

**KARADENİZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**





KARADENİZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ



Karadeniz Teknik Üniversitesi Fen Bilimleri Enstitüsünde

Unvanı Verilmesi İçin Kabul Edilen Tezdir.

Tezin Enstitüye Verildiği Tarih : / /

Tezin Savunma Tarihi : / /

Tez Danışmanı :

Trabzon

ÖNSÖZ

Bu çalışmayla sınırlı kalmamak üzere engin ufkundan, bilgeliğinden ve özgün fikirlerinden her daim istifade ettiğim, bu tez çalışmasının araştırma, geliştirme ve yazım aşamalarının yönlendirilmesi ve denetlenmesinde büyük katkılarda bulunan danışmanım Sayın Prof. Dr. Çetin CÖMERT'e sonsuz şükranlarımı sunarım.

Lisansüstü eğitime ve akademiye adım atmam konusunda beni teşvik eden ve ilham veren Prof. Dr. Oğuz GÜNGÖR'e bir teşekkürü borç bilir, saygılarımı sunarım.

Lisansüstü eğitimimin ders döneminde kendisinden aldığım dersler ve önerileriyle ilerlememe önemli katkıda bulunan Sayın Prof. Dr. Murat EKİNCİ'ye bir teşekkürü borç bilirim.

Mesai arkadaşlığı ile sınırlı kalmayıp gerek akademide gerekse akademi dışında desteğini, ağabeyliğini ve bilgilerini esirgemeyen Arş. Gör. Dr. Ziya USTA'ya çok teşekkür ederim.

Geliştirme ve yazım süreçlerinde bilgilerini, önerilerini ve eleştirilerini benimle paylaşan kıymetli dostlarım Arş. Gör. Eren Can SEYREK'e, Makine Mühendisi Emre ÇELİK'e ve desteklerini gördüğüm hocalarıma ve arkadaşlarıma teşekkür ederim.

Bu çalışmada kullanılan EmguCV isimli OpenCV kütüphanesinin .NET dilleri üzerinde kullanılmasını sağlayan eklentiye benimle karşılık beklemeden paylaşan, eklentinin geliştiricisi ve dağıtıcısı Canming HUANG'a teşekkür ederim.

Bu tez çalışması KTÜ BAP tarafından FBA-2021-9488 kodlu proje kapsamında desteklenmektedir.

Başta annem, babam ve eşim olmak üzere maddi ve manevi desteklerini hep yanımda hissettiğim aileme minnettarlığımı belirtmek isterim.

Alper Tunga AKIN
Trabzon, 2021

TEZ ETİK BEYANNAMESİ

Yüksek Lisans Tezi olarak sunduđum “Görme Engelliler İçin Bir Artırılmış Gerçeklik Uygulaması Geliştirilmesi” başlıklı bu çalışmayı baştan sona kadar danışmanım Prof. Dr. Çetin CÖMERT’in sorumluluğunda tamamladığımı, verileri/örnekleri kendim topladığımı, deneyleri/analizleri ilgili laboratuvarlarda yaptığımı/yaptırdığımı, başka kaynaklardan aldığım bilgileri metinde ve kaynakçada eksiksiz olarak gösterdiğimi, çalışma sürecinde bilimsel araştırma ve etik kurallara uygun olarak davrandığımı ve aksinin ortaya çıkması durumunda her türlü yasal sonucu kabul ettiğimi beyan ederim. 16/12/2021

Alper Tunga AKIN

İÇİNDEKİLER

	<u>Sayfa No.</u>
ÖNSÖZ.....	III
TEZ ETİK BEYANNAMESİ.....	IV
İÇİNDEKİLER.....	V
ÖZET	VII
SUMMARY	VIII
ŞEKİLLER DİZİNİ	IX
TABLolar DİZİNİ.....	X
1. GENEL BİLGİLER.....	1
1.1. Giriş	1
1.2. Problem Tanımı	2
1.3. Çalışmanın Amacı	2
1.4. Metodoloji	3
1.5. Temel Kavramlar.....	3
1.5.1. Artırılmış Gerçeklik (AR)	3
1.5.2. Derin Öğrenme Modelleri	4
1.5.2.1. Görüntü Üzerinden Obj e Tespiti	8
1.5.2.2. Monoküler Derinlik Çıkarımı.....	10
1.5.3. Navigasyon	12
1.6. Benzer Çalışmalar.....	13
2. YAPILAN ÇALIŞMALAR.....	18
2.1. Materyaller.....	18
2.1.1. Veri Kaynakları	18
2.1.2. Kullanılan Geliştirme Ortamları ve Araçları.....	18

2.1.2.1. Unity Oyun Motoru	18
2.1.2.2. Anaconda	19
2.1.2.3. Spyder	19
2.1.2.4. Visual Studio Code	19
2.1.2.5. Flask	20
2.1.2.6. OpenCV	20
2.1.2.7. PyTorch	20
2.1.3. Kullanılan Donanım	21
2.2. Yöntem	22
2.2.1. Genel Uygulama Mimarisi	22
2.2.2. İstemci Uygulaması	23
2.2.3. Web Servislerinin Geliştirilmesi	26
2.2.3.1. Obje Tespiti Web Servisi	27
2.2.3.2. Derinlik Çıkarımı Web Servisi	28
3. BULGULAR VE TARTIŞMA	30
4. SONUÇ VE ÖNERİLER	36
5. KAYNAKÇA	38
ÖZGEÇMİŞ	

Yüksek Lisans Tezi

ÖZET

GÖRME ENGELLİLER İÇİN BİR ARTIRILMIŞ GERÇEKLİK UYGULAMASI
GELİŞTİRİLMESİ

Alper Tunga AKIN

Karadeniz Teknik Üniversitesi
Fen Bilimleri Enstitüsü
Harita Mühendisliği Anabilim Dalı
Danışman: Prof. Dr. Çetin CÖMERT
2021, 52 Sayfa

Bu çalışma kapsamında görme engelli bireylerin yaşamlarını kolaylaştırmak adına canlı görüntü kaydı yapmakta olan akıllı telefonlar ile çalışan bir sesli artırılmış gerçeklik uygulaması geliştirilmiştir. Uygulama anlık telefon kamerası görüntüsü üzerinden obje tespiti, engel bildirim-i-derinlik çıkarımı sağlamakta; bununla birlikte çevrimiçi harita altlıkları üzerinden navigasyon sağlamakta ve bu çıktıları sesli geri bildirimle görme engelli kullanıcıya sunmaktadır. Bu işlevleri yerine getirirken derin öğrenme tekniklerinin problem çözme yeteneklerinden faydalanmaktadır. Hesap yükü ve depolama alanı bakımından oldukça maliyetli derin öğrenme modellerinin web servislerine dönüştürülerek kullanılması sayesinde uygulama, kullanıcı cihazında bir yük oluşturmadan orta segment ve altı cihazlarda dahi sonuç üretebilen kolay ulaşılabilir, ekonomik bir çözüm niteliği kazanmıştır. Ayrıca, dünya genelinde geliştiricilerin katılımına açık, açık kaynak kodlu ve yerli bir yazılım ürünü elde edilmiştir.

Anahtar Kelimeler: Artırılmış Gerçeklik, Obj e Tanıma, Navigasyon, Monoküler Derinlik Çıkarımı

Master Thesis

SUMMARY

DEVELOPMENT OF AN AUGMENTED REALITY APPLICATION FOR VISUALLY
IMPAIRED

Alper Tunga AKIN

Karadeniz Technical University
The Graduate School of Natural and Applied Sciences
Geomatics Engineering Graduate Program
Supervisor: Prof. Dr. Çetin CÖMERT
2021, 52 Pages

Within the scope of this study, an audio augmented reality application that works with smartphones recording live images has been developed to facilitate visually impaired individuals' lives. The application provides object detection and obstacle notification-depth extraction via instant phone camera image. It also provides navigation via online maps and presents these outputs to the visually impaired user with audible feedback. While performing these functions, it benefits from the problem-solving capabilities of deep learning techniques. Thanks to the use of deep learning models via web services, which are very costly in terms of computational load and storage, the application has become an easily accessible, economical solution that can produce results even in the low-end or mid-end devices without causing a load on the user device. In addition, an open-source and domestic software product, open to the participation of developers around the world, was obtained.

Keywords: Augmented Reality, Object Detection, Navigation, Monocular Depth Extraction

ŞEKİLLER DİZİNİ

Sayfa No.

Şekil- 1. Bir ANN mimarisi örneği. (inputs: girdiler, layer: katman, outputs: çıktılar, neurons: nöronlar, connections: ağırlıklar) (Rashid, 2016)	5
Şekil- 2. İki farklı görüntüde farklı dönüklükte bulunan X karakteri. (URL-2, 2021).....	6
Şekil- 3. Bir konvolüsyon operasyonu örneği (Goodfellow vd., 2016)	7
Şekil- 4. Bir CNN mimarisi örneği. (Convolution Layer: Konvolüsyon katmanı, feature map: öznelik haritası, fully connected layer: son aşamadaki sinir ağı) (Borjali vd., 2020)	7
Şekil- 5. Grid bazında obje sınıflarına göre olasılık haritası ve obje tespitleri (Redmon vd., 2016)	9
Şekil- 6. 4 farklı ölçekteki detaylar (sarı,yeşil,mavi,siyah) ve nihai derinlik haritası (kırmızı) (Lee vd., 2019).....	12
Şekil- 7. Örnek bir rota ve bu rotayı takip etmek için gerekli direktifler	13
Şekil- 8. Uygulamanın Genel Mimarisi.....	23
Şekil- 9. Directions API yanıtı örneği. Talimatlar (mavi kutucuk) ve bu talimatların ilişkilendirildiği coğrafi konumlar (kırmızı kutucuk).	25
Şekil- 10. Uygulama arayüzü (1 - İlk çalıştırmada boy girişi; 2 - Uygulamanın koşumu esnasındaki arayüz, navigasyon işlemini açma-kapama butonu ve sanal şütör). 26	
Şekil- 11. Decorator yapıları ve Flask fonksiyonları. (depthExtract() fonksiyonunun çalışması için istemcinin "...endpoint.../pass4depth" adresine görüntü POST etmesi gerekmektedir.).....	27
Şekil- 12. Objeye tespiti web servisi iş akışı.....	28
Şekil- 13. Derinlik çıkarımı web servisi iş akışı (Akın ve Cömert, 2021)	29
Şekil- 14. Ölçülen Mesafe - Servis Üretimi Mesafe (kırmızı: ölçülen, mavi: servis üretimi) (Akın ve Cömert, 2021)	31
Şekil- 15. Test görüntülerine karşılık derinlik haritaları (Akın ve Cömert, 2021).....	32
Şekil- 16. YOLO ve benzer modellerin kıyaslaması (Redmon, Farhadi, 2018).	34

TABLULAR DİZİNİ

	<u>Sayfa No.</u>
Tablo 1. İş istasyonu sistem konfigürasyonu	21
Tablo 2. Akıllı telefonların sistem konfigürasyonları	22
Tablo 3. Derinlik çıkarımı servisi doğruluk metrikleri (Akın ve Cömert, 2021)	31
Tablo 4. YOLO performans tablosu (URL-28, 2021)	33
Tablo 5. Çalışma kapsamında geliştirilen uygulamanın benzer çalışmalarla kıyaslanması 35	



1. GENEL BİLGİLER

1.1. Giriş

İnsanoğlu dış dünyadan edindiği bilgilerin yaklaşık %85'ini görme duyusu ile alır. Görüntü bilgisi insan için dış dünyayı algılayıp yorumlama ve etkileşime geçme noktasında birinci dereceden önem arz etmektedir. Ayrıca, çocukluk döneminden itibaren pek çok psikomotor beceri de yetişkin davranışlarının gözlemlenip taklit edilmesiyle edinilir ve iletişim becerilerinin de ilk adımı göz kontağı kurulması ile atılır (URL-1, 2021). Dolayısıyla görme duyusunun eksikliği ya da tümenden yoksunluğu bireyin yaşam kalitesini direkt olarak olumsuz yönde etkileyen bir durumdur.

Dünya Sağlık Örgütü (DSÖ)' nün "World Report on Vision" başlıklı raporuna göre 2.2 milyar insan bir görme engeli taşımaktadır (Dünya Sağlık Örgütü, 2019). Bu da demektir ki, dünya nüfusunun yaklaşık %29'u görme duyusu eksikliğinin getirdiği olumsuzluklardan etkilenmektedir. Öyle ki; karşıdan karşıya geçmek, karşılaşılan insanları tanımak, yolda yürürken engellerden kaçınmak, bulunulan konumdan bir hedef konuma doğru yola koyulmak, giyinmek gibi çevredeki nesnelerin tanınmasını gerektiren tüm temel günlük aktivitelerde görme engelli bireyler güçlük çekmekte ve bir asistana ihtiyaç duymaktadırlar. Bu asistan bir evcil hayvan ya da bir rehber birey olabilmektedir fakat böylesi bir asistanın mutlak gerekliliği sebebiyle birey bağımsız bir yaşam sürdürememektedir. Ancak günümüzde teknolojinin insan yaşamına hızlı entegre olacak biçimde ilerlemesi, görme engelli bireylerin bireysel bağımsızlıklarını artırmaya ve hayat standartlarını yükseltmeye yönelik çözümlerin geliştirilmesini mümkün kılmaktadır.

Günümüzde gerek giyilebilir teknolojiler gerekse tablet ve akıllı telefon gibi mobil cihazlar üzerinden görme engelli bireylere yönelik çözümler üretilmektedir. Bu çözümlerde bireylere normal şartlar altında görme duyusu ile elde edilen obje türü, engel, rota gibi bilgiler ses veya titreşim yoluyla iletilmektedir. Çünkü görme engelli bireylerde görme duyusundaki eksikliğe karşılık haptik (dokunsal) ve işitsel beceriler gelişmiştir (URL-1, 2021). Bu çözümlerin üretimi aktif bir çalışma alanı olup; bu çözümler teknolojinin ilerleme hızına paralel olarak görme engelli bireylerin yaşamlarına daha verimli katkı verecek ve bireyler tarafından daha kolay edinilebilecek biçimde geliştirilmeye devam etmektedirler.

1.2. Problem Tanımı

Görme engelli bireyler için geliştirilen çözümlerde karşılaşılan obje türü ve engel, karşılaşılan obje ile kullanıcı arası mesafe, bilgisi kullanıcıya genelde iki yöntemle verilmektedir. Bunlardan ilki obje türü ve coğrafi konum karşılığı belirli QR kod, levha, tabela gibi işaretçilerin kamera görüntüsü üzerinden algılanmasıyla gerçekleştirilir. İkincisi ise kullanıcının bulunduğu ortamın fiziksel yapısının modellenmesi için kullanılan sensörler vasıtasıyla gerçekleştirilir. İlk yöntemde bireyin günlük yaşamda karşılaşılabileceği rastlantısal, işaretçinin obje türü ve konum bilgisi karşılığı uygulamanın veritabanında bulunmayan, objelerin bilgisinin kullanıcıya sunulması olanaklı değildir. İkinci yöntem grubundaki çalışmalar da söz konusu sensörlerin ekonomik maliyeti ve bu sensörlerden gelen verinin işlenmesindeki hesaplama maliyeti sebebiyle kolay gerçekleştirilebilir çözümler değildir. Bu çalışma kapsamında, obje türü ve engel bilgilerinin ikinci gruba dahil olabilecek bir yöntemle, ancak yalnızca bir akıllı telefon kamerası görüntüsü üzerinden, düşük maliyetle nasıl üretilip sunulabileceği problemi ele alınmıştır. Bunun yanı sıra, obje türü ve engel bilgisine ek olarak, navigasyon işlemi de gerçekleştirip tüm bu çıktıları sesli olarak görme engelli kullanıcıya tek bir yazılım uygulaması aracılığıyla sunmanın yolları aranmıştır.

1.3. Çalışmanın Amacı

Bu çalışmada görme engelli bireylerin görme duyusu yoksunluğu sebebiyle edinemedikleri obje türü ve engel bilgisini bir akıllı telefon kamerası üzerinden web servisleri aracılığıyla üreten, akıllı telefonun konum servisleri üzerinden de navigasyon sağlayıp tüm bu çıktıları sesli olarak sunan bir artırılmış gerçeklik (AR) uygulaması üretilmesi amaçlanmıştır. Web servisleri vasıtasıyla bilgi üretilip sağlayacağından, kullanıcı cihazının konfigürasyonundan bağımsız bir uygulama olacaktır. Aynı zamanda açık erişimle sunulacağı için görme engelli bireyler için kolay ulaşılabilir ve ekonomik bir çözüm niteliği taşıyacaktır. Bununla birlikte, açık kaynak kodlu yazılım bileşenleri kullanılarak geliştirilmesi ve kaynak kodlarının çevrimiçi kod depolarında paylaşılması dünya genelinden yazılım geliştiricilerin de katkıda bulunmasına olanak tanıyacaktır.

1.4. Metodoloji

Çalışma süresince gerçekleştirilen işlem adımları aşağıda sıralanmıştır.

- Konuya ilişkin literatürün kavranabilmesi için gerekli altyapıyı sağlamak amacıyla ilgili temel kavramların irdelenmesi,
- Konuya ilişkin literatürün araştırılması,
- Anlık görüntü üzerinden obje tespiti gerçekleştiren bir derin öğrenme modelinin web servisine dönüştürülmesi,
- Anlık görüntü üzerinden derinlik çıkarımı-engel bildirimini gerçekleştiren bir derin öğrenme modelinin web servisine dönüştürülmesi,
- Anlık konum ve hedef konum arasında navigasyon gerçekleştiren uygulama bileşeninin tasarlanması,
- Çıktıların kullanıcıya sesli olarak sunulmasını sağlayan uygulama bileşeninin tasarlanması,
- Uygulamanın test edilmesine yönelik deneysel çalışmaların gerçekleştirilmesi,
- Test sonuçlarının irdelenmesi.

1.5. Temel Kavramlar

1.5.1. Artırılmış Gerçeklik (AR)

AR, gerçek dünyanın bilgisayar üretimi bilgi ile iyileştirilmesi (Furth, 2006) ya da eş bir deyişle sentetik bilgi ile gerçek çevrenin entegre edilmesidir (Bimber ve Raskar, 2005). Bu entegrasyon, gerçekliğin artırımı, bireyin içinde bulunduğu gerçeklikte, duyu organlarıyla edinemediği bilginin bireye duyu organları ile algılayabileceği biçimde sunulması ile gerçekleşir. Bu şekilde gerçekliği artırıcı işlevleri yerine getiren uygulamalar AR uygulaması olarak adlandırılırlar. AR uygulamaları bu işlevleri gerçekleştirirken kullandıkları veriyi gerek üzerinde koştukları cihazlardaki iç sensörler, gerekse ilave sensörler vasıtasıyla edinirler. İç sensörlere örnek olarak kamera, Küresel Konumlandırma Sistemi (Global Positioning System - GPS) alıcısı, uçuş süresi (Time-of-Flight (- ToF), Ataletsel Ölçüm Birimi (IMU (Inertial Measurement Unit - IMU); ilave sensör örnekleri olarak da ultrason, LiDAR ve RADAR verilebilir. Bu çalışmada yol haritası sunulan

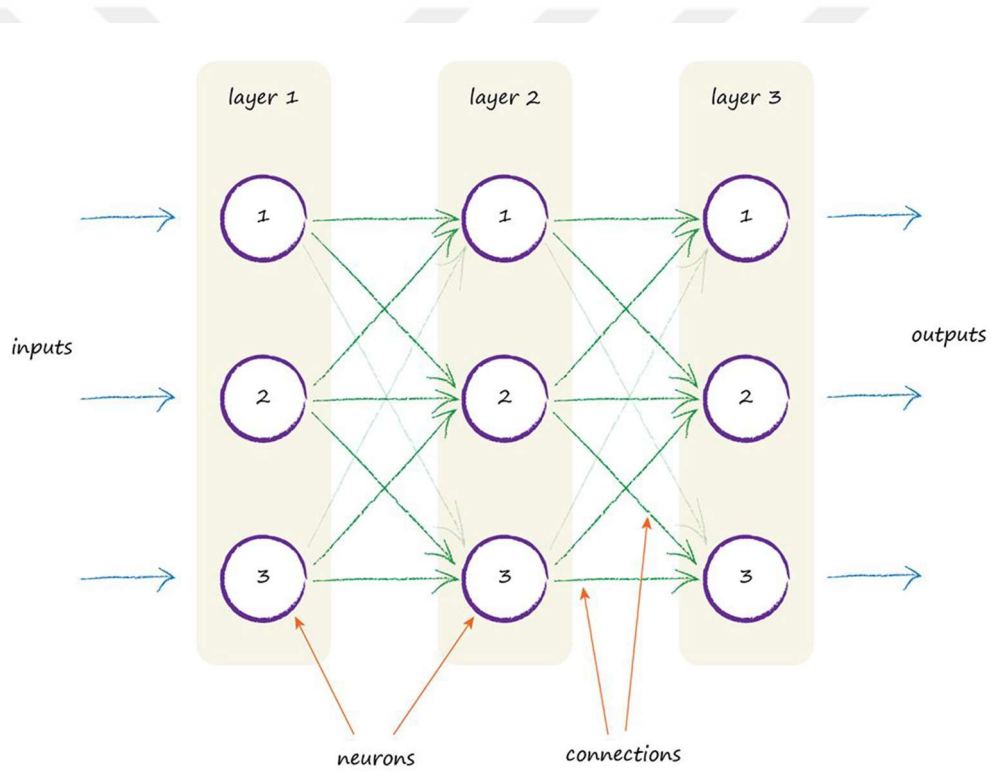
uygulama akıllı telefonun kamera, GPS alıcısı ve IMU sensörlerini kullanmaktadır. Kamera, görme engeli bulunmayan bireylerin görme duyusuyla edindikleri obje türü ve engel bilgisinin üretilmesinde kullanılmaktadır. GPS alıcısı ve IMU ise navigasyon işlemlerinin gerçekleştirilebilmesi için gereklidir. Bahsedilen çıktılar kullanıcıya işitsel kanalla algılatmak üzere sese dönüştürülmektedir. Beş duyu organının tümü işlevini yerine getirebilen bireylerin gerçekliği eğer 5 boyutlu olarak nitelendirilirse, görme engelli bireylerde gerçeklik 4 boyutludur denilebilir. Buradan hareketle bu uygulama, görme engelli bireyin gerçekliğinin boyutlarını artırıcı bir rol üstlendiğinden, bir AR uygulaması olarak nitelendirilmektedir.

1.5.2. Derin Öğrenme Modelleri

Derin öğrenme (Deep Learning - DL), makine öğrenmesi alanının bir alt kümesi olup, yapay ve konvolüsyonel sinir ağları (Artificial Neural Network - ANN, Convolutional Neural Network - CNN) temelleri üzerine inşa edilmiş bir teknikler bütünüdür. DL teknikleri, sınıflandırma ya da değer kestirimi gibi problemler bağlamında aralarındaki ilişki net bir biçimde formülize edilemeyen bağımsız (girdi) ve bağımlı (çıkıtı) değişkenler arasındaki ilişkiyi, hatanın geriye yayılımına (back-propagation) dayanan iteratif öğrenme süreçleriyle çözerler. Bu öğrenme süreçleri “eğitim” olarak adlandırılır. Eğitim kavramının okuyucunun zihninde somutlaştırılabilmesi için ANN ve CNN yapılarının çalışma prensiplerinin irdelenmesinde fayda vardır.

Bir ANN'nin temel yapı taşı yapay nöronlardır ve ANN bu nöronların birbirlerine ağırlık katsayıları üzerinden bağlantılı olduğu katmanlı bir ağ yapısıdır (Zurada, 1992). Burada katman, ardışık nöron dizilerini ifade eder (Şekil 1). ANN'lerin, ileri yayılım (feed-forward) ve geri yayılım olmak üzere iki temel işlevi vardır. İleri besleme safhası rastgele başlangıç ağırlıkları (initial) ile başlar. Bu safhada her nöron, kendisine bağlı olan ağırlıklar ile girdi vektörü elemanlarının ağırlıklandırılmış toplamından bir çıkıtı (net) değeri üretir. Bu çıkıtı değeri bir aktivasyon fonksiyonuna verilir ve $f(\text{net})$ değeri elde edilir. $f(\text{net})$ ardıl katmandaki nörona gönderilir ki bu değer, ardıl katmandaki nöronun girdisi niteliğindedir (Fausett, 1994). Bu işlemin ardışık her katmanda gerçekleşmesinin ardından, son katmanda elde edilen çıkıtı vektörü elemanları (output-observed) ile beklenen çıkıtı vektörü (desired) arasındaki fark (desired - output) belirlenen bir hata fonksiyonuna verilerek hata değeri

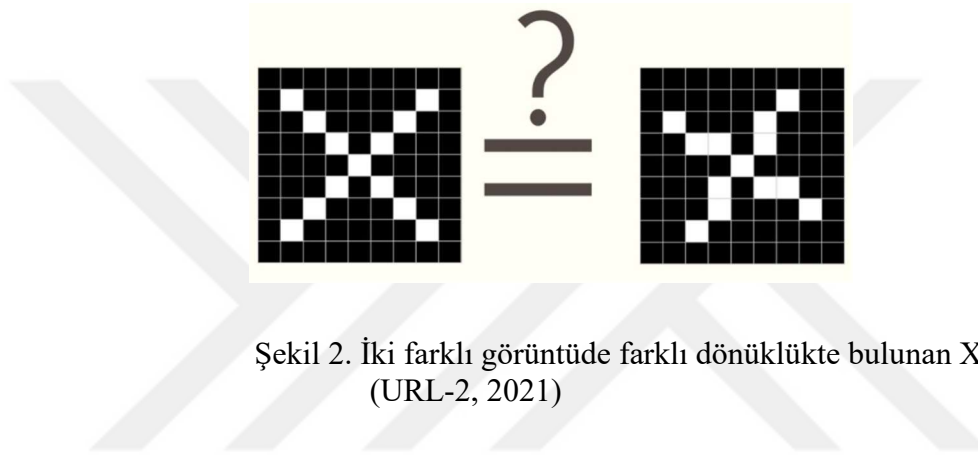
hesaplanır. Eğer ki bu hata değeri belirli bir eşik değerin üzerinde ise elde edilen çıktı ile beklenen çıktı arasındaki fark, gradyan düşüş (gradient descent) yöntemleriyle net değerlerinin değişimine ($f'(net)$) göre, ağırlıklara dağıtılır (Bishop, 2006). Bu işlem geri yayılım safhasında gerçekleşir ve buna “hatanın geriye yayılımı” denir. Güncellenen ağırlıklar ile bir sonraki iterasyona geçilir. Hata değeri belirlenen eşik değerin altına düşüncüye dek iterasyonlar devam eder. İterasyonlar sonucu hata değerin eşik değerin altına düşmesiyle ağıın “eđitimi” tamamlanmış olur. Böylelikle ANN, girdi ve çıktı veri kümeleri arasındaki ilişkiyi öğrenmiş olur ki, ilgili problemde eğitim veri kümesinde olmayan her yeni girdi için bir yeni çıktı değeri üretebilir duruma gelir. Buna ağıın “test” edilmesi denir. Test işlemi geri besleme içermeyen, salt bir ileri besleme işlemidir.



Şekil 1. Bir ANN mimarisi örneđi. (inputs: girdiler, layer: katman, outputs: çıktıları, neurons: nöronlar, connections: ağırlıklar) (Rashid, 2016)

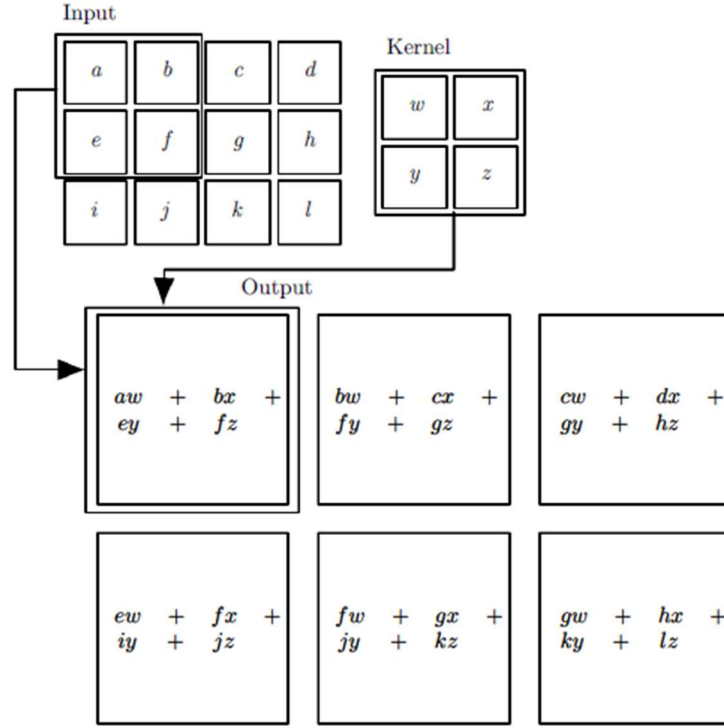
Girdi verisinin bir ANN yapısına verilebilecek şekilde yalnızca kendine dair nitelikleri temsil ettiđi, düzenli bir yapıda olmadığı durumlar olabilir. Verinin yapısını düzensizleştiren durumlar verideki gürültü, görüntü sınıflandırma problemi özelinde örneklendirilirse görüntü üzerindeki objenin ölçeđi, dönüklüğü gibi verinin edinilmesi

esnasındaki rastlantısal etkenler ya da verinin edinildiği sensör ve çevreye dair etkiler olabilir. Bu gibi durumlarda veri ANN'ye ham olarak aktarılmamalıdır. Verinin kendisini tanımlayıcı, ait olduğu sınıfa dair öznitelikleri çıkarılıp ANN'ye aktarılmalıdır. Öznitelik çıkarımı ihtiyacı ile birlikte CNN yapıları ortaya çıkmıştır. Şekil 2'deki örneğe bakıldığında iki görüntü üzerinde farklı dönüklükte bulunan iki X karakteri görülmektedir. CNN yapılarıyla bu iki görüntünün aynı sınıfta sınıflandırılması, yani iki görüntünün de X karakteri sınıfına dahil edilmesi için öncelikli olarak iki görüntü üzerinden X karakterine dair öznitelikler çıkarılır ve bu öznitelikler üzerinden sınıflandırma yapılır.

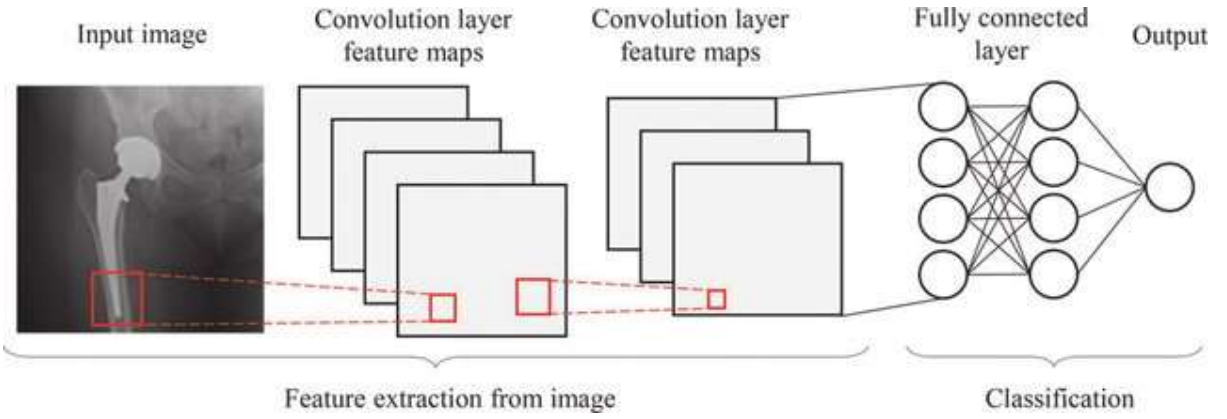


Şekil 2. İki farklı görüntüde farklı dönüklükte bulunan X karakteri.
(URL-2, 2021)

CNN'ler temel olarak, katmanlarının en az birinde konvolüsyon operasyonu barındıran özel sinir ağlarıdır (Goodfellow vd., 2016). Konvolüsyon operasyonu, bir görüntü işleme problemi üzerinden ele alınırsa, görüntünün bir operatör pencere (kernel) ile filtrelenmesi işlemidir (Şekil 3). ANN'lerdeki nöronların yerini konvolüsyon katmanlarında filtre operatörlerinin aldığı düşünülürse, ağırlıkların yerini de operatörün elemanları (Şekil 3'teki w, x, y, z) almıştır. Ardışık konvolüsyon katmanlarıyla girdi görüntü üzerindeki objelerin yönelim, ölçek, gürültü ve aydınlatma gibi etkenlerden bağımsız, objeyi tanımlayıcı, öznitelikleri filtrelenir. Filtrelenen özniteliklerin oluşturduğu öznitelik haritası CNN'nin son aşaması olan sinir ağına vektörize edilerek verilir (Şekil 4). Sinir ağının çıktısı ile beklenen çıktı kıyaslanarak ANN'ler ile aynı şekilde CNN'nin eğitimi sağlanır. CNN'lerin eğitimi ile konvolüsyon operatörlerinin elemanları ve son aşamadaki sinir ağının ağırlıkları öğrenilen parametreler olurlar.



Şekil 3. Bir konvolüsyon operasyonu örneği (Goodfellow vd., 2016)



Şekil 4. Bir CNN mimarisi örneği. (Convolution Layer: Konvolüsyon katmanı, feature map: öznelik haritası, fully connected layer: son aşamadaki sinir ağı) (Borjali vd., 2020)

CNN ve ANN yapıları, başta görüntü işleme problemleri olmak üzere çeşitli problemlerin çözümüne yönelik, probleme göre şekillenen katman kombinasyonları ve mimarilerle DL modellerini oluştururlar. DL modelleri bir problem özelinde, girdi ve çıktı

niteliğindeki veri kümeleri arasındaki formülize edilemeyen ilişkiyi eğitim süreçleri ile öğrenirken çok yüksek miktarda eğitim verisine ve hesaplama gücü yüksek işlemcilerle ihtiyaç duyarlar. Söz edilen eğitim süreçleri yüksek kapasiteli grafik işlemciler (Graphics Processing Unit - GPU) kullanılarak yürütülmektedir. Öyle ki bu çalışma kapsamında görüntüden obje tespiti için tercih edilen YOLO (You Only Look Once) (Redmon vd., 2016) modeli 123,000 adet görüntüden oluşan COCO (Common Object in Context) (Lin vd., 2014) veri seti ile NVIDIA V100 16GB GPU kullanılarak, görüntüden derinlik tespiti için tercih edilen BTS (from Big To Small) (Lee vd., 2019) modeli ise 23,488 adet görüntüden oluşan KITTI (Geiger vd., 2013) veri seti ile NVIDIA 1080TI 11GB GPU kullanılarak eğitilmiştir (URL-3, URL-4, 2021). Buradan anlaşılacağı üzere DL modellerinin eğitim işlemi hesaplama gücü ve süresi bakımından oldukça maliyetlidir. Eğitim süreçlerinin bu maliyetinden ötürü günümüzde, belirli problemlerin çözümüne yönelik eğitilmiş DL modelleri ön eğitilmiş (pre-trained) olarak yayınlanmaktadır. Ön eğitilmiş bir DL modeli, eğitim süreci sonunda öğrenilen parametrelerin bir bağlantılı çizge (connected graph) yapısında test için yeniden kullanılabilir biçimde dışarı aktarılmasıyla oluşur. Böylelikle ön eğitilmiş modeli edinen bir geliştirici, modeli bir eğitim sürecine tabi tutmaksızın, direkt olarak test modunda uygulamasına dahil edebilir. Bu süreç eğitimin aktarılması (transfer of learning) olarak adlandırılır (URL-5, 2021). Bu çalışma kapsamında da YOLO ve BTS modelleri ön eğitilmiş olarak kullanılmışlardır.

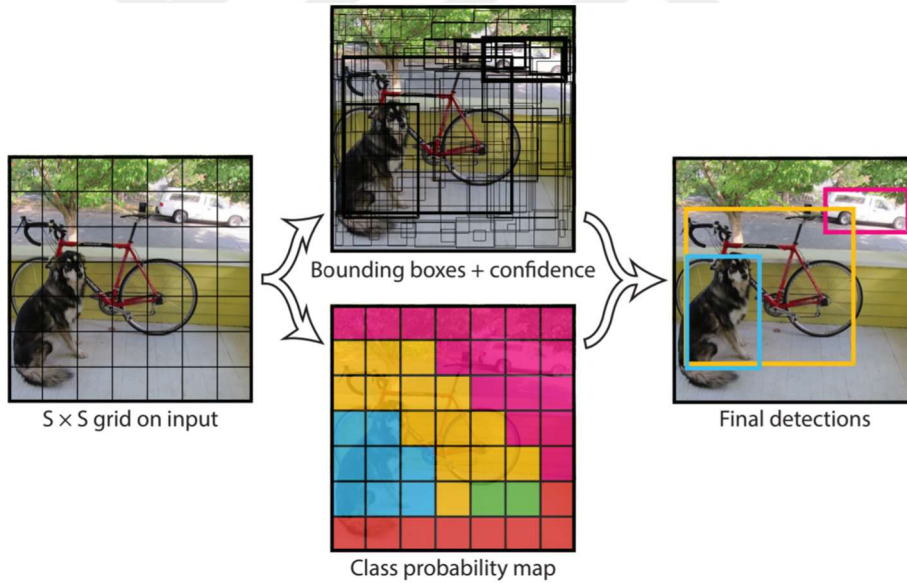
1.5.2.1. Görüntü Üzerinden Obje Tespiti

DL modelleri görüntüden obje tespiti için, RGB (Kırmızı, Yeşil, Mavi bant sıralamasına göre kodlanmış) görüntü ve bu görüntüye karşılık obje etiketleri verilerek eğitilirler. Etiketler obje sınıfını ve bu objenin görüntü üzerindeki piksel biriminde konumunu içerir. Bu eğitim süreci sonucunda model test aşamasında, yani uygulamalardaki kullanımında, eğitim için seçilen veri seti içindeki tüm objelerin sınıflarını ve konumlarını görüntü üzerinde bulabilir duruma gelir.

Bu çalışmada olduğu gibi, canlı video akışı kareleri üzerinden anlık obje tespiti sağlayan DL modelleri incelendiğinde modelin tahmin hızı (inference time), saniyelik görüntü karesi hızı (Frame Per Second - FPS) ve doğruluk metriklerine göre Single Shot MultiBox Detector (SSD) (Liu vd., 2016) ve YOLO isimli iki modelin ön plana çıktığı görülmektedir. Bu iki modeli belirtilen metriklere göre emsallerinden avantajlı kılan

özelliklerinden biri, girdi görüntü karesini eşit gridlere bölerek yalnızca obje barındırma olasılığı yüksek gridler üzerinde tanıma yapmalarıdır (Şekil 5). Bir diğer özellik de aynı amaca yönelik kullanılan modellere göre, daha az parametre hesaplaması gerektiren öznitelik çıkarıcı konvolüsyon katmanları kullanıyor olmalarıdır (Liu vd., 2020). Ayrıca, tek seferde birden fazla obje ile etiketlenmiş görüntüler ile eğitilebilir ve test esnasında tek bir görüntü karesi üzerinden birden fazla obje tanıyabilir olmaları da diğer modellere kıyasla bu modelleri üstün kılmaktadır.

YOLO modeli, SSD modeline göre yüksek fps değerlerinde çalışabiliyor olması sayesinde bu çalışmada tercih edilen model olmuştur (Zhao vd., 2019). YOLO modelinin, COCO veri seti ile eğitilmiş bir versiyonu ön eğitimli olarak çalışmaya dahil edilmiştir. COCO, 91 genel obje sınıfı içeren ve 123,000 etiketli RGB görüntüden oluşan bir veri setidir. Buradaki genel sözcüğü, COCO'nun makalesinde "4 yaşındaki bir çocuk tarafından rahatça tanınabilen" olarak ifade edilmiştir.



Şekil 5. Grid bazında obje sınıflarına göre olasılık haritası ve obje tespitleri (Redmon vd., 2016)

1.5.2.2. Monoküler Derinlik Çıkarımı

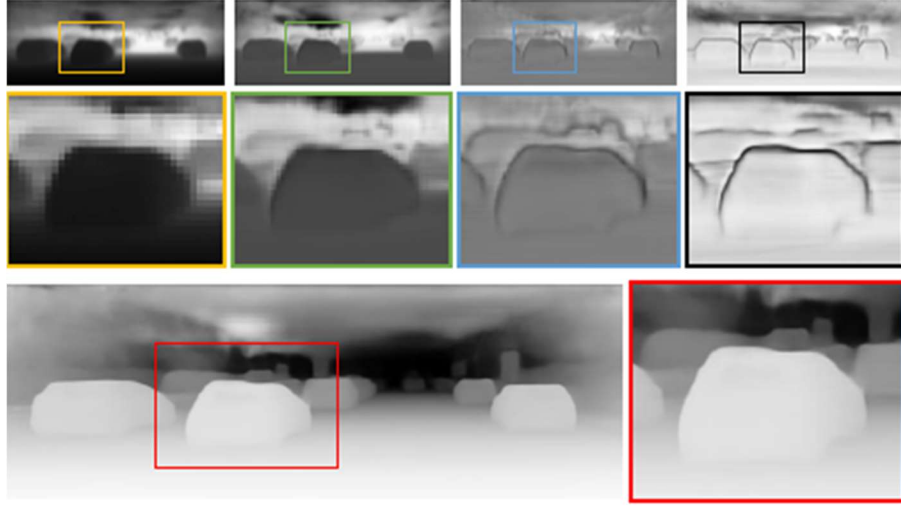
Görüntüden metrik derinlik bilgisi çıkarımını akıllı telefon ya da tablet gibi mobil cihazlar aracılığıyla sağlayan çalışmalar, geometrik ve algısal yaklaşımlar olmak üzere iki ana gruba ayrılabilir. Geometrik yaklaşımda derinlik bilgisi, çoklu görüntü geometrisi prensiplerine göre, stereo görüntülerin eşleştirmesi ile elde edilir. Cihaz kamerası ile alınan görüntü kaydının karelerinin birbirleri arasındaki yönelme elemanlarının bilinmesiyle görüntü kareleri birbirleriyle eşleştirilerek epipolar düzlem üzerinden kamera-obje arasındaki derinlik hesaplanır (Hartley ve Zisserman, 2003). Böylesi çözümlerin hafıza gereksinimi ve hesaplama maliyeti bakımından pahalı çözümler olduğu bilinmektedir (Khan vd., 2020).

Algısal yaklaşım grubunda ise monoküler RGB görüntü ve bu görüntüye karşılık gelen görüntüyle eş boyutlardaki derinlik haritaları ile eğitilen DL modelleri bulunur. Bu derinlik haritalarındaki bilgi, piksel düzeyinde, RGB görüntünün çekildiği kameranın gerçek dünya objeleri arasındaki mesafeyi ifade eder. Söz edilen mesafenin değişim niceliği, RGB görüntü üzerinden doğrudan ya da türetilmiş olarak elde edilebilecek birçok özneliğin değişiminden yararlanılarak elde edilebilir. Bu öznelikler görüntü üzerindeki piksellerin farklı renk uzaylarındaki değer değişimleri, perspektif değişimleri, görüntüdeki objelerin şekil ve ölçek değişimleri, objelerin kameradan olan mesafeye bağlı rölatif hiyerarşisi, obje sınırları ve dokular üzerindeki keskinleşme ya da blurlaşma gibi bilgiler olarak örneklendirilebilir. Monoküler derinlik çıkarımı sağlayan DL ağları, eğitimleri sırasında öğrendikleri RGB görüntü ve derinlik haritası arasındaki ilişkiyi bu gibi öznelikler ile öğrenirler. Söz edilen öznelikler insanoğlunun derinlik algılamak üzere göz önünde bulundurduğu kriterler düşünülerek örnek verilmiştir. Ancak bir DL modelinin derinlik çıkarımı probleminde kullandığı özneliklerin tam olarak ne oldukları ve derinlik haritası oluşumuna hangi özneliğin ne derecede etkisi olduğu, çözümü DL modellerine bırakılmış birçok problemde olduğu gibi, modelin eğitim sürecine bırakılmıştır.

Bu çalışmada, dış mekan (outdoor) görüntülerinden monoküler derinlik çıkarımı işlemini gerçekleştirmek için, ilgili literatürde BTS ismiyle bilinen DL modelinin, KITTI veri seti ile eğitilmiş ön eğitilmiş bir versiyonu, bir web servisine dönüştürülerek kullanılmıştır (URL-4, 2021). KITTI, GPS, IMU, lidar ve kamera takımı donanımlarını taşıyan bir otomobil ile oluşturulmuş; dış mekanlarda çekilmiş RGB görüntüler ve bu görüntülerin çekim anındaki yönelme parametrelerini barındıran bir veri setidir. Yönelme

parametreleri kullanılarak stereo değerlendirme ile her bir görüntüye karşılık derinlik haritası üretilmektedir.

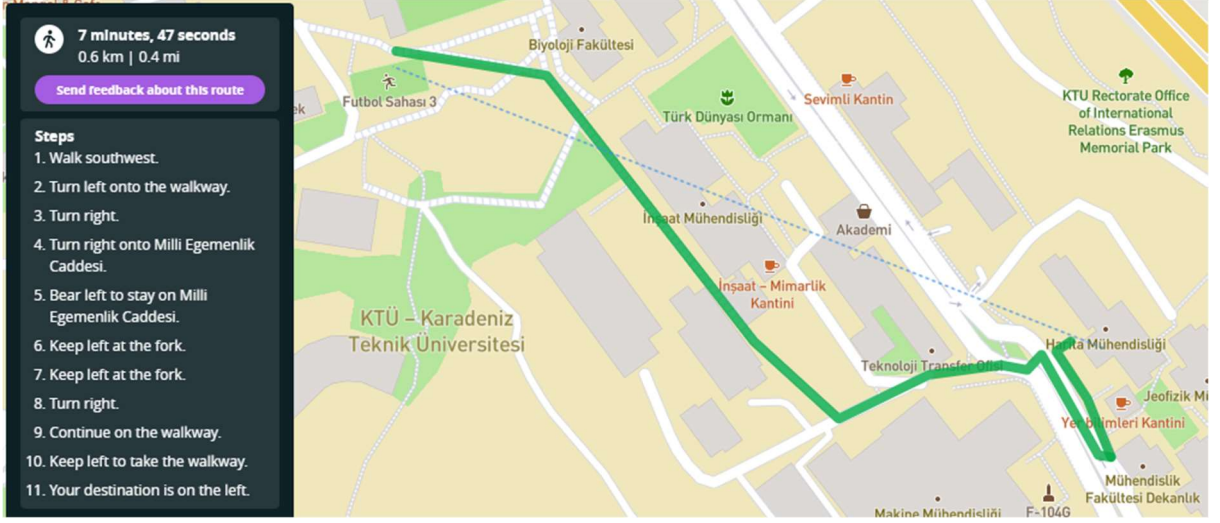
BTS modeli, doğruluk metrikleri ve koşum zamanına göre en yüksek performansı göstermiş iki DL modelinden biridir (Khan vd., 2020). BTS modeli bir encoding-decoding ardışık yapısına sahip DL modelidir. Encoding aşamasında girdi görüntü üzerindeki objelere ait öznitelikler çıkarılır ve decoding aşamasında bu öznitelikler kullanılarak görüntü üzerindeki her bir piksele ait derinlik bilgisi türetilir. BTS modelinin aynı amaca yönelik tasarlanmış DL modellerine kıyasla başarısı, girdi görüntü üzerindeki objelere ait özniteliklerle çıktı derinlik haritası arasındaki ilişkiyi emsallerine kıyasla daha etkili biçimde kurmasına dayanmaktadır. Bu üstünlüğü sağlayan yapı, decoding aşamasındaki “local planar guidance” katmanıdır. Bu katman sayesinde diğer DL modellerinden farklı olarak, encoding aşamasında elde edilen öznitelik haritasındaki özniteliklerin, decoding aşamasında derinlik haritası oluşturulması için upsample edilmesi (sık örnekleme - boyut yükseltme) sırasında oluşan uzamsal bilgi kaybının önüne geçilmektedir. Eş bir deyişle, görüntü keskinleştirmeye benzer bir yaklaşımla, keskinlik kaybının önüne geçen bir yapı ortaya konulmuş ve derinlik haritasındaki obje sınırları emsal modellere kıyasla daha yüksek hassasiyette oluşturulmuştur. Ayrıca bu işlem H boyutlarındaki bir görüntü için, H, H/2, H/4 ve H/8 olmak üzere 4 farklı çözünürlükteki öznitelik haritaları üzerinde uygulandığından dolayı derinlik haritası üzerinde farklı ölçekteki objelerin temsil edilmesi noktasında da model üstün performans sağlamaktadır (Akın ve Cömert, 2021) (Şekil 6).



Şekil 6. 4 farklı ölçekteki detaylar (sarı,yeşil,mavi,siyah) ve nihai derinlik haritası (kırmızı) (Lee vd., 2019)

1.5.3. Navigasyon

Navigasyon, bir hedef konuma ulaşmak amacıyla gerçekleştirilen yönlendirme işlemlerinin tümüdür. Bu çalışma kapsamında navigasyon görme engelli bireyin, kullandığı akıllı telefonun anlık konumuna ve anlık yönelimine (direction - heading) göre, belli bir varış noktasına sesli olarak yönlendirilmesi şeklinde gerçekleştirilmiştir. Navigasyona yönelik sunulan yazılım geliştirme araçları, bu çalışmada amaçlanan ya da benzeri uygulamalarda çevrimiçi harita altlıklarını kullanmaktadırlar. Bu harita altlıklarına örnek olarak Google Haritalar, Bing Haritalar, HERE ve bu çalışmada kullanılan kitle kaynaklı bir açık veri olan (crowdsourced open data) OpenStreetMap (URL-6, 2021) verilebilir. Harita altlıkları üzerinden ilgili geliştirme araçlarıyla, kullanıcıların anlık konumları ve ulaşmak istedikleri hedef coğrafi konum arasında en uygun rota hesaplanır (Şekil 7). Kullanıcının, mobil cihazın GPS alıcısı ve IMU sensörlerinden gelen anlık verinin değerlendirilmesiyle, en uygun rota üzerinde hedefe doğru ilerlemesi sağlanır.



Şekil 7. Örnek bir rota ve bu rotayı takip etmek için gerekli direktifler

Günümüzde mobil cihazların GPS alıcılarının kapalı mekanlarda sağladığı konum doğruluğunun henüz desimetre üzeri seviyede olmasından dolayı, navigasyonun kapalı mekanlardaki uygulanabilirliği ek donanım gerektirmektedir. Kapalı mekanlarda navigasyon, BLE (Bluetooth-Low-Energy) cihazları ya da Wi-fi erişim noktaları kullanılarak gerçekleştirilen geriden kestirmelerle, kullanıcı ortamda lokalize edilerek sağlanır (Kandalan ve Namuduri, 2020). Bunun yanı sıra, ilgili cihazların harita altlığı üzerinde konumlandırılabilmesi için, kapalı mekanların kat planlarının da harita altlığına işlenmesi gerekmektedir. Söz edilen yükümlülüklerden ötürü bu çalışmada geliştirilen uygulamanın navigasyon işlevi, GPS sinyalinin güçlü olduğu, dış mekanlarla sınırlı tutulmuştur.

1.6. Benzer Çalışmalar

Görme engelli bireylere yönelik çevreyle etkileşim ve navigasyon sağlayan akıllı telefonlar üzerine geliştirilmiş çözümlere hem akademik literatürde hem de yazılım sektöründe rastlanmaktadır. İlerleyen paragraflarda yazılım sektöründen akademik literatürden çalışmalar sırayla bu tez çalışmasında amaçlanan hedefler bağlamında irdelenmektedir. Yazılım sektöründen verilebilecek örnekler çoğaltılabilir olup, internet araştırmaları sonucunda ön plana çıkan uygulamalara irdelenmede yer verilmiştir. Bu çalışma

ile olabildiğince yakın çıktılar sağlayan akademik çalışmaların incelenmesine özen gösterilmiş ve bu çalışmalar 2017-2021 yılları arasında yayınlanmış çalışmalardan seçilmiştir.

Ülkemizde çeşitli bakanlıklarca yürütülmekte olan Görengöz (URL-7, 2021) projesine ait Navibaston isimli yazılım uygulaması akıllı telefonlar ile kullanılmaktadır. Konum servisi ile kullanıcının çevrimiçi harita altlıkları üzerinden sesli navigasyonunu sağlamaktadır. Aynı zamanda sesli olarak metin okuma özelliğine de sahiptir. Ancak uygulamada erişim kısıtlaması bulunmakta, yalnızca ilgili bakanlık tarafından uygun görülen vatandaşlara internet adresi üzerinden sunulmaktadır. Ayrıca proje başarı metrikleri, sistem mimarisini açıklayıcı bir yayın ya da kullanıcı geri dönüşleri gibi bilgilerin paylaşılmamış olması sebebiyle daha detaylı bir irdeleme yapılamamaktadır.

Ülkemizden bir başka girişim olarak WeWALK projesi (URL-8, 2021), ultrasonik sensör barındıran bir baston ile göğüs ve baş hizasında bulunan 0.8-2.5 m (metre) aralığındaki engellerin algılanıp kullanıcıya sesli bildirimini sağlamaktadır. Bunun yanı sıra bluetooth ile kullanıcının akıllı telefonuna bağlanarak konum servisleriyle çevrimiçi harita altlıkları üzerinden navigasyon sağlamaktadır. Projenin ek donanım yükümlülüğü getiriyor olması, kolay ulaşılabilirliğini ve ekonomikliğini sorgulatan bir etken olmaktadır.

TapTapSee (URL-9, 2021) isimli akıllı telefonlar ile ücretsiz olarak kullanılabilen bir başka uygulama, görme engelli kullanıcının çektiği fotoğraflar üzerinden obje tanıma ve obje türünün sesli geri bildirimini sağlamaktadır. CloudSight (URL-10, 2021) isimli API (Application Programming Interface - Uygulama Programlama Arayüzü) ile obje tanıma işlemi uzak sunucuda gerçekleşmekte, kullanıcıya 7-10 sn içerisinde geri bildirim verilmektedir. İşlemin uzak sunucuda gerçekleşmesi kullanıcı donanımında yük oluşturmamaktadır ancak servisin yanıt süresi göz önünde bulundurulduğunda, uygulamanın gerçek zamanlı bir kullanıma uygun olmadığı söylenebilir.

ARIANNA (pAth Recognition for Indoor Assisted NavigatioN with Augmented perception) (URL-11, 2021) (Gallo vd., 2013) isimli AR uygulaması, GPS sinyalinin yetersiz olduğu iç mekanlarda (indoor) görme engelli kullanıcıların navigasyonunu sağlamak üzere geliştirilmiştir. Bu uygulamada, iç mekanlarda zeminde işaretlenmiş QR kodlar ve QR kodların arasına çekilmiş renkli şeritlerin oluşturduğu yollar kullanılmaktadır. QR kod akıllı telefon kamerasıyla algılanıp uzak sunucuya gönderilmektedir. Uzak sunucuda QR kodun konum karşılığı bulunup kullanıcının hedef konuma ulaşması için takip etmesi gereken rota kullanıcıya gönderilmektedir. Kullanıcının bu rota üzerinde tutulması,

zemindeki renkli şeritlerin akıllı telefon kamerasının görüş alanında bulunduğu sürece kullanıcıya titreşimli geri bildirim verilmesi ile sağlanmaktadır.

ARIANNA uygulamasındaki zemin işaretçileri zorunluluğunu ortadan kaldıran ve ARIANNA+ olarak isimlendirilen bir çalışma sunulmuştur (Lo Valvo vd., 2021). iOS işletim sistemli cihazlarda ARKit (URL-12, 2021) AR kütüphanesi fonksiyonlarını kullanarak kullanıcının, zemin işaretçisi bulunmayan sanal yollar üzerinde navigasyonu sağlanmıştır. ARKit ve bu kütüphanenin Android işletim sistemindeki alternatifi olan ARCore (URL-13, 2021) gibi kütüphaneler, ToF ve LiDAR sensörlerini kullanarak kameranın algıladığı sahnenin fiziksel yapısının çıkarımını sağlamaktadırlar. ARIANNA+ çalışmasında ARKit sayesinde, sahnedeki zemin ve zemin üzerindeki yapıların ayrılması sağlanmış ve kullanıcının üzerinde engel bulunmayan yatay yer zemini üzerinde hareket etmesi sağlanmıştır. Ayrıca bu yeniliğe ek olarak uygulamaya, anlık görüntü üzerinden obje tespiti yeteneği eklenmiş ve bunun için SSD ve Inception (Szegedy vd., 2015) DL modellerini kullanmıştır. Uygulamanın ARKit kütüphanesi temeli üzerine geliştirilmesinin doğurduğu Apple firmasının belirli cihazlarını kullanma yükümlülüğü bu çalışmaya bir donanım bağımlılığı getirmektedir. Dolayısıyla kolay ulaşılabilir ve ekonomik olma açısından kısıt taşımaktadır.

Lin ve diğerleri (2017), görme engelli kullanıcıya karşısındaki engelin türünü ve engelle kullanıcı arasındaki mesafe-derinlik bilgisini, bu tez çalışmasına benzer bir yaklaşımla, sesli olarak sunan bir istemci-sunucu mimarisi geliştirmişlerdir. Bu çalışmaya göre kullanıcının akıllı telefon kamerasından alınan canlı görüntü akışı kareleri sunucuya gönderilmekte, bu görüntü kareleri sunucuda işlenerek obje türü ve mesafe bilgisi üretilmektedir. Objeye türü ve derinlik bilgisi metin veri formatında istemciye gönderilmekte ve istemci tarafında text-to-speech (metinden konuşma sesi üretimi, TTS) teknolojisi ile sese dönüştürülerek kullanıcıya sunulmaktadır. Objeye türü bilgisi sunucuda koşan YOLO ve Fast R-CNN (Ren vd., 2015) DL modelleri ile çıkarılmaktadır. Derinlik bilgisi çıkarımı için de kameranın odak ekseninin zemine paralel olduğu varsayımına dayanan bir üçgen benzerliği çözümü tercih edilmiştir. Çalışma, istemci-sunucu mimarisi üzerine kurulu oluşu ve anlık obje tespiti için DL modelleri kullanmasıyla birlikte, bu tez çalışmasında amaçlandığı gibi donanımdan bağımsız bir ürün olma niteliği taşımaktadır. Ancak derinlik çıkarımında tercih edilen yaklaşım, temelinde yatan varsayımın üretilen derinlik bilgisinin doğruluğunu canlı kullanımda olumsuz yönde etkileyebileceği için, riskli görülmüştür. Öyle ki, çalışmanın test

sonuçları incelendiğinde çıkarılan derinlik bilgisinin hata miktarının 10 m ve üzeri uzaklıklarda giderek arttığı görülmektedir.

Sun ve diğerleri (2019), görme engelli kullanıcının karşılaştığı engelleri 18 m²'lik bir alan içerisinde algılayarak titreşim ve ses ile geri bildirim veren bir çözüm geliştirmişlerdir. Çözümü Google ARCore' un selef teknolojisi olan Google Tango Project (URL-14, 2021) destekli akıllı telefonlar üzerinde geliştirmişlerdir. Google Tango destekli cihazlar kızılötesi (infrared - IR) derinlik sensörü barındıran cihazlar olup bu sensör sayesinde çevrenin fiziksel yapısını modelleyebilen cihazlardır. Google Tango destekli bir cihazla anlık olarak kullanıcının karşılaştığı engeller ile arasındaki mesafe bilgisi üretilip, kullanıcının titreşim ve sesli geri bildirimle kendisini bu engellerden sakınması sağlanmıştır. Açıklanan bu çözümün belirli donanımlara bağımlılığı söz konusudur. Ayrıca, Google firması 2018 yılında Tango projesini sonlandırıp ARCore projesini başlatmıştır. Dolayısıyla bu çözüm şimdiki haliyle sürdürülebilir değildir.

Bauer ve diğerleri (2020), görme engelli kullanıcının karşılaştığı engel türü ve engel-kullanıcı arası mesafe bilgisini anlık görüntü üzerinden DL modelleri ile üretilen ses ve titreşimle geri bildirim sağlayan bir istemci-sunucu mimarisi geliştirmişlerdir. İstemci tarafındaki donanım bileşenleri iki akıllı saat, bir akıllı telefon ve bir kablosuz cep kamerasından oluşmaktadır. Kablosuz kamera akıllı telefona Wi-fi ile bağlanmaktadır. Anlık kamera görüntüsü Wi-fi üzerinden akıllı telefona aktarılıp uzak sunucuya gönderilmektedir. Uzak sunucuda YOLO modeli ile engelin türü, (Laina vd., 2016) DL modeli ile de 0,5-20 m arası mesafe bilgisi üretilip istemciye gönderilmektedir. Bu bilgiler istemcide sese ve titreşime dönüştürülüp kullanıcıya sunulmaktadır. Söz edilen iki akıllı saat kullanıcıya titreşim gönderilmesinde kullanılmaktadır. Çalışma, istemci-sunucu mimarisi kurulması, engel türü-mesafe bilgilerinin DL modelleri ile üretilmesi yönleriyle bu tez çalışması ile yakın benzerlik göstermektedir. Ancak ulaşılabilirliği hayli kısıtlayan bir ek donanım yükümlülüğü bulunmaktadır. Öyle ki, akıllı telefonun kendi kamerası yerine harici bir kamera kullanılması hem maliyet fazlalığı hem de şarj edilmesi gereken ilave bir cihaz anlamına gelmektedir.

Yukarıdaki irdelemeler ışığında rahatlıkla söylenebilir ki, görme engellilere yönelik geliştirilen mobil çözümlerin tümü erişilebilirlik, ek donanım bağımlılığı, maliyet, anlık kullanım performansı ve üretilen bilgilerin doğruluğu açılarından birtakım kısıtlamalar içermektedirler. Bu tez çalışması sonucunda bu kısıtlamalardan arı bir yazılım uygulaması ortaya konulmaya çalışılmıştır. Dahası, görüntü üzerinden obje tespiti, görüntü üzerinden

derinlik çıkarımı ve çevrimiçi harita altlıkları üzerinden navigasyon işlevlerinin tümünün aynı uygulamada bir araya geldiği başka bir çalışmanın bulunmaması da bu çalışmanın özgünlüğünü arttırmaktadır



2. YAPILAN ÇALIŞMALAR

2.1. Materyaller

2.1.1. Veri Kaynakları

Görüntüden obje tespiti ve görüntüden derinlik çıkarımı sağlayan, YOLO ve BTS modellerinin koşturulduğu web servisleri için gerekli olan veri, akıllı telefon kamerasıyla alınan canlı videonun kareleridir. Bu görüntülerin edinilmesinde HUAWEI FIG-LX1 ve ASUS Zenfone 6 marka-model Android işletim sistemli telefonlarla çekilen görüntüler kullanılmıştır. HUAWEI FIG-LX1 cihazı 13 MP (Megapiksel) çözünürlüğünde 3.46 mm odak uzaklığında kameraya sahiptir. 4160x3120 çerçeve boyutlarında 30 fps hızında görüntü kaydı yapmaktadır. ASUS Zenfone 6 cihazı 13 MP çözünürlüğünde 3.78 mm odak uzaklığında kameraya sahiptir. 4096x2304 çerçeve boyutlarında 30 fps hızında görüntü kaydı yapmaktadır (URL-15, 2021).

Navigasyon için Mapbox Web Services API (URL-16, 2021) kullanılmıştır. Bu API Mapbox firmasının, çevrimiçi harita altlıkları üzerinden navigasyon ve geocoding gibi araçları sunduğu bir servisler bütünüdür. Mapbox altlık olarak, OpenStreetMap haritaları üzerine kendi ürettiği veriler ile katkıda bulunduğu altlıkları kullanmaktadır. OpenStreetMap, dünya genelinden katkıda bulunmak isteyen herkesin, hava fotoğrafları üzerinden bir haritaya konu olabilecek objeleri işleyebildiği ve işlenmiş objeleri düzenleyebildiği kolaboratif bir kitle kaynak verisidir. Geliştirici ve kullanıcılar için ücretsiz sunulmaktadır.

2.1.2. Kullanılan Geliştirme Ortamları ve Araçları

2.1.2.1. Unity Oyun Motoru

Unity, öncelikli olarak bilgisayarlar, konsollar ve mobil cihazlar için video oyunları ve simülasyonları geliştirmek için kullanılan çapraz platform bir oyun motorudur. C# programlama dilini kullanarak on yedi farklı platformda kullanabilen uygulamalar

geliştirmek mümkündür. Özellikle AR ve VR (virtual reality) uygulamalarında geliştiricilere birçok hazır kod ve oyun objesi sunmaktadır. Arayüz objeleri ve oyun objeleri ile kod betikleri (script) arasındaki ilişkiyi kolaylaştırmak adına sezgisel programlamayı olanaklı kılmaktadır. Ayrıca kullanıcı ve geliştiricileri etkileşim içinde tutmak ve yardımlaşmalarını sağlamak için zengin açık kaynak depoları sunmaktadır (URL-17, 2021). Bu çalışma kapsamında istemci uygulaması geliştirilmesi için kullanılmıştır.

2.1.2.2. Anaconda

Anaconda, Python programlama diline yönelik geliştirilmiş bir kütüphane ve çevre yönetim platformudur. Platformun odak noktasında veri bilimi çalışmaları bulunmaktadır (URL-18, 2021). Anaconda içerisinde erişilen Python ekosistemindeki tüm bileşenler sunucu uygulamalarının, web servislerinin, geliştirilmesi için kullanılmıştır.

2.1.2.3. Spyder

Anaconda içerisinde sunulan bir kod editörüdür. Bilimsel hesaplamalar için geliştirilmiştir. Bilimsel hesaplamalarda tercih edilmesinin başlıca sebebi diğer editörlerden farklı olarak, bir Python betiği oluşturulduğunda oluşan ara değişkenleri değişken gözlemcisi aracıyla (variable explorer) gözlemlenebilir kılmasıdır (URL-19, 2021). Bu çalışmada sunucu uygulamasındaki betiklerin düzenlenmesinde kullanılmıştır.

2.1.2.4. Visual Studio Code

Bu çalışmada Unity oyun motorundaki geliştirme aşamalarında C# betiklerinin derlenmesinde kullanılan, Microsoft firmasının sunduğu ücretsiz bir kod editörüdür. C# dilinin kütüphane ve çevre yönetim platformu olan .NET ve Unity' ye yönelik eklentileri bulunmaktadır (URL-20, 2021).

2.1.2.5. Flask

Flask, web uygulamaları geliştirilmesinde kullanılan bir Python kütüphanesidir. Bu uygulamalar web siteleri, API'ler ya da bu çalışmada olduğu gibi web servisleri olabilir (URL-21, 2021).

2.1.2.6. OpenCV

OpenCV, açık kaynaklı bir bilgisayarlı görü projesidir. Python da dahil olmak üzere birçok programlama diline ait uygulamalara kütüphane olarak eklenebilmektedir. Birçok bilgisayarlı görü (computer vision - CV) algoritması ve bu algoritmalara altlık teşkil eden görüntü işleme fonksiyonlarını bünyesinde barındırmaktadır. Gelişen bilgisayarlı görü tekniklerine göre güncellenmeye devam etmektedir (URL-22, 2021). Bu çalışmada, kamera ile alınan görüntülerin web servislerinde işlenmeye hazır hale getirilmesinde kullanılmıştır.

2.1.2.7. PyTorch

Pytorch, açık kaynak kodlu bir Python DL kütüphanesidir (Paszke vd., 2019). Birçok DL mimarisini ve yeni DL mimarileri oluşturmak için gerekli olan yapıları geliştiricilere sunmaktadır. Diğer DL kütüphanelerine kıyasla düşük düzey (low-level) bir kütüphane olduğu için mimari oluşturmak amacıyla kullanılan fonksiyonların geliştirici tarafından modifiye edilmesine daha yüksek düzeyde olanak sunmaktadır (URL-23, 2021). Ayrıca GPU çekirdekleri arasında işlem paralelleştirme işlemi diğer kütüphanelere göre, kodlama yükü bakımından, PyTorch ile daha kolaydır. Bu çalışmada, sunucu uygulamasındaki YOLO ve BTS modellerinin yönetiminde kullanılmıştır.

2.1.3. Kullanılan Donanım

Test aşamalarında sunucu olarak ve tüm geliştirme süreçlerinde DELL Precision T3640 model iş istasyonu kullanılmıştır. İş istasyonunun sistem konfigürasyonu Tablo-1’deki gibidir.

Tablo 1. İş istasyonu sistem konfigürasyonu

Kasa / Çipset	300 Watt (%85) Dikey Kasa, Intel PCH-H W48
İşlemci	Intel Xeon W-1250 (6 Core) (3.30GHz, 4.70GHz Turbo, 12MB)
RAM	16GB DDR4 2933MHz ECC (2x8GB) (maksimum 128GB)
GPU	nVidia Quadro P2200 (5GB GDDR5x, 160bit, Pro 3D)
Sabit Disk	512GB M.2 PCIe NVMe SSD + 1TB (7.200rpm) SATA HDD
İşletim Sistemi	Windows 10 Pro (64 bit)

İstemci uygulamasının koşacağı test cihazı olarak Asus Zenfone 6, HUAWEI FIG-LX1 model orta segment ve altı olarak nitelendirilebilecek Android akıllı telefonlar kullanılmıştır. Farklı donanım özellikleri ve farklı işletim sistemi sürümlerindeki cihazlar olmaları sebebiyle seçilmişlerdir. Bu cihazların konfigürasyonları Tablo-2 deki gibidir.

Tablo 2. Akıllı telefonların sistem konfigürasyonları

	Asus Zenfone 6	HUAWEI FIG-LX1
CPU	Intel Atom(x86)-1.6 GHz	Kirin(x64)-2.36 GHz
Depolama	16 GB	32 GB
Android API (Versiyon)	21 (Android 5.0)	28 (Android 9.0)
RAM Boyutu	2 GB	3 GB
GPS	Var	Var
Jiroskop Sensörü	Var	Var
İvmeölçer Sensörü	Var	Var
Batarya	3300 mAh	3000 mAh
Ana Kamera	13 MP	13 MP

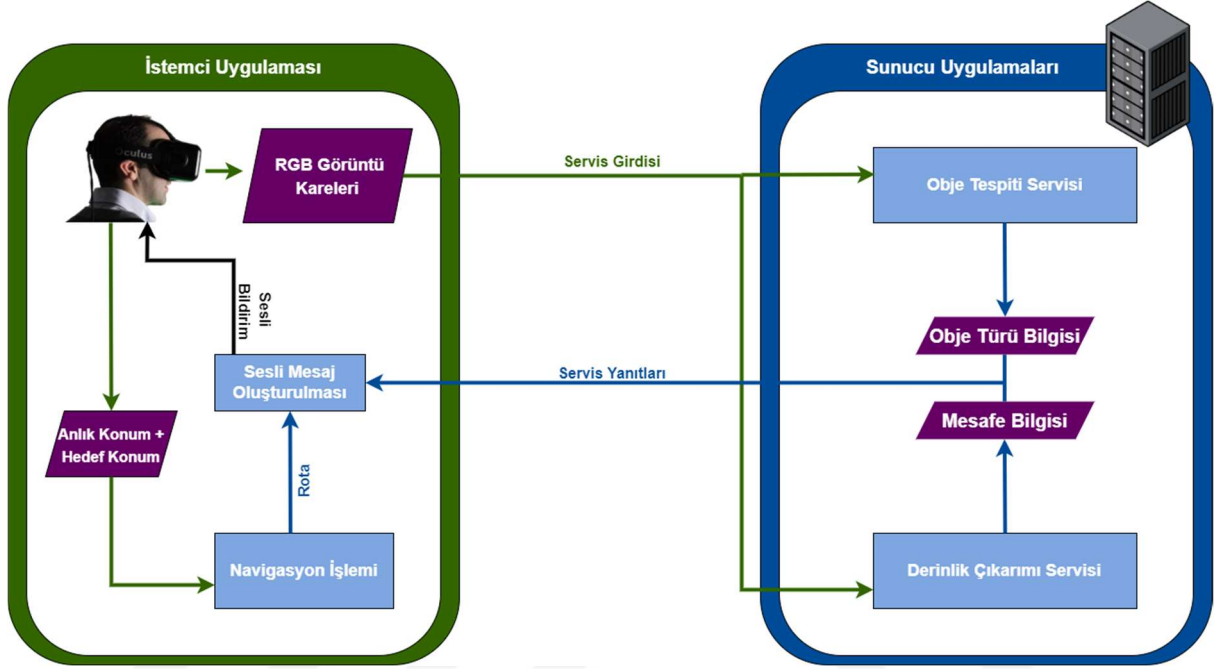
2.2. Yöntem

2.2.1. Genel Uygulama Mimarisi

Belirtildiği üzere, görme engellilere yönelik bir AR uygulaması geliştirilmiştir. Uygulama girdi olarak akıllı telefon kamerasının çektiği canlı video akışının karelerini almaktadır. Bu girdiye karşılık kullanıcının karşılaştığı objenin türünü ve objeyle aradaki mesafeyi üretmektedir. Üretilen bu bilgiler kullanıcıya sesli olarak sunulmaktadır. Bunun yanı sıra, çevrimiçi harita altlıkları üzerinden sesli navigasyon sağlamaktadır. Uygulamanın koşaacağı telefon kullanıcının göz hizasına bir VR gözlük ya da benzeri bir sabitleyiciyle ya da göğüs bölgesine telefon kamerasını açıkta bırakacak şekilde bir kuşakla sabitlenmektedir.

Uygulama mimarisi istemci uygulaması ve sunucu uygulamaları olmak üzere iki temel bölümden oluşmaktadır. İstemci uygulamasında görüntü karelerinin sunucuya internet bağlantısı üzerinden gönderilmesi, sunucudan gelen yanıtların sese dönüştürülmesi ve

navigasyon işlemleri gerçekleştirilmektedir. Sunucu tarafında ise görüntüden obje tespiti ve derinlik çıkarımı sağlayan DL modelleri birer web servisi olarak koşturmaktadır. Gelen görüntü karelerinden üretilen obje türü ve mesafe bilgileri istemciye gönderilmektedir. Şekil 8’de geliştirilen çözümün genel mimarisi verilmiştir.



Şekil 8. Uygulamanın Genel Mimarisi

2.2.2. İstemci Uygulaması

İstemci uygulamasının geliştirilmesi için Unity oyun motoru kullanılmıştır. Bölümün devamında tarif edilen işlemlerin yerine getirebilmesi için gereken fonksiyonlar ve kullanıcı arayüzü C# programlama dili ile kodlanarak, test cihazlarında kullanım için .apk uzantılı Android uygulamasına dönüştürülmüştür. Uygulama web servisleriyle çalıştığı için yüklendiği cihazda mobil veri kullanımına izin verilmesi gerekmektedir.

Uygulama kullanıcı cihazına ilk yüklendiğinde kullanıcıdan metrik birimde boy bilgisi istenmektedir. Bunun sebebi derinlik çıkarımı servisinden dönen yanıtın kullanıcı boyunun %43'üne bölünerek adım birimde kullanıcıya döndürülüyor olmasıdır. Çünkü bir adım, insan boyunun yaklaşık %43'üne karşılık gelmektedir (URL-24, 2021). Eğer kullanıcı

biliyorsa, adım uzunluğu da girdi olarak alınabilmektedir. Doğuştan görme engelli bir bireyin zihninde metrik birimler bir çağrışım yapamayacağından literatürdeki derinlik çıkarımı sağlayan diğer çalışmalarda da mesafe bilgisi titreşime ya da “yakın-orta-uzak” gibi mesajlara dönüştürülerek görme engelli kullanıcıya sunulmuştur. Bu çalışmada adım birimde sesli yanıt verilmesi tercih edilmiştir.

Uygulama koşturulduğunda cihaz kamerası direkt olarak açılmakta ve canlı görüntü kareleri ilgili web servislerine internet bağlantısı üzerinden gönderilmeye başlamaktadır, bunun için bir kullanıcı istemi gerekmemektedir. Kamera ile alınan canlı görüntü kareleri sunucudaki obje tespiti ve derinlik çıkarımı web servislerine, ilgili web servisinde bulunan DL modelinin giriş boyutlarına göre OpenCV kütüphanesiyle yeniden boyutlandırılarak aynı anda gönderilmektedir. Bu servislere görüntü karesinin yanı sıra görüntü orta noktasının indisi, ekranda tanımlanan sanal şütörün piksel biriminde koordinatları, de gönderilmektedir. Bu işlem kullanıcının tam karşısındaki, sanal şütörün isabet ettiği, objeye dair obje türü ve mesafe bilgisinin görüntünün tümünden üretilen bilgiden filtrelenip istemciye gönderilmesi için yapılmaktadır. Görüntü alımının 30 fps hızında olması ve bir görüntü üzerinde aynı anda sesli mesaja dönüştürülemeyecek kadar fazla objenin tespit edilebileceği sebeplerinden ötürü, kullanım hızının iyileştirilmesi adına, bu filtrelemeye başvurulmuştur. Bu filtrelemenin ardından elde edilen servis yanıtları istemci tarafında sesli mesaja dönüştürülerek kullanıcıya sunulmaktadır. Kullanıcının kendisine hızla yaklaşan bir obje olması, kişisel alan ihlali ya da sosyal mesafe ihlali gibi durumlarda kendini korumaya alabilmesi için, eğer ki derinlik çıkarımı servisiyle görüntü üzerinde kullanıcıya belirli bir eşik değerden daha yakın bir mesafe değeri saptanırsa, filtreleme devre dışı bırakılmaktadır. Bu eşik için 1.83 m olan sosyal mesafe değeri seçilmiştir (URL-25, 2021).

Kullanıcının navigasyon işlevini kullanabilmesi için uygulamayı cihazına ilk yüklediğinde mobil veri kullanımının yanı sıra konum servislerinin de aktifleştirilmesine izin vermesi gerekmektedir. Navigasyonun başlatılabilmesi için öncelikle kullanıcının varmak istediği hedefi uygulamaya sesli olarak vermesi gerekmektedir. Kullanıcı uygulama arayüzünde navigasyonu aktif hale getirdiğinde kullanıcıdan ses girdisi alınmaktadır. Bu ses girdisi Unity eklenti mağazasından edinilebilen speech-to-text eklentisi ile metne dönüştürülmektedir (URL-26, 2021). Metin formatında elde edilen bu hedef bilgisi Mapbox Geocoding API ile enlem-boylam bilgisine dönüştürülmektedir. Kullanıcının anlık konumu ile hedef konum arasındaki en uygun rota Mapbox Directions API ile edinilmektedir. Directions API’den yanıt olarak gelen JSON (JavaScript Object Notation) dosyası en uygun

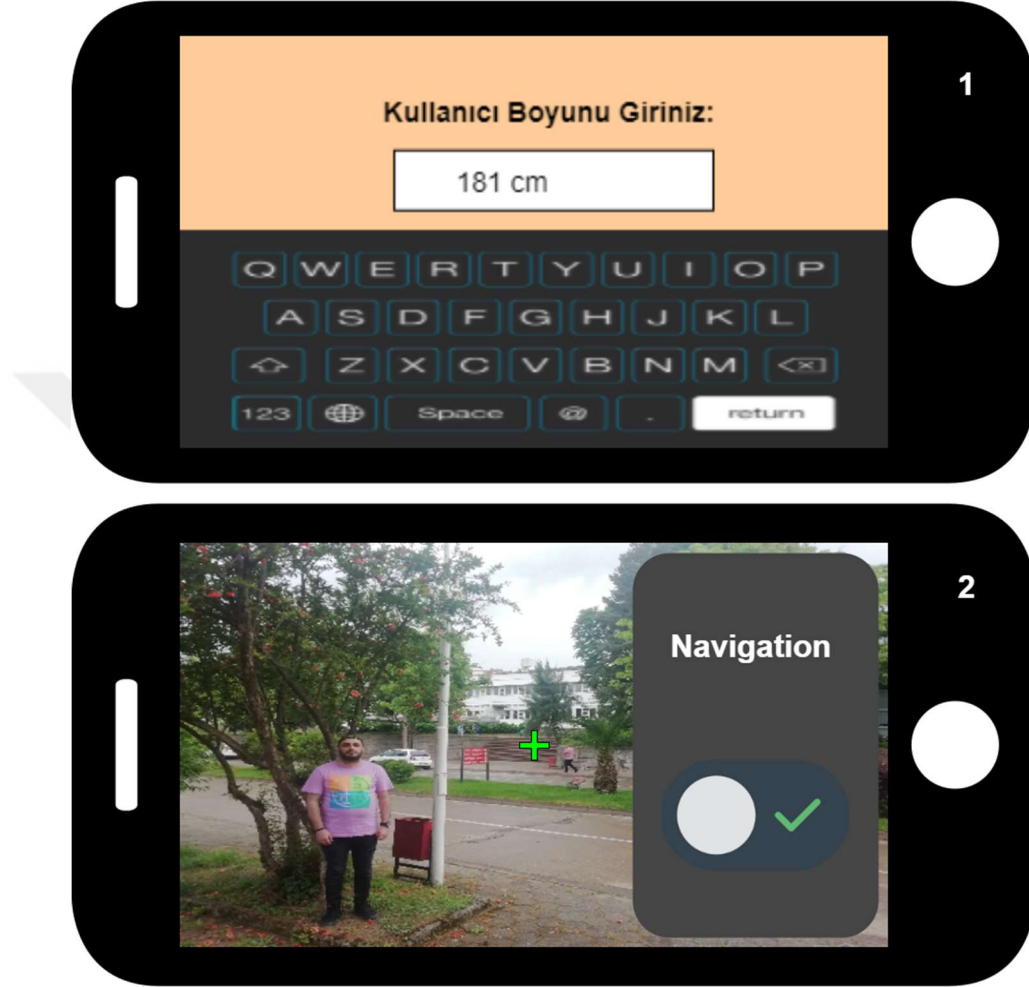
rotayı takip etmeyi gerektiren talimatları ve bu talimatların ilişkilendirildiği coğrafi konumları içermektedir. Kullanıcının anlık konumuna göre ilgili talimat sese dönüştürülerek kullanıcı hedef konuma ulaştırılmaktadır. Directions API yanıtı olan örnek bir JSON dosyası Şekil 9’ da verilmiştir.

```

JSON
  routes
    0
      weight_name : "pedestrian"
      weight : 576.072
      duration : 509.145
      distance : 662.065
      legs
        0
          via_waypoints
          admins
            weight : 576.072
            duration : 509.145
            steps
              0
                intersections
                maneuver
                  type : "depart"
                  instruction : "Walk southeast on the walkway."
                  bearing_after : 147
                  bearing_before : 0
                location
                  0 : 39.775273
                  1 : 40.994396
                name : ""
                duration : 39.834
                distance : 52.648
                driving_side : "right"
                weight : 42.218
                mode : "walking"
              1
                intersections
                maneuver
                  type : "turn"
                  instruction : "Turn right."
                  modifier : "right"
                  bearing_after : 279
                  bearing_before : 165
                location
                  0 : 39.775545
                  1 : 40.99397
                name : ""
                duration : 4.93
                distance : 7
  
```

Şekil 9. Directions API yanıtı örneği. Talimatlar (mavi kutucuk) ve bu talimatların ilişkilendirildiği coğrafi konumlar (kırmızı kutucuk).

Yukarıda bahsedilen tüm servis yanıtlarının sese dönüştürülmesinde Unity eklenti mağazasından edinilebilen text-to-speech eklentisi kullanılmıştır (URL-27, 2021). İstemci uygulamasının arayüz tasarımına Şekil 10'dan ulaşılabilir.



Şekil 10. Uygulama arayüzü (1 - İlk çalıştırmada boy girişi; 2 - Uygulamanın koşumu esnasındaki arayüz, navigasyon işlemini açma-kapama butonu ve sanal şütör)

2.2.3. Web Servislerinin Geliştirilmesi

Objeye tespiti ve derinlik çıkarımı sağlayan YOLO ve BTS modelleri farklı birer Flask fonksiyonuna dönüştürülmüştür. Modellerin kullanılmasında ve GPU üzerindeki hafıza yönetiminde Pytorch kütüphanesi kullanılmıştır. Modellerin dönüştüğü fonksiyonlar Python'a özgü decorator yapıları sayesinde birer URL ile ilişkilendirilmiş ve web servisine

dönüştürülmüşlerdir (Şekil 11). Decorator yapısı sayesinde istemcinin istek gönderdiği URL'ye atanan Flask fonksiyonu aktif olup istemciye yanıt göndermektedir. Bu fonksiyonların bulunduğu sunucu uygulamasının koşturulduğu andan itibaren servisler istemciden istek alabilir duruma gelmektedirler. Öyle ki, sunucu uygulaması koşturulduğunda modellerin bağlantılı çizge yapıları öğrenilmiş parametrelerle birlikte birer girdi alabilen Pytorch modeline dönüşmektedirler. Sunucu uygulaması kodlarına <https://github.com/alpertungakin/DepthandRecogApp> adresinden erişilebilir.

```
@app.route("/pass4depth", methods = ["POST"])
def depthExtract():

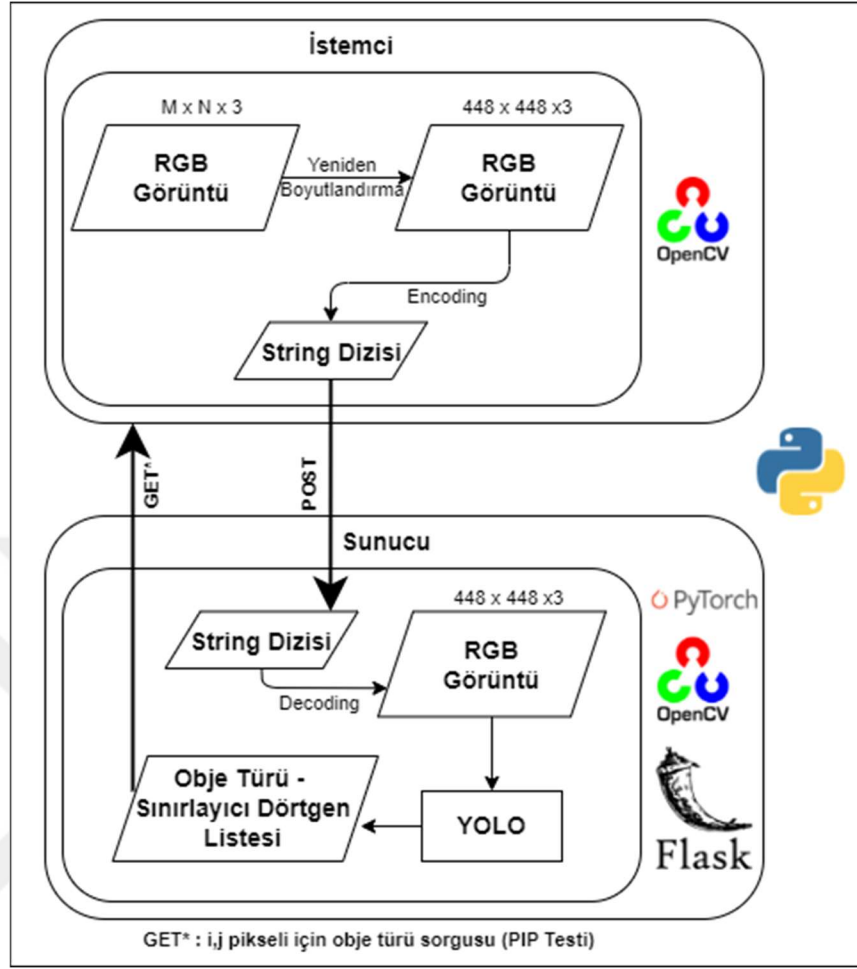
@app.route("/pass4detection", methods = ["POST"])
def objectDetection():
```

Şekil 11. Decorator yapıları ve Flask fonksiyonları. (depthExtract() fonksiyonunun çalışması için istemcinin "...endpoint.../pass4depth" adresine görüntü POST etmesi gerekmektedir.)

2.2.3.1 Obje Tespiti Web Servisi

Anlık görüntü akışının kareleri istemci tarafında öncelikle YOLO modelinin giriş boyutlarına (448x448x3) göre yeniden boyutlandırılmaktadır. Ardından yeniden boyutlandırılan görüntü karesi encode edilmekte, yani bir string dizisi olarak ikili dosyaya dönüştürülmektedir. Bu ikili dosya ve görüntü orta noktası indisi paketlenerek obje tespiti servisine POST metodu ile gönderilmektedir.

Obje tespiti servisine POST edilen bu paket, ikili dosya ve indis olarak sunucuda geri ayrıştırılmaktadır. Bu ikili dosya decode edilip, yeniden RGB görüntüye dönüştürülerek YOLO modeline girdi olarak verilmektedir. YOLO modeli ise bu görüntüden, tespit ettiği objelerin türlerini ve sınırlayıcı dörtgenlerini (bounding box) içeren bir liste üretmektedir. Bu liste içerisinde, kullanıcının odaklandığı objenin türünün seçilmesi için, görüntü orta noktasının indisi listedeki sınırlayıcı dörtgenlerle PIP (Point-In-Polygon) testine tabi tutulmaktadır. PIP testi sonucu "True" dönen sınırlayıcı dörtgenin içerisindeki objenin türü metin veri olarak GET metodu ile istemciye gönderilmektedir. Bu servisin iş akışı Şekil 12'de görülmektedir.



Şekil 12. Objeler tespiti web servisi iş akışı

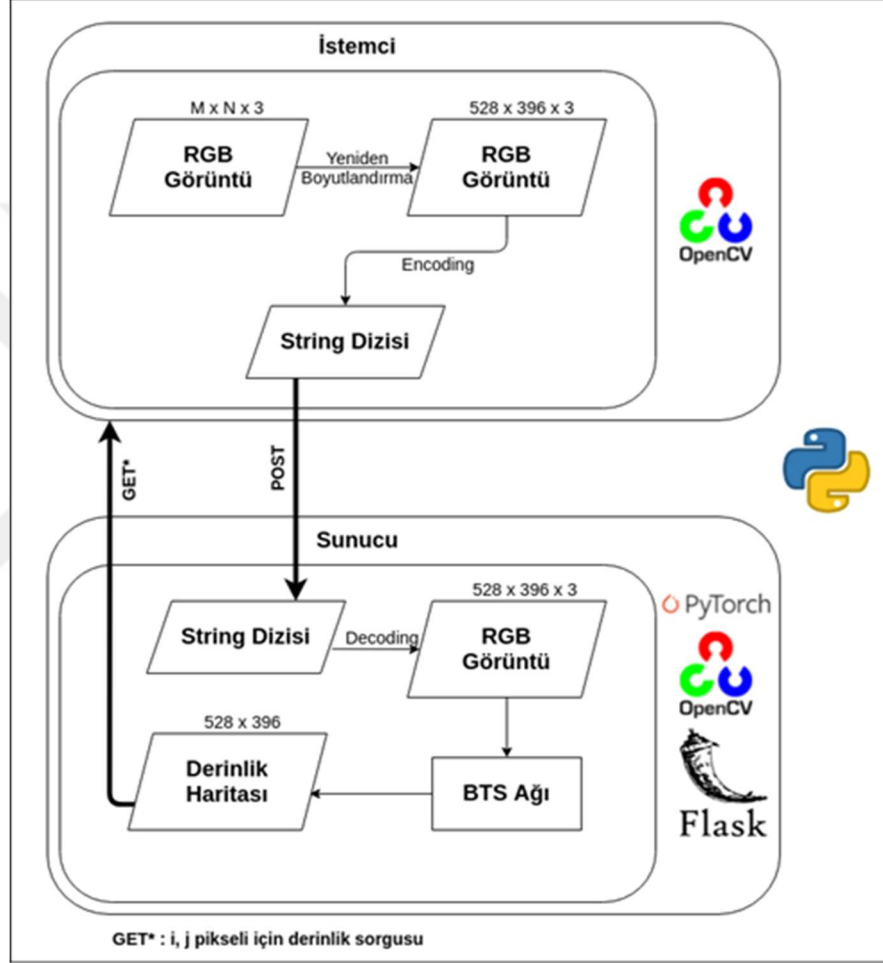
2.2.3.2. Derinlik Çıkarımı Web Servisi

İstemci tarafında görüntü kareleri öncelikle, boyutların düşürülmesi suretiyle, yeniden boyutlandırılmaktadır. Görüntünün düşürüldüğü yeni boyut, veri aktarım hızı, BTS modelinin yanıt süresi ve BTS modelinin giriş boyutları arası oran (sıra/sütun) göz önünde bulundurularak deneysel belirlenmiştir. Yeniden boyutlandırılan görüntü karesi encode edilmektedir. Encode işlemi sonucunda oluşan ikili dosya, derinlik sorgusu yapılacak görüntü orta noktasının indis bilgisi ile birlikte paketlenerek derinlik çıkarımı servisine POST metodu ile gönderilmektedir.

Derinlik çıkarımı servisine POST edilen bu paket, ikili dosya ve indis olarak sunucuda geri ayrıştırılmaktadır. İkili dosya decode edilip BTS modeline görüntü girdisi

olarak verilmektedir. BTS modeli ise bu girdiye karşılık bir derinlik haritası üretmektedir. Derinlik haritası üzerinden alınan, sunucuya gelen indis bilgilerine karşılık gelen metrik birimdeki derinlik bilgisi-mesafe, GET metodu ile istemci tarafından çekilmektedir.

Hem derinlik çıkarımı hem de obje tespiti servislerinde görüntünün yeniden boyutlandırılması ve encode-decode işlemleri OpenCV ile gerçekleştirilmiştir. Derinlik çıkarımı web servisinin iş akışına Şekil 13' dan ulaşılabilir.



Şekil 13. Derinlik çıkarımı web servisi iş akışı (Akın ve Cömert, 2021)

3. BULGULAR VE TARTIŞMA

Derinlik çıkarımı servisi için belirtildiği üzere BTS modeli ön eğitilmiş olarak edinilip kullanılmıştır. Ancak böylesi bir uygulamada, engel bildirimini doğruluk düzeyi görme engelli bir birey için hayati önem arz edebileceğinden, BTS modeli için akıllı telefon kamerası görüntüleri üzerinden doğruluk metrikleri üretilip irdelenmiştir. Bunun için, Mean Absolute Error (Ortalama Mutlak Hata-MAE), Root Mean Square Error (Karesel Ortalama Hata-RMSE), Square Relative Error (Karesel Bağıl Hata-SqRel) metrikleri kullanılmıştır. Bu metrikler şu şekilde tanımlanır:

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N \left(\frac{d_i - d'_i}{d_i} \right)^2} \quad (1)$$

$$MAE = \frac{1}{N} \sum_{i=1}^N |d_i - d'_i| \quad (2)$$

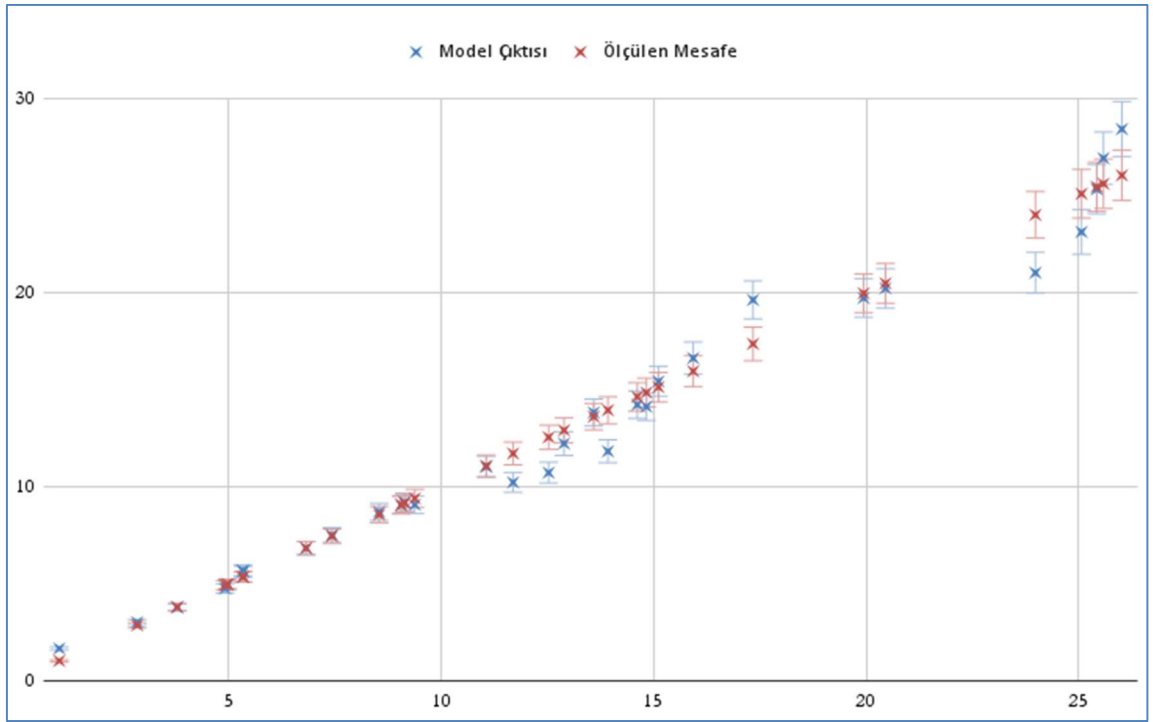
$$SqRel = \frac{1}{N} \sum_{i=1}^N \left(\frac{|d_i - d'_i|^2}{d_i} \right) \quad (3)$$

burada d_i ölçülen, d'_i ise model ile çıkarılan derinlik ve N test objesi sayısıdır.

KTÜ Merkez Kampüsü içerisinde Huawei FIG-LX1 akıllı telefon ile 5 adet RGB dış mekan görüntüsü çekilmiştir. Görüntülerin çekildiği noktalardan, görüntülerde kadraja girmiş 31 adet objeye elektronik uzaklık ölçer cihaz ile eğik mesafe ölçümü yapılmıştır. Objelerin tümünün cihaz kamerasına uzaklığı 1 - 28 metre aralığındadır. Bu şekilde test veri seti oluşturulmuştur. Elektronik uzaklık ölçer ile eğik mesafe okuması yapılan 31 objenin test görüntüleri üzerindeki indisleri ile okunan mesafeler ilişkilendirilmiştir. Ardından test görüntülerinin sırayla, derinlik çıkarımı servisi aracılığıyla, derinlik haritaları edinilmiştir. Bu derinlik haritaları üzerinden test objelerinin indislerine karşılık gelen derinlik bilgileri çıkarılmış ve sahada ölçülen eğik mesafelerle kıyaslanmıştır (Tablo-3). Tablodaki “d”, objelerin kameraya olan mesafelerini belirtmektedir. Ölçülen mesafe ve servis üretimi mesafe arasındaki farkın ölçülen mesafenin artışına göre gözlemi Şekil 14’te verilmiştir. RGB test görüntüsü örneklerine karşılık üretilen derinlik haritalarına Şekil 15’ten erişilebilir (Akın ve Cömert, 2021).

Tablo 3. Derinlik çıkarımı servisi doğruluk metrikleri (Akın ve Cömert, 2021)

Mesafe (d)	MAE	RMSE	SqRel
d<10	0,173	0,245	0,036
d<15	0,465	0,757	0,063
d<20	0,530	0,843	0,067
d>20	0,718	1,107	0,080



Şekil 14. Ölçülen Mesafe - Servis Üretimi Mesafe (kırmızı: ölçülen, mavi: servis üretimi) (Akın ve Cömert, 2021)



Şekil 15. Test görüntülerine karşılık derinlik haritaları (Akin ve Cömert, 2021)

Elde edilen metrikler ışığında rahatlıkla söylenebilir ki, derinlik çıkarımı servisinin doğruluğu verilen literatür bağlamında tatmin edicidir. Fakat doğruluk metrikleri performans değerlendirmesi için başlı başına yeterli değildir. Derinlik çıkarımının çalışma menzili ve web servisinin yanıt süresi de bu değerlendirmede göz önünde bulundurulmalıdır. Derinlik çıkarımı servisinin yanıt süresi 140 ms dolaylarında gözlemlenmiş ve çalışma menzili 20 m olarak belirlenmiştir. Yanıt hızı ve çalışma menzili bakımından da literatürdeki diğer çalışmalara kıyasla derinlik çıkarımında yüksek performans sağlanmıştır. Ancak şunu belirtmek gerekir ki, 140 ms'lik yanıt süresi yerel bir sunucu üzerinden gözlemlenmiştir.

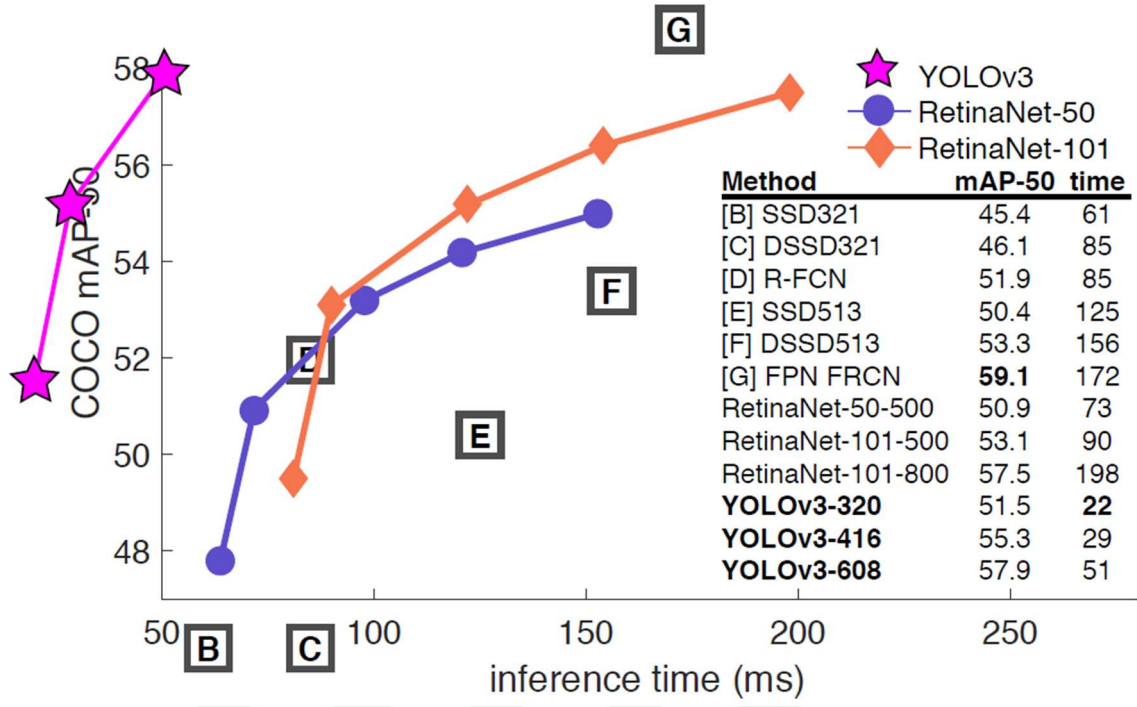
Gerçek kullanıcıların hücrenel veri bağlantısıyla edinecekleri deneyimlerde bu süre bağlantı kalitesine göre farklılık gösterebilir.

Objeye tespiti servisi için doğruluk metrikleri aşağıda verilmiştir. Metriklerin sunumunda görme engelli bir kullanıcının dışarıda karşılaşabileceği COCO veri seti içerisindeki en yaygın objelere ilişkin YOLO modelinin sunduğu doğruluk metrikleri kullanılmıştır. Bu objeler insan, araba, otobüs, motosiklet, trafik ışığı, trafik işaretçisi, bisiklet olarak belirlenmiştir.

Tablo 4. YOLO performans tablosu (URL-28, 2021)

Objeye Sınıfı	İlgili Sınıf Etiketine Sahip Eğitim Görüntüsü Sayısı	mAP (Mean Average Precision, Ortalama Hassasiyet) (%)
İnsan	254	77.4
Bisiklet	6	61.4
Araba	46	48.6
Motosiklet	5	79.1
Otobüs	7	72
Trafik Işığı	14	24
Trafik İşaretçisi	2	82.8

Şekil 16'da YOLO modelinin benzer amaçla kullanılan DL modelleriyle bir kıyaslaması verilmiştir. Model tahmin hızı ve mAP dengesine göre YOLO diğer modellerin önüne geçmektedir. Bu çalışmada YOLO modelinin koştugu web servisinin yanıt hızı 27 ms olarak gözlemlenmiştir.



Şekil 16. YOLO ve benzer modellerin kıyaslaması (Redmon ve Farhadi, 2018).

Literatür araştırmasında sunulan görme engellilere yönelik çalışmalar ile bu tez çalışmasının üretilen çıktılar ve sağlanan performans başarımları bağlamındaki kapsamlı kıyaslamasına Tablo-4'ten ulaşılabilir. Tablodaki "-", ilgili çalışmada söz konusu özelliğin bulunmadığını ya da o özelliğe ilişkin verinin paylaşılmadığını, "+" bulunduğunu, "?" ise birtakım kısıtlamalar ile bulunduğunu ifade etmektedir. Bu kısıtlamalar "1.6. Benzer Çalışmalar" başlığı altında açıklanmıştır.

Tablo 5. Çalışma kapsamında geliştirilen uygulamanın benzer çalışmalarla kıyaslanması

	Göregöz	WeWALK	TapTapSee	ARIANNA	Lo Valvo vd., 2021	Lin vd., 2017	Sun vd., 2019	Bauer vd., 2020	Geliştirilen Uygulama
Obje Tespiti	-	-	+	-	+	+	-	+	+
Derinlik Çıkarımı	-	+	-	-	-	?	+	+	+
Harita Altlıkları Üzerinden Navigasyon	+	+	-	-	-	-	-	-	+
Indoor Navigasyon	-	-	-	+	+	-	-	-	-
Donanım Bağımlılığı	-	+	-	?	+	-	+	+	-
Obje Tespiti Doğruluk Bilgisi	-	-	-	-	0.90	0.60	-	0.74	0.63
Derinlik Çıkarımı Hatası (RMSE) (m)	-	-	-	-	-	-	-	0.672	0.843
Derinlik Çıkarımı Menzili Bilgisi (m)	-	2.50	-	-	-	10.00	2.00	20.00	20.00
Obje Tespiti Hızı Bilgisi (ms)	-	-	10000	-	-	-	-	25?	27
Derinlik Çıkarımı Hızı Bilgisi (ms)	-	-	-	-	-	-	-	55?	140

(Bauer vd., 2020) çalışmasında kullanılan WiFi bağlantılı kamera ile akıllı telefon arasındaki görüntü aktarımında 81 ms' lik bir sürenin geçtiğinin belirtilmek gerekmektedir. Bu durumda, hücresel veri bağlantısıyla sunucuya görüntü aktarımı göz ardı edilirse, derinlik çıkarımı servisinin hızı yaklaşık 136 ms, obje tespiti servisinin hızıysa 106 ms olarak değerlendirilebilir.

Tablo-4'ten de anlaşılacağı üzere, barındırılan işlev sayısı ve bu işlevlere ilişkin performans metrikleri bağlamında bu tez çalışmasında geliştirilen uygulama literatürdeki diğer çalışmalardan üstün konumdadır. Öyle ki, obje tespiti, derinlik çıkarımı ve navigasyon işlevlerinin tümünü bünyesinde barındıran ve bu işlevlere yönelik daha iyi performans metrikleri sunan bir çalışmaya literatürde rastlanmamıştır.

4. SONUÇ VE ÖNERİLER

Bu tez çalışmasında görme engelli bireylerin yaşam kalitelerini artırmaya yönelik bir yardımcı AR uygulamasının geliştirilme süreci açıklanmıştır. Uygulama anlık telefon kamerası görüntüsü üzerinden obje tespiti, engel bildirim-i-derinlik çıkarımı sağlamakta; bununla birlikte çevrimiçi harita altlıkları üzerinden navigasyon sağlamakta ve bu çıktıları sesli geri bildirimle görme engelli kullanıcıya sunmaktadır. Bu işlevleri yerine getirirken DL tekniklerinin problem çözme yeteneklerinden faydalanmaktadır. Hesap yükü ve depolama alanı bakımından oldukça maliyetli DL modellerinin web servislerine dönüştürülerek kullanılması sayesinde uygulama, kullanıcı cihazında bir yük oluşturmadan orta segment ve altı cihazlarda dahi sonuç üretebilen kolay ulaşılabilir, ekonomik bir çözüm niteliği kazanmıştır.

Uygulamanın ilerleyen süreçte görme engelli gerçek kullanıcılar tarafından test edilmesi sağlanacaktır. Kullanıcılardan sağlanan geri dönüşlere göre kullanım tecrübesi iyileştirmeleri yapılacaktır. Bunun yanı sıra kullanıcı geri dönüşleri uygulamaya eklenmesi muhtemel yeni özelliklerin belirlenmesi konusunda da yardımcı olacaktır.

Uygulamanın literatürdeki diğer çalışmalara kıyasla üstün özellikler sunmasına karşın birtakım kısıtlamaları da mevcuttur. Bunlardan birincisi derinlik çıkarımı servisinde kullanılan DL modelinin dış mekan görüntülerinden oluşan KITTI veri setiyle eğitilmiş olması, dolayısıyla iç mekandaki bir kullanım için servisin elverişli olmayışıdır. Buna ek olarak, derinlik servisinin ürettiği derinlik bilgisinin doğruluğunun, ilgili literatür bağlamında yeterli olmasına rağmen, yükseltilmesi gerektiği düşünülmektedir. Derinlik çıkarımı servisinin iç mekanlara da uyarlanması için KITTI veri setine ek olarak, iç mekan görüntüleri barındıran NYU Depth (Silberman vd., 2012) veri setinin de eğitim sürecine dahil edilerek BTS modelinin yeniden eğitilmesi bir çözüm olarak düşünülmüştür. Üretilen derinlik bilgisinin doğruluk düzeyinin yükseltilmesi için ise, telefonda sunucuya görüntünün aktarımı sırasında uygulanan görüntü boyutu düşürülmesi işlemindeki küçültme oranının azaltılması düşünülen çözümlerden biridir. Bu boyut düşürme işlemine sunucu ile istemci arasındaki iletişim süresinin kısaltılması ve sunucuda GPU üzerinde hesaplanan parametre sayısının azaltılması için başvurulmuştur. BTS modeli her bir girdi piksele karşılık bir derinlik bilgisi ürettiği için görüntü boyutunun büyümesi daha fazla konvolüsyon

parametresi hesaplanması anlamına gelmektedir. Aynı DL modelinin kullanılması şartı altında görüntü küçültme oranının azaltılması için sunucuda daha yüksek kapasiteli bir GPU'nun kullanılması gerekmektedir. Derinlik çıkarımı doğruluğunun artırılması için BTS modelinin boyut indirilmesi uygulanmış akıllı telefon kamerası görüntüleri ve bu görüntülere karşılık gelecek derinlik haritaları ile yeniden eğitilmesi de düşünülen çözümlerden biridir. Bu iyileştirme sürecinin gerçekleştirilmesi ve bulgularının paylaşılması bir gelecek çalışma olarak hedeflenmektedir.

Tüm bunların yanı sıra, obje tespiti için kullanılan ön eğitilmiş YOLO modeli, eğitiminde kullanılmış COCO veri setinde bulunmayan engelli yollarında kullanılan dokunsal parke taşı nesnesine ait görüntüler de veri setine eklenerek yeniden eğitilecektir. Bu eğitimin sonucunda elde edilecek metriklerle birlikte yeni bir performans değerlendirmesinde bulunulacaktır. Bu eğitimle eş zamanlı olarak, OpenStreetMap altlığı üzerine belirli bir pilot çalışma sahasındaki dokunsal parkeyle kaplı engelli yolları da işlenerek geliştirilen uygulamaya bu yollar üzerinden navigasyon sağlama özelliği kazandırılacaktır. Kamera görüntüsü üzerinden engelli yollarının tanınmasıyla ARIANNA uygulamasında olduğu gibi kullanıcının engelli yolu üzerinde rotasına devam edebilecek şekilde yönlendirilmesi sağlanacaktır. Uygulamaya bu şekilde yalnızca engelli yolları üzerinden navigasyon yeteneği kazandırılması bir gelecek çalışma olarak hedeflenmektedir. Öyle ki, (Aktaş vd., 2020) çalışmasında dokunsal parke yüzeyi görüntülerinden oluşan bir veri seti ile eğitilmiş bir nesne tanıma DL modeli geliştirilmiş ve performans değerlendirmesinde bulunulmuştur. Gelecek çalışma olarak ise, yukarıda söz edilen hedef çalışmaya benzer görme engellilere yönelik bir navigasyon uygulaması hedefi belirlemişlerdir.

5. KAYNAKÇA

- Akın, A. T. ve Cömert, Ç., 2021. Testing of a Deep Learning Model Providing Monocular Depth Estimation on Mobile Devices via Web Service, 5th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT), IEEE, 43-46.
- Aktaş, A., Doğan, B. ve Demir, Ö., 2020. Derin öğrenme yöntemleri ile dokunsal parke yüzeyi tespiti, Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi, 35(3), 1685-1700.
- Bauer, Z., Dominguez, A., Cruz, E., Gomez-Donoso, F., Orts-Escolano, S. ve Cazorla, M., 2020. Enhancing perception for the visually impaired with deep learning techniques and low-cost wearable sensors, Pattern Recognition Letters, 137, 27–36.
- Bimber, O. ve Raskar, R., 2005. Spatial augmented reality: merging real and virtual worlds, AK Peters/CRC Press.
- Bishop, Christopher M., 2006. Pattern recognition, Machine learning 128.9.
- Borjali, A., Chen, A. F., Muratoglu, O. K., Morid, M. A. ve Varadarajan, K. M., 2020. Deep learning in orthopedics: how do we build trust in the machine?, Healthcare Transformation.
- Fausett, Laurene V., 2006. Fundamentals of neural networks: architectures, algorithms and applications, Pearson Education India.
- Furht, B., 2008. Encyclopedia of multimedia, Springer Science & Business Media.
- Gallo, P., Tinnirello, I., Giarré, L., Garlisi, D., Croce, D. ve Fagiolini, A., 2013. ARIANNA: pAth recognition for indoor assisted navigation with augmented perception, arXiv preprint, arXiv:1312.3724
- Geiger, A., Lenz, P., Stiller, C. ve Urtasun, R., 2013. Vision meets robotics: The kitti dataset, The International Journal of Robotics Research, 32, 11, 1231-1237.
- Goodfellow, I., Bengio Y. ve Courville A., 2016. Deep learning, MIT press.
- Hartley, R. ve Zisserman, A., 2003. Multiple view geometry in computer vision, Cambridge university press.
- Kandalan, R.N. ve Namuduri, K., 2020. Techniques for Constructing Indoor Navigation Systems for the Visually Impaired: A Review, IEEE Transactions on Human-Machine Systems, 50, 492–506.
- Khan, F., Salahuddin, S. ve Javidnia, H., 2020. Deep Learning-Based Monocular Depth Estimation Methods—A State-of-the-Art Review, Sensors.

- Laina, I., Rupperecht, C., Belagiannis, V., Tombari, F. ve Navab, N., 2016. Deeper depth prediction with fully convolutional residual networks, Fourth international conference on 3D vision (3DV), IEEE, 239-248.
- Lee, J. H., Han, M. K., Ko, D. W. ve Suh, I. H., 2019. From big to small: Multi-scale local planar guidance for monocular depth estimation, [arXiv preprint](#), arXiv:1907.10326.
- Lin, B.-S., Lee, C.-C. ve Chiang, P.-Y., 2017. Simple Smartphone-Based Guiding System for Visually Impaired People, [Sensors](#), 17, 1371.
- Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P. ve Zitnick, C.L., 2014. Microsoft COCO: Common Objects in Context, [Lecture Notes in Computer Science](#), 740–755.
- Liu, L., Ouyang, W., Wang, X., Fieguth, P., Chen, J., Liu, X. ve Pietikäinen, M., 2020. Deep Learning for Generic Object Detection: A Survey, [International Journal of Computer Vision](#), 128, 261–318.
- Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y. ve Berg, A.C., 2016. SSD: Single Shot MultiBox Detector, [Lecture Notes in Computer Science](#), 21-37.
- Lo Valvo, A., Croce, D., Garlisi, D., Giuliano, F., Giarré, L. ve Tinnirello, I., 2021. A Navigation and Augmented Reality System for Visually Impaired People, [Sensors](#), 21, 3061.
- Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., ... ve Chintala, S., 2019. Pytorch: An imperative style, high-performance deep learning library, [Advances In Neural Information Processing Systems](#), 32, 8026-8037.
- Rashid, T., 2016. Make your own neural network, CreateSpace Independent Publishing Platform.
- Redmon, J., Divvala, S., Girshick, R. ve Farhadi, A., 2016. You only look once: Unified, real-time object detection, [In Proceedings Of The IEEE Conference On Computer Vision And Pattern Recognition](#), 779-788.
- Redmon, J. ve Farhadi, A., 2018. Yolov3: An incremental improvement, [arXiv preprint](#) arXiv:1804.02767.
- Silberman, N., Hoiem, D., Kohli, P., ve Fergus, R., 2012. Indoor segmentation and support inference from rgb-d images. In European conference on computer vision, Springer, Berlin, Heidelberg, 746-760.
- Sun, M., Ding, P., Song, J., Song, M. ve Wang, L., 2019. “Watch Your Step”: Precise Obstacle Detection and Navigation for Mobile Users Through Their Mobile Service, [IEEE Access](#), 7, 66731–66738.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V. ve Rabinovich, A., 2015. Going deeper with convolutions, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 1-9.

World Health Organization (WHO), 2019. World Report On Vision.

Zhao, Z.-Q., Zheng, P., Xu, S.-T. ve Wu, X., 2019. Object Detection With Deep Learning: A Review, IEEE Transactions on Neural Networks and Learning Systems, 30, 3212–3232.

Zurada, Jacek M., Introduction to artificial neural systems, Vol. 8. St. Paul: West, 1992.

URL-1, <http://www.korleriegitimvekalkindirma.org/?news=gorme-yetersizliginin-tanimi-ve-gormeyetersizliginin-gelisim-uzerinde-etkisi/>, Görme Yetersizliğinin Tanımı Ve Görme Yetersizliğinin Gelişim Üzerinde Etkisi, 30 Kasım 2021.

URL-2, <https://towardsdatascience.com/a-high-level-explanation-of-popular-neural-network-architectures-49dce63f02e8>, A High Level Explanation Of Popular Neural Network Architectures, 30 Kasım 2021.

URL-3, <https://github.com/ultralytics/yolov5>, YOLOv5, 30 Kasım 2021.

URL-4, <https://github.com/ErenBalatkan/Bts-PyTorch>, BTS – PyTorch, 30 Kasım 2021.

URL-5, <https://machinelearningmastery.com/how-to-use-transfer-learning-when-developing-convolutional-neural-network-models/>, Transfer Learning in Keras with Computer Vision Models, 30 Kasım 2021.

URL-6, <https://www.openstreetmap.org/>, OpenStreetMap, 30 Kasım 2021.

URL-7, <http://gorengoiz.aile.gov.tr>, GörenGöz, 30 Kasım 2021.

URL-8, <https://wewalk.io/tr/>, Dünyanın En Akıllı Bastonu ve Mobil Uygulaması, 30 Kasım 2021.

URL-9, <https://taptapseeapp.com/instruction.html>, How It Works, 30 Kasım 2021.

URL-10, <https://cloudsight.ai/>, CloudSight, 30 Kasım 2021.

URL-11, <http://in.sight.srl/arianna/>, Arianna, 30 Kasım 2021.

URL-12, <https://developer.apple.com/augmented-reality/arkit/>, More to Explore with ARKit 5, 30 Kasım 2021.

URL-13, <https://developers.google.com/ar>, ARCore, 30 Kasım 2021.

URL-14, <https://www.theverge.com/2017/12/15/16782556/project-tango-google-shutting-down-arcore-augmented-reality>, Google's Project Tango is shutting down because ARCore is already here, 30 Kasım 2021.

- URL-15, <https://www.gsmarena.com/compare.php3?idPhone1=5953&idPhone2=8961>, Compare Specs, 30 Kasım 2021.
- URL-16, <https://docs.mapbox.com/api/overview/>, Web Services APIs, 30 Kasım 2021.
- URL-17, <https://assetstore.unity.com/>, Unity Asset Store, 30 Kasım 2021.
- URL-18, <https://www.anaconda.com/products/individual>, Your data science toolkit, 30 Kasım 2021.
- URL-19, <https://docs.spyder-ide.org/current/index.html>, Welcome to Spyder's Documentation, 30 Kasım 2021.
- URL-20, <https://code.visualstudio.com/docs>, Visual Studio Code, 30 Kasım 2021.
- URL-21, <https://flask.palletsprojects.com/en/2.0.x/#>, Flask, 30 Kasım 2021.
- URL-22, <https://opencv.org/>, OpenCV, 30 Kasım 2021.
- URL-23, <https://viso.ai/deep-learning/pytorch-vs-tensorflow/>, Pytorch vs Tensorflow: A Head-to-Head Comparison, 30 Kasım 2021.
- URL-24, <https://www.scientificamerican.com/article/bring-science-home-estimating-height-walk/>, Stepping Science: Estimating Someone's Height from Their Walk, 30 Kasım 2021.
- URL-25, <https://www.cdc.gov/coronavirus/2019-ncov/prevent-getting-sick/prevention.html>, Protect Yourself, 30 Kasım 2021.
- URL-26, <https://cloud.google.com/speech-to-text/>, Speech-to-Text, 30 Kasım 2021.
- URL-27, <https://cloud.google.com/text-to-speech>, Text-to-Speech, 30 Kasım 2021.
- URL-28, shorturl.at/gxyTU, YOLOv5 Tutorial, 30 Kasım 2021.

ÖZGEÇMİŞ

2012 yılında Düzce Anadolu Öğretmen Lisesi'ni bitirdi. 2013 yılında Karadeniz Teknik Üniversitesi Harita Mühendisliği bölümünde eğitim görmeye hak kazandı ve 2018 yılında mezun oldu. 2018 yılında Karadeniz Teknik Üniversitesi Harita Mühendisliği Anabilim dalında tezli yüksek lisans çalışmalarına başladı. 2019 yılından bu yana Karadeniz Teknik Üniversitesi Harita Mühendisliği bölümünde Araştırma Görevlisi olarak görev yapmaktadır. B2 seviyesinde İngilizce bilmektedir.

