

AKILLI TELEFONLAR İLE NESNE TESPİTİ PROJESİ

Müberra ÇELİK – Taha Batuhan TÜRK
180202102 - 180202007

Bilgisayar Mühendisliği Bölümü
Kocaeli Üniversitesi
180202102@kocaeli.edu.tr - 180202007@kocaeli.edu.tr

Özet-Proje , Bulut sistemleri ve görüntü işleme algoritmaları kullanılarak gerçek zamanlı nesne tespiti yapmayı amaçlayan, React Native teknolojisi ile yazılmış bir mobil uygulamadır.

Anahtar Kelimeler : Bulut Bilişim Teknolojileri , Görüntü İşleme, Veritabanı, Google Cloud, TensorFlow, API, Python, JavaScript, React Native .

I. GİRİŞ

Android ve IOS cihazlarda kullanılabilen uygulamamızda, kullanıcının çektiği fotoğraf ya da galeriden seçeceği herhangi bir fotoğraf Google Cloud ortamındaki sunucumuza request olarak gönderilir. Sunucuda kullandığımız görüntü işleme algoritması fotoğraftaki objeler ve obje sayısı tespit edilip tekrar response olarak işlenmiş görüntü döndürülür. Gelen işlenmiş görüntü telefonda kullanıcıya gösterilir.

II. YÖNTEM

Mobil uygulama Android Studio ortamında JavaScript programlama dili kullanan cross-platform React-Native ile yazılmıştır. Sunucu tarafında Google Cloud kullanılmış olup görüntü işleme programı sunucuya gömülmüştür. Görüntü işleme programı Python ile yazılmış olup, Google tarafından geliştirilen TensorFlow kütüphanesi kullanılmıştır. Kütüphane YOLOv3 yapay sinir ağı ile çalışmaktadır. YOLO (You Only Look Once) görüntüdeki nesneleri konumlarıyla beraber tek seferde tespit edebilen bir yapay sinir ağıdır. Görüntüyü hücrelere bölerek her bir hücrede nesneler için confidence değerlerini ve bounding boxları tahmin etmeye çalışan çıktı vektörleri üretir.

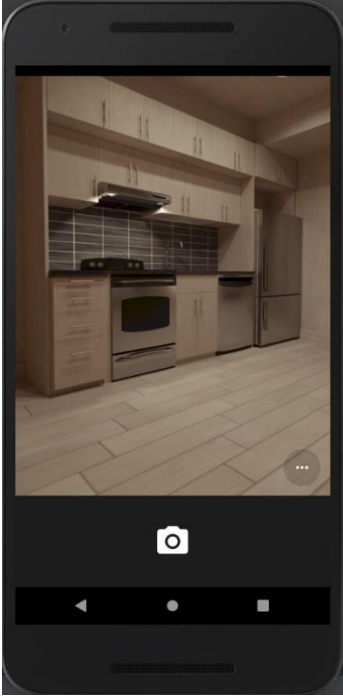
III. DENEYSEL SONUÇLAR

Uygulama açıldığında kullanıcının karşısına, kameradan fotoğraf çekebileceği veya galeriden bir fotoğraf seçebileceği iki seçenek çıkar. (Şekil.1)



Şekil.1 Kullanıcıyı karşılayan seçim ekranı

Kamera butonuna tıklandıktan sonra kullanıcının kameradan fotoğraf çekebileceği ekran Şekil.2 de gösterilir.



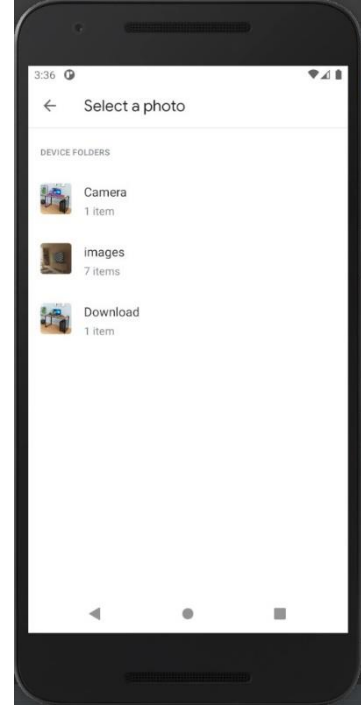
Şekil.2 Kullanıcının fotoğraf çekebileceği kamera ekranı

Kullanıcı fotoğraf çektikten sonra tekrar fotoğraf çekmek için geri tuşuyla kamera ekranına geri dönebilir, ana ekrana dönmek isterse çarpı işaretine tıklayarak dönebilir. Ok tuşuna basıldığında ise resmi sunucuya request olarak gönderir. (Şekil.3)



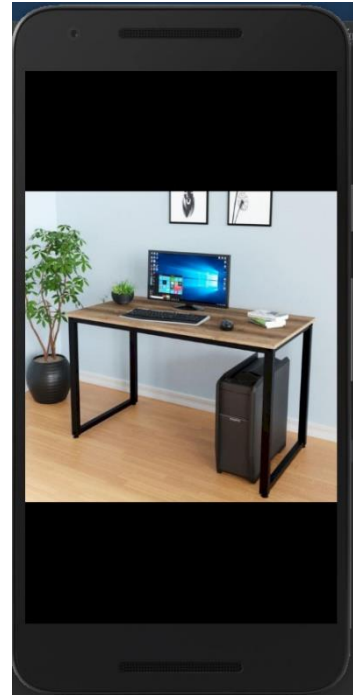
Şekil.3 Kullanıcının fotoğraf çektikten sonra karşısına gelen ekran

Kullanıcı isterse Galeri butonuna tıklayarak galerisindeki fotoğrafları görüntüleyebilir ve galeriden bir fotoğraf seçebilir. (Şekil.4)



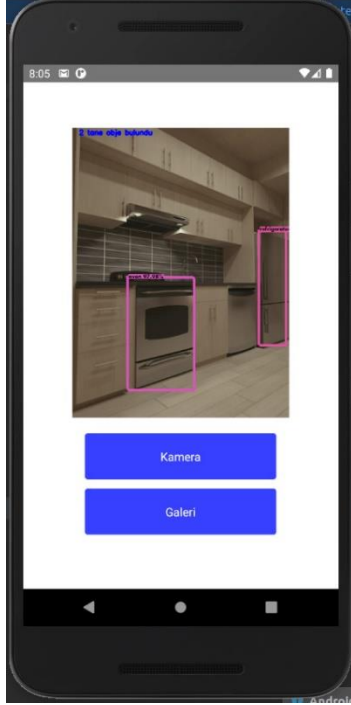
Şekil.4 Kullanıcının galeri ekranı

Kullanıcı kendi galerisinden seçtiği fotoğrafı görüntüleyebilir. (Şekil.5)

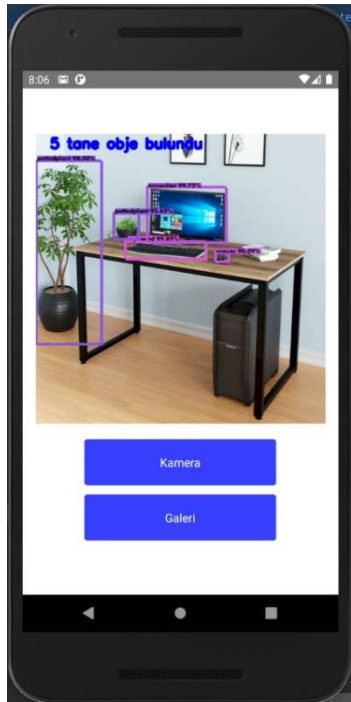


Şekil.5 Seçilen fotoğrafın görüntülenmesi

Kullanıcının kameradan çektiği (Şekil.6.1) veya galeriden seçtiği fotoğraf (Şekil.6.2), bulut ortamında görüntü işleme yapıldıktan sonra fotoğraftaki nesneler ve nesnelerin sayısı kullanıcıya gösterilir.



Şekil.6.1 Kameradan çekilen fotoğrafın işlenmiş görüntüsü



Şekil.6.2 Galeriden seçilen fotoğrafın işlenmiş görüntüsü

IV. AKIŞ DİYAGRAMI



V. SONUÇ

Projeyi geliştirirken mobil uygulama için cross-platform desteği sağlayan React-Native' in JavaScript tabanlı olmasından dolayı öncelikle JavaScript öğrendik. Sürüm farklarından dolayı React-Native ve kütüphaneler arasında birçok uyumsuzluk yaşadık. Bu uyumsuzlukları çözmek için github, StackOverflow gibi kaynaklardan yararlandık.

Sunucu tarafında Google, Amazon, IBM, Firebase gibi Cloud ortamlarını araştırdık. Bizim için en uygun ve en düşük maliyetli çözüm olan Google Cloud kullanmaya karar verdik.

Görüntü işleme tarafında klasik olan OPENCV kütüphanesini tek başına yeterli görmediğimiz için, derin öğrenme ve yapay sinir ağları etkileşimi sunan Tensorflow kütüphanesinde YOLOV3 ve COCO verisetini kullanan hazır bir Python scripti kullandık.

Projede en çok zorlandığımız kısım tüm işlemleri birbiri ile etkileşimli ve senkronize etmek oldu. Bu projede kendi API mizi oluşturmayı, cloud üzerinde işlemler yapmayı ve mobil platform alanında JavaScript tabanlı React-Native dilini öğrendik.

VI. KAYNAKÇA

- [1] Github (n.d.). Retrieved from
<https://github.com/tbturk/Object-Detection-API>
<https://github.com/joltup/m-fetch-blob>
- [2] HACI VELİ KARAKUŞ (n.d.). Retrieved from
<https://medium.com/@karakus.haciveli/darknet-yolov3-h%C4%B1zl%C4%B1-bak%C4%B1%C5%9F-ddc9cd5582ea>
- [3] NAPPD (n.d.). Retrieved from
<https://enappd.com/blog/pick-images-from-camera-gallery-in-react-native-app/78/>