

T.C.
SAKARYA ÜNİVERSİTESİ
BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ

BSM 498 BİTİRME ÇALIŞMASI

MOBİL DESTEKLİ DERİN ÖĞRENME İLE
BİTKİ TANIMA VE ÖZELLİK ÇIKARIMI

B151210074 – Yusuf GÖREN
B161210059 – Habibullah METİN
B161210113 – Muhammed Emin SANDIKÇI

Fakülte Anabilim Dalı : BİLGİSAYAR MÜHENDİSLİĞİ
Tez Danışmanı : Dr. Öğr. Üyesi Muhammed Fatih
ADAK

2019-2020 Bahar Dönemi

T.C.
SAKARYA ÜNİVERSİTESİ
BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ

**MOBİL DESTEKLİ DERİN ÖĞRENME İLE
BİTKİ TANIMA VE ÖZELLİK ÇIKARIMI**

BSM 498 - BİTİRME ÇALIŞMASI

Adı SOYADI

Fakülte Anabilim Dalı : BİLGİSAYAR MÜHENDİSLİĞİ

Bu tez .. / .. / ... tarihinde aşağıdaki jüri tarafından oybirliği / oyçokluğu ile kabul edilmiştir.

.....
Jüri Başkanı

.....
Üye

.....
Üye

ÖNSÖZ

Günümüzdeki bilgisayar bilimlerindeki ilerleme, insan gibi düşünen ve davranan sistemlerin geliştirilmesi, 1950'li yıllardan beri sürmektedir. Yapay zeka olarak isimlendirilen bu alan, insan düşünme ve davranışlarını taklide yönelik olduğundan, nöroloji, psikoloji ve mühendislik gibi farklı disiplinleri kapsayan geniş bir alana yayılmıştır.

İnsan gibi düşünebilen ve davranabilen sistemlerin geliştirilmesi için yapılan çalışmalarda bugün için gelinen nokta, henüz yapay zekanın tam olarak geliştirilememiş olmasıdır. Yapay zekanın yapılabirliği üzerinde yapılan felsefi tartışmalar bir yana, düşüncenin salt fiziksel süreçlere indirgenebildiği kabul edilse bile, henüz beynin tüm fonksiyonları tam olarak çözülemediğinden, bugün için yapılabilmesi henüz mümkün gözükmemektedir. Fakat konu üzerinde yapılan çalışmalar farklı alanlarda hızla devam etmektedir. Burada şunu da belirtmek gerekir ki, yapay zekanın yapılamayacağını savunanlar, konu üzerinde karşıt görüşlü araştırmacılar ile aynı araştırma ve geliştirme çalışmalarını yürütmektedirler. Çünkü her iki tür araştırmacının yapmaya çalıştıkları şey gözlemlenebilen nesnel olayların benzerini yapabilmektedir. Bu olayların, yani düşüncenin beyinde gözlenebilen fiziksel süreçlere indirgenerek bir algoritmasının oluşturulup oluşturulamayacağı felsefi bir yorumdur.

Bu farklı alanlarda yapılan çalışmaların ortaya çıkardığı teknolojik ürünler, işletmelerde de sıklıkla kullanılmaktadır. Çünkü ortaya çıkan ürünler, insan özelliklerinin sınırlı da olsa belli bir kısmına sahip olabildiğinden, belirli işlerde insanların yerine onlardan daha verimli olarak kullanılmaktadır. Bu ise yönetim açısından bakıldığında, işletme de verimlilik artışı ve hata oranları ve birtakım diğer masraflarda azalmalara sebep olduğundan oldukça önemlidir. Yönetim açısından aynı oranda dikkat gösterilmesi gereken nokta, yeni teknolojilerin kullanılabilmesine yönelik, çalışanların eğitilmeleri ve çalışanların gelişen teknolojiye uyumunun sağlanması konusudur.

İÇİNDEKİLER

ÖNSÖZ.....	iii
İÇİNDEKİLER.....	iv
ŞEKİLLER LİSTESİ.....	vii
ÖZET.....	viii
 BÖLÜM 1.	
GİRİŞ.....	9
 BÖLÜM 2.	
YAPAY ZEKA.....	11
2.1. Evrişimli Sinirsel Ağlar(CNN).....	12
 BÖLÜM 3.	
VERİ VE VERİ SETİ NEDİR.....	13
3.1. Veri Setinin Önemi.....	14
3.2. Projede Kullanılan Veri Seti.....	15
 BÖLÜM 4.	
KULLANILAN YAPAY ZEKA YÖNTEMLERİ VE TEKNİKLERİ.....	17
4.1. Kullanılan Kütüphaneler ve Uygulama Geliştirme Ortamı.....	17
4.2. Veri Setinin Colab Ortamına Yüklenmesi.....	17
4.3. Kütüphanelerin Projeye Dahil Edilmesi.....	18
4.4. Seçilen Veri Setindeki Klasörlerin Oluşturulması.....	18
4.5. Veri Setinin Rastgele Ayrıştırılması.....	18
4.6. Veri Setine Augmentation Yönteminin Uygulanması.....	18
4.7. Yapay Sinir Ağının Oluşturulması.....	19
4.7.1. Tanımlanmaların yapılması.....	19
4.7.2. MobileNetV2 ağının tanıtılması.....	22

4.8. Modelin Derlenmesi ve Başarım ve Kayıp Oranlarının Elde Edilmesi.....	23
4.9. Başarım ve Kayıp Grafiklerinin Çizdirilmesi.....	24
4.10. Modelin Kaydedilmesi.....	24
 BÖLÜM 5.	
ANDROİD VE WEB SUNUCU YAZILIMLARININ GELİŞTİRİLMESİ.....	25
5.1. MainActivity Sayfasının Yapısı.....	26
5.1.1. MainActivity arayüz tasarımı.....	26
5.1.2. Kamera fonksiyonlarının işleyişi.....	27
5.1.3. OpenUtils fonksiyonunun işleyişi.....	27
5.1.4. MainActivity'deki diğer fonksiyonların işleyişi.....	28
5.1.5. Resmin kaydedilmesi ve kırılması.....	28
5.2. CameraPreview Sayfasının Yapısı.....	29
5.2.1. CameraPreview fonksiyonlarının işleyişi.....	29
5.3. Classify Sayfasının Yapısı.....	30
5.3.1. Classify arayüz tasarımı.....	30
5.3.2. Classfy onCreate fonksiyonunun işleyişi.....	31
5.3.3. ConvertBitmapToByteBuffer fonksiyonunun işleyişi.....	32
5.3.4. PrintTopKLabels fonksiyonunun işleyişi.....	32
5.3.5. Bitki özelliklerinin çıkarılması.....	33
5.4. Web Sunucu Yazılımının Geliştirilmesi.....	33
5.4.1. Mobil uygulama kısmının işleyişi.....	33
5.4.2. Web sunucu kısmının işleyişi.....	34
 BÖLÜM 6.	
UYGULAMA.....	35
 BÖLÜM 7.	
SONUÇLAR VE ÖNERİLER.....	36
 KAYNAKLAR.....	 37

ÖZGEÇMİŞ.....	38
---------------	----

BSM 498 BİTİRME ÇALIŞMASI DEĞERLENDİRME VE SÖZLÜ SINAV TUTANAĞI.....	39
---	----

ŞEKİLLER LİSTESİ

Şekil 2.1.	Geleneksel Evrişimli Sinir Ağı mimarisi.....	12
Şekil 2.2.	İki Boyutlu Evrişim.....	12
Şekil 3.1.	Aglaonema.....	15
Şekil 3.2.	Antoryum.....	16
Şekil 3.3.	Deve Tabanı.....	16
Şekil 3.4.	Kasımpatı.....	16
Şekil 4.1.	Seçim Ekranı.....	17
Şekil 4.2.	Conv2D Gösterimi.....	20
Şekil 4.3.	Depthwise Conv2D.....	20
Şekil 4.4.	Global Average Pooling.....	21
Şekil 4.5.	Batch Normalization Formülü.....	21
Şekil 4.6.	RELU Fonksiyon Grafiği.....	22
Şekil 4.7.	Softmax Formülizasyon.....	22
Şekil 4.8.	Bottleneck Residual Block.....	23
Şekil 4.9.	Eğitim Sonucu.....	23
Şekil 4.10.	Eğitim ve Test Başarı Grafiği.....	24
Şekil 4.11.	Eğitim ve Test Kayıp Grafiği.....	24
Şekil 5.1.	Örnek Uygulama Çıktısı.....	25
Şekil 5.2.	MainActivity Sayfasının Görüntüsü.....	26
Şekil 5.3.	MainActivity XML Görüntüsü.....	27
Şekil 5.4.	Kırpıcı Sayfasının Görünümü.....	29
Şekil 5.5.	Classify Sayfasını Görüntüsü.....	31
Şekil 5.6.	Modelin Yüklendiği Kısım.....	31
Şekil 5.7.	RGB Çevirici Kod.....	32
Şekil 5.8.	Mobilde Sunucu Örnek Çıktısı.....	34

ÖZET

Anahtar kelimeler: Derin Öğrenme, Mobil Uygulama, Bitki Tanıma

Bu projenin amacı kullanıcıların bitkileri daha hızlı bir şekilde tanımlarını sağlamak amacıyla mobil ortam destekği sağlanarak geliştirilmiştir. Mobil desteği sayesinde uygulamanın taşınabilmesi kolaylaştırılmıştır.

Kullanıcıların herhangi bir çiçek resminin sınıflandırılması konusunda yardımcı olan bu sistem, derin öğrenme yardımıyla yapılmıştır. Android cihazlarda da çalışabilen bu sistem ile kullanıcılar istedikleri çiçeğin fotoğrafını çekerek onların türlerini ve o bitkiye ait özellikleri öğrenebilecekler.

Bu proje sayesinde insanlar bitkileri daha hızlı bir şekilde tanıyabilmesi ve bitkiyle alakalı bilgilere daha hızlı bir şekilde ulaşabilmesini sağlamaktadır.

BÖLÜM 1. GİRİŞ

Günümüzde makine öğrenmesi ve derin öğrenmedeki teknolojilerin gelişmesi botanik alanında da yapay zekanın kullanım alanlarını arttırmaktadır. Ev bitkisi alırken bitkilerle alakalı bilgilere genelde internet üzerinde yada bilgi sahibi kimseler sayesinde ulaşılmaktadır bu da ev bitkisi seçerken zaman kaybına neden olmaktadır, bizlerde bu probleme çözüm oluşturabilecek bir proje üzerinde çalışmaya karar verdik. Bu proje herhangi bir ev bitkisinin anlık olarak türünün tespit edilmesi ve tespit edilen bitki ile alakalı bilgilere hızlı ve etkili bir şekilde ulaşılabilmesi amacıyla geliştirilmiştir, hem zaman hemde mekan fark etmeksizin sadece telefona yüklenen bir uygulama sayesinde bitkinin tespiti sağlanmaktadır.

Bu projede derin öğrenme metodları kullanılarak camera api ile android telefon üzerinden bitkinin resmini çekip resmi ön işlemlere tabi tutarak modele yollanır ve tahmin edilen bitkinin ismi ve doğruluk oranı ile birlikte o bitkiye ait ortam ısısı, sulama, saksı değişimi süresi, güneş ışığı ve destek besin bilgilerinide ekrana yazdıran bir program yapılmıştır. Diğer bir yandan python üzerinde geliştirilen HTTP işlemleri için geliştirilmiş olan Flask framework ile python üzerinden geliştirilne program ile android üzerinden çekilen resim HTTP protokolü üzerinden sunucuya gönderilir, resim sunucu üzerindeki python programı üzerinden oluşturduğumuz modele girdi olarak verilir, modelin çıktısı response mesajı ile android uygulamasına gönderilerek uygulama ekranında sonuç gösterilir.

Pekin Ormancılık Üniversitesi yaptıkları çalışmalarında 100 farklı bitki için toplamda 10000 görüntüden oluşan 26 katmanlı derin öğrenme modeli geliştirilmiş ve bu modelin (BJFU100) eğitiminde kullanılan görüntüler 28 mm eşdeğerli odak uzaklığına sahip bir ana lens ve 3120×4208 çözünürlüğe sahip telefonlarla çekilmiştir. Model de % 91,78'lik bir tanıma oranı elde edilmiştir. Min Gao, Lang Lin ve Richard O. Sinnott geliştirdikleri iOS tabanlı mobil uygulama ile %95'e varan doğruluk oranı ile 122/125 çeşit bitki ve 47/50 bitki cinsini tanımlayabilmektedir.

Geliştirilen bu proje mobil ortamda olmasından dolayı hem kullanışlı hem de hızlı olmasından dolayı camera uygulamasını java tabanlı Camera adlı kütüphanesi kullanılarak geliştirilmiştir. Geliştirilen model ise mobil uygulama üzerinden saklanacağından mobil uygulamalara göre tasarlanmış olan CNN yapısıyla geliştirilmiş eğitim sonucunda oluşan ağırlık dosyasının boyutunun küçük olması nedeniyle MobileNet ağı projede derin öğrenme modeli olarak kullanılmaktadır.

BÖLÜM 2. YAPAY ZEKA

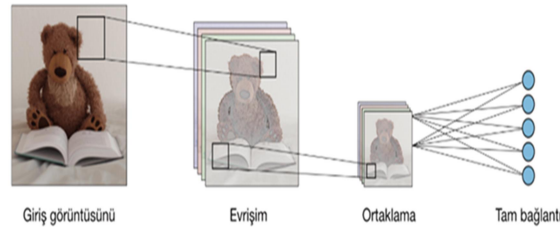
Yapay Zeka insanın makinelerle etkileşimini sağlayan, insanlığa faydası olan, bazı bilim insanlarına göre de insanoğlunu yok edeceğini düşünülen bütüncül bir teknolojidir. Başka bir tanımı ise bir bilgisayarın veya bilgisayar kontrolündeki bir robotun çeşitli faaliyetleri zeki canlılara benzer şekilde yerine getirme kabiliyetidir [1]. Yapay Zeka matematiksel yöntemler, istatistiksel yöntemler, nörolojik yöntemler ve deneysel yöntemler gibi birçok farklı yöntem kullandığından dolayı çok sayıda alt alana ayrılmıştır. Bu alanlar şunlardır:

- **Makine Öğrenmesi** : Bu alan genel anlamda istatistiksel ve matematiksel yöntemler kullanır. Örneğin Doğrusal Regresyon, Polinomal Regresyon, Destek Vektör Makineleri, Karar Ağaçları, Naive Bayes, Lojistik Regresyon, K-NN, Birlikte Kural Çıkarımı gibi tahmin, sınıflandırma ve kümeleme algoritmalarını kullanır.
- **Derin Öğrenme** : Bu alan insan beyin yapısını temel alır ve Beynin öğrenme modelini kendinde uygular. Destekli, Desteksiz ve Takviyeli Öğrenme olmak üzere 3 farklı alt alana bölünmüştür. Yapay Sinir Ağlarının (ANN) yanı sıra Evrişimli Sinir Ağları (CNN) ve Yinelemeli Sinir Ağları (RNN) gibi birçok sinirsel ağ modeli bulunmaktadır.
- **Örüntü Tanıma** : Belli bir örüntüye ve yapıya sahip yapıların, şekillerin tahmini ve tanınması konusunu ele alır. Desteksiz Öğrenme başlığı altındaki sinirsel ağ modellerini kullanır.
- **Nesne Tanıma** : Belli bir nesne tanınmasını ele alır. Hırsız tespiti bu konunun temel başlığıdır. Makine Öğrenmesinde Destek Vektör Makineleri, Derin Öğrenmede ise Evrişimli Sinir Ağları algoritmaları genelde bu alanda kullanılır.
- **Görüntü İşleme** : Bir resmin ya da fotoğrafın içerdiği kırmızı, mavi ve yeşil renklere göre pixel matrisinin bulunmasını esas alır.
- **Doğal Dil İşleme** : Herhangi bir dile ait kelimelerin bulunması, noktalama işaretlerinin ayrıştırılması, kelime kök ve gövdelerinin bulunması gibi ve

daha birçok işlem yapılabilir.(örneğin : chatbot) Doğal dil işleme kütüphaneleri kullanılır ardından makine öğrenmesi algoritmaları kullanarak örneğin bir restorandaki müşterilerin geri dönüşlerindeki cümlelere göre makineye müşterinin yemeği beğenip beğenmediği öğretilir.

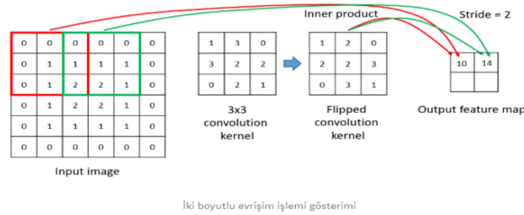
2.1. Evrişimli Sinirsel Ağlar (CNN)

Evrişimli Sinirsel Ağlar genellikle bir görüntünün filtrelenmesi ve tanınması için kullanılan önemli yöntemlerden birisidir. Bu ağlar günümüzde görüntü tanıma uygulamalarında sıklıkla kullanılmaktadır. Geleneksel Evrişimli Sinir Ağı mimarisi aşağıda verilen görselde gösterilmiştir. (Şekil 1.1)



Şekil 2.1. Geleneksel Evrişimli Sinir Ağı mimarisi

Günümüzde bu ağlar 2 boyutlu ya da 3 boyutlu evrişim yöntemlerini kullanırlar. Örneğin 2 boyutlu evrişimde 2 boyutlu bilgiye uygulanacak filtrenin x ve y eksenine göre simetrisi alınır. Tüm değerler matriste eleman eleman çarpılır ve çıkış matrisinin ilgili elemanı olarak kaydedilir [2]. Aşağıdaki şekilde 2 boyutlu evrişim açıklanmıştır. (Şekil 1.2)



Şekil 2.2. İki Boyutlu Evrişim

BÖLÜM 3. VERİ VE VERİ SETİ NEDİR

Veri, bilginin yapılandırılıp kayıt altına alınıp, kolay analiz edilebilmesi için bir araya getirilmesine denir. Bir veya birden fazla bilgiden oluşan kümedir. Veri genellikle araştırma, gözlem, deney, sayım, ölçüm yoluyla elde edilir. Başlıca veri türleri:

- Nitel veri (Qualitative data) Birimlere dayalı ölçümlerle değil sayımlarla değerleri belirtilebilen, belli bir yapısal veya kurumsal özelliği taşımakla (veya taşımamakla) simgelenen veriler. Cinsiyet, saç rengi vb. nitel özellikleri belirten veriler. Bir denneğin niteliklerini belirten verilerdir. Eğitim durumu, ev sahibi olup olmama... Nitel veriler iki gruba ayrılır.
 - Sınıflanabilen Nitel Veri: Birbirinden bağımsız isim bildiren, kod ve numara ile gösterilebilen, sınıflara ayrılan verilerdir. Taşıtlar: kara, hava ve deniz taşıtları:
 - Sıralanabilen Nitel Veri: Belirli bir miktar belirtmeyen, bir sıra ya da dereceye göre elde edilen verilerdir. Öğrencilerin başarı durumu: pekiyi, iyi, orta... Ordu rütbeleri: albay, yarbay, subay, astsubay ...
- Nicel veri (Quantitative data) birimlerin ölçüm ve tartım sonucu değerleri saptanan sayısal özelliklerini belirten, aralıklı ölçekli veya orantılı ölçekli değişkenler. Bir deneyin sayılabilir, ölçülebilir özelliğini veren verilerdir. Sürekli nicel veri ve kesikli nicel veri olmak üzere iki türü vardır. Örneğin golf toplarının sayısı, ölçüsü, fiyatı, bir testteki skor vs.
 - Sürekli nicel veri: Ondalıklı değerler alabilen nicel verilerdir. Boy uzunluğu, kilo, vs.
 - Kesikli nicel veri: Sayılarak elde edilen ve tam sayılı değerlerdir. Nüfus, öğrenci sayısı, hane halkı sayısı.

- Kategorik veri (Categorical data): tanımladığınız veriyi bir kategoriye koyar [3].

Veri seti (veya veri kümesi) bir veri koleksiyonudur. Tablo halinde veriler söz konusu olduğunda, bir veri kümesi bir veya daha fazla veri tabanı tablosuna karşılık gelir; burada bir tablonun her sütunu belirli bir değişkeni temsil eder ve her satır söz konusu veri kümesinin belirli bir kaydına karşılık gelir. Veri kümesi, veri kümesinin her üyesi için bir nesnenin boyu ve ağırlığı gibi her bir değişken için değerleri listeler. Her değer veri olarak bilinir. Veri setleri ayrıca bir belge veya dosya koleksiyonundan oluşabilir [4].

3.1. Veri Setinin Önemi

Veri seti bir makine öğrenmesi projesinde önemli yer kaplar. İşin zor kısmı ise uygun veri setini bulmaktır. Çünkü modelin başarısında modelin kendisi kadar öğretilen veri seti de önemlidir. Eğer veri setinde yeterli çeşitlilikte ve sayı da veri yoksa bu uygulamanın gerçek verilerle olan başarısını bir hayli düşürür. Ancak veri setinde birbirine çok benzeyen verilerde olmamalı, çünkü bu durum modeli ezberlemeye iter. Eğer model ezberleme yaparsa test verisinde başarısı ne kadar yüksek olursa olsun gerçek verilerle karşılaştığı zaman yanlış bir sonuç döndürme olasılığı oldukça yüksektir.

Bunun dışında eğer elde yeterince veri yoksa bunların çoğaltılması gerekir. Eğer veriniz resimlerden oluşuyorsa verinizi arttırmak için bazı kod bloklarını kullanabilirsiniz. Bu kodları kullanarak resimleri döndürebilir, küçültebilir, yakınlaştırebilirsiniz. Bu şekilde elinizde daha fazla veri olur. Diğer önemli problem ise veri setinin eğitim ve test verileri olarak ikiye ayrılmasıdır. Önemli kısım her zaman eğitim verisinin test verisinden fazla olmasıdır. Örneğin %70 eğitim %30 test şeklinde ayrılabilir. Ayrıca eğitim verisindeki veriler test verisinde kesinlikle olmamalıdır. Model test verisinde daha önce görmediği verilerle karşılaşmalıdır.

Veri setiniz ne kadar büyük olursa başarı olasılığınızda o kadar artar. Bunun eksi yanı ise modelin eğitim süresi ve verileri saklama alanı artar. Eğer veriniz çok büyükse modeli

eğitmek tahmin edilenden daha fazla vakit alır. Bunu engellemek için daha iyi donanımına sahip olmak gerekir.

3.2. Projede Kullanılan Veri Seti

Bizim veri setimize gelirse, şuan ki durumdan dolayı veriler internetten bir eklenti yardımıyla çekilmiştir. Veri seti hakkında daha detaylı bilgi vermek gerekirse, veri seti daha çok gündelik yaşantımızda karşımıza çıkan ve evlerde yetiştirilen bitkilerden oluşmaktadır. Her bitki için 100 fotoğraf bulunmaktadır. Veri setinde toplam 43 bitki bulunmaktadır. Bu bitkiler Afrika menekşesi, Aloe vera, Antoryum, Deve tabanı, Kaktüs, Kentia palmyesi, Kuşkonmaz, Para ağacı, Paşa kılıcı, Afelandra, Aglaonema, Areka palmyesi, Atatürk çiçeği, Difenbahya çiçeği, Barış çiçeği, Yılbaşı çiçeği, Gloksinya, Guzmania, Yuka bitkisi, Dua Çiçeği, Aşk Merdiveni, Acem borusu, Arap Yasemini, Begonvil Sarmaşığı, Kumkuat ağacı, Salon Çamı, Katır tırnağı, Sabin ardıcı, Telgraf çiçeği, Orman Gülü, Sardunya Çiçeği, Orkide, Schefflera, Kasımpatı, Sıkla men çiçeği, begonya çiçeği, petunya çiçeği, Fulya çiçeği, kraton bitkisi, Zambak, Zakkum, Sümbül, anemon çiçeği. Bu çiçekler ev bitkileri türünde en çok satılan bitkilerdir. Aşağıda veri setinden birkaç örnek fotoğrafa yer verilmiştir.



Şekil 3.1. Aglaonema

Yukarıdaki resim bir aglaonema örneğidir. Veri setinin içinde ezberlemeyi önlemek için karışık denilebilecek resimler de vardır. Karışıktan kasıt çiçeğin resimde belirgin olmadığı durumlardır.



Şekil 3.2. Antoryum

Yukarıdaki resim eğitim veri seti içerisindeki antoryum bitkisine ait bir örnek fotoğrafı temsil etmektedir.



Şekil 3.3. Deve Tabanı

Yukarıdaki resim eğitim veri seti içerisindeki deve tabanı bitkisine ait bir örnek fotoğrafı temsil etmektedir.



Şekil 3.4. Kasımpatı

Yukarıdaki resim kasımpatıya aittir. Kasımpatıların çok farklı renkleri olduğu için veri setinde diğer renk kasımpatılara da yer vermeye çalışılmıştır.

BÖLÜM 4. KULLANILAN YAPAY ZEKA YÖNTEMLERİ VE TEKNİKLERİ

Projede Transfer Learning kullanarak ImageNet kütüphanesi ile eğitilmiş ve mobil uygulamalara yönelik olarak geliştirilen MobileNetV2 ağı kullanılmıştır. Bu tarz ağlar birçok fotoğraf ile eğitildiği için Çoklu Sınıflandırma problemlerinde iyi çalışırlar. Bundan dolayı MobileNetV2 ağının kullanılması uygun görülmüştür. 43 farklı bitki çeşidi için ağ eğitildiğinde %87 eğitim başarımları oranı ve %74 geçerleme başarımları oranı yakalamıştır.

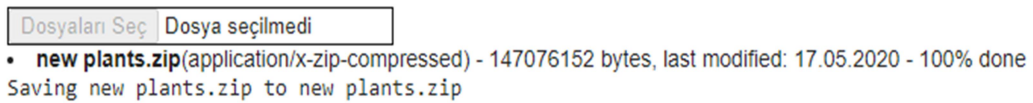
4.1. Kullanılan Kütüphaneler ve Uygulama Geliştirme Ortamı

Projenin genel alanı Görüntü Tanıma olduğundan Derin Öğrenme kütüphaneleri kullanılmıştır. Dünyada en iyi çalışan ve açık kaynaklı olan, Google ve bağımsız geliştiriciler tarafından geliştirilen Tensorflow ve onun alt kütüphanesi diyebileceğimiz Keras kullanılmıştır. Projenin mobil platformda yayınlanması için de Android seçilmiştir.

Oluşturduğumuz yapay sinirsel ağın daha hızlı efektif eğitilebilmesi için yine Google'ın desteklediği Colab çevrimiçi çalışma aracı kullanılmıştır. Programlama dili olarak Python seçilmiştir. Android platformunda uygulamayı çıkarmak için Android Studio ve Java programlama dili kullanılmıştır.

4.2. Veri Setinin Colab Ortamına Yüklenmesi

Veri setinin Colab ortamında kullanabilmek için upload edilmesi gerekir. Bu sayede veriseti kullanılabilir.



Şekil 4.1. Seçim Ekranı

4.3. Kütüphanelerin Projeye Dahil Edilmesi

Python kısmında kullanılan başlıca kütüphaneler şunlardır : os, zipfile, shutil, random, numpy, matplotlib, Tensordlow, Tensorflow Lite, ve Keras kütüphanesindeki optimizers, image, Model, preprocessing, Dense, GlobalAveragePooling2D ve Transfer Learning olarak kullanılan MobileNet kullanılmıştır.

4.4. Seçilen Veri Setindeki Klasörlerin Oluşturulması

Bu kısımda ilgili zip dosyasındaki veriseti os kütüphanesi vasıtasıyla açılarak ardından tüm klasör ve veriler dışarı aktarılmıştır. Ardından tüm sınıf isimlerini içeren bir liste tanımlayıp bitki isimleri listeye sırayla atanmıştır. Son olarak da Python'daki try bloğu ile eğitim ve test klasörleri ve bunların altında foreach döngüsü ile sınıflara ait klasörler oluşturulmuştur.

4.5. Veri Setinin Rastgele Ayrıştırılması

Bu kısımda yazılan split_data fonksiyonu ile her bir sınıftaki veriler %80 eğitim %20 test olarak ayrılır. Ayrılan bu verileri önceden oluşturulan ilgili klasörlere foreach döngüsü ve shutil kütüphanesinin copyfile fonksiyonu kullanılarak yerleştirilmiştir ve bu sayede veri seti her sınıf adına eğitim ve test olarak ayrılmıştır.

4.6. Veri Setine Augmentation Yönteminin Uygulanması

Augmentation yöntemi basit tanımıyla veri setini çoğaltma olarak düşünülebilir. Yani veri seti beklenilenden az veri içeriyorsa ya da daha iyi bir sonuç alabilmek için veri setine Augmentation yöntemi uygulanabilir. Augmentation yöntemi veri seti rescale yapıldıktan sonra yani görüntünün piksel değerlerinin 0 ila 1 arasında normalizasyon işlemine tabi tutulduktan sonra yapılabilir. Başlıca Augmentation yöntemleri şunlardır:

- Ölçeklendirme (Scaling) : Bu teknik ile herhangi bir görüntünün büyütme ve küçültme işlemleri yapılabilir.
- Kaydırma (Translation) : Bu teknik sayesinde görüntüyü yukarı, aşağı, sağa ve sola kaydırma ve kırpma işlemleri yapılabilir.
- Döndürme (Rotation) : Bu teknik, görüntüyü istenilen dereceler arasında rastgele döndürerek aynı resmin birçok farklı versiyonunu oluşturur.
- Yansıtma (Flipping) : Bu teknik, görüntünün istenilen ekseninde(örneğin x, y, $x=y$, $x=-y$, vs.) yansıtılmasını sağlar.
- Yakınlaştırma (Zooming) : Bu teknik, görüntünün yaklaşmasını veya uzaklaşmasını sağlar.
- Doldurma (Filling) : Bu teknik, görüntüde kaybolan piksellerin girilen parametreye göre(Örneğin 'nearest' parametresinde kendisine en yakın piksellere bakarak doldurur) doldurulmasını sağlar.

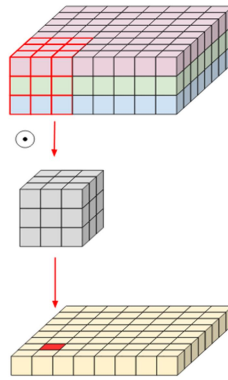
Bu işlemler yapıldıktan sonra eğitim ve test verileri üzerinde görüntülerin kaç kaç piksel olarak ayarlanacağı ve ne tip bir veriye sahip olduğu girilmelidir. (örneğin : binary ya da categorical)

4.7. Yapay Sinir Ağının Oluşturulması

Yapay sinir ağının oluşturulması 2 kısımdan oluşur. Bunlar yapay sinir ağının tanımlamalarının yapılması ve MobileNetV2 ağının tanıtılmasıdır.

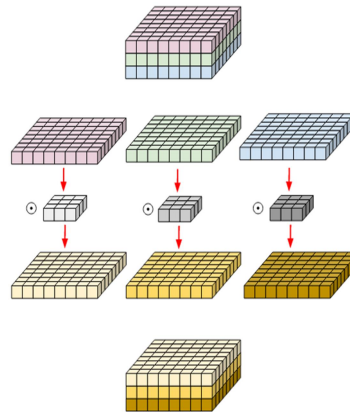
4.7.1. Tanımlamaların yapılması

- Conv2D : 2 boyutlu Evrişimli Sinir Ağı tanımlarken kullanılır. Çekirdek (kernel) matrisin giriş tensoruna maskelendiği ve her bir çekirdek matrisinin elemanının ilgili giriş tensorundaki matris ile çarpılarak çıkan sonuçların sırayla toplanması ve çekirdek matrisinin yerini aldığı giriş tensorundaki matrisin ortadaki elemanına yazılır. Kenarların boyutu konvolüsyon sonucu olan matriste yer almaz. Eğer kenarların da olması isteniyorsa “padding” parametresi “same” olmalıdır. $N \times N$ boyutundaki her bir çekirdek (kernel) matrisi giriş tensorunda konvolüsyon işlemi yapmak için gezinir. (Şekil 4.2)



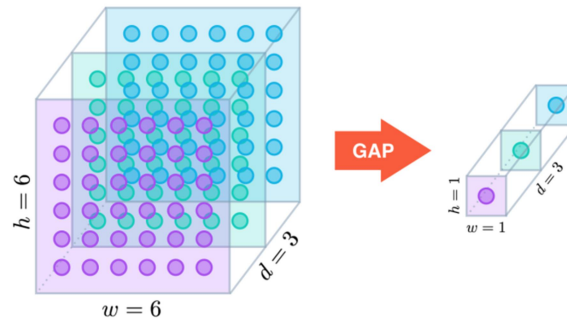
Şekil 4.2. Conv2D Gösterimi

- DepthWise Conv2D : Konvolüsyon işleminde gelen veri fotoğraf formatında olduğu için bu verinin 3 kanalı vardır. Bunlar da RGB olarak adlandırılan fotoğraftaki kırmızı tonları gösteren R bileşeni yeşil tonları gösteren G bileşeni ve mavi tonları gösteren B bileşenidir. Conv2D'den farklı olarak bu üç kanalın her biri ayrılarak her birine uygun olduğu kanalla eşleşen çekirdek(kernel) matrisi uygulanır.



Şekil 4.3. Depthwise Conv2D

- Global Average Pooling2D : İlgilenilen 2 boyutlu bir matrisin her bir elemanının toplanıp eleman sayısına bölünmesi işlemi olarak adlandırılır. Pooling işlemlerinin genel amacı boyut küçültmedir ve küçülen boyuta karşılık matrisin nasıl doldurulacağı pooling işleminin tipi ile belirlenir. Eğer “average pooling” kullanılıyorsa ortalama “max pooling” kullanılıyorsa pool_size kadarlık bir matristeki maksimum değer alınır.



Şekil 4.4. Global Average Pooling

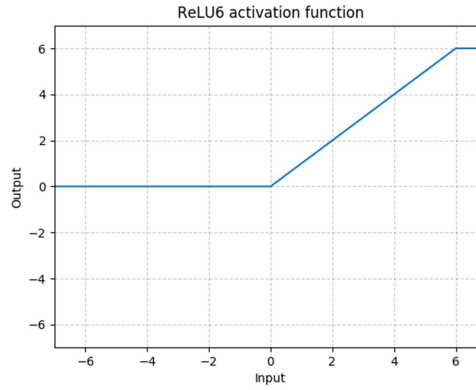
- Expansion Layer : Temel amacı gelen verinin kanal sayısını büyütmektir. Kanal sayısı expansion factor parametresinde verilen değerle çarpılarak büyür.
- Batch Normalization : Genel anlamıyla bir yapay sinir ağının gizli katmanlarındaki parametreleri normalizasyon işlemine tabi tutar. Normalizasyon formülünde de görüldüğü gibi net değerin normalize edilmiş halini bulabilmek için net değeri net değerlerin ortalamasından çıkarıp (mü değeri) standart sapmasının karesini alıp belirlenen epsilon değeri ile toplanıp bölüldüğünde ilgili net değerin normalize edilmiş hali bulunulacaktır. Epsilon değeri Batch Normalization katmanını tanımlarken parametre olarak verilebilir.[5]

$$z_{norm}^{(i)} = \frac{z^{(i)} - \mu}{\sqrt{\sigma^2 + \epsilon}}$$

Şekil 4.5. Batch Normalization Formülü

- Projection Layer : Temel amacı gelen verinin kanal sayısını küçültmektir.
- Dropout : Kendisinden önce gelen katmanın ağırlık değerlerinin Dropout katmanında verilen parametrenin değeri oranında kesilmesini sağlar. Bu sayede ağız ezberleme yapması önlenmiş olur.
- Dense : Genelleştirilmiş yapay sinir ağına verilen katmandır.
- RELU6 : Rectified Linear Unit'(Düzleştirilmiş Doğrusal Birim) in farklı bir versiyonudur. X ekseninin negatif değerlerinde 0 pozitif değerlerinde

doğrusal olarak artan ve X eksenini 6 ve 6'dan büyük olduğunda 1 döndüren bir aktivasyon fonksiyonudur.[6]



Şekil 4.6. RELU Fonksiyon Grafiği

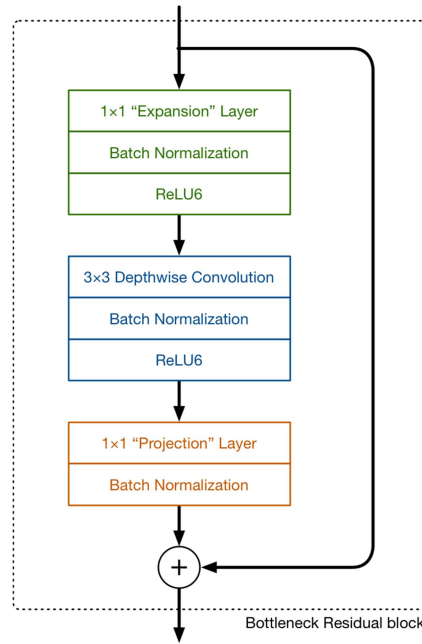
- Softmax : Softmax aktivasyon fonksiyonu $x \in \mathbb{R}^n$ skorlarının bir vektörünü girdi olarak alan ve $p \in \mathbb{R}^n$ çıkış olasılık vektörünü oluşturan genelleştirilmiş bir lojistik fonksiyon olarak görülebilir.[7]

$$p = \begin{pmatrix} p_1 \\ \vdots \\ p_n \end{pmatrix} \quad \text{buna karşılık} \quad p_i = \frac{e^{x_i}}{\sum_{j=1}^n e^{x_j}}$$

Şekil 4.7. Softmax Formülizasyon

4.7.2. MobileNetV2 ağına tanıtılması

MobileNetV2 ağı kendi içinde 16 blok katmanına sahiptir. Her bir katmanda her blok birbiriyle aynı değil ancak bloklarda Batch Normalization, Conv2D, Depthwise Conv2D, Zero Padding2D, Expansion Layer, Projection Layer gibi katmanlar bulunmaktadır. Ayrıca katmanlarda residual connection bulunmaktadır ve her bir blok Bottleneck Residual Block olarak isimlendirilir. MobileNetV2 ana blok yapısı aşağıdaki şekilde gösterilmiştir.(Şekil 4.8)



Şekil 4.8. Bottleneck Residual Block

4.8. Modelin Derlenmesi ve Başarım ve Kayıp Oranlarının Elde Edilmesi

Modelin derlenmesi aşamasında kayıp(loss) parametresi olarak sınıflandırma yapıldığı için kategorik çarpım entropisi(Örneğin bir sayının tahmininde ortalama hata kareleri toplamı daha faydalı olabilir.) kullanılmıştır. Optimizasyon algoritması olarak RMSprop fonksiyonu seçilmiştir ve öğrenme katsayısı olarak %0,1 uygun görülmüştür. Metrik olarak başarımlı(accuracy) seçilmiştir. Epoch değeri olarak 20 ideal görülmüştür.

Bu değerlerin bazıları teorik bilgiler ışığında bazıları ise denem yanılma yöntemiyle belirlenmiştir. Başarımdaki önemli kriterler sinirsel ağı ezberleme yapmaması, iyi bir şekilde çalışabilmesi ve belli bir hata payıyla doğru sonuçlar üretebilmesidir. Aşağıda son epochtaki eğitim ve test veri setindeki başarımlı ve kayıp oranları gösterilmiştir.

```

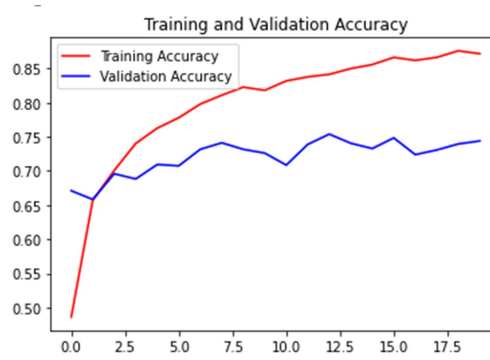
Epoch 20/20
228/228 [=====] - 93s 409ms/step - loss: 0.4179
- accuracy: 0.8715 - val_loss: 1.2811 - val_accuracy: 0.7439

```

Şekil 4.9. Eğitim Sonucu

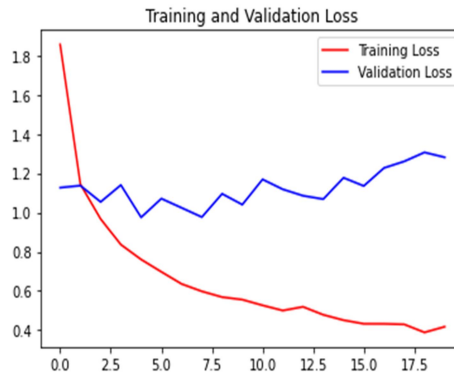
4.9. Başarım ve Kayıp Grafiklerinin Çizdirilmesi

Bu aşamada başarı ve kayıp oranlarının epoch değerlerine göre çizdirilmesi Python'ın görselleştirme kütüphanesi olan Matplotlib ile yapılmıştır. Eğitim ve test veri setindeki başarı yüzdesi aşağıdaki şekilde gösterilmiştir. (Şekil 4.10)



Şekil 4.10. Eğitim ve Test Başarı Grafiği

Eğitim ve test için kayıp oranları aşağıdaki şekilde gösterilmiştir. (Şekil 3.10)



Şekil 4.11. Eğitim ve Test Kayıp Grafiği

4.10. Modelin Kaydedilmesi

Son aşama olarak Python'da keras dosyası(h5 uzantılı dosya) olarak kaydedilen model Tensorflow'un TF Lite modülüyle tflite uzantılı dosyaya dönüştürülmüştür. Bu dosya Python'da oluşturduğumuz yapay sinir ağı modelini saklar. TF Lite dosyası daha sonra Android Studio'da mobil uygulamanın çalışmasında kullanılacaktır.

BÖLÜM 5. ANDROID VE WEB SUNUCU YAZILIMLARININ GELİŞTİRİLMESİ

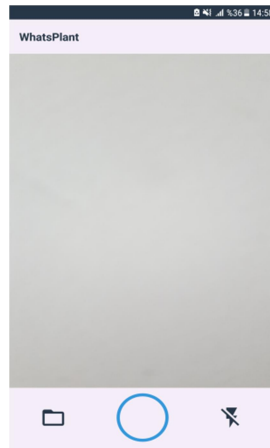
Projenin android kısmı üç bölümden oluşmaktadır bunlar; uygulama içi Api kamera kullanarak resimlerin çekildiği yada galeriden seçildiği kısım, çekilen veya galeriden seçilen resmin kırıldığı kısım en son olarak kırılan resmin türünün bulunduğu ve bulunan türe ait özelliklerin listelendiği kısım. Resim çekme sayfasında FrameLayout kullanılarak CameraPreview sayfasından gelen kamera görüntüsü gösterilmektedir ve bu sayfada galeriden resim çekme, flaş durumunu ayarlama ve resim çekmek için butonlar bulunmaktadır bu sayfada kamera resim çekmek istendiğinde ilk olarak mCamera.takePicture fonksiyonunu çağırır bu fonksiyon çağırılması tamamlandığı zaman yani resim verileri hazır durumda olduğunda Camera.PictureCallback dinleyici fonksiyonu devreye girer ve bu fonksiyonunun alt işlevi olarak onPictureTaken çalışır, bu fonksiyonda kameradan gelen byte dizisi şeklindeki veriler uygulamanın cache klasörü altında bir jpeg uzantısı dsya oluşturularak verileri oluşturduğumuz dosyaya yazmaktayız. Yazma işlemi gerçekleştirildikten sonra cropImage fonksiyonu ile resim kırpıcı sayfasına gönderilmektedir. Kırpma işlemi tamalandıktan sonra kaydedilen resim Classify sayfasına gönderilerek kırılan resim ImageView ile ekrana basılır, türü bul butonuna basıldığında tfflite uzantılı modele resim ilk önce bitmap verinin RGB değerleri bulunur sonrasında normalizasyon işlemi yapılarak gönderilir ve modelden sonuç olarak gelen tür bilgisi doğruluk değeriyle birlikte ekrana yazılır ve türe ait özelliklerde ilgili alanlarda listelenir.



Şekil 5.1. Örnek Uygulama Çıktısı

5.1. MainActivity Sayfasının Yapısı

MainActivity program açıldığında ilk işleyen kısımdır yani fotoğraf çekme, fotoğraf seçme ve flaş ayarlarının yapıldığı kısımdır. MainActivity sınıfının kurucu fonksiyonunda Xml tarafında oluşturulan elemanların tanımlanması yapılır ve butonlara basıldığında ne yapılacağını belirten fonksiyonlar tanımlanmıştır. Fotoğraf çekme butonunda basıldığında android'in kendi fotoğraf ve galeri sekmesi açılır. Galeriyi sayfasını açmak için intent sınıfına parametre olarak (Intent.ACTION_PICK) parametresi verilir. Galeriden seçilen resmin yolu kırpıcı sayfasına iletilir. Flaş butonu basıldığında ise flashMode değişkeni değerini değiştirerek flaş durumunu belirlemiş oluruz.



Şekil 5.2. MainActivity Sayfasının Görüntüsü

5.1.1. MainActivity arayüz tasarımı

MainActivity XML kısmında container olarak relativelayout kullanıldı. Relativelayout kullanılmasının nedeni nesneleri herhangi bir konuma yerleştirebilmemize imkan sağlıyor. Bu sayfada üç adet buton ve bir adet textureview kullanıldı. Birinci buton galeriden resim çekmemizi sağlayacak olan butondur. İkinci buton resmi çekmemizi sağlayacak olan buton ve son olarak flaş açma ve kapamayı sağlayan bir buton kullanıldı. Resim alma ve flaş açma kapama butonu kullanılan textureview elemanın 70 dp altında bulunmaktadır. Resim çekmeyi sağlayan buton ise textureview elemanının 95 dp altında ve textureview elemanının alt orta kısmında bulunmaktadır.



Şekil 5.3. MainActivity XML Görüntüsü

5.1.2. Kamera fonksiyonlarının işleyişi

Sayfa yüklendiğinde onCreate fonksiyonu içerisinde openCamera fonksiyonu çağırılır. Fonksiyon çağırıldığı zaman mCamera nesnesine getCameraInstance fonksiyonu üzerinden Camera.open() fonksiyonu ile kamera açılarak nesneye ataması yapılır. Kamera ataması yapıldıktan sonra mPreview değişkenine oluşturduğumuz CameraPreview sayfasından bir nesne oluştururuz ve parametre olarak mCamera değişkenini veririz. Sonrasında CameraPreview sayfasından gelen her bir görüntüyü ekrana basmak için ise FrameLayout nesnesine preview.addView(mPreview) fonksiyonu ile CameraPreview sayfasını FrameLayout'a atamasını yapmış oluruz. Son olarak setFlashMode, setFocusMode ve setRotation fonksiyonları ile kameraya flaş modunu, fokus modunu ve resmin döndürme oranını atayarak kamerayı hazır hale getirmiş oluruz.

5.1.3. OpenUtils fonksiyonunun işleyişi

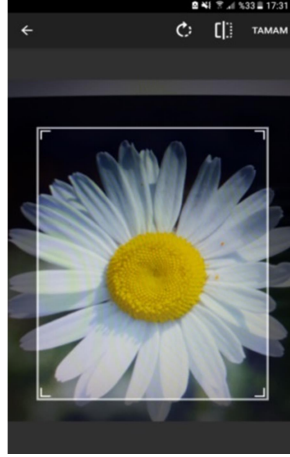
Bu fonksiyon içerisinde buton tanımlamaları yapılmaktadır. İlk olarak captureButton ile autoFocus fonksiyonu çalışarak kameranın focuslanması sağlanır ve focus işlemi tamamlandığında mCamera.takePicture ile kameradan görüntünün alınması sağlanır. Diğer bir buton olan flashButton ile ilk olarak flashModeChange fonksiyonu çalışır bu fonksiyon flashMode değişkenindeki değere göre flaş modunun ayarlanması sağlanır. Son olarak galleryButton ile Intent.ACTION_PICK parametresi ile galeri sayfasının açılması sağlanır.

5.1.4. MainActivity'deki diğer fonksiyonların işleyişi

Camera.PictureCallback nesnesinden türetilmiş olan mPicture fonksiyonu kamera çekimi tamamlandığında geriye dönen resim verisini uygulamanın içerisindeki cache dosyasına image.jpeg olarak kaydetmemizi sağlamaktadır ve kaydetme işlemi tamamlandıktan sonra cropImage fonksiyonu çalışmaktadır, bu fonksiyon kırpıcı sayfasına çektiğimiz resmi yollamamızı sağlamaktadır, kırpılma işlemi tamamlandıktan sonra onActivityResult fonksiyonu ile kırpılma sonucununda oluşan resmi uygulamanın cache dosyasına kaydedilmektedir. Kayıt işlemi tamamlandıktan sonra resmin yolu Classify sayfasına yollanarak sayfa açılmaktadır. Eğer resim galeriden çekilecek ise yine seçim sonucunu onActivityResult fonksiyonu üzerinden dinleyip seçilen resmin yolu Classify sayfasına yollanarak sayfa açılmaktadır.

5.1.5. Resmin kaydedilmesi ve kırpılması

BtnCapture butonuna basıldığında resim kaydedilir ve sonrasında CameraCaptureSession .CaptureCallback nesnesinden oluşturulan dinleyici fonksiyon Kamera ön izlemesini yeniler ve kırpıcıyı çalıştırır. Kırpıcı fonksiyonu kaydedilen resmin yolunu ve kırpıldıktan sonraki oluşturulan yolunu hazır kullandığımız kırpıcı olan CropImageActivity sayfasına parametre olarak verilir. Kırpıcı sayfasında kıpılma işlemi tamamlandığı zaman onActivityResult fonksiyonuna kırpılan resmin bilgileri gelir. Kırpma işlemi tamamlandığı zaman türün belirleneceği Classify sayfasına yönlendirme yapılır. Kullanıcı fotoğraf seçme kısmından yapıyor ise galeri yada fotoğraflardan seçtiği fotoğrafın yolu onActivityResult fonksiyonunda belirlenir ve kırpıcı sayfasına bu bilgiler parametre olarak gönderilir.



Şekil 5.4. Kırpıcı Sayfasının Görünümü

5.2. CameraPreview Sayfasının Yapısı

Bu sayfada açılan kameradan gelen görüntüleri ekrana basarak uygulamadaki herhangi bir değişiklik durumunda kamera ön izlemesinin durdurulması yada eğer önceden durdurulmuş isede önizlemenin tekrar başlatılmasını sağlamak ayrıca uygulama ilk açıldığında önizleme başlamadan önce kameranın rotasyon(yönlendirme) ve ekrana görselleri basabilmemiz için SurfaceHolder nesnesinin kameraya tanıtılması yapılır.

5.2.1. CameraPreview fonksiyonlarının işleyişi

Kamera canlı görüntüsünün tutulabilmesi için Android'in şart koştuğu kurallar vardır. Bu kurallardan bir tanesi SurfaceView sınıfını extend edip SurfaceHolder.Callback interface'ini de implement eden bir sınıf olmasıdır. Bu sınıfı tanımladıktan sonra SurfaceHolder ile oluşturulan görüntüleri ekranda sürekli tutmak için kullanılır. SurfaceHolder.Callback ise FrameLayout'daki değişikliklerde bilgi almak için kullanılan bir arabirimdir. SurfaceHolder.Callback çağrısının üç adet tanımlanması gereken metodu vardır bunlar surfaceCreated, surfaceChanged ve surfaceDestroyed tanımlanması gerekmektedir. İlk olarak surfaceCreated fonksiyonu ile kameradan gelen görüntü 90 derece yatık olduğundan onu düzeltmek amacıyla setDisplayOrientation fonksiyonu ile rotasyon ataması yapılır, atama yapıldıktan sonra kameradan gelen verileri oluşturduğumuz mHolder değişkeniyle senkron

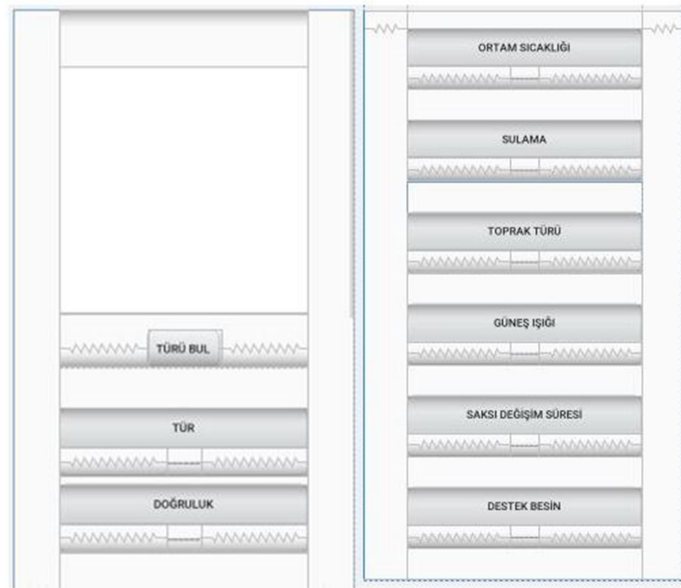
olarak gönderebilmemiz için kameranın setPreviewDisplay fonksiyonuyla kameradan gelen verileri artık holder aracılığı ile ekrana basabilmemize olanak sağlayacaktır sonrasında ise kameranın startPreview fonksiyonu ile önizleme başlatılır. surfaceChanged fonksiyonu ile FrameLayout üzerinde herhangi bir değişim sonucunda önizlemenin kapatılması yada kapalı ise tekrardan başlatılması yapılmaktadır. surfaceDestroyed fonksiyonu eğer uygulama kapatılacak yada alt sekmeye alınacak ise önizlemeyi kapatmamıza olanak sağlayacaktır.

5.3. Classify Sayfasının Yapısı

Classify sayfası galeriden seçilen resim yada fotoğrafı çekilip kaydedilen resmin yolunu alarak sayfadaki imageview nesnesi ile resim ilk açılışta ekranda gösterilmiş oluyor. Türü butonunun basıldığında imageview' de gösterilen resmin her bir pikselin RGB kodu whatteplant.tflite isimli modele gönderilerek modelden gelen float dizisinde en büyük doğruluk değeri seçilir ve label.txt dosyasından seçilen değere karşılık türün ismi bulunur ve ekranda türü ve doğruluk değerleri yazılmış olur. Diğer bir yandan türü bulunduktan sonra tür ismine göre önceden oluşturduğumuz feature.json dosyasındaki bitkilerin özelliklerinden tahmin edilen bitkinin adı kullanılarak bitkinin özellikleri tespit edilir ve ekranda ilgili alanlara özellikler listelenir.

5.3.1. Classify arayüz tasarımı

Classify XML kısmında container olarak main activity olduğu gibi relativelayout kullanıldı. Kırpılmış resmi görüntüleye bilmek için sayfanın ortasında bir adet ImageView nesnesi kullanıldı. Resmin altına türün bulunmasını sağlayacak bir buton ve butonun hemen altında ise türü ve doğruluk bilgilerinin listelendiği kutucuklar bulunmaktadır. Türü ve doğruluğun listelendiği kutuların altında ise bitkiyle alakalı ortam ısısı, sulama, toprak türü, güneş ışığı, saksı değişim süresi ve destek besin gibi özelliklerin listelendiği aynı türü ve doğrulukta olduğu gibi bu özelliklerde XML ile oluşturulmuş kutularla ekranda listelenmesi sağlanmaktadır.



Şekil 5.5. Classify Sayfasını Görüntüsü

5.3.2. Classfy onCreate fonksiyonunun işleyişi

Classify sayfası açıldığında modele gönderilecek olan resmin genişlik, yükseklik ve piksel boyutunu belirleyen değişkenlere 224, 224 ve 3 değerleri sırasıyla atanır. Resmin byte bilgilerini RGB çevirebilmek için image_mean ve image_std değişkenlerine 128 değeri atanır. Alınan resmin her bir bitini kutan intValues adında integer dizisi oluşturulur ve boyutuna genişlik ve yükseklik atanır. Oluşturduğumuz tflite değişkenine oluşturduğumuz linear.tflite dosyası atanır.

```
private MappedByteBuffer loadModelFile() throws IOException {
    AssetFileDescriptor fileDescriptor = this.getAssets().openFd(chosen);
    FileInputStream inputStream = new FileInputStream(fileDescriptor.getFileDescriptor());
    FileChannel fileChannel = inputStream.getChannel();
    long startOffset = fileDescriptor.getStartOffset();
    long declaredLength = fileDescriptor.getDeclaredLength();
    return fileChannel.map(FileChannel.MapMode.READ_ONLY, startOffset, declaredLength);
}
```

Şekil 5.6. Modelin Yüklendiği Kısım

LabelList değişkenine label.txt dosyasından okuduğumuz verileri sırasıyla ataması yapılır. ByteBuffer nesnesinden imgData değişkeni oluşturularak her bir pixel' in float cinsinden RGB kodlarının tutulmasını sağlar. imgData değişkeninin boyutunu

genişlik, yükseklik, pixel boyutu ve bit sayısı ile çarpılarak oluşturulur. Türü bul butonuna basıldığında imageView' deki resmin bitmap cinsinden verisi alınır, alınan veri getResizedBitmap fonksiyonu ile genişliği ve yüksekliği verilen boyutlarda tekrardan bitmap verisi oluşturulur. convertBitmapToByteBuffer fonksiyonu ile bitmap verisinden çıkarılan her bir RGB değerini float olacak şekilde ByteBuffer değişkenine eklenerek bu değişken modele girdi olarak verilir ve çıktılar float dizi olarak labelProbArray değişkenine atanır. En son olarak printTopKLabels fonksiyonu ile sonuçlar ekrana yazdırılır. Kırpıcı sayfasından gelen resim uri getParcelableExtra ile alınarak imageView' e resim atanmış olur.

5.3.3. ConvertBitmapToByteBuffer fonksiyonunun işleyişi

Bu fonksiyonda ilk olarak imageView nesnesinden aldığımız bitmap verisinden getPixels metodu ile ilk pikselden başlayarak her bir piksel değerleri intValues değişkenine atanır. Genişlik ve yükseklik değerleri kadar döngüyle her bir piksel değeri alınır. Alınan değerden bir formülizasyon ile her bir 0 ila 255 arası RGB değerleri imgData değişkenine atanmış olur.

```
int pixel = 0;
for (int i = 0; i < DIM_IMG_SIZE_X; ++i) {
    for (int j = 0; j < DIM_IMG_SIZE_Y; ++j) {
        final int val = intValues[pixel++];

        imgData.putFloat((((val >> 16) & 0xFF) - IMAGE_MEAN) / IMAGE_STD);
        imgData.putFloat((((val >> 8) & 0xFF) - IMAGE_MEAN) / IMAGE_STD);
        imgData.putFloat((((val & 0xFF) - IMAGE_MEAN) / IMAGE_STD);
    }
}
```

Şekil 5.7. RGB Çevirici Kod

5.3.4. PrintTopKLabels fonksiyonun işleyişi

Bu fonksiyonda önceden oluşturduğumuz PriorityQueue nesnesine labelList listesi içerisindeki tür sayısı kadar her bir gelen veriyi sırayla öncelikli kuyruk listesine ekliyoruz. Her eklemede liste sonu kontrolü yaparak eğer liste sonuna gelindiye sortedLabels.poll() komutu ile gelen diğer büyük verileri listenin başından eklemeye başlaması için göstergeyi listenin başına çeker. Sıralama yapıldıktan sonra listenin

son elemanın tür ismini topLables değişkenine atanır ve doğruluk oranında topConfidence değişkenine atanır. Atamalar bittikten sonra tür ve doğruluk oranını listeleyen textview' lere değişken değerlerinin ataması yapılır.

5.3.5. Bitki özelliklerinin çıkarılması

Özellik çıkarımında ilk olarak assets klasörü altındaki bitkilerin özelliklerinin tutulduğu feature.json dosyasının okunması için loadJSONFromAsset fonksiyonu yazılmıştır. Bu fonksiyon dosyayı önce byte dizisi olarak okuduktan sonra string formatına çevrilerek sonrasında get_feature fonksiyonundaki obj isimli JSONObject nesnesine gönderilir. Bu fonksiyonda json formatına çevrildikten sonra döngü kullanarak her bir bitkiyi tek tek kontrol ederek eğer tahmin edilen bitki ismi ile okunan bitkinin ismi aynı ise okunan bitkinin özellikleri feature adlı JSONObject nesnesine ataması yapılır. Atama Yapıldıktan sonra change_featurelabels fonksiyonu çağrılarak classify sayfasındaki ilgili alanlara bitkinin özellikleri listelenmesi için ekrana yazdırma işlemi gerçekleştirilir.

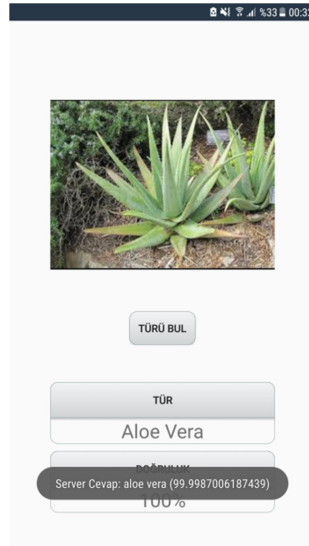
5.4. Web Sunucu Yazılımının Geliştirilmesi

Android uygulama üzerinden çektiğimiz bitki görüntüsünü localde çalışan bir python yazılımı ile görüntüyü HTTP protokolü aracılığı ile android uygulamasından sunucuya göndererek python ile programladığımız sunucuda resmi ön işlemlere tabi tuttukten sonra kaydetmiş olduğumuz modele ile test edip test sonucunu yine HTTP üzerinden response mesajı olarak android uygulamasına göndererek uygulamada ekranda sonuç yazılmaktadır.

5.4.1. Mobil uygulama kısmının işleyişi

Android kısmında HTTP protokolü üzerinden işlemler yapabilmemiz için öncelikle OkHttp3 kütüphanesini ekleemiz gerekmektedir. Bu uygulamada classify butonuna basıldığında local olarak model sonucu üretildikten sonra en son olarak onCreate fonksiyonunda getServerModel fonksiyonu çağrılmaktadır, bu fonksiyon ile bitmap formatındaki resim verisini ilk olarak jpeg formatına çevirerek sonrasında jpeg

verisini byte dizisine dönüştürmektedir. Sonrasında RequestBody değişkeni ile oluşturduğumuz byte dizisini androidFlask.jpg ismi ile <http://localhost:5000/> adresine göndermektedir. Gönderilen isteğe karşın sunucudan bir response mesajı gelir ise gelen mesaj response.body() fonksiyonu ile mesajdan sonuç çıkarılarak değer ekrana yazdırılır.



Şekil 5.8. Mobilde Sunucu Örnek Çıktısı

5.4.2. Web sunucu kısmının işleyişi

Sunucu kısmında android üzerinde de olduğu gibi burda da HTTP işlemleri yapabilmemizi sağlayacak olan python framework olan Flask kullanılmaktadır. İlk olarak Flask ile <http://localhost:5000/> adresini dinleyen bir uygulama yapılmaktadır. Gelen POST verisindeki image adlı dosya ilk olarak local klasör olan images klasöründe kaydedilmektedir. Kaydedilme işlemi tamamlandıktan sonra resim keras.preprocessing içerisindeki image fonksiyonu ile resim verisi alınır, sonrasında resim verisi normalizasyon işleminden geçtikten sonra modele girdi olarak verilir. Modelin test sonucu ise android uygulamasına cevap olarak gönderilir.

BÖLÜM 6. UYGULAMA

Bu uygulamada ilk olarak android bir telefona projenin apk dosyasını yüklemeliyiz. Yükleme tamamlandıktan sonra kamera ve dahili depolama izinlerini onayladıktan sonra istenirse resim galeriden yada direkt olarak uygulama üzerinden çekilir sonrasında çekilen resim kırpıcı sayfasına yollanır. Bu sayfada resim üzerinde kırpma yada resmi döndürme işlemleri gerçekleştirilebilir. Kırpma işlemi tamamlandığında türün ve özelliklerin belirleneceği ekran karşımıza gelmektedir. Bu sayfada türü bul butonuna basıldığında direkt olarak sonuçlar ilgili kutucukların içerisinde listelenecektir. Telefonda yapmış olduğumuz testler sonucunda her bitki için on adet resim denenerek toplamda 430 resim içerisinde 321 tenesini doğru bilmektedir bu da yaklaşık olarak %74 doğruluk oranı karşılık gelmektedir. Sonuçta da görüldüğü gibi eğitilen modelin validation oranıyla mobil test oranı aynı çıkmaktadır.

BÖLÜM 7. SONUÇLAR VE ÖNERİLER

Bu projenin çalışma konusu yapay zeka konularından nesne tanımanın mobil platformda gerçekleştirilmesidir. Bu problem için en uygun ağı MobileNetV2 olduğunu gördük, yeterli bir başarıma sahip olması mobil platforma uygulanmasının daha kolay olması ve diğer modellere göre daha hızlı çalışması bu ağı seçiminde etkili olmuştur. Kamera uygulamasının geliştirilmesinde Camera kütüphanesi kullanılmıştır, bu kütüphane CameraManager kütüphanesinden oluşturulduğundan flaş ve odaklanma otomatik hale getirildiğinden bir avantaj sağlanmaktadır. Yapılan testle sonucunda tüm bitkiler için %74 başarımla elde edilmiştir.

Yaklaşık 50 bitkinin tanımlanması ve başlıca özelliklerinin kullanıcıya aktarılması sağlanmıştır. Daha yüksek başarımla sahip model ile daha iyi sonuçlar elde edilebilir fakat bunun için daha geniş veri seti ve daha donanımlı cihazlar gerekmektedir. Bu uygulama bir drone entegre edilerek daha kapsamlı bir tanımlama sistemi oluşturulabilir ya da bu uygulama bir sulama sistemine entegre edilerek bitkilerden daha çok verim alınması sağlanabilir.

KAYNAKLAR

- [1] Artificial Intelligence Britannica.com, Eriřim Tarihi: 18.05.2020
- [2] <https://medium.com/@ayyucekizrak/deri%CC%87ne-daha-eri%CC%87ne-evri%C5%9Fimli-sinir-a%C4%9Flar%C4%B1-2813a2c8b2a9>, Eriřim Tarihi: 20.05.2020
- [3] <https://medium.com/@zurister/depth-wise-convolution-and-depth-wise-separable-convolution-37346565d4ec>, Eriřim Tarihi: 20.05.2020
- [4] <https://alexisbcook.github.io/2017/global-average-pooling-layers-for-object-localization/>, Eriřim Tarihi: 20.05.2020
- [5] <https://towardsdatascience.com/review-mobilenetv2-light-weight-model-image-classification-8febb490e61c>, Eriřim Tarihi: 20.05.2020
- [6] <https://stanford.edu/~shervine/l/tr/teaching/cs-230/cheatsheet-convolutional-neural-networks>, Eriřim Tarihi: 22.05.2020
- [7] <https://machinethink.net/blog/mobilenet-v2/>, Eriřim Tarihi: 22.05.2020

ÖZGEÇMİŞ

Muhammed Emin Sandıkçı, 22.12.1998 de Rize’ de doğdu. İlk, orta ve lise eğitimini Çayeli’de tamamladı. 2016 yılında Çayeli Barbaros Teknik Lisesi, Bilgisayar Bölümü’nden mezun oldu. 2016 yılında Sakarya Üniversitesi Bilgisayar Mühendisliği Bölümü’nü kazandı. 2019 yılında ZDC Bilişim Danışmanlık Ltd. Şti. Şirketinde yazılım stajını ve 2019 yılında da TürkTelekom Telekomünikasyon A.Ş. Şirketinde donanım stajını yapmıştır. SAÜ Bilgisayar Mühendisliği Bölümünde eğitimine devam etmektedir.

Yusuf Gören 05.07.1997 de İstanbul’da doğdu. İlk, orta ve lise eğitimini İstanbul’da tamamladı.2015 yılında Behiye Dr. Nevhiz Işıl Anadolu Lisesi’nden mezun oldu.2015 yılında Sakarya Üniversitesi Bilgisayar Mühendisliği Bölümü’nü kazandı.2019 yılında Un-maş ta donanım stajını yaptı.

Habibullah Metin, 24.05.1998 de İstanbul’da doğdu. İlk, orta ve lise eğitimini İstanbul’da tamamladı. 2016 yılında Beylikdüzü Birey Temel Lisesi’nden mezun oldu. Aynı yılda Sakarya Üniversitesi Bilgisayar Mühendisliği Bölümü’nü kazandı. 2018 yılında Türkiye Halk Bankası’nda yazılım stajını ve 2019 yılında yine Halk Bankası’nda donanım stajını yapmıştır.

BSM 498 BİTİRME ÇALIŞMASI DEĞERLENDİRME VE SÖZLÜ SINAV TUTANAĞI

KONU : Mobil Destekli Derin Öğrenme ile Bitki Tanıma ve Özellik Çıkarımı
ÖĞRENCİLER (Öğrenci No/AD/SOYAD): (B16210113/Muhammed Emin/SANDIKÇI)
(B151210074/Yusuf/GÖREN)(B161210059/Habibullah/METİN)

Değerlendirme Konusu	İstenenler	Not Aralığı	Not
Yazılı Çalışma			
Çalışma klavuza uygun olarak hazırlanmış mı?	x	0-5	
Teknik Yönden			
Problemin tanımı yapılmış mı?	x	0-5	
Geliştirilecek yazılımın/donanımın mimarisini içeren blok şeması (yazılımlar için veri akış şeması (dfd) da olabilir) çizilerek açıklanmış mı?			
Blok şemadaki birimler arasındaki bilgi akışına ait model/gösterim var mı?			
Yazılımın gereksinim listesi oluşturulmuş mu?			
Kullanılan/kullanılması düşünülen araçlar/teknolojiler anlatılmış mı?			
Donanımların programlanması/konfigürasyonu için yazılım gereksinimleri belirtilmiş mi?			
UML ile modelleme yapılmış mı?			
Veritabanları kullanılmış ise kavramsal model çıkarılmış mı? (Varlık ilişki modeli, noSQL kavramsal modelleri v.b.)			
Projeye yönelik iş-zaman çizelgesi çıkarılarak maliyet analizi yapılmış mı?			
Donanım bileşenlerinin maliyet analizi (prototip-adetli seri üretim vb.) çıkarılmış mı?			
Donanım için gerekli enerji analizi (minimum-uyku-aktif-maksimum) yapılmış mı?			
Grup çalışmalarında grup üyelerinin görev tanımları verilmiş mi (iş-zaman çizelgesinde belirtilebilir)?			
Sürüm denetim sistemi (Version Control System; Git, Subversion v.s.) kullanılmış mı?			
Sistemin genel testi için uygulanan metotlar ve iyileştirme süreçlerinin dökümü verilmiş mi?			
Yazılımın sızma testi yapılmış mı?			
Performans testi yapılmış mı?			
Tasarımın uygulamasında ortaya çıkan uyumsuzluklar ve aksaklıklar belirtilerek çözüm yöntemleri tartışılmış mı?			
Yapılan işlerin zorluk derecesi?	x	0-25	
Sözlü Sınav			
Yapılan sunum başarılı mı?	x	0-5	
Soruları yanıtlama yetkinliği?	x	0-20	
Devam Durumu			
Öğrenci dönem içerisindeki raporlarını düzenli olarak hazırladı mı?	x	0-5	
Diğer Maddeler			
Toplam			

DANIŞMAN (JÜRİ ADINA): DR. ÖĞR. ÜYESİ MUHAMMED FATİH ADAK
DANIŞMAN İMZASI: