

AKILLI TELEFONLAR İLE NESNE TESPİTİ

Cumali TOPRAK

Mühendislik Fakültesi, Bilgisayar Mühendisliği
Kocaeli Üniversitesi
Kocaeli, Türkiye
cumalitoprakk@gmail.com

Berkay Efe ÖZCAN

Mühendislik Fakültesi, Bilgisayar Mühendisliği
Kocaeli Üniversitesi
Kocaeli, Türkiye
berkayefeozen@gmail.com

Özetçe— Bu projede akıllı telefonlar kullanılarak nesne tespiti yapmak amaçlanmıştır. Bu kapsamda mobil programlama teknikleri, bulut veritabanı kullanımı, ve sunucuda uygulama barındırma gibi konularda bilgi sahibi olunması amaçlanmıştır. Projeyi özetlersek telefondan çekilen bir görüntünün veya telefonun galerisinden seçilen bir görüntünün react native uygulaması kullanılarak sunucuya gönderilip burada görüntü işleme uygulamasıyla nesnelerin tespiti yapılması ve verilerin burada barındırılması amaçlanmıştır. Son olarak işlenen görüntünün de telefona tekrar geri gönderilmesi amaçlanmıştır. Projemizde React native mobil uygulama programlama framework'u, amazon uzak sunucusu ve depolama alanı kullanılmıştır.

Anahtar Kelimeler—React native, Amazon, Görüntü İşleme, Nesne tespiti.

Abstract— In this project, it is aimed to detect objects using smart phones. In this context, it is aimed to have information about mobile programming techniques, cloud database usage, and application hosting on the server. If we summarize the project, it is aimed that an image taken from the phone or an image selected from the phone's gallery is sent to the server using the react native application, and the objects are detected with the image processing application and the data is stored here. Finally, it is aimed to send the processed image back to the phone. React native mobile application programming framework, amazon remote server and storage area were used in our project.

Keywords—React native, Amazon, Image Processing, Object Detection .

I. GİRİŞ

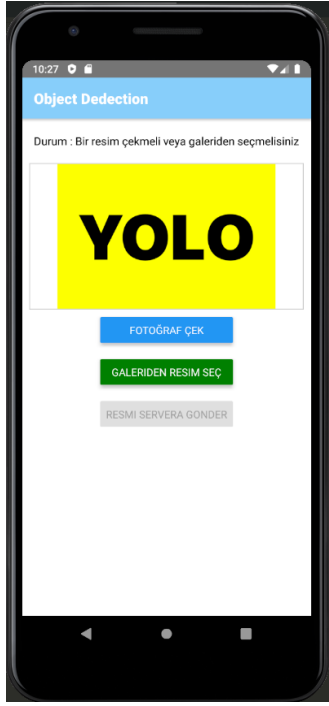
Projemizde react native mobil programlama frameworku kullanmış bulunmaktayız. Projemizi oluştururken herhangi bir ucuncu parti uygulama kullanmadık(örn. Expo). Server tarafında ise bize en uygun olan barındırma ortamının amazon olduğuna karar verdik ve amazon hesabi aktif. Back-End uygulamamız olan nesne tanıma uygulamamızı sunucuya tasıdık. Aynı zamanda nesne depolama işlemini de bu platformda gerçekleştirdik. Projemizin detaylarına girecek olursak Görüntü işleme kısmında birçok alternatif teknolojiyi araştırdık. Fakat bircogu ya maliyetli ya da tam olarak bizim ihtiyaçlarımızı karşılamıyordu. Biz bu yüzden öncelikle YOLOv4 algoritmasının deep learning kutuphanesi olan keras ile kullanimini projemize uyguladık. Fakat sonradan algımız sonuçlarda bu algoritma görüntü işleme kısmını sorunsuz yapmakta fakat 20 saniyelere varan işlem süresiyle çok uzun zaman almaktaydı. Biz de bu amaçla Farklı

alternatiflere yönelmeye karar verdik. YOLOv3 algoritmasının keras kutuphanesine implementasyonunu görüntü işleme uygulamamız için kullanmaya karar verdik. Bu algoritmanın işlem süresi çok daha kısa sürede (yaklaşık 1.sn) alıyordu. Tek dezavantajı olan sadece 80 nesne tanımasıydı. Bizim amacımız tüm nesneleri tanıyan bir modeldi. Bunun için araştırmalar yaptığımız zaman gördük ki bunun için model eğitmemiz gerekiyordu. Fakat model eğitmek için de elimizde çok fazla dataset olması gerekiyordu. Bunu elde etmek çok zor olmasının yanı sıra bu datasetleri eğitmek için işlem gücü çok yüksek bilgisayarların kullanılması gerekiyordu. Biz de bu sebepten dolayı bu algoritmayı kullanmaya karar verdik. Tanıyacağı nesne sayısı sınırlı olmasına rağmen uygulama o belirli nesneler üzerinden yüzde 95'i aşkın doğrulukla çalışıyor. Bu algoritmayla eşleşen nesnelerin konumunu bulmuş olduk. Sıradaki bu nesneleri diktörtgen bir kutu içerisinde göstermek gerekiyordu. Bunun için araştırmalar yaptığımız zaman gördük ki ihtiyacımız olan şey OpenCV kutuphanesiydi. Bu kutuphaneyi de kullanarak konumlarını daha önce tespit ettığımız nesneleri yeşil bir dikdörtgen kutu içerisine alıp bu nesneyi yüzde kaç doğru tanıdığını belirten oran yanında nesnenin ismini de yazdırdık. Aynı zamanda gönderilen resim üzerinde toplam kaç nesnenin tanındığını da resmin sol üst köşesine yazdırdık. Bunu tekrar istemci olan mobil uygulamaya geri gönderdik. Projemizde istek yapmak için ayrıca rest api kullandık. Sunucu tarafında python kullandığımız için flask frameworkunu kullandık. Aslında öncelikle bir diğer python frameworku olan django kullanıyorduk, fakat sonra yaptığımız araştırmalar da öğrendik ki flask frameworku küçük ölçekli ve orta ölçekli uygulamalar için daha verimli çalışan bir frameworkmus biz de bu yüzden uygulamamızı flask frameworkuna döndürdük. Uygulamamızın detaylarına daha sonraki bölümlerde gireceğiz.

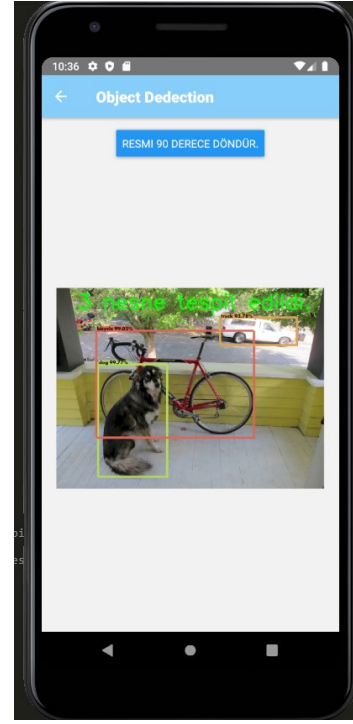
II. TEMEL BİLGİLER

Projeye başlarken öncelikle arayüz kısmını yapmaya başladık. Bu amaçla arayüz kısmında resim gönderme işlemi için react native'de bulunan hazır bir modul kullandık. Bu modul yardımıyla hem galeriden fotoğraflardan hem de anlık olarak çektiğimiz bir resmi api üzerinden istekte bulunarak sunucuda çalışan uygulamaya gönderebildik. Sunucu tarafında YOLOv3 algoritması keras deep learning kutuphanesi ve bir görüntü işleme kutuphanesi olan OpenCV

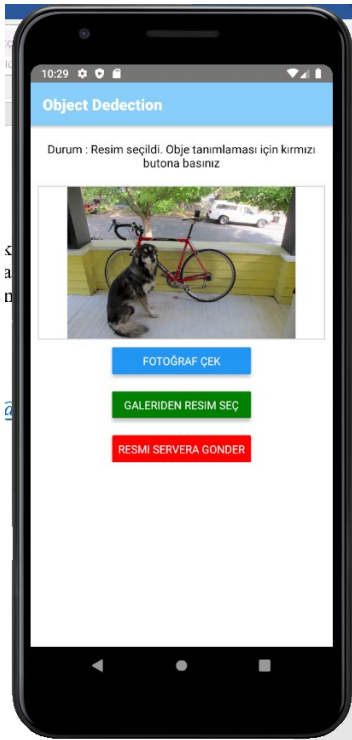
Bulut bilgisayara gönderilecek fotoğrafın elde edilmesinde React Native Image Pickerkütüphanesi kullanılmıştır. Elde edilen fotoğraf React Native Fetch Blob kütüphanesi yardımıyla bulut sunucuya gönderilmiştir. Elde edilen fotoğraf base64 tipine dönüştürülerek post request ile upload işlemi gerçekleştirilmiştir.



Şekil2: Uygulamanın başlangıç ekranı



Şekil4: İşlenmiş fotoğrafın gösterimi



Şekil3: Çekilmiş fotoğrafın gösterimi

D. Bulut Veri Tabanı Üzerinde Görüntü İşlemenin Yapılması

Görüntü işleme sürecini bulutta çalıştırabilmek için bulut veri tabanları ilgili bir takım araştırmalarda bulunduk. Bizim için en iyi bulut sisteminin AWS olduğuna karar verdik. Bu bulut sunucu üzerinde çalışacak işletim sistemini Ubuntu Server 20.04 LTS olarak belirledik.

Bulut bilgisayara gelen resmi alabilmek için http kullandık. Bu http ile gelen requestleri handle etmek için python programlama dilinde Flask framework ünü kullandık. Post request ile gelen resimde yolov3 ile nesne tanımlaması yaptırdıktan sonra response olarak işlenmiş resim döndürülmüştür.

IV. GERÇEKLEŞTİRİLEN TESTLER VE DEĞERLENDİRME

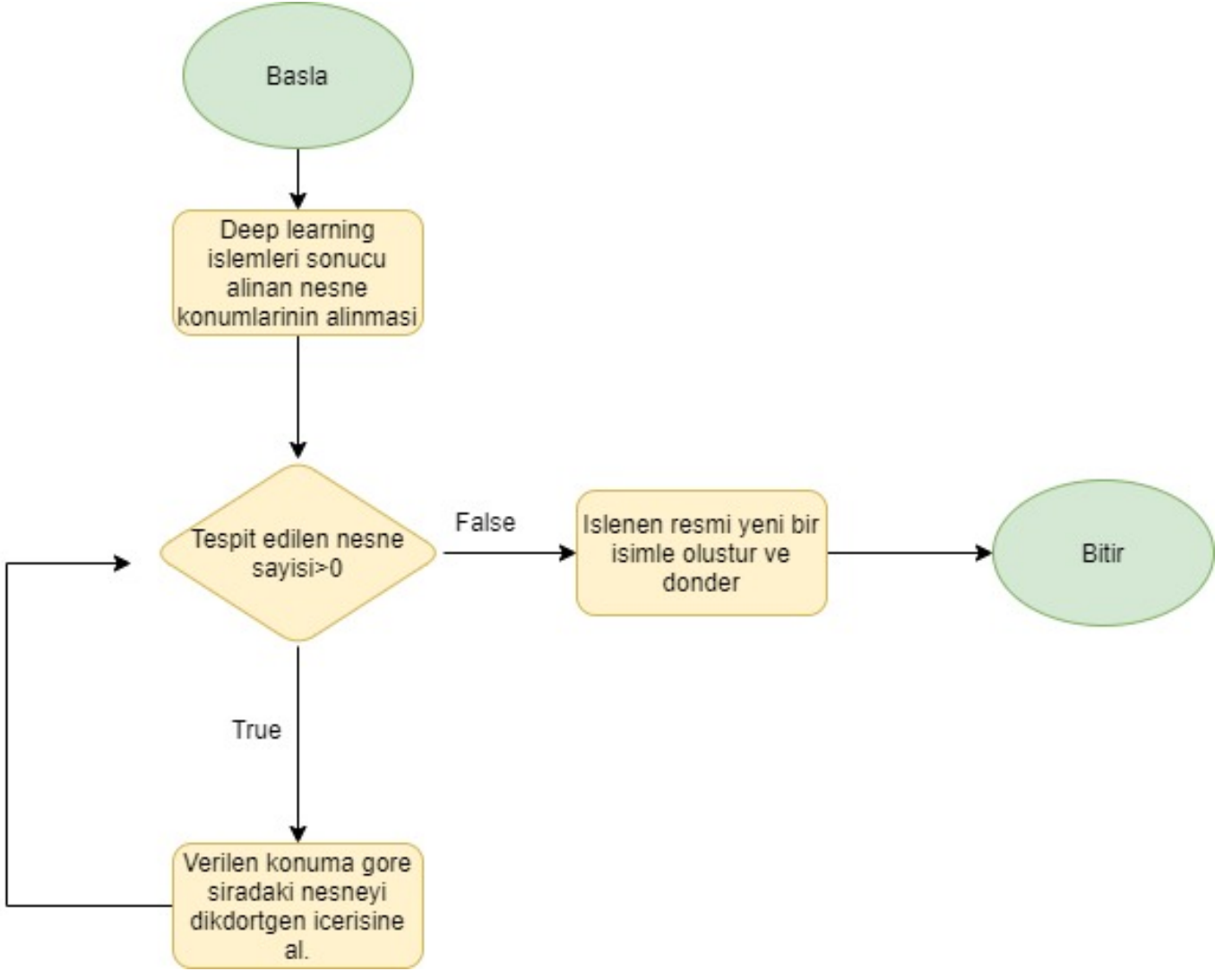
Gerçekleştirdiğimiz testler ve çözülen sorunlar sonucunda istemci-sunucu olarak gerçekleştirdiğimiz proje sorunsuz olarak çalışmaktadır. Programın doğru çalıştığından emin olmak için birden çok bilgisayarda birden çok kez çalıştırdık. Bu proje sonunda sunucu-client ve bulut kavramını anlamış, gerçek bir projeye uygulamış olduk.

V. ÖZET VE SONUÇLAR

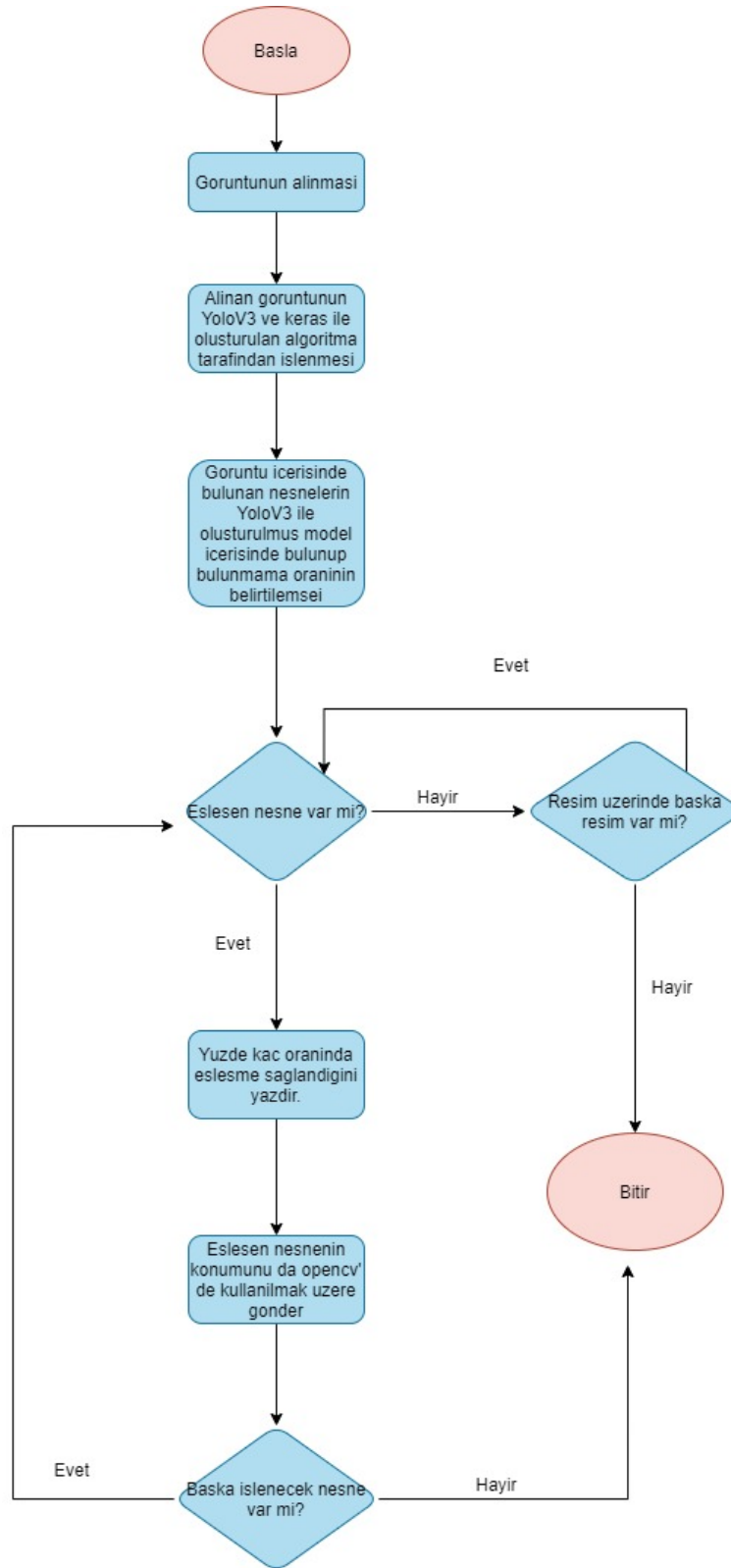
Yaptığımız bu proje sonucunda bizden istenilen tüm isterleri eksiksiz olarak uyguladık. Gerçekleştirilen testler sonucu da bunun hatasız olarak çalıştığını gördük. Bu projeyle birlikte istemci-sunucu kavramını daha iyi anlamış olduk. Bu proje an itibari ile gelişime açık durumdadır.

VI. KAYNAKÇA

- [1] <https://stackoverflow.com/>
- [2] <https://medium.com/@ahmetxgenc/pre-trained-modeller-kullanman>
- [3] <https://www.youtube.com/>
- [4] <https://www.udemy.com/>
- [5] <https://github.com/theAIGuysCode/Object-Detection-API>



Şekil5: Görüntü İşleme Akış Şeması



Şekil6: YoloV3 Algoritmasının Akış Şeması