

Yazılım Geliştirme Laboratuvarı-II

2. Proje

Akıllı Durak/Navigasyon ve Sesli Yönlendirme Sistemi

Gökdeniz Yılmaz
Bilişim Sistemleri Mühendisliği
Kocaeli Üniversitesi
211307083

Mustafa İnce
Bilişim Sistemleri Mühendisliği
Kocaeli Üniversitesi
221307109

Taha Rıdvan Öztürk
Bilişim Sistemleri Mühendisliği
Kocaeli Üniversitesi
221307097

1. Giriş

Bu proje, görme engelli bireyler için sesli navigasyon sağlamak üzere tasarlanmış mobil tabanlı bir uygulamanın geliştirilmesine odaklanmaktadır. Birincil amaç, görme engelli kullanıcıların sesli rehberli bir sistem kullanarak otobüs duraklarına gidip gelmelerini sağlamaktır. Kullanıcılar varış noktalarını sözlü olarak girebilmekte ve bu bilgiler daha sonra metne dönüştürülerek hedef doğrultusunda sesli talimatlar vermektedir. Aynı zamanda Görüntü işleme kullanılarak canlı görüntü üzerinde kırmızı ışık, yeşil ışık ve engeller gibi olumsuz durumlarda kullanıcı sesli olarak uyarılmaktadır. Bu çözüm, gelişmiş mobil teknolojilerden ve kullanıcı dostu arayüzlerden yararlanarak görme engelli bireylerin hareketliliğini ve bağımsızlığını artırmayı amaçlamaktadır.

2. Literatür Taraması

Görme engelliler için sesli yönlendirme uygulamaları, GPS ve dijital harita hizmetlerine dayanır. Mevcut navigasyon uygulamaları üzerinde yapılan araştırmalar, görme engelliler için farklı konum tespit uygulamaları alternatiflerine ulaşılsa da, bu uygulamalar genellikle kullanıcılara belirli özellikleri tam olarak sunamazlar ve genellikle tek bir yapı üzerine kurulurlar.

Bu uygulamada, kullanıcının ve hedef konumun bilgileri alınarak gerçek zamanlı haritalama işlemleri gerçekleştirilir ve kullanıcıya oluşan rotaya göre aralıklarla direktifler verilerek yönlendirme işlemi yapılır. Ayrıca, görme engelliler için büyük bir problem olan yaya geçitlerindeki trafik işaretlerinin tespiti yapılarak sesli direktiflerle uyarı mekanizması eklenmiştir. Bu sayede, gerçek zamanlı haritalama ile kullanıcıya sesli direktiflerle yönlendirme yapılarak hem görüntü tespiti yapılır hem de kullanıcıya yaya geçitlerinde kolaylık sağlanır. Bu iki teknoloji birbirine entegre edilmiştir.

3. Yöntem

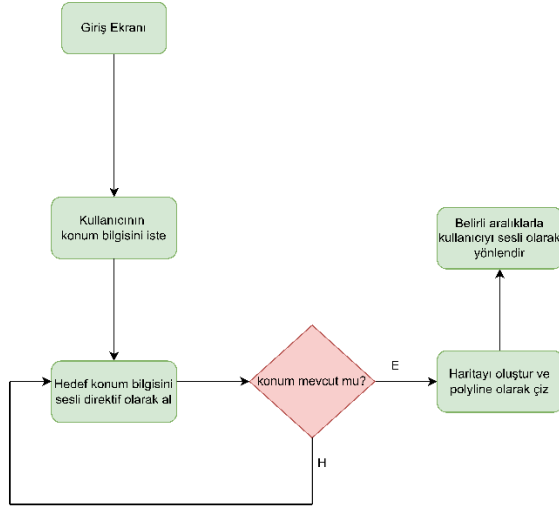
Projeye ilk olarak uygulamadaki gereksinimlerin tespit edilmesi ile başlanmıştır.

3.1 Kullanılan Teknolojiler ve Araçlar

- Android Studio: Android tabanlı proje geliştirme ortamı olarak seçilmiştir. Popülerliği ve dokümantasyon konusunda fazla çeşitliliği olduğundan tercih sebebidir.
- Kotlin Programlama Dili: Google API'lerinin Kotlin ile örneklerinin çokça bulunması nedeniyle Kotlin programlama dili tercih edilmiştir.
- Harita ve Navigasyon: Harita ve Navigasyon amaçları için Google'ın Maps, Geocoding ve Fused Location Provider API'leri tercih edilmiştir.

- Speech-to-Text ve Text-to-Speech: Görme engelli kullanıcılarla iletişim amacıyla yine Google'ın SpeechRecognizer ve Text-to-Speech API'leri tercih edilmiştir.
- Engel ve Durum Farkındalığı: Engellerin tespiti ve çevre farkındalığı için YoloV8 tabanlı bir engel ve trafik lambası tespit modeli kullanılmıştır.

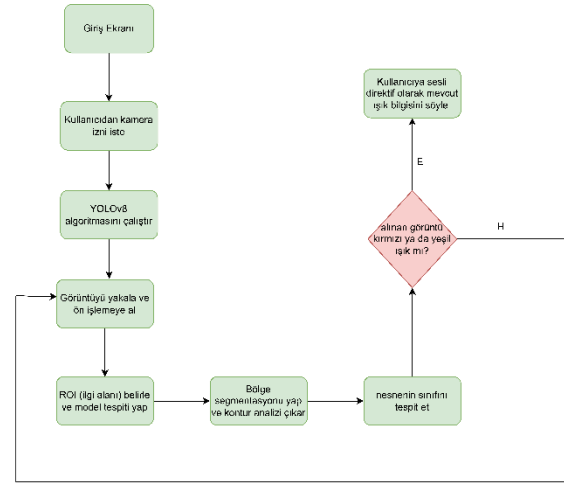
3.2 Algoritma ve Akış Diyagramları



Şekil 1 - Uygulama Genel Akış Diyagramı

3.2.1 Uygulama Genel İşleyişi

Uygulama izin verildikten sonra kullanıcının cihazından konum bilgisini çekerek başlar. Sonrasında Speech-to-Text fonksiyonu ile kullanıcının gitmek istediği konumu alır ve Routes API'ye gönderir. Google Routes üzerinde rota hesaplanır ve maps API'si üzerinde ekrana çizdirilir. Sonrasında hedef Directions API'ye gönderilir ve rota adımları çıkartılır. Aynı zamanda bu adımlar Text-to-Speech özelliği ile kullanıcıya bildirilir. Aynı zamanda TensorFlowLite Görüntü işleme modeli ile kamera görüntüsü sürekli olarak izlenir ve trafik lambası, yolda engel ve park halinde araç gibi durumlarda kullanıcı sesli olarak uyarılır. Hedefe varıldığında Kullanıcı bilgilendirilir ve akış sona erer.



Şekil 2 - Görüntü İşleme ile Tespit Akış Şeması

4. Sistem Tasarımı

Uygulamanın hedef kitlesi görme engelli kullanıcılar olduğu için mevcut uygulama arayüzünde basit ve kullanımı kolay bir tasarım oluşturmak hedeflenmiştir. Uygulama giriş ekranında kullanıcının kullanabileceği 2 adet işlev bulunmaktadır. Bunlar, kullanıcıdan sesli direktif alma ve görüntü tespiti işlevleridir.

4.1 Navigasyon Ekranı

Bu ekranda kullanıcıdan izinler alındıktan sonra mevcut ve hedef konum belirlenir ve rota oluşturulur. Ardından kullanıcıya belirli aralıklarla sesli talimatlar verilir.

4.2 Görüntü Tespit Ekranı

Bu ekranda kullanıcıdan kamerasını kullanmak için izin alındıktan sonra yayalar için trafik lambası tespiti yapılabilmesi için nesne tespiti gerçekleştirilir. Eğer alınan görüntüde hedef nesneler bulunuyorsa, kullanıcı sesli direktifler aracılığıyla uyarılır.



Şekil 3 - Engel Tespit Ekranı ve Örnek Yakalama Görüntüsü

5. Uygulamanın Geliştirilmesi

5.1 Kullanılan İzinler:

5.1.1 android.permission.ACCESS_COARSE_LOCATION

Bu izin, uygulamanın kaba konum verilerine erişmesine izin verir. Bu, genellikle GPS gibi daha kesin konum belirleme yöntemlerine başvurmadan önce kullanılan daha az doğru konum verilerini içerir. Bu izin, uygulamanın yaklaşık konum bilgilerine erişmesine izin verir.

5.1.2 android.permission.ACCESS_FINE_LOCATION

Bu izin, uygulamanın hassas konum verilerine erişmesine izin verir. Bu, GPS gibi daha kesin konum belirleme yöntemlerini kullanarak daha doğru konum bilgilerini içerir. Bu izin, uygulamanın kesin konum bilgilerine erişmesine izin verir.

5.2 Fonksiyonlar

5.2.1 getLastLocation()

Bu yöntem, cihazın son bilinen konumunu döndürür. Bu, cihazın önceki konum güncellemelerini kullanarak son konumu sağlar. Bu yöntem, konum bilgisini hızlı bir şekilde alır ve pil kullanımını en aza indirir, çünkü daha önce alınmış bir konum bilgisini kullanır. Ancak, cihazın son konumu güncel olmayabilir, özellikle cihaz yakın zamanda konum hizmetlerini kullanmamışsa veya konum hizmetleri etkin değilse.

5.2.2 getCurrentLocation()

Bu yöntem, cihazın mevcut konumunu döndürür. Bu yöntem, daha doğru ve güncel bir konum sağlamak için cihazın konum hizmetlerini etkin olarak kullanır. Bu, GPS, Wi-Fi, mobil veri veya diğer sensörler gibi cihazın mevcut konumu belirlemek için kullanabileceği tüm mevcut kaynakları kullanır. Ancak, bu yöntem, cihazın konum hizmetlerini aktif olarak kullanması gerektiğinden pil kullanımını artırabilir ve bu nedenle daha fazla enerji tüketir.

5.2.3 requestLocationUpdates()

requestLocationUpdates() fonksiyonu, Android platformunda konum güncellemelerini dinlemek ve belirli bir aralıkta konum bilgilerini almak için kullanılır. Bu fonksiyon, bir LocationManager veya FusedLocationProviderClient gibi bir konum yöneticisi üzerinden çağrılır.

5.2.4 onSaveInstanceState()

Bu fonksiyon, bir Android Activity veya Fragment'in geçici durumunu (state) kaydetmek için kullanılır. Bu yöntem, Activity veya Fragment'in beklenmedik bir şekilde durdurulması veya sonlandırılması durumunda,

kullanıcının çalışma oturumunu kurtarmak için önemlidir.

5.3 Kullanılan API'ler

5.3.1 Google Routes API

Bu API, kullanıcılara farklı ulaşım yöntemleriyle (araba, yürüme, bisiklet, toplu taşıma) belirli bir kaynaktan hedefe yönlendirme bilgileri sağlar. Kullanıcılar, iki nokta arasındaki mesafeyi, tahmini varış süresini, tahmini seyahat süresini ve önerilen rotayı alabilir. Ayrıca, API, trafik durumu ve yol koşullarını da dikkate alarak en iyi rotayı sağlar.

5.3.2 Google Directions API

Bu API, belirli bir başlangıç ve bitiş noktası arasında yol tarifi almak için kullanılır. Kullanıcılar, iki nokta arasındaki yolu, farklı ulaşım modları için talimatları ve yol tarifini alabilirler. Ayrıca, rota üzerindeki özelliklerin (örneğin, benzin istasyonları, restoranlar) listesini de alabilirler.

5.3.3 Google Geocoding API

Bu API, bir adresi (veya tersi, enlem ve boylamı) bir coğrafi konuma dönüştürmek için kullanılır. Yani, kullanıcılar bir adres girdiğinde veya bir enlem ve boylam girdiğinde, API bu konumu belirler ve coğrafi konuma dönüştürür. Ayrıca, coğrafi konumlar için tam adresleri almak için de kullanılabilir.

6. Test ve Doğrulama

Uygulama geliştirilirken API'ler eklenmeden önce örnek uygulamalarda API'lerin kullanılacak özellikleri test edilmiş ve ardından ana uygulamaya eklenmiştir.

6.1 Rota Testi

Google API'lerinden Routes API kullanılmıştır. Örnek Uygulaması üzerinden test edilerek istediğimiz şekilde çalıştığı gözlemlenmiş ve uygulamaya entegre edilmiştir.

6.2 Maps API Testi

Google Maps API'si uygulamaya eklenerek haritanın yüklendiği ve konumun tespit

edilebildiği test edilmiş ve uygulamaya eklenmiştir.

6.3 Speech-to-Text API Testi

Gürültülü ve değişken seslerin olduğu ortamlarda denenerek trafiğin yoğun olduğu bölgeler gibi yerlerde çalışabildiği test edilmiş ve uygulamaya eklenmiştir.

6.4 Geocoding API Testi

Speech to text ile önceden adresi bilinen bir konum uygulamaya iletilerek verdiği adresler karşılaştırılmış, doğru olduğu görülmüş ve uygulamaya eklenmiştir.

7. Referanslar

- <https://developers.google.com/maps/documentation/routes?hl=en>
- <https://developers.google.com/maps/new-map-style-opt-in?hl=en>
- <https://developers.google.com/maps/documentation/geocoding/?hl=en>
- <https://developers.google.com/maps/documentation/geolocation>
- <https://docs.ultralytics.com/hub/app/android/>
- <https://rockyshikoku.medium.com/using-yolov8-object-detection-model-on-android-18e51a519ba8>
-