification based on improved VGG deep learning network model. *International Journal of Computational Intelligence Systems*, 13(1), 559-564.
- Elsharif, A. A., Dheir, I. M., Mettleq, A. S. A., & Abu-Naser, S. S. (2020). Potato Classification Using Deep Learning. *International Journal of Academic Pedagogical Research (IJAPR)*, 3(12).
- ROY, Kyamelia; CHAUDHURI, Sheli Sinha; PRAMANIK, Sayan. Deep learning based real-time Industrial framework for rotten and fresh fruit detection using semantic segmentation. *Microsystem Technologies*, 2021, 27.9: 3365-3375.
- Chougule, A., Pawar, A., Kamble, R., Mujawar, J., & Bhide, A. (2021). Recognizing Fresh and Rotten Fruits Using Deep Learning Techniques. In *Data Engineering and Intelligent Computing* (pp. 205-212). Springer, Singapore.
- Chen, Y., An, X., Gao, S., Li, S., & Kang, H. (2021). A deep learning-based vision system combining detection and tracking for fast on-line citrus sorting. *Frontiers in Plant Science*, 12, 171.
- Korchagin, S. A., Gataullin, S. T., Osipov, A. V., Smirnov, M. V., Suvorov, S. V., Serdechnyi, D. V., & Bublikov, K. V. (2021). Development of an Optimal Algorithm for Detecting Damaged and Diseased Potato Tubers Moving along a Conveyor Belt Using Computer Vision Systems. *Agronomy*, 11(10), 1980.
- Leelavathy, B., Sri Datta, Y. S. S., & Rachana, Y. S. (2021). Quality Assessment of Orange Fruit Images Using Convolutional Neural Networks. In *Proceedings of International*

- Conference on Computational Intelligence and Data Engineering* (pp. 403-412). Springer, Singapore.
- Abu-Jamie, T. N., Abu-Naser, S. S., Alkahlout, M. A., & Aish, M. A. (2022). Six Fruits Classification Using Deep Learning.
- Aish, M. A., Abu-Naser, S. S., & Abu-Jamie, T. N. (2022). Classification of pepper Using Deep Learning.
- Sofu, M. M., Er, O., Kayacan, M. C., & Cetişli, B. (2016). Design of an automatic apple sorting system using machine vision. *Computers and Electronics in Agriculture*, 127, 395-405.
- Mukhtar, T., Jamil, S., Arif, U., Razzaq, W., & Wasif, M. (2021). Lemon Grading and Sorting Using Computer Vision. *Engineering Proceedings*, 12(1), 55.
- Chopra, H., Singh, H., Bamrah, M. S., Mahbubani, F., Verma, A., Hooda, N., ... & Singh, A. K. (2021). Efficient fruit grading system using spectrophotometry and machine learning approaches. *IEEE Sensors Journal*, 21(14), 16162-16169.
- Liu, C. S., Hwang, J. C., Chen, L. R., & Fu, C. C. (2009, May). Development of new structure of brushless DC servo motor for ceiling fan. In *2009 4th IEEE Conference on Industrial Electronics and Applications* (pp. 2640-2643). IEEE.
- Meng, T., & Yu, L. (2009, December). The design of microcomputer control system for step motor. In *2009 Second International Symposium on Computational Intelligence and Design* (Vol. 1, pp. 446-449). IEEE.
- Aydoğan, H. (2019). Sanal ortamda tasarlanan programlanabilir denetleyicinin gerçek ortam uygulaması ve analizi.
- Oluwaleye, S., Oguntosin, V., & Idachaba, F. (2021). Conceptual design of smart multi-farm produce dehydrator using a low-cost programmable logic controller and raspberry pi. *F1000Research*, 10.
- Bhargava, A., & Bansal, A. (2021). Fruits and vegetables quality evaluation using computer vision: A review. *Journal of King Saud University-Computer and Information Sciences*, 33(3), 243-257.
- Batchelor, B. G. (Ed.). (2012). *Machine vision handbook* (pp. 801-870). London, UK:: Springer.
- BÜYÜKARIKAN, B., & ÜNCÜ, İ. S. (2019). BİLGİSAYARLI GÖRÜ SİSTEMLERİ İÇİN SİSTEM TASARIMI VE KONTROLÜ. *Selçuk Üniversitesi Mühendislik, Bilim Ve Teknoloji Dergisi*, 7(1), 228-240.
- PK, F. A. (1984). What is Artificial Intelligence?. "Success is no accident. It is hard work, perseverance, learning, studying, sacrifice and most of all, love of what you are doing or learning to do"., 6
- Learning, M. (1997). Machine learning. *Kluwer Academic Publishers*, 23(2), 3.
- Chollet, F. (2021). *Deep learning with Python*. Simon and Schuster.
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT press.
- HANBAY, K., & Hüseyin, Ü. Z. E. N. (2017). Nesne tespit ve takip metotları: Kapsamlı bir derleme. *Türk Doğa ve Fen Dergisi*, 6(2), 40-49.

- Karasulu, B. (2013). Videolardaki Hareketli Nesnelerin Tespit Ve Takibi İçin Uyarlanabilir Arkaplan Çıkarımı Yaklaşımı Tabanlı Bir Sistem. *Uludağ Üniversitesi Mühendislik Fakültesi Dergisi*, 18(1), 93-110.
- Talu, M. F. (2010). Nesne takip yöntemlerinin sınıflandırılması. *İstanbul Ticaret Üniversitesi Fen Bilimleri Dergisi*, 9(18), 45-34.
- TELÇEKEN, M., & KUTLU, Y. Detecting Tagged People in Camera Images Kamera Görüntülerinde Etiketlenen Kisilerin Tespit Edilmesi.
- Redmon, J., & Angelova, A. (2015, May). Real-time grasp detection using convolutional neural networks. In *2015 IEEE international conference on robotics and automation (ICRA)* (pp. 1316-1322). IEEE.
- Zhang, J., Huang, M., Jin, X., & Li, X. (2017). A real-time chinese traffic sign detection algorithm based on modified YOLOv2. *Algorithms*, 10(4), 127.
- Altan, G. (2019). DeepGraphNet: grafiklerin sınıflandırılmasında derin öğrenme modelleri. *Avrupa Bilim ve Teknoloji Dergisi*, 319-327.
- Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. M. (2020). Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*.
- Zhu, L., Geng, X., Li, Z., & Liu, C. (2021). Improving yolov5 with attention mechanism for detecting boulders from planetary images. *Remote Sensing*, 13(18), 3776.

## DİZİN

**A**

ABSTRACT	v, vii
Asenkron	43
ayrıştırma	iv, xiii, 7, 9, 10, 11, 12, 14, 17, 31, 46, 47, 48

**B**

bant	2, 5, 6, 8, 10, 11, 13, 14, 17, 31, 43
bilgisayarlı görü	6
Boyun	40
bölütleme	3

**Ç**

Çapa Kutuları	29
çürük	33

**D**

DenseNet	34
derin öğrenme	iv, xi, 2, 5, 7, 12, 17, 18, 23, 31, 47
doğruluk	7

**E**

Endüstri	3, 1, 49
Etiketleme	32

**G**

Görüntü	17
görüntü işleme	2, 3
Güven Skoru	29

**H**

hasat	1
Havuzlama Katmanı	31

**I****J**

Jetson Nano	vii, 9, 14, 44
-------------	----------------

**K**

Kalman filtresi	25
katman	2
Konveyör	9
kusurlu	3

**M**

Makine öğrenmesi	19
meyve	iv, 1, 2, 3, 5, 8, 10, 11, 13, 20, 47, 48

**N**

Nesne	16
Nesne takibi	23
nesne tespiti	24

**O**

Omurga	40
otomatik	iv, 1, 3, 5, 6, 7, 9, 12, 14, 17, 18, 31, 47

**Ö**

ön işlem	2, 3, 5, 7, 32
öznitelik	3

**P**

parçacık filtreleme	25
PLC	15

**R**

Raspberry Pi	7
--------------	---

**S**

sebze	1, 2, 3, 47, 48
servo	11
Sistem	2, 7, 8, 9, 12, 41, 43, 45, 46, 48, 52
Step motor	14

**T**

Tarım	1
taşma	8, 10, 11, 13, 43

**V**

veri seti	8, <i>Bakın</i>
-----------	-----------------

**Y**

yapay zekâ	17
YOLO	iv, v, xi, 2, 26, 29, 30, 31, 43, 47





# TEKNOVERSITE



teknoversite **AYRICALIĞINDASINIZ**

**İSTE**

