



MÜHENDİSLİK FAKÜLTESİ
YAPAY ZEKA MÜHENDİSLİĞİ BÖLÜMÜ

BİL 496 FİNAL RAPORU

ÜRÜN ADI

EVI (Eye of Visually Impaired)

TAKIM ADI

PIXELPUNKS

TAKIM ÜYELERİ

Zeynep Meriç AŞIK

Emre BELİKIRIK

Meriç DEMİRÖRS

İÇİNDEKİLER

1. <u>PROJENİN AMACI</u>	3
1.1. <u>Hedef Kullanıcı Kitlesi</u>	3
1.2. <u>Proje Fikri ile İlgili Ürün ve Hizmetler</u>	4
1.3. <u>Bizim Çözüm Önerimiz</u>	4
2. <u>KULLANICI SENARYOLARI & GEREKSİNİM ANALİZİ</u>	5
2.1. <u>Kullanıcı Senaryoları</u>	5
2.2. <u>Gereksinim Analizi</u>	7
3. <u>SİSTEM FONKSİYONLARI</u>	8
3.1. <u>APK</u>	8
3.2. <u>Server</u>	9
3.3. <u>YOLO</u>	9
3.4. <u>Depth Model</u>	10
3.5. <u>LLM</u>	11
4. <u>SİSTEM MİMARİSİ, UNSURLARIN İLİŞKİSİ VE DETAYLARI</u>	12
5. <u>TAKIM İÇİ İŞ BÖLÜMÜ</u>	17
<u>Meriç DEMİRÖRS</u>	17
<u>Emre BELİKIRIK</u>	17
<u>Zeynep Meriç AŞIK</u>	17
<u>REFERANSLAR</u>	18
<u>FİNAL DEMO</u>	18
<u>GITHUB REPOSITORY</u>	18

1. PROJENİN AMACI

Görme engelli bireyler yaşamları boyunca birçok engelle karşılaşır. Bu engellerden en zorlusu ise çevrelerindeki nesneleri ve engelleri algılayamama ve bu yüzden de hareket ederken veya bir işi yaparken zorlanmalarıdır. Sunumunu yaptığımız EVI projesi, görme engelli bireylere hayatlarındaki bu zorluklarda yardım etmek üzere tasarlanan bir projedir. Bunu gerçekleştirmek için birden fazla görüntü işleme modeliyle birlikte büyük dil modeli de kullanarak kullanıcıya sesli geri bildirim verir.

Kalabalık sokaklarda gezinmek, alışılmadık iç mekanlarda hareket etmek veya dışarıda açık alanlarda bulunmak olsun; görme engelli bireyler, güvenliklerini tehlikeye atan ve bağımsızlıklarını engelleyen çok sayıda durumla karşı karşıya kalırlar. Görme engelli bireylerin çevreyi algılama ve tanıma şeklinin sağlıklı kişilerden farklı olarak ses yoluyla olduğunu düşünürsek asıl sorun olan görsel bilgisizliği sözel bir yolla çözüm olarak bu bireylere sunmak gerekmektedir.

Geleneksel olarak görme engelli bireyler; bastonlar, rehber köpekler veya onlara yardımcı olacak ikinci bir kişi gibi dışarıdan gelecek yardımlara bel bağlamıştır. Bu yardımlar her ne kadar önemli olsa da, bireyin etrafındaki çevreyi; kişiye sözel olarak anlatabilecek, her zaman ulaşılabilir, gerçek zamanlı bir çözüm sağlamazlar. Dışarıdan alınan bu yardımlara rağmen görme engelli bireyler için en basit işleri bile yapmak sağlıklı bir insanla karşılaştırılamayacak derecede zordur.

Görme engelli bireyler ne yazık ki hayatlarını etkileyen çok sayıda engelle ve sınırlamalarla karşılaşır. Bu zorlukları yaşayan kişilere yardım etmek, onların güvenliklerini sağlamak ve onları başka yardımlara bağımlı olmayan özgür bireylere dönüştürmek toplum eşitliğinde önemli bir adımdır.

Kafa seviyesindeki nesneler, kaldırımdaki banklar veya yürüyüş yolunu tıkayan tehlikeli nesneler olsun, görme engelli bireyler bu engellerden kaçınmak için etraflarından yardım beklemektedir. Bu da hareket kabiliyetlerini önemli ölçüde kısıtlar sonuç olarak da kişisel özgürlüklerini sınırlar ve onları daha yüksek kaza, yaralanma ve hatta sosyal izolasyon riskine sokar.

Ayrıca çevreleri hakkında gerçek zamanlı bilgilerinin olmaması görme engelli bireylerin etraflarındaki dünyayı tam olarak deneyimlemelerini ve onunla ilişki kurmalarını engellemektedir, onları yeni yerler keşfetmek veya açık hava etkinliklerine katılmak gibi fırsatlardan mahrum etmektedir.

Bu nedenle bu soruna çözüm bulmak son derece önemlidir. Görme engelli bireylerin çevrelerini algılama ve anlama yeteneğini güçlendirerek; güvenliklerini, bağımsızlıklarını ve genel yaşam kalitelerini önemli ölçüde artırılabilir.

1.1. Hedef Kullanıcı Kitlesi

Dünya genelinde 323 milyon (%4,1), Türkiye genelinde 1,04 milyon (%1,4) kişi görme engellidir [1, 2] ve projenin kullanılabilirliği düşünüldüğünde başarısı halinde birçok bireyin yaşamını kolaylaştıracağı da ortadadır.

Görme engelli bireylerin güvenle gezinebildiği, dolaşabildiği veya bilmediği ortamlara kolayca girebildiği, çevresi hakkında bilinçli kararlar verebilecekleri bir dünya, toplumun ve görme engelli bireylerin refahını arttıracaktır.

Bu sorunu ele almak sadece görme engelli bireylerin hayatlarını iyileştirmekle kalmıyor, aynı

zamanda daha kapsayıcı ve erişilebilir bir toplumu da teşvik ediyor. Engelleri yıkmak ve görsel yetenekleri ne olursa olsun her bireyin çevrelerindeki dünyaya tam olarak katılabilmesini sağlamak için bir ortam hazırlıyor.

1.2. Proje Fikri ile İlgili Ürün ve Hizmetler

Bu konuda daha önceden yapılmış çözümlere göz atacak olursak farklı makaleler incelendikten sonra en çok kullanılan object detection/classification/localization modellerinin çoğunlukla YOLO varyantları olduğunu görülür [3, 4, 5, 6, 7]. Bunların yanı sıra sadece belirli alanlarda kullanılmak üzere tanımlanan sistemlerde derin öğrenme modellerinin yerine yerleşik sistemlerden de faydalanılarak geliştirilen projeler vardır.

Yöntemlerin neredeyse tamamı kurdukları sistemleri herkes tarafından kullanılabilir hale getirmek için telefon kameraları veya kafa kameraları gibi giyilebilir cihazlar üzerine bir sistem kurmanın önemine değinmiştir ve buna müsait olan projeler araştırılıp incelenmiştir [3, 10]. Kameralardan sahnenin yerleştirilmesini yapmak için derinlik sensörleri [4] ve ışık sinyalinin gidiş-dönüş süresinin ölçen (ToF) kameralar da kullanılan projeler vardır.

Bu yöntemlerde kullanıcıyı bilgilendirmek için işitsel [3, 4, 6, 7, 8, 9] ve/veya dokunsal [7, 8, 9] geri bildirimler kullanılır.

Ayrıca RGB resimlerden derinlik algılaması yapan modeller üzerinde de araştırma yapıp makaleleri inceledikten sonra genel olarak Unet tarzı bir encoder-decoder [11, 12, 13, 14] yapısı kullanıldığı görülmüştür. Bunun yanı sıra transformer [11, 12, 13, 14] gibi farklı teknikler kullanılan modeller de vardır.

Konuyla direkt olarak alakalı olan veya sadece belirli bir noktada fikir verebilecek konudan uzak olan birçok ayrı ayrı makale incelenmiştir, makalelerin büyük bir çoğunluğu görme engelli bireylere veya başka bir engeli olan bireylere yardım etme amaçlı projelerin makaleleridir. Sadece konuyla en alakalı olan makaleler “Referanslar” kısmında sıralanmıştır.

1.3. Bizim Çözüm Önerimiz

Önerisi sunulan proje, ihtiyacı olan insanlara yardım etmek için birden fazla modeli bir arada kullanan, makine öğrenimini içeren bir çözüm önermektedir.

Bireyin önündeki çevreyi sürekli olarak izleyen bir kamera kullanılarak ve ardından bu kameranın yakaladığı görsel verileri bir görüntü modeline besleyerek çevre hakkında bilgi sahibi olunmaktadır.

Çevredeki nesneler ve bireye olan mesafeleri hakkında bilgileri toplayan görüş modelleri tarafından üretilen çıktılar daha sonra büyük dil modeline (LLM) aktarılır. Bu model, girdileri analiz edip buna göre yanıt verir ve görme engelli kullanıcılara izlenebilecek mevcut yollar veya kaçınılması gereken potansiyel tehlikeler gibi önemli bilgileri iletir.

Bu sistemi uygulanabilir bir çözüm kılan şey rehber köpeklerinin aksine herkes tarafından ulaşılabilir olması, bastonların aksine çevreyi bir bütün halinde bireye yansıtması ve onlara yardımcı olan ikinci bir kişinin aksine her zaman ulaşılabilir olmasıdır. Böylece hem hareket etmesi zor olan mekanlarda hem de bireyin tanımadığı alanlarda gezinmenin yaratabileceği zorlukları ortadan kaldırmayı hedeflenmektedir.

Proje, bu sorunu doğrudan ele alarak görme engelli bireylere çevreleri hakkında gerçek zamanlı bilgi sağlayarak onların hayatlarını kolaylaştırmayı amaçlamaktadır. Bireylere sağlayacağı çevresel

farkındalık onların güvenliklerini artırır, bağımsızlıklarını destekler ve nihayetinde genel yaşam kalitelerini artırır.

Proje kapsamında sunmuş olduğumuz çözüm önerisinin, daha önce geliştirilen örnek projelere göre benzerlik ve farklılıkları bulunmaktadır. Şu ana kadarki çözümler incelendiğinde çevredeki engelleri tanıma konusunda benzer yaklaşımlar izlense de eskiye nazaran değişik modellerin değişik şekillerde birleştirilip kullanılması farklılığı dışında sistemin en iç katmanında bir LLM modelinin çalışması önceki fikir önerileri ile aradaki en büyük fark ve en büyük özgünlüktür.

Önceki çözümler incelendiğinde akıllı telefon kamerası kullanımından YOLO varyantı object detection/classification/localization modelleri ve RGB resimden derinlik haritası çıkaran modellerin kullanımına kadar çoğu noktada ortak ve benzer yönler bulunmaktadır, ancak bu yöntemlerin yukarıda açıklandığı şekilde birleştirilip kullanıldığı bir sistem gözlemlenen araştırmalarda görülmemiştir. Yararlanılan modellerin kullanılış şekilleri bakımından proje önerisi bir özgünlük göstermektedir.

Özgün noktalar arasından en öne çıkanı ise önceki yöntemlerden farklı olarak sistemin kendisi tarafından belirlenen kurallar etrafında bir yönlendirme yapılmamasıdır. Bunun yerine kendisine verilen girdileri mantıksal açılardan düşünerek değerlendiren ve kullanıcıya buna göre bir geri dönüş sağlayan bir LLM modeli bulunmasıdır.

2. KULLANICI SENARYOLARI & GEREKSİNİM ANALİZİ

2.1. Kullanıcı Senaryoları

2.1.1. Kullanıcı Senaryosu: Üniversite öğrencisinin üniversite bahçesinden ve otoparkından geçip ana binaya ulaşması

1. Kullanıcının üniversite içerisinde bahçede oturuyorken ana binaya varmaya çalıştığını düşünelim.
2. Kullanıcı telefonundan uygulamayı açıp (telefon otomatik olarak server'a bağlanacak ve resimleri göndermeye başlayacak) ana binaya doğru bahçede ilerleyecek ve bu sırada da uygulamadan "şu kadar ileride ağaç var", "bu kadar sağında insan yürüyor biraz sola dön" gibi çevreyle ilgili geri dönüşler alarak herhangi bir engelle karşılaşmadan yeşillik alandan çıkabilecek.
3. Kullanıcının çevreden tamamen kopuk olmadığını, etrafı hakkında genel bir bilgisinin ve varsayımlarının olduğu düşünüldüğünde bu esnada karşılaşılabilecek merdiven gibi uygulama desteğinin yalnız başına yeterli olmayacağı durumlarda baston yardımıyla ilerledikten sonra otoparka geldiğini düşünebiliriz.
4. Otoparkta karşıdan karşıya geçmek için etrafı dinleyip telefonunu sola ve sağa çevirip uygulamadan herhangi bir engel uyarısı geri dönüşü almadığında güvenli bir şekilde karşıya geçişini gerçekleştirecektir.
5. Çevre hakkında bilgi sahibi olduğu için uygulamanın engeller hakkında geri dönüşlerinin ve bastonunun yardımıyla engellere takılmadan ana binaya kadar ulaşacaktır.

2.1.2. Kullanıcı senaryosu: Bina lobisinden bahçedeki masalara ulaşmak

1. Kullanıcının bina lobisindeki koltuklarda oturduğunu ve bahçedeki kamelyadaki masalara ulaşmak istediğini varsayalım.
2. Kullanıcı telefonundan uygulamayı açıp (telefon otomatik olarak server'a bağlanacak ve resimleri göndermeye başlayacak) koltuktan kalkar ve baston yardımıyla kapıya kadar ilerleyip kapıdan çıkar ve yerdeki kör patikalarını izlemeye başlar.
3. Patikaları izlerken telefon yardımıyla önünde bir insan veya hayvan olup olmadığını;

bahçedeki saksılardan, çöp kovalarından birine çarpıp çarpmayacağını sesli geri dönüş olarak alır ve yönünü ona göre ayarlar.

4. Kamelyaların bulunduğu bölgeye kadar baston ve uygulamanın yardımıyla geldikten sonra etrafı hakkındaki bilgilerini ve varsayımlarını da kullanarak kamelyaların olduğu tarafa dönerek o yöne doğru ilerlemeye başlayacaktır.
5. Kamelyalara yaklaştıkça uygulamadan "şu kadar ileride masa var", "bu kadar sağda sandalye var" gibi geri dönüşlerde alarak baston yardımıyla da masalardan birini seçip oturabilir.

2.1.3. Kullanıcı senaryosu: Dışarıda parkta yürüyüş yapmak

1. Kullanıcının içinde engeller barındıran bir parkta yürüyüş yapma istediğini düşünelim.
2. Kullanıcı telefonundan uygulamayı açıp (telefon otomatik olarak server'a bağlanacak ve resimleri göndermeye başlayacak) başlangıç noktasından parkın içinde ilerleyecek.
3. Bu sırada karşısına çıkabilecek insanların, bankların, kedi, köpek veya başka tür hayvanların, çöp kovalarının ve benzeri engellerin kendisine ne kadar uzaklıkta olduğuna dair sesli geri dönüşler alarak rotasına ona göre ayarlayacak.
4. Bu süreçte kullanıcı telefonunu sağa sola çevirerek veya bastonuyla genel çevre bilgisini kullanarak etrafındaki nesneler hakkında bilgi sahibi olabilir.
5. Böylece herhangi bir nesneye ya da insana çarpmadan patikada ilerleyebilecek ve yürüyüşünü tamamlayabilecek.

2.1.4. Kullanıcı senaryosu: Market alışverişi yapmaya gitmek

1. Kullanıcının yürüyüş mesafesindeki bir markete gitmek istediğini düşünelim.
2. Bu sırada evinin olduğu binadan çıkması ve sokaktaki kaldırımda kalarak markete ulaşması gerekiyor.
3. Kullanıcı telefonundan uygulamayı açıp (telefon otomatik olarak server'a bağlanacak ve resimleri göndermeye başlayacak) markete kadar ilerleyen, önceden bildiği rotayı takip etmeye başlayacak.
4. Bu sırada karşısına çıkabilecek kaldırımda yürüyen diğer yayaların, ağaçların, çöp kutularının ya da torbaların, yerde duran dükkan tabelalarının ve benzeri nesnelerin uzaklığı sesli geri dönüş olarak uygulama tarafından sağlanacak.
5. Bu süreçte kullanıcı kendisine bir nesnenin ya da insanın yaklaştığını hissettiği/duyduğu yöne telefonu çevirebilir ve herhangi bir engel uyarısına göre de yönünü değiştirebilir ve engeli atlatabilir.
6. Böylece duyduğu ve bastonuyla hissettiği durumları da göz önüne alarak, uygulamadan gönderilen bildirimlerle güvenle evinden markete gidip gelebilir.

2.1.5. Kullanıcı senaryosu: Site girişinden apartmana ulaşmak

1. Kullanıcının halihazırda açık olan uygulaması sayesinde sitenin giriş kapısından geçiş yapar.
2. Uygulama, site içerisindeki binaları, yolları ve önemli alanları tespit eder ve bu bilgileri telefonuna ileterek kullanıcıya rehberlik eder.
3. Kullanıcı, apartmanın girişine yaklaştığında, telefonunun sesli anlatımını dinleyerek giriş kapısının yerini öğrenir.
4. Kullanıcı, apartmanın içine girdiğinde, uygulamanın sesli rehberliği ile apartman içindeki yolları ve merdivenleri öğrenir.
5. Kullanıcı, dairenin bulunduğu kata ulaşmak istediğinde, uygulamanın sesli anlatımı sayesinde asansörü veya merdivenleri bulur.
6. Kullanıcı, dairenin bulunduğu kapıya yaklaştığında, telefonunun geri bildirimi ile hangi

yönde kapı olduğunu daha rahat şekilde anlar ve güvenli şekilde evine ulaşır.

2.1.6. Kullanıcı senaryosu: Evden otobüs durağına ulaşmak

1. Kullanıcı, evinden çıkıp otobüs durağına gitmeyi hedefliyor.
2. Evinden çıktığında, uygulamayı açar ve etrafının bilgisini almak için telefonla etrafını tarar.
3. Uygulama, kullanıcının evinin çevresindeki engel olabilecek nesneleri tanır ve bu bilgileri kullanıcıya ileterek rehberlik yapar.
4. Kullanıcı, otobüs durağına ulaşmak istediğinde, telefonunun sesli anlatımını dinleyerek doğru yönlendirmeleri takip eder.
5. Kullanıcı, yolda karşısına çıkabilecek kaldırımları, yaya geçitlerini ve kavşakları uygulamanın sesli rehberliği ile dikkatlice geçer.
6. Otobüs durağına yaklaştığında, telefonunun sesli anlatımı ile durağa doğru güvenle ilerler.

2.2. Gereksinim Analizi

Hedef Kullanıcı Kitle (1.1.) başlığı altında da değinilen milyonlarca insanın ortak problemine çözüm olarak geliştirmek istenilen çözüm, görme engelli bireylerin bulunduğu çevrede rahatça hareket etmelerine yardımcı olacak bir uygulama yapmaktır. Hemen hemen herkesin akıllı telefonlara sahip olduğu bu dönemde kullanıcının akıllı telefonundan çevresinin bilgisini arkada çalışan bir bilgisayara gönderip o bilgisayarda gönderilen bilgilerin analizini yapıp sonuçları bireye işitsel yolla ileterek yardım eden bir proje geliştirmek için gerekenler sırasıyla aşağıdaki gibidir:

- Görme engelli bireylerin bulundukları çevrede başkalarından bağımsız olarak kendi başlarına hareket edebilmeleri ve baston/rehber köpeği gibi dışarıdan alınan yardımların müsait olmadığı zamanlarda bile çevre hakkında kapsayıcı ve bireyi yönlendirici bir geri dönüş yapılabilmesi için bu proje önemlidir ve bu bireylerin istekleri ile ihtiyaçları göz önünde bulundurulduğunda da gereklidir.
- Çevrenin görsel verisini yakalamak için kamerası olan bir akıllı telefon gerekmektedir, ayrıca bireye işitsel geri dönüşler de bu cihaz üzerinden yapılacağı için tercihen hoparlör çıkışı olan veya kulaklıkla çıkış sağlayabilecek bir cihaz gerekmektedir.
- Akıllı telefondan bilgisayara görsel veri gönderebilmek için telefon üzerinden APK ile veri iletimi sağlanmalıdır, aynı şekilde bilgisayarda elde edilen çıktıların da telefona iletilmesi için metin iletimi sağlanmalıdır. Bu sebeple projenin kullanılabilirliği açısından veri iletişiminin hızlı ve eksiklik sağlanabileceği bir ortam projenin optimal performansı göstermesi için önemlidir.
- Arkada çalışacak olan bilgisayar, kendisine iletilen bilgileri farklı görüntü işleme modellerinden geçirip elde ettiği çıktıları uygun bir formatla LLM'e besleyeceğinden bütün bu işlemleri hızlıca yapıp LLM'den elde edilen sonucu görme engelli bireyin akıllı telefonuna gecikme yaşamadan göndermelidir. Bu işlemleri ve bilgi aktarmalarını uygulamanın çalışma mantığına uygun olarak gerçek zamanlı olmaya yakın bir şekilde gerçekleştirecek bir bilgisayar gerekmektedir.
- Kullanıcıya en iyi yardımı en makul sürede sunabilmek için kullanılan uygulamanın yanında bir de baston/rehber köpeği gibi bireyin kendisine işitsel yolla iletilen bilgileri sağlamasına yardımcı olabilecek dışarıdan bir yardım, uygulamanın çalışmasında ve işleminde yardımcı olacaktır.
- Projede en iyi performansın sağlanabilmesi için gerekli ortam özellikleri düşünüldüğünde sistemin uygun kullanım alanları modellerin doğru ve hızlı bilgi verebileceği ortamlardır,

bunun için de görüntü modellerine karmaşık ve kalabalık gelmeyecek, veri aktarımın hızlı ve eksiksiz yapılabilmesi için arkadaki bilgisayarla iletişimin net olacağı ortamlar projede optimal çözümlerin sunulacağı çevrelerdir. Modellerin yavaş ve/veya hatalı çalışacağı ortamlarda ise projenin kullanılabilirliği azalacaktır.

- Kullanıcının en güncel hizmeti alabilmesi için telefon üzerinden sisteme bağlanacağı APK'nın son versiyon APK olması proje açısından daha uygun olacaktır.

3. SİSTEM FONKSİYONLARI

3.1. APK

- Flutter modülü ile çalışır.
- Sesleri oynatmak için audioplayers modülünü kullanır.
- Kamera görüntülerine erişmek için camera modülünü kullanır.
- İnternet üzerinden app.py ile iletişim kurmak için http modülünü kullanır.
- Kullanıcıya sesli geri bildirim vermek için Flutter TTS modülünü kullanır.
- "/process_image" path'inden app.py'a cameradan yakalanan resimleri işler ve gönderir.
- app.py tarafından gönderilen metinleri "/get_ilm_output" path'i üzerinden alır ve kullanıcıya sesli geri dönüş sağlar.

Uygulamanın arayüzü görme engelli bireylere en uygun, olabildiğince basit ve kullanımı kolay olacak şekilde ayarlanmıştır. Kullanıcılar 2 ayrı durum arasında geçiş yapmayı sağlayan bir butonla uygulamayı başlatıp durdururlar. Uygulamanın 2 durumdaki arayüz görüntüleri aşağıdaki gibidir:

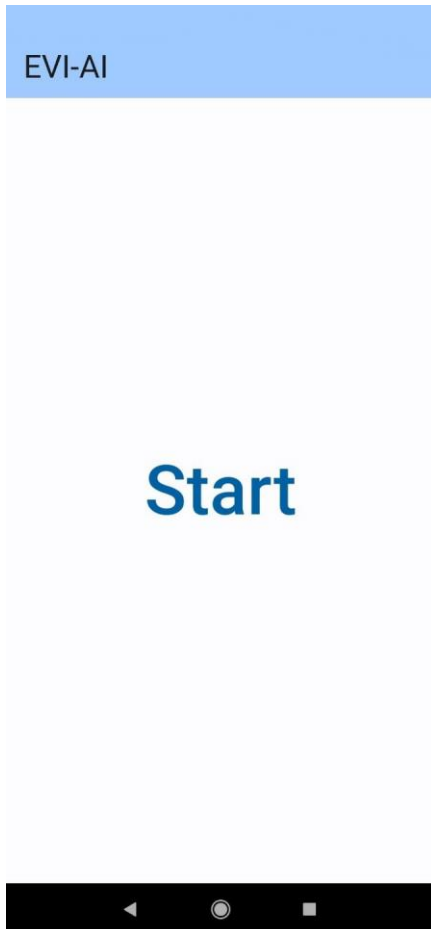


Fig. 1. Uygulama Açılış Arayüzü



Fig. 2. Başlatıldığı Andaki Arayüz görüntüsü

Uygulama ilk başlatıldığında “Press to Start” sesli geri dönüşüyle kullanıcıya uygulamanın kullanıma hazır olduğu bildirilir. “Start” butonunda basılmasıyla birlikte uygulama aktif duruma geçer ve kullanıcıya “Checking for obstacles, be cautious” sesli geri dönüş sağlayarak uygulamanın çalışmaya başladığını, kameradan görüntü almaya başladığını ve birazdan da server’den kullanıcının durumuna uygun yönergeler ileteceğini belirtir. Kullanıcı “Stop” butonuna basana kadar uygulama ve server arasında resim ve yönerge alışverişi devam eder, uygulama da yönergeleri aldığı an kullanıcıya sesli geri dönüş sağlar. Kullanıcı “Stop” butonuna bastığında tekrardan “Press to Start” sesli geri dönüşü ile uygulamanın dinlenme durumuna geçtiğini, artık kameradan görüntü almadığı gibi server’den da yönerge beklemediğini ve artık sesli geri dönüş sağlamayacağını kullanıcıya bildirir.

3.2. Server (app.py)

- Kullanıcıyla iletişim kurmak için Flask modülünü kullanır.
- "/process_image" path'inden APK tarafından gönderilen resimleri alır, işler ve user.py'ın okuması için ortak kullanılan bir lokasyona kaydedilir.
- user.py tarafından yine başka bir ortak lokasyona kaydedilen LLM text'ini okuyarak işler ve "/get_llm_output" path'i üzerinden APK'ya gönderir.
- Resim işlemek, diske kaydetmek için OpenCV modülünü kullanır.
- Resim işlemek için numpy modülünü kullanır.
- Resimleri kodlamak ve de kodlamak için base64 modülünü kullanır.
- Dosyaların güncelliğini kontrol etmek için os modülünü kullanır.

3.3. YOLO

- ultralytics modülünü kullanarak girdi resimler üzerinde obje tespiti, sınıflandırması ve lokalizasyonu yapılır.
- OpenCV modülü kullanarak modelin çıktıları üzerinden tespit edilen objeler hakkında geliştiriciye ekstra bilgiler sağlanır.

Daha detaylı anlatım için 2 ayrı girdi resim ile model çıktılarını inceleyecek olursak:



Soldaki resimde YOLO modelinin 2 motosiklet ve 1 insan tespit ettiğini görülür, ancak sağdaki resim için herhangi bir obje tespit edilmemiş.



YOLO çıktısı:

```
[((10, -30), 'person', 1),  
((20, -40), 'motorcycle', 2),  
((30, 40), 'motorcycle', 3)]
```



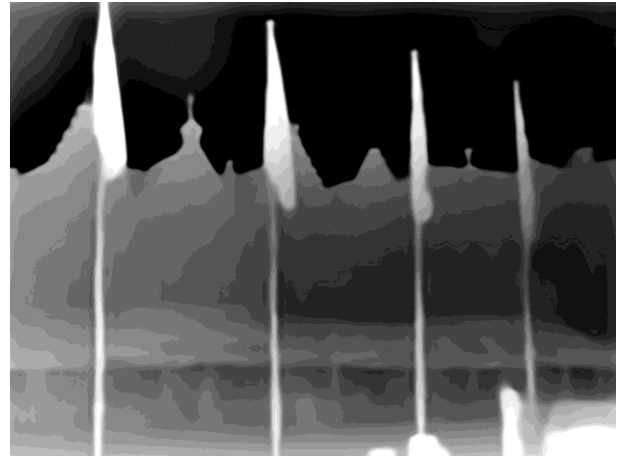
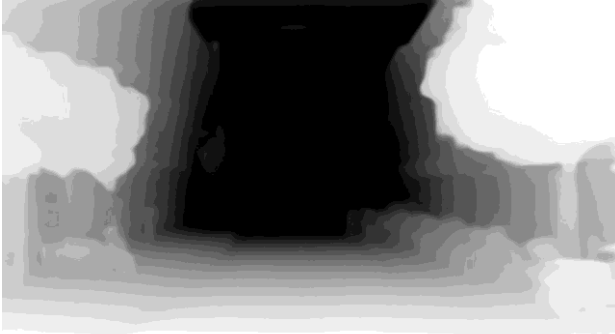
YOLO çıktısı:

```
[]
```

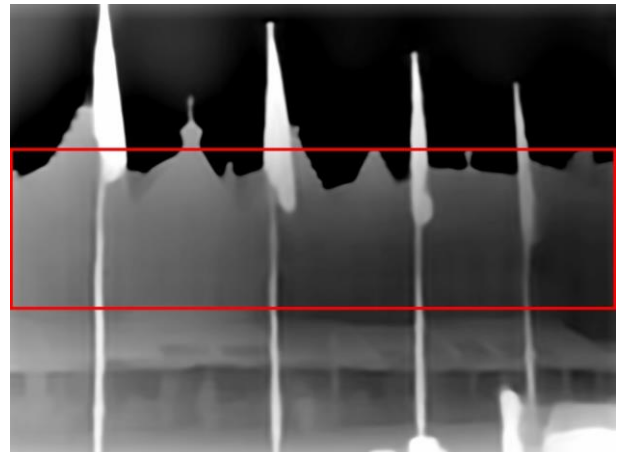
3.4. Depth Model

- torch modülünü kullanarak “ZoeDepth” derinlik tespiti modeli ile girdi resimler üzerinde derinlik tespiti yapılır.
- YOLO modelinin çıktısında tespit edilen objelere veya çıktı verilmemişse kullanıcının karşısına dikkat edilerek derinlik tespiti yapılır.

Daha detaylı anlatım için 2 ayrı girdi resim ile model çıktılarını inceleyecek olursak:



YOLO modelinin çıktısına göre derinlik tespiti yapılacak alanları gösterecek olursak:



YOLO ve DEPTH modelinin çıktıları bir prompt oluşturmak için birleştirilince ise elde ettiğimiz sahne bilgileri şöyledir:

[(5, -40, 2, "motorcycle"),
(3, 40, 3, "motorcycle")]

[(5, -30, id=0)]

YOLO modelinin obje tespit edip etmemesine göre, modelin objeye verdiği confidence score'una göre ve DEPTH modelinin tespit ettiği derinliğe göre eşik değerinin üstünde kalan tespitler (tehlike sayılanlar yani) sahne bilgisine dahil edilir. (3, 40, 3, "motorcycle") şeklindeki sahne bilgisi ise sırasıyla “metre uzaklık”, “sol sağ derece bilgisi”, “id” ve “obje adı” bilgileri taşır (YOLO tespiti olmayan sahnelerde “obje adı” bilgisi eksik olacaktır).

3.5. LLM

- YOLO ve DEPTH modellerinin çıktıları birleştirildikten sonra duruma göre promptlarla birleştirilir.
- Ollama ve langchain modüllerini kullanarak “llama2” büyük dil modelinden girdi promptlara kısa ve durumu özetleyici çıktı metinler elde edilir.

Yukarıda verilen sahne bilgilerinin hazır promptlarla birleştirilmesi üzerine modellere verilecek promptları gösterecek olursak aşağıdaki örnekler verilebilir:

“There is a system to help blind people, this system captures the obstacles ahead the user via camera and calculates how far and how many angles left or right are this obstacles from user. You should guide the user. Now I will provide the outputs of this system and you will give clear instructions based on where are the obstacles and guide the user with short sentences, don't output anything except guidance sentences. Here is the captured information of the scene(output only short sentences to guide user):

*motorcycle at 5 meters ahead at half left
motorcycle at 3 meters ahead at half right user
should go forward*

Now write a short sentence to guide the user.”

“There is a system to help blind people, this system captures the obstacles ahead the user via camera and calculates how far and how many angles left or right are this obstacles from user. You should guide the user. Now I will provide the outputs of this system and you will give clear instructions based on where are the obstacles and guide the user with short sentences, don't output anything except guidance sentences. Here is the captured information of the scene(output only short sentences to guide user):

*obstacle at 5 meters ahead at half left user
should go forward*

Now write a short sentence to guide the user.”

LLM modellerinin sayılar üzerinde istenilen doğrulukta çalışmadığını tespit ettikten sonra “sol sağ derece bilgisi” full left, half left, slightly left, forward, slightly right, half right, full right gibi

metinsel yönlere dönüştürülür. Ve eğer modeller tarafında engel tespit edilmezse prompt “no obstacle detected, go forward” şeklinde belirlenir ve LLM modelini es geçerek direkt kullanıcıya gönderilir.

Soldaki ve sağdaki resimler için sırasıyla LLM çıktılarını gösterecek olursak:

“Please proceed forward with caution, there is a motorcycle ahead at 5 meters and 3 meters to your left and right.”

“Please move forward slightly to avoid the obstacle at 5 meters ahead at left.”

Bunlar gibi örnek çıktılar kullanıcıya sesli geri dönüş sağlamak için internet üzerinden kullanıcıya geri gönderilir.

4. SİSTEM MİMARİSİ, UNSURLARIN İLİŞKİSİ VE DETAYLARI

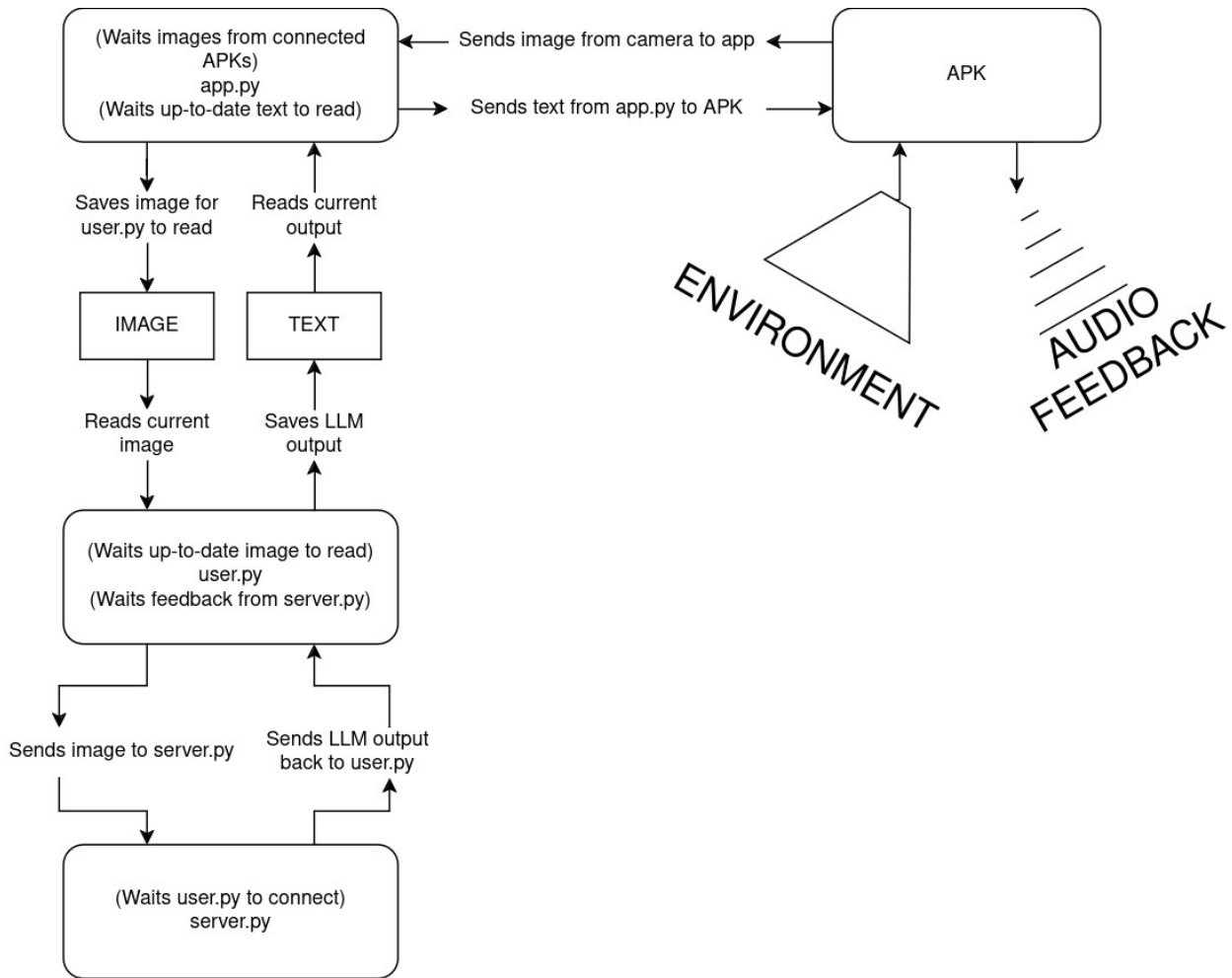


Fig. 3. Sistem Mimarisi

Sistem mimarisi 4 ana kısımdan oluşur, app.py, user.py ve server.py parçaları dağıtıcı tarafından kontrol edilir; kullanıcı sadece APK parçası ile ilgilenir. Sistemin çalışmasını adım adım anlatacak olursak sırasıyla:

1. APK internet üzerinden app.py'ya telefon kamerasından yakalanan görüntüyü gönderir
2. APK'lar tarafından bağlantı bekleyen app.py, bir resim teslim aldığı anda resmi işler ve user.py'ın kullanımına uygun şekilde ortak diske kaydeder.
3. app.py tarafından güncel bir resim kaydedilmesini bekleyen user.py, diske kaydedilen resmi okur ve TCP protokolüne uygun bir şekilde server.py'la RTSP bağlantısıyla resmi paylaşır.
4. user.py'dan bağlantı beklemekte olan server.py, RTSP üzerinden resim teslim aldıktan sonra ayrı bir subprocess başlatarak resmi işler ve işlemler sürecinde elde edilen LLM çıktısını tekrar aynı bağlantı üzerinden user.py'a iletir
5. server.py'dan LLM çıktısını teslim alan user.py, app.py'ın okuması için LLM çıktısını ortak diske kaydeder.
6. user.py'ın güncel bir LLM çıktısını diske kaydetmesini bekleyen app.py, LLM çıktısını diskten okuduktan sonra internet üzerinden APK'ya gönderir.
7. app.py'dan metin geri dönüşü bekleyen APK LLM çıktısını teslim aldıktan sonra kullanıcıyı uyarmak için telefon ile sesli geri bildirim verir.

Bütün bu işlemler bir döngü içerisinde birbirini/güncel veriyi bekleyen işlemler olarak devam eder.

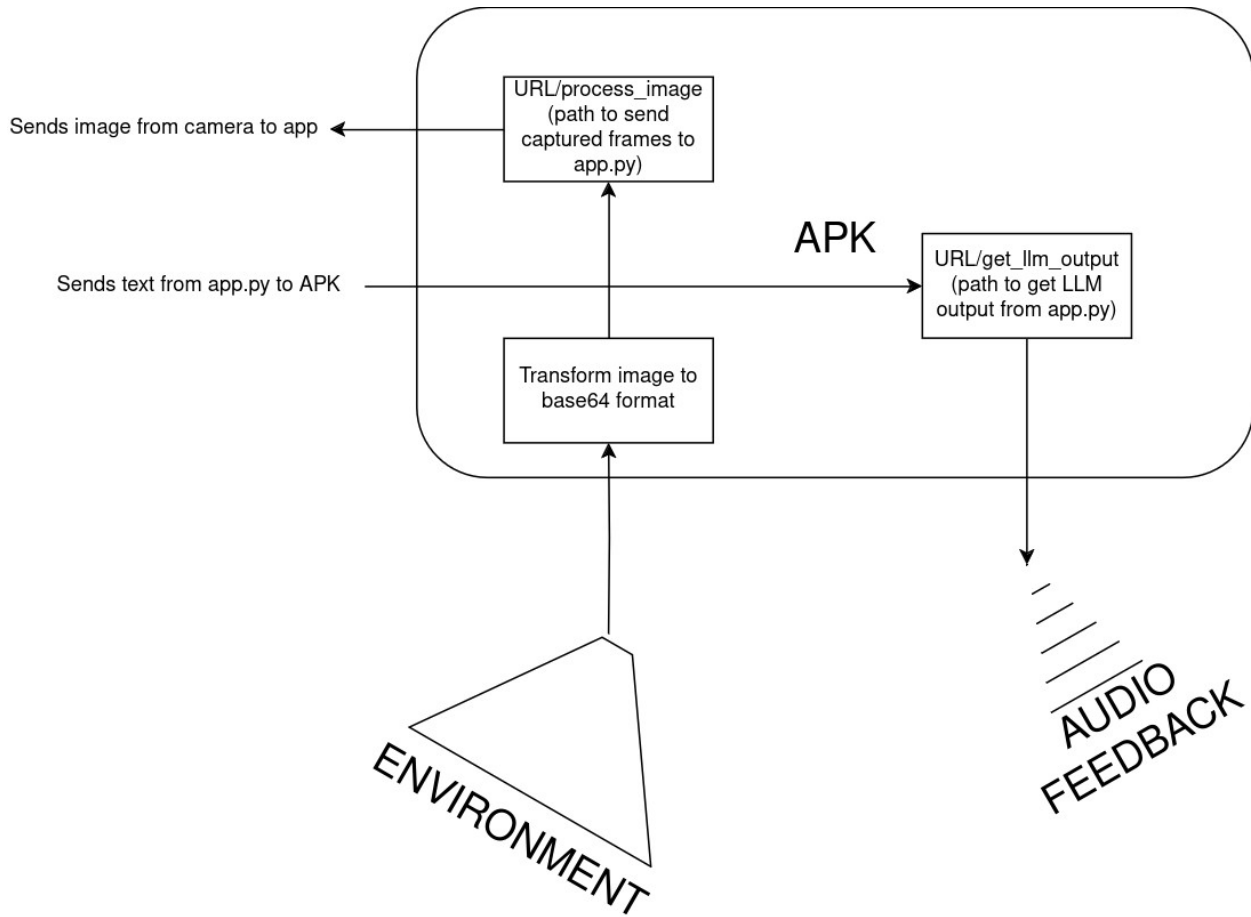


Fig. 4. APK-Backend İlişkisi

APK parçasını daha detaylı inceleyecek olursak resim gönderme ve metin teslim alma sırasında yukarıda bahsi geçen modüller ve paketler kullanılarak aşağıdaki adımlar izlenir:

1. Telefon kamerasından görüntü alınır.
2. Alınan görüntüyü internet üzerinden daha verimli iletebilmek için base64 formatına çevrilir.
3. app.py ile bağlantı kurulacak internet adresinin "process_image" path'ine base64 formatındaki veri iletilir.

1. app.py ile bağlantı kurulacak internet adresinin “get_llm_output” path’inden metin teslim alınır.
2. Telefonun ses çıkışı cihazından sesli geri bildirim verilir.

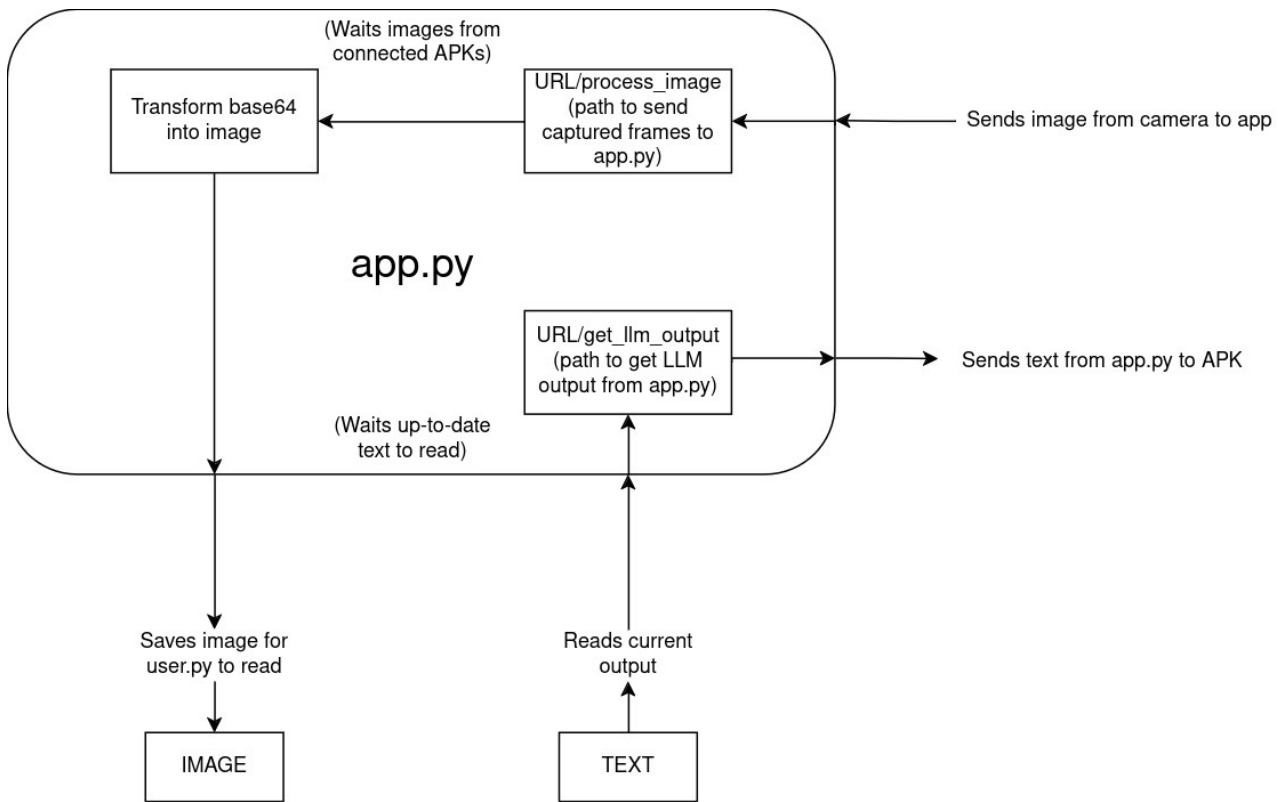


Fig. 5. APK Server-Backend İlişkisi

app.py parçasını daha detaylı inceleyecek olursak resim teslim alma/kaydetme ve metin okuma/gönderme sırasında yukarıda bahsi geçen modüller ve paketler kullanılarak aşağıdaki adımlar izlenir:

1. İnternet üzerinden iletişim kurulan internet adresinin “process_image” path’inden base64 formatında resim teslim alınır.
 2. base64 formatındaki resim RGB formatına çevirilir.
 3. RGB formatındaki resim user.py’nın kullanımı için ortak diske kaydedilir.
-
1. Ortak diskteki metin dosyasının son modifikasyon tarihi kontrol edilerek user.py tarafından güncel bir LLM çıktısı kaydedilmiş mi diye kontrol edilir.
 2. Diskteki dosya okunur ve internet adresinin “get_llm_output” path’i üzerinden APK’ya gönderilir.

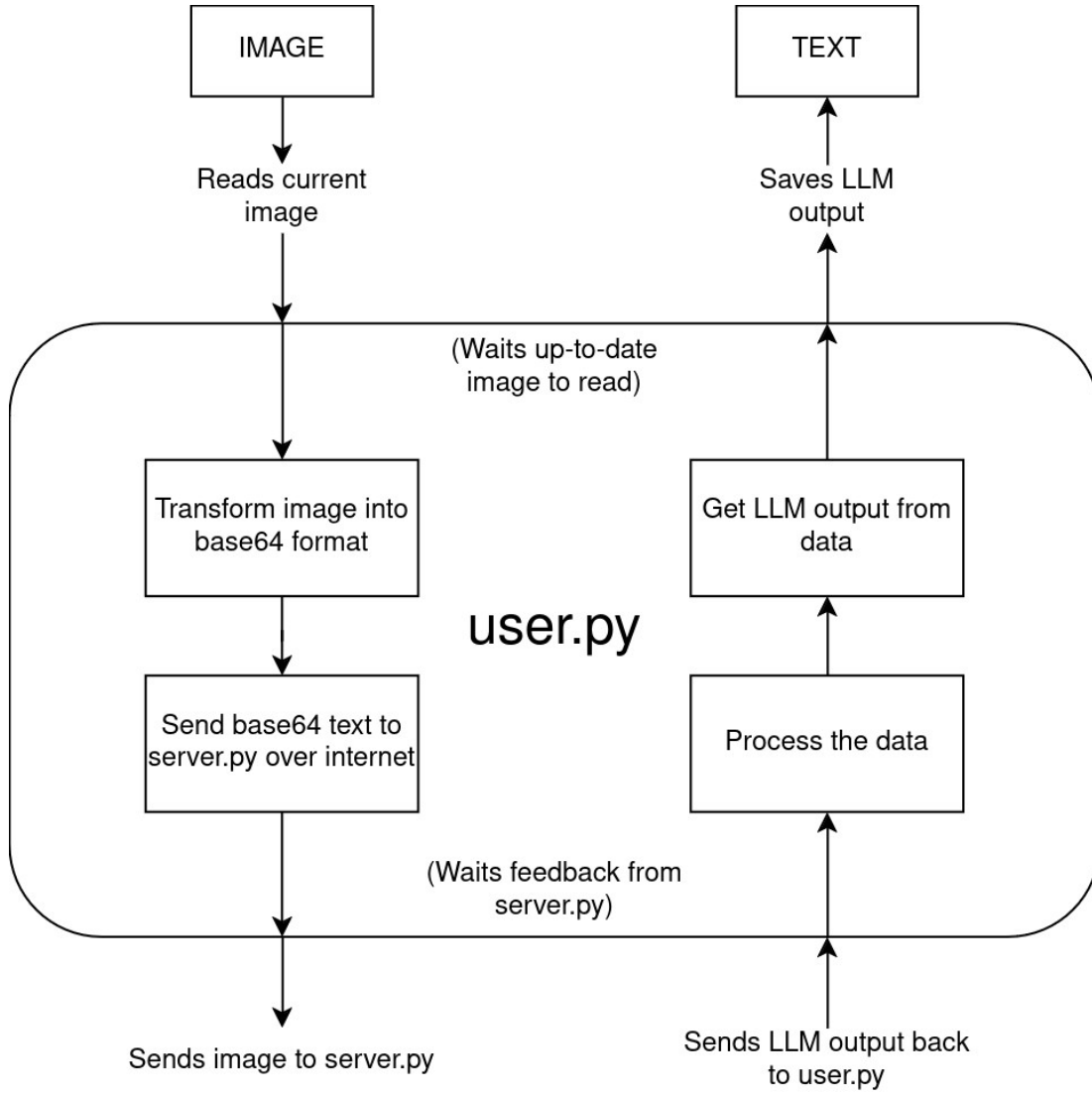


Fig. 6. Backend Unsurlarının Birbiriyle İlişkisi (user.py)

`user.py` parçasını daha detaylı inceleyecek olursak resim okuma/gönderme ve metin teslim alma/kaydetme sırasında yukarıda bahsi geçen modüller ve paketler kullanılarak aşağıdaki adımlar izlenir:

1. Ortak diskteki resim dosyasının son modifikasyon tarihi kontrol edilerek `app.py` tarafından güncel bir resim kaydedilmiş mi diye kontrol edilir.
2. Diskten okunan resim internet üzerinden daha verimli iletebilmek için base64 formatına çevrilir.
3. `server.py`'la kurulan internet bağlantısı üzerinden base64 formatındaki resmi içeren bir internet paketi gönderilir.
1. `server.py`'la kurulan internet bağlantısı üzerinden paket dinlenmeye başlanır.
2. LLM çıktısını içeren internet paketi alındığında paket işlenir
3. Paketten çıkartılan metin verisi `app.py`'ın okuması için ortak diske kaydedilir.

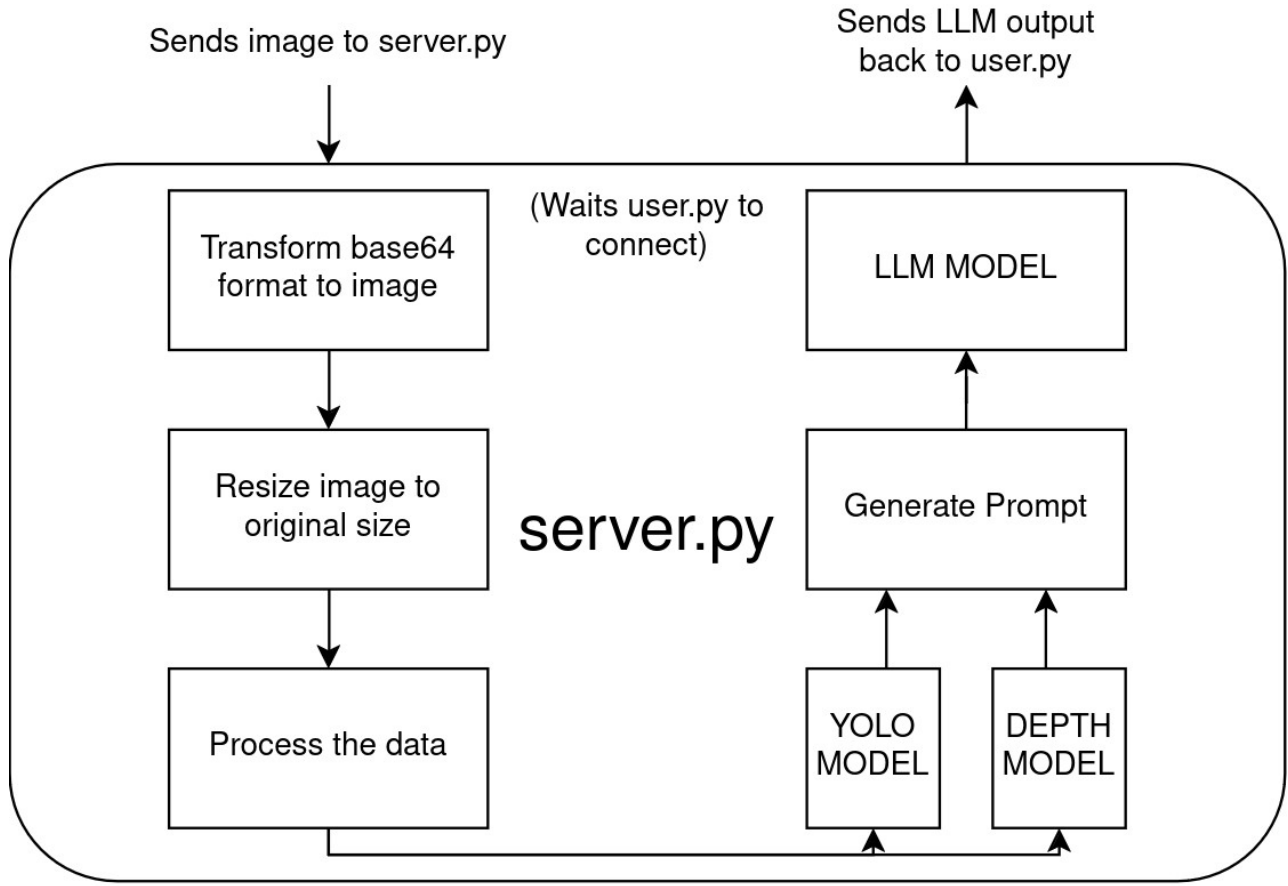


Fig. 7. Backend Unsurlarının Birbiriyle İlişkisi (server.py)

`server.py` parçasını daha detaylı inceleyecek olursak resim teslim alma/işleme ve metin üretme/gönderme sırasında yukarıda bahsi geçen modüller ve paketler kullanılarak aşağıdaki adımlar izlenir:

1. `user.py`'la kurulan internet bağlantısı üzerinden paket dinlenmeye başlanır.
2. base64 formatında resim verisi içeren paket alınır ve paket işlenir.
3. Paketten çıkartılan resim base64 formatından RGB formatına çevrilir ve orijinal boyutuna döndürülüp işlenir.
4. Elde edilen resim YOLO ve DEPTH modellerine beslenir.
5. YOLO modelinin tespit ettiği objeler hakkındaki verilerle DEPTH modelinin tahmin ettiği derinlik bilgisi birleştirilerek kullanıcının eğer yakında obje varsa: ne kadar yakında ve sağda/solda ne olduğunu veya yakında tanımlanamayan bir engel varsa: ne kadar yakında ve sağda/solda engel olduğunun bilgisi elde edilir.
6. Önceden denenip LLM modelinde istenilen tipte çıktılar verdiği test edilen prompt'larla elde edilen sahne bilgisi birleştirilir ve LLM'e beslenir.
7. `user.py`'la kurulan internet bağlantısı üzerinden LLM'den alınan metni içeren bir internet paketi gönderilir.

5. TAKIM İÇİ İŞ BÖLÜMÜ

Meriç DEMİRÖRS

- Pre-trained YOLO modelleri araştırması ve object detection, classification, localization kodlarının yazılması.
- Sunucu bilgisayarda kullanıcı ve sunucu tarafının işlemleri arasındaki iletişim için resim ve metin aktarımını sağlayacak TCP protokollü RTSP bağlantısı
- Sunucu bilgisayarda lokal olarak çalışacak LLM modeli araştırmasıyla kodların yazımı ve proje formatına uygun prompt geliştirmeleri,
- Sunucu bilgisayarda çalışacak kullanıcı ve sunucu tarafı işlemleri koşturacak kodların yazılması ve proje işlem hattının bu kodlara gömülmesi,
- Projeye uygun derinlik tespit modelleri araştırılması ve test edilmeye uygun ortam oluşturulması,
- Geçici olarak bir Text-to-Speech modülü eklenmesi

Emre BELİKIRIK

- Pre-trained Depth Estimation modellerinin araştırılması
- Frame üzerinden depth map oluşturulup tespit edilen objenin tahmini uzaklığının döndürülmesi
- Pre-trained YOLO modellerinin araştırılması ve depth modeli ile birleştirilip Object Detection + Depth Estimation birleştirilmiş prediction kodunun yazılması
- Farklı Depth ve YOLO modelleri için projeye uygunluk testi gerçekleştirilmesi
- YOLO'nun nesne tespit edemediği senaryoda Depth modeli üzerinden engel tespiti yapan yapının geliştirilmesi

Zeynep Meriç AŞIK

- Pre-trained depth modellerinin araştırılması
- Mobil uygulamanın arayüz tasarımı
- Mobil uygulamadan backend'e bağlantı kurulması (telefon kamerasından frame gönderimi)
- Backend'den mobil uygulamaya bağlantı kurulması (LLM çıktısı alımı)
- Mobile uygulama seslendirme işlemi (giriş, başlangıç ve text-to-speech olarak LLM çıktısı sesi)

REFERANSLAR

SOURCE FOR VISUALLY IMPAIRED PEOPLE STATISTICS:

- [1] <https://www.blindlook.com/tr/blog/detay/gorme-engelli-nufusu>
- [2] https://aile.gov.tr/media/130921/eyhgm_istatistik_bulteni_ocak_23.pdf

VISION MODELS FOR VISUALLY IMPAIRED PERSONS PAPERS:

MAGICEYE: AN INTELLIGENT WEARABLE TOWARDS INDEPENDENT LIVING OF VISUALLY IMPAIRED

- [1] Obstacle Detection, Depth Estimation And Warning System For Visually Impaired People
- [2] Embedded implementation of an obstacle detection system for blind and visually impaired persons
- [3] Real-Time Object Detection for the Visually Impaired
- [4] Assisting Blind People with Object Detection and Localization Using Deep Learning
- [5] AssistiveGlasses: A Wearable Object Detection System for the Visually Impaired
- [5] Machine Learning-Based Obstacle Detection and Navigation Assistance for Visually Impaired People

RGB TO DEPTHPMAP PAPERS:

- [1] LiteDepth: Digging into Fast and Accurate Depth Estimation on Mobile Devices
- [2] AdaBins: Depth Estimation using Adaptive Bins
- [3] BinsFormer: Revisiting Adaptive Bins for Monocular Depth Estimation
- [4] DepthFormer: Exploiting Long-Range Correlation and Local Information for Accurate
- [5] Monocular Depth Estimation
- [6] Transformer-Based Attention Networks for Continuous Pixel-Wise Prediction

[Final Demo](#)

[Github Repository](#)