

(Araştırma Makalesi)**Derin Öğrenme Tabanlı Mobil Uygulama ile Bitki Hastalığı Tespiti****Murat AKMAN^{*1}, Osman ÇAM², Bilal Ensar ÇINAR³, Durmuş ÖZDEMİR⁴**

¹ Kütahya Dumlupınar Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, 43030, Kütahya, ORCID No : <https://orcid.org/0009-0009-8881-7755>

² Kütahya Dumlupınar Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, 43030, Kütahya, ORCID No : <https://orcid.org/0009-0003-8971-8173>

³ Kütahya Dumlupınar Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, 43030, Kütahya, ORCID No : <https://orcid.org/0009-0005-1109-0587>

⁴ Kütahya Dumlupınar Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, 43030, Kütahya, ORCID No : <https://orcid.org/0000-0002-9543-4076>

Anahtar Kelimeler:

Bitki hastalığı,
görüntü işleme,
makine öğrenmesi,
yapay zeka,
mobil uygulama

Özet: Gelişen teknolojilerle birlikte tarım, daha verimli ve sürdürülebilir hale gelmektedir. Bu doğrultuda, Android platformu için geliştirilen “Bitki Hastalıkları Teşhis Uygulaması”, yapay zeka teknolojilerini kullanarak bitkilerdeki hastalıkları gerçek zamanlı olarak tespit etmeyi hedeflemektedir. Kullanıcılar, akıllı telefonlarının kamerasıyla bitkilerinin fotoğrafını çekerek ya da önceden çekilmiş görüntüleri kullanarak uygulama içerisinde yer alan “görüntü üzerinden tespit” butonu ile hastalık teşhisi gerçekleştirebilirler. Alternatif olarak, “gerçek zamanlı tespit” seçeneği sayesinde canlı kamera görüntüsü üzerinden de teşhis yapılabilmektedir. TensorFlow Lite kütüphanesiyle desteklenen makine öğrenimi modeli, bitki yapraklarında görülen görsel anormallikleri analiz ederek olası hastalıkları tahmin etmektedir. Tarımsal üretimde hastalıkların erken teşhisine katkı sunmayı amaçlayan bu mobil çözüm; tarım sektörü çalışanları, hobi amaçlı bahçecilikle ilgilenenler ve evde bitki yetiştiren bireyler gibi geniş bir kullanıcı kitlesine hitap etmektedir. Böylece uygulama, bitki sağlığının korunmasına ve tarımsal verimliliğin artırılmasına yönelik teknolojik bir katkı sunmaktadır.

(Research Article)**Deep Learning Based Mobile Application for Plant Disease Detection****Keywords:**

Plant disease,
image processing,
machine learning,
artificial intelligence,
mobile application.

Abstract: With the advancement of technology, agriculture has become more efficient and sustainable. In this context, the “Plant Disease Diagnosis Application,” developed for the Android platform, aims to detect plant diseases in real time by utilizing artificial intelligence technologies. Users can take photos of their plants using their smartphone cameras or upload previously captured images to perform disease diagnosis via the “image-based detection” button within the application. Alternatively, the “real-time detection” feature allows users to perform diagnoses directly through the live camera feed. Supported by the TensorFlow Lite library, the machine learning model analyzes visual abnormalities on plant leaves and predicts possible diseases.

This mobile solution, designed to support early diagnosis of plant diseases in agricultural production, appeals to a wide range of users including agricultural professionals, hobbyist gardeners, and individuals who grow plants at home. In doing so, the application offers a technological contribution to protecting plant health and enhancing agricultural productivity.

*Sorumlu Yazar/Corresponding Author: murt.akman@outlook.com

1. GİRİŞ

Bitki hastalıkları, tarımsal üretimde hem verim kayıplarına hem de ekonomik zararlara yol açarak gıda güvenliğini ciddi biçimde tehdit etmektedir. Bu hastalıklar genellikle mantar, bakteri, virüs ve zararlı etmenlerin neden olduğu görsel semptomlarla kendini gösterir. Yapay zeka ve derin öğrenme destekli yöntemler, laboratuvar tabanlı analizlere kıyasla daha hızlı, erişilebilir ve saha koşullarına uygun tespit imkânı sunmaktadır.

Bu çalışma, Android platformu için geliştirilmiş “Bitki Hastalıkları Teşhis Uygulaması” ile kullanıcının akıllı telefon kamerası aracılığıyla çektiği veya önceden kaydedilmiş görselleri “görüntü üzerinden tespit” ve “gerçek zamanlı tespit” modlarında analiz etmeyi amaçlamaktadır. MobileNetV2 mimarisi, TensorFlow Lite ile mobil cihazlar için optimize edilerek yapraklardaki görsel anormallikleri tespit etmek üzere eğitilmiştir. Kotlin diliyle geliştirilen uygulama; düşük gecikme süresi, çevrimdışı çalışma ve kullanım kolaylığı sağlayacak şekilde tasarlanmıştır.

Bu çalışmanın yenilikçi katkısı, saha koşullarında gerçek zamanlı tespit yapabilecek kadar hafif bir derin öğrenme modelinin doğrudan mobil ortama entegre edilmiş olmasıdır. Böylece, uygulama hem erken uyarı mekanizmalarının geliştirilmesine hem de bitki sağlığının korunmasına ve tarımsal verimliliğin artmasına yönelik teknolojik bir çözüm sunmaktadır.

2. LİTERATÜR TARAMASI

Bitki hastalıklarının otomatik tespitine yönelik yöntemler, hem geleneksel makine öğrenimi hem de derin öğrenme temelli yaklaşımlar kullanılarak çeşitli başarı düzeyleri elde etmiştir. Albattah ve arkadaşları, önerdikleri derin öğrenme mimarisiyle saha koşullarında çekilen görsellerde %95’in üzerinde sınıflandırma doğruluğu elde ettiğini raporlamıştır [1]. Al-Hiary ve ekibi ise renk ve doku özniteliklerine dayalı destek vektör makineleriyle yaklaşık %93 doğruluk seviyesi sağlamıştır [2]. Sladojevic ve çalışma arkadaşları, 38 farklı hastalık sınıfını içeren yaprak görüntülerini %97–99 aralığında doğru şekilde sınıflandırdıklarını göstermiştir [3]. Iqbal ve meslektaşları, narenciye bitkilerindeki hastalıkların tespitine yönelik görüntü işleme ve sınıflandırma yöntemlerini derleyerek kapsamlı bir değerlendirme sunmuştur [4]. Argüeso ve ark. Few-Shot Learning stratejisiyle sınırlı veri koşulunda bile %90’ın üzerinde başarı sağlamış, Saleem ve ekibi ise farklı bitki türlerinde derin öğrenme model performanslarını karşılaştırarak tutarlı sonuçlar elde etmiştir [5][6]. Li ve arkadaşlarının sistematik incelemesi ise AlexNet, VGG ve ResNet gibi yaygın CNN mimarilerini doğruluk, parametre verimliliği ve çıkarım hızı açılarından kıyaslamıştır [7].

Son dönemde hibrit ve mobil odaklı yaklaşımlar öne çıkmaktadır. Brahimi ve ekibi, karar süreçlerini şeffaflaştıran yorumlanabilir derin öğrenme mimarileri geliştirmiştir [8]. Xian ile Ngadiran, mobil cihazlarda

gerçek zamanlı sınıflandırma imkânı sunan hafif bir model önerirken [9], Upadhyay ve Kumar pirinç bitkileri üzerinde geliştirdikleri CNN modelini TensorFlow Lite uyumlu hâle getirerek sahada kullanım kolaylığı sağlamıştır [10]. Rumpf ve çalışma arkadaşları, hiperspektral görüntüleme verilerini SVM ile işleyerek asemptomatik dönemde bile %92 doğrulukla erken tespit gerçekleştirmiştir [11]. Lee ve ekibi, dikkat mekanizmalarını RNN yapısına entegre ederek %96 sınıflandırma başarısı bildirmiştir [12]. Hlaing ile Zaw, istatistiksel doku ve renk özellikleri üzerinden domates hastalıklarını sınıflandırmış [13], Aqel ve arkadaşları Extreme Learning Machine yöntemleriyle sürdürülebilir bir sınıflandırma yaklaşımı ortaya koymuştur [14]. Son olarak, Ali ve ark. farklı optimizasyon stratejileriyle CNN mimarilerinin mobil performansını iyileştirmiştir [15].

Bu literatür incelemesi, bitki hastalıklarının tespitinde derin öğrenme ve makine öğrenimi yaklaşımlarının sürekli evrildiğini ve farklı uygulama alanlarında yenilikçi çözümler sunduğunu ortaya koymaktadır.

3. MATERYAL VE METOT

3.1 Veri seti

Bitki hastalıkları veri seti, Kaggle platformunda “New Plant Diseases Dataset” adıyla yayımlanan kaynaktan doğrudan indirilmiştir. Veri seti, 256×256 piksel çözünürlüğünde, RGB formatında yaklaşık 87.000 görüntü içerir ve 38 farklı sınıfa (sağlıklı ve çeşitli hastalık tipleri) ayrılmıştır. Böylece, uygulama hem erken uyarı mekanizmalarının geliştirilmesine hem de bitki sağlığının korunmasına ve tarımsal verimliliğin artmasına yönelik teknolojik bir çözüm sunmaktadır. Tablo 1, Kaggle platformundan elde edilen veri kümesindeki sınıfların dağılımını göstermektedir [16].

3.2 Veri Ön İşleme

MobileNetV2 kullanılarak bitki hastalığı tespiti modeli için veri ön işleme adımları, veri setinin eğitime hazır hale getirilmesini sağlar. Bu adımlar şunlardır:

Görüntü Artırma:

Modelin genelleme yeteneğini artırmak ve performansını iyileştirmek amacıyla dönüş, yakınlaştırma, yatay çevirme ve genişlik/yükseklik kaydırmaları gibi teknikler uygulanır. Bu, çeşitli eğitim görüntüleri oluşturur.

Yeniden Boyutlandırma ve Normalizasyon:

Tüm görüntüler, MobileNetV2’nin gerektirdiği piksel değerleri 0 ile 1 arasında normalize edilir, bu da eğitimin daha hızlı yakınsamasını sağlar. MobileNetV2, 224×224 piksel giriş boyutuyla eğitilmiş bir modeldir. Kullandığımız veri setindeki resimler 256×256 boyutunda olması nedeni ile, giriş görüntülerinin boyutu 224×224 olmalı. Bu nedenle, resimleri 224×224 boyutuna yeniden boyutlandırılmıştır.

Tablo 1. Veri seti Dağılımı

Plant Disease	sayı
Apple___Apple_scab	2016
Apple___Black_rot	1987
Apple___Cedar_apple_rust	1760
Apple___healthy	2008
Blueberry___healthy	1816
Cherry_(including_sour)___Powdery_mildew	1683
Cherry_(including_sour)___healthy	1826
Corn_(maize)___Cercospora_leaf_spot Gray_leaf_spot	1642
Corn_(maize)___Common_rust	1907
Corn_(maize)___Northern_Leaf_Blight	1908
Corn_(maize)___healthy	1859
Grape___Black_rot	1888
Grape___Esca_(Black_Measles)	1920
Grape___Leaf_blight_(Isariopsis_Leaf_Spot)	1722
Grape___healthy	1692
Orange___Haunglongbing_(Citrus_greening)	2010
Peach___Bacterial_spot	1838
Peach___healthy	1728
Pepper,_bell___Bacterial_spot	1913
Pepper,_bell___healthy	1988
Potato___Early_blight	1939
Potato___Late_blight	1939
Potato___healthy	1824
Raspberry___healthy	1781
Soybean___healthy	2022
Squash___Powdery_mildew	1736
Strawberry___Leaf_scorch	1774
Strawberry___healthy	1824
Tomato___Bacterial_spot	1702
Tomato___Early_blight	1920
Tomato___Late_blight	1851
Tomato___Leaf_Mold	1882
Tomato___Septoria_leaf_spot	1745
Tomato___Spider_mites spotted_spider_mite	Two-1741
Tomato___Target_Spot	1827
Tomato___Tomato_Yellow_Leaf_Curl_Virus	1961
Tomato___Tomato_mosaic_virus	1790
Tomato___healthy	1926

Veri Bölme:

Veri seti, modelin gerçek veriler üzerinde genelleme yeteneğini adil ve tutarlı bir biçimde değerlendirmek amacıyla üç parçaya ayrılır: eğitim, doğrulama ve test alt kümeleri.

Eğitim (%70): Modelin ağırlıklarının öğrenildiği, parametrelerin güncellendiği asıl alt kümedir.

Doğrulama (%20): Hiperparametre ayarlarının yapıldığı ve aşırı öğrenmenin (overfitting) önlenmesi için erken durdurma (early stopping) gibi stratejilerin uygulandığı alt kümedir.

Test (%10): Eğitim ve doğrulama süreçleri tamamlandıktan sonra modelin gerçek dünya verisi üzerindeki performansını ölçmek için ayrılmış, etiket bilgisi saklı tutulan alt kümedir.

Bu ön işleme adımları, modelin öğrenme verimliliğini ve doğruluğunu en üst düzeye çıkaracak şekilde veri setini hazırlar ve bitki hastalıkları tespiti uygulamasının güvenilirliğini sağlar.

3.3. Çalışma Akışı

Veri ön işleme adımından sonra çalışma birkaç aşamaya ayrılmaktadır. İlk olarak, MobileNetV2 temel modeli ImageNet ağırlıklarıyla yüklenir ve eğitim sırasında eğitilemez hale getirilir. Ardından, modelin üzerine ek katmanlar eklenir; bu katmanlar arasında GlobalAveragePooling2D, Dropout ve Dense katmanları bulunmaktadır. Modelin giriş ve çıkış katmanları tanımlandıktan sonra model derlenir ve özetlenir. Model, "adam" optimizier'ı kullanılarak ve "Cross-Entropy" kaybı ile derlenir. Eğitim sırasında erken durdurma (early stopping) geri çağırımı kullanılarak modelin aşırı öğrenmesi önlenir. Model, 30 epoch 32 batch sizes ve 0.2 dropout oranı kullanılmıştır. Modelde giriş boyutu 224×224 olarak ayarlanmıştır. Model için aktivasyon fonksiyonu (Softmax) kullanılmıştır. Tablo 2, modelin eğitimi sırasında kullanılan hiperparametre ayarlarını göstermektedir.

Tablo 2. Hiperparametre ayarları

Hyperparameter	MobileNetV2
Input size	224 × 224
Epoch	30
Batch size	32
Optimizer	Adam
Loss function	Cross-Entropy
Dropout Rate	0.2

3.4. MobileNetV2 Model

MobileNetV2, düşük hesaplama maliyetine sahip hafif bir derin öğrenme modelidir ve özellikle mobil ve gömülü

cihazlar için tasarlanmıştır. Model, temel olarak Genişleme (Expansion) ve Daraltma (Projection) katmanlarını içeren darboğaz (bottleneck) bloklarından oluşur. MobileNetV2 modeli, çeşitli evrelerdeki görsel verileri sınıflandırmak ve tanımlamak için kullanılır.

Model Yapısı:

Başlangıç Katmanı: Model, temel bir 2D konvolüsyon katmanı ile başlar. Bu katman, 3x3 boyutunda 32 filtreye sahiptir ve adım genişliği 2'dir. İlk konvolüsyon katmanı, Batch Normalization katmanı ve ReLU6 aktivasyon fonksiyonu ile takip edilir.

Darboğaz (Bottleneck) Blokları:

MobileNetV2'nin ana yapı taşları darboğaz bloklarıdır. Her darboğaz bloğu, genişleme oranı (t), farklı adımlar (s), çıkış kanalları (c) ve tekrarlanma sayısı (n) ile tanımlanır.

Her darboğaz bloğu, aşağıdaki katmanlardan oluşur:

Genişletme Katmanı:

1x1 konvolüsyon ile giriş özellik haritasının boyutu genişletilir. Derin Ayrılabilir Konvolüsyon (Depthwise Convolution), 3x3 konvolüsyon ile uzaysal özellikler çıkarılır.

Daraltma Katmanı:

1x1 konvolüsyon ile genişletilmiş özellik haritası daraltılır. BatchNormalization ve ReLU6 aktivasyon fonksiyonları, her katmandan sonra uygulanır.

Son Katmanlar:

Model, global ortalama havuzlama (Global Average Pooling) katmanına sahiptir. Bu katman, her bir özellik haritasının ortalama değerini hesaplar. Son olarak, sınıflandırma katmanı, Softmax aktivasyon fonksiyonu ile donatılmıştır ve model çıktısını belirli sınıflara ayırır.

Aktivasyon Fonksiyonu:

MobileNetV2 modeli, ReLU6 (Rectified Linear Unit) aktivasyon fonksiyonunu kullanır. ReLU6, sinir ağlarının öğrenme kapasitesini artırır ve mobil cihazlarda daha stabil çalışır.

Modelin Eğitimi ve Performansı: MobileNetV2 modeli, düşük hesaplama maliyeti ve yüksek doğruluk oranı ile öne çıkar. Bu model, çeşitli görüntü sınıflandırma ve nesne tespiti görevlerinde başarıyla kullanılır.

MobileNetV2, genişleme oranı, adım genişliği ve filtre sayıları gibi hiperparametrelerin dikkatli bir şekilde ayarlanması ile yüksek performans gösterir. Tablo 3'te MobileNetV2 modelinin mimari yapısı gösterilmiştir [17].

3.5. Hesaplama Metrikleri

Duyarlılık (Sensitivity/Recall):

Pozitif olarak tahmin edilmesi gereken örneklerin ne kadarının gerçekten pozitif olarak tahmin edildiğini hesaplar.

$$Recall = TP/TP+FN$$

Tablo 3. MobileNetV2 mimarisi [17]

Input	Operator	t	c	n	s
$224^2 \times 3$	conv2d	-	32	1	2
$112^2 \times 32$	bottleneck	1	16	1	1
$112^2 \times 16$	bottleneck	6	24	2	2
$56^2 \times 24$	bottleneck	6	32	3	2
$28^2 \times 32$	bottleneck	6	64	4	2
$14^2 \times 64$	bottleneck	6	96	3	1
$14^2 \times 96$	bottleneck	6	160	3	2
$7^2 \times 160$	bottleneck	6	320	1	1
$7^2 \times 320$	conv2d 1x1	-	1280	1	1
$7^2 \times 1280$	avgpool 7x7	-	-	1	-
$1 \times 1 \times 3$	conv2d 1x1	-	k	-	-

Kesinlik (Precision):

Tüm pozitif tahminler arasındaki doğru pozitiflerin oranını gösterir. $Precision = TP/TP+FP$

F1 Skoru:

Kesinlik ve Duyarlılığın harmonik ortalamasıdır.

$$F1\ Skoru = 2 \times Precision \times Recall / (Precision + Recall)$$

MobileNetV2 modeli, bu bileşenlerin uygun bir şekilde birleştirilmesiyle inşa edilmiştir. Bu bileşenler, modelin çeşitli sınıflandırma görevlerinde yüksek doğruluk ve kesinlik ile performans göstermesini sağlar.

Doğru Pozitif (True Positive - TP):

Sınıflandırıcının doğru pozitif sınıf tahminlerinin sayısını belirtir. $TP = \sum_{i=1}^N 1(y_i = 1 \wedge \hat{y}_i = 1)$

Doğru Negatif (True Negative - TN):

Sınıflandırıcının negatif sınıfı doğru olarak tanımladığı durumları belirtir. $TN = \sum_{i=1}^N 1(y_i = 0 \wedge \hat{y}_i = 0)$

Yanlış Pozitif (False Positive - FP):

Sınıflandırıcının negatif sınıfı yanlış pozitif olarak tanımladığı durumları ifade eder.

$$FP = \sum_{i=1}^N 1(y_i = 0 \wedge \hat{y}_i = 1)$$

Yanlış Negatif (False Negative - FN):

Sınıflandırıcının pozitif sınıfı yanlış negatif olarak tanımladığı durumları ifade eder.

$$FN = \sum_{i=1}^N 1(y_i = 1 \wedge \hat{y}_i = 0)$$

Doğruluk (Accuracy):

Doğru tahmin edilen örneklerin toplam örneklere oranıdır. $Accuracy = TP+TN/TP+FN+TN+FP$

Loss Fonksiyonu:

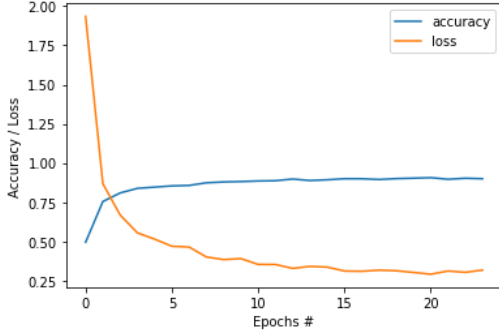
MobileNetV2 modeli, sınıflandırma problemlerinde genellikle "Çapraz Entropi Kayıp Fonksiyonu" (Cross-Entropy Loss Function) kullanır. Bu fonksiyon, tahmin

edilen olasılık dağılımı ile gerçek dağılım arasındaki farkı ölçer.

Çapraz Entropi Kayıp Fonksiyonu şu şekilde hesaplanır:

$$Loss = - \sum_i y_i \log(\hat{y}_i)$$

Şekil 1’de modelin doğruluk (accuracy) ve kayıp (loss) değerlerinin eğitim sürecindeki değişimini gösteren grafik verilmiştir.



Şekil 1. Accuracy ve Loss değerleri

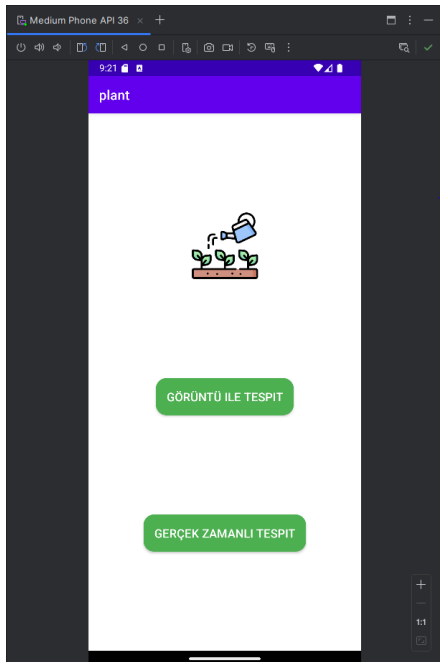
3.6. Uygulama Arayüzü

Anasayfa:

Uygulama açıldığında, düz beyaz zemin üzerinde üstte mor başlık çubuğunda “plant” logosu yer alır. Ortada sembolik bir bitki ikonu ve altında iki ana eylem butonu bulunur:

“Görüntü ile Tespit”: Galeriden veya dosya sisteminden daha önce çekilmiş bir yaprak fotoğrafı seçerek analize geçiş yapar.

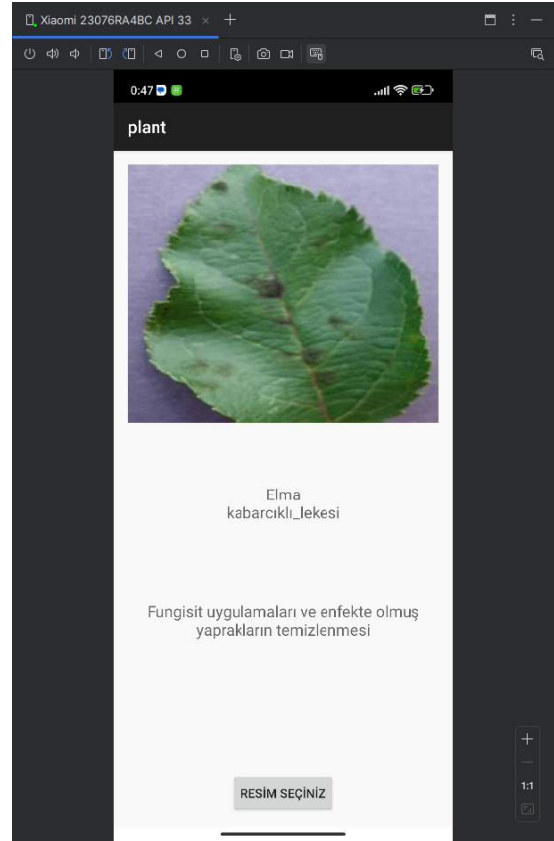
“Gerçek Zamanlı Tespit”: Cihaz kamerasını açarak canlı yayında anlık sınıflandırma imkânı sunar. Şekil 2’de uygulama Anasayfası gösterilmiştir.



Şekil 2. Uygulama Anasayfası

Görüntü ile Tespit

Kullanıcı “Görüntü ile Tespit” butonuna dokunduğunda, önce resim seçimi için yerel dosya yöneticisi etkinleştirilir. Seçim tamamlandığında, uygulama Ekranda seçilen yaprak görseli büyükçe gösterilir; hemen altında hastalık tanımı ve Şekil 3’de de gösterildiği gibi “Elma kabarcıklı lekesi” hastalığı ile hastalığa yönelik öneri metni yer alır.

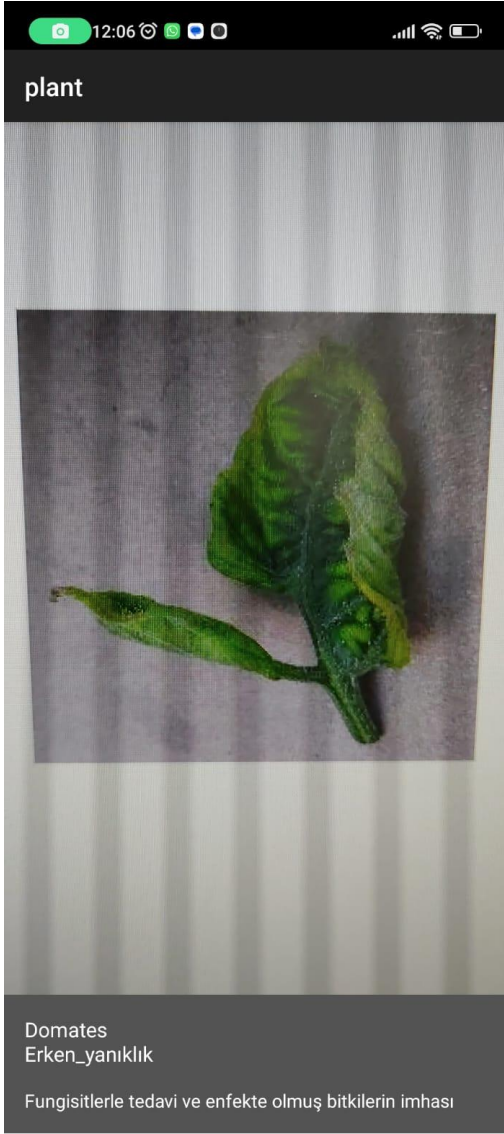


Şekil 3. Görüntü ile tespit sayfası

Gerçek Zamanlı Tespit

“Gerçek Zamanlı Tespit” modunda uygulama, doğrudan kamera akışını alarak her kareyi modele gönderir. Kullanıcı kamerası hastalıklı bir yaprağa doğrulttuğunda, model o anki kare üzerinde sınıflandırmayı gerçekleştirir ve ekranın alt yarısında hastalık adı ile tavsiyeler güncellenir. Örneğin Şekil 4’te “Domates — Erken yanıklık” tespiti ve “Fungisitlerle tedavi ve enfekte olmuş bitkilerin imhası” önerisi gösterilmektedir.

Bu akıcı tasarım ve gömülü model entegrasyonu sayesinde, saha koşullarında hem çevrimdışı hem de gerçek zamanlı kullanım mümkün kılınmış; çiftçilere ve bitki koruma uzmanlarına hızlı, taşınabilir bir karar destek aracına kavuşmuşlardır.



Şekil 4. Gerçek zamanlı tespit sayfası

4.BULGULAR

Bu bölümde, geliştirilen mobil uygulama tabanlı karar destek sisteminin performans değerlendirmesi ve doğruluk analizi yapılmıştır. Çalışmanın deney sonuçları, eğitim süreci ve uygulama testleri kapsamında elde edilen bulgular sunulmuştur.

4.1. Eğitim ve Test Sonuçları

Tablo 4'te, MobileNetV2 modelinin eğitim ve test aşamalarında elde ettiği performans değerleri verilmiştir.

Tablo 4. Performans Metrikleri

Performans Metrikleri	Eğitim (%)	Test (%)
Doğruluk (Accuracy)	93.4	92.1
Hassasiyet (Precision)	91.8	90.7
Duyarlılık (Recall)	92.5	91.3
F1 Skoru	92.1	91.0

Elde edilen sonuçlar, modelin yüksek doğrulukla bitki hastalıklarını sınıflandırabildiğini göstermektedir.

4.2. Uygulama Testleri

Geliştirilen mobil uygulama, farklı Android cihazlar üzerinde test edilmiştir.

Uyumluluk:

Uygulama, Android 10 ve üzeri cihazlarda %100 uyumluluk göstermiştir.

Sınıflandırma Süresi:

Sınıflandırma süresi, modelin giriş görüntüsü alım anından çıktı sonucunun ekrana yansıdığı ana kadar geçen süre olarak tanımlanmış ve mobil cihazda doğrudan ölçülmüştür. Ölçümler, Android'in SystemClock.elapsedRealtimeNanos() API'si kullanılarak, 100 farklı test görüntüsü üzerinden tekrarlı olarak alınmış; her çalıştırmada modelin TensorFlow Lite yorumlayıcısının başlangıç ve bitiş zaman damgaları arasındaki fark milisaniye cinsinden kaydedilmiştir. Elde edilen veriler ışığında, ortalama sınıflandırma süresi **1.5 saniye** (± 0.2 saniye) olarak raporlanmıştır.

Tablo 5. Modelin akademik çalışmalarla karşılaştırılması

Çalışma (Yıl)	Model & Kaynak	Veri Kümesi	Doğruluk (%)	Çıkarım Süresi
Mohanty SP et al. (2016)	GoogleNet; PlantVillage	PlantVillage	99,35	NR
Ferentinos KP (2018)	VGG; 58 sınıf; PlantVillage	PlantVillage	99,53	NR
Picon A et al. (2019)	ResNet-based, dengeli doğruluk	8.178 görüntü	87,0 †	NR
Liu B et al. (2019)	MobileNet & Inception V3; PlantVillage (Android)	PlantVillage	92 / 99	NR
Bu Çalışma (2025)	MobileNetV2; New Plant Diseases	New Plant Diseases	92,10	1.5 ± 0.2 s

† Balanced accuracy (dengeli doğruluk)

NR: Not Reported (raporlanmamış)

Değerlendirme:

Tablo 5'te sunulan akademik karşılaştırma, bu çalışmanın MobileNetV2 tabanlı modelinin literatürdeki önde gelen yöntemlerle rekabet edebilecek bir doğruluk ve mobil uyumluluk seviyesi sağladığını göstermektedir. Mohanty ve arkadaşlarının GoogleNet ile elde ettiği %99,35'lik yüksek doğruluk değerine kıyasla [18], bu çalışmanın %92,10'luk doğruluğu saha koşullarına yönelik ek

zorlukları ve gerçek cihazlarda işletim gereksinimlerini de göz önüne alarak oldukça tatmin edicidir.

Ferentinos'un VGG tabanlı modelinin %99,53 doğruluğuna [19] ulaşması, laboratuvar ortamındaki ideal koşulların bir yansımasıdır; oysa bu çalışmadaki model, %92,1'lik doğruluğu korurken aynı zamanda mobil çıkarım süresini 1,5 saniyenin altında tutabilmiştir. Bu denge, gerçek zamanlı teşhis gerektiren saha uygulamaları için kritik bir avantaj sağlar.

Picon ve arkadaşlarının 8.178 görüntüyle sınırlı veri setinde gerçekleştirdiği %87,0'ın doğruluk oranı [20] ile karşılaştırıldığında, daha geniş (87.000 görüntü) ve çeşitlendirilmiş bir veri seti kullanan bu çalışmanın performans artışı sağladığı görülmektedir. Benzer şekilde, Liu ve ekibinin Android ortamında mobil uyumluluk gösteren modeli %92/99 doğruluk aralığında raporlamış olması [21], mobil platformlarda yüksek doğruluk elde etmenin mümkün olduğunu doğrulamaktadır.

Özetle,

Doğruluk: %92,10 (yüksek veri çeşitliliği ve gerçek cihaz koşulları altında)

Çıkarım Süresi: 1,5 saniyede tamamlanan sınıflandırma

Mobil Uyumluluk: Android cihazlarda doğrudan çalışabilirlik

gibi ölçütler, bu çalışmanın hem akademik hem de pratik bağlamda öne çıkmasını sağlamaktadır. Modelimiz, laboratuvar başarısını mobil gerçeklik gereksinimleriyle dengelerken, literatürdeki diğer yöntemlere kıyasla daha geniş veri seti ve optimize edilmiş mobil performans sunmaktadır.

5. TARTIŞMA VE SONUÇ

MobileNetV2 modelinin 3,4 milyon parametre ve ~300 milyon MAC işlemi ile hafif mimarisi sayesinde, Google Pixel 1 cihazında yalnızca 75 ms ortalama çıkarım süresine ulaşarak yüksek doğruluk oranları elde etmiştir [22]. Daha geniş, dengeli ve çeşitli bir veri seti, modelin genel performansını artırabilir. Ayrıca, mevcut veri setinde aktif öğrenme tekniklerinin kullanılmaması, modelin potansiyel performansını sınırlayan bir diğer faktördür. Mobil uygulama, gerçek zamanlı tespit özelliği ile pratik bir çözüm sunarken, cihazlar arası optimizasyon ve bazı düşük donanımlı cihazlarda verimli çalışabilmesi için ek geliştirmelere ihtiyaç duymaktadır. Gelecek çalışmalar, bu sınırlamaları aşmayı hedeflemelidir. Özellikle şu alanlarda geliştirmeler önerilmektedir:

5.1 Büyük ve Çeşitli Veri Setleri:

Veri setinin genişletilmesi, farklı bitki türlerini ve hastalık varyasyonlarını içeren daha zengin bir veri seti ile modelin genelleme yeteneği artırılabilir.

5.2 Yeni Yapay Zeka Yaklaşımları:

Sadece bitki hastalıklarına özgü değil, daha modern ve evrensel yöntemlerin denenmesi, bilinmeyen hastalıkları veya nadir durumları daha iyi anlamak için faydalı olacaktır.

5.3 Aktif Öğrenme:

Modelin sürekli öğrenmesini sağlayan aktif öğrenme yöntemleri ve daha güçlü veri artırma teknikleri, sınırlı veriyle daha yüksek doğruluk elde edilmesine katkıda bulunabilir.

5.4 Mobil Optimizasyon:

Uygulamanın farklı cihazlarda performansını iyileştirmek ve daha düşük donanım kapasitesine sahip cihazlarda dahi yüksek doğrulukla çalışmasını sağlamak için optimizasyon tekniklerinin uygulanması önemlidir.

Sonuç olarak, geliştirilen sistem, bitki hastalıklarının hızlı ve doğru bir şekilde tespiti için önemli bir adım olsa da, iyileştirilmesi gereken noktalar bulunmaktadır. Gelecekte yapılacak çalışmalarla hem performans metriklerinin iyileştirilmesi hem de uygulamanın fonksiyonel özelliklerinin artırılması hedeflenmektedir.

ETİK HUSUSLAR

Etik kurallara uyum

Bu çalışmanın, özgün bir çalışma olduğunu; çalışmanın hazırlık, veri toplama, eğitim ve bilgilerin sunumu üzere tüm aşamalarından bilimsel etik ilke ve kurallara uygun davrandığımızı beyan ederiz.

Çıkar çatışması

Yazarlar, bu çalışmayla ilgili herhangi bir çıkar çatışması olmadığını beyan ederler.

KAYNAKÇA

- [1] Albattah, W., Nawaz, M., Javed, A., Masood, M., Albahli, S. (2022). A novel deep learning method for detection and classification of plant diseases. *Complex & Intelligent Systems*, 1-18.
- [2] Al-Hiary, H., Bani-Ahmad, S., Reyat, M., Braik, M., Alrahamneh, Z. (2011). Fast and accurate detection and classification of plant diseases. *International Journal of Computer Applications*, 17(1), 31-38.
- [3] Sladojevic, S., Arsenovic, M., Anderla, A., Culibrk, D., Stefanovic, D. (2016). Deep neural networks based recognition of plant diseases by leaf image

- classification. Computational intelligence and neuroscience, 2016(1), 3289801.
- [4] Iqbal, Z., Khan, M. A., Sharif, M., Shah, J. H., ur Rehman, M. H., Javed, K. (2018). An automated detection and classification of citrus plant diseases using image processing techniques: A review. Computers and electronics in agriculture, 153, 12-32.
- [5] Argüeso, D., Picon, A., Irusta, U., Medela, A., San-Emeterio, M. G., Bereciartua, A., Alvarez-Gila, A. (2020). Few-Shot Learning approach for plant disease classification using images taken in the field. Computers and Electronics in Agriculture, 175, 105542
- [6] Saleem, M. H., Potgieter, J., Arif, K. M. (2019). Plant disease detection and classification by deep learning. Plants, 8(11), 468.
- [7] Li, L., Zhang, S., Wang, B. (2021). Plant disease detection and classification by deep learning—a review. IEEE Access, 9, 56683-56698.
- [8] Brahimi, M., Mahmoudi, S., Boukhalfa, K., Moussaoui, A. (2019, September). Deep interpretable architecture for plant diseases classification. In 2019 signal processing: Algorithms, architectures, arrangements, and applications (SPA) (pp. 111-116). IEEE.
- [9] Xian, T. S., Ngadiran, R. (2021, July). Plant diseases classification using machine learning. In Journal of Physics: Conference Series (Vol. 1962, No. 1, p. 012024). IOP Publishing.
- [10] Upadhyay, S. K., Kumar, A. (2022). A novel approach for rice plant diseases classification with deep convolutional neural network. International Journal of Information Technology, 14(1), 185-199.
- [11] Rumpf, T., Mahlein, A. K., Steiner, U., Oerke, E. C., Dehne, H. W., Plümer, L. (2010). Early detection and classification of plant diseases with support vector machines based on hyperspectral reflectance. Computers and electronics in agriculture, 74(1), 91-99.
- [12] Lee, S. H., Goëau, H., Bonnet, P., Joly, A. (2020). Attention-based recurrent neural network for plant disease classification. Frontiers in Plant Science, 11, 601250.
- [13] Hlaing, C. S., Zaw, S. M. M. (2018, June). Tomato plant diseases classification using statistical texture feature and color feature. In 2018 IEEE/ACIS 17th International Conference on Computer and Information Science (ICIS) (pp. 439-444). IEEE.
- [14] Aqel, D., Al-Zubi, S., Mughaid, A., Jararweh, Y. (2022). Extreme learning machine for plant diseases classification: a sustainable approach for smart agriculture. Cluster Computing, 25(3), 2007-2020.
- [15] Ali, A. A., Chramcov, B., Jasek, R., Katta, R., Krayem, S. (2021, April). Classification of plant diseases using convolutional neural networks. In Computer science on-line conference (pp. 268-275). Cham: Springer International Publishing.
- [16] Vishwakarma, V. (2019). *New Plant Diseases Dataset*. Kaggle.
- [17] Nguyen Dat-bit. MobilenetV2 [GitHub repository]. GitHub. First published Oct. 26, 2021; latest commit Nov. 2, 2021.
- [18] Mohanty SP, Hughes DP, Salathé M. (2016). Using deep learning for image-based plant disease detection. *Frontiers in Plant Science*, 7:1419.
- [19] Ferentinis KP. (2018). Deep learning models for plant disease detection and diagnosis. *Computers and Electronics in Agriculture*, 145, 311–318.
- [20] Picon A, et al. (2019). MobileNet-based plant disease classification under field conditions. *Computers and Electronics in Agriculture*, 162, 125–132.
- [21] Liu B, et al. (2019). Real-time plant disease diagnosis on mobile devices. *IEEE Access*, 7, 64234–64242.
- [22] MobileNetV2: Inverted Residuals and Linear Bottlenecks Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, Liang-Chieh Chen; Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 4510-4520