

# PROJE 3: Nesne Tanıma Uygulaması

Erdem Nayın 180202050

Bilgisayar Mühendisliği Bölümü

Kocaeli Üniversitesi

[erdem\\_nayin@hotmail.com](mailto:erdem_nayin@hotmail.com) – [180202050@kocaeli.edu.tr](mailto:180202050@kocaeli.edu.tr)

## 1-)Özet

Daha önceden kıstasları belirlenen uygulama geliştirildi. Uygulama geliştirilirken cross platform olarak çalışabilen bir mobil uygulama geliştirme, ilgili Cloud servisleriyle haberleşerek ve ilgili nesne tanıma servisleriyle haberleşerek uygulamaya gönderilen fotoğrafta bulunan nesnelerin tanınması ev ekranda yazdırılması, kodlama becerisini geliştirmek, bunun yanında geliştirilen uygulamanın dinamik bir şekilde kullanılmasını sağlamak amaçlandı.

## 2-)Giriş

Uygulamayı çalıştırdığınızda karşınıza açılış ekranı gelir. Bu ekranda en üstte uygulamanın ismi, altında ise iki adet buton bulunmaktadır.

Bu butonlar sırasıyla “Kameradan Fotoğraf Çek” ve “Galeriden Fotoğraf Seç” idir.

“Galeriden Fotoğraf Seç” butonuna tıkladığınızda programı ilk kez çalıştırıyorsanız uygulama sizden aygıtınızın depolama alanına erişim izni ister.

Bu izni vermezseniz program ilgili hata mesajını yazdırarak sizi AçılışEkranı’na geri yönlendirir. Eğer izin verirsiniz aygıtınızda kullandığınız Galeri

uygulaması açılarak buradan bir fotoğraf seçmeniz beklenir.

“Kameradan Fotoğraf Çek” butonuna tıkladığınızda programı ilk kez çalıştırıyorsanız uygulama sizden aygıtınızın depolama alanına erişim izni ve kameranıza erişip onu kullanma ister.

Bu izni vermezseniz program ilgili hata mesajını yazdırarak sizi AçılışEkranı’na geri yönlendirir. Eğer izin verirsiniz aygıtınızda kullandığınız kamera uygulaması açılarak buradan bir fotoğraf çekmeniz beklenir.

Eğer fotoğraf seçilmezse, fotoğraf çekilmezse veya hatalı işlemler olursa uygulama hata mesajı yazdırarak AçılışEkranı’na geri döner. Eğer fotoğraf seçerseniz veya yüklerseniz ilgili fonksiyonlar tetiklenerek fotoğrafın datası api’ye post edilir.

Daha sonra cloud ortamında ilgili nesne depolanır ve nesne tanıma işlemleri ilgili servisleri tarafından yapılır. Bu işlemler yapıldıktan sonra fotoğrafta bulunan nesnelerle ilgili tahminler ve fotoğrafın neresinde olduğunu belirten BoundingBox dataları uygulamaya geri döner.

Daha sonra fotoğrafın altında, fotoğrafın konumunu beliren uri text mesajının altında; fotoğrafta tanınan nesnelerle ilgili tahminler görünür hale gelir.

### 3-)Yöntem

Proje JavaScript programlama dili, React Native CLI, NodeJS, Android Studio, AWSGatewayService, AWSLambdaFunction, AWS Rekognition Service, AWS S3Bucket cloud geliştirme ortamları kullanılarak gerçekleştirildi.

Uygulama geliştirilirken react-native-image-picker gibi hazır kütüphanelerden ve fonksiyonlardan yararlanıldı. Bulunmayan fonksiyonlar ekibimiz tarafından yapıldı.

Proje yapılırken birçok problemle karşılaşıldı. Problemlerin çözümünde, aynı problem veya benzeri bir problemle karşılaşmış arkadaşlarımızla görüş alışverişi yapıldı, internette daha önce bu problemlerle karşılaşan insanların bu problemlerin üstesinden nasıl geldiği araştırıldı ve farklı bakış açıları kazanıldı. Bulunan birçok farklı çözümden problemlerimizin çözümlerine uygun olanları seçildi.

Projeye ilk olarak kullanıcı ara yüzünün nasıl olması gerektiği, örnek uygulamaları inceleyerek nasıl anlaşılabilir ve kolay kullanılabilir bir uygulama yapılabileceği düşünüldü.

Daha sonra bu kısım ilgili text ve view alanları yerleştirilerek, altlarına da kullanıcıların etkileşime girerek galeriden fotoğraf seçebileceği veya telefonun kamerasından fotoğraf yükleyebileceği butonlar konuldu.

Butonların, textlerin ve view nesnelerinin özellikleri ve konumlanma özellikleri belirlendi.

Daha sonra uygulamanın çalıştırıldığı depolama alanından fotoğraf alıp uygulama içine entegre etme işleminin yapımına başlandı.

Bu problemin çözümünde react-native-image-picker kütüphanesi import edildi. Bu kütüphanede bulunan fonksiyonlar kullanılarak hem kullanılan aygıtın depolama ünitesine erişerek daha önceden hazır bir şekilde bulunan fotoğrafların uygulamaya aktarılması hem de kullanıcının uygulamayı kullandığı esnada aygıtının kamerasına erişerek fotoğraf çekmesi ve akabinde uygulamaya bu fotoğrafın eklenmesi gerçekleştirildi.

Daha sonra alınan bu resmin base64string formatındaki datası alındı. Bu data türü hem fotoğrafı API'ye yollarken işimize yarar hem de fotoğraf işleme algoritmaları için gerekli olduğu için bu tür kullanıldı.

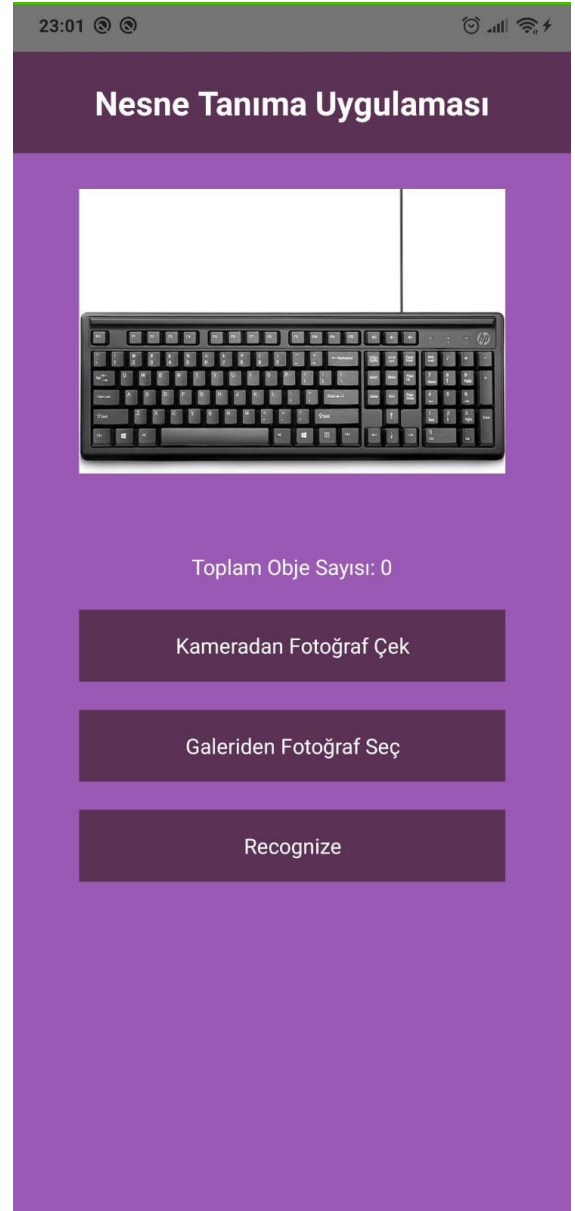
Daha sonra AmazonGateway ve Amazon Lambda Functions kullanılarak fotoğraf AmazonS3Bucket cloud ortamında depolandı. Daha sonra burada depolanan fotoğraf yine Amazon'un servisi olan Amazon Rekognition servisine gönderildi. Bu servis bizim için fotoğraftaki nesneleri tanımlama işlemini gerçekleştirerek fotoğrafta bulunan nesnelerin neler olabileceği hakkındaki tahminlerini ve servisin fotoğrafta algılayabildiği nesnelerin bounding box koordinat bilgileri yine bu Gateway'ler kullanılarak uygulamamıza geri döndürüldü.

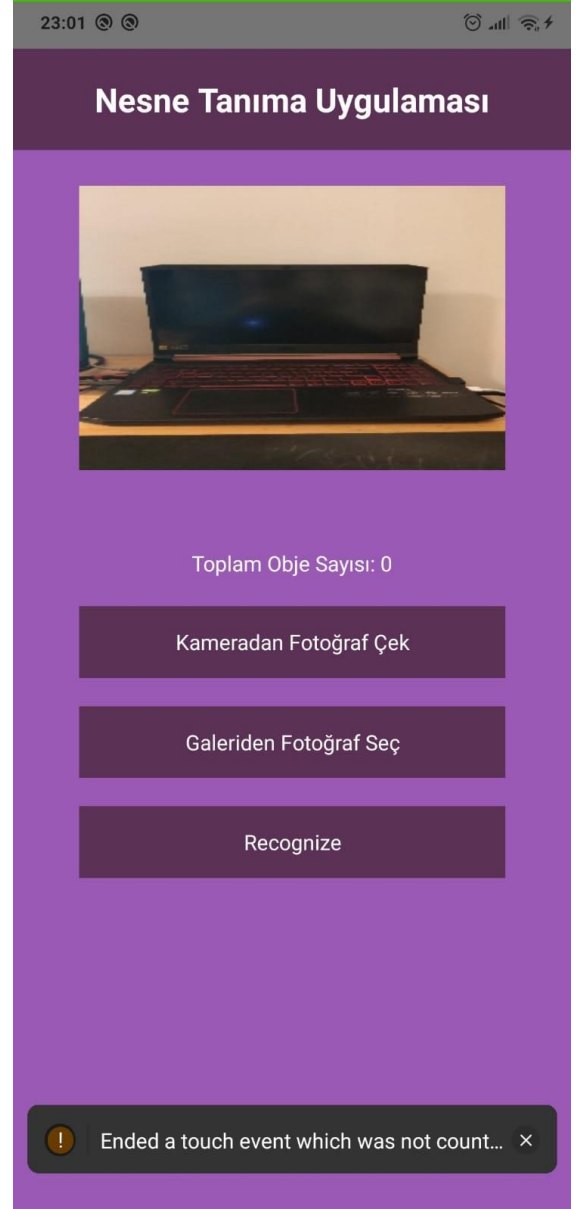
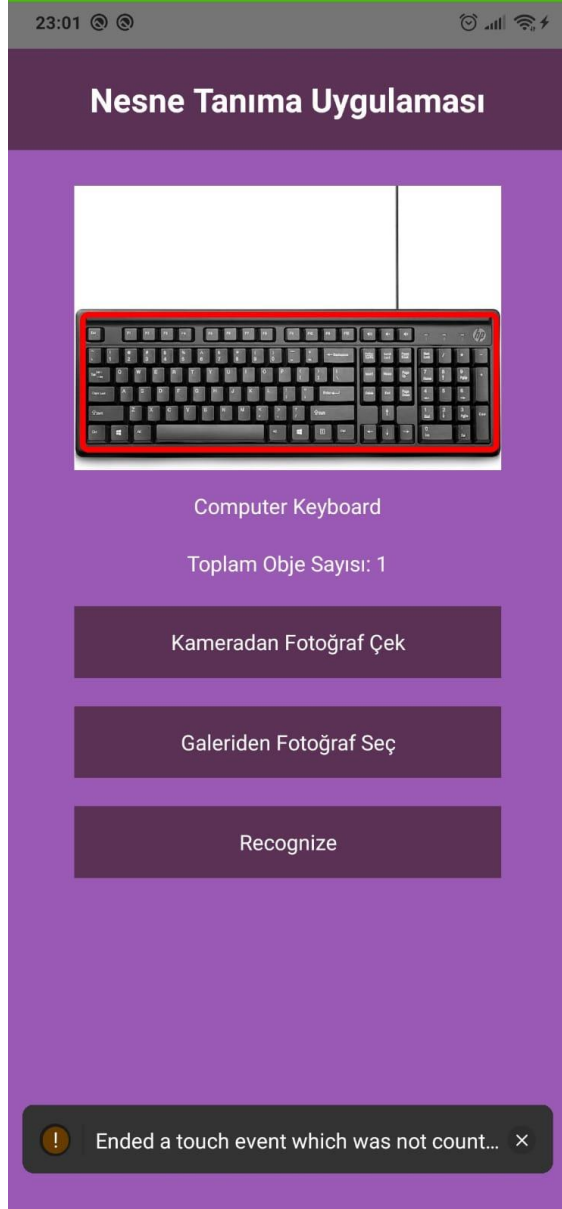
Daha sonra kullanıcının yüklediği fotoğrafın altında ilgili fotoğrafta bulunan nesnelerin ne olduğu ile ilgili tahminler burada kullanıcıya sunuldu.

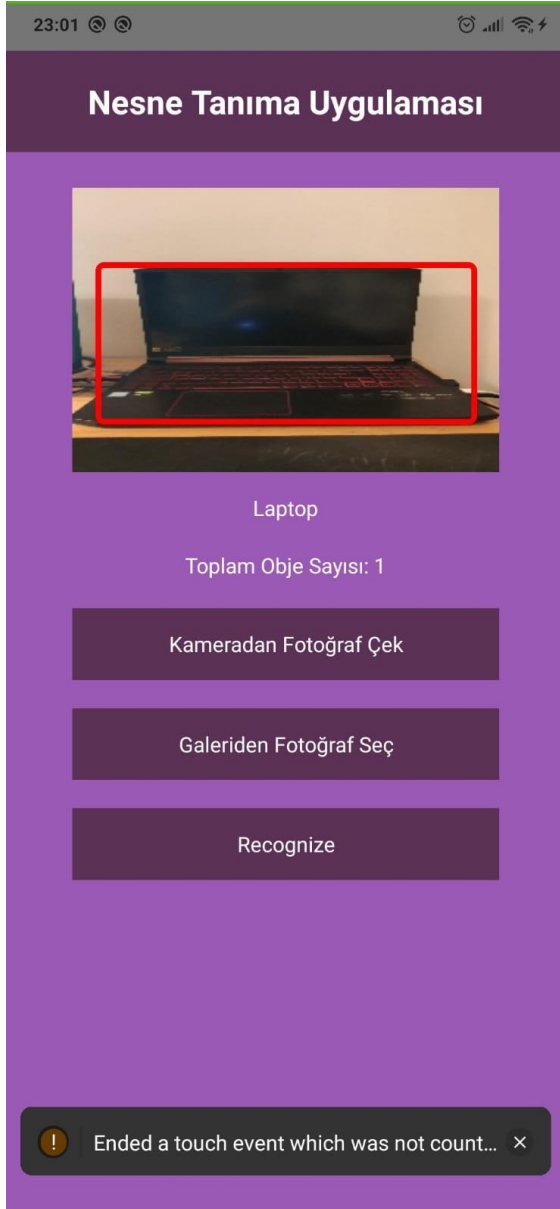
#### 4-)Sonuç

Sonuçta React Native, NodeJS, Android Studio kullanılarak cross platform çalışan bir uygulama gerçekleştirildi. Uygulamamız ile serverlar arasında API işlemleri gerçekleştirildi. Uygulamadan AmazonS3Bucket bulut servisine data yollandı. Amazon Rekognition servisi kullanılarak görüntü işleme yapıldı ve görüntüde bulunan nesnelerin tespiti yapıldı. Lambda Fonksiyonlarıyla haberleşme sağlandı. İlk kez mobil uygulama geliştirilmiş oldu. İlk kez serverlar arası iletişim gerçekleştirilmiş oldu.

#### 5-)Deneyisel Sonuçlar







## 6-)Yalancı Kod

- Başla
- Açılış ekranını göster.
- 
- Eğer Kameradan Fotoğraf Çek butonuna basıldıysa ilgili eventHandler'ı çalıştır.
- Kamera arayüzünü aç ve kullanıcının fotoğraf çekmesini bekle.
- Kullanıcı fotoğraf çektiyse ve işlem başarılıysa hata fırlatmadan geri dön.
- Kullanıcı fotoğraf ekranını açmak için izinleri vermedi, fotoğraf çekti fakat kaydetmedi,fotoğraf çekmediyse ilgili uyarıları ekrana yazarak Açılış Ekranına geri dön.
- Eğer Galeriden Fotoğraf Seç butonuna tıklandıysa ilgili eventHandler'ı çalıştır.
- Galeri arayüzünü aç ve kullanıcının galeriden fotoğraf seçmesini bekle.
- Kullanıcı fotoğraf seçtiyse ve işlem başarılıysa hata fırlatmadan geri dön.
- Kullanıcı fotoğraf seçmek için galeriye ulaşma iznini vermediyse, fotoğraf seçti fakat vazgeçti ve yüklediysen ilgili uyarıları ekrana yazarak AçılışEkranına geri dön.
- Çekilen veya seçilen fotoğrafın base64string datasını API'ye gönder.
- Lambda Function kullanarak bundan sonraki işlemleri yap.
- Fotoğrafın datasını Amazon S3 Bucket serverında depola.
- Amazon S3 Bucket serverında depolanan fotoğrafın adresini Amazon Rekognition servisine gönder.
- İşlenmiş fotoğrafın datasını uygulamaya geri döndür.
- Geri dönen datada bulunan ilgili tahminleri ekranda yazdır.
-

## 7-)Kaynakça

- <https://medium.com/zenofai/object-detection-in-react-native-app-using-aws-rekognition-ebd39b1270c2>
- <https://medium.com/zenofai/serverless-web-application-architecture-using-react-with-amplify-part1-5b4d89f384f7>
- <https://reactnative.dev/docs/environment-setup>
- <https://reactnative.dev/docs/running-on-device>
- <https://aboutreact.com/example-of-image-picker-in-react-native/>
- <https://reactnative.dev/docs/network#handling-the-response>
- <https://stackoverflow.com/questions/56891033/facing-issue-failed-to-install-the-app-make-sure-you-have-the-android-develop>
- <https://stackoverflow.com/questions/17587751/gradle-execution-failed-for-task-processdebugmanifest>
- <https://stackoverflow.com/questions/5427195/how-to-change-the-minsdkversion-of-a-project>
- <https://github.com/react-native-image-picker/react-native-image-picker/issues/1165>
- <https://www.w3docs.com/snippets/javascript/how-to-convert-the-image-into-a-base64-string-using-javascript.html>
- <https://medium.com/@cherifmezdari/image-upload-base64-to-server-react-native-18488988bf49>
- <https://docs.aws.amazon.com/rekognition/latest/dg/images-displaying-bounding-boxes.html>
- <https://stackoverflow.com/questions/47830061/undefined-is-not-an-objectevaluating-imagepickermanager-showimagepicker>
- <https://www.positronx.io/react-native-pick-images-from-camera-gallery-example/>
- <https://developer.aliyun.com/mirror/npm/package/react-native-image-picker-enhancement>
- <https://stackoverflow.com/questions/44246110/android-error-while-using-exifinterface>
- <https://stackoverflow.com/questions/42064283/could-not-connect-to-react-native-development-server-on-android>
- <https://lifesaver.codes/answer/solved-undefined-imagepicker-showimagepicker>
- <https://github.com/react-native-image-picker/react-native-image-picker/issues/494>
- <https://stackoverflow.com/questions/19465049/changing-api-level-android-studio>
- <https://stackoverflow.com/questions/54698734/how-to-fix-read-property-showimagepicker-of-undefined-react-native-with-expo>
- <https://rollbar.com/blog/top-10-javascript-errors/#:~:text=related%20errors%20below.,2.,method%20on%20an%20undefined%20object.&text=This%20is%20essentially%20the%20sa>

me,uses%20a%20different%20e  
rror%20message.