

**KOCAELİ ÜNİVERSİTESİ**  
**BİLGİSAYAR MÜHENDİSLİĞİ**  
**AKILLI TELEFONLAR İLE NESNE TESPİTİ**

Süleyman AYAZ  
200201118  
suleyman.ayaz.edu@gmail.com

## GİRİŞ

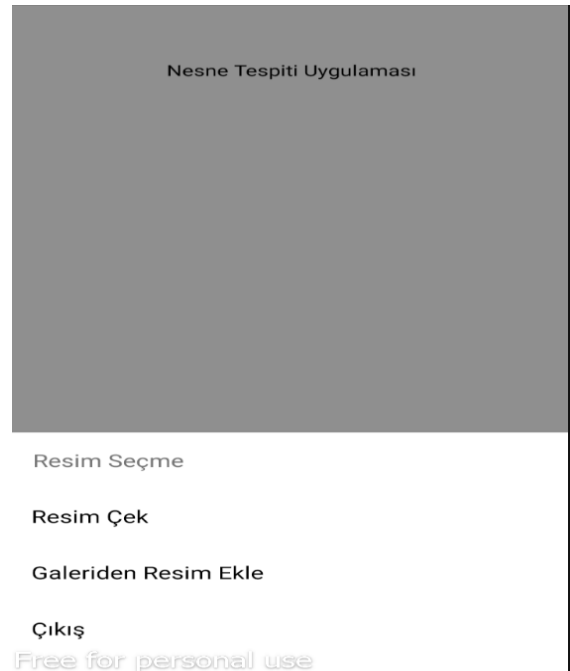
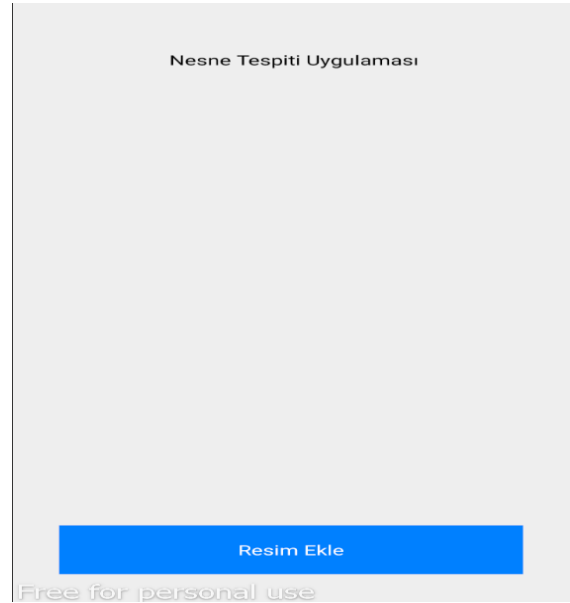
Bu projede React Native programlama dilini kullanarak mobil uygulama yapmamız beklenmektedir. Bu uygulama hem telefon kamerasından anlık görüntü alınacaktır (resim çekilecektir) hem de telefon galerisinden de resim seçilebilecektir. Elde edilen görüntüler bulut ortamındaki cloud ortama upload edilecektir. Bulut üzerinde gerekli görüntü işleme tooları kullanılarak resim üzerindeki nesneler algılanıp nesneler kare içerisine alınacak ve uygulamaya tekrar geri gönderilecektir. Mobil uygulama hem emülatör hem de telefondan gerçekleştirilmelidir.

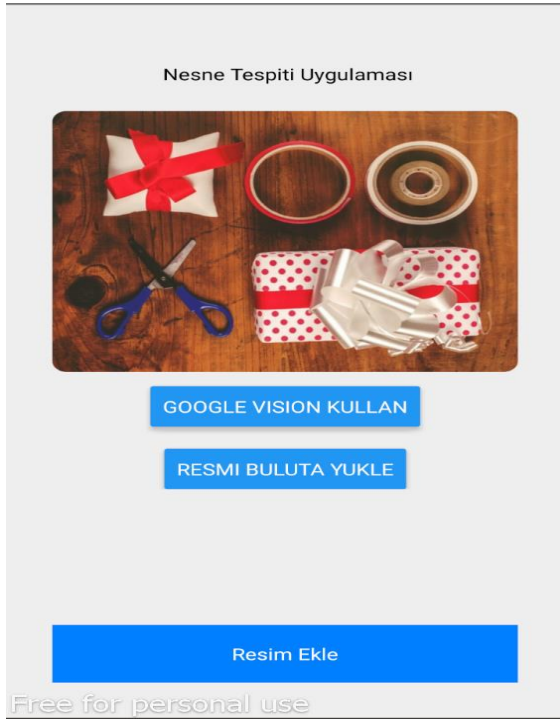
## YÖNTEM

Bu mobil uygulamada cloud ortamına yüklenicek resim seçilirken yada çekilirken react-native-image-crop-picker modülünden, react-native-sv modülünden yararlanılmıştır. Firebase cloud ortamına gerekli api bağlantısı yapıldıktan sonra bulut ortamına atılır. Google Vision ile api bağlantısını kurarak resmin base64 string halini apiyle beraber yolluyoruz ve sonuç olarak bize response.json yolluyor. Bu dosyada istediğimiz görüntü verilerin işlenmiş hali dönüyor. Bu

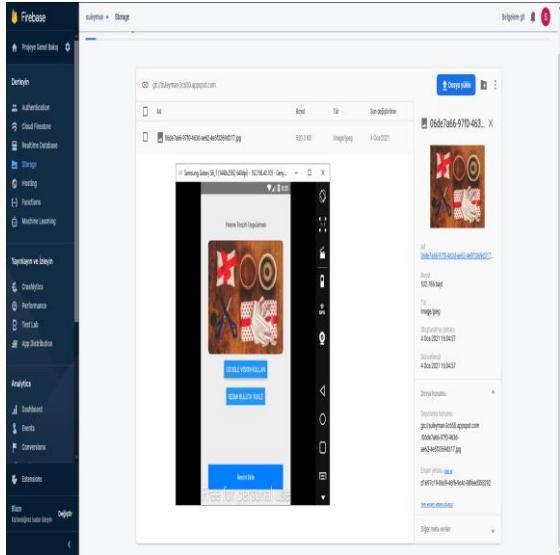
## DENEYSEL SONUÇLAR

React Native platformunu kullanarak mobil resim çekme ve resimlerden resim seçme işlemi yapılmıştır.





Resim buluta yükle butonuna tıklandıktan sonra resim FireBase cloud ortamına yüklenir.



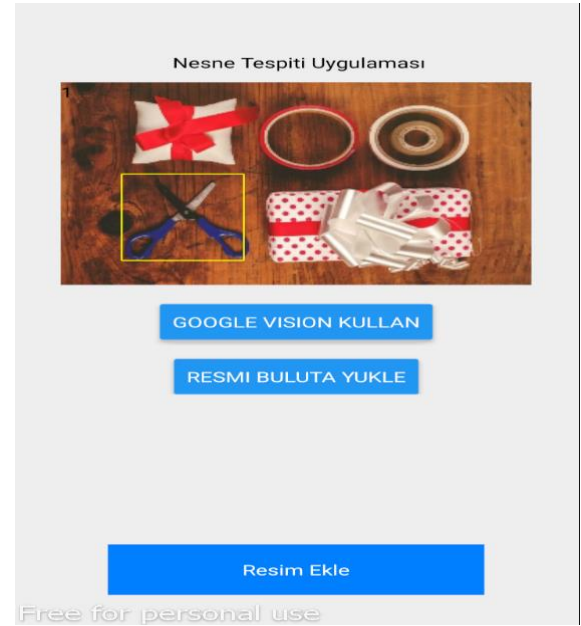
Google vision apisin kullan butonuna tıkladığımızda google vision apisı üzerinden google visiona bağlanıp bulut ortamındaki resim içeriği hakkında bilgileri response.json dosyasına yazıp bize yollar bu dosya içeriğini konsola yazdırılır.

```
To reload the app press "r"
To open developer menu press "d"

[Tue Jan 05 2021 21:13:50.142] BUNDLE ./index.js

[Tue Jan 05 2021 21:13:52.656] LOG Running "proj" with {"rootTag":1}
[Tue Jan 05 2021 21:14:33.445] LOG file:///storage/emulated/0/Pictures/62e90fa6-3b9b-4e25-b0d1-b06254565c25.jpg
[Tue Jan 05 2021 21:14:33.288] LOG Name: Scissors
[Tue Jan 05 2021 21:14:33.300] LOG Confidence: 0.926649
[Tue Jan 05 2021 21:14:33.300] LOG x1 : 38.870265
[Tue Jan 05 2021 21:14:33.300] LOG y1 : 90.595168
[Tue Jan 05 2021 21:14:33.301] LOG x2 : 116.66751000000001
[Tue Jan 05 2021 21:14:33.301] LOG y2 : 90.595168
[Tue Jan 05 2021 21:14:33.301] LOG x3 : 116.66751000000001
[Tue Jan 05 2021 21:14:33.302] LOG y3 : 175.934732
[Tue Jan 05 2021 21:14:33.302] LOG x4 : 38.870265
[Tue Jan 05 2021 21:14:33.302] LOG y4 : 175.934732
```

Google vision apisini kullan butonuna tekrar tıkladığımızda google vision apisı üzerinden gelen bilgiler doğrultusunda resimdeki nesneler kare içine alınır ve resimdeki nesne sayısı resime basılır.



## GOOGLE VISION

Google Cloud tarafından sunulan Vision API, REST ve RPC API'leri aracılığıyla önceden eğitilmiş güçlü makine öğrenimi modelleri sunar. Görüntülere etiket atayarak önceden tanımlanmış milyonlarca kategoriye göre hızlıca sınıflandırma yapabilirsiniz. Ayrıca nesneleri ve yüzleri algılayabilir, basılı ve el yazısı metinleri okuyabilir, görüntü kataloğunuzda değerli meta veriler oluşturabilirsiniz.

Aşağıdaki Fonksiyonları barındırır:

- **Face Detection:** Konturlarla birlikte yüzleri ve yüz işaretlerini tespit etmek.
- **Landmark recognition:** Görüntüdeki işaretleri tanımlama.
- **Logo Detection:** Görüntüdeki logo tanımlama
- **Label Detection :** Bir görüntü için genelleştirilmiş etiketler sağlar. Her etiket için metinsel bir açıklama, güven puanı ve güncellik derecelendirmesi döndürür.
- **Text Detection:** Bir görüntü için optik karakter tanıma(OCR) metin tanıma ve makine kodlu metne dönüştürme.
- **Object Localization:** Tek bir görüntüde tanıma birden çok nesne için genel etiket ve sınırlayıcı kutu açıklamaları sağlar
- **Crop hint detection:** Kırpılın görüntü için bir sınırlayıcı poligon, bir güven puanı ve her istek için orjinal görüntüye göre bu belirgin bölgenin önemli bir bölümünü sağlar.
- **Web entities and pages:** Bir görüntüye bir dizi ile ilgili web içeriği sağlar

## FIREBASE ML\ML KİT

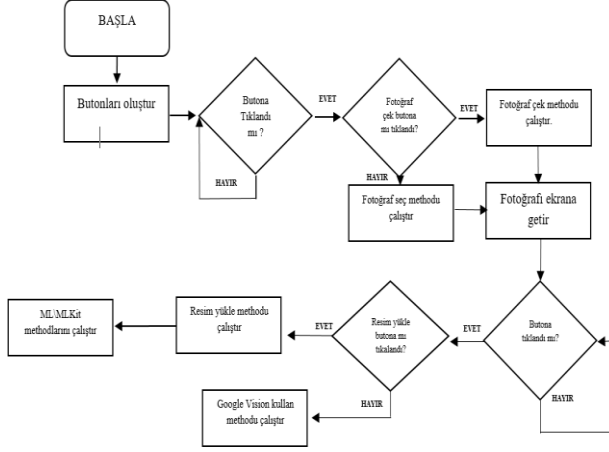
Firestore ML Kit, mobil geliştiricilerin uygulamalarına makine öğrenme yeteneklerini eklemelerini kolaylaştıran mobil bir SDK'dır. Aşağıdaki fonksiyonları barındırır:

- **Text Recognition:** Resimlerden metin tanımak ve çıkarmak.
- **Face Detection:** Konturlarla birlikte yüzleri ve yüz işaretlerini tespit etmek.
- **Object Detection and Tracking:** To detect, track and classify objects in camera and static images.
- **Image Labelling:** Kameradaki ve statik görüntüdeki nesneleri algılamak, izlemek ve sınıflandırmak.
- **Barcode scanning:** Barkodları taramak ve işlemek.
- **Landmark recognition:** Görüntüdeki işaretleri tanımlama.
- **Language ID:** Metnin dilini tespit etmek.
- **On-Device Translation:** Metnin bir dilden diğerine çevrilmesi.
- **Smart Reply:** Önceki mesajlara dayalı metinsel cevaplar üretilmesi.

Bunun dışında özel görüntü sınıflandırma makinesi öğrenme modellerini (.tflite modelleri) **AutoML** yardımıyla kullanabilirsiniz.

ML Kit, temel olarak mobil uygulamaya makine öğrenme yeteneklerini dahil etme ve kullanma karmaşıklığını minimuma indiren bir sistemdir.

## AKIŞ DİYAGRAMI



## SONUÇ

Bu projede React-Native programlama dilini kullanarak bir mobil uygulama geliştirilmiştir. Bu uygulama projenin isterleri olan seçilen yada çekilen resimi cloud ortamına upload etme işlevini yerine getirmektedir. Firebase ve Google cloud platformlarının sağladığı Google Vision uygulamalar mobil program tarafından api bağlantısı yaparak iletişim haline geçmiş olup yollanan resimin içeriği hakkında bilgileri elde edilmiştir. Projenin bir diğer isteri resim içerisindeki nesneleri kare içine alma ve nesne sayısını gösterme işlevleri Google vision apisi sonucunda dönen veriye göre işlem yapıлып resime işlenmiştir.

## PROJEDE KARŞILAŞILAN SORUNLAR

- Google vision apisin bağlanmasında react-native'nin desteklemediği fs modülü yüzünden çalışmaması
- Image-crop-picker modülünün grandle ve react native bazı sürümlerinden sürümünden dolayı çalışmaması
- Firebase ve Google Cloud platformlarının kredi kartı bilgilerinin girilmeden kullanılamaması

## ÇALIŞMA TAKVİMİ

- Kaynakların toplanması ve listelenmesi :6gün
- Projenin teorik kısmının yazılması :1 hafta
- Projenin nihai olarak yazılması :2 gün

## KULLANILAN YÖNTEMLER

- Proje React-native programlama diliyle yazılmıştır.

## KAYNAKÇA

- <https://cloud.google.com/vision>
- <https://cloud.google.com/vision/docs/labels>
- <https://cloud.google.com/vision/docs/detecting-landmarks>
- <https://cloud.google.com/vision/docs/object-localizer>
- <https://firebase.google.com/docs/ml-kit>
- <https://firebase.google.com/docs/ml/label-images>
- <https://firebase.google.com/docs/ml/android/label-images>
- <https://medium.com/hardwareand/firebase-ml-kit-ile-android-text-recognition-metin-tan%C4%B1ma-22c08f3aaf8b>
- <https://developers.google.com/ml-kit>
- <https://developers.google.com/ml-kit/vision/object-detection>
- <https://developers.google.com/ml-kit/vision/object-detection/android>
- <https://stackoverflow.com/questions/64022942/build-failing-after-installing-react-native-image-crop-picker>
- <https://github.com/react-native-svg/react-native-svg>