



**T.C.**  
**KARABÜK ÜNİVERSİTESİ**  
**MÜHENDİSLİK FAKÜLTESİ**

**FAKE OR REAL FACE**

**SERHAT TEPELİ**  
**BURCU SARI**

**DANIŞMAN: Dr. HAKAN KUTUCU**

**2020**

# **FAKE OR REAL FACE**

**2011010205013 SERHAT TEPELİ**  
**2012010205075 BURCU SARI**

**Karabük Üniversitesi**  
**Mühendislik Fakültesi**  
**Bilgisayar Mühendisliği Bölümünde**  
**Bitirme Projesi Tezi**  
**Olarak Hazırlanmıştır.**

**KARABÜK**  
**2020**

## **KABUL**

Burcu SARI ve Serhat TEPELİ tarafından hazırlanan “FAKE OR REAL FACE” başlıklı bu projenin Bitirme Projesi Tezi olarak uygun olduğunu onaylarım.

Dr. Hakan KUTUCU

Bitirme Projesi Danışmanı, Bilgisayar Mühendisliği Anabilim Dalı .....

Bilgisayar Mühendisliği bölümü , bu tez ile, Bitirme Projesi Tezini onamıştır  
Dr. Hakan KUTCU  
BÖLÜM BAŞKANI

...../...../2020

.....

## **TEŞEKKÜR**

Bu tez çalışmasının planlanmasında, araştırılmasında, yürütülmesinde, oluşumunda ilgi ve desteğini esirgemeyen, engin bilgi ve tecrübelerinden yararlandığımız, yönlendirme ve bilgilendirmeleriyle çalışmamızı bilimsel temeller ışığında şekillendiren sayın Dr. Hakan KUTUCU' ya ve manevi desteğini esirgemeyen Şenol Güneş'e sonsuz teşekkürlerimizi sunarız.

**Bitirme Projesi**  
**YÜZ FOTOĞRAFLARINDA MANİPÜLASYON TESPİTİ**

**BURCU SARI**

**SERHAT TEPELİ**

**Karabük Üniversitesi**

**Mühendislik Fakültesi**

**Bilgisayar Mühendisliği**

**Dr. Hakan Kutucu**

**ÖZET**

Günümüzde insanlar sıklıkla Faceapp gibi programlarla fotoğraf manipülasyonu yapabilmektedir. Bu programlarda yapılan manipülasyonların ayırt edilebilmesi için CNN mimarisi kullanılacaktır. CNN’i eğitmek için gerekli olan sahte fotoğraflar Faceapp programında oluşturulmuş gerçek fotoğraflar ise insanlardan toplanmıştır.. CNN’ler sahte / gerçek yüzleri tanımak için yerel görüntü dokusu bilgilerini yakalayabilmektedir. CNN mimarisi ile birlikte sahte ve gerçek fotoğrafların ayırt edilebilmesi sağlanacaktır. Derin öğrenme ile oluşturulmuş model daha sonra bir mobil uygulama ile birlikte kullanılacaktır. Mobil uygulamada kullanıcının fotoğraf yüklediği ve yüklenen fotoğrafın tespitinin yapıldığı basit bir ara yüz oluşturulacaktır.

**Anahtar Kelimeler:** Derin Öğrenme, Yüz fotoğraflarında manipülasyon

**Bitirme Projesi**  
**YÜZ FOTOGRAFLARINDA MANİPÜLASYON TESPİTİ**  
**BURCU SARI**  
**SERHAT TEPELİ**  
**Karabuk University**  
**Faculty of Engineering**  
**Department of Computer Engineering**  
**Dr. Hakan Kutucu**  
**ABSTRACT**

Today, he is able to manipulate photos with programs such as Faceapp among people. CNN architecture will be used in these programs. The fake photos required for training CNNs were collected from people for real photos created in the Faceapp program. CNNs can capture local image textures to recognize fake / real faces. Together with the CNN architecture, it will ensure that fake and real photos are distinguished. Model created with deep learning then work with a mobile application. A simple interface will be created.

**Keywords:** Deep Learning, Manipulation in face photos

## İçindekiler

KABUL.....	i
TEŞEKKÜR.....	ii
ÖZET.....	iii
ABSTRACT .....	iv
ŞEKİLLER LİSTESİ.....	vi
BÖLÜM 1.....	1
GİRİŞ .....	1
1. Kapsam .....	1
2. Gereçesi ve Amacı .....	1
3. Literatür Özeti.....	2
BÖLÜM 2.....	4
1. Yöntemler .....	4
1.1. Derin Öğrenme.....	4
2. Uygulama Adımları.....	5
2.1. Veri Setinin Oluşturulması.....	5
2.2. Giriş Verisinin Oluşturulması .....	7
2.3. Sinir Ağının Oluşturulması.....	7
2.4. Oluşturulan Modelin Eğitilmesi .....	10
2.5. Modelin Test Aşaması.....	10
2.6. Modelin Android Studio ile Optimize Edilmesi .....	11
2.7. Android Studioda Arayüz Oluşturulması .....	11
2.8. Test Aşaması .....	11
3. Uygulama Teknolojileri.....	11
3.1. Keras .....	11
3.2. Derin Öğrenme Araçları.....	12
3.3. Python .....	12
3.4. Derin öğrenme ile Android Studio.....	12
BÖLÜM 3.....	13
1. Veri Setinin Düzenlenmesi .....	13
2. Modelin Eğitimi.....	14
3. Android Uygulamasının Oluşturulması.....	19
KAYNAKÇA.....	21

## ŞEKİLLER LİSTESİ

Şekil 1 CNN şeması [9].....	8
Şekil 2 Evrişim Katmanı [9].....	8
Şekil 3 Ortaklama Katmanı [9].....	9
Şekil 4 Tam bağlantı Katmanı [9].....	9
Şekil 5 ReLU Aktivasyon Fonksiyonu [9] .....	10
Şekil 6 Fake Veri Seti Dosya Görünümü .....	13
Şekil 7 Real Veri Seti Dosya Görünümü.....	13
Şekil 8 Model Şablonu .....	15
Şekil 9 Derin Öğrenme Arayüz .....	15
Şekil 10 Real Test Dosyası.....	16
Şekil 11 Fake Test Dosyası .....	16
Şekil 12 Real Fotoğrafın Sınıflandırma Sonucu .....	17
Şekil 13 Fake Fotoğrafın Sınıflandırma Sonucu.....	17
Şekil 14 Sunucu IP Adresi Çıktısı.....	18
Şekil 15 Fotoğraf seçme .....	19
Şekil 16 Fake Fotoğraf Çıktısı .....	20
Şekil 17 Real Fotoğraf Çıktısı .....	20

## TABLO LİSTESİ

Tablo 1 Fake Fotoğraf verisinden örnekler .....	6
Tablo 2 Oynanmış Fotoğrafların orijinal halleri .....	6
Tablo 3 Real Fotoğraf verisinden Örnekler .....	7



# BÖLÜM 1

## GİRİŞ

### 1. Kapsam

Fotoğraf düzenleme programları geliştikçe fotoğrafların gerçek ya da sahte olduğunu ayırt etmek giderek zor bir hal alıyor. Proje kapsamında derin öğrenme mimarileri kullanılarak evrimsel bir sinir ağı oluşturulacaktır. Model eğitimi gerçek ve sahte fotoğrafların kullanılmasıyla gerçekleştirilecektir. Bu eğitim sonucunda gerçek fotoğraflar ile sahte fotoğrafların ayırt edilebilmesi sağlanacaktır. Mobil uygulama kısmında ise kullanıcıdan fotoğraf yüklemesi istenecek, yüklenen fotoğrafın sahte ya da gerçek fotoğraf olup olmadığının bilgisi verilecektir.

### 2. Gerekçesi ve Amacı

Görselleri manipüle eden birçok yazılım bulunmaktadır ve kullanımları oldukça yaygındır. Günümüzde fotoğraf ile oynama işlemi bu yazılımlar sayesinde çok kolay hale gelmiştir. İnsanlar bu yazılımlar sayesinde cinsiyet değişikliği, gençleştirme yaşlandırma gibi işlemlerle kendilerine farklı görünümler vererek çok ileri seviyelerde manipülasyon işlemi yapılabilmektedir. Bu işlemler fotoğraflarda ki yüzlerin gerçek mi sahte mi olduğunun anlaşılmasını zorlaştırmaktadır, fotoğraf manipülasyonu ciddi bir sorun haline gelmiştir. Bu yazılımları yalnızca fotoğraf ve video sanatçıları kullanmamaktadır, bu tür yazılımlar günlük hayatta da insanlar tarafından en çok kullanılan yazılımlardır. İnsanlar fotoğraflar üzerinde yüz şekilleri, saç renkleri, göz renkleri, yaşlandırma, gençleştirme gibi birçok oynama yaparak kendilerine farklı bir insan görünümü verebilmektedir. Fotoğraflar üzerinde bu şekilde yapılan manipülasyonlar sahte yüzler ve sahte kanıtlar ile güvenliğe tehdit oluşturabilir ya da insanları kandırma amacıyla kullanılabilir. Bu projede bahsi geçen yazılımlar kullanılarak insanların kendi oluşturdukları başka bir insan görünümünde, gerçek halinden çok farklı değişikliklerle manipüle edilmiş yüz fotoğraflarının tespiti amaçlanmıştır.

### 3. Literatür Özeti

Literatürde manipüle edilmiş fotoğraflar ile gerçek fotoğrafları ayırt edebilen çalışmalara rastlanmıştır. En benzer çalışma olarak “Fake Face Detection Methods: Can They Be Generalized?” çalışması ve “Real or Fake: An Empirical Study and Improved Model For Fake Face Detection” bulundu.

Fake Face Detection Methods: Can They Be Generalized?” çalışmasında, yöntem olarak bu derin öğrenmenin alt dalı olan GAN mimarisi kullanılmıştır. Manipüle edilmiş fotoğrafları üretmek için çeşitli teknolojiler mevcuttur. “Derin öğrenmedeki son gelişmelerle, image morphing, Snap-Chat, (CGFI), GAN ve Face2Face gibi teknolojiler kullanarak şimdi sorunsuz bir şekilde manipüle edilmiş, gerçek zamanlı görüntüler / videolar oluşturmak mümkündür” [1] GAN mimarisi iki düşman sinir ağı gibi çalışmaktadır. Çalışmada iki sinir ağı kullanılmıştır. Ağlardan bir tanesi üretici ağ olarak çalışır, girdi olarak verilen fotoğraflardan manipüle edilmiş fotoğraflar üretir. Diğer ağ ise ayırt edici ağ olarak çalışır, üretici ağın çıktısı ve veri seti ile eğitim yapılır. Çıktı olarak da fotoğrafın real ya da fake fotoğraf olup olmadığını ayırt eder. Bizim projemizden farkı üretici ağıdır. Projemizde sahte yüz görüntülerinin ayırt edilmesi için derin öğrenme ile fotoğrafları ayırt etmektir. “Real or Fake: An Empirical Study and Improved Model For Fake Face Detection” çalışmasında ise GAN mimarisi ile CNN mimarisi incelenmiştir. Gerçek ve sahte yüz karşılaştırması yapılmıştır. Üretilmiş fotoğraflar ile gerçek yüz fotoğraflarının dokuları farklıdır. “Sahte yüzlerin doku istatistikleri doğal yüzlerden büyük ölçüde farklıdır. CNN'ler dokuyu sahte yüz tespiti için önemli bir ipucu olarak alır. ResNet-18 modeli eğitim verileri ve test verileri ise neredeyse el değmemiş sahte yüzleri tespit etmede mükemmeldir.” [2]

Adobe şirketinin de bu konuyla ilgili çalışmaları mevcuttur. Adobe, Max konferansında şirketin yeni 'About Face' aracını tanıtmıştır. About Face, hala geliştirilme aşamasında olan bir çalışmadır. “Berkeley Üniversitesi araştırmacılarıyla ortak çalışan Adobe, bir fotoğrafın değiştirilip değiştirilmediğini ve nasıl değiştirildiğini tespit etmek için şirketin Sensei AI adlı yapay zekasını kullanan bir araç yarattı.” [3] About Face Makine öğrenmesine odaklanmıştır. Fotoğraf piksellerini inceleyip, hangi bölgelerde değişiklik yapıldığını gösteren bir ısı haritası oluşturmaktadır. Fotoğraflardaki

değişikliği tespit edip, hangi bölgelerde ne kadar değişiklik yapıldığını da göstermektedir. About Face makine öğrenimi aynı fotoğrafın hem gerçek hem de sahte verisi kullanılarak bir eğitim gerçekleştirilmiştir. Bu yüzden sadece Adobe Photoshop programı ile yapılan değişiklikleri tespit edebilmektedir.

## BÖLÜM 2

### 1. Yöntemler

Gerçek ve sahte yüz fotoğraflarını ayırt edebilmek için manipüle edilmiş fotoğraflar ve gerçek manipüle edilmemiş fotoğraflar kullanılarak eğitim yapılacaktır. Eğitim için gerekli olan fotoğrafların bir kısmı Yonsei üniversitesinin açık kaynak olarak veri setini paylaşması ile elde edilmiştir. Diğer kısmı için çeşitli insanlara ait yüz fotoğrafları elde edilmiştir. Veri seti olarak bulunan fotoğraflardan bazıları manipüle edilerek, bazıları ise gerçek halleriyle bırakılarak eğitim gerçekleştirilecektir. Veri seti elde edildikten sonra derin öğrenme yöntemi ile fotoğraflar ayırt edilecektir. Yeni fotoğraf yüklendiğinde bu eğitime göre yüklenen fotoğrafın gerçek mi yoksa sahte mi olduğuna dair tahmin yapılacaktır. Projenin mobil uygulama kısmında ise kullanıcıdan fotoğraf yüklemesi istenecek ve yüklenen fotoğrafın gerçek fotoğraf ya da sahte fotoğraf olduğu söylenecektir.

Projenin gerçekleştirilebilmesi ve fotoğrafların gerçek mi sahte mi olduğunun öğretilmesi için kullanılacak yöntem derin öğrenme yöntemidir. Derin öğrenme, verilen bir veri kümesi ile çıktıları tahmin edebilen modelin eğitilmesine olanak sağlar.

#### 1.1. Derin Öğrenme

Derin Öğrenme bir makine öğrenme yöntemidir. Verilen bir veri kümesi ile çıktıları tahmin edecek modeli eğitmeye olanak sağlar. Modeli eğitmek için hem denetimli hem de denetimsiz öğrenme kullanılabilir. Temelde insanı taklit eden bir öğrenme biçimini otaya koyar. “Derin öğrenme temel olarak verinin temsilinden öğrenmeye dayalıdır. Bir görüntü için temsil denildiğinde; piksel başına yoğunluk değerlerinin bir vektörü veya kenar kümeleri, özel şekiller gibi özellikler düşünülebilir.” [4] Denetimli Öğrenme, girdilere ve beklenen çıktılarına sahip etiketli veri setlerini kullanmayı içerir. Denetimli öğrenmeyi kullanarak bir Yapay Zeka eğitirken, ona bir girdi verir ve beklenen çıktıyı söylersiniz.” [5] “Denetimli öğrenmede her örnek bir giriş değeri (vektör) ve bir de çıkış değeri (denetleyici sinyal) içeren bir çifttir. Denetlenen bir öğrenme

algoritması, eğitim verilerini analiz eder ve yeni örneklerin haritalanması için kullanılabilecek bir çıkarım fonksiyonu üretir. Bu sayede öğrenme algoritmanız, girdilerden ilgili hedeflere bir fonksiyon oluşturur. Eğer hedefler bazı sınıflarda ifade edilirse sınıflandırma problemi olarak adlandırılır.” [6] Eğer giriş (x) ve çıkış (Y) değişkeni varsa ve girişten çıkışa kadar eşleştirme işlevini öğrenmek için algoritma kullanıyorsak bu denetimli öğrenmedir. Tamamen basit bir fonksiyon mantığında işlemektedir.

Sınıflandırma: sistem, hangi verinin, hangi koşullarda, hangi sınıfa ait olacağı bilgisi ile sınıflandırılarak eğitilirse, yeni veri setindeki veriyi de öğrendiklerine benzer biçimde sınıflandırabilir.

## **2. Uygulama Adımları**

### **2.1. Veri Setinin Oluşturulması**

Sinir ağının eğitilmesi için .jpeg formatında hem gerçek hem sahte fotoğraf verileri kullanılmıştır. 897 adet gerçek fotoğraf verisi, 739 adet sahte fotoğraf verisi bulunmaktadır. Birçok insanın fotoğrafları çekilmiştir ve bir kısmı gerçek fotoğraflarda kullanılırken bir kısmı da sahte fotoğraf üretmek için kullanılmıştır. Sahte fotoğraf verilerinin tamamı Faceapp programı kullanılarak gençleştirme, yaşlandırma, cinsiyet değiştirme gibi işlemlerle manipüle edilmiştir. Çekilen fotoğraflar kullanılmadan önce photoscape adında bir fotoğraf düzenleme programında 600x600 piksel boyutlarına dönüştürülmüştür. Gerçek ve manipüle edilmiş fotoğrafların piksel boyutları aynıdır. Veri seti eğitim ve test verisi olarak ikiye bölünmesi gerekir.



Tablo 1 Fake Fotoğraf verisinden örnekler



Tablo 2 Oynanmış Fotoğrafların orijinal halleri



Tablo 3 Real Fotoğraf verisinden Örnekler

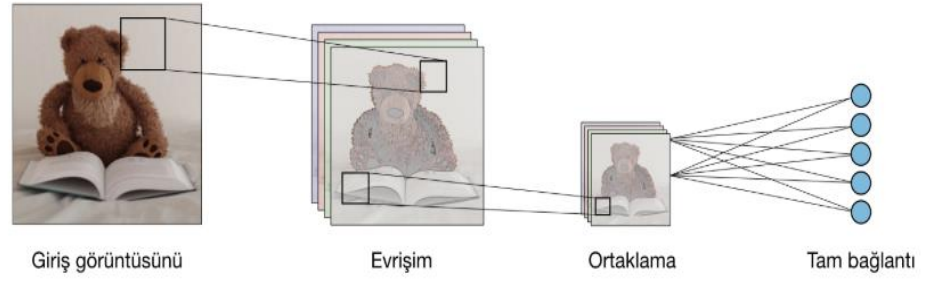
## 2.2.Giriş Verisinin Oluşturulması

**OpenCV:** Resimlerde yapılacak olan işlemler için gerekli olan kütüphane Opencv kütüphanesidir. “OpenCV (Açık Kaynak Bilgisayarlı Görme Kitaplığı) açık kaynaklı bir bilgisayar görme ve makine öğrenimi yazılım kütüphanesidir” [7]. OpenCV kütüphanesi ile birlikte resim veriler daha rahat bir şekilde okunabilmektedir. OpenCV kütüphanesi ile birlikte resimler okunacak ve giriş verisi olarak kullanılacaktır.

## 2.3. Sinir Ağının Oluşturulması

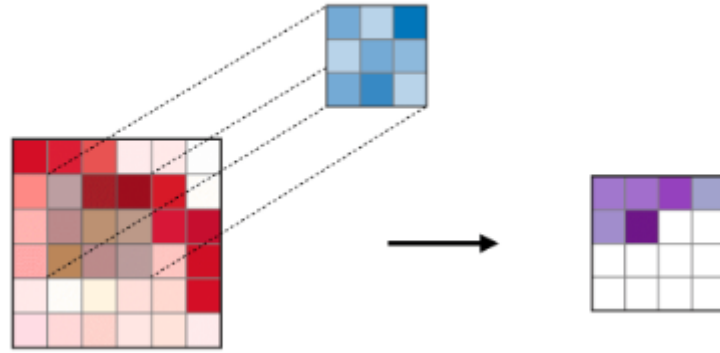
**Konvolüsyonel Sinir Ağları (CNN):** Bir girdi görüntüsü alır ve bu girdide ki çeşitli görüşleri ayırt edebilen bir derin öğrenme algoritmasıdır. “Konvolüsyonel Sinir Ağları (Convolution Neural Network-CNN) çok katmanlı algılayıcıların (Multi Layer Perceptron-MLP) bir türüdür” [4].“CNN’ler görüntüleri girdi olarak almak için tasarlanmış yapılardır ve bilgisayarlı görmede etkili bir şekilde kullanılmaktadır. CNN, bir veya birden çok konvolüsyonel katman ve standart çok katmanlı bir sinir ağı gibi bir veya daha fazla tamamen bağlı katmanlardan oluşmaktadır” [8].





Şekil 1 CNN şeması [9]

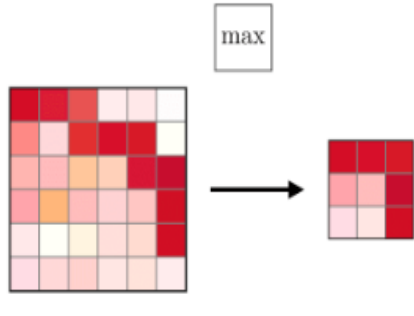
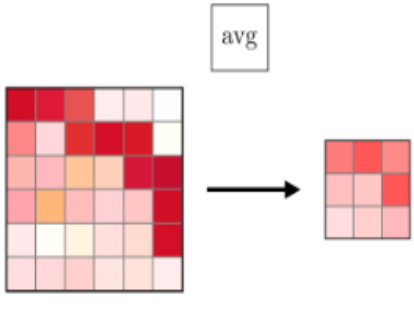
Bu mimaride Evrişim katmanı, havuz katmanı ve tam bağlantı bulunmaktadır. “Evrişim katmanı (CONV) evrişim işlemlerini gerçekleştiren filtreleri, I girişini boyutlarına göre tararken kullanır. Hiperparametreleri F filtre boyutunu ve S adımını içerir. Elde edilen çıktı O, öznitelik haritası veya aktivasyon haritası olarak adlandırılır.” [9]



Şekil 2 Evrişim Katmanı [9]

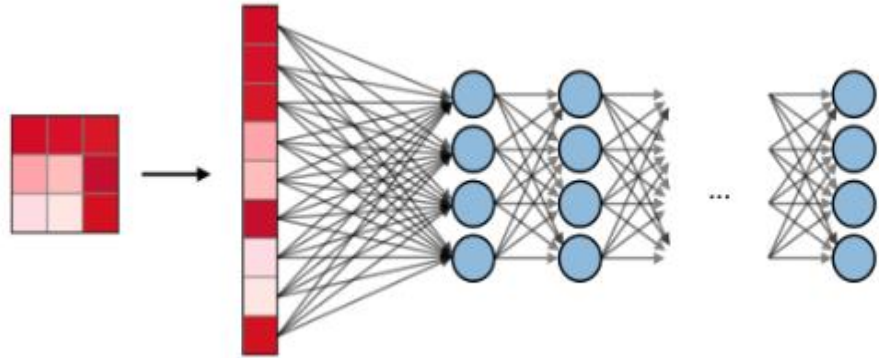
“Ortaklama katmanı (POOL), tipik olarak bir miktar uzamsal değişkenlik gösteren bir evrişim katmanından sonra uygulanan bir örnekleme işlemidir. Özellikle, maksimum ve ortalama ortaklama, sırasıyla maksimum ve ortalama değerin alındığı özel ortaklama türleridir.” [9] Ortaklama katmanında genellikle maksimum ortaklama yöntemi kullanılır.



Maksimum ortaklama	Ortalama ortaklama
Her ortaklama işlemi, geçerli matrisin maksimum değerini seçer	Her ortaklama işlemi, geçerli matrisin değerlerinin ortalaması alır
	

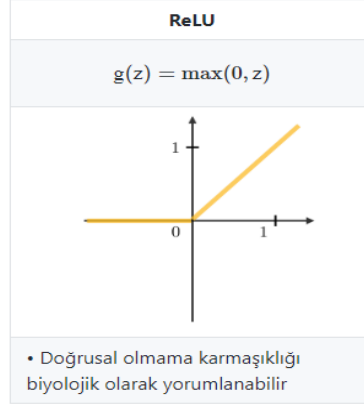
Şekil 3Ortaklama Katmanı [9]

“Tam bağlı katman (FC), her girişin tüm nöronlara bağlı olduğu bir giriş üzerinde çalışır. Eğer varsa, FC katmanları genellikle CNN mimarisinin sonuna doğru bulunur ve sınıf skorları gibi hedefleri optimize etmek için kullanılabilir.” [9]



Şekil 4 Tam bağlantı Katmanı [9]

**Aktivasyon Fonksiyonu:** CNN mimarisinde yaygın olarak kullanılan aktivasyon fonksiyonları kullanılmaktadır. “Düzeltilmiş doğrusal birim katmanı (ReLU), (g)'nin tüm elemanlarında kullanılan bir aktivasyon fonksiyonudur. Doğrusal olmamaları ile ağıın öğrenmesi amaçlanmaktadır” [9] Geleneksel CNN bir görüntüyü sınıflandırmakta kullanılır. Nesnenin olasılığını tahmin eder.



Şekil 5 ReLU Aktivasyon Fonksiyonu [9]

**Optimizasyon Algoritması:** Derin öğrenme uygulamalarında öğrenme işleme temelde bir optimizasyon problemidir. "Optimizasyon algoritmalarında öğrenme katsayısının ayarlanması modelin eğitiminde kritik bir rol oynamaktadır." [10] Adam optimizasyon algoritması birinci dereceden gradyan tabanlıdır. "Stokastik nesnel fonksiyonlar, alt dereceli momentlerin uyarlamalı tahminlerine dayanarak Yöntemin uygulanması basittir, hesaplama açısından verimlidir, bellek gereksinimlerinin az olması, degradelerin köşegen ölçeklendirilmesinde değişmez, ve veri ve / veya parametreler bakımından büyük olan problemler için çok uygundur." [11]

#### 2.4. Oluşturulan Modelin Eğitilmesi

Eğitim işlemi fit() işlevi ile gerçekleşmektedir. Eğitim verileri, hedef verileri, doğrulama seti, aynı anda eğitilecek veri miktarı her bir epoch'ta verilerin yerlerinin değişmesi için suffle, eğitim sırasında çıktıları görebilmek için verbose ve veri setinin kaç kez model üzerinden geçerek eğitileceğini belirleyen epoch sayısı gibi birçok parametre almaktadır.

#### 2.5. Modelin Test Aşaması

Modelin test aşamasında eğitilmiş olan modele yeni verilerle tahmin de bulundurulacaktır. Girdi olarak verilen verinin tahminiyle modelin başarı oranı karşılaştırılacak ve ne oranda başarılı olduğu hesaplanacaktır.

## **2.6.Modelin Android Studio ile Optimize Edilmesi**

Keras’da eğitilmiş ve testleri bitirilmiş olan model, android studio’ya optimize edilebilmesi için öncelikle bir python sunucusu oluşturulmuştur. Eğitilmiş olan model bu kısımdan bulunmaktadır. Python sunucusu Android studio ile entegre edilerek optimize edilmiştir.

## **2.7.Android Studioda Arayüz Oluşturulması**

Android Studio android cihazda çalışan uygulamalar geliştirmek için araçları sağlayan bir platformdur. Oluşturulacak olan ara yüzde kullanıcının fotoğraf yüklemesini sağlayan ve yüklenen fotoğrafta tahmin yapılmasını sağlayan bir buton ve Python sunucusuna erişimin sağlanması için bir edittext oluşturulacaktır. Aynı zamanda yüklenen fotoğraf kullanıcıya gösterilecektir. Yüklenen fotoğrafta tahmin işlemi yapıldıktan sonra yapılan tahmin ekranda gösterilecektir.

## **2.8. Test Aşaması**

Model android studio ile optimize edildikten ve arayüzü oluşturulduktan sonra yeni verilerle test edilecek ve analizi yapılacaktır.

# **3. Uygulama Teknolojileri**

## **3.1.Keras**

Derin öğrenme için çeşitli kütüphaneler kullanılmaktadır.“Keras, Python ile yazılmış ve TensorFlow, CNTK veya Theano'nun üzerinde çalışabilen üst düzey bir sinir ağları API'sidir.” [12].Derin sinir ağları ile hızlı deneyler sağlamak için tasarlanmıştır. Yüksek seviyeli bir API’dir. Keras ile bir ağ modeli oluşturulur ve verilen eğitim verisi ile bu model eğitilir, eğitim işlemi gerçekleştirildikten sonra test verisi ile olması gereken/beklenen değer tahmin edilip modelin performansı sorgulanır. “Theano, çok boyutlu dizileri içeren matematiksel ifadeleri tanımlamanızı, optimize etmenizi ve verimli bir şekilde değerlendirmenizi sağlayan bir Python kütüphanesidir. NumPy'nin üzerine inşa edilmiştir”. [13] Theano kendisinden sonra çıkan birçok kütüphaneye referans olmuştur. Tensorflow, Açık kaynak kodlu bir derin öğrenme kütüphanesidir. “TensorFlow ve makine öğrenmeye

başlamayı kolaylaştıran üst düzey Keras API'sini kullanarak modeller oluşturun ve eğitin.” [14]

### 3.2.Derin Öğrenme Araçları

**Kendi bilgisayarımızda:** Kullanılan dilin seçimine göre platformlar da değişmektedir. Python için Pycharm ve Anaconda IDE’leri ön plana çıkarken C/C++ programlama dilleri için Visual Studio kullanımı daha avantajlıdır.

**Bulut Ortamında:** Microsoft Azure Notebook (sadece CPU) ve Google Colab (GPU desteği var) herhangi bir kurulum gerektirmeksizin uygulama geliştirmeyi sağlamaktadır.

### 3.3.Python

Derin öğrenme birçok programlama dili ile yazılabilmektedir. Fakat bazı derin öğrenme kütüphaneleri dil konusunda sınırlıdır. Kullanılacak kütüphaneye göre dil seçimi yapılmaktadır. Python birçok derin öğrenme kütüphanesinin arka planında kullanılmasından dolayı en çok tercih edilen programlama dilidir. “Python Guido van Rossum tarafından tasarlanmış dinamik tipte bir programlama dilidir. Ruby gibi python’da programcılar arasında kolay okunacak şekilde tasarlanmıştır. Nesne yönelimli bir programlama dilidir.” [15]

### 3.4.Derin öğrenme ile Android Studio

Android Studio, android uygulama geliştiricileri için tasarlanan geniş kapsamlı bir programdır. “Android Studio, Android uygulamalarının geliştirildiği, üst seviye özelliklere sahip ve Google tarafından da önerilen resmi programlama aracıdır” [16] Derin öğrenme modellerinin android studio’da kullanımı için tensorflow lite sürümü geliştirilmiştir. “Makine öğrenmesi (ML), uygulamalarınıza açıkça programlanmadan deneyimlerinizden otomatik olarak öğrenme ve iyileştirme yeteneği sağlayan bir programlama tekniğidir. Bu, özellikle görüntüler ve metinler gibi yapılandırılmamış verileri kullanan veya kazanan spor takımını tahmin etme gibi çok sayıda parametreyle ilgili sorunlar kullanan uygulamalar için çok uygundur.” [17]

## BÖLÜM 3

### 1. Veri Setinin Düzenlenmesi

Derin öğrenme modelinin eğitilmesi için daha Fake ve Real şeklinde oluşturulmuş fotoğraf veri seti Train ve Test verisi olarak ikiye bölünmüştür. Train ve Test verisi de kendi içinde Fake ve Real verisi olarak ikiye bölünmüştür. Bu şekilde modelin eğitimi için gerekli olan fotoğraflar Train ve Test dosyalarından çağrılacaktır. Train dosyasından gelen fotoğrafları model eğitim için kullanırken Test dosyasından gelen veriyi de test etmek için kullanmaktadır.



Şekil 6 Fake Veri Seti Dosya Görünümü



Şekil 7 Real Veri Seti Dosya Görünümü

## 2. Modelin Eğitimi

Model Eğitimi yapabilmek için öncelikle python dosyasına Train ve Test dosyasında ki fotoğraflar çağrılmıştır. Okunan fotoğraf resize ve reshape edilmiştir. Model eğitiminde başarı oranını yükseltmek amacıyla bazı teknikler uygulanmıştır. Görüntü verisini yeniden boyutlandırılmıştır. Alınan resmin genişlik ve yükseklik boyutları değiştirilmiştir. Alınan görüntü verisinin en boy oranı 64x64 olarak belirlenmiştir. Boyutlandırılan resim diziye aktarılmıştır. Sınıflandırma işlemi için gereken beklenen çıktı, Fake fotoğraf için 0, Real fotoğraf için 1 olarak belirlenmiştir. Model eğitiminde optimizasyon algoritması olarak adam kullanılmıştır. Sequential Model her katmanın tam olarak bir giriş tensörüne ve çıkış tensörüne sahip olduğu düz bir katman yığını i.in uygundur.

**Konvolüsyonel Katmanı:** Katman olarak Conv2D kullanılmıştır. Bir çıkış tensorü üretmek için katman girdisi ile bir çıkış çekirdeği üretir.

**Loss Fonksiyonu:** Tasarlanan modelin hata oranını aynı zamanda başarı oranının gösteren fonksiyondur. Loss fonksiyonu son katmanda tanımlanmaktadır. Oluşturulan model de tek etiket kategorizasyonu için kullanılan bir loss fonksiyon olan categorical\_crossentropy kullanılmıştır.

**Epoch Sayısı:** Model eğitilirken verilerin tamamı eğitime katılamaz. Parçalar halinde eğitimde yer alabilirler. İlk parça eğitildikten sonra modelin başarıyı test edilir ve başarıya göre geri yayılım ile ağırlıklar güncellenir. Daha sonra yeni eğitim kümesi ile model tekrar eğitilip ağırlıklar tekrar güncellenir. Bu işlem her bir eğitim adımında tekrarlanarak model için en uygun ağırlık değerleri hesaplanmaya çalışılır. Bu eğitim adımlarının her birine “epoch” denilmektedir. Modelde epoch sayısı 18 verilmiştir.

**Batch Boyutu:** Modelin aynı anda kaç veriyi işleyeceğini gösterir. Bu projede batch boyutu 16 olarak belirlenmiştir.

Model fit edilmiştir.

**Metrik:** Tahminlerin etiketlere ne sıklıkta eşit olduğunu hesaplar.

**Yapılandırma:** Model eğitimi için yapılandırma yapılması gerekmektedir. Bu yapılandırmalar ise kerastaki fit ve compile methodları ile yapılabilmektedir. Fit methodunda parametre olarak optimizasyon

algoriması batch boyutu, eğitim verilerinin her epoch'dan önce karıştırılıp karıştırılmayacağını belirten shuffle, Doğrulama verisi olarak kullanılacak eğitim verilerinin kesri olan validation split, işlem sırasında ekrana ayrıntılı verilerin dökülmesini sağlayan verbose parametresi verilmiştir. Compile methodunda ise parametre olarak optimizasyon algoritması, loss fonksiyonu, metrik değeri belirlenmiştir.

```
Model: "sequential_1"
```

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 62, 62, 32)	896
max_pooling2d_1 (MaxPooling2D)	(None, 31, 31, 32)	0
conv2d_2 (Conv2D)	(None, 29, 29, 32)	9248
max_pooling2d_2 (MaxPooling2D)	(None, 14, 14, 32)	0
flatten_1 (Flatten)	(None, 6272)	0
dense_1 (Dense)	(None, 128)	802944
dense_2 (Dense)	(None, 2)	258

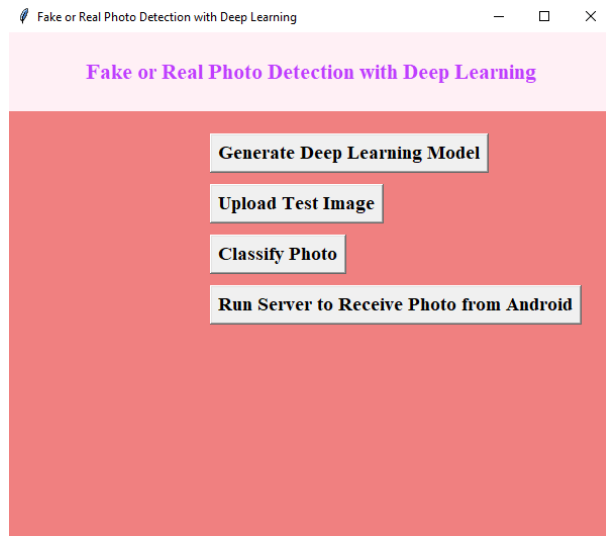
```

Total params: 813,346
Trainable params: 813,346
Non-trainable params: 0
None

```

Şekil 8 Model Şablonu

Eğitilen modelin başarımlı oranı %70'dir. İstenilen başarımlı oranına ulaştıktan sonra model ağırlıkları save methoduyla kaydedilmiştir. Modelin üretilmesi, test edilmesi ve Android için sunucunun çalıştırılması amacıyla bir arayüz oluşturulmuştur.

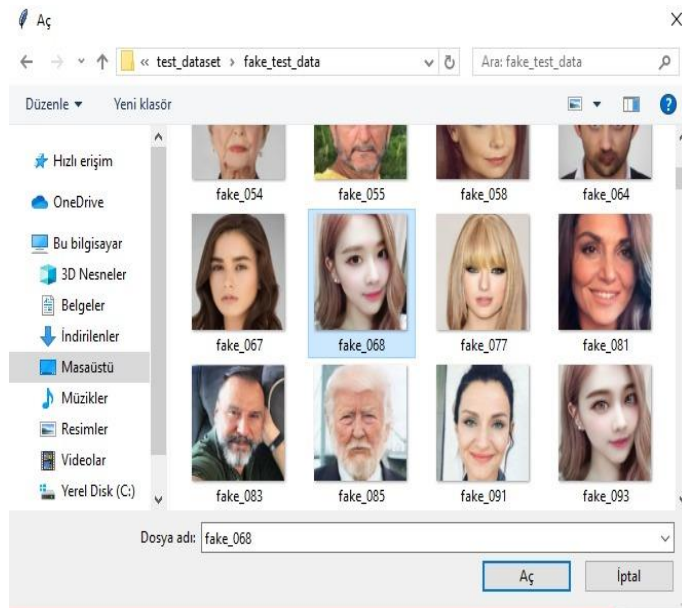


Şekil 9 Derin Öğrenme Arayüz

Arayüzde oluşturulan “Generate Deep Learning Model” butonuna tıklayarak model üretimi gerçekleştirilmektedir. Kısımda eğer kaydedilmiş bir model var ise o modelin ağırlıkları yüklenir. Kaydedilmiş bir model yok ise o zaman derin öğrenme eğitimi gerçekleştirilmektedir. “Upload Test Image” butonuna tıklandığında ise test veri setine ulaşırız. Test veri seti arasından fotoğraf seçimi yapılabilmektedir. Test dosyasından fotoğraf seçilip fotoğraf yüklemesi yapılmaktadır.



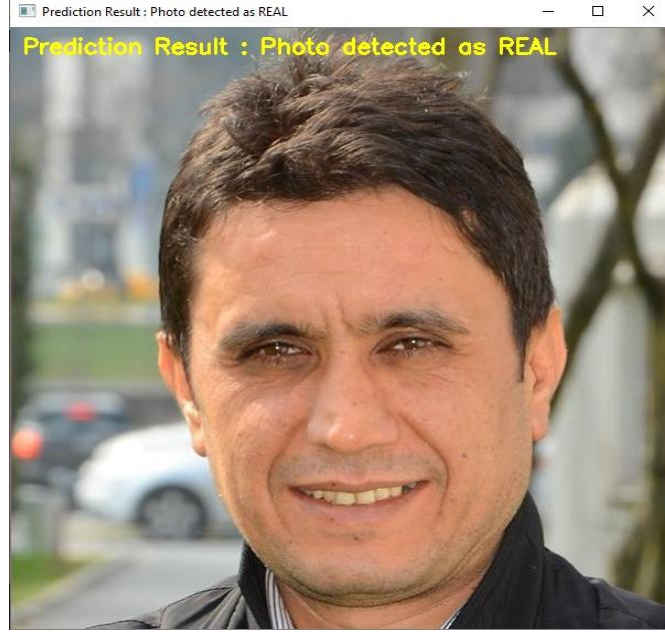
Şekil 10 Real Test Dosyası



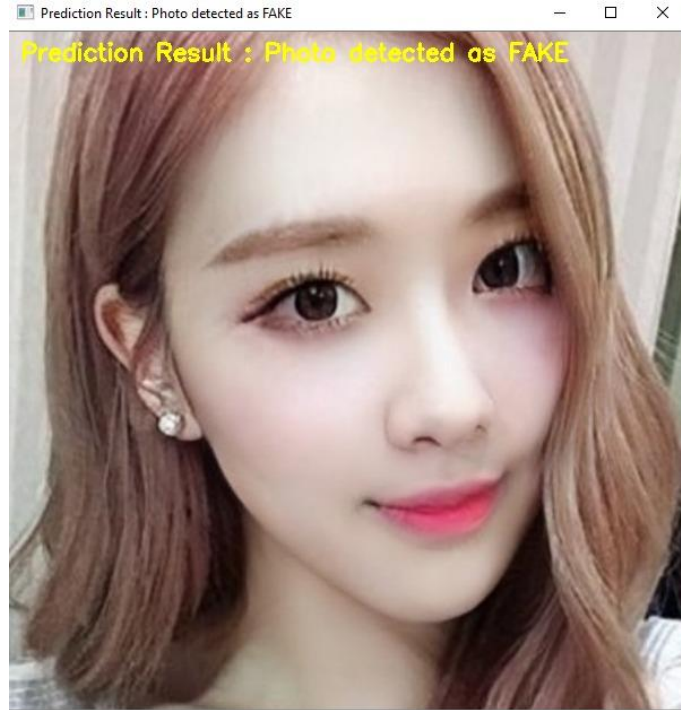
Şekil 11 Fake Test Dosyası



“Classify Photo” butonuna tıklıldığında ise yüklediğimiz fotoğrafı ve sınıflandırma sonucunu görebilmekteyiz. Bu şekilde arayüz kısmından da eğitilmiş olan modelin başarısını test edebilmekteyiz.



Şekil 12 Real Fotoğrafın Sınıflandırma Sonucu



Şekil 13 Fake Fotoğrafın Sınıflandırma Sonucu

Kaydedilen modelin Android Studio da çağırılıp kullanılması açısından Python server oluşturulmuştur. Android studiodan yüklenen fotoğraf sunucuda değerlendirilecektir. Sunucuda değerlendirilen fotoğrafların sonuçları yine Android Studioya gönderilecektir. Python Arayüzünde oluşturulan “Run Server to Receive Photo From Android” butonuyla oluşturulan sunucu Android uygulamasında kullanılmak amacıyla çalıştırılmaktadır. Bu şekilde oluşturulan model Android uygulaması ile optimize edilmiştir. Ve çıktı olarak sunucun IP adresini vermektedir.

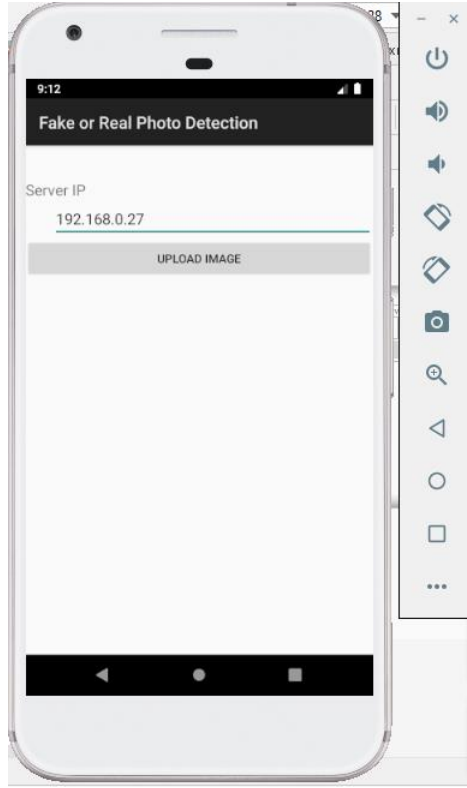


```
None
192.168.0.27
```

Şekil 14 Sunucu IP Adresi Çıktısı

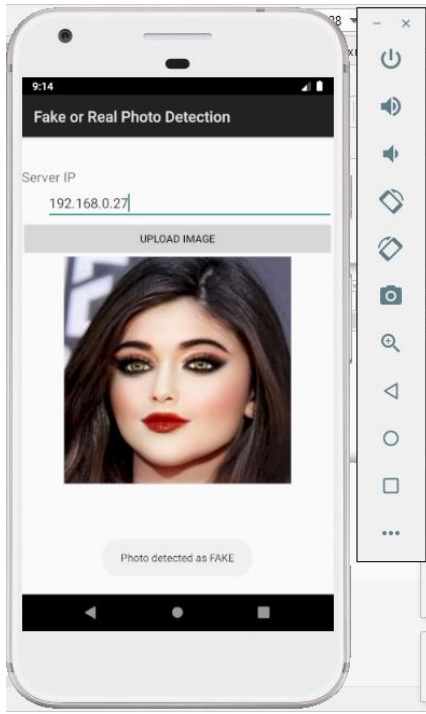
### 3. Android Uygulamasının Oluřturulması

Android uygulama kısmında bir adet buton bir adet imageview ve bir adet edittext kullanılarak basit bir arayüz tasarlanmıřtır. İlk ekranda fotoğraf seçimi ve sunucuya erişim sağlanmaktadır. Oluřturulan edittext kısmından sunucu IP adresi girildikten sonra fotoğraf seçimi için oluřturulan “UPLOAD IMAGE” butonuna tıklayarak yapılmaktadır. Butona tıklandıktan sonra galeriye girerek fotoğraf seçimi yapılmaktadır.

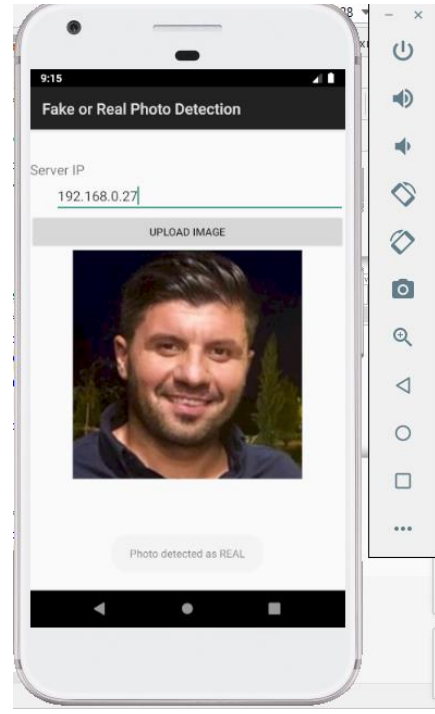


řekil 15 Fotoğraf seçme

Seilen fotoğraf bytearraya dnřtrlp Python sunucusuna gnderilmiřtir. Fotoğrafin Fake veya Real olduėu olduėunun bilgisi Python sunucusundan gelmektedir. Fake fotoğraflar iin gelen “Photos detected as FAKE”, Real fotoğraf iin gelen “Photos detected as REAL” bilgisi ekranda Toast mesajı olarak gsterilmektedir. Aynı zamanda seilen fotoğraf oluřturulan imageview de ekranda gsterilmektedir.



řekil 16 Fake Fotoğrağ ıktısı



řekil 17 Real Fotoğrağ ıktısı

## KAYNAKÇA

- [1] R. R. B. R. W. B. Ali Khodabakhsh, «Fake Face Detection Methods: Can They Be Generalized?,» Department of Information Security and Communication Technology Norwegian University of Science and Technology, Gjøvik, Norway, 2018.
- [2] X. Q. J. J. H. S. T. Zhengzhe Liu, «Real or Fake: An Empirical Study and Improved Model for Fake Face Detection,» %1 içinde *ICLR 2020*, Addis Ababa, Ethiopia, 2019.
- [3] «<https://www.webtekno.com/adobe-photoshop-about-face-h79362.html>,» [Çevrimiçi]. [Erişildi: 11 12 2019].
- [4] B. D. H. H. B. Abdülkadir ŞEKER, «Derin Öğrenme Yöntemleri ve Uygulamaları Hakkında Bir İnceleme,» *Gazi Mühendislik Bilimler Dergisi*, 2017.
- [5] N. Y. Şimşek, «Derin Öğrenme Nedir ve Nasıl Çalışır?,» 24 01 2019. [Çevrimiçi]. Available: <https://medium.com/@nyilmazsimsek/derin-%C3%B6%C4%9Frenme-deep-learning-nedir-ve-nas%C4%B1l-%C3%A7al%C4%B1%C5%9F%C4%B1r-2d7f5850782>. [Erişildi: 23 10 2019].
- [6] «<https://deepinsightr.com/>,» 05 02 2018. [Çevrimiçi]. Available: <https://deepinsightr.com/2018/02/05/supervised-unsupervised-machine-learning/>. [Erişildi: 24 10 2019].
- [7] «<https://opencv.org/about/>,» [Çevrimiçi]. [Erişildi: 11 12 2019].
- [8] İ. H. C. Gül Gündüz, «Derin Öğrenme Algoritmalarını Kullanarak Görüntüden Cinsiyet,» *SAKARYA UNIVERSITY JOURNAL OF COMPUTER AND INFORMATION SCIENCES*, 2019.
- [9] S. A. Afshine Amidi. [Çevrimiçi]. Available: <https://stanford.edu/~shervine/l/tr/teaching/cs-230/cheatsheet-convolutional-neural-networks>. [Erişildi: 25 10 2019].
- [10] C. T. B. Gazel SER\*, «Derin Sinir Ağları ile En İyi Modelin Belirlenmesi,» 2019.

- [11] J. L. B. Diederik P. Kingma, «ADAM: A METHOD FOR STOCHASTIC OPTIMIZATION,» 2015.
- [12] <https://keras.io/>. [Çevrimiçi]. [Erişildi: 01 12 2019].
- [13] <https://pypi.org/project/Theano/>. [Çevrimiçi]. [Erişildi: 01 12 2019].
- [14] <https://www.tensorflow.org/about>. [Çevrimiçi]. [Erişildi: 01 12 2019].
- [15] [Çevrimiçi]. Available: <https://www.python.tc/python-nedir/>,. [Erişildi: 1 11 2019].
- [16] <https://gelecegiyazanlar.turkcell.com.tr/konu/android/egitim/android-201/android-studioyu-taniyalim>. [Çevrimiçi]. [Erişildi: 06 12 2019].
- [17] <https://developer.android.com/ml>. [Çevrimiçi]. [Erişildi: 06 12 2019].