



**T.C.
AMASYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**TÜTÜN YAPRAĞI ÜZERİNDE BULUNAN HASTALIKLARI
MORFOLOJİK GÖRÜNTÜ İŞLEME METOTLARIYLA TESPİT
EDEN GERÇEK ZAMANLI GÖMÜLÜ SİSTEM TASARIMI**

YÜKSEK LİSANS TEZİ

CEMİL ERGİN

TEMMUZ

CEMİL ERGİN

TEKNOLOJİ VE İNOVASYON YÖNETİMİ ANABİLİM
DALI

TEMMUZ 2024

**TÜTÜN YAPRAĞI ÜZERİNDE BULUNAN HASTALIKLARI
MORFOLOJİK GÖRÜNTÜ İŞLEME METOTLARIYLA TESPİT EDEN
GERÇEK ZAMANLI GÖMÜLÜ SİSTEM TASARIMI**

Cemil ERGİN

**YÜKSEK LİSANS TEZİ
TEKNOLOJİ VE İNOVASYON YÖNETİMİ ANABİLİM DALI**

**Danışman
Dr. Öğr. Üyesi Çetin Cem BÜKÜCÜ**

**AMASYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

TEMMUZ 2024

Yüksek Lisans Tezi Kabul ve Onay Sayfası

Cemil ERGİN tarafından hazırlanan “Tütün Yaprağı Üzerinde Bulunan Hastalıkları Morfolojik Görüntü İşleme Metotlarıyla Tespit Eden Gerçek Zamanlı Gömülü Sistem Tasarımı “ adlı tez çalışması aşağıdaki jüri tarafından OY BİRLİĞİ ile Amasya Üniversitesi Fen Bilimleri Enstitüsü Teknoloji ve İnovasyon Yönetimi Anabilim Dalında YÜKSEK LİSANS TEZİ olarak kabul edilmiştir.

Danışman: Dr. Öğr. Üyesi Çetin Cem BÜKÜCÜ

Teknoloji ve İnovasyon Anabilim Dalı, Amasya Üniversitesi

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

Başkan: Doç. Dr. Levent GÖKREM

Elektronik Anabilim Dalı, Tokat Gaziosmanpaşa Üniversitesi

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

Üye: Dr. Öğr. Üyesi Yavuz ÜNAL

Bilgisayar Yazılımı Anabilim Dalı, Amasya Üniversitesi

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

Tez Savunma Tarihi: 12/07/2024

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

.....

Doç. Dr. Ümit YILDIRIM

Fen Bilimleri Enstitüsü Müdürü

ETİK BEYAN

Amasya Üniversitesi Fen Bilimleri Enstitüsü Tez Yazım Kurallarına uygun olarak hazırladığım bu tez çalışmada;

- 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.

Cemil ERGİN

12.07.2024

TÜTÜN YAPRAĞI ÜZERİNDE BULUNAN HASTALIKLARI MORFOLOJİK
GÖRÜNTÜ İŞLEME METOTLARIYLA TESPİT EDEN GERÇEK ZAMANLI
GÖMÜLÜ SİSTEM TASARIMI
(Yüksek Lisans Tezi)

Cemil ERGİN

AMASYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
Temmuz 2024

ÖZET

Tütün bitkisinin yetiştirme koşulları esnek olduğundan Türkiye'nin iç Anadolu bölgesi dışında tüm bölgelerinde yetiştirilmektedir. Ekonomik açıdan büyük öneme sahip tütün bitkisi 2022 yılında 828, 9 milyon dolarlık ihracat rakamına ulaşmıştır. Tütün yapraklarında meydana gelen hastalıklar, tütün bitkisinin büyüme ve gelişmesini olumsuz etkilemektedir. Bu durum verim kaybına dolayısı ile maddi kayıplara neden olmaktadır. Tütün yaprakları üzerindeki hastalıkların gözlem yoluyla tespiti zaman kaybına neden olabildiği gibi hatalı da olabilmektedir. Bu nedenle tütün yapraklarında meydana gelen hastalıklı alanların tespiti amacıyla gerçek zamanlı çalışan gömülü bir sistem modeli tasarlanmıştır. Bu sistemde, kameradan alınan görüntüler gömülü sisteme aktarılmaktadır. Daha sonra python yazılımı ile morfolojik görüntü işleme işlemi uygulanarak yapraklar üzerindeki hastalıklı alanlar tespit edilmiş ve gösterilmiştir. Ayrıca tütün yapraklarını hastalıklı ve sağlıklı olarak sınıflandırmak amacıyla evrişimsel sinir ağları modelleri kullanılmıştır. Samsun ilinde fotoğrafı çekilen 1600 sağlıklı ve 1600 hastalıklı tütün yaprağı içeren bir veri seti üzerinde bu modellerin performansları değerlendirilmiştir. Sistem, sınıflandırma işleminin sonucunda 3 modelde %93 ve üzeri başarı oranları elde etmiştir.

Sayfa Adedi : 94
Anahtar Kelimeler : Tütün yaprak hastalıkları, Gömülü sistemler, Evrişimli sinir ağları, Sınıflandırma
Danışman : Dr. Öğr. Üyesi Çetin Cem BÜKÜCÜ

DESIGN OF A REAL-TIME EMBEDDED SYSTEM THAT DETECTS DISEASES ON TOBACCO LEAVES USING MORPHOLOGICAL IMAGE PROCESSING

METHODS

(M. Sc. Thesis)

Cemil ERGİN

AMASYA UNIVERSITY
INSTITUTE OF SCIENCE

July 2024

ABSTRACT

Since it does not have very rigorous conditions of cultivation, the tobacco plant is produced in all regions of Turkey, except for the Central Anatolia region. The tobacco plant, which has huge economic importance, has reached an export figure of \$828.9 million in 2022. Diseases emerging in tobacco leaves act against the normal and healthy growth of the tobacco plant. This case causes yield loss and thus financial losses. This detection of diseases in tobacco leaves does cause time loss and could also be inaccurate. So a real-time embedded system model has been designed to detect the area of tobacco leaves that are diseased. In all these systems, images taken from the camera are transferred to the embedded system. After that comes the application of morphological image processing, with the help of a Python software, which allows the detection and display of the areas of the leaves affected by such disease. Classification of tobacco leaves into categories such as diseased and healthy was also done for developing convolutional neural network models. Performance was tested on a dataset of 1600 healthy and 1600 diseased tobacco leaves pictured in Samsun province. Through the classification process, it succeeded in success rates of 93% or more in individual three models.

Number of pages : 94
Keywords : Tobacco leaf diseases, Embedded systems, Convolutional neural networks, Classification
Supervisor : Dr. Öğr. Üyesi Çetin Cem BÜKÜCÜ

ÖN SÖZ VE TEŞEKKÜR

Bu çalışmanın gerçekleştirilmesinde desteklerini esirgemeyen, çalışma sırasında yönelttiğimiz her soruya sabırla cevap veren ve her zaman ulaşılabilir olan, birlikte çalışma fırsatı bulduğum için kendimi çok şanslı hissettiğim değerli danışmanım Dr. Öğr. Üyesi Çetin Cem BÜKÜCÜ ‘ye en içten teşekkürlerimi sunarım. Ayrıca, yüksek lisans sürecimde bana verdikleri destek ve iş birliği için kıymetli sınıf arkadaşlarıma da sonsuz teşekkür ederim. Proje çalışmalarım süresince yanımda olan, bana inanan ve destekleyen sevgili eşim Züleyha ERGİN'e ve sabrı için biricik kızım Ahsen ERGİN'e de minnettarım.



İÇİNDEKİLER

	Sayfa
ÖZET	iv
ABSTRACT.....	v
ÖN SÖZ VE TEŞEKKÜR.....	vi
İÇİNDEKİLER	vii
ÇİZELGELER DİZİNİ	x
ŞEKİLLER DİZİNİ	xi
RESİMLER DİZİNİ.....	xiii
SİMGELER VE KISALTMALAR DİZİNİ	xv
1. GİRİŞ.....	1
1.1. Literatür Tarama.....	4
2. DONANIM	12
2.1. Gömülü Sistemler.....	12
2.1.1. Raspberry pi 3 model b+	14
2.2. Raspberry Pi Uyumlu Kamera	19
3. YAZILIM.....	21
3.1. Görüntü İşleme.....	21
3.2. Görüntü İşleme Adımları	23
3.2.1. Ön işlem.....	24
3.2.2. Özellik çıkarımı	32
3.3. Raspberry Pi Os İşletim Sistemi.....	34
3.3.1. Raspberry pi kartına işletim sisteminin kurulması	35
3.3.2. Raspberry pi'ye ssh(uzak bağlantı) bağlantısı	37

3.3.3. Thonny ide programı	39
3.4. Python Programla Dili.....	40
3.4.1. Kullanılan kütüphaneler	42
4. YAPAY ZEKÂ	47
4.1. Derin Öğrenme (Deep Learning)	47
4.1.1. Evrişimsel sinir ağları.....	48
4.1.2. Önceden eğitilmiş evrişimsel sinir ağları mimarileri	50
5. YÖNTEM	57
5.1. Prototip Tasarımı.....	57
5.2. Yöntem Adımları.....	61
5.2.1. Kameradan görüntülerin alınması	61
5.2.2. Görüntüleri yeniden boyutlandırma.....	62
5.2.3. Gri tona dönüşüm	62
5.2.4. Eşik(Threshold) değeri tespiti.....	62
5.2.5. Morfolojik işlemler.....	62
5.2.6. Kontur(Contours) belirleme	63
5.2.7. Tütün yaprakları üzerindeki hastalıklı alanların tespiti	63
6. PERFORMANS ÖLÇÜTLERİ VE ESA MİMARİLERİ İLE SINIFLANDIRILMA İŞLEMİ	70
6.1. Başarım Ölçütleri	70
6.1.1. Karmaşıklık matrisi(confusion matrix)	70
6.1.2. Doğruluk (Accuracy).....	71
6.1.3. Duyarlılık (Recall).....	72
6.1.4. Precision(Hassasiyet).....	72
6.1.5. F1 skor(F-score)	72

6.1.6. Logaritmik kayıp (Loss)	73
6.1.7. Roc eğrisi(roc curve)	73
6.1.8. Intersection over union (IoU)	74
6.1.9. Transfer öğrenme eğitim verileri.....	74
6.2. Transfer Öğrenme Modelleri.....	75
6.2.1. VGG16 modeli	76
6.2.2. Resnet50 modeli	77
6.2.3. EfficientNetB6 modeli.....	78
6.3. Transfer Öğrenme Modellerinin Karşılaştırılması	80
6.4. Gerçek Zamanlı Çalışan Sistem Performans Değerleri.....	81
6.4.1. Gömülü sistem tasarımı performans değerleri	81
6.4.2. Bilgisayar ortamında çalışan sistemin performans değerleri.....	81
6.4.3. Gözlemsel inceleme ve görüntü işleme sonuçlarının karşılaştırılması.....	82
7. SONUÇLAR VE ÖNERİLER.....	84
KAYNAKLAR	86
ÖZGEÇMİŞ.....	94

ÇİZELGELER DİZİNİ

Çizelge	Sayfa
Çizelge 2. 1. Kaan hwk-w kamera özellikleri.....	20
Çizelge 6. 1. Eğitim verileri.....	74
Çizelge 6. 2. Vgg16 modeli değerlendirme verileri.....	77
Çizelge 6. 3. Resnet50 modeli değerlendirme verileri.....	78
Çizelge 6. 4. Resnet50 modeli değerlendirme verileri.....	79
Çizelge 6. 5. ESA modelleri değerlendirme verileri.....	80
Çizelge 6. 6. Gömülü sistem tasarımı karmaşıklık matrisi	81
Çizelge 6. 7. Gömülü sistem değerlendirme verileri	81
Çizelge 6. 8. Bilgisayar görüntü işleme karmaşıklık matrisi	82
Çizelge 6. 9. Bilgisayar görüntü işleme değerlendirme verileri	82
Çizelge 6. 10. Hastalıklı alanların gözlemsel ve bilgisayarlı görüntüleme yöntemleri ile sayımı ve karşılaştırılması.....	83

ŞEKİLLER DİZİNİ

Şekil	Sayfa
Şekil 2. 1. Gömülü sistem tasarımı	13
Şekil 2. 2. Raspberry pi 3 model b+ pin bağlantıları	17
Şekil 3. 1. Görüntü işleme adımları	23
Şekil 3. 2. Aşındırma işlemi uygulaması	29
Şekil 3. 3. Genişleme işlemi uygulaması	30
Şekil 3. 4. Açma işlemi uygulaması.....	31
Şekil 3. 5. Kapama işlemi uygulaması.....	32
Şekil 3. 6. Özellik çıkarım teknikleri	33
Şekil 3. 7. Raspberry pi'ye SSH bağlantısı	37
Şekil 3. 8. OpenCV bileşen yapısı	43
Şekil 3. 9. Matplotlib grafik örnekleri	45
Şekil 4. 1. Yapay zekâ, makine öğrenmesi, derin öğrenme kronolojisi.....	47
Şekil 4. 2. CNN katmanları.....	49
Şekil 4. 3. Model boyutu ve imagenet doğruluk oranları	50
Şekil 4. 4. VGG16 mimarisi	52
Şekil 4. 5. VGG16 mimarisi ile oluşturulan model	52
Şekil 4. 6. Artık (Residual) Öğrenme	53
Şekil 4. 7. Resnet50 mimarisi ile oluşturulan model	54
Şekil 4. 8. EfficientNet mimarisi	56
Şekil 4. 9. EfficientNet-B6 mimarisi ile oluşturulan model	56
Şekil 5.1. Tütün yaprak hastalıklı alan sayısı ve yüzdelik dilimi	59
Şekil 5.2. Gerçekleştirilen yöntem aşamaları	61

Şekil 5. 3. Akış diyagramı.....	61
Şekil 5. 4. Tütün hastalık tespiti aşamaları	64
Şekil 5. 5. Hastalıklı tütün yaprakların gösterilmesi.....	66
Şekil 5. 6.Sağlam tütün yaprağı	66
Şekil 5. 7. Tütün yaprakları hastalık sayıları ve oranları	69
Şekil 6. 1. İkili sınıflandırma için karmaşıklık matrisi	71
Şekil 6. 2. Roc eğrisi tanımlanması	73
Şekil 6. 3. Karmaşıklık matrisi	76
Şekil 6. 4. ROC eğrisi	76
Şekil 6. 5. Karmaşıklık matrisi	77
Şekil 6. 6. ROC eğrisi	78
Şekil 6. 7. Karmaşıklık matrisi	79
Şekil 6. 8. ROC eğrisi	79
Şekil 6. 9. ESA modellerinin doğruluk oranları	80

RESİMLER DİZİNİ

Resim	Sayfa
Resim 1. 1. Hastalıklı tütün yaprağı.....	1
Resim 1. 2. Tütün hastalıkları	2
Resim 2. 1. Raspberry pi 3 model b+.....	17
Resim 2. 2. Raspberry pi kartına alüminyum ve fan soğutucu uygulaması.....	19
Resim 2. 3. Raspberry pi 3 muhafaza kutusu.....	19
Resim 2. 4. Kaan hwk-w kamera modülü.....	20
Resim 3. 1. Piksellerden görüntü oluşturulması	21
Resim 3. 2. Renkli görüntünün gri görüntüye çevrilmiş gösterimi.....	24
Resim 3. 3. Bulanıklaştırma işlemi	25
Resim 3. 4. Klasik eşikleme yöntemi.....	27
Resim 3. 5. Orijinal görüntü ve histogram grafiği	28
Resim 3. 6. Aşındırma işlemi uygulanmış görüntü.....	29
Resim 3. 7. Aşındırma işlemi uygulanmış görüntü.....	30
Resim 3. 8. Açma işlemi uygulanmış resim	31
Resim 3. 9. Kapama işlemi uygulanmış resim.....	32
Resim 3. 10. Raspberry pi os masaüstü görünümü	35
Resim 3. 11. Raspberry Pi imager programı arayüzü	36
Resim 3. 12. Raspberry pi os ilk açılış ekranı.....	37
Resim 3. 13. Raspberry pi SSH özelliğinin aktif edilmesi	38
Resim 3. 14. VNC Viewer ile Raspberry pi kartına bağlanması	38
Resim 3. 15. Thonny IDE program arayüzü	39
Resim 3. 16. 2023 yılı programlama dilleri popülerlik oranları	41

Resim 5. 1. Gerçek zamanlı tütün yaprak hastalık tespiti yapan model	57
Resim 5. 2. Hastalıklı alanların mavi daire içerisinde gösterilmesi.....	58
Resim 5. 3. Gerçek zamanlı tütün yaprak hastalık tespiti yapan modelin çalışması	59
Resim 5. 4. Hastalıklı tütün yapraklarının bilgisayar ortamında tespit edilmesi	67
Resim 5. 5. Gerçek zamanlı hastalık tespiti ekran çıktısı	68
Resim 6. 1. Hastalıklı tütün yaprak veri seti	75
Resim 6. 2. Sağlıklı tütün yaprak veri seti	75



SİMGELER VE KISALTMALAR DİZİNİ

Bu çalışmada kullanılmış bazı simgeler ve kısaltmalar, yanda açıklamaları verilmek üzere aşağıda listelenmiştir.

Simgeler

Açıklama

F	Fonksiyon
$w, a, b, c, x, y, x_1, x_2$	Eşitlik değişkenleri
T	Zaman
A, B	Veri kümeleri
Δ	Delta işareti
Θ	Aşınma işlemi
$\emptyset$	Boş küme
$\oplus$	Genleşme işlemi
$\circ$	Açma işlemi
$\cdot$	Kapama işlemi

Kısaltmalar

Açıklama

RGB	Kırmızı, yeşil, mavi renkler
CNN	Derin evrimsel sinir ağlar
MFF	Çoklu özellik füzyon
RNN	Tekrarlayan yapay sinir ağları
İHA	İnsansız hava aracı
ORB	Özellik tanıma algoritması
PNN	Olasılıksal sinir ağları
YSA	Yapay sinir ağları

YOLOv3	Nesne tespit algoritması
FO	Solgunluk hastalığı
A&D	Analog dijital dönüştürücü
DA	Dijital analog dönüştürücü
SBC	Tek kart bilgisayar sistemini
GPIO	Genel giriş çıkış birimi
PCB	Devre kartı
ARM	İşlemci mimarisi
USB	Universal seri port
HDMI	Yüksek çözünürlüklü multimedya ara birimi
İOT	Nesnelerin interneti
VGA	Video grafik arabirimi
ESA	Evrişimli sinir ağları
SSH	Güvenli veri iletimi
VNC	Sanal ağ sistemi
BGR	Mavi, yeşil, kırmızı renk uzayı
H	Hastalıklı alan
THS	Toplam hastalık sayısı
THA	Hastalıklı alanların toplam yüzeyde kapladığı alan
TP	Toplam piksel sayısı
ROC	Alıcı işletim karakteristiği
AUC	ROC eğrisi altındaki alan
TPR	Doğru olasılık oranı
FPR	Yanlış olasılık oranı
FO	Fusarium oxysporum

1. GİRİŞ

Endüstri 3.0 ile hayatımızın her alanına giren bilgisayar ve bilgisayar sistemleri hayatın her alanında vazgeçilmez birer parçası olmuştur. Evlerdeki kullanılan eşyalardan, fabrikalarda kullanılan robotlara kadar her alanda bilgisayar sistemleri kullanılmaktadır. Üretimde otomasyon ve dijitalleşmenin artmasıyla ortaya çıkan endüstri 4.0 yapay zekâ, nesnelerin interneti gibi teknolojilerin kullanılması üretim süreçlerindeki verimlilik ve başarıyı artırmıştır (Öztürk ve Şahin, 2018).

Bitki yaprak hastalıkları, bitkilerin büyüme ve gelişmesini olumsuz etkileyen önemli bir sorundur. Bu hastalıklar, bitkilerin yaprakları üzerinde oluşan lekeler, resim 1.1’de görüldüğü gibi lekelerin büyümesi, kahverengileşme, sararma gibi belirtilerle kendini gösterir. Bitki yaprak hastalıklarının erken tespiti, bitki sağlığı ve verimliliği açısından kritik öneme sahiptir. Hastalıkların erken teşhisi, hastalığın yayılmasını ve bitkilerin etkilenme derecesini önemli derecede azaltmaktadır (Çetiner, 2021).

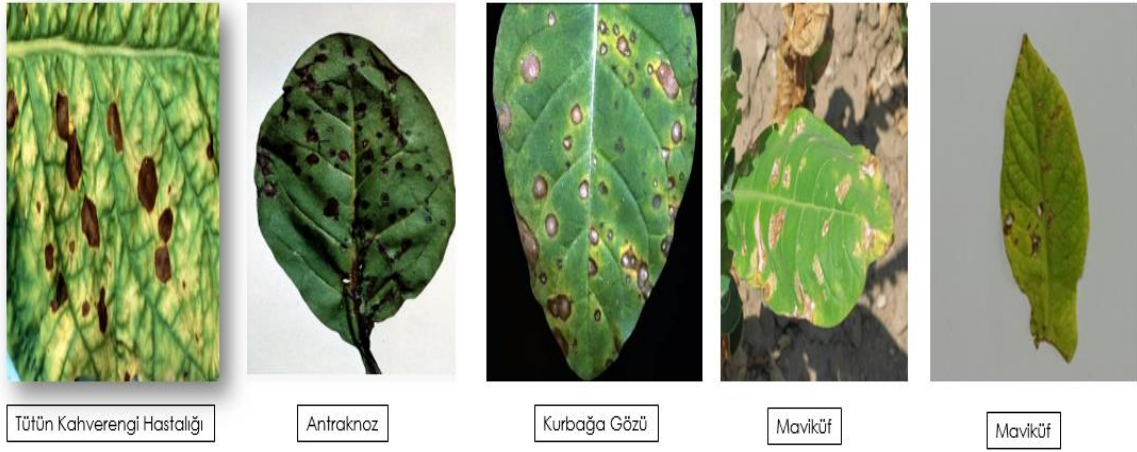


Resim 1. 1. Hastalıklı tütün yaprağı

Bitki yaprak hastalıklarının tespiti, bitki sağlığını korumak ve hastalığın yayılmasını önlemek için büyük bir öneme sahiptir. Geleneksel yöntemler olan gözlem yapmak, yaprakların hastalık belirtilerini kontrol etmek ve bitkilerin genel sağlık durumunu gözlemlmek hastalığın tespit edilmesinde etkili yöntemlerdir. Ancak geleneksel yöntemlerle hastalık belirtilerinin tespit edilmesi zaman alıcı ve hatalı olabilir. Bu nedenle, bitki yaprak hastalıklarının tespit edilmesinde yapay zekâ destekli uygulamaların kullanılması hastalıkların erken tespit edilmesinde daha etkin bir rol oynamaktadır.

Tütün yaprak hastalıkları, tütün yapraklarında bulunan çeşitli hastalıkların belirtileridir. Bu hastalıklar tütünün büyümesini ve verimini olumsuz yönde etkiler. Tütün yaprak hastalıklarına genellikle mantarlar, bakteriler ve virüsler gibi mikro organizmalar neden olur.

Tütün yaprak hastalıklarının belirtileri lekelenme, esmerleşme, solma, kabarma, kuruma ve yaprak dökümü olarak ortaya çıkar. Bu belirtiler hastalığın türüne ve şiddetine bağlıdır. Hastalık ilerledikçe bitki büyümesi ve yaprak verimi azalır ve hatta bitkiler ölebilir (Hastalık Zararlılar ve Arızalar Faktörü, b.t.). Tütün yaprak hastalıklarının bazıları Resim 1.2’de gösterilmektedir.



Resim 1. 2. Tütün hastalıkları

Tütün yaprak hastalıkları, tütün bitkisinin büyüme ve verimlilik sürecini olumsuz etkileyen önemli bir sorundur. Bu hastalıklar, tütün üreticileri için ekonomik kayıplara neden olmakta ve tütünün kalitesini düşürmektedir. Bu nedenle, tütün yaprak hastalıklarının tanınması, önlenmesi ve kontrolü tütün yetiştiriciliği için hayati öneme sahiptir.

Yapay zekâ, insan zekâsı ile ilgili zihinsel işlevleri bilgisayarlaştırma yeteneği olarak tanımlanır. Bunların modeller yardımıyla analiz edilmesi, formüle edilmesi, yapay sistemlere uygulanmasıdır (Erkutlu vd., 2023) .

Yapay zekâ, insan benzeri düşünme ve zekâ yeteneklerini gösterme kabiliyeti bilgisayar sistemlerine kazandıran bir bilim dalı olarak tanımlanabilir. Yapay zekânın temel hedefi, bilgisayarları, öğrenme, problem çözme, metin analizi, algılama, planlama ve karar verme

gibi insan benzeri yeteneklerle donatmaktır. Bu yetenekler, yapay zekânın geniş bir uygulama yelpazesine sahip olmasını sağlar (Copeland, 2023).

Görüntü işleme, dijital görüntülerin bilgisayarlar aracılığıyla analiz edilmesi, işlenmesi işlemidir. Görüntü işleme, görüntü iyileştirme, özellik çıkarma, örüntü tanıma, hedef tespiti, görüntü sıkıştırma vb. gibi birçok uygulama alanında kullanılmaktadır (Erdem Aytan vd.,1993).

Görüntü işleme, yararlı bilgiler elde etmek veya bir görüntünün görsel kalitesini artırmak amacıyla görüntüler üzerinde aritmetik işlemler gerçekleştirmek için hesaplama algoritmalarının kullanılmasıdır. Dijital görüntüleri işlemek için evrişim ve filtreleme gibi matematiksel işlemler kullanılmaktadır (Başaran ve Elbasan 2023).

Yapay zekâ ve görüntü işleme günümüzde tarım alanlarında sıkça kullanılmaktadır. Seraların gözlemlenmesi, analiz edilmesi, yabancı otların kontrolü, bitki ve toprak izleme gibi konularda, özellikle sulama, gübreleme, ilaçlama gibi tarımsal girdilerde etkinliği arttırarak sürdürülebilir tarıma önemli katkılar sağlamaktadır. Bu alanlardaki başarılı uygulamalar, tarımın sürdürülebilirliği açısından önemli bir rol oynamaktadır (Çakmakçı ve Çakmakçı, 2023). Kullanılan bu teknolojiler sayesinde üretim kalitesi ve tarım verimliliği artmaktadır.

Bitki verimini artırmak için yaprak hastalıkları tahmini hemen yapılmalıdır. Bu nedenle görüntülerin toplanması, özellik çıkarımını yapılması ve gerçek zamanlı olarak sınıflandırılması gerekmektedir. Ürünlerin hastalık tespitinin hızlı ve otomatik yapılması ürün kalitesi açısından son derece önemlidir.

Tütün yaprakları, Türkiye için ekonomik ve sosyal açıdan büyük bir öneme sahiptir. Türkiye, dünyada tütün üretiminde önemli bir yere sahip olan önemli ülkelerden biridir. Tütün, tarım sektöründe önemli bir gelir kaynağıdır ve birçok çiftçi tarafından yetiştirilmektedir (Şahin ve Taşlıgil, 2013).

Günümüzde tarım sektörü, teknolojik gelişmeler ve yapay zekâ kullanımı nedeniyle büyük bir dönüşüm geçiriyor. Bu dönüşüm, üretim verimliliğinin artması, hastalıkların erken tespiti ve kaynakların daha verimli kullanılması gibi birçok faydayı beraberinde getirmektedir (Oktan ve Güzel, 2022).

Bu faydalardan biri de tütün yaprak hastalıklarının tespiti ve yönetimidir. Tarım sektöründe görüntü işleme ve yapay zekâ teknolojilerinin kullanılması hastalıkların erken teşhisine ve hızlı müdahale edilmesine olanak sağlamaktadır. Tütün yaprak görüntülerinin bu yöntemlerle analiz edilmesiyle hastalık tespiti gerçekleştirilecektir. Yapay zekâ algoritmaları, büyük veri setlerini hızlı bir şekilde analiz edebilir ve hastalık teşhisinde yüksek düzeyde sonuçlar ortaya koyabilmektedir.

Bu tez çalışmasında tütün yapraklarında meydana gelen hastalıkların tespiti için bir çalışma yapılmıştır. İlk olarak morfolojik görüntü işleme yöntemleriyle yaprak üzerindeki hastalıklı alanlar tespit edilmiştir. Yaprak üzerindeki hastalıklı alanların belirlenmesinden sonra hastalıklı bölgelerin sayılarının tespiti yapılmış ve hastalıklı yapraklar ayrılmıştır. Hastalıklı yaprakları gerçek zamanlı olarak tespit edebilmek için gömülü sistem tasarımı gerçekleştirilmiştir. Son olarak derin öğrenme yöntemleri kullanılarak hastalıklı ve sağlıklı yapraklar birbirinden ayrılarak sınıflandırma işlemi yapılmıştır.

1.1. Literatür Tarama

Soni ve arkadaşları yaptıkları çalışmada, ürünlerin hastalık tespitinin hızlı ve otomatik yapılmasının son derece önemli olduğunu vurgulamışlardır. Özellikle büyük arazilerde bu yaklaşım son derece faydalıdır. Önerilen modelde salatalık, guava, yer fıstığı, balkabağı ve börülcenin yaprak hastalıklarının otomatik ve verimli bir şekilde tanımlanması hedeflenmiştir. Tavsiye edilen bu yaklaşımda işlem adımları; ön işleme, eşikleme, Sobel filtresi, morfolojik genişletme ve ROI (Region of Interest) çıkarma olarak belirlenmiştir. MATLAB yazılımı tüm bu işlemleri gerçekleştirmektedir. Morfolojik görüntü işlemede resim öğelerini bölütlemek ve nesnenin şeklini korumak için genişletme (dilatasyon) yaklaşımı kullanılmıştır. Bu yöntem sonucunda %98,27 doğruluk oranına ulaşılmıştır (Soni vd., 2021).

Bükücü yaptığı çalışmada, cam yüzeylerde üretim esnasında meydana gelebilecek hasarları tespit etmek için gömülü sisteme dayalı, otomatik kontrol sistemi gerçekleştirmiştir. Çalışmada sağlam, batık hatalı ve çizik hatalı olmak üzere 3 sınıftan oluşan 90 adet resim kullanılmıştır. Sistemin gerçekleşmesi amacıyla kamera, mikro sürücü kartlarında oluşan bir bant sistemi kurulmuştur. Gömülü sistem, açık kaynak kodlu, morfolojik görüntü işleme teknikleriyle çalışan ve Gaussian metodu ile sınır tespiti yapabilen bir yazılım içermektedir.

Sistemin başarı oranı "Destek Vektör Makinası", "Kuadratik Diskriminant Analizi" ve "Orta Ölçekli Ağaç" sınıflandırıcılar kullanılarak ölçülmüştür ve %93.2 başarı oranı elde edilmiştir. Sistemin uygulanması cam fabrikasında test edilmiş ve yüksek bir başarı oranıyla hata tespiti gerçekleştirilmiştir. Bu çalışma, gömülü sistemlerin kalite kontrol süreçlerinde kullanımının avantajlarını göstermektedir. Bu tür sistemler, ekonomik, ergonomik ve işlem gücü yetenekleri nedeniyle yaygınlaşmaktadır ve üretim süreçlerinde önemli bir yer tutmaktadır (Bükücü, 2021).

Gupta ve arkadaşları kiraz yapraklarında görülen küllenme hastalığının tespitinde eşikleme tekniği ve morfolojik işlemler kullanılmasını önermiştir. İstenilen hastalığın ortaya çıkarılması için bir algoritma tasarlanmıştır. Arka planın görüntüden çıkarılması ve ardından istenilen hastalığın ortaya çıkarılması amaçlanmıştır. Performansın artırılması amacıyla gürültüyü gidermek için morfolojik işlemler kullanılmıştır. Önerilen bu yöntem otomatik ve karmaşıklığı azdır (Gupta vd., 2017).

Goyal, hindistanda yaptığı bir çalışmada morfolojik görüntü işlemenin ilkeleri ve uygulamaları hakkında derinlemesine bir inceleme gerçekleştirmiştir. Temel morfolojik oparetörlerden başlayarak pratik ve yararı kanıtlanmış bilgiler vermiştir. Morfolojik işlemler genel olarak segmentasyon sırasında meydana gelen kusurları gidermek için kullanılır. Morfolojik işlemenin amacı görüntü üzerindeki verileri basitleştirmek, temel şekillerini korumak ve gürültüyü ortadan kaldırmaktır. Bu işlemi gerçekleştirirken matematiksel morfolojinin dili olarak kabul edilen küme teorisi kullanılır (Goyal, 2011).

Shao ve arkadaşları, tütün yaprak hastalıklarının tanıma doğruluğunu artırmak amacıyla BP(Backpropagation) sinir ağını optimize eden genetik algoritmaya dayalı, çok özellikli bir otomatik tanımlama sistemi geliştirmişlerdir. İlk olarak hastalık konum bilgisini tespit etmek amacıyla Nobuyuki Otsu tarafından geliştirilen otomatik eşikleme yapan otsu metodu kullanılmıştır. Daha sonra hastalıklı alanı çıkarmak için GrabCut işlevi başlatılmıştır. Renk, dış hat ve doku özelliklerini toplamak için çok katmanlı kontur, renk momentleri ve gri seviye eş-oluşum matrisi kullanılmıştır. Tütün yaprak hastalıkları olan, kahverengi leke, hava lekesi, orman yangını hastalıkları üzerinde yapılan bu çalışmada başarı oranı ortalama %92.5 olarak ölçülmüştür. Bu tanımlama sistemi sayesinde tütün hastalıklarının teşhis edilebileceği ve daha sonra bu sonuçlara göre üretim için gerekli yönlendirmelerin yapılacağı bildirilmiştir (Shao vd., 2017).

Karhan ve arkadaşları, gerçekleştirdikleri çalışmada kayısı meyvesinde yaprak delen(çil) hastalığının bir sonucu olarak ortaya çıkan leke hastalığını morfolojik görüntü işlemleri kullanarak tespit etmişlerdir. Kırmızı, Yeşil, Mavi kanallardan oluşan görüntüler tek kanallı gri tonlu görüntülere göre daha fazla anlamsal bilgi içermektedirler. Renkli görüntülerde morfolojik olarak çalışabilmek için RGB kanallara ayrılmıştır. Kayısı üzerinde meydana gelen lekeli alanı tespit etmek amacıyla renkli görüntülerin piksel sayılarına bakılarak histogram analizi gerçekleştirilmiş, morfolojik işlemlerden geçirilmiş, uygun eşik değerine göre ikili görüntüye çevrilmiştir. Kayısı üzerindeki sağlam kısımlar ile hastalıklı kısımlar(çilli) ayrılmıştır. Çalışma sonucunda elde edilen verilerin doğruluk oranı başarılıdır. Kayısı üzerindeki lekeler morfolojik yöntemler kullanılarak karmaşık yöntemlere ihtiyaç duymadan başarılı bir şekilde yapılabileceği belirtilmiştir (Karhan vd., 2011).

Gögçe ve arkadaşları, yaptıkları çalışmada, 35 adet hastalıklı yaprak üzerinde çeşitli hastalıkların oluşturduğu lekelerinin tespiti için görüntü işleme yöntemleri kullanılarak bir algoritma oluşturmuşlardır. Geliştirilen algoritma, renkli görüntünün 3 kanalı ile gri görüntünün tek kanalından elde edilen standart sapma görüntüsü ile oluşturulmuştur. Bu yöntem sayesinde oluşturulan görüntülerdeki hastalıklı alanlar, önerilen algoritma kullanılarak başarı ile tespit edilmiştir. Farklı özellikteki görüntüler kullanılmış olmasına rağmen, geliştirilen algoritma sayesinde yaprak üzerindeki farklı renk, desen ve yapıdaki birçok lekenin tespiti mümkün olmaktadır. Bu çalışmanın literatürdeki diğer çalışmalardan farkı, leke ayrımı yapmaksızın geniş bir leke yelpazesini başarılı bir şekilde tespit edebilmesidir (Gögçe vd., 2021).

Lin ve arkadaşları yaptıkları çalışmada, bitki hastalıklarının sürekli olarak morfolojik ve fiziksel değişimlere uğramasından dolayı bu hastalık görüntülerinin sınıflandırmasının zor olduğunu belirtmişlerdir. Bu nedenle, CNN modeli tarım alanlarında düşük doğruluğa sahiptir. Çin’de gerçekleştirilen çalışmada, tarla koşullarında tütün hastalıklarının tanımı için CAMFFNet adında bir CNN modeli önerilmiştir. Çoklu özellik füzyonu (MFF), özellik çıkarımlarında bilgi kaybını azaltmak için kullanılmıştır. Ayrıca arka plan etkisini azaltmak amacıyla koordinat dikkat (CA) modülü kullanılmıştır. Elde edilen bulgular sonucunda CAMFFNet modelinin tütün hastalıkları doğruluk derecesi %89,71 olarak ölçülmüştür.

Klasik CNN modellerine göre doğruluk oranları daha iyi olduğundan, bitki hastalıkları tespitinde daha güvenilir bir model olduğu belirtilmiştir (Lin vd., 2022).

Sindhu yaptığı çalışmada, bitki verimini artırmak için yaprak hastalıkları tahminin hemen yapılmasının öneminden bahsetmiş ve bu nedenle görüntülerin toplanması, özellik çıkarımını yapılması ve gerçek zamanlı olarak sınıflandırılması amacıyla bir model oluşturmuştur. Bu sınıflandırmalar yaprağın hasta olup olmadığını göstermektedir. Sınıflandırma yapmak amacıyla DBN(Deep Belief Network) kullanılmıştır. Bu yöntem diğer CNN, RNN vb. modellere göre daha performans sağladığı görülmüştür (Sindhu, 2020).

Fan ve arkadaşları derin öğrenme yöntemi kullanılarak yaptıkları çalışmada İHA tarafından çekilen tarla resimlerinden tütün bitkisi tespitinin yapılması amacıyla yeni bir algoritma önermişlerdir. İlk olarak elde edilen görüntüler morfolojik işleminden geçirilmektedir. Daha sonra tütün bitkisi içeren ve içermeyen görüntülerden oluşan veri seti sınıflandırma yapılmak amacıyla evrimsel sinir ağı(CNN) oluşturulur ve eğitilir. Son olarak tütün dışındaki bitkileri uzaklaştırmak amacıyla son işlem uygulanır. Bu algoritma sonucunda tütün tespiti için iyi bir performansa ulaşılmış ve sırasıyla ortalama 0,9370, 0,9126 ve %3,94 doğruluk ve hata değerlerine ulaşmıştır (Fan vd., 2018).

Xu ve arkadaşları, yaptıkları çalışmada tütün yaprak hastalıklarını tanıma ve teşhisindeki düşük verimlilik oranlarını iyileştirmek amacıyla ORB algoritmasına dayalı yeni bir hastalık tanıma yöntemi geliştirilmişlerdir. Geliştirilen FAST14-24 algoritması ORB algoritmasındaki düşük verimliliğe sahip olan görüntü kenarlarını algılama oranını artırmıştır. Bu yöntemin geliştirilmesi sırasında morfolojik özellikler, renk özellikleri, doku ve tütün hastalığı lekelerinin çıkartılmasıyla toplam 28 adet parametre ortaya çıkmıştır. FAST14-24 algoritması sonucunda tütün kurbağa gözü ve kahverengi leke hastalığının erken aşamadaki tanımlanmasında %92 başarı oranı, orta ve geç aşamasındaki tanıma oranı %96 olarak ölçülmüştür (Xu vd., 2022).

Ulutürk ve Uğur yaptıkları çalışmada, bitkilerin sınıflandırılmasında geleneksel yöntemler yerine otomatik bitki tanıma sistemi tasarımı hedeflemişlerdir. Bunu gerçekleştirmek amacıyla yapraklar 2'ye bölünerek bir sistem tasarımı gerçekleştirilmiştir. Bitki yaprakları ön işleme tabi olduktan sonra 7 tane morfolojik özellik çıkarılmıştır. Yarım yaprak görüntüleri kullanılarak morfolojik özelliklere sahip 3 ek özellik daha çıkarılmıştır. Bu

özelliklerden yararlanılarak yapraklar ana eksenine göre yönlendirilerek alan, kapsam ve dış merkezlik özellikleri çıkarılmıştır. Bu özellikler PNN(olasılıksal sinir ağına) girdisi sağlanmış ve yaprak tanımda %92,5 tanıma doğruluğuna ulaşılmıştır (Ulutürk ve Uğur, 2012).

Atalı ve arkadaşları, görüntüler üzerinde meydana gelen tahribatın giderilmesi amacıyla yapay zekâ kullanarak bu sorunu çözmeyi hedeflemişlerdir. Yapılan çalışmada, yapay sinir ağları kullanılarak görüntüdeki tahribatın derecesi morfolojik operatörler kullanılarak incelenmiştir. Yapay sinir ağı üzerinde eğitilen görüntüler, dört farklı düzeyde dairesel ortalama filtresinin uygulanması sonucu meydana gelen tahribatları temsil etmektedir. Bu değerlere dayanarak, yapay sinir ağı üzerinde 2. ve 3. derecedeki dairesel ortalama filtresi kullanılarak oluşturulan morfolojik görüntüde %75 oranında bir başarı elde edilmiştir. Ancak, 4. derece aşınma uygulanan görüntüde orijinalden sapmalar nedeniyle sonuçlar daha olumsuz olarak tespit edilmiştir (Atalı vd., 2016).

Laily yaptığı çalışmada, karanfil yaprakları ve tütün yaprak hastalıkları tespitinde Perceptron(Tek Katmanlı Öğrenme) yöntemi algoritması kullanarak yapay sinir ağı yöntemi uygulamıştır. Tütün yaprak hastalıklarından kahverengimsi kırmızı leke, beneklenme, siyah noktalar olmak üzere 8 tanesi analiz edilmiştir. İlk olarak hastalıktan etkilenmiş ve etkilenmemiş yapraklar toplanmıştır. Daha sonra hastalık verisi taşıyan yapraklar sınıflandırılmıştır. Yapraklarda görülen hastalıklar iki kutuplu değerlerle temsil edilerek yapay sinir ağına eğitim ve test verisi olarak kullanılmıştır. Eğitim aşamasında 20 yaprak, test aşamasında 10 yaprak kullanılarak gerçekleştirilen bu yöntemde %66 oranında başarı sağlanmıştır (Laily, 2013).

Avila-George ve arkadaşları tütün yapraklarında ortaya çıkan maviküf hastalığının yapraklara verdiği hasarı tespit etmek için android temelli bir yazılım geliştirmişlerdir. Tütün yapraklarının görüntülenmesinde yapay sinir ağı olarak bilinen patern tanıma tekniği kullanılmıştır. Gerçekleştirilen eğitim ve test işlemlerinde 40 adet fotoğraf kullanılmış ve %97,2 oranında başarı sağlanmıştır. YSA modelini gerçekleştirmek amacıyla ön işleme, özellik çıkarma ve sınıflandırma işlemleri yapılmıştır (Avila-George vd., 2018).

Sun ve arkadaşları, tütün hastalıklarının tanınması amacıyla yeni bir görüntü sınıflandırma yöntemi önermişlerdir. Çoklu dikkat sınıflandırma ağı(MACN) olarak isimlendirilen bu

yöntem sınıflandırma ağını 2 açıdan geliştirmektedir. İlk olarak transfer öğrenmeye dayalı düşük seviyeli özelliklerin elde dilmesidir. 2.olarak da düşük seviyeli özellikleri modelleyerek yüksek seviyeli özelliklerin çıkarılması olarak bildirilmiştir. Tütün kahverengi lekesi, tütün vahşi ateş, maviküf ve sağlam yapraklardan elde edilen veri seti kullanılmıştır. Sınıflar arasındaki farkları öğrenmek amacıyla yeni bir çoklu dikkat modülü(MAM) tasarlanmıştır. Veri kümesi üzerinde eğitilen transfer öğrenme modellerinden VGG16, ResNet50 ve InceptionV3 arasında yapılan karşılaştırmalar sonucunda ilk özellik çıkarımı için transfer öğrenme yöntemi InceptionV3 modelinin en yüksek doğruluğa sahip olduğu belirlenmiştir. Transfer öğrenme modelleri ve önerilen MACN modeli arasında yapılan karşılaştırmada en yüksek doğruluk oranının %85,56 ile MACN'ye aittir. Bu oran diğer mevcut yöntemlere göre önemli ölçüde iyi olduğunu kanıtlamaktadır (Sun vd., 2020).

Aptoula ve Yanıkoğlu, imageClef12 bitki tanıma veri setini kullanılarak yapraklardan bitki tanımlanmasını hedeflemişlerdir. Önerilen bitki tanıma yönteminde matematiksel morfolojiye dayalı 2 tanımlayıcı önerilmektedir. İlk olarak yaprak kenarlarındaki periyodik desenleri yakalayabilmek için morfolojik kovaryans hesaplamasından oluşmaktadır. İkinci olarak yaprağın anatomik yapısında oluşan damarlanma özelliklerini yakalayan sirküler kovaryans histogramının kullanılmasıdır. Yapılan bu çalışma sonucunda var olan operatörlerden daha iyi sonuç verdiği hesaplanmıştır (Aptoula ve Yanikoglu, 2013).

Huawei bulut geliştirici birliği tarafından yapılan çalışmada, gerçek zamanlı görüntülerden nesneleri takip etmek için kullanılan YOLOv5 ile tütün yapraklarındaki hastalıkların tespiti için bir sistem tasarlanmıştır. Tütün veri setinin elde edilmesi ve etiketlenmesinden sonra tütün yaprağı algılama ve eğitim adımları uygulanmıştır. Hastalık tespit sitemine entegre edilen yolov5 ile hastalığı türü, yeri ve miktarı yapraklarda belirgin bir şekilde işaretlenmiştir. Bu sistem yüksek doğruluk ve yüksek tanıma oranına sahiptir (Geliştirilmiş YOLOv5'i Temel Alan Python Tütün Yaprağı Hastalığı Tespit Sistemi, b.t.).

Chowdhury ve arkadaşları, domates yapraklarında meydana gelen hastalıkları sınıflandırmak amacıyla 18.161 domates görüntüsü üzerinde EfficientNet derin öğrenme mimarisi kullanarak çalışmışlardır. Çalışmada yaprakları farklı bölgelere ayırmak için U-net ve kişiselleştirilmiş U-net'in performansı belirlenmiştir. Modeller ikili sınıflandırma, altılı sınıflandırma ve onlu sınıflandırmalar ile karşılaştırmalı olarak rapor edilmiştir. Modellerin sırası ile doğruluk oranları %98,66, %98,5 ve %98,73 olarak ölçülmüştür. EfficientNet-B7

ile yapılan ikili ve altılı sınıflandırmada %99,95 ve %99,12 doğruluk oranlarına ulaşılmıştır. Sonuç olarak Modifiye edilmiş U-net'in arka plan segmentasyonu için en uygun olduğu, EfficientNet-B7 ise diğer mimarilere göre ayırt edici özelliğinin daha iyi olduğu gözlemlenmiştir (Chowdhury vd., 2021).

Swasono ve arkadaşları, zararlı saldırılarından hasar görmüş tütün yapraklarının otomatik olarak sınıflandırılması üzerine çalışmışlardır. Tütün yaprakları hasat sonrası döneme ait kurutulmuş kahverengi formdadır. Gerçekleştirilen bu çalışmada Evrişimsel sinir ağı(CNN) yöntemlerinden VGG16 mimarisi kullanılmıştır. VGG16 eğitiminin doğruluğunu artırmak ve hızlandırmak amacıyla ilk olarak transfer öğrenme gerçekleştirilmiştir. Transfer öğrenme ve VGG16 kullanılarak yapılan çalışmada 3 sınıf hastalık üzerinde çalışılmış ve eğitim sonunda çok yüksek oranda doğruluk tespit edilmiştir (Swasono vd., 2019).

Dhaya yaptığı çalışmada, Fusarium oxysporum hastalığının bitkilerin yapraklarında sararmaya, solgunluğa, büyüme geriliğine ve meyve ve sebzelerde çürümeye neden olabileceğini belirtmiştir. Bu hastalığın genellikle domates, tütün ve patates gibi birçok ürünü etkilediğini vurgulamıştır. Çalışmada, domates yapraklarındaki fusarium oxysporum (FO) hastalığının tespit edilmesi hedeflenmiştir. Önerilen algoritmada doğruluğu artırmak amacıyla yaprak üzerindeki hastalıklar 2 defa sınıflandırılarak tanımlanmıştır. 87 bin görüntünün kullanıldığı bu çalışmada bu görüntülerin %60'ı hastalıklı yaprak, %40'ı sağlam yapraklardır. Önerilen bu çalışma sonucunda hastalıklı yaprakları %96 oranında bulmuştur (Dhaya, 2020).

Türkoğlu, yaptığı araştırmada bitki türlerinin tanımlanması, yeni türlerin tespiti ve bitki hastalıklarının tespiti konularında yapay zekâ ve görüntü işleme tabanlı sistemlerin kullanıldığını belirtmiştir. Bitki üzerinde meydana gelen hastalıkların tespit edilmesinde var olan yöntemler geliştirilmiştir. Ayrıca hastalıklarının tespiti için kenar adım yöntemi ve deri öğrenme modeli kullanılmıştır. Geliştirilen bu modeller 15 adet veri seti kullanılarak değerlendirilme yapılmıştır. Bu çalışmanın sonucunda bitki türlerini bulma ve hastalıkların sınıflandırması konusunda yüksek başarı sağlanmıştır (Türkoğlu, 2019).

Benli, yaptığı çalışmada açık kaynaklı bir veri seti kullanılarak oluşturulan "New Plant Disease" veri seti üzerinde altı farklı veri genişletme tekniği ile yaklaşık 83.107 görüntü kullanmıştır. Bu görüntüler üzerinde 6 katmanlı evrişimsel sinir ağı kullanılarak

sınıflandırma mimarisi geliştirilmiştir. Model üzerinde ilk olarak dört farklı bitki yaprağı çeşidindeki en yüksek veri sayısına sahip görüntüler için çalışma yapılmıştır. Daha sonra New Plant Disease veri setindeki tüm veri seti incelenmiştir. Elde edilen sonuçlar, %92 oranında başarı sağlandığını göstermiştir. Bu çalışma kapsamında, performansın değerlendirilmesinde için transfer öğrenme mimarileri olan Google Net, ResNet50, Vgg-19, Vgg-16, DenseNet mimarileri kullanılmıştır. Bu değerlendirme sonucunda, en yüksek başarı oranına %90 ile Vgg mimarilerinden özellikle Vgg-19 mimarisi ulaşmıştır (Benli, 2021).



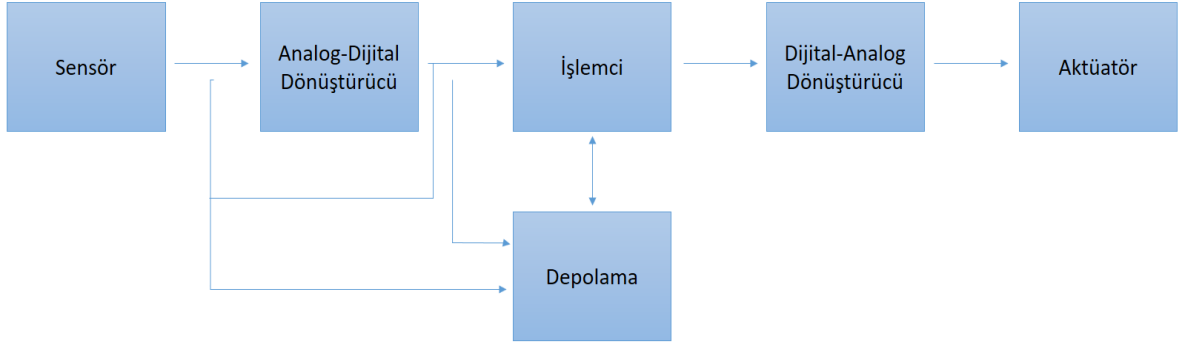
2. DONANIM

Bir projeyi oluşturan elemanlar için donanım, projenin gerçekleştirilmesi ve çalıştırılması için gereken tüm fiziksel bileşenleri ifade eder. Bu, bilgisayarlar ve işlemcilerden depolama aygıtlarına, giriş/çıkış cihazlarından sensörler, gömülü sistemler ve iletişim modüllerine kadar geniş bir alanı kapsar. Donanım bileşenleri, yazılımın çalışmasını ve projenin belirlenen hedeflere ulaşmasını sağlamak için çok önemlidir.

2.1. Gömülü Sistemler

Gömülü sistemler, belirli görevleri yerine getirmek üzere tasarlanmış, bilgisayar sistemlerine veya elektronik cihazlara gömülü sistemler denir (Türk ve Lüy, 2021). Gömülü sistemler, kendilerine özgü yazılımı olan mikro denetleyici tabanlı bilgisayar donanım sistemleri olarak da tanımlanabilir. Bu sistemler, tek başına bağımsız olarak çalışabilen veya daha büyük bir sistemin içinde kullanılabilen cihazlardır. Gömülü sistemler, insan vücudu gibi düşünülebilir; çünkü insan beyni vücut içinde belirli işlevleri yerine getirirken, gömülü sistemler de mikro denetleyici veya mikro işlemci aracılığıyla belirli görevleri yerine getirirler. İnsan beyninin vücutta işlev görmesi gibi, gömülü sistemler de belirli görevleri yerine getirmek için optimize edilmiş donanım ve yazılım kombinasyonuna sahiptirler.

Gömülü sistemler, bilgi işlem sistemleri olmalarına rağmen, kullanıcı ara yüzleri açısından oldukça çeşitli olabilir. Bu cihazlar, tek bir görevi yerine getirmek amacıyla tasarlanmış cihazlarda kullanıcı ara yüzünden yoksun olabilirken, mobil cihazlardaki gibi karmaşık grafik kullanıcı ara yüzlerine sahip olabilirler. Kullanıcı ara yüzleri, butonlar, ledler, dokunmatik ekran algılama gibi özellikler içerebilir. Bazı sistemler ise uzaktan kullanıcı ara yüzlerini kullanır. Bu çeşitlilik, gömülü sistemlerin kullanım amacına, tasarım gereksinimlerine ve kullanıcı ihtiyaçlarına bağlıdır. Gömülü sistemin temel yapısını oluşturan blok diyagram Şekil 2.1’de verilmiştir.



Şekil 2. 1. Gömülü sistem tasarımı

Gömülü Sistem Tasarımını oluşturan;

- Sensörler, fiziksel miktarları ölçer ve bu ölçümleri elektronik bir cihaz tarafından okunabilir bir elektrik sinyaline dönüştürerek kaydeder. Bu ölçümler daha sonra işlenmek üzere kullanılır.
- Analogdan dijitale dönüştürücüler (A&D dönüştürücüler), sensörlerin gönderdiği analog sinyalleri dijital sinyallere çevirirler. Bu, bilgisayarlar gibi dijital işlemcilerin okuyabileceği verileri sağlar.
- İşlemciler ve özel işlemciler (ASIC'ler) dijital verileri işlerler. Ölçülen değerleri işler, analiz eder ve bellekte saklarlar.
- Dijitalden analoğa dönüştürücüler (DA dönüştürücüler), işlenen dijital verileri analog sinyallere dönüştürürler. Bu, çıkış sinyallerini etkileyen fiziksel süreçlere geri dönüş sağlar.
- Aktüatörler, DA dönüştürücüler tarafından verilen çıktıları alır, bu çıktıları beklenen gerçek çıktılarıyla karşılaştırır ve onaylanmış çıktıları saklar. Bu, bir sistemde kontrol edilen işlemleri gerçekleştirmek için kullanılır.

Gömülü sistemler genellikle düşük güç tüketimi, hızlı tepki ve güvenilirlik gibi özelliklere sahip olacak şekilde tasarlanır (Lutkevich, 2020). Ayrıca sınırlı kaynaklar nedeniyle verimli çalışacak şekilde optimize edilirler. Gömülü sistemler çeşitli sektörlerde kullanılır ve günlük hayatımızın birçok alanında bulunur. Ev aletleri, tıbbi cihazlar, endüstriyel otomasyon, akıllı

ev teknolojisi, telekomünikasyon, uzaktan kumanda sistemleri, otomotiv endüstrisi, robotik ve diğer birçok uygulama alanında yaygın olarak kullanılmaktadır.

Gömülü sistemler, gerçek zamanlı işletim ortamlarında çalışır ve donanım ile iletişim kurmak için gerçek zamanlı işletim sistemi kullanır (Mellit vd., 2023). Bu sistemler, verilen görevleri yerine getirmek amacıyla yeterli hızda tasarlanır ve görev sırasında meydana gelen küçük sapmalardan etkilenmezler. Bu tür sistemlerde, sıklıkla sadeleştirilmiş linux işletim sistemi sürümleri kullanılır. Bununla birlikte, gömülü sistemler için özelleştirilmiş işletim sistemleri de bulunur.

Son yıllarda çeşitli alanlarda kullanılabilen birçok gömülü sistem kartı üretilmiştir. Bunların en çok kullanılanları, arduino, nvidia jetson nano, asus tinkercard, beaglebone black orange pi, odroid xu4, raspberry pi modellerinden oluşmaktadır. Bu tez çalışmasında en çok kullanılan model olan raspberry pi modeli ile gömülü sistem tasarımı gerçekleştirilmiştir.

2.1.1. Raspberry pi 3 model b+

Cambridge Üniversitesi Raspberry Pi Vakfı'ndan Eben Upton bilgisayar bilimleri eğitime katkı sağlamak amacıyla özellikle eğitim kurumlarında kullanılabilen, düşük maliyetli, ekonomik ve kredi kartı boyutlarında bir bilgisayar üretimine başlamıştır (Mathe vd., 2022). Raspberry Pi piyasaya 2012 yılında sürülmüş ve düşük fiyatı, taşınabilirliği, kullanım kolaylığı ve özellikleri nedeniyle büyük talep görmüştür (Jolles, 2021). Raspberry Pi sadece mini bilgisayar olarak değil aynı zamanda mikro denetleyici olarak da kullanılabilmektedir. Çeşitli alanlarda gerçek dünyayı algılamak ve kontrol etmek için geniş bir uygulama yelpazesine sahiptir.

Raspberry Pi'nin single board computer (sbc) yani tek kart bilgisayar sistemini kullanmaktadır. "Tek Kart Bilgisayarlar" veya İngilizce adıyla "Single Board Computer (SBC)", mikroişlemci, GPIO (genel amaçlı giriş ve çıkış) pinleri, RAM ve bir bilgisayarın gereksinim duyduğu diğer bileşenleri tek bir devre kartı (PCB) üzerinde barındıran bilgisayarlardır (Boyacıgil ve Uzun, 2023). Masaüstü bilgisayarlardan farklı olarak, SBC'ler daha düşük güç tüketimi sağlar ve daha küçük boyutludur. Genellikle ARM mimarisine dayanırken, bazı modellerde x86 işlemci mimarisi de bulunmaktadır. SBC'lerin donanımsal olarak değiştirilemez olması, kullanıcılara daha fazla güvenlik sağlamaktadır. Raspberry Pi,

üzerindeki 40 adet pini kullanarak çeşitli kontrol uygulamalarını beraber çalıştırabilmektedir. Bluetooth ve Wi-Fi yetenekleri sayesinde iletişim hızlı, etkili ve kolaydır. Ekran ve kamera modülü ve usb portları sayesinde sistemi kodlamak ve kontrol etmek kolaydır. Raspberry Pi kartı sıcaklık ve nem kontrolü, motor kontrolü ve ev otomasyonda kullanıldığı gibi, görüntü işleme, yapay zekâ, makine öğrenmesi gibi alanlarda da sıkça kullanılmaktadır.

Raspberry Pi, farklı amaçlar için çok yönlü bir şekilde kullanılabilir:

- USB fare, klavye ve HDMI bağlantısıyla monitöre bağlanarak raspbian desktop imajı masaüstü bilgisayar olarak kullanılabilir.
- Raspbian işletim sistemi kurulduktan sonra ethernet veya wlan bağlantısı ile web sunucusu indirilip web sunucusu olarak kullanılabilir.
- Raspberry Pi, dosya sunucusu olarak kullanılabilir ve depolama cihazlarıyla birleştirilebilir.
- Owncloud gibi bulut tabanlı sistemlerle Raspberry Pi, bir bulut dosya sunucusuna dönüştürülebilir.
- Raspberry Pi, kendi kamerası veya USB web kamerası ile ağ kamerası veya kayıt sistemi olarak kullanılabilir.
- Raspberry Pi'ye hoparlör ve mikrofon eklenebilir, bu sayede google assistant veya amazon gibi uygulamalarla beraber kullanılabilir.
- Raspberry Pi, farklı sistemleri entegre eden bir IoT servis hub olarak konumlandırılabilir. OpenHAB gibi sistemlerle çeşitli teknolojiler entegre edilebilir ve özelleştirilmiş komutlar belirlenebilir.
- Raspberry Pi kamera portuna veya usb portuna kamera takılarak görüntü işleme cihazı olarak kullanılabilir.
- Raspberry Pi gerçek zamanlı çalışan uygulamalarda kullanılabilir.

Gerçekleştirilen bu tez çalışmasında gömülü sistem tasarımı için Raspberry Pi 3 Model B+ kullanılmıştır. Raspberry Pi model ayrıca açık kaynak işletim sistemlerini sıkça kullanmaktadır. Çalışmada kullandığımız Raspberry Pi 3 Model B+ kartına linux çekirdeği kullanan raspberry pi os işletim sistemi kurularak kodlama gerçekleştirilmiştir. Raspberry pi 3 model b+ kart görünümü Resim 2.1’de, pin ve port bağlantıları Şekil 2.2’de gösterilmiştir.

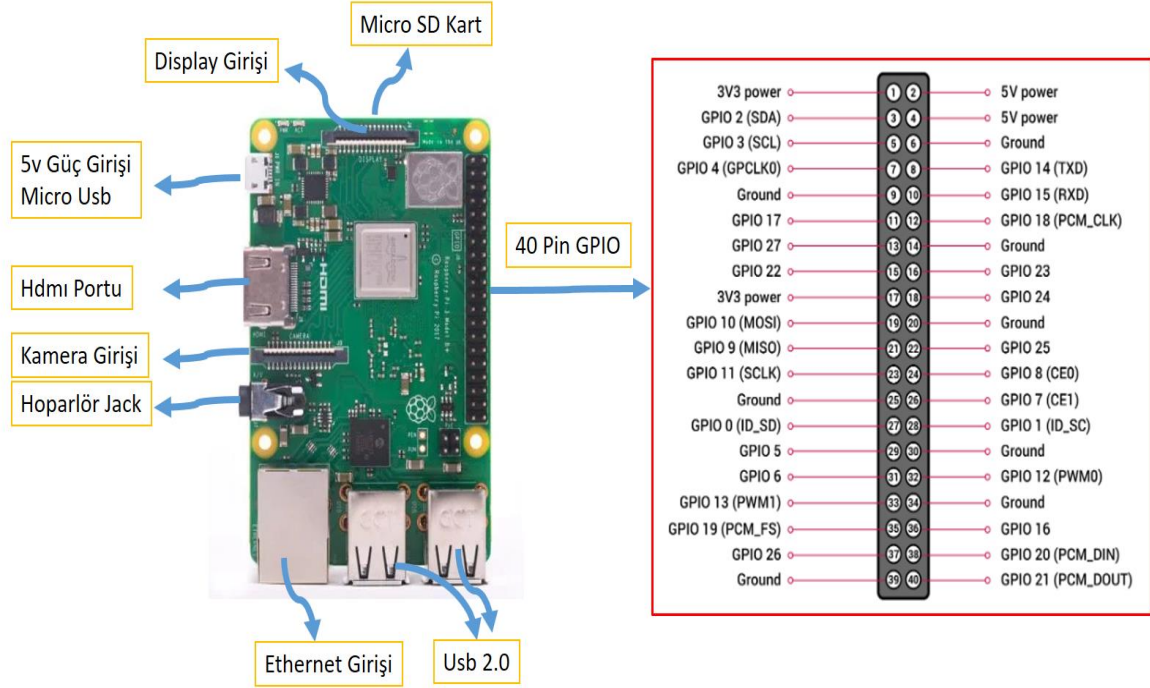
Raspberry Pi 3 Model B+ teknik özellikleri:

- İşlemci: Broadcom BCM-2837B0, Cortex-A53/AMRv8, Quad-Core 64-bit SoC @ 1.4GHz
- Ram: 1GB LPDDR2 SDRAM
- Bağlantı:
 - 2.4GHz ve 5GHz IEEE 802.11.b/g/n/ac kablosuz yerel ağ bağlantısı
 - Bluetooth 4.2, BLE
 - Gigabit Ethernet, USB 2.0 üzerinden (Maksimum 300Mbps)
 - 4 × USB 2.0 Port
- Gpio: Genişletilmiş 40 adet pin
- Video ve Ses:
 - 1 × HDMI
 - MIPI DSI ekran portu
 - MIPI CSI kamera portu
 - 4 Kutuplu 3.5mm Ses + kompozit video portu
- Multimedya:
 - H.264, MPEG-4 decode (1080p@30)
 - H.264 encode (1080p@30)
 - OpenGL ES 1.1, 2.0 Grafik
- Hafıza: Micro SD Kart Girişi, işletim sistemi ve veri depolama için kullanılır
- Güç Beslemesi:

- 5V/2.5A DC, Mikro USB port üzerinden
- 5V DC, GPIO pinler üzerinden
- Ethernet Üzerinden Güç (Power over Ethernet, PoE) (ayrı poe hat gerekmektedir)
- Çevre Şartları: Çalışma sıcaklığı, 0–50°C
- Üretim Desteği: Ocak 2023'e kadar üretim desteği



Resim 2. 1. Raspberry pi 3 model b+



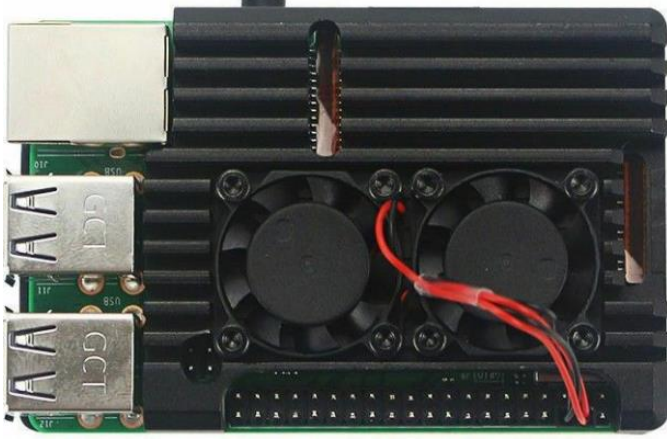
Şekil 2. 2. Raspberry pi 3 model b+ pin bağlantıları

Raspberry pi kartlarının avantajları:

- Kendi işletim sistemine sahip olması
- USB çıkış
- Düşük maliyet
- Genişletilebilir hafıza
- Bluetooth, Wi-Fi bağlantı yapılması
- HDMI (high definition multimedia interface) çıkış
- VGA (video graphics array) çıkış
- Merkezi işlem biriminin olması
- Taşınabilmesinin kolay olması
- Açık kaynaklı yazılımları desteklemesi
- Mouse ve klavye bağlantısı
- Farklı elektronik kartlar ile birlikte çalışabilme
- Dokunmatik ekran bağlanabilmesi
- Düşük voltajla çalışabilmesi
- Kurulumun ve kullanımının basit olması

Raspberry pi kartlarının dezavantajları

- Raspberry kartlarının en büyük dezavantajlarından bir tanesi ısınma problemidir. Üzerinde dâhili bir fan olmayan bu karta daha sonradan soğutucu özellikler eklenebilmektedir. Resim 2.2’de soğutucu uygulaması görülmektedir.
- Yoğun programlarda performans düşüklüğü
- HD videolarda performans düşüklüğü
- Diğer kartlara göre daha pahalı olması
- Muhafazasız olması ve daha sonradan takılan harici bir kutuya ihtiyaç duyar. Resim 2.3’te muhafaza kutusu gösterilmektedir.



Resim 2. 2. Raspberry pi kartına alüminyum ve fan soğutucu uygulaması



Resim 2. 3. Raspberry pi 3 muhafaza kutusu

2.2. Raspberry Pi Uyumlu Kamera

Raspberry Pi geliştirme kartında tümleşik olarak yer alan video portuna takılabilen bir kamera modülü bulunmaktadır. Bu kamera modülü ile birçok uygulama yapılabilmektedir. Ancak Bu çalışmadan görüntü kalitesi önemli olduğundan bu kameranın yerine Raspberry pi'ye usb portu üzerinden takılan Resim 2.4'te gösterilen Kaan hwk-w kamera modülü kullanılmıştır. Kameranin özellikleri Çizelge 2.1'de verilmiştir.



Resim 2. 4. Kaan hwk-w kamera modülü

Çizelge 2. 1. Kaan hwk-w kamera özellikleri

Görüntü İşleme Bölümü Özellikleri	
Sensör Tipi/Boyutu	CMOS 1 / 2.9"
Sensör Çözünürlüğü	2MP
Sinyal/Gürültü Oranı	>50dB
Görüntü Formatı	FHD 1920x1080 (1080P) HD 1280x720 (720P) 640x360
Çerçeve Hızı	30fps/25fps
Kodlama/Sıkıştırma	H.264, MJPEG
Sayısal Perde Hızı (Enstantane)	Otomatik
Çalışma Gerilimi/Akımı	DC5V / 200mA

Diğer Özellikler	
Çalışma Sıcaklığı	-20 / +60°C
Mekanik Bağlantı	1/4" Monopod/Tripod Bağlantı Yuvası
Kablo Boyu	≤ 5 mt

Lens Özellikleri	
Lens Dış Çapı	D14mm
Lens Bağlantı Çapı	M12 x P0.5
Etkin Görüntü Çapı	Ø 1 / 2.7" (6.9mm)
F.No	2.5 ± %5
Odak Uzunluğu	3.92mm
Görüş Açısı	107° (D) x 89° (H) x 48° (V) ±5
Odak Uzaklığı (Sabit)	2 mt
Lens Katman Yapısı	1G + 3P

Yazılım Özellikleri	
İşlem Sistemi Uyumluluğu	MS Windows 7, 8, 10 Linux MAC OS
Uygulama Uyumluluğu	Zoom MS Teams Google DUO Skype

3. YAZILIM

Yazılım, bilgisayar sistemlerinde veya elektronik cihazlarda belirli görevleri yerine getirmek üzere tasarlanmış programlar ve veri kümeleridir. Bu programlar, donanım üzerinde çalışarak kullanıcıların ihtiyaçlarına cevap verir, işlevler gerçekleştirir ve veri işleme süreçlerini yönetir. Bu tez çalışmasında gerçekleştirilen görüntü işleme yöntemleri, kullanılan kütüphaneler ve gömülü sistem işletim sistemleri bu bölümde incelenmiştir.

3.1. Görüntü İşleme

Görüntü işleme, görüntülerin elde edilmesiyle başlar. Satır ve sütunlar halinde düzenlenmiş diziler ya da matrislerdir. Görüntüleme cihazı, her piksel için belirli özellikleri tanımlayan bir sayıyı veya bu özellikleri küçük bir sayı kümesini kaydeder. Bu sayılar, görüntüdeki piksellerin dikey ve yatay konumlarına karşılık gelen bir dizi satır ve sütun halinde düzenler.

İki boyutlu veya üç boyutlu bir sahnenin mekânsal temsiline görüntü denir. Sensörlerden gelen görüntünün bilgisayara aktarılmasıyla dijital görüntü oluşmaktadır (Peker, 2009). Dijital görüntüyü elde edilmesi görüntü kaynağına ve yazılıma göre farklılık göstermektedir. Her bir piksel değeri, parlaklığı, rengi, yoğunluğu veya görüntü işleme yoluyla elde edilen sınıflandırılmış bir değeri temsil eden bilgileri içerir. Bu piksellerden binlercesi bir araya gelerek Resim 3.1.'deki görüntüyü oluşturur.



Resim 3. 1. Piksellerden görüntü oluşturulması

Görüntü çeşitleri:

Binary görüntü; her piksel için yalnızca iki olası değer alır, 0 değerinde siyah ve 1 veya 255 değerinde beyaz olmaktadır. Bu tür görüntüler genellikle kenar tespiti veya basit nesne algılama gibi uygulamalarda kullanılır.

Gri renk seviyeli görüntü; her pikseldeki ışık yoğunluğu bilgisi içermektedir. Tek renkli görüntüler olarak ifade edilir. Görüntü işleme algoritmalarında sıklıkla kullanılır ve gri tonlamalı resimlerin işlenmesi veya analizi için idealdir.

Renkli görüntü; renkli görüntü birden fazla renk kanalı içeren bir görüntüdür. Bu kanallar genellikle kırmızı, yeşil ve mavi renklerin bir araya gelmesinden oluşur. Renkli görüntü işleme süreci, bu renkli görüntüler üzerinde çeşitli eylemlerin gerçekleştirilmesini içerir. Bu süreç rengin düzeltilmesini, rengin filtrelenmesini, rengin bölümlendirilmesini ve renk tabanlı nesnelerin tanınmasını içermektedir.

Çok spektrumlu görüntü; görüntüleme amacıyla farklı spektrum bölgelerini kullanarak nesneleri görselleştirme yöntemidir. Bu yaklaşımın amacı, insan gözüyle algılanabilen ancak önceki yaklaşımlarla mutlaka tespit edilemeyen bilgileri elde etmektir.

Temel anlamda görüntü işleme bir kamera tarafından çekilen görüntünün analogdan dijitalle dönüştürülmesi ve belirlenen bir amaca göre yeni bir görüntü oluşturulmasıdır (Altunay, 2012). Dijital görüntü analog görüntülerden örneklem alınması ile elde edilir. Görüntüler Sayısal formatlarda temsil edildiği bir dizi piksel üzerinde çalışır. Her piksel, bir renk değerine veya gri tonuna sahip olabilir ve bir görüntü, bu piksellerin bir araya gelmesiyle oluşur. Görüntü işleme, bu pikseller üzerinde matematiksel işlemler yaparak, görüntüdeki bilgileri çıkarır ve analiz eder.

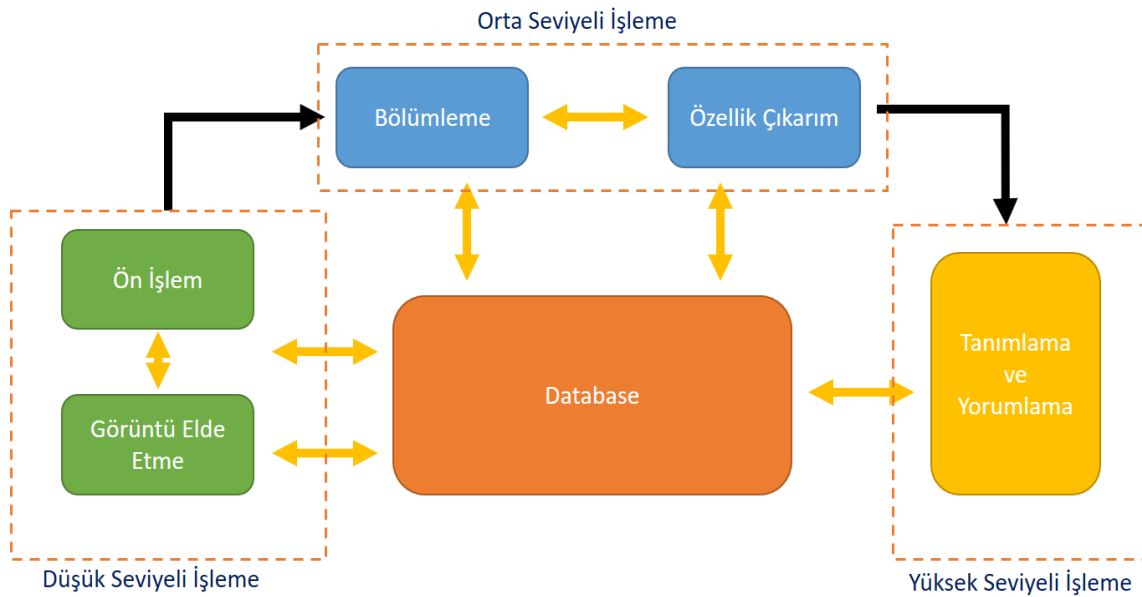
Görüntü işleme kaydedilmiş görüntülerin bilgisayar ortamında bilgisayar yazılımları ile kullanım amacına uygun olarak değiştirilmesi işlemidir (Görüntü İşleme Teknolojisi, 2022). Bu işlem, verilerin ölçülüp değerlendirilmesinden sonra başka bir aygıtta okunabilir hale getirilmesi veya dijital platformlar arasında aktarılması amacıyla gerçekleştirilir. Sinyal işleme, bir sinyal üzerinde matematiksel veya hesaplama temelli operasyonların uygulanmasıyla ilgilenen bir alandır. Görüntü işleme ise farklı bir işlemdir. Görüntü işleme,

genellikle kaydedilmiş olan mevcut görüntülerin işlenmesi anlamına gelmektedir. Bu işlem, mevcut resim ve grafiklerin değiştirilmesi, dönüştürülmesi, iyileştirilmesi veya analiz edilmesiyle ilgilenir.

Görüntü işleminin faydaları uygulama tekniğine bağlı olarak değişir. Her bir görüntü işleme tekniği farklı bir perspektiften yaklaşılır. Örneğin, tıp alanında, röntgen veya MRI gibi görüntüleme teknikleriyle elde edilen görüntüler hastalıkların teşhisi ve tedavisi için analiz edilir. Otomotiv endüstrisinde, görüntü işleme teknikleri, araçlar için sürüş yardım sistemlerinin geliştirilmesinde kullanılır. Günümüzde tasarım, üretim, güvenlik, sağlık hizmetleri, elektronik cihazlar, makineler, yapı sektörü, otomotiv endüstrisi gibi birçok alanda görüntü işleme teknolojileri yaygın bir şekilde kullanılmaktadır.

3.2. Görüntü İşleme Adımları

Görüntülerin elde edilmesi en temel adımlardan ilkidir. Daha sonra görüntü üzerinde ön işleme adımları gerçekleştirilir. Bu adımlar görüntüyü giderme, parlaklık ayarlama veya keskinleştirme aşamalarından oluşur. Daha sonra görüntü üzerinde özellik çıkarımı ve görüntü tanıma adımları gerçekleştirilir. Bu adımlar görüntüdeki nesneleri veya özellikleri tanımlar. Son olarak elde edilen bilgilere dayanarak görüntülerin yorumlanması aşamaları gerçekleştirilir. Şekil 3.1.'de görüntü işleme adımları gösterilmektedir.



Şekil 3. 1. Görüntü işleme adımları (Çayıroğlu, 2020)

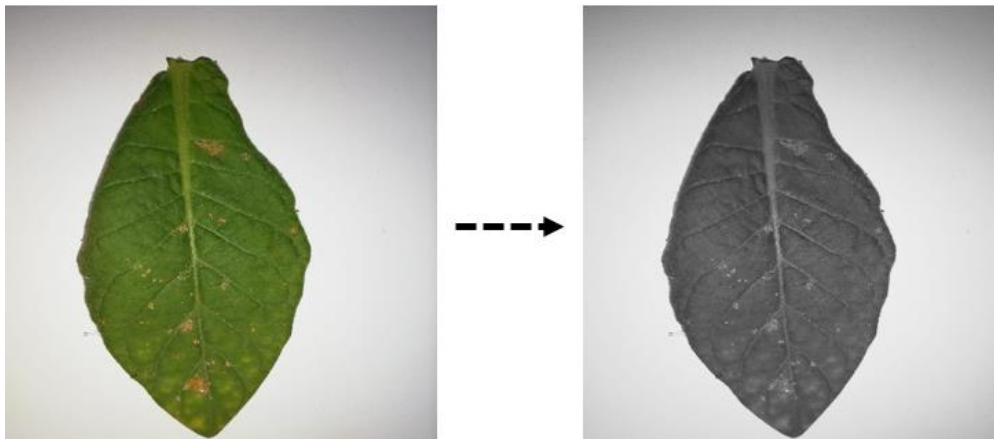
Görüntü işleme, yapay zekâ teknikleriyle birleştirildiğinde daha güçlü ve etkili bir analiz aracı haline gelir. Yapay zekâ algoritmaları, görüntüleri analiz etmek ve öğrenmek için kullanılır. Bu algoritmalar, büyük veri setlerini kullanarak desenleri tanımlayabilir ve görüntülerdeki nesneleri veya özellikleri tespit edebilir. Bu sayede, hastalıklı tütün yapraklarının tespiti gibi karmaşık görevlerde daha yüksek doğruluk ve hız elde edilebilir.

3.2.1. Ön işlem

Görüntü işlemede ön işleme, görüntü verilerini analiz ve işlemeye hazırlayan bir dizi işlemdir. Ön işleme adımları çeşitli amaçlar için gerçekleştirilebilir; Örneğin, görüntü kalitesini iyileştirmek, gürültüyü azaltmak, parlaklığı artırmak, kenarları vurgulamak veya görüntüdeki istenmeyen öğeleri kaldırmak için kullanılabilir.

Gri tona dönüştürme

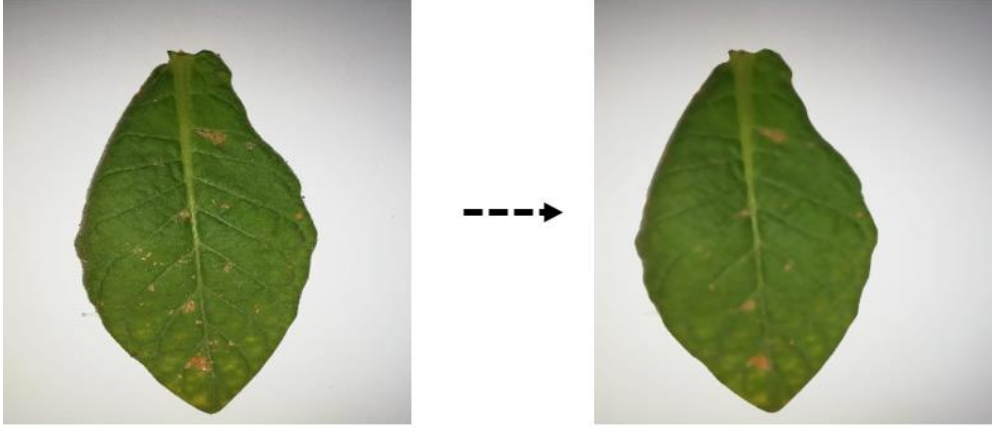
Görüntü işlemede gri tona dönüştürme işlemi, renkli bir görüntüyü gri tonlamalı görüntüye dönüştürme işlemidir. Bu işlem görüntüdeki her pikselden renk bilgisini kaldırır ve geriye yalnızca parlaklık bilgisi kalır. Bu, her pikselin yalnızca bir parlaklık değerine sahip olduğu ve görüntünün siyah, beyaz ve gri tonlarından oluştuğu anlamına gelmektedir. Gri tonlamalı görüntüler, sonraki analiz veya işleme adımlarında daha hızlı ve daha düşük bellek kullanımına olanak tanır. Ayrıca bazı görüntü işleme algoritmaları gri tonlamalı görüntülerle daha iyi çalışabilir. Renkli bir görüntünün gri görüntüye dönüşmüş görüntüsü Resim 3.2.'de verilmiştir.



Resim 3. 2. Renkli görüntünün gri görüntüye çevrilmiş gösterimi

Bulanıklaştırma(blurring)

Görüntü işlemede bulanıklaştırma olarak da bilinen bu işlem, görüntüdeki ayrıntıların azaltılması veya keskinliğin ortadan kaldırılması işlemidir. Bu işlem, görüntüdeki gürültüyü azaltmak, nesneleri gizlemek veya odaklanmamış bir görüntünün daha pürüzsüz ve estetik açıdan daha hoş görünmesini sağlamak gibi çeşitli amaçlar için kullanılabilir. Bulanıklaştırma, görüntüdeki keskin kenarları yumuşatır ve pikseller arasındaki geçişleri yumuşatır. Yapılan tez çalışmasında medyan bulanıklaştırma (median blurring) yöntemi kullanılmıştır. Medyan bulanıklaştırma bir pikselin değerini, çevresindeki piksellerin medyan değeri ile değiştirilerek elde edilir. Bulanıklaştırma işlemi tütün yaprağı üzerinde Resim 3.3'te gösterilmektedir.



Resim 3. 3. Bulanıklaştırma işlemi

Renk uzayı dönüşümü

Bu tez çalışmasında lab renk uzayına dönüştürme işlemi gerçekleştirilmiştir. Lab renk uzayı, renk bilgisini parlaklık (L), yeşil-magenta (a^*), ve mavi-sarı (b^*) bileşenleri olarak üç farklı bileşenle ifade eden bir renk uzayıdır (Othman, 2020). Gerçekleştirilen çalışmada sadece parlaklık bileşenini kullanılmaktadır. Bu parlaklık bilgisi, görüntü işleme ve analizinde farklı uygulamalarda kullanılabilir. Örneğin, görüntü segmentasyonu, kenar tespiti, nesne tanıma ve diğer birçok işlemde kullanılabilir. Lab renk uzayı, insan gözünün renkleri algılama şekline daha yakın olduğu için, görüntü işleme ve analizinde sıkça kullanılan bir renk uzayıdır.

Görüntü eşikleme(thresholding)

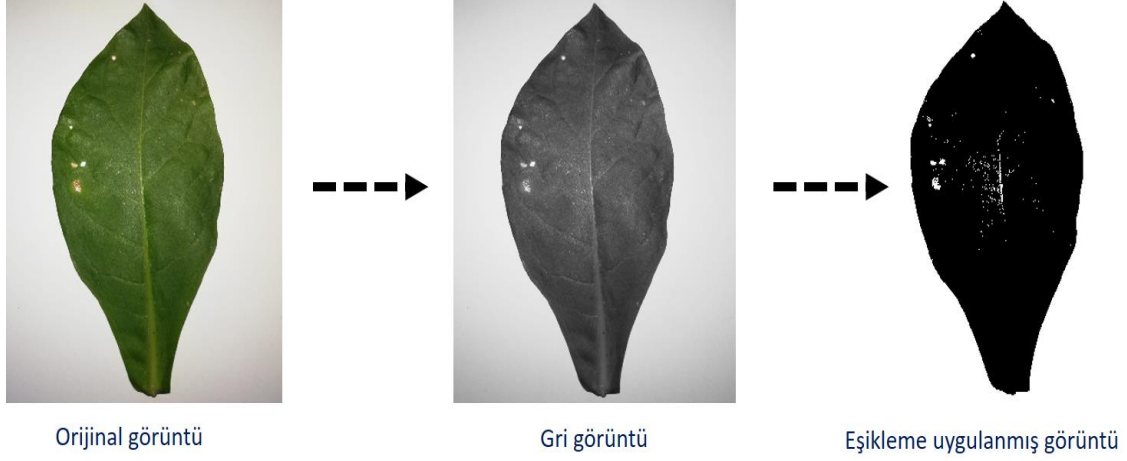
Makine öğrenimi yöntemlerini kullanarak görüntülerden bilgi ve özellik çıkarma eşikleme yöntemi ile gerçekleştirilir. Eşikleme, görüntüleri bölümlere ayırmak için görüntü işlemede sıklıkla kullanılan önemli bir yaklaşımdır. Görüntü işlemedeki en temel konulardan biridir. Eşikleme görüntüyü farklı bölgelere ayırmak için kullanılmaktadır (Elen, 2020).

Görüntüdeki nesneleri arka plandan ayırmak ve istenmeyen nesneleri kaldırmak için eşikleme yöntemi kullanılır. Eşikleme, görüntüdeki piksellerin belirli bir eşiği geçip geçmediğini kontrol ederek nesneleri ayırmak için kullanılır. Görüntüdeki her piksel, belirlenen eşik değerine göre karşılaştırılır ve eşik değerini geçen pikseller nesne olarak kabul edilirken, geçemeyen pikseller arka plan olarak kabul edilir. Bu şekilde, görüntüdeki nesneleri arka plandan ayırmak ve istenmeyen nesneleri kaldırmak için eşikleme yöntemi kullanılır. Eşik değeri, uygulandığı görüntüye bağlı olarak ayarlanır ve bu değer, nesnelerin ve arka planın farklılıklarına bağlı olarak değişebilmektedir. Eşik değeri(T) ve eşik değer fonksiyonuna($f(x,y)$) ait eşitlik denklem 3.1 de verilmiştir.

$$f(x,y) = \{ 255, \text{ eğer } f(x,y) > T(x,y) \text{ ise}$$

$$\{ 0, \text{ eğer } f(x,y) \leq T(x,y) \text{ ise (Boyacıgil, 2022)} \quad (3.1)$$

Eşik değerinin doğru belirlenmesi görüntünün içeriğinin bozulmaması için kritik öneme sahiptir. Bu çalışmada kullanılan klasik eşikleme tekniği, gri tonlarından oluşan bir görüntüyü siyah ve beyaz piksellerden oluşan bir ikili görüntüye dönüştürmek için kullanılır. Bu yöntem, basit ve sıklıkla kullanılan bir yöntemdir. Denklem 3.1’de gösterilen bu yöntemde tek bir eşik değeri kullanılarak gerçekleştirilmiştir. Eşik değerinin belirlenmesi sürecinde gri piksellerin T eşik değerinden büyük yada küçük olduğuna bakılmaktadır. Eşikleme işlemi uygulanmış bütün yaprağı Resim 3.4’te gösterilmektedir.



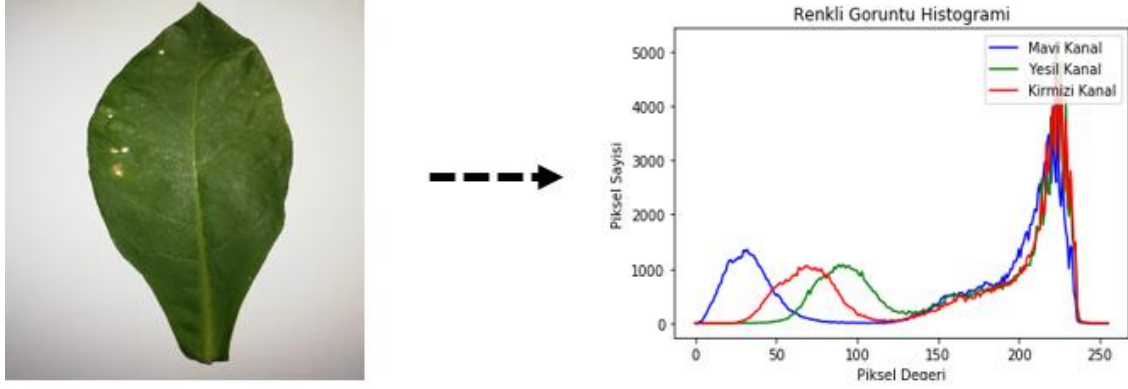
Resim 3. 4. Klasik eşikleme yöntemi

Görüntü histogramı

Görüntü histogramı, bir görüntüdeki piksellerin renk veya gri tonlama değerlerinin dağılımını (frekansını) gösteren bir grafikdir (Horasan, 2016). Histogram, görüntü analizi ve işlenmesi için önemli bir araçtır. Görüntü histogramı, görüntünün her noktasındaki pikselleri tespit ederek piksel sayısını görüntüler. Bu sayede histogramdan görüntüye ait çeşitli bilgiler çıkarılır. Histogram analizi, bir görüntünün renk dağılımının veya gri tonlamanın yanı sıra karşıtlık, parlaklık ve renk dengesi gibi özelliklerini anlamak için kullanılır. Histogram, görüntüdeki piksellerin yoğunluk değerlerini gösteren bir grafik olarak görüntülenir.

Görüntü histogramı, piksel dağılımını inceleyerek görüntünün genel özelliklerini anlamamıza yardımcı olur. Bir görüntü histogramı, örneğin görüntünün genel olarak koyu mu yoksa açık mı olduğunu, renk tonlarının ne kadar tekdüze veya heterojen olduğunu ve hangi renklerin baskın olduğunu gösterir.

Histogram analizi sıklıkla görüntü işleme algoritmalarında kullanılır. Örneğin histogram analizi, bir görüntüyü geliştirmek veya düzeltmek için kontrastın ayarlanması, parlaklığın değiştirilmesi veya renk dengesinin değiştirilmesi gibi işlemlerde kullanılır. Ayrıca histogram analizi görüntü bölütleme, nesne tespiti ve görüntü sınıflandırma gibi uygulamalarda da sıkça kullanılmaktadır. Resim 3.5'te görüntü histogramı görülmektedir.



Resim 3. 5. Orijinal görüntü ve histogram grafiği

Morfolojik görüntü işleme

Morfoloji, görüntüleri şekiller temelinde işleyen geniş kapsamlı bir görüntü işleme işlemleri kümesidir. Bu işlem, dijital görüntüdeki eksik noktaların doldurulması veya fazlalıkların giderilmesi amacıyla geometrik hesaplamalar kullanılarak gerçekleştirilir. Bu tür işlemler görüntü işleme alanında önemlidir ve birçok uygulamada kullanılmaktadır (Aydın, 2021). Eklenicecek veya çıkarılacak olan bölgeler, daire, kare ve dikdörtgen gibi yapısal elemanların görüntü üzerinde gezdirilmesiyle belirlenir.

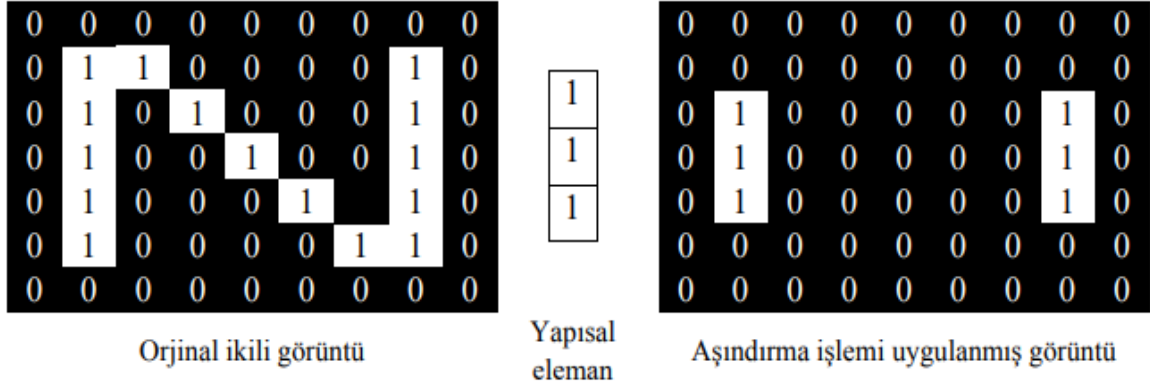
Morfolojik görüntü işleme, şekil analizi, kenar tespiti, nesne tanıma, nesne ayırma ve gürültü giderme gibi uygulamalarda yaygın olarak kullanılmaktadır. Bu yöntem, görüntüdeki nesnelerin boyutunu, şeklini, uzantısını ve bağlantısını belirlemek için operatörler ve filtreler kullanır. Bu yöntem tıp, endüstriyel görüntüleme, robotik, otomasyon gibi alanlarda yaygın olarak kullanılan bir görüntü işleme tekniğidir.

Aşındırma işlemi(Erosion)

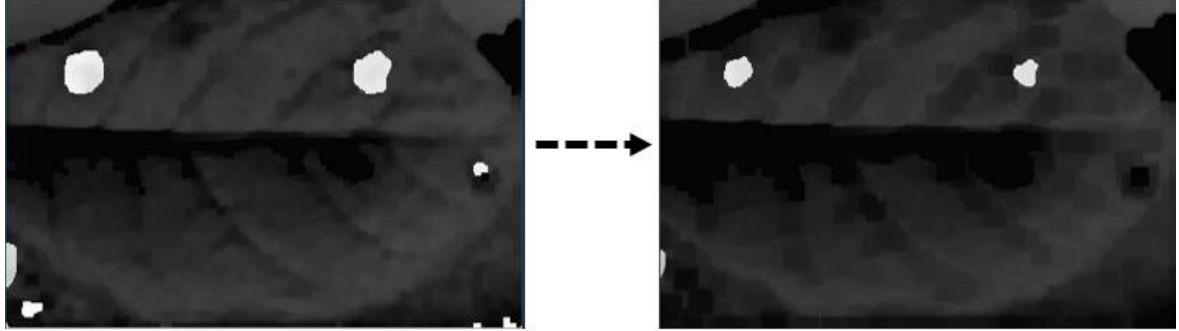
Görüntü işlemede kullanılan aşındırma, görüntüdeki piksellerin yoğunluğunu azaltarak veya tamamen ortadan kaldırarak görüntünün belirli kısımlarındaki ayrıntıları kaldıran veya bulanıklaştıran bir işlemdir. Orijinal görüntüye göre daha az piksel içerir. Aşındırma işlemi uygulanmış görüntüde nesneler küçülür, delikler büyür, birbirine bağlı alanlar ayrılır (Altunkurt ve Kahrıman, 2011). Aşındırma işlemi genellikle gürültü azaltma, kenar iyileştirme veya görüntü sıkıştırma gibi amaçlarla kullanılır. Aşındırma işlemi, morfolojik işlemler adı verilen bir dizi matematiksel işlemi içerir. Aşındırma işlemi için matematiksel

denklem 3.2’de, görsel olarak aşındırma işlemi Şekil 3.2’de ve Resim 3.6’da aşağıda gösterilmiştir.

$$A \ominus B = \{z | (B)z \subseteq A\} \quad (3.2)$$



Şekil 3. 2. Aşındırma işlemi uygulaması

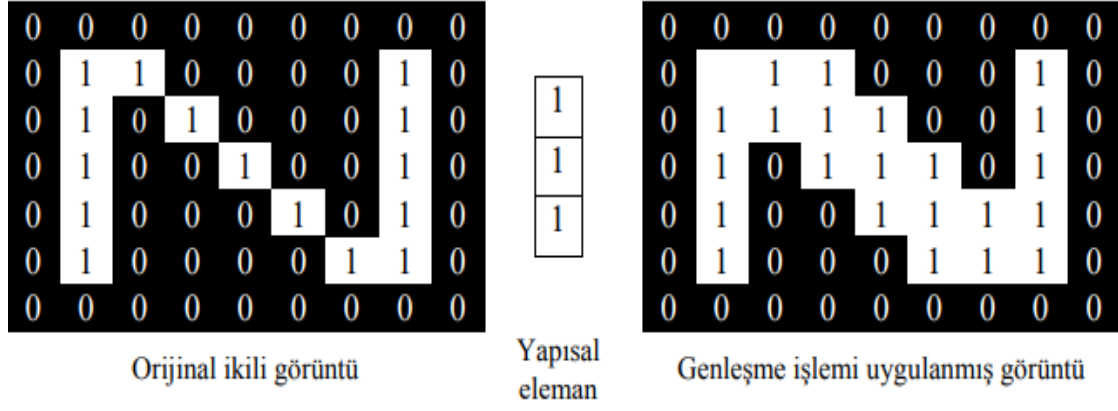


Resim 3. 6. Aşındırma işlemi uygulanmış görüntü

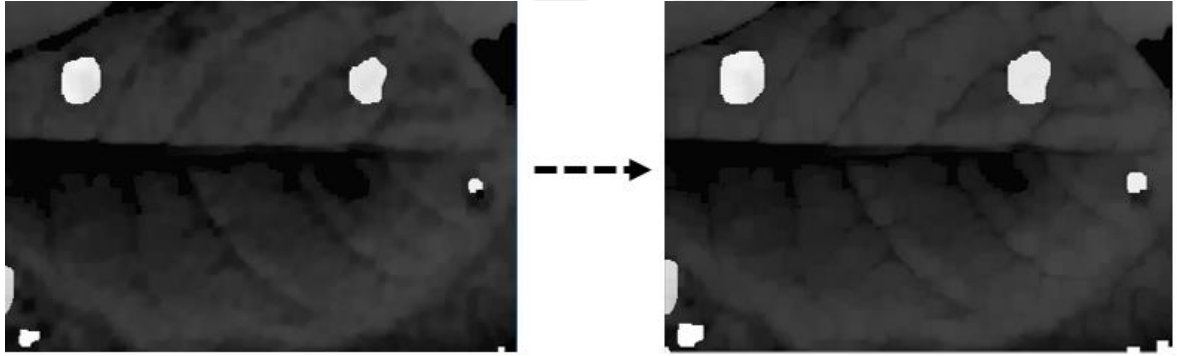
Genişleme işlemi(dilation)

Morfolojik genişleme, bir görüntüdeki nesnelerin veya yapıların boyutunu veya alanını artırmak için kullanılan morfolojik bir işlemdir (Acar ve Bayram, 2009). Genellikle bu işlemi uygulamak için bir yapı elemanı (kernel) kullanılır. Bu işlem, bir görüntüdeki nesnelerin kenarlarını ve sınırlarını keskinleştirmek, delikleri tıkmak, küçük boşlukları doldurmak için kullanılır. Bu işlem, görüntü işleme alanında özellikle nesne tespiti, kenar tespiti ve şekil analizi gibi uygulamalarda yaygın olarak kullanılmaktadır. Genişleme işlemi ile ilgili matematiksel denklemler 3.3’de, görsel açıklamalar ise Şekil 3.3’te ve Resim 3.7’de ayrıntılı olarak sunulmuştur.

$$A \oplus B = \{z | (B)z \cap A \neq \emptyset\} \quad (3.3)$$



Şekil 3. 3. Genişleme işlemi uygulaması

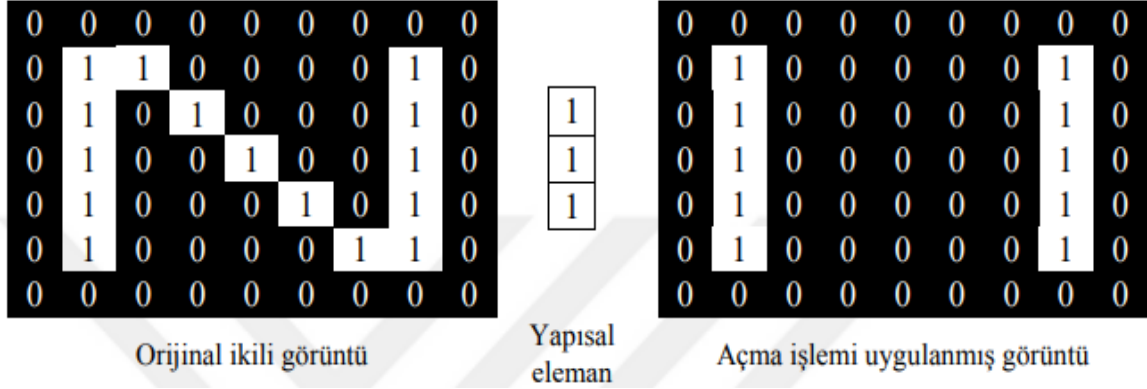


Resim 3. 7. Aşındırma işlemi uygulanmış görüntü

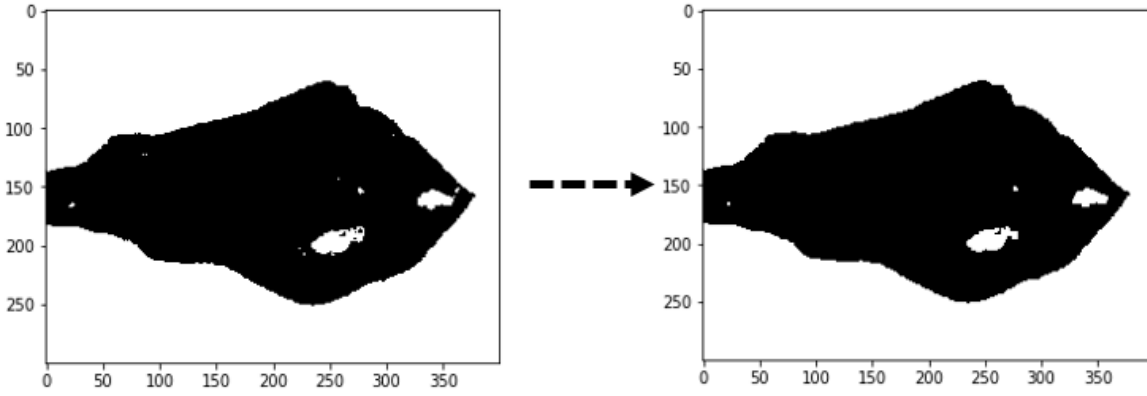
Açma işlemi (opening)

Morfolojik açma işlemi görüntü üzerinde sırası ile aşınma ve genişleme işlemlerinin uygulanmasıdır. Açma işlemi görüntüdeki nesnelerin sınırlarını düzeltmek, küçük boşlukları doldurmak ve gürültüyü azaltmak amacıyla kullanılır (Boztoprak vd., 2007). Bu işlem genellikle, görüntüdeki ince detayları azaltmak ve nesnelerin daha belirgin hale gelmesini sağlamak için uygulanır. Açma işlemi, başlangıçta erozyon işlemi uygulayarak nesneleri küçültür ve ardından bu küçültülmüş nesneleri genişletmek için genişleme işlemi yapar. Sonuç olarak, nesnelerin kenarları daha yumuşak ve düzgün hale gelirken, küçük aralıklar kapanır. Opening işlemi ile ilgili matematiksel denklemler 3.4'de, görsel açıklamalar ise Şekil 3.4'te ve Resim 3.8'de ayrıntılı olarak sunulmuştur.

$$A \circ B = (A \ominus B) \oplus B \quad (3.4)$$



Şekil 3. 4. Açma işlemi uygulaması



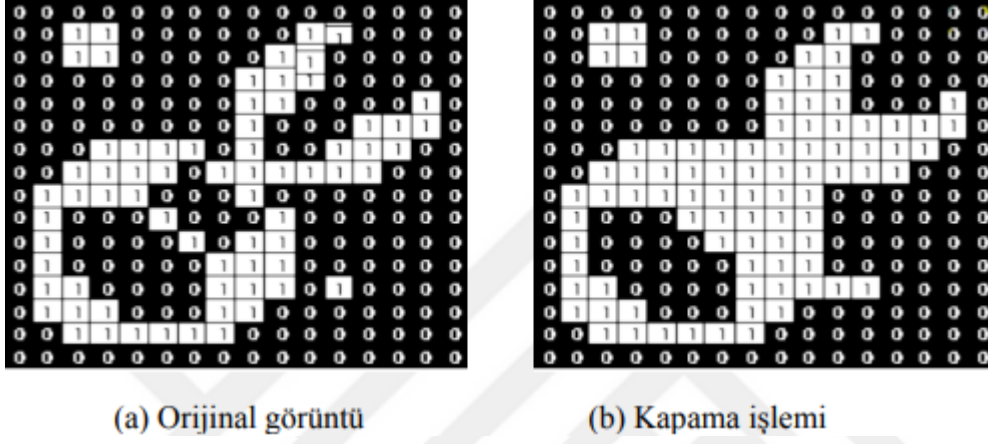
Resim 3. 8. Açma işlemi uygulanmış resim

Kapama işlemi(closing)

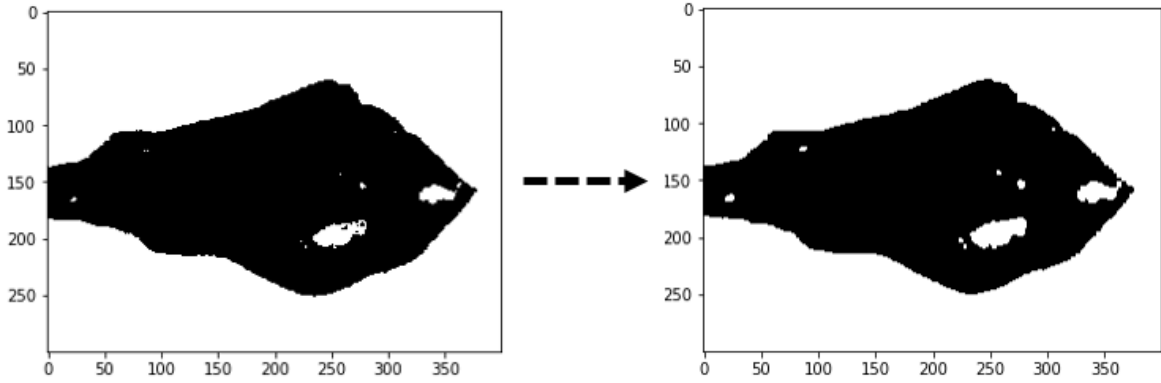
Kapanma işlemi (closing), açma işleminin tersidir. Görüntüde ilk olarak genişleme işlemi uygulanır, ardından aşınma işlemi gerçekleştirilir. Morfolojik kapama işlemi görüntülerdeki nesnelerin kenarlarını düzeltmek, eksik bağlantıları tamamlamak ve gürültüyü azaltmak için kullanılır (Jimeno-Morenilla vd., 2014). Kapama işlemi, nesnelerin daha belirgin hale gelmesi veya nesneler arasındaki boşlukların doldurulması gereken durumlarda genellikle kullanılır. Nesnelerin sınırlarını düzeltmek için kullanılan bu teknik, özellikle kenar tespiti ve nesne ayırma gibi işlemlerde etkilidir. Ayrıca, görüntülerdeki küçük gürültüleri azaltmak ve nesneler arasındaki boşlukları doldurmak için de tercih edilir. Kapanma işlemi, görüntüdeki detayları korurken gürültüyü azaltma yeteneğine sahiptir. Özellikle, görüntülerdeki küçük nokta veya çizgi gibi gürültülerin giderilmesinde etkilidir. Bu sayede,

görüntü üzerinde daha net ve anlaşılır bilgiler elde edilir. Kapanma işlemi ile ilgili matematiksel denklemler 3.5'te, görsel açıklamalar ise Şekil 3.5'te ve Resim 3.9'da ayrıntılı olarak sunulmuştur.

$$A \cdot B = (A \ominus B) \oplus B \quad (3.5)$$



Şekil 3. 5. Kapama işlemi uygulaması



Resim 3. 9. Kapama işlemi uygulanmış resim

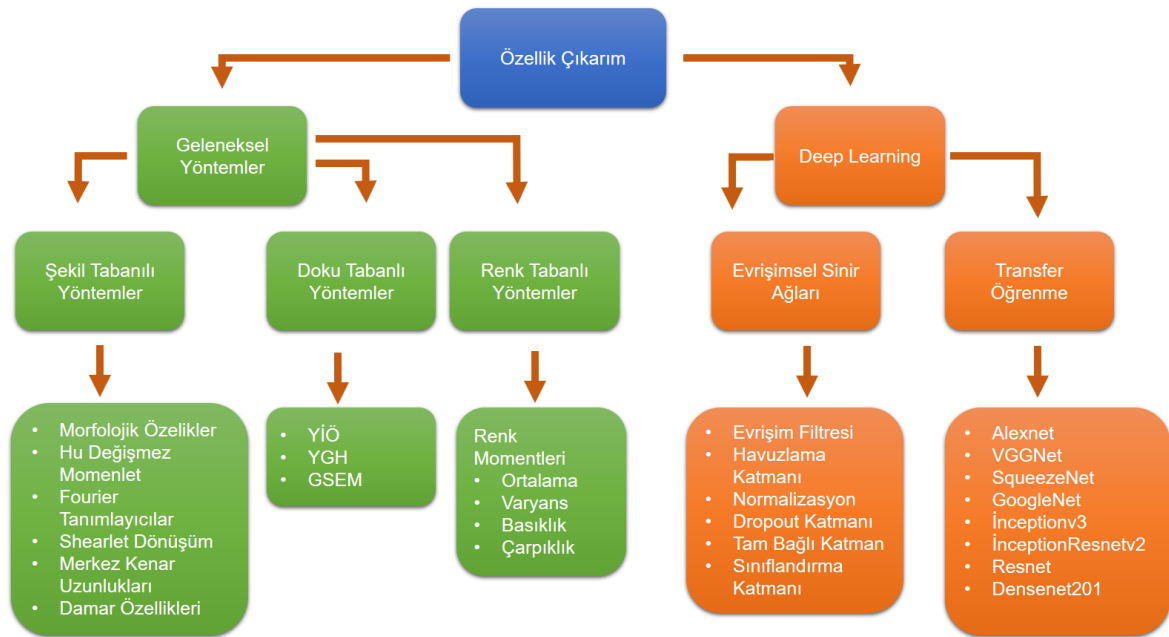
3.2.2. Özellik çıkarımı

Özellik çıkarma işlemi, nesne tanıma alanında son derece önemli bir rol oynar. Bu işlem, nesne tanıma sistemlerinin başarı oranını önemli ölçüde etkiler. Özellik çıkarma, nesnelerin daha küçük ve özgün niteliklerle temsil edilmesini sağlayarak nesne tanıma sistemlerinin performansını artırır. Özellik çıkarma aynı zamanda makine öğrenimi ve derin öğrenme gibi alanlarda da büyük bir öneme sahiptir. Bu yöntem, karmaşık veri setlerinden anlamlı

özelliklerin çıkarılmasını sağlar ve eğitim sürecinde kullanılan verilerin boyutunu azaltır. Böylece, nesne tanıma algoritmalarının daha hızlı ve verimli çalışmasına olanak tanır (Toraman, 2016).

Nesne tanıma sistemlerinde kullanılan özellik çıkarma yöntemleri oldukça çeşitlidir. Bu yöntemler, genellikle görüntülerdeki nesnelerin belirgin özelliklerini vurgulayarak tanıma işlemini kolaylaştırmayı amaçlarlar. Bu özellikler, nesnelerin benzersizliğini ve ayırt edici niteliklerini yansıtmalıdır. Özellik çıkarma aşamasında kullanılan algoritmaların seçimi ve parametrelerin ayarlanması, nesne tanıma sisteminin başarısını büyük ölçüde belirler. Bu nedenle, doğru özellik çıkarma yöntemlerinin seçilmesi ve bu yöntemlerin veri setine özgü ihtiyaçlara uygun şekilde ayarlanması son derece kritik bir öneme sahiptir.

Bazı özellik çıkarma yöntemleri, kenar tespiti algoritmalarını kullanarak görüntülerdeki nesnelerin kenarlarını ve köşelerini vurgular. Diğer yöntemler ise renk, şekil veya dokusundaki özelliklere dayanarak nesneleri tanımaya çalışır. Her bir yöntemin amacı, nesnelerin benzersiz niteliklerini yakalamak ve bu özellikleri kullanarak daha doğru ve güvenilir nesne tanıma sonuçları elde etmektir. Özellik çıkarım teknikleri Şekil 3.6'da gösterilmiştir.



Şekil 3. 6. Özellik çıkarım teknikleri

Geleneksel yöntemler

Bu tez araştırması içerisinde şekil tabanlı yöntemlerden morfolojik özellikler ve önceden eğitilmiş evrişimsel sinir ağları mimarileri incelenmiştir. Esa ve derin öğrenme yöntemleri yapay zekâ başlığı altında incelenmiştir.

Morfolojik özellikler

Morfolojik özellikler, görüntülerden çıkarılan ve morfolojik görüntü işleme teknikleri kullanılarak elde edilen özelliklerdir. Bu özellikler nesnelerin şekli, boyutu, yapısı ve topolojik özellikleri gibi yapısal özelliklerini tanımlamak için kullanılır (Kumar ve Bhatia, 2014). Morfolojik özellikler görüntü analizinde önemli bir rol oynamaktadır. Bu özellikler bir nesnenin şekil, boyut, yapı ve topolojik özellikler gibi yapısal özelliklerini tanımlamak için kullanılır. Görüntülerdeki nesnelerin sınıflandırılması, tanımlanması ve takip edilmesi gibi birçok uygulama bu özelliklerin kullanılmasını gerektirmektedir.

Morfolojik özellikler, bir nesnenin şeklini ve yapısını tanımlamak için kullanılan matematiksel ölçütlerdir. Bu özellikler nesnelerin geometrik özelliklerini ifade eder ve nesne sınıflandırma, tanımlama ve izleme gibi birçok uygulama için önemlidir. Örneğin bir nesnenin alanı, çevresi gibi özellikler morfolojik özelliklere örnek olabilir.

Literatürde kullanılan morfolojik özellikler, nesnelerin görüntü analizi için kullanılabilecek bir dizi özelliği içerir. Bunlar arasında alan, çevre, yüzey alanı, hücre sayısı, eğrilik, merkezi moment, uzunluk, genişlik ve diyagonal uzunluk yer almaktadır. Bu özellikler nesnelerin şekli, boyutu ve topolojik özellikleri hakkında istatistiksel bilgi sağlar. Bu özellikler nesne tanıma, sınıflandırma, tıbbi görüntü işleme gibi çeşitli uygulamalar için kullanışlıdır.

3.3. Raspberry Pi Os İşletim Sistemi

Raspberry kartının çalışabilmesi için işletim sistemi yüklenmesi gerekmektedir. Raspberry Pi'nin işlemci mimarisi nedeniyle, yalnızca windows 10 IoT core gibi sınırlı bir windows sürümü yüklenebilir (Özdemir, 2018). Ancak, windows 10 IoT işletim sisteminin kullanıcılar için sunduğu kısıtlı arayüz nedeniyle kullanılması oldukça zordur. Windows 10

IoT Core'a kod yüklemek için IDE'ye ihtiyaç duyulurken, ayrıca cihazda derleme veya kod düzenleme de oldukça zordur. Raspberry Pi'de linux tabanlı işletim sistemleri sıkça kullanılmaktadır. Raspberry Pi üzerinde sorunsuz bir şekilde çalışan işletim sistemleri arasında ubuntu, Archlinux, Fedora, Puppy Linux, Slackware, Chromium OS, FreeBSD, NetBSD, openSUSE, RISC OS gibi işletim sistemleri bulunmaktadır.

Farklı birçok farklı işletim sistemi bulunmasına rağmen Raspberry pi kartına özel olarak çıkardığı Raspberry Pi OS(eski adıyla Raspbian) işletim sistemi resmi olarak desteklenmektedir. Temel programlar ve yardımcı programlar yüklü olarak gelmektedir. Yazılımcıların işlerini kolaylaştırmak amacıyla 35.000 den fazla kütüphane yüklü olarak gelmektedir. Resim 3.10'da raspberry pi os masaüstü görünümü verilmiştir.



Resim 3. 10. Raspberry pi os masaüstü görünümü

3.3.1. Raspberry pi kartına işletim sisteminin kurulması

Bu tez çalışmasında raspberry pi os işletim sistemi kullanılmıştır. Kurulum için gerekli adımlar aşağıda verilmiştir.

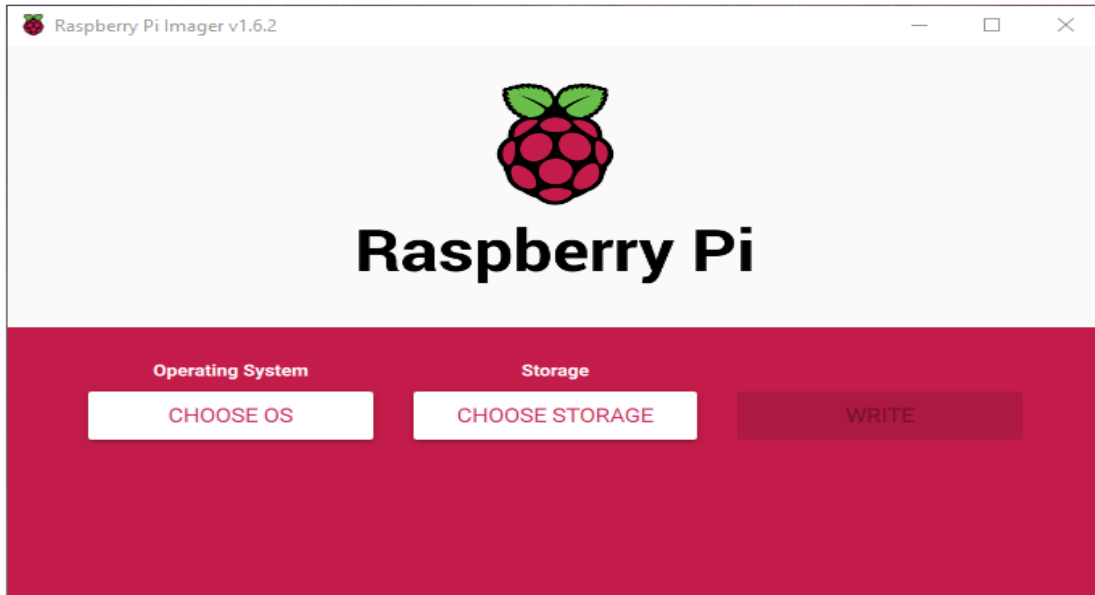
Gerekli malzemeler:

- Raspberry pi 3 kart

- Fare ve klavye
- Hdm1 kablo
- Mikro sd kart(16 gb)
- Ekran
- Güç adaptörü

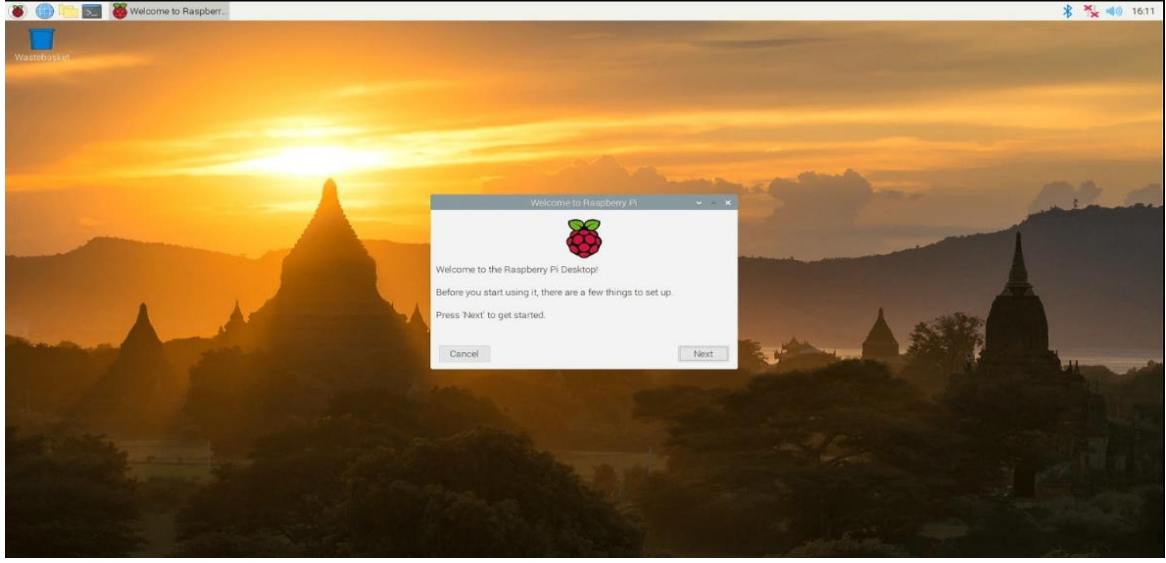
Raspberry Pi OS işletim sistemini bir Micro SD Karta kurabilmek için, Raspberry Pi'nin resmi yazılımı olan "Raspberry Pi Imager" programını bilgisayara indirilmesi ve kurulması gerekmektedir. Bu program, Micro SD kartına işletim sisteminin yazılmasını ve kurulumunun gerçekleştirilmesini sağlar (Diker, 2023).

Resim 3.11'de gösterilen Raspberry Pi Imager programı bilgisayara kurulduktan sonra program ara yüzünden, choose os butonuna basılarak kurulacak işletim sistemi seçilir. Ardından choose storage butonu ile sd kart seçildikten sonra write butonu ile işletim sistemi hafıza kartına yazdırma işlemi gerçekleştirilir.



Resim 3. 11. Raspberry Pi imager programı arayüzü

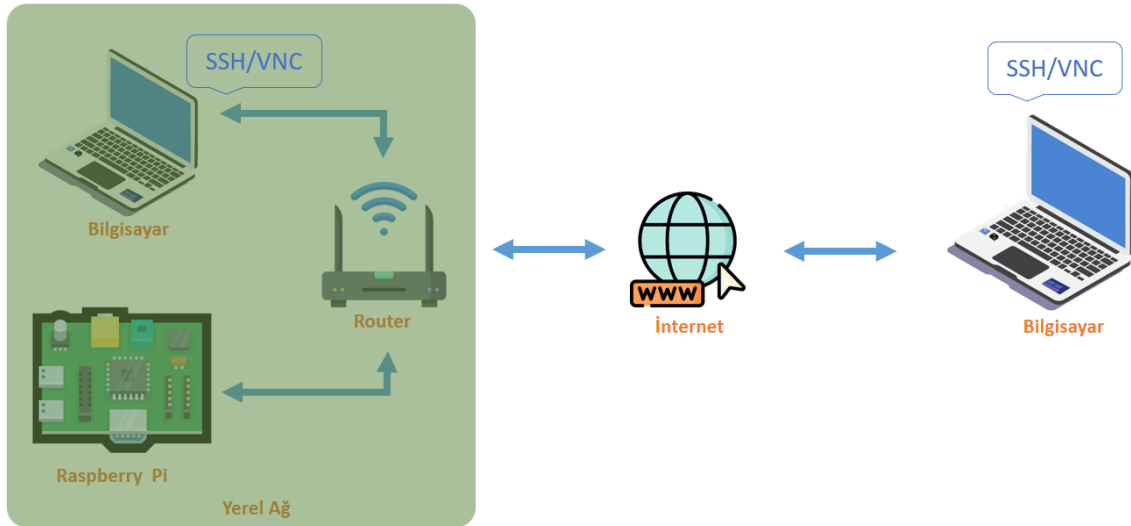
Raspberry pi os işletim sistemi yüklendikten sonra karşımıza Resim 3.12'deki masaüstü ekranı gelmektedir.



Resim 3. 12. Raspberry pi os ilk açılış ekranı

3.3.2. Raspberry pi'ye ssh(uzak bağlantı) bağlantısı

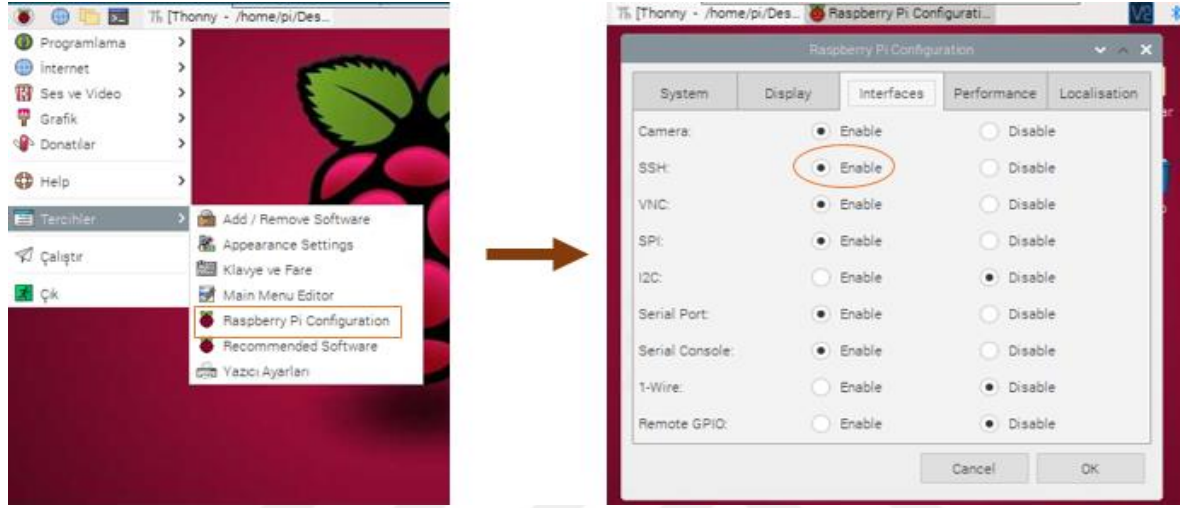
Raspberry Pi 3, uzaktan erişim ve yönetim için SSH (secure shell) protokolünü desteklemektedir. Bu sayede SSH bağlantısı kurarak Raspberry pi'yi herhangi bir cihazdan uzaktan kontrol edebilir ve yönetilebilmektedir. Şekil 3.7'de SSH bağlantı gösterilmektedir.



Şekil 3. 7. Raspberry pi'ye SSH bağlantısı

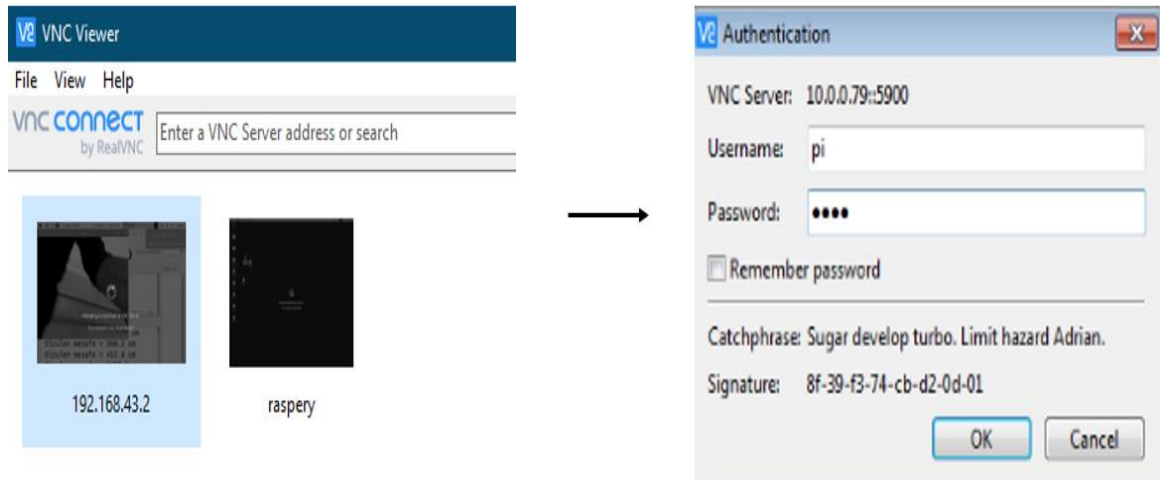
Uzaktan bağlantı için Raspberry Pi kartını çalıştırmak ve cihazınızın başarılı bir şekilde ethernet veya wifi aracılığıyla bir ağa bağlantısının yapılması gerekmektedir. Raspberry Pi bir ağa bağlı olması oldukça önemlidir, aksi takdirde uzaktan bağlantı gerçekleştirilemez.

Kartı ağı bağladıktan sonra üst paneldeki menüyü açarak ve "Tercihler" sekmesi ve ardından "Raspberry Pi Yapılandırması/ Raspberry Pi Configuration" sekmesi seçilmelidir. Açılan pencerede "Arayüz/interfaces" sekmesine açılarak "SSH" seçeneğini etkinleştirilmelidir. Yapılan değişiklikleri kaydetmek için "OK" düğmesine tıklayarak kart yeniden başlatılmalıdır. Resim 3.13'te SSH ayarı gösterilmektedir.



Resim 3. 13. Raspberry pi ssh özelliğinin aktif edilmesi

SSH etkinleştirildikten sonra Raspberry pi karta VNC Viewer programıyla istenilen bilgisayardan bağlanılabilir. Resim 3.14' te gösterilen VNC Viewer programına Raspberry pi kartının ip adresi girildikten sonra kullanıcı adı ve şifre bölümleri girilerek giriş yapılır (Niamh, 2023).



Resim 3. 14. Vnc viewer ile raspberry pi kartına bağlanması

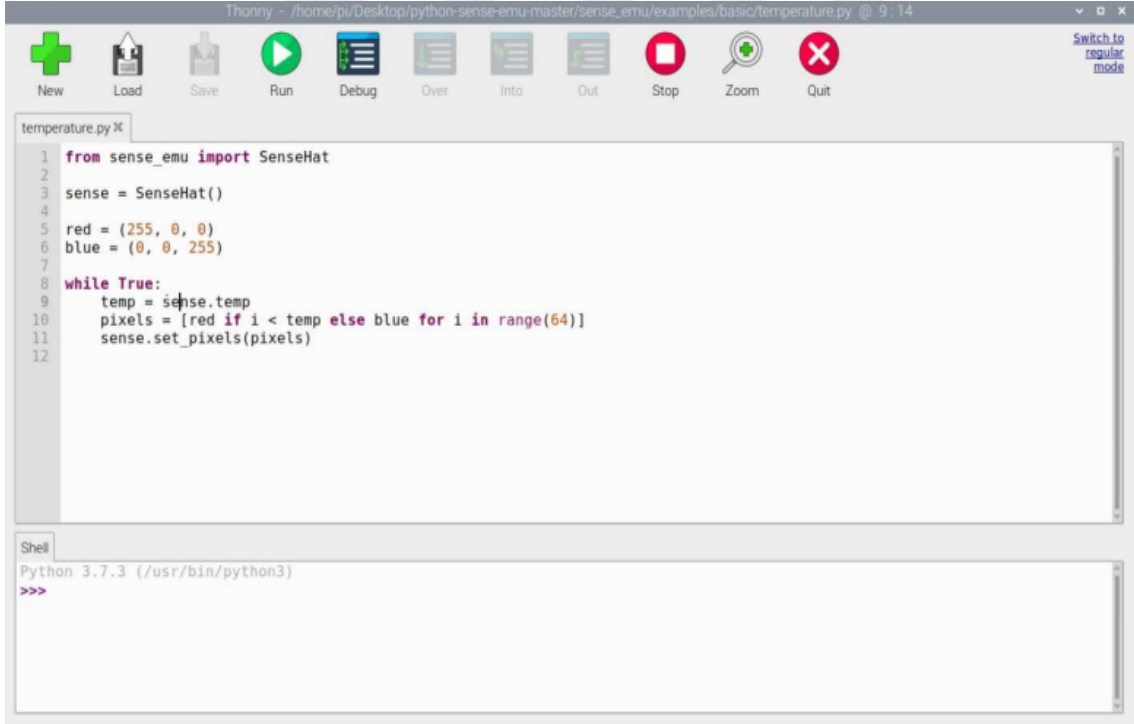
Varsayılan raspberry pi kullanıcı adı ve şifre:

- Kullanıcı adı: pi
- Şifre: ahududu

Varsayılan şifre güvenlik önlemleri nedeniyle değiştirilmelidir.

3.3.3. Thonny ide programı

Gerçekleştirilen çalışmada Raspberry pi os işletim sistemi içerisinde hazır olarak gelen Thonny IDE programı kullanılmıştır. Kod yazımı, hata ayıklama, değişken izleme ve programın adım adım çalıştırılması gibi temel özellikleri içeren bu program 2015 yılında ortaya çıkmıştır (Zipponi, 2021). Resim 3.15'te thonny programı ara yüzü gösterilmektedir.



Resim 3. 15. Thonny IDE program ara yüzü

Gerçekleştirilen tez çalışmasında gömülü sistem uygulamaları dışında, bilgisayar ortamında görüntü işleme ve transfer öğrenme uygulamaları geliştirilirken jupyter ve spyder ide programları kullanılmıştır.

3.4. Python Programla Dili

Python, esnek ve dinamik bir programlama dilidir. Yorumlanabilir, nesne yönelimli, yüksek seviyeli bir programlama dilidir ve aynı zamanda açık kaynak kodlu olarak geliştirilmiştir. (Hlavatý vd., 2022). Büyük şirketler, üniversiteler hem bilimde hem de Ar-Ge çalışmalarında, Python'u yaygın bir şekilde kullanmaktadır. Dilin tasarımı, özellikle öğrenmeyi kolaylaştırmaya odaklanmıştır; bu da onu yeni başlayanlar veya ilkokul/lise seviyesindeki öğrenciler için ideal bir seçim yapar.

Python, basit ve öğrenmesi kolay bir sözdizimine sahiptir. Okunabilirliği yüksek bir dildir. Bu özellikleri sayesinde, program bakım maliyetini azdır. Bunun yanı sıra, Python, program modülerliğini ve kodun tekrar kullanılabilirliğini destekleyen modül ve paketleri beraberinde getirmektedir. Python'un yorumlayıcısı ve geniş standart kütüphanesi, kullanıcılarına ücretsiz ve özgür bir şekilde dağıtılabilecek şekilde, çeşitli platformlarda kullanılabilir hale getirir (Yılmaz, 2019).

Python, 1980'lerin sonlarında Hollanda'daki Centrum Wiskunde & Informatica'da Guido van Rossum tarafından, ABC programlama dilinin yerini alması amacıyla geliştirilmiştir (Sarı, 2023). Python'un açık kaynak olması geliştiricilerin dilin gelişimine katkı sağlamasını ve yeni özellikler eklemesini kolaylaştırmaktadır (Tiryaki ve Balaman, 2021). Bu da dilin sürekli güncellenmesini ve geliştirilmesini sağlamaktadır. Python, açık kaynaklı bir dil olup, genellikle Ruby, JavaScript ve Scheme gibi dillerle karşılaştırılır. Python'un diğer dillerden farkını belirleyen önemli özellikleri, basit kullanımı, yeni başlayanlar için kolay öğrenilebilir olması, çeşitli uygulamalara entegre edilebilmesi ve Mac, Windows ve Linux gibi birçok işletim sisteminde çalışabilmesidir. Diğer dillerle kıyaslandığında, Python sadece güçlü bir dil değil, aynı zamanda programcılara yaklaşık üç ila beş kat daha hızlı kod yazma olanağı sunan bir dil olarak öne çıkmaktadır.

TIOBE endeksi, farklı eğitim kaynakları, satıcılar ve mühendislerin arama sonuçlarına dayanarak popüler programlama dillerini sıralayan bir endekstir. Her ay, en çok aranan programlama dillerini belirleyen TIOBE Endeksi, sektördeki dillerin popüleritesini gösteren bir ölçüdür. Resim 3.16'da programlama dillerinin popülerlik oranlarını gösterilmektedir.

Jul 2024	Jul 2023	Change	Programming Language	Ratings	Change
1	1		 Python	16.12%	+2.70%
2	3	▲	 C++	10.34%	-0.46%
3	2	▼	 C	9.48%	-2.08%
4	4		 Java	8.59%	-1.91%
5	5		 C#	6.72%	-0.15%
6	6		 JavaScript	3.79%	+0.68%
7	13	▲	 Go	2.19%	+1.12%

Resim 3. 16. 2024 yılı programlama dilleri popülerlik oranları (TIOBE Index for July, 2024)

Python kullanım alanları:

- Web uygulamaları
- Pygame ve panda3d gibi kütüphanelerle oyun uygulamaları geliştirme
- Makine öğrenimi ve yapay zekâ projeleri geliştirme
- Pandas, numpy ve matplotlib gibi kütüphanelerle görselleştirme
- Masaüstü grafik kullanıcı ara yüzleri oluşturmak
- İnternet sitelerinden veri toplama işlemleri
- İş uygulamaları
- Ses ve video uygulamaları
- 2D ve 3D modeller oluşturabilir
- Mikro denetleyicilerle ve gömülü sistemlerle çalışabilir
- Görüntü İşleme
- Tıp ve farmakoloji
- Veri analizi ve modelleme
- Sinirbilim ve Psikoloji
- Astronomi
- Robotik, otonom araçlar, iş, meteoroloji gibi alanlarda kullanılabilir.

3.4.1. Kullanılan kütüphaneler

Yazılım kütüphanesi, yazılım geliştirme sürecinde kullanılan, belirli işlevleri veya görevleri yerine getiren hazır kod parçalarının bir araya getirildiği bir kaynaktır. Tez çalışmasında kullanılan opencv, numpy ,os ve matplotlib kütüphaneleri bu başlık altında incelenmiştir.

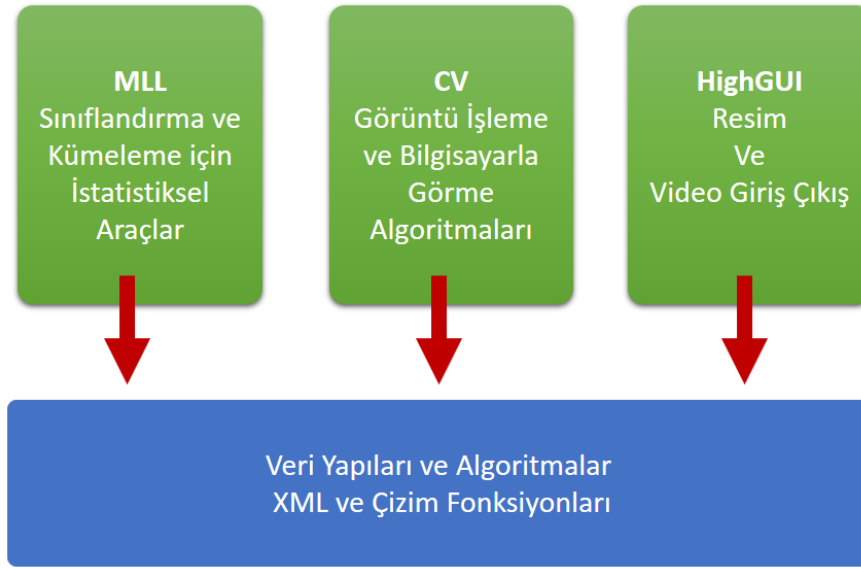
Opencv

OpenCV (Open Source Computer Vision) görüntü işleme alanında kullanılan açık kaynaklı bir kütüphanedir ve bu kütüphane 1999 yılında Intel tarafından geliştirilmeye başlanmıştır. Geliştirme sürecini Nvidia, AMD ve Google gibi birçok şirket ve topluluk desteklemiştir (Akkoca ve Buluş, 2021). OpenCV, bilgisayarlı görü alanında geniş bir temel sağlamayı ve ticari ürünlerde makine algısının kullanımını hızlandırmayı hedefleyen bir kütüphanedir. OpenCV, görüntü işleme, nesne tanıma ve makine öğrenimi gibi uygulamalarda geliştiricilere güçlü bir araç seti sunmayı hedeflemektedir. OpenCV'nin 47.000'den fazla kullanıcıdan oluşan bir topluluğu ve 18 milyondan fazla indirildiği tahmin edilmektedir. Bu kütüphane Google, Yahoo, Microsoft, Intel, IBM, Sony, Honda, Toyota gibi birçok köklü şirket tarafından kullanılmaktadır (Elkiran, 2020).

OpenCV'nin ilk sürümü 2000 yılında alfa sürümüyle piyasaya sürülmüştür. Bu kütüphane başlangıçta C programlama dilinde geliştirmiş ve algoritmalarının çoğu daha sonra C++ dilinde genişletilmiştir. OpenCV Açık kaynak kodlu olup BSD lisansı altında geliştirilmiştir. BSD lisansı, kütüphanenin projelerde ücretsiz olarak kullanılabileceği anlamına gelir. OpenCV platformdan bağımsız bir kütüphanedir, bu nedenle Windows, Linux, Android, Mac OS ve iOS dâhil olmak üzere birden fazla platformda çalışır. Ayrıca OpenCV uygulamaları, C++, C, Python, Java, Matlab ve topluluk tarafından geliştirilen farklı dillerle kullanılabilen wrapper'lar aracılığıyla Perl ve Ruby programlama dilleri kullanılarak kolaylıkla geliştirilebilmektedir. Bu esneklik sayesinde yapay zekâ yazılım geliştirmek isteyenler de OpenCV'den faydalanabilirler.

OpenCV, açık kaynaklı bir kütüphane ve gömülü sistem uygulamalarda gerçek zamanlı tanımayı sağlamak için yaygın olarak kullanılan bir görüntü işleme uygulamasıdır. OpenCV kütüphanesi hızlı ve pratik kodlamaya olanak sağlar ve birçok gömülü platform için derlenebilmektedir (Deepthi ve Sankaraiah, 2011). OpenCV kütüphanesi temel görüntü

işleme görevlerini içerir ve mantıksal ve aritmetik işlemler sağlar. Aynı zamanda nesne tespiti ve takibi gibi karmaşık işlemleri de içermektedir. OpenCV kütüphanesini oluşturan bileşenlerin yapısı Şekil 3.8’de verilmiştir.



Şekil 3. 8. OpenCV bileşen yapısı (Erişti, 2010)

OpenCV kütüphanesini oluşturan bileşenler: (Güler, 2018)

- Çekirdek(Core): OpenCV'nin temel fonksiyonlarını ve veri yapılarını içerir. Matris, nokta, boyut gibi veri yapılarının yanı sıra görseller üzerinde çizim yapmak ve XML işlemleri için gerekli fonksiyonları da sağlar.
- HighGui: Görüntü görüntüleme, pencere yönetimi ve grafik kullanıcı arayüzleri için gereken işlevselliği içerir. Önceki sürümlerde, dosya sistemi aracılığıyla görüntü dosyalarını okuma ve yazma yeteneği bulunmaktadır.
- Imgproc: Filtreleme operatörleri, kenar algılama, nesne algılama, renk alanı yönetimi, renk dönüşümü ve eşikleme gibi birçok fonksiyonu bünyesinde barındıran bir pakettir.
- Imgcodec'ler: Dosya sistemini kullanarak görüntüler ve videolar için okuma/yazma işlemlerini yapmak için gerekli fonksiyonları içermektedir.
- Videoio: Kameralara ve video cihazlarına erişim sağlar ve görüntü yakalama ve yazma işlemlerini gerçekleştirmek için gerekli işlevselliği içerir. Bu, kullanıcıların

kameralarından veya video cihazlarından görüntü veya video almasına, bu verileri işlemesine ve gerektiğinde kaydetmesine olanak tanımaktadır.

Python programlama dilinde opencv kütüphanesini kullanmak için “import cv2” şeklinde import edilir.

NumPy

NumPy (Sayısal Python olarak da bilinir), Python'da bilimsel hesaplama için temel bir kütüphanedir. Bu kütüphane çok boyutlu dizilerin ve matrislerin kullanımını kolaylaştırarak daha verimli sayısal hesaplamalara olanak sağlamaktadır. NumPy, ndarray veri yapısını (N boyutlu dizi) kullanmaktadır (Harris vd., 2020). Bu, veri depolamak için güçlü ve verimli bir veri yapısı sunmaktadır. Bu dizi yapısı Python'un listelerinden daha verimlidir ve vektörler ve matrisler üzerindeki işlemler için tasarlanmıştır.

NumPy, genellikle veri bilimi, makine öğrenimi, görüntü işleme ve diğer bilimsel veya mühendislik disiplinlerinde yaygın bir şekilde kullanılmaktadır. Veri analizi ve işleme için güçlü bir platform oluşturmak için numpy kütüphanesi, diğer Python kütüphaneleri (pandas, numpy, matplotlib gibi) ile entegre olarak da kullanılmaktadır.

Python programlama dilinde numpy kütüphanesini kullanmak için “import numpy as np” şeklinde import edilir.

Matplotlib

Matplotlib, Python'un en popüler veri görselleştirme kütüphanelerinden biridir. Bu kütüphane, John Hunter başta olmak üzere birçok katkıcı tarafından geliştirilmiş olup, dünya genelinde birçok bilim insanı tarafından kullanılan bu yazılıma büyük bir emek harcanmıştır (Hunter, 2008). Matplotlib, Python'daki veri bilimi yığnında önemli bir yer tutan bir veri görselleştirme kütüphanesidir ve NumPy, Pandas ve diğer ilgili kütüphanelerle uyum içinde çalışır.

Matplotlib, çoğunlukla grafiksel uygulama geliştirmeye odaklanmış bir Python kütüphanesidir. Bu kütüphane, etkileşimli iki ve üç boyutlu grafiklerin geliştirilmesinde kullanılır ve yüksek kaliteli görüntüler oluşturulabilmektedir (Ranjani vd., 2019). Kullanıcı dostu arayüzlere sahip olup geniş çeşitlilikte grafikler üretmekte oldukça esnek bir yapıya sahiptir. Şekil 3.9'da gösterildiği gibi çizgi, sütun, dağılım, pasta grafikleri ve histogramlar gibi grafik türlerini oluşturmak için idealdir. Ayrıca, güç spektrumu gibi özel uygulamaların geliştirilmesinde de kullanışlı bir yapıya sahiptir.



Şekil 3. 9. Matplotlib grafik örnekleri

Python programlama dilinde matplotlib kütüphanesini kullanmak için “import matplotlib.pyplot as plt ” şeklinde import edilir.

Os

Python'da işletim sistemi işlemlerini gerçekleştirmek için "OS" kütüphanesi kullanılır. Bu kütüphane dosya yönetimi, sistem bilgilerine erişim, ortam değişkenleri ve sistem komutları gibi birçok işlemi gerçekleştirmek için kullanılmaktadır.

"OS" kütüphanesinin kullanıldığı yerlere örnekler aşağıdadır:

- Dosya ve dizin oluşturma, silme, taşıma ve yeniden adlandırma.
- Çalışma dizinlerini değiştirme.
- Sistemdeki dosya ve dizinleri listeleme.
- Dosya ve dizinlerin varlığını kontrol etme.
- Sistem hakkındaki bilgilere erişme

- Sistem komutlarını çalıştırma.

Python programlama dilinde os kütüphanesini kullanmak için “import os” şeklinde import edilir.

Matplotlib, numpy, opencv, os kütüphaneleri bu çalışmanın gömülü sistem programlanması aşamasında kullanılmıştır. Bunların dışında bilgisayar ortamında bütün yaprakları hastalıklarını tespiti için ek olarak os.path, glob, sys, shutil modülleri de kullanılmıştır.

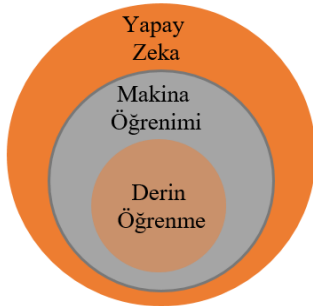


4. YAPAY ZEKÂ

Yapay zekâ, bilgisayar sistemlerinin insan benzeri düşünme yetilerini taklit etmeye çalışan bir teknoloji alanıdır. Derin Öğrenme ise yapay zekânın bir alt dalıdır ve büyük veri setlerinden öğrenme sağlayarak karmaşık modeller oluşturmaktadır. Tez çalışmasında sınıflandırma amacıyla kullanılan derin öğrenme modelleri bu kısımda incelenmiştir.

4.1. Derin Öğrenme (Deep Learning)

Teknolojinin gelişmesiyle birlikte görsel ve videoların artması, etiketsiz veri miktarının da artmasına neden olmuştur. Bu nedenle, bu verilerden anlamlı sonuçlar çıkarılması ve verilerin analiz edilmesi, önemli bir konu olarak ortaya çıkmaktadır (Luo vd., 2021). Ortaya çıkan sorunların çözümünü gerçekleştirmek amacıyla görüntü işleme algoritmalarından yararlanılmaktadır. Bu algoritmalar yararlı bilgilerin seçilmesi, metin, resim ve ses dosyalarına anlamlı içerik eklenmesi, yüz tanıma, plaka tespiti ve nesne tespiti gibi birçok alanda kullanılmaktadır. Ancak veri hacmi ve karmaşıklığı arttıkça geleneksel yöntemler artık yeterli olmamaktadır. Bu nedenle güçlü öğrenme yetenekleri ve yüksek doğruluk sağlayan derin öğrenme yöntemleri son yıllarda popüler hale gelmiştir. Derin öğrenme yöntemleri yapay sinir ağlarını temel alır ve büyük veri kümelerini analiz etmek ve karmaşık ilişkileri keşfetmek için kullanılır. Bu yöntemler, önceden belirlenmiş kurallara dayanmak yerine, insandan bağımsız, verilerin kendisinden öğrenerek sonuçlar üretmektedir. Yapay zekâ, makine öğrenmesi, derin öğrenme kronolojisi Şekil 4.1’de verilmiştir.



Şekil 4. 1. Yapay zekâ, makine öğrenmesi, derin öğrenme kronolojisi

Derin öğrenme, beyin nöronlarının yapısından etkilenecek geliştirilmiştir. Bu buluş, araştırmacıların birden fazla ağı eğitmek için daha fazla çaba harcamasına yol açmıştır (Bolouri, 1995). Derin öğrenme birden fazla katmandan oluşmaktadır. Oluşturulan her

katmanın girdisi kendinden önce gelen katmanın çıktısıdır. Giriş katmanı ilk katman, son katman ise çıkış katmanıdır. Bu iki katman arasındaki katmanlara gizli katman denmektedir (Deng ve Yu, 2013).

Görüntü işleme alanında derin öğrenme yöntemleri nesne algılama, görüntü sınıflandırma ve segmentasyon gibi görevleri yerine getirebilmektedir. Örneğin, bir görüntüdeki nesneler tanımlanabilir ve sınıflandırılabilir. Ayrıca görüntüyü parçalara ayırarak ve her bir parçanın ne olduğunu belirleyerek görüntüleri daha ayrıntılı bir şekilde analiz eder. Bununla birlikte, derin öğrenme yöntemlerinin kullanımında çeşitli zorluklar vardır. İlk olarak, derin öğrenme modelleri genellikle büyük miktarda veri gerektirir ve öğrenme süreci zaman alıcıdır. Buna ek olarak, modellerin karmaşıklığı nedeniyle hesaplama gücü ve bellek gereksinimleri fazladır.

4.1.1. Evrışimsel sinir ağları

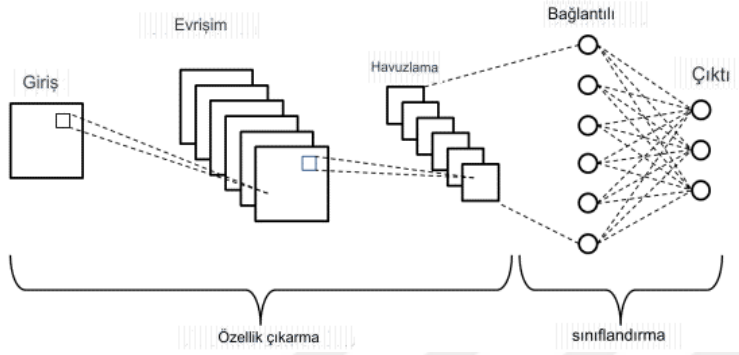
Nöronlar arasındaki bağlantıyı hayvanlardan almış ileri beslemeli bir yapay sinir ağıdır. CNN sınıflandırma işleminin direk olarak görüntüler ve videolardan öğrenen popüler bir model olarak öne çıkmaktadır (Cireşan vd., 2012). Evrışimli sinir ağları, verilerden özellikleri analiz etmek ve çıkarmak için nöronal ağırlıklar ve öğrenme algoritmaları kullanır.

Evrışimli sinir ağları özellikleri:

- Segmentasyon veya arka plandan nesne ayırma gibi ön işleme adımları olmadan gelişmiş görevleri gerçekleştirir.
- Geleneksel görüntü tanımlayıcılarına kıyasla yüksek performans sunar.
- Bu yeni yaklaşım, daha karmaşık ve zorlu görevlerin üstesinden gelmek için esneklik sağlar.
- Nesnelerin daha iyi anlaşılması daha doğru sonuçlar elde edilmesini sağlar.
- Veri tabanının boyutuna ve karmaşıklığına bağlı olarak ölçeklenebilirlik sunar.
- Yüksek doğruluk ve düşük hata oranları gibi avantajlar sağlar.

- Öğrenme sürecindeki veri gereksinimlerini azaltır ve daha az etiketli veri kullanımına olanak sağlar.
- Görüntü işleme alanında yeni olanaklar yaratır ve daha gelişmiş uygulamaların geliştirilmesine katkı sağlar.

Evrişimli sinir ağı(CNN) katmanlardan oluşmaktadır. Bu katmanlar Şekil 4.2’de verilmiştir.



Şekil 4. 2. CNN katmanları

CNN Katmanları:

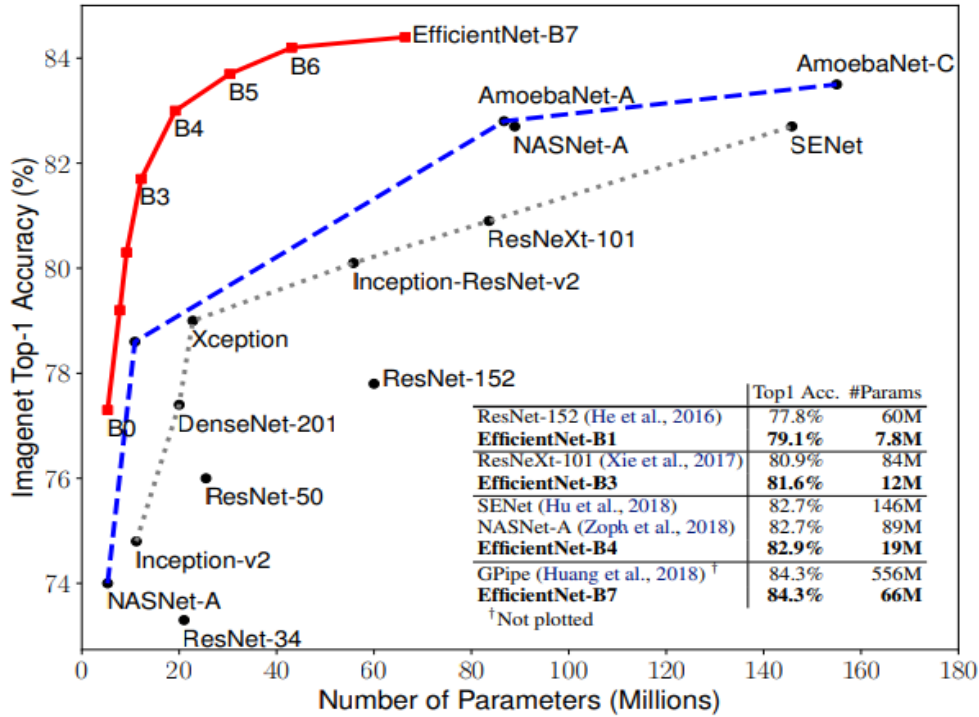
- **Konvolüsyon katmanı:** Giriş verilerine belirli filtreler veya çekirdekler uygulayarak farklı özelliklerin tanınmasını sağlar. Konvolüsyon işlemi, filtrelerin giriş verileri üzerinde kaydırılması ve çarpma yoluyla bir özellik haritası oluşturulmasıyla gerçekleştirilir.
- **Relu (Rectified Linear Unit) aktivasyon katmanı:** konvolüsyon sonuçlarına doğrusal olmayan bir aktivasyon fonksiyonu uygular. Relu negatif değerleri sıfıra dönüştürerek ağı daha iyi öğrenmesini sağlar.
- **Havuzlama katmanı:** konvolüsyon sonucunu en aza indirmek ve önemli bilgileri korumak için kullanılır. En yaygın kullanılan havuzlama işlemi maksimum havuzlamadır. Bu işlem, her bölgeden en büyük değeri alarak özellik haritasını en aza indirir.
- **Tam bağlantılı katman:** Havuzlama katmanından elde edilen özellik haritası, sınıflandırma işleminin gerçekleştirildiği tam bağlantılı katmandan beslenir. Bu

katmanlarda nöronlar arasındaki tüm bağlantılar bulunur ve çıktılar hesaplanmaktadır.

4.1.2. Önceden eğitilmiş evrimsel sinir ağları mimarileri

Convolutional neural networks (CNN), evrimsel sinir ağlarına dayalı birçok mimari geliştirilmiştir. Bu tez çalışmasında önceden eğitilmiş (transfer learning) modellerden vgg-16, resnet-50, efficientNetB6 mimarileri kullanılmıştır. Transfer öğrenme, önceden eğitilmiş bir modelin özellik çıkarma kısmını alır ve hedef görevin sınıflandırma kısmını yeniden eğitir. Başka bir deyişle, kaynak görevde öğrenilen özellikler hedef görevde kullanılarak daha az veriyle daha iyi sonuçlar elde edilir.

Transfer öğrenme, bir görevde kazanılan bilginin diğer görevlere aktarılmasını sağlar (Hunter, 2018). Önceden eğitilmiş modeller genellikle büyük veri kümeleri üzerinde eğitilir ve çeşitli özellikler öğrenir. Bu özellikler genel nesne tanıma, duygu analizi, yüz tanıma ve doğal dil işleme (NLP) gibi birçok görev için kullanılır. Model boyutu ve imagenet doğruluk oranları Şekil 4.3'te verilmiştir.



Şekil 4. 3. Model boyutu ve imagenet doğruluk oranları (Wang vd., 2020)

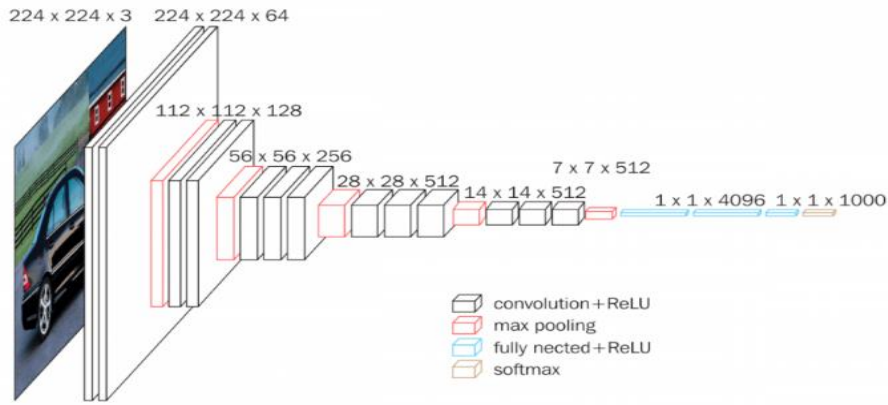
Transfer öğrenmenin avantajları:

- Daha az veriyle eğitim: Transfer öğrenimi, yetersiz veriyle bile başarılı sonuçlar elde edebilir.
- Daha hızlı eğitim: Önceden eğitilmiş modelin özellik çıkarım kısmı kullanıldığından sadece sınıflandırma kısmı yeniden eğitilir. Bu, eğitim süresini azaltır.
- Daha iyi genelleme: Önceden eğitilmiş model genel özellikleri öğrenebildiğinden, hedef görevdeki benzer özellikleri tanımayı da öğrenebilir.

VGGNet

Oxford Üniversitesi'nden K. Simonyan ve A. Zisserman'ın “Very Deep Convolutional Networks for Large-Scale Image Recognition” adlı araştırma makalesinde, 1000 adet sınıf ve 14 milyon görüntü kullanılarak oluşturulan veri setinde ImageNet'in yarışmasında kayda değer bir başarı elde etmiştir. Modelleri %92'lik bir başlangıç doğruluğu elde etmiş ve ilk beş model arasına girmeyi başarmıştır (Kılıç, 2023). Bu model 224 x 224 RGB görüntüsü kullanılmaktadır. Model, 16 veya 19 katmanlı olmak üzere iki farklı varyasyonda gelir. Her katman, 3x3 boyutunda evrişimli filtreler ve ardından Relu aktivasyon fonksiyonu ile birleştirilmiş iki katmanlı bir evrişimli bloktan oluşmaktadır. Ardından, 2x2 boyutunda maksimum havuzlama (max pooling) işlemi uygulanır. Bu yapı, daha derin ve daha karmaşık modellerin oluşturulmasına olanak tanır.

VGGNet, evrişimli sinir ağlarının derinliğinin performans üzerinde önemli bir etkisi olduğunu göstermiştir (Prasetyo vd., 2022). Ayrıca, VGGNet, önceden eğitilmiş ağırlıkların kullanılmasıyla transfer öğrenme için de popüler bir seçenek olmuştur. Bu tez çalışmasında VGGNet 16 modeli kullanılmıştır. Kullanılan mimarinin yapısı Şekil 4.4'te, oluşturulan model ise şekil 4.5'te verilmiştir.



Şekil 4. 4. VGG16 mimarisi (Varshney, 2020).

Model: "model"

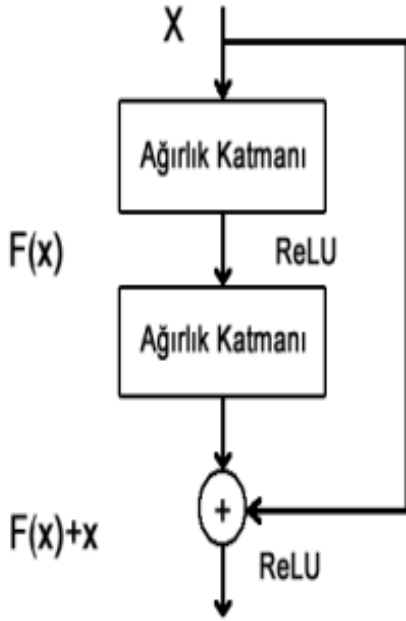
Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, 128, 128, 3)]	0
block1_conv1 (Conv2D)	(None, 128, 128, 64)	1792
block1_conv2 (Conv2D)	(None, 128, 128, 64)	36928
block1_pool (MaxPooling2D)	(None, 64, 64, 64)	0
block2_conv1 (Conv2D)	(None, 64, 64, 128)	73856
block2_conv2 (Conv2D)	(None, 64, 64, 128)	147584
block2_pool (MaxPooling2D)	(None, 32, 32, 128)	0
block3_conv1 (Conv2D)	(None, 32, 32, 256)	295168
block3_conv2 (Conv2D)	(None, 32, 32, 256)	590080
block3_conv3 (Conv2D)	(None, 32, 32, 256)	590080
block3_pool (MaxPooling2D)	(None, 16, 16, 256)	0
block4_conv1 (Conv2D)	(None, 16, 16, 512)	1180160
block4_conv2 (Conv2D)	(None, 16, 16, 512)	2359808
block4_conv3 (Conv2D)	(None, 16, 16, 512)	2359808
block4_pool (MaxPooling2D)	(None, 8, 8, 512)	0
block4_pool (MaxPooling2D)	(None, 8, 8, 512)	0
block5_conv1 (Conv2D)	(None, 8, 8, 512)	2359808
block5_conv2 (Conv2D)	(None, 8, 8, 512)	2359808
block5_conv3 (Conv2D)	(None, 8, 8, 512)	2359808
block5_pool (MaxPooling2D)	(None, 4, 4, 512)	0
flatten (Flatten)	(None, 8192)	0
dense (Dense)	(None, 10)	81930
dense_1 (Dense)	(None, 2)	22

Total params: 14796640 (56.44 MB)		
Trainable params: 81952 (320.12 KB)		
Non-trainable params: 14714688 (56.13 MB)		

Şekil 4. 5. VGG16 mimarisi ile oluşturulan model

ResNet

ResNet (Residual Network), derin öğrenme (deep learning) alanında önemli bir ağ mimarisi ve yöntemidir. ResNet, Kaiming He, Xiangyu Zhang, Shaoqing Ren ve Jian Sun tarafından 2015 yılında geliştirilmiş ve ImageNet sınıflandırma yarışmasında büyük başarı elde etmiştir (He vd., 2016). Bu mimari, geleneksel mimarilerden biri olan VGGNet'ten farklı olarak mikro mimari modüllere dayanmaktadır. Resnet, VGGNet'e göre birçok avantajı vardır. Daha derindir, ancak daha küçük bir ağ boyutuna ve daha az parametreye sahiptir. ResNet, aşırı derin sinir ağlarının eğitimi ve performansını iyileştirmek için özgün bir yaklaşım sunmaktadır. ResNet, bir önceki katmanın çıktısını bir sonraki katmanın girişine eklemek suretiyle oluşturulan "residual bağlantılar" veya "skip connections" kullanır. Bu, ağın daha derin hale gelmesine izin verir, çünkü bilgi kaybını önler ve aşırı uyum sorununu azaltmaktadır. Başka bir deyişle, ResNet, katmanlar arasındaki hataların doğrudan iletilmesine izin vererek, daha kolay eğitim ve daha iyi performans sağlar. ResNet, diğer bazı derin sinir ağı yapılarına kıyasla daha derin ağlar kullanır. Özellikle "ResNet-50" ve "ResNet-101" gibi modeller, yüzlerce katman içerebilir. Bu, ağın daha karmaşık özellikleri öğrenmesine olanak tanır. Resnet mimarisi ile derinlik artsa da öğrenme hızlı bir şekilde gerçekleşmektedir. Şekil 4.6'da residual öğrenme katmanları verilmiştir.



Şekil 4. 6. Artık (Residual) Öğrenme

Bu tez çalışmasında kullanılan ResNet50, 50 ağırlık katmanından oluşan bir mimariye sahiptir. Model 224 x 224 RGB görüntüsü kullanılmaktadır. ResNet, Vgg-16 ve Vgg-19'dan çok daha derin olmasına rağmen, tam bağlantılı katmanlar yerine global ortalama havuzlama yöntemi kullanarak model boyutunu 102 MB seviyesine düşürmüştür. Bu sayede model boyutu önemli ölçüde daha küçük hale gelmektedir. Bu çalışmada kullanılan ResNet50 mimarisi ile gerçekleştirilen modelin yapısı Şekil 4.7'da verilmiştir.

Model: "model_1"

Layer (type)	Output Shape	Param #	Connected to
input_2 (InputLayer)	(None, 128, 128, 3)	0	[]
conv1_pad (ZeroPadding2D)	(None, 134, 134, 3)	0	['input_2[0][0]']
conv1_conv (Conv2D)	(None, 64, 64, 64)	9472	['conv1_pad[0][0]']
conv1_bn (BatchNormalizati on)	(None, 64, 64, 64)	256	['conv1_conv[0][0]']
conv1_relu (Activation)	(None, 64, 64, 64)	0	['conv1_bn[0][0]']
pool1_pad (ZeroPadding2D)	(None, 66, 66, 64)	0	['conv1_relu[0][0]']
pool1_pool (MaxPooling2D)	(None, 32, 32, 64)	0	['pool1_pad[0][0]']
...			
conv5_block3_add (Add)	(None, 4, 4, 2048)	0	['conv5_block2_out[0][0]', 'conv5_block3_3_bn[0][0]']
conv5_block3_out (Activati on)	(None, 4, 4, 2048)	0	['conv5_block3_add[0][0]']
flatten_1 (Flatten)	(None, 32768)	0	['conv5_block3_out[0][0]']
dense_2 (Dense)	(None, 10)	327690	['flatten_1[0][0]']
dense_3 (Dense)	(None, 2)	22	['dense_2[0][0]']
=====			
Total params: 23915424 (91.23 MB)			
Trainable params: 327712 (1.25 MB)			
Non-trainable params: 23587712 (89.98 MB)			

Şekil 4. 7. Resnet50 mimarisi ile oluşturulan model

EfficientNet

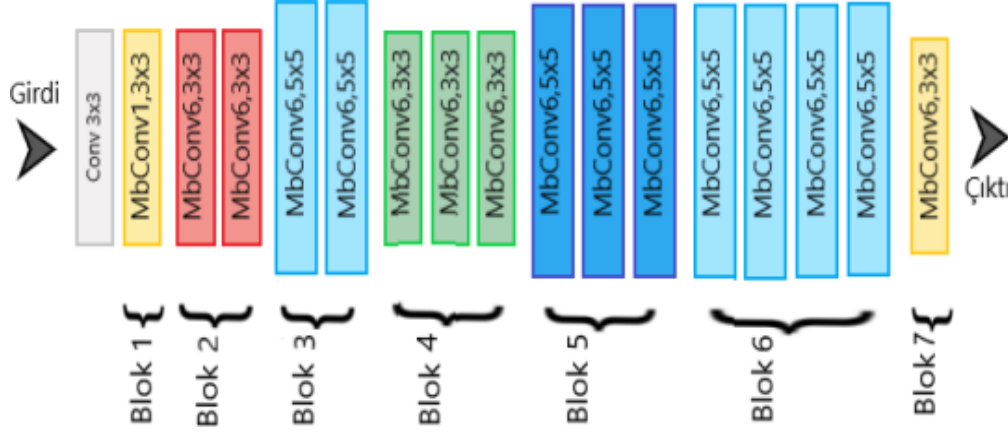
EfficientNet, ölçeklenebilir ve yüksek performanslı yapay sinir ağı mimarilerinden biridir. 2019 yılında t Mingxing Tan ve Quoc V. Le tarafından "Rethinking Model Scaling for

Convolutional Neural Networks” makalesinde yayınlanarak tanıtılmıştır. Geliştirilen bu ağ mimarisi, özellikle görüntü işleme görevlerinde yüksek doğruluk sağlayan ve aynı zamanda hesaplama kaynaklarını daha etkili bir şekilde kullanabilen bir yapıyla dikkat çekmektedir (Tan ve Le, 2019).

Tan ve Le ağın sadece derinliğini, genişliğini ve çözünürlüğünü azaltmak yerine, belirli bir ölçek dâhilinde ağın derinliğini, genişliğini ve çözünürlüğünü birlikte azaltmanın ağın performansı üzerinde olumlu bir etkiye sahip olduğunu gözlemlemişlerdir (Türkmen, 2021). Sonuç olarak, ağın derinliğinin, genişliğinin ve çözünürlüğünün eşit oranda artırılmasına veya azaltılmasına izin veren yeni bir yaklaşım ortaya çıkmıştır. EfficientNet, B0'dan başlayıp B7'de biten 8 farklı modelden oluşan ve modeller büyüdükçe kullanılan parametre sayısı artmakla beraber performansı da artmaktadır.

Derin öğrenme mimarilerinde, daha küçük ve daha verimli modeller oluşturarak performansı artırmak temel hedef olmuştur. EfficientNet modeli, ESA mimarilerinden farklı olarak modelin boyutunu küçültürken derinlik, genişlik ve simetriyi ölçerek daha iyi sonuçlar elde etmeyi amaçlamaktadır. Kullanılabilirlik ölçüm sürecinin ilk adımı, kullanıcı arayüzü altındaki farklı değişkenler arasındaki ilişkileri tanımlamak için bir ızgara arama algoritması kullanmaktır. Bu süreç, doğru uyumu belirlemek için yükseklik, genişlik ve çözünürlük arasındaki doğru dengeyi bulur. Bu katsayılar daha sonra ağ tabanını boyut aralığına ölçeklendirmek için kullanılır. Bu şekilde, EfficientNet, kaynakların daha iyi kullanıldığı ve daha kompakt modellerin elde edildiği bir yaklaşım sunmaktadır.

EfficientNet'in dikkate değer bir özelliği de MBConv (Mobil İkilik Evrişim) katmanlarıdır. Bu katmanlar, ResNet bloklarının aksine, kanal sayısını önce genişletip sonra sıkıştırarak çalışırlar (Sandler vd., 2018). Bu sayede hesaplama miktarını azaltırken aynı zamanda atlama bağlantıları ile zenginleştirilmiş kanal katmanlarını birbirine bağlarlar. Şekil 4.8 EfficientNetB0 modelinin bu özelliğini göstermektedir. Bu yöntem, hesaplama kaynaklarını daha verimli bir şekilde kullanmaya ve ağın performansını artırmaya yardımcı olur.



Şekil 4. 8. EfficientNet mimarisi

Bu tez çalışmasında EfficientNet-B6 mimarisi kullanılmıştır. Model 669 katmandan oluşmakta ve RGB görüntü boyutu 528x528 çözünürlüktedir. Yapılan çalışmada kullanılan EfficientNet-B6 mimarisi ile gerçekleştirilen modelin yapısı Şekil 4.9'de verilmiştir.

Model: "model"

Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	[(None, 128, 128, 3)]	0	[]
rescaling (Rescaling)	(None, 128, 128, 3)	0	['input_1[0][0]']
normalization (Normalization)	(None, 128, 128, 3)	7	['rescaling[0][0]']
rescaling_1 (Rescaling)	(None, 128, 128, 3)	0	['normalization[0][0]']
stem_conv_pad (ZeroPadding2D)	(None, 129, 129, 3)	0	['rescaling_1[0][0]']
stem_conv (Conv2D)	(None, 64, 64, 56)	1512	['stem_conv_pad[0][0]']
...			
top_conv (Conv2D)	(None, 4, 4, 2304)	1327104	['block7c_add[0][0]']
top_bn (BatchNormalization)	(None, 4, 4, 2304)	9216	['top_conv[0][0]']
top_activation (Activation)	(None, 4, 4, 2304)	0	['top_bn[0][0]']
flatten (Flatten)	(None, 36864)	0	['top_activation[0][0]']
dense (Dense)	(None, 10)	368650	['flatten[0][0]']
dense_1 (Dense)	(None, 2)	22	['dense[0][0]']

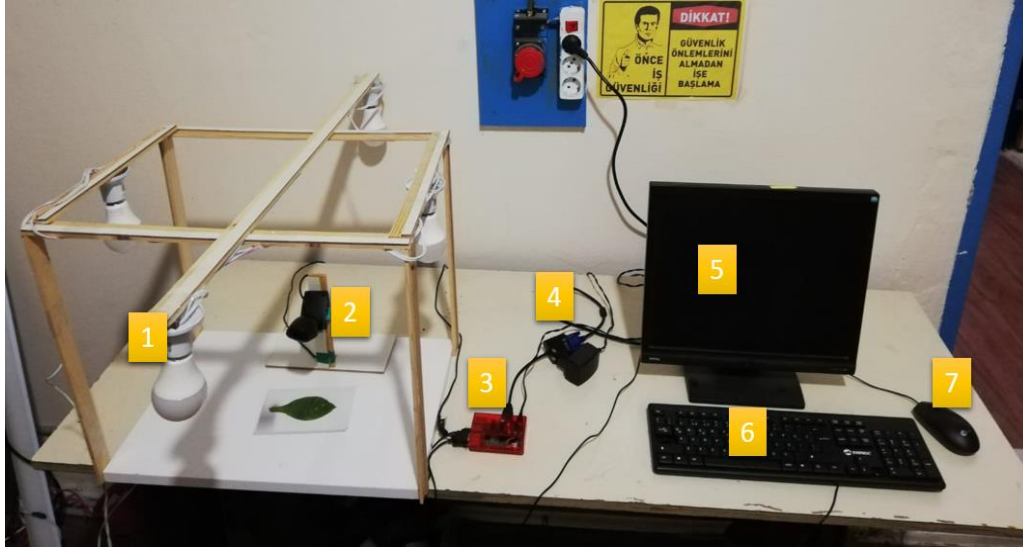
=====
Total params: 41328815 (157.66 MB)
Trainable params: 368672 (1.41 MB)
Non-trainable params: 40960143 (156.25 MB)

Şekil 4. 9. EfficientNet-B6 mimarisi ile oluşturulan model

5. YÖNTEM

Gerçekleştirilen tez çalışmasında modelin tasarlanması ve yaprak üzerinde meydana gelen hastalıklı alanların tespit edilmesi için uygulanan adımlar bu kısımda yer almaktadır.

5.1. Prototip Tasarımı



Resim 5. 1. Gerçek zamanlı tütün yaprak hastalık tespiti yapan model

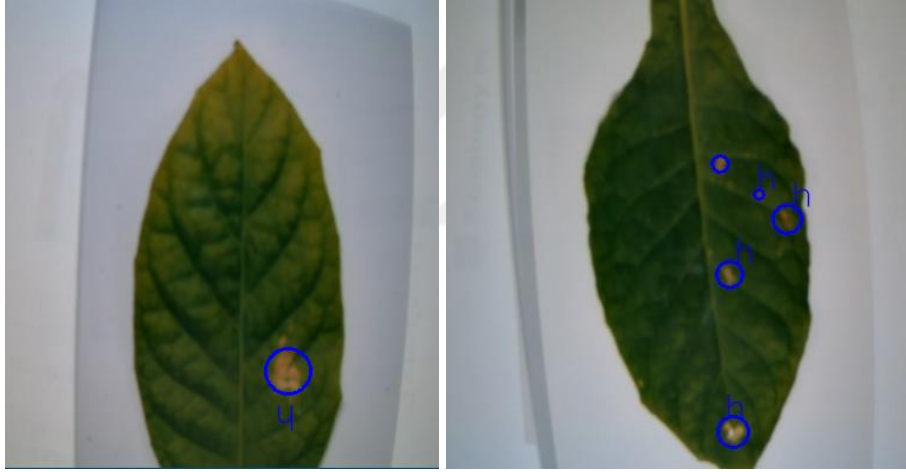
Kullanılan donanımlar;

1. 4 Adet LED lamba
2. Kaan web kamera
3. Raspberry Pi 3 Model B+
4. Hdm1 vga dönüştürücü
5. Monitör
6. Klavye
7. Fare

Resim 5.1’de tütün yaprakları üzerindeki hastalıkları gerçek zamanlı olarak tespit eden bir örnek model tasarımı görülmektedir. Bu model tasarımı, dört ayağa yerleştirilmiş ve her köşesinde aydınlatmayı sağlamak için led lamba kullanılmıştır. Aydınlatma, yapraklar

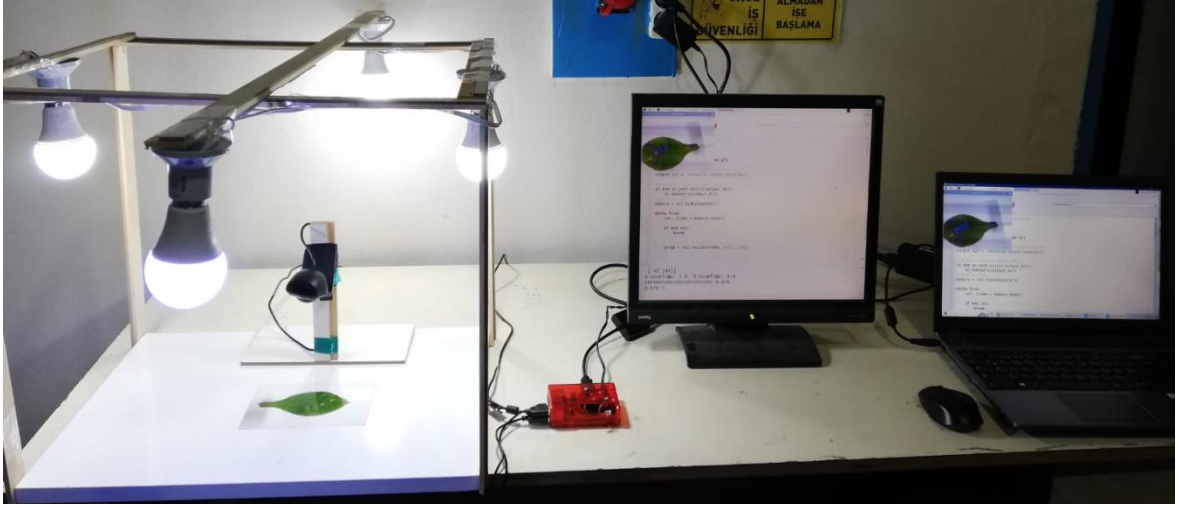
üzerindeki hastalıklı bölgelerin doğru bir şekilde tespiti için çok önemlidir. Yeterli aydınlatmanın olmadığı alanlarda, modelin doğruluk oranının azalma eğiliminde olduğu gözlemlenmiştir.

Model yüzeyinde tütün yapraklarının büyüklüğüne ve ışığın konumuna göre yüksekliği ve konumu değiştirilebilen bir USB kamera bulunmaktadır. Bu kamera aracılığıyla alınan görüntüleri Raspberry Pi 3 kartına aktarmaktadır. Alınan bu görüntüler, görüntü işleme teknikleri kullanılarak işlenir. Bu işleme süreci, hastalıklı bölgeleri belirlemek ve vurgulamak için özel olarak tasarlanmış algoritmalar içermektedir. Görüntü işleminin sonucunda Raspberry Pi 3 kartına bağlı olan ekranda, gerçek zamanlı olarak tespit edilen hastalıklı alanlar görsel olarak görüntülenir. Bu alanlar Resim 5.2’de görüldüğü gibi mavi daire içerisinde alınarak h harfi gösterilmiştir.



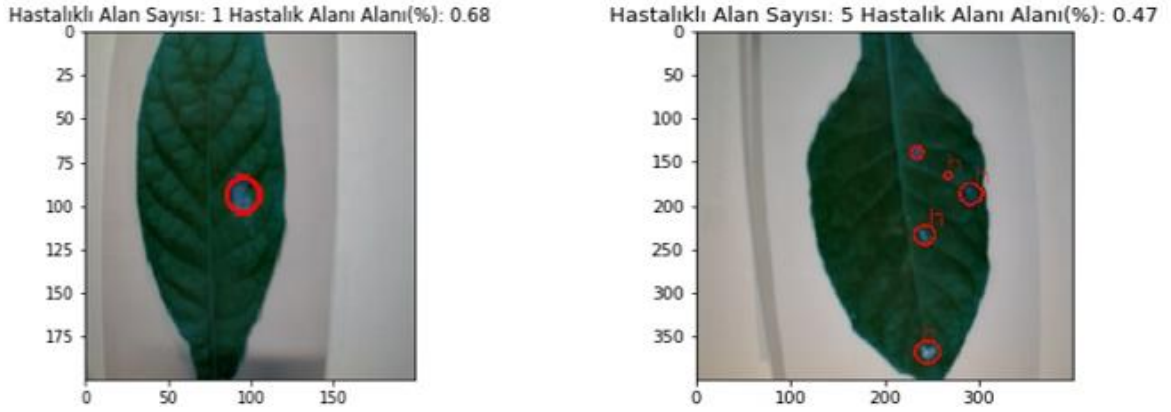
Resim 5. 2. Hastalıklı alanların mavi daire içerisinde gösterilmesi

Raspberry Pi kartına HDMI bağlantısı üzerinden ekran eklenerek gerçek zamanlı hastalık tespiti ekran üzerinden takip edilmiştir. Ancak bazı durumlarda harici monitör bağlamak mümkün veya uygun olmayabilir. Bu durumda ya da gerçek zamanlı görüntüyü aynı ağ üzerinden başka bir bilgisayarda izlemek istenildiğinde, görüntüyü Raspberry Pi 3 kartından aynı ağdaki başka bir bilgisayara aktaran VNC programı yardımıyla izlenebilir. Resim 5.3’de kameradan gelen görüntüler Raspberry Pi 3 kartında işlendikten sonra HDMI bağlantısı ile monitöre, VNC programı ile de aynı ağdaki dizüstü bilgisayara aktararak görüntüler izlenmektedir.



Resim 5. 3. Gerçek zamanlı tütün yaprak hastalık tespiti yapan modelin çalışması

Ayrıca, model tarafından tespit edilen hastalıkların sayısı ve kapladığı alan bilgisi belirlenen klasöre kaydedilmektedir. Klasöre kayıt edilen bu görüntüler kırmızı daire için de ve üzerinde h harfi ile gösterilmektedir. Bu, elde edilen verilerin daha sonradan analiz edilmesi ve değerlendirilmesi için kullanılmasını sağlamaktadır. Böylece, tarım uzmanları veya kullanıcılar, tütün bitkilerindeki hastalıkların yayılma durumu hakkında bilgi edinebilir ve gerekli önlemleri alabilirler. Şekil 5.1’de hastalıklı yaprak ile ilgili bilgiler verilmiştir.



Şekil 5.1. Tütün yaprak hastalıklı alan sayısı ve yüzdelik dilimi

Gerçek zamanlı çalışan bu sistemde, Raspberry Pi kartının performansı önemli bir faktördür. Raspberry Pi kartının özellikleri ne kadar gelişmişse, sistem o kadar hızlı çalışmaktadır. Bununla birlikte, Raspberry Pi kartının sıcaklığı da performans üzerinde etkilidir. Bu nedenle, kartın etkin bir şekilde soğutulması, uzun süreli ve güvenilir performans için çok önemlidir.

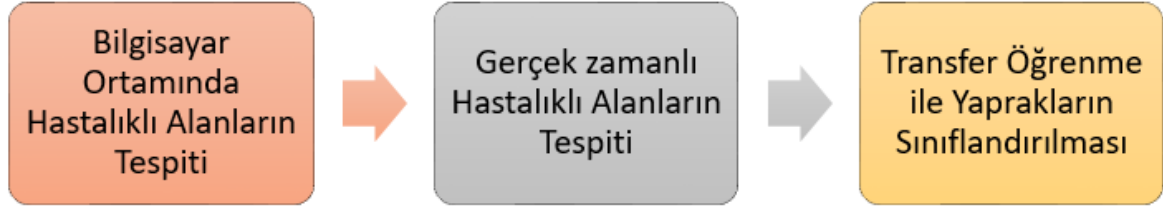
Sistem performansını etkileyen bir diğer önemli faktör, USB kamerasından alınan görüntülerin kalitesi ve en-boy oranıdır. Görüntü kalitesi, hastalıkların doğru şekilde tespit edilmesi için önemli bir faktördür. Aynı zamanda, görüntü boyutu da Raspberry Pi kartının performansını önemli derecede etkilemektedir. Görüntü boyutu 300x300'ün üzerine çıktığında, sistem performansında belirgin bir yavaşlama gözlemlenmiştir. Bu nedenle, görüntü boyutu 200x200 olarak tercih edilmiştir. Bu performans iyileştirmeleri, sistemin daha etkili ve hızlı çalışmasını sağlamak adına yapılmıştır. Raspberry Pi kartının en uygun sıcaklıkta çalışması ve uygun görüntü parametrelerinin seçilmesi, sistemin performansı için önemlidir.

Tütün bitkisinin ekilmesi, dikilmesi ve hasat süreci, iklim, tütün türü ve yetiştirildiği bölgeye bağlı olarak değişiklik göstermektedir. Ancak genel olarak tütün hasadı, bitkinin olgunlaştığı dönemde gerçekleştirilir. Tütün bitkisi büyüdükçe yaprakları gelişir ve olgunlaşır. Hasat genellikle yapraklar tamamen olgunlaştığında yapılır, çünkü bu dönemde tütün yaprakları en iyi kaliteye ulaşmaktadır. Bu çalışmada tütün yapraklarının toplandığı Bafra İlçesinde tütün yapraklarının toplanması temmuz ayının başında başlar ve eylül ayının sonuna kadar devam eder. Bitki her gün büyür yaprakları gelişir ve sürekli olarak hasat edilir. Gerçekleştirilen gerçek zamanlı tütün yaprakları hastalık tespiti çalışmasında yaprakların test ve deneme süreçleri tütün hasat zamanı sonrasında yapılmıştır. Bu problemi aşmak amacıyla hasat döneminde çekilen resimler, gerçek boyutlarına yakın bir şekilde bastırılmıştır. Model üzerinde deneme ve test süreçleri bu basılan fotoğraflarla gerçekleştirilmiştir.

Bu tasarım, tütün yaprakları hastalık tespiti sürecini otomatikleştirmek ve hızlandırmak için kullanılabilecek etkili bir model sunmaktadır.

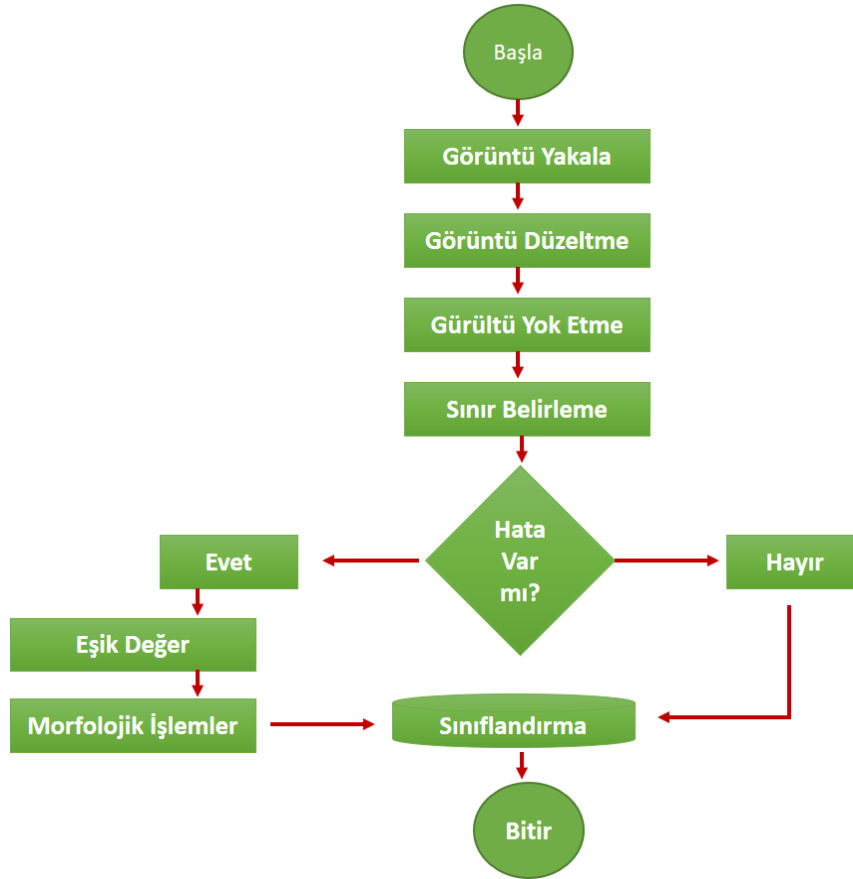
Gerçekleştirilen tez çalışmasında tütün yapraklarındaki hastalıklı bölümlerin tespit edilmesi ve yaprakta kapladığı alan hesaplanması işlemleri bu kısımda anlatılmıştır. Kullanılan yöntem ve teknikler, yazılımlar bu kısımda gerçekleştirilmiştir. Tütün yapraklarında meydana gelen hastalıklar ilk olarak bilgisayar ortamında işlenerek hesaplanmıştır. Daha sonra gömülümü sistem tasarımı ile gerçek zamanlı olarak hastalıklı alanlar tespit edilmiş ve hastalıklı yapraklar ayrılmıştır. Son olarak transfer öğrenme yöntemleriyle tütün yapraklarının hastalıklı ve sağlam şeklinde sınıflandırma işlemi gerçekleştirilmiştir.

5.2. Yöntem Adımları



Şekil 5.2. Gerçekleştirilen yöntem aşamaları

Tütün yaprakları tespiti için gerçekleştirilen adımlar Şekil 5.2’de gösterilmiştir. Gömülü sistem tasarımının kodlanması ve bilgisayar ortamında yapılan çalışmanın kodlanması python programlama dili ile yapılmıştır. Bu kodlamanın blok şeması Şekil 5.3’te gösterilmiştir.



Şekil 5.3. Akış diyagramı

5.2.1. Kameradan görüntülerin alınması

Python'da OpenCV kütüphanesini kullanarak kamera görüntüsü almak veya kaydedilmiş bir video dosyasını kullanmak için, cv2.VideoCapture() metodunu kullanılır (VideoCapture

Class Reference, b.t.). Dâhili bir kamera kullanılacaksa `cv2.VideoCapture(0)`, harici bir USB kamera kullanılacaksa `cv2.VideoCapture(1)` metodu kullanılmalıdır.

5.2.2. Görüntüleri yeniden boyutlandırma

Raspberry Pi kartına usb ile bağladığımız kaan marka kamera 1920*1080 boyutlarında video kayıt yapmaktadır. Kameradan gelen görüntünün boyutunu 200*200 olarak ayarlanmıştır. Kameradan gelen görüntü ne kadar büyük olursa Raspberry pi kartının performansı da bir o kadar düşmektedir.

5.2.3. Gri tona dönüşüm

`Cv2.cvtColor()` işlevi, OpenCV kütüphanesinde renk dönüştürme işlemlerini gerçekleştirmek için kullanılmaktadır. Bu işlev, dönüştürülmek istenen görüntüyü ve hedef renk uzayını parametre olarak alır. Bu çalışmada görüntü HSV (ton, doyma, parlaklık) renk uzayından BGR (mavi, yeşil, kırmızı) renk uzayına dönüştürülmüştür.

5.2.4. Eşik(Threshold) değer tespiti

Eşik değer, genellikle bir görüntüdeki pikselleri iki gruba ayırmak için kullanılan bir değerdir. Eşikleme, piksellerin belirli bir eşik değerinin üstünde veya altında olup olmadığını belirlemek için kullanılmaktadır. Bu işlem görüntü işlemede önemli bir adımdır. Örnek olarak, eşikleme bir görüntüdeki belirli parlaklık seviyelerini veya piksel değerlerini ayarlamak için kullanılabilir. Eşik değeri genellikle görüntüdeki belirli bir yoğunluk seviyesini temsil eder ve pikseller bu eşik değerine göre gruplandırılır.

5.2.5. Morfolojik işlemler

Morfolojik işlemler, görüntü işleme alanında kullanılan bir dizi matematiksel operasyonlardır. Bu işlemler, görüntülerdeki şekil ve yapıları değiştirmek veya vurgulamak için kullanılmaktadır. Genellikle yapısal eleman veya kernel adı verilen bir matris kullanılarak gerçekleştirilir. Bu yapısal elemanlar, işlemin türüne ve uygulanacak şekle göre tanımlanmaktadır.

Morfolojik operasyonları python'da kullanmak için önce `kernel = np.ones((5, 5), np.uint8)` kodu ile bir çekirdek oluşturulur. Daha sonra erozyon işlemi için `erosion = cv2.erode(image, kernel, iterations=1)` kodu kullanılır. Genişletme (dilation) işlemi için `dilation = cv2.dilate(image, kernel, iterations=1)` kodu uygulanır. Açma (opening) işlemi için `opening = cv2.morphologyEx(image, cv2.MORPH_OPEN, kernel)` kodu kullanılır ve kapama (closing) işlemi için `closing = cv2.morphologyEx(image, cv2.MORPH_CLOSE, kernel)` kodu kullanılır.

5.2.6. Kontur(Contours) belirleme

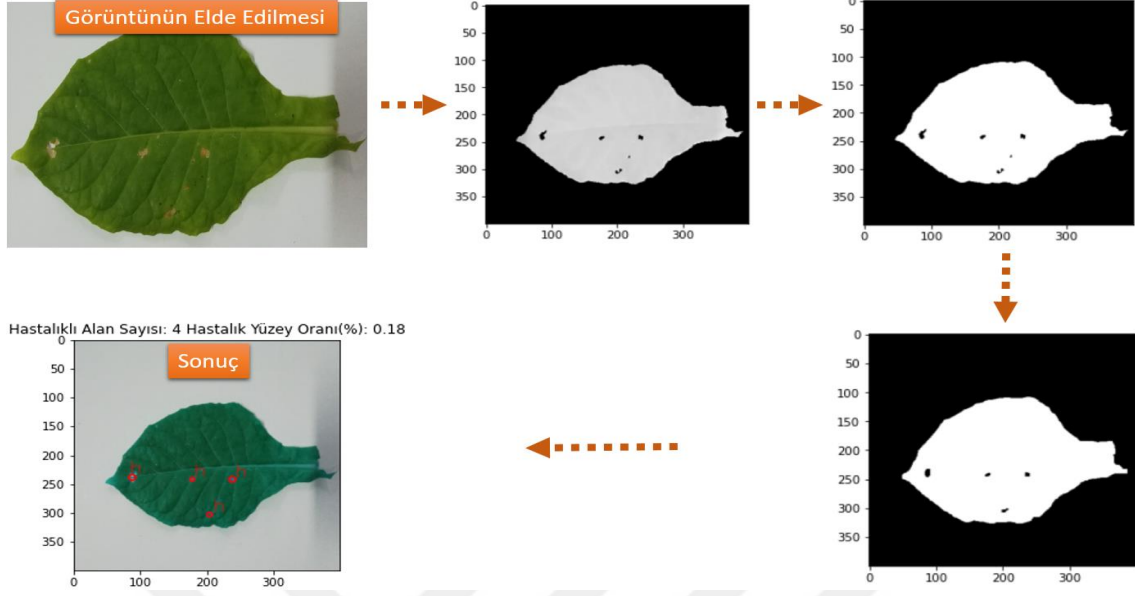
Kontur, bir nesnenin sınırını oluşturan tüm noktaların birleştirilmesiyle oluşmaktadır. Genellikle aynı renk veya yoğunluğa sahip piksellerin sınırlarını temsil eder. Kontur bir nesnenin dış sınırlarını veya kenarlarını temsil eden bir dizi noktalardan oluşmaktadır (Chakraborty, 2021).

Ana hatlar bir nesnenin sınırlarını temsil ederken, bunların amacı genellikle bir eşik değeri verilen köşeleri azaltarak sürekli çizgileri basitleştirmektir. Yani konturlama, nesne sınırlarındaki karmaşık yapıyı korurken köşe sayısını azaltarak daha düzgün eğriler elde edilmesini amaçlamaktadır.

Bu, piksellerin birleştirilmesiyle oluşturulan bir çizgidir ve bir nesnenin dış sınırını temsil eder. Kontur, şeklin genel yapısını korurken gereksiz ayrıntıları veya köşeleri azaltarak daha basit bir yapı sağlar. Bu, daha az karmaşık ve anlaşılması daha kolay bir gösterim sağladığından, nesnelerin belirlenmesi, izlenmesi, tanımlanması veya diğer nesne tespiti işlemleri için önemli bilgiler sağlamaktadır. Örneğin, nesnelerin alanını, merkezini veya şeklini belirlemek için konturların analizi sıkça kullanılmaktadır.

5.2.7. Tütün yaprakları üzerindeki hastalıklı alanların tespiti

Tütün yaprakları üzerindeki hastalıklı alanların belirlenmesinde ilk olarak görüntü üzerindeki hastalıklı alanlar arka plandan ayrılmıştır. Arka plandan ayrılan alanların piksel sayıları hesaplanmıştır. Piksel değeri 0-10000 arasında olanlar hastalıklı alan olarak kabul edilmiş ve bu alanlar kırmızı/mavi daire içerisinde ve h harfi ile gösterilmiştir. Görüntünün elde edilmesinden, sonuç bölümüne kadar olan işlem basamakları Şekil 5.4'te gösterilmiştir.



Şekil 5. 4. Tütün hastalık tespiti aşamaları

Görüntü işleme sürecindeki adımlar detaylandırıldığında, ilk olarak tütün yaprak resimleri yeniden boyutlandırılmış ve median blur filtresi uygulanarak gürültü azaltılmıştır. Daha sonra, resim LAB renk uzayına dönüştürülmüş ve renk doygunluğu artırılarak daha canlı renkler elde edilmiştir. Yeşil ve kırmızı renk kanalları kullanılarak yeni bir görüntü oluşturulmuş, yüksek yeşil değerlerine sahip pikseller belirli bir eşik değerine göre ayarlanmış ve resim tersine çevrilmiştir. Görüntü binarize edilmiş, morfolojik işlemler (açma ve kapama, erozyon, genişletme) uygulanarak temizlenmiş, hastalıklı alanlar morfolojik olarak daire şekline benzetilmiş ve konturlar belirlenmiştir. Bu konturlar belirli alan aralıklarına göre filtrelenmiş bu kontur değerine sahip alanlar daire olarak kabul edilmiş ve hastalıklı alan olarak işaretlenmiştir. Hastalıklı alanlar belirlendikten sonra yapraklar hastalıklı ve sağlıklı olarak da sınıflandırılmıştır. Son olarak, bu sınıflamalar görsel olarak işaretlenmiş ve analiz sonuçları ekranda sunulmuştur.

Hastalıklı alanın toplam yüzeyde kapladığı alan yazılım tarafından hesaplanmış ve grafik olarak gösterilmiştir. Bu hesaplamalar ışık şiddeti, görüntü boyutu gibi özelliklere göre değişim gösterebilmektedir. Bu işlemler sırasında hastalıklı alanlar morfolojik operatörlerin de yardımıyla daire şekline benzetilerek hesaplama yapılmıştır. Ayrıca çok küçük hastalıklı alanlar yine morfolojik operatörler yardımıyla yok edilerek hesaplamaların dışında

bırakılmıştır (Bükücü ve Gökrem, 2020). Önemsiz sayılabilecek bu hastalıklı alanların yok edilmesi isteğe bağlı olarak değiştirilebilmektedir.

Bu tez çalışmasında Bafra bölgesinde yetiştirilen tütünlerden 1600 hastalıklı yaprak resmi ve 1600 sağlam tütün yaprağı resmi çekilmiş ve çalışmalar bu yapraklar üzerinde gerçekleştirilmiştir.

Hastalıklı alanların tespiti için hesaplamalar yapılmıştır. Bunlar, hastalıklı alan sayısı (THS), hastalıklı alanların toplam yüzeyde kapladığı alan ve görüntüdeki toplam piksel sayılarını ifade etmektedir. Böylece hastalıklı bölgelerin toplam yüzeydeki oranını belirtir. Bu ölçümler, hastalıklı bölgelerin sayısını, toplam hata oranını ve ürünün genel kalitesini değerlendirmek için kullanılmaktadır. Denklem 5.1’de THA hesaplanmıştır.

- THS: toplam hastalık sayısı
- THA: hastalıklı alanların toplam yüzeyde kapladığı alan
- TP: toplam Piksel sayısı

$$THA=THS/TP \quad (5.1)$$

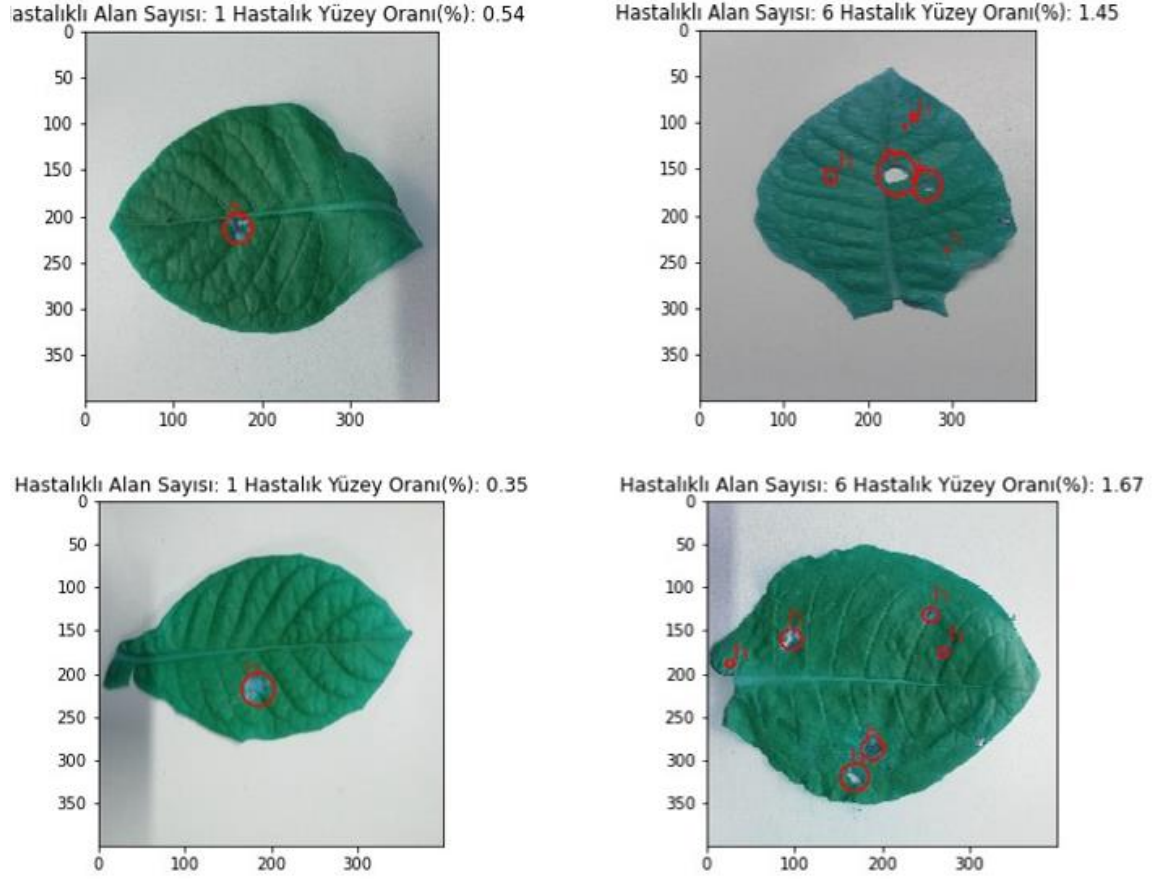
Bilgisayar ortamında tütün yaprakları hastalıklı alanların tespit edilmesi

İlk olarak gerçek zamanlı hastalık tespiti uygulamalarının yapılmasının zor olduğu durumlarda, bilgisayar ortamında hastalık tespiti için bir çalışma yapılmıştır. Bu, gerçek zamanlı hastalık tespiti sistemlerinin imkânsız olduğu ortamlarda kullanılabilecek bir çözüm oluşturmayı hedeflemiştir.

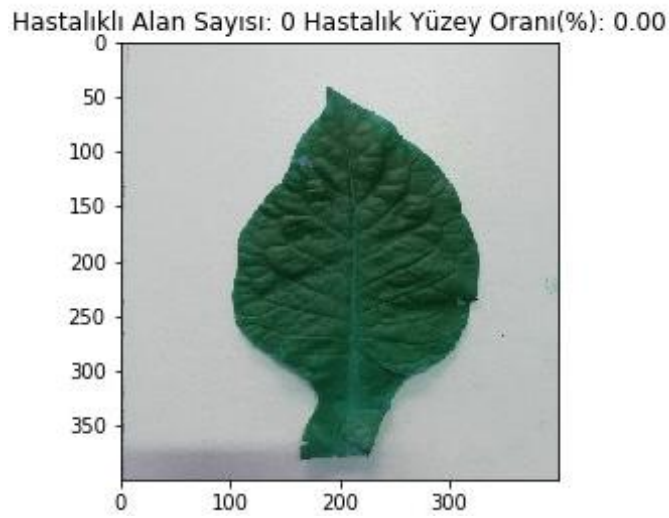
Ayrıca, bilgisayar ortamında gerçekleştirilen bu çalışma ile tütün ekili tarladan elde edilen görüntüler internet yardımıyla merkezdeki bilgisayara aktarılarak hastalık tespiti hızlıca yapılabilir.

Elde edilen tütün yaprak resimleri ilk olarak bilgisayar ortamına aktarılmış ve bu resimler üzerinde hastalıklı alanlar tespit edilmiştir. Şekil 5.5’te hastalıklı alan sayısı ve hastalıklı

alanın tüm alanda kapladığı alan yüzdelik oranları gösterilmiştir. Şekil 5.6'da ise sağlam tütün yaprağı görüntüsü verilmiştir.

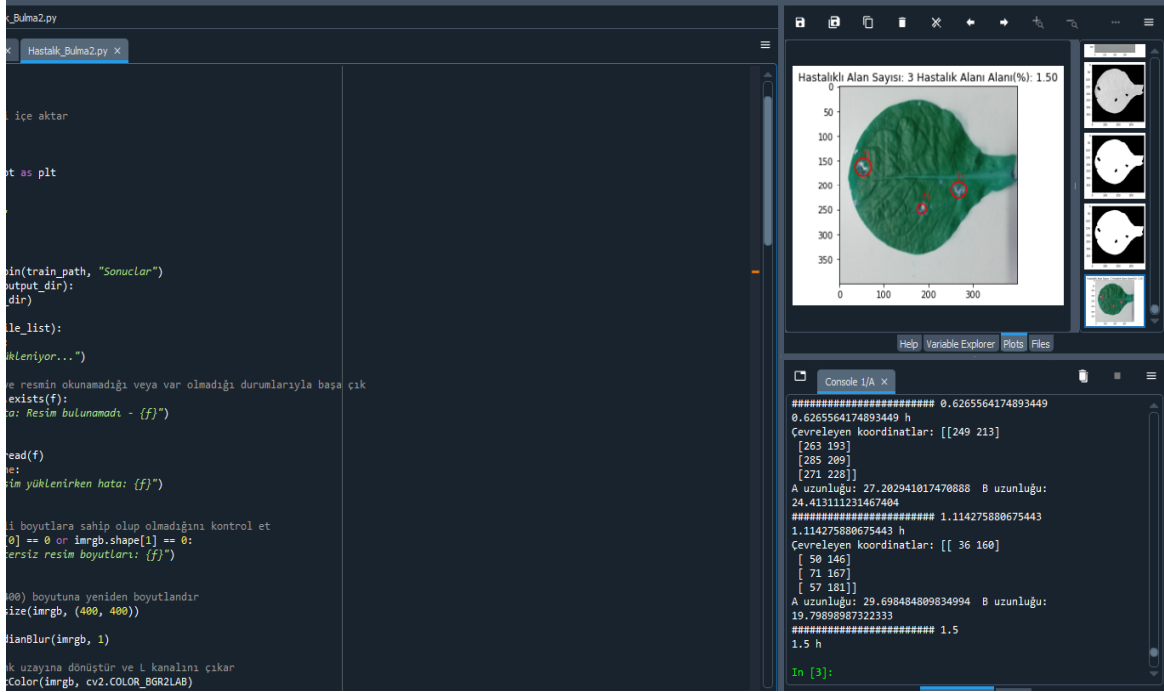


Şekil 5. 5. Hastalıklı tütün yaprakların gösterilmesi



Şekil 5. 6.Sağlam tütün yaprağı

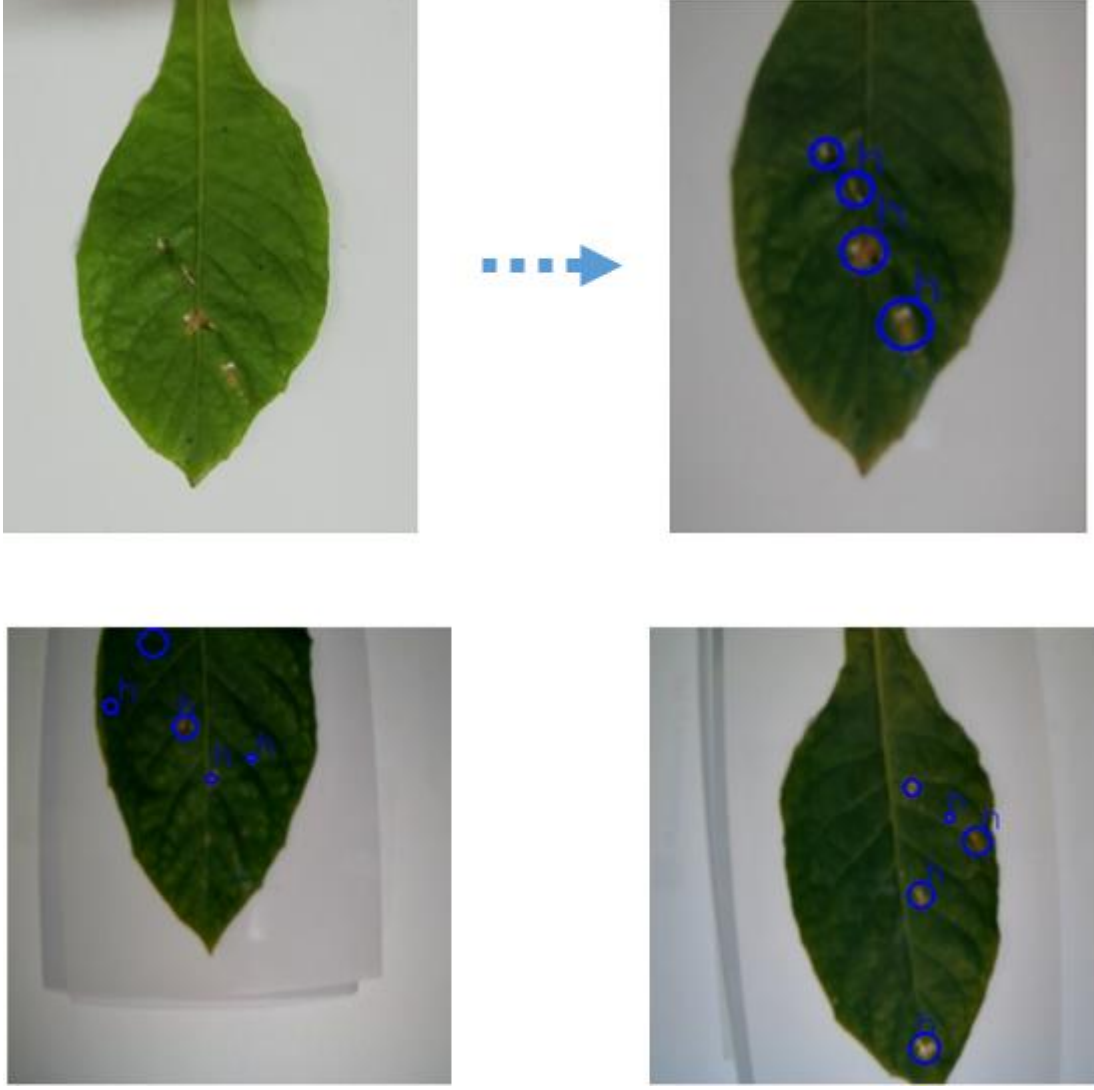
Gerçek zamanlı hastalık tespiti için çalışma imkânı olmayan durumlar da kullanılabilmesi amacıyla bilgisayar ortamında hastalık tespiti gerçekleştirilmiştir. Belirlenen klasördeki tüm yaprak resimleri için hastalık tespiti gerçekleştirilir ve çıktılar sonuçlar klasörüne kayıt edilir. Bilgisayar ortamında hastalık tespiti çalışması Resim 5.4'te gösterilmiştir.



Resim 5. 4. Hastalıklı tütün yapraklarının bilgisayar ortamında tespit edilmesi

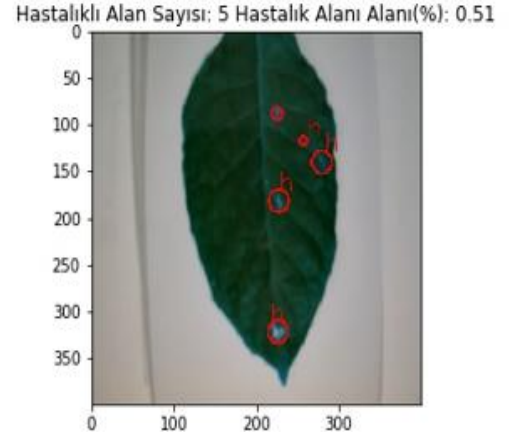
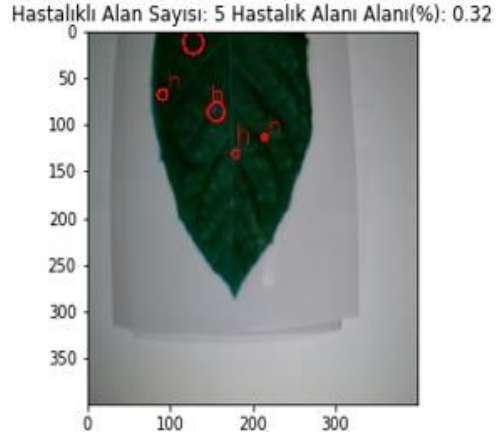
Gerçek zamanlı hastalık tespiti

Tütün yaprakları üzerinde meydana gelen hastalıkların tespiti, gerçek zamanlı tespit yapılan raspberry pi kart ile tasarlanan model üzerinden gerçekleştirilmiştir. Kamradan alınan görüntü boyutları 200x200 piksel boyutlarına düşürülmüştür. Eşik değeri uygulaması yapıldıktan sonra belirlenen kontur değerleri içerisinde kalan hastalıklı alanlar mavi daire içerisinde h harfi ile anlık olarak ekran da gösterilmiştir. Resim 5.5'te tütün yaprakları hastalıklı alanların ekrandaki görüntüsü gösterilmiştir.



Resim 5. 5. Gerçek zamanlı hastalık tespiti ekran çıktısı

Gerçek zamanlı sistem çalıştırıldığında, tütün yapraklarındaki hastalıklı alanlar tespit edilmekte ve hastalıklı yapraklar "hastalik_tespit_sonuc_lari" adlı klasöre kaydedilmektedir. Ayrıca, tütün yaprakları üzerindeki hastalıkların sayısı ve tüm yüzeye oranı bilgileri "hastalik_oran_lari" adlı başka bir klasöre kaydedilmektedir. Bu kayıtlar, ilerleyen dönemlerde hastalıkların detaylı bir şekilde incelenmesi ve analiz edilmesi amacıyla kullanılabilir. Hastalıklı alan sayısı ve oranı, özellikle Şekil 5.7'de gösterilmiştir. Hastalıklı alanlar kırmızı renk daire içinde ve h harfi ile gösterilmiştir. Bu şekil üzerindeki bilgiler, sistemin tütün bitkilerindeki hastalıkları ne kadar başarıyla tespit ettiğini ve hastalıklı bölgelerin genel yüzey içindeki oranını göstermektedir.



Şekil 5. 7. Tütün yaprakları hastalık sayıları ve oranları

6. PERFORMANS ÖLÇÜTLERİ VE ESA MİMARİLERİ İLE SINIFLANDIRILMA İŞLEMİ

Sınıflandırma, veri analizi ve öngörülebilirlik açısından önemli bir role sahiptir. Belirli bir veri kümesindeki örnekleri farklı sınıflara veya kategorilere ayırmak için kullanılır (Alan ve Karabatak, 2020). Bu çalışmada, iki sınıflı sınıflandırma yapılmış ve 3 adet sınıflandırma modeli kullanılmıştır. Modellerin performansı, doğruluk ve uygulama esnekliği gibi kriterler üzerinde detaylı bir değerlendirme yapılmıştır.

6.1. Başarım Ölçütleri

Derin öğrenme performans metrikleri, bir modelin başarımını değerlendirmek amacıyla kullanılır ve modelin ne kadar iyi performans gösterdiğini belirlemek için çeşitli ölçümler sağlar (Ünal, 2015). Gerçekleştirilen çalışmada ESA modelleri arasından VGG16, Resnet50 ve EfficientNetB6 modelleri incelenmiş ve sınıflandırma parametreleri değerlendirilmiştir. Bu parametreler karmaşıklık matrisi(confusion matrix), f1 puan(f1 score), kayıp(los), doğruluk(accuracy) ve ROC eğrisinden oluşmaktadır (Atacak, 2024).

6.1.1. Karmaşıklık matrisi(confusion matrix)

Sınıflandırma modellerinin doğruluk sonuçları genellikle paylaşılsa da, elde edilen bu doğruluk değerlerinin gerçeği tam olarak yansıtmayı yansıtmadığı konusunda belirsizlikler bulunmaktadır. Bu nedenle, model performansını daha detaylı bir şekilde değerlendirmek ve hataları anlamak için çeşitli teknikler geliştirilmiştir. Bu tekniklerden biri de karmaşıklık matrisi oluşturmaktır. Karmaşıklık matrisi, bir makine öğrenimi modelinin performansını özetleyen bir matristir (Aliyev ve Güneş, 2023). Genel olarak sınıflandırma modellerinin performansını değerlendirmek için kullanılmaktadır. Her girdi örneği için modelin tahminlerini gerçek etiketlerle karşılaştırarak oluşturulur. Bu matris, aşağıdaki dört temel ögeyi içermektedir.

- Gerçek Pozitif (TP): Modelin doğru bir şekilde bir pozitif durumu tahmin ettiği durum sayısıdır.

- Gerçek Negatif (TN): Modelin doğru bir şekilde bir negatif durumu tahmin ettiği durum sayısıdır.
- Yanlış Pozitif (FP): Modelin yanlış bir şekilde bir pozitif durumu tahmin ettiği durum sayısıdır.
- Yanlış Negatif (FN): Modelin yanlış bir şekilde bir negatif durumu tahmin ettiği durum sayısıdır.

İkili sınıflandırma için kullanılan matris 2x2'lik bir tabloyu temsil etmektedir. Ancak, çok sınıflı sınıflandırma için kullanılan matrisin şekli, sınıf sayısına eşit olacaktır. Yani, n sınıf için n^n boyutlarında bir matris olacaktır. İkili sınıflandırma için karmaşıklık matrisi Şekil 6.1'de gösterilmiştir.

		TAHMİN	
		Negatif	Pozitif
GERÇEK	Negatif	TP	FP
	Pozitif	FN	TN

Şekil 6. 1. İkili sınıflandırma için karmaşıklık matrisi

6.1.2. Doğruluk (Accuracy)

Doğruluk (Accuracy), bir sınıflandırma modelinin doğru tahminlerinin, toplam tahminlere oranını ifade eden bir performans ölçüsüdür. Doğruluk oranı, 0 ile 1 arasında değer alır ve 0 en düşük performansı, 1 ise en yüksek performansı temsil etmektedir. Doğruluk oranı, denklem 6.1'deki formülle hesaplanır.

$$\text{Accuracy} = (TN + TP)/(TP+FP+FN+TN) \quad (6.1)$$

Bu denklemin sonucu, bir sınıflandırma modelinin genel başarısını göstermektedir. Ancak, dengesiz veri setleri veya özellikle ilgili olmayan sınıfların büyük çoğunluğuna sahip durumlarda, doğruluk oranının tek başına yeterli bir değerlendirme olmayabilir. Bu durumlarda, diğer performans parametreleri de göz önünde bulundurulmalıdır.

6.1.3. Duyarlılık (Recall)

Duyarlılık, pozitif durumların (gerçek pozitifler) ne kadar başarılı bir şekilde tahmin edildiğini gösteren bir performans ölçüsüdür. Bu metrik, özellikle bir sınıflandırma modelinin, örneğin bir hastalığın varlığını doğru bir şekilde tespit edip edemediğini değerlendirmek için kullanılmaktadır. Formülü denklem 6.2' de gösterilmiştir.

$$\text{Duyarlılık} = TP / (TP + FN) \quad (6.2)$$

6.1.4. Precision(Hassasiyet)

Precision (Hassasiyet), bir sınıflandırma modelinin pozitif olarak tahmin ettiği durumların, gerçekte pozitif olan durumların ne kadarına denk geldiğini gösteren bir performans ölçüsünü ifade etmektedir. Formülü denklem 6.3' de gösterilmiştir.

$$\text{Hassasiyet} = TP / (TP + FP) \quad (6.3)$$

6.1.5. F1 skor(F-score)

F1 skor, bir sınıflandırma modelinin başarısını değerlendirmek için hassasiyet (precision) ve duyarlılık (recall) değerlerinin harmonik ortalamasını kullanmaktadır. Bu metrik, modelin pozitif durumları doğru ve kapsamlı bir şekilde tahmin etme yeteneğini ölçmektedir. F skor özellikle dengesiz sınıflı veri setlerinde, yani sınıflar arasında örnek sayısındaki büyük farklar olduğu durumlarda kullanışlıdır. F-score, denklem 6.4'deki formülle hesaplanır.

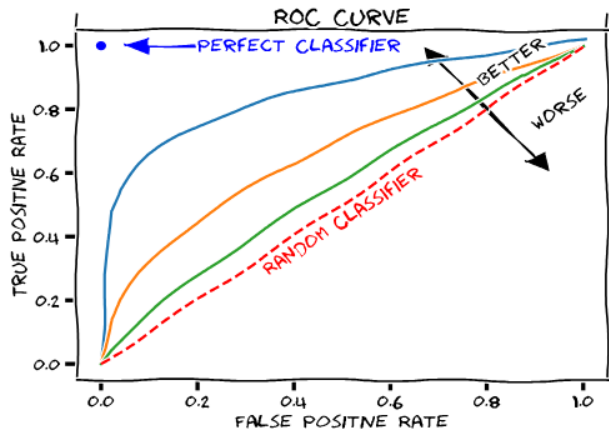
$$\text{F1 Skor} = 2 * (\text{Duyarlılık} * \text{Hassasiyet}) / (\text{Duyarlılık} + \text{Hassasiyet}) \quad (6.4)$$

6.1.6. Logaritmik kayıp (Loss)

Logaritmik kayıp, gerçek sınıf etiketi ile modelin tahmini arasındaki farkı ölçmektedir. Bir örneğin belirli bir sınıfa ait olma olasılığını ölçer ve bu olasılıkları değerlendirir. Logaritmik kaybın değeri ne kadar küçükse modelin performansı o derece iyidir. Bu ölçüm, özellikle çok sınıflı sınıflandırma problemlerinde ve olasılık tahminlerinde model performansını değerlendirmek için sıkça kullanılır.

6.1.7. Roc eğrisi(roc curve)

AUC - ROC eğrisi, sınıflandırma modellerinin performansını değerlendirmek için kullanılan bir ölçüdür. ROC (Receiver Operating Characteristic) eğrisi, çeşitli eşik değerlerinde duyarlılık ve özgüllük arasındaki ilişkiyi gösteren grafiği ifade eder. AUC (Area Under the Curve), bu eğri altındaki alanı göstermektedir. AUC, modelin sınıflar arasında ne kadar iyi ayırım yaptığını ölçer. Yüksek bir AUC değeri, modelin 0 sınıfını 0 olarak ve 1 sınıfını 1 olarak tahmin etmede, ya da diğer bir deyişle, pozitif ve negatif örnekleri birbirinden ayırmada ne kadar başarılı olduğunu göstermektedir. Özellikle, AUC'nin yüksek olması, modelin hastalığı olan ve olmayan durumları doğru bir şekilde ayırt etme yeteneğini ifade etmektedir. Şekil 6.2'de roc eğrisi tanımlanması verilmiştir.



Şekil 6. 2. Roc eğrisi tanımlanması (Tao, 2021)

Roc eğrisi hesaplama formülü denklem 6.5 ve denklem 6.6' da verilmiştir.

$$TPR = TP / (TP + FN) \quad (6.5)$$

$$FPR = FP / (FP + TN) \quad (6.6)$$

6.1.8. Intersection over union (IoU)

İki kümenin örtüşme miktarını ölçen bir metriktir ve özellikle bilgisayarla görme ve görüntü işleme alanlarında yaygın olarak kullanılmaktadır. Bu metrik, nesne tespiti (object detection) ve segmentasyon (segmentation) algoritmalarının performansını değerlendirmede önemli bir araçtır (Aydın vd., 2023).

$$IoU = \text{iki kümenin kesişim alanı} / \text{iki kümenin birleşim alanı} \quad (6.7)$$

Intersection over Union (IoU) değerinin 1'e yakın olması, tespit edilen ve gerçek alanlar arasında yüksek bir örtüşme sağlandığını gösterir.

6.1.9. Transfer öğrenme eğitim verileri

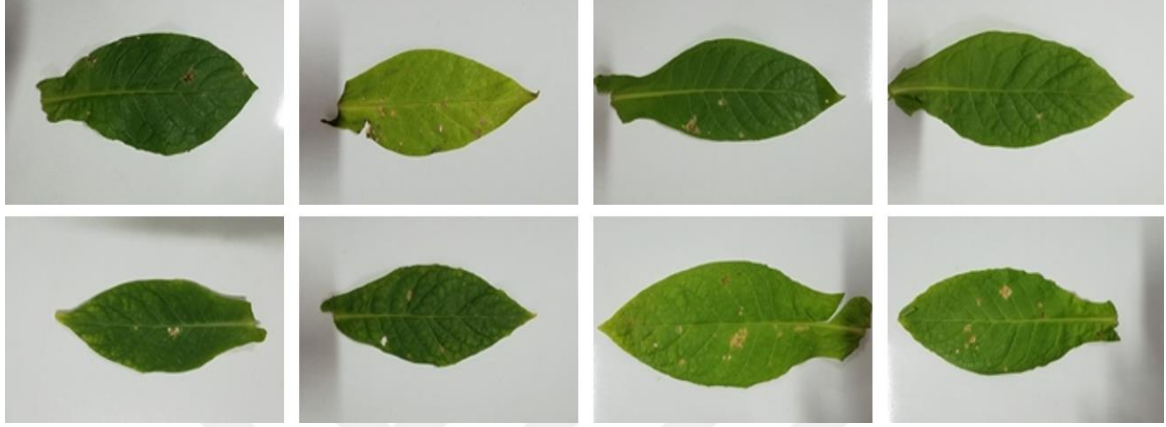
Tütün yaprak hastalıklarının “sağlam” ve “hastalıklı” olarak sınıflandırmak için evrimsel sinir ağlarına (ESA) dayalı VGG16, Resnet50 ve EfficientNetB6 mimarileri kullanılmış ve performansları ölçülmüştür. Veri seti olarak Bafra ilçesinde yetişen tütün bitkisi yapraklarının resimleri çekilerek veri seti oluşturulmuştur. 1600 hastalıklı ve 1600 sağlıklı tütün yaprağı fotoğrafı çekilmiştir. Bu resimlerin 2560'ı eğitim, 640'si doğrulama için ayrılmıştır. Eğitim için ayrılan resimler, veri artırma yöntemleri kullanılarak artırılmış ve böylece toplamda 5120 sağlıklı ve 5120 hastalıklı resim elde edilmiştir. Bu genişletilmiş veri seti ile çalışmalar gerçekleştirilmiştir. Çizelge 6.1'de eğitim verileri ve Resim 6.1 ve Resim 6.2'de veri seti örneği görülmektedir.

Çizelge 6. 1. Eğitim verileri

	Toplam Resim Sayısı 3200 Adet		
	Eğitim için ayrılan resim %80	Doğrulama için ayrılan resim %20	Artırılmış eğitim verisi %80
Adet	2560	640	10240

ESA mimarileri VGG16, Resnet50 ve EfficientNetB6 mimarileri kullanılırken keras ve tensorflow kütüphaneleri kullanılmıştır. Çalışma ortamı olarak Jupyter ide kullanılmıştır.

Ayrıca tüm bu modellerin performanslarını eşit bir şekilde değerlendirmek amacıyla eğitim aşaması intel(R) 3.nesil core(TM) i7-3630qmcpu işlemcisi ve nvidia geforce gt 650m ekran kartına sahip bilgisayar kullanılarak gerçekleştirilmiş ve karmaşıklık matrisleri hesaplanmıştır.



Resim 6. 1. Hastalıklı tütün yaprak veri seti



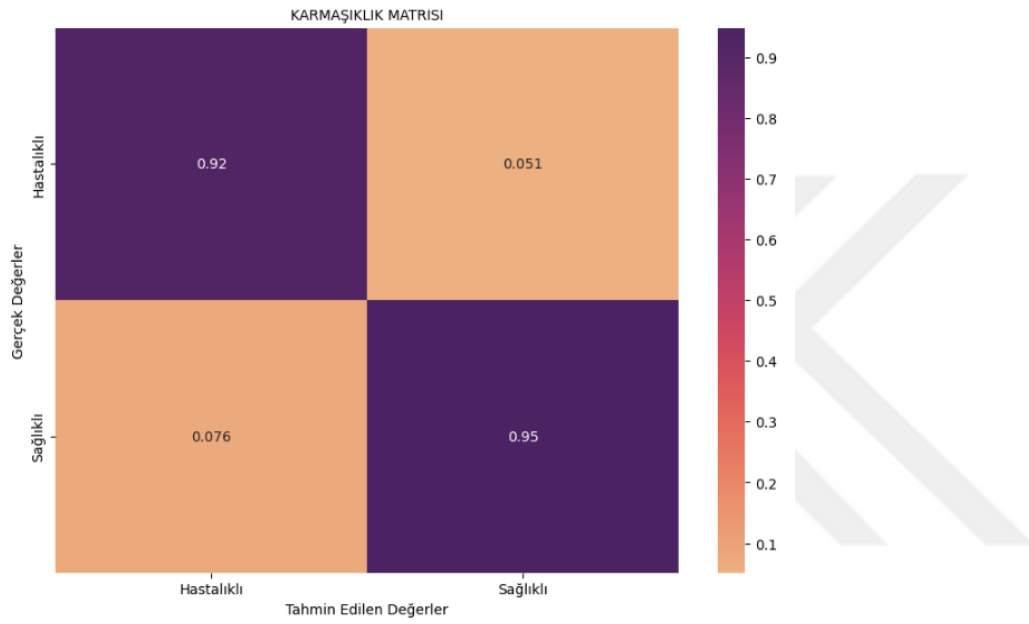
Resim 6. 2. Sağlıklı tütün yaprak veri seti

6.2. Transfer Öğrenme Modelleri

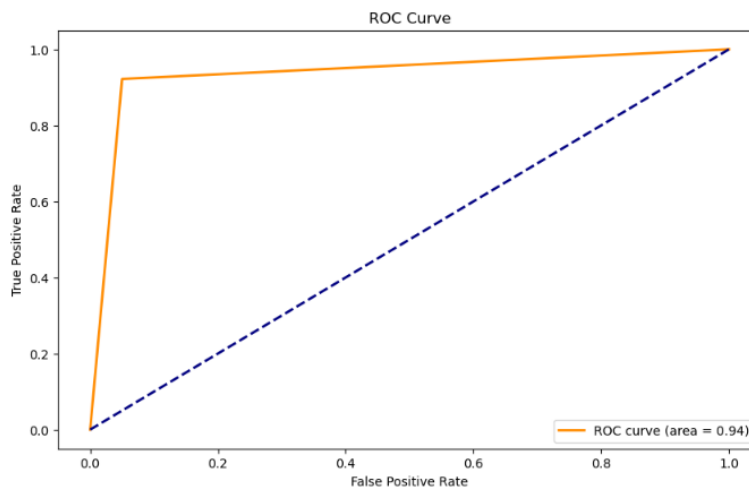
Transfer öğrenme, bir modelin bir görevde öğrendiği bilgiyi başka bir görevde yeniden kullanma tekniğidir (Çiftçi ve Tekin, 2023). Bu çalışmada kullanılan transfer öğrenme modellerinin performansları bu kısımda incelenmiştir.

6.2.1. VGG16 modeli

Veri seti %80 eğitim ve %20 test için ayrılmıştır. Resim boyutları 224 x 224 piksel olarak ayarlanmıştır. Model eğitilirken “Sağlam” ve “Hastalıklı” olarak etiketlenmiştir. VGG16 modelinin Karmaşıklık matrisi Şekil 6.3’te roc eğrisi Şekil 6.4’te ve eğitime ait değerlendirme verileri Çizelge 6.2’de gösterilmiştir.



Şekil 6. 3. Karmaşıklık matrisi



Şekil 6. 4. ROC eğrisi

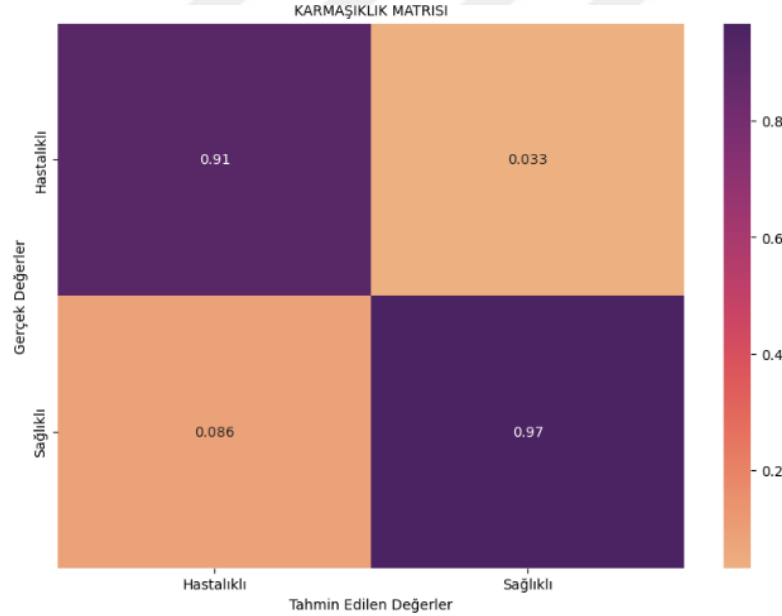
Çizelge 6. 2. Vgg16 modeli değerlendirme verileri

Doğruluk (Accuracy)	Precision (Hassasiyet)	Kayıp(Loss)	Duyarlılık (Recall)	F1-Puanı (F1-score)
0.93	0,93	0.43	0.93	0.93

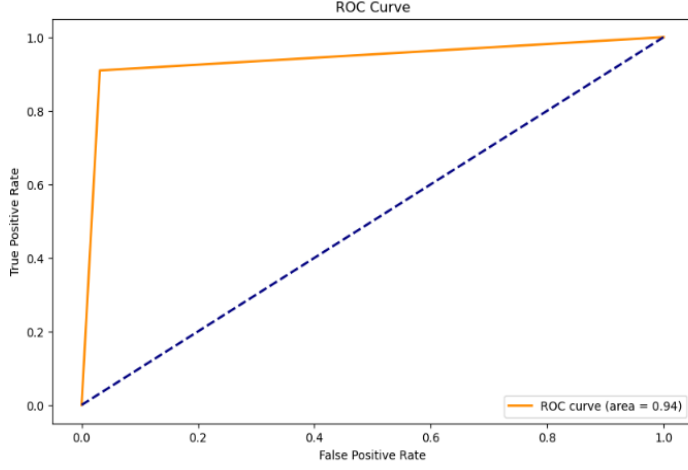
Vgg16 modelinin doğruluk oranı %93 olarak hesaplanmış kayıp ise 0.43 olarak hesaplanmıştır. Modelinizin genel performansının iyi olabilmesi için kayıp oranının düşürülmesi gerekmektedir.

6.2.2. Resnet50 modeli

Veri seti %80 eğitim ve %20 test için ayrılmıştır. Resim boyutları 224 x 224 piksel olarak ayarlanmıştır. Model eğitilirken “Sağlam” ve “Hastalıklı” olarak etiketlenmiştir. Resnet50 modelinin karmaşıklık matrisi Şekil 6.5’te roc eğrisi Şekil 6.6’de ve eğitime ait değerlendirme verileri Çizelge 6.3’te gösterilmiştir.



Şekil 6. 5. Karmaşıklık matrisi



Şekil 6. 6. ROC eğrisi

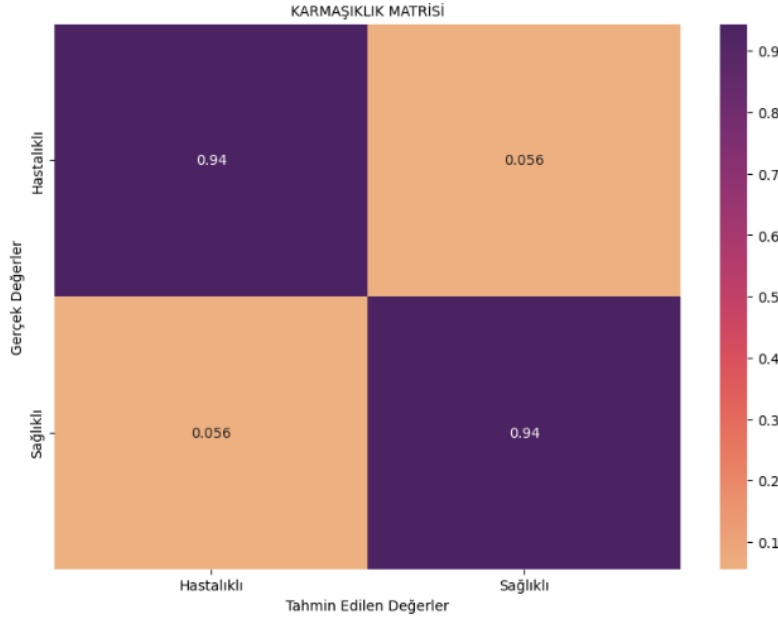
Çizelge 6. 3. Resnet50 modeli değerlendirme verileri

Doğruluk (Accuracy)	Precision (Hassasiyet)	Kayıp(Loss)	Duyarlılık (Recall)	F1-Puanı (F1-score)
0.93	0,94	0.46	0.93	0.93

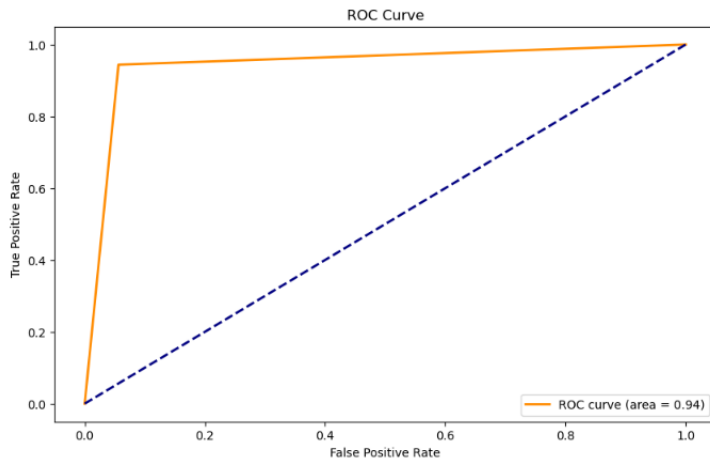
Resnet50 modelinin doğruluk oranı %93 olarak hesaplanmış kayıp ise 0.46 olarak hesaplanmıştır. Doğruluk değer başarılı olmasına rağmen kayıp değeri fazladır.

6.2.3. EfficientNetB6 modeli

Veri seti %80 eğitim ve %20 test için ayrılmıştır. Resim boyutları 512x 512 piksel olarak ayarlanmıştır. Model eğitilirken “Sağlam” ve “Hastalıklı” olarak etiketlenmiştir. EfficientNetB6 modelinin karmaşıklık matrisi Şekil 6.7’de roc eğrisi Şekil 6.8’de ve eğitime ait değerlendirme verileri Çizelge 6.4’te gösterilmiştir.



Şekil 6. 7. Karmaşıklık matrisi



Şekil 6. 8. ROC eğrisi

Çizelge 6. 4. Resnet50 modeli değerlendirme verileri

Doğruluk (Accuracy)	Precision (Hassasiyet)	Kayıp(Loss)	Duyarlılık (Recall)	F1-Puanı (F1-score)
0.94	0,94	0.16	0.94	0.94

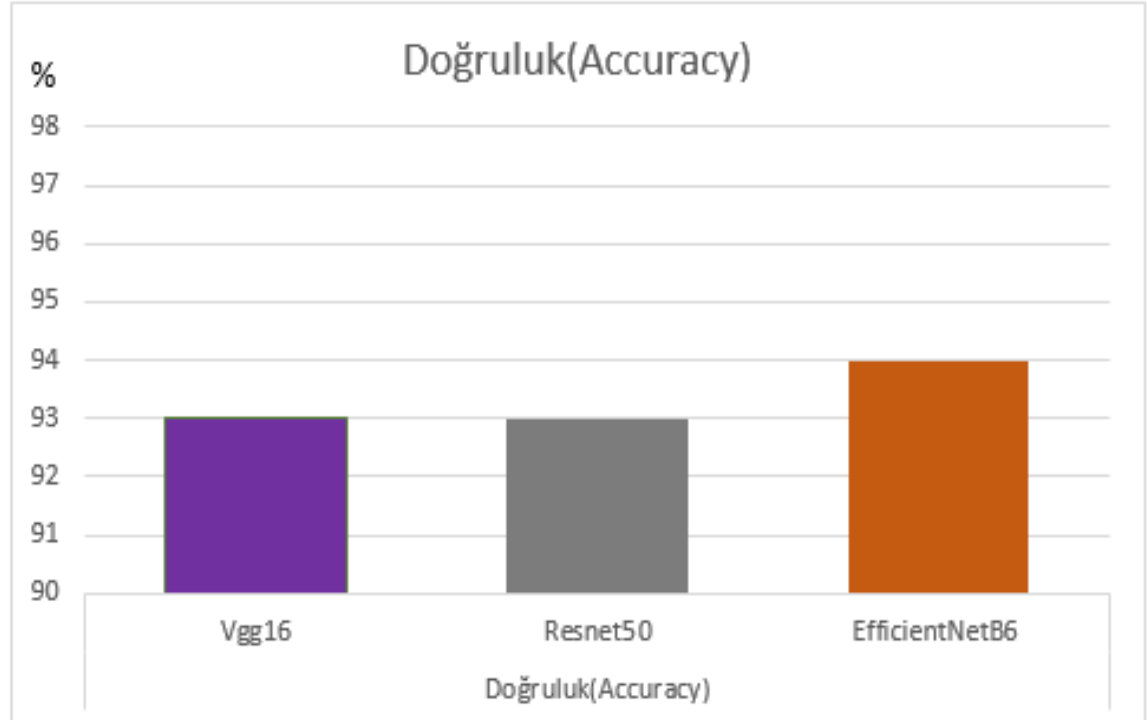
EfficientNetB6 modelinin doğruluk oranı %94 olarak hesaplanmış kayıp ise 0.16 olarak hesaplanmıştır. Doğruluk ve kayıp oranlarına bakıldığında başarılı model olduğu görülmektedir.

6.3. Transfer Öğrenme Modellerinin Karşılaştırılması

Çalışmada kullanılan modellerin eğitim verileri sonuçları Çizelge 6.5'te, doğruluk oranları Şekil 6.9'da gösterilmiştir.

Çizelge 6. 5. ESA modelleri değerlendirme verileri

	Doğruluk (Accuracy)	Precision (Hassasiyet)	Kayıp(Loss)	Duyarlılık (Recall)	F1-Puanı (F1-score)
VGG16	0.93	0,93	0.43	0.93	0.93
Resnet50	0.93	0,94	0.46	0.93	0.93
EfficientNetB6	0.94	0,94	0.16	0.94	0.94



Şekil 6. 9. ESA modellerinin doğruluk oranları

Tütün yaprak hastalıklarını sınıflandırmak için kullanılan modeller arasında, "sağlam" ve "hastalıklı" sınıfları ayırt etme yeteneği en yüksek doğruluk oranına sahip EfficientNetB6 modelidir. Bu, modelin genel olarak doğru tahminlerde bulunduğunu ve sağlam ile hastalıklı

yaprakları başarılı bir şekilde ayırt ettiğini göstermektedir. Ancak, diğer modellerde yüksek bir başarı oranına rağmen, bu modellerde kayıp oranının EfficientNetB6 modelinden daha yüksek olması performansını olumsuz etkilemektedir. Kayıp oranının yüksek olması, modelin gerçek değerlere daha az benzer tahminlerde bulunduğunu göstermektedir.

6.4. Gerçek Zamanlı Çalışan Sistem Performans Değerleri

Gerçek zamanlı olarak çalışan sistem tasarımında 30 adet sağlıklı ve 30 adet hastalıklı yaprak veri seti kullanılmıştır.

6.4.1. Gömülü sistem tasarımı performans değerleri

Gerçek zamanlı hastalıklı yaprakları tespit eden gömülü sistem tasarımının karmaşıklık matrisi Çizelge 6.6’da, performans metrikleri Çizelge 6.7 de verilmiştir.

Çizelge 6. 6. Gömülü sistem tasarımı karmaşıklık matrisi

		Tahmin	
		Sağlam	Hastalıklı
Gerçek	Sağlam	29	1
	Hastalıklı	3	27

Çizelge 6. 7. Gömülü sistem değerlendirme verileri

Doğruluk (Accuracy)	Precision (Hassasiyet)	Duyarlılık (Recall)	F1-Puanı (F1-score)
0.93	0,90	0.96	0.92

Karmaşıklık matrisi hesaplamasının sonucunda hastalıklı ve sağlıklı bütün yapraklarını ayırt eden gömülü sistem tasarımı %93.33 oranında bir başarı elde etmiştir.

6.4.2. Bilgisayar ortamında çalışan sistemin performans değerleri

Bilgisayar ortamında gerçekleştirilen çalışmada sistemin karmaşıklık matrisi Çizelge 6.8’de, performans metrikleri Çizelge 6.9’de verilmiştir.

Çizelge 6. 8. Bilgisayar görüntü işleme karmaşıklık matrisi

		Tahmin	
		Sağlam	Hastalıklı
Gerçek	Sağlam	29	1
	Hastalıklı	2	28

Çizelge 6. 9. Bilgisayar görüntü işleme değerlendirme verileri

Doğruluk (Accuracy)	Precision (Hassasiyet)	Duyarlılık (Recall)	F1-Puanı (F1-score)
0.95	0,96	0.93	0.94

Karmaşıklık matrisi hesaplamasının sonucunda hastalıklı ve sağlıklı tütün yapraklarını ayırt eden sistem tasarımı %95 oranında bir başarı elde etmiştir.

6.4.3. Gözlemsel inceleme ve görüntü işleme sonuçlarının karşılaştırılması

30 adet hastalıklı tütün yaprağından elde edilen görüntüler üzerinde, yapraklarda bulunan hastalıklı bölgeler öncelikle gözlemsel yöntemle sayılmıştır. Daha sonra, gömülü sistem ve bilgisayar ortamında uygulanan görüntü işleme teknikleri kullanılarak hastalıklı alanlar hesaplanmış ve sonuçlar Çizelge 6.10'da sunulmuştur.

Çizelge 6. 10. Hastalıklı alanların gözlemsel ve bilgisayarlı görüntüleme yöntemleri ile sayımı ve karşılaştırılması

Tütün Yaprakları Görüntüleri		Gözlem (hastalıklı alan sayısı)	Gömülü Sistem Görüntü İşleme	Fark	İou	Bilgisayar. Görüntü İşleme	Fark	İou
	1	1	1	0	1	1	0	1
	2	1	1	0	1	1	0	1
	3	1	0	1	0	0	1	0
	4	1	1	0	1	1	0	1
	5	2	1	1	0,5	2	0	1
	6	4	3	1	0,75	3	1	0,75
	7	3	3	0	1	3	0	1
	8	2	2	0	1	2	0	1
	9	3	3	0	1	3	0	1
	10	3	3	0	1	3	0	1
	11	6	6	0	1	6	0	1
	12	1	1	0	1	1	0	1
	13	1	0	1	0	0	1	0
	14	2	2	0	1	2	0	1
	15	4	3	1	0,75	4	0	1
	16	8	6	2	0,75	7	1	0,87
	17	3	2	1	0,66	2	1	0,66
	18	6	5	1	0,83	5	1	0,83
	19	6	6	0	1	6	0	1
	20	4	4	0	1	4	0	1
	21	3	3	0	1	3	0	1
	22	1	1	0	1	1	0	1
	23	2	2	0	1	2	0	1
	24	5	4	1	0,8	4	1	0,8
	25	6	6	0	1	6	0	1
	26	1	0	1	0	1	0	1
	27	3	3	0	1	3	0	1
	28	4	2	2	0,5	2	2	0,5
	29	2	2	0	1	2	0	1
	30	3	3	0	1	3	0	1

Çizelge 6.10 incelendiğinde, 1 numaralı tütün yaprağında 1 adet hastalıklı alan tespit edilmiştir. Bu hastalıklı alan, hem gerçek zamanlı çalışan gömülü sistem hem de bilgisayar ortamında yapılan görüntü işleme sonucunda doğru bir şekilde tanımlanmıştır. Diğer tütün resimleri de aynı yöntemle incelenmiş ve benzer doğrulukta sonuçlar elde edilmiştir. Çalışma sonuçları ile gözlemsel değerlendirme sonuçlarının birbirine çok yakın olması, çalışmanın başarılı bir şekilde gerçekleştirildiğini göstermektedir.

7. SONUÇLAR VE ÖNERİLER

Bu çalışma, tütün yapraklarında görülen hastalıkları tespit etmek amacıyla görüntü işleme teknikleri ve derin öğrenme algoritmaları kullanarak bir model geliştirilmiş ve değerlendirilmiştir. Çalışmanın başlıca hedefleri arasında, tütün yapraklarının sağlığını değerlendirmek için etkili bir görüntü işleme tabanlı yöntem geliştirmek, hastalıkları erken teşhis etmek ve tarım verimliliğini artırmak yer almaktadır. Çalışmada elde edilen veri seti Samsun Bafra ilçesine ait tütün yaprak resimleridir. Yapraklar 1600 sağlam ve 1600 hastalıklı olmak üzere 2 sınıftan oluşmaktadır.

Gerçek zamanlı olarak tütün yapraklarındaki hastalıklı alanları tespit etmek amacıyla geliştirilen gömülü sistem tabanlı modelde, bilgisayar tabanlı görüntü işleme teknikleri kullanılmıştır. Bu model, tütün yapraklarındaki hastalıkları belirlemek için renk uzayı dönüşümü ve morfolojik görüntü işleme yöntemlerini kullanmaktadır. Tespit edilen hastalıklı alanlar morfolojik görüntü işleme teknikleri uygulanarak daire şekline benzetilmiş ve belirlenen bu alan daire içerisine alınarak "h" harfi ile etiketlenmiştir, bu sayede görsel olarak kolaylıkla ayırt edilebilmektedir. Ayrıca, hastalıklı bölgelerin tüm alanındaki yüzdesi hesaplanmış ve bu bilgiler kaydedilmiştir. Tütün uzmanlarının incelemesi için bu bilgiler kayıt altına alınmıştır. Sonuç olarak, gömülü sistem tütün yapraklarındaki hastalıklı alanları başarıyla tespit etmiş ve bu yaprakları hastalıklı veya sağlıklı olarak sınıflandırmıştır. Bu sınıflandırma sonucunda, sistem %93.33 oranında bir başarı sağlamıştır.

Kayıt altına alınan bu veriler, hastalığın yayılma derecesi ve etkisi hakkında önemli bilgiler sağlamaktadır. Bu bilgiler tütün eksperlerine ve çiftçilere, hastalıklı alanların genişliğini değerlendirme ve tedavi süreçlerinde yardımcı olacaktır.

Bu çalışma, bilgisayar ortamında yapılan bir offline analizi de içermektedir. Tütün yapraklarından elde edilen görüntüler, bilgisayar ortamına aktarılarak detaylı bir görüntü işleme sürecinden geçirilmiştir. Offline analizde, gerçek zamanlı çalışan modele benzer bir yaklaşım kullanılmış ve hastalıklı alanlar daire şeklinde işaretlenmiş ve üzerlerine "h" harfi eklenerek görsel olarak vurgulanmıştır. Bu işaretlemeler, hastalıklı bölgelerin daha rahat ve hızlı bir şekilde tanımlanabilmesini sağlamaktadır. Bu işlem sonucunda, hastalıklı ve sağlıklı yapraklar tespit edilmiş ve sınıflandırılmıştır. Offline analiz gerçek zamanlı çalışma dışında da hastalıkların tespit edilmesine olanak tanımaktadır. Elde edilen görüntülerin daha önce

kaydedilmiş olması veya gerçek zamanlı bir sistemin mümkün olmadığı durumlar için bir çözüm sunmaktadır. Bilgisayar ortamı üzerinde gerçekleştirilen bu çalışmanın sonucunda %95 oranında başarı elde edilmiştir.

Yapılan çalışma, hem gerçek zamanlı hem de offline koşullarda etkili bir hastalık tespit yöntemi sunmaktadır. Gerçekleştirilen çalışma ile tütün tarımında hastalıkların belirlenmesi için geniş bir uygulama alanı bulabilir ve tarım uzmanlarına, hastalıkla mücadele stratejilerini geliştirmeleri konusunda önemli bilgiler sağlayabilir.

Tez çalışmasında, tütün yapraklarının "hastalıklı" ve "sağlıklı" olarak iki sınıfa ayrılması amacıyla VGG16, ResNet50 ve EfficientNetB6 evrişimli sinir ağı modelleri detaylı bir şekilde değerlendirilmiştir. Tütün yapraklarından elde edilen veri seti, görüntü artırma teknikleri kullanılarak artırılmıştır ve bu veri seti üzerinde üç farklı model eğitilmiştir. Elde edilen sonuçlar incelendiğinde, EfficientNet modelinin diğer modellere göre %94'lük doğruluk değeriyle en yüksek başarı oranına sahip olduğu belirlenmiştir. Bu durum, EfficientNetB6 modelinin tütün yapraklarındaki özellikleri daha etkili bir şekilde öğrenerek, daha hassas ve doğru tahminler yapabildiğini göstermektedir. Bu bulgular, tarım sektöründe hastalıkların etkili bir şekilde belirlenmesi için bu derin öğrenme modellerinin kullanılabilirliğini desteklemektedir. Yapılacak yeni çalışmalara önderlik edeceği görülmektedir.

Yapılacak ileriki çalışmalarda, tütün yapraklarının "hastalıklı" ve "sağlıklı" olarak 2'li sınıflandırılması yöntemi geliştirilerek sınıflandırma performansı artırılabilir. Bu geliştirme, hastalıkların spesifik adlarının belirlenmesi için kullanılabilir. Ayrıca, farklı evrişimli sinir ağı modelleri üzerinde sınıflandırma yönteminin performansı karşılaştırmalı olarak değerlendirilebilir, böylece en etkili modelin belirlenmesi sağlanabilir.

KAYNAKLAR

- Acar, U. ve Bayram, B. (2009, 11-15 Mayıs). *Morfolojik Görüntü Filtreleri ile İkonos Görüntülerinden Otomatik Bina Çıkarımı*. TMMOB Harita ve Kadastro Mühendisleri Odası 12. Türkiye Harita Bilimsel ve Teknik Kurultayı, Ankara.
- Akkoca, E. ve Buluş, E. (2021). Telea ve Naiver Stokes Algoritmaları Kullanılarak Görüntülerdeki Bozulmaları Düzeltme. *Türkiye Bilişim Vakfı Bilgisayar Bilimleri ve Mühendisliği Dergisi*, 14(2), 77-85.
- Alan, A. ve Karabatak, M. (2020). Veri Seti - Sınıflandırma İlişkisinde Performansa Etki Eden Faktörlerin Değerlendirilmesi. *Fırat Üniversitesi Mühendislik Bilimleri Dergisi*, 32(2), 531-540.
- Aliyev, H. ve Güneş, P. (2023). Makine Öğrenme Algoritmalarını Kullanan Duygu Tahminine Dayalı Müzik Öneri Sistemi. *Anadolu Bil Meslek Yüksekokulu Dergisi*, 18(68), 189-234.
- Altunay, A. (2012). Geleneksel Medyadan Yeni Medyaya: Görüntü Yüzeyi. *Selçuk Üniversitesi Sosyal Bilimler Enstitüsü Dergisi*, 27, 33-44.
- Altunkurt, Ö. ve Kahrıman, M. (2011). Gerçek Zamanlı Olarak, Anfis İle Renk Tabanlı Nesne Tespit Ve Motorlu Sistem İle Takip Edilmesi. *SDU Teknik Bilimler Dergisi*, 1(1), 1-5.
- Aptoula, E. and Yanikoglu, B. (2013, September). *Morphological Features for Leaf Based Plant Recognition*. 2013 20th IEEE International Conference on Image Processing (ICIP), Melbourne.
- Atacak, İ. (2024). Kalp Yetmezliği Tahmininin Kategorik Olarak Farklı Tip Makine Öğrenmesi Yöntemleri ile Uygulanmasına Yönelik Bir Değerlendirme Çalışması. *EMO Bilimsel Dergi*, 14(1), 73-85.
- Atalı, G., Özkan, S. S. ve Karayel, D. (2016). Morfolojik Görüntü İşleme Tekniği ile Yapay Sinir Ağlarında Görüntü Tahribat Analizi. *Akademik Platform Mühendislik ve Bilim Dergisi*, 4(1), 1-7.
- Avila-George, H., Valdez-Morones, T., Pérez-Espinosa, H., Acevedo-Juárez, B. and Castro, W. (2018). Using Artificial Neural Networks for Detecting Damage on Tobacco Leaves Caused by Blue Mold. *International Journal of Advanced Computer Science and Applications (IJACSA)*, 9(8), 579-583.
- Aydın, A. (2021). Bilgisayar Bilimlerinde Teorik ve Uygulamalı Araştırmalar. T. Talan ve C. Aktürk (Ed.), *Derin Öğrenme*, (s. 105-129). İstanbul: Efe Akademi Yayınları.
- Aydın, S., Taşyürek, M. ve Öztürk, C. (2023). MR Görüntülerinde Evrişimli Sinir Ağlar Kullanılarak Alzheimer Hastalık Tespiti. *Erciyes Üniversitesi Fen Bilimleri Enstitüsü Fen Bilimleri Dergisi*, 39(3), 357-368.

- Başaran, Z. ve Elbasan, B. (2023). Fizyoterapi ve Rehabilitasyonda Güncel Yazılım Teknolojisi: Görüntü İşleme Tekniği. *Gazi Sağlık Bilimleri Dergisi*, 8(2), 79-88.
- Benli, S. N. (2021). *Derin Evrişimli Yapay Sinir Ağı Kullanarak Meyve Yaprığı Hastalık Tespiti*. Yüksek Lisans Tezi, Bilecik Şeyh Edebali Üniversitesi Fen Bilimleri Enstitüsü, Bilecik.
- Bolouri, H. (1995). Book Review: Fundamentals of Neural Networks – Architectures, Algorithms, and Applications: L. FAUSETT. *International Journal of Electrical Engineering & Education*, 32(3), 284–285. doi: 10.1177/002072099503200320.
- Boyacıgil, M. F. (2022). *Görüntü İşleme Teknikleriyle Bir Endüstriyel Kalite Kontrol Uygulaması*. Yüksek Lisans Tezi, Necmettin Erbakan Üniversitesi Fen Bilimleri Enstitüsü, Konya.
- Boyacıgil, M. F. ve Uzun, Y. (2023). Raspberry PI Tabanlı Görüntü İşleme Uygulaması Geliştirilmesi. *Osmaniye Korkut Ata Üniversitesi Fen Bilimleri Enstitüsü Dergisi*, 6(2), 1067-1078.
- Boztoprak, H., Çağlar, M. F. ve Merdan, M. (2007, 14-18 Kasım). *Alternatif Morfolojik Bir Yöntemle Plaka Yerini Saptama*. XII. Elektrik, Elektronik, Bilgisayar, Biyomedikal Mühendisliği Ulusal Kongresi, Eskişehir.
- Bükücü, Ç. C. ve Gökrem, L. (2020). A New Prototype That Performs Real-Time Error Detection in Glass Products. *Uluslararası Mühendislik Araştırma ve Geliştirme Dergisi (UMAGD) / International Journal of Engineering Research and Development (IJERAD)*, 12(2), 510-519.
- Bükücü, Ç. C. (2021). *Görüntü İşleme Teknikleri ile Cam Ürünlerde Hata Tespiti*, Doktora Tezi, Gaziosmanpaşa Üniversitesi Lisansüstü Eğitim Enstitüsü, Tokat.
- Chowdhury, M. E., Rahman, T., Khandakar, A., Ayari, M. A., Khan, A. U., Khan, M. S., Emadi, N. A., Raez, M. B. I., Islam, M. T. and Ali, S. H. M. (2021). Automatic and Reliable Leaf Disease Detection Using Deep Learning Techniques. *AgriEngineering*, 3(2), 294-312.
- Cireşan, D. C., Meier, U., Masci, J., Gambardella L. M. And Schmidhuber, J. (2011, July). *Flexible, High Performance Convolutional Neural Networks for Image Classification*. IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence, Barcelona.
- Çakmakçı, M. F. ve Çakmakçı, R. (2023). Uzaktan Algılama, Yapay Zekâ ve Geleceğin Akıllı Tarım Teknolojisi Trendleri. *Avrupa Bilim ve Teknoloji Dergisi*, (52), 234-246.
- Çetiner, H. (2021). Yaprak Hastalıklarının Sınıflandırılabilmesi İçin Önceden Eğitilmiş Ağ Tabanlı Derin Ağ Modeli. *Adıyaman Üniversitesi Mühendislik Bilimleri Dergisi*, 8(15), 442-456. DOI: 10.54365/adyumbd.988049.
- Çiftçi, M. E. ve Tekin, R. (Mart 2023). *Türk İşaret Dilinin Transfer Öğrenme Yöntemleriyle Sınıflandırılması*. Uluslararası Bilişim Kongresi Bildiriler Kitabı, Batman.

- Deepthi, R. S. and Sankaraiah, S. (2011, September). *Implementation of Mobile Platform Using QT and Opencv for Image Processing Applications*. 2011 IEEE Conference on Open Systems, Langkawi.
- Deng, L., Yu, D. (2013). Deep Learning: Methods and Applications. *Foundations and Trends® in Signal Processing*, 7(3-4), 197-387. doi: 10.1561/20000000039.
- Dhaya, R. (2020). Flawless Identification of Fusarium Oxysporum in Tomato Plant Leaves by Machine Learning Algorithm. *Journal of Innovative Image Processing (JIIP)*, 2(04), 194-201.
- Elen, A. (2020). Görüntü İkileştirme için Global Eşikleme Yöntemleri Üzerine Bir İnceleme. *Mühendislik Bilimleri ve Araştırmaları Dergisi*, 2(2), 38-49.
- Elkiran, H. (2020). *Occ-Opencv Kütüphanesi İçin Blok Tabanlı Programlama Aracı*. Yüksek Lisans Tezi, İstanbul Sabahattin Zaim Üniversitesi Bilgisayar Bilimi ve Mühendisliği, Yüksek Lisans Programı, İstanbul.
- Erdem Aytan, A., Öztürk, Y. ve Örgen, E. (1993). Görüntü İşleme. *İ.Ü. Diş Hekimliği Fakültesi Dergisi*, 27(4), 273-277.
- Erişti, E. (2010, 10-12 Şubat). *Görüntü İşlemede Yeni Bir Soluk, Opencv*. Akademik Bilişim'10, XII. Akademik Bilişim Konferansı Bildirileri, Muğla.
- Erkutlu, H., Erdemir Ergün, E., Köseoğlu, İ. ve Vurgun, T. (2023). Yapay Zekâ ve Örgütsel Davranış. *Nevşehir Hacı Bektaş Veli Üniversitesi Sosyal Bilimler Enstitüsü Dergisi*, 13(3), 1403-1417. DOI: 10.30783/nevsosbilen.1246678.
- Fan, Z., Lu, J., Gong, M., Xie, H. and Goodman, E. D. (2018). Automatic Tobacco Plant Detection in UAV Images via Deep Neural Networks. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 11(3), 876-887.
- Goyal, M. (2011). Morphological Image Processing. *IJCST*, 2(4), 59.
- Gögçe, M., Turgut K. ve Hocaoglu, A. K. (2021). *Spot Detection on Leaves*. 2021 29th Signal Processing and Communications Applications Conference (SIU), İstanbul.
- Gupta, V., Sengar, N., Dutta, M. K., Travieso, C. M. and Alonso, J. B. (2017, July). *Automated Segmentation of Powdery Mildew Disease from Cherry Leaves Using Image Processing*. 2017 International Conference and Workshop on Bioinspired Intelligence (IWOBI), Funchal. doi: 10.1109/IWOBI.2017.8006454.
- Güler, E. (2018). *Fındık Zarı Uygunluğunun Opencv ile Belirlenmesi İçin Yazılım ve Donanım Geliştirilmesi*. Yüksek Lisans Tezi, Selçuk Üniversitesi Fen Bilimleri Enstitüsü, Konya.
- Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., del Río, J. F., Wiebe, M., Peterson, P., Gérard-Marchant, P., Sheppard, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke, C. and Oliphant, T. E. (2020). Array Programming with NumPy. *Nature*, 585(7825), 357-362.

- He, K., Zhang, X., Ren S. and Sun J., (2016, June). *Deep Residual Learning for Image Recognition*. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas.
- Hlavatý, M., Kozáková, A. and Haffner, O. (2022, September). *Application for Python Programming Language Education Developed by Unity Engine*. International Conference Cybernetics & Informatics, Visegrád.
- Horasan, D. (2016). *Mermer Yüzeylerinin Görüntü İşleme Yöntemi ile Değerlendirilmesi*. Tezsiz Yüksek Lisans Bitirme Projesi, Pamukkale Üniversitesi Fen Bilimleri Enstitüsü, Denizli.
- İnternet 1: Diker, S. (2023). Raspberry Pi Headless Kurulum ve Konfigürasyon Süreçleri, URL: <https://www.cozumpark.com/raspberry-pi-headless-kurulum-ve-konfigurasyon-surecleri/>. Son Erişim Tarihi:10.12.2023.
- İnternet 2: Geliştirilmiş YOLOv5'i Temel Alan Python Tütün Yaprağı Hastalığı Tespit Sistemi, (b.t.). <https://blog.csdn.net/cheng2333333/article/details/126450401>. Son Erişim Tarihi: 30.10.2023.
- İnternet 3: Görüntü İşleme Teknolojisi. (2022) Yapay Zekâ, Teknocak. <https://teknocak.com/goruntu-isleme-teknolojisi/115>. Son Erişim Tarihi: 01.11.2023.
- İnternet 4: Çayıroğlu, İ. (2020), Görüntü İşleme Ders Notları. URL: http://www.ibrahimcayiroglu.com/Dokumanlar/GoruntuIsleme/Goruntu_Isleme_Ders_Notlari-1.Hafta.pdf. Son Erişim Tarihi: 01.11.2023.
- İnternet 5: Copeland, B. J. (2023). Artificial Intelligence. Encyclopedia Britannica. URL: <https://www.britannica.com/technology/artificial-intelligence> Son Erişim Tarihi: 2.8.2024.
- İnternet 6: Chakraborty, D. (2021). OpenCV Contour Approximation, URL: <https://pyimagesearch.com/2021/10/06/opencv-contour-approximation/>. Son Erişim Tarihi: 11.12.2023.
- İnternet 7: Hastalık Zararlılar ve Arızalar Faktörü, (b.t.). Tütün Eksperleri Derneği. <http://www.tutunekspers.org.tr/kaynaklar/hastaliklar/hastalik-zararlılar-ve-arizalar-faktoru>. Son Erişim Tarihi: 26.10.23.
- İnternet 8: Hunter, J. D. (2008). History (Project Information). The Matplotlib. <https://matplotlib.org/stable/users/project/history.html>. Son Erişim Tarihi:11.12.2023.
- İnternet 9: Hunter, D. (2018). Transfer Öğrenimi (Transfer Learning). Dev Hunter Yapay Zekâ, Robotik ve Sinirbilim. <https://devhunteryz.wordpress.com/2018/07/28/transfer-ogrenimi-transfer-learning>. Son Erişim Tarihi: 26.10 2023.
- İnternet 10: Lutkevich, B. (2020), Embedded System. URL: <https://www.techtarget.com/iotagenda/definition/embedded-system>. Son Erişim Tarihi: 1.12.2023.

- İnternet 11: Niamh, B. (2023). Connecting to a computer from the vnc connect portal. URL: <https://help.realvnc.com/hc/en-us/articles/9987343661469-Connecting-to-a-computer-from-the-VNC-Connect-Portal>. Son Erişim Tarihi:10.12.2023.
- İnternet 12: Tao, K. (2021), The AUC-ROC Curve. URL: https://udrc.ushe.edu/news/2021/research_skills/20210407AucRocCurve.html. Son Erişim Tarihi: 11.12.2023.
- İnternet 13: TIOBE Index for July 2024. <https://www.tiobe.com/tiobe-index>. Son Erişim Tarihi:3.8.2024.
- İnternet 14: Varshney, P. (2020). VGGNet-16 Architecture: A Complete Guide. URL: <https://www.kaggle.com/code/blurredmachine/vggnet-16-architecture-a-complete-guide>. Son Erişim Tarihi: 29.11.2023.
- İnternet 15: VideoCapture Class Reference, Class For Video Capturing From Video Files, Image Sequences or Cameras. Open Source Computer Vision. https://docs.opencv.org/4.x/d8/dfe/classcv_1_1VideoCapture.html. Son Erişim Tarihi: 11.12.2023.
- İnternet 16: Zipponi, F. (2021). Programmation MicroPython sans fil avec Thonny. URL: <https://10-raisons.fr/tr/programmation-micropython-sans-fil-avec-thonny/>. Son Erişim Tarihi:10.12.2023.
- Jimeno-Morenilla, A., Pujol, F. A., Molina-Carmona, R., Sánchez-Romero, J. L. and Pujol, M. (2014). Trajectory-based morphological operators: a model for efficient image processing. *The Scientific World Journal*, vol. 2014, 1-9.
- Jolles, J. W. (2021). Broad-scale Applications of the Raspberry Pi: A Review and Guide for Biologists. *Methods in Ecology and Evolution*, 12(9), 1562-1579.
- Karhan, M., Oktay, M.O., Karhan, Z. ve Demir, H. (2011, 16-18 Mayıs). *Morfolojik Görüntü İşleme Yöntemleri ile Kayıslarda Yaprak Delen (Çil) Hastalığı Sonucu Oluşan Lekelerin Tespiti*. 6th International Advanced Technologies Symposium (IATS'11), Elazığ.
- Kılıç, Ş. (2023). *Derin Öğrenme Yöntemleri Kullanılarak Giyilebilir Sensörlerden Kişi Tanıma*, Doktora Tezi, Ankara Üniversitesi Fen Bilimleri Enstitüsü, Ankara.
- Kumar G. ve Bhatia, P. K. (2014, February). *A Detailed Review of Feature Extraction in Image Processing Systems*. Fourth International Conference on Advanced Computing & Communication Technologies, Rohtak.
- Laily, D. (2013). Deteksi penyakit pada daun tembakau dengan menerapkan algoritma artificial neural network. *Simetris Jurnal Teknik Mesin Elektro dan Ilmu Komputer*, 3(1), 51-58.
- Lin, J., Chen, Y., Pan, R., Cao, T., Cai, J., Yu, D., Chi, X., Cernava, T., Zhang, X. and Chen, X. (2022). *CAMFFNet: A novel convolutional neural network model for tobacco*

- disease image recognition*. Computers and Electronics in Agriculture, 202(11), 107390.
- Luo, W., Xing, J., Milan, A., Zhang, X., Liu, W. and Kim, T.-K. (2021). Multiple Object Tracking: A Literature Review. *Artificial Intelligence*, 293,103448. doi: 10.1016/j.artint.2020.103448.
- Mathe, S. E., Pamorthy, A. C., Kondaveeti, H. K. and Vappangi, S. (2022, February). *A Review on Raspberry Pi and its Robotic Applications*. 2nd International Conference on Artificial Intelligence and Signal Processing (AISP), Vijayawada.
- Mellit, A., Benghanem, M., Kalogirou, S. and Pavan, A. M. (2023). An embedded system for remote monitoring and fault diagnosis of photovoltaic arrays using machine learning and the internet of things. *Renewable Energy*, 208, 399-408.
- Oktan, E. ve Güzel, B. (2022). Tarım ve Yapay Zekâ. M. Bilen (Ed.), *Yapay Zekanın Değiştirdiği Dinamikler*, (s. 199-224). Eğitim Yayınevi.
- Othman, O. A. O. (2020). *Evreşimsel Sinir Ağları Tabanlı Otomatik Görüntü Renklendirme*. Yüksek Lisans Tezi, Selçuk Üniversitesi Fen Bilimleri Enstitüsü, Konya.
- Özdemir, F. M. (2018). *Gömülü Sistemlerde Yüz Tanıma Algoritmalarının Karşılaştırılması*. Yüksek Lisans Tezi, Afyon Kocatepe Üniversitesi Fen Bilimleri Enstitüsü, Afyon.
- Öztürk, K. ve Şahin, M. E. (2018). Yapay Sinir Ağları ve Yapay Zekâ'ya Genel Bir Bakış. *Takvim-i Vekayi*, 6(2), 25-36.
- Peker, M. (2009). *Görüntü İşleme Tekniği Kullanılarak Gerçek Zamanlı Hareketli Görüntü Tanıma*. Yüksek Lisans Tezi, Sakarya Üniversitesi Fen Bilimleri Enstitüsü, Sakarya.
- Prasetyo, E., Suciati, N. and Fatichah, C. (2022). Multi-level residual network VGGNet for fish species classification. *Journal of King Saud University-Computer and Information Sciences*, 34(8), 5286-5295.
- Ranjani, J., Sheela, A. and Meena, K. P. (2019, April). *Combination of NumPy, SciPy and Matplotlib/Pylab -a good alternative methodology to MATLAB - A Comparative Analysis*. 2019 1st International Conference on Innovations in Information and Communication Technology (ICIICT)
- Sandler, M., Howard, A., Zhu, M., Zhmoginov, A. and Chen, L.- C. (2018, June). *MobileNetV2: Inverted Residuals and Linear Bottlenecks*. IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA.
- Sarı, C. (2023). *Python Programlama Dili Kullanılarak Günlük Maksimum Yağış Verilerinin Trend ve Risk Analizleri*. Yüksek Lisans Tezi, İstanbul Teknik Üniversitesi Lisansüstü Eğitim Enstitüsü, İstanbul.
- Shao, Y., Xuan, G., Zhu, Y., Zhang, Y., Peng, H., Liu, Z. and Hou, J. (2017). Research on automatic identification system of tobacco diseases. *The Imaging Science Journal*, 65(4), 252-259.

- Sindhu, D. (2020). Prediction Of Disease In Tobacco Leaf Using Deep Belief Network. *ICTACT Journal on Data Science and Machine Learning*, 2(1), 133-136.
- Soni, A., Soni J. K. and Hota S. (2021, September). *Detection of Multiple Crop Diseases using Image Processing Techniques*. 2021 IEEE High Performance Extreme Computing Conference (HPEC), Waltham. doi: 10.1109/hpec49654.2021.9622812.
- Sun, Y., Wang, H. Q., Xia, Z. Y., Ma, J. H. and Lv, M. Z. (2020). Tobacco-disease Image Recognition via Multiple-Attention Classification Network. *Journal of Physics: Conference Series*, vol. 1584, 012008.
- Swasono, D. I., Tjandrasa, H. And Fathicah, C. (2019, July). *Classification of tobacco leaf pests using VGG16 transfer learning*. 12th International Conference on Information & Communication Technology and System (ICTS), Surabaya.
- Şahin, G. ve Taşlıgil, N. (2013). Türkiye’de tütün (Nicotiana Tabacum L.) Yetiştiriciliğinin Tarihsel Gelişimi ve Coğrafi Dağılımı / Le Developpement Historique et la Dispersion Géographique de la Cultivation de Tabac en Turquie. *Doğu Coğrafya Dergisi*, 18(30), 71-102.
- Tan, M. and Le, Q. V. (2019, June). *EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks*. Proceedings of the 36th International Conference on Machine Learning, ICML 2019, Long Beach, 9-15 2019, 6105-611436. Uluslararası Makine Öğrenimi Konferansı Bildirileri, ICML 2019, Long Beach.
- Tiryaki, S. H. ve Balaman, F. (2021). Açık Kaynak Kodlu Yazılımlardan Scratch, Arduino ve Python Kullanımı Hakkında Öğrenci Görüşleri. *Journal of Computer and Education Research*, 9(18), 831-852.
- Toraman, S. (2016). *Kızılötesi Spektroskopisi Kullanılarak Kan Örneklerinden Kolon Kanserine Yönelik Özellik Çıkarımı*, Doktora Tezi, Fırat Üniversitesi Fen Bilimleri Enstitüsü, Elazığ.
- Türk, F. ve Lüy, M. (2021). Gömülü Sistemler ve Mühendislikte Uygulama Alanları. *Uluslararası Mühendislik Araştırma ve Geliştirme Dergisi (UMAGD) / International Journal of Engineering Research and Development (IJERAD)*, 13(3), 256-265.
- Türkmen, A. (2021). *Konvolüsyonel Yapay Sinir Ağları ve Görüntü İşleme Teknikleri ile Biyomedikal Görüntülerden Diyabetik Retinopati Teşhisi*. Yüksek Lisans Tezi, Ankara Üniversitesi Fen Bilimleri Enstitüsü, Ankara.
- Türkoğlu, M (2019). *Görüntü İşleme Tabanlı Bitki Türleri ve Hastalıkları Tanıma*, Doktora Tezi, İnönü Üniversitesi Fen Bilimleri Enstitüsü, Malatya.
- Ulutürk, C. ve Uğur, A. (2012, 2-4 Temmuz). Recognition of leaves based on morphological features derived from two half-regions. International Symposium on Innovations in Intelligent Systems and Applications (INISTA), Trabzon.
- Ünal, Y. (2015). *Makine Öğrenmesi Yöntemleriyle Bel Bölgesi Rahatsızlıklarının Tanısı*, Doktora Tezi, Selçuk Üniversitesi Fen Bilimleri Enstitüsü, Konya.

- Wang, X., Kondratyuk, D., Kitani, K. M., Movshovitz-Attias, Y. and Eban E. (2020). Multiple Networks are More Efficient than One: Fast and Accurate Models via Ensembles and Cascades. *Computing Research Repository (CoRR)*, December 2020.
- Xu, M., Li, L., Cheng, L., Zhao, H., Wu, J., Wang, X., Li, H. and Liu, J. (2022). Tobacco Leaves Disease Identification and Spot Segmentation Based on the Improved ORB Algorithm. *Scientific Programming*, vol. 2022, 1-12.
- Yılmaz, T. (2019). *Raspberry Pi Kartı Kullanarak Nesne Tespit ve Takip Robotunun Tasarımı ve Modellenmesi*. Yüksek Lisans Tezi, Fırat Üniversitesi Fen Bilimleri Enstitüsü, Elazığ.



ÖZGEÇMİŞ

Adı-Soyadı : Cemil ERGİN

Bilimsel Faaliyetler (Yayınlar, Bildiriler, Katıldığı Projeler)

- 1-) Ergin, E., & Bükücü, Ç. C. (2024, Mart). *Gömülü sistem tasarımı ile tütün yapraklarındaki hastalıklı alanların tespit edilmesi*. In 5. International Mediterranean Scientific Research Congress (s. 1159). Mersin, Türkiye.

