



Araştırma Makalesi (Research Article)

Gönderi (Received) 14/06/2024  
Kabul (Accepted) 18/08/2024  
Yayın (Published) 31/08/2024

## Yenilebilir Yabani Bitki Görüntülerinin Derin Öğrenme Tabanlı Sınıflandırılması: Mobil Uygulama Örneği

Mohamad ALHAJ RABIA <sup>a</sup>, İrem Nur ECEMİŞ <sup>a,\*</sup>, Mustafa KARHAN <sup>a</sup>

<sup>a</sup> Bilgisayar Mühendisliği Bölümü, Mühendislik Fakültesi, Çankırı Karatekin Üniversitesi, Çankırı, TÜRKİYE  
\* Sorumlu yazar e-mail adresi: iremnurecemis@karatekin.edu.tr

### Özet

Bu çalışmada, yenilebilir yabani bitkilerin (YYB) tanımlanması amacıyla geliştirilen bir mobil uygulama tanıtılmaktadır. MobileNetV2 mimarisi kullanılarak oluşturulan derin öğrenme modeli, 35 farklı bitki türünü tanımlayabilmektedir. Model, toplamda 16.500 görüntüden oluşan geniş bir veri setiyle eğitilmiştir. Eğitim sürecinde veri artırma teknikleri kullanılarak modelin genelleme yeteneği geliştirilmiştir. Bu teknikler, görüntülerin döndürülmesi, kaydırılması, yakınlaştırılması ve yatay olarak çevrilmesi gibi çeşitli işlemleri içermektedir. Flask API aracılığıyla entegre edilen model, React Native ile geliştirilen mobil uygulama üzerinden kullanılabilir hale getirilmiştir. Uygulama, kullanıcıların bitkiler hakkında bilgi edinmelerini ve bu bilgileri favorilerine eklemelerine imkan sağlar. Ek olarak, kullanıcıların önceki taramalarını ve favori bitkilerini listelemelerine olanak tanır. Geliştirilen sistem, kullanıcı dostu arayüzü ve yüksek doğruluk oranıyla YYB'nin tanımlanmasında etkili bir çözüm sunmaktadır. Elde edilen sonuçlar, modelin eğitim doğruluğunun %85 ve doğrulama doğruluğunun %82 olduğunu göstermektedir. Bu çalışma, mobil cihazlar üzerinden YYB tanımlama alanında önemli bir adım atmakta ve doğa meraklıları, botanikçiler ve araştırmacılar için değerli bir araç sunmaktadır. Geliştirilen uygulama, benzer çalışmalara kıyasla daha geniş bir veri seti ve yüksek doğruluk oranı ile dikkat çekmektedir.

**Anahtar kelimeler:** Yenilebilir Yabani Bitkiler, Derin Öğrenme, MobileNetV2, React Native

## Deep Learning Based Classification of Edible Wild Plant Images: A Mobile Application Model

### Abstract

In this study, a mobile application developed for the identification of edible wild plants (YYB) is introduced. The deep learning model created using the MobileNetV2 architecture can identify 35 plant species. The model was trained with a large dataset consisting of 16,500 images in total. The model's generalization ability was increased by using data augmentation techniques during the training process. These techniques include various operations such as rotating, panning, zooming, and horizontally flipping the images. The model, integrated via the Flask API, was made available via the mobile application developed with React Native. The application allows users to obtain plant information and add it to their favorites. In addition, it allows users to list their previous scans and favorite plants. The developed system offers an effective solution for identifying YYB with its user-friendly interface and high accuracy rate. The results show that the model's training accuracy is 85%, and the validation accuracy is 82%. This study takes an essential step in identifying YYB via mobile devices and provides a valuable tool for nature enthusiasts, botanists, and researchers. The developed application attracts attention with its larger data set and higher accuracy rate than similar studies.

**Keywords:** Edible Wild Plants, Deep Learning, MobileNetV2, React Native

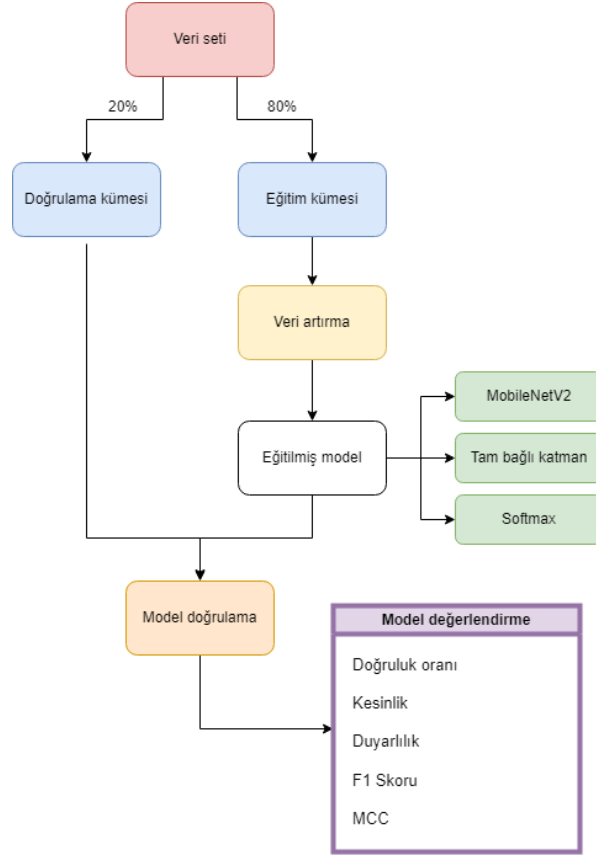
## 1. Giriş (Introduction)

Önemli geçim ve tıbbi kaynak sağlayıcıları olma rolleri nedeniyle yenilebilir yabani bitkilerin (YYB) insanlar tarafından algılanarak tanımlanması çok önemlidir. Gıda ve Tarım Örgütü (FAO), YYB'yi "Doğal veya yarı doğal ekosistemlerde kendi kendine yeten popülasyonlarda doğal olarak büyüyen ve doğrudan insan müdahalesi olmadan gelişebilen bitkiler" olarak tanımlamaktadır. YYB, alternatif anlamda, "Hiçbir zaman ekilmemiş ve evcilleştirilmemiş, ancak orijinal habitatlarından elde edilebilen bitki türleri" anlamına gelmektedir [1]. YYB'ler geçmişten günümüze gelen farklı yapıdaki toplulukların ve kültürlerin önemli bir parçası olmuştur. Ekili ürünlere takviye ürünler, kıtlık gıdaları ve gelir kaynakları olarak hizmet etmiştir [2-4]. Bu yabani bitkiler toplumda yaşayan yerli halklar için istikrarlı gıda kaynakları sağlamada ve gıda kıtlığı dönemlerinde bir geçiş yolu bulmada önemli bir rol oynamaktadır [5]. Bitki türleri açısından, meyveler ve ağaçlar insanlar tarafından tüketilen YYB'nin önemli ve baskın bileşenleri olarak vurgulanmaktadır [6]. Literatürde farklı ülkelerde farklı tür YYB üzerine yapılmış çeşitli çalışmalar bulunmaktadır. Hindistan'da [7], Nepal'de [8] ve Etiyopya'da [9] yürütülenler gibi YYB ile ilgili çeşitli araştırmalardan elde edilen bilgiler kullanılarak YYB hakkında geniş bir veritabanı oluşturulması tanınabilirlik açısından yarar sağlayabilmektedir. Her yerde bulunan bu bitkiler, doğru bir şekilde tanımlanırsa hayati bir besin kaynağı olarak hizmet edebilir. Bununla birlikte, doğru bilgilerin bulunmaması halinde zehirli bitkilerle karıştırılmasına yönelik potansiyel bir tehlike oluşturabilmektedir. Sağlıklı bir hayat sürdürebilmek için zehirli bitkilerden kaçınmaya öncelik vermek çok önemlidir. Bu bilgi, özellikle kamp veya yürüyüş gibi açık hava etkinlikleriyle uğraşan kişiler için hayati önem arz etmektedir. Bu özel başlık altında, bitkileri hassas bir şekilde tanıma ve ayırt etme yeteneğine sahip bir araç, kullanıcıların bitki örneklerini doğal ortamlarından güvenli bir şekilde toplamasına imkan sunacaktır.

Bu çalışmanın amacı, kullanıcıların YYB'yi doğru bir şekilde tanımasına yardımcı olmaktır. Buna ek olarak botanik örneklerin uygulamalarına ilişkin bilgi sağlamak yine günlük hayatı kolaylaştıran bir faktördür. Çalışma kapsamında geliştirilen akıllı telefon uygulaması, kullanıcılar için bitki tanımayı gerçekleştirmek için derin öğrenme yöntemi kullanmaktadır. Uygulamanın amacı, çeşitli bitki türlerine ilişkin genel bilgiler ve görsel temsiller sunarak kullanıcıların bitkileri doğal ortamlarında güvenli bir şekilde toplama becerilerini geliştirmektir. Bu, hem kişisel rahatlığın korunmasına hem de doğaya karşı duyulan ilginin artmasına katkı sağlamaktadır.

## 2. Materyal ve Metot (Material and Method)

Modellenen uygulamanın geliştirilme adımları Şekil 1'deki akış diyagramında sunulmuştur.



Şekil 1. Akış diyagramı (Flowchart)

## 2.1. Veri kümesi (Dataset)

Bu çalışmada derin öğrenme yöntemi kullanılarak yenilebilen yabancı bitkilerin sınıflandırılmasına yönelik kullanılan veri kümesi, Kaggle platformundan alınmıştır. Bu veri seti, 35 farklı YYB türüne ait yaklaşık 16.500 görüntü barındırmaktadır. Her sınıfta bulunan görüntü sayısı 400 ile 500 arasında farklılık göstermektedir. Bu görüntüler, Flickr'ın API'si aracılığıyla elde edilmiştir ve tüm halkın kullanımına açıktır [10]. Bu çeşitlilik, modelin doğruluğunu arttırmak ve farklı çevresel koşullarda bile doğru tanımlamalar yapabilmesini sağlamak için önemlidir. Veri seti, modelin eğitiminde ve testinde kullanılmak üzere %80 eğitim ve %20 doğrulama setlerine ayrılmıştır. Kullanılan veri setinin görsel örnekleri Şekil 2'de verilmiştir.

## 2.2. Derin öğrenme (Deep learning)

Yapılan çalışmada bitki türlerini sınıflandırmak için derin öğrenme algoritmaları kullanılmıştır. Veri artırma, kullanılan ilk tekniktir. Derin sinir ağlarının eğitimi için veri artırma, aşırı uyum riskini azaltmak için genellikle kapsamlı veri kümelerine dayanmaktadır. Ancak gerçek dünyada kullanım söz konusu olduğunda yalnızca az miktarda etiketlenmiş veriye erişilebilirlik sağlanabilmektedir. Veri artırma, eğitim verilerinin miktarını ve çeşitliliğini arttırmayı içeren bir tekniktir. Görüntü verileriyle çalışan derin öğrenme modellerinin eğitiminde yaygın olarak kullanılır. Eğitim verilerinin yeterliliğini ve çeşitliliğini arttırmadaki yüksek etkinliği ile bilinen bu teknik, görüntü verileri üzerinde çalışan derin öğrenme modellerinin başarılı uygulamalarına entegre edilmiştir [12].

Görüntü iyileştirme çok sayıda avantaj sunmaktadır. Öncelikle etiketli veri toplama ve açıklama ekleme gibi geleneksel yöntemlere kıyasla daha etkin bir yöntemdir. Bu fayda özellikle veri bağlamında kaynak kıtlığı olan çalışmalar için çok önemlidir. Dahası, görüntü büyütmenin kesinliği, orijinal verilere dayanmasından kaynaklanır, böylece makine öğrenimi modellerinin eğitimi ve

tahminlerin geliştirilmesi için sağlam bir temel oluşturulur. Büyütme sırasında görüntülerin kasıtlı olarak manipülasyonu, dengeli ve çeşitli veri kümelerinin oluşturulmasını kolaylaştırmaktadır. Bu durum modelin önceden kullanılmamış veriler üzerindeki genelleme yapma yeteneğini arttırmaktadır. Bunlara ek olarak, modellerin eğitim verilerine aşırı uyum sağlaması durumunu (overfitting) önlemede görüntü artırma yöntemi önemli bir rol oynamaktadır. Ayrıca görüntü artırma, gerçekleştirilen işlemler sonundaki test doğruluğunu arttırmaktadır. Arttırılmış verilerle eğitilmiş modeller, test doğruluğunu önemli ölçüde arttırmakta, bu da oluşturulan modelin güvenilirliğini ve performansını olumlu yönde etki etmektedir [13].



**Şekil 2.** Veri kümesi görüntü örnekleri [11] (Dataset image examples)

Keras'taki ImageDataGenerator sınıfı, verileri artırma sürecini basitleştirmektedir. ImageDataGenerator sınıfı, görüntü verilerini değiştirmek için kullanılacak bir dizi büyütme tekniği sunmaktadır. Keras ImageDataGenerator'a ek olarak, Albumentations paketi kullanılarak da benzer işlemler gerçekleştirilebilmektedir. Daha fazla veri ekleyerek eğitim verilerini arttırmakta, çeşitliliğini arttırmakta ve modelin daha geniş bir veri deseni yelpazesi elde etmesine olanak sağlamaktadır. Veri çeşitliliğinin artırılması, modelin sağlanan verilerin sınıflandırılmasındaki sağlamlığını ve kararlılığını arttırmaktadır [14].

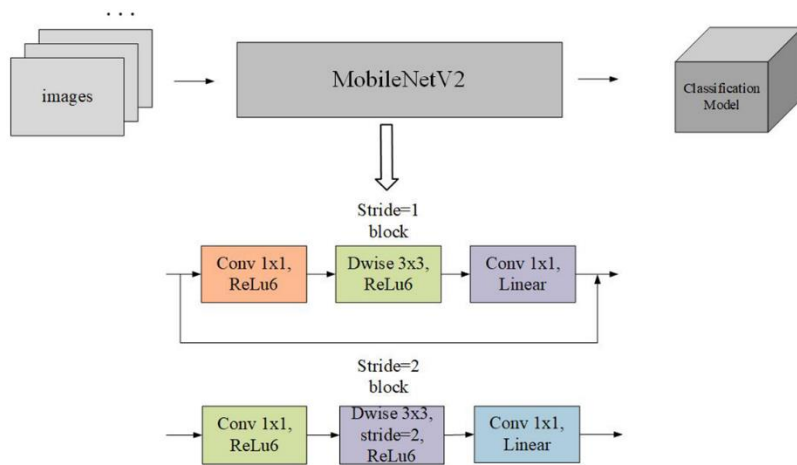
Veri kümesinin çeşitliliğini arttırmak ve modelin genelleme kapasitesini geliştirmek için veri artırma teknikleri uygulanmıştır. Kullanılan bu metodolojiler, modelin daha çeşitli bir veri kümesi kullanılarak eğitilmesini kolaylaştırır. Bu sayede gerçek hayattaki performansın yükselmesini sağlar. Yeniden ölçeklendirme, piksel değerlerini 0-255 aralığından 0-1 aralığına dönüştürerek kullanılan model için öğrenmenin hızını artırır. Döndürme aralığı özelliği, modelin, görüntülere 20 dereceye kadar rastgele döndürmeler uygulayarak çeşitli perspektiflerden çekilen bitki fotoğraflarını doğru bir şekilde tanımlamasını sağlar. Genişlik kaydırma aralığı parametresi, görüntüleri yatay olarak %20 büyüklüğünde rastgele hareket ettirerek modelin yatay yer değiştirmeleri iyi bir şekilde ele almasını sağlar. Yükseklik kaydırma aralığı işlevi, modelin dikey konumlandırmadaki değişiklikleri işleme yeteneğini geliştirmek için görüntülere %20'lik bir aralıkla rastgele dikey yer değiştirmeler sunar. Kayma aralığı işlevi, görüntülere 0,2'lik bir kayma uygulayarak modelin görüntülerdeki bozulmaları tespit etmesine yardımcı olur. Yakınlaştırma aralığı özelliği, fotoğraflara %20'lik rastgele yakınlaştırma uygulayarak modelin çeşitli ölçeklerdeki görüntüleri algılamasına yardımcı olur. Yatay çevirme işlevi görüntüleri yatay olarak rastgele yansıtır ve modelin yansıtılan görüntüleri tanımlamasına ve anlamasına olanak tanır. Doldurma işlevi, veri büyütme sırasında oluşabilecek boş alanları en yakın komşu piksellerin piksel değerleriyle doldurarak veri bütünlüğünü sağlar. Bu

stratejiler, modelin genel performansını ve genelleme kapasitesini arttırmaya yönelik önemli ve etkin adımlardır.

Transfer öğrenimi, yeni bir görev için öğrenme sürecini hızlandırmak amacıyla benzer işlerden önceden edinilen bilgilerden yararlanmayı içerir. Transfer öğrenimi, önceki öğrenme ve deneyimlerden, bu tür bilgiler olmadan elde edilebilecek olandan daha etkili bir şekilde yararlanarak yeni fakat birbiriyle bağlantılı kavramlar edinmeyi amaçlamaktadır. Transfer öğreniminin avantajı genellikle belirli öğrenme görevlerinde istenen performans düzeyini elde etmek için gereken eğitim örneklerinin miktarındaki azalmayla ölçülmektedir. Bu, ilgisiz problemler için gerekene kıyasla daha az sayıda örnek anlamına gelir ve örneklerin karmaşıklığında bir azalmaya yol açar. Bir öğrenici ilgili kavramları anladığında, sadece birkaç eğitim örneğini gözlemleyerek yeni bir kavramı veya süreci genellebilir ve anlayabilir [15].

Öte yandan, transfer öğrenme eğitim & test alanlarının ve görevlerin farklı olmasına izin verir. Temel amacı, seçilen alanlardan etiketlenmiş verileri ödünç almak veya kullanılan verilerden bilgi çıkarmak suretiyle bir makine öğrenme algoritmasının bu alandaki doğruluk sonuçlarını yükseltmesine yardımcı olmaktır. Bu nedenle yarı denetimli ve aktif öğrenmeye kıyasla minimal insan denetimi altında modeller öğrenmenin farklı bir stratejisi olarak görülebilir. Gerçek dünyada, transfer öğrenmenin birçok örneği görülmektedir [16].

Google ekibi, 2018 yılında MobileNetV2 ağını önermiştir. MobileNetV1 ağının en önemli özelliği, derinlik-seviyeli ayrılabilir konvolüsyonların kullanılması ve modeli küçültmek için iki hiperparametrenin eklenmesidir. Ancak, ağ yapısı VGG ağına benzer ve ResNet ağındaki kısa yol bağlantısına benzer bir bağlantı içermez. Bazı kullanıcılar, MobileNetV1 ağındaki DW konvolüsyonunun eğitim sürecinde hızla işlevini yitirdiğini ve ağın etkisinin tatmin edici olmadığını bildirmiştir. MobileNetV2 ağı, MobileNetV1 temel alınarak geliştirilmiş ve yapı, Tersine Artıklar (Inverted Residuals) ve Doğrusal Bottleneck tanıtılarak ayarlanmıştır [17]. Şekil 3'te MobileNetV2 ağ yapısı gösterilmiştir. MobileNetV1 ağının temelinde derinlik-seviyeli ayrılabilir konvolüsyon yer aldığı için, konvolüsyon yöntemi hesaplamaları ve parametre sayısını önemli ölçüde azaltabilir. Bu yapı, MobileNetV2'nin performansını büyük ölçüde arttıracaktır. Ayrıca, tersine artık tasarımı ReLU6 aktivasyon fonksiyonunu, son konvolüsyon katmanı ise doğrusal aktivasyon fonksiyonunu kullanır. Son konvolüsyon, aktivasyon fonksiyonundan önce özellik haritasını sıkıştırır ve ReLU aktivasyon fonksiyonu negatif giriş değerleri için 0 çıktısı verir, bu da bilgi kaybına neden olabilir. Doğrusal aktivasyon fonksiyonunun kullanılması, kaybedilen bilgiyi azaltabilir.



Şekil 3. MobileNetV2 mimari yapısı [17] (MobileNetV2 architecture structure)

Modelin eğitimi için MobileNetV2 transfer öğrenme yöntemi kullanılmıştır. MobileNetV2, mobil cihazlarda verimli çalışacak şekilde optimize edilmiş hafif ve güçlü bir derin öğrenme modelidir. Bu verimli model, hesaplama karmaşıklığını azaltmak ve performansı arttırmak için derin ayırık evrişimler tarafından sağlanır. Transfer öğrenme yöntemi ile önceden eğitilmiş model, yeni veri setine adapte edilerek yüksek doğruluk oranlarına (accuracy) ulaşmıştır. Bu çalışmada, geliştirilen model ile %85 eğitim doğruluğu ve %82 doğrulama doğruluğu elde edilmiştir.

MobileNetV2'nin mimarisi, 3x3 derinlik evrişimleri ve ardından 1x1 nokta evrişimlerinden oluşan bloklara dayanmaktadır. Derinlik evrişimleri giriş kanallarını bağımsız olarak filtreler, nokta evrişimleri ise bunların filtrelediği çıkışları birleştirir. Düzenli evrişimler hem filtreleme hem de toplama adımlarını bir arada gerçekleştirir, dolayısıyla hesaplama açısından çok ağırdırlar. MobileNetV2, bu işlemi ayırarak doğruluğu korurken hesaplama verimliliğini sağlamaktadır. MobileNetV2, modelin daha az veriyle genelleme yapma yeteneğini arttıran ters çevrilmiş artık bloklar ve doğrusal darboğazlar sunmaktadır. Tersine çevrilmiş artıklar dar katmanları, yani darboğaz katmanlarını daha geniş katmanlara bağlayarak özelliklerin verimli hesaplama maliyetleriyle çıkarılmasına olanak tanır. Doğrusal darboğazlar, boyutsallığın azaltılmasında temsil gücünün korunmasına yardımcı olur.

MobileNetV2'nin girişi 224 x 224 piksel renkli bir görüntüdür ve ağ, her bitki türü için bir olasılık puanı üretir. Bu süreç, tutarlılığı ve girdinin işlenmeye uygun şekilde hazırlanmasını sağlamak için görüntülerin yeniden boyutlandırılması ve normalleştirilmesiyle başlar. Bu çalışma kapsamında yapılan model eğitimi, 15 dönemde (epoch) 0,001 öğrenme oranında gerçekleştirilmiştir. Her dönem, tüm eğitim verilerinin üzerinden geçerek modelin parametreleri tekrar tekrar öğrenmesine ve değiştirmesine olanak tanır. Verimli aktarım öğrenimi için MobileNetV2 mimarisi, böylesine büyük ve çeşitli bir gövdeden öğrenilen özelliklerin uygulamaya konulması için sağlam bir başlangıç noktası görevi gören, önceden eğitilmiş ImageNet ağırlıkları kullanılarak başlatılır. Include\_top parametresi de False olarak ayarlanır, böylece son bağlanan katman tam olarak kullanılmaz. Bu durum, uygulamaya özel olarak uyarlanmış katmanlar eklenerek modeli bitki türleri sınıflandırmasının bu özel görevi için genişletilebilir hale getirir.

Bunlara ek olarak, bu model için yakınsama hızı ile eğitim kararlılığı arasında bir denge sağlarken, optimizasyon süreci sırasındaki adım boyutunun orta düzeyde olmasını sağlayacak bir öğrenme oranı seçilmiştir. Eğitimde, adaptif öğrenme oranı tekniği ve Adam optimizasyon algoritması kullanılmıştır. Adam, diğer iki stokastik gradyan inişi genişletmesi olan AdaGrad ve RMSProp'un kombinasyonudur. Adam, gradyanların birinci ve ikinci dereceden moment tahminlerine dayalı olarak her parametre için öğrenme oranlarını ayarlar, böylece gürültülü problemlerin varlığında bile seyrek gradyanlarla başa çıkabilir. Bu optimizatör, parametreleri güncellerken taşmayı önleyecek ve modeldeki performansı arttıracaktır. Softmax kaybı fonksiyonu, tahmin edilen olasılık dağılımları ile gerçek puanlar arasındaki çarpıklığı ölçmüştür. Bu yapı çok sınıflı sınıflandırma için popüler bir seçenektir ve farklı bölgelerde kullanıldığı bilinmektedir. Çünkü her sınıf için log kaybını hesaplar ve bunları toplayarak tek bir standart kayıp değeri sağlar. Bu kaybın en aza indirilmesi, tahmin edilen olasılıkların girdi verilerinin gerçek dağılımına mümkün olduğunca yakın bir şekilde eşleşeceği şekilde modelin ayarlanmasına yardımcı olur.

TensorFlow, makine öğrenimi algoritmalarını belirlemek için bir çerçeve ve bunları çalıştırabilmek için bir platform sunmaktadır. TensorFlow kullanılarak uygulanan bir hesaplama, cep telefonları ve tabletler gibi küçük ölçekli cihazlardan yüzlerce iş istasyonu ve binlerce GPU'dan oluşan büyük ölçekli dağıtılmış sistemlere kadar çok az değişiklik gerektirerek veya hiç değişiklik gerektirmeden çeşitli heterojen sistemler üzerinde yürütülebilir. Bu teknoloji son derece uyarlanabilir ve derin sinir ağlarının hem eğitim hem de çıkarım algoritmaları da dahil olmak üzere çok çeşitli algoritmaların

uygulanması için kullanılabilir. TensorFlow, bilgisayar bilimi alanında ve diğer çeşitli disiplinlerde araştırma yapmak ve makine öğrenimi sistemlerini operasyonel ortamlarda uygulamak için kullanılmıştır. Bu teknolojinin kullanımı, konuşma tanıma, bilgisayarla görme, robotik, bilgi alma, doğal dil işleme, coğrafi bilgi çıkarma ve hesaplamalı ilaç keşfi dahil olmak üzere ondan fazla alanı kapsamaktadır [18].

TensorFlow Lite (TF Lite), mobil, gömülü cihazlar için özel olarak geliştirilmiş bir işlevselliktir. TF Lite, bu dosya formatında optimize edilmiş bir sürüme dönüştürülmesini sağlar. TensorFlow, bir makine öğrenimi modelini optimize etmek, sıkıştırmak ve bahsedilen formata dönüştürmek için birçok farklı teknik sağlar. TensorFlow dünyada birçok uygulama yapımında kullanılmaktadır. Yapay zeka uygulamaları için makine öğrenmesi yöntemlerinin ve sonuçlarının uygulanmasını sağlamaktadır. TF Lite, özellikle yerel çıkarım için tasarlanmış bir hesaplamalı bir platformdur. Asıl olarak ucuz bilgisayar donanımlarında kullanılmak üzere tasarlanmıştır. Programcıların modellerini uyumlu donanım ve Nesnelerin İnterneti (IoT) cihazlarında yürütmesine olanak tanıyarak cihaz içinde kullanılan yapay zekayı kolaylaştırır [19].

TF Lite, TensorFlow gibi standart bir formatta veya ONNX tarafından dönüştürülebilir bir formatta eğitilmiş bir modeli alır ve bir TFLite model dosyası (.tflite) oluşturur. Dönüştürmeden sonra, model dosyası bir istemci cihazına dağıtılabilir. Ardından TFLite yorumlayıcısı kullanılarak yerel olarak çalıştırılabilir. TFLite, Android, IOS ve Linux (Raspberry Pi dahil) gibi mobil ve gömülü platformlarda birçok programlama dilinde çıkarım yapmayı destekler [20]. Geliştirilen model, TensorFlow kullanılarak eğitilmiş ve daha sonra TensorFlow Lite (tflite) formatına dönüştürülmüştür. TF Lite modelleri, mobil cihazlarda daha küçük boyutları ve hızlı çalışmaları sayesinde tercih edilmektedir. Bu dönüşüm, modelin mobil uygulamada kullanılabilmesini sağlamıştır.

### **2.3. Flask API (Flask API)**

Flask yapısı, istenilen cihaz üzerinde web uygulaması oluşturmaya olanak tanıyan kütüphaneler, araçlar ve yeni teknoloji imkanları sunan bir frameworktür. Bu imkanlar bir web uygulaması, bir blog, bir web sayfası veya bir takvim uygulaması olabilir. Flask, Python ile yazılmış hafif bir web uygulamanın ana yapılarından. WSGI araç takımı ve Jinja2 şablon motoru üzerine kuruludur. Flask, mikroframework kategorisine girer. Mikroframeworkler genellikle dış kütüphanelere çok az bağımlılığı olan veya hiç bağımlılığı olmayan frameworklerdir [21]. Ayrıca Flask, Python projeleri için geliştiricilerin kolayca web uygulamaları oluşturmalarını sağlamaktadır. Farklı ölçeklerdeki geliştirilen projelere entegre edilebilecek yapıdadır. Örneğin, basit çaplı web uygulamalarından, dünyanın her yanında kullanılan geniş çaplı sosyal ağ sitelerine kadar birçok alanda kullanılabilir. Temel prensipler aynıdır ve bu prensipler kolayca herhangi bir ölçekli web tabanlı uygulamaya uygulanabilir [22].

Modelin pratikte kullanılabilirliğini arttırmak için bir Flask uygulaması geliştirilmiştir. Bu yapı geliştirilen projede modelin tahminlerini yapabilmesi için bir API olarak kullanılmıştır. Flask uygulaması, tflite formatındaki modeli yüklemekte ve tahminler için hazır hale getirmektedir. Model, gelen görüntülerin işlenmesi ve tahminlerin yapılması için optimize edilmiştir. İstemci tarafından gönderilen görüntüleri alır ve tahmin sonuçlarını döndürür. İstemci tarafından base64 formatında gönderilen görüntüler, sunucu tarafında çözülür ve tflite modeline giriş olarak verilir. Model, görüntüyü işleyerek işlemler sonucunda tahmin edilen bitki türünü ve doğruluk oranını verir.

### **2.4. Mobil uygulama (Mobile application)**

React Native framework yapısı, React JavaScript Kütüphanesi'ni kullanarak JavaScript ile çapraz platformda derlenmiş mobil uygulamalar geliştirmek amacıyla kullanılan açık kaynaklı şekilde

sunulmuş bir frameworktür. React Native, kullanıcı arayüzleri oluşturmak için bir JavaScript kütüphanesi olan ReactJS'ye dayanır. React, başlangıçta Facebook tarafından web uygulamalarının kullanıcı arayüzlerini oluşturmak için geliştirilmiştir. Uygulamanın kullanıcı arayüzü elemanlarını yazmak için JavaScript içinde XML gömülü bir sözdizimi olan JavaScript XML'i (JSX) kullanmaktadır. React, birbirinden bağımsız organizasyonların ve geliştirici topluluğunun katkılarıyla gelişerek React Native kullanarak mobil uygulama geliştirmeyi destekleyecek hale evrilmiştir. Bu nedenle daha hızlı ve daha güvenilirlerdir [23, 24]. React Native ilk kez piyasaya sürüldüğünde sadece IOS desteğine sahip iken daha sonraki zamanlarda ana yapı üzerine Android desteği eklenmiştir. Günümüzde hala yapısal olarak genişlemeye devam etmektedir. Facebook, daha açık kaynaklı olmaya başlamış ve React Native için de bu yaklaşımı seçmiştir. Kaynak kodu tamamen açık değildir, yine de Facebook bu hedefe ulaşmayı amaçlamaktadır. Ayrıca toplumsal yapının frameworkü geliştirmeye katkıda bulunacağını düşünmektedir [25].

Çapraz platform mobil geliştirme frameworklerinin kullanımı, günümüzde mobil uygulamalar oluşturmanın popüler bir yoludur. React Native ve Expo, Android ve IOS uygulamaları oluşturmak için React kullanarak daha farklı yöntemler sunan iki popüler seçenektir. Expo, React Native'in üzerine bir soyutlama katmanı sağlamaktadır, bu da geliştirici deneyimini iyileştirmekte ve süreci basitleştirmektedir [26]. Expo, React Native yapısı kullanılarak oluşturulmuş yerel web ve mobil uygulamaları için geliştirilmiş bir frameworktür. Kamera ve GPS konumuna erişim gibi çeşitli geliştirme araçları ve özellikler sunmaktadır. Bir kütüphane oluşturulurken en önemli Expo özelliği, bir sunucuyu başlatma ve geliştirme amaçları için bir mobil emülatörde veya mobil cihazda uygulamayı çalıştırmayı kolaylaştıran Expo CLI olmuştur [27].

Mobil uygulama, React Native kullanılarak geliştirilmiştir ve kullanıcı dostu bir arayüze sahiptir. Uygulama, kendi içerisinde anlaşılabilirliği arttıran çeşitli ekranlar ve işlevler sunmaktadır. Uygulamanın ana ekranı, kullanıcıların galeri veya kamera aracılığıyla görüntü yüklemelerine olanak tanımaktadır. Yüklenen görüntü, base64 formatına dönüştürülerek Flask API'ye gönderilir ve tahmin sonucu alınır. Şekil 4a'da mobil uygulamanın ana ekranı gösterilmiştir.

Sonuç ekranı, modelin tahmin ettiği bitki türü hakkında bilgi sağlamaktadır. Bu ekranda bitkinin adı, 4 görüntüden oluşan bir slayt gösterisi, alternatif adları, açıklaması, yenilebilir kısımları ve veri kaynağı gibi bilgiler sunulur. Ayrıca, bitkinin favorilere eklenmesini sağlayan bir buton bulunmaktadır. Bu bitki bilgileri, çeşitli web sitelerinden toplanmış ve JSON formatında düzenlenmiştir. Şekil 4b'de mobil uygulamanın sonuç ekranı gösterilmiştir.

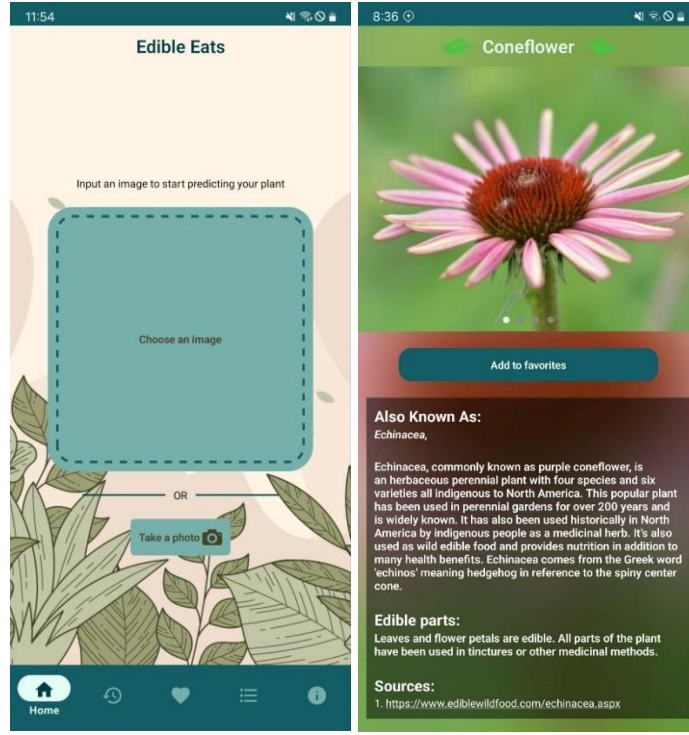
Favoriler ekranı, kullanıcıların favori bitkilerini ekleyip saklayabileceği bir liste sunmaktadır. Her bir favori öğesi için bitkinin adı, bitki resmi ve eklenme tarihi gösterilir. Kullanıcı, listedeki herhangi bir öğeye tıklayarak bitki sayfasına yönlendirilebilir. Şekil 5a'da uygulamanın favoriler ekranı gösterilmiştir.

Geçmiş ekranı, kullanıcıların daha önce tarattığı bitkilerin bir listesini göstermektedir. Her bir öğe için bitkinin adı, bitki resmi ve tarama tarihi gösterilir. Ayrıca, "hepsini sil" butonuna tıklanarak listedeki tüm öğeler silinebilir. Şekil 5b'de mobil uygulamanın geçmiş ekranı gösterilmiştir.

Bitkiler ekranı, modelin eğitiminde kullanılan tüm bitkilerin listesini tutmaktadır. Bu ekranda, uygulamanın sınıflandırabileceği tüm YYB gösterilir ve kullanıcı herhangi bir bitkiye tıkladığında bitki sayfasına yönlendirilir. Şekil 6a'da mobil uygulamanın bitkiler ekranı gösterilmiştir.

Hakkında ekranı, proje ve geliştirici hakkında bilgi sağlamaktadır. Bu ekranda, projenin GitHub deposunun bağlantısı bulunur ve mevcut uygulama sürümünü gösteren bir metin yer alır. Bu bilgiler, kullanıcıların proje hakkında daha fazla bilgi edinmesine ve geliştiriciyle iletişime geçmesine olanak tanımaktadır. Şekil 6b'de mobil uygulamanın hakkında ekranı gösterilmiştir.

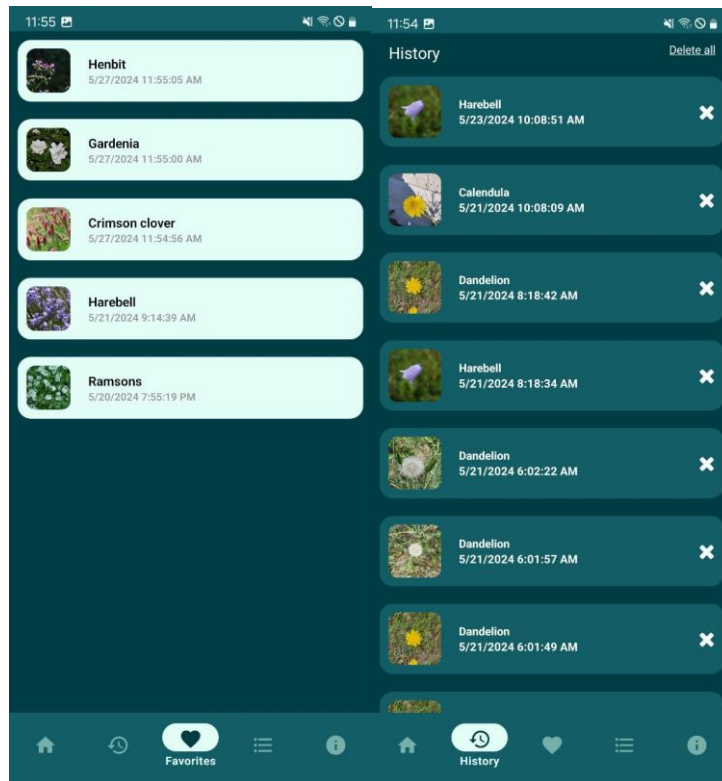




a)

b)

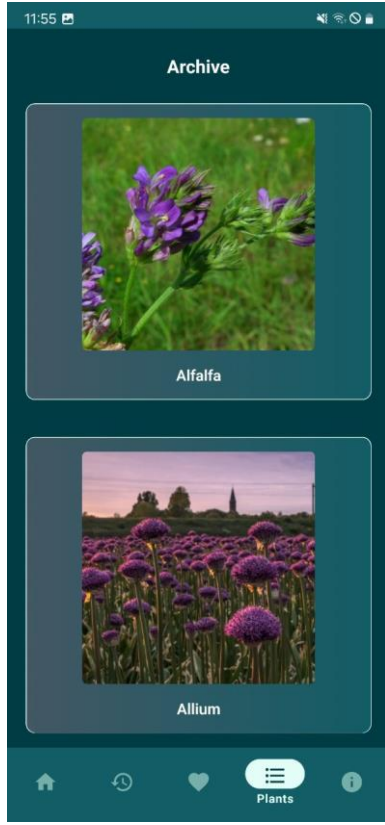
Şekil 4. a) Mobil uygulamanın ana ekranı b) Mobil uygulamanın sonuç ekranı (a) Home screen of the mobile application b) Results screen of the mobile application)



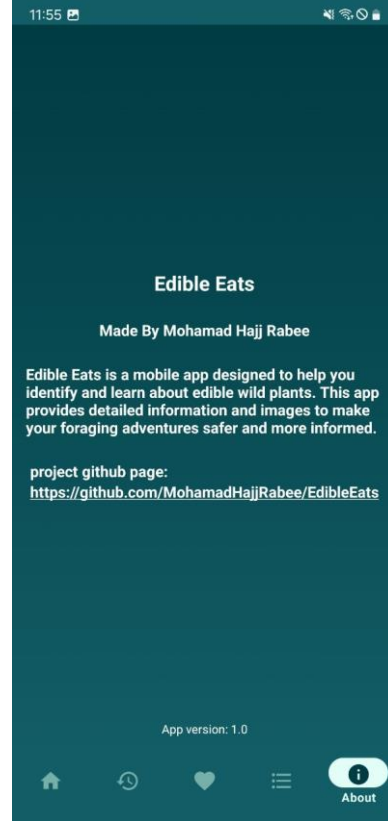
a)

b)

Şekil 5. a) Mobil uygulamanın favoriler ekranı b) Mobil uygulamanın geçmiş ekranı (a) Favorites screen of the mobile application b) History screen of the mobile application)



a)



b)

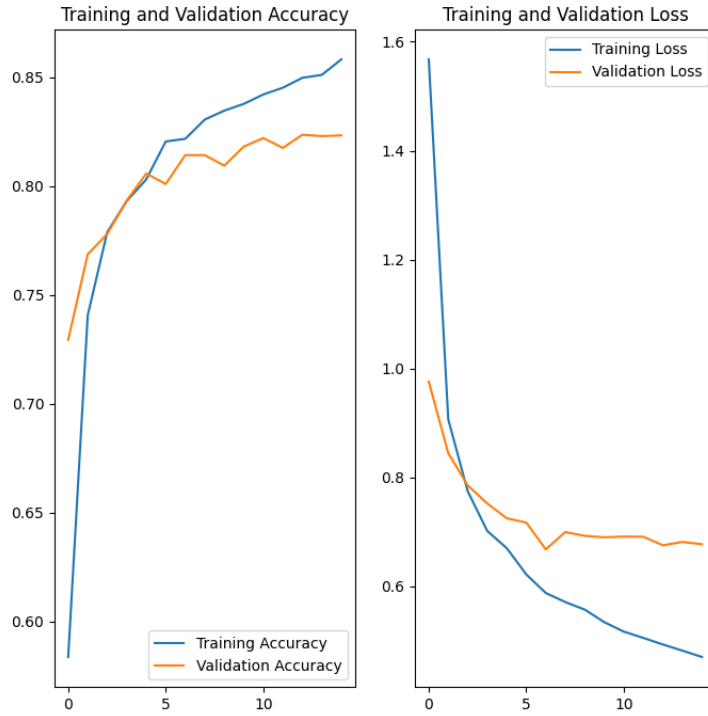
**Şekil 6. a)** Mobil uygulamanın bitkiler ekranı **b)** Mobil uygulamanın hakkında ekranı (a) About screen of the mobile application b) Plants screen of the mobile application)

### 3. Sonuçlar ve Tartışma (Results and Discussion)

#### 3.1. Model eğitim süreci (Model training process)

Modelin eğitim ve doğrulama sürecinde doğruluk ve kaybın değişimini anlamak için oluşturulan grafik, Şekil 7’de sunulmaktadır. Grafikte görüldüğü gibi, eğitim doğruluğu ve doğrulama doğruluğu zamanla artarken, eğitim kaybı ve doğrulama kaybı azalmaktadır. Bu durum, modelin veriye uyum sağladığını ve öğrenme sürecinde ilerleme kaydettiğini göstermektedir. Ancak, doğrulama doğruluğunun belirli bir noktada plato yaptığı ve doğrulama kaybının da belirli bir değerden sonra azalmadığı gözlemlenmektedir. Bu, modelin aşırı öğrenmeye başlamış olabileceğini ve ek düzenleme veya veri arttırma tekniklerine ihtiyaç duyulabileceğini işaret edebilir. Modelin eğitim sonuçlarından %85 doğruluk ve %82 doğrulama doğruluğu olarak elde edilmiştir.

Karışıklık matrisi, çalışmada kullanılan sınıflandırıcı performansının kapsamlı bir sunumudur. Doğruluk (accuracy), duyarlılık (recall) veya başka değerlendirme ölçütlerini hesaplamamanın bir yolu şeklinde açıklanamaz. Sınıflandırıcının iç işleyişine dair daha derin bir anlayış sunan bir açıklamadır. Karışıklık matrisi ve türevlerinin incelenmesi, sınıflar ve veri örneklerine atanan etiketlerin temsil ettiği anlamlar ve kavramlar arasındaki ilişkiler hakkında güçlü ipuçları sağlar. Karışıklık matrisi, sınıflandırıcının zayıf yönlerini ortaya çıkarır ve model performansının iyileştirilmesi için rehberlik eder. Karışıklık matrisinin analizi, farklı veri özellikleri ve nesnelere arasındaki ilişkiler hakkında da bilgiler sunabilmektedir [28].



Şekil 7. Modelin öğrenme eğrisi (Learning curve of the model)

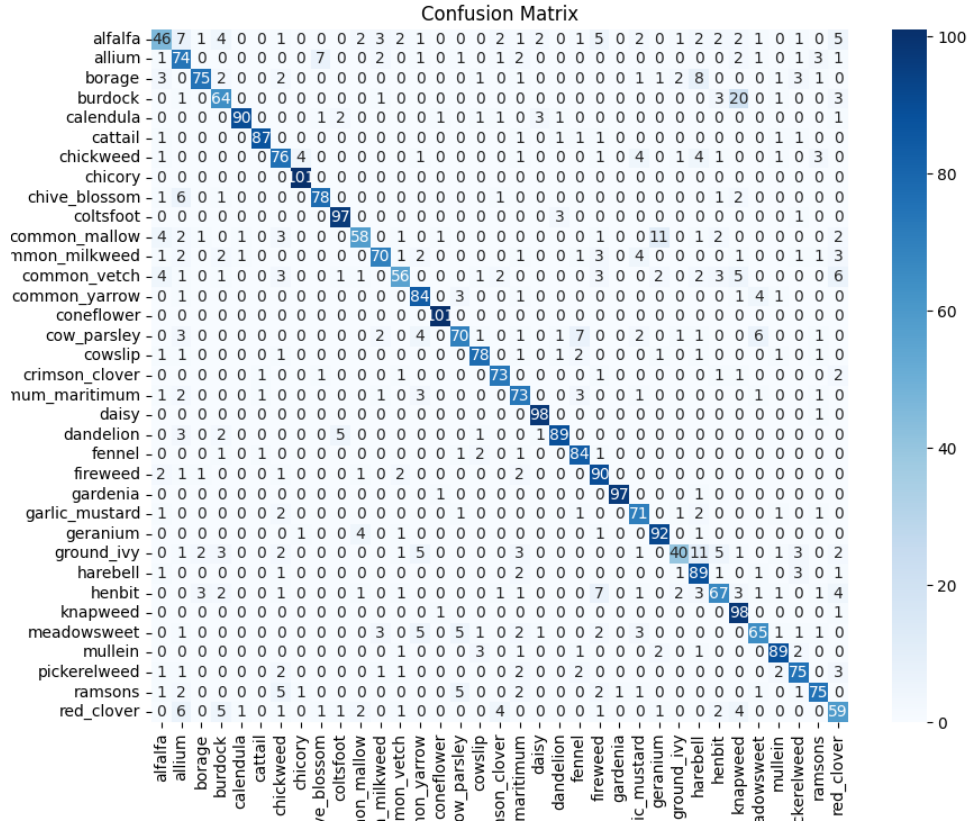
Modelin performansını daha detaylı değerlendirmek için karışıklık matrisi kullanılmıştır. Karışıklık matrisi, her bir sınıf için doğru ve yanlış sınıflandırmaları görselleştirerek, modelin hangi sınıflarda güçlü ve zayıf olduğunu belirlemeye yardımcı olmaktadır. Şekil 8’de modelin karışıklık matrisi gösterilmektedir. Karışıklık matrisi, modelin genel performansını ve hangi bitki türlerinin doğru veya yanlış tanımlandığını göstermektedir. Matriste diagonal üzerindeki değerler doğru sınıflandırmaları, diğer hücreler ise yanlış sınıflandırmaları temsil etmektedir. Bu analiz, modelin hangi bitki türlerini karıştırdığına dair değerli bilgiler vermektedir ve gelecekteki iyileştirme çalışmaları için yönlendirme sağlamaktadır.

### 3.2. Model performans değerlendirmesi (Model performance evaluation)

Makine öğrenimi, veri madenciliği ve yapay zeka uygulamalarının farklı alanlarında (öneri sistemleri ve sosyal ağ analizi gibi) yanlış ve negatif örnekler sürekli artmaktadır. Örnek olarak çoğu web sayfası çoğu sorgu için alakasız sonuçlar sunmaktadır. Bu tür durumlarda sınıflandırma doğruluğu mantıklı bir değerlendirme ölçütü değildir, çünkü her zaman negatif olan sınıflandırıcıyı fazla değerli kılmaktadır. Sınıf dengesizliğini doğruluğun maliyet-duyarlı versiyonlarıyla ayarlamak da işe yaramamaktadır. Çünkü bu sadece doğru negatiflerin faydasını azaltmakla kalmaz aynı zamanda yanlış pozitiflerin maliyetini de düşürür. Bu durumda iyi bir çözüm, doğru negatifleri tamamen göz ardı etmek ve performans ölçütü olarak yanlış pozitif oranı yerine, pozitif tahminler arasındaki doğru pozitiflerin oranı olan kesinliği (precision) kullanmaktır. Bu bağlamda, doğru pozitif oranı genellikle duyarlılık (recall) olarak adlandırılır [29, 30].

Kesinlik, modelin doğru pozitif tahminlerinin, yaptığı toplam pozitif tahminlere oranını ifade eder. Modelin pozitif olarak tahmin ettiği sonuçların ne kadarının gerçekten doğru olduğunu gösterir. Bu modelin kesinlik değeri 0.8267’dir. Kesinlik değerinin formülü aşağıda verilmiştir:

$$\text{Kesinlik} = \frac{TP}{TP+FP} \quad (1)$$



Şekil 8. Modelin karışıklık matrisi (Confusion matrix of the model)

Duyarlılık, modelin doğru pozitif tahminlerinin, tüm gerçek pozitif vakalara oranını ifade eder. Modelin ne kadar doğru pozitif vakayı yakaladığını gösterir. Bu modelin duyarlılık değeri 0.8232'dir. Duyarlılık metriğinin formülü aşağıda verilmiştir:

$$Duyarlılık = \frac{TP}{TP+FN} \quad (2)$$

F1 Skor, kesinlik ve duyarlılık metriklerinin harmonik ortalamasıdır. Bu metrik, hem yanlış pozitif hem de yanlış negatif tahminleri dikkate alarak genel model performansını değerlendirir. Bu modelin F1 Skoru 0.8204'tür. F1 Skor hesaplama formülü aşağıda verilmiştir:

$$F1\ Skoru = 2 \times \frac{Kesinlik \times Duyarlılık}{Kesinlik + Duyarlılık} \quad (3)$$

F1 Skor pozitif ve negatif sınıflarda asimetrik bir yapıdadır. Tamamlayıcı tahminler ve gerçek etiketler göz önüne alındığında, F1 Skor farklı bir sonuç verebilir. Ayrıca, mikro F1'in nadir etiketler üzerindeki performanstan daha az etkilendiği, makro F1 Skorum ise her etiketin F1 Skorunu eşit olarak ağırlıklandırdığı genel olarak bilinmektedir [31].

Matthews Korelasyon Katsayısı (Matthews Correlation Coefficient), modelin sınıflandırma kalitesini ölçen bir metriktir. Hem pozitif hem de negatif sınıflardaki doğru ve yanlış tahminleri dikkate alır ve dengeli sınıflar için uygun bir ölçümdür. Bu modelin MCC değeri 0.8183'tür. MCC değerinin hesaplama formülü aşağıda verilmiştir:

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \quad (4)$$

Doğruluk, modelin doğru tahminlerinin, yaptığı toplam tahminlere oranını ifade etmektedir. Modelin genel olarak ne kadar doğru tahmin yaptığını gösterir. Doğruluk metriğinin formülü aşağıda verilmiştir. Bu modelin doğruluk değeri 0.8232'dir. Hesaplanan metrik sonuçları Tablo 1'de sunulmuştur.

$$\text{Doğruluk} = \frac{TP+TN}{TP+TN+FP+FN} \quad (5)$$

**Tablo 1.** Performans metrik sonuçları (Performance metric results)

	<b>Sonuç</b>
<b>Kesinlik</b>	0.8267
<b>Duyarlılık</b>	0.8232
<b>F1 Skoru</b>	0.8204
<b>MCC</b>	0.8183
<b>Doğruluk</b>	0.8232

#### 4. Sonuç (Conclusion)

Sunulan makale kapsamında geliştirilen sistem, kullanıcı dostu arayüzü ve elde edilen yüksek doğruluk oranıyla YYB'nin tanımlanmasında günümüzde kullanılması kar sağlayacak önemli bir çözüm sunmaktadır. Elde edilen sonuçlarda, MobileNetV2 tabanlı oluşturulan modelin eğitim doğruluğunun %85 ve doğrulama doğruluğunun %82 olduğu görülmektedir. Bu çalışma, mobil cihazlar üzerinden YYB tanımlama alanında önemli bir yol katedilmesini sağlamıştır. Doğa meraklıları ve doğa üzerindeki araştırmacılar için mobilde rahatça kullanılabilir bir araç sunmaktadır. Sunulan mobil uygulama, gerçek zamanlı bitki tanımlama yeteneğini arttırabilmek için, kullanıcıya anlık geri bildirim gönderebilme ve bitkileri daha hızlı tanımlama özellikleriyle daha da iyileştirilebilir. Yapılacak bu iyileştirmeler, kullanıcıların doğada araştırma yaparken bitkileri rahatça ve hızlı bir şekilde tanımlayıp bilgi edinmelerini sağlayarak kullanıcı memnuniyetini arttıracaktır. Ayrıca bu uygulama üzerine, bitki hakkındaki ek bilgiler (benzer türdeki farklı bitki örnekleri, doğada bulunma alanları vb.) ve özellikler eklenerek kullanıcı deneyimlerinin olumlu yönde etkilenmesi sağlanabilir. Sunulan uygulama düzenli olarak güncellenmelidir ve uygulama kullanıcı geri bildirimlerine dayalı olarak geliştirilebilir. Uygulama farklı dillerde desteklenerek uluslararası kullanımı kolaylaştırılabilir. Bu sayede uygulama içerisinde bir topluluk platformu oluşturularak kullanıcıların kendi aralarında bilgi alışverişi yapmaları ve uygulama içerisine kendi çektikleri bitki fotoğraflarını eklemeleri bir başka pozitif özellik oluşturacaktır.

#### Semboller (Symbols)

TF	TensorFlow
TP	Doğru pozitif tahminler
FP	Yanlış pozitif tahminler
FN	Yanlış negatif tahminler
TN	Doğru negatif tahminler
MCC	Matthews Korelasyon Katsayısı

## **Deklarasyon ve Etik Standartlar (Declaration and Ethical Standards)**

Yazarlar bu makalenin araştırılması, yazarlığı ve/veya yayınlanmasıyla ilgili olarak herhangi bir potansiyel çıkar çatışması beyan etmemiştir. Bu makalenin yazarları, bu çalışmada kullanılan materyal ve yöntemlerin etik kurul izni ve/veya yasal-özel izin gerektirmediğini beyan eder.

## **Yazar Katkısı (Author Contribution)**

Mohamad ALHAJ RABIA sunulan fikri tasarladı. Mohamad ALHAJ RABIA ve Mustafa KARHAN teoriyi ve uygulamayı geliştirdi. İrem Nur ECEMİŞ ve Mustafa KARHAN bu çalışmanın gidişatını ve bulgularını denetleyerek düzenledi. Tüm yazarlar sonuçları tartıştı ve makaleyi son haline getirdi.

## **Kaynaklar (References)**

- [1] Shaheen, S., Ahmad, M., Haroon, N., "Edible wild plants: a solution to overcome food insecurity," *Edible Wild Plants: An Alternative Approach to Food Security* 41-57 (2017). Doi: 10.1007/978-3-319-63037-3.
- [2] Teklehaymanot, T., Giday, M., "Ethnobotanical study of wild edible plants of Kara and Kwego semi-pastoralist people in Lower Omo River Valley, Debub Omo Zone, SNNPR, Ethiopia," *Journal of Ethnobiology and Ethnomedicine* 6:1 1-8 (2010). Doi: 10.1186/1746-4269-6-23.
- [3] Jamdhade, V. M., "Wild edible plants used by the tribes of Panvel and Uran Tahsils in Alibaugh District, India: Ethnobotanical application and tribal recipes," *Journal of Botanical Research* 4:1 13-19 (2022). Doi: 10.30564/jbr.v4i1.4280.
- [4] Ojelel, S., Kakudidi, E. K., "Wild edible plant species utilized by a subsistence farming community in Obalanga sub-county, Amuria district, Uganda," *Journal of Ethnobiology and Ethnomedicine* 11:1 1-8 (2015). Doi: 10.1186/s13002-015-0037-7.
- [5] Kohila, A., Mary Kensa, V., "Survey of wild edible plants of Dhanakarkulam Panchayath, Tirunelveli District, Tamil Nadu, India," *Kongunadu Research Journal* 6:2 20-27 (2019). Doi: 10.26524/krj297.
- [6] Kebede, A., Tesfaye, W., Fentie, M., Zewide, H., "An ethnobotanical survey of wild edible plants commercialized in Kefira Market, Dire Dawa City, eastern Ethiopia," *Plant* 5:2 42-46 (2017). Doi: 10.11648/j.plant.20170502.13.
- [7] Panda, S. P., Mazhar, Z., Chakraborty, K., Dasgupta, S., Kamila, P. K., Hameed, S. S., Sharief, M. U., "Diversity of wild edible fruit plants of Acharya Jagadish Chandra Bose Indian Botanic Garden, Howrah, West Bengal, India," *Plant Archives* 24:1 1463-1472 (2024).
- [8] Gautam, R. S., Shrestha, S. J., Shrestha, I., "Wild edible fruits of Nepal," *International Journal of Applied Sciences and Biotechnology* 8:3 289-304 (2020). Doi: 10.3126/ijasbt.v8i3.31561.
- [9] Addis, G., Urga, K., Dikasso, D., "Ethnobotanical study of edible wild plants in some selected districts of Ethiopia," *Human Ecology* 33:1 83-118 (2005). Doi: 10.1007/s10745-005-1656-0.
- [10] Partridge, R. "Wild Edible Plants," Kaggle, 2024. Erişim: 27 Ağustos 2024. [Çevrimiçi]. Erişim adresi: <https://www.kaggle.com/datasets/ryanpartridge01/wild-edible-plants>.

- [11] Partridge, R. "Wild Edible Plant Classifier," Github, 2024. Erişim: 27 Ağustos 2024. [Çevrimiçi]. Erişim adresi: <https://github.com/Achronus/wep-classifier>.
- [12] Yang, S. Xiao, W. Zhang, M. Guo, S. Zhao, J. and Shen, F. "Image Data Augmentation for Deep Learning: A Survey," arXiv, 2023. Erişim: 27 Ağustos 2024. [Çevrimiçi]. Erişim adresi: <https://arxiv.org/abs/2204.08610>.
- [13] Tatar, A., Haghighi, M., Zeinijahromi, A., "Experiments on image data augmentation techniques for geological rock type classification with convolutional neural networks," Journal of Rock Mechanics and Geotechnical Engineering (2024). Doi: 10.1016/j.jrmge.2024.02.015.
- [14] Rahmatullah, P. Abidin, T. F. Misbullah, A. Nazaruddin, A. "Effectiveness of Data Augmentation in Multi-class Face Recognition," 2021 5th International Conference on Informatics and Computational Sciences (ICICoS), 64-68 (2021). Doi: 10.1109/ICICoS52554.2021.9651780.
- [15] Yang L., Hanneke S., Carbonell J., "A theory of transfer learning with applications to active learning," Machine Learning 90 161-189 (2013). Doi: 10.1007/s10994-012-5310-7..
- [16] Pan S. J., "Transfer Learning," Data Classification: Algorithms and Applications 21 537-570 (2014).
- [17] Yong L., Ma L., Sun D., Du L., "Application of MobileNetV2 to waste classification," PLOS ONE 18 1-16 (2023). Doi: 10.1371/journal.pone.0282336..
- [18] Abadi M., Agarwal A., Barham P., Brevdo E., Chen Z., Citro C., Corrado G. S., Davis A., Dean J., Devin M., "TensorFlow: Large-scale machine learning on heterogeneous distributed systems," arXiv (2016). arXiv:1603.04467..
- [19] Rashidi, M., "Application of TensorFlow Lite on embedded devices: A hands-on practice of TensorFlow model conversion to TensorFlow Lite model and its deployment on Smartphone to compare model's performance," Dissertation 1-43 (2022).
- [20] Verma G., Gupta Y., Malik A. M., Chapman B., "Performance evaluation of deep learning compilers for edge inference," 2021 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW) 858-865 (2021). Doi: 10.1109/IPDPSW52791.2021.00137..
- [21] Vyshnavi V. R., Malik A., "Efficient way of web development using Python and Flask," International Journal of Recent Research Aspects 6:2 16-19 (2019).
- [22] Alemu M. B., "REST API: Implementation with Flask-Python," Lisans, Lapland UAS, (2014).
- [23] React Native, "React Native Documentation," [Çevrimiçi]. Available: <https://reactnative.dev/>. [Erişim: 27 Ağustos 2024].
- [24] Fentaw A. E., "Cross platform mobile application development: a comparison study of React Native Vs Flutter," Yüksek Lisans, University of Jyväskylä Faculty of Information Technology, (2020).
- [25] Danielsson W., "React Native application development," Linköpings universitet, Swedia 10:4 1-10 (2016).

[26] Hutri H., "Comparison of React Native and Expo," Yüksek Lisans, Lappeenranta–Lahti University Software Engineering and Digital Transformation, (2023).

[27] Järveläinen, H., "Creating a React Native UI component Library," Lisans, Turku UAS Information and Communications Technology (2024).




[28] Krstinić D., Braović M., Šerić L., Božić-Štulić D., "Multi-label classifier performance evaluation with confusion matrix," Computer Science & Information Technology 10 1-14 (2020). Doi: 10.5121/csit.2020.100801.

[29] Flach P., Kull M., "Precision-recall-gain curves: PR analysis done right," Advances in Neural Information Processing Systems 28 1-9 (2015).

[30] Xie Y., Zhu C., Zhou W., Li Z., Liu X., Tu M., "Evaluation of machine learning methods for formation lithology identification: A comparison of tuning processes and model performances," Journal of Petroleum Science and Engineering 160 182-193 (2018). Doi:10.1016/j.petrol.2017.10.028

[31] Lipton, Z. C., Elkan, C., Naryanaswamy, B., "Thresholding classifiers to maximize F1 score," arXiv (2014). arXiv:1402.1892.

### Yazar Biyografileri (Author Biographies)

 Mohamad Alhaj Rabia	Mohamad Alhaj Rabia, Çankırı Karatekin Üniversitesi Bilgisayar Mühendisliği bölümünde okumaktadır. Makine öğrenmesi yöntemlerinin kullanılmasıyla mobil uygulama geliştirme alanında çalışmalar yapmaktadır.
 İrem Nur Ecemiş	İrem Nur Ecemiş, Çankırı Karatekin Üniversitesi Bilgisayar Mühendisliği bölümünde Araştırma Görevlisi olarak çalışmaktadır. Ankara Üniversitesi Bilgisayar Mühendisliği Bölümünde doktora eğitimine devam etmektedir.
 Mustafa Karhan	Mustafa Karhan, Çankırı Karatekin Üniversitesi Bilgisayar Mühendisliği bölümünde Doçent unvanıyla görev yapmaktadır. Yapay zeka ve görüntü işleme alanında yayınları bulunmaktadır.