

AKILLI TELEFONLAR İLE NESNE TESPİTİ

Doğukan Kertlez, Kadir Yüksel

170202024, 170202059

Bilgisayar Mühendisliği

Kocaeli Üniversitesi

170202024@kocaeli.edu.tr, 170202059@kocaeli.edu.tr

Özet

Proje’de ,kullanıcı tarafından seçilen veya çekilen fotoğrafları buluta gönderip gerekli nesne tanıma işlemlerini yaptıktan sonra kullanıcıya yani uygulamaya geri döndüren bir yapı kurulmuştur .Bu yapıda react native için expo frameworku kullanılmıştır.

1.Giriş

Projeyi tamamlarken gerekli olan bütün araçlar iletişime geçerek elde etmek istediğimiz sonuç için senkronize biçimde çalışarak uygulamayı tamamlar.

Uygulamamız ilk olarak kullanıcıdan göndermek istediği fotoğrafı seçmesi yada bu fotoğrafı o anda kamerayla çekip kaydetmesi için seçenekler sunar.kullanıcı bu seçeneklere göre galerisinden bir fotoğrafı seçer yada kamerası ile yeni bir fotoğraf çeker ve bu fotoğrafı artık işlemler yapılması üzere gönderilir.

Fotoğraf gönderildiği yerde istenilen özellikleri elde etmek için işlemlere tabi tutulur ve bu işlemler sonucunda gönderdiğimiz fotoğraftaki nesneleri tespit edip bu nesnelerin kareler içine alması sağlanır.daha sonra bu nesnelerin adeti ve her bir nesne için ayrı ayrı kare içine alma ve ne olduğu bilgisi fotoğrafa yazılır .elde edilen fotoğraf daha sonra uygulamaya geri dönmek üzere kaydedilir.

Fotoğraf bu işlemlerden sonra uygulamaya geri çekilir ve kullanıcıya gösterilir.

2.Temel Bilgiler

Projemiz Visual studio code tümleşik geliştirme ortamında , React native –expo kullanılarak gerçekleştirilmiştir.

Bulut platformu olarak Amazon web services kullanılmıştır.

Ayrıca postman ve termius uygulamaları da proje de kullanılmıştır.

3.Tasarım

Projemiz en başta tasarlanırken seçilen çevreler kritik önem taşımaktaydı çünkü uyumu ve iletişimi sağlamak asıl meselelerden biriydi.

Bunun sonucunda proje’de react native expo ,aws,termius ,postman kullanıldı .

Kullanıcının gördüğü kısım olan uygulama arayüzümüz, kullanılmak isteyen fotoğrafın çekildiği veya seçildiği kısımdır. Kullanıcıya bu iki seçenek sunulur ,bu doğrultuda seçilen fotoğraf işlemlere tutulmak üzere alınır .kullanıcı sadece bu seçimi yapar ve işlemler sonucunda oluşturulan fotoğrafını görüntüler.

Kullanıcının seçiminden sonra uygulama aldığı bilgilere göre uzaktaki sunucumuza fotoğrafı gönderir ve fotoğraf tensorflow kütüphanesi sayesinde javascript ile nesne tanıma işlemini yapar .

Bu işlem sonucunda fotoğraf’daki nesneler kare içine alınır ve ne olduğu bilgisi eklenir.en son kaç adet nesne tanındıysa yazılır ve fotoğraf tekrar kaydedilir.

Kaydedilen fotoğraf uygulamaya gelir ve kullanıcıya gösterilir .bu işlemler tersium uygulamasından da takip edilebilir.

4. Kazanımlar

- Mobil uygulama gerçekleştirmede gerekli olan çevre birimlerinin iletişimi konusunda ilerleme kaydedildi.
- Mobil uygulamanın yayınlanması konusunda deneyimler elde edildi.
- Nesne tanıma algoritmalarında bilgi elde edildi.
- Fotoğraf işleme konusunda bilgi elde edildi.

5. Kullanılan Bazı Fonksiyonlar

- **pickImage()**: galeriden resim seçmek için kullanılır .resmi seçtikten sonra gerekli verileri set eder.
- **takePicture()**:kameradan resim çeker.
- **getCurDate()**: resim için zaman bilgisini alır.
- **uploadImage()**: servera resim yüklemek için kullanılır.resmi bilgileriyle birlikte yükler.
- **savePicture ()**: yakalanan fotoyu kaydetmek için kullanılır daha sonra upload eder .
- **styles ()**: uygulama dizaynı ile ilgili değerleri içerir.
- **useEffect ()**: galeri iznini ayarlar .

6. Kaynakça

[1][https://stackoverflow.com/\(genel\)](https://stackoverflow.com/(genel))

[3]<https://emineakturblog.wordpress.com/2020/02/20/postman-nedir/>

[3]<https://devnot.com/2019/tensorflow-nedir-nasil-kullanilir/>

[4]<https://github.com/tensorflow/tfjs-models/tree/master/coco-ssd>

[5]<https://www.npmjs.com/package/@tensorflow-models/coco-ssd>

[6]<https://docs.termius.com/>

[7]<https://learning.postman.com/docs/publishing-your-api/documenting-your-api/>

[8] <https://docs.aws.amazon.com/>

[9]<https://mreorhan.medium.com/expo-ile-mobil-uygulama-geli%C5%9Firme-1e5ccc2ebef7>

7. Ekran görüntüleri



Object Detection Server /home/ubuntu/objectDetectionServer			
Filter		Actions	
Name	Date Modified	Size	Kind
..			
node_modules	1/3/2021, 8:06 PM	--	folder
Yeni klasör	1/3/2021, 7:15 PM	--	folder
out.png	1/5/2021, 1:24 AM	930.36 kB	png
package-lock.json	1/3/2021, 7:15 PM	84.56 kB	json
index.js	1/4/2021, 5:32 PM	9.47 kB	js
package.json	1/3/2021, 7:15 PM	565.00 B...	json

```
ubuntu@ip-172-31-89-162:~/objectDetectionServer$ sudo npm run dev
> objectdetectionserver@1.0.0 dev /home/ubuntu/objectDetectionServer
> nodemon index

[nodemon] 2.0.6
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting 'node index index.js'

=====
Hi there 🍌 Looks like you are running TensorFlow.js in Node.js. To speed things up dramatically, install our node backend, which binds to TensorFlow C++, by running
$ npm i @tensorflow/tfjs-node, or npm i @tensorflow/tfjs-node-gpu if you have CUDA. Then call require('@tensorflow/tfjs-node'); (-gpu suffix for CUDA) at the start
of your program. Visit https://github.com/tensorflow/tfjs-node for more details.
=====
Coco-ssd loaded successfully
```

8. Tasarım Gösterimi

