



T.C.
ULUDAĞ ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ



Görüntü İşlemeyle Zararlı Böceklerin
Ve Bitki Hastalığının Tespiti

Muhammed Celal Yılmaz
031890020

Ecenur Karakaya
031890003

Tunahan Top
031890062

TASARIM DERSİ FİNAL RAPORU
2023

T.C.
ULUDAĞ ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

Görüntü İşlemeyle Zararlı Böceklerin
Ve Bitki Hastalığının Tespiti

Muhammed Celal Yılmaz
031890020

Ecenur Karakaya
031890003

Tunahan Top
031890062

Projenin Danışmanı: Doç. Dr. Pınar Kırıcı

Jüri Üyesi:

Jüri Üyesi :

ÖZET

Tarım insanoğlunun her zaman temel geçim kaynağını oluşturmuştur. Artan nüfusla birlikte gelen gıda sorunu ve ürünlerin çeşitli etkenlerden dolayı zarar görmesi bir problemdir. Ürünlerin kalite ve verimlerinin yükseltilmesi ve yetiştirme sürecinde ürünlerin muhafazası ve yetiştirilen ortamın kalitesi tarımın gelişmesi açısından çok önemlidir. Teknolojinin gelişmesi ve birçok alanda hayatımıza girmesiyle beraber tarım alanında da gelişmeler yaşanmıştır. Geleneksel tarım yerini çeşitli parametrelerin akıllı sistemlerle beraber kontrol edildiği akıllı sera sistemlerine bırakıyor. Çalışmamız, yetiştirdiğimiz bitkinin hastalık durumunu ve hastalık yapıcı mikroorganizmaları görüntü işleme ve makine öğrenmesi teknolojileriyle tespit eden bir uyarı sistemidir. Raporumuzda bu konuya yönelik akademik makale ve literatür çalışmalarını inceledik.

ABSTRACT

Agriculture has always been the main source of livelihood for mankind. The food problem that comes with the increasing population and the damage of the products due to various factors is a problem. Increasing the quality and yield of the products and the quality of the products during the growing process and the quality of the growing environment are very important for the development of agriculture. With the development of technology and its entry into our lives in many areas, there have been developments in the field of agriculture. Traditional agriculture is leaving its place to smart greenhouse systems where various parameters are controlled together with intelligent systems. Our work is a warning system that detects the disease status of the plant we grow and disease-causing microorganisms with image processing and machine learning technologies. In our report, we examined academic articles and literature studies on this subject.

İÇİNDEKİLER

	<u>Sayfa No</u>
ÖZET.....	iii
ABSTRACT.....	iii
İÇİNDEKİLER.....	iv
ŞEKİLLER DİZİNİ.....	vi
TABLolar.....	vii
1. GİRİŞ	1
2. Sera Sistemi	3
3. Tarla Bitkilerinde Böcek Sınıflandırılması ve Tespiti	3
3.1 Materyal ve Metod	4
3.2 Haşere Tespiti İçin Bir Görüntü İşleme Algoritması	5
4. Görüntü İşlemeyle Bitki Hastalık Tespiti Aşamaları	8
5. Meyve Üretiminde Görüntü İşleme Teknikleri	9
6. Görüntü İşlemeyle Marul Bitkisinin Sağlık Durumu	15
7. Kayısı Hastalık Tespiti	17
8. Görüntü İşleme Teknikleri İle Bağ Yaprak Uyuzu Hasarının Belirlenmesi	17
9. Şeker Pancarı Yaprak Leke Hastalığı Örneği	18
10. Tarımda Görüntü İşleme ve Literatür Çalışma Sonuçları	19

10.1	Görüntü işleme	20
10.2	İlgili Literatür ve Çalışmaların İncelenmesi	20
10.3	Malzemeler ve Yöntemler	21
10.4	Deney Sonuçları	23
10.5	Sonuç	24
11.1	Genel Bulgular	24

PROJEDE KULLANILAN TEKNOLOJİLER VE PROJE METODU 26

1.1	Makine Öğrenmesi	26
1.2	Derin Öğrenme	26
1.3	Tensorflow	27
1.4	Python	27
1.5	Anaconda	28
1.6	Keras	28
1.7	Görüntü işleme	28
1.8	Veri Seti	28
1.9	Çalışmada kullanılan fiziksel donanım	29
2.1	Ön İşlem	30
2.2	Image Augmentation (Veri Çoğaltma)	30
2.3	Data Cleaning (Veri Temizleme)	34
2.4	CNN (Evrşimli Sinir Ağları)	34

Proje Aşamaları 35

13.	Kaynakça	52
14.	Teşekkür	56

ŞEKİLLER DİZİNİ

	<u>Sayfa No</u>
Şekil 1.1	Akıllı bir sistemin tipik bir modelidir 3
Şekil 1.2	Otomasyon sistemiyle kontrol edilen bir arayüz örneğidir 3
Şekil 1.3	Döndürme işlemiyle elde edilen böceklerdir 4
Şekil 1.4	CNN modeliyle böcek sınıflandırma aşamaları 5
Şekil 1.5	Sınıflandırma sonrası çıktıları 5
Şekil 1.6	Akış şeması 6
Şekil 1.7	Böcek resimleri 6
Şekil 1.8	İki seviyeli eşiklemenin akış şeması 7
Şekil 1.9	Parazit çıktı 7
Şekil 2.1	Parazit sonrası 8
Şekil 2.2	Back propagation yöntemi 9
Şekil 2.3	ANN (Artificial Neural Network) yöntemi 9
Şekil 2.4	Kiraz tespiti 10
Şekil 2.5	Elmarın boyutlarına göre çıkan sonuçlar 11
Şekil 2.6	Görüntü ön işleme aşaması 12
Şekil 2.7	K-means ile fındık tespiti 12
Şekil 2.8	Resim üzerinde yapılan işlem 13
Şekil 2.9	Gül sapı örnek çıktı 14
Şekil 3.1	Uzaklık ölçme metodu 15
Şekil 3.2	Tipik bir marul tespit modeli 16
Şekil 3.3	Modelin marul tespit aşaması 17
Şekil 3.4	Mimari tasarım 21
Şekil 3.5	Output 22
Şekil 3.6	Medyan filtresi 23
Şekil 3.7	Deney Sonuçları 24
Şekil 3.9	Marul bitkisinde uç yanıkların ısı haritası 25

Şekil 4.1	DL ve ML arasındaki ilişki	27
Şekil 4.2	Sırasıyla sağlıklı, hastalıklı ve üzerinde böcek olan marul fotoğrafları	29
Şekil 4.3	Görüntü ölçeklendirme	30
Şekil 4.4	Görüntü kırpma	31
Şekil 4.5	Görüntü çevirme	31
Şekil 4.6	Rastgele çevirme	32
Şekil 4.7	Parlaklık	32
Şekil 4.8	Kontrast	33
Şekil 4.9	Görüntü yakınlaştırma	33
Şekil 5.1	Evrişimsel sinir ağı katmanları	34
Şekil 5.2	Maydanoz ve lahana bitkisi yetiştirme aşamaları	35
Şekil 5.3	Fesleğen bitkisi yetiştirme aşamaları	36
Şekil 5.4	Marul bitkisi yetiştirme aşamaları	36
Şekil 5.5	Sağlıklı marul bitkisi örnek	37
Şekil 5.6	Hastalıklı marul bitkisi örnek	37
Şekil 5.7	Pest detected etiketine örnek	37
Şekil 5.8	Projemizin genel klasör yapısı	38
Şekil 5.9	Değerlerin belirlenmesi ve veri setine yükleme kısmı	39
Şekil 6.1	Train,test ve validation değerlerini oranlama	39
Şekil 6.2	Model mimarisi oluşturma ve data augmentation kısmı	40
Şekil 6.3	Model özeti	41
Şekil 6.4	Epoch kısmı	41
Şekil 6.5	Modelin çıktıları	42
Şekil 6.6	Eğitilen modelin sonuçlarını gösterme	43
Şekil 6.7	Modelin kaydedilmesi	43
Şekil 6.8	Pycharm ıde kullanarak yaptığımız backend dosyası	44
Şekil 6.9	Postman ile API doğruluğunu test etme	45

Şekil 7.1 Resim yükleme kısmımız ve diğer dosyalar	46
Şekil 7.2 Projenin arayüz kısmı	47
Şekil 7.3 Hastalıklı etikete sahip marul örneği	47
Şekil 7.4 Hastalıklı etikete sahip marul örneği	48
Şekil 7.5 Hastalıklı etikete sahip marul örneği	48
Şekil 7.6 Sağlıklı etikete sahip marul örneği	49
Şekil 7.7 Sağlıklı etikete sahip marul örneği	49
Şekil 7.8 Pest detected etiketine sahip marul örneği	50
Şekil 7.9 Pest detected etiketine sahip marul örneği	50
Şekil 8.1 Pest detected etiketine sahip marul örneği	51
Şekil 8.2 Pest detected etiketine sahip marul örneği	51

Tablolar Dizini

Sayfa No

Tablo 1.1	Tensorflow ile eğitilen veri setinin çıktıları	17
Tablo 1.2	Veri setinin sınıflara göre resim sayıları	29
Tablo 1.3	Donanımsal bilgiler	29
Tablo 1.4	Modelin doğruluk ve kayıp grafiği	42

1. GİRİŞ

Tarım insanoğlunun her zaman temel geçim kaynağını oluşturmuştur. Artan nüfusla birlikte gelen gıda sorunu ise halen üzerine düşülmesi gereken bir problemdir. Ürünlerin kalite ve verimlerinin yükseltilmesi ve yetiştirme sürecinde ürünlerin muhafazası ve yetiştirilen ortamın kalitesi tarımın gelişmesi açısından çok önemlidir. Teknolojinin gelişmesi ve birçok alanda hayatımıza girmesiyle beraber tarım alanında da gelişmeler yaşanmıştır. Geleneksel tarım yerini çeşitli parametrelerin akıllı sistemlerle beraber kontrol edildiği akıllı sera sistemlerine bırakıyor. İklim krizi ve birçok etkenden dolayı bu akıllı sistemlerin kullanılması son derece önemlidir. Bitkilerin zararlı organizmalar ve diğer etkenlerden olumsuz etkilenmemesi adına çalışmalar devam etmektedir. Bilgisayar biliminin önemli disiplinleri arasında yer alan görüntü işleme ve makine öğrenmesi konuları yardımıyla bu sorunu ortadan kaldıracabiliriz.

Görüntü işleme teknikleri, her tür imgeye değişen başarı oranlarında pratik olarak uygulanmaya çalışılmaktadır. İlk görüntü işleme uygulaması 1964'de ABD'de uydu görüntüsünün gürültüsünden kaynaklanan lekeleri temizlemek için yapıldı. Tüm çok disiplinli bilim alanlarında olduğu gibi, görüntü işleme de hatalı bilgi sorunları yaşamaktadır. Bu disiplin optik, elektronik, matematik, fotoğraf bilimi, bilgisayar teknolojisini içinde barındıran oldukça geniş bir yelpazeye sahiptir. (Yılmaz ve diğerleri, 2018)

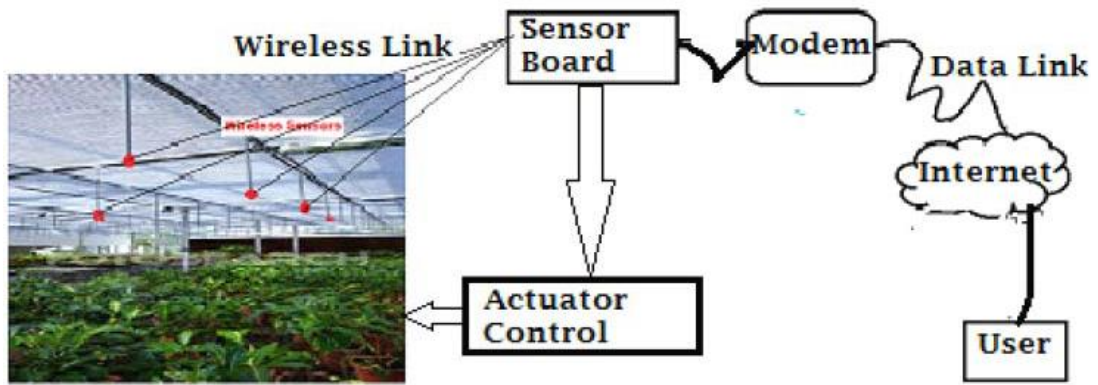
Görüntü işleme dijital ortama aktarılmış görseller üzerinden ihtiyaca göre bilgiler elde etmek için farklı tekniklerle kullanılan bir yöntemdir. Görüntü işleme, kaydedilmiş olan belirli görüntülerin işlenip mevcut resim ve grafiklerin iyileştirilmesi ya da karşılaştırılması için kullanılmaktadır. Günümüzde alınan görüntüler değişik programlarla işleme tabi tutulmaktadır. İşlem neticesinde kullanıcının kullandığı yöntemlere göre farklı sonuçlar ortaya çıkmaktadır. Görüntü işleme sistemlerin maliyeti azaltan en önemli özelliklerinden biri gerçek zamanlı görme işlemidir. Gerçek zamanlı görme, maliyeti düşürdüğü gibi, insandan daha hızlı işlem yapması ve güvenilirliği sayesinde tercih edilen bir sistem olmaktadır. Görüntü işleme yöntemleri, günümüzde 2 ve 3 boyutlu yüzey ölçümü ve modellenmesi gibi pek çok işlemde kullanılmaktadır. (Yılmaz ve diğerleri, 2018)

Dünyada birçok farklı alanda kullanılabilen görüntü işleme hızla gelişen teknolojiler arasında yer almaktadır. Aynı zamanda mühendislik ve bilgisayar bilimleri gibi disiplinlerin temel araştırma alanlarından biridir. Görüntü işleme bilgisayar yardımıyla görüntülerin işlenmesi ve yorumlanmasıdır. Görüntü işleme tıp, kriminoloji, savunma sanayi gibi alanlar dışında kalite kontrol, robotlar, kaynak sistemleri ve ölçmede de kullanılır. Tasarlanan otomasyon sistemleri ile sınıflandırma işlemi eskiye göre daha hızlı, pratik ve kaliteli bir şekilde yapılmaktadır. Buda ülkemiz endüstrisi ve işgücü kalitesini üst noktaya çıkarmada olumlu bir etki doğuracaktır. Görüntü işlemenin ilk ve en önemli faktörü ışıktır. Bilgisayar vb. elektronik donanımlarla alınan görüntünün işlenebilmesinde, ortamdaki ışık faktörü uygun olmadan, başarılı sonuçlara ulaşılamamaktadır. Özellikle açık havada, güneş ışığı gibi farklı açılardan gelen bir ışık söz konusu olduğunda, bu durum sonuçlara çok büyük ölçüde etki etmektedir. Ancak görüntüsü alınacak malzemeye gelen ışık faktörü sabit tutulabildiğinde, daha başarılı sonuçlar alınabilmektedir. Bu halde de, siyah beyaz değil, dijital kameraların dışarıdan gelen ışık faktöründen daha kolay bir şekilde ayırt edebileceği renklerin kullanılması, sonuçların doğruluk oranını arttırmaktadır. Ayrıca yüksek çözünürlüklü kameraların kullanılması ile kalite kontrolünün çok daha başarılı bir şekilde yapılabilceği açıktır. (Ağın ve diğerleri, 2016)

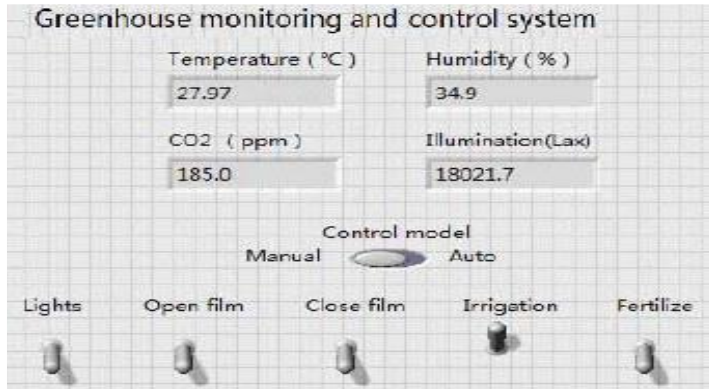
Tarımın gelişmesinde bilgisayar teknolojilerini kullanmak ve ürünlerin sürdürülebilirliğini, verimini ve muhafazasını sağlamak son derece önemlidir. Çiftçilerin ürünlerdeki zararlı organizmaları ve bitkilerdeki hastalık durumlarını tespit etmesi zordur. Bu noktada görüntü işleme, derin öğrenme gibi uygulamalar devreye girmektedir. Ayrıca tasarruf açısından bu teknolojileri kullanmak maliyet açısından da uygun olacaktır. Mesela etkili su kullanımı amaçlayan bir çalışma aynı zamanda gübre kullanımının azaltılmasına da katkı sağlayabilmektedir. (Ağın ve diğerleri, 2016)

2. Akıllı Sera Tasarımı Ve IOT Tabanlı İzleme Sistemi

Sera, bitkilerin yetişmesi için uygun şartların kontrol edilip düzenlendiği örtülü bir ortamdır. Akıllı sera sistemi ise IOT (Nesnelerin İnterneti) tabanlı ve otomasyon ile sera sistemini kontrol ettiğimiz sera çeşididir. Bu şekilde akıllı teknolojilerin kullanılması üretim verimini ve kontrolünü arttıran bir unsurdur. (Hu ve diğerleri, 2020)



Şekil 1.1 akıllı bir sistemin tipik bir modeli , (Hu ve diğerleri,2020)



Şekil 1.2 Otomasyon sistemiyle kontrol edilen bir arayüz örneği
(Hu ve diğerleri,2020)

3. Tarla Bitkilerinde Böcek Sınıflandırması ve Tespiti

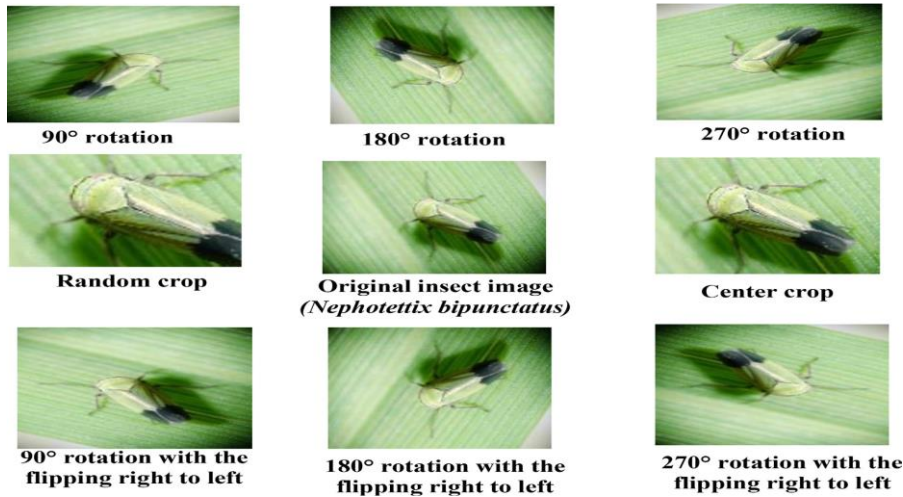
Tarım sektöründe modern makine öğrenimi tekniklerini kullanmak gıda ihtiyacını karşılamak için büyük bir öneme sahiptir. Böcek tespiti, çiftçiler için zorlu bir süreçtir ve uzman

taksonomistlere ihtiyaç duyar. ANN, SVM, KNN, NB ve CNN gibi makine öğrenme teknikleri böcek tespitinde bize yardımcı olur. Ayrıca Keras ve Tenserflow gibi teknolojiler, tespit ve sınıflandırma konusunda bize yardımcı olan frameworklere örnektir.

(Kasinathan ve diğerleri, 2021)

3.1 Materyal ve Metodlar :

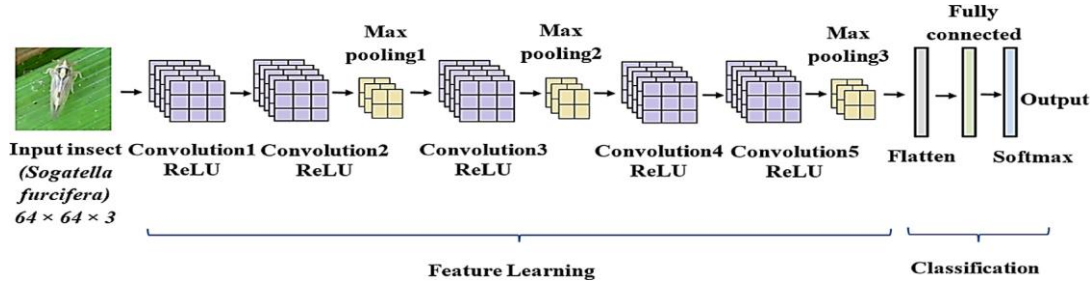
- 1- **Sınıflandırma İçin Böcek Veri Seti:** Bu makalede farklı bitki türleri için Wang ve Xie veri seti kullanılmıştır. Bu veri setini kullanarak çalışmalar yapılmıştır.
- 2- **Görüntü Ön İşleme:** Görüntülerdeki gürültüyü azaltmak ve keskinleştirmek için uygulanır.
- 3- **Görüntü büyütme:** Böcek görüntüleri yeni bir piksel boyutuna ayarlanır ve üzerinde büyütme, döndürme ve kırpma gibi işlemler gerçekleştirilir. (Kasinathan ve diğerleri, 2021)



Şekil 1.3 Döndürme işlemiyle elde edilen böcekler , (Kasinathan ve diğerleri,2021)

- 4- **Böcek Sınıflandırma:** Böcekleri sınıflandırmak için CNN (Evrişimli Sinir Ağı), SVM (Destek Vektör Makinesi), KNN (k-en yakın komşular), ANN (Yapay Sinir Ağları), NB (Naive Bayes) gibi teknolojiler kullanılmıştır.
- 5- **Böcek Tespiti:** Böcek sınıflandırma için modellerimizi kullanırız.

(Kasinathan ve diğerleri, 2021)



Şekil 1.4 CNN modeliyle böcek sınıflandırma aşamaları , (Kasinathan ve diğerleri,2021)

6- Sınıflandırmanın Doğruluğu: Modelin sınıflandırma doğruluğunu hesaplamak için $TP+TN/ TP+TN+FP+FN$ kullanılır.

TP, FP, FN ve TN doğru pozitif, yanlış temsil eder. böcek görünür halde doğru sınıflandırıldıysa TP olarak kabul edilir. Aksi takdirde FN olarak kabul edilir. Mevcut olmayan böcek sınıflandırma ise TN olarak kabul edilir. Aksi halde FP olarak Kabul edilir. Sınıflandırma modellerinin doğruluk sıralaması $NB < ANN < KNN < SVM < CNN$ şeklinde ifade edilir. Önerilen model CNN olmakla birlikte hesaplama süresi diğerlerine göre daha uzun sürmüştür. (Kasinathan ve diğerleri, 2021)



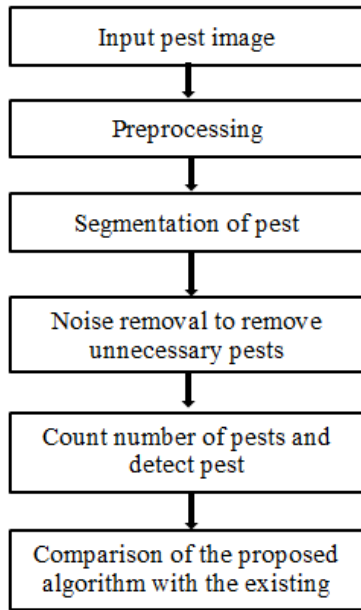
Şekil 1.5 Sınıflandırma sonrası çıktılar , (Kasinathan ve diğerleri,2021)

3.2 Haşere Tespiti İçin Bir Görüntü İşleme Algoritması

Haşere istilası, tarımda verimi düşüren önemli sorunlardandır .Çiftçilerin bu sorunu önlemek için kullandığı böcek ilaçları ise bazı sağlık sorunlarını beraberinde getirir. Bu makalede bunu önlemek için verimli bir algoritma geliştirmekten bahsediliyor. Bu sorunu

çözmek için bir robot geliştirilebilir ve uygun ilaç kullanımıyla bu sorun ortadan kaldırılabilir. Görüntü işleme yöntemiyle zararlının 3D görüntüsü elde edilir. Modern tekniklerle izlenen serada haşere tespiti için segmentasyon tekniği kullanılmış. Burada entropi tabanlı eşikleme önerilmiş ve sonuçların karşılaştırması Fuzzy c-ortalama yöntemi ile yapılmıştır. Genişletilmiş bölge büyütme algoritması kullanılmıştır. Görüntüler veri tabanından alınmış ve uygun renk modeli belirlenmiştir. Doygunluk renk modeli en uygun renk modeli olarak seçilmiştir. Otsu ve manuel eşikleme denenmiş ve en iyi sonuçlar manuel eşiklemede alınmıştır.

(N. Pavithra ve diğerleri, 2017)



Şekil 1.6 Akış şeması

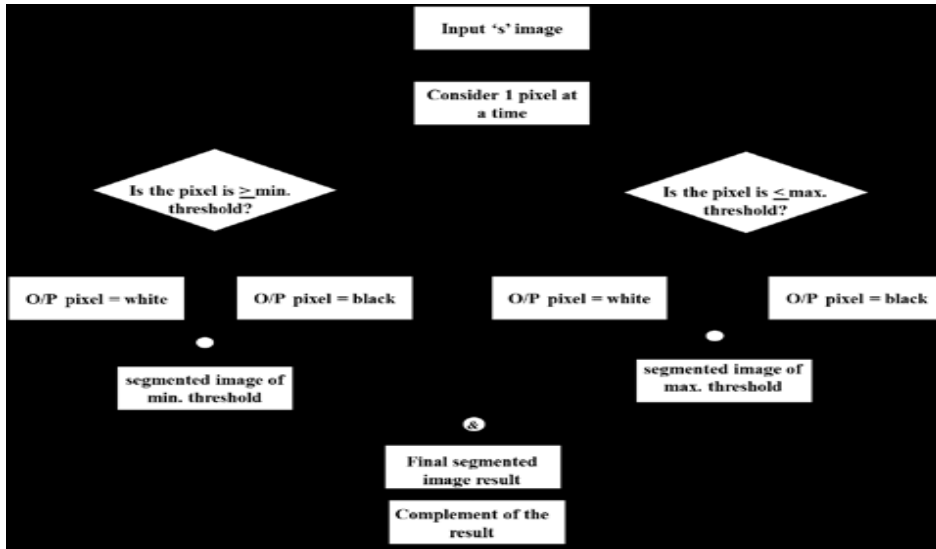
(N.Pavithra ve diğerleri,2017)



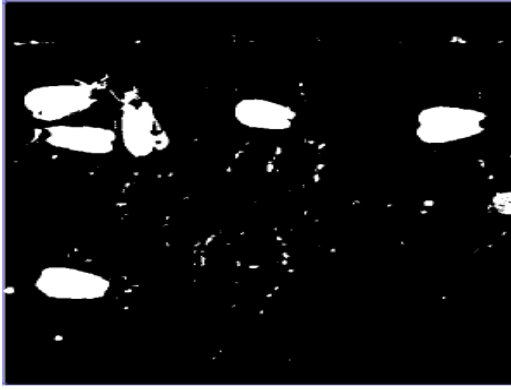
Şekil 1.7 Böcek resimleri

(N.Pavithra ve diğerleri,2017)

İki seviyeli eşiklemenin akış şeması ise Şekil-1.8’de yer almaktadır.



Şekil 1.8 İki seviyeli eşikleme akış şeması , (N.Pavithra ve diğerleri,2017)



Şekil-1.9 Parazit çıktısı (N.Pavithra ve diğerleri,2017)

Gereksiz pikselleri kaldırmak için gürültü giderme işlemi yapılır. Şekil-2.1'de parazitin gürültü giderildikten sonra elde edilen görüntüsünü göstermektedir. Algoritma önce MATLAB R2011b'de test edilmiş ve ardından verilog kullanılarak FPGA'da da aynısı uygulanmıştır. (N. Pavithra ve diğerleri, 2017)



Şekil 2.1 Gürültü giderme işleminden sonraki çıktılar. (N.Pavithra ve diğerleri,2017)

4. Görüntü İşlemeyle Bitki Hastalık Tespiti Aşamaları

Bitki hastalığı tespiti için temel adımlar ve görüntü işleme kullanılarak sınıflandırma aşamamızı şu şekilde özetleyebiliriz:

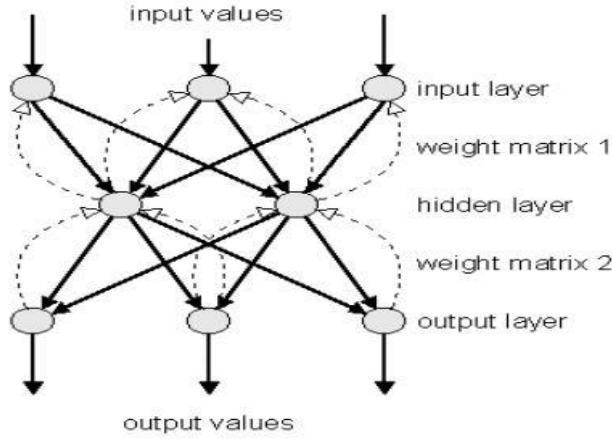
Image Acquisition (Görüntü Edinme): Bitki yaprağının görüntüleri kamera aracılığıyla yakalanır. Bu görüntü RGB (Kırmızı, yeşil ve mavi) formundadır. Burada renk dönüşüm yapısı uygulanır.

Image Pre-processing (Görüntü Ön İşleme): Görüntüdeki paraziti gidermek ve diğer nesneleri kaldırmak için görüntü işlenir. Kırpma, yumuşatma, contrast artırma, RGB görüntüleri gri görüntülere dönüştürme, histogram eşitleme gibi yöntemler kullanılır.

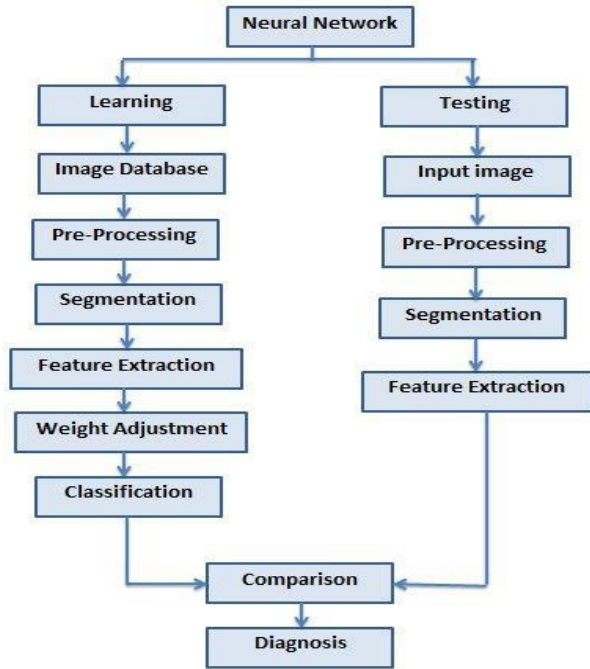
Image Segmentation (Görüntü Bölme): Segmentasyon, görüntünün çeşitli bölümlerine bölünmesi anlamına gelir. Sınır ve nokta algılama algoritmasını kullanarak segmentasyon ile RGB görüntüsü, HIS modeline dönüştürülür, K-kümeleme algoritması ile nesnenin sınıflandırılması için kullanılır, otsu eşik algoritması ile eşikleme, gri seviyeli görüntülerden ikili görüntüler oluşturur.

Feature Extraction (Özellik Çıkarma): Öznitelik çıkarımı, tanımlama için önemlidir. Görüntü işlemenin birçok uygulamasında özellik çıkarma kullanılır. Renk, doku, morfoloji, kenarlar bunlara örnektir. (S. D. Khirade ve diğerleri, 2015)

Sınıflandırma: ANN (Yapay Sinir Ağları) ve Back propagation (Geri Yayılma) kullanılarak sınıflandırma yapılır ve nihayetinde model test edilir.



Şekil 2.2 Back propagation yöntemi , (S. D. Khirade ve diğerleri,2015)



Şekil 2.3 ANN (Artificial Neural Network) prensibi , (S. D. Khirade ve diğerleri,2015)

5. Meyve Üretiminde Görüntü İşleme Teknikleri

Küreselleşen dünyada meyve ticaretinin en önemli unsurlarından birisi ürün kalitesidir. Dünya pazarında farklı toplumların farklı boy ve renkteki meyve tercihleri bilinmektedir. Gelişen teknolojinin bu taleplerin karşılanmasında kullanılması son derece doğaldır. Görüntü işleme tekniği kullanılarak, günümüzde özellikle endüstriyel tarım alanında başarılı sonuçlar elde edilmiştir. Bitkilerin dış yüzey hastalıkları, hataları, renklerinin belirlenen özelliklere uygunluğu kontrol edilebilmektedir. Bunlarla birlikte ürünlerin ebatlarına, renklerine göre sınıflandırılması yapılabilmektedir. (K. Sabancı ve diğerleri, 2016)

Meyve üretimi açısından Türkiye önemli bir potansiyele sahiptir. Ülke coğrafyası göz önüne alındığında her bölgede meyvecilik yapılmaktadır. Bir araştırmaya göre ülkemizde kiraz, yöresine ve cinsine göre nisan ayından ağustos ayına kadar yetiştirilebilen bir meyvedir. (M.B.Eminoğlu ve diğerleri, 2015)

Kiraz meyvesinin toplanması çok dikkat edilmesi gereken bir konudur, zira kiraz toplanmadığında rengi, olgunluk seviyesi değişebilir ve ekonomik değeri kaybolabilir. Bu sorunun çözümü için HSV tekniğini kullanan bir sistem geliştirilmiştir. (K. Berk ve diğerleri, 2017)

HSV renklere göre sınıflandırmalar için yapılan kullanılmaktadır. HSV renk kodları RGB renk kodlarının koordinat sisteminde sunulmasını sağlayan bir sistemdir. HSV (Hue, Saturation, Value) renk modeli renkleri, renk özü, rengin yoğunluğu ve parlaklık çerçevesinde tanımlayarak değerlendirir. RGB ise kırmızı, yeşil ve mavi renkleri barındırır.

(H. Hassannejad ve diğerleri, 2016)

Kiraz tespiti araştırmasında HSV renk kodları kullanılarak yapılan uygulama uygun ışıktaki %100 doğru tespitle bulunarak kirazların tespitini yapmıştır. Algoritmada istenilen kırmızılıktaki kirazlar tespit edildiğinde arka plan siyaha çevrilmiş ve kirazlar beyaz renge boyanmıştır. Bu sayede kirazın varlığının tespiti beyaz noktalar vasıtası ile anlaşılmıştır.

(Şekil-2.4), (E. Sert ve diğerleri,2010)



Şekil 2.4 Kiraz tespiti (E. Sert ve diğerleri,2010)

Diğer bir araştırmada MATLAB yazılımı yardımı ile RGB renk kodları kullanılarak elma ve şeftali sınıflandırılması için bir çalışma yapılmıştır.

Elma sınıflandırılması için aşağıdaki işlem adımları takip edilmiştir;

- Başlangıçta elma resmi gri seviyeye çevir
- Ardından kenarları keskinleştir
- Elma alanını beyaz, dışını siyah olarak doldur (R3)
- Fotoğrafın ölçüsü çıkar
- Fotoğrafın 1. satırından en son satıra kadar tara ve a ölçüsünü bul
- Fotoğrafın 1. sütunundan en son sütuna kadar tara ve b ölçüsü bul
- Fotoğrafı 45 derece döndür, 1. satırdan en son satıra kadar tara ve c ölçüsünü bul
- Elma parametrelerinden sınıflandırma değerlerini bul
- Grafik ekranda sınıflandırma sonucunu göster, (E.Sert ve diğerleri,2010)

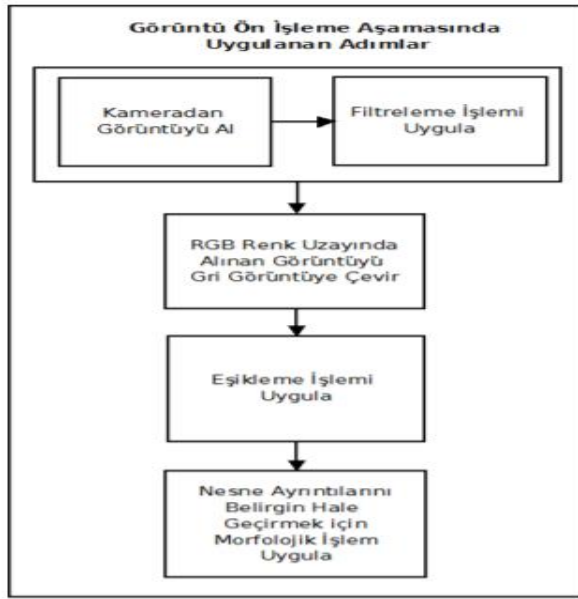


Şekil 2.5 Elmaların boyutlarına göre çıkan sonuçlar

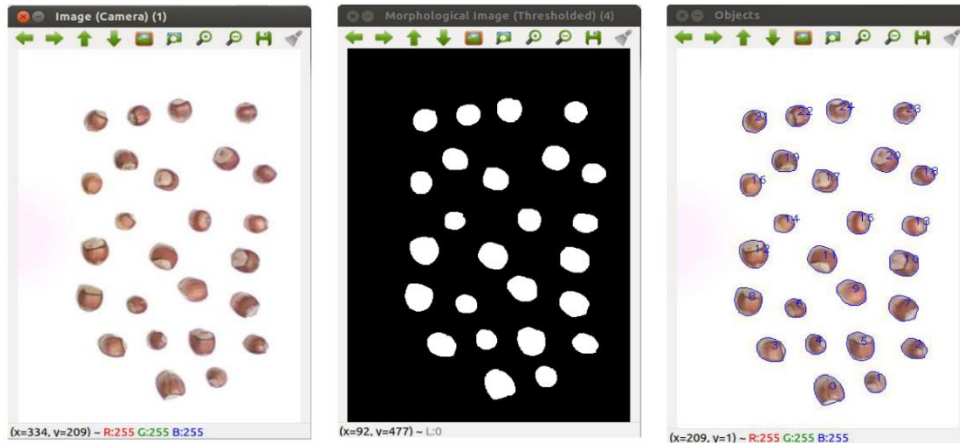
(E. Sert ve diğerleri,2010)

Fındık meyvesinin tespit ve sınıflandırılması için yapılan araştırmada ise üç aşamalı bir yöntem kullanılmıştır. İlk aşama görüntü ön işleme, 2. aşama nesne bulma ve özellik çıkarımı işlemi ve son aşama sınıflandırma olarak belirlenmiştir. Görüntü ön işleme aşamasında, kameradan alınan görüntü üzerinde sırasıyla filtreleme, resmin grileştirilmesi ve ikili resme çevrilmesi işlemleri uygulanmaktadır. Bu işlemlerin gerçekleştirilmesinden sonra görüntü üzerinde yer alan ve ilgilenilen nesneler daha belirgin ve kolay işlenebilir hale getirilmektedir. Şekil 2.6'da görüntü ön işleme aşamasında uygulanan adımlar sunulmaktadır.

(S. Solak ve diğerleri, 2018)



Şekil 2.6 Görüntü ön işleme aşaması (S.Solak ve diğerleri,2018)



Şekil-2.7 K-means ile fındık tespiti (S. Solak ve diğerleri, 2018)

Bu çalışmada ortama farklı boyutlarda fındıklar bırakılmış sonrasında ortalama tabanlı ve K-means yöntemleri uygulanarak fındıklar tespit edilmeye çalışılmıştır. Her iki yöntemde de fındıkların tamamının tespit edildiği görülmüştür. (Şekil 2.7)
(S. Solak ve diğerleri, 2018)

Diğer bir araştırma ise robotik kivi hasadı araştırmasıdır, MATLAB programında gerçekleştirilmiştir. Kurulan sistemle kameradan alınan görüntü işleme tabi tutulmuş ve elde edilen sonuçlar değerlendirilmiştir. Bu çalışmada öncelikle ağacın dalındaki kivilerin kamera ile görüntüleri alınmış, görüntü işlemeye tabi tutulmuş, kivi meyvesinde kahverengi tonlar

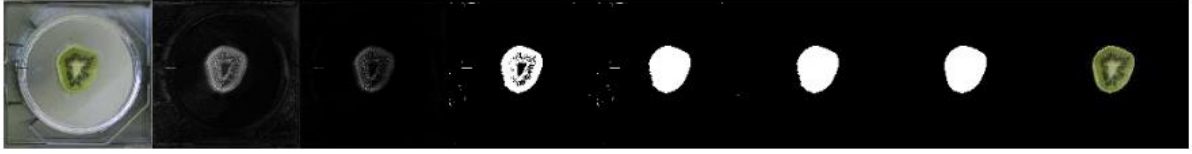
temel renk olduđu için, bu renklerin ayırt edilmesi sađlanmıř ve filtreleme sađlanmıřtır. Daha sonra elde edilen görüntü siyah beyaza dönüřtürülmüřtür. Kivileri belirten renk deđerleri programda belirtilmiřtir. Çalışma sonucunda ıřıklı ortamlarda kivilerin tespit edilebildiđi ancak ıřıđı az olduđu ortamlarda denemelerin bařarısız olduđu görülmüřtür.

(H. Polatcı ve diđerleri, 2017)

Görüntü iřleme yöntemleri ile meyve kurutma iřleminin ařamaları ve meyvenin deđiřimlerinin belirlemesi incelenmesi çalışmasında ise siyah beyaz cisim görüntüsünden faydalanılarak yüzey alanı ve Heywood Dairesellik Faktörü hesaplanmıřtır. Yüzey alanı beyaz piksellerin sayılması ile elde edilmiřtir ürünün boyutu ile ilgili dođrudan bilgi vermektedir.

Görüntü iřleme analizlerinde ortalama 900 mm²'ye yakın bir alan azalması olduđu belirlenmiřtir. Bilgisayar ortamında resim üzerinde yapılan iřlemler řekil 2.8'de gösterilmiřtir.

(H. Polatcı ve diđerleri, 2017)



řekil 2.8 Resim üzerinde yapılan iřlem (H.Polatcı ve diđerleri,2017)

Yine ölkemizde yapılan bir çalışmada güllere zarar veren gül kořnilinin görüntü iřleme teknikleri ile tespiti amaçlanmıřtır. Gül yađı, uçucu yađların önemli bir bileřenidir. Gül çiçeklerinden elde edilir. Ayrıca Isparta bölgesine büyük ekonomik katkısı bulunmaktadır. Genel olarak kozmetik ve eczacılıkta kullanılır. (H. Polatcı ve diđerleri, 2017)

Çalışmada, 13 adet farklı yerlerden alınan 1,5 cm'lik gül dalları kullanıldı. Çekilen fotođraflar MATLAB uygulamasında morphological image processing metodu ile iřlemden geçirildi. Çalışma sonucunda gül kořnilinin istila ettiđi 13 dal örneđinde yapılan arařtırmalar bařarıyla tamamlanmıřtır. Dalların renkleri ve zararlı canlının farklı yařam evreleri birbirine çok yakın bulundu ve bu da görüntülerde istenmeyen renk dalgalanmalarına ve kumlamalara neden olduđu görüldü. Bazı gül saplarının iřlem sırasında etrafındaki renkler ile girdiđi iřlemlerden sonra gerçeđinden daha büyük görüldüđü de tespit edilmiřtir. (řekil 2.9)

(A. Yılmaz ve diđerleri, 2018)



Şekil 2.9 Gül sapı örnek çıktı (A.Yılmaz ve diğerleri,2018)

Işığın yoğunluğu ve geliş açısı görüntü işleme sonucunu etkileyen en temel etkenlerden biri olmuştur. Işığın nesneler üzerine geliş açılarındaki farklılık, var olan renklerin değişik algılanması sonucunu doğurmuş ve sonuçlarda yanılmalar ortaya çıkmıştır. Bu sebepten ötürü, çalışma istenen sonuç değerlerine ulaşamamıştır. Ancak ortamda tek tip aydınlatma sağlanırsa, başarılı sonuçların elde edilebileceği; bir başka deyişle ortamın sabit ışıktaki olması gerektiği anlaşılmıştır. (M. Kabak ve diğerleri, 2017)

Bu çalışmada incelediğimiz görüntü işleme tekniklerinin ışığının yetersiz ya da farklı açılardan gelmesi durumlarında işlemlerin başarıya ulaşılması zorlaşmış hatta engellenmiş olduğu gözlemlenmiştir. (M.Kabak ve diğerleri,2017)

Görüntü işlemenin endüstride farklı alanlar farklı teknikler ile kullanıldığı görülmektedir. Her bir sektörde farklı teknikler ve sistemler kullanılmış ve farklı oranlarda doğruluk içeren sonuçlara ulaşılmıştır. Ayrıca dijital kameralarla yapılan her görüntü işleme tekniğinde, ışığın çok büyük bir öneminin olduğu görülmektedir. Görme olayı, nesneler üzerine düşen ışığın yansmasıyla başlar. Bir nesne üzerine düşen ışık demetleri, bir mercek üzerine düştüğünde, mercek arkasında tek bir noktada birleşerek görüntüyü oluştururlar. Işığın yansması, görüntü işlemede de en önemli faktördür. (M. Kabak ve diğerleri, 2017)

Wiener tekniği gürültü temizleme gibi işlemler yapmayı mümkün kılan filtreleme tekniği olarak karşımıza çıkmaktadır. Wiener filtreleme metodu genelde bozulmuş görüntünün yeniden oluşturulmasında kullanılır ve oldukça da başarılıdır, teorik olarak da en uygun çözümlerden birisidir. (M. Kabak ve diğerleri, 2017)



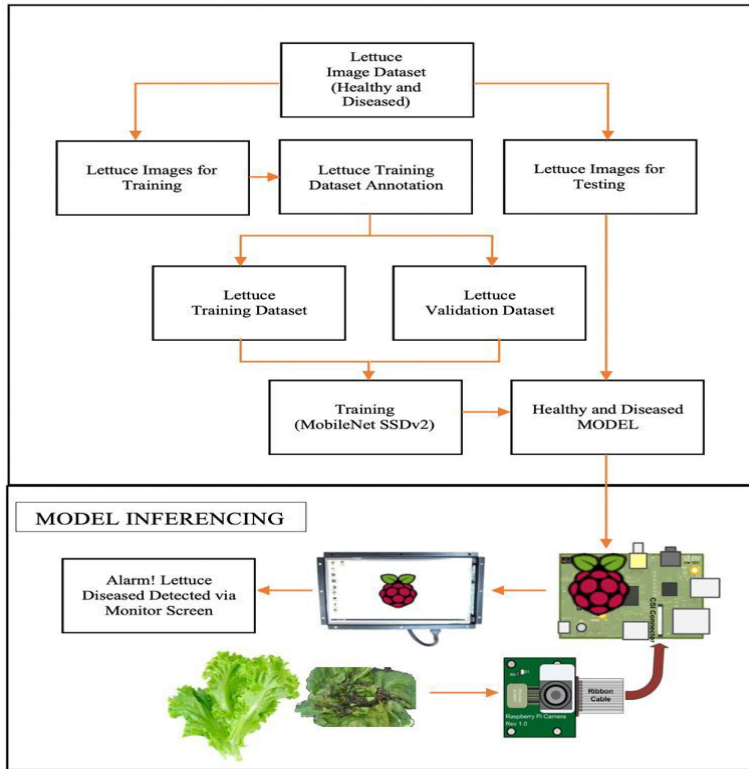
Şekil 3.1 Uzaklık ölçme metodu (M. Kabak ve diğerleri, 2017)

Şekil-3.1’de uzaklık ölçme metotları gösterilmiştir. Öklid yöntemi kolay uygulanabilirliği nedeniyle sıklıkla tercih edilmektedir. Öklid uzaklık metodunda, iki nokta arasındaki mesafe esas alınmakta, bu iki nokta dışındaki bilgiler hesaba katılmamaktadır. Manhattan yöntemi bir şehirde kare şeklinde bloklar arasında bir taksinin kat edeceği mesafe yaklaşımını esas almaktadır. Mahalanobis yöntemi, sadece merkezlere olan uzaklık değil, standart sapma değerleri de hesaplamada dikkate alınmaktadır. Bu nedenle Mahalanobis uzaklık metoduyla yapılan sınıflandırmaların daha yüksek doğrulukta sınıflandırmalara imkân verdiği gözlemlenmiştir. RGB renkleri ile Mahalanobis yöntemi kullanılarak yapılan sınıflandırma, HSV uzayında yapılan sınıflandırmaya göre daha yüksek doğruluk oranına da sahiptir. (M. Kabak ve diğerleri, 2017)

6. Görüntü İşlemeyle Marul Bitkisinin Sağlık Durumu

Derin öğrenme ve mikrodenetleyici kullanılarak marulun sağlıklı olup olmadığı incelenmiştir bu makalede. (Alon ve diğerleri, 2020)

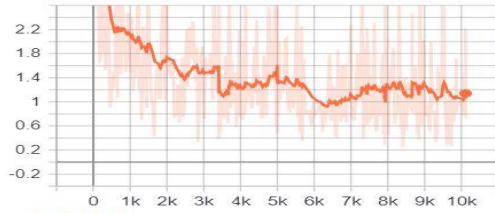
Daha sonra elde edilen marul görüntüleri eğitilmiştir. Eğitim için %80 marul görüntüsü kullanıldı. Kalan %20 ise test için kullanılmış. Eğitilen .xml dosyaları tensorflow girdilerini oluşturmuştur. Daha sonra bu dosyalar .csv dosyasına dönüştürüldü. Ayrıca görüntünün marul olup olmadığını tespit eden bir model de oluşturulmuştur. Tensorflow ile eğitilen model ve hastalık tespiti sonucu elde edilen grafik ve sonuçlar ise tablo 1.1’deki gibidir. (Alon ve diğerleri, 2020)



Şekil 3.2 Tipik bir marul tespit modeli , (Alon ve diğerleri,2020)

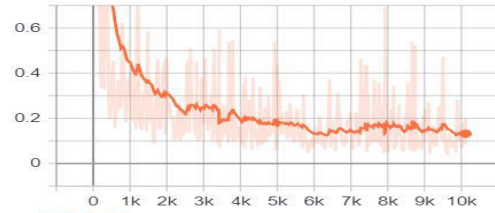
Losses

Loss/classification_loss
tag: Losses/Loss/classification_loss

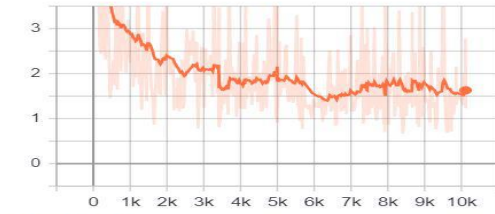


Loss/localization_loss
tag: Losses/Loss/localization_loss

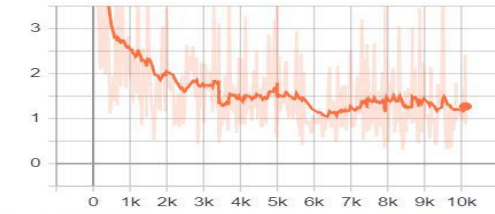
Loss/localization_loss
tag: Losses/Loss/localization_loss



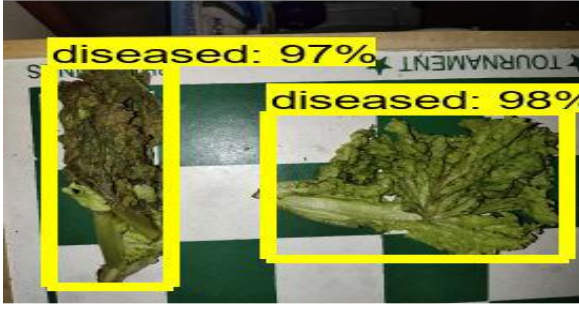
TotalLoss
tag: Losses/TotalLoss



clone_loss
tag: Losses/clone_loss



Tablo 1.1 Tensorflow ile eğitilen veri setinin çıktıları. (Alon ve diğerleri,2020)



Şekil 3.3 Modelin marul tespit aşaması (Alon ve diğerleri,2020)

7. Kayısı Hastalık Tespiti

Bu çalışmada kayısıda meydana gelen hastalıkların tespiti için bir sistem geliştirilmiştir. Kayısı meyvesindeki çil hastalığının belirtilerine bakılmıştır. Çil hastalığı kayısının; yaprak, meyve, tomurcuk ve genç dalları üzerinde görülmektedir. Hastalığın tespiti için derin öğrenme modellerinden AlexNet mimarisinin performansı değerlendirilmiştir. Bu model, ImageNet veri tabanının bir alt kümesi üzerinde eğitilmiştir. Ayrıca model, bir milyondan fazla görüntü ile eğitilmiştir. Bu görüntüler klavye, fare, kalem ve birçok hayvan gibi 1000 farklı nesne kategorisinden oluşmaktadır. Sınıflandırma algoritması olarak KNN istatistiksel öğrenme metodu kullanılmıştır. Derin öğrenme algoritması olarak AlexNet kullanılmıştır. Bu model, 3 adet maxpool katmanı, 5 adet konvolüsyon katmanı, 3 adet tam bağlı katman, softmax katmanı, 7 adet relu katmanı, normalizasyon katmanı, 2 adet dropout katmanı, 2 adet giriş ve çıkış katmanı olmak üzere toplam 25 katmandan oluşmaktadır. Çalışmada, AlexNet mimarisinin fc6 katmanından 4096 boyutlu öznitelik vektörü elde edilmiştir. Sonrasında KNN sınıflandırıcı kullanılarak başarı oranına bakılmıştır. Yapılan deneysel çalışmada veri seti Nikon 7200d kamerası ile oluşturulmuştur. Görüntülerin her biri; 3 kanallı (RGB) renkli ve 4000×6000 çözünürlüklü görsellerden oluşmaktadır. Veri setinde 308 adet sağlıklı ve 960 çil hastalığı oluşmuş kayısı yaprağı görseli içermektedir. Eğitim ve test verilerini ayırmak için 10 kat çaprazlama yöntemi kullanılmıştır. Eğitim kısmında 864 sağlıklı ve hastalıklı görüntü ve önerilen yöntemin test edilmesi için 96 görüntü kullanılmıştır. AlexNet, Vgg16 ve Vgg19 modelleri kullanılarak KNN yöntemi ile sınıflandırma performansı hesaplanmıştır. Elde edilen sonuçlara göre Vgg16 modeli kullanılarak %94.8 olarak en yüksek başarı elde edilmiştir.

(M. TÜRKOĞLU ve diğerleri, 2018)

8. Görüntü İşleme Teknikleri ile Bağ Yaprak Uyuzu Hasarının Belirlenmesi

Bağ yaprak uyuzu, havaların ısınması ile yeni oluşan yapraklardan beslenmeye başlamaktadır. Genellikle yapraklarda bulunsada bazen çiçek tomurcuklarında ve çiçeklerde de görülebilmektedir. Hasar oluşan yapraklar fotosentezi tam olarak yapamamaktadır. Bu hastalığı

getiren canlı gözle görülemeyecek kadar küçüktür. Bitkide küçük kabarcıklar halinde bir hasar meydana getirir. Bu kabarcıklara gal adı verilmektedir. Çalışmanın veri setini oluşturan görüntüler, 4000x3000 piksel çözünürlükte kamera ile çekilmiştir. Görüntü işleme için MATLAB programlama dili kullanılmıştır. Programı çalıştırmak için dört çekirdekli ve i5 işlemcili bir bilgisayar kullanılmıştır. (Altaş ve diğerleri,2021)

Alınan görüntüler bilgisayar ortamına aktarılmıştır. Mevcut görüntüler içerisinde uyuz zararını temsil eden rastgele 50 görüntü ele alınmıştır. Bu görüntüler MATLAB çalışma ortamına tanımlanmıştır. Sonrasında görüntüler $l*a*b$ renk uzayına dönüştürülmüştür. Görüntüler çeşitli faktörlere göre bölünebilmektedir. Görsellerin kümelenmesi için K-aracı kümeleme yöntemi kullanılmıştır. Çalışmada kümeleme, Öklid uzaklık fonksiyonuna göre yapılmıştır. Kümeleme işlemin yapıldıktan sonra lekeli kısımlar siyah-beyaz görüntüye çevrilmiştir. (Altaş ve diğerleri,2021)

Bu çalışmada, bağ yaprak uyuz zararlısının verdiği hasarı tespit etmek için MATLAB'ın image processing toolbox modülü kullanılmıştır. Görüntülere bakıldığında, 50 görüntü içinde en yüksek %87 ve en düşük %10 zarar oranında tespit edilmiştir. Çalışmada uzman değerlendirmesi ve görüntü işleme tekniği ile belirlenen hasar düzeyleri birbirine yakın bulunmuştur. Görüntü işleme sonuçları ile uzman değerlendirmesi arasındaki farklara bakıldığında; regresyon katsayısı $R^2 = 0.968$, $p < 0.01$; Pearson korelasyon katsayısı: $r = 0.984$, $p < 0.01$ olarak belirlenmiştir. İki yöntem arasındaki bu ilişki istatistiki olarak önemlidir, bu yüzden hasar düzeyinin belirlenmesinde uzman değerlendirmesine alternatif olarak oluşturulan görüntü işleme yönteminin kullanılması mümkündür. (Altaş ve diğerleri, 2021)

9. ŞEKER PANCARI YAPRAK LEKE HASTALIĞI ÖRNEĞİ

Şeker pancarı yaprak leke hastalığı, şeker pancarını etkileyen en önemli ve yaygın hastalıklardan biridir. Bireysel yaprak lekeleri hemen hemen daireseldir. Olgunluk seviyeleri 3-5 mm çapında ölçülmektedir. Hastalık ilerledikçe, tekli lekeler birleşmekte ve şiddetli enfekte olan doku ilk önce sarı ve daha sonra kahverengi hale gelmektedir. Sağlıklı yapraklar yeşil kalmakta ve daha az etkilenmektedir. Bu hastalık pancarın en dış yapraklarından başlayarak ve hastalık dıştan içe doğru gelişme gösterir. Bu şekilde ilerleyerek tüm yaprakları kurutmaktadır. Bu arada pancar hayatta kalabilmek için sürekli olarak göbek kısmından yeni yapraklar çıkarmaktadır. Bunun sonucunda pancar enerjisini sürekli olarak yeni yaprak oluşturmaya harcayarak yeterli irilikte pancar kök gelişimi gerçekleştirememektedir. (Altaş ve diğerleri, 2019)

Yaprak üzerindeki hastalıklı bölgelerin alanı şu şekilde oranlanır; ilgili bölgeleri oluşturan piksel sayısının toplamının yaprağı oluşturan toplam piksel sayısına oranlanarak yaprak üzerindeki kapladığı alandır. Bu da hastalığın şiddetini göstermektedir.

(Altaş ve diğerleri, 2019)

Bir şeker pancarı yaprak lekesi görüntüsü ele alındığında uzmanın değerlendirilmesi sonucu hastalık şiddeti %50 bulunan bir yaprakta, görüntü işleme sonucunda hastalık şiddeti %48 hesaplanmıştır. Gözlem (%50) ile görüntü işleme (%48) karşılaştırıldığında birbirlerine çok yakın bir tam sayı değeri bulunmuştur. Görüntü işleme yöntemi kullanmanın sağladığı hassasiyetin ve doğruluğun önemi bu yolla anlaşılmaktadır. Geliştirilen görüntü işleme algoritmaları ile hastalıklı alanlar tam olarak belirlenmiştir. Yapılan çalışma bu yöntemin gözlem ile hastalık belirleme yönteminin yerine kullanılabileceğini, hatta kesin değerler vermesinden dolayı yüksek oranda tercih nedeni olabileceğini göstermektedir. Çalışmada şeker pancarında yaprak leke hastalığı başarıyla belirlenmiştir, dolayısıyla bu yöntemin farklı bitki türlerinde görülen hastalık ve zararlı tespitinde de rahatlıkla kullanılabileceği sonucuna varılabilir.

(Altaş ve diğerleri, 2019)

10. Tarımda Görüntü İşleme ve Literatür Çalışma Sonuçları

Modern tarımda verimin artırılması amacıyla teknolojik gelişmelerin kullanımı mahsulün iyileştirilmesi için çok önemlidir. Her yıl bir çok bitki yetiştirici zararlı böcek sebebi ve çeşitli hastalıklar nedeniyle büyük kayıplar yaşıyor.

(Bartolome T. ve diğerleri, 2014)

Bitki hastalıklarını ve zararlı böcekleri tespit etmek, tarım alanında büyük bir zorluktur. Ayrıca ilaç kullanımı hem insan sağlığı açısından zararlı olabilmekte hem de maddi olarak büyük bir yük haline gelebilmektedir. Bu nedenle ilaç kullanımı en aza indirirken hastalık ve istila ile mücadele etmek için etkili yöntemler geliştirilmelidir. Görüntü analizi teknikleri tarım bilimine kapsamlı bir şekilde uygulanarak mahsuller için maksimum koruma sağlayarak sonuçta daha iyi mahsul üretimine yol açabilir. Geçmişten günümüze kadar zararlı istilalarının izlenmesi insan gücüne bağlıdır, ancak günümüzde insan emeğini ve hatalarını en aza indirmek için otomatik izleme sistemleri geliştirilmektedir. Bu çalışma, pirinç tarlalarındaki zararlı yoğunluklarını tahmin etmek için otomatik bir tespit sistemi kurarak zararlıları tespit etmek ve ayıklamak için farklı görüntü işleme tekniklerinin uygulanmasını konu almaktadır. Deney

sonuçları, önerilen sistemin pirinç tarlalarındaki zararlıları tespit etmede verimli bir çözüm yolu sağladığını göstermektedir.

(Bartolome T. ve diğerleri, 2014)

10.1 Görüntü işleme

Görüntü işleme, fotoğraf ve video gibi kaynaklardan alınan görüntülerin analizi ve işlenmesidir. Görüntü işlemede üç ana adım vardır; ilki, yakalanan görüntülerin bir bilgisayarın işleyebileceği ikili değerlere dönüştürülmesidir; ikincisi, görüntü geliştirme ve veri sıkıştırmadır; ve üçüncüsü işlenmiş görüntünün görüntülenmesi veya yazdırılmasından oluşan çıktı adımıdır. Görüntü işleme, uydu hava haritalandırılması, makine ile görme, bilgisayar tabanlı örüntü tanıma ve tarımsal ürünlerin verimliliğini artırma gibi uygulamalarda kullanılır. (Bartolome T. ve diğerleri, 2014)

10.2 İlgili Literatür ve Çalışmaların İncelenmesi

Görüntü analizine dayalı otomatik böcek tanımlama sistemleri oluşturmak için birkaç girişimde bulunulmuştur. Bir çalışmada Hindistan'ın Kuzey Bengal bölgelerindeki çay bahçelerinde sekiz ana böcek üzerinde yoğunlaşmıştır. Bu çalışmada korelasyon tabanlı özellik seçimi ve artımlı geri yayımlı sinir ağı uygulanmıştır. (Samanta ve diğerleri, 2012)

Başka bir çalışmada araknidleri ve diğer eklembacaklıları tanıma konusunda uzman olmayanlar için tanımlamayı kolaylaştırmak için bilgisayar tabanlı bir örüntü tanıma sistemi kurmuşlardır. Araştırmacılar, ilaç kullanımını azaltmak için sera bitkilerinde erken zararlı tespitini araştırmışlardır. Kurulan sistemin sistemin amacı yapraklardaki zararlıların tespitidir. (Do. ve diğerleri, 1999)

Başka bir çalışmada ceviz bitleri için sinir ağı tabanlı bir tanımlama sistemi geliştirilmiştir. Görüntü tanımlamak için kamera sistemi kurulup farklı görüntüler elde edilmiştir. Daha sonra elde edilen görüntüler ikili dosyaya dönüştürülmüş ve 114x134 piksele yeniden boyutlandırılmıştır. Verileri işlemek için farklı görüntü işleme teknikleri kullanmışlardır. (Al-Saqer ve diğerleri, 2012)

Bir makalede böcekleri tanımlamak için yeni bir yöntem önerildi. Görüntü eşleştirme için entegre bölge eşleştirme ve ikili ağaç karmaşık dalgacık dönüşümünü kullandılar.

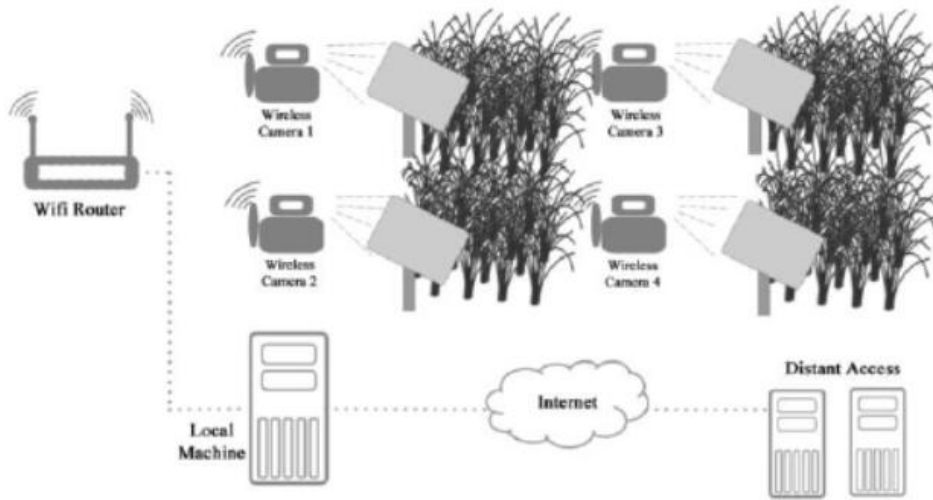
Böceklerin toplanan görüntüleri 248x200 piksele indirgenir ve ortalama kaydırma algoritması kullanılarak filtrelenir. İşlenen görüntü, renk özelliği kullanılarak k-mean algoritmasına dayalı olarak bölgelere ayrılır. Pikselleri renk özelliklerine göre bölgelere ayırmak için piksel bazında bir segmentasyon kullandılar.

(L. Zhu ve diğerleri, 2011)

10.3 Malzemeler ve Yöntemler

A. Görüntü elde etme

Araştırmacılar zararlıları yakalamak için yapışkan tuzaklarla birlikte kablosuz kameralardan (kameralar su ve doğrudan güneş ışığına karşı korumalıdır) oluşan bir ağ kurmuştur. Kullanılan kameralar 8 megapiksel çözünürlükte saniyede 10 kare yakalayabilen CISCO Linksys Wireless-G Internet Home Monitoring kameradır. Yakalanan görüntüler Intel i3 işlemci ve 4 GB RAM donanıma sahip bir bilgisayarda işlenmiştir. Sistemin mimari tasarımı Şekil 1’de gösterilmiştir.



Şekil 3.4 Mimari Tasarım , (Bartolome T. ve diğerleri,2014)

B. Görüntü Ön İşleme

Görüntü ön işleme, durağan görüntünün işlenerek daha kullanışlı olan gelişmiş bir görüntü oluşturulmasını sağlar. RGB renk modelinde, her renk kırmızı, yeşil ve mavinin bileşimi olarak görülür. RGB renkli görüntünün saklanması için geniş bir alan gerekir ve işlenmesi çok zaman alır. Bu çalışmada gri tonlamalı görüntü yeterlidir, bu nedenle araştırmacılar aşağıdaki formülle RGB görüntüyü gri tonlamalı görüntüye dönüştürürler:

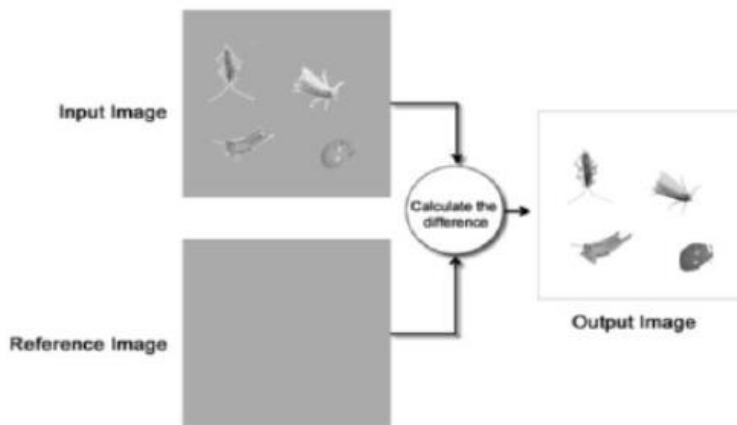
$$I(x,y) = 0.2989xR + 0.5870xG + 0.1140xB \quad (1.2)$$

(Bartolome T. ve diğerleri, 2014)

C. Görüntüdeki Zararlıların Tespiti

Araştırmacılar, kameradan yakalanan görüntülerin piksel değerlerini karşılaştırdı. Farkın tespiti için iki görüntü kullanılır; ilk görüntü, karşılaştırma amacıyla referans piksel değerlerini temsil eden referans görüntü olarak kullanılırken, ikinci görüntü giriş görüntüsü olarak görev yapar. İki görüntü karşılaştırılmış ve piksel değerlerindeki farklılıklar belirlenmiştir. Girilen görüntü piksel değerleri, referans görüntü piksel değerlerine eşit değilse girilen görüntü pikseli, çıktı görüntü pikseli olarak kaydedilmiştir. (Şekil-3.5)

(Bartolome T. ve diğerleri, 2014)



Şekil 3.5 Output (Bartolome T. ve diğerleri,2014)

D. Görüntünün Filtrelenmesi

Filtreleme, görüntünün görünümünü farklı aydınlatma koşullarının neden olduğu gürültüden temizleme işlemidir. Dijital görüntü işlemede kullanılabilir bir sonuç elde etmek için filtreleme gerekmektedir. Bu çalışmada araştırmacılar medyan filtresi kullanmışlardır. (Bartolome T. ve diğerleri,2014)

119	114	113	111	110	111	112
111	111	112	115	112	116	119
114	132	122	124	121	112	122
112	111	121	126	121	115	121
113	112	123	121	122	115	111
122	112	112	122	112	113	112
121	111	112	134	111	112	115

Neighborhood value = 122

121, 121,121,121,122,
122,123,124,126

Median = 122

Şekil 3.6 Medyan Filtresi (Bartolome T. ve diğerleri,2014)

E. Tespit Edilen Zararlıların Çıkarılması

Bu aşama, tespit edilen zararlının görüntüden çıkarılmasıdır. Bir önceki aşamanın çıktı görüntüsünün görüntü piksel değerleri, zararlının koordinatlarını tespit etmek için hem yatay hem de dikey olarak taranır. Zararlının başlangıç ve bitiş noktaları belirlendikten sonra matris kaydedilmiştir.

(Bartolome T. ve diğerleri,2014)

10.4 Deney Sonuçları

Önerilen sistem Pampanga Eyalet Tarım Üniversitesi uygulama çiftliğinde test edildi. Test edilen alan 50x50 metre genişliğindedir. Örnekler, test alanına kurulan 4 adet kamera tarafından yakalanan yapışkan tuzaklar üzerinden toplanmıştır. Önerilen sistem art arda beş gün boyunca test edilmiştir ve sonuçlar şekil-3.7'deki tablodaki gibidir. (Bartolome T. ve diğerleri,2014)

	Camera 1	Camera 2	Camera 3	Camera 4	Total
Day 1	15	12	13	10	50
Day 2	12	10	12	10	44
Day 3	10	13	12	11	46
Day 4	14	15	10	13	52
Day 5	12	11	10	5	38

Şekil 3.7 Deney Sonuçları (Bartolome T. ve diğerleri,2014)

10.5 Sonuç

Bu çalışmada, otomatik tespit ve ayıklama sistemi kurulmuş, yakalanan görüntüdeki zararlıları tespit etmek için farklı görüntü işleme yöntemleri kullanılmıştır. Araştırmacılar yakalanan görüntüdeki zararlıların tespiti için arka plan modellemesi kullandılar ve farklı aydınlatma koşullarından dolayı ortaya çıkan gürültüyü gidermek için medyan filtre kullandılar. Algılanan nesneyi görüntüden çıkartmak için görüntü hem yatay hem de dikey olarak taranarak koordinatlar belirlenir ve nesne görüntüsü kaydedilir.Çalışmadan çıkan sonuçlar umut vericidir, ancak gerekliliklere ulaşmak için hem materyallerde hem de yöntemlerde çeşitli iyileştirmeler yapılmalıdır. (Bartolome T. ve diğerleri,2014)

11.1 Genel Bulgular

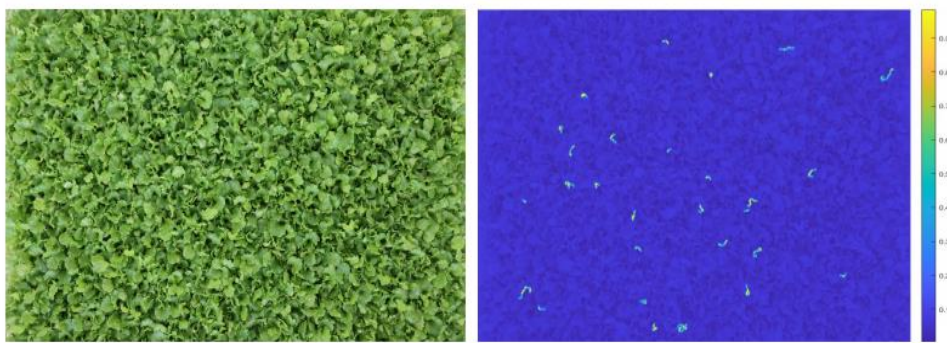
Bu çalışmada, daha kaliteli ve düşük maliyetli tarım ürünü yetiştirme amacıyla akıllı sera kullanımı araştırılmıştır. Araştırmacılar, çeşitli arduino sensörler yardımı ile alınan verileri kullanarak otonom veya geliştirilen mobil uygulama üzerinden manuel olarak seraya müdahale edilecek bir sistem geliştirmişlerdir. (Öztürk ve diğerleri, 2021)

Bu makalede, tarla koşullarında depolanmış tahıl böceklerini tespit etmek için Faster R-CNN modeline dayanan nesne algoritması kullanılmıştır. Derin evrişimli sinir ağları aracılığıyla küçük böceklerin tespitinin doğruluğunu arttırmak için başlangıç ağı geliştirilmiştir. (Z. Shi ve diğerleri, 2020)

Bu makalede, gıda endüstrisi ve tarım alanında görüntü işleme teknolojisinin kullanımını ele alınmaktadır. Tarım ürünlerinde kalite; boyut, renk, şekil, doku ve kusurlara bakılarak ölçülmektedir. Bu çalışmada gıdaların manuel olarak denetlenmesini sağlayan bilgisayar görüş sistemi kullanılmaktadır. Bilgisayarla görme tabanlı kalite denetimi dört ana adımdan oluşmaktadır bunlar; edinme, segmentasyon, öznelik çıkarımı ve sınıflandırmadır. Bu çalışmada araştırmacılar her adımda çeşitli yöntem ve algoritmalar kullanmış ve karşılaştırmıştır. Çalışmada meyve ve sebzelerin tek bir açıdan fotoğrafları esas alınmıştır, farklı yönlerden çekilen görüntüler ile birlikte sistemin performansının artırılması amaçlanmaktadır. (Bhargava ve diğerleri, 2018)

Bu çalışmada araştırmacılar, görüntü işleme ve makine öğrenimine dayalı olarak anormal hidroponik marul yapraklarının tespiti üzerine çalışmışlardır. Sarı ve çürümüş yapraklar maruldaki ana anormal yaprak türleridir. Bu çalışma, makine öğrenim modelleri ile marulun sarı ve çürümüş yapraklarını tespit etmeyi amaçlamaktadır. Marul görüntülerinin RGB özellik sayısını azaltmak için tek yönlü varyans analizi uygulanmıştır. (Ruizhe Yang ve diğerleri, 2021)

Bu çalışmada, marul bitkisinde uç yanık tespiti üzerine çalışılmıştır. Tek bir yaprağı veya bitkiyi baz alan diğer araştırmaların aksine dikey bir açıdan aynı anda birden fazla bitki üzerinde yanık tespiti amaçlanmıştır. (R. Gozzovelli ve diğerleri, 2021)



Şekil 3.9 Marul bitkisinde uç yanıkların ısı haritası , (R. Gozzovelli ve diğerleri,2021)

PROJEDE KULLANILAN TEKNOLOJİLER VE PROJE METODU

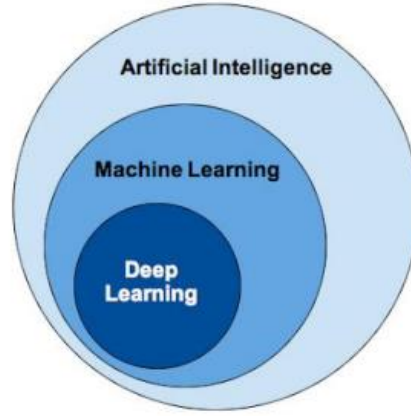
1.1 Makine Öğrenmesi

Makine öğrenmesi bir dataset üzerinde tahmin ve karar elde etmek için oluşturulan matematiksel süreci ifade eder. Makine öğrenmesi, yapay zekanın bir alt dalıdır ve öğrenmeye dayalı olarak geliştirilir. Makine öğrenmesi algoritmaları ile tahmin için kullanılacak veriler eğitilir ve sınıflandırılır. Öğrenme yeteneklerine bağlı olarak makine öğrenmesi algoritmaları üç şekilde sınıflandırılır; denetimli, denetimsiz ve takviyeli öğrenme. Sınıflandırma ve regresyon denetimli öğrenme kategorisinde incelenirken gerçek zamanlı karar modeli pekiştirmeli öğrenme kategorisinde incelenir. Denetimli öğrenme, girdi verilerini kullanarak tasarlanan model üzerinden tahminlerde bulunur. Denetimsiz öğrenme çok daha karmaşık işlemler için kullanılır. Makine öğrenmesi teknikleri genel olarak verilerden yararlanarak insan davranışlarını analiz etmek ve işletmelerin buna uygun üretim yapmalarını sağlamak için kullanılmaktadır. Bunun dışında günümüzde çok farklı amaçlar için makine öğrenimi kullanılabilmektedir. (Bal F. ve diğerleri, 2021)

1.2 Derin Öğrenme

Derin öğrenme ilk defa Igor Aizenberg tarafından 2000'lerin başında dile getirilmiş, asıl popülerliğini ise 2016 yılından itibaren kazanmaya başlamıştır. Derin öğrenme, makine öğrenme algoritmalarını ve yapay sinir ağlarını kapsayan çok daha derin ve kompleks bir yapıdır. Derin öğrenme, yapılandırılmamış verilerin işlenmesini maksimum seviyede yapması, yüksek kaliteli sonuçlar üretmesi ve gereksiz maliyetlerden kaçınması gibi avantajlara sahiptir. Diğer taraftan daha fazla veriye ihtiyaç duyma ve donanım açısından yüksek maliyete neden olma gibi dezavantajları mevcuttur. Doğal dil işleme, otonom araç teknolojisi, yüz tanıma gibi alanlarda kullanılmaktadır.

(Bal F. ve diğerleri,2021)



Şekil 4.1 DL ve ML arasındaki ilişki , (Bal F. ve diğerleri,2021)

1.3 Tensorflow

TensorFlow, 2015 yılında Google tarafından geliştirilen ve günümüzde bir çok kişi tarafından kullanılan açık kaynak bir makine öğrenmesi yazılımı kütüphanesidir. Farklı alanlarda da kullanılabilmesine rağmen ana amacı derin sinir ağlarının eğitimi ve modellenmesi üzerinedir. Esnek mimarisi sayesinde mobil ve uç cihazlara kadar kolay dağıtıma olanak sağlamaktadır.

Çalışma biçimi olarak, Tensor adı verilen skaler, vektörel veya n-boyutlu diziyi girdi olarak alarak bir çıktı değeri döner. Tensorflow’da tüm işlemler Graph isimli bir hesaplama kümesinde gerçekleştirilir. (Ayşe A. ve diğerleri, 2020)

1.4 Python

Python, Guido Van Rossum tarafından geliştirilen nesne yönelimli bir programlama dilidir. İçerdiği bir çok açık kaynaklı kütüphanesi sayesinde Web tabanlı uygulamalar, masaüstü uygulamaları, bilimsel ve matematiksel hesaplamalar gibi bir çok alanda uygulama geliştirmek mümkündür. Tıp, astronomi, makine öğrenmesi gibi bir çok alanda çeşitli kütüphaneler içermektedir. Bu çalışmada geliştirilen uygulama Python programlama dili kullanılarak Jupyter Notebook geliştirme ortamında yazılmıştır. Geliştirilen uygulama Windows işletim sisteminde test edilmiş ve buna göre sonuçlar elde edilmiştir. (Mahmut S. ve diğerleri, 2021)

1.5 Anaconda

Anaconda, ücretsiz ve 1500'den fazla açık kaynak paketi içeren bir paket yöneticisi sistemidir. Anaconda dağıtımı Windows, Linux ve MacOS gibi işletim sistemlerinde kullanılabilen paketleri içerir. İçerisinde barındırdığı Jupyter, JupyterLab, Spyder ve RStudio gibi çeşitli IDE'leri kullanarak veri bilimi projeleri oluşturulmasına olanak sağlar. Geliştirilen proje dosyalarını başka geliştiriciler ile de paylaşmamıza ve farklı platformlarda da projeler geliştirmemizi sağlamaktadır. (Mahmut S. ve diğerleri, 2021)

1.6 Keras

Keras, Python programlama dili kullanılarak yazılmış bir açık kaynaklı sinir ağı kütüphanesidir. Keras kütüphanesi kullanılarak yapay sinir ağları oluşturmak ve çalıştırmak mümkündür. Keras, kullanıcı dostu, modüler ve genişletilebilir bir yapıya sahiptir. Keras kullanılarak geliştirilen modeller görselleştirilebilir ve kaydedilebilir. (Mahmut S. ve diğerleri, 2021)

1.7 Görüntü işleme

Görüntü işleme, fotoğraf ve video gibi kaynaklardan alınan görüntülerin analizi ve işlenmesidir. Görüntü işlemede üç ana adım vardır; ilki, yakalanan görüntülerin bir bilgisayarın işleyebileceği ikili değerlere dönüştürülmesidir; ikincisi, görüntü geliştirme ve veri sıkıştırma; ve üçüncüsü işlenmiş görüntünün görüntülenmesi veya yazdırılmasından oluşan çıktı adımıdır. Görüntü işleme, uydu hava haritalandırılması, makine ile görme, bilgisayar tabanlı görüntü tanıma ve tarımsal ürünlerin verimliliğini artırma gibi uygulamalarda kullanılır. (Bartolome T. ve diğerleri, 2014)

1.8 Veri Seti

Bu çalışmada yetiştirilen marullardan elde edilen görüntüler veri seti oluşturmada kullanılmıştır. Bu veri setinde 231 sağlıklı, 241 hastalıklı ve 45 böcekli görüntü kullanılmıştır (Çizelge-1.2). Görüntüleri sınıflandırırken bu üç duruma dikkat edilmiştir. Veri setinde bulunan görüntüler 256x256 piksel boyutunda, ve jpg formatındadır. Şekil 4.2'de sağlıklı, hastalıklı ve üzerinde böcek olan marul fotoğrafları verilmiştir.

Sınıf İsmi	Veri Seti
Healthy	231
Diseased	241
Pest Detected	45
Toplam	517

Tablo 1.2 Veri setinin sınıflara göre resim sayıları



Şekil 4.2 Sırasıyla sağlıklı, hastalıklı ve üzerinde böcek olan marul fotoğrafları

1.9 Çalışmada kullanılan fiziksel donanım

Makine öğrenmesi uygulamalarında kullanılan algoritmalar çok fazla hesaplama yapmaktadırlar ve bu çok fazla kaynak tüketimine neden olmaktadır. Bu çalışma MSI GE76 RAIDER model bir bilgisayar üzerinde gerçekleştirilmiştir. Bu bilgisayara ait donanım bilgileri Çizelge 3.1’de gösterilmiştir.

İşlemci	12th Gen Intel(R) Core(TM) i7-12700H
Bellek	16 GB DDR5 RAM
Disk	WDC PC SN540 SDDPNPF-1T00-1032
GPU	NVIDIA GeForce RTX 3060 Laptop GPU
Ağ Bağlantısı	Intel(R) Wi-Fi 6 AX201 160MHz

Tablo 1.3 Donanımsal bilgiler

2.1 Ön İşlem

Geliştirilen sistemde kullanılacak marul veri setindeki her bir resmin boyutunun 768x1024 piksel olmasından dolayı algoritmanın çok fazla kaynağa ihtiyaç duyacağı öngörülmüştür. Bu yüzden 256x256 piksel boyutundaki resimlerin küçültülmesine ihtiyaç doğmuştur. Resimler ön işlemde geçirilerek boyutları 256x256 olarak yeniden boyutlandırılmıştır. Tüm resimler aynı yükseklik ve genişlik değerine sahiptir. Resimlerin yeniden boyutlandırılmasında '<https://www.resizetnow.com/tr>' isimli site kullanılarak boyutlandırılmıştır. (Kumar, 2022).

2.2 Image Augmentation (Veri Çoğaltma)

Veri çoğaltma, bir veri kümesinin bir veya birden fazla parametrelerinin değiştirilmiş versiyonlarını oluşturarak veri kümesinin boyutunu arttırmak için kullanılan bir tekniktir. Tahmin değerlerinde gerçeğe yakında sonuçlar üretmek için çekilen resimlere ek olarak çok sayıda eğitim verisi gerekir. Bu yüzden daha iyi sonuçlar aldığımız bir model oluşturmak için genellikle veri çoğaltma tekniğinden yararlanılır (Kumar, 2022).

Yaygın olarak kullanılan veri çoğaltma tekniklerinden bazıları aşağıda verilmiştir.

a) Görüntü Ölçeklendirme

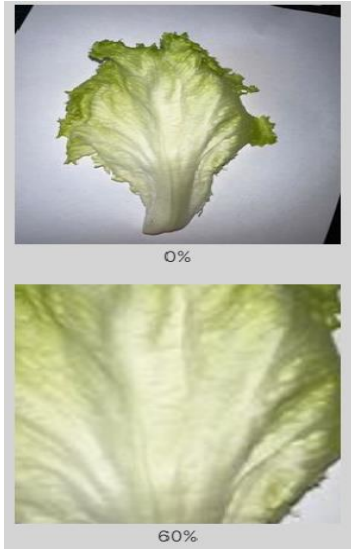
Resmi yeniden boyutlandırma işlemidir, görüntü verilen boyuta göre yeniden boyutlandırılır. (Kumar, 2022).



Şekil 4.3 Görüntü ölçeklendirme

b) Görüntü Kırpma

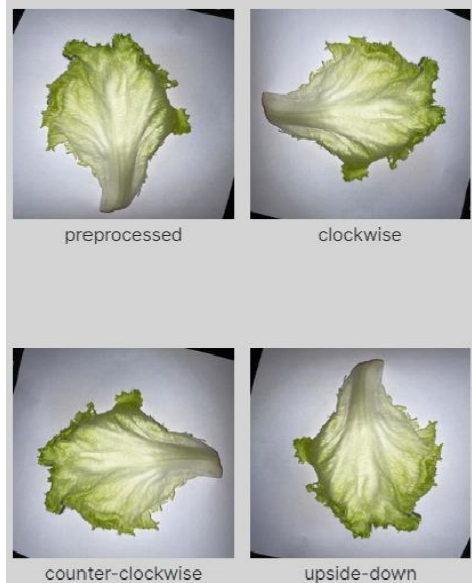
Görüntünün sadece belirli bir kısmı kırpılarak işleme tutulur. (Kumar, 2022).



Şekil 4.4 Görüntü kırpma

b) Görüntü Çevirme

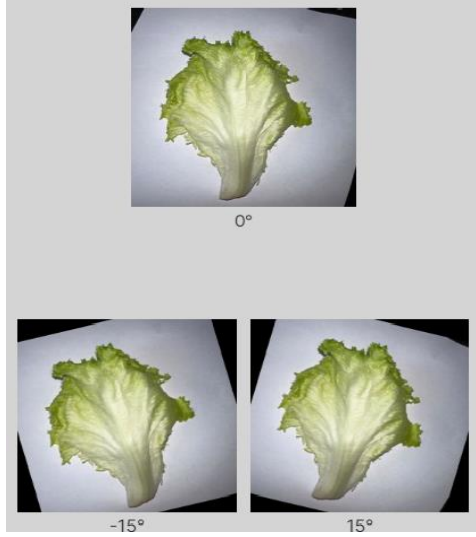
Görüntü çevirme, dikey ve yatay olarak piksel satır ve sütunlarını tersine çevirme işlemidir. (Kumar, 2022).



Şekil 4.5 Görüntü çevirme

d) Rastgele çevirme

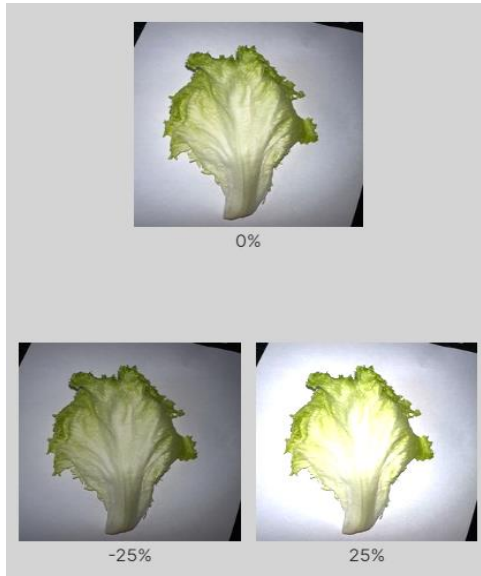
Görüntüyü sadece dikey ve yatay olarak değil 0'dan 360 dereceye kadar belirli bir derecede saat yönünde döndürme işlemidir. (Kumar, 2022).



Şekil 4.6 Rastgele çevirme

d) Parlaklık

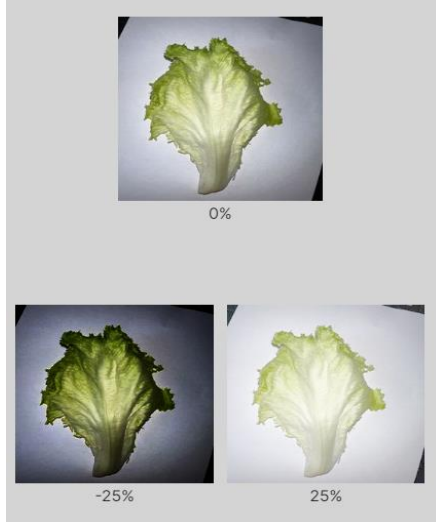
Görsellerin parlaklık değerlerini değiştirerek önceki haline kıyasla daha parlak veya daha koyu görseller elde edilmesi işlemidir. Bu işlemde amaç, farklı parlaklık seviyelerinde modeli eğitilmesini sağlamaktır. (Kumar, 2022).



Şekil 4.7 Parlaklık

e) Kontrast

Görüntüdeki kontrast değiştirilerek karanlık ve parlak alanlar arasında bir ayırım sağlanır. (Kumar, 2022).



Şekil 4.8 Kontrast

f) Görüntü yakınlaştırma ve büyütme

Yakınlaştırma, görüntüyü yakınlaştırır veya görüntünün etrafına yeni piksel değerleri ekleyerek görüntüyü uzaklaştırır. 1.0'dan küçük değerler görüntüyü yakınlaştırırken, 1.0'dan büyük değerler görüntüyü uzaklaştırır. (Kumar, 2022).



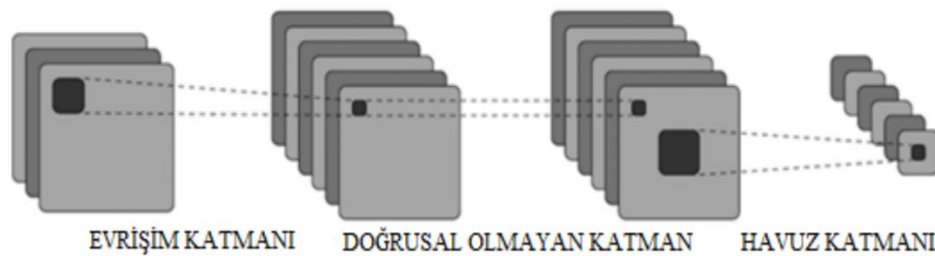
Şekil 4.9 Görüntü yakınlaştırma

2.3 Data Cleaning (Veri Temizleme)

Veri kalitesi, geliştirilen projelerde genellikle ana bir konudur. Güvenilirliğin artması için çok sayıda görüntü ön işleme tekniğinden yararlanır. Bunlardan birisi olan veri temizleme görüntüdeki gürültülerin ve tutarsızlıkların giderilmesi işlemidir. Herhangi bir değişkenin eksik değerlerinin doldurulması için farklı yollar vardır; eksik değerler içeren görüntüler atılabilir, değişken ortalaması eksik değerler için kullanılabilir, aynı sınıfa ait tüm görseller için değişken ortalaması kullanılabilir, varolan değerlere bağlı olarak en uygun değer seçilip kullanılabilir. Veri temizleme uygulanırken bir diğer sorun ise gürültü temizlemedir. Gürültü, resimde meydana gelen bir hatadır. Gürültüyü gidermek için binning, kümeleme (clustering) ve regresyon (regression) gibi veri düzeltme tekniklerinden yararlanır. (Ayşe O. ve diğerleri, 2003)

2.4 CNN (Evrişimli Sinir Ağları)

CNN, derin öğrenmenin çeşitli algoritmalarından önemli bir tanesidir. CNN, insan beyninden ilham alınarak ortaya çıkmıştır ve tıpkı insan beynindeki sinir ağları gibi görüntüyü sınıflandırmak, benzer görüntüleri kümelemek ve nesne tanımlamak için kullanılır. CNN, görüntüyü çeşitli katmanlarda işler, bu katmanlar sırasıyla evrişim katmanı, doğrusal olmayan katmanlar, havuz katmanları ve tamamen bağlı katmanlardır. Bu katmanlar şekil-5.1’de gösterilmiştir. En yoğun matematiksel işlemlerin gerçekleştiği katman evrişim katmanıdır ve görüntüdeki düşük ve yüksek seviyeli özellikleri bulmak için kullanılmaktadır. Doğrusal olmayan katmanda doğrusal olmayanlık tespit edilir. Havuzlama katmanı parametrelerin azaltılması ve sistem üzerindeki ağırlık sayısının azaltmak amacı ile kullanılmaktadır. (Bekir A. ve diğerleri, 2020)



Şekil 5.1 evrişimsel sinir ağı katmanları

Proje Aşamaları

Bitki Yetiştirme Ve Veri Seti Hazırlama Aşamaları :

Amacımız projemizde kullanacağımız bitkinin hastalık ve sağlık durumlarını gözlemlemek ve ek olarak bitkiye zarar verebilecek bazı böceklerin tespitini yaparak bitkinin uygun şartlarda yetişmesini sağlamak için bazı etkenleri kontrol etmek ve bunu bir uyarı sistemi haline getirmektir. Veri bilimi, deep learning ve görüntü işleme projeleri için ilk adım çalışılacak uygun bir dataset hazırlamaktır.

Projemize ilk olarak kendi datasetlerimizi oluşturmak için bitki yetiştirmeyle başladık. Marul, fesleğen, lahana, maydanoz gibi bitkiler yetiştirmeye başladık. Bitkilerin yetişmesi ve tohumların çimlenmesi için uygun sıcaklık, nem ve ortam sağladık. Yetiştirme aşamasında bazı sorunlarla karşılaştık da en uygun verinin marul bitkisinden oluşacağını tespit ettik. Projemizde ilk aşamamız çalışabileceğimiz datasetimizi toplamaktı. Bitkimizi yetiştirme aşamasında düzenli olarak fotoğrafladık. Aşağıdaki resimlerde yetiştirme sürecindeki bazı aşamalar yer alıyor.



Şekil 5.2 Maydanoz ve lahana bitkisi yetiştirme aşamaları



Şekil 5.3 Fesleğen bitkisi yetiştirme aşamaları



Şekil 5.4 Marul bitkisi yetiştirme aşamaları

Projemizin sistemine en uygun bitki olduđu için projemizde marul bitkisini dataset olarak kullanmaya karar verdik.Ve yetiřme sürecinde sađlıklı yaprak fotođrafları çektik.Bitki yapraklarının hastalıklı durumlarını gözlemlemek için bitki ve yapraklarını yetiřmesi için uygun olmayan řartlara maruz bıraktık.Son olarak da bitkiye zarar verebilecek böcekleri sistemin içersine dahil ettik.Böceklerden bazıları bitkinin ortamına dođal olarak dahil olsada veri setimizi zenginleřtirmek için biz de böcek ekledik.

Etiketler

Datasetimizde kullanmak üzere 3 tane etiketin olması gerektiđine karar verdik.

Bu etiketlerimiz:

- 1-Sađlıklı (Healthy)
- 2-Hastalıklı (Disease)
- 3-Böcek tespit edildi. (Pest detected)

Ařađıda bu etiketlere uygun bazı resimlere yer verilmiřtir :



řekil 5.5 Sađlıklı marul bitkisi örnek



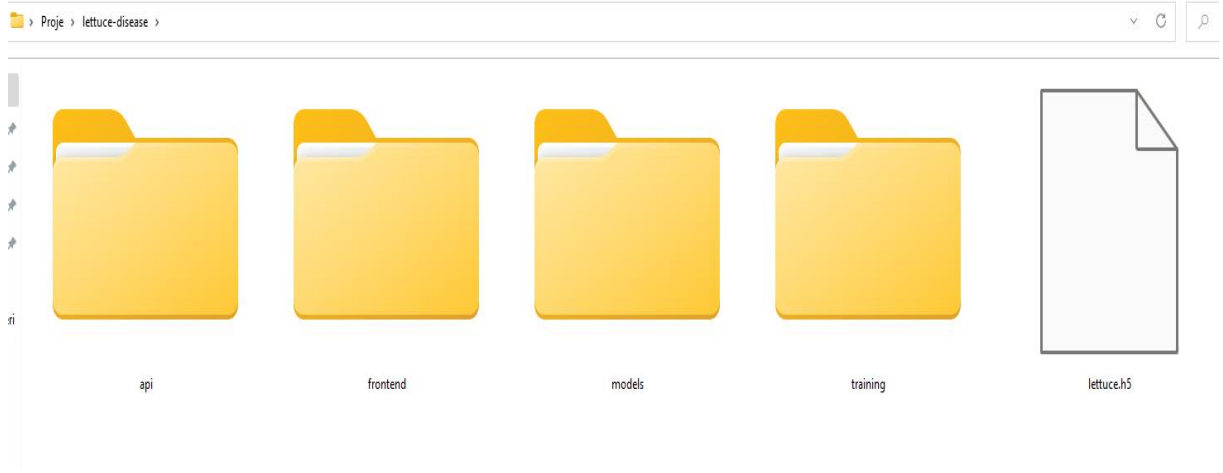
řekil 5.6 Hastalıklı marul bitkisi örnek



řekil 5.7 Pest detected etiketine örnek

Proje Klasörü

Verilerimizi topladıktan sonra projemizi oluşturmaya başladık. Ve proje klasörümüzü oluşturduk ve datasetimizi ve modeli eğiteceğimiz dosyayı training klasörüne yerleştirdik. Training klasöründe modelimizi eğittik. Projemizin arayüzünü oluşturmak için backend ve frontend klasörleri oluşturduk. Eğittiğimiz model models klasörüne kaydedildi.



Şekil 5.8 Projemizin genel klasör yapısı

Model Eğitim Aşaması :

Öncelikle gerekli kütüphanelerimizi ekledik. Modelimizi jupyter notebook kullanarak eğittik. Resimlerimizin boyutunu 256*256 olarak ayarladık. Fotoğraflarımızı boyutlandırmada 'https://www.resizenow.com/tr' sitesinden yararlandık. Modelimizin batch size değeri ise 32 belirlendi. Yani her batch 32 resme sahip. Ve 50 adımdan oluşacak bir epoch tanımlaması yapıldı. Modelimizi tensorflow kullanarak yükledik. Böylece ana dizine bağlı 3 tane sınıfımız oluşmuş oldu. RGB ile alakalı olan 'CHANNELS' değerimiz ise 3 olarak verildi. Resimlerimizi 'shuffle' ile karıştırarak yüklüyoruz. 3 sınıfa ait 517 resmimiz bu şekilde yüklenmiş oluyor.

```

In [4]: BATCH_SIZE = 32
        IMAGE_SIZE = 256
        CHANNELS=3
        EPOCHS=50

In [5]: dataset = tf.keras.preprocessing.image_dataset_from_directory(
        "Samples",
        seed=123,

        shuffle=True,
        image_size=(IMAGE_SIZE,IMAGE_SIZE),
        batch_size=BATCH_SIZE
    )

Found 517 files belonging to 3 classes.

In [6]: class_names = dataset.class_names
        class_names

Out[6]: ['Disease', 'Healthy', 'Pest-detected']

```

Şekil 5.9 Değerlerin belirlenmesi ve veri setine yükleme kısmı

Modelimizde %80 training,%10 test,%10 validation oranı belirledik.Ve veri setimize bu oranları uyguladık.Batch size 32olarak alındığından dataset uzunluğu 17 çıkıyor.

```

In [19]: len(dataset)

Out[19]: 17

In [20]: train_size = 0.8
        len(dataset)*train_size

Out[20]: 13.600000000000001

In [41]: train_ds = dataset.take(5)
        len(train_ds)

Out[41]: 5

In [42]: test_ds = dataset.skip(5)
        len(test_ds)

Out[42]: 12

In [43]: val_size=0.1
        len(dataset)*val_size

Out[43]: 1.7000000000000002

In [44]: val_ds = test_ds.take(5)
        len(val_ds)

Out[44]: 5

In [45]: test_ds = test_ds.skip(5)
        len(test_ds)

Out[45]: 7

```

Şekil 6.1 Train,test ve validation değerlerini orarlama

Data Augmentation Ve Model Mimarisi Oluşturma

Data augmentation kısmında modelimizde rotasyonlar,döndürmeler,kontrast arttırma ve yeniden boyutlandırma gibi bazı işlemler uyguladık.Ve bazı filtreleme işlemleri uygulayarak CNN (Convolutional Neural Network) ile modelimizi sınıflandırdık.Ardından modelimizi inşa ettik.Model mimarimizi tamamladık.'softmax' ile sınıflarımız normalize ediliyor.Burda yaptığımız preprocessing işlemleriyle modelimizi eğitime hazır hale getirdik.

```
resize_and_rescale = tf.keras.Sequential([
    layers.experimental.preprocessing.Resizing(IMAGE_SIZE, IMAGE_SIZE),
    layers.experimental.preprocessing.Rescaling(1./255),
])

data_augmentation = tf.keras.Sequential([
    layers.experimental.preprocessing.RandomFlip("horizontal_and_vertical"),
    layers.experimental.preprocessing.RandomRotation(0.2),
])

input_shape = (BATCH_SIZE, IMAGE_SIZE, IMAGE_SIZE, CHANNELS)
n_classes = 3

model = models.Sequential([
    resize_and_rescale,
    layers.Conv2D(32, kernel_size = (3,3), activation='relu', input_shape=input_shape),
    layers.MaxPooling2D((2, 2)),
    layers.Conv2D(64, kernel_size = (3,3), activation='relu'),
    layers.MaxPooling2D((2, 2)),
    layers.Conv2D(64, kernel_size = (3,3), activation='relu'),
    layers.MaxPooling2D((2, 2)),
    layers.Conv2D(64, (3, 3), activation='relu'),
    layers.MaxPooling2D((2, 2)),
    layers.Conv2D(64, (3, 3), activation='relu'),
    layers.MaxPooling2D((2, 2)),
    layers.Conv2D(64, (3, 3), activation='relu'),
    layers.MaxPooling2D((2, 2)),
    layers.Flatten(),
    layers.Dense(64, activation='relu'),
    layers.Dense(n_classes, activation='softmax'),
])

model.build(input_shape=input_shape)
```

Şekil 6.2 Model mimarisi oluşturma ve data augmentation kısmı

Modelimizin özellikleri sekildeki gibidir.Eğitmemiz gereken params değerleri ve modelin diğer özellikleri verilmiştir.


```
model.summary()
```

Model: "sequential_2"

Layer (type)	Output Shape	Param #
sequential (Sequential)	(32, 256, 256, 3)	0
conv2d (Conv2D)	(32, 254, 254, 32)	896
max_pooling2d (MaxPooling2D)	(32, 127, 127, 32)	0
conv2d_1 (Conv2D)	(32, 125, 125, 64)	18496
max_pooling2d_1 (MaxPooling2D)	(32, 62, 62, 64)	0
conv2d_2 (Conv2D)	(32, 60, 60, 64)	36928
max_pooling2d_2 (MaxPooling2D)	(32, 30, 30, 64)	0
conv2d_3 (Conv2D)	(32, 28, 28, 64)	36928
max_pooling2d_3 (MaxPooling2D)	(32, 14, 14, 64)	0
conv2d_4 (Conv2D)	(32, 12, 12, 64)	36928
max_pooling2d_4 (MaxPooling2D)	(32, 6, 6, 64)	0
conv2d_5 (Conv2D)	(32, 4, 4, 64)	36928
max_pooling2d_5 (MaxPooling2D)	(32, 2, 2, 64)	0
flatten (Flatten)	(32, 256)	0
dense (Dense)	(32, 64)	16448
dense_1 (Dense)	(32, 3)	195

=====
 Total params: 183,747
 Trainable params: 183,747
 Non-trainable params: 0

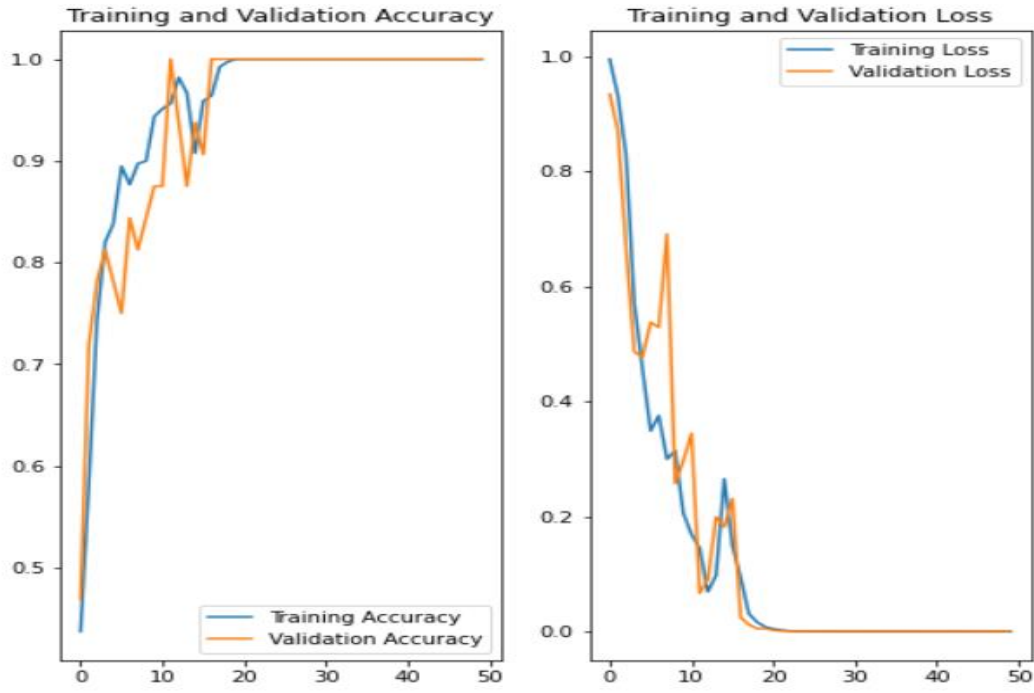
Şekil 6.3 Model özeti

```
Epoch 13/50
13/13 [=====] - 6s 492ms/step - loss: 0.0690 - accuracy: 0.9820 - val_loss: 0.0903 - val_accuracy: 0.9375
Epoch 14/50
13/13 [=====] - 6s 490ms/step - loss: 0.0975 - accuracy: 0.9666 - val_loss: 0.1990 - val_accuracy: 0.8750
Epoch 15/50
13/13 [=====] - 6s 493ms/step - loss: 0.2650 - accuracy: 0.9075 - val_loss: 0.1818 - val_accuracy: 0.9375
Epoch 16/50
13/13 [=====] - 6s 496ms/step - loss: 0.1490 - accuracy: 0.9589 - val_loss: 0.2310 - val_accuracy: 0.9062
Epoch 17/50
13/13 [=====] - 6s 489ms/step - loss: 0.0944 - accuracy: 0.9640 - val_loss: 0.0243 - val_accuracy: 1.0000
Epoch 18/50
13/13 [=====] - 6s 492ms/step - loss: 0.0311 - accuracy: 0.9923 - val_loss: 0.0120 - val_accuracy: 1.0000
Epoch 19/50
```

```
In [59]: scores = model.evaluate(test_ds)
3/3 [=====] - 0s 112ms/step - loss: 0.2679 - accuracy: 0.9792
```

Şekil 6.4 Epoch kısmı

Modelimizi 50 epoch ile eğittik. Her epoch 13 adımdan oluşuyor ve doğruluk ve kaybımız resimde yer almaktadır. Bu kısmın süresi bilgisayarın performansına göre değişebilir. Modelin genel doğruluk ve kaybı ise şekilde görülmektedir.



Tablo 1.4 Modelin doğruluk ve kayıp grafiği

Eğittiğimiz modelin doğruluk grafiği aşağıda verilmiştir. Grafiğimizde başlangıçta düşük bir doğruluk ve yüksek bir kayıpla başladık ama ileriki aşamalarda doğruluğumuz artarken kaybımız azalıyor. Ve epoch aşamamızın sonlarında değerimiz stabil hale geldi yani yeterli bir epoch sayısında modelimizi eğittik.



Şekil 6.5 Modelin çıktıları

İlk batch değerimizi alarak sonuçları görmek için bazı örnekleri çıktı olarak aldık. Çıktılarda doğruluk değeri de yer almaktadır.

Resimler 0-255 bit arası kodlanır.Tenser verilerimize numpy dönüşümü uyguladık ve ardından matplotlib kütüphanesi yardımıyla 'uint8' dönüşümü uygulayarak çıktı aldık.Modelde 'actual class','predicted class' ve 'confidence' değerleriyle eğitimimizin sonuçlarını test ettik.Predict fonksiyonunda ilk batch değerleri alındı ve oran 100 ile çarpılarak 'confidence' değeri belirlendi.

```
def predict(model, img):
    img_array = tf.keras.preprocessing.image.img_to_array(images[i].numpy())
    img_array = tf.expand_dims(img_array, 0)

    predictions = model.predict(img_array)

    predicted_class = class_names[np.argmax(predictions[0])]
    confidence = round(100 * (np.max(predictions[0])), 2)
    return predicted_class, confidence

plt.figure(figsize=(15, 15))
for images, labels in test_ds.take(1):
    for i in range(9):
        ax = plt.subplot(3, 3, i + 1)
        plt.imshow(images[i].numpy().astype("uint8"))

        predicted_class, confidence = predict(model, images[i].numpy())
        actual_class = class_names[labels[i]]

        plt.title(f"Actual: {actual_class},\n Predicted: {predicted_class}.\n Confidence: {confidence}%")

    plt.axis("off")
```

Şekil 6.6 Eğitilen modelin sonuçlarını gösterme

En son adımda ise eğittiğimiz modeli auto increment şekilde klasörümüze kaydediyoruz.

```
import os
model_version=max([int(i) for i in os.listdir("../models") + [0]])+1
model.save(f"../models/{model_version}")

WARNING:absl:Found untraced functions such as _jit_compiled_convolution_op, _jit_compiled_convolution_op, _jit_compiled_convolution_op, _jit_compiled_convolution_op, _jit_compiled_convolution_op while saving (showing 5 of 7). These functions will not be directly callable after loading.

INFO:tensorflow:Assets written to: ../models/4/assets

INFO:tensorflow:Assets written to: ../models/4/assets

model.save("../lettuce.h5")
```

Şekil 6.7 Modelin kaydedilmesi

Projenin Backend Bölümü

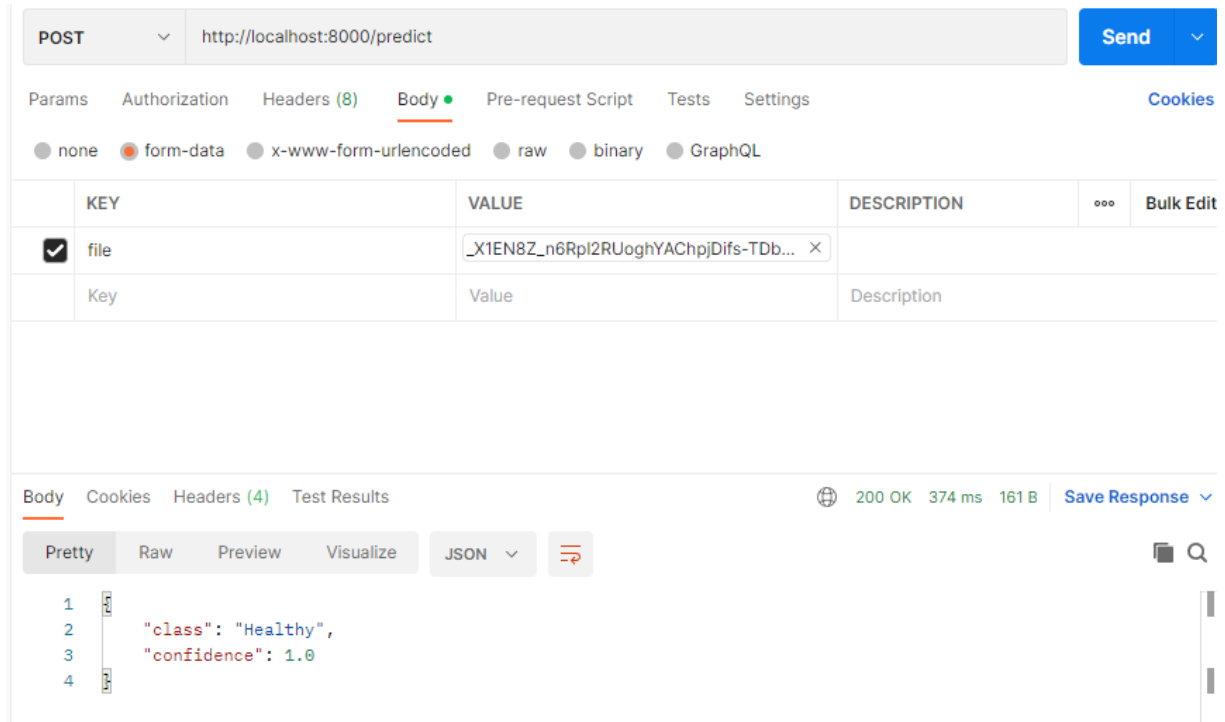
Model eğitiminden sonra projemizin backend kısmına başladık. Modellerimizi yüklemek için Fast Api server ve Tf serving kullandık. Kaydettiğimiz modelleri ve sınıflarımızı tanımladık.

```
app=FastAPI()
origins = [
    "http://localhost:3000",
]
app.add_middleware(
    CORSMiddleware,
    allow_origins=origins,
    allow_credentials=True,
    allow_methods=["*"],
    allow_headers=["*"],
)
MODEL = tf.keras.models.load_model("../models/4")
CLASS_NAMES = ["Disease", "Healthy", "Pest-detected"]
@app.get("/ping")
async def ping():
    return "Hi !"
def read_file_as_image(data) -> np.ndarray:
    image=np.array(Image.open(BytesIO(data)))
    return image
@app.post("/predict")
async def predict(
    file: UploadFile = File(...)
):
    image = read_file_as_image(await file.read())
    img_batch=np.expand_dims(image,0)
    predictions=MODEL.predict(img_batch)
    predicted_class=CLASS_NAMES[np.argmax(predictions[0])]
    confidence = np.max(predictions[0])
    return{
        'class':predicted_class,
        'confidence':float(confidence)
    }
```

Şekil 6.8 Pycharm ıde kullanarak yaptığımız backend dosyası

Fast Api bir Rest Api geliştirme aracıdır.Yüksek performansa sahip olduğundan projemizde bu API den yararlandık.Tf serving ise makine öğrenimi uygulamaları için yüksek performans sağlayan bir sistemdir.Predict kısmında dosyamızı yüklüyoruz.Ve local host tanımlaması ve çalışacağımız model klasörünü belirliyoruz.Ayrıca k,sınıflarımızı da belirttik.Hangi sınıf daha yüksek orana sahipse predict kısmında o sınıfı gösterecek.

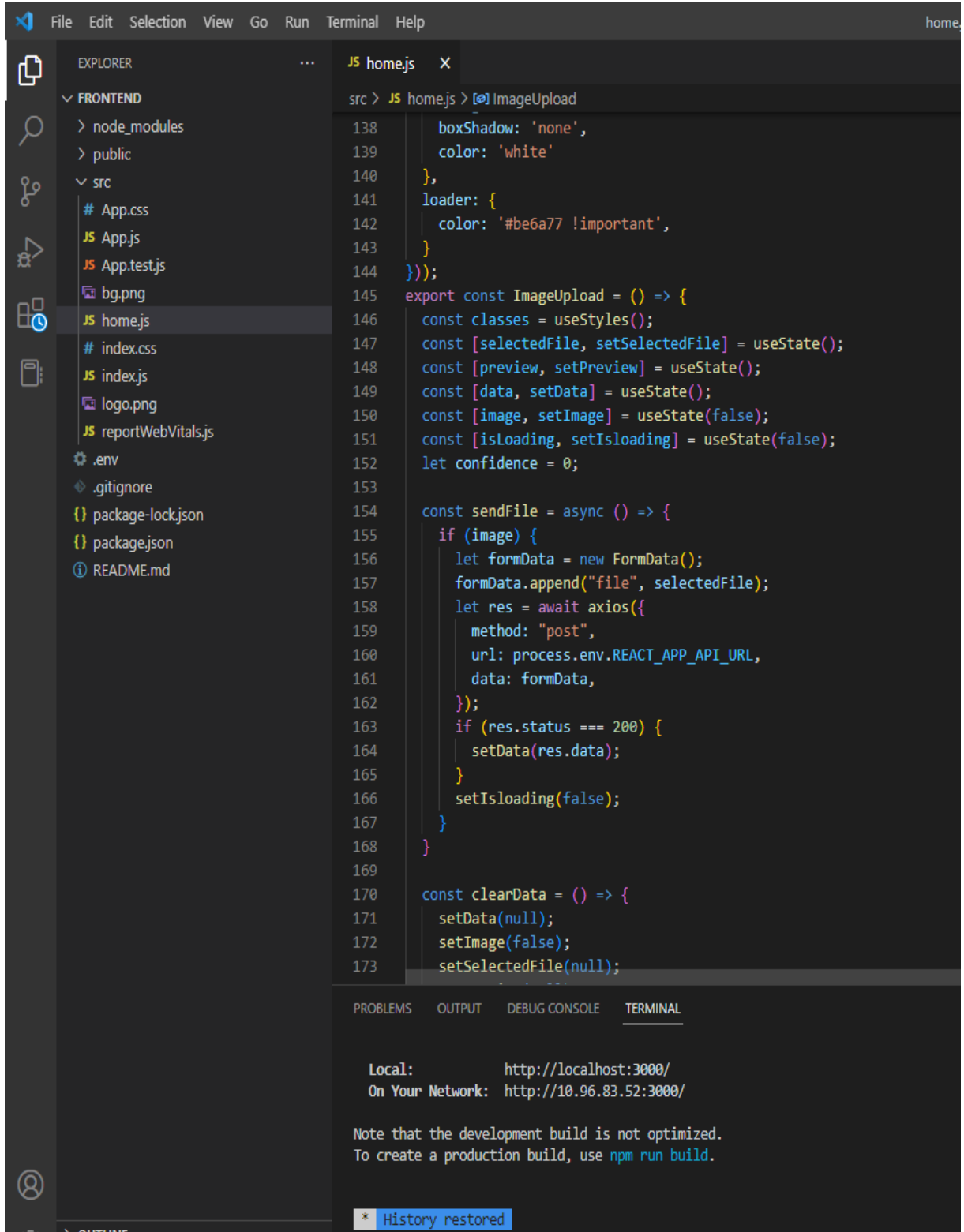
API testi için Postman kullandık ve burada elde ettiğimiz sonuçları gördük. Postman API testini gözlemleyen bir API platformudur. Resimde de görüldüğü üzere API doğru bir şekilde çalışmaktadır. Böylelikle projemizin frontend kısmına geçebiliriz.



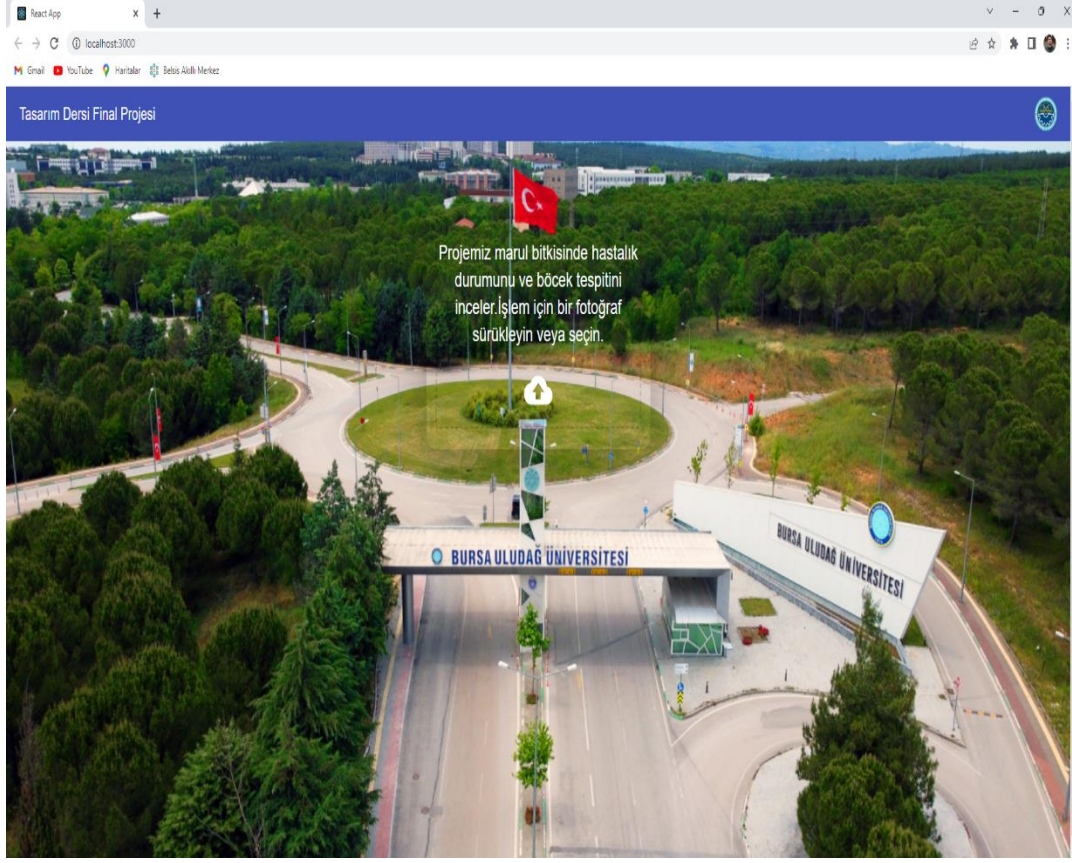
Şekil 6.9 Postman ile API doğruluğunu test etme

Projenin Frontend Bölümü

Frontend arayüzümüzün dosya yapısı ve kodları aşağıda verilmiştir. Local host'da çalışan projemize uygun html ve css tanımları yaptık. Dosya yükleme fonksiyonumuz şekildeki gibidir. Bu fonksiyon bir html fragment dönüyor. Ayrıca API url tanımlamamızı da yaptık. Local host 8000 de çalışan backend kısmından aldığımız url ile local host 3000 de projemizi ayağa kaldırdık. Arayüzümüzü ReactJs kullanarak oluşturduk. Visual studio code ide kullandık.



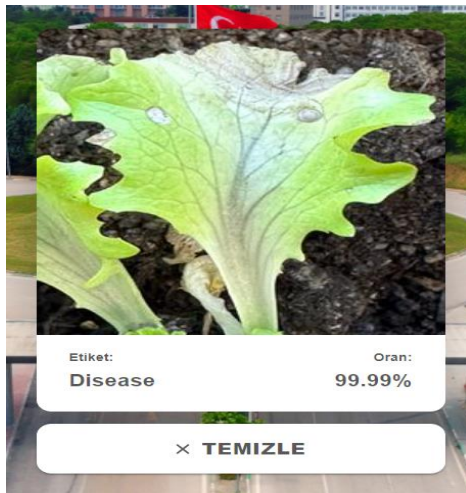
Şekil 7.1 Resim yükleme kısmımız ve diğer dosyalar



Şekil 7.2 Projenin arayüz kısmı

Sonuçlar

Aşağıdaki resimlerde ise sonuçlarımızı farklı etiketler için gözlemliyoruz.



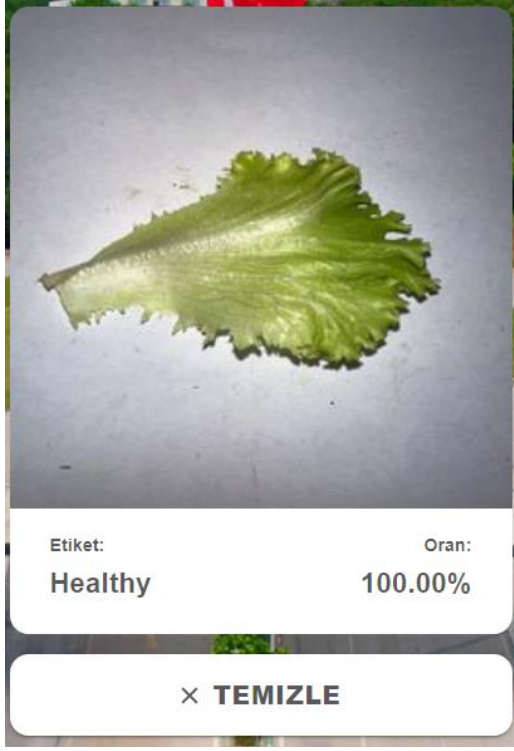
Şekil 7.3 Hastalıklı etikete sahip marul örneği



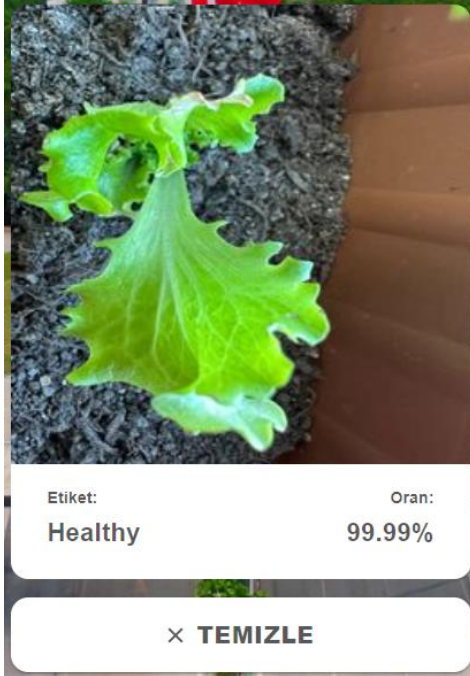
Şekil 7.4 Hastalıklı etikete sahip marul örneği



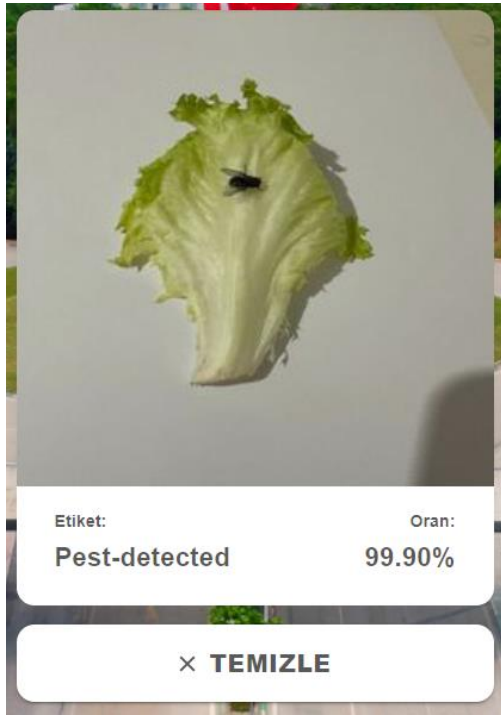
Şekil 7.5 Hastalıklı etikete sahip marul örneği



Şekil 7.6 Sağlıklı etikete sahip marul örneği



Şekil 7.7 Sağlıklı etikete sahip marul örneği



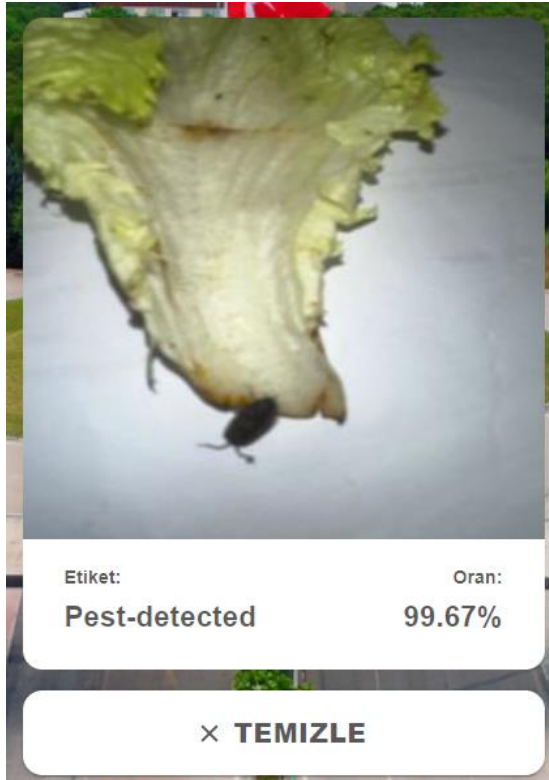
Şekil 7.8 Pest detected etiketine sahip marul örneği



Şekil 7.9 Pest detected etiketine sahip marul örneği



Şekil 8.1 Pest detected etiketine sahip marul örneği



Şekil 8.2 Pest detected etiketine sahip marul örneği

13. KAYNAKÇA

Hu, X., Sun, L., Zhou, Y. et al. Review of operational management in intelligent agriculture

based on the Internet of Things. *Front. Eng. Manag.* 7, 309–322 (2020).
<https://doi.org/10.1007/s42524-020-0107-3>

Kasinathan T., Singaraju D., Uyyala S.R.

(2021) *Information Processing in Agriculture*, 8 (3), pp. 446-457.

Insect classification and detection in field crops using modern machine learning techniques

N. Pavithra and V. S. Murthy, “An Image Processing algorithm for pest detection”,
pices,
vol. 1, no. 3, pp. 24-26, Jul. 2017.

S. D. Khirade and A. B. Patil, "Plant Disease Detection Using Image Processing,"
2015 International Conference on Computing Communication Control and
Automation, 2015, pp. 768-771, doi: 10.1109/ICCUBEA.2015.153.

Alon et al., *International Journal of Emerging Trends in Engineering Research*,
8(4), April 2020, 1073 - 1078

<http://www.warse.org/IJETER/static/pdf/file/ijeter20842020.pdf>

K. Sabancı, M. F. Ünleşen, and Y. Dilay, “Karaman Yöresinde Yetiştirilen Elma
Çeşitlerinin Sınıflandırma Parametrelerini Görüntü İşleme Teknikleri Kullanarak
Belirlenmesi,” 2016.

M. B. Eminoğlu, R. Öztürk, A. İ. Acar, and V. Kalinkara, “MEYVE HASADINDA
KULLANILAN HASAT PLATFORMLARININ ÇALIŞMA KOŞULLARININ
İYİLEŞTİRİLMESİ YÖNÜNDEN DEĞERLENDİRİLMESİ,” *J. Eng. Sci. Des.*, vol. 3, no. 3,
pp. 233–238, 2015.

K. Berkk, K. Artan, and M. Orhan, “AĞAÇTAKİ KİRAZIN TESPİT EDİLMESİ,”
2017.

H. Hassannejad, G. Matrella, P. Ciampolini, I. De Munari, M. Mordonini, and S. Cagnoni,
“Food image recognition using very deep convolutional networks,” *MADiMa 2016 -
Proc. 2nd Int. Work. Multimed. Assist. Diet. Manag. co-located with ACM Multimed.
2016*, pp. 41–49, 2016.

- E. Sert, D. Taşkin, and N. Suçsuz, "GÖRÜNTÜ İŞLEME TEKNİKLERİ ile ŞEFTALİ ve ELMA SINIFLANDIRMA," *tujs Trak. Univ J Sci*, vol. 11, no. 2, pp. 82–88, 2010.
- S. SOLAK and U. ALTINIŞIK, "Görüntü İşleme Teknikleri ve Kümeleme Yöntemleri Kullanılarak Fındık Meyvesinin Tespit ve Sınıflandırılması," *SAÜ Fen Bilim. Enstitüsü Derg.*, 2018.
- H. Polatçı, A. K. Yıldız, O. Saraçoğlu, E. Adsız, and B. Aksüt, "Görüntü İşleme Yöntemleri Kullanılarak Kivi Meyvesinin Kuruma Performansı ve Renk Değişiminin Belirlenmesi," pp. 105–112, 2017.
- M. Kabak, F. Sağlam, and A. Aktas, "Farklı uzaklık hesaplama yaklaşımlarının TOPSIS üzerinde kullanılabilirliğinin incelenmesi," *J. Fac. Eng. Archit. Gazi Univ.*, vol. 32, pp. 35–43, 2017.
- A. Yılmaz, A. Çakır, and O. Demirözer, "Counts of rose soft scale (*Rhodococcus perornatus* Cockerell & Parrott) (Hemiptera: Coccidae) by using image processing methods," 2018.
- M. TÜRKOĞLU and D. HANBAY, "Apricot Disease Identification based on Attributes Obtained from Deep Learning Algorithms," 2018 International Conference on Artificial Intelligence and Data Processing (IDAP), 2018, pp. 1-4, doi: 10.1109/IDAP.2018.8620831.
- Altaş, Ziya & Ozguven, Mehmet & Yanar, Yusuf. (2019). Bitki Hastalık ve Zararlı Düzeylerinin Belirlenmesinde Görüntü İşleme Tekniklerinin Kullanımı:Şeker Pancarı Yaprak Leke Hastalığı Örneği.
- Altaş Z. , Özgüven M. M. , Dilmaç M. Görüntü İşleme Teknikleri ile Bağ Yaprak Uyuzu Hasarının Belirlenmesi. Gaziosmanpaşa Bilimsel Araştırma Dergisi. 2021; 10(3): 77-87.
- Bartolome T. ,Tanguilig IiiJohnny L. Miranda,Bobby Dioquino Gerardo International Journal of Computer and Communication Engineering 3(3):189-192, DOI:10.7763/IJCCE.2014.V3.317
- Bal F. , Kayaalp F. Review of machine learning and deep learning models in agriculture.International Advanced Researches and Engineering Journal. 2021; 5(2): 309-323.
- Öztürk, E., Çelik, Y. & Kırıcı, P. (2021). Akıllı Tarımda Sensör Uygulaması. Avrupa Bilim ve Teknoloji Dergisi, (28), 1279-1282.

Z. Shi, H. Dang, Z. Liu and X. Zhou,
"Detection and Identification of Stored-Grain Insects Using Deep Learning: A More Effective Neural Network," in IEEE Access, vol. 8, pp. 163703-163714, 2020, doi: 10.1109/ACCESS.2020.3021830.

Bhargava, Anuja & Bansal, Atul. (2018). Fruits and Vegetables Quality Evaluation Using Computer Vision:A Review. Journal of King Saud University - Computer and Information Sciences.33. 10.1016/j.jksuci.2018.06.002.

Ruizhe Yang, Zhenchao Wu, Wentai Fang, Hongliang Zhang, Wenqi Wang, Longsheng Fu, Yaqoob Majeed, Rui Li, Yongjie Cui, Detection of abnormal hydroponic lettuce leaves based on image processing and machine learning, Information Processing in Agriculture, 2021

R. Gozzovelli, B. Franchetti, M. Bekmurat and F. Pirri,
"Tip-burn stress detection of lettuce canopy grown in Plant Factories,"
2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW), 2021, pp. 1259-1268, doi: 10.1109/ICCVW54120.2021.00146.

Ağın, O. & Malaslı, M. Z. (2016). Görüntü İşleme Tekniklerinin Sürdürülebilir Tarımdaki Yeri ve Önemi: Literatür Çalışması . Tarım Makinaları Bilimi Dergisi , 12 (3) , 199-206 . Retrieved from <https://dergipark.org.tr/tr/pub/tarmak/issue/35278/391531>

R. K. Samanta and I. Ghosh, "Tea insect pests classification base on artificial neural networks," International Journal of Computer Engineering Science, vol. 2, June 2012.

M. T. Do et al., "A test of a pattern recognition system for identification of spiders," Bulletin of Entomological Research, 1999.

S. M. Al-Saqer, "A robust recognition system for pecan weevil using artificial neural networks," American Journal of App. Sci., 2012.

L. Zhu and Z. Zhang, "Insect recognition based on integrated region matching and dual tree complex wavelet transform," Journal of Zhejiang University-SCIENCE C (Computers & Electronics), 2011.

Demirbas, A. A. & Çınar, A. (2020). Nesne Sınıflandırma İşlemi İçin Tensor İşleme Birimi ve CPU Performans Karşılaştırması . Bilgisayar Bilimleri ve Teknolojileri Dergisi , 1 (1) , 10-15

Terzi, M. S. (2021). Derin öğrenme ile göğüs röntgenlerinden hastalık teşhisi. (Yayınlanmamış Yüksek Lisans Tezi). Necmettin Erbakan Üniversitesi, Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı, Konya.

ALEEM, Sidra, et al. Random data augmentation based enhancement: a generalized enhancement approach for medical datasets. *arXiv preprint arXiv:2210.00824*, 2022.

OĞUZLAR, Ayşe. Veri ön işleme. *Erciyes Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi*, 2003, 21.

AKSOY, Bekir; HALİS, Helin Diyar; SALMAN, Osamah Khaled Musleh. Elma bitkisindeki hastalıkların yapay zekâ yöntemleri ile tespiti ve yapay zekâ yöntemlerinin performanslarının karşılaştırılması. *International Journal of Engineering and Innovative Research*, 2020, 2.3: 194-210.

14. TEŞEKKÜR

Projemizin literatür araştırmasında makalelerini incelediğimiz ve atıf verdiğimiz değerli bilim insanlarına, makalelere açık kaynak erişim sağlayan platformlara ve bize bu projede yol gösteren danışman hocamız Doç. Dr. Pınar Kırıcı'ya teşekkür ederiz.