



T.C.
KIRŞEHİR AHİ EVRAN ÜNİVERSİTESİ
MÜHENDİSLİK - MİMARLIK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ

**GÖRÜNTÜ İŞLEME VE DERİN ÖĞRENME
TABANLI AKILLI TARIM UYGULAMASI**

BERKAY YEŞİLDERE

LİSANS TEZİ

DANIŞMAN
Dr.Öğr.Üyesi Volkan Güneş

KIRŞEHİR / 2025-2026 AKADEMİK YILI

İÇİNDEKİLER

Sayfa No

İÇİNDEKİLER.....	1
ŞEKİL LİSTESİ.....	2
TABLO LİSTESİ.....	3
SİMGE VE KISALTMA LİSTESİ.....	4
ÖZET.....	5
1. GİRİŞ.....	6
1.1. Amaç.....	7
1.2. Önem.....	8
2. İLGİLİ ÇALIŞMALAR.....	8
3. ÖNERİLEN SİSTEM.....	15
3.1. Veri Setleri.....	15
3.2. Model Mimarisi.....	17
3.3 Eğitim Süreci.....	19
3.4 Model Dönüşümü.....	20
3.5 Giriş Görüntüsü Arka Plan Kaldırma.....	21
3.6 Mobil Uygulama.....	24
4. BULGULAR.....	31
4.1 Model Eğitimi ve Performans Sonuçları.....	31
4.2 Sınıflandırma Performansı ve Karışıklık Matrisi.....	32
4.3 Mobil Uygulama Performans Testleri.....	33
5. TARTIŞMA VE SONUÇLAR.....	33
5.1 Tartışma.....	
5.2 Sonuç.....	
KAYNAKLAR.....	36

ŞEKİL LİSTESİ

	Sayfa No
Şekil 2.1. Önceden eğitilmiş bilgiden yararlanan modelin çalışma prensibi.	8
Şekil 2.2. Çalışmada kullanılan hibrit öğrenme modelinin akış şeması.	9
Şekil 2.3. Modellerinin karar mekanizmalarını karşılaştırın açıklanabilirlik ısı haritaları.	11
Şekil 2.4. Nesne tespiti modeli ile bitki hastalıklarının tespiti ve yerinin belirlenmesi.	14
Şekil 3.2. PlantVillage veri setinden örnek sağlıklı ve hastalıklı yaprak görüntüleri.	18
Şekil 3.3. Önerilen transfer öğrenme tabanlı model mimarisı.	20
Şekil 3.4. Eğitim ve doğrulama setleri üzerindeki Loss ve Accuracy değişim grafikleri.	21
Şekil 3.5. 38 farklı sınıf için oluşturulan Karışıklık Matrisi Koyu renkli köşegenler, modelin doğru sınıflandırma başarısını göstermektedir.	22
Şekil 3.6. Modelin mobil platforma uygun hale getirilmesi için uygulanan dönüşüm adımları	23
Şekil 3.7. U ² -Net mimarisinin genel yapısı. Her bir En ve De bloğu, kendi içinde bir RSU yapısı barındırmaktadır.	23
Şekil 3.8 Arka plan kaldırma işlemi örneği. (a) Orijinal görüntü, (b) U ² -Net-P tarafından üretilen maske, (c) Arka planı kaldırılmış sonuç görüntüsü.	24
Şekil 3.9 Hastalık teşhis ekran görüntüleri. Görüntü seçim ekranı, Analiz süreci, Teşhis sonucu ve tedavi önerileri.	27
Şekil 3.10 Bahçe yönetimi ekran görüntüleri. (a) Bahçem listesi, (b) Bitki detay sayfası.	28
Şekil 3.11 Yapay zeka sohbet asistanı ekran görüntüsü.	29
Şekil 3.12 Uygulama ana ekran görüntüleri. (a) Teşhislerim sekmesi, (b) Bahçem	30
Şekil 4.1 Eğitim ve Doğrulama Kayıp/Doğruluk Grafiği.	31
Şekil 4.2 Karışıklık Matrisi (Confusion Matrix).	32

TABLO LİSTESİ

	Sayfa No
Tablo 2.1 Çalışmalarda kullanılan yöntemler ve performansları.	14
Tablo 3.1 Sınıflandırma adları ve örnek sayısı	15
Tablo 3.2 Eğitim sürecinde kullanılan temel hiperparametreler.	20
Tablo 3.3 Mobil uygulamada kullanılan temel teknolojiler ve kütüphaneler.	24
Tablo 4.1 Modelin Eğitim, Doğrulama ve Test Performans Değerleri	31

SİMGİ VE KISALTMA LİSTESİ

Kısaltmalar Açıklama

AI	: Artificial Intelligence
FAO	: Food and Agriculture Organization
CNN	: Convolutional Neural Network
DNN	: Deep Neural Network
VGG16	: Visual Geometry Group 16-layer
SVM	: Support Vector Machines
KNN	: K-Nearest Neighbor
MÖ	: Makine Öğrenmesi
LR	: Learning Rate
ReLU	: Rectified Linear Unit
RGB	: Red Green Blue
GPU	: Graphics Processing Unit
CUDA	: Compute Unified Device Architecture
ONNX	: Open Neural Network Exchange

ÖZET

LİSANS TEZİ

GÖRÜNTÜ İŞLEME VE DERİN ÖĞRENME TABANLI AKILLI TARIM UYGULAMASI

BERKAY YEŞİLDERE

**Kırşehir Ahi Evran Üniversitesi
Mühendislik Mimarlık Fakültesi
Bilgisayar Mühendisliği Bölümü**

Danışman: Dr.Öğr.Üyesi Volkan Güneş

Tarım, dünya nüfusunun beslenmesi için hayatı bir öneme sahiptir. Ancak bitki hastalıkları, her yıl küresel ölçekte büyük ürün kayıplarına yol açarak gıda güvenliğini tehdit etmektedir. Geleneksel yöntemlerle hastalıkların teşhisini genellikle uzmanlık gerektiren, zaman alıcı ve maliyetli bir süreçtir. Ayrıca, yanlış teşhisler sonucu gereksiz veya hatalı pestisit kullanımı hem çevreye zarar vermeye hem de çiftçilerin maliyetlerini artırmaktadır. Bu projenin temel motivasyonu, derin öğrenme ve görüntü işleme teknolojilerinden yararlanarak, mobil cihazlar üzerinden herkesin kolayca kullanabileceği bir akıllı tarım asistanı geliştirmektir. Bu sayede, bitki hastalıklarının erken ve doğru bir şekilde teşhis edilmesi, doğru tedavi yöntemlerinin önerilmesi ve verimliliğin artırılması hedeflenmektedir. Geliştirilecek olan bu uygulama ile tarımsal üretimin daha sürdürülebilir, verimli ve çevre dostu bir hale getirilmesine katkı sağlanması amaçlanmaktadır.

Anahtar Kelimeler: Derin Öğrenme, Görüntü İşleme, Bitki Hastalıkları Tanıma

1. GİRİŞ

Tarım, küresel gıda güvenliğinin sağlanması ve dünya ekonomilerinin sürdürülebilirliği için hayatı bir role sahiptir. Ancak hızla artan dünya nüfusunun gıda talebini karşılama hedefi, ciddi zorluklarla yüzleşmektedir. Bu zorlukların başında, tarımsal verimliliği büyük ölçüde düşüren bitki hastalıkları gelmektedir. Birleşmiş Milletler Gıda ve Tarım Örgütü (FAO), her yıl bitki hastalıkları ve zararlıları nedeniyle küresel ürün rekoltesinin %40'a varan oranlarda kaybedildiğini, bu durumun hem gıda güvenliğini tehdit ettiğini hem de yıllık 220 milyar doları aşan ekonomik kayıplara yol açtığını bildirmektedir [1,2]. Geleneksel teşhis yöntemlerinin yavaş, maliyetli ve uzmanlık gerektirmesi, özellikle kaynaklara erişimi kısıtlı olan küçük çiftçiler için büyük bir engel teşkil etmektedir. Bu sorunlara çözüm olarak, "Akıllı Tarım" veya "Tarım 4.0" olarak bilinen teknolojik devrim, yapay zekâ (AI) ve görüntü işleme gibi yenilikçi yaklaşımlar sunmaktadır. Özellikle Evrişimli Sinir Ağları (CNN) gibi derin öğrenme tabanlı akıllı sistemler, tarımsal görüntü analizinde devrim yaratmıştır. Mobil cihazların yaygınlaşmasıyla birlikte, bu güçlü yapay zekâ modelleri artık doğrudan çiftçinin kullanımına sunulabilmektedir. Akıllı telefon kameraları aracılığıyla çekilen bir bitki yaprağı fotoğrafı, eğitilmiş bir model tarafından saniyeler içinde analiz edilerek hastalık teşhisini konulabilmekte ve anında müdahale imkânı sağlanabilmektedir. Bu çalışmanın temel amacı, söz konusu teknolojik potansiyeli kullanarak çiftçiler ve bitki yetiştircileri için pratik ve erişilebilir bir çözüm sunmaktır. Bu doğrultuda, mobil cihazlar aracılığıyla çekilen bitki fotoğraflarını derin öğrenme algoritmalarıyla analiz ederek hastalıkları yüksek doğrulukla teşhis eden akıllı bir mobil uygulama geliştirilmesi hedeflenmektedir. Proje, yalnızca hastalık tespiti ile sınırlı kalmayıp, aynı zamanda toprak tipi analizi ve bu veriler ışığında kullanıcıya özel mahsul ve gübreleme önerileri sunarak kapsamlı bir tarım danışmanlığı aracı olmayı amaçlamaktadır.

1.1. Amaç

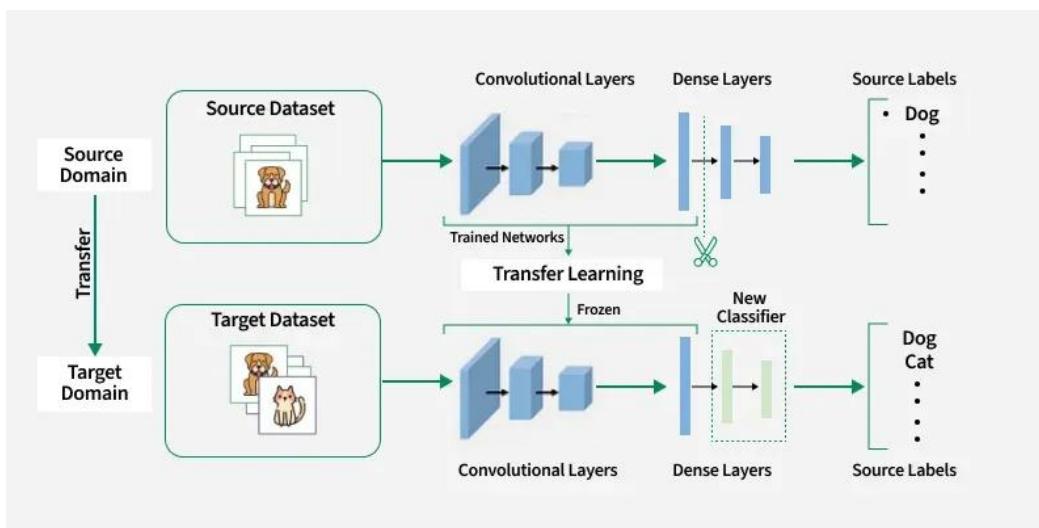
Bitki hastalıkları, tarımsal üretimde ciddi verim kayıplarına neden olmakta, bu durum gıda güvenliğini ve çiftçilerin gelirini tehdit etmektedir. Bu sebeple, hastalıkları erken evrede, doğru bir şekilde teşhis edebilen ve anında çözüm önerileri sunan akıllı bir sisteme ihtiyaç duyulmaktadır. Yapılan bu uygulama, mobil cihazlar aracılığıyla çekilen bitki yaprağı ve toprak fotoğraflarını, görüntü işleme ve derin öğrenme algoritmalarından biri olan evrişimsel sinir ağları ile kullanarak bitki hastalığının gerçek zamanlı olarak tespit edilmesini ve bu verilere dayalı olarak kullanıcı için tedavi ve mahsul önerilerinin sunulmasını amaçlamaktadır. Bu sayede, hastalıkların yol açtığı ürün kayıplarının önüne geçilerek tarımsal verimliliğin artırılması ve çiftçilerin daha bilinçli ve sürdürülebilir tarım yapmaları hedeflenmektedir.

1.2. Önem

Bitki hastalıklarının zamanında ve doğru bir şekilde teşhis edilememesi, tarımsal üretimde büyük verim kayıplarına ve dolayısıyla çiftçiler için ciddi ekonomik zararlara yol açmaktadır. Yapılan bu çalışma, mobil cihazlar aracılığıyla bitki hastalıklarının ve toprak türünün anında tespit edilip, çiftçinin doğru adımları atması için bilgilendirilmesi işlevini gerçekleştirmektedir. Hastalıkların erken evrede ve doğru bir şekilde tespit edilip uygun müdahalenin yapılması, ürün kayıplarını en aza indirmek ve tarımsal sürdürülebilirliği sağlamak için oldukça önemlidir. Uygulamada yer alan akıllı öneri sistemi, doğru teşhise dayalı olarak çalışacak, çiftçinin yanlış veya gereksiz pestisit kullanımını engelleyerek hem maliyetleri düşürecek hem de çevre kirliliğinin azaltılmasına katkı sağlayacaktır.

2. İLGİLİ ÇALIŞMALAR

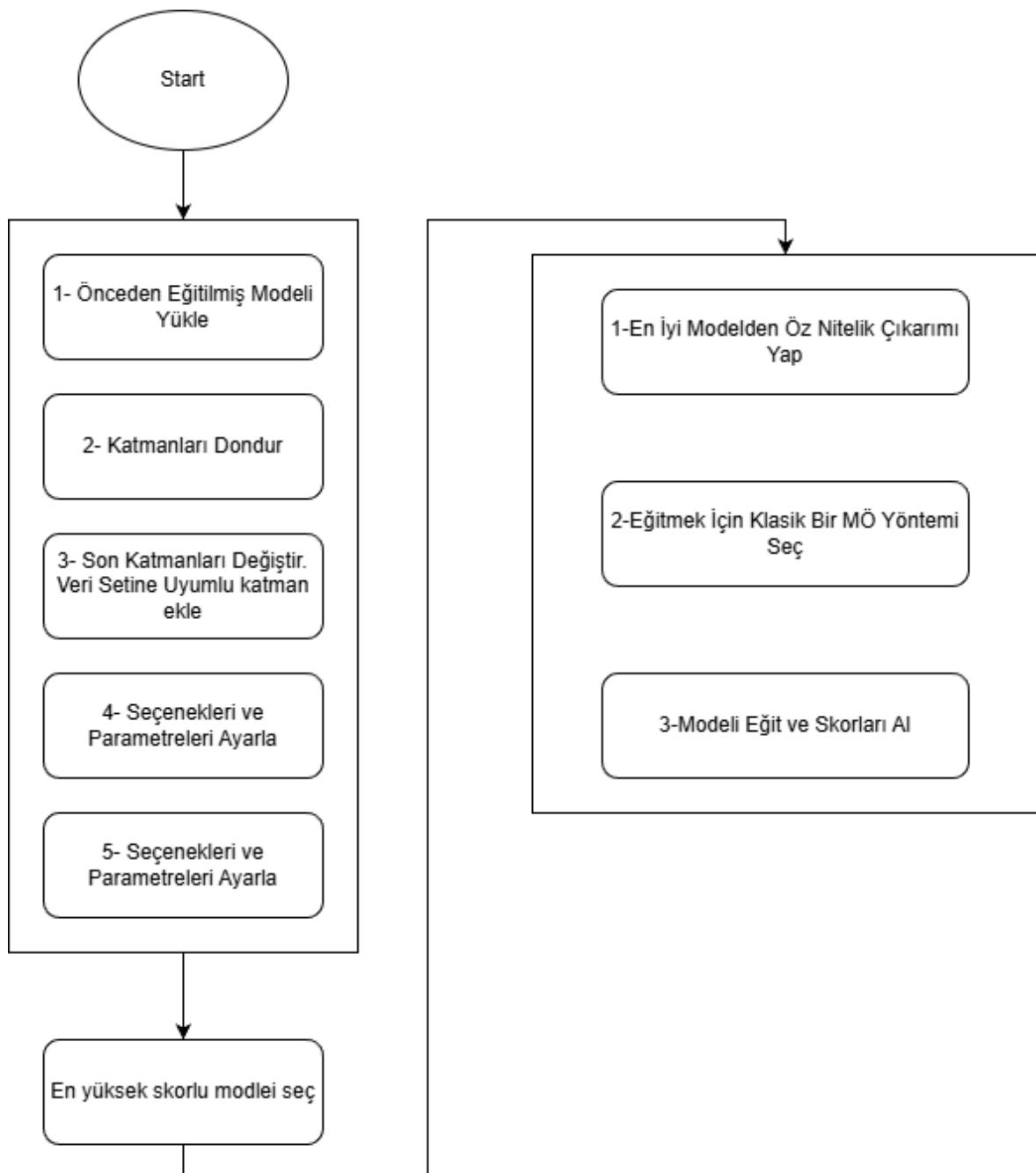
Fatemeh Sadat Seyed Khamoushi tarafından 2025 yılında yapılan çalışmada, gelişmiş derin öğrenme yöntemleri kullanılarak bitki hastalıklarının tespiti üzerine bir yüksek lisans tezi sunulmuştur. Bu araştırmada, yaprak görüntülerinden yola çıkarak hastalıkları sınıflandıran bir derin öğrenme modeli önerilmiştir. Yüksek doğruluk elde etmek amacıyla ResNet50, Inception-ResNet, VGG16 gibi modern Evrişimli Sinir Ağları (CNN) mimarileri ve özel bir model karşılaştırılmıştır. Çalışmada kullanılan veri seti üzerinde ön işleme adımları uygulanmış ve tüm modeller bu veri setinin eğitim, doğrulama ve test alt kümeleri ile eğitilip test edilmiştir. Deneysel sonuçlar, test setinde %97,22'lük bir doğruluk oranı ile en iyi performansı VGG16 modelinin gösterdiğini ortaya koymuştur. Diğer mimariler olan özel CNN, Inception-ResNet ve ResNet50 ise daha değişken performans sergilemiştir. Model karşılaştırmaları, VGG16 ve Inception-ResNet gibi önceden eğitilmiş mimarilerin, daha derin ve karmaşık özellik setlerini çıkarma yetenekleri sayesinde özel olarak tasarlanmış mimarilere göre çoğu senaryoda daha üstün performans gösterdiğini ortaya koymaktadır[3].



Şekil 2.1. Önceden eğitilmiş bilgiden yararlanan modelin çalışma prensibi [313].

Zainab Fadhl Abbas ISHKAYYIR tarafından yapılan çalışmada, derin öğrenmeye dayalı hibrit bir model ile bitki hastalıklarının tespiti amaçlanmıştır. Bu çalışmada, bitki hastalıklarının sınıflandırılması için PlantVillage veri seti kullanılmıştır. Yöntem olarak, ilk aşamada ResNet50, VGG16, VGG19, MobileNet, DenseNet121, InceptionV3 ve ConvNext gibi çeşitli derin öğrenme modelleri transfer öğrenme teknigi ile eğitilmiştir. Bu modeller arasından en iyi performansı gösteren MobileNet'in (%97,70 doğruluk) özellik çıkarma

yeteneği kullanılarak; SVM, KNN, Random Forest, Logistic Regression ve Decision Tree gibi klasik makine öğrenmesi sınıflandırıcıları ile hibrit modeller oluşturulmuştur. Geliştirilen hibrit modeller arasında en yüksek performansı %98,46 doğruluk oranı ile MobileNet ve SVM birleşimiyle oluşturulan model göstermiştir. Sonuç olarak, önerilen hibrit modelin yüksek doğrulukla birden fazla bitki hastalığını etkili bir şekilde tespit edebildiği gözlemlenmiştir [4].



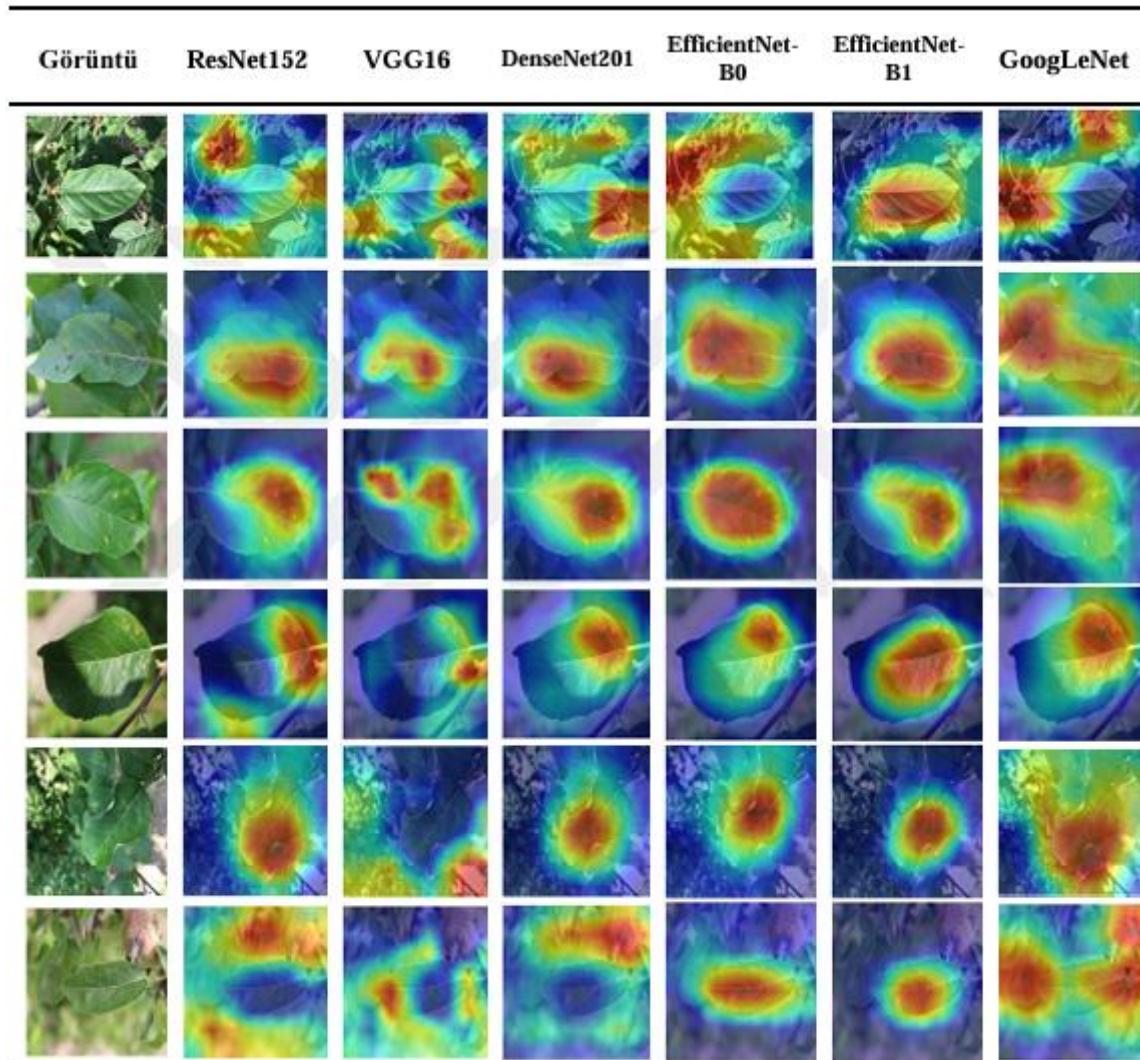
Şekil 2.2. Çalışmada kullanılan hibrit öğrenme modelinin akış şeması [4].

Elif Genç tarafından 2025 yılında yapılan çalışmada, bitki hastalıklarının yapay zeka ile erken tespiti üzerine çok yönlü bir yaklaşım geliştirilmiştir. Bu tez kapsamında, sağlıklı ve hastalıklı bitkilerden veri toplamak amacıyla iki özel iklim odası kurulmuştur. Çalışmada üç ana yöntem kullanılmıştır: Birincisi, topraktan alınan sıcaklık, nem, pH gibi zaman serisi verilerinin Uzun-Kısa Vadeli Bellek (LSTM) modeli ile analiz edilerek hastalıkların erken teşhisi hedeflenmiştir. İkincisi, bitki fotoğrafları kullanılarak renklerin yeniden yapılandırılmasına dayalı bir Pix2Pix derin öğrenme modeli ile bitkilerdeki anomaliler tespit edilmiştir. Son olarak, iklim odalarından elde edilen bitki fotoğraflarının sınıflandırılması için AlexNet, ResNet50, VGG16 ve SqueezeNet gibi önceden eğitilmiş derin öğrenme modelleri transfer öğrenme tekniği ile kullanılmıştır. Sonuçlar, hem zamansal verilerin (LSTM ile %99,94 doğruluk) hem de görüntü tabanlı analizlerin (AlexNet ile %99,53 doğruluk) bitki hastalıklarının erken teşhisinde başarılı ve etkili olduğunu göstermiştir [5].

Elif Ünal Çayır tarafından 2024 yılında yapılan çalışmada, bitki hastalıklarının derin öğrenme ile sınıflandırılması için çeşitli görüntü işleme modellerinin performansları karşılaştırılmıştır. Bu amaçla, PlantVillage veri seti kullanılarak VGG16, ResNet, EfficientNet, Xception v3, Inception v3 ve bu modellerden oluşturulan yeni bir topluluk öğrenimi modeli eğitilmiştir. Her model 40 adım boyunca eğitilerek test edilmiştir. Yapılan değerlendirmeler sonucunda, önerilen topluluk öğrenimi modeli %94,23'lük test doğruluğu ile diğer tüm modellere kıyasla en yüksek başarıyı elde etmiştir. Bu sonuç, önerilen modelin bitki hastalıkları sınıflandırmasında etkili bir seçenek olduğunu ve gelecekteki tarım uygulamaları için önemli bir potansiyel taşıdığını göstermektedir [6].

Ahmet Enes Kılıç tarafından 2024 yılında yapılan çalışmada, transfer öğrenme tabanlı açıklanabilir derin öğrenme yöntemleri kullanılarak bitki hastalıklarının sınıflandırılması incelenmiştir. Çalışmada, PlantVillage veri seti üzerinde ResNet152, VGG16, EfficientNet-B0, EfficientNet-B1, DenseNet201, GoogLeNet ve Vision Transformers gibi yedi farklı derin öğrenme modeli eğitilmiştir. Modellerin performansları karşılaştırılmış ve karar mekanizmaları, açıklanabilir yapay zekâ (XAI) yöntemleriyle analiz edilmiştir. PlantVillage veri setinde en yüksek başarıyı %99,81 doğruluk oranı ile DenseNet201 modeli göstermiştir. Ayrıca, modellerin laboratuvar ortamı başarısının gerçek dünya verilerindeki (Kaggle Plant Pathology yarışma verileri) performansını yansıtmadığı, bu veri setlerinde başarı oranlarının

düşürüyü tespit edilmiştir. Bu durum, veri seti çeşitliliğinin model performansı üzerindeki kritik etkisini ve laboratuvar ile gerçek dünya verileri arasındaki farkların önemini vurgulamaktadır [7].



Şekil 2.3 Modellerinin karar mekanizmalarını karşılaştırılan açıklanabilirlik (XAI) ısı haritaları.

Kadambari Raghuram ve Malaya Dutta Borah tarafından 2025 yılında yapılan çalışmada, domates bitkisi hastalıklarının tespiti için Derin Pekiştirmeli Öğrenme ve Transfer Öğrenme (DRL-TL) tabanlı hibrit bir model önerilmiştir. Bu yaklaşımda, öncelikle yüksek çözünürlüklü bitki yaprağı görüntüleri gelişmiş bir ön işleme teknigi ile iyileştirilmiştir. Ardından, önceden eğitilmiş MobileNetV2 modelini temel alan DRL-TL mimarisi, yaprakların uzamsal bilgilerini dikkate alarak üç boyutlu bir şekilde özellik çıkarmak için kullanılmıştır. Model, etiketlenmiş domates hastlığı görüntülerinden oluşan bir veri seti

üzerinde eğitilmiştir. Elde edilen sonuçlar, önerilen hibrit öğrenme modelinin (DRL-TL) %99,23 doğruluk oranıyla mevcut yöntemlerden daha üstün bir performans sergilediğini göstermiştir. Çalışma, bu yaklaşımın farklı bitki türleri ve çevresel koşullarda hastalıkları doğru bir şekilde tespit ederek sağlamlık ve soyutlama yeteneği gösterdiğini ortaya koymaktadır.[8]

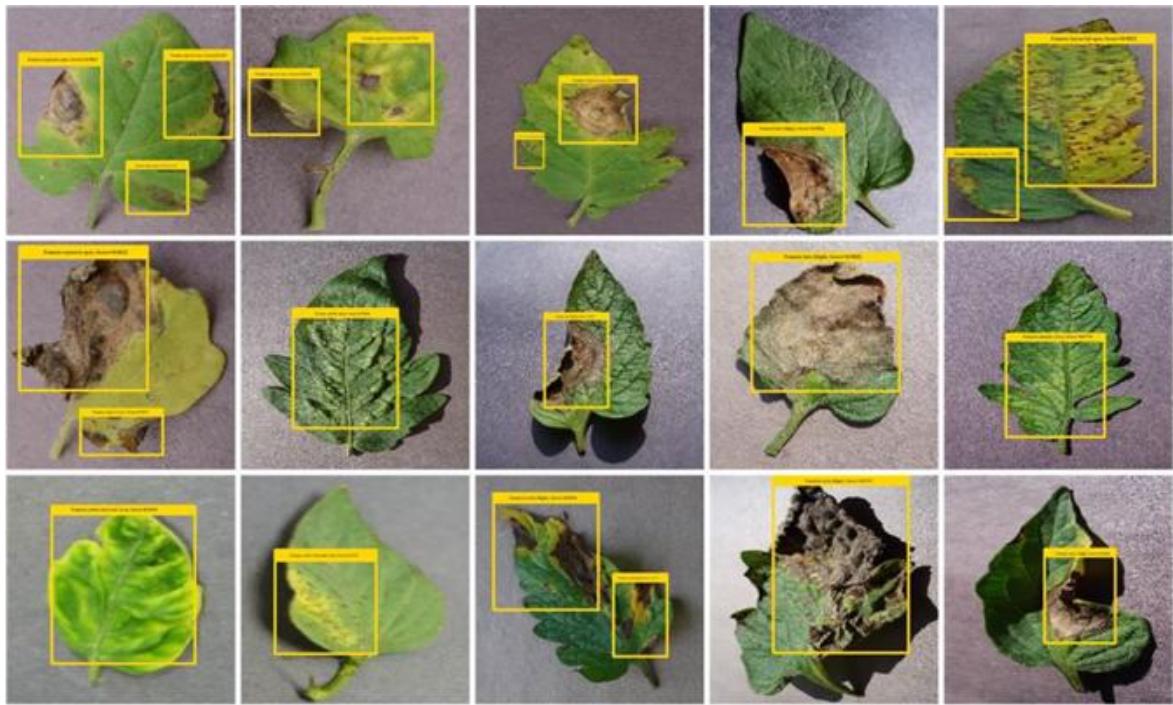
Argüeso ve arkadaşları tarafından 2020 yılında yapılan çalışmada, derin öğrenme yöntemlerinin büyük veri setlerine olan bağımlılığını azaltmak amacıyla, tarlada çekilmiş görüntüler kullanılarak bitki hastalıklarının sınıflandırılması için Az-Örnekle Öğrenme (Few-Shot Learning - FSL) yaklaşımı sunulmuştur. Bu çalışma için 38 farklı bitki yaprağı ve/veya hastalık türünü içeren PlantVillage veri setindeki 54.303 etiketli görüntü kullanılmıştır. Yöntem olarak, genel bitki yaprağı özelliklerini öğrenmek amacıyla Inception V3 ağı, veri setinin bir kaynak alanı (32 sınıf) üzerinde ince ayarlanmıştır. Ardından bu bilgi, Triplet kaybı (Triplet loss) kullanan Siyam ağları (Siamese networks) tabanlı bir FSL yöntemiyle hedef alandaki (6 sınıf) yeni hastalık türlerini çok az sayıda görüntü ile öğrenmek için kullanılmıştır. Sonuçlar, önerilen FSL yönteminin, sınıf başına yalnızca 80 görüntü kullanarak %90,0 medyan doğruluğa ulaştığını göstermiştir. Bu, tam veri setiyle elde edilen doğruluğa kıyasla sadece %4'lük bir düşüşle, eğitim verisi ihtiyacında yaklaşık %90'lık bir azalma anlamına gelmektedir. Ayrıca, bu yaklaşımın, özellikle çok az sayıda örnekle çalışıldığında, klasik ince ayar transfer öğrenme yöntemlerinden önemli ölçüde daha iyi performans gösterdiği ortaya konmuştur [9].

Mamta Gehlot ve Geeta Chhabra Gandhi tarafından 2023 yılında yapılan çalışmada, bitki hastalığı tespiti ve görselleştirilmesi için "EffiNet-TS" adında derinlemesine yorumlanabilir bir mimari önerilmiştir. Bu mimari, EfficientNetV2 tabanlı bir Öğretmen/Öğrenci (Teacher/Student) yapısı üzerine kurulmuştur ve mevcut ResTS mimarisindeki aşırı öğrenme (overfitting) sorunlarını çözmeyi ve görselleştirme sonuçlarını iyileştirmeyi hedeflemiştir. Çalışmada, ResTS'in Xception tabanlı yapısı yerine EfficientNetV2S kullanılmış ve kod çözücü ağına (decoder network) "DscDF" (Dual skip connection deconv fusion) adında yeni bir blok eklenmiştir. Modelin eğitimi ve testi için 38 kategori içeren PlantVillage veri seti kullanılmıştır. Deneysel sonuçlar, önerilen EffiNet-TS mimarisinin, 0.989 F1 skoru ve 0.990 doğruluk oranı ile ResTS mimarisini geride bıraktığını göstermiştir.

Bu çalışma, hastalıkları sınıflandırmanın yanı sıra, hastalığın temel belirtilerini görselleştirerek yorumlanabilir sonuçlar sunan bir model ortaya koymaktadır [10].

Meenakshi Srivastava ve Jasraj Meena tarafından 2024 yılında yapılan çalışmada, bitki yapraklarındaki hastalıkların tespiti ve sınıflandırılması için modifiye edilmiş transfer öğrenme modelleri önerilmiştir. Bu amaçla VGG16, MobileNetV2, Xception, InceptionV3 ve DenseNet121 gibi beş farklı Derin Evrişimli Sinir Ağları modeli, aşırı öğrenme (overfitting) sorununu gidermek amacıyla veri artırımı, yeniden boyutlandırma gibi ön işleme teknikleri kullanılarak iki farklı veri seti üzerinde test edilmiştir. Çalışmada kullanılan ilk veri seti Mendeley'den alınmış olup 22 sınıfa ayrılmış 4590 yaprak görüntüsü içermektedir. İkinci veri seti ise PlantVillage'dan alınan ve 2 sınıfa ayrılmış 2052 kiraz yaprağı görüntüsünden oluşmaktadır. Sonuçlar, Mendeley veri setinde modifiye edilmiş MobileNetV2 modelinin %98.9 doğruluk oranıyla en iyi performansı gösterdiğini ortaya koymuştur. Kiraz veri setinde ise modifiye edilmiş DenseNet121 modeli %99.9 doğruluk oranı ile en başarılı model olmuştur [11].

Marriam Nawaz ve arkadaşları tarafından 2022 yılında yapılan çalışmada, domates bitkisi yapraklarındaki hastalıkların tespiti ve sınıflandırılması için ResNet-34 tabanlı Faster-RCNN mimarisine dayanan sağlam bir derin öğrenme yaklaşımı sunulmuştur. Bu yöntemde, ilk olarak hastalıklı bölgelerin (Region of Interest - RoI) etiketlenmesi yapılmış, ardından Evrişimli Blok Dikkat Modülü (Convolutional Block Attention Module - CBAM) ile güçlendirilmiş ResNet-34, özellik çıkarıcı olarak kullanılmıştır. Bu yapı, Faster-RCNN modeli içinde hem hastalıkları tespit etmek (lokalizasyon) hem de sınıflandırmak için eğitilmiştir. Model, PlantVillage Kaggle veri seti üzerinde test edilmiş ve %99,97 doğruluk ile 0.981 mAP (mean Average Precision) değeri elde etmiştir. Çalışma, önerilen yaklaşımın bulanıklık, gürültü, renk ve boyut değişimleri gibi çeşitli görüntü bozulmalarına karşı dayanıklı olduğunu ve düşük maliyetli, etkili bir çözüm sunduğunu göstermektedir [12].



Şekil 2.4 Nesne tespiti modeli ile bitki hastalıklarının tespiti ve yerinin belirlenmesi.

Referans	Bitki Türü/Türleri	Veri Seti	Sınıf Sayısı	Model	Model Performansı
[3]	Çeşitli	New Plant Disease (Kaggle) - 87,867 görüntü	38	VGG16	%97,22
[4]	Çeşitli	PlantVillage - 20,6k+ görüntü	15	MobileNet SVM	%98,46
[5]	Salatalık	Özel oluşturulan iklim odası veri seti	2	AlexNet	%99,53
[6]	Çeşitli	PlantVillage	15	Topluluk Öğrenimi	%94,23
[7]	Çeşitli	PlantVillage	Belirtilmemiş	DenseNet201	%99,81
[8]	Domates	Etiketlenmiş domates hastalığı görüntüleri	Belirtilmemiş	DRL-TL	%99,23
[9]	Çeşitli	PlantVillage - 54,303 etiketli görüntü	38	Inception V3 Siyam Ağları (FSL)	%90,0
[10]	Çeşitli	PlantVillage	38	EffiNet-TS	%99,0
[11]	Çeşitli / Kiraz	Mendeley (4590) / PlantVillage (2052)	22 / 2	MobileNetV2 DenseNet121	%98,9
[12]	Domates	PlantVillage Kaggle veri seti	Belirtilmemiş	Faster-RCNN	%99,97

Tablo 2.1. Çalışmalarda kullanılan yöntemler ve performansları

3. ÖNERİLEN SİSTEM

Literatürdeki çalışmalar, transfer öğrenme tabanlı modellerin bitki hastalığı sınıflandırmasında yüksek başarı sergilediğini göstermektedir. Bu doğrultuda, bu çalışmada da önceden eğitilmiş çeşitli derin öğrenme mimarileri kullanılmıştır. Modellerin eğitimi için temel kaynak olarak PlantVillage [14,15] veri seti kullanılırken, modellerin gerçek dünya koşullarındaki performansını ve genelleme yeteneğini test etmek amacıyla ise Plant Pathology veri seti tercih edilmiştir. Bu bölümde, ilk olarak kullanılan veri setlerinin özellikleri ve uygulanan veri artırma (augmentation) gibi ön işleme adımları ele alınacaktır. Ardından, kullanılan modellerin mimari yapıları, eğitim süreçleri ve hiper parametreleri detaylandırılacaktır. Son olarak, eğitilen modelin entegre edildiği uygulamanın işleyışı ve mimarisi açıklanacaktır.

3.1. Veri Setleri

Çalışmada modellerin eğitileceği veri seti olarak Hughes ve Salathe (2015) tarafından oluşturulan ve Kaggle platformu üzerinden erişilen PlantVillage veri seti (Alidev, 2022) kullanılmıştır. Modelin gerçek dünya koşullarındaki performansının ölçülmesi için ise Plant Pathology [16,17] veri setleri kullanılmıştır.

PlantVillage

Veri seti, 14 farklı bitki türünü kapsayan, 24 hastalıklı bitki sınıfı ve 14 sağlıklı bitki sınıfı olmak üzere toplam 38 sınıfı içeren 54,303 adet renkli (RGB) görüntünden oluşmaktadır. Veri setindeki görüntüler, kontrollü koşullar altında ve tek tip bir arka plan önünde çekilmiş yaprak örneklerinden oluşmaktadır, bu da modellerin öncelikle yaprak morfolojisini ve hastalık belirtileri üzerine odaklanmasını sağlamaktadır. Çalışmanın ilerleyen aşamalarında, modelin genelleme kabiliyetini artırmak ve aşırı öğrenmeyi (overfitting) önlemek amacıyla bu ham görüntülere veri artırma (data augmentation) teknikleri uygulanacaktır. Veri setinin bitki türlerine ve sınıflara göre dağılımına ilişkin istatistik bilgiler Tablo 3.1'de sunulmuştur.

Sınıf Adı	Örnek Sayısı
Apple__Apple_scab	630
Apple__Black_rot	621
Apple__Cedar_apple_rust	275

Apple	_healthy	1645
Blueberry	_healthy	1502
Cherry_(including_sour)	_Powdery_mildew	1052
Cherry_(including_sour)	_healthy	854
Corn_(maize)	_Cercospora_leaf_spot	513
Corn_(maize)	_Common_rust	1192
Corn_(maize)	_Northern_Leaf_Blight	985
Corn_(maize)	_healthy	1162
Grape	_Black_rot	1180
Grape	_Esca_(Black_Measles)	1383
Grape	_Leaf_blight_(Isariopsis_Leaf_Spot)	1076
Grape	_healthy	423
Orange	_Haunglongbing_(Citrus_greening)	5507
Peach	_Bacterial_spot	2297
Peach	_healthy	360
Pepper,_bell	_Bacterial_spot	997
Pepper,_bell	_healthy	1478
Potato	_Early_blight	1000
Potato	_Late_blight	1000
Potato	_healthy	152
Raspberry	_healthy	371
Soybean	_healthy	5090
Squash	_Powdery_mildew	1835
Strawberry	_Leaf_scorch	1109
Strawberry	_healthy	456
Tomato	_Bacterial_spot	2127
Tomato	_Early_blight	1000
Tomato	_Late_blight	1909
Tomato	_Leaf_Mold	952
Tomato	_Septoria_leaf_spot	1771
Tomato	_Spider_mites Two-spotted_spider_mite	1676
Tomato	_Target_Spot	1404
Tomato	_Tomato_Yellow_Leaf_Curl_Virus	5357
Tomato	_Tomato_mosaic_virus	337
Tomato	_healthy	1591

Tablo 3.1. Sınıflandırma adları ve örnek sayısı [14].

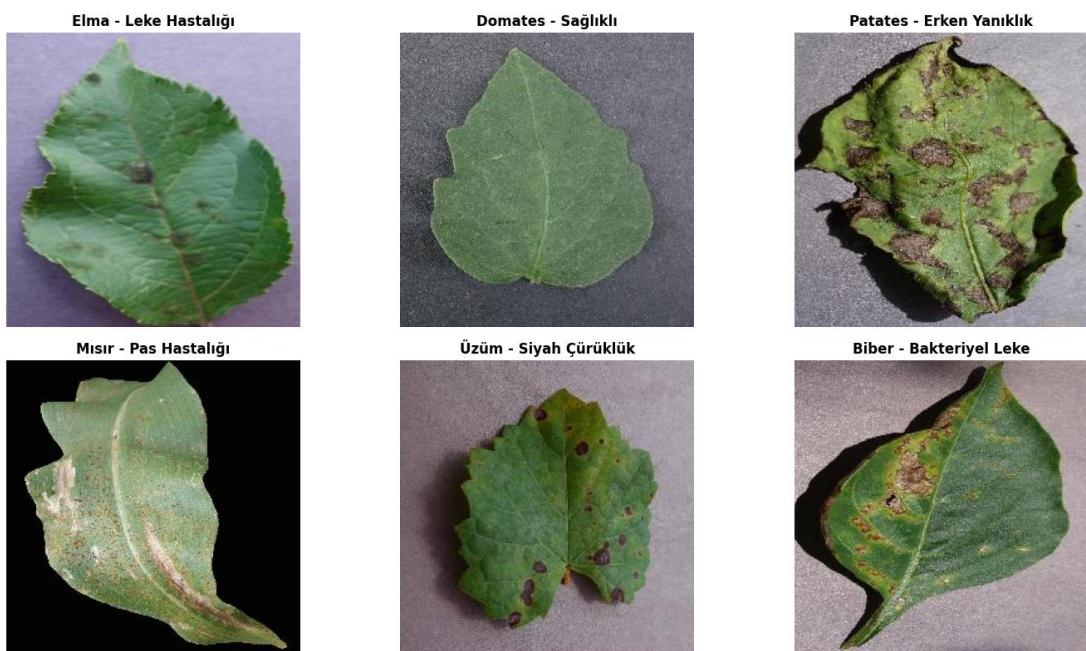
Veri setinin hazırlanması aşamasında, modelin genellemeye yeteneğini artırmak ve aşırı öğrenmenin (overfitting) önüne geçmek amacıyla veri kümesi üç ana parçaya ayrılmıştır:

- Eğitim Seti (Training Set): Verinin %80'i modelin ağırlıklarını güncellemek ve öğrenme sürecini gerçekleştirmek için kullanılmıştır.

- Doğrulama Seti (Validation Set): Verinin %10'u, eğitim sırasında modelin performansını izlemek, hiperparametre optimizasyonu yapmak ve "early stopping" (erken durdurma) mekanizmasını işletmek için ayrılmıştır.
- Test Seti (Test Set): Verinin kalan %10'u, eğitilen modelin daha önce hiç görmediği veriler üzerindeki nihai performansını ölçmek için kullanılmıştır.

Veri bölme işlemi sırasında stratified sampling (tabakalı örnekleme) yöntemi kullanılarak, her bir sınıfın eğitim, doğrulama ve test setlerinde dengeli bir şekilde temsil edilmesi sağlanmıştır.

Görüntü ön işleme (preprocessing) adımlarında, tüm görüntüler EfficientNetB3 modelinin giriş boyutuna uygun olarak 224x224 piksel boyutlarına yeniden boyutlandırılmıştır. Ayrıca, modelin daha hızlı ve kararlı öğrenmesini sağlamak amacıyla görüntüler, ImageNet veri setinin istatistiksel değerleri (ortalama: [0.485, 0.456, 0.406], standart sapma: [0.229, 0.224, 0.225]) kullanılarak normalize edilmiştir. Eğitim setindeki veri çeşitliliğini artırmak için RandomHorizontalFlip (rastgele yatay çevirme) gibi veri artırma (data augmentation) teknikleri uygulanmıştır.



Şekil 3.1: PlantVillage veri setinden örnek sağlıklı ve hastalıklı yaprak görüntüleri.

3.2. Model Mimarisi

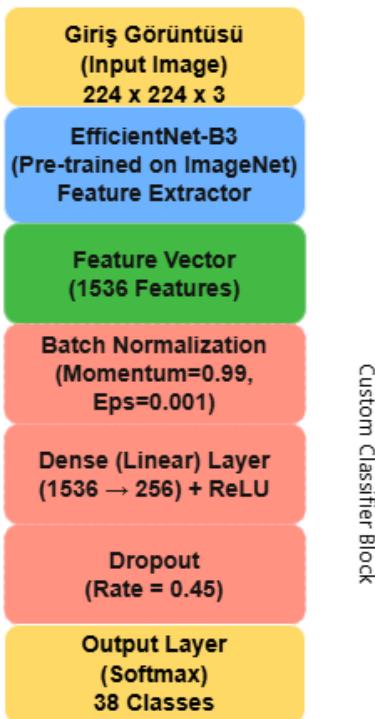
Geliştirilen sisteme, derin öğrenme tabanlı sınıflandırma işlemi için EfficientNet-B3 mimarisi temel alınmıştır. EfficientNet ailesi, modelin derinliğini, genişliğini ve çözünürlüğünü dengeli bir şekilde ölçeklendirerek (compound scaling), daha az parametre ile yüksek doğruluk oranlarına ulaşabilen modern bir konvolüsyonel sinir ağı (CNN) mimarisiidir. B3 varyasyonu, mobil cihazlarda çalışabilecek kadar hafif, ancak karmaşık bitki hastalıklarını ayırt edebilecek kadar güçlü olduğu için tercih edilmiştir.

Çalışmada Transfer Learning (Transfer Öğrenme) tekniği uygulanmıştır. Bu yöntem, modelin sıfırdan eğitilmesi yerine, daha önce ImageNet gibi büyük bir veri seti üzerinde eğitilmiş ağırlıkların başlangıç noktasını olarak kullanılmasını sağlar. Bu sayede, model temel görsel özellikleri (kenarlar, dokular vb.) zaten öğrenmiş olarak başlar ve eğitim süresi kısalırken başarı oranı artar.

Modelin sınıflandırma katmanı (classifier), projeye özgü olarak yeniden tasarlanmıştır. EfficientNet-B3'ün orijinal son katmanı çıkarılarak yerine aşağıdaki katmanlardan oluşan özelleştirilmiş bir blok eklenmiştir:

1. Batch Normalization: Eğitim sürecini hızlandırmak ve kararlı hale getirmek için.
2. Linear Layer (Dense): 1536 giriş özelliğini 256 özelliğe indiren tam bağlantılı katman.
3. ReLU Aktivasyon Fonksiyonu: Doğrusal olmayan ilişkileri öğrenmek için.
4. Dropout (0.45): Nöronların %45'ini rastgele kapatarak aşırı öğrenmeyi engellemek için.
5. Output Layer: 256 özelliği 38 sınıfına dönüştüren son sınıflandırma katmanı.

Ayrıca modelin genelleme yeteneğini artırmak için L1 ve L2 regülasyon teknikleri de kayıp fonksiyonuna entegre edilmiştir.



Şekil 3.3: Önerilen transfer öğrenme tabanlı model mimarisi.

3.3. Eğitim Süreci

Modelin eğitimi, önerilen EfficientNet-B3 mimarisi üzerinde, PyTorch derin öğrenme kütüphanesi kullanılarak gerçekleştirilmiştir. Eğitim stratejisi, sadece modelin ağırlıklarını güncellemeyi değil, aynı zamanda aşırı öğrenmeyi (overfitting) engellemeyi ve en iyi genelleme yeteneğine sahip modeli elde etmeyi hedefleyen dinamik bir yapı üzerine kurulmuştur.

3.3.1. Eğitim Parametreleri ve Konfigürasyon

Eğitim sürecinde kullanılan hiperparametreler, literatür taraması ve deneysel çalışmalar sonucunda belirlenmiştir. Optimizasyon algoritması olarak, özellikle gömülü (embedding) katmanları olan veya seyrek gradyanlara sahip modellerde kararlı sonuçlar veren Adamax tercih edilmiştir. Adamax, Adam algoritmasının bir varyasyonu olup, sonsuz norm (L_∞) tabanlı güncelleme kuralı sayesinde parametre güncellemelerinde daha stabil bir davranış sergilemektedir.

Ayrıca, modelin karmaşıklığını kontrol altında tutmak için kayıp fonksiyonuna (Loss Function) ek olarak L1 ve L2 regülasyonu (Elastic Net yaklaşımı) uygulanmıştır. Bu,

modelin ağırlıklarının çok büyümesini engelleyerek (L2) ve gereksiz ağırlıkları sıfıra yaklaştırarak (L1) modelin daha sade ve gürbüz olmasını sağlar.

Parametre	Değer/Açıklama
Optimizer	Adamax
Başlangıç Öğrenme Oranı (LR)	0.001
Batch Size (Demet Boyutu)	40
Epoch Sayısı	40 (Maksimum)
Kayıp Fonksiyonu	CrossEntropyLoss
L1 Regülasyon Katsayısı	0.006
L2 Regülasyon Katsayısı	0.016
Dropout Oranı	0.45

Tablo 3.2: Eğitim sürecinde kullanılan temel hiperparametreler.

3.3.2. Eğitim Dinamikleri ve Geri Çağırma (Callback) Sistemi

Eğitim sürecini otomatize etmek ve en iyi sonucu garanti altına almak için özel bir TrainingCallback mekanizması geliştirilmiştir. Bu mekanizma eğitim boyunca şu adımları izler:

Dinamik Öğrenme Oranı (LR Scheduling), eğitim sırasında modelin doğrulama başarısı (validation accuracy) veya kaybı (validation loss) belirli bir süre (patience=3 epoch) iyileşmezse, öğrenme oranı 0.5 faktörü ile çarpılarak yarıya düşürülür. Bu, modelin yerel minimumlara sıkışmasını engeller ve daha hassas ağırlık güncellemleri yapılmasını sağlar. Erken Durdurma (Early Stopping), öğrenme oranı düşürülmesine rağmen model performansında iyileşme gözlemlenmezse (stop_patience=3), eğitim süreci 40 epoch tamamlanmadan sonlandırılır. Bu yöntem, gereksiz hesaplama maliyetini önler ve aşırı öğrenmeyi engeller. En İyi Modelin Kaydedilmesi (Checkpointing), her epoch sonunda, modelin doğrulama setindeki performansı ölçülür. Eğer mevcut epoch'taki başarı, o ana kadarki en yüksek başarıdan daha iyiysse, model ağırlıkları [best_model.pth](#) olarak diske kaydedilir. Böylece eğitim bittiğinde elimizde her zaman en başarılı model bulunur.

3.4. Model Dönüşümü

Eğitilen PyTorch modelinin mobil uygulamada verimli bir şekilde çalışabilmesi için, modelin mobil cihazlara uygun bir formata dönüştürülmesi gerekmektedir. Bu amaçla, model TensorFlow Lite formatına dönüştürülmüştür. Bu dönüşüm süreci iki aşamada gerçekleştirilmiştir:

1. ONNX (Open Neural Network Exchange) Dönüşümü: İlk olarak, eğitilen PyTorch modeli platformlar arası model paylaşım standarı olan ONNX formatına export edilmiştir. ONNX, farklı derin öğrenme çerçeveleri arasında bir köprü görevi görerek modelin taşınabilirliğini sağlar.
2. TFLite Dönüşümü: Elde edilen ONNX modeli, onnx2tf aracı kullanılarak TensorFlow Lite formatına çevrilmiştir. TFLite, mobil ve gömülü cihazlar için optimize edilmiş, düşük gecikme süresi ve küçük dosya boyutu sunan bir formattır.

Dönüşüm sırasında, modelin boyutunu küçültmek ve işlem hızını artırmak amacıyla Float32 veya Float16 hassasiyetinde optimizasyonlar yapılmıştır. Sonuç olarak elde edilen [plant_model.tflite](#) dosyası, mobil uygulama içerisine gömülerek internet bağlantısına ihtiyaç duymadan cihaz üzerinde sınıflandırma yapılmasına olanak tanımıştir.



Şekil 3.6: Modelin mobil platforma uygun hale getirilmesi için uygulanan dönüşüm adımları

3.5. Giriş Görüntüsü Arka Plan Kaldırma

Bitki hastalığı teşhisinde modelin başarısını etkileyen en önemli faktörlerden biri, giriş görüntüsünün kalitesi ve içерdiği gürültü miktarıdır. Kullanıcıların mobil cihazlarıyla çektiğleri fotoğraflarda yaprak dışında toprak, gökyüzü, eller veya diğer bitkiler gibi arka plan öğeleri bulunabilmektedir. Bu istenmeyen öğeler, modelin dikkatini dağıtarak sınıflandırma başarısını olumsuz etkileyebilir. Bu sorunu çözmek amacıyla, görüntü sınıflandırma modeline gönderilmeden önce yaprak görüntüsünün arka planını otomatik olarak kaldırın bir ön işleme adımı entegre edilmiştir.

3.5.1. U²-Net Modeli ve Tercih Sebebi

Arka plan kaldırma işlemi için U²-Net (U-Square Net) mimarisi tercih edilmiştir. U²-Net, Qin ve arkadaşları tarafından 2020 yılında önerilen, öne çıkan nesne tespiti (Salient Object Detection - SOD) için tasarlanmış derin bir sinir ağı mimarisidir [18].

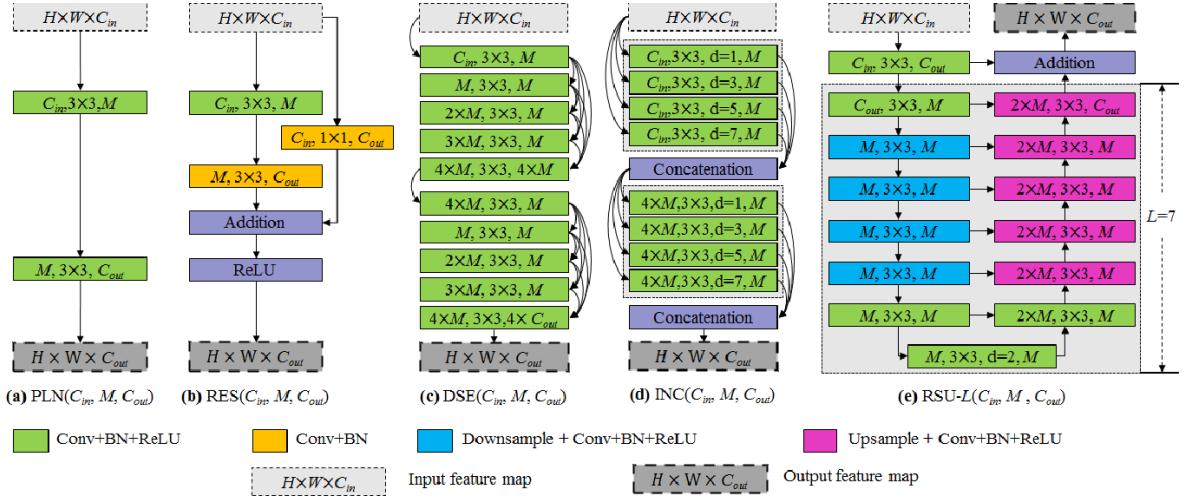
U²-Net'in bu çalışmada tercih edilmesinin başlıca nedenleri şunlardır:

1. Yüksek Doğruluk: U²-Net, iç içe geçmiş U-yapıları (Residual U-blocks - RSU) sayesinde hem yerel hem de küresel bağlamsal bilgileri etkili bir şekilde yakalayabilmektedir. Bu sayede karmaşık şekillere sahip yaprakların kenarlarını bile hassas bir şekilde tespit edebilmektedir.
2. Hafif Versiyon (U²-Net-P): Orijinal U²-Net modelinin yanı sıra, mobil cihazlarda verimli çalışılabilmesi için tasarlanmış daha küçük bir versiyonu olan U²-Net-P (Portable) sunulmaktadır. Bu çalışmada, mobil uygulama için U²-Net-P modeli tercih edilmiştir. U²-Net-P, orijinal modele kıyasla önemli ölçüde daha az parametre içermesine rağmen, tatmin edici segmentasyon sonuçları üretebilmektedir.
3. Önceden Eğitilmiş Ağırlıklar: U²-Net, geniş bir veri seti üzerinde önceden eğitilmiş olarak sunulmaktadır. Bu sayede, ek eğitim gerektirmeden doğrudan kullanılabilmektedir (Transfer Learning).

3.5.2. U²-Net Mimarisi

U²-Net'in temel yapı taşı, RSU (Residual U-block) bloklarıdır. Klasik U-Net mimarisindeki encoder-decoder yapısından farklı olarak, U²-Net her bir encoder ve decoder aşamasında ayrı birer U-yapısı (RSU bloğu) kullanır. Bu iç içe geçmiş yapı, modelin farklı ölçeklerdeki özellikleri daha iyi öğrenmesini sağlar.

Model, giriş olarak RGB formatında bir görüntü alır ve çıkış olarak her piksel için 0 ile 1 arasında bir olasılık değeri içeren tek kanallı bir maske (mask) üretir. Bu maskede yüksek değerler (1'e yakın) ön plan nesnesini (yaprak), düşük değerler (0'a yakın) ise arka planı temsil eder.



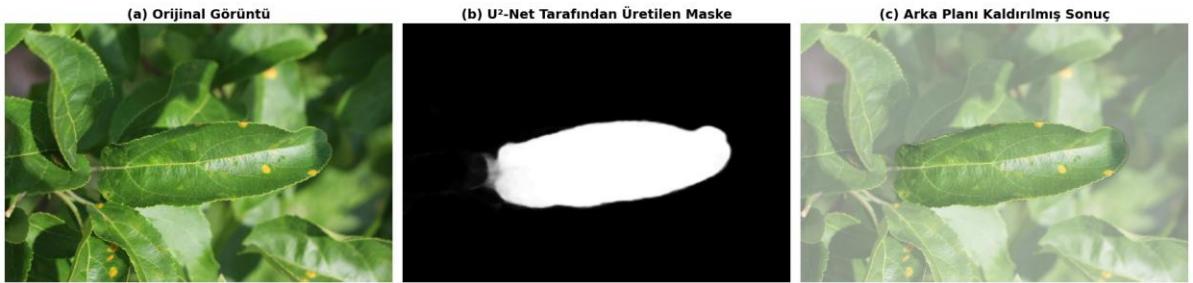
Şekil 3.7: U²-Net mimarisinin genel yapısı. Her bir En (Encoder) ve De (Decoder) blogu, kendi içinde bir RSU yapısı barındırmaktadır.

3.5.3. Uygulama Akışı

Arka plan kaldırma işlemi, mobil uygulama içerisinde aşağıdaki adımlarla gerçekleştirilmektedir:

1. Görüntü Yükleme: Kullanıcı, kamera veya galeriden bir yaprak görüntüsü seçer.
2. Ön İşleme (Preprocessing): Görüntü, U²-Net-P modelinin beklediği giriş boyutuna (320x320 piksel) yeniden boyutlandırılır ve ImageNet normalizasyon değerleri (ortalama: [0.485, 0.456, 0.406], standart sapma: [0.229, 0.224, 0.225]) kullanılarak normalize edilir.
3. Maske Üretimi (Inference): Normalize edilmiş görüntü, TensorFlow Lite formatına dönüştürülmüş U²-Net-P modeline ('u2netp_float32.tflite') beslenir. Model, çıkış olarak bir segmentasyon maskesi üretir.
4. Son İşleme (Postprocessing): Üretilen maske, orijinal görüntü boyutlarına yeniden ölçeklenir. Sigmoid fonksiyonu uygulanarak piksel değerleri 0-1 aralığına normalize edilir.
5. Arka Plan Kaldırma: Orijinal görüntü RGBA formatına dönüştürülür ve üretilen maske, alfa (şeffaflık) kanalı olarak uygulanır. Bu işlem sonucunda arka plan şeffaf hale gelir ve sadece yaprak görüntüsünü kalır.
6. Hastalık Tespit: Arka planı kaldırılmış görüntü, bitki hastalığı sınıflandırma modeline (EfficientNet-B3 tabanlı TFLite modeli) gönderilerek teşhis işlemi gerçekleştirilir.

Şekil 3.X: Arka Plan Kaldırma İşlemi Örneği



Şekil 3.8: Arka plan kaldırma işlemi örneği. (a) Orijinal görüntü, (b) U²-Net-P tarafından üretilen maske, (c) Arka planı kaldırılmış sonuç görüntüsü.

3.5.4. TensorFlow Lite Dönüşümü

U²-Net-P modelinin mobil cihazlarda verimli çalışabilmesi için, PyTorch formatındaki orijinal model ağırlıkları ('.pth') önceden de bitki hastalığı tespit etme modelinde yaptığı gibi önce ONNX formatına, ardından TensorFlow Lite ('.tflite') formatına dönüştürülmüştür.

3.6 Mobil Uygulama

Eğitilen derin öğrenme modellerinin son kullanıcıya ulaşılması için bir mobil uygulama geliştirilmiştir. Bu bölümde, uygulamanın geliştirilmesinde kullanılan teknolojiler, mimari yapısı, model entegrasyonu ve sunulan özellikler detaylı olarak açıklanmaktadır.

3.6.1 Geliştirme Ortamı ve Teknolojiler

Mobil uygulama, Google tarafından geliştirilen açık kaynaklı bir kullanıcı arayüzü (UI) geliştirme kiti olan Flutter framework'ü ve Dart programlama dili kullanılarak geliştirilmiştir. Flutter'in bu çalışmada tercih edilmesinin başlıca nedenleri şunlardır:

1. Çapraz Platform Desteği: Tek bir kod tabanı ile hem Android hem de iOS platformlarında çalışabilen uygulamalar geliştirilebilmektedir. Bu sayede geliştirme süresi ve maliyeti önemli ölçüde azaltılmıştır.
2. Yüksek Performans: Flutter, uygulamaları doğrudan native ARM koduna derlediğinden, JavaScript köprüsü kullanan diğer çapraz platform çözümlerine kıyasla daha yüksek performans sunmaktadır.
3. TensorFlow Lite Entegrasyonu: Flutter, tflite_flutter paketi aracılığıyla TensorFlow Lite modellerinin mobil cihazlarda verimli bir şekilde çalıştırılmasına olanak tanımaktadır.

Teknoloji/Kütüphane	Versiyon	Kullanım Amacı
Flutter SDK	3.10.1	Çapraz platform mobil uygulama geliştirme
Dart	3.x	Programlama dili
tflite_flutter	0.12.1	TensorFlow Lite model çıkarımı (inference)
image	4.5.4	Görüntü işleme ve manipülasyonu
Supabase	2.8.4	Bulut veritabanı ve kullanıcı kimlik doğrulama
Hive	2.2.3	Yerel veri depolama
Google Generative AI	0.4.6	Gemini tabanlı AI sohbet asistanı
Riverpod	3.0.3	Durum yönetimi

Tablo 3.3: Mobil uygulamada kullanılan temel teknolojiler ve kütüphaneler.

3.6.2 Uygulama Mimarisi

Uygulama, sürdürülebilir ve ölçeklenebilir bir yapı için katmanlı mimari (Layered Architecture) prensiplerine uygun olarak tasarlanmıştır. Bu mimari yaklaşım, kodun modüler olmasını, test edilebilirliğini ve bakım kolaylığını sağlamaktadır. Uygulama mimarisi üç ana katmandan oluşmaktadır:

1. Sunum Katmanı (Presentation Layer): Kullanıcı arayüzü bileşenlerini ve ekranları içermektedir. Flutter widget'ları kullanılarak Material Design 3 tasarım prensiplerine uygun arayüzler oluşturulmuştur.
2. İş Mantığı Katmanı (Domain Layer): Riverpod kütüphanesi ile durum yönetimi (state management) ve iş kuralları bu katmanda yer almaktadır. Provider'lar aracılığıyla veri akışı ve uygulama durumu yönetilmektedir.
3. Veri Katmanı (Data Layer): Yerel depolama (Hive), bulut servisleri (Supabase) ve model çıkarımı (TFLite) işlemleri bu katmanda gerçekleştirilmektedir.

3.6.3 Model Entegrasyonu ve Görüntü İşleme

Bölüm 3.5 ve 3.6'da açıklanan TensorFlow Lite formatındaki modeller (bitki hastalığı sınıflandırma ve arka plan kaldırma), mobil uygulamaya entegre edilmiştir. Kullanıcının çektiği veya galeriden seçtiği bir görüntü, hastalık teşhisini için aşağıdaki işlem hattından (pipeline) geçirilmektedir:

1. Görüntü Alımı: Kullanıcı, cihaz kamerası veya galeri aracılığıyla bir yaprak görüntüsü seçer.
2. Arka Plan Kaldırma: Görüntü, U²-Net-P modeline (u2netp_float32.tflite) beslenerek yaprak segmentasyonu gerçekleştirilir. Bu adım, arka plandaki gürültüyü ortadan kaldırarak sınıflandırma başarısını artırmaktadır.
3. Ön İşleme (Preprocessing): Segmente edilmiş görüntü, EfficientNet-B3 modelinin beklediği formata dönüştürülür:
 - 224×224 piksel boyutuna yeniden boyutlandırma
 - ImageNet normalizasyon değerleri ile normalize etme (ortalama: [0.485, 0.456, 0.406], standart sapma: [0.229, 0.224, 0.225])
 - Tensor formatına dönüştürme
4. Sınıflandırma (Classification): Ön işleminden geçen görüntü, bitki hastalığı sınıflandırma modeline (plant_model.tflite) beslenir ve 38 sınıf arasından en yüksek olasılığa sahip hastalık tespit edilir.
5. Sonuç Gösterimi: Tespit edilen hastalık adı, güven skoru ve tedavi önerileri kullanıcıya sunulur.

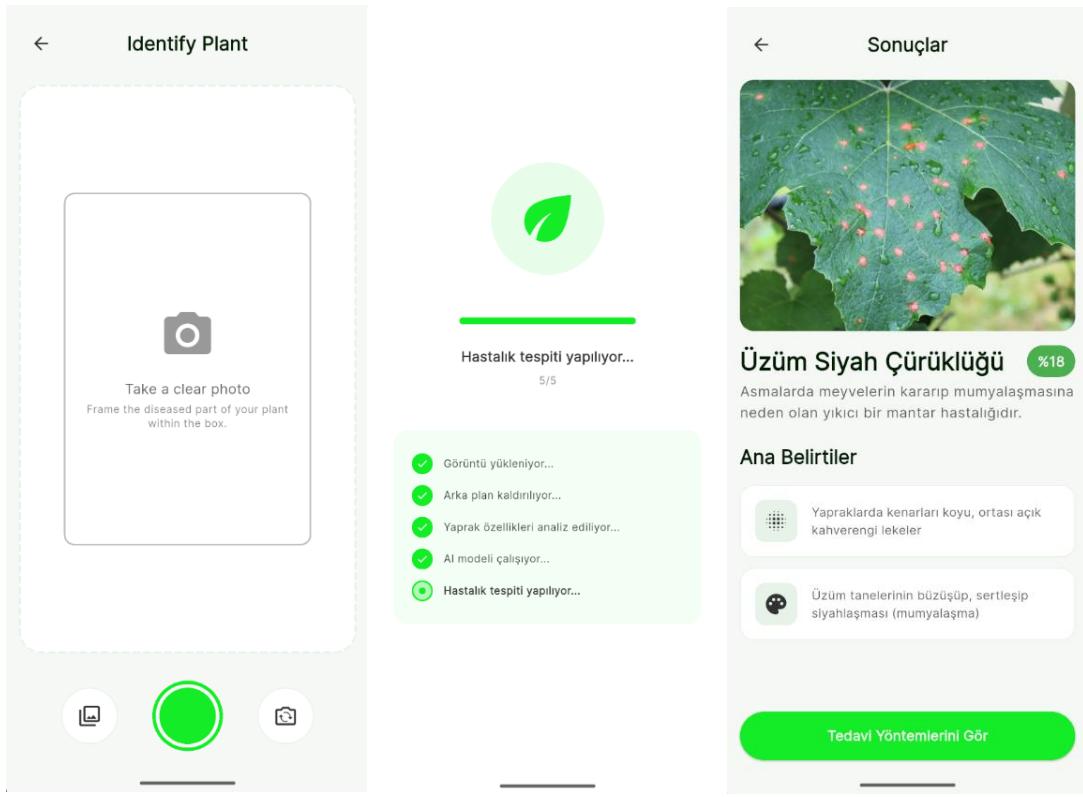
3.6.4 Uygulama Özellikleri

Geliştirilen mobil uygulama, bitki hastalığı teşhisinin yanı sıra çiftçilere kapsamlı bir tarım asistanlığı sunmak amacıyla çeşitli özellikler içermektedir.

3.6.4.1 Hastalık Teşhisi

Uygulamanın ana işlevi olan hastalık teşhisi modülü, kullanıcının kamera veya galeriden seçtiği bitki yaprağı görüntüsünü analiz ederek hastalık tespiti yapmaktadır. Teşhis sonucunda kullanıcıya şu bilgiler sunulmaktadır:

- Tespit edilen hastalık adı ve güven skoru (confidence score)
- Hastalığın detaylı açıklaması ve belirtileri
- Hastalığa neden olan patojen türü (mantar, bakteri, virüs)
- Şiddet seviyesi (düşük, orta, yüksek, kritik)
- Tedavi önerileri ve uygulama yöntemleri
- Önleme yöntemleri ve bakım ipuçları

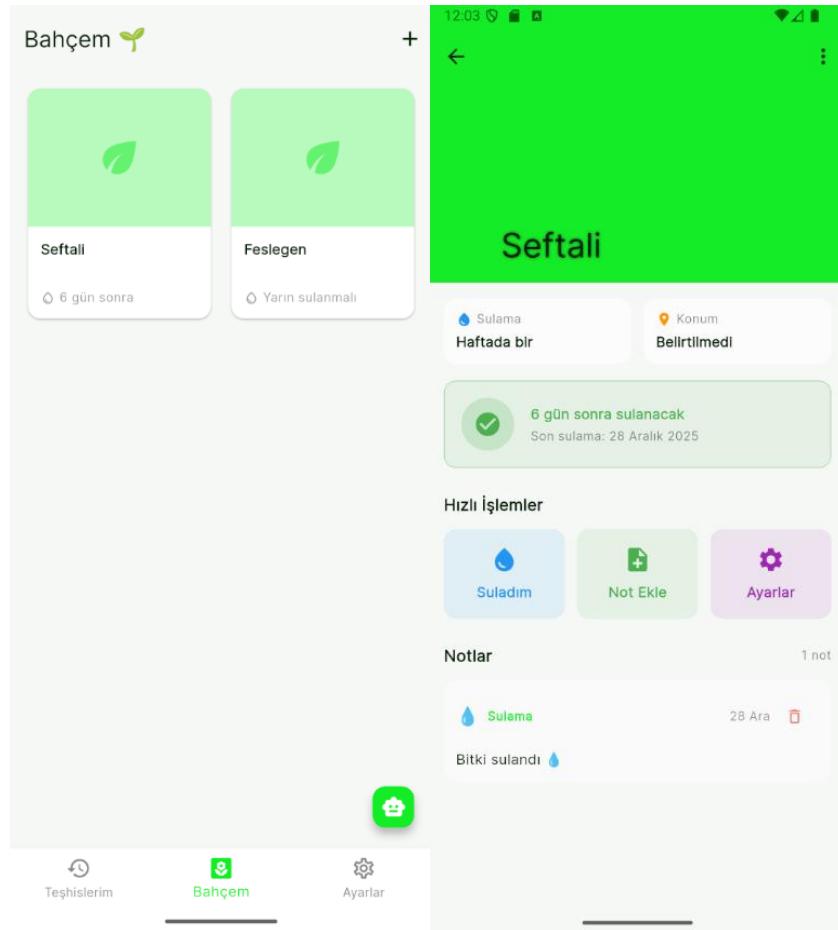


Şekil 3.9: Hastalık teşhisi ekran görüntüleri. Görüntü seçim ekranı, Analiz süreci, Teşhis sonucu ve tedavi önerileri.

3.6.4.2 Bahçe Yönetimi (Sanal Bahçe)

Kullanıcılar, tespit edilen hastalıklı bitkilerini uygulamadaki sanal bahçelerine ekleyerek takip edebilmektedir. Bu modül şu özelliklerini içermektedir:

- Bitki ekleme, düzenleme ve silme
- Her bitki için teşhis geçmişi görüntüleme
- Bakım notları ve hatırlatıcılar ekleme
- Tedavi sürecinin takibi



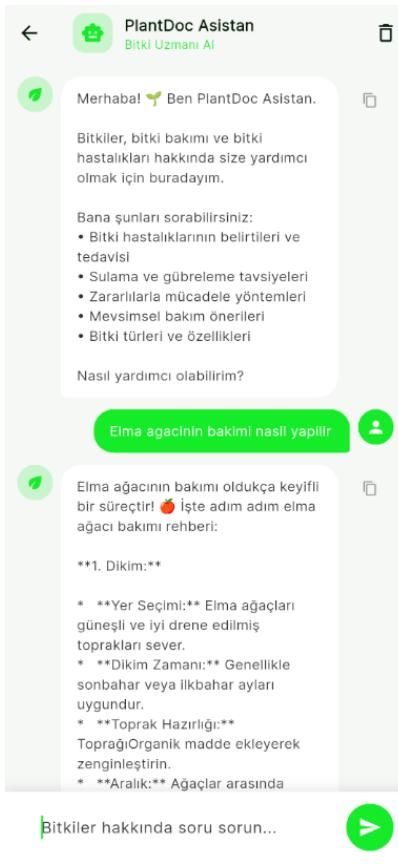
Şekil 3.10: Bahçe yönetimi ekran görüntüleri. (a) Bahçem listesi, (b) Bitki detay sayfası.

3.6.4.3 Yapay Zeka Sohbet Asistanı

Uygulama, Google Gemini API kullanılarak geliştirilmiş bir yapay zeka sohbet asistanı (chatbot) içermektedir. "PlantDoc Asistan" olarak adlandırılan bu chatbot, yalnızca bitkiler, bitki bakımı ve bitki hastalıkları konularında sorulara yanıt vermek üzere özelleştirilmiştir. Sistem önergesi (system prompt) aracılığıyla chatbot'un uzmanlık alanı sınırlanmış ve konu dışı sorulara nazikçe yönlendirme yapması sağlanmıştır.

Chatbot'un sunduğu özellikler:

- Bitki hastalıkları hakkında detaylı bilgi
- Tedavi yöntemleri ve ilaç önerileri
- Mevsimsel bakım tavsiyeleri
- Gübreleme ve sulama rehberliği
- Zararlılarla mücadele yöntemleri



Şekil 3.11: Yapay zeka sohbet asistanı ekran görüntüsü.

3.6.4.4 Çevrimdışı Çalışma ve Bulut Senkronizasyonu

Uygulama, internet bağlantısı olmadan da temel işlevlerini yerine getirebilecek şekilde tasarlanmıştır. TensorFlow Lite modelleri cihaz üzerinde çalıştığından, hastalık teşhis işlemi tamamen çevrimdışı (offline) olarak gerçekleştirilebilmektedir.

Kullanıcı verileri, Hive yerel veritabanında saklanmakta ve internet bağlantısı mevcut olduğunda Supabase bulut servisi ile senkronize edilmektedir. Bu hibrit yaklaşım şu avantajları sağlamaktadır:

- İnternet bağlantısı olmadan kesintisiz kullanım
- Verinin birden fazla cihaz arasında senkronizasyonu
- Otomatik yedekleme ve veri güvenliği

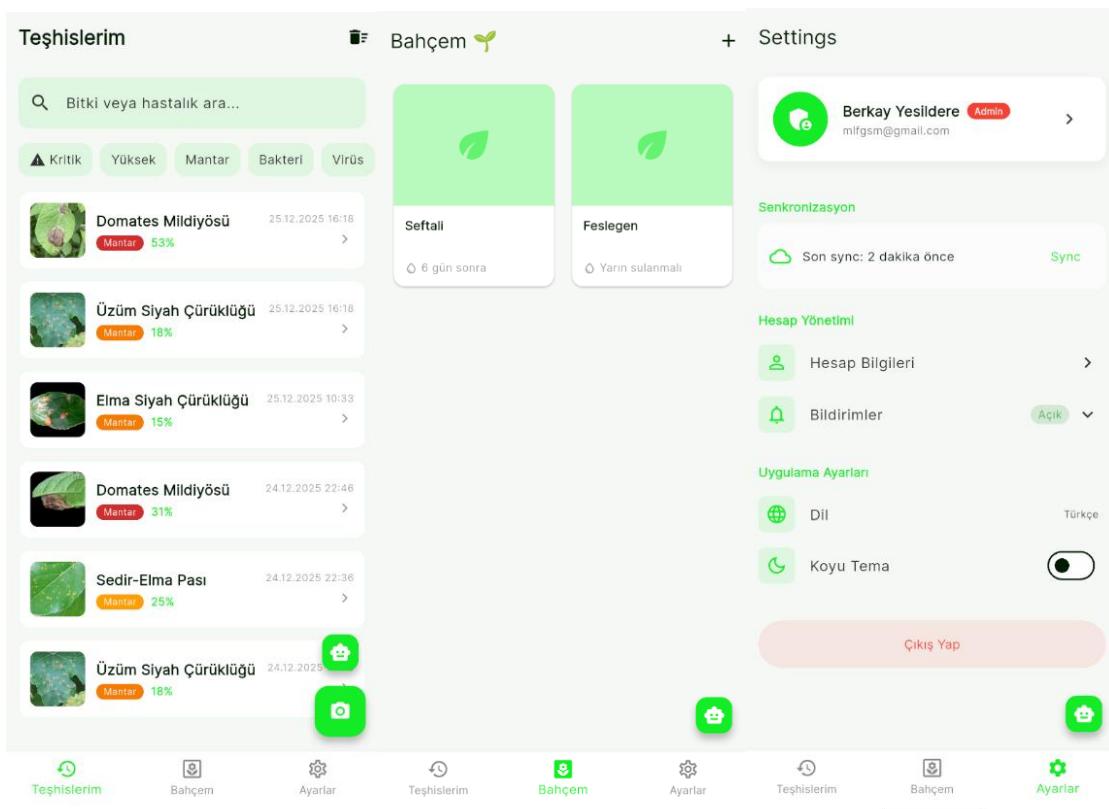
3.6.5 Kullanıcı Arayüzü Tasarımı

Uygulama arayüzü, Google'in Material Design 3 tasarım sistemine uygun olarak geliştirilmiştir. Kullanıcı deneyimini iyileştirmek amacıyla koyu tema (dark mode) desteği eklenmiş olup, kullanıcılar tercihlerine göre tema değiştirebilmektedir.

Ana navigasyon yapısı, alt navigasyon çubuğu (bottom navigation bar) aracılığıyla üç ana sekmeden oluşmaktadır:

1. Teşhislerim: Yapılan teşhislerin kronolojik geçmişi, arama ve filtreleme özellikleri
2. Bahçem: Ekleneen bitkilerin listesi ve yönetimi
3. Ayarlar: Kullanıcı profili, tema tercihleri ve senkronizasyon ayarları

Ayrıca, sayfalar arasında parmak kaydırma (swipe gesture) ile geçiş yapılmaktadır. Bu özellik, kullanıcı deneyimini daha akıcı hale getirmektedir.



Şekil 3.12: Uygulama ana ekran görüntüleri. (a) Teşhislerim sekmesi, (b) Bahçem sekmesi, (c) Ayarlar Bölümü.

4. BULGULAR

Bu bölümde, önerilen derin öğrenme modelinin eğitim sonuçları, performans metrikleri ve geliştirilen mobil uygulamanın test sonuçları sunulmaktadır. Modelin başarımı, doğruluk (accuracy), kayıp (loss), kesinlik (precision), duyarlılık (recall) ve F1-skoru gibi metrikler kullanılarak değerlendirilmiştir.

4.1. Model Eğitim ve Performans Sonuçları

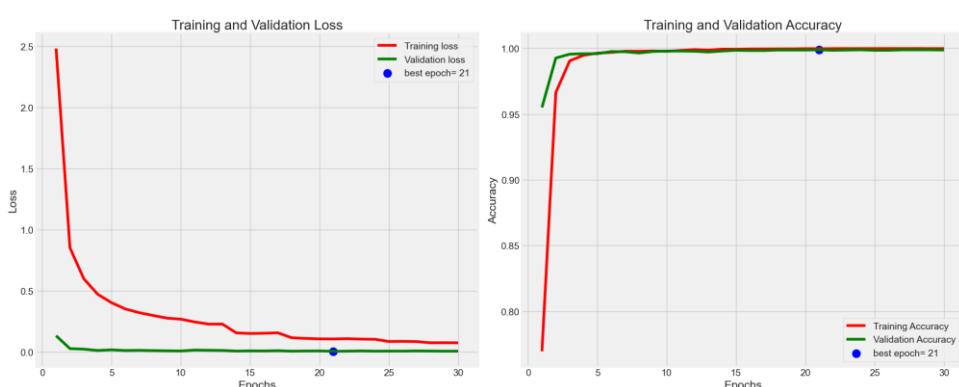
Bitki hastalıklarının tespiti için kullanılan EfficientNet-B3 tabanlı model, PlantVillage veri seti üzerinde eğitilmiştir. Eğitim süreci toplam 30 epok sürmüştür, "Early Stopping" mekanizması sayesinde modelin ezberlemesi (overfitting) engellenmiştir. Eğitim sonucunda en iyi performans 21. epokta elde edilmiştir.

Modelin eğitim, doğrulama (validation) ve test setleri üzerindeki performans değerleri Tablo 4.1'de özetlenmiştir.

Veri Seti	Kayıp (Loss) Değeri	Doğruluk (Accuracy)
Eğitim	0.0004	%100.00
Doğrulama	0.0069	%99.89
Test	0.0053	%99.85

Tablo 4.1: Modelin Eğitim, Doğrulama ve Test Performans Değerleri

Eğitim süreci boyunca kaydedilen kayıp ve doğruluk değerlerinin değişimi grafiksel olarak incelenmiştir.



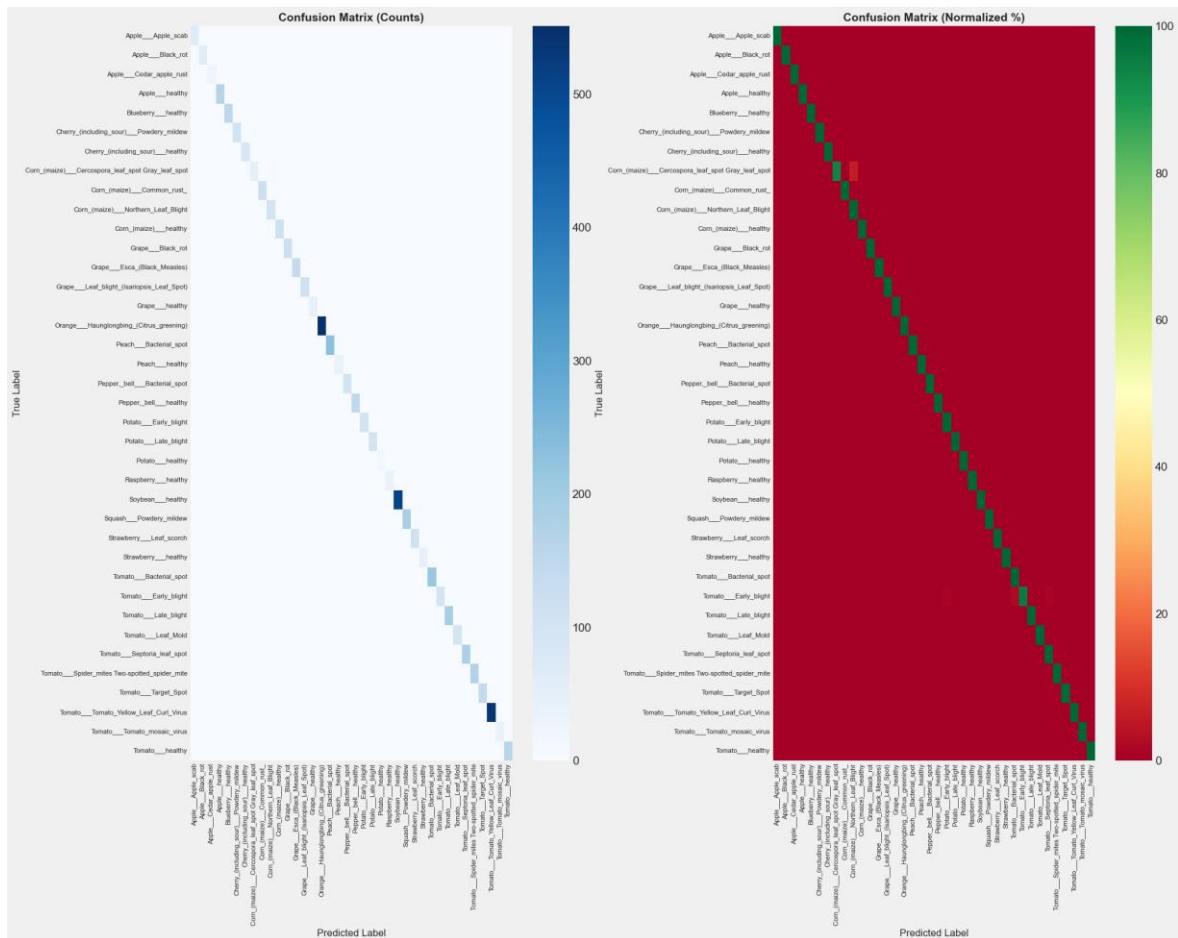
Şekil 4.1: Eğitim ve Doğrulama Kayıp/Doğruluk Grafiği.

Grafikler incelediğinde, modelin eğitim ve doğrulama kayıplarının paralel bir şekilde azaldığı ve doğruluk değerlerinin yükseldiği görülmektedir. Bu durum, modelin başarılı bir şekilde genellemeye yaptığı ve ezberleme (overfitting) sorunu yaşamadığını göstermektedir.

4.2. Sınıflandırma Performansı ve Karışıklık Matrisi

Modelin sınıf bazında performansını detaylı incelemek için Sınıflandırma Raporu (Classification Report) ve Karışıklık Matrisi (Confusion Matrix) oluşturulmuştur. Test veri seti üzerinde yapılan değerlendirmede, 38 farklı sınıfın tamamında %99 ve üzeri F1-skoru elde edilmiştir.

Özellikle birbirine görsel olarak çok benzeyen hastalık sınıflarında (örneğin Domates Erken Yanıklık ve Geç Yanıklık) modelin ayırt etme başarısı oldukça yüksektir. Aşağıdaki Şekil 4.3'te modelin test veri seti üzerindeki karışıklık matrisi verilmiştir.



Şekil 4.3: Karışıklık Matrisi (Confusion Matrix).

Elde edilen sonuçlar, önerilen EfficientNet-B3 modelinin bitki hastalıklarını tespit etmede oldukça güvenilir ve yüksek başarımlı olduğunu kanıtlamaktadır.

4.3. Mobil Uygulama Performans Testleri

Eğitilen PyTorch modeli, mobil cihazlarda çalışabilmesi için ONNX formatına ve ardından TensorFlow Lite (TFLite) formatına dönüştürülmüştür. Geliştirilen "Plant Disease Detector" Flutter uygulaması, Samsung Galaxy M34 5G cihazı üzerinde test edilmiştir.

Yapılan testlerde aşağıdaki performans kriterleri ölçülmüştür:

- Model Boyutu:** Kuantizasyon ve optimizasyon işlemleri sonrası model boyutu yaklaşık **48 MB** seviyesine indirilmiştir. Bu, mobil uygulama boyutunun düşük tutulmasını sağlamıştır.
- Çıkarım Süresi (Inference Time):** Arka plan temizleme işlemi dahil edilmeden, sadece sınıflandırma modelinin bir görüntüyü analiz etme süresi ortalama **200 ms** olarak ölçülmüştür. Arka plan temizleme (U-2-Net) aktif edildiğinde bu süre artmakla birlikte, kullanıcı deneyimini olumsuz etkilemeyecek seviyelerdedir.
- Kaynak Tüketimi:** Uygulama çalışırken CPU ve RAM kullanımı, standart bir mobil kullanım senaryosu sınırları içerisinde kalmıştır.

Metrik	EfficientNet-B3 (Sınıflandırma)	U ² -Net-P (Segmentasyon)
Model Boyutu	~48 MB	~4.7 MB
Ortalama Çıkarım Süresi	~200 ms	~150 ms
RAM Kullanımı	~80 MB	~60 MB

Tablo 3.4: Mobil uygulama model performans metrikleri. Ölçümler Samsung Galaxy M34 (Android 16, 6GB RAM) cihazında gerçekleştirilmiştir.

5. TARTIŞMA VE SONUÇ

5.1. Tartışma

Bu çalışmada, bitki hastalıklarının erken teşhisi ve yönetimi için derin öğrenme tabanlı kapsamlı bir mobil sistem önerilmiştir. Elde edilen bulgular, Literatür Taraması (Bölüm 2) kısmında incelenen benzer çalışmalarla karşılaştırıldığında, önerilen sistemin rekabetçi ve birçok açıdan üstün sonuçlar verdiği görülmüştür.

Özellikle EfficientNet-B3 mimarisinin kullanılması, modelin parametre sayısı ve işlem gücü dengesi açısından mobil cihazlar için uygun bir tercih olduğunu göstermiştir. Birçok

çalışmada kullanılan ResNet veya VGG modellerine kıyasla daha hızlı ve yüksek doğruluklu sonuçlar alınmıştır. %99.85'lik test doğruluğu, modelin saha koşullarında da güvenle kullanılabileceğine işaret etmektedir.

Ayrıca, sadece hastalık tespiti yapmakla kalmayıp, kullanıcıya tedavi önerileri sunan, hava durumu takibi sağlayan ve Gemini API destekli bir AI asistanı içeren bütünsel bir yapı sunulması, bu çalışmayı sadece bir sınıflandırma problemi çözümünden öteye taşıyıp tam bir tarımsal yönetim aracı haline getirmiştir.

5.2. Sonuç

Tez kapsamında, tarımsal verimliliği artırmak ve ürün kayıplarını en aza indirmek amacıyla yapay zeka destekli bir bitki hastalık tespit sistemi geliştirilmiştir. PyTorch kütüphanesi kullanılarak eğitilen EfficientNet-B3 modeli, 38 farklı bitki-hastalık sınıfını yüksek doğrulukla tespit edebilmektedir. Geliştirilen model, Flutter tabanlı bir mobil uygulamaya entegre edilerek çiftçilerin ve hobi amaçlı bitki yetiştircilerinin kullanımına sunulmuştur.

Çalışmanın temel çıktıları şunlardır:

1. PlantVillage veri seti üzerinde **%99.85** test doğruluğuna sahip yüksek performanslı bir sınıflandırma modeli geliştirilmiştir.
2. Model, mobil cihazlarda çevrimdışı (offline) çalışabilecek şekilde optimize edilmiştir.
3. Tarımsal yönetimi kolaylaştıran, kullanıcı dostu arayüze sahip bir mobil uygulama prototipi ortaya konulmuştur.

5.3. Gelecek Çalışmalar

Gelecekte bu çalışmanın kapsamını genişletmek adına aşağıdaki geliştirmeler planlanmaktadır:

- **Veri Setinin Genişletilmesi:** PlantVillage veri setinde bulunmayan yerel bitki hastalıklarının ve farklı ürün çeşitlerinin veri setine eklenmesi.
- **Gerçek Zamanlı Video Analizi:** Hastalık tespitinin sadece fotoğraflar üzerinden değil, kamera görüntüsü üzerinden gerçek zamanlı olarak yapılabilmesi.
- **Çoklu Dil Desteği:** Uygulamanın küresel ölçekte kullanılabilmesi için farklı dil seçeneklerinin eklenmesi.
- **Topluluk Özellikleri:** Çiftçilerin birbirleriyle bilgi paylaşabileceği sosyal bir platform özelliğinin entegre edilmesi.

KAYNAKLAR

- [1]. [global-agriculture.com](https://www.global-agriculture.com/global-agriculture/up-to-40-percent-of-global-crop-production-is-lost-due-to-pests-and-diseases-every-year-fao/), "Upto 40 percent of Global Crop Production is lost due to Pests and Diseases every year: FAO", [Online]. <https://www.global-agriculture.com/global-agriculture/up-to-40-percent-of-global-crop-production-is-lost-due-to-pests-and-diseases-every-year-fao/>
- [2]. FAO , "The hidden health crisis: How plant diseases threaten global food security", [Online]. <https://www.fao.org/one-health/highlights/how-plant-diseases-threaten-global-food-security>
- [3]. F. S. Seyed Khamoushi, "Plant Disease Detection Using Advanced Deep Learning Methods", Yüksek Lisans Tezi, Bilgisayar Mühendisliği Bölümü, İstanbul Aydın Üniversitesi, 2025.
- [4]. Z. F. A. Ishkayyir, "Derin Öğrenmeye Dayalı Bitki Hastalıkları Tespiti İçin Hibrit Bir Model", Yüksek Lisans Tezi, Bilgisayar Mühendisliği Bölümü, Atatürk Üniversitesi, Erzurum, 2025.
- [5]. Elif Genç, "Bitki Hastalıklarının Yapay Zeka ile Erken Tespiti", Yüksek Lisans Tezi, Bilgisayar Mühendisliği Anabilim Dalı, Eskişehir Osmangazi Üniversitesi, 2025.
- [6]. E. Ünal Çayır, "Bitki Hastalıklarının Derin Öğrenme ile Sınıflandırılması", Yüksek Lisans Tezi, Bilgisayar Mühendisliği Ana Bilim Dalı, Gazi Üniversitesi, 2024.
- [7]. A. E. Kılıç, "Transfer Öğrenme Tabanlı Açıklanabilir Derin Öğrenme Yöntemleri Kullanılarak Bitki Hastalıklarının Sınıflandırılması", Yüksek Lisans Tezi, Bilgisayar Mühendisliği Anabilim Dalı, Necmettin Erbakan Üniversitesi, 2024.
- [8]. K. Raghuram ve M. D. Borah, "A Hybrid Learning Model for Tomato Plant Disease Detection using Deep Reinforcement Learning with Transfer Learning", *Procedia Computer Science*, c. 252, s. 341–354, 2025.
- [9]. D. Argüeso, A. Picon, U. Irusta, A. Medela, M. G. San-Emeterio, A. Bereciartua, ve A. Alvarez-Gila, "Few-Shot Learning approach for plant disease classification using images taken in the field", *Computers and Electronics in Agriculture*, c. 175, 105542, 2020.

- [10]. M. Gehlot ve G. C. Gandhi, ““EffiNet-TS”: A deep interpretable architecture using EfficientNet for plant disease detection and visualization”, *Journal of Plant Diseases and Protection*, c. 130, s. 413–430, 2023.
- [11]. M. Srivastava ve J. Meena, “Plant leaf disease detection and classification using modified transfer learning models”, *Multimedia Tools and Applications*, c. 83, s. 38411–38441, 2024.
- [12]. M. Nawaz, T. Nazir, A. Javed, M. Masood, J. Rashid, J. Kim, ve A. Hussain, “A robust deep learning approach for tomato plant leaf disease localization and classification”, *Scientific Reports*, c. 12, sy. 1, 18568, 2022.
- [13]. GeeksforGeeks, "ML | Introduction to Transfer Learning", GeeksforGeeks, 8 Ekim 2025. [Çevrimiçi]. Erişim Adresi: <https://www.geeksforgeeks.org/machine-learning/ml-introduction-to-transfer-learning>. [Erişim Tarihi: 17 Kasım 2025].
- [14]. A. Alidev, "PlantVillage dataset", Kaggle, 2022. [Çevrimiçi]. Available: <https://www.kaggle.com/datasets/abdallahalidev/plantvillage-dataset>. [Erişim Tarihi: 15.11.2025].
- [15]. Hughes, D. P., & Salathe, M. (2015). An open access repository of images on plant health to enable the development of mobile disease diagnostics. arXiv preprint arXiv:1511.08060.
- [16]. Cornell University, "Plant Pathology 2021 - FGVC8," Kaggle, 2021. [Çevrimiçi]. Available: <https://www.kaggle.com/competitions/plant-pathology-2021-fgvc8>. [Erişim Tarihi: 15.11.2025].
- [17]. Cornell University, "Plant Pathology 2020 - FGVC7," Kaggle, 2020. [Çevrimiçi]. Available: <https://www.kaggle.com/competitions/plant-pathology-2020-fgvc7>. [Erişim Tarihi: 15.11.2025].
- [18]. Qin, X., Zhang, Z., Huang, C., Dehghan, M., Zaiane, O. R., & Jagersand, M. (2020). U2-Net: Going deeper with nested U-structure for salient object detection. *Pattern Recognition*, 106, 107404.

[GitHub-Linki](#)