

T.C.
HARRAN ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

BİTİRME ÖDEVİ

GÖRME ENGELLİLER İÇİN DERİN ÖĞRENME TABANLI MOBİL
UYGULAMA VE ASİSTAN GÖZLÜK

Emirhan ÇİFTÇİ
Yusuf Yasin KAYA
Metin Tarık KİKİ
Yusuf Ali BEKÇİ
Dr. Öğr. Kemal GÜNER

ŞANLIURFA

2023

Görme Engelleiler İçin Derin Öğrenme Tabanlı Mobil Uygulama ve Asistan Gözlük

**Emirhan ÇİFTÇİ
Yusuf Yasin KAYA
Metin Tarık KİKİ
Yusuf Ali BEKÇİ**

**T.C.
HARRAN ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

BİTİRME ÖDEVİ

**GÖRME ENGELLİLER İÇİN DERİN ÖĞRENME TABANLI MOBİL
UYGULAMA VE ASİSTAN GÖZLÜK**

**Emirhan ÇİFTÇİ
Yusuf Yasin KAYA
Metin Tarık KİKİ
Yusuf Ali BEKÇİ
Dr. Öğr. Kemal GÜNER**

ŞANLIURFA

2023

Bitirme Ödevi Yönergesi uyarınca hazırlanmış ve anılan bölüme sunulmuştur.

Şanlıurfa 2023

**Emirhan ÇİFTÇİ
Yusuf Yasin KAYA
Metin Tarık KİKİ
Yusuf Ali BEKÇİ**

ONAY

Doç. Dr. İ. Berkan AYDİLEK
Bölüm Başkanı

Arş. Gör. Harun ÇİĞ
Üye

Dr. Öğr. Üyesi Kemal GÜNER
Komisyon Başkanı

Danışman

Arş. Gör. Songül AKDAĞ
Üye

İÇİNDEKİLER

TEŞEKKÜR

1.GİRİŞ.....	
1.1.Özet.....	
1.2.Problem Durumunun Tanımlanması.....	
1.3.Amaç ve Hedefler.....	
2.LİTERATÜR TARAMASI.....	
3.YENİLİKÇİ YÖNLER.....	
4.YÖNTEM.....	
4.1.İmplementasyon Yönteminin Belirlenmesi ve Optimizasyon Çalışmaları.....	
4.2.İmplementasyonun Uygulanma Yöntemi.....	
4.3.Demo Uygulama.....	
5.KULLANILAN TEKNOLOJİLER.....	
6.BULGULAR.....	
7.SONUÇLAR VE ÖNERİLER.....	
KAYNAKLAR.....	
ÖZGEÇMİŞ.....	

TEŞEKKÜR

Öncelikle bitirme projesinin konusunun belirlenmesi, proje kapsamında ilgili konu üzerine yapılan geliştirmeler ve geliştirme fikirleri konusunda isteklerimizi göz önünde bulundurup süreç boyunca bize yardımcı olan proje danışmanımız Dr. Öğr. Üyesi Kemal Güner'e tüm sorularımıza yanıt verdiği, araştırma konuları ve proje ilerleyişiyle ilgili gerekli yönlendirmeleri yaptığı, yardımını esirgemediği için teşekkür ederiz. Üniversite eğitim hayatımız boyunca özellikle proje boyunca araştırmasını yaptığımız konseptler ve bunların öğrenilmesi ile ilgili ders içerikleri ve bu konseptlerle ilgili yönlendirmelerle ilgili makine öğrenmesi konusunda Dr. Öğr. Üyesi Mehmet Emin Tenekeci'ye, Yapay Zeka konseptleri ve paralelleştirme konseptlerinin ders içeriklerinde bulundurulmasıyla ilgili Bilgisayar Mühendisliği Bölüm Başkanı Doç. Dr. İbrahim Berkan Aydılek'e, ilgili konseptlerin daha detaylı uygulamaları ve işlenmesi ile ilgili özellikle örüntü tanıma ders içeriklerinin zenginliği ve bize kattıkları dolayısıyla Dr. Öğr. Üyesi Serdar Çiftçi'ye teşekkürlerimizi sunarız.

Yusuf Yasin KAYA

Yusuf Ali BEKÇİ

Emirhan ÇİFTÇİ

Metin Tarık KİKİ

2023

1. GİRİŞ

1.1. Özet

Engelsiz bir yaşam ortamının oluşması için gerek ve yeter şart dezavantajlı bireylerin sorunlarına mümkün olan en yüksek seviyede fayda getirecek çözümlerin sunulmasıdır. Bu projede, görme engelli bireylerin yaşam kalitesini arttıracak bir sistem geliştirilerek bu bireylere yönelik bir sosyal fayda sağlamak hedeflenmiştir. Geliştirilmesi planlanan sistemin hedefi, görme engelli bireylerin çevresindeki nesneler & varlıklar hakkında bu bireylere detaylı, hızlı ve net şekilde bilgi vermek, bu nesneler & varlıklarla etkileşime geçerken kazara kendisine zarar verme ihtimalini minimuma indirmektir. Bu bağlamda öncelikle bir mobil uygulama geliştirilerek gerekli modüllerin gerçekleştirilmesi bu uygulama üzerinden sağlanacaktır. Mobil uygulama yazılımının ve sistemin gerekli modüllerinin geliştirilmesi tamamlandığında, sistemin işlevselliğini arttırmak amacıyla sistemin üzerinde çalışabileceği bir gözlük tasarlanacaktır. Projemizin hedefini gerçekleştirmek üzere geliştirilecek yazılım ve gözlük sistemi, sürekli olarak kamera erişimi sağlayarak kullanıcı ile etkileşime geçmeye hazır olacak, kullanıcı ihtiyaç duyduğu anda cihaz kamerasından elde edilen görüntüleri, içerisine entegre edilen derin öğrenme modelini kullanarak işleyip sesli olarak kullanıcıya bilgi verecektir. Bu yazılım sistemini kullanan bireyin etrafındaki varlıklar hakkında bilgilerin seslendirilmesi, 3D ses simülasyonları kullanılarak sağlanacaktır. Böylece bireyler sadece etrafında ne olduğunu değil, bunların mesafeleri ve yönleri hakkında da bilgiyi hızlı ve net olarak edinecek, buna göre daha rahat hareket edebileceklerdir.

1.2. Problem Durumunun Tanımlanması

Bir görme engelli bireyin temelde en büyük sorunu etrafındaki nesneler & varlıkları tanımak ve bu nesne & varlıklarla etkileşime geçerken kendisine zarar verme potansiyelinin bulunmasıdır. Bu bağlamda bu bireyler, çevrelerini zarar görmeden tanımak üzere baston gibi çeşitli aletler kullanmakta ve yer yer etrafındaki diğer bireylerin kendilerine yardımcı olmasına ihtiyaç duymaktadır. Bir görme engelli bireyin, etrafında özellikle başka insanlara ihtiyaç duymadan çevresini tanımasına yardımcı olabilecek bir cihaz, bu bireyin hayatını önemli ölçüde kolaylaştıracak, daha özgür biçimde çevresiyle etkileşime geçebilmesini sağlayacaktır. Görme engelli bireylere günlük yaşamlarında onlara yardımcı olmak üzere geliştirilen çeşitli cihazlar ve yazılımlar bulunmaktadır. Ancak henüz piyasadaki ürünler, gerçekleştirmek istediğimiz ürünün işlevlerine sahip değildir.

Projemizin hedefi, görme engelli bireylerin etrafındaki nesneler & varlıkları detaylı olarak algılamasını sağlamak ve bunu başka bir insanın desteğine ihtiyaç duymadan yapay zekâ ve görüntü işleme yöntemlerini kullanarak yapmaktır. Bu doğrultuda, görme engelli bireyler için öncelikle bir mobil uygulama ve uygulamanın yaptığı işlemleri daha işlevsel düzeyde gerçekleştirebilecek özel bir gözlük tasarlanması amaçlanmaktadır. Gerçekleştirilmesi planlanan mobil uygulamanın ve sonraki aşamada geliştirilmesi planlanan gözlüğün görevi, kameradan aktif olarak görüntü alıp, bu görüntüleri sayısal veriye dönüştürerek görüntü işleme ve örüntü tanıma algoritmalarıyla işleyerek görme engelli insanlara etrafındaki varlıklar/nesnelerle ilgili bu nesne/varlıkların öncelik seviyelerine göre sesli bir şekilde ne olduklarını, mesafelerini ve 3D ses simülasyon teknikleri kullanarak (yani yönü söylemek yerine sesleri o yönden algılayacağı şekilde) bulundukları yönleri algılamasını sağlamaktır.

1.3. Amaç ve Hedefler

Projemizin amacı, görme engelli insanların etrafındaki nesneleri daha kolay algılamasını sağlamaktır. Bu bağlamda amaç, etraftaki nesneleri tanıyan ve önceliklerine göre kullanıcıya seslendiren bir gözlük geliştirmektir.

Bu bağlamda gerçekleştirilmesi planlanan işler:

1. YOLO (You Only Look Once) görüntü işleme algoritmasını ve açık kaynak olan Darknet Framework ile eğitilmiş modelleri kullanarak performans testlerini gerçekleştirip, ana sistemi, yazılımsal optimizasyon teknikleri ve uygun algoritmalar kullanarak düşük performanslı mobil cihazlar üzerinde çalıştırılabilir hale getirmek.

2. Birden fazla nesne aynı anda seslendirilemeyeceğinden bu işlemin sırayla yapılması gerekmektedir. Örneğin kaldırımda yürürken önce önde giden kişinin belirtilmesi, sonra sağında belirecek olan ağacın ve en son olarak da belediye ekiplerince açılmış su boru tesisatına ait çukurun belirtilmesi gibi. Seslendirme işleminin, algılanan nesne sayısına bağlı olarak alacağı vakit de değişkendir. Bu bağlamda daha yüksek tehlike arz eden bir nesnenin en son seslendirilmesini önlemek amacıyla nesneler için bir öncelik tablosu oluşturulacak ve bu önceliğe göre seslendirme yapacak modül geliştirilecektir.

3. Nesnelerin seslendirilmesi esnasında görme engelli bireyin, seslendirilen nesnenin yönünü algılayabilmesi amacıyla 3D ses simülasyonları uygulama içerisine entegre edilecektir. Böylece birey, seslendirilmekte olan nesnenin ne tarafta olduğunu, ek bir seslendirme işlemine gerek kalmadan hem duyup hem de algılayabilecektir.

4. Basit bir arayüze sahip, görme engelli bireyler tarafından yeterince kolay şekilde kullanılacak bir mobil uygulama geliştirilecektir.

5. Görüntü işleme işlemlerinin, yerel cihazlar üzerinde düşük performansta çalışması durumuna karşı B planı olarak, üzerinde görüntü işleme işlemlerinin daha hızlı bir şekilde gerçekleştirilebileceği uzak sunucu/bulut sunucu sistemi tasarlanacak ve tüm yazılım sistemi, hesaplamaların bu sunucu üzerinden gerçekleştirileceği şekilde tasarlanacaktır.

6. Amaç doğrultusunda yüksek doğruluk oranına sahip olacak şekilde derin öğrenme modellerinin çalıştırılabilmesi için yeterince yüksek çözünürlüğe ve kayıt özelliklerine sahip olan ve yazılan kodların üzerinde çalıştırılabileceği bir akıllı gözlük hazır alım yöntemiyle temin edilecektir. Bu gözlüğün, mobil uygulamaya kıyasla daha gelişmiş bir kullanıcı deneyimi sunması hedeflenmektedir. Burada amaç, birden fazla kamera veya mesafe sensörü ile daha hatasız mesafe algılama ve yön algısı sunmanın yanında, boynunda veya elinde sürekli bir telefon taşıma zorunluluğunu ortadan kaldırarak daha ergonomik bir deneyim sunmak hedeflenmektedir.

2. LİTERATÜR TARAMASI

Gerçekleştirmek istediğimiz projemizle aynı ya da benzer amacı taşıyan farklı projeleri araştırdık ve bulduğumuz ürünler arasında, yapmak istediğimiz şeyi gerçekleştiren hiçbir ürün bulamadık. İlgili ürünlerin her biri kendi avantaj ve dezavantajlarına sahiptir. Projemizde gerçekleştirmek istediğimiz hedefler, bu dezavantajları ortadan kaldırmak ve kullanıcılara daha etkin bir deneyim sunmaktır.

Araştırmalarımız sonunda bulduğumuz benzer amaçlı projeler:

"DeepSight: Görme Engellilere Yönelik Derin Öğrenme Tabanlı Mobil Uygulama" - Bu proje, görme engellilerin günlük yaşam aktivitelerini kolaylaştırmak amacıyla derin öğrenme algoritmalarını kullanarak mobil bir uygulama geliştirir. Uygulama, nesneleri tanımlama, metin okuma ve navigasyon gibi özellikleri içerir.

"WeWalk Akıllı Baston: WeWalk" – Bu proje görme engelli bireylerin günlük yaşamlarını kolaylaştırmak için tasarlanmış bir akıllı bastondur. Bu yenilikçi baston, geleneksel bastonlara eklenen özelliklerle donatılmıştır. WeWalk, ultrasonik sensörler ve yapay zeka algoritmaları kullanarak etrafındaki engelleri tespit eder ve kullanıcıya sesli uyarılar verir. Ayrıca, akıllı telefonlarla entegre olabilen Bluetooth teknolojisi sayesinde navigasyon özellikleri sunar, sesli mesajları alır ve çağrılarını yanıtlayabilir. WeWalk, görme engelli bireylerin günlük hayatta daha bağımsız ve güvenli hareket etmelerini sağlamak için tasarlanmış bir akıllı bastondur.

"Aira Akıllı Gözlük": Aira, görme engelli bireylere yardımcı olmak için tasarlanmış bir akıllı gözlük ve hizmet platformudur. Kullanıcılar, gerçek zamanlı görüntü aktarımı ve uzaktan yardım sağlayan Aira temsilcileriyle iletişim kurabilirler. Akıllı gözlük, kullanıcının çevresini görüntüler ve temsilciler yönlendirmeler ve bilgi sağlar. Aira, kullanıcılara güvenli bir şekilde hareket etmeleri, alışveriş yapmaları ve bağımsız olmaları için yardımcı olur.

"IrisVision – Elektronik Gözlük": IrisVision elektronik gözlükler, görme engelliler ve görüşü kısıtlı olan bireyler için geliştirilmiş, FDA (Food and Drug Administration) tarafından onay almış bir gözlük sistemidir. Samsung'un VC kafa bandı ve bir akıllı telefon ikilisinden oluşacak şekilde çalışır. 14 kata kadar yakınlaştırma ve 70 dereceye kadar görüş açısı sunar. Ayrıca sesli komut özelliğini destekler.

3. YENİLİKÇİ YÖNLER

Sistemin, şu ana kadar geliştirilmiş benzer amaçlı ürünlerle karşılaştırıldığında ön plana çıkan özellikleri şunlardır:

- 3D ses simülasyonu sayesinde çevredeki nesneler ve varlıkların yönleri ve mesafeleri ile ilgili daha detaylı bilgiyi anında ve sezgisel olarak elde edebilmeyi sağlamak.

- Nihai ürün olarak ortaya konulacak gözlük sisteminin sensörler ve örüntü tanıma metodlarını birlikte kullanarak bireyin çevresi hakkındaki bilgileri başka insanlara ihtiyaç olmadan kullanıcıya iletmek.

- Sistem, yöntem başlığında açıklandığı üzere yalnızca yerel cihazlarda yeterince yüksek başarı oranı ile çalışacak şekilde gerçekleştirildiği taktirde internet bağlantısına ihtiyaç duymadan çalışabilir olması.

Bu belgede açıklanan proje ile benzer amaçla geliştirilen projelerden biri, Aira isimli mobil uygulama ve Envision Glasses isimli gözlük çifti. Bu yazılım-gözlük çifti, görme engelli bireylerin bir çağrı merkezi sistemine bağlanıp, etrafını çevresi hakkındaki bilgileri karşı taraftaki bir yetkiliye iletmekte ve bu yetkiliden destek almaktadır. Bu yöntemin en büyük iki dezavantajı, internet bağlantısı zorunluluğu ve görme engelli bireyin bu sistem üzerinden destek alabilmesi için karşı tarafta muhatap olacağı başka bir insana ihtiyaç duyması.

Başka bir proje olarak, Android cihazlar için Mesut Şimşek tarafından Google Play Store üzerinde yayınlanmakta olan DijiGöz isminde bir mobil uygulama tasarlanmıştır. Bu uygulama, mobil cihazın kamerası üzerinden algılama yaparak, yazı okuma, algılanan nesneler hakkında sesli olarak bilgi verme işlevlerine sahiptir. Ancak algılanan nesnelerle ilgili verilen bilgiler, özellikle bu belgenin ana konusu olan proje ile kıyaslandığında çok kısıtlı olmakla birlikte, yayınlandığı platform olan Google Play Store üzerindeki kullanıcı geri bildirimleri incelendiğinde memnuniyet oranının düşük olduğu ve bu durumun sebebinin kullanıcılar tarafından sıklıkla bildirildiği üzere nesne algılama esnasındaki doğruluk oranının oldukça düşük olması olarak gözlemlenmektedir.

Görme engelli bireylere çevreleri ile ilgili yönlendirme yapmak, destek olmak üzere geliştirilen bir diğer ürün ise WeWalk. Bu ürün görme engelli bireylere sarı şeritleri takip ederek yön bilgisi sunmakta ve etrafındaki engelleri algılayarak titreşim vermektedir. Telefon ile bağlantı kurma seçeneği ile, yakınından geçilen otobüs durağı, restoran vb. mekanlar hakkında bilgi verebilmektedir. Ancak burada da cihazı kullanan birey etrafındaki varlıklar hakkında yalnızca yüzeysel olarak bilgi edinebilmekte -örneğin engel ile karşılaşıldığında sadece titreşim verilmesi-, çevresi ile etkileşime geçmediği sürece etrafındaki şeylerin ne olduklarıyla ilgili bilgi edinebilmesi söz konusu değildir.

Anlaşıldığı üzere geliştirdiğimiz projemizin asıl hedefleri, ortaya koymak istediğimiz ürünümüzün bu dezavantajlara sahip değildir ve görme engelli bireylere tam anlamıyla otonom olarak destek verebilecek bir ürün olması hedeflenmektedir.

4. YÖNTEM

Projemizin geliştirilmesi esnasında uygulayacağımız yöntemler, projenin kapsamı ve implemente etmeyi hedeflediğimiz konseptlerin zorluğu göz önüne alındığında, bu yöntemleri test planları, bu testlerin sonuçları ve mevcut donanım imkanlarının elverişliliğine göre implemente etmek gerekmektedir. Bu doğrultuda, proje geliştirme planı 3 parçaya ayrılmış, öncesinde mevcut donanım ve teknolojilerin incelenmesi, buradan elde edilen bilgiler doğrultusunda mümkün olan en uygun implementasyon yönteminin kararlaştırılması, bu yöntemlerin daha küçük bir ortamda test edilmesi ve en son elde edilen sonuçlar doğrultusunda nihai ürünün implemente edilmesi kararlaştırılmıştır. İlerleyen başlıklarda (4.1, 4.2, 4.3) süreç boyunca yaptığımız çalışmalar anlatılmaktadır.

4.1 başlığı süreç içerisinde hangi donanımın, ne tarz bir implementasyon yöntemi ile nasıl sonuçlar vereceğinin testini; 4.2 başlığı bu testler sonucunda ortaya konulacak nihai ürünle ilgili kararlaştırılan yol haritası ve implementasyonun tam anlamıyla nasıl yapılacağını; 4.3 ise süreç boyunca ilerlememiz ve demo ürününün implementasyonunu.

4.1 Implementasyon Yönteminin Belirlenmesi ve Optimizasyon Çalışmaları

Geliştirme hızının yüksek olmasından dolayı başlangıç aşamasındaki çalışmalar ve test sonuçlarını hızlıca analiz edebilmek için Python programlama dilini kullandık. Python için sağlanan görüntü işleme için OpenCV, nesne tanımlama için YOLO (you only look once – sadece bir kere bak) algoritması, matris işlemlerini kolaylaştıran NumPy, yazıları sese dönüştüren Google Text To Speech ve uygulamamızı mobil uygulama haline getirebilmek için Kivy ve KivyMD kütüphanelerini kullanarak bir demo oluşturduk.

Normal şartlarda YOLO algoritmasını kullanarak kendi modelimizi eğitme imkanı sunulmakta ancak başlangıç aşamasında performans testi için halihazırda eğitilmiş bir derin öğrenme modelini kullandık.

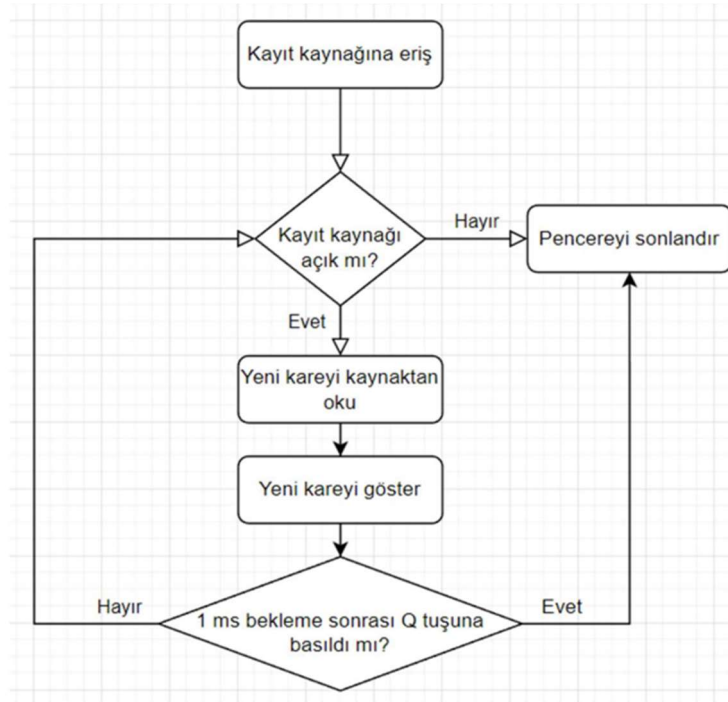
Hedefimiz, algoritmamızın bir grafik işlemci barındırmayan küçük cihazlar üzerinde çalışması olduğundan ilk aşamada OpenCV ile CPU üzerinde çalışarak kameradan canlı görüntü alma performansını test ettik ve optimizasyon yöntemlerini araştırdık. Görüntü işleme, derin öğrenme, yapay sinir ağları gibi sistemlerin hızlı çalışması için işlemlerin paralelleştirilmesi yani aynı anda yürütülmesi gerekir. Çünkü çok sayıda yapay nöron ve çok sayıda piksel bulunduğundan bu seviyelerde yapılan işlemlerin sayısı milyonlara çıkmaktadır. Bu bağlamda GPU'lar çok sayıda(binlerce) çekirdek barındırdığından işlemleri paralelleştirmek yani aynı anda çalışacak hale getirmek CPU'lara göre çok daha hızlı sonuç verir. Ancak burada CPU üzerinde çalışmaya karar vermemizdeki sebep küçük cihazlarda alacağımız sonucu canlandırmaya çalışmamızdır.

İkinci olarak Darknet üzerinde açık kaynak olarak sağlanan YOLO algoritmasının 320p derin öğrenme modelini kullanarak uygulamamızın çalışma performansını ölçtük, algoritmamızın algıladığı tüm nesnelerin ekranda yazı halinde listelenmesini ve hangi nesnelerin nasıl bir doğruluk oranıyla algılandığını saptadık. Bu bağlamda küçük nesnelerin algılanma oranının düşük olduğu, CPU üzerinde çalışırken performansın oldukça düşük olduğu sonucuna vardık.

Nesneleri daha doğru ve görüntülerin daha güncel şekilde algılanabilmesi için kameramızın yüksek kare oranıyla kayıt yapması gerekmekte. Ancak YOLO algoritmasını kameramızın işlediği her kare için çalıştırmamız CPU ile çok yavaş olduğundan bu algoritmanın sadece belirli kareler için çalıştırılması gerektiği ve bu karelerin nasıl seçileceği üzerine yoğunlaştık.

Kayıt içerisinde anlık olarak algılanan nesnelerin sayısı, onları seslendirmenin çok uzun süreceği kadar fazla olabiliyor. Bu bağlamda her nesne için birer öncelik belirleyip, ilgili anda seslendirilecek nesne sayısını sınırlandırdık.

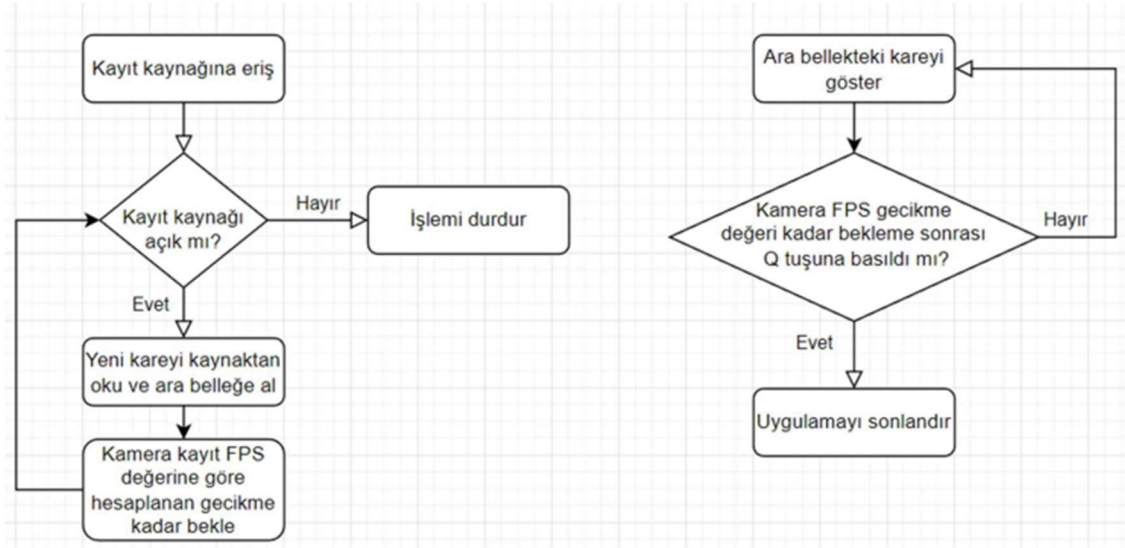
İlk aşamada kamera kontrollerinin çalışması için OpenCV ile basitçe kameradan görüntü almayı ve bu görüntünün ne kadar stabil olduğunu test etmeyi denedik. Bu aşamada kameradan alınan görüntüde takılmalar olduğunu gördük. Uygulama içerisinde bu takılmaları minimuma indirmek ilk hedefimizdi. Çünkü takılmalar aslında görüntünün güncel olmasını engelleyen bir durum. Proje hedefimizi gerçekleştirebilmek için kameradan alınan en güncel görüntüye ihtiyacımız var.



Şekil 1 - İlk görüntü testi

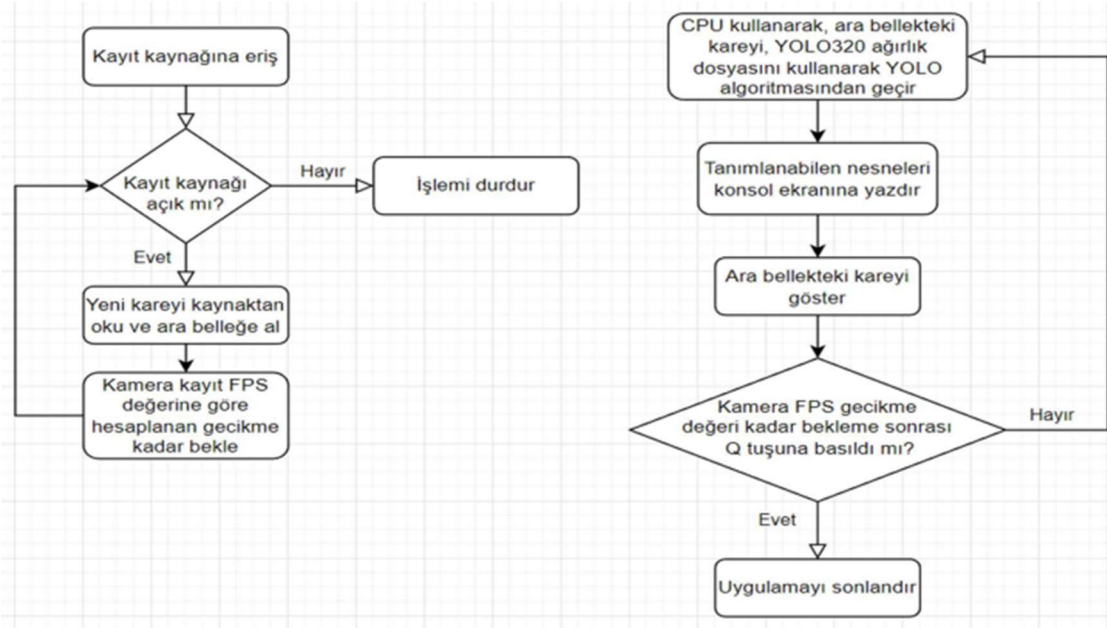
Takılma sebeplerini, bu sorunu yaşayan diğer insanların açtığı konuları internet üzerinde araştırmamız sonrasında bu takılmaların, görüntünün kaynaktan alınması esnasındaki gecikme ile

bu görüntünün ekranda görüntülenmesi esnasında yaşanan gecikme ile bağlantılı olduğunu gördük. Bu sorunu çözmek için kameradan görüntü alma işlemi ile görüntüyü ekrana verme işlemini asenkron olarak çalışan farklı fonksiyonlar haline getirmeyi denedik.



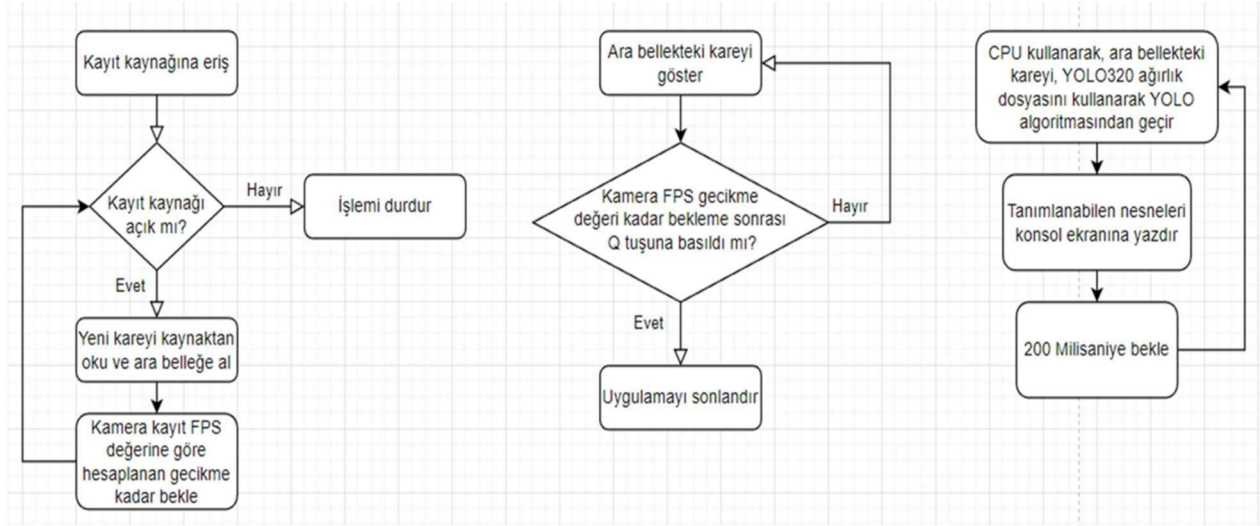
Şekil 2 - Asenkron görüntü alma testi

Asenkron kare alma denemesinin başarılı olması sonrasında sıradaki adım YOLO algoritmasını kullanarak ilk demoyu oluşturmaktı. Küçük cihazlarda hızlı çalışabilmesi için daha yüksek performansla çalışan YOLOV3 320p versiyonunu kullanmaya karar verdik. Algoritmayı ve ağırlıkları nasıl kullanabileceğimiz konusunda Murtaza's Workshop – Robotics AI isimli sosyal medya kanalının sağladığı kaynağı referans alarak, Darknet tarafından sağlanan COCO Names isimli nesne sınıfları ile ilk nesne tanımlama demomuzu oluşturduk.



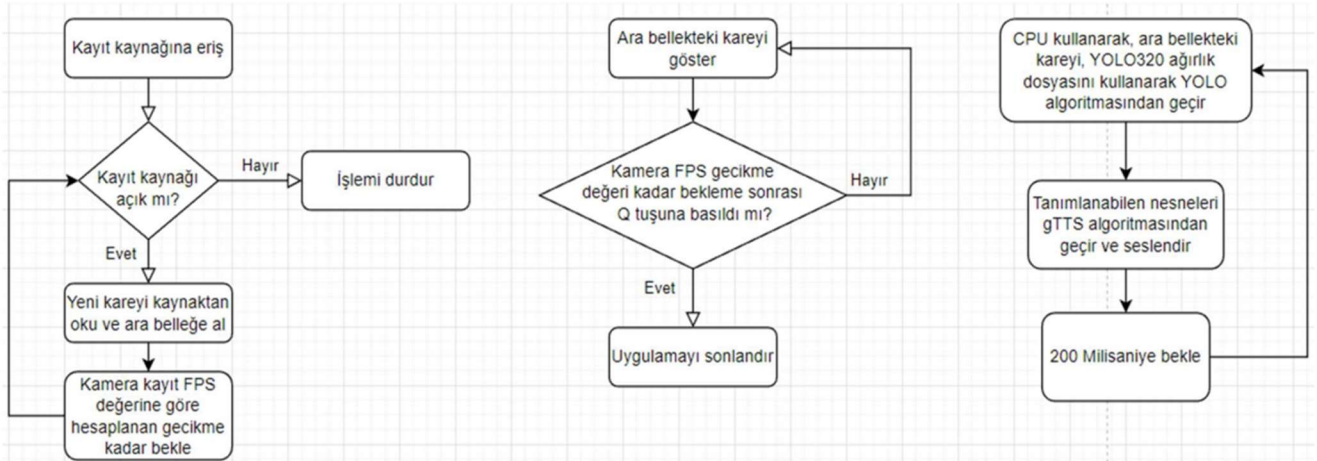
Şekil 3 - YOLO İlk Deneme

Burada CPU kullanma sebebimiz, küçük cihazlardakine benzer performansta bir sonuç elde ederek nasıl çalışacağına dair bir fikir edinmektir. Bu işlemde YOLO algoritmasından her bir kareyi geçirmek çok uzun zaman alıyordu. Her bir kare yaklaşık 1.5 saniyede sonuç verdiğinden ara bellekteki karelerimiz güncel olsa da nesne tanımlama işlemi eski kare ile işlem yapmış oluyordu. Bu sorunu çözmek için nesne tanımlama işleminin çalışma sıklığını bütün kareler yerine bazı karelerde çalışacak şekilde azaltıp, asenkron olarak çalışan farklı bir fonksiyona taşımaya karar verdik.



Şekil 4 - Yavaşlatılmış YOLO İşleme Denemesi

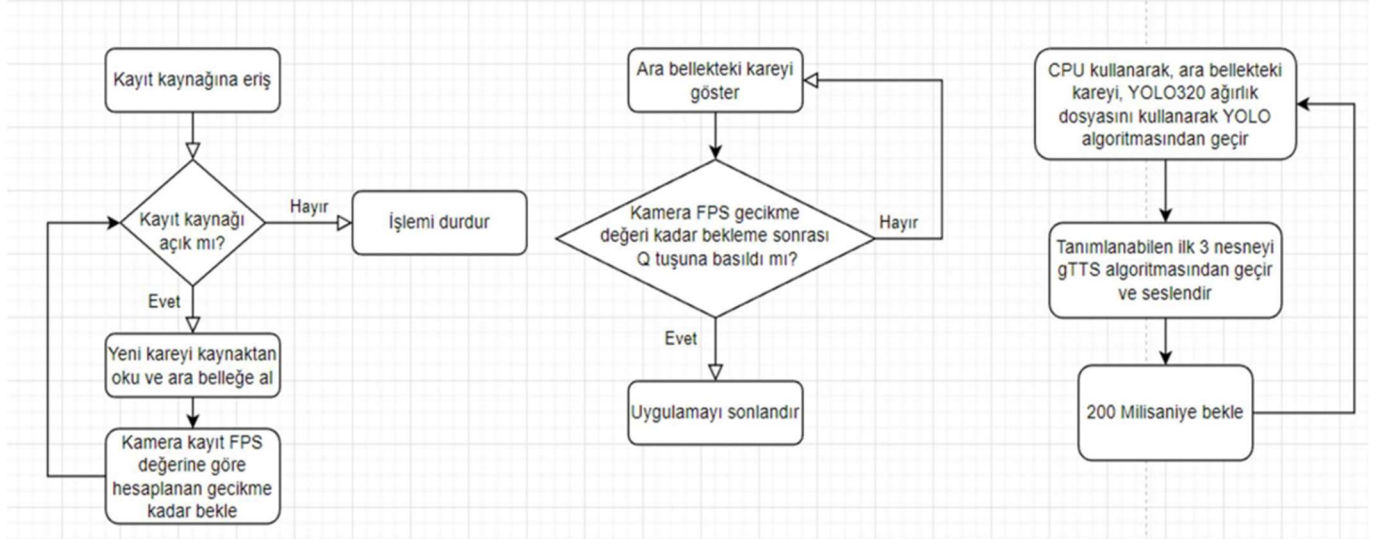
Burada elde ettiğimiz sonuçta hem görüntüdeki takılmalar büyük oranda azaldı, hem YOLO çalışma sıklığı saniyede 5 defaya indiği için performans yükseldi. Sıradaki aşamada tanımlanan nesneleri seslendirmeyi denedik.



Şekil 5 - Nesne Seslendirme Denemesi

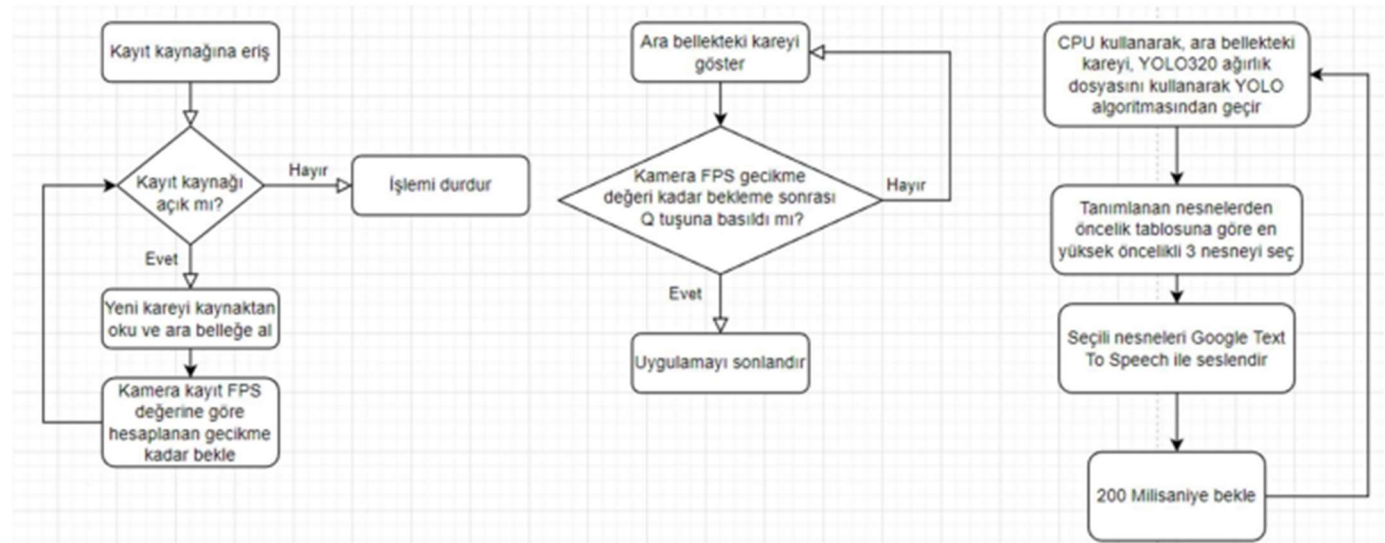
Anlık görüntü üzerinde tanımlanan her bir nesneyi seslendirmek Google Text To Speech kütüphanesi kullanarak yaklaşık 1 saniye vakit almakta. Bu durumda eğer tanımlanan nesne sayısı

fazla olursa sıradaki karenin okunması gecikmekte. Bu sorunu çözmek için seslendirilecek nesneleri limitlemeye karar verdik.



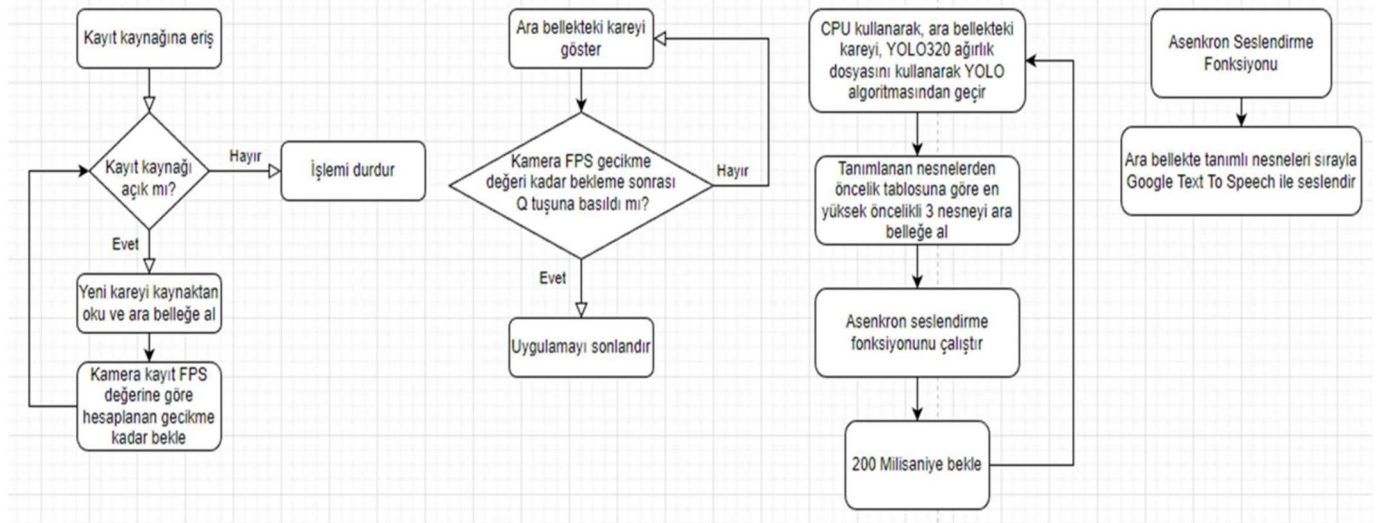
Şekil 6 - Limitli Sayıda Seslendirme Denemesi

Limitli sayıda nesne seslendirme yeterli oranda başarılı oldu ancak farklı bir sorunla karşılaştık. Tanımlanan nesneler rastgele sıra ile listeleniyordu ve bu da önemsiz nesnelerin daha önemli nesnelerin listede bulunmamasına sebep olabiliyordu. Bu bağlamda COCO Names isimli nesne sınıf dosyasından okunan verilerin her birine öncelik numarası atamaya karar verdik.



Şekil 7 - Öncelik Numarasına Göre Nesne Seçimi

Son olarak yapmamız gereken şey seslendirme aşamasının YOLO algoritmasının çalışmasını geciktirmesini engellemektir. Çünkü sesler oynatılırken görüntüde büyük çaplı değişiklikler olması durumu için YOLO algoritmasının arka planda en güncel nesneleri algılamaya devam etmesi gerekmektedir. Bunun için ses oynatma işlemini de asenkron olarak çalışan farklı bir fonksiyona taşımaya karar verdik.



Şekil 8 - Asenkron Seslendirme

Buradaki çalışmamız ilk demo için istediğimiz gibi sonuç vermekte. Sıradaki çalışmalarımız, buradan elde ettiğimiz sonuçlar ve bu sonuçlara göre tasarladığımız algoritma ile sistemin mobil uygulama üzerinde entegre edilmesi üzerine olacaktır.

4.2 İmplementasyonun Uygulanma Yöntemi

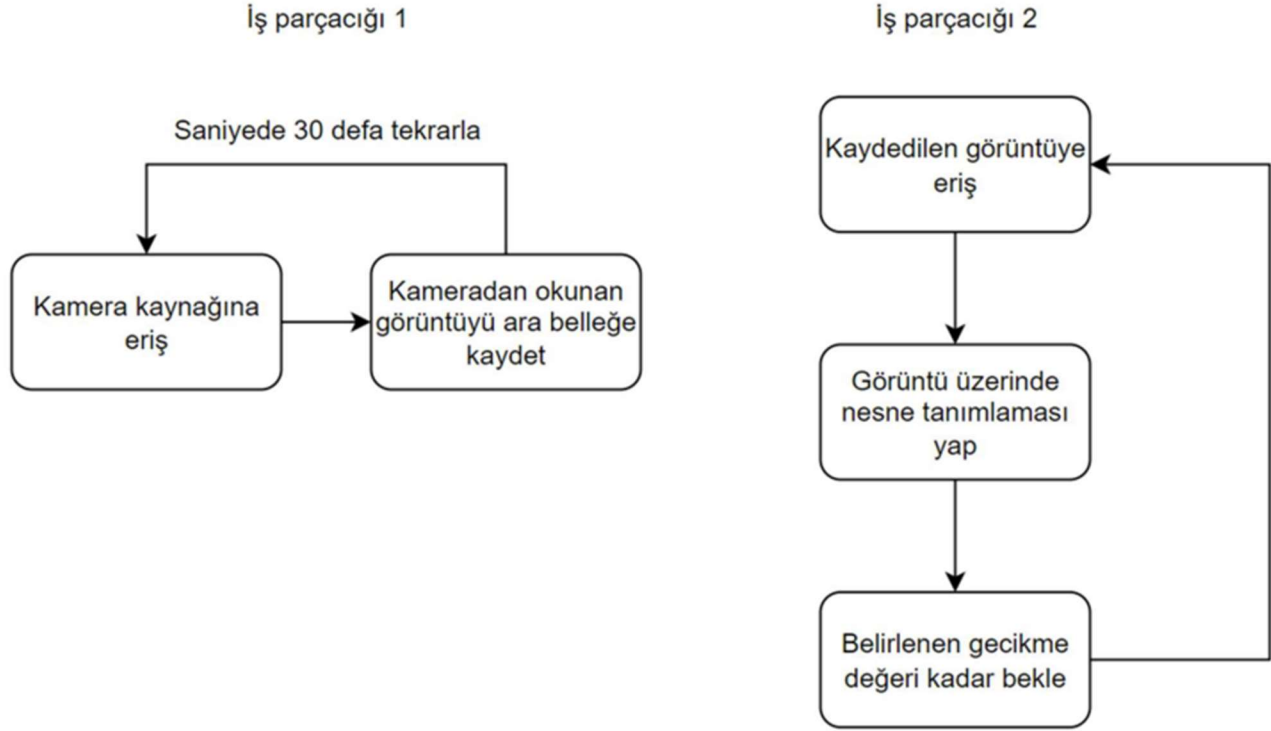
Proje hedefleri doğrultusunda ortaya konulacak ilk ürün, mobil cihazlarda görme engelli bireylerin kullanabileceği kontrol şemalarına sahip bir mobil uygulama olacaktır. Mobil uygulama geliştirme platformları arasında çapraz platform ile native(doğal, platforma özgü) uygulama geliştirme ortamları arasındaki performans farkının, Ihor Demedyuk ve Nazar Tsybulskyi tarafından gerçekleştirilen ve inheritasoft web sayfasında rapor halinde yayınlanan testlerin sonucunda 6 kata kadar çıktığı gözlemlenmiştir. Yapılan testler, bu projenin gerçekleştirilmesinde kullanılacak bazı görüntü işleme metotlarını da içermektedir. Bu sebepten, çapraz platform uygulama geliştirme ortamları tercih edilmemiştir. Bunun yerine mobil uygulama, native kodlama ile gerçekleştirilecektir. Bunun için Android işletim sistemi için sağladığı concurrent(eşzamanlı) fonksiyon kütüphanelerinin performansı ve OpenCV kütüphanesinin bu dil için bulunması dolayısı ile JAVA programlama dili kullanılacaktır. IOS işletim sistemi için yüksek performansı ve OpenCV kütüphanesini kullanabilmek üzere C++ kodlarını, Objective C dili ile arada köprü kurarak çalıştırmaya uygun yapısı dolayısıyla Swift programlama dili kullanılacaktır. Nesne tanımlama için başlangıç aşamasında YOLO algoritması kullanılacaktır. Tanımlanan nesneler ses aracılığıyla uygulamayı kullanan kişiye Google Text to Speech kütüphanesini kullanılarak iletilecektir. Uygulamanın test aşamalarında YOLO algoritması ve Joseph Chet Redmon'a ait web sayfasında açık kaynak olarak sağlanan Darknet framework'ü kullanılarak COCO(Common Objects in Context) veri seti üzerinde hali hazırda farklı çözünürlükler kullanılarak eğitilmiş modeller denenecektir. COCO veri seti, nesne tanımlama, segmentasyon ve etiketleme için kullanılan, açık kaynak olarak yayınlanan bir veri setidir. YOLO CNN (konvolüsyonel sinir ağları) kullanır. Nesne tespiti yapan bu algoritmanın açılımı "You Only Look Once (Sadece Bir Kez Bak)". Algoritmanın bu isimde olması, nesne tespiti yaparken tek bir kare üzerinde tek seferde tanımlama işlemini gerçekleştirdiğini ifade eder. WebCam, video ve resimlerde kullanılan YOLO, resimlerde tek kare üzerinde işlem yaparken video ve WebCam'da görüntüyü karelere ayırıp onlar üzerinde ayrı ayrı işlem yapar.

Proje hedefleri doğrultusunda geliştirilecek yazılım sisteminin, grafik işlemcisi zayıf olan veya grafik işlemci barındırmayan küçük cihazlar üzerinde çalışabilir olması gerektiğinden, ilk aşamada OpenCV ile CPU üzerinde çalışarak kameradan canlı görüntü alma performansı test edilecek ve optimizasyon yöntemleri araştırılacak. Görüntü işleme, derin öğrenme, yapay sinir ağları gibi sistemlerin hızlı çalışması için işlemlerin paralelleştirilmesi yani aynı anda yürütülmesi gerekmektedir. Çünkü çok sayıda yapay nöron ve çok sayıda piksel bulunduğundan bu seviyelerde yapılan işlemlerin sayısı milyonlara çıkmaktadır. Bu bağlamda GPU'lar çok sayıda(binlerce) çekirdek barındırdığından işlemleri paralelleştirmek yani aynı anda çalışacak hale getirmek CPU'lara göre çok daha hızlı sonuç verir. Ancak burada testlerin CPU üzerinde gerçekleştirilecek olmasının sebebi küçük ve zayıf cihazlarda alınacak sonucu canlandırmaya çalışmaktır. Performansı iyileştirmek amacıyla kameradan görüntüyü alacak iş birimi(thread) ile bu görüntüyü işleyecek birim asenkron olarak çalıştırılacak, performans ve sonucun doğruluğu anlamında optimum bir nokta bulunana kadar birim zamanda işlenecek kare sayısı kademeli olarak düşürülerek test işlemi gerçekleştirilecektir.

Mobil uygulama, gözlük ve sunucu işleme aşamalarının testi ve optimizasyonunun sağlanması aşamalarında nesne tanımlama işlemi için açık kaynak olarak yayınlanan Darknet

framework ve COCO veri seti kullanılarak eğitilmiş modeller kullanılacaktır. Bu doğrultuda kameradan görüntü okumak için ilk aşamada, bilgisayar CPU'yu frekans anlamında limitleyerek kameradan OpenCV kütüphanesi ile görüntü alma işlemlerinin performansı test edilecek. GPU yerine CPU kullanılma sebebi, mobil cihazlardaki küçük mikro işlemcilerin performansına yakın düzeyde sonuç alarak, küçük cihazlardaki performansı simüle etmek. Yapılacak test işlemleri için 320p, 416p, 608p çözünürlükler için tasarlanan modeller ile zayıf cihazlar için tasarlanmış tiny isimli model kullanılacaktır. Bu aşamada test ortamında elde edilen cihaz performansı ve nesne tanımlama doğruluk oranı verileri raporlanacaktır.

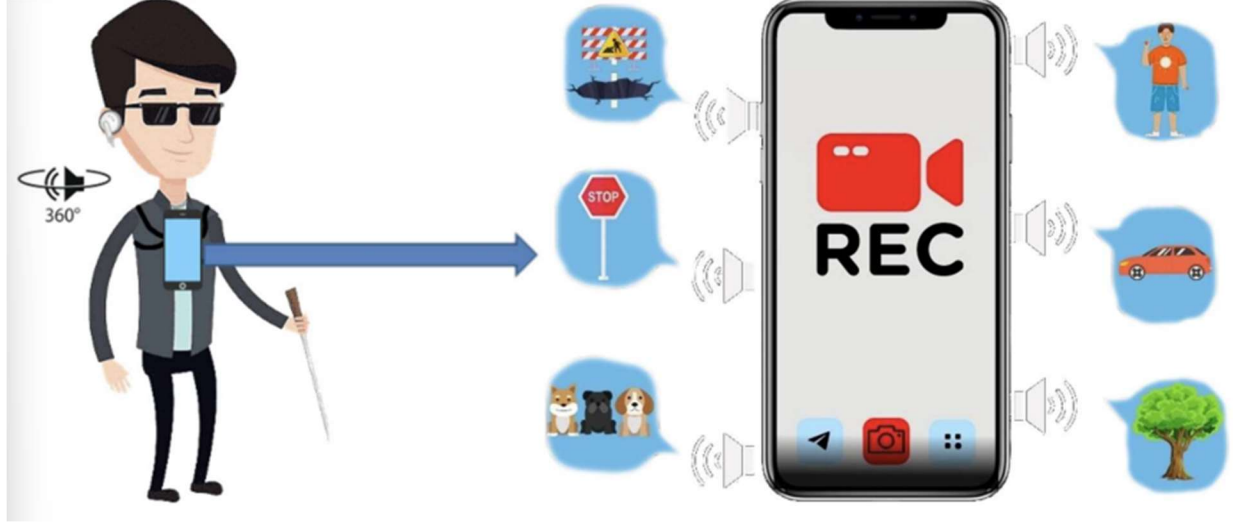
İkinci aşamada, raporu tutulmuş verilere dayanarak uygulama performansının istenenden daha düşük olduğu noktalarda bu duruma sebep olan etkenler saptanacak. Saptanan etkenlerden sistemsel olanların etkisi, paralel programlama konseptleri kullanılarak minimuma indirilecektir. Burada uygulama çoklu process ve çoklu thread olacak şekilde çalıştırılıp test edilecek. Çoklu thread halinde çalıştırırken paylaşımlı bellek (Shared Memory), çoklu process olarak çalıştırılırken mesaj gönderme (Message Passing) yöntemleri test edilecek. İki yöntemin de performans sonuçları raporlanacak ve daha iyi sonuç veren metot sonraki aşamalarda tercih edilecek. Yapay sinir ağı modelinin işlenmesinden kaynaklanan etkenin performans etkisi, birim zamanda üzerinde hesaplama yapılacak kare sayısı kademeli olarak düşürülerek minimuma indirilecek ve her bir kademe için elde edilen sonuçlar raporlanacaktır. Yazılım sisteminin ilerleyen aşamalarında elde edilen veriler üzerinden işlevsellik ve performans arasında optimum dengeye sahip olan birim zamanda kare işleme oranı değerleri tercih edilecektir.



Şekil 9 - Birim zamanda işlenen kare adedinin farklı değerlerde test edilmesi

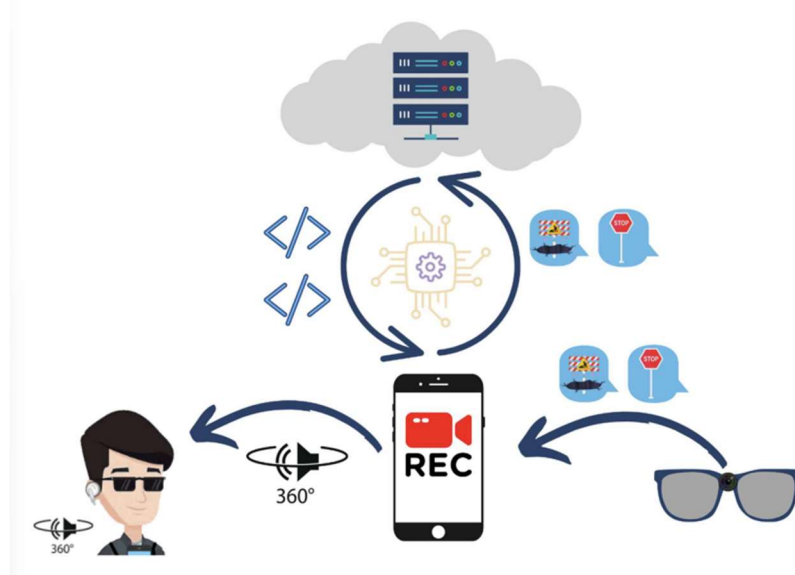
Üçüncü aşamada, uygulamanın resimler üzerinden elde ettiği verileri sesli olarak kullanıcıya iletmek için bir yazılım modeli tasarlanacaktır. Burada Google tarafından açık kaynak olarak tasarlanan Google Text To Speech kütüphanesi kullanılacaktır. Bir görsel içerisinde birden fazla tanımlama yapılabilir. Bu verilerin seslendirilmesi ancak sırayla yapılarak mümkün olur. Bu durumda ortaya çıkan problem, tanımlanan hangi varlığın daha önce seslendirilmesi gerektiğidir. Burada yapılacak ek çalışma, kullanılan varlık/nesne sınıflandırma kütüphanesinin elemanları arasında öncelik tablosu oluşturmak ve seslendirmelerin bu öncelikler doğrultusunda yapılmasını sağlamaktır. Örneğin, uygulamayı kullanan bireyin karşısında hareketli bir araba ve kedi bulunması durumunda, araba daha büyük bir tehlike arz ettiğinden daha öncelikli olarak seslendirilmesi gerekmektedir.

Sıradaki aşamada 3D ses simülasyon sistemleri araştırılarak, uygun bir ses simülasyon tekniği ile, sistemi kullanan bireyin, algılanan nesne/varlıkları bulundukları yönlerden algılaması sağlanacaktır. Böylece yön bilgisi için ekstra bir seslendirme yapılmamış olacak ve kullanıcının, yön bilgilerini doğru bir şekilde algılaması sağlanacaktır. Tasarlanacak 3D ses simülasyonunun, algılanan varlıkların/nesnelerin mesafelerine göre de yazılım sistemini kullanan bireye uyarıcı bir ses vermesi ve bu uyarıcı ses düzeyinin, algılanan mesafeye göre ayarlanması sağlanacaktır. Böylece örneğin mutfak gibi bir ortamda bulunan görme engelli birey, kafasını açık bırakılmış olan bir kapağa veya benzeri bir cisme yaklaştıracak olursa bir uyarı sesi duyarak bu nesneye çarpmaktan kaçınabilecektir.



Şekil 10 - Mobil uygulamanın temsili genel çalışma şeması

Şimdiye kadar yapılan çalışmaların sonucunda, varlık/nesne tanımlama modelinin küçük cihazlar üzerinde çalışma performansı, tanımlamaların doğruluk oranı ve gecikme değerleri bakımından raporlanacaktır. Bu raporlara dayanarak kullanılabilirlik olarak düşünüldüğünde performansın yetersiz kalması durumunda, geliştirilen yazılım sistemi, yüksek performanslı işlem gücüne sahip uzak bir sunucu üzerinde çalıştırılmak üzere düzenlenecektir.



Şekil 11 - Uzak sunucu kullanılarak çalıştırılacak sistemin temsili çalışma şeması

Uzak sunucu üzerinde çalışacak modelin gecikme değerlerini minimuma indirmek üzere çeşitli resim sıkıştırma yöntemleri, farklı internet hızları/bant genişliği ve farklı saniye başına iletilen görüntü sayısı değerleri ile test edilecek ve bu değerler test sonuçlarından elde edilecek hız/doğruluk oranına göre ayarlanarak ideal bir seviyede tutulmaya çalışılacaktır. Ortaya çıkacak ilk ürün mobil uygulama olacaktır. Şimdiye kadar anlatılan çalışmalar tamamlandıktan sonra üzerinde kamera bulunacak bir gözlük ve tasarlanan yazılım sisteminin, üzerinde yeterli performansta çalışabileceği, gözlük çerçevesine entegre edilebilir bir devre sistemi tasarlanacaktır.

4.3 Demo Uygulama

Başlık 4.1 ve 4.2’de yaptığımız çalışmalar ve araştırmalar sonucunda elde ettiğimiz veriler ve kararlar neticesinde ilk demo uygulamamızı Android işletim sistemi üzerinde yapmaya karar verdik.

4.2’de bahsi geçen 3D ses simulasyonu için kullanmaya karar verdiğimiz teknoloji, Waves firmasının NX ismini verdiği sanal çevresel ses teknolojisi. NX sanallaştırma teknolojisi, cihaz üzerinde oynatılan seslere yön algısı verilebilmesi için geliştirilen bir 3D ses sanallaştırma teknolojisidir. Firma, Dell, Audeze gibi büyük şirketlere de hizmet sağlamakta olup, bu alandaki başarısını kanıtlamıştır.

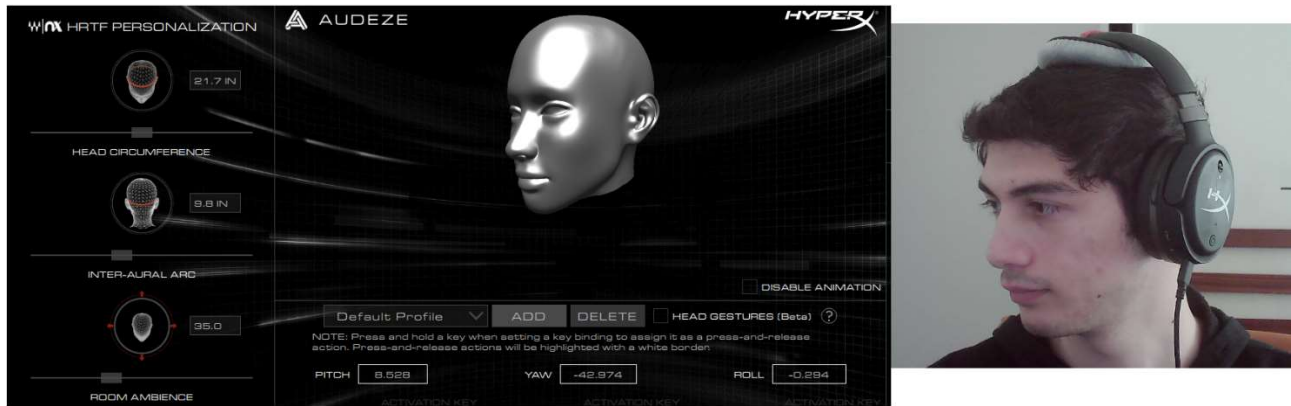
Projemizin geliştirme sürecinde, NX ses sanallaştırma ürün ailesi içerisinde bulunan NX head tracker isimli cihaz ve Waves NX mobile yazılımını ve ilk demo için içerisine NX head tracker sistemi entegre edilmiş ve hali hazırda elimizde bulunan Audeze Mobius / Cloud Orbit S model isimli kulaklığı kullanmaya karar verdik.

NX Head Tracker:

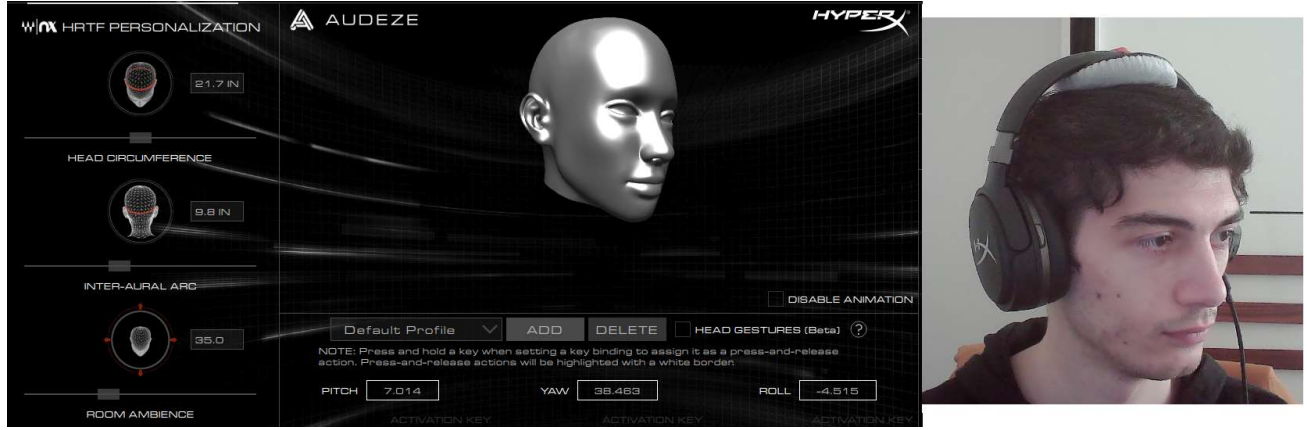
NX head tracker ürünü, kulaklığın üzerine tutturulan bir kablosuz kafa hareket takip cihazıdır. Kulaklığın üzerine tutturulur ve gerçek zamanlı olarak kafa hareketini takip eder. Elde ettiği verileri eşleştirildiği NX yazılımına Bluetooth üzerinden gönderir.

Waves NX Mobile:

NX mobil uygulaması, NX head tracker cihazından gelen verilere göre ses sanallaştırmasını gerçekleştirerek ses kaynağının yönünü, kullanıcının kafasının dönük olduğu yöne göre yeniden düzenleyip, 3D ses sanallaştırma teknikleri kullanarak, kullanıcının stereo bir kullaklık kullanmasına rağmen sesi sanki etrafında birden çok farklı yönde hoparlör varmış gibi, sesin tam olarak hangi açıdan geldiğini algılamasını sağlar.



Şekil 12 - NX Head Tracker Uygulama Görüntüsü



Şekil 13 - NX Head Tracker Uygulama Görüntüsü

Mobil cihazların performansları çok yüksek olmadığından, YOLO Tiny modeli ve Tensor Flow Lite kütüphanesi, proje amacına en uygun seçenekler olarak bulunmakta. Bu yüzden geliştirdiğimiz demo mobil uygulamada, Tensor Flow Lite kütüphanesi ve YOLO Tiny modelini kullandık.

Demo mobil uygulamada, görüntü işleme sonuçlarını analiz edebilmek için arayüz modülüne ek olarak bir kamera modülü bulundurmaya karar verdik. Proje tamamlanıp, son ürün olarak ortaya konulduğunda, görsel olarak sadece arayüz modülü bulunacak. Bu seçiminin sebebi, uygulamanın hedef kitlesi olan görme engelli bireylere, ekranda bir kamera görüntüsü göstermenin anlamsız olması, dolayısıyla telefon bataryasını ve performansını boşa tüketecek olması.



Şekil 14 - Demo mobil uygulama, ana ekran ve kamera modülleri

Uygulama çalışması esnasında kamera üzerinde algıladığı görsellerin X eksen konumlarına göre analiz yapmakta. Bu analiz sonucunda nesne isimleri sese dönüştürülmekte. Sese dönüştürülen veriler, çıkış ses aygıtına stereo kanallarının ses seviyeleri, X eksen konumuna göre değiştirilerek aktarılmakta. Buradaki amaç, nesne ne kadar soldaysa sesi o kadar soldan, nesne ne kadar sağda ise o kadar sağdan iletmek. Bu esnada devreye giren Waves NX head tracker cihazı, kullanıcı kafasını ilgili yöne çevirdiğinde sesi tekrar merkezlemektedir. Böylece görme engelli bireyin, karşısında duran nesnenin yönünü daha iyi algılayabilmesi sağlanacaktır.

Burada uygulamaya gelen X pozisyonu, en solda ise 0, en sağda ise kamera görüntüsünün X eksen maksimum boyutudur. Ancak ses kanalı ses ayarlaması [0, 1] kapalı aralığında yapılabilmektedir. Bu sebepten, elde ettiğimiz X pozisyonu değerini normalize edip, stereo kanalı ses ayarlamasını buna göre yapmaktayız. Uyguladığımız normalizasyon şu şekildedir:

$$x = \frac{\text{nesne } X \text{ pozisyonu}}{\text{kamera maksimum } X}$$

Burada elde ettiğimiz X değeri ile sağ kanal ve sol kanal ses seviyesini şu şekilde ayarlamaktayız:

$$\text{sol kanal} = 1 - x$$

$$\text{sağ kanal} = x$$

En son elde edilen ses ayarlamaları sonucunda elde ettiğimiz çıktı, NX head tracker destekli kulaklığa aktarılmaktadır.

Burada elde ettiğimiz sonuçlar, uygulama içerisinde Google Text To Speech kütüphanesini kullanacak şekilde yazdığımız sınıfın, ses oynatma metoduna gönderilmektedir. İlgili metod sesi, çoklu kanal olarak verebilmek üzere .wav formatında byte çıktısı alır ve bu çıktı, ses oynatan modül tarafından okunarak yukarıda bahsettiğimiz hesaplama ile elde edilen stereo ses seviyelerini ayarlayarak oynatır. Ses oynatma işlemi sonrasında kullanıcı kafasını ilgili yöne çevirdikçe, NX head tracker modülü ve 3D ses simülasyonu sonucunda ses merkeze yaklaşmakta ve bu sayede proje hedefindeki yön algısı elde edilebilmektedir.

5. KULLANILAN TEKNOLOJİLER

Projemizin geliştirme süreci boyunca kullandığımız ve ilerleyen süreçlerinde kullanacağımız teknolojiler şunlardır: Android Studio, Java, C++, IOS, Swift, NX Head Tracker, YOLO Algoritması, TensorFlow Lite kütüphanesi.

IOS:

Mobil cihaz piyasasında Android ve IOS işletim sistemleri, günümüzde kullanılan işletim sistemleri olduğundan ve projemizin yüksek performansla çalışması için native olarak geliştirilmesi gerektiğinden, ayrı ayrı iki işletim sistemini hedef alacak şekilde, Android için Java ile, IOS için Swift ile, ilerleyen aşamalarda alt seviye kodlama için C++ ile geliştirme yapmaya karar verdik. Hem Java hem de Swift tarafında C++ fonksiyonları ile iletişim kurarak çalışmak mümkündür.

IOS, Apple Inc. tarafından geliştirilen ve kullanıcıların iPhone, iPad ve iPod Touch gibi cihazlarda kullanabileceği bir mobil işletim sistemidir. IOS, 2007 yılında ilk iPhone ile birlikte piyasaya sürüldü. Apple'ın kurucusu Steve Jobs'un liderliğindeki ekibin çalışmaları sonucunda geliştirilen iOS, kullanıcı dostu arayüzü, yüksek performansı ve güvenliği ile dikkat çekti.

IOS, başlangıçta sadece Apple'ın kendi cihazlarında kullanılabilen bir işletim sistemi olarak tasarlandı. Ancak zamanla iPod Touch, iPad ve Apple Watch gibi diğer cihazlarda da kullanılmaya başlandı. IOS, App Store üzerinden kullanıcılara binlerce uygulama sunmakta ve Apple ekosistemi içinde entegre bir deneyim sağlamaktadır. Ayrıca, IOS güncellemeleriyle düzenli olarak yeni özellikler ve iyileştirmeler getirilmektedir.

Android Studio

Android Studio, Android uygulamalarının geliştirilmesi için resmi entegre geliştirme ortamıdır. İşte Android Studio hakkında daha detaylı bilgiler, tarihçesi ve özellikleri:

Android Studio, 2013 yılında Google tarafından duyurulmuş ve geliştiricilerin Android uygulamalarını daha kolay, hızlı ve verimli bir şekilde geliştirebilmeleri için tasarlanmıştır. Öncesinde, Android uygulama geliştirme için Eclipse gibi çeşitli entegre geliştirme ortamları kullanılmaktaydı. Ancak Android Studio'nun piyasaya sürülmesiyle birlikte, Android geliştirme süreci büyük ölçüde iyileştirilmiştir.

Android Studio, Java ve Kotlin programlama dillerini desteklemektedir. Java, geleneksel olarak Android uygulama geliştirmede kullanılan bir dil olmuştur. Ancak Kotlin, son zamanlarda popülerlik kazanan bir alternatif olarak kullanılmaktadır ve Android Studio'da tam desteklenmektedir.

Android Studio'nun özellikleri ve avantajları şunlardır:

1. Gelişmiş Kod Düzenleme: Android Studio, geliştiricilere zengin bir kod düzenleme deneyimi sunar. Akıllı kod tamamlama, hata ayıklama, kod navigasyonu ve otomatik düzeltme gibi özelliklerle geliştirme sürecini kolaylaştırır.

2. Görsel Tasarım Editörü: Android Studio, kullanıcı arayüzü tasarlama için güçlü bir görsel tasarım editörü içerir. Bu editör, kullanıcıların uygulama arayüzünü sürükle-bırak yöntemiyle kolayca oluşturmalarını sağlar.

3. Emülatör Desteği: Android Studio, geliştiricilere farklı cihazlarda uygulamalarını test etmek için entegre bir emülatör sunar. Bu sayede uygulamaların farklı ekran boyutlarında, cihazlarda ve Android sürümlerinde nasıl görüldüğünü görmek mümkün olur.

4. Derleme ve Dağıtım Araçları: Android Studio, geliştiricilere APK (Android Package) dosyası oluşturma ve uygulamaları Google Play Store gibi dağıtım platformlarına yükleme işlemlerini kolaylaştıran araçlar sunar.

5. Performans Analizi: Android Studio, uygulama performansını analiz etmek için çeşitli araçlar içerir. Hafıza kullanımı, işlemci performansı ve ağ trafiği gibi faktörleri izlemek ve optimize etmek mümkündür.

Android Studio'nun sürekli güncellenen bir yapıya sahip olduğunu belirtmek önemlidir. Google, düzenli olarak yeni özellikler eklemekte ve hataları düzeltmektedir. Ayrıca, Android Studio, Firebase, Google Cloud Platform ve diğer Google hizmetleriyle entegre çalışarak geliştiricilere geniş bir ekosistem sunmaktadır.

Android Studio, geliştiricilere güçlü bir araç seti sağlayarak Android uygulama geliştirme sürecini hızlandırır ve kolaylaştırır. Google'ın desteği ve geniş bir topluluğunun olması, Android geliştirme topluluğunun büyümesini ve yenilikçi uygulamaların ortaya çıkmasını sağlamaktadır.

NX Head Tracker:

NX head tracker isimli cihaz, kullanıcının kafa hareketlerini takip edip, mobil cihaza veya bilgisayara aktarmaya yarar. Bu cihaz sayesinde kullanıcının kulağına, sağ, sol, yukarı aşağı gibi yönlerden vermeye çalıştığımız seslerin, kullanıcı kafasını o bölgeye çevirdiğinde merkezlenmesi sağlanır. NX head tracker Bluetooth 4 ile çalışır. Canlı olarak kullanıcının kafasının hangi yöne dönük olduğunu eşlendiği yazılıma bildirir.

YOLO Algoritması:

YOLO (You Look Only Once) algoritması, resim üzerinde tek bir kareden tarama yaparak nesne algılayabilir. Çoğu alternatif nesne algılama algoritmasına kıyasla, bu özelliği sayesinde çok daha hızlı çalışır. Piyasadaki en hızlı resim üzerinden algılama yapabilen nesne algılama algoritmalarından biridir. Bizim projemizde hedef aldığımız cihazlar, mobil cihazlar olduğundan ve ilerleyen süreçte bir gözlük üzerine gömülü küçük donanımlar geliştirmemiz gerektiğinden, hızlı çalışan bir algoritmaya ihtiyacımız olmakta. Bundan dolayı projemizde kullanmayı kararlaştırdığımız nesne algılama algoritması YOLO olmuştur.

TensorFlow Lite Kütüphanesi:

TensorFlow Lite kütüphanesi, çeşitli makine öğrenmesi modellerini, Android cihazlar, mikro kontrolcüler ve gömülü sistemler gibi cihazlar üzerinde kullanılabilir kılmaya yarayan, bu işi hızlandırmak amaçlı geliştirilmiş bir kütüphanedir. Özel olarak geliştirilmiş, bellek kullanımını, çalışma gecikmesi değerlerini ve işlemciye binecek yükü minimum seviyede tutmak üzere çeşitli yöntemler kullanmaktadır. Bu özellikleri ve hedef aldığımız cihazlar tam olarak TensorFlow Lite'ın hedeflediği cihazlar aynı grupta olduğundan, makine öğrenmesi modellerini bu kütüphaneyi kullanarak çalıştırmaya karar verdik.

6. BULGULAR

```
public BorderedText(final int interiorColor, final int exteriorColor, final float textSize) {
    interiorPaint = new Paint();
    interiorPaint.setTextSize(textSize);
    interiorPaint.setColor(interiorColor);
    interiorPaint.setStyle(Style.FILL);
    interiorPaint.setAntiAlias(false);
    interiorPaint.setAlpha(255);

    exteriorPaint = new Paint();
    exteriorPaint.setTextSize(textSize);
    exteriorPaint.setColor(exteriorColor);
    exteriorPaint.setStyle(Style.FILL_AND_STROKE);
    exteriorPaint.setStrokeWidth(textSize / 8);
    exteriorPaint.setAntiAlias(false);
    exteriorPaint.setAlpha(255);

    this.textSize = textSize;
}
```

Şekil 15 - Demo uygulamanın nesne ismini kutu içine çizmesi için gerekli sınıf fonksiyonları

```
public Pair<Float, Float> getStereoRatio(float totalX, float x) {
    float ratio = x / totalX;
    return new Pair<>(1-ratio, ratio);
}

public void drawText(
    final Canvas canvas, final float posX, final float posY, String text, Paint bgPaint) {

    float width = exteriorPaint.measureText(text);
    float textSize = exteriorPaint.getTextSize();
    float textX;

    // Ekranı üçe bölerek metni orta bölgeye hizalama
    float screenThird = canvas.getWidth() / 3.0f;

    if (getObjectPosition(posX).equals("Solda")) {
        // Sol tarafta ise metni orta bölgenin solunda çiz
        textX = screenThird - width / 2.0f;
    } else if (getObjectPosition(posX).equals("Sağda")) {
        // Sağ tarafta ise metni orta bölgenin sağında çiz
        textX = 2 * screenThird - width / 2.0f;
    } else {
        // Orta bölgede ise metni orta bölgede çiz
        textX = canvas.getWidth() / 2.0f - width / 2.0f;
    }
}
```

Şekil 16 - Demo uygulamanın nesne ismini kutu içine çizmesi için gerekli sınıf fonksiyonları

```
text = getObjectPosition(posX) + " " + text;

Paint paint = new Paint(bgPaint);
paint.setStyle(Paint.Style.FILL);
paint.setAlpha(160);
canvas.drawRect(textX, (posY + textSize), (textX + width), posY, paint);
canvas.drawText(text, textX, (posY + textSize), interiorPaint);
Pair<Float, Float> ratio = getStereoRatio(canvas.getWidth(), posX);
sv.playText(text, leftChannel: 1.0f, rightChannel: 1.0f);
}

public void drawLines(Canvas canvas, final float posX, final float posY, Vector<String> lines) {
    int lineNum = 0;
    for (final String line : lines) {
        drawText(canvas, posX, posY: posY - getTextSize() * (lines.size() - lineNum - 1), line);
        ++lineNum;
    }
}

public void setInteriorColor(final int color) { interiorPaint.setColor(color); }

public void setExteriorColor(final int color) { exteriorPaint.setColor(color); }
```

Şekil 17 - Demo uygulamanın nesne ismini kutu içine çizmesi için gerekli sınıf fonksiyonları

```
public float getTextSize() { return textSize; }

public void setAlpha(final int alpha) {
    interiorPaint.setAlpha(alpha);
    exteriorPaint.setAlpha(alpha);
}

public void getTextBounds(
    final String line, final int index, final int count, final Rect lineBounds) {
    interiorPaint.getTextBounds(line, index, count, lineBounds);
}

public void setTextAlign(final Align align) {
    interiorPaint.setTextAlign(align);
    exteriorPaint.setTextAlign(align);
}
}
```

Şekil 18 - Demo uygulamanın nesne ismini kutu içine çizmesi için gerekli sınıf fonksiyonları

```
public class SpatialVocalizer {

    private static final int languageSuccess = 0;
    private static final int languageError = -1;
    private static final String ttsFileName = "ttsAudioFile.wav";
    private static final String Class_TAG = "SpacialVocalizer";
    private static final String TTS_TAG = "SpacialVoc... TTS:";

    private static final int PlaybackRate = 44100;

    TextToSpeech tts;
    private Context appContext;
    private float leftChannel = 1.0f;
    private float rightChannel = 0.0f;
    private boolean isPlaying = false;

    AudioFormat audioFormat = new AudioFormat.Builder()
        .setEncoding(AudioFormat.ENCODING_PCM_16BIT)
        .setSampleRate(PlaybackRate)
        .setChannelMask(AudioFormat.CHANNEL_OUT_MONO)
        .build();

    AudioAttributes audioAttributes = new AudioAttributes.Builder()
        .setContentType(AudioAttributes.CONTENT_TYPE_SPEECH)
        .build();
}
```

Şekil 19 - Demo uygulama, nesne seslendirme sınıfı implementasyonu

```
private ArrayList<Pair<String, Pair<Float, Float>>> textQueue = new ArrayList<>();

public SpatialVocalizer(Context context, Locale locale) {
    tts = new TextToSpeech(context, new TextToSpeech.OnInitListener() {
        @Override
        public void onInit(int status) {
            if (status == TextToSpeech.SUCCESS) {
                if (setLanguage(locale) == languageSuccess) {
                    setupTTS( pitch: 0.6f, speechRate: 1.0f);
                    appContext= context;
                } else {
                    Log.e(TTS_TAG,
                        msg: "TTS language is not valid");
                }
            } else {
                Log.e(TTS_TAG,
                    "TTS INIT ERROR, STATUS: ".concat(String.valueOf(status))
                );
            }
        }
    });
}
```

Şekil 20 - Demo uygulama, nesne seslendirme sınıfı implementasyonu


```
public int setLanguage(Locale locale) {  
    int result = tts.setLanguage(locale);  
    if (result == TextToSpeech.LANG_MISSING_DATA  
        || result == TextToSpeech.LANG_NOT_SUPPORTED) {  
        Log.e(TTS_TAG, msg: "LANGUAGE NOT SUPPORTED");  
        return languageError;  
    } else {  
        return languageSuccess;  
    }  
}
```

Şekil 21 - Demo uygulama, nesne seslendirme sınıfı dil ayarı fonksiyonu

```
public void setupTTS(float pitch, float speechRate) {  
    tts.setPitch(pitch);  
    tts.setSpeechRate(speechRate);  
  
    tts.setOnUtteranceProgressListener(new UtteranceProgressListener() {  
        @Override  
        public void onStart(String s) { Log.d(Class_TAG, msg: "UTR on start"); }  
  
        @Override  
        public void onDone(String s) {  
            Log.d(Class_TAG, msg: "UTR on done:");  
            byte[] audioData = readSound();  
            playSound(audioData);  
        }  
  
        @Override  
        public void onError(String s) {  
            Log.e(Class_TAG, msg: "UTR error on TTS");  
        }  
    });  
}
```

Şekil 22 - Demo uygulama, nesne seslendirme sınıfı TextToSpeech implementasyonu

```
private void playNext() {  
    if (textQueue.size() > 0) {  
        textToWav(textQueue.get(0).first);  
        textQueue.remove(index: 0);  
    }  
}  
  
public void playText(  
    String text,  
    float leftChannel,  
    float rightChannel  
) {  
    textQueue.add(new Pair<>(text, new Pair<>(leftChannel, rightChannel)));  
    textToWav(text);  
}
```

Şekil 23 - Demo uygulama, nesne seslendirme sınıfı text oynatma ve kuyruk yönetimi fonksiyonları

```
private void textToWav(  
    String text  
) {  
    String utrId = "spatialVocalizerUtrId";  
    Bundle params = new Bundle();  
    params.putString(TextToSpeech.Engine.KEY_PARAM_UTTERANCE_ID, utrId);  
  
    File path = appContext.getApplicationContext().getFilesDir();  
    File ttsFile = new File(path, ttsFileName);  
    tts.synthesizeToFile(text, params, ttsFile, utrId);  
}
```

Şekil 24 - Demo uygulama, nesne seslendirme sınıfı yazıyı wav formatına dönüştüren fonksiyon

```
private byte[] readSound() {  
    File path = appContext.getApplicationContext().getFilesDir();  
    File file = new File(path, ttsFileName);  
    byte[] audioData = new byte[(int) file.length()];  
    try {  
        InputStream inputStream = new FileInputStream(file);  
        inputStream.read(audioData);  
        inputStream.close();  
    } catch (IOException e) {  
        e.printStackTrace();  
    }  
  
    return audioData;  
}
```

Şekil 25 - Demo uygulama, nesne seslendirme sınıfı dosyadan ses okuma fonksiyonu

```
private void playSound(byte[] audioData) {  
    AudioTrack at = new AudioTrack.Builder()  
        .setAudioFormat(audioFormat)  
        .setAudioAttributes(audioAttributes)  
        .setBufferSizeInBytes(audioData.length)  
        .build();  
  
    at.write(audioData, 0, audioData.length);  
    at.setPlaybackRate(PlaybackRate / 2);  
  
    float left = textQueue.get(0).second.first;  
    float right = textQueue.get(0).second.second;  
    at.setStereoVolume(left, right);  
    textQueue.remove(0);  
  
    at.setNotificationMarkerPosition(audioData.length / 2);  
  
    at.setPlaybackPositionUpdateListener(new AudioTrack.OnPlaybackPositionUpdateListener() {  
        @Override  
        public void onMarkerReached(AudioTrack audioTrack) { Log.d(Class_TAG, "ON PLAY END"); }  
  
        @Override  
        public void onPeriodicNotification(AudioTrack audioTrack) {  
  
        }  
    });  
    at.play();  
}
```

Şekil 26 - Demo uygulama, nesne seslendirme sınıfı dosyadan okunan sesi oynatan fonksiyon

7. SONUÇLAR VE ÖNERİLER

Sonuç olarak dönem içerisinde proje geliştirme süreci boyunca çeşitli optimizasyon çalışmaları ve araştırmalar sonucunda mobil cihazlar üzerinde kameradan nesne algılayabilen ve bunu kullanıcıya nesnenin yönünü algılayabilecek şekilde iletebilen bir demo uygulama gerçekleştirdik.

Demo uygulama, iki farklı arayüz içermektedir. İlk arayüz, süreci başlatabilmek üzere gerekli komutu tuş aracılığıyla almaktadır. İkinci arayüz, sonuçların tarafımızca gözlenebilmesi için kamera modülünden gelen veriler üzerinden yapılan algılamalarla ilgili sonuçları ekrana bastırmaktadır. Görsellerin konumlarına göre ses sağa ya da sola doğru orantılı şekilde ayarlanmakta ve kullanıcı kafasını sesin geldiği yöne doğru hareket ettirdikçe ses merkeze yaklaştırılmaktadır. Yaptığımız testler sonucunda ses yönünü algılayabildik ve kullanıcının kafasını doğru yöne doğru hareket ettirebildiğini gözlemledik. Hedeflerimiz doğrultusunda sistemin çalışma performansını başarılı bulduk.

Projemiz, Tubitak 2209-A programı kapsamında destek almaktadır. Buraya kadar olan süreç boyunca ilk demo uygulamamızı gerçekleştirmeyi başarmış bulunmaktayız. Destek süreci boyunca projemizi geliştirmeye devam edeceğimizden, yol haritamızda, gerçekleştirilmesi gereken ek işlemler bulunmaktadır. Çalışmamız boyunca gerçekleştirmeye devam edeceğimiz iş paketlerimiz şunlardır:

- 1- Demo uygulamadan elde ettiğimiz sonuçları göz önünde bulundurarak, görme engellilerin kullanımına özel olacak bir arayüz tasarlanacaktır.
- 2- Uygulama, yeni tasarlanan arayüz ile hem Android hem de IOS işletim sistemleri için ayrı ayrı ve native (Java, Swift, C++ kullanan) geliştirme ortamlarında geliştirilecektir.
- 3- Üzerinde çevre algılaması için sensörler ve kamera bulunan fiziksel bir gözlük tasarlanacak, bu gözlük aracılığı ile çevre tanımlaması, mobil uygulamaya kıyasla çok daha etkin ve geniş ölçekte gerçekleştirilebilecektir. Bu sayede sistemi kullanacak birey, sistemden tam anlamıyla verim alabilecektir.

KAYNAKLAR

- 1- Waves Audio | Mixing, Mastering & Music Production Tools. (n.d.). waves.com.
<https://www.waves.com/>
- 2- Global blindness was slowing prior to pandemic study reveals | Orbis
<https://www.orbis.org/en/news/2021/new-global-blindnessdata#:~:text=2021%20has%20seen%20the%20official,%2Dto%2Dsevere%20visual%20impairment>
- 3- Aira and the Envision Glasses – Aira
<https://aira.io/envision/>
- 4- WeWALK Smart Cane – Smart Cane for the Visually Impaired
<https://wewalk.io/en/>
- 5- Flutter vs Native vs React-Native: Examining Performance, Ihor Demedyuk & Nazar Tsybulskyi,
<https://inveritasoft.com/blog/flutter-vs-native-vs-react-native-examining-performance>
- 6- Flutter vs React Native vs Native: Deep Performance Comparison, Ihor Demedyuk & Nazar Tsybulskyi,
<https://inveritasoft.com/blog/flutter-vs-react-native-vs-native-deep-performance-comparison>
- 7- Survival Strategies for the Robot Rebellion (pjreddie.com)
<https://pjreddie.com/>
- 8- Technology, D. (2021). Home - DeepSightTM. DeepSightTM.
<https://deepsight.com/>
- 9- Zeeshan, M. (n.d.). Home. IrisVision
<https://irisvision.com/>

ÖZGEÇMİŞ

Emirhan ÇİFTÇİ:

Ben Emirhan ÇİFTÇİ, 25 Kasım 1999 tarihinde Şanlıurfa'nın Eyyübiye ilçesinde dünyaya geldim. İlkokul 1. sınıf ve 2. sınıfı Karaköprü ilçesinde bulunan İMKB İlköğretim okulunda okudum. Sonrasında Gaziantep şehrine taşındık. 3. sınıfı Karataş İMKB ilköğretim okulunda 4-5-6-7 ve 8.nci sınıfları Gaziantep Turgut Özal Ortaokulunda okudum ve ortaokulu bu okulda tamamlamış oldum. Gaziantep'te girdiğim TEOG sınavının sonucunda Dumlupınar Anadolu Lisesi'ni kazandım. 9. sınıf boyunca okul adına Sırasıyla Dart, Basketbol, Bocce ve Bowling sporlarında temsil ettim ve dereceler yaptım. Lise yıllarımın geri kalanını Şanlıurfa, Ankara ve Gaziantep'te geçirdim sırasıyla GAP Anadolu lisesi, Sezai Karakoç Anadolu Lisesi, Faruk Nafiz Çamlıbel Anadolu Lisesi, Özel Karataş Uygur Temel Lisesi'nde okuyup mezun oldum. Üniversite sınav sonucunda Harran Üniversitesi Bilgisayar Mühendisliği bölümünü kazandım. Üniversite hayatımın ilk yılında Otonom ziraat araçları ve mobil uygulaması üzerine bir proje sayesinde Nasa Space Apps Chalange yarışmasında dördüncülük ve teknokent sponsorluğu kazandım. Daha sonrasında 1,5 senelik bir COVID-19 dönemi yaşadım ve online eğitime tabii tutuldum. 3. sınıf ve 4. sınıf sürecinde kendimi birçok alanda (Web tasarım, web programlama, Network, Yapay zeka, Mobil uygulama ve siber güvenlik) geliştirme fırsatı kazandım. İlk stajımı Tarım Kredi Teknoloji şirketinde Network ve Donanım üzerinde 30 iş günlük staj yaptım. Staj projeleri olarak ise Sanal Kampüs ağı ve Sanal Şirket ağı kurulum işlemleri ve konfigürasyon işlemlerini yaptım. Ek olarak da 4. sınıf okurken üniversite içerisinde HUBAP dairesinde bilgi işlemde yarı zamanlı olarak çalıştım. 4.Sınıfta ise bitirme projem olan Asistan Gözlük projesini Teknofest 2023 ve TÜBİTAK 2209-A Programlarına başvuruda bulunduk. Teknofest Yarı final hakem değerlendirmesinde 78 puan aldık ve yarı finalde elendik. TÜBİTAK 2209-A programından ise onay aldık ve 12 aylık süreçlik proje yapım sürecine girdik. 4.Sınıf sonunda ikinci 30 iş günlük stajımı Ayzasoft şirketinde yapıp mezuniyetimi elime alacağım. Bundan sonraki amacım ise kurmuş olduğum EQO Software adlı Şanlıurfa Teknokent Firmasının gelişimini ilerletmek ilgili olduğum alanlarda gelişmek ve faydalı bir insan olmaya çalışmak olacaktır.

Yusuf Yasin KAYA:

Ben Yusuf Yasin Kaya. 25.02.2001 tarihinde Şanlıurfa Merkez'de doğdum. Şu an aktif olarak Harran Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği programında son sınıf öğrencisiyim. 2007 yılında babamın tayini sebebi ile Uşak'a taşındık ve 1,5 yıl boyunca Uşak Bir Eylül İlköğretim Okulunda eğitim gördüm. Ardından Şanlıurfa'ya döndük ve sırasıyla İbrahim Tatlıses İlköğretim ve Vatan İlköğretim Okullarında eğitimimi tamamladım. Lise öğrenimimi Şanlıurfa'nın seçkin okullarından olan ÇEAŞ Anadolu Lisesi'nde tamamladım. Üniversite 2. sınıftan itibaren kendimi mesleğime ve gelişimime adadım. Üniversite 3. sınıftan itibaren Tiyatro Kulübü Başkanlığı yapmaktayım. Ayrıca 4. sınıfta TÜBİTAK 2209A programında ikinci döneme hak kazanırken Teknofest Engelsiz Yaşam kategorisinde yarı finalde elendik. 2022 yazında AyzaSoft Yazılım ve Donanım şirketinde staj yaptım ve stajımın akabinde benden memnun kaldılar ve orada sektöre atıldım. 1 yıldır AyzaSoft bünyesinde çalışıyorum. Ayrıca alanında uzman ve sektörde hâkim olan birçok yazılımcı/bilgisayar mühendisi ile çalıştım. Şuan ise Şanlıurfa

Emirhan ÇİFTÇİ

Yusuf Yasin KAYA

Metin Tarık KİKİ

Yusuf Ali BEKÇİ

Görme Engelleiler İçin Derin Öğrenme Tabanlı Mobil Uygulama ve Asistan Gözlük

Teknokent TGB-1'de kendi şirketim olan ICED Company'i kurdum ve burada çalışmalarımı sürdürüyorum.

Yusuf Ali BEKÇİ:

Ben Yusuf Ali Bekçi. 17 Ekim 1999 yılında Şanlıurfa'da doğdum. İlkokul ve ortaokul eğitimimi Şanlıurfa'nın Haliliye ilçesinde bulunan Cengiz Topel okulunda okudum. Lise eğitimimin ilk iki yılını Doğuş Kolejinde tamamladıktan sonra Çözüm Temel Lisesine geçiş yapıp orada lise eğitimimi tamamladım. Üniversite sınavı sonucunda Harran Üniversitesi Bilgisayar Mühendisliği bölümünü kazandım. Üniversite de tiyatro kulübü başkan yardımcılığı yaptım. 3.sınıfta Teknofest ve TÜBİTAK için projeler içinde buldum Teknofest'te yarı final, TÜBİTAK'ta ise proje onayı aldım. Şu an Şanlıurfa Teknokent bünyesinde şirketim bulunmak ve Amerika'da Directful isimli şirkette uzaktan(remote) staj yapmaktayım.

Metin Tarık Kiki:

4 Aralık 1997 tarihinde Ankara'nın Keçiören ilçesinde dünyaya geldim. Hayatımın önemli kısmını doğduğum bölgede geçirdim. İlk okul ve orta okul sürecimi toplamda iki defa okul değiştirerek tamamladım. Bu dönemlerin ardından lise sınavı sonrasında Kalaba Lisesi'ni kazandım. Lise sürecinde bir takım sağlık sorunları yaşayıp, devamsızlık sınıırım artık dolduğundan 3. yıl açık liseye geçmek durumunda kaldım. Bu süreç boyunca üniversite sınavına 3 yıl boyunca girmedim. 4. yıl ise üniversite sınavı için bir kursa kaydolup Harran Üniversitesi'ni kazandım. Üniversite okuduğum süreç boyunca birinci yılın ikinci dönemi pandemi süreci başladı ve toplam 1.5 yıllık bir süreç uzaktan eğitim ile devam etti. Üçüncü yıl yüz yüze eğitim tekrar başladı. 2022 yılı yaz stajımı Ayzasoft yazılım şirketinde WPF Framework'ü ile Windows ortamında yazılım geliştirmek üzere yaptım. Dördüncü yılımda 20 Şubat günü meydana gelen Kahramanmaraş depremi dolayısıyla son dönemim tekrar uzaktan devam etti. Bu süreçte Turkcell tarafından düzenlenen Gençlere Yatırım Geleceğe Yazılım projesi kapsamında IOS işletim sistemi üzerinde Swift programlama dili ile yazılım geliştirmek üzere eğitim aldım. Süreç sonunda Turkcell bünyesinde zorunlu stajımı gerçekleştirmek üzere şirket ile anlaşma sağladık. Şu anda son yılımdayım ve tüm derslerimi başarıyla verdiğim taktirde bu dönem mezun olacağım. Üniversite eğitimim boyunca C++, Python, Javascript, Java programlama dillerini öğrendim ve bu dillerle çeşitli projeler geliştirdim. Bu konular dışında, Görüntü İşleme ve Makine Öğrenmesi konularında kendimi geliştirip, bu alanlarda çalışmak istiyorum.