

**Mühendislik ve Mimarlık Fakültesi**

**Bilgisayar Mühendisliği Bölüm**

**İleri Mobil**

**Dersi Proje Formu**

|  |
| --- |
| **PROJE BAŞLIĞI** |
| **İnsanların ruh hallerine anlama ve buna göre yorum yapma.**  **(Psychology Detector)** |

|  |  |
| --- | --- |
| **ÖĞRENCİ BİLGİLERİ** | |
| **Öğrenci Numara** | 19010011054 |
| **Öğrenci Ad Soyad** | RABİYA KURAN |

|  |
| --- |
| **Danışman Ad Soyad:** Dr. Hasan Ali AKYÜREK |

İçindekiler

[PROJENİN AMACI 3](#_Toc106115548)

[PROJENİN YÖNETİMİ 3](#_Toc106115549)

[PROJENİN YAPIM AŞAMALARI 3](#_Toc106115550)

[PROJEDE KULLANILAN YAZILIMLAR VE SİTELER 6](#_Toc106115551)

[YOUTUBE ANLATIM LİNKİM 6](#_Toc106115552)

[KAYNAKLAR 7](#_Toc106115553)

# PROJENİN AMACI

İnsanların ruh hallerine anlama ve buna göre gerekli yorumları yapma ve insanlara istedikleri zaman istedikleri yerde yardımcı olmak amaçlanmaktadır. Kullanıcı uygulamayı açtığında telefonun kamerası ile veya galeriye kaydettiği fotoğrafı seçip kolay bir şekilde uygulamaya dahil olabilir. İstenilen fotoğraf seçildikten saniyeler sonra ekrana duygunun sınıf başlığı ve ilgili yazı ekranda gözükür. Kullanıcının bu yazıyı okuyarak iyi bir yönlendirme yapmak amaçlanmaktadır.

# PROJENİN YÖNETİMİ

Proje kapsamında belirlenen uygulamamın içinde kişinin görüntüsü alınacak, alınan görüntü tflite modelinde işlenerek modelde mod tespiti yapılacak ve sonrasında kişinin duygu durum bilgisi modelden alınacaktır. Alınan veriler sonucunda kişinin durum bilgisine göre mutlu mu, üzgün mü, kaygılı mı vs. buna göre uygun mesaj bilgisi kullanıcıya verilecek. Kullanıcı eğer duygu durumunda yardım isterse ona göre ipuçları verilecek.

# PROJENİN YAPIM AŞAMALARI

İlk olarak teachable machine’yi kullanarak **tablo1**’de görülen karşılaştırmalara göre toplam 9904 fotoğraf kullanarak makinemi eğittim. Daha sonra eğitme kısmını tamamladıktan sonra indirme işlemini gerçekleştirdim. İndirdiğim .rar dosyasının içinde ‘model\_unquant.tflite’ ve ‘labels.txt’ olmak üzere 2 dosyam oluştu. Labels dosyamda ayırdığım başlık bilgileri bulunmakta.

**Tablo1**.yüz ifadeleri

|  |  |  |
| --- | --- | --- |
| 1-) Sad | 1657 adet fotoğraf | Üzgün bir yüz ifadesi varsa, |
| 2-) Happy | 1486 adet fotoğraf | Mutlu bir yüz ifadesi varsa, |
| 3-) Worried | 1425 adet fotoğraf | Endişeli veya kaygılı bir yüz ifadesi varsa, |
| 4-) Surprised | 1200 adet fotoğraf | Şaşırmış bir yüz ifadesi varsa, |
| 5-) Disgust | 1389 adet fotoğraf | Tiksinen bir yüz ifadesi varsa, |
| 6-) Angry | 1465 adet fotoğraf | Öfkeli bir yüz ifadesi varsa, |
| 7-) Scared | 1282 adet fotoğraf | Korkmuş bir yüz ifadesi varsa, |

Eğittiğim modelimi Visual Studio Code uygulaması içinde bulunan uygulamamın assets klasörünün içerisine gömdüm. TensorFlow uygulamasının çalışabilmesi için ilk önce pubspec.yaml dosyasına ‘tflite:’ paketini ve assets(varlıklar) kısmına modelinin ve labellerin yollarını ayrı ayrı göstererek ekledim. İkinci olarak android/app/build.gradle yoluna **şekil1**’deki kodu ekledim.

    aaptOptions {

        noCompress 'tflite'

        noCompress 'lite'

    }

**Şekil1**

Tflite işlemleri bittikten sonra fotoğraf çekebilmemize veya galerimizden var olan bir fotoğraf alabilmemize yarayan image\_picker pluginini pubspec.yaml doyasına ‘image\_picker:^0.6.4‘ versiyonunu ekledim. Bu şekilde programın çalışması için gerekli her şey eklenmiş oldu.

TensorFlow.dart dosyama gerekli paketleri **Şekil2**’de görüldüğü gibi ekledim. Firebase veri tabanından gerekli durumlar halinde veri çekeceğim için fire store paketini ekledim.

import 'package:cloud\_firestore/cloud\_firestore.dart';

import 'package:donemuygulamam/servisler/yetkilendirmeservisi.dart';

import 'package:flutter/material.dart';

import 'package:image\_picker/image\_picker.dart';

import 'package:tflite/tflite.dart';

Şekil2

Fotoğrafları kameradan çekmek için ayrı, galeriden çekmek için ayrı metotlar yazdım. Veri tabanından verileri çekmek için iki metot kullandım. **Şekil3.a**’da görüldüğü gibi galeriden fotoğraf seçilmesi için ve kameradan fotoğraf çekilmesi için ayrı butonlar oluşturdum.

**Şekil3**.a **Şekil3**.b

Fotoğraf çekildikten sonra ise yazıların gelmesi için fotoğrafın altına ayrı bir container oluşturdum. Bu şekilde tasarımımı **Şekil3.b’**de görüldüğü gibi ayarladım.

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

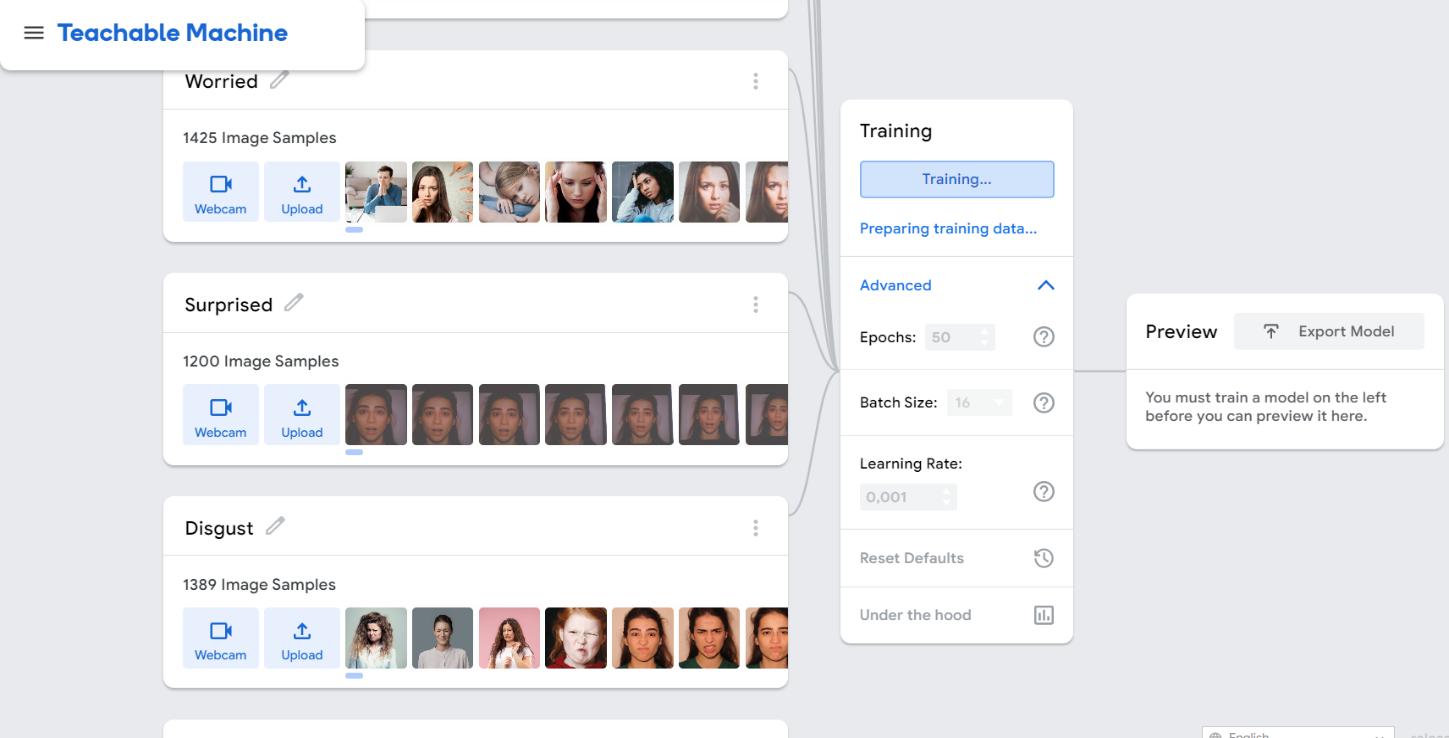
**Şekil4**.endişe durumu **Şekil5.**öfke durumu

**Şekil4**’te ve **şekil5**’te görüldüğü gibi fotoğrafın durumuna göre hangi duruma karşılık geliyorsa veri tabanından o durumun için yazılan yazı çekilip kullanıcıya fotoğrafın alt kısmında gösteriliyor.

Ektradan gerekli durumlarda kullanıcı uygulamanın giriş kısmında bulunan hastalık testlerine yönlendiriliyor. Bu şekilde kullanıcıya kolaylıklar sağladım.

# PROJEDE KULLANILAN YAZILIMLAR VE SİTELER

Makinemi eğitebilmek için **Şekil5**’te görülen Teachable Machine Sitesini kullandım. Burada 9904 fotoğraf kullanarak makinemi eğittim.



Şekil5

Visual studio code ara yüzüyle dart dilini kullanarak flutterın esnekliğinden faydalanarak uygulamamı gerçekleştirdim.

Fotoğraflar ve diğer bilgiler için pixabay, kaggle, pexels, evrim ağacı sitelerinden faydalandım.

Bilmediğim ve eklemem gereken bilgiler için flutter.dev sitesinde yararlandım.

# YOUTUBE ANLATIM LİNKİM

<https://youtu.be/fd9d6C-88VI>

# KAYNAKLAR

<https://evrimagaci.org/duygular-ve-yuz-ifadeleri-yuzumuz-duygularimizi-nasil-ifade-eder-7555>

<https://teachablemachine.withgoogle.com/train/image>

<https://flutter.dev/>

<https://www.kaggle.com/>

<https://www.pexels.com/tr-tr/>

<https://pixabay.com/tr/>