Derin Öğrenme Tabanlı Akıllı Tarım ve Uygulamaları

Yazılım Mühendisliği Ana Bilim Dalı

Yüksek Lisans Tezi

Furkan Alp Esen

ORCID 0000-0002-2157-3057

Tez Danışmanı: Doç. Dr. Aytuğ Onan

Kasım 2022

İzmir Kâtip Çelebi Üniversitesi Fen Bilimleri Enstitüsü öğrencisi **Furkan Alp Esen** tarafından hazırlanan **Derin Öğrenme Tabanlı Akıllı Tarım ve Uygulamaları** başlıklı bu çalışma tarafımızca okunmuş olup, yapılan savunma sınavı sonucunda kapsam ve nitelik açısından başarılı bulunarak jürimiz tarafından YÜKSEK LİSANS TEZİ olarak kabul edilmiştir.

ONAYLAYANLAR:

Tez Danışmanı: **Doç. Dr. Aytuğ Onan**
İzmir Kâtip Çelebi Üniversitesi

Jüri Üyeleri:

Doç. Dr. Aytuğ Onan
İzmir Kâtip Çelebi Üniversitesi

Doç. Dr. Vahide Bulut
İzmir Kâtip Çelebi Üniversitesi

Dr. Öğr. Üyesi Emin Borandağ
Manisa Celal Bayar Üniversitesi

Savunma Tarihi: 08.12.2022

Yazarlık Beyanı

Ben, **Furkan Alp Esen**, başlığı **Derin Öğrenme Tabanlı Akıllı Tarım ve Uygulamaları** olan bu tezimin ve tezin içinde sunulan bilgilerin şahsıma ait olduğunu beyan ederim. Ayrıca:

- Bu çalışmanın bütünü veya esası bu üniversitede Yüksek Lisans derecesi elde etmek üzere çalıştığım süre içinde gerçekleştirilmiştir.
- Daha önce bu tezin herhangi bir kısmı başka bir derece veya yeterlik almak üzere bu üniversiteye veya başka bir kuruma sunulduysa bu açık biçimde ifade edilmiştir.
- Başkalarının yayımlanmış çalışmalarına başvurduğum durumlarda bu çalışmalara açık biçimde atıfta bulundum.
- Başkalarının çalışmalarından alıntıladığımda kaynağı her zaman belirttim. Tezin bu alıntılar dışında kalan kısmı tümüyle benim kendi çalışmamdır.
- Kayda değer yardım aldığım bütün kaynaklara teşekkür ettim.
- Tezde başkalarıyla birlikte gerçekleştirilen çalışmalar varsa onların katkısını ve kendi yaptıklarımı tam olarak açıkladım.

Tarih: 22.11.2022

Derin Öğrenme Tabanlı Akıllı Tarım ve Uygulamaları

Öz

Geleneksel tarım faaliyetlerine yapay zekayı entegre edebilmek için yeterli donanıma ve bilgiye sahibiz. Bu çalışma kapsamında geliştirilen mobil uygulama ile yapay sinir ağları kullanılarak, çiftçilerin mobil uygulama üzerinden bitkilerin hastalıklarını tespit edebilmesi sağlanmıştır. Eğitilen derin öğrenme modelleri sayesinde üreticiler; 14'ten fazla hastalıklı veya sağlıklı bitki yapraklarının fotoğrafını çekerek yapay zeka ile rahatlıkla durumunu öğrenebilir ve yayılmakta olan hastalığa oldukça hızlı bir şekilde tedavi uygulayabilir hale gelmiştir. Üretici; farklı araziler üzerinde ektiği veya hasat ettiği bitkileri uygulama envanterine kaydederek her yerden kontrol edebilir, envanterinde yetiştirilmekte olan bitkilerin de belirli aralıklar ile fotoğraflarını çekerek, mobil uygulama üzerinden derin öğrenme modelini kullanarak gelişim oranlarını hesaplayıp veritabanına kayıt edebilmektedir. Üreticiler; belirli aralıklar ile görsellerini toplayarak sahip oldukları bitkilerin gelişim grafiklerini uygulama içerisinde takip edebilmektedirler. Ziraatçılar ve çiftçiler uydu veya insansız hava araçlarından alınan görüntüleri kullanarak mobil uygulama içerisinde araziye yayılan su, ilaç, kuruma vb değişimleri kontrol edilebilmektedirler.

Anahtar Sözcükler: Yazılım, yapay zeka, derin öğrenme, bilgisayarlı görü, arazi, bitki, yaprak, veriseti, sınıflandırma, model eğitim ve testleri, veritabanı, servis yapısı, mobil uygulama, entegrasyon ve testler

Deep Learning Based Smart Agriculture and Applications

Abstract

We have sufficient equipment and knowledge to integrate artificial intelligence into traditional agricultural activities. With the mobile application developed within the scope of this study, artificial neural networks were used to enable farmers to detect plant diseases through the mobile application. Thanks to the trained deep learning models, manufacturers; By taking pictures of more than 14 diseased or healthy plant leaves, it has become able to easily learn its condition with artificial intelligence and apply treatment to the spreading disease quite quickly. Producer; It can control the plants that it has planted or harvested on different lands by recording it in the application inventory, and can also take pictures of the plants grown in its inventory at certain intervals, calculate the development rates using the deep learning model via the mobile application and save it to the database. Producers; by collecting their images at certain intervals, they can follow the development graphics of the plants they have in the application. Agriculturalists and farmers can use the images taken from satellite or unmanned aerial vehicles to control the changes in water, medicine, drying etc. spread over the land within the mobile application.

Keywords: Software, artificial intelligence, deep learning, computer vision, land, plant, leaf, dataset, classification, model training testing, database, service architecture, mobile application, integration and testing

Tez çalışmamı başta aileme, öğretmenlerime ve en çok da her zaman ilham kaynağım olan değerli 'Ziraat Mühendisi' dedem Veli Doğan'a ithaf ediyorum.

Teşekkür

Tez çalışmama katkıda bulunan; saygı değer danışman hocam Aytuğ ONAN'a, görüşlerini bildirmekten çekinmeyen ulaştığım diğer hocalarıma ve esinlendiğim benzer çalışmaları yapan literatürdeki ve sektördeki herkese teşekkür ederim.

İçindekiler

Yazarlık Beyanı ii

Öz iii

Abstract iv

Teşekkür vi

Şekiller Listesi x

Tablolar Listesi xii

Kısaltmalar Listesi xiii

Semboller Listesi xvi

1 Giriş 1

1.1 Hastalık Tespiti 6

1.2 Bitki Sınıflandırma 7

1.3 Arazi Yapısı Tanımlama 8

1.4 Zararlı Böcek Tanıma 9

1.5 Akıllı Sulama 10

1.6 Zararlı Bitki Tanıma 11

1.7 Toprak Yönetimi 12

1.8 Verim Kestirimi 12

2 Literatür Taraması 14

3 Metodoloji 17

3.1 Yapay Zeka 17

3.2 Yapay Zeka Türleri 18

3.2.1 Yapay Dar Zeka 18

3.2.2 Yapay Genel Zeka 19
3.2.3 Yapay Süper Zeka 19
3.2.4 Makine Öğrenmesi 19
3.2.5 Derin Öğrenme 19
3.3 Bilgisayarlı Görü 21
3.4 Problemler ve Uygulanan Yöntemler 22
3.4.1 Konvolüsyonel (Evrişimsel) Sinir Ağları (CNN) 23
3.4.1.1 Evrişimsel Sinir Ağları 23
3.4.1.2 Evrişimsel Sinir Ağları Çalışma Prensibi 24
3.4.1.3 Evrişimsel Sinir Ağı Türleri 25
3.4.2 Aktarımlı Öğrenme (Transfer Learning) 25
3.4.2.1 AlexNet 26
3.4.2.2 DenseNet 27
3.4.2.3 VGG16 28
3.4.2.4 ResNet18 29
3.4.2.5 EfficientNet B0 – EfficientNet B7 30
3.4.3 Hastalık Tespiti 31
3.4.3.1 Veriseti 31
3.4.3.2 Sinir Ağı Yapısı ve Eğitimi 33
3.4.4 Bitki Gelişim Takibi 35
3.4.4.1 Mimari Yapı 36
3.4.4.2 Uygulama 39
3.4.5 Arazi Durum Kontrolü 42
4 Gerçekleştirim 50
4.1 Kullanılan Araçlar 50
4.1.1 Python 50
4.1.1.1 FastAPI 51

4.1.2 Flutter 53
4.1.3 SQLite 59
4.2 Genel İşleyiş 61
4.2.1 Hava Durumu Özelliği 62
4.2.2 Envanter Yapısı 63
4.2.3 Servis Yapısı 63
4.2.4 Hastalık Tespiti Yapısı 63
4.2.5 Gelişim Takibi Yapısı 63
4.2.6 Arazi Durum Tespiti Yapısı 64
5 Deneysel Sonuçlar ve Tartışma 65
5.1 Hastalık Tespiti Çıkarımlar 65
5.2 Gelişim Takibi Çıkarımlar 74
5.3 Genel Mimari Tasarım Çıkarımlar 76
6 Sonuç 77
Kaynaklar 79
Özgeçmiş 89

Şekiller Listesi

Şekil 1.1 Akıllı tarımın uygulandığı alan 2
Şekil 1.2 Akıllı tarım sistemleri örneği 4
Şekil 1.3 Akıllı tarım ve alt alanlarından bazıları 6
Şekil 1.4 Hastalıklı Bitki 7
Şekil 1.5 Yaprak görüntülerinden sınıflandırılmış bitkiler 8
Şekil 1.6 Uydu görüntüsü üzerinde ayrıştırılmış alanlar 9
Şekil 1.7 Zararlı böcek temsili gösterim 10
Şekil 1.8 Akıllı sulama temsili gösterim 10
Şekil 1.9 Zararlı bitkilerin görsel veriler ile tespit edilmesi 11
Şekil 1.10 Toprak Yönetimi temsili gösterim 12
Şekil 1.11 Ürün veriminin zaman içerisinde tahmin edilmesi 13
Şekil 3.1 Yapay zeka ve alt alanları 20
Şekil 3.2 Yapay sinir ağı 22
Şekil 3.3 Fotoğraf üzerinde tespit edilip işaretlenen insanlar ve nesneler 22
Şekil 3.4 Görsel veri üzerinden hayvanları bulunması 22
Şekil 3.5 Evrişimsel sinir ağının yapısı 24
Şekil 3.6 Aktarımlı öğrenme ve normal öğrenme ile sınıflandırma 26
Şekil 3.7 AlexNet sinir ağının yapısı 27
Şekil 3.8 DenseNet sinir ağının yapısı 28
Şekil 3.9 VGG16 sinir ağının yapısı 29
Şekil 3.10 ResNet18 sinir ağının yapısı 30
Şekil 3.11 EfficientNet sinir ağının yapısı 31
Şekil 3.12 Verisetinin içeriğinden bir kesit 32
Şekil 3.13 Bitki hastalık tespiti akış diyagramı 33
Şekil 3.14 Tahmin görüntüleri, (a) EfficientNet B7, (b) AlexNet 35
Şekil 3.15 MaskRCNN mimarisi ile cadde üzerindeki araçların, yayaların ve farklı nesnelerin tespit edilmiş hali 36
Şekil 3.16 R-CNN mimarisi 37

Şekil 3.17 Segmente edilmiş bir görüntü, (a) Semantik segmentasyon, (b) Örnek segmentasyonu 38
Şekil 3.18 Gelişim takibi yapısı..... 39
Şekil 3.19 MaskRCNN birinci girdi ve çıktı örneği, (a) Gelişimdeki bir bitki, (b) Maskelenmiş hali..... 40
Şekil 3.20 MaskRCNN ikinci girdi ve çıktı örneği, (a) Gelişimdeki bir bitki, (b) Maskelenmiş hali..... 40
Şekil 3.21 MaskRCNN üçüncü girdi ve çıktı örneği, (a) Gelişimdeki bir bitki, (b) Maskelenmiş hali..... 41
Şekil 3.22 Gelişim takibi grafiği 41
Şekil 3.23 Arazi durum kontrolü görüntü işleme süreci..... 43
Şekil 3.24 Uydu görüntüsü üzeirnde ekim arazilerinin işlenmiş hali..... 44
Şekil 3.25 Uydu görüntüsü üzerinde mahsul verimi için yapılan çalışma 45
Şekil 3.26 Uydu görüntüsü üzerinde örnek bir çalışma 46
Şekil 3.27 OpenCV kütüphanesinin çalışma şekli 46
Şekil 3.28 Görüntü işleme süreci ve kullanıcıya gönderme..... 47
Şekil 3.29 Üreticinin uygulama üzerinden gönderdiği görüntü 47
Şekil 3.30 Servisten gelen haritalanmış (mapping) görüntü 48
Şekil 3.31 Burdur gölü yakınındaki arazilerin uydu görüntüsü 48
Şekil 3.32 Servisten gelen haritalanmış görüntü..... 48
Şekil 3.33 Servisten gelen farklı haritalanmış görüntü 49
Şekil 4.1 Python FastAPI hastalık tespiti servisinden görüntü..... 52
Şekil 4.2 Python FastAPI envanter sistemi ve gelişim takibi servisinden görüntü 52
Şekil 4.3 Mobil uygulama anasayfa hava durumu ve envanter durumu pasta grafiği 54
Şekil 4.4 Mobil uygulama envanter sayfası 54
Şekil 4.5 Mobil uygulama envantere bitki ekleme sayfası..... 55
Şekil 4.6 Mobil uygulama bitki detay sayfası 55
Şekil 4.7 Mobil uygulama örnek mısır verisi ile hatalık tahmini..... 56
Şekil 4.8 Mobil uygulama örnek fasulye bitkisi verisi ile gelişim hesaplama 56
Şekil 4.9 Mobil uygulama fasulye bitkisi gelişim grafiği 57
Şekil 4.10 Mobil uygulama arazi değişim tespit sayfası..... 57
Şekil 4.11 Mobil uygulama örnek arazi verisi ile değişimi görüntüleme..... 58

Şekil 4.12 Mobil uygulama farklı metod kullanarak arazinin değişimini görüntüleme 58
Şekil 4.13 Mobil uygulama farklı metod ve farklı arazi verisi kullanarak değişimini görüntüleme..... 59
Şekil 4.14 Gelişim takibi ilişkisel veritabanı yapısı 60
Şekil 4.15 Bitki envanteri SQLite veritabanı json formatındaki örnek görüntü..... 60
Şekil 4.16 Sistemin mimari şeması 61
Şekil 5.1 AlexNet eğitim ve tahmin görüntüleri, (a) Eğitim, (b) Tahmin 67
Şekil 5.2 DenseNet161 eğitim ve tahmin görüntüleri, (a) Eğitim, (b) Tahmin..... 68
Şekil 5.3 ResNet18 eğitim ve tahmin görüntüleri, (a) Eğitim, (b) Tahmin..... 69
Şekil 5.4 VGG16 eğitim ve tahmin görüntüleri, (a) Eğitim, (b) Tahmin 70
Şekil 5.5 EfficientNetB0 eğitim ve tahmin görüntüleri, (a) Eğitim, (b) Tahmin ... 71
Şekil 5.6 EfficientNetB7 eğitim ve tahmin görüntüleri, (a) Eğitim, (b) Tahmin ... 72
Şekil 5.7 Modellerin genel performans karşılaştırması 73
Şekil 5.8 MaskRCNN modeli örnek çıktısı 74
Şekil 5.9 MaskRCNN gelişim takibi girdi ve çıktı örneği, (a) Girdi, (b) Çıktı..... 75
Şekil 5.10 MaskRCNN gelişim takibi girdi ve çıktı başka bir örneği, (a) Girdi, (b) Çıktı 75

Tablolar Listesi

Tablo 3.1 Modellere göre eğitim süreleri ve başarımları 34
Tablo 5.1 Modellerden alınan sonuçlar 66
Tablo 5.2 AlexNet tahmin sonuçları..... 67
Tablo 5.3 DenseNet161 tahmin sonuçları 68
Tablo 5.4 ResNet18 tahmin sonuçları 69
Tablo 5.5 VGG16 tahmin sonuçları 70
Tablo 5.6 EfficientNetB0 tahmin sonuçları..... 71
Tablo 5.7 EfficientNetB7 tahmin sonuçları..... 72

Kısaltmalar Listesi

DL	Deep Learning
YSA	Yapay Sinir Ağı
TL	Transfer Learning
YZ	Yapay Zeka
ML	Makine Öğrenmesi
SOA	Service Oriented Architecture
OOP	Object Oriented Programming
PL	Programming Language
CNN	Convolutional Neural Network
VGG	Very Deep Convolutional Neural Network
ResNet	Residual Networks
DenseNet	Densely Connected Convolutional Neural Networks
RCNN	Region Based Convolutional Neural Networks
MaskRCNN	Masked Region Based Convolutional Neural Networks
MDS	Mendeley Dataset
DB	Database
ACC	Accuracy
DA	Data augmentation
DC	Diseases Classification
IEEE	The Institute of Electrical and Electronics Engineers
CV	Computer Vision
OD	Object Detection

IB	Image Based
FC	Fully Connected
PL	Pooling Layer
FL	Flatten Layer
AD	Akış Diyagramı
BE	Best Epoch
YBİH	Yeşil Bitki Örtüsü İndex Haritaları
HR	High Resolution
API	Application Programming Interface
UI	User Interface
UX	User Experience

Bölüm 1

Giriş

İnsan ihtiyaçlarının yüzdelik dilimde en çok yer kaplayan kısmını gıda ihtiyacı oluşturmaktadır. Artan gıda taleplerini karşılamak için üretim faaliyetlerinin de artması gerekmektedir. Bu beklenen artış; üretimin daha verimli yapılması ile sağlanacaktır. Tarımsal alandaki verimleri artırmanın bir çok yolu bulunmaktadır [2]. Başta kaliteli tohumların kullanılması, sulama, gübreleme ve hastalıkların doğru tespit edilerek etkili ilaçlamaların yapılması gibi parametreler üzerinde çalışmalar yapılarak kayda değer farkları elde edebiliriz. Fakat her ne kadar üzerinde çalışmalar artsa da zaman içerisinde geleneksel yöntemler azaltılarak yerini akıllı tarım çözümlerine bırakmalıdır [2]. Geleneksel yöntemler asla göz ardı edilmemelidir çünkü akıllı tarım yapabilmek için birçok yazılımın ham maddesi olan verileri geleneksel yöntemler ile elde edebiliriz. Toplanan verilerin değerlendirilmesi, analizlerinin yapılması, programların geliştirilmesi ve bakımların yapılması kendi içerisinde farklı süreçleri beraberinde getirmektedir. Tüm bu süreçlerin aslında temeline baktığımızda verinin ne kadar önemli olduğunu görebiliriz [3].

Akıllı tarım faaliyetlerinin ilerletilebilmesi için geleneksel yöntemlerden toplanan verilerin incelenmesi gerekir. İncelen veriler ile tespit edilen sorunların dağılımına kabaca baktığımızda genel olarak sorunların; toprak, bitki ve kullanılan üretim malzemelerin üzerinde dağıldığınız görebiliriz. Bu dağılımlar aslında bizlere çözümün de paralel olarak nerelerde olacağını işaret etmektedir. Problemlerin çözüme ulaşılabilmesi için verilerin istatistiksel sonuçları göz önünde bulundurarak yazılımların üretilmesi gerekmektedir. Yapay zeka bize bu yazılımların geliştirilmesine zemin hazırladığı için izlenecek yol haritası da yapay zeka ve alt alanlarından devam etmek olacaktır [1]. Yapay zeka daha genel bir tanım olmakla beraber bu tez kapsamında derin öğrenme başlığına odaklanacağız. Yapılacak

geliştirmelerin; analizlerin, alınan sonuçların metrikler ile değerlendirmesi ile uygulamaların üretilmesi derin öğrenme yöntemine dayanacaktır.

Dünya genelinde; üniversitelerde, ticari kurumlarda farketmeksizin akıllı tarım alanında uygulamalarını artırmaya yönelik yapay zeka çalışmalarının yapıldığı bilinmektedir. Artık günümüzde devletler, tarımsal faaliyetlerdeki yetersizliğin veya dezavantaj durumları ekarte etmek amacıyla birçok gıda ürününe ait verileri toplamakta ve analitik tabanlara oturtmak için çalışmalar yapmaktadır [2]. Yukarıda bahsi geçen, tarımın yapıldığı alanlarda etkin olarak rol oynayan parametrelerin her biri (toprak, bitki, su, gübre vs) analitik olarak değerlendirilmekte ve çözümler üretilmeye çalışılmaktadır. Başlıca bazı devletlerin ve kurumlar tarafından çalışmalara örnek olması amacıyla allta bulunan çalışmaları inceleyebiliriz.

İlk örnek Singapur'dan:

15 Haziran 2020 tarihli yazıda bir başlık görülmektedir:

"Singapurlu çiftçiler; gıda güvenliğimizi artırmak için yüksek teknolojiye geçiyoruz."

Şekil 1.1: Akıllı tarımın uygulandığı alan [73]

Bu şekilde bir açıklama yapmalarının altındaki nedenlerden bazıları şu şekilde:

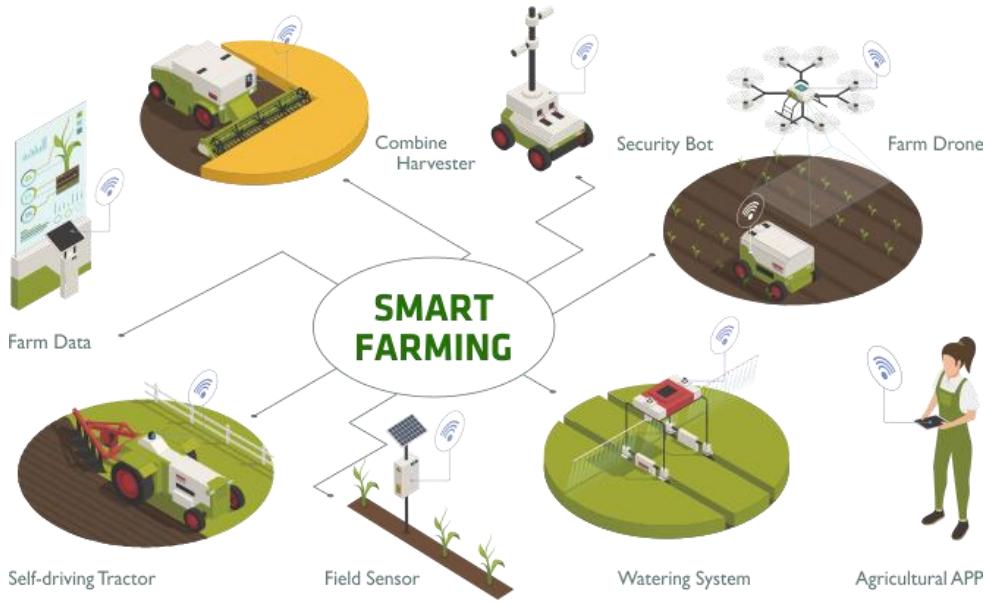
İlk neden olarak; tüm dünya çapında etkili olan, küresel iklim değişikliği nedeniyle üretimini yaptıkları gıdaların verimindeki %10 düşüşten bahsetmektedirler. Değişken hava koşullarının tahmin edilemez oluşu, üreticiler için kötü bir durum. Gıda üretiminin ne zaman düşüşe geçeği, hangi durumda stokların piyasa sürüleceği gibi parametreler bir ülkenin adeta bir zinciridir [4]. Eğer ki üreticiler, piyasaya belirsiz zamanlarda ürün çıkarırsa veya stokların dağıtımında yaşanacak lojistik gecikmelerin bu zincirin kopmasına neden olursa o ülke zarar etmeye başlar. Yaşadığımız dönem itibariyle Rusya-Ukrayna savaşının neden olduğu buğday gibi tahıl ürünlerinin üretimi ve dağıtımı hem ülkemiz için hem de diğer ülkeler için ciddi derecede sorun haline gelmiştir. Bu durumun düzeltilmesi için gereken politik adımlar atılmış fakat çözüme kavuşması zaman almıştır. Singapurlu çiftçiler; bahsettikleri bu global sorunları aşmak adına modern tarımın getirdiği kendi kendine yetebilir çözümlerin uygulamalarını yapmaktadır.

İkinci neden olarak; var olan tarım arazilerinin artık üretim kapasitesinin yeterli gelmemesi. Bulundukları coğrafya itibariyle toprak alanın az, insan nüfusunun fazla olması üretimin yetersizliğini beraberinde getirmektedir [5]. Üretim yetersiz kalması, dışarıdan gıda alımına yol açtığı için şu anda kullandıkları arazileri daha verimli hale getirmek istiyorlar. Doğal olarak ilk düşündükleri çözüm, dikey tarımın yapılması. Dikey olarak ürünlerin yetiştirilip kontrol edilmesi fiziki zorluklara yol açacağı için modern tarıma ve akıllı tarıma geçmeyi hedeflemektedirler. Akıllı tarım uygulamaları ile gıdaların; üretimi, bakımı, toplanması ve dağıtımı daha kolay hale geleceğini söylüyorlar.

En nihayetinde sorunlar ne kadar fazla olsa da asıl odaklandıkları şey; kendi kendilerine nasıl daha fazla yetebiliriz sorusunun cevabı olan modern ve akıllı tarım ile çözümler üretiyor olmalarıdır.

Singapurlu yerel bir üretici olan Deniel Wong'un şu sözleri aslında anlatmak istediklerimizi özetlemektedir. "*Modern agriculture is 95% science.*" Modern tarım; %95 bilimdir.

İkinci örnek mevcut durumun kıyaslamasının yapıldığı makalede bulunur [5]. Genel olarak makale içerisinde, Singapur'da tespit edilen sorunlara benzer sorunların olduğu görünür. Bu sorunlara ek olarak su kalitesi, küresel pandemik süreçleri vs söyleyenebilir. Fakat bu makalede sorunlar üzerinde değil de çözümler üzerinde durulduğu için o kısımlara odaklanmak gerekmektedir. Üretilen çözümler, doğrudan yazılım ve teknoloji tabanlıdır. Şekil 1.2'de görüldüğü üzere otonom ve insansız araçlar ile kontrol yazılımlarının entegre şekilde çalıştığı anlaşılabilir.



Şekil 1.2: Akıllı tarım sistemleri örneği [75]

Yukarıda bulun görselde sistemin nasıl çalıştığına kabaca bakacak olursak; tarım alanı farklı donanımlar ile entegre edilerek kontrol edilebilir hale getirilmiştir. Donanımlar fiziksel olarak veri elde etme ve müdahale etme amacıyla kullanılmıştır ve donanımlar, tasarlanan yazılım ürünü ile haberleşerek bitki, hava, su bilgisi gibi verileri göndermektedir. Yazılım, gelen verileri üreticilere anlaşılabilir halde sunmaktadır [4]. Üretici, program üzerinde bilgileri ve analizleri inceledikten sonra aksiyon alarak üretimi daha sağlıklı ve sürdürülebilir hale getirmiş olur.

Genel olarak durumu değerlendirmek gerekirse dünya genelinde insanın ihtiyacı olan gıda üretimi için farklı sorunlar ortaya çıkarken aynı zamanda farklı çözümler de üretilmektedir. Benzer durumlar ülkemizde de görülmektedir. Biz bu tez çalışmasını yaparak kendi kendimize yetebilir modern tarıma katkıda bulunmaya çalışılmıştır.

Gerçekleştirdiğimiz akıllı tarım uygulaması ile üreticilerin gündelik işlerini kolaylaştıracak ve üretimi sağlıklı takip edebilir hale gelmesini hedefledik.

Üreticiler; düzenli olmayan yaklaşımlarla, sürekli dışardan etkenlere maruz kaldığında verimin düşüşe geçtiğini bilmektedirler. Fakat önlem almak ve uygulama konularında zayıf kalabilmektedir. Onları olumlu yönde aksiyon almaları için geliştirdiğimiz uygulama ile yönlendirebilecekleri düşünülmektedir. Elbette geri dönüşler alınarak kullanım konusundaki kolaylıklar yeniden değerlendirilebilir.

Uygulamanın asıl amacı üreticinin geleneksel tarımdan modern tarıma yani akıllı tarıma adım atmasını sağlamak. Bu adımı sıradan yaklaşımlar ile değil de yapay zeka ile destekleyerek üreticin veya ilgili ziraat mühendisinin karar vermesinde, üretimdeki sorunları görmesinde, maliyetlerinin düşürülmesinde birer destek sistemi şeklinde çalışması beklenmektedir. Yapay zeka destek sistemlerinin birçok alanda olduğu gibi tarımsal alanda da kısa zaman içerisinde hayatımıza girmesi beklenmektedir. Elbette ki bu süreci kolaylaştıracak olan en büyük etken unutmamalıdır. Bu en büyük etken mobil cihazlardır. Mobil cihazların yaygınlaşması ve daha çok işlem yapabilir hale gelmesi bu tez çalışmasının temeline kadar etki etmektedir. Mobil cihazlar ile gündelik hayatta kontrol altına aldığımız şeylerin arasına tarımı da koymamak olmaz. Düzenli ve sağlıklı şekilde üreticinin; ürünü, araziyi, hava şartları gibi önemli faktörleri takip edebilmesi ve gerekli aksiyonları alması ile oldukça verimli sonuçlar elde edebileceği anlamına gelmektedir.

Akıllı tarım ve modern tarım alanları üzerine yapılan çalışmaların dünya genelinde ulaştığı noktalardan bazılarına değinilmiştir. Akademik ve sektörel anlamda bazı çalışmaların ortak problemleri incelendiğinde konu ile ilgili alt başlıklar Şekil 1.3 üzerinde görülmektedir.

Şekil 1.3: Akıllı tarım ve alt alanlarından bazıları [74]

Yapay zeka ve derin öğrenme yöntemlerinin akıllı tarımda uygulamasına birçok çalışma ve uygulama bulunmaktadır. Bunlar arasında; hastalık tespiti, bitki sınıflandırma, arazi yapısı tanımlama, zararlı böcek tespiti, toprak yönetimi ve akıllı sulama sıralanabilir [1].

1.1 Hastalık Tespiti

Yaşayan tüm canlıların ortak özellikleri arasında büyüme, gelişme ve ölüm süreçleridir. Bitkilerde de bu süreçlerin olduğunu biliyoruz. İlk çimlenmeden itibaren gelişim sürecideki bitkiler; farklı sebepelerden ötürü hastalıklara yakalanabilirler. Genel sebepler arasında; zararlı böcekler, bakteriler, toprak yapısı, diğer zararlı bitkiler gibi nedenler gösterilebilir. Bu durum bize hastalık sebepleri farklı ise tedavi yöntemleri de farklı olacaktır sonucunu çıkarmamızı sağlayacaktır. Ancak burada odaklanılması gereken kısım, hastalıkların tespiti olacaktır.

Çeşitli bitki türleri, hastalıklarını farklı şekilllerde bizlere göstermektedir. Bazı bitkiler üzerinde lekelenmeler gözlemlenirken bazılarının da meyverleri üzerinde lekelenmeler meydana gelir. Bu lekelenmlerin hangi sebepten meydana geldiğini ve tedavisinin ne olduğunu sıradan bir üretici kolaylıkla anlayamaz ve bunun için bilir kişiyi yani ziraat mühendisini çağırır. Ziraat mühendisi gözlemleri ve bilgisi birleştirerek karar verme sürecine geçer. Eğer ki yetkinliği varsa hastalığın sebebini ve

tedavisinin ne olacağını üreticiye anlatarak bilgilendirir. Sonrasında tedabi süreci başlar ve bitki iyileşinceye kadar devam eder. Makalede ve bizim çalışmamızda üzerinde sıklıkla durulan kısım bahsedilen bu insanlı süreci akıllı tarım adı altında yapay zeka ile yürütmektir. Yapay zeka insan tarafından belirlenen bitkiler ve hastalıkları hakkında toplanan görsel verilerden o bitkilere ait lekeli görsellerin hangi hastalıklara denk geldiğini öğrenir. Farklı metrikler ile testlerden geçen yapay zeka yüzdelik ihtimaller ile üreticiye hastalığın neler olabileceğini seçenekler halinde sunar. Üretici herhangi bir bilgi sahibi olmasa bile yetiştirdiği bitkinin hastalığını öğrenmiş olur. Ziraat mühendisini bilgilendirerek en son kararın yine bir insan tarafından verilmesi ile karar destek sistemini işletmiş olur.

Hastalık tespiti süreci farklı şekillerde de yapılabilir fakat en nihayetinde veriler görüntü tabanlı olmasa da problemin çözümü için bitkiler ve onlara ait hastalık isimlerine ihtiyaç olmaktadır.

Şekil 1.4: Hastalıklı bitki [56]

1.2 Bitki Sınıflandırma

Akıllı tarımın yapıldığı araziler veya çeşitli yapılar içerisinde farklı türde bitkilerin olması her biri için farklı bakım, sulama, ilaçlama gibi süreçleri beraberinde getirecektir. Görüntü tabanlı bitki sınıflandırma algoritmaları kullanılarak, büyük-küçük tüm arazilerin içerisinde yetişen bitkilerin sınıflandırmasını yapılabilir.

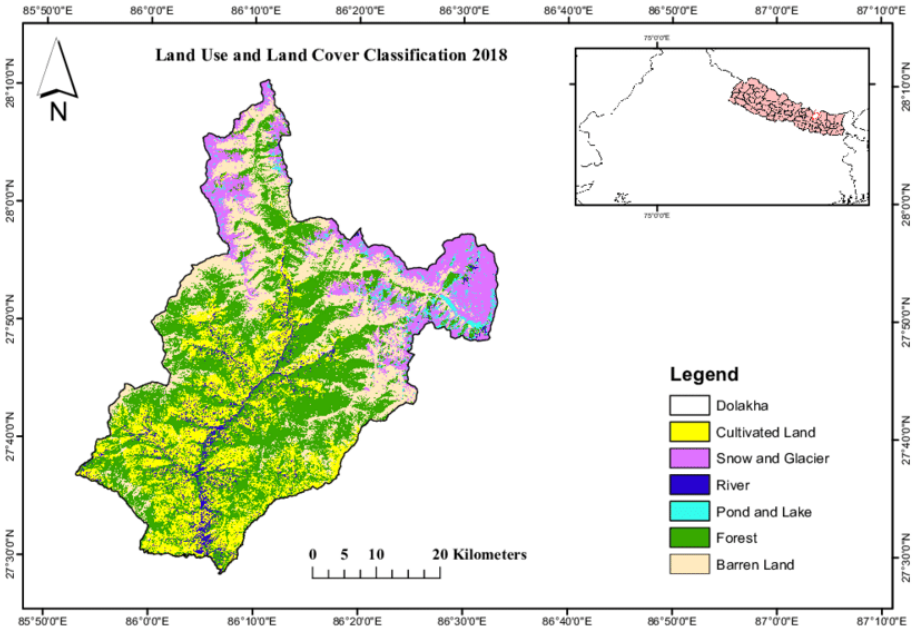
Otomasyon teknikleri ürünler arasında genelleştirilemediğinden, araştırmacılar ürüne özel sistemler geliştirmeye odaklanıldı. Damar morfolojisine dayalı olarak önerilen bitki tanımlaması. domates bitkileri için robotik tarıma dayalı otomatik verim tahmini önerildi [1].

Şekil 1.5: Yaprak görüntülerinden sınıflandırılmış bitkiler [61]

1.3 Arazi Yapısı Tanımlama

Görüntü tabanlı olarak düşünülün bu problem türü, üreticinin yetiştirdiği ürünlerin bulunduğu arazilerin ayrımının yapılabilmesi için düşünülmüştür. Uydu görüntüleri veriler halinde yapay zeka modellerine verilerek arazi türlerini tespit etmek için kullabilmektedir. Farklı arazi türleri; su yolları, otoyollar, ekim yapılan araziler, hasat edilmiş ekimi bekleyen araziler olarak çeşitli şekillerdedir. Uydu görüntülerinden ayrıştırılan arazi türleri var olan üreticiler ve yeni üreticillik yapmak isteyen kişiler için oldukça faydalı bilgileri barındırabilmektedir.

Uydu verileri, arazi örtüsü sınıflandırması için tek veri kaynağı değildir. IoT tabanlı teknolojilerin gelişmesiyle birlikte insansız hava araçları (İHA), mahsul izleme için etkili bir araç haline geldi [1].



Şekil 1.6: Uydu görüntüsü üzerinde ayrıştırılmış alanlar [57]

1.4 Zararlı Böcek Tanıma

Akıllı tarıma katkısı olacak olan diğer konu, bitkilerin gelişim sürecine olumsuz yönde etki eden bir durum olan zararlı böceklerin derin öğrenme yöntemi ile tespit edilerek gerekli ilaçlamalar yapılarak ve diğer önlemlerin alınmasında önemli rol oynamaktadır.

Haşere tanıma sadece nesne tanıma değildir; özel bir şekilde ele alınması gereken daha karmaşık bir görevdir. Cheng ve ark. [76], karmaşık bir arka planda derin artık öğrenme yoluyla haşere tanımlaması gerçekleştirildi. Ding ve Taylor [77] ve Zhu ve ark. [78], güve görüntülerinin sınıflandırılması için derin öğrenme tekniklerini kullanıldı [1].

Şekil 1.7: Zararlı böcek temsili gösterim [58]

1.5 Akıllı Sulama

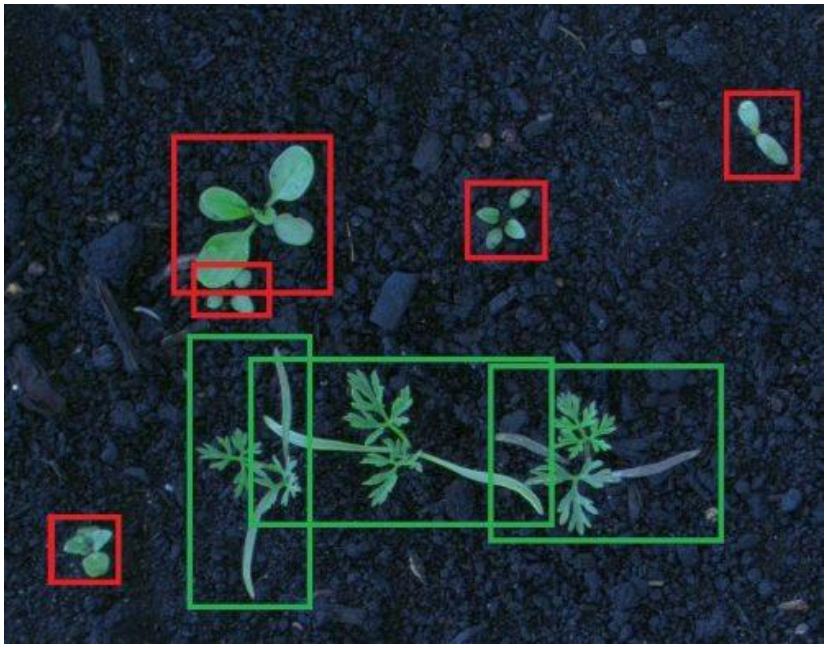
Üreticilerin ulaşmakta ve dağıtımında zorlandırkları bir durum olan sulama, olabildiğince verimli şekilde yapılmalıdır. Sulamadan verim alınabilmesi için boşa harcama yapılmaması gereklidir. Üretici kaynağında bulunan suyu hagi bölgelerde daha çok kullanmalı ya da daha az kullanmalı sorusunun cevabını aramaktadır. Görüntü tabanlı veriler ile arazide bulunan bitkilerin susuz kalan kısımlarını yapay zeka ile tespit ederek doğru sulama yapması sağlanmalıdır.

Sulama sürecinin verimli yönetimi için bitkilerin su durumunun tespit edilmesi önemlidir. AlZu'bi ve ark. [79], IoT sensörlerinin akıllı sulama kararları vermek için makine öğrenimi yöntemleriyle çalıştığı görüntü işleme konseptleri önerdi. Şarkı ve ark. [80], sulanmış bir mısır tarlasında toprak nemi içeriğini tahmin etmek için derin inanç ağını makroskopik hücresel otomata (MCA) yaklaşımıyla birleştiren yeni bir model önerdi [1].

Şekil 1.8: Akıllı sulama temsili gösterim [59]

1.6 Zararlı Bitki Tanıma

Bitkilerin gelişiminde olumsuz yönde etki eden zararlı böcekler olduğu kadar zarar bitkiler de mevcuttur. Bu bitkilerin; görüntü tabanlı verilerden bitkilerin ne olduğu tespit edildiği gibi yine görüntü tabanlı verilerden zararlı bitkilerin de yapay zeka modelleri ile sınıflandırması yapılarak bulunması gerekmektedir. Zararlı bitkiler toprak içerisinde bitkinin gerekli besinleri alabilmesi için veriler maddelerden faydalarak gelişim göstermektedir. Düzenli periyotlar ile üretici kendi yetiştirdiği bitkilerden farklı olan zararlı bitkileri bulup temizlemesini yapmalıdır.



Şekil 1.9: Zararlı bitkilerin görsel veriler ile tespit edilmesi [60]

Genel hatlarıyla bizim çalışmamıza büyük ölçüde etki etmiş olan "Smart farming becomes even smarter with deep learning" makalesi akıllı tarımın yapay zeka ile daha etkin bir biçimde yapılabileceğini bizlere göstermektedir.

Diğer bir çalışma olan "Deep Learning for smart agriculture" yukarda bahsedilen birçok alt problem ve çözümler için (hastalık tespiti, sulama, zararlılar tespiti vs) ortak kümeye sahiptir. Ayrışan durumları aşağıda görebiliriz.

1.7 Toprak Yönetimi

Üretici tarafından yetiştirilen bitkilere ev sahipliği yapan toprak, verimli şekilde kullanılmalıdır. Daha sağlıklı ve fazla ürün almak amacıyla yapılan yanlış yöntemler nedeniyle toprak yapıları zarar görmektedir. Gereksiz ilaçlama veya gübrelemeler yapılarak topraktaki yararlı bakteriler gibi birçok canlı öldürülmektedir. Tek sefer için fazla ürün almak amacıyla yapılan hatalar çoğunlukla uzun vadeli zararlar vererek geri dönülemez ya da çok zor telafi edilen durumlara götürmektedir. Topraklar kontrol altında tutularak gerekli analizler ve raporlamalar yapılarak yapay zeka ile desteklenmelidir.

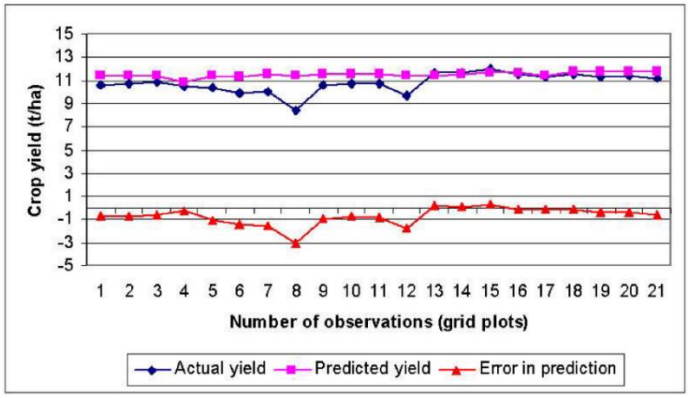
Şekil 1.10: Toprak yönetimi temsili gösterim [61]

Makine öğrenimi tekniklerini kullanarak, uygun ekinleri ve toprak için en iyi gübre türünü kolayca tanınabilir. Gauss çekirdek tabanlı SVM (Destek Vektör Makineleri), Bayes Ağları ve YSA çoğunlukla daha iyi toprak yönetimi için kullanılır. Toprağın kalitesi, seçilen mahsuldeki pH düzeyine, EC'ye, birincil, mikro ve makro besinlere bağlıdır, bu nedenle pH'ın sınıflandırılması ve ilgili toprak besin indeksleri toprağın sağlığını incelemeye yardımcı olur [2].

1.8 Verim Kestirimi

Yıllık üretimden elde edilen hasılat bilgileri ile istenilen analizler yapılabilmektedir. Kullanılan gübre, ilaç, su ve diğer maddelerin değerleri kayıt altında tutularak bir sonraki seneler için alt yapı hazırlanabilir. Hava şartları veya beklenmedik olayların görülmesi bu durumun dışında tutularak makine öğrenmesi modelleri ile bazı

tahminler yapılabilir. ‘Random Forest Algorithm’ (Rastgele Orman Algoritması) ile mısır mahsülü verim kestirimi yapılarak %99.74 başarı puanına ulaşılmıştır. Verim tahmininde; ‘Random Forest Algorithm’ diğer algoritmalara göre kıyaslandığında daha çok başarım (doğru sınıflandırılmış kayıtların toplam kayıt sayısına oranı: ACC = CR/TR) [92] göstermiştir. Gelecekte çifçilerin kayıp oranlarını hesaplamasında yardımcı olacağı da düşünülmektedir.



Şekil 1.11: Ürün veriminin zaman içerisinde tahmin edilmesi [62]

Her mahsulün verim tahminine konsantre olunması gereklidir. Mahsul verimi tahmininde makine öğrenmesi ve derin öğrenme algoritmaları uygulanmıştır. Çiftçilerin neyi ne zaman ekeceklerine karar vermelerine yardımcı olmaktadır [2].

Bölüm 2

Literatür Taraması

Senthil Kumar Swami Durai ve ark. çalışmasında [81] akıllı tarım için makine öğrenmesi ve derin öğrenme tekniklerini kullanarak çiftçilerin ihtiyacı olacak konularda veriler toplayarak çeşitli sorunlara çözümler üretmenin yollarını aramışlardır. Çiftçilerin ürettiği çok büyük verileri; veri madenciliği, veri analizi, makine öğrenmesi ve IOT yöntemleri değerlendirerek daha az emekle daha çok verimli ve akıllı yöntemler üzerinde denemeler yapmışlardır.

Achim Walter ve ark. çalışmasında [16] sürdürülebilir ve geliştirilebilir akıllı tarım yöntemleri üzerinde durulmuştur. Akıllı tarım faaliyetleri için karşılabiliecek zorluklar da değerlendirilmiştir. Teknoloji, enstitüler, çeşitlilik ve ağlar bir araya getirilerek çifçilerden marketlere kadar tüm süreçte kullanılabilecek dağıtım ağının mümkün olduğundan bahsetmiştir. Otonom araçlar, hava araçları, sensörler ve teknik yeterlilikler kullanılarak üreticilerin akıllı tarım yapabilmeleri bir fırsat olarak görülmektedir.

Jash Doshi ve ark. çalışmasında [15] IOT teknolojileri kullanılarak çifçiler için; hava durumumdan, toprak yapısına kadar sensörler vasıtasıyla düzenli olarak takip edebilecekleri sistemi anlatmaktadır. Oluşturdukları sensörlü prototip cihaz ile çiftçilerin yeterli teknik bilgileri olmasa bile rahatlıkla kullanabileceklerdir. Prototip cihaz ile gönderdikleri verileri mobil uygulamadaki ekranlar ile izleyebilmektedirler.

Elsayed Said Mohammed ve ark. çalışmasında [18] akıllı tarım için zirai yönetimin ne şekillerde yapılabileceğinden bahsetmiştir. IOT cihazlar kullanılarak verilerin yakında gelecek olan 5G teknolojisi ile bulut sistemlerine gönderilerek yönetilebilmelidir. Verilerin akıllıca analizleri ve iyi planlamalar ile zirai faaliyetlerde önemli değişiklikler elde edilecektir. Arazilerin çok büyük olmasına gerek olmadan ter türlü

arazide çalışması hedeflenmektedir. Belirli aralıklar ile yerleştirilen IOT cihazlarından yüksek doğruluğa sahip kaliteli verilerin toplanması oldukça önemlidir.

Sharada P. Mohanty ve ark. çalışmasında [12] bitki yapraklarındaki hastalıkların tespit etmeye çalışmışlardır. 256 x 256 piksel türündeki renkli ve siyah beyaz resimlerden 54.306 tane görüntü toplayarak 38 farklı sınıfa ayırıp verisetlerini oluşturmuşlardır. Aynı zamanda PlantVillage verisetini de kullanmışlardır. Aktarımlı öğrenme yöntemi ile AlexNet ve GoogLeNet mimarilerini eğitilmiştir. Verisetlerini farklı yüzdelik oranlar ile bölerek eğitimden geçirdikleri modellerin kıyaslamaları yapılmıştır. Mimarilerin çapraz verisetleri ile eğitiminden 0.9934 değerine sahip olan GoogLeNet'tir. Renkli fotoğraflar çoğunlukla her iki model için de yüksek başarım [92] getirmiştir.

Mahmudul Hassan ve ark. çalışmasında [8] aktarımlı öğrenme yöntemi ile evrişimli sinir ağlarını kullanarak bitki yapraklarında hastalık tahmin etmeye çalışmışlardır. InceptionV3 , InceptionResNetV2, MobileNetV2 ve EfficientNetB0 modellerini PlantVillage veriseti ile eğiterek başarım [92] sonuçları karşılaştırılmıştır. Çapraz bölütleme ile veriler farklı oranlarda bölünmüştür. Elde edilen sonuçlar arasında %99.69 başarım [92] oranı ile MobileNetV2 olmuştur.

Biao Wang ve ark. çalışmasında [5] uydu görüntüleri üzerinden arazi değişim tespitini hedeflemişlerdir. Veriseti olarak asya bölgesine ait IKONOS-2 ve WorldVıew-3 kullanılmıştır. 400 x 400 piksel şeklinde verileri kullaranak geliştirilen metodolar ile değişim bölgeleri tespit edilmiştir.

V. S. Martins ve ark. çalışmasında [82] derin öğrenme içerisinde aktarımlı öğrenme yöntemlerini kullarak yüksek çözünürlüklü uydu görüntülerinden yanan arazileri haritalandırmışlardır. Veriseti olarak Landsat-8 OLI, PlanetScope-0, ve PlanetScope-1 kullanılmıştır. U-Net mimarisi verisetleri ile eğitildikten sonra afrika kıtası üzerinde farklıo bölgelerde denemeler yapılmıştır. Bölgesel olarak kuzey afrikada başarım [92] oranı yüksek görülmüştür.

Pail Oskar Gislason ve ark. çalışmasında [83] makine öğrenmesi yöntemi olan rastgele orman algoritması ile arazileri sınıflandırmışlardır. Landsat uydu verileri ile yapılan testlerde Colorado eyaletinde su, toprak, arazi, dağ gibi10 farklı zemin türü ayrıştırılımıştır.

Minghu Wu ve ark. çalışmasında [84] MaskR-CNN mimarisini kullanrak nesne tespiti yapılmıştır. 1300 x 800 piksek formatında COCO verisetini kullanarak 118.287 eğitim, 5000 doğrulama ve 40.670 görseli model eğitim için kullanmışlardır. Aynı zamanda PASCAL VOC07 veriseti de kullanılmıştır. ResNet mimarisinin türevleri modelin kemik kısmında çaprazlanarak F1 değerleri alınmıştır. COCO veriseti ile sınır kutulama yöntemi kullarak eğitilen MASKR-CNN modeli, örnek segmantasyonu yapılan modele göre yüksek başarım [92] elde etmiştir.

Yongliang Qiao ve ark. çalışmasında [85] hayvancılık alanında, büyük baş hayvanların MASKR-CNN mimarisi görsel üzerinden ayrıştırılması yapılmıştır. Hayvanların düzenli olarak takip edilebilmesi, hareketlerinin analiz edilmesi, sağlık durumu, kilo kontrolü gibi önemli konular üzerinde durulmuştur. Modellerin eğitimlerinde kullanılan verisetleri, yerleştirilen kamera ile 1920 x 1080 çözünürlüğünde görüntüler akratarak toplanmıştır. Geliştirilen görüntü 3 adet segmantasyonu modeli arasında MASKR-CNN modeli 0.92 başarım [92] puanı ile en yüksek puanı almıştır.

Melissande Machefer ve ark. çalışmasında [86] insansız hava araçlarından çekilen arazi fotoğrafların içerisinden bitkilerin sayısını, MASKR-CNN mimarisi ile belirlemiştir. Verisetini toplamak için düzenli olarak arazi üzerinde 50m yükseklikten uçarak kayıtlar alınmıştır. Ardından COCO veriseti ve toplanan diğer çeşitli sayıda veriler ile eğitim yapılmıştır. 10 x 10 m arazi üzerinden alınan görütülerin üzerinde bitkiler olukça etkili şekilde ayrıtırılmıştır. 0.997 precision değerine sahip model ile tespit edilen bitkiler, yüksek doğruluk oranı ile adetleri sayılmıştır.

Bölüm 3

Metodoloji

Tüm çalışma boyunca; geliştirilen ve kullanılan teknikler, araçlar, mimariler ve yapısal tasarımlar bu bölüm kapsamında alt başlıklar halinde incelenecektir.

3.1 Yapay Zeka

Yapay zeka; insan zekasını, düşünce tarzını taklit ederek benzer sonuçları alabilmek adına geliştirilmiş matematiksel yöntemler bütünüdür.

İkinci dünya savaşının getirdiği yıkıcı etkiyi son vermek adına İngiliz matematikçiler tarafından, şifreleri deşifre etmek amacıyla basit düzeyde bilgisayarlar geliştirilmiştir. İlk işlevleri temel fonksiyonları yapmak olan bu bilgisayarlar daha sonra, savaşa son veren İngiliz matematikçilerin arasında bulunan Alan Turing adlı kişi tarafından farklı amaçlara yöneltilemeye başlandı. Geliştirdiği Turing Makinesi (Turing Machine) ile insan-bilgisayar etkileşimini (human-computer interaction) ortaya atarak deneyler yaptı. Yaptığı bu deneylerde; programların, insanlar gibi düşünerek karar vermesini görmek ve karşısındakinin insan mı program mı olduğunu anlamayı hedefledi [13]. Bu süreç artık insan zekasının bir takliti ve öğrenicisi görülerek yapay zeka adı altında literatüre kazandırıldı. Günümüze kadar da devam eden çalışmalar ile artık daha başarılı sonuçlara imzalar atılmaktadır. Donanımsal geliştirmeler ile işlem yapabilme kapasitesi artırılan bilgisayarlar sayesinde yapay zekalar insan gibi karar verebilme yetilerini her geçen gün daha da artırmaktadır. Sanattan teknolojiye, oyundan eğitime kadar birçok alanda karşımızda birer ‘bot’ adı verilen programları görmekteyiz. Bu programlar sayesinde hayatımızda kararlar alırken veya yeni bir şey öğrenirken çok fazla zorluk yaşamıyoruz. Hayatımıza oldukça hızlı bir şekilde girmekteler ve biz de oldukça hızlı adapte oluyoruz.

Yapay zeka oldukça geniş kapsama sahip olduğu için genel tanımların ardında getirdiği alt alanları da incelemeliyiz.

3.2 Yapay Zeka Türleri

Genel olarak iki gruba ayırarak ele alırsak:

Yapay Dar Zeka (ANI) olarak da bilinen zayıf yapay zeka, belirli görevleri gerçekleştirmek için eğitilen bir yapay zeka türüdür [34]. Zayıf yapay zeka, şu anda etrafımızdaki yapay zekanın büyük bir kısmını yönlendiriyor. Zayıf olmaktan tamamen uzak olduğundan, bu yapay zekayı 'dar' olarak daha doğru bir şekilde tanımlayabiliriz; bu yapay zeka, Apple'dan Siri, Amazon'dan Alexa, IBM Watson ve otonom araçlar gibi oldukça sağlam uygulamalara olanak sağlıyor.

Güçlü yapay zeka, Yapay Genel Zeka (AGI) ve Yapay Süper Zekadan (ASI) oluşur. Yapay genel zeka (AGI) ya da genel yapay zeka, bir makinenin insanlarla eşit zekaya sahip olduğu teorik bir yapay zeka şeklidir; sorun çözen, öğrenen ve gelecek planları yapan, kendinin farkında bir bilince sahiptir. Yapay Süper Zeka (ASI) -aynı zamanda süper zeka olarak da bilinir- insan beyninin zekasını ve sınırlarını aşan bir zeka türüdür. Güçlü yapay zeka günümüzde herhangi bir pratik örnekten yoksun bir şekilde hala teori aşamasında olsa da, bu, yapay zeka araştırmacılarının güçlü yapay zekayı geliştirmeyi denemediği anlamına gelmiyor. Bu arada, yapay süper zekanın en iyi örnekleri bilimkurguda görülebilir; tıpkı 2001: A Space Odyssey'deki insanüstü, başınabuyruk bilgisayar asistanı HAL gibi [34].

3.2.1 Yapay Dar Zeka

Dar yapay zeka, insanlığın yapay zeka geliştirme konusunda şu ana dek ulaşabildiği en uç noktadır. Otonom araçlar ve kişisel dijital yardımcılar gibi gerçek dünyada gördüğünüz tüm yapay zeka örnekleri bu kategoriye girer. Bunun nedeni, yapay zekanın gerçek zamanlı olarak kendi başına düşünüyormuş gibi göründüğü durumlarda bile aslında dar kapsamlı çeşitli süreçli koordine ediyor ve önceden belirlenmiş bir çerçeve dahilinde kararlar alıyor olmasıdır. Yapay zekanın “düşünme sürecinde” bilinç veya duygular yer almamaktadır [33].

3.2.2 Yapay Genel Zeka

Genel yapay zekaya sahip bir bilgisayar sistemi teorik olarak, karmaşık sorunları çözebilir, belirsiz durumlarda karar alabilir ve mevcut durumu değerlendirirken önceki bilgilerinden yararlanabilir. Böyle bir sistem; yaratıcılık ve hayal gücü açısından insanlarla aynı düzeyde olur ve dar yapay zekaya göre çok daha geniş bir görev yelpazesini başarıyla ele alabilir [33].

3.2.3 Yapay Süper Zeka

Yapay süper zeka düzeyine ulaşmış bir bilgisayar sistemi; bilimsel yaratıcılık, genel bilgelik ve sosyal beceriler dahil olmak üzere hemen her alanda insanlardan daha başarılı olacaktır [33].

3.2.4 Makine Öğrenmesi

Makine öğrenmesi, bilgisayar sistemlerinin yapay zeka elde etmek için uyguladığı bir süreçtir. Makine öğrenmesi, verilerdeki kalıpları belirlemek için algoritmaları kullanır. Ardından, bu kalıplar kullanılarak tahmin yapabilen bir veri modeli oluşturulur.

Makine öğrenmesi modelleri, veri alt kümeleri üzerinde eğitilir. Modeli eğitmek için kullanılan veriler analiz edilecek veri kümesinin tamamını doğru şekilde temsil ediyorsa algoritmanın hesapladığı sonuçlar daha doğru olur. Makine öğrenmesi modeli, görevini faydalı ve güvenilir olacak kadar hızlı ve doğru bir şekilde yerine getirecek ölçüde iyi eğitildiğinde dar yapay zeka elde edilmiş olur [33].

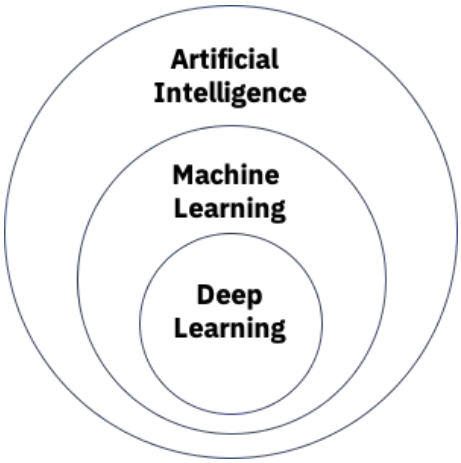
3.2.5 Derin Öğrenme

Derin öğrenme, beynin yapısından (sinir ağları olarak bilinir) ilham alınarak oluşturulmuş algoritma ağlarının kullanıldığı gelişmiş bir makine öğrenmesi türüdür.

Derin sinir ağlarında iç içe yerleştirilmiş nöral düğümler bulunur ve yanıtlanan her soru ilgili bir dizi başka soruya yol açar.

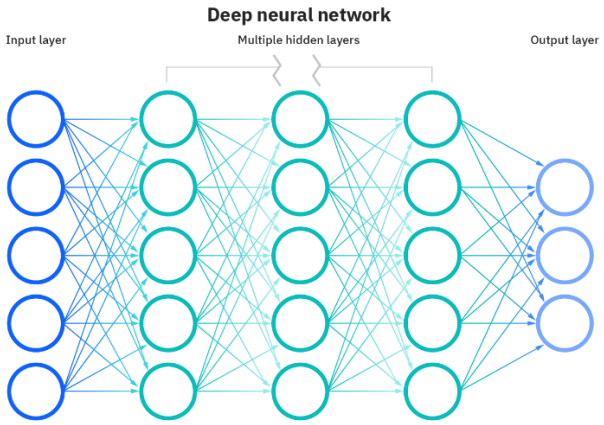
Derin öğrenme eğitimi için genellikle büyük bir veri kümesi gerekir. Derin öğrenmeye yönelik eğitim kümeleri bazen milyonlarca veri noktasından oluşur. Bu büyük veri kümeleri üzerinde eğitilen derin sinir ağları, belirsizlik içeren durumlarda basit ağlara göre daha iyi çalışabilir. Bu nedenle derin sinir ağları, görüntü tanıma gibi uygulamalar için kullanışlıdır. Görüntü tanıma uygulamasında yapay zekanın, görüntüde neler olduğunu belirleyebilmek için önce şekillerin kenarlarını bulması gerekir. Ayrıca, satranç gibi karmaşık oyunlarda insan becerilerinin ötesine geçen yapay zeka sistemleri de derin öğrenme tarafından eğitilir [33].

Derin öğrenmenin daha anlaşılır olması için Şekil 3.1 incelenebilir.



Şekil 3.1: Yapay zeka ve alt alanları [34]

Derin öğrenme, aslında nöral ağlardan oluştuğunu söylemiştik. Derin öğrenmedeki "derin" ifadesi, üçten fazla katmandan oluşan (girişler ve çıkışlar dahil) ve bir derin öğrenme algoritması olarak görülebilecek nöral bir ağı ifade eder. Bu, genellikle aşağıdaki şema kullanılarak temsil edilir [33].

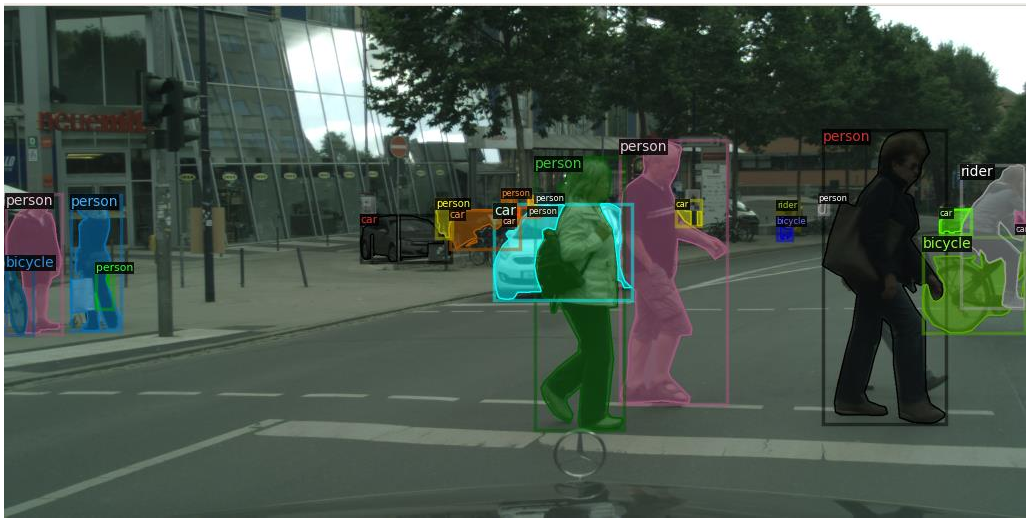


Şekil 3.2: Yapay sinir ağı [34]

Derin öğrenmenin ve yapay sinir ağlarını incelediğimize göre bizim çalışmamızın temelini oluşturan görüntü tabanlı problemlerin çözümünde ifade ettiğimiz diğer bir kısma bakalım.

3.3 Bilgisayarlı Görü

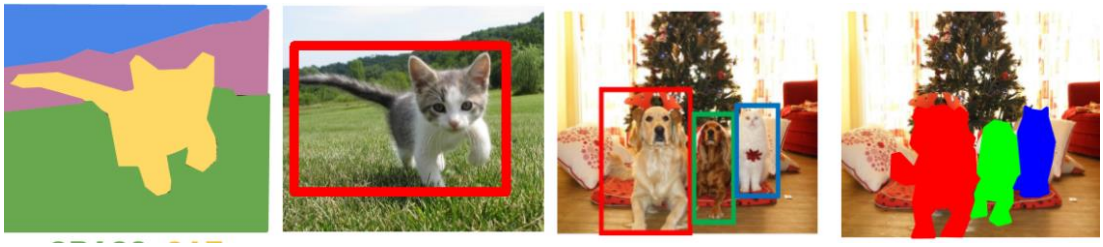
Bilgisayarlı görü (computer vision – CV), makinelerin dijital görüntüleri görmesini ve içeriklerini anlamasını sağlayan teknikleri geliştiren bir yapay zeka disiplinidir. Bilgisayarlı görü, makineleri çevrelerindeki dünyayı görmeleri ve anlamaları için eğitir. Bu yapay zeka disiplini sürücüsüz otomobiller, yüz tanıma ve artırılmış gerçeklik dahil olmak üzere birçok teknoloji alanındaki yenilikler için kritik öneme sahiptir. Bilgisayarlı görü, üretilen dijital görsel verilerin miktarının hızla artması nedeniyle günümüzün en önemli AI disiplinlerinden birisine dönüşmüştür. Görsel verilerin artması, aynı zamanda bilgisayarlı görü algoritmalarının eğitilmesini de kolaylaştırıyor. Çalışma mantığı olarak da küçükten büyüğe mantığıyla çalışan bir görsel işleme tekniği kullanır. Önce pikseller ve renkler gibi basit, sonrasındaysa çizgiler ve nesneler gibi daha karmaşık özellikleri tespit ve analiz eder [35].



Şekil 3.3: Fotoğraf üzerinde tespit edilip işaretlenen insanlar ve nesneler [68]

Birçok görsel tipteki probleme çözüm olarak sunmak oldukça etkilidir. Derin öğrenme algoritmalarının sinir ağlarına input olarak verilecek görsellerin anlam kazanarak çıkması beklenir. Algoritma, görseller üzerinde kalıpsal şeyleri farkederek adeta bir harita çıkarır. Bu harita üzerinde işaretlemeleri yaparak benzer haritaya sahip olan nesnelerin kalıplarının da ne anlama geldiğini bulmuş olur. Elbetteki görselin kalitesi ve sayısı bu süreçte etkili parametreler olarak rol oynacaktır.

Şekil 3.3 numaralı örnekte gösterildiği gibi bilgisayarlı görü; akıllı mobeseler, otonom araçlar, güvenlik sistemleri, tıbbi görüntüleme, oyunlar ve akıllı tarım problemleri gibi görsel veriler ile çözüme kavuşabilecek problemler için oldukça ideal olduğu söylenebilir.



Şekil 3.4: Görsel veri üzerinden hayvanların bulunması [63]

3.4 Problemler ve Uygulanan Metotlar

Derin öğrenme algoritmaları, sinir ağları ile problemin çözümüne farklı şekillerde yaklaşabilir. Bu yaklaşımlar problemin türüne ve çözümüne göre değişkenlik

gösterebilir. Sinir ağları gelen verinin hangi türde olduğu ile ilgilenmez. Ses, resim, sensör verileri, sayısal veriler şeklinde farklı veri türleri problemlerin çözümüne değişkenlik gösterir. Eğer ki çevre fotoğrafı çekilmiş bir görsel üzerinde çeşitli nesneler, hayvanlar veya insanlar bulunmak istenirse derin öğrenme algoritmalarının sinir ağları görsel veriler ile eğitilip test edilmelidir. Doğal olarak problem görüntü problemine dönüşmüş olur. Fakat ses verileri kullanılarak sinir ağlarını eğitmek, görsel problem için yanlış bir yaklaşım olacağı için sonuçların da doğru olması beklenemez.

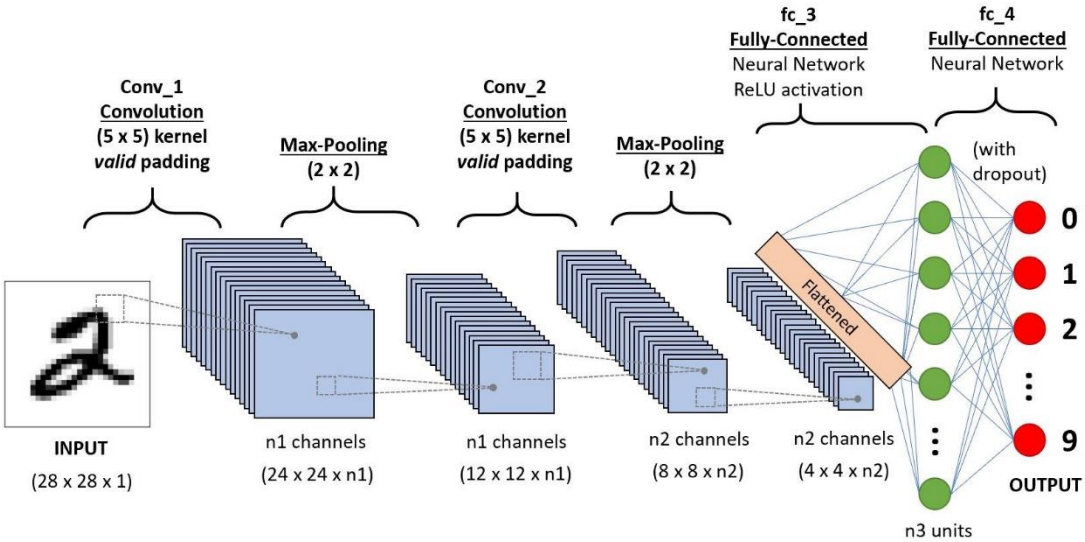
Akıllı tarımın yapılabilmesi için önceki bölümlerde bazı problemlerden ve çözümlerinin derin öğrenme yöntemleri ile olabileceğinden bahsedilmişti. Çalışmamızda görsel tabanlı problemler üzerinde yoğun şekilde durulmuştur. Başlıca saymak gerekirse; bitki yaprakları üzerinden hastalık tespiti, bitkinin büyümesini takip etmek için düzenli görüntülerin alınması ve takip edilmesi şeklinde bahsedebiliriz. Ancak uygulanan ve geliştirilen çözümlerin daha iyi anlaşılabilmesi için önce görsel tabanlı sinir ağlarını incelemekte fayda var. Daha sonra problemler ve çözümleri özelinde devam edeceğiz. Görsel tabanlı sinir ağlarının yapısını alt kısımda detaylı olarak görebiliriz.

3.4.1 Konvolüsyonel (Evirişimsel) Sinir Ağları (CNN)

3.4.1.1 Evrişimsel Sinir Ağları

Bir evrişimsel sinir ağı (ConvNet / Convolutional neural networks - CNN), bir girdi görüntüsünü alıp, görüntüdeki çeşitli görünüşleri/nesneleri birbirinden ayırabilen derin öğrenme algoritmasıdır. Evrişimli sinir ağları, temel olarak görüntüleri sınıflandırmak (örneğin gördüklerini isimlendirmek), benzerlikle kümelemek (fotoğraf arama) ve sahnelerde nesne tanıma yapmak için kullanılan derin yapay sinir ağlarıdır [37].

Çalışma yapısına geçmeden önce aşağıdaki resimde görüldüğü üzere sıradan bir sinir ağının gösteriminden farklı olduğuna dikkat etmek gerekir. Normal sinir ağlarını da içerisinde barındırmaktadır fakat evrişimsel sinir ağı (CNN) mimarisinin kendine özel getirmiş olduğu katmanlar da bulunmaktadır. Birçok türde mimari için de benzer türde farklılıkların olabileceği dikkate alınmalıdır.



Şekil 3.5: Evrişimsel sinir ağının yapısı [64]

3.4.1.2 Evrişimsel Sinir Ağları Çalışma Prensibi

Çalışmanın bu kısmında; evrişimli sinir ağlarını bitki yaprağı görüntüsü verisetleri ile eğiterek, bitki yaprakları görselleri üzerinden hastalık tespiti yapılmıştır.

Evrişimsel sinir ağlarının farklı problemler ile kullanılabilir. Fakat bizim çalışmamızda odaklandığımız problemler görsel veri odaklı olduğu için çalışma şekli ve diğer anlatımlar da görsel veriler ile devam edilecektir [49].

Evrişimsel sinir ağları ana tipte katmana sahiptir ve yapısı gereği sürekli ileri doğru akan bir algoritma barındırır. Sırasıyla katmanlar; evrişimsel katman (convolutional layer), havuzlama katmanı (pooling layer) ve tamamıyla bağlı (fully-connected) katmanı şeklindedir [49].

Evirişimsel katman, yapının ilk katmanı olduğu için veri ile ilk karşılaşan katmandır. Görsel verinin bu katmanda hesaba girebilmesi için önce matris formatında olması gerekir. Matris şeklinde evrişimsel katmanda filtre adı verilen matrisler kullanılarak çarpım işlemi yapılır. Filtre görselin üzerinde sıra sıra kaydırılarak çarpılmaya devam edilir ve sonuç olarak filtreden geçen görüntümüz yeni formatta bir matris haline gelir. Diğer bir katman olan havuzlama katmanına geçer. Burada ise girdideki fazla parametreler azaltılarak boyutsal indirgeme sağlar. Ardından tamamıyla bağlı katmana geçerek 0 ile 1 arasında bir sınıflandırma işlemine dönüşür. Elbette tüm bu durum

göründüğü kadar sade bir yapıda değildir. Bazı fonksiyonlar ve parametreler ile desteklenerek çeşitli sonuçlar elde edilmektedir.

3.4.1.3 Evrişimsel Sinir Ağı Türleri

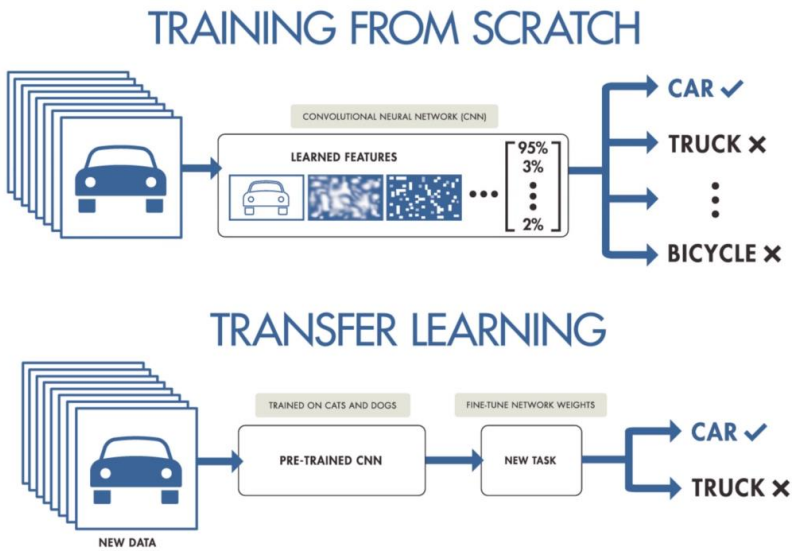
Evrişimli sinir ağları temel yapı olarak günümüz problemlerine yeterli gelmeyebiliyor. Daha etkin sonuçlar almak adına performans değeri yüksek evrişimli sinir ağı modelleri üretilmiştir. Bu modeller içerdikleri katman sayılarının yüksek olması ve işlem kapasitesi yüksek donanımlar ile eğitildiği için birçok benzer problem için hazır altyapı sağlamaktadırlar. Aşağıda bazılarını görebiliriz.

- AlexNet [29]
- VGGNet [32]
- DenseNet [30]
- ResNet [31]
- EfficientNet [33] (En güçlü CNN mimarisi olarak da geçer)

3.4.2 Aktarımlı Öğrenme (Transfer Learning)

Aktarımlı öğrenme; yapay sinir ağlarının tekrar kullanılabilmesi, daha başarılı sonuçlar elde etmek için geliştirilmiş bir yöntemdir.

Yapay sinir ağları farklı problemler için farklı şekillerde tasarlanır ve amacı doğrultusunda kullanılıp geliştirilmeye devam eder. Örneğin metinsel bir problem üzerinde çalışılıyor ise geliştirilen model metinsel problemler için kullanılıp geliştirilmeye devam edilmelidir. Aynı şekilde bizim çalışmamızda ele aldığımız problem, görsel tabanlı bir problem olduğu için geliştirilen mimariler yani yapay sinir ağları görüntü verileri ile eğitilerek sonuçlar üzerinde analizler yapılmalıdır. Daha önceden farklı görüntü sınıflandırma problemleri için Google, Nvidia, Microsoft gibi firmaların derin öğrenme çalışmalarında kaynak olması veya geliştirme yapılan programlama dilinde içerisindeki kütüphaneler ile eğitilmiş modeller üzerinde çalışmalar yapabiliyoruz [65]. Çalışmamızda işlenen bitki yapraklarından elde edilen görüntülerin hangi hastalık grubuna ait olduğunu tahmin edebilmek için eğitilen bazı modelleri kendi kaynaklarımız ve verisetimiz ile tekrar eğiterek sonuçlar aldık. Aktarımlı öğrenmenin önemi ilerleyen kısımlarda daha iyi anlaşılacaktır.



Şekil 3.6: Aktarımlı öğrenme ve normal öğrenme ile sınıflandırma [65]

Aktarımlı öğrenme yöntemi; yukardaki görselden de anlaşılacağı üzere, görüntü üzerinde ne olduğunu anlamak için standart yöntemden farklı olarak sıfırdan bir model eğitimi yapmak yerine daha önceden büyük veri setleri ile eğitilmiş modelleri kullanarak sınıflandırma yapmaktır. Standart yaklaşımda olduğu gibi kendi problemimize ait veriseti toplayarak ön-eğitimli(pretrained) model ile tahmin edebilebilmektedir. Problemler değişkenlik göstereceği için verisetleri de farklılık gösterecektir fakat az veri ile sıfırdan eğitilen bir modele göre daha yüksek başarım [92] vereceği için çok sık tercih edilen bir yöntemdir. Tahmin ve başarım [92] sonuçları verinin kalitesi gibi farklı parametrelere değişkenlik göstereceği için kesin olarak iyi bir yöntemdir denilemez. Karşılaştırılmalar yapılarak artıları ve eksileri değerlendirilmelidir.

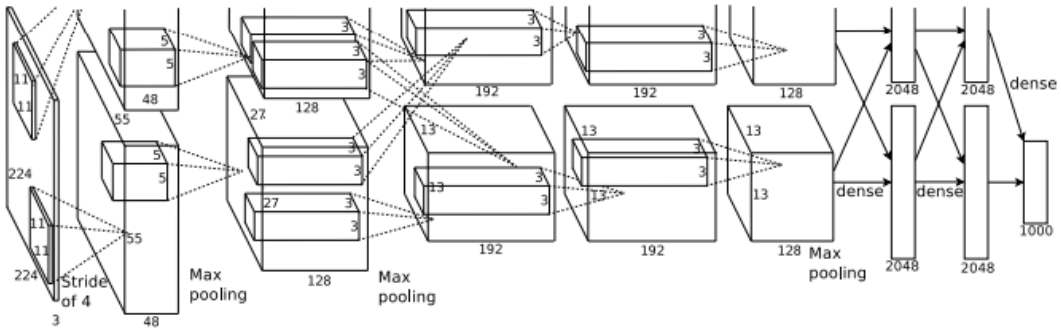
Çalışmamızda kullanılan bazı ön-eğitimli evrişimli sinir ağı olan aktarımlı öğrenme modelleri alt kısımdadır.

3.4.2.1 AlexNet

Alex Krizhevsky tarafından Ilya Sutskever ve Krizhevsky'nin danışmanı olan Geoffrey Hinton ile işbirliği içinde tasarlanan bir evrişimsel sinir ağı mimarisinin adıdır [46].

Ağ ağırlıkları olan sekiz katman içerir; ilk beşi evrişimseldir ve geri kalan üçü tamamen bağlantılıdır. Son tam bağlı katmanın çıktısı, 1000 sınıf etiketi üzerinde bir dağıtım üreten 1000 yollu bir softmax'a beslenir. Ağ, tahmin dağılımı altında doğru

etiketin log-olasılığının eğitim durumları arasındaki ortalamayı en üst düzeye çıkarmaya eşdeğer olan çok terimli lojistik regresyon hedefini en üst düzeye çıkarır [87].



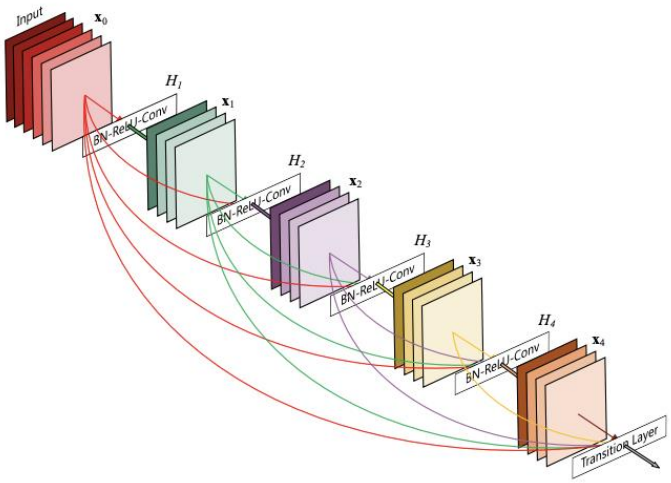
Şekil 3.7: AlexNet sinir ağının yapısı [87]

AlexNet mimarisi, yalnızca son denetim terimiyle sınıflandırma göreviyle ilgilenir. Önceden eğitilmiş AlexNet mimarisinin amacının, son çıktı katmanında sınıflandırma hatasının en aza indirilmesi için filtre katmanlarını ve ağırlıkları öğrenmek olduğu belirtilmektedir [88].

3.4.2.2 DenseNet

DenseNet, her katmanı diğer tüm katmanlara ileri beslemeli bir şekilde bağlayan yoğun evrişim ağıdır. Halbuki L katmanlı geleneksel evrişimli ağlar, her katman ve sonraki katman arasında bir tane olmak üzere L bağlantıya sahiptir. DenseNet ağı L(L+1)/2 doğrudan bağlantıya sahiptir [89]. Her katman için, önceki tüm katmanların özellik haritaları girdi olarak kullanılır ve kendi özellik haritaları, sonraki tüm katmanlara girdi olarak kullanılır. DenseNet'lerin birkaç zorlayıcı avantajı vardır: kaybolan gradyan sorununu hafifletirler, özellik yayılımını güçlendirirler, özelliklerin yeniden kullanımını teşvik ederler ve parametre sayısını önemli ölçüde azaltırlar.DenseNet'ler, yüksek performans elde etmek için daha az hesaplama gerektirirken, çoğunda en son teknolojiye göre önemli iyileştirmeler elde eder [89].Tüm katmanları (eşleşen özellik haritası boyutlarıyla) doğrudan birbirine bağladığımız Yoğun Bloklar aracılığıyla katmanlar arasındaki yoğun bağlantıları kullanan bir tür evrişimsel sinir ağıdır. İleri beslemeli doğayı korumak için, her katman

önceki tüm katmanlardan ek girdiler alır ve kendi özellik haritalarını sonraki tüm katmanlara iletir [47].



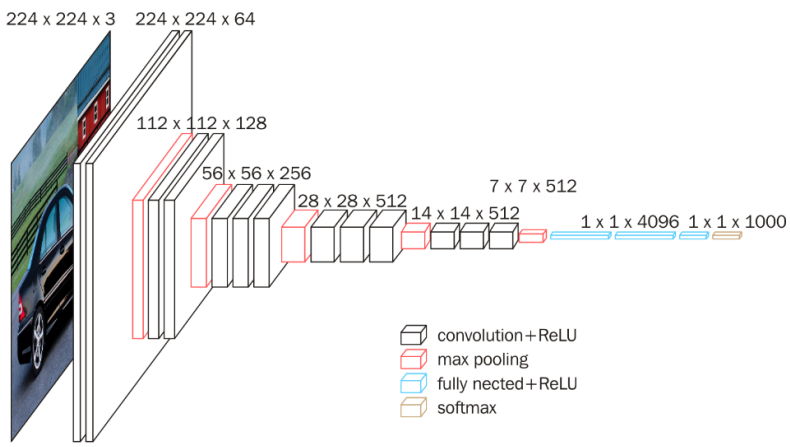
Şekil 3.8: DenseNet sinir ağının yapısı [67]

3.4.2.3 VGG16

VGG16, 2014 yılında ILSVR(Imagenet) yarışmasını kazanmak için kullanılan bir evrişim sinir ağı (CNN) mimarisidir. Bugüne kadar mükemmel görüntü modeli mimarisinden biri olarak kabul edilmektedir [90].

Basit bir ağ modeli olup öncesindeki modellerden en önemli farkı evrişim katmalarının 2'li ya da 3'li kullanılmasıdır.Tam bağlantı (FC) katmanında 7x7x512=4096 nöronlu bir öznitelik vektörüne dönüştürülür. İki FC katmanı çıkışında 1000 sınıflı softmax başarımı [92] hesaplanır. Yaklaşık 138 milyon parametre hesabı yapılmaktadır. Diğer modellerde olduğu gibi girişten çıkışa doğru matrislerin yükseklik ve genişlik boyutları azalırken derinlik değeri (kanal sayısı) artmaktadır [49].

Tüm mimari boyunca tutarlı bir şekilde evrişim ve maksimum havuz katmanları. Sonunda 2 FC'ye (tamamen bağlı katmanlar) ve ardından çıktı için bir softmax'a sahiptir. VGG16'daki 16, ağırlıkları olan 16 katmana sahip olduğunu ifade eder. Bu ağ oldukça büyük bir ağdır ve yaklaşık 138 milyon (yaklaşık) parametreye sahiptir [90].



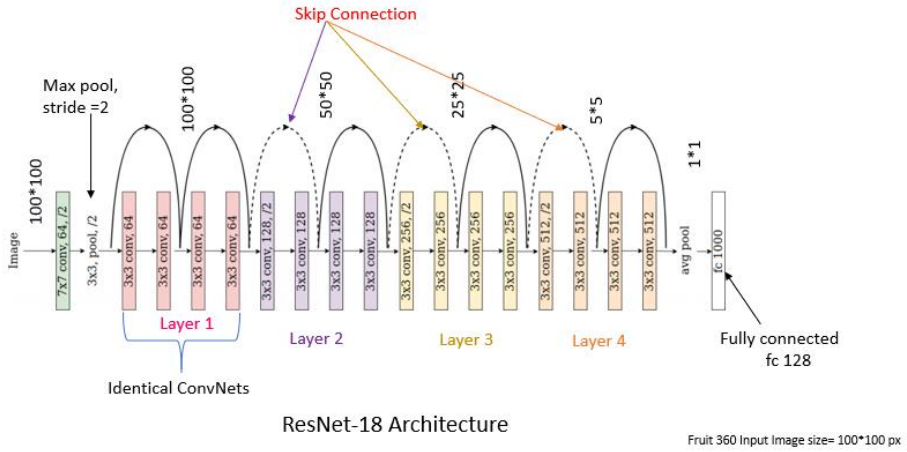
Şekil 3.9: VGG16 sinir ağının yapısı [90]

3.4.2.4 ResNet18

ResNet'i popüler yapan nedenlerden biri, yalnızca bir kimlik kısayolu sunan basit tasarım stratejisidir. Büyük başarısına rağmen, kimlik kısayolunun zayıflığı, takip çalışmaları ile analiz edilmiştir. Kimlik kısayolu, özellikleri korumak için kalan blokları atlar ve sonuç olarak ağın temsil gücünü sınırlar. ResNets'in dezavantajı, ağın öğrenme kapasitesini azaltan ve doğrusal olmayan kısayollarla hafifletmeyi öneren çöken alan sorununa neden olmasıdır [91].

VGGNet'in temel modelini, iki tasarım kuralına sahip çoğunlukla 3×3 filtreli düz bir ağ olarak alındı. Aynı çıktı özellik haritası boyutu için, katmanlar aynı sayıda filtreye sahiptir ve özellik haritası boyutu ise yarıya indirildiğinde, katman başına zaman karmaşıklığını korumak için filtre sayısı iki katına çıkar [91]. Ağ, global ortalama havuzlama katmanı ve softmax ile 1000 yollu tam bağlantılı katmanı şeklinde sonlandırılır.

Düz ağa dayalı olarak, düz sürümü artık versiyona dönüştürmek için kısayol bağlantıları eklenir [91].

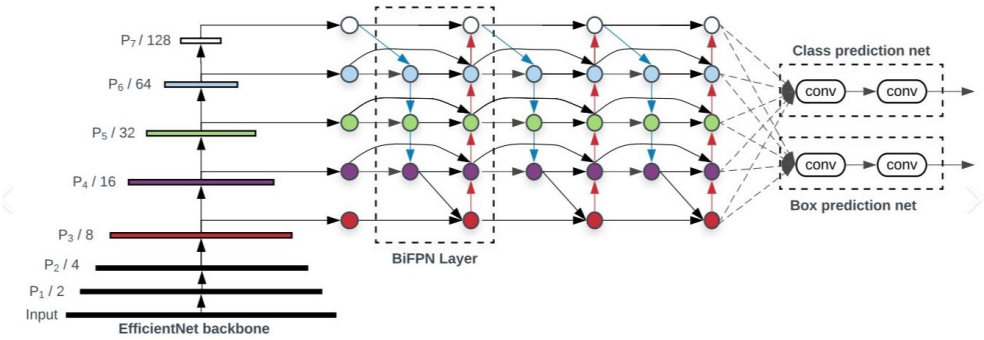


Şekil 3.10: ResNet18 sinir ağının yapısı [69]

3.4.2.5 EfficientNet B0 – EfficienNet B7

EfficientNet, bir grup evrişimli sinir ağı modeli olarak düşünülebilir. Ama tabi ki incelikleri sayesinde adından da anlaşılacağı gibi kendisinden önceki modellerin çoğundan efektif/verimli (efficient)dir. EfficientNet grubu B0-B7 arasında 8 tane modelden oluşur ve sayı büyüdükçe hesaplanan parametre sayısı ve doğruluk artar [50].

Yeni bir temel ağ tasarlamak, önceki ConvNet'lerden çok daha iyi doğruluk ve verimlilik sağlayan EfficientNets adlı bir model ailesi elde etmek amacıyla onu ölçeklendirmek için sinir mimarisi araması kullanılıyor. Özellikle EfficientNet-B7, mevcut en iyi ConvNet'ten 8,4 kat daha küçük ve 6,1 kat daha hızlı çıkarım yaparken, ImageNet, son teknoloji %84,3 ilk 1 doğruluğa ulaşır. EfficientNet'ler ayrıca CIFAR-100 (%91,7), Flowers (%98,8) ve diğer 3 aktarımlı öğrenme veri setinde daha iyi aktarım yapar ve en son teknoloji doğruluğu elde eder. Büyüklük sırasına göre daha az parametreye sahiptir [32].



Şekil 3.11: EfficientNet sinir ağının yapısı [71]

3.4.3 Hastalık Tespiti

Çalışmamız içerisinde ilk odaklandığımız ve bizi en çok zorlayan konu hastalık tespiti olmuştur. Verisetinin az olması nedeniyle yapılan çalışmalar da sınırlı kapasitedeydi. Sıradan bir sınıflandırma problemi gibi duran bu problem birçok yerde teoride kalmış ve ciddi anlamda kayda değer bir çalışma çok azdır. Bu nedenle derin öğrenme yöntemlerinin getirdiği özellikler kullanılarak veri artırmalar (data augmentation) yapılmıştır. Veri artırma tekniği; elde var olan verileri kullanarak benzer verilerin oluşturulma işlemidir. Derin öğrenme algoritmaları, makine öğrenmesi algoritmalarına nazaran az veriyle eğitim ve test işleriminde çok kötü sonuçlar vermektedir. Önceki bölümlerde bahsedildiği üzere derin öğrenme algoritmaları çok fazla veriye ihtiyaç duyar. İhtiyaç duyulan veriler yeterince elde edilemez ise farklı yöntemlere başvurmak gerekebilir.

Hastalık tespiti ve sınıflandırma yapabilmek adına ilk yaptığımız veriseti toplamaktı ve o şekilde ilerledik. Literatürdeki verisetlerini inceleneyerek en ideal olanı seçtik.

3.4.3.1 Veriseti

Kullanılan veriseti, alınan çıktıların yani tahminleme işleminin ne derece başarılı olacağını belirler.

Bu çalışma içerisinde kullanılan veriseti, hastalıklı yaprak görüntülerinden oluşan verisettir [38]. Herkese açık şekilde kullanıma açıktır. Bitki yapraklarına ait görüntüleri barındırmaktadır. Çeşitli bitkilerin hastalıklı yapraklarının ve sağlıklı

yaprakların resimleri toplanarak oluşturulmştur. Her bitki ve hastalık türü eşit derecede hazırlanmamıştır.

Verisetinin içerdiği görüntü sayısı ve dağılımları incelemek gerekirse, veriseti 2 farklı şekilde sunulmaktadır. İlk senaryo veriseti veri artırma (data augmentation) yöntemini içermeyen yani görüntü sayısının ve çeşitliliğin az olduğu durum. Bu senaryoyu yapmadık çünkü modelimizin başarısı veri sayısına bağlı olduğu için diğer senaryoyu yani veri artırma yöntemi uygulanmış olanı seçtik. Seçilen verisetine ait durum:

Class	Plant Name	Diseases Name	Causes Name	Virus	Type of Diseases	No. Of Images
C27	Apple	Apple Black Rot			Rot	1000
C28	Apple	Apple Scap				1000
C29	Corn	Corn Common Rust			Rust	1000
C30	Cornn	Corn Gray Leaf Spot			Spot	1000
C31	Grape	Grape Black Rot				1000
C32	Tomato	Tomato Bacterial Spot	Spot		Bacteria - Spot	1000
C33	Tomato	Tomato Late Blight			Pest	100
C34	Tomato	Tomato Leaf Mold				1000
C35	Tomato	Tomato Mosaic Virus	Tomato Mosaic		Viral	1000

Şekil 3.12: Verisetinin içeriğinden bir kesit [38]

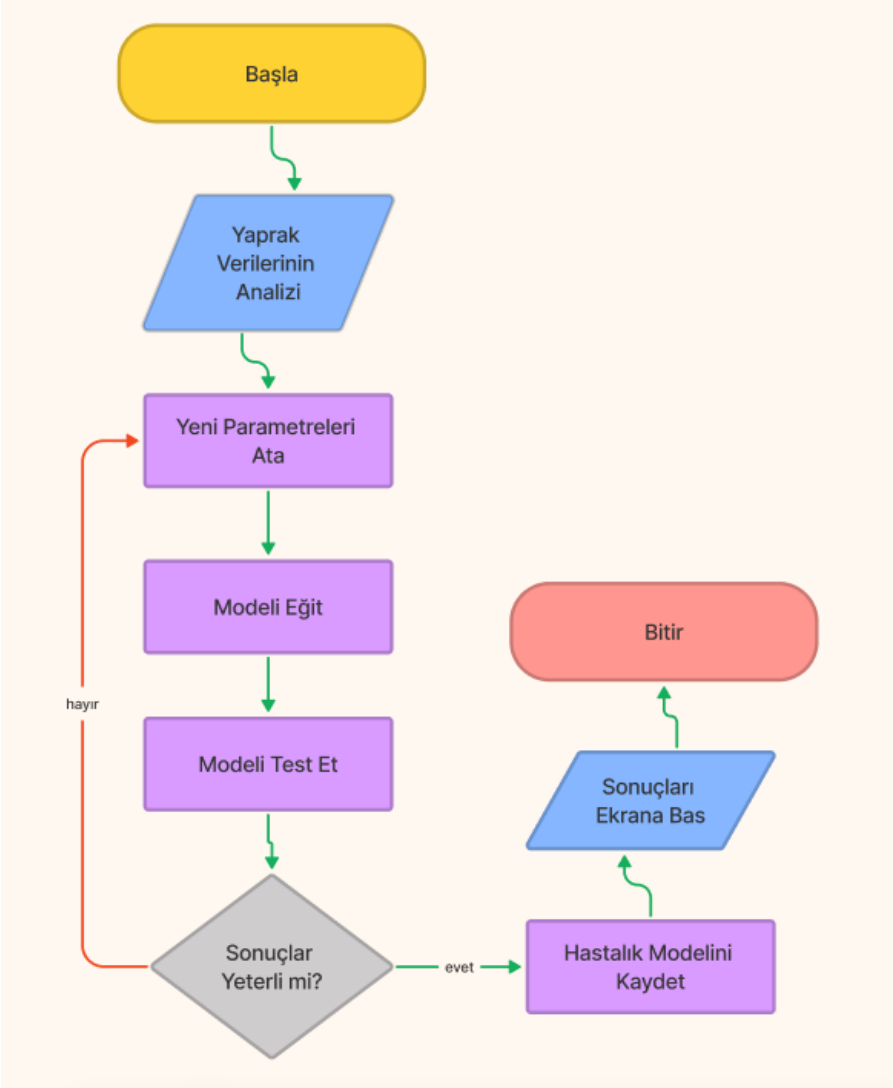
Hastalık tespiti için kullandığımız veriseti 39 farklı görüntü sınıfı içermektedir. Yukarıda bulunan kesit örneğinde farklı türde bitkilerin olduğunu ve o bitki türlerine ait çeşitli hastalıkların bazılarını görebiliriz.

Veriseti içerisinde; 15 tane farklı küme vardır. Kümeler kendi içerisinde ait olduğu bitkiye ait hastalık sınıfı bulundurmaktadır. Elma bitkisi hastalıkları kümesinde 4 adet, kiraz için 2 adet, mısır için 4 adet, üzüm için 4 adet, portakal için 1 adet, şeftali için 2 adet, patates için 3 adet, çilek için 2 adet, domates için 10 adet, biber için 2 adet, soya fasulyesi için 1 adet, kabak için 1 adet, yabanmersini için 1 adet ve en son olarak tüm sınıfların olmaması ihimaline karşın yaprakların olmadığı alan olduğun anlamak için 1 adet yapraksız kısım sınıfı bulunmaktadır.

Veriseti doğrudan eğitimde kullanılamayacağı için veri arttırma yöntemi uygulandıktan sonra bölütlemeler yapılmıştır. Veriseti %60'ı eğitim verisi, %30'u test verisi ve %10 luk kısmı doğrulama olarak ayrılmıştır. Her bir sınıf için; 600 resim eğitime, 300 resim teste ve 100 resim de doğrulama için klasörlenmiştir.

3.4.3.2 Sinir Ağı Yapısı ve Eğitimi

Üretici tarafından yetiştirilen bitkilerin zaman içerisinde hastalıklara yakalandığı bilinmektedir. Bu hastalıkların ne olduğunu bulabilmek için farklı yöntemler mevcut fakat biz çalışmamızda derin öğrenme mimarisi olan evrişimsel sinir ağlarını kullarak tespit ettik. Aktarımlı öğrenme modellerinden evrişimli sinir ağlarını(CNN) kendi topladığımız veriseti ile eğiterek probleme çözüm getirmiş olduk.



Şekil 3.13: Bitki hastalık tespiti akış diyagramı

Bitki yapraklarından hastalık tespiti için tüm eğitim ve test sürecinin işleyişini Şekil 3.13'teki diyagram üzerinde görebiliriz.

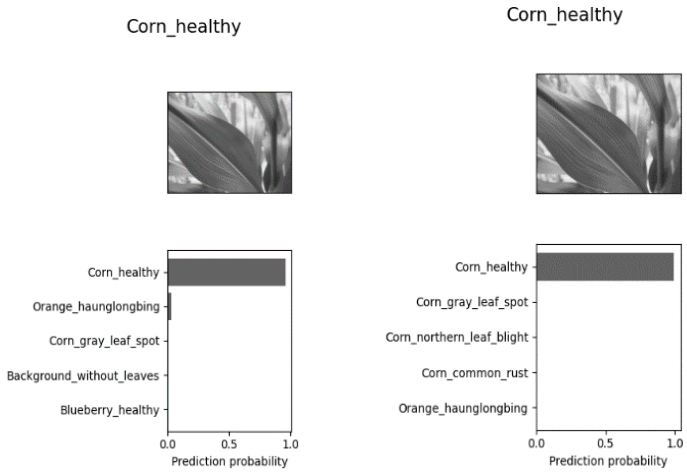
Verisetleri farklı şekillerde bölünerek eğitim ve test süreçlerine başlanılmaktadır. Biz verisetimizin büyük bir kısmını eğitim için ayırmayı tercih ettik. Geriye kalan kısım ile modelimizi test ederek başarım [92] sonuçlarını değerlendirdik. Yeterli başarım [92] değerlerine ulaşıncaya kadar eğitim sürecine devam ettik. Her bir aktarımlı öğrenme modelleri olan AlexNet, ResNet18, VGG16, EfficientNetB0, EfficientNetB7 ve DenseNet ile eğitim süreçlerini tekrar ettik. Aşağıdaki tabloda modellerin eğitim süreleri, tekrar sayıları (epoch değeri) ve başarım oranlarını (accuracy) [92] görebiliriz.

Tablo 3.1: Modellere göre eğitim süreleri ve başarımları

Model Adı	Eğitim Süresi	Toplam Tekrar Sayısı	En İyi Tekrar Numarası	Başarım (Acc)
AlexNet	163m 24s	100	82	0.8535
DenseNet	925m 17s	100	71	0.8765
VGG16	367m 21s	100	100	0.9488
ResNet18	139m 1s	100	56	0.9289
EfficientNet B7	602m 50s	75	97	0.9248
EfficientNet B0	190m 20s	100	40	0.8852

Başarım [92] oranlarının birbirine yakın olması ile eğitim süreleri arasında doğrusal bir ilişki yoktur. Her modelin kendine ait evrişimli katman sayısı, havuzlama katmanı sayısı ve tam bağlantı katmanların derinliği farklıdır. Katman sayılarının farklı olması ya da derinliklerden ziyade yapay sinir ağlarının içerisindeki düğümlerin (node) ağırlık değerleri de öenmli bir faktördür. Yapay sinir ağlarının her katmanı belirli sayısal ağırlık değerlerine sahiptir ve her katmanın sonucu diğer katmana geçtiğinde bu ağırlık değeri ile çarpılır. Çok sayıda katmanın olması ya da ağırlıkların yüksek olması yapay sinir ağı modelinin var olan nesneyi ezberlemesi anlamına gelen 'Dropout' a düşebilir. Bu durumun olması hiç istenmez çünkü ezberleyen model kesinlikle yanlış tahminleme yapar. Hataya düşmemek için biz de yapay sinir ağının ezberlerini bozmak için bazı tambağımlı katmanların içerisinden silme işlemi yapıyoruz. Aynı zamanda düğümlerin ağırlık değerlerini de değiştirerek yeniden eğitime alıyoruz. Genel olarak bu sürecin literatürdeki adı hiper parametre ayarlaması (hyper parametere tuning) dır ve sıkça tercih edilen bir yöntemdir.

Yeterli görülen eğitimlerin ardından örnek olarak yaptığımız tahminlemeyi alt kısımda görebiliriz.



(a) (b)

Şekil 3.14: Tahmin görüntüleri, (a) EfficientNet B7, (b) AlexNet

İnternetten rastgele seçilen sağlıklı bir mısır yaprağına ait görüntü, eğitilen farklı modellere verilerek sonuçları kıyaslanmıştır. Çıktılarından İlk tercihlerine bakarak iki modelin de başarılı tahmin yaptığını söyleyebiliriz. İkinci ve sonraki tercihlerinde farklı sınıfları işaret etmeleri de oldukça normaldir. Çünkü bütün modeller birbirinden farklı yapılara sahiptir.

3.4.4 Bitki Gelişim Takibi

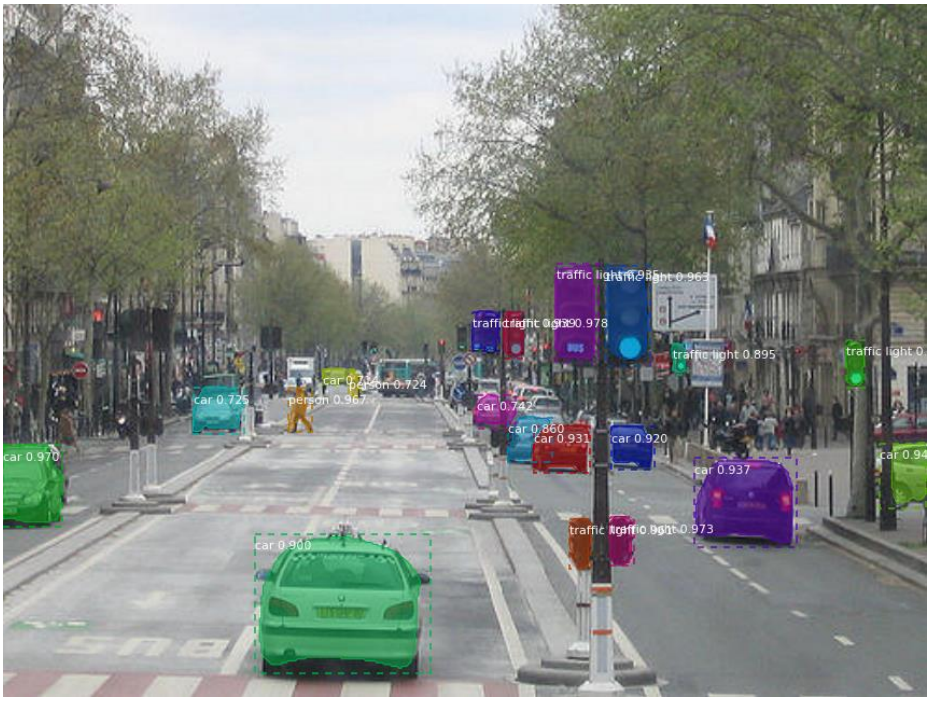
Çalışmanın bu kısmında MaskRCNN sinir ağını fasulye bitkisi görselleri ile eğiterek; bitkinin gelişim aşaması takip edilmiştir.

Geleneksel yöntemlerde bitki gelişimi, farklı yöntemler ile takip edilebilmektedir. Çoğunlukla insan faktörü kullanılarak yapılmaktadır. Ürünleri yetiştiren üreticiler veya çalışanlar düzenli olarak ekimin yapıldığı alanları kontrol etmek zorundadır. Yeterli olgunluğa erişim sağlanamaz ise (çevresel faktörler hariç) harcanan emek, zaman ve maliyetler boşa gitmiş olacaktır. Bu durum üretim sisteminin otomatize edilmesi zaman ve maliyet takibi için büyük önem taşımaktadır. Düzenli olarak üründen elde edilen veriler ile haftalık, aylık veya yetiştirilen ürüne göre farklı

periyodik zamanlar da kontrol edilebilir. Bitkinin boy uzunluğu, yaprak sayısı veya gövde kalınlığı gibi veriler analiz edilebilir verilerdir. Biz de probleme yaprak sayıları ve yaprakların kapladıkları alanları baz alarak çözüme gittik. Yaprak sayıları ve yaprağın kapladığı alanı elde edebilmek adına elbette görsel verileri toplayarak derin öğrenme modelleri için kullandık. Alt bölümde derin öğrenme yapısı, verisetinin toplanması ve kullanılması gibi detayları göreceğiz.

3.4.4.1 Mimari Yapı

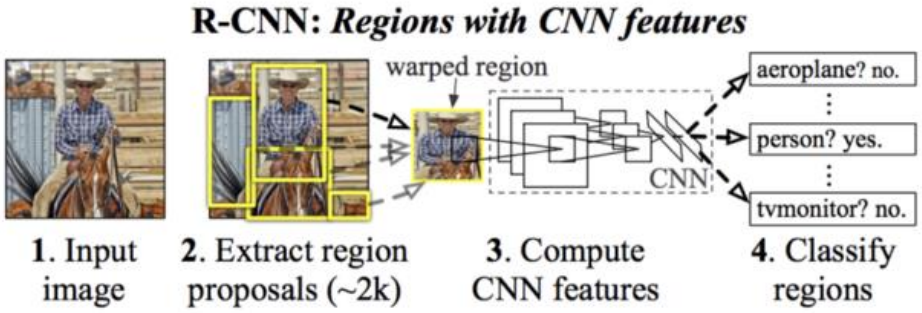
Aktarımlı öğrenme yöntemlerine (transfer learning) ve nasıl çalıştıklarına değinmiştik. Bitkinin gelişimini takip edebilmek için yaprak sayıları ve yaprakların kapladıkları alanları hesaplayabilmek için bu aktarımlı öğrenme yöntemine giderek çözüme ulaştık. Görüntüler üzerinden nesne tespit etme ve ayrıştırma problemlerine çözüm amacıyla Facebook tarafından geliştirilen aktarımlı öğrenme yöntemi olan 'MaskRCNN (Maskelenmiş Bölge Tabanlı Evrişimsel Sinir Ağı)' mimarisi kullanmılmıştır. MaskRCNN ile çeşitli görüntüler üzerinden nesne maskelemesi yapılabilmektedir.



Şekil 3.15: MaskRCNN mimarisi ile cadde üzerindeki araçların, yayaların ve farklı nesnelerin tespit edilmiş hali [7]

MaskRCNN mimarisi temelinde R-CNN (Region Based Convolutional Neural Network) mimarisine dayanmaktadır. Bölge tabanlı nesne tespit etme algoritması 2014 yılında Girshick tarafından geliştirilmiştir. MaskRCNN'i anlamak için önce R-CNN'i incelemek gerekmektedir.

Evrişimsel sinir ağları (CNN) çeşitli varyasyonlar ile karşımıza çıkmaktadır. Önceki bölümlerde görüntü verileri ile nasıl çalıştığına basit düzeyde değinmiştik. Ancak problemlerin büyümesi ve çeşitlenmesi ile sıradan evrişimli sinir ağları yukarıda görüldüğü üzere farklı mimarilere dönüşümektedir. Her gelişim kendi içerisinde CNN yapılarını kullanarak ayrı bir seviyeye geçmiştir. İlk olarak R-CNN yapısına ardından da MaskRCNN yapısına dönüşmüştür. CNN'den bahsedildiği için şimdi R-CNN yapısından bahsedilecektir.



Şekil 3.16: R-CNN mimarisi [70]

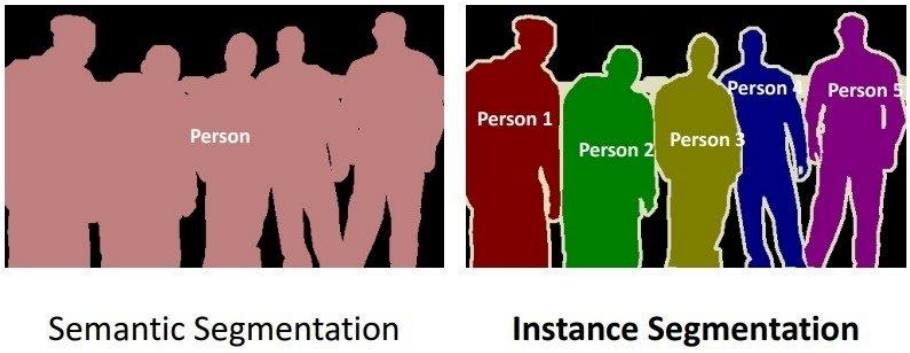
R-CNN algoritması temelinde 4 adımı beraberinde getirir. İlk adım görüntü verilerin aldındığı adımdır. Ardından 'Extract Region Proposals' adımına geçerek görsel verinin içeriğini amaca uygun şekillerde parçalayarak bölge bölge ayırma işlemi yapılır [45]. Ayrılan bölgeler üçüncü adımda evrişimli sinir ağına gelerek birer sınıflandırma işlemine tabi tutulur. Sınıflandırma tahminleri son kısımda toplanarak tüm sonuçlar analiz edilir. Kabaca anlattığımız bu sistem aslında verilerin karmaşıklığı ve boyutlarının artmasıyla oldukça fazla işlem gücü gerektirmektedir. Görsel veriler hem ön işleme girer hem de sınıflandırma işlemine girer. Doğal olarak performans değerlerini iyileştirmek için farklı çalışmalar da yapılmaktadır. Faster R-CNN algoritması buna oldukça güzel bir örnektir. Hatta Facebook tarafından yayınlanan 'MaskRCNN' adlı makalede kıyaslamaları ve ayrımları da görülebilir.

Mask R-CNN; evrişimli sinir ağı (CNN) yapısında, görüntü segmantasyonu ve örnek segmentasyonu açısından son teknoloji ürünüdür. Detaylandırmak gerekirse, ilk

adımda görüntü segmentasyonuna bakmak gerekir. Görüntü segmentasyonu resim içerisindeki çizgiler, eğrileri ve lokal sınırları bulmaktır. Bu sınırları semantik segmantasyon ve örnek segmentasyonu olarak 2 şekilde bulabiliriz.

Semantik segmentasyon, nesne örneklerini ayırt etmeden her pikseli sabit bir kategoriler grubuna sınıflandırır. Başka bir deyişle, semantik segmentasyon, benzer nesnelerin piksel seviyesinden tek bir sınıf olarak tanımlanması/sınıflandırılması ile ilgilenir [45].

Örnek Segmentasyonu veya Örnek Tanıma, bir görüntüdeki tüm nesnelerin doğru algılanmasıyla ilgilenirken aynı zamanda her örneği kesin olarak bölümlere ayırır. Bu nedenle, nesne algılama, nesne lokalizasyonu ve nesne sınıflandırmasının birleşimidir. Başka bir deyişle, bu tür bölümleme, benzer örnekler olarak sınıflandırılan her nesne arasında net bir ayrım sağlamak için daha da ileri gider [45].



(a) (b)

Şekil 3.17: Segmente edilmiş bir görüntü (a) Semantik segmentasyon, (b) Örnek segmentasyonu [45]

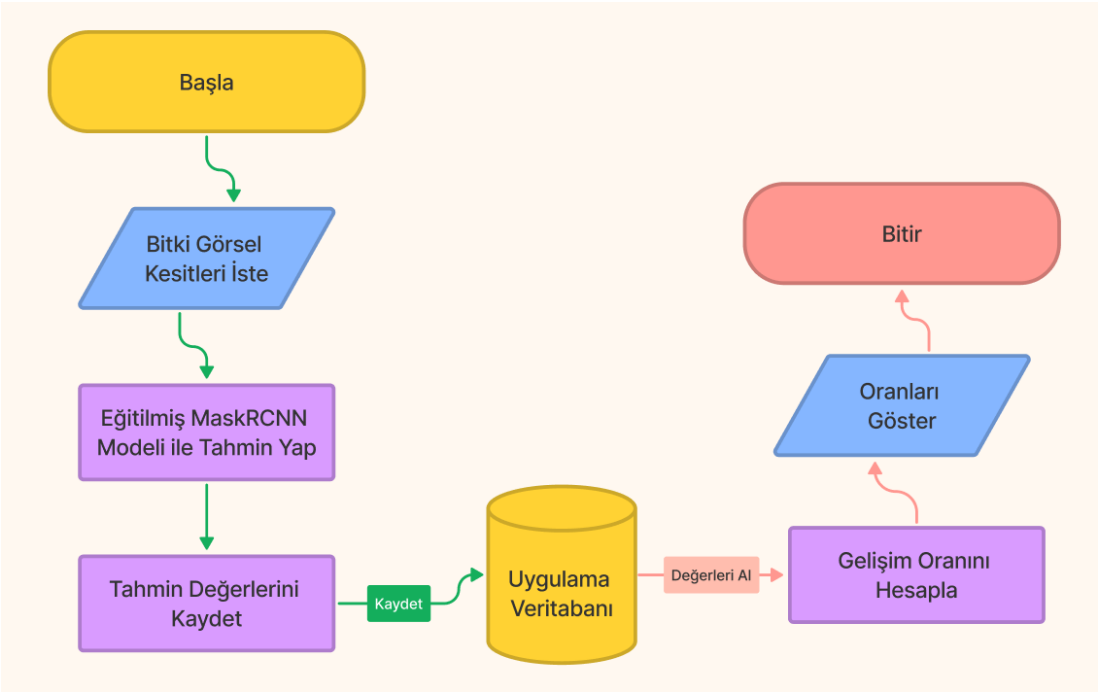
Soldaki resimde semantik segmentasyon ile nesneler bir varlık gibi olarak sınıflandırılmış. Görüntünün özneleri arka plandan ayrıldığı için arka plan segmentasyonu olarak da bilinir.

Sağdaki resimde örnek segmentasyon ile bir bütün olarak gösterilen nesnelerin her biri ayrı birer varlık olarak ayrılır ve ön plan segmentasyonu olarak da geçer.

Geriye ayrılan kısımların tahminlemesi kalır. Elbette bu süreçte MaskRCNN yapısı içerinde nesnelerin, sınırlandırmaları ve çıkarımları yapılmak zorundadır.

MaskRCNN'in avantajları arasında; basit eğitilebilir olması, her görevde mevcut tek model girişlerinden daha iyi performans göstermesi, Faster R-CNN'e göre çok daha verimli olması ve ensek yapısı sayesinde çeşitli görüntü problemlerinin çözümüne uyarlanabilecek esneklikte olması sayılabilir.

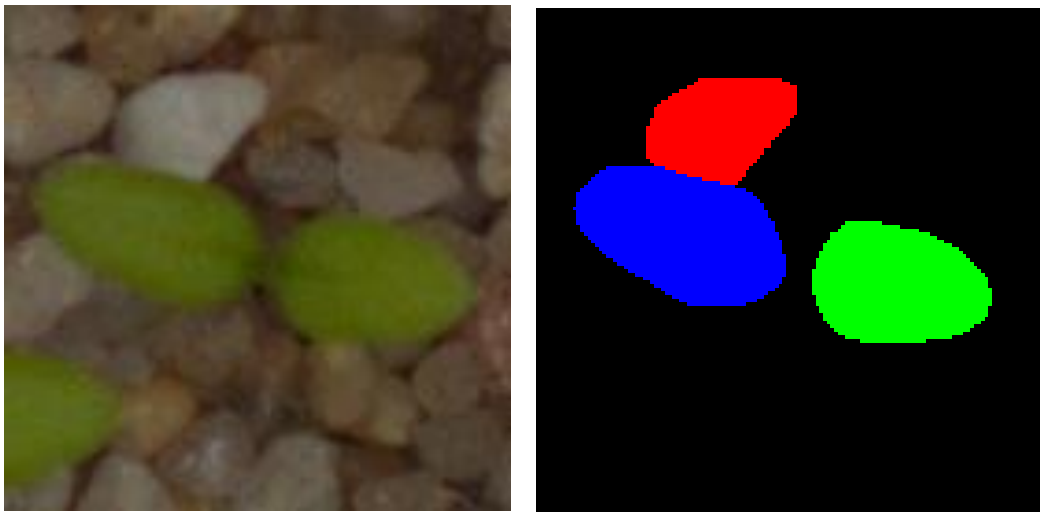
3.4.4.2 Uygulama



Şekil 3.18: Gelişim takibi yapısı

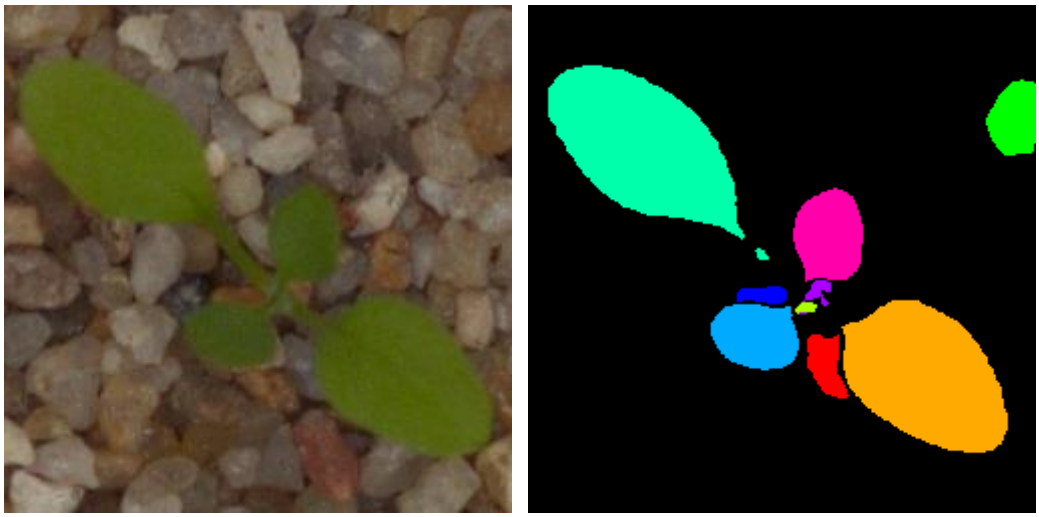
Bitki görüntüleri üzerinden gelişim takibi yapabilmek için işletilen tüm süreci Şekil 3.18 üzerinde görebiliriz.

MaskRCNN mimarisinin çeşitli problemlere çözüm üretmek için esnek bir yapıya sahip olduğu biliyoruz. Bu esneklik sayesinde biz de kendi verisetlerimizi kullanarak yapay sinir ağı model eğitimlerini yaptık. Yaklaşık olarak modellerin eğitimleri 1 aydan fazla zaman almıştır. Bitkinin gelişim aşamasındaki görüntüleri ile eğitilen modelimiz artık ayrım yapabilir hale geldi. Görüntü üzerinde ayrımını yaptığı bitkileri maskelenmiş şekilde elde ettik.



(a) (b)

Şekil 3.19: MaskRCNN birinci girdi ve çıktı örneği, (a) Gelişimdeki bir bitki, (b) Maskelenmiş hali.



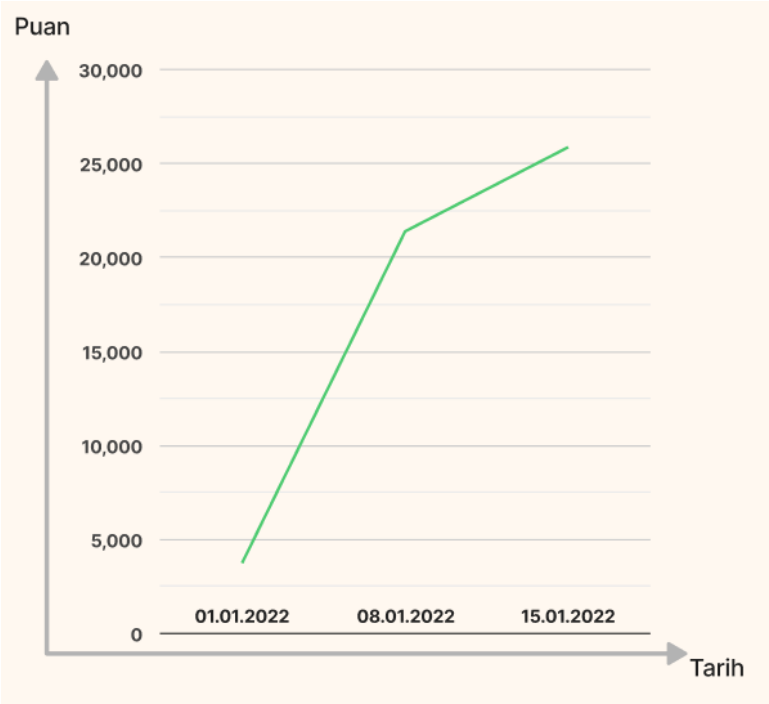
(a) (b)

Şekil 3.20: MaskRCNN ikinci girdi ve çıktı örneği, (a) Gelişimdeki bir bitki, (b) Maskelenmiş hali.

(a) (b)

Şekil 3.21: MaskRCNN üçüncü girdi ve çıktı örneği, (a) Gelişimdeki bir bitki, (b) Maskelenmiş hali.

Yukarıdaki resimde de görüldüğü üzere eğitilen MaskRCNN derin öğrenme modeli, bitki yapraklarını renkli şekilde maskeleyerek çıktı olarak verdi. Fakat bitki yaprağı bulunmayan bölgelerde de maskeleme yaparak doğruluk payını düşürmüştür. Sadece yeşil kısımların maskelenmesi için eğitilen modelin yüksek doğruluğa sahip olması gerekmektedir.



Şekil 3.22: Gelişim takibi grafiği

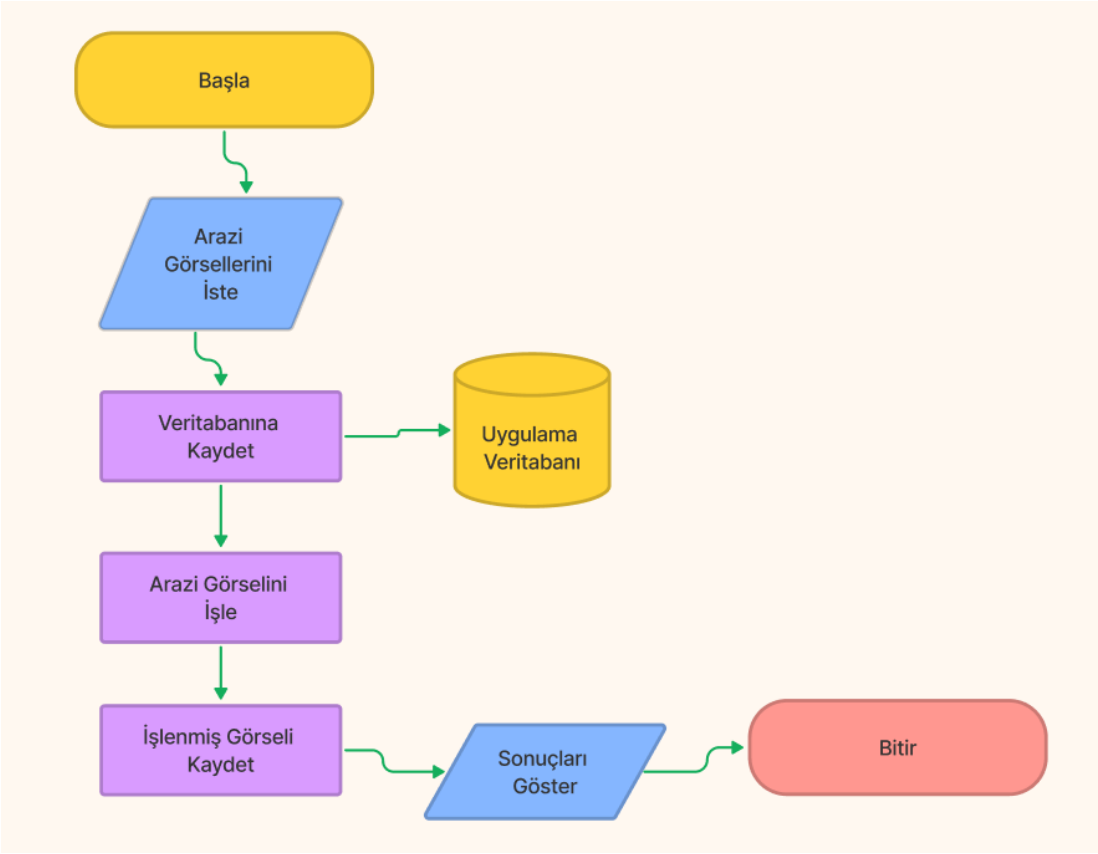
Buradaki kritik nokta, modelin görsel içerisindeki bitki yapraklarını ve gövdelerinin bulunduğu yeşil kısımlarını tespit ederek maskelemektir. Maskelenen alanların gelişim takibindeki asıl rolü, sahip oldukları alanları hesaplatarak bir değer elde edilmesidir. Periyodik olarak , bitki yapraklarının hesaplanan alanları bitkinin gelişimini takip etmek için kullanılmaktadır. Alanların arasındaki farkların artması; bitkinin hızlı gelişim gösterdiğini, alanlar arasındaki farkların artmaması ya da az artması bitkinin yeterince gelişim gösteremediğini ifade etmektedir. Doğal olarak bu durum 'Time-Series' zamansal problemlere dönüşmektedir. Kritik kısım, hesaplamaların doğru sonuçlara götürebilmesi için hesaplanan alanların doğru olması gerekmektedir. Eğer model yanlış veya hatalı tespit edip maskleme işlemi yaparsa alan hesabı da yanlış olacaktır. En sonda da gelişimin takibindeki oranlar hatalı çıkarak bitkinin ne durumda olduğu anlaşılamayacaktır. Örneğin eksi değerde bir oranın çıkması beklenemez. Çünkü büyüme yani artma olması durumu hep artı yöndedir. Eksi yönde gelişim söz konusu olamaz. Farkların eksi çıkması gelişimin tersi yönde gittiğini gösterir bu da gelişim olmaz küçülme olur.

Genel olarak dikkat edilmesi gereken kısımları, uygulamaları, yöntemleri ve hedeflerin neler olduğunu değerlendirmiş olduk.

3.4.5 Arazi Durum Kontrolü

Çalışmanın bu kısmında uydu araçlarından veya insansız hava araçlarından alınan arazi görüntüleri ile arazi üzerindeki değişimlerin takibi yapılmıştır.

Üretim yapılan araziler yıl içerisinde farklı şekillerde görünebilmektedir. Hasat öncesinde, hasat sırasında, hasatın kaldırılmasına yakın zamanda ve hasatın sonrasında farklı görüntüye sahip olmaktadır. Arazilerin irili ufaklı, engebeli, ağaçlık, kayalık veya bataklık gibi çeşitli olması üreticinin işini epey zorlaşırmaktadır. Akıllı tarımın asıl amaçlarından biri olan üreticinin işini nasıl kolaylaştıracağımızın üzerine düşündük ve bazı çözümler geliştirdik. Bu çözümlerin çalışması için öncelikle üretici yetiştirdiği ürünlerin bulunduğu arazileri yukarıdan periyodik olarak görüntülemesi gerekmektedir. Toplanan görüntüler işlenerek, üreticiye arazinin durumunu daha iyi görebilmesi için sonuçlar çıkarılacaktır. Arazi yapısının değişimini kontrol edebilmek için yürütülen süreci Şekil 3.23 üzerindeki diyagramda görebiliriz.



Şekil 3.23: Arazi durum kontrolü görüntü işleme süreci

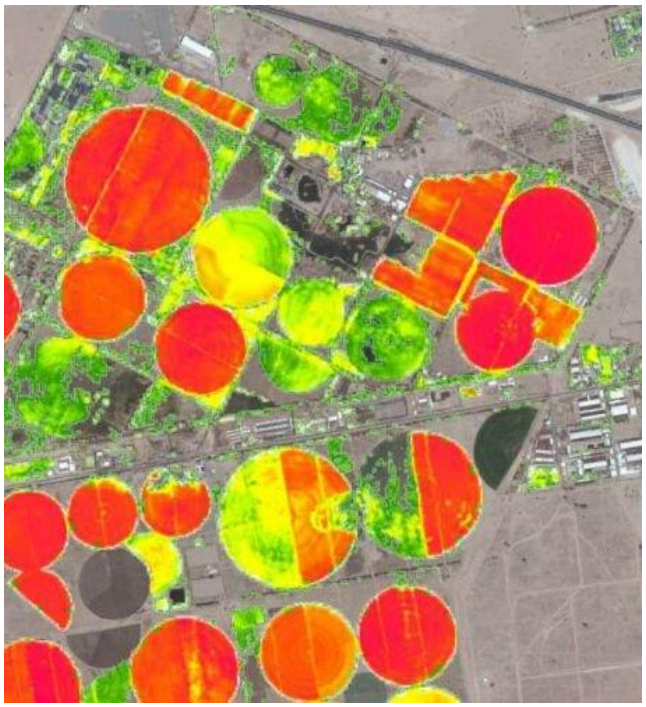
İnternetten rastgele seçilen sağlıklı bir mısır yaprağına ait görüntü, eğitilen farklı modellere verilerek sonuçları kıyaslanmıştır. Çıktılarından İlk tercihlerine bakarak iki modelin de başarılı tahmin yaptığını söyleyebiliriz. İkinci ve sonraki tercihlerinde farklı sınıfları işaret etmeleri de oldukça normaldir. Çünkü bütün modeller birbirinden farklı yapılara sahiptir.

Sayısal görüntü işleme bilgisayar algoritmaları kullanarak sayısal resimler üzerinde görüntü işlemenin gerçekleştirilmesidir. Sayısal sinyal işlemenin bir alt konusu olarak kabul edilen sayısal görüntü işleme, analog görüntü işlemeye göre birçok avantaja sahiptir. Sayısal görüntü işlemede giriş verilerine uygulanabilecek algoritmalar daha fazladır ve analog görüntü işlemeye göre işlem sırasında ortaya çıkabilecek gürültü artışı ya da sinyal bozulması gibi problemler önlenebilir. Görüntüler iki boyuttan daha fazla boyutta tanımlanabildiğinden beri sayısal görüntü işleme çok boyutlu sistemler şekline modellenebilmektedir [39].

Çeşitli alanlarda oldukça yaygın bir kullanıma sahip olan bu teknoloji derin öğrenmenin getirdiği en büyük faydalardan biri olarak görülmektedir. Sağlıktan

fotoğrafçılığa, otonom araçlardan tarım alanına kadar yaygın kullanım alanları görebiliriz.

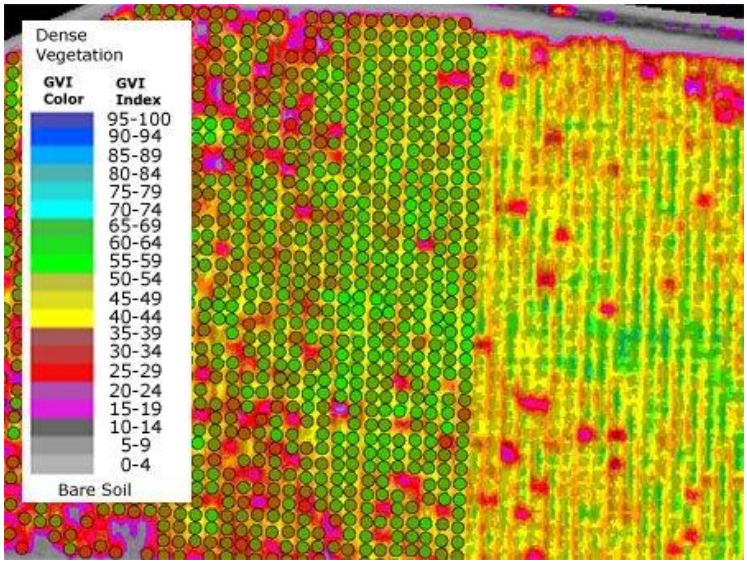
Özellikle tarım alanındaki faydaları bizim için önem taşımaktadır. Uydulardan, drone'lardan veya çeşitli hava araçlarından alınan ekin görüntüleri ile hasadı izlemek, yabancı ot oluşumunu tespit etmek veya besin eksikliğini belirlemek gibi farklı analizler elde edilebilir. Aynı zamanda hasadın yapıldığı arazi ve etrafındaki diğer arazilerin ne durumda olduğu da anlaşılabilir. Su geçiş güzergahlarındaki değişimler, sıcaklık haritaları veya erozyon durumları daha geniş ölçekten çekilen görüntülerin işlenmesiyle kontrol edilebilir.



Şekil 3.24: Uydu görüntüsü üzerinde ekim arazilerinin işlenmiş hali [40]

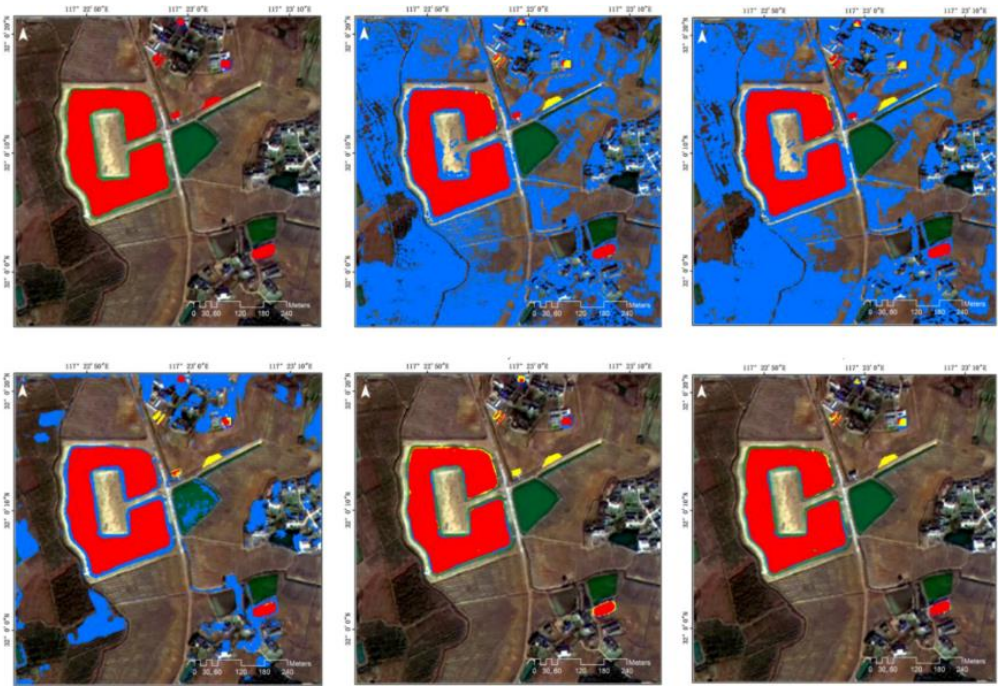
Yeşil bitki örtüsü indeks haritaları, çifçilere tarlalarındaki mevcut yeşil bitki örtüsü miktarının mutlak bir ölçümünü sağlar. Toprağın sağlığını ve nem ölçümünü belirler. Bu haritalar, mahsüllerini etkileyen girdilerin çevresel, fiziksel ve kültürel faktörlerin entegre sonucunu gösterir. Yeşil bitki örtüsü haritaları, keşif ve örnekleme yapacanızı, sorunlu alanların kapsamını ve ciddiyetini, etkilerini koruduğunuzu, kuraklık ve sel gibi olası sorunları çözmek için harekete geçeceğini gösteren önemli bir gelir artırıcı araçtır [40].

Yukarıda bulunan çalışma ‘Satellite Imaging Corporation’ kendi web sitesinden alınan bir örnektir. ‘NDVI Maps’ adını verdikleri yeşil bitki örtüsü haritaları ile uydu görüntüleri üzerinde neler yaptıklarını ve akıllı tarım açısından ne kadar büyük bir öneme sahip olduğunı açıkca göstermektedir. Kuveyt’te çekilen uydu görüntüsü üzerindeki mahsul verimini gösterdikleri diğer bir örnek alttaki görselde gösterilmiştir.



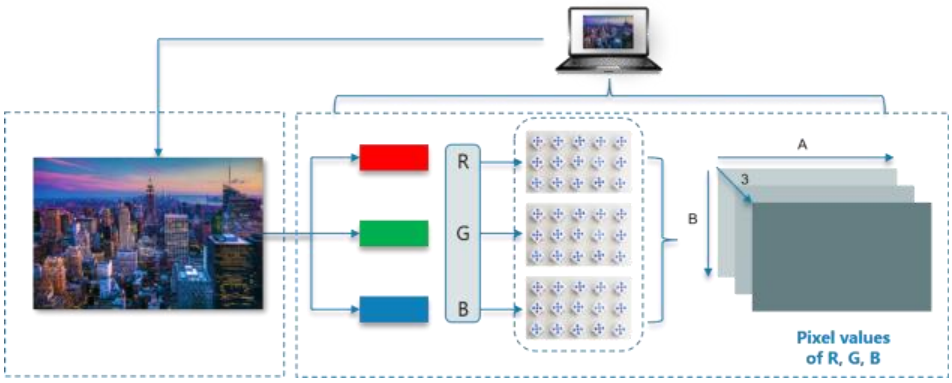
Şekil 3.25: Uydu görüntüsü üzerinde mahsul verimi için yapılan çalışma [40]

‘Image Fusion-Based Land Cover Change Detection Using Multi-Temporal High-Resolution Satellite Images [5]’ adlı diğer çalışmada bizim çalışmamıza benzer görüntü işleme ile arazilerin değişimleri incelenmiştir. Bu çalışma içerisinde ayrıca farklı mateatiksel formüller geliştirilerek alınan sonuçlar karşılaştırılmıştır.



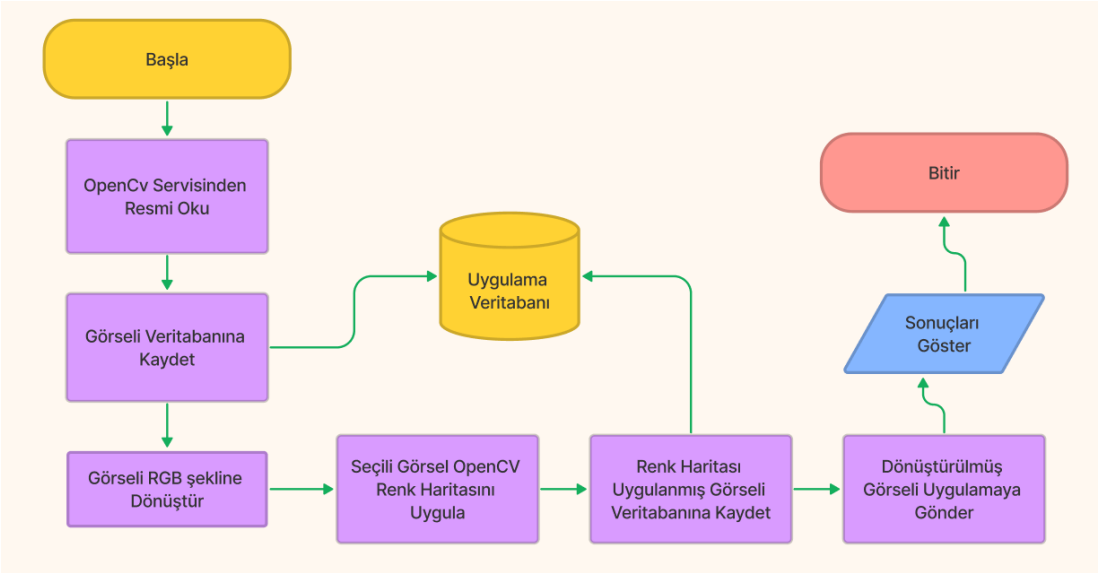
Şekil 3.26: Uydu görüntüsü üzerinde örnek bir çalışma [5]

Servis halinde sunduğumuz, arazi yapısının durumu hakkında bilgi sahibi olmamızı sağlayan yapının temelinde python programlama diline ait görüntü işleme kütüphaleri bulunmaktadır. Oldukça fazla seçeneğe sahip olan kütüphaler ile görüntü işleme üzerinde etkili sonuçlar alınabilmektedir. Bu kütüphane programala dili içerisinde 'OpenCV' ismi ile kullanılmaktadır.



Şekil 3.27: OpenCv kütüphanenesinin çalışma şekli [41]

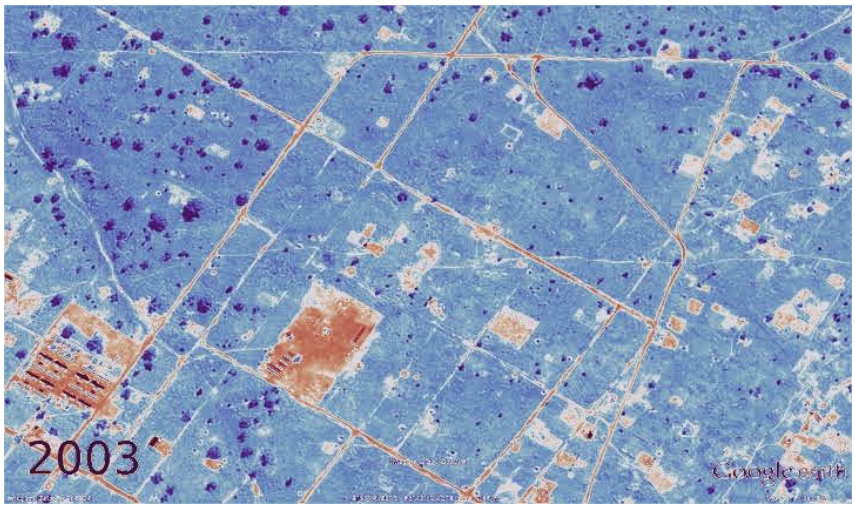
Kütüphaneyi kullanabilmek için python programlama dilinin çevre (environment) yapısını ve içerisinde modüllerin kurulması gerekmektedir. Ardından dökümantasyonda gösterildiği şekilde kullanımlar yapılabilir. Kodumuzun görüntü işleme sürecini daha net gösterebilmek adına akış diyagramı oluşturulmuştur.



Şekil 3.28: Görüntü işleme süreci ve kullanıcıya gönderme

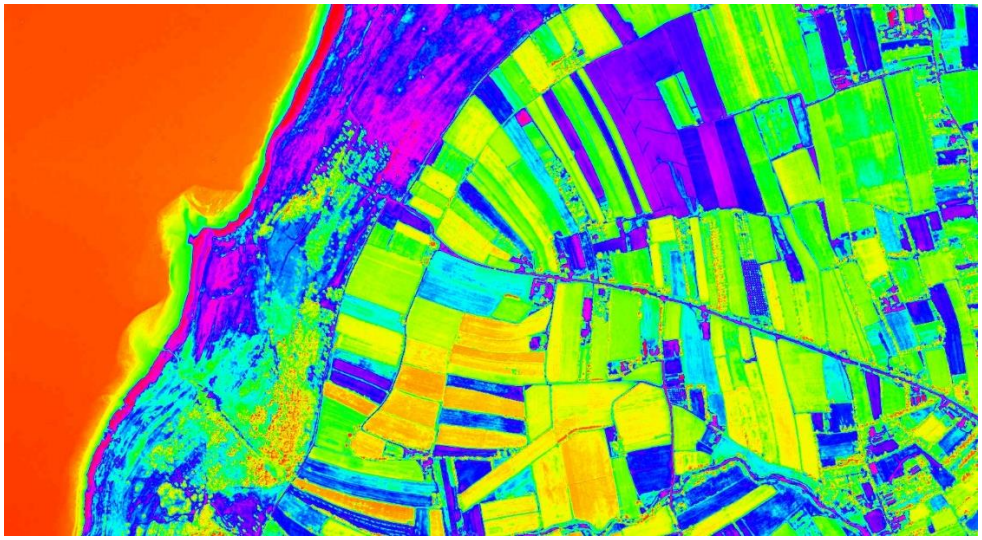
Kullanıcının uygulama üzerinden servise gönderdiği ve servis üzerinden kullanıcıya gönderilen görüntüler veritabanlarında saklanmaktadır. İlerleyen süreçlerde; yıllık hasat durumu izleme, dönemsel toprak yapısındaki ve bitki örtüsü üzerindeki değişimleri analiz edebilmek bu sistemle oldukça kolay hale getirişmiş olacaktır. Üretici kendi arazisinin yapısını dışarıya bağlı kalmadan daha kolay yönetilebilir alt yapıya sahip olacaktır.

Şekil 3.29: Üreticinin uygulama üzerinden gönderdiği görüntü

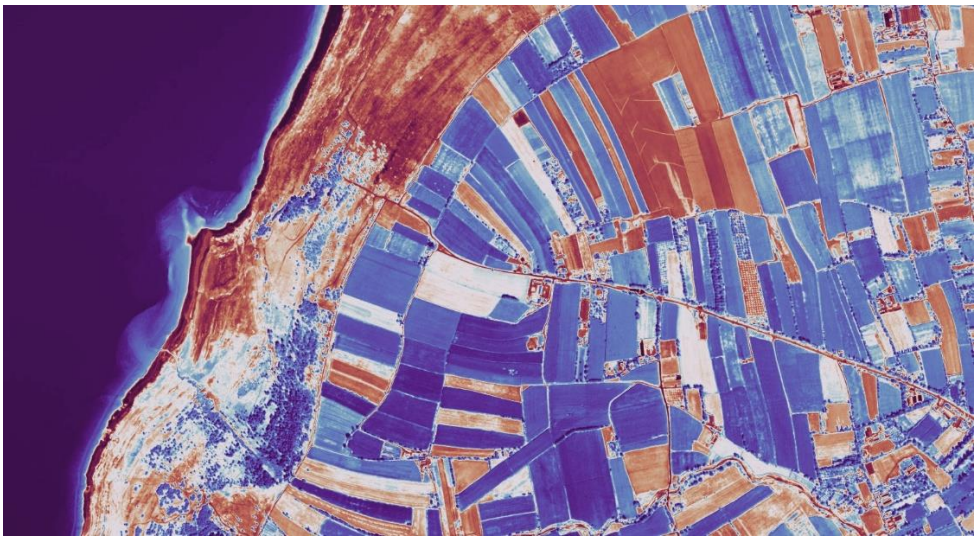


Şekil 3.30: Servisten gelen haritalanmış (mapping) görüntü

Şekil 3.31: Burdur Gölü yakınındaki arazilerin uydu görüntüsü



Şekil 3.32: Servisten gelen haritalanmış görüntü



Şekil 3.33: Servisten gelen farklı haritalanmış görüntü

Yukarıdaki Burdur Gölü üzerinden çekilmiş uydu görüntüsünün servise gönderildikten sonraki hali yani haritalanmış (mapping) görüntüyü incelersek, mahsül yetiştirilen arazilerin birbirinden farklı olduğunu ve bazı arazilerin tam verimli kullanılamadığını görürüz. Mahsülün üretildiği alanların orta kısımlarında veya tüm alanın susuzluktan yeterince gelişemediği sonucuna varabiliriz.

Bölüm 4

Gerçekleştirim

Çalışmanın teorik kısımları ele alındı. Bu kısımda canlı bir uygulamaya dönüştürülmesinin nasıl gerçekleştirildiği anlatılmaktadır.

4.1 Kullanılan Araçlar

4.1.1 Python

Python, nesne yönelimli, yorumlamalı, birimsel (modüler) ve etkileşimli yüksek seviyeli bir programlama dilidir.Girintilere dayalı basit söz dizimi, dilin öğrenilmesini ve akılda kalmasını kolaylaştırır. Bu da ona söz diziminin ayrıntıları ile vakit yitirmeden programlama yapılmaya başlanabilen bir dil olma özelliği kazandırır. Modüler yapısı, sınıf dizgesini (sistem) ve her türlü veri alanı girişini destekler. Hemen hemen her türlü platformda çalışabilir (Unix, Linux, Mac, Windows, Amiga, Symbian). Python ile sistem programlama, kullanıcı arabirimi programlama, ağ programlama, web programlama, uygulama ve veri tabanı yazılımı programlama gibi birçok alanda yazılım geliştirebilirsiniz. Büyük yazılımların hızlı bir şekilde prototiplerinin üretilmesi ve denenmesi gerektiği durumlarda da C ya da C++ gibi dillere tercih edilir [43].

Uygulamanın geliştirilmesinde oldukça önemli olan programlama dilidir. Servis mimarisi (backend), görüntü işleme ve yapay sinir ağlarının oluşturulması eğitilmesi ve test işlemlerinin her adımında kemik yapı olarak python programlama dili kullanılmıştır.

Python, esnek bir yapıya sahip olduğu için istatistik alanından, web programlamaya, oyun geliştirmeden derin öğrenme ve makine öğrenmesine kadar birçok alana hitap etmektedir. Uygulamamız içerisinde sıkça matris kütüphaneleri, dizi kütüphaneleri, resim kütüphaneleri ve sistem kütüphaneleri kullanılmıştır. Aynı zamanda web sunucuları oluşturmak için kullanılan FastAPI modülü ve SQLite3 veritabanı ile de entegre şekilde çalışmaktadır.

Python içerisindeki, Pytorch kütüphanesini kullanarak aktarımlı öğrenme ile geliştirilen yapay sinir ağlarının geliştirilmesi, eğitim yapılması, test işlemlerini ve tahmin işlemlerini kendi ekran kartlarımız üzerinde gerçekleştirdik.

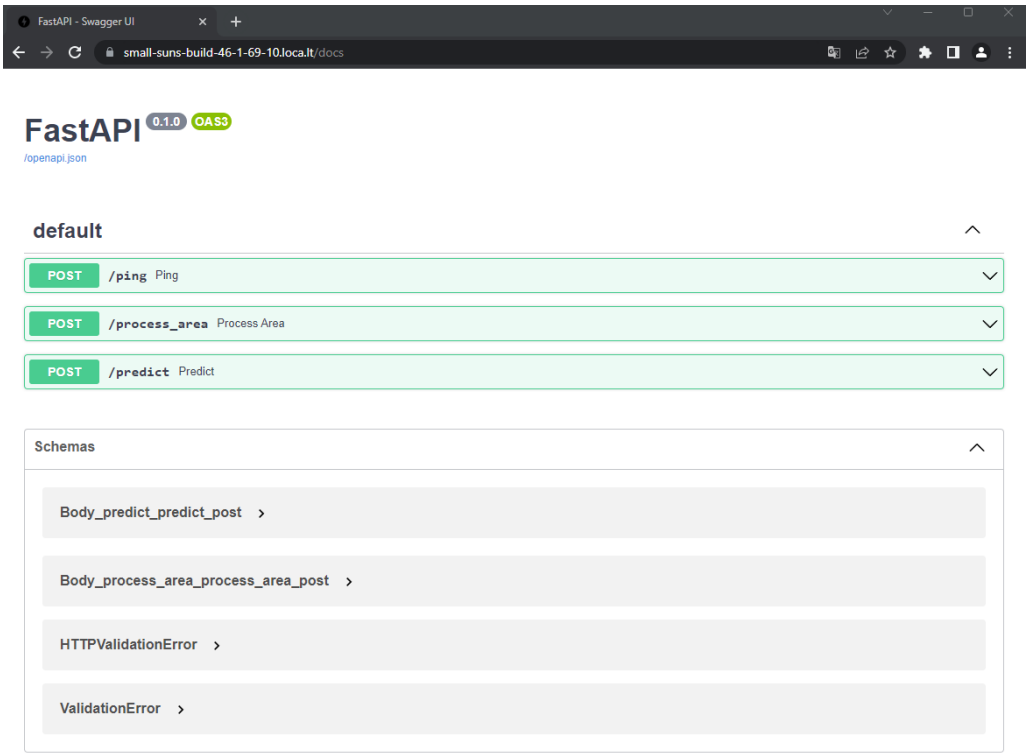
Bitki yaprakları üzerinden hastalık tespitini bir sınıflandırma problemi olarak Pytorch kütüphanesi ile yaptık. Tahminlerin daha kolay yapılabilmesi ve mobil arayüzden kolayca ulaşılabilmesi için geliştirdiğimiz en performanslı yapay sinir ağı modelini FastAPI servisi ile kullanıma açtık. Gelişim takibi süreci için de benzer süreç yürütülmüştür. Fakat gelişim takibi sürecinin en arkasında Python'ın Keras kütüphanesi bulunmaktadır. Keras ve Pytorch yapay sinir ağları konusunda gerçek uygulamalarda ve akademik çalışmalarda yüksek oranda tercih edilmektedir.

4.1.1.1 FastAPI

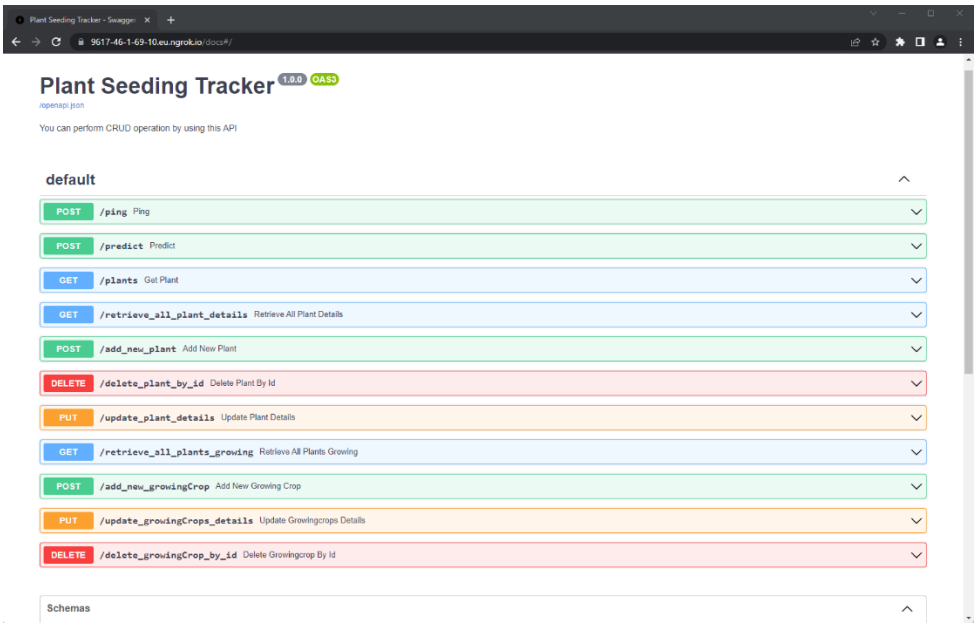
FastAPI, standart Python türü ipuçlarına dayalı 'Python 3.7' versiyonu ve daha fazlası ile 'Uygulama Programlama Arayüzü (API)' oluşturmaya yönelik modern, hızlı (yüksek performanslı) bir web çerçevesidir [54].

Temel olarak; yüksek performansa sahip olması, hızlı kodlanabilir olması, insan kaynaklı hata oluşma ihtimalinin düşük olması, her yerde yazılabilir ve derlenebilir olması, kullanımının ve öğreniminin kolay olması, standartlara dayalı olması (otomatik dokümante edilebilirlik), json schema yapısına uyumlu olması gibi çok önemli özelliklere sahip olması tercih edilebilirliğini yüksek tutmaktadır.

Mobil arayüze FastAPI servisleri ile bağlanarak veri iletişimi yapılmıştır. İstek-cevap tepki süreleri oldukça düşük olduğu için son kullanıcının farkedilebileceği bir yapıdadır.



Şekil 4.1: Python FastAPI hastalık tespiti servisinden görüntü



Şekil 4.2: Python FastAPI envanter sistemi ve gelişim takibi servisinden görüntü

Derin öğrenme modellerinin geliştirilmesi tamamlandıktan sonra mobil arayüzden erişilebilir olmalıdır. Mobil arayüz farklı bir programlama dili ile yazıldığı için doğrudan erişim mümkün değildir. Bu yüzden ‘Uygulama Programlama Arayüzü (API)’ne ihtiyaç vardır. Bu noktada FastAPI’yi devreye katarak ‘Temsili Aurum

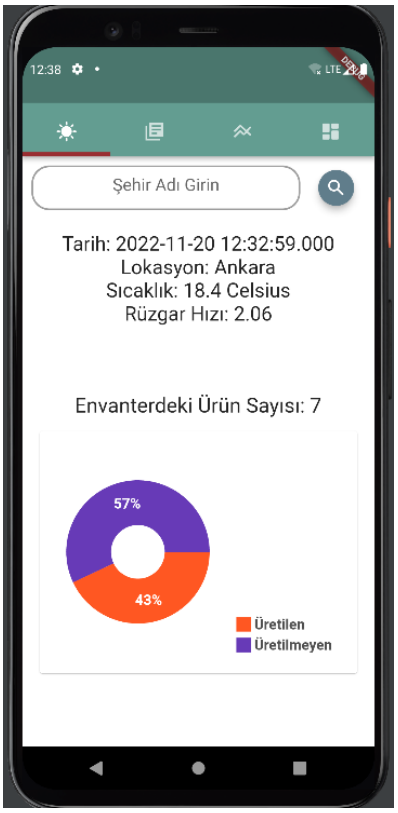
Aktarımı Mimarisi (Representational State Transfer Architecture) (REST)' mimari yaklaşımı ile servislerimizi oluşturduk. Pytorch ve Keras kütüphaneleri ile oluşturduğumuz modelleri servislerin içerisine koyarak 'Tekdüzen Kaynak Bulucuları (URL)'den yani tarayıcıdan erişebilir hale getirdik. Geriye mobil uygulama içerisinden URL'e istek atarak cevap almak kaldı. Tüm istek-cevap akışı sırasında servislerimiz SQLite3 veritabanına bağlı şekilde kayıtları saklamaktadır.

4.1.2 Flutter

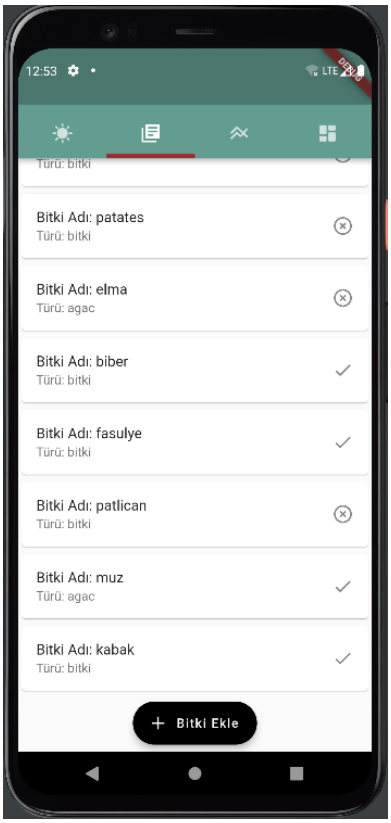
Flutter, tek bir kod tabanından yerel olarak derlenmiş, çok platformlu uygulamalar oluşturmak için 'Google'ın açık kaynak kodlu bir çerçevesidir.

Kullanıcının doğrudan erişim sağlayacağı kısımı mobil cihazlarda çalışabilir şekilde tasarladık. Kullandığımız 'Flutter' çerçevesi bize tek kod yazarak hem 'Android' hem de 'IOS' platformlara uygulama çıkarmamıza imkan verdiği için asıl tercih sebebimizdir. Aynı zamanda programlama dili olarak 'Dart' programlama dilini kullanmaktadır. Nesne yönelimli olan bu programlama dili oldukça esnek yapılar sunarak temiz kod yapısına yöneltmektedir. Diğer çapraz platform dillerine nazaran (React-Native olabilir) daha hızlı derleme sürelerine sahip ve yüksek performans için yerel (native) yürütme (build) tasarımını barındırmaktadır.

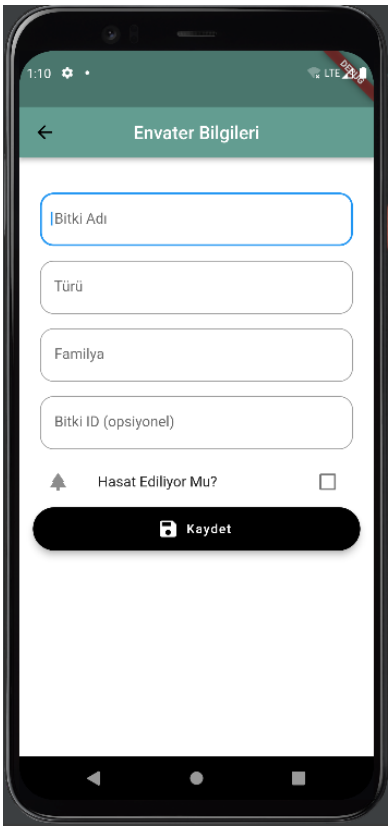
Uygulamamızda kullanıcı, bulunduğu konuma göre hava durumunu, üretimini yaptığı veya yapmadığı mahsüller için envanter yapısını ve o mahsüller için hastalık tespitini, envantere mahsül ekleme çıkarma, ekili olan mahsül ve olmayanların oranlarını, üretilen mahsüllerin büyüme oranların takibini yapabilmekte, ekim yapılan araziyi veya diğer arazilerin durumlarını görebilmek için kullanabileceği servislere erişebilmektedir.



Şekil 4.3: Mobil uygulama anasayfa hava durumu ve envanter durumu pasta grafiği



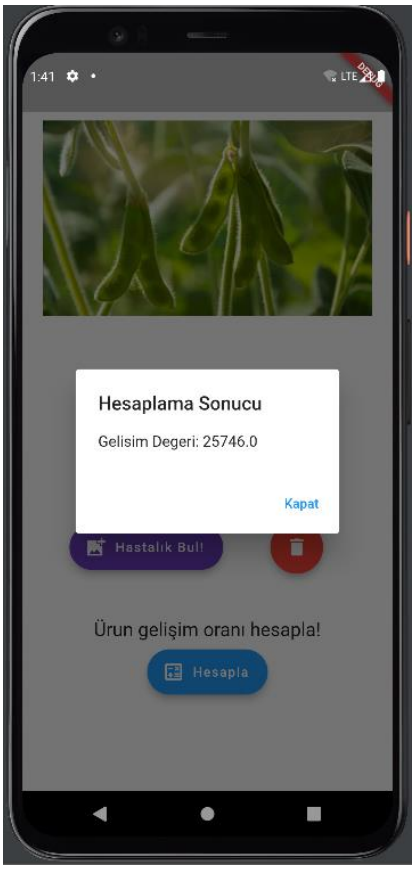
Şekil 4.4: Mobil uygulama envanter sayfası



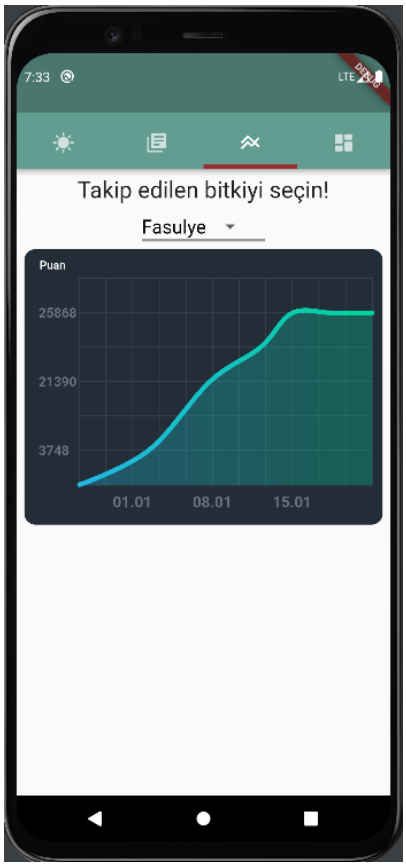
Şekil 4.5: Mobil uygulama envantere bitki ekleme sayfası

Şekil 4.6: Mobil uygulama bitki detay sayfası

Şekil 4.7: Mobil uygulama örnek mısır verisi ile hastalık tahmini



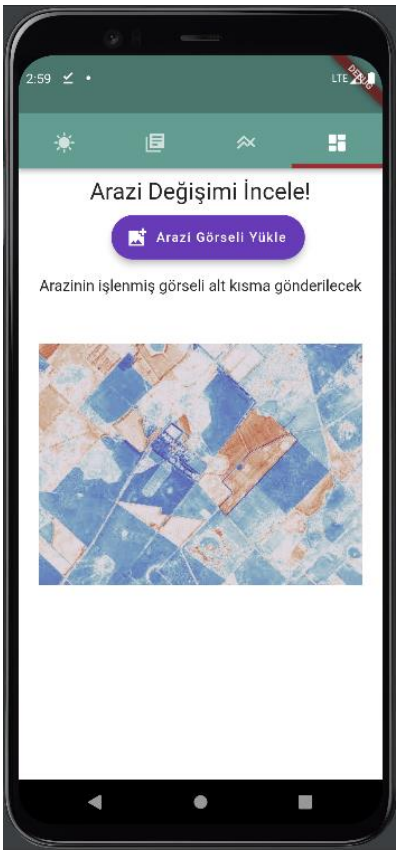
Şekil 4.8: Mobil uygulama örnek fasulye bitkisi verisi ile gelişim hesaplama



Şekil 4.9: Mobil uygulama fasulye bitkisi gelişim grafiği

Şekil 4.10: Mobil uygulama arazi değişim tespit sayfası

Şekil 4.11: Mobil uygulama örnek arazi verisi ile değişimi görüntüleme



Şekil 4.12: Mobil uygulama farklı metod kullanarak arazinin değişimini görüntüleme

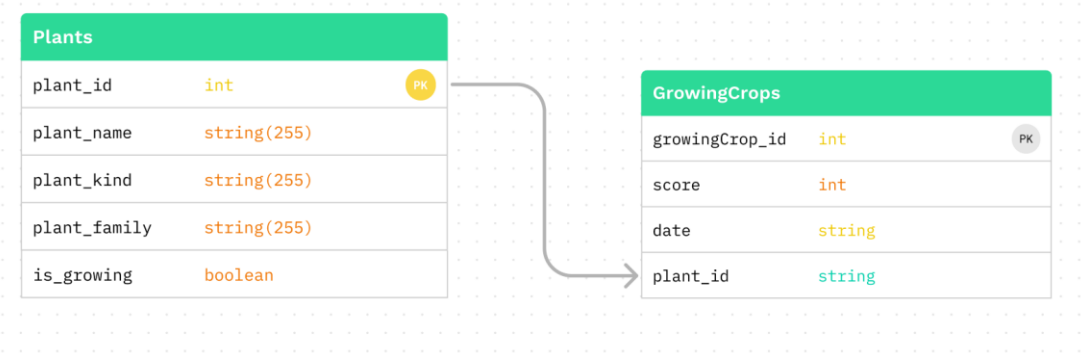
Şekil 4.13: Mobil uygulama farklı metod ve farklı arazi verisi kullanarak değişimini görüntüleme

4.1.3 SQLite

SQLite, dünyada en çok dağıtılan ve tavsiye edilen kaynak kodları halka açık, tamamen C/C++ programlama dilleriyle geliştirilmiş sunucu yazılımı ve yapılandırma gereksinimi olmayan, işlemsel ve ilişkisel bir SQL veritabanı motorudur [53].

SQLite3 veritabanı, python programlama dili ile oldukça kolay entegrasyonlara sahiptir. Ayrıca verilere hızlı erişim sağladığı için FastAPI servislerini (backend) bağlayarak verileri ilişkisel yapıda tuttuk.

Gelişim takibi için oluşturduğumuz ilişkisel SQLite3 veritabanından kesit:



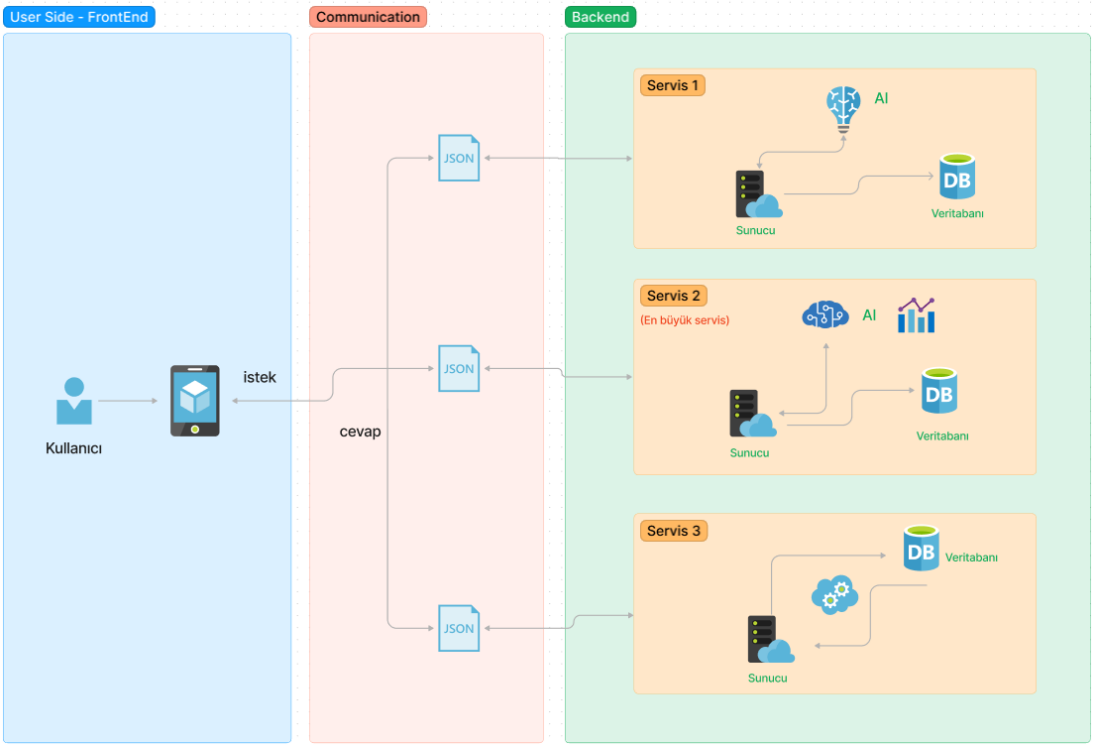
Şekil 4.14: Gelişim takibi ilişkisel veritabanı yapısı


```
{
  "plant_name": "patlican",
  "plant_kind": "bitki",
  "plant_family": "bitki",
  "is_growing": false,
  "plant_id": "p",
  "id": 6
},
{
  "plant_name": "muz",
  "plant_kind": "agac",
  "plant_family": "agac",
  "is_growing": true,
  "plant_id": "m",
  "id": 7
},
{
  "plant_name": "kabak",
  "plant_kind": "bitki",
  "plant_family": "Cucurbitaceae",
  "is_growing": true,
  "plant_id": "k",
  "id": 8
},
{
  "plant_name": "misir",
  "plant_kind": "bitki",
  "plant_family": "Poaceae",
  "is_growing": true,
  "plant_id": "mi",
  "id": 9
}
```


Şekil 4.15: Bitki envateri SQLite veritabanı json formatındaki örnek görüntü

4.2 Genel İşleyiş

Geliştirilen sistem ‘end-to-service’ yani kullanıcıdan servise şeklinde bir yapıda tasarlanmıştır. Üretici ihtiyacı olduğu durumda mobil cihazından giriş yaparak servislere ulaşabilmektedir.



Şekil 4.16: Sistemin mimari şeması

Sistemin genel işleyişine bakacak olursak; geliştirilen mobil uygulama, kullanıcıya çeşitli özellikleri sunmaktadır. Her bir özellik arka tarafta kendine özel mimarilere sahiptir. Kullanıcı; hava durumu, envanter ve envantere bağlı grafikleri görmek isterse sistem üzerinden ‘Servis 1’e istek atar. Tüm servisler REST mimarisinde tasarlandığı için haberleşme JSON formatında yapılmaktadır. Servis 1’e düşen istek sunucu üzerinden veritabanına erişerek envanter bilgilerini ya da hava durumu bilgilerini gönderir. Üretici; envanterdeki bir bitkinin yetiştirilip yetiştirilmemesi önemli olmamaksızın hastalığını öğrenmek isteyebilir. Hastalığını öğrenmek istediği bitkiye giderek tahmin işlemi için görseli telefondan yükler ve ‘Servis 1’e gönderilir. Gönderilen bu istek sunucu üzerinde yüklü olan, eğitimlerini yaptığımız derin öğrenme modelleri arasında en çok performansa sahip olan modele gider. Model gelen görüntüdeki hastalığın neler olabileceğini tahmin ederek 3 farklı tahmin yüzdesi ve

etiket çıkarır. Üretici uygulama üzerinde en güçlü tahmini ve diğerlerini görüntülüyebilir. Üretici; yetiştirmeye başladığı ürünün zaman içerisinde takibini yapmak ister ise uygulama üzerinden envanterindeki bitkiye giderek bitkinin anlık görselini yükler. Gelişim takibi süreç alacağı için 1 ay içerisinde düzenli periyotlarda görsel yüklemeye devam etmelidir. Yüklenen resimler 'Servis 2'ye gelerek veritabanında tutulur. Eğitimini tamamladığımız derin öğrenme modeli gelen görüntüler üzerinden farklı değerler elde ederek veritabanına kaydeder. Bu kaydedilen veriler mobil uygulama üzerinden gönderilen bir diğer istek sayesinde grafiksel olarak ekranda gösterilir. Eğer ki üretici, drone ile veya farklı hava araçlarıyla elde ettiği arazi görüntüsüne sahipse (düzenli olarak takip etmek için çoğu zaman arazi gözlem altında tutulur) 'Servis 3'e o görüntüyü göndererek işlenmiş halini geri sunucudan alabilir. Sunucu tarafında çalıştırılan görüntü işleme kütüphaneleri pythonda yazılmıştır. Birçok farklı işi tek bir uygulama üzerinden yapılabilmesi için karmaşık ama gerekli servisleri bir arada haberleşimini sağladık.

4.2.1 Hava Durumu Özelliği

Çifçilerin ürün yetiştirirken veya mahsül sonrası uygulamaları yaparken maruz kaldıkları en yüksek öncelikli faktör doğa olaylarıdır. Ülkemizde 4 farklı mevsim yaşandığı için güneşli, yağmurlu, karlı vs oldukça çeşitli hava olaylarına şahitlik ediyoruz. Mecburen doğa koşullarından kaçmak yerine barışık yaşamak zorundayız. Hava durumunu düzenli takip ederek yetiştirelecek mahsüllerin planlanması, ilaçlama, gübreleme gibi uygulamalar yapılmalıdır. Bu nedenle en çok ihtiyaç olabilecek bu özelliği mobil uygulamamız içerisine de entegre ettik. Arama çubuğundan bulunduğu bölgeye ait hava durumuna, sıcaklığa ve nem yüzdesi gibi değerlere erişebilmektedir. Dışardan servis hizmeti olarak açık kaynak kodlu şekilde kullanılabilmektedir.

Yukarıda bulunan 'Geocoding API' servisini kullanarak bulunulan bölgenin koordinatları (enlem-boylam) alınarak, Flutter içerisindeki 'Weather' paketi ile hava durumuna ve birçok parametreye de erişilmektedir. Entegrasyonlarda 'API Key' anahtarları kullanılmıştır.

4.2.2 Envanter Yapısı

SQLite3 veritabanı kullanarak çiftçinin o döenm içerinde üretimin yaptığı veya daha önceden yetiştirilen bitkileri kayıt altında tutulmaktadır. Üretici istediği zaman yeni bir bitkiyi sistemine ekleme çıkarma işlemini yapabilmektedir. Aynı zamanda o an içerisinde envanterindeki toplam ürünün yüzde kaçının üretimde olduğunu ya da üretimde olmadığını grafik üzerinde görebilmektedir.

4.2.3 Servis Yapısı

Python FastAPI çerçevesi kullanılarak oluşturulan servisler, derin öğrenme modellerinden en performanslı olanları ile entegre şekilde hazırlanmıştır. Tahmin işlemlerinin yapılabilmesi için gerekli haberleşme trafiği ve verilerin uygun formatlara dönüştürülmesi (işlemeye hazır hale getirilmesi) servis yapıları üzerinde yapılmıştır. Fazla yükler dağıtılarak modüler bir mimari tasarlamaya özen gösterilmiştir. Sadece veri transferinin yapıldığı kısımları değil aynı zamanda diğer işlemler için de modülerlik ön planda tutulmuştur.

4.2.4 Hastalık Tespiti Yapısı

Hastalık tespitinin uygulama içerisinde yapılması imkansız olduğu için kendi oluşturduğumuz servisler üzerinden gerçekleştirdik. Bitkinin sahip olduğu hastalığı öğrenmek için çiftçi öncelikle hastalıklı yaprağın görüntüsüne sahip olmalıdır. Eğer bulunmuyorsa mobil cihazın kamerası ile fotoğrafını çekerek cihazına kayıt edebilir. Uygulama üzerinden çektiği fotoğrafı ‘Hastalık Tespit Et’ butonuyla yükleyebilir. Servisten cevap dönmesinin ardından ekranda tahmin sonuçlarını görebilir. 3 farklı tahmin ve yüzdelik oranlarından yola çıkarak tedaviye başlayabilir.

4.2.5 Gelişim Takibi Yapısı

Üreticiler, düzenli olarak mahsülün sağlıklı gelişim gösterdiğinde emin olmak ister. Bunu yapabilmesi için ya sürekli izleyip takip etmesi gerekir ya da izlemesi için birkaç kişi görevlendirmesi gerekir. Arazilerin çok küçük olmadığı göz önünde bulundurulursa çok kolay ve ucuz olmayacağı ortadadır. Bunun yerine modern tarım

tekniklerinden kamera ile düzenli izlenmesi daha mantıklıdır. Aynı zamanda kameralardan gelen görüntülerin bir yapay zeka tarafından yorumlanıp gelişimlerine ait değerlerin elde edilmesi oldukça etkili bir çalışma haline gelmektedir. Biz mobil uygulama içerisine de bu özelliği koyarak durumu biraz daha kolaylaştırmak istedik. Üretici, düzenli aralıklar ile yetişen ürünün üstten görüntüsünü çekerek 'Servis 2'ye gönderir. Ardından derin öğrenme modeli tarafından ürününü kapladığı alan hesaplanarak ve veritabanına kayıt edilir. Veritabanında o bitkiye ait veriler topluca okunarak gelişim değerlerini servisten uygulamaya gönderilir.

4.2.6 Arazi Durum Tespiti Yapısı

Çiftçi, arazi durumunun değişimi, üretilen mahsulün arazide susuz kalmış veya beslenememiş kısımlarını, fazla gübreye veya ilaca maruz kısımlarını görebilmek için tüm araziyi dolaşması gerekmektedir. Fakat hava araçları ile görüntü elde ederek 'Servis 3'e gönderirse, tüm araziyi gezmek yerine tek bir fotoğraftan değişimleri izleyebilir. Servis 3 gelen görüntüleri işleyerek arazi üzerindeki değişimleri direk olarak çiftçiye gösterebilmektedir. Bu şekilde üretici paradan ve zamandan yüksek oranda kar elde etmiş olmaktadır.

Bölüm 5

Deneysel Sonuçlar ve Tartışma

5.1 Hastalık Tespiti Çıkarımlar

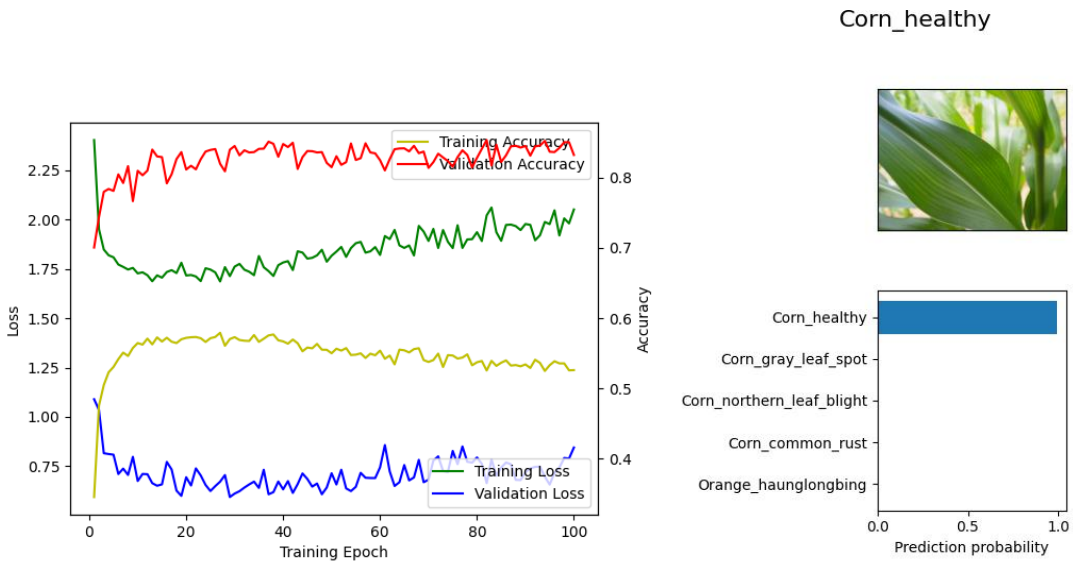
Genel olarak tüm sistemi göz önünde bulundurursak deneysel sonuçların daha çok derin öğrenme modellerin geliştirilmesinde ve iyileştirilmesi kısmında olduğunu söyleyebiliriz. Algoritmik yaklaşımlar, servis tasarımları ve kodlaması, mobil uygumanın tasarımı ve kodlanması değişkenlik gösterebilir. Ancak ilk önce derin öğrenme modelleri üzerinde yapılan çalışmalardan elde edilen sonuçları incelenmelidir.

Hastalık tespiti yapabilmek amacıyla aktarımlı öğrenme yöntemiyle geliştirilmiş derin öğrenme modellerini kullanılmıştı. Kendi topladığımız ve ön işleme aşamasından geçirdiğimiz verileri bu modelleri eğitmek için kullandık. Eğitilen modellerin her biri farklı doğruluk değerlerine sahip oldu. Aynı zamanda test görselleri de kullanılarak verdiği sonuçlar da aşağıda incelenecektir.

Tablo 5.1: Modellerden alınan sonuçlar

Model Adı	Eğitim Süresi	Toplam Tekrar Sayısı	En İyi Başarım Tekrarı	Başarım (Acc)
AlexNet	163m 24s	100	82	0.8535
DenseNet	925m 17s	100	71	0.8765
VGG16	367m 21s	100	100	0.9488
ResNet18	139m 1s	100	56	0.9289
EfficientNet B7	602m 50s	75	97	0.9248
EfficientNet B0	190m 20s	100	40	0.8852

Yukarda bulunan tablo detaylı olarak incelediğinde eğitim süreleri arasındaki farklar ve doğruluk oranları göze çarpmaktadır. Eğitim sürelerinin yüksek olması ciddi maliyeti de beraberinde getirmektedir. Her bir eğitimin tamamlanması ve başarım [92] değerini değerlendirerek yeniden eğitimlere katılması çok zaman almaktadır. Kendi bilgisayar donanımımız üzerinde (Nvidia GTX 1660 ekran kartı + AMD Ryzen 3500X işlemci) haftalar sürmüştür. 6 farklı modelin yukarıda görünen 'Accuracy' değerlerine yükseltilebilmesi için yaklaşık 7 haftalık süre harcanmıştır. Bazı modeller %90 üzerinde 'Accuracy' değerine ulaşamadığı için ekstra eğitimden geçmeleri gerekmektedir. Fakat çeşitli aktarımlı öğrenme modelleri deneyerek bu zaman farklı şekilde değerlendirilebilir. %90'ın üzerinde başarıma [92] sahip modellerin bile mobil uygulamadan gelen görüntülerin üzerindeki tahminleri bile yanılabilmektedir. Bu yüzden daha fazla kaliteli veri ile eğitime katılması gerekmektedir. Yine de elimizde ki bu modellerin aldıkları çıktılardan bazıları bizi mutlu etmeye yetmiştir. Alt kısımda test verisi ile modellerin nasıl sonuçlar verdiğini görebiliriz.

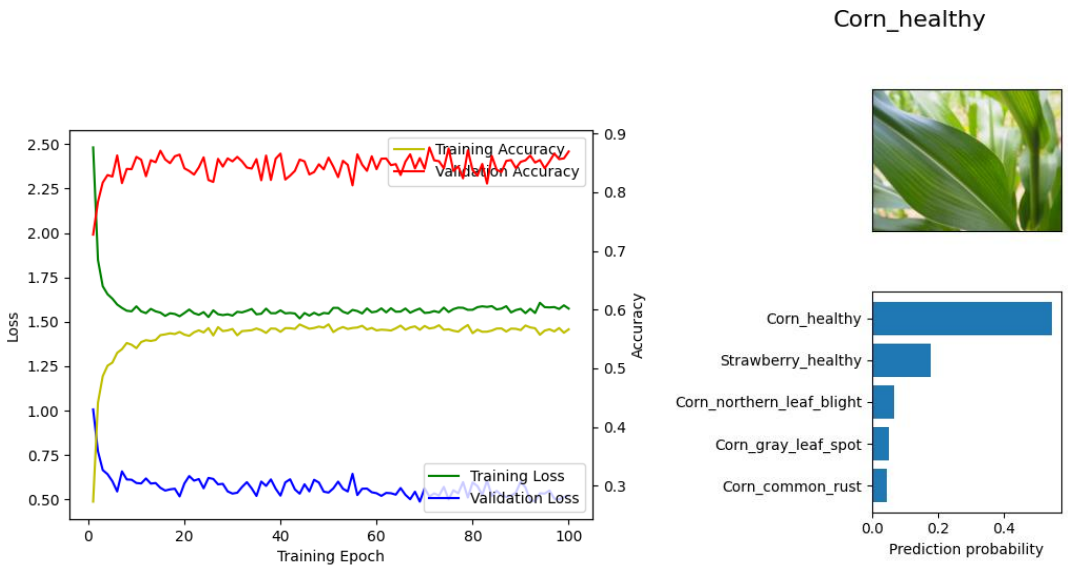


(a) (b)

Şekil 5.1: AlexNet eğitim ve tahmin görüntüleri, (a) Eğitim, (b) Tahmin

Tablo 5.2: AlexNet tahmin sonuçları

İlk Tahmin	Corn_healthy	0.9959
İkinci Tahmin	Corn_gray_leaf_spot	0.0020
Üçüncü Tahmin	Corn_nırthen_leaf_blight	0.0012
Dördüncü Tahmin	Corn_common_rust	0.0006
Beşinci Tahmin	Orange_hounglongbing	-4.698

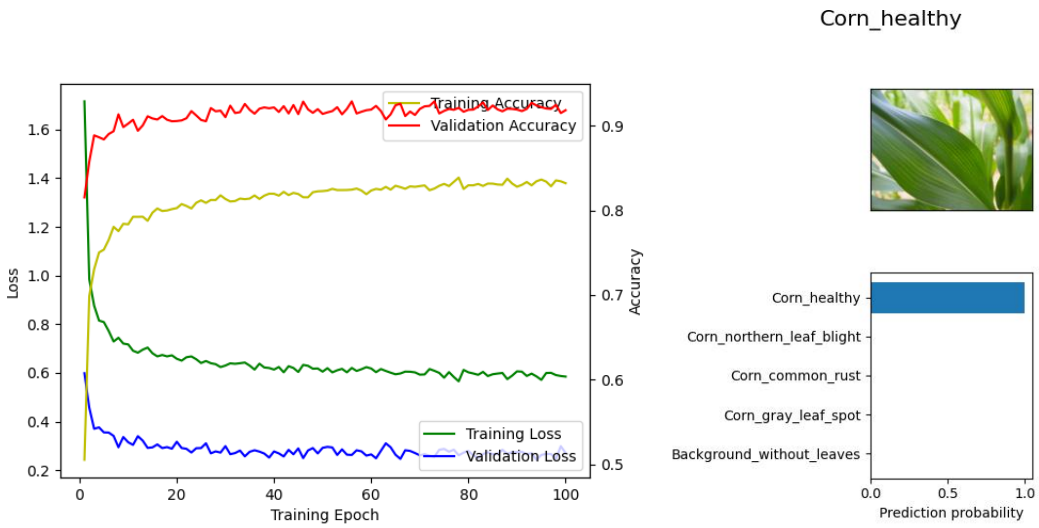


(a) (b)

Şekil 5.2: DenseNet161 eğitim ve tahmin görüntüleri, (a) Eğitim, (b) Tahmin

Tablo 5.3: DenseNet161 tahmin sonuçları

İlk Tahmin	Corn_healthy	0.5468
İkinci Tahmin	Strawberry_healthy	0.1775
Üçüncü Tahmin	Corn_northen_leaf_blight	0.0682
Dördüncü Tahmin	Corn_gray_leaf_spot	0.0505
Beşinci Tahmin	Corn_common_rust	0.0439

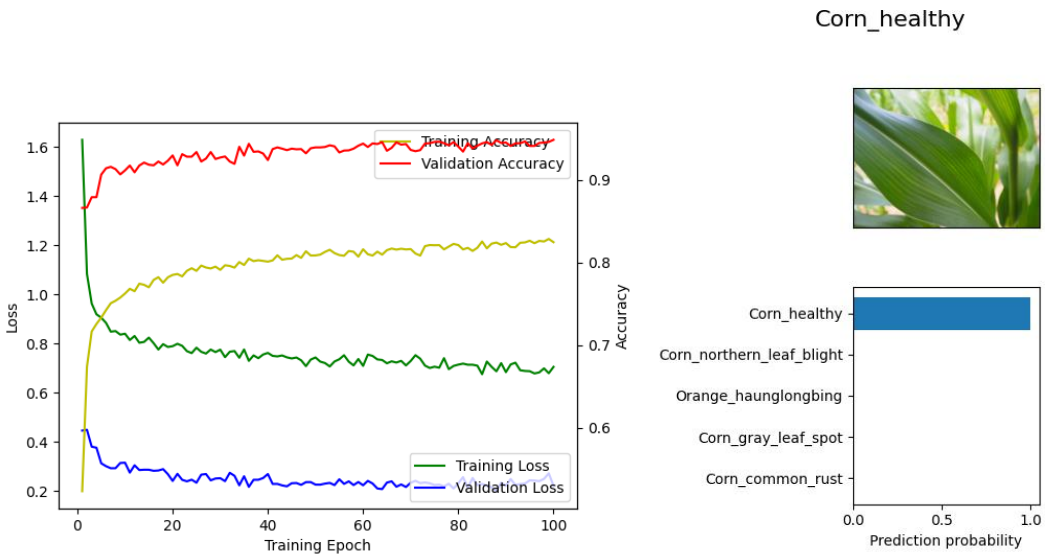


(a) (b)

Şekil 5.3: ResNet18 eğitim ve tahmin görüntüleri, (a) Eğitim, (b) Tahmin

Tablo 5.4: ResNet18 tahmin sonuçları

İlk Tahmin	Corn_healthy	0.9999
İkinci Tahmin	Corn_northen_leaf_blight	-6.7031
Üçüncü Tahmin	Corn_common_rust	-5.6159
Dördüncü Tahmin	Corn_gray_leaf_spot	-4.8736
Beşinci Tahmin	Background_without_leaves	-1.8715

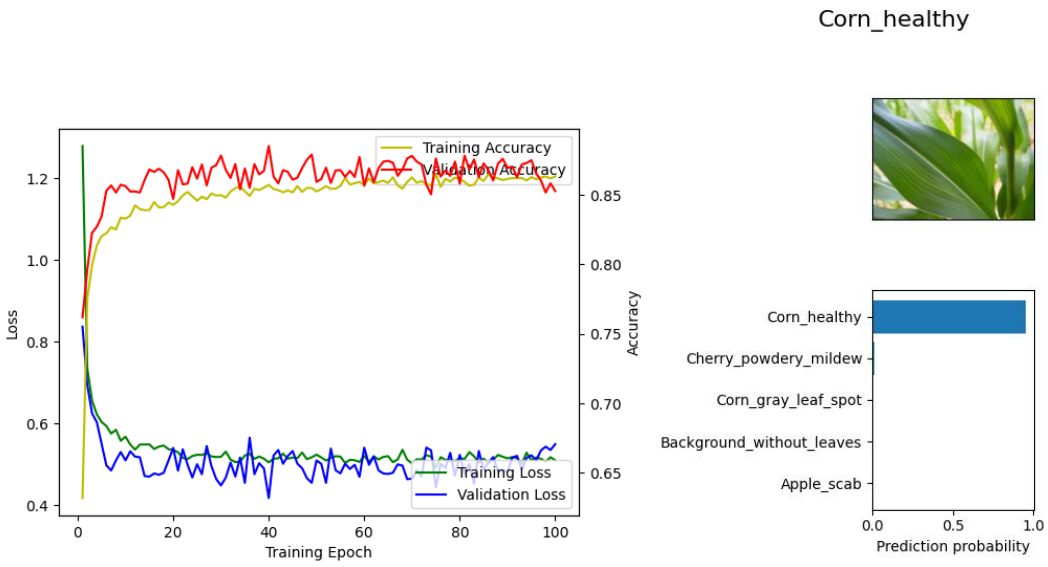


(a) (b)

Şekil 5.4: VGG16 eğitim ve tahmin görüntüleri, (a) Eğitim, (b) Tahmin

Tablo 5.5: VGG16 tahmin sonuçları

İlk Tahmin	Corn_healthy	1.0000
İkinci Tahmin	Corn_northen_leaf_blight	-2.0528
Üçüncü Tahmin	Orange_haunglongbing	-2.5728
Dördüncü Tahmin	Corn_gray_leaf_spot	-3.3931
Beşinci Tahmin	Corn_common_rust	-9.1961

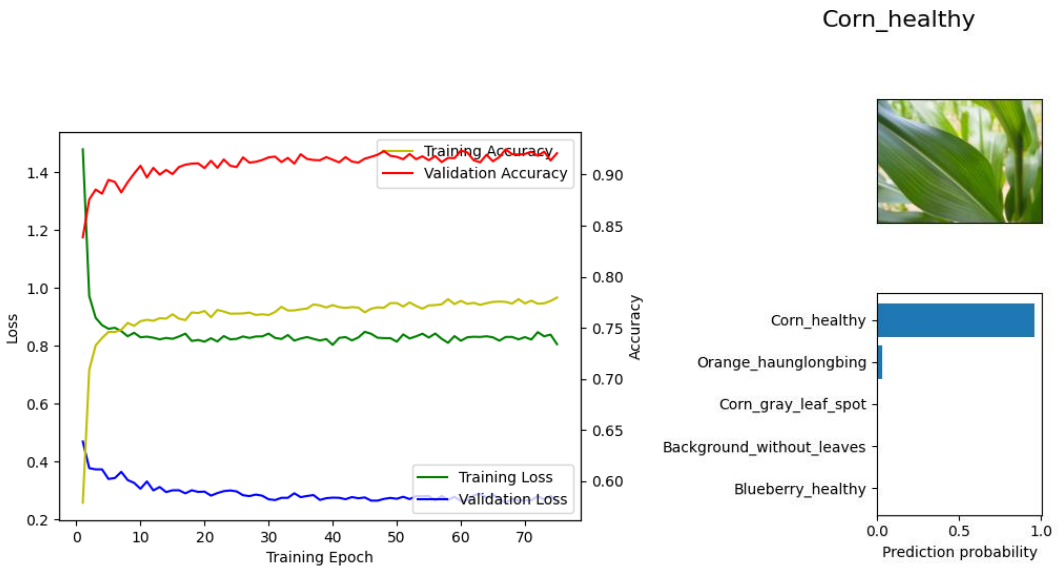


(a) (b)

Şekil 5.5: EfficientNetB0 eğitim ve tahmin görüntüleri, (a) Eğitim, (b) Tahmin

Tablo 5.6: EfficientNetB0 tahmin sonuçları

İlk Tahmin	Corn_healthy	0.9540
İkinci Tahmin	Cherry_powdery_mildew	0.0145
Üçüncü Tahmin	Corn_gray_leaf_spot	0.0058
Dördüncü Tahmin	Background_without_leaves	0.0046
Beşinci Tahmin	Apple_scab	0.0036



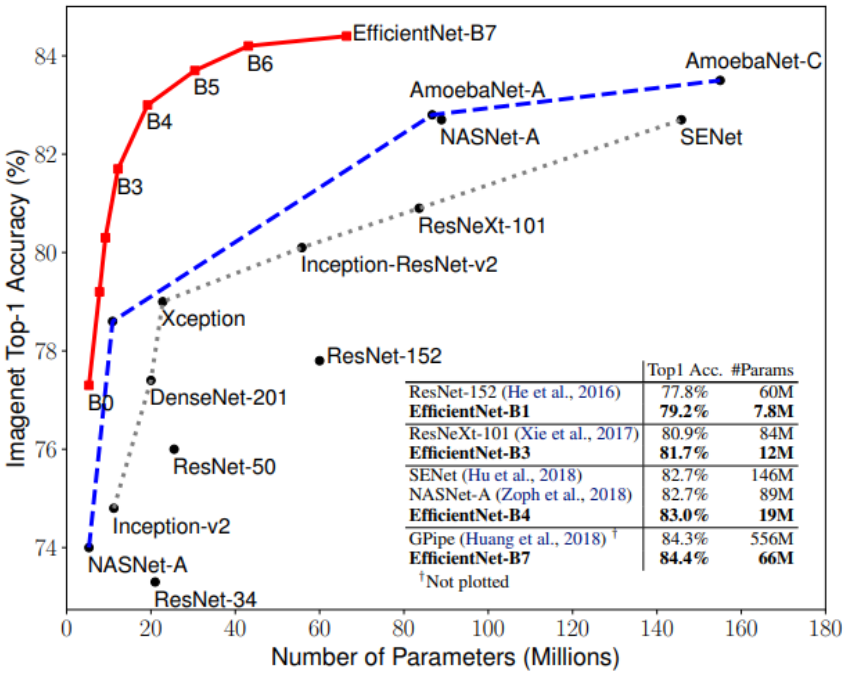
(a) (b)

Şekil 5.6: EfficientNetB7 eğitim ve tahmin görüntüleri, (a) Eğitim, (b) Tahmin

Tablo 5.7: EfficientNetB7 tahmin sonuçları

İlk Tahmin	Corn_healthy	0.9598
İkinci Tahmin	Orange_haunglongbing	0.0303
Üçüncü Tahmin	Corn_gray_leaf_spot	0.0031
Dördüncü Tahmin	Background_without_leaves	0.0021
Beşinci Tahmin	Blueberry_healthy	0.0019

Modellerin performans karşılaştırmalarına bakarsak EfficientNet serisinin (kırmızı çizgi ile gösterilen) özellikle EfficientNet-B7 modelinin diğer modellere göre daha çok performanslı olduğunu görebiliriz.



Şekil 5.7: Modellerin genel performans karşılaştırması [70]

Bitki yapraklarının hastalıklarını tespit etmek ve sınıfladırmak amacıyla oldukça fazla çalışma bulunabilir. Fakat henüz ticari amaçlı kullanılabilecek kadar etkili değildir. Üzerinde geliştirilmelere devam edilmesi gerekmektedir. Gelişimi doğrudan etkileyecek olan ise veri setinin çeşitliliği ve saysının artması olacaktır. Biz bu çalışmada Mendeley Plant Diseases Dataset'i kullandık.

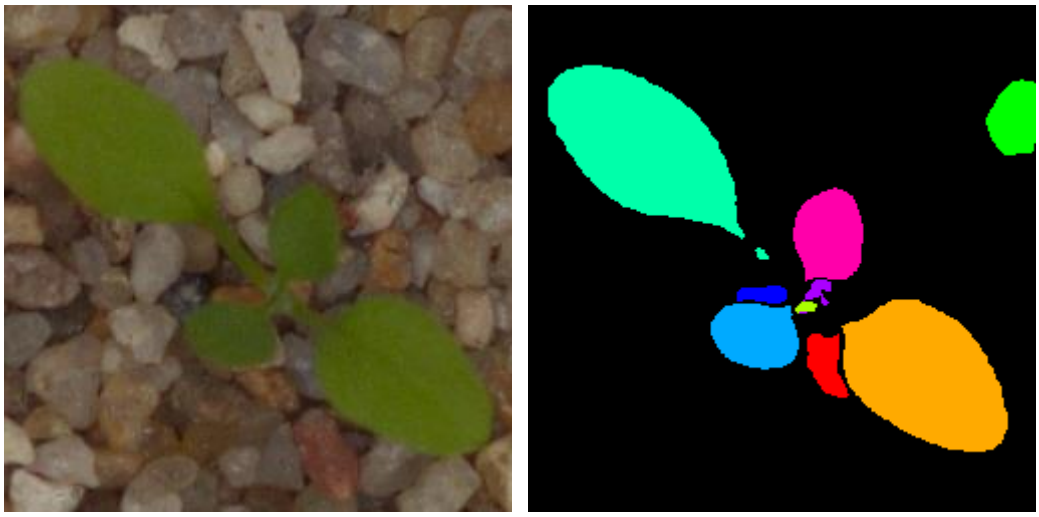
Bitki yapraklarındaki hastalıkların tespiti için farklı derin öğrenme teknikleri eğitilmiştir. Eğitimden sonra alınan sonuçlar arasında en iyi başarımı [92] veren %94.88 ile VGG16 mimarisi olmuştur. Süre olarak da DenseNet'e ve EfficientNetB0'a göre oldukça düşüktür. ResNet18 eğitim süresi olarak oldukça düşük ve başarım [92] olarak ortalamanın biraz üstünde bir değer çıkarmıştır. Üzerinde çalışılan bu modeller optimizasyon olarak gelişmiş donanımlı sistemlerde etkili olabilir ancak düşük donanımlı daha çok ihtiyaç olabilecek yerlerde etkili olmaayabilir. Düşük donanım gereksinimleri için diğer mimariler tercih edilebilir. İlk tercih sırasında MobileNetV2 gelebilir.

5.2 Gelişim Takibi Çıkarımlar

Gelişim takibi için kulllanılan MaskRCNN yapısı çok fazla probleme çözüm sunduğu için kullanılan verisetleri de o problemlere ait olacaktır. Mesela MSCOCO (Microsoft tarafından toplanan ve kullanıma açık) veriseti ile yapılan bir çalışmanın örnek çıktısına bakarsak:

Şekil 5.8: MaskRCNN modeli örnek çıktısı [52]

Geniş ölçekte çekilmiş olan fotoğraf üzerinde, nesneleri ve insanları yüksek doğruluk oranları ile maskelemiştir. Bizim problemimiz fotoğraf üzerinde bitki yapraklarının kapladığı alanları bulmak olduğu için ona uygun veriseti ile eğitim ve testler yapılarak geliştirildi. Aldığımız sonuçlar yeterli seviyede bulundu.



(b) (b)

Şekil 5.9: MaskRCNN gelişim takibi girdi ve çıktı örneği, (a) Girdi, (b) Çıktı.

(a) (b)

Şekil 5.10: MaskRCNN gelişim takibi girdi ve çıktı başka bir örneği, (a) Girdi, (b) Çıktı.

Tüm bu başarımla [92] yetinmek tabi ki de gerekli değildir. Çıktı fotoğrafları üzerinde de görüldüğü üzere bitki yaprağı olmayan bölgelerde de maskeleme işlemi yapılmıştır. Test fotoğrafı ve eğitim kalitesi bu durumun yaşanmaması için çok önemlidir. Başka bir çalışma için benzer bir işlem yapılacaksa mutlaka daha kaliteli ve fazla sayıda verinin olması avantaj sağlayacaktır.

5.3 Genel Mimari Tasarım Çıkarımlar

Sistemin planlamasından kodlamasına kadar birçok teknik denenmiştir. Mobil uygulama tarafından, servis yapısına kadar olabildiğince modülerlik göz önünde bulundurulmuştur. SOLID yazılım tasarım presiblerin hepsini uygulamaya yeterli zaman olmadığı için bazı kısımlar karmaşık hal almıştır. Bu tarz karmaşık durumların yaşanmaması için olabildiğince servis odaklı (service oriented architecture) yapılar ve SOLID yapıları göz önünde bulundurulmalıdır. Zaman içerisinde artan kod sayısı, bazen yapıları kurgularken fazladan işlemler yapmak zorunda bırakabiliyor. Düzenli olarak kodlar incelenerek yeniden düzenleme (refactor) yapılabilir. Kod karmaşıklığını azaltabilmek adına fazla efor sarfederek zaman ve enerji kaybettiğimiz durumlarla karşı karşıya kaldık. Yazılımın doğası gereği, sorunlar her zaman çıkabilir fakat işe hakim ve planlı ilerlemek bu sorunları minimum seviyeye indirebilmektedir.

Bölüm 6

Sonuç

Genel olarak modern tarım ve akıllı tarım uygulamaları üzerinde çalışmalar yapmak zaman içerisinde daha çok önem kazanacaktır. Artan insan nüfusu ve artan taleplerle, gıda zincirleri içerisinde geleneksel yöntemler ile yönetmek her geçen gün zorlaşmaktadır. Eldeki kaynaklar; su, toprak, bitki çeşitliliği vs korunarak tasarruf yapılabilir.

Teknolojik gelişmeleri, olabildiğince faydalı şekilde kullaranarak modern tarım için adımlar atılabilir. Maliyetleri düşürerek daha sistematik yapılar kurulabilir. Geleneksel tarım içerisindeki insan faktörünü olabildiğince dışarıda tutarak çok daha verimli sonuçlar elde edilebilir. Doğa olayları veya iklim değişikliği konularında maalesef daha dikkatli olmaktan başka seçenek bulunmamaktadır. Gereksiz tüketimler tespit edilerek otomasyona bağlanabilir. Böylece hem doğaya bırakılan atıklar azalabilir hem de zaman ve maliyetler düşebilir. Yaptığımız çalışmada ve diğer çalışmalarda benzer olarak ortak nokta; zamandan ve paradan tasarruf etmektir. Tarım yapılan arazilerde, seralarda veya daha büyük alanlarda, ilk başta teknolojik yatırım yapmak maaliyetli görünebilir. Fakat ileriye dönük planlamalar yapıldığında; kazançların yüksek olması ve daha sürdürülebilir yapının kurulması mümkündür. Yatırımlar için öncülük edecek olan kişiler tarafından akıllı tarım sistemlerinin ve yapay zekanın önemi kavranmış olmalıdır. Sadece kazanç odaklı düşünmek kötü sonuçları beraberinde getirebilir. Çünkü teknonolojik cihazlara sahip olarak normal şekillerde kullanmaktan ziyade; içerisinde yürütülen programların planlanması, geliştirilmesi ve bakımlarının da yapılması gerekecektir.

İçerik itibari ile yapay zeka karar destek sistemlerinin geliştirilmesi oldukça uzun bir süreçtir. Özellikle de hastalık tespiti veya gelişim takibi gibi konularda verisetlerinin toplanması, modellerin geliştirilmesi ve eğitilmesi yüksek maliyetli işler arasındadır.

Bulunulan bölgede yetiştirilen bitki çeşitliliği ve arazi yapısının farklılaşması bu süreci daha da arttırmaktadır. Bitki çeşidine göre görsellerin toplanması, manuel kameralar ile fotoğraflanması iş gücü gerektirmektedir. Yapay zeka modelinin eğitimleri sırasında harcanan zamanlar, donanımın gücüne bağlı olmakla beraber programcı da bilgi sahibi olmalıdır. Yetkinliği bulunmayan programcıların üretecekleri sinir ağı modelleri ile yanlış sonuçlar elde edeceği için maliyetler de artabilir.

Akıllı tarım için dikkat edilmesi gereken bir diğer bir konu ise tarım faaliyetleri yapan kişilerin yenilikçi teknolojiler ile interaktif çalışmabilmeleridir. Modern sistemlerin kurulmasından kullanımına kadar tüm süreç boyunca insan eli değmek zorundadır. Bu durumu çözebilmek için akıllı tarım ve modern tarım kavramları tüm çalışanlara anlatılmalıdır. Teorik kısımlarda bilgi sahibi olan çalışanların sahada uygulamalar yapabilmesi için gerekli deneysel ortamlar hazırlanmalıdır. Yeterli bilgi ve donanıma sahip kişiler kendileri akıllı tarıım süreçlerini yönetebilir hale gelecektir. Aynı zamanda bilgi sahibi kişiler bulundukları bölge içerisindeki değişken koşulları; hava, su, rüzgar, toprak yapısı vs değerlendirerek çıkan problemlere de çözüm üretebileceklerdir. Dünya genelinde de süreçler benzer şekilde yürütülmektedir. Fakat önceliğimiz kendi yaşadığımız coğrafyayı iyi anlamak olmalıdır. 4 mevsimin yaşandığı, farklı iklimlere sahip coğrafyada doğal olarak çeşitliliklere sahibiz. Toplumsal yaşantılarımız, geleneklerimiz ve dilimiz üzerinde bulunduğumuz topraklar ile şekillenmektedir. Çalışmları, ‘kendi kendimize yetebilir sistemler’ adı altında planlayarak (Singapur’da olduğu gibi) akıllı tarım uygulamaları ile öncülük edebiliriz. Üretim faaliyetlerinde yapay zeka sistemlerini dahil ederek karar destek sistemleri oluşturarak, lojistik ve nakliye işlemlerini otomatize ederek fazla maaliyetlerden kaçınarak, tüketiciye sunulan ürünlerden geri dönüşler alınarak tüm yapı güncellenebilir. Verim kelimesi sadece tarlalardaki ürünlerin üretiminde bir odak noktası olmamalıdır. Planlanan tüm süreçlerde ilk olarak verimin önemi düşünülerek ciddi atılımlar gerçekleştirilebilir.

Kaynaklar

[1] Ünal Z. Smart Farming Becomes Even Smarter With Deep Learning. IEEE Xplore 2020; doi.org:10.1109/ACCESS.2020.3000175

[2] Kathiva A. Deep Learning for Smart Agriculture. International Journal of Engineering Research & Technology (IJERT) 2021; 5(1): doi.org: 10.17577/IJERTCONV9lS05027

[3] O'Shaughnesy S, Kim M, Kim Y, Lee S, Kim H, Shekailo J. Towards Smart Farming Solutions in the U.S. and South Korea: A Comparison of the Current Status. ScienceDirect 2021; doi.org: 10.1016/j.geosus.2021.12.002

[4] Suganya E, Sountharrajan S, Shandilya S.K, Karthiga M. IOT in Agriculture Investigation on Plant Diseases and Nutrient Level Using Image Analysis Techniques. ScienceDirect 2019; doi.org: 10.1016/B978-0-12-817356-5.00007-3

[5] Wang B, Choi J, Choi S, Lee S, Wu P, Gao Y. Image Fusion-Based Land Cover Change Detection Using Multi-Temporal High-Resolution Satellite Images. Remote Sensing MDPI 2017; doi.org: 10.3390/rs9080804

[6] Khaki S, Wang L. Crop Yield Prediction Using Deep Neural Networks. Frontiers in Plant Science 2019; doi.org: 10.3389/fpls.201900621

[7] Kaiming H, Georgia G, Piotr D, Ross G, Facebook AI Research(FAIR). Mask R-CNN. IEEE Xplore 2017; doi.org: 10.1109/ICCV.2017.322

[8] Hassan M, Maji A. K, Jasinski M, Leonowicz Z, Jasinska E. Identification of Plant-Leaf Diseases Using CNN and Transfer-Learning Approach. MDPI 2021; doi.org: 10.33390/electronics10121388

[9] Yang Q, Shi L, Han J, Zha Y, Zhu P. Deep Convolutional Neural Networks for Rice Grain Yield Estimation at the Ripening Stage Using UAV-Based Remotely Sensed Images. ScienceDirect ELSEVIER 2019

[10] Zhou Y, Luo J, Feng L, Yang Y, Chen Y, Wu W. Long-Short-Term-Memory-Based Crop Classification Using High-Resolution Optical Images and Multi-Temporal SAR Data. GIScience & Remote Sensing 2019; doi.org: 10.1080/15481603.2019.1628412

[11] Samiei S, Rasti P, Vu L. J, Buitink J, Rousseau D. Deep Learning-Based Detection of Seedling Development. National Library of Medicine Pubmed 2020; doi.org: 10.1186/s13007-020-00647-9

[12] Mohanty P. S, Hughes D, Salathe M. Using Deep Learning for Image-Based Plant Disease Detection. National Library of Medicine 2016; doi.org:10.3389/fpls.2016.01419.

[13] Chen W. The Electrical Engineering Handbook. Digital Image Processing; 2005. 891-910.-13

[14] Mohammed E. S, Belal AA, Abd-Elmabod K. S, El-Shirbeny M, Gad A, Zahran B. M. Smart Farming for Improving Agricultural Management. ScienceDirect 2021; doi.org: 101016/j.ejrs.2021.08.007

[15] Doshi J, Patel T, Bharti K. S. Smart Farming using IOT, a Solution for optimally monitoring farming conditions. ScienceDirect ELSEVIER 2019: doi.org: 10.1016/j.procs.2019.11.016

[16] Walter A, Finger R, Huber R, Buchmann N. Smart farming is key to developing sustainable agriculture. OPINION 2017; doi.org: 10.1073/pnas.1007462114

[17] Giselsson T. M, Jergen N. R, Jensen K. R, Dyrmann, Midtiby S. H. A Public Image Database for Benchmark of Plant Seedling Classification Algorithms. ArXiv 2017; doi.org: 10.48550/arXiv.1711.05458

[18] Brahimi M, Said M, Kamel B, Moussaoui A. Deep Interpretable Architecture For Plant Diseases Classification. ArXiv 2019; doi.org: 10.48550/arXiv.1905.13523

[19] Hereda I. Large-Scale Plant Classification with Deep Neural Networks. ArXiv 2017.

[20] Cap H. Q, Uga H, Kagiwada S, Iyatomi H. LeafGAN: An Effective Data Augmentation Method for Practical Plant Disease Diagnosis. ArXiv 2020; doi.org: 1048550/arXiv.2002.10100

[21] Wolny A, Cerrone L, Vijayan A, Tofanelli R, Barro B. A, Louveaux M ve diğerleri. Accurate and versatile 3D segmnetation of plant tissuses at cellular resolution. ELife 2020; doi.org: 10.7554/eLife.57613

[22] Singh D, Jain N, Jain P, Kayal P, Kumawat S, Batra N. PlantDoc: A Dataset for Visual Plant Disease Detection. ArXiv 2019; doi.org: 10.48550/arXiv.1911.10317

[23] Lee H. S, Chan S. C, Wiikin P, Remagnino P. Deep-Plant: Plant Identification with convolutional neural networks. ArXiv 2015; doi.org: 10.48550/arXiv.1506.08425

[24] Sa I, Chen Z, Popovic M, Khanna R, Liebisch F, Nieto J ve diğerleri. WeedNet: Dense Semantic Weed Classification Using Multispectral Images and MAV for Smart Farming. ArXiv 2017; doi.org: 10.48550/arXiv.1709.03329

[25] Wang J. Z, Turko R, Shaikh O, Park H, Das N, Hohman F ve diğerleri. CNN Explainer: Learning Convolutional Neural Networks with Interactive Visualization. ArXiv 20; doi.org: 10.48550/arXiv.2004.15004

[26] Girshick R. Fast R-CNN. ArXiv 2015; doi.org: 10.48550/arXiv.1504.08083

[27] Beery S, Wu G, Rathod V, Votel R, Huang J. Context R-CNN: Long Term Temporal Context for Per-Camera Object Detection. ArXiv 2020; doi.org: 10.48550/arXiv.1912.03538

[28] Wang Z, Yan W, Oates T. Time Series Classification from Scratch with Deep Neural Networks: A Strong Baseline. ArXiv 2016; doi.org: 10.48550/arXiv.1611.06455

[29] Lai G, Chang W, Yang Y, Liu H. Modeling Long and Short-Term Temopral Patterns with Deep Neural Networks. ArXiv 2018; doi.org: 10.48550/arXiv.1703.07015

[30] Touvron H, Cord M, Sablayrolles A, Synnaeve G, Jegou H. Going deeper with Image Transformers. ArXiv 2021; doi.org: 10.48550/arXiv.2103.17239

[31] Xie Q, Dai Z, Hovny E, Luong M, Le V. Q. L. Unsupervised Data Augmentation for Consistency Training. ArXiv 2019; doi.org: 10.48550/arXiv.1904.12848

[32] Tan M, Le V. Q. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. ArXiv 2016; doi.org: 10.48550/arXiv.1905.11946

[33] Microsoft. Yapay zeka nedir? [İnternet]. 2022 [erişim tarihi 07.09.2022]. https://azure.microsoft.com/tr-tr/resources/cloud-computing-dictionary/what-is-artificial-intelligence/#benefits.

[34] IBM. Yapay zeka [İnternet]. 2022 [erişim tarihi 08.09.2022]. https://www.ibm.com/tr-tr/cloud/learn/what-is-artificial-intelligence.

[35] Innova. Bilgisayarlı Görü: Makineler Dünyayı Nasıl Görür? [İnternet]. 2022 [erişim tarihi 13.09.2022]. https://www.innova.com.tr/tr/blog/buyuk-veri-blog/bilgisayarli-goru-makineler-dunyayi-nasil-gorur.

[36] DeepAI. Convolutional Neural Network [İnternet]. 2022 [erişim tarihi 12.09.2022]. https://deepai.org/machine-learning-glossary-and-terms/convolutional-neural-network.

[37] IBM. Convolutional Neural Networks [İnternet]. 2022 [erişim tarihi 21.09.2022]. https://www.ibm.com/cloud/learn/convolutional-neural-networks.

[38] Mendeley Data. Data For: Identification of Plant Leaf Diseases Using a 9 Layer Deep Convolutional Neural Network [İnternet]. 2022 [erişim tarihi 21.09.2022]. https://data.mendeley.com/datasets/tywbtsjrjv.

[39] Microsoft. Görüntü İşleme nedir? [İnternet]. 2022 [erişim tarihi 20.09.2022]. https://azure.microsoft.com/tr-tr/resources/cloud-computing-dictionary/what-is-computer-vision/.

[40] Satellite Imaging Crop. Precision Agriculture Mapping [İnternet]. 2022 [erişim tarihi 21.09.2022]. https://www.satimagingcorp.com/applications/natural-resources/agriculture/.

[41] Medium. OpenCV Python Tutorial – Computer Vision With OpenCV In Python [İnternet]. 2022 [erişim tarihi 13.09.2022]. https://medium.com/edureka/python-opencv-tutorial-5549bd4940e3#:~:text=OpenCV%20Python%20is%20nothing%20but%20a%20wrapper%20class,integrate%20it%20with%20other%20libraries%20which%20use%20NumPy.

[42] Wikipedia. Yapay Zeka [İnternet]. 2022 [erişim tarihi 14.09.2022]. https://tr.wikipedia.org/wiki/Yapay_zek%C3%A2.

[43] Wikipedia. Python [İnternet]. 2022 [erişim tarihi 05.09.2022]. https://tr.wikipedia.org/wiki/Python.

[44] Medium. Region-Based Convolutional Neural Network (RCNN) [İnternet]. 2022 [erişim tarihi 24.09.2022]. https://medium.com/analytics-vidhya/region-based-convolutional-neural-network-rcnn-b68ada0db871.

[45] Viso. Everything about Mask R-CNN: A Beginner's Guide [İnternet]. 2022 [erişim tarihi 22.08.2022]. https://viso.ai/deep-learning/mask-r-cnn/.

[46] Wikipedia. AlexNet [İnternet]. 2022 [erişim tarihi 24.07.2022]. https://en.wikipedia.org/wiki/AlexNet.

[47] Paperswithcode. DenseNet [İnternet]. 2022 [erişim tarihi 01.08.2022]. https://paperswithcode.com/model/densenet?variant=densenet-161.

[48] Medium. ResNet (Residual Network) Nedir? [İnternet]. 2022 [erişim tarihi 14.07.2022]. https://suhedacilek.medium.com/resnet-residual-network-nedir-49105e642566.

[49] Medium. Derine Daha Derine: Evrişimli Sinir Ağları [İnternet]. 2022 [erişim tarihi 15.07.2022]. https://ayyucekizrak.medium.com/deri%CC%87ne-daha-deri%CC%87ne-evri%C5%9Fimli-sinir-a%C4%9Flar%C4%B1-2813a2c8b2a9#:~:text=VGG%2D16,1000%20s%C4%B1n%C4%B1fl%C4%B1%20softmax%20ba%C5%9Far%C4%B1m%C4%B1%20hesaplan%C4%B1r.

[50] Medium. Ölçeklendirme ile CNN Modellerinin Doğruluk ve Verimliliğini Artırma: EfficientNet [İnternet]. 2022 [erişim tarihi 19.08.2022]. https://ayyucekizrak.medium.com/%C3%B6l%C3%A7eklendirme-ile-cnn-modelinin-do%C4%9Fruluk-ve-verimlili%C4%9Fini-art%C4%B1rma-efficientnet-cb6f2b6512de.

[51] KDNuggets. Computer Vision in Agriculture [İnternet]. 2022 [erişim tarihi 18.08.2022]. https://www.kdnuggets.com/2021/09/computer-vision-agriculture.html.

[52] Github. Mask_RCNN [İnternet]. 2022 [erişim tarihi 16.08.2022]. https://github.com/matterport/Mask_RCNN.

[53] Wikipedia. SQLite [İnternet]. 2022 [erişim tarihi 05.06.2022]. https://tr.wikipedia.org/wiki/SQLite.

[54] FastAPI. FastAPI [İnternet]. 2022 [erişim tarihi 05.06.2022]. https://fastapi.tiangolo.com/.

[55] DTNPF. Prograssive Farmer [İnternet]. 2022 [erişim tarihi 16.06.2022]. https://www.dtnpf.com/agriculture/web/ag/crops/article/2018/07/11/five-steps-successful-plant-disease-2.

[56] Pinterest. Leafs [İnternet]. 2022 [erişim tarihi 16.06.2022]. https://www.pinterest.com.mx/pin/755619643706097135/.

[57] Pawan T. Impacts of Land Use and Climate Chance: A Study of Dolakha, Nepal. ResearchGate 2020.

[58] GoodHouseKeeping. How to Get Rid of Ants [İnternet]. 2022 [erişim tarihi 17.06.2022]. https://www.goodhousekeeping.com/home/a46670/how-to-get-rid-of-ants/.

[59] Radio Crafts. Smart Irrigation. [İnternet]. 2022 [erişim tarihi 09.09.2022]. https://radiocrafts.com/applications/smart-irrigation/.

[60] WeedBot. How does the laser weeding work? [İnternet]. 2022 [erişim tarihi 09.09.2022]. https://weedbot.eu/weedbot-technology/.

[61] Getactiveabc. Growing Skills Programme [İnternet]. 2022 [erişim tarihi 02.07.2022]. https://getactiveabc.com/events/growing-skills-programme-2/.

[62] Mayralt. Crop Yield Prediction Dataset [İnternet]. 2022 [erişim tarihi 18.06.2022]. https://mayralt.blogspot.com/2021/05/crop-yield-prediction-dataset-github.html.

[63] Blog CSDN. Classification [İnternet]. 2022 [erişim tarihi 19.06.2022]. https://blog.csdn.net/u013250416/article/details/81281798.

[64] MarkTechPost. Understanding The Concept of Convolutional Neural Networks (CNNs) [İnternet]. 2022 [erişim tarihi 16.06.2022]. https://www.marktechpost.com/2022/01/24/a-detailed-understanding-of-convolutional-neural-networks/.

[65] Medium. Introducing Transfer Learning as Your Next Engine to Drice Future Innovations [İnternet]. 2022 [erişim tarihi 11.06.2022]. https://medium.datadriveninvestor.com/introducing-transfer-learning-as-your-next-engine-to-drive-future-innovations-5e81a15bb567.

[66] Medium. The Architecture & Implementation of AlexNet [İnternet]. 2022 [erişim tarihi 26.07.2022]. https://medium.com/analytics-vidhya/the-architecture-implementation-of-alexnet-135810a3370.

[67] AI Pool. DenseNet [İnternet]. 2022 [erişim tarihi 24.07.2022]. https://ai-pool.com/m/densenet-1568742493.

[68] Neurohive. VGG16 – Convolutional Network for Classification and Detection [İnternet]. 2022 [erişim tarihi 22.07.2022]. https://neurohive.io/en/popular-networks/vgg16/.

[69] Blog CSDN. Resnet [İnternet]. 2022 [erişim tarihi 15.07.2022]. https://blog.csdn.net/weixin_26752765/article/details/108132640.

[70] Papers. RCNN [İnternet]. 2022 [erişim tarihi 12.07.2022]. https://papers.readthedocs.io/en/latest/imagedetection/rcnn/.

[71] Awesome Open Source. Efficientdet PyTorch [İnternet]. 2022 [erişim tarihi 12.07.2022]. https://awesomeopensource.com/project/romulus0914/EfficientDet-PyTorch.

[72] SAFEF. Singapore's Farmers: Going High-Tech to Boost Our Food Security [İnternet]. 2020 [erişim tarihi 08.09.2022]. https://safef.org.sg/blog/singapores-farmers-going-high-tech-to-boost-our-food-security/.

[73] The Conversation. We throw away a third of the food we grow – here's what to do about waste [İnternet]. 2020 [erişim tarihi 08.09.2022]. https://theconversation.com/we-throw-away-a-third-of-the-food-we-grow-heres-what-to-do-about-waste-165997.

[74] Barbados Today. Six barriers to digital inclusion [İnternet]. 2020 [erişim tarihi 08.09.2022]. https://barbadostoday.bb/2021/09/14/btcolumn-six-barriers-to-digital-inclusion/.

[75] Concaveagri. Smart Farming [İnternet]. 2020 [erişim tarihi 15.09.2022]. https://concaveagri.com/smart-farming/.

[76] Cheng X, Zhang Y, Chen Y, Wu Y, Yue Y. Pest identification via deep residual learning in complex background. ScienceDirect 2017; doi.org: 10.1016/j.compag.2017.08.005

[77] Ding W, Taylor G. Automatic moth detection from trap images for pest management. ScienceDirect 2016; doi.org:10.1016/j.compag.2016.02.003

[78] Zhu L. Q, Ma M. Y, Zhang Z, Zhang Y. P, Wu W, Wang D. D. Hybrid deep learning for automated lepidopteran insect image classification. ResearchGate 2016; doi.org: 10.1080/00305316.2016.1252805

[79] Alzu'bi S, Hawashin B, Mujahed M, Jararweh Y, Grupta B. B. An efficient employment of internet of multimedia things in smart and future agriculture. ResearchGate 2019; doi.org:10.1007/s11042-0197367-0

[80] Song X, Ganlin Z, Liu F, Li C. D, Zhao Y, Yang J. Modeling spatio-temporal distribution of soil moisture by deep learning-based cellular automata model. ResearchGate 2016; doi.org: 10.1007/s40333-016-0046-0

[81] Durai S. K. S, Shamili D. M. Smart farming using Machine Learning and Deep Learning techniques. ScienceDirect 2022; doi.org: 10.1016/j.dajour.2022.100041

[82] Martins S. V, Roy P. D, Huang H, Boschetti L, Zhang K. H, Yan L. Deep learning high resolution burned area mapping by transfer learning from Landsat-8 to PlanetScope. Remote Sensing of Environment 2022; doi.org:10.1016/j.rse.2022.113203

[83] Gislason O. P, Benediktsson A. J, Sveinsson R. J. Random Forests for land cover classification. Pattern Recognition Letters 2006; doi.org:10.1016/j.patrec.2005.08.001

[84] Wu M, Yue H, Wang J, Haung Y, Liu M, Jian Y. Object detection based on RGC mask R-CNN. IET Journals 2020; doi.org: 10.1049/iet-ipr.2019.0057

[85] Qiao Y, Truman M, Sukkariech S. Cattle segmentation and contour extraction based on Mask R-CNN for precision livestock farming. ScienceDirect Computers and Electronics in Agriculture 2019; doi.org: 10.1016/j.compag.2019.104958

[86] Machefer M, Lemarchand F, Bonnefond V, Hitchins A, Sidiropoulos P. Mask R-CNN Refitting Strategy for Plant Counting and Sizing in UAV Imagery. MDPI Remote Sensing 2020; doi.org: 10.3390/rs12183015

[87] Krizhevsky A, Sutskeve I, Hinton E. G. ImageNet Classification with Deep Convolutional Neural Networks. Communications of the ACM 2017; doi.org: 10.1145/3065386

[88] Han X, Zhong Y, Cao L, Zhang L. Pre-Trained AlexNet Architecture with Pyramid Pooling and Supervision for High Spatial Resolution Remote Sensing Image Scene Classification. MDPI Remote Sensing 2017; doi.org: 10.3390/rs9080848

[89] Huang G, Liu Z, Maaten V. D. L, Weinberger Q. K. Densely Connected Convolutional Networks. ArXiv 2016; doi.org: 10.48558/arVix.1608.06993

[90] Medium. Step by Step VGG16 implementation in Keras for Beginners [İnternet]. 2019. [erişim tarihi 22.10.2022]. https://towardsdatascience.com/step-by-step-vgg16-implementation-in-keras-for-beginners-a833c686ae6c.

[91] Medium. Deep Residual Learning for Image Recognition (ResNet Paper explained) [İnternet]. [erişim tarihi 25.10.2022]. https://medium.com/analytics-vidhya/deep-residual-learning-for-image-recognition-resnet-paper-explained-26b29e0fa73e.

[92] Niyaz Q, Sun W, Javaid Y A, Alam M. A Deep Learning Approach for Network Instrusion Detection System. BICT 2015; doi.org: 10.4105/eai.3-12-2015.2262516.

[93] Esen F. A., Onan A. Derin Öğrenme Yöntemleri ile Bitki Yaprakları Üzerindeki Hastalıkların Sınıflandırılması. Avrupa Bilim Dergisi Dergipark 2022; doi.org: 10.23590/ejosat.1181081.

Özgeçmiş

Adı Soyadı: Furkan Alp Esen

E-mail (1): y200237007@ogr.ikcu.edu.tr

E-mail (2): furkanalpesen@gmail.com

Eğitim:

2016–2020 Manisa Celal Bayar Üniversitesi, Yazılım Müh. Bölümü

2021–2023 İzmir Kâtip Çelebi Üniversitesi, Yazılım Müh.

İş Deneyimi:

2019 – 2019 Tirad Information Services

2020 – 2021 Codeo Engineering A.Ş

2021 – 2022 AloTech a Cloud Company & Call Center Studio A.Ş

Yayınlar:

1. Derin Öğrenme Yöntemleri ile Bitki Yaprakları Üzerindeki Hastalıkların Sınıflandırıması