

TEKNOFEST
HAVACILIK, UZAY VE TEKNOLOJİ
FESTİVALİ

AKILLI ULAŞIM YARIŞMASI
PROJE DETAY RAPORU

PROJE ADI: AIPARK

TAKIM ADI: AIPARK

TAKIM ID: T3-22183-200

TAKIM SEVİYESİ: Üniversite-Mezun

DANIŞMAN ADI: Dr. Öğ. Üyesi Adnan Özsoy

1. Proje Özeti (Proje Tanımı)

AIPark bilgisayarlı görü teknolojisi ile cadde ve sokaklardaki boş park yerlerini tespit eder. Kullanıcıların gitmek istedikleri yere tahmini ulaşma vakitlerinde boş park yeri bulma olasılıklarını hesaplar ve mobil uygulama aracılığıyla gösterir.

Park alanları kameralarla sürekli olarak izlenir. Boş park noktalarının konumu mevcut araçlardan öğrenilir, elle veri girilmesine gerek kalmaz.

AIPark'ın kaydettiği veriler otopark çıkışına park yapılması, izin verilen park süresinin aşılması, park yasağı olan yerlere park yapılması gibi ihlallerin saptanmasında da kullanılabilir.

2. Problem/Sorun:

Sosyal bir etkinliğe gitmek istediğimizde sokaklarda uygun bir park yeri bulamamak çok can sıkıcı bir hal alabiliyor. Valelerin birçoğumuz tarafından pek “güvenilir” bulunmadığı da bir gerçek. İşlek bir caddede boş park yeri ararken hem bizim hem de caddedeki diğer herkesin zamanı boşa harcanmış oluyor.

Ayrıca gün geçtikçe artan trafik problemine kalıcı çözümler bulabilmek için trafiğe çıkan araçların davranış rutinleri hakkında da bir fikir sahibi olunması gerekiyor. Bu da ortaya bir trafik datası ihtiyacı çıkarıyor. Mobese gibi kaynaklarından gelen trafik verisinin problemi ise yapılandırılmış veri olmaması.

AIPark olarak çözümlerimizi;

- Kullanıcılara en verimli park yeri bulma tecrübesini yaşatabilmek
- Yapılandırılmış bir trafik veri seti oluşturmak

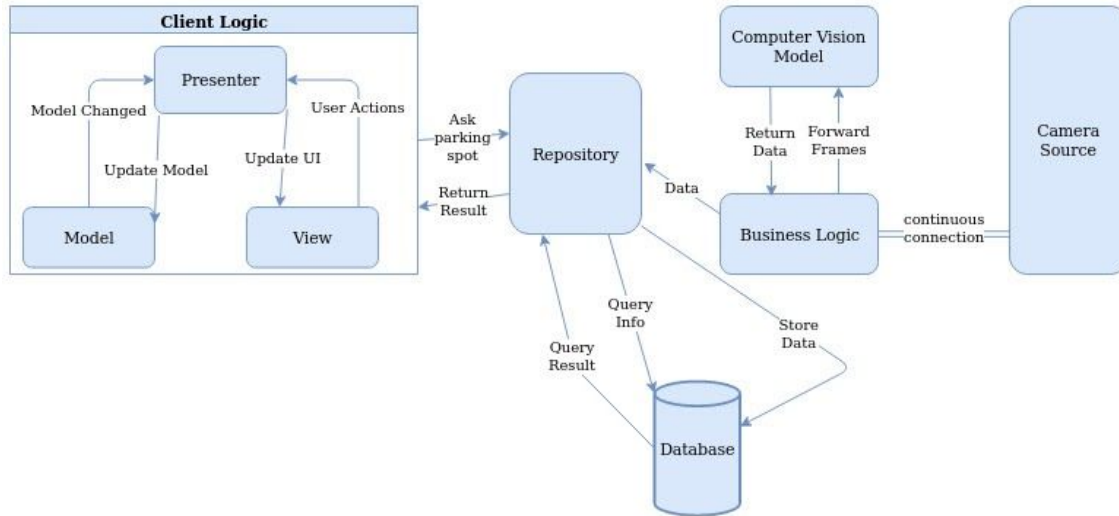
sorunlarına yönelik geliştirdik ve geliştirmeye devam ediyoruz.

3. Çözüm

Temel olarak, sokakları net bir açıyla izleyen kameralar aracılığıyla her dakika güncel park yeri durumunu takip eden bir sistem sunuyoruz.

Bir kullanıcı belirli bir cadde hakkında park yeri bilgisi istediğinde, o andaki boş park yerlerinin sayısı ve kullanıcının istenen yere ulaşma anında boş bir park yeri bulma olasılığı kullanıcıya iletilir. Bu olasılık noktaların önceki günlerdeki tekrar dolma sürelerinden hesaplanır. Kullanıcı bu bilgiler ışığında nerede park yeri arayacağına daha etkili bir biçimde karar verir.

Topladığımız dakikalık park yeri sayısı ve park yeri yeniden dolma süreleri şehrin farklı bölgelerindeki trafik yoğunluk durumunu devamlı bir şekilde inceleyip bununla ilgili başka akıllı çözümler üretilmesi konusunda bir altyapı sağlar.



İş Mantığı

Algoritma boş park yerlerini bulmak için zaten park etmiş durumda olan araçların konumlarını kullanır. Görüntüdeki konumu belli bir süre boyunca sabit kalan araçlar park etmiş sayılır, bu araçlar ayrıldığında eski konumları geçerli bir park noktası olarak kabul edilir. Duran araçların görüntüdeki konumlarının değişmemesi için kameranın sabit kalması gerekir.

Park etmiş bir aracın önüne bir engel geldiğinde yerinin artık boş sayılmaması veya boş sayılan bir park yerinin üzerinden bir araç geçtiğinde burasının kısa süreli olarak işgal edilmiş kabul edilmemesi için yeni boşalan ve yeni dolan park alanlarının zamanlayıcılara konması gerekir.

Bu işlemleri yapan bileşen yukarıdaki diyagramda Business Logic olarak adlandırılmış ve daha ayrıntılı olarak "[ekteki akış diyagramında](#)" da gösterilmiştir.

API

Diyagramda Repository olarak adlandırılan bu bileşen diğer bileşenler ile veritabanı arasındaki bağlantıyı sağlar.

İş mantığından HTTP istekleri ile gelen anlık boş park noktası sayısı ve boş nokta dolma sürelerini veritabanına işler. Mobil uygulamada canlı değişim olabilmesi için Firestore güncellemesi de burada yapılır.

Mobil uygulamadan istek geldiğinde boş yer bulma ihtimalini hesaplama işlemi burada yapılır. Bunun için şu anda boş olan yerlerin en az birinin kullanıcı ulaştığında boş olma ihtimali önceki yeniden dolma sürelerinden normal dağılım formülleriyle hesaplanır. O anda boş park yeri yoksa önceki günlerde verilen saatteki boş yer sayılarından yola çıkarak ihtimal hesaplanır.

Bilgisayarlı Görü Modeli

Verilen bir video karesinden görüntüdeki bütün arabaların karedeki koordinatlarını bulan bileşendir. Yan ve dikey görüntüyü destekler. Ayrıca Mask R-CNN ve YoloV3 teknolojileri arasında seçim yapmayı sağlar.

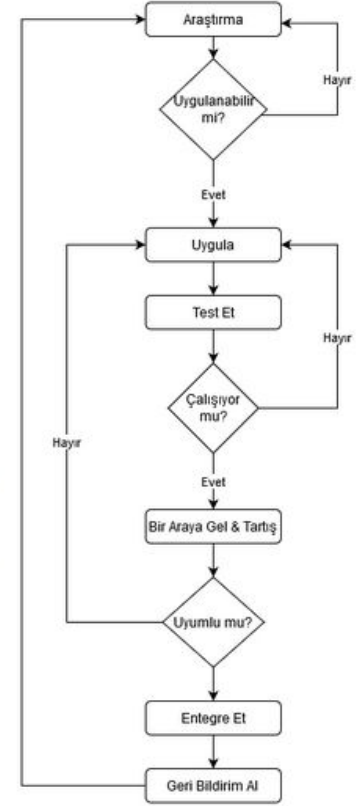
İstemci

İstemci, "Client", gideceği istikametteki boş park yeri bilgisi için API ile haberleşir. Bizim prototipimizde bu bir Android mobil uygulama ile temsil edilmiştir. Mobil uygulamasının kullanıcı ara yüzüne "[buradan](#)" ulaşabilirsiniz.

4. Yöntem

Projemizin gerek prototip geliştirme gerek fikir olarak olgunlaşma aşamasında aşağıda gösterilen diyagramdaki adımlar izlenmiştir;

- İlerlemek için gerekli bilgiler araştırma yaparak öğrenilir
- Edinilen bilgi uygulamaya geçmek için yeterli mi?
- Hayır ise araştırmaya geri dönülür
- Evet ise uygulamaya başlanır
- Uygulama test edilir
- Beklenen sonuç elde edildi mi?
- Hayır ise uygulamaya geri dönülür
- Evet ise grup üyeleri bir araya gelir
- Grup üyeleri bir araya gelerek birbirlerine kaydettikleri ilerleme hakkında aktarım yapar
- Aktarımlar sonucu kaydedilen ilerlemelerin birbirleri üzerinde bozucu bir etkisi olup olmadığına karar verilir
- Bozucu bir etki varsa uygulama aşamasına geri dönülür
- Yok ise entegrasyon aşamasına geçilir
- Entegrasyon aşamasında proje üzerinde yapılan geliştirmelerin birbirleriyle uyumlu bir biçimde çalışması sağlanır
- Danışman ile bir araya gelinir son toplanmadan bu yana yapılan geliştirmeler gösterilir. Bir sonraki adım hakkında fikir alınır ve araştırma aşamasına geri dönülür



Projemizin en hayati bileşenlerinden biri olan “Kamera görüntüsünden araba saptama”

işlemi için iki farklı bilgisayarlı görü teknolojisi üzerinde deneyler gerçekleştirdik. Bunlar;

- “[YoloV3](#)” (You Only Look Once)

Yolo konvansiyonel yapay sinir ağları kullanılarak geliştirilmiş bir obje saptama algoritmasıdır. En önemli özelliği bütün kareye sadece bir yapay sinir ağlarından geçirmesi ve dolayısıyla son derece hızlı olmasıdır

- “[Mask R-CNN](#)”

Faster R-CNN algoritmasının devamıdır. Konvansiyonel yapay sinir ağları kullanılmıştır. Bir kareyi bir çok kez yapay sinir ağlarına sokarak YoloV3’e göre daha doğru sonuçlar üretir.

Projemizin performansına direkt bir etkisi olduğu için bu iki algoritma arasında bazı testler yaptık. Sonuçlar aşağıdaki gibidir;

Bu iki algoritmayı önce GPU ve CPU da çalıştırarak kıyasladık. Sonuçlarımızın görsel sonuçlarına aşağıda bağlantılar ile ulaşabilirsiniz;

- “[GPU vs CPU Algoritma Karşılaştırması - Grafik](#)“
- “[GPU vs CPU Algoritma Karşılaştırması - Tablo](#)”

YoloV3 sıradan bir GPU’da bile neredeyse gerçek zamanlı işlem yapabiliyor ve Mask R-CNN’den iki donanımda da çok daha hızlı. Peki sonuçları doğru gösteriyor mu? İlgili test sonuçlara bu “[görselden ulaşabilirsiniz](#)”.

YoloV3 genelde daha kötü sonuçlar üreten bir algoritma olarak bilinmesine rağmen bizim oluşturduğumuz üç senaryonun ikisinde görüldüğü gibi daha iyi sonuçlar vermiştir. Özellikle en hayati olan üst üste gelen arabalar senaryosunda aradaki fark görmezden gelinemez. Ayrıca hız testlerindeki CPU’da 1 fps(saniyede 1 kare)’lik sonucu da göz önüne aldığımızda hem donanım maliyetini azaltmak hem de doğruluk oranını arttırmak için yolumuza YoloV3 ile devam etmeye karar verdik.

5. Yenilikçi (İnovatif) Yönü

Projemizin tamamen yenilik getiren iki yönü var. Bunlar;

- Cadde/sokak üzerinde kamera tabanlı park yeri bulma
- Kamera tabanlı park yeri bulma işleminin el ile park yeri girilmeden, park yerlerini kendi başına öğrenerek yapılması

Yaptığımız araştırmalarda Türkiye’de kamera tabanlı park yeri hizmeti veren bir şirkete rastlamadık ancak dünya pazarında bir örnek bulduk. “[Parkam](#)“ Tel Aviv tabanlı bir İsrail şirketi ve seçilen otoparklarda park yerlerinin hem kare içindeki koordinatlarını hem de dünya üzerindeki koordinatlarını girerek kullanıcıya seçtiği park yerine kadar navigasyon hizmeti sağlıyor. Ayrıca yasaklı yerlere yapılan parkları tespit etmek için emniyet güçlerine destek sağlıyor.

Diğer kamera tabanlı trafik çözümlerinin aksine AIPark’ta her bir park yerinin elle tanımlanması gerekmez. Tak-çalıştır özelliğe sahip olduğu için herhangi bir kamerayı izlemeye başladığı andan itibaren park noktalarının yerlerini sokaktaki araçlardan öğrenir.

AIPark park yeri/sokak durumunu devamlı olarak takip eder. Bu devamlı verilerden istenen bir zamanda boş park yeri bulunabilme olasılığı hesaplanabilir. Bu veriler ayrıca trafik yoğunluğu izlenmesi için de kullanılabilir.

6. Uygulanabilirlik

Projemiz evrensel değerlerin üzerine kendi katma değerimizi ekleyerek oluşturduğumuz yerli ve milli bir yazılım projesidir. Yüksek Çözünürlük (HD) kalitesinde çekim yapabilen herhangi bir kamera ve trafiği kaldırabilecek herhangi bir sunucu ile hızlıca hayata geçebilir. Sahaya inme aşamasında karşılaşma ihtimalimiz olan bazı engeller aşağıdaki gibidir. Bu engeller hakkında daha detaylı bilgi için “[Riskler](#)” bölümünde bulunan tabloyu inceleyebilirsiniz

- *Bürokratik Engeller*
Sokaklara kamera konulması sırasında çıkabilecek bir takım hukuki süreçler
- *Sahada Karşılaşılabilecek Fiziksel Engeller*
Kameralarımızın fiziki durumuna gelebilecek zararlar
- *Yazılımsal Engeller*
Geliştirme sırasında öngörülmeleyen zorluklarla karşılaşılması

Yukarıda belirtilen muhtemel engellerle karşılaşılmaması durumunda uygulayacağımız B planları ve projeye etkileri “[Riskler](#)” bölümünde mevcuttur.

7. Tahmini Maliyet ve Proje Zaman Planlaması



Benzer bir proje olan Parkam için ücretlendirme veya şirket değeri bilgisi bulamadığımız için karşılaştırma yapamıyoruz.

Prototip geliştirmesi sırasında telefon kameralarımızı ve kişisel bilgisayarlarımızı kullandığımız için prototip maliyeti göz ardı edilebilecek seviyededir. Üretim aşaması ile ilgili maliyetler aşağıdadır;

Dizilim Adı	Kamera Ücreti	Sunucu	Sunucu Maliyeti	Market Ücretleri	Brüt Mühendis Maaşı(Toplam)
En Düşük Maliyet	92 TL	Yerel Rasp. pi	243 TL	25\$+99\$/yıl	20.000 TL
Orta Doğruluk	153 TL	Velocihost CPU	\$65/ay	25\$+99\$/yıl	20.000 TL
Yüksek Doğruluk	825 TL	Velocihost GPU	\$149/ay	25\$+99\$/yıl	20.000 TL

8. Proje Fikrinin Hedef Kitle (Kullanıcılar):

Projemizin hedef kitesini temel olarak 2 kategoriye ayırabiliriz. Bunlar;

1. Son Kullanıcılar

Bu kategorideki kullanıcılar proje diyagramındaki “Client” paketi üzerinden sistemle etkileşime geçeceklerdir. Araç kullanan herhangi bir insan için Client(İstemci) paketi aracılığı ile sistemden edinilen park yeri bilgisi son derece faydalı olacaktır.

2. Ürünü mü kendi ürünlerini geliştirmekte kullanacak olan üreticiler

Daha önce de bahsettiğimiz gibi projemiz son kullanıcılara park yeri bilgisi sunarken aynı zamanda elde ettiği park yeri verilerini yapısal bir şekilde tutar ve gelecekte trafik yoğunluğu, araç sahiplerinin davranışları gibi alanlarda üretilebilecek sayısız uygulamalara da altyapı sağlayacaktır.

Sokaktaki park yerlerinin dinamik olarak fiyatlandırılması, hatalı parkların cezalandırılması sayısız uygulama fikirlerinden sadece bir kaç tanesi. Başta Devlet, emniyet güçleri, belediyeler olmak üzere bu alanlarda çalışmak isteyen herkes bu verilerden bizim aracılığımızla faydalanabilir

9. Riskler

Risk Tanımı	Olasılık	Etki	B Planı
Devletten/Belediyeden izin alınamaması	Düşük	Yüksek	Hukuki destek ile anlaşmaya varılması
Yetersiz donanım kalitesi	Düşük	Orta	Daha az güç gerektiren teknolojilere geçilmesi
Kullandığımız bileşenlerin lisans sorunu yaratması	Düşük	Yüksek	Hukuki destek ile anlaşmaya varılması
Kameraların ortam şartlarından kötü etkilenmesi	Düşük	Yüksek	Kameraların yeniden konumlandırılması
Sahada geliştirme sürecinde öngörülmeleyen zorluklar çıkması	Orta	Düşük	Yeni sürümlerle sorunun çözülmesi

10. Proje Ekibi

Takım Lideri: Anıl Helvacı

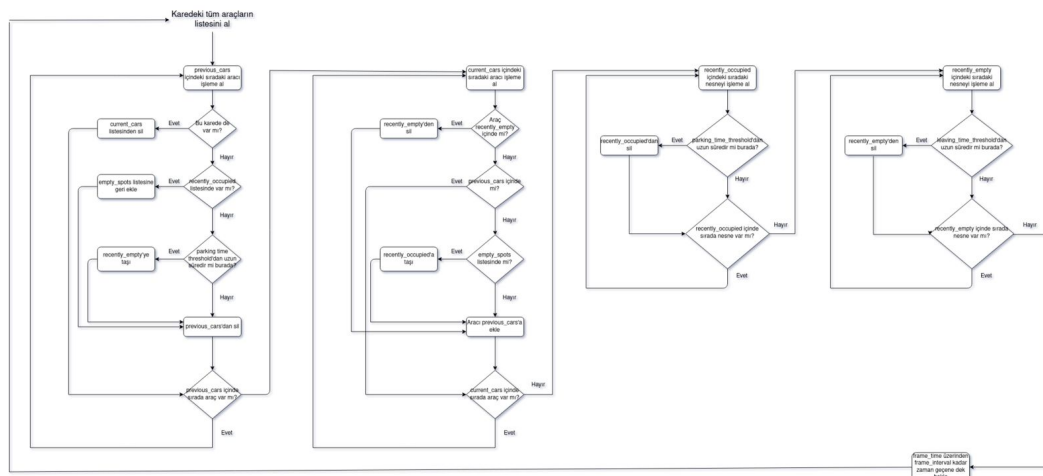
Adı Soyadı	Projedeki Görevi	Okul	Tecrübesi
Anıl Helvacı	B. Görü, Mobil	Hacettepe Ü.	Makina Öğrenmesi, Android SDK
Cihat Duman	İş Mantığı, API	Hacettepe Ü.	Rest API, Algoritma
Dr. Öğ. Üyesi Adnan Özsoy	Danışman	Hacettepe Ü.	https://web.cs.hacettepe.edu.tr/~aozsoy/

11. Kaynaklar

- www.velocihost.net/
- <https://www.parkam.com/>
- <https://pjreddie.com/darknet/yolo/>
- <https://arxiv.org/abs/1703.06870>

Ekler

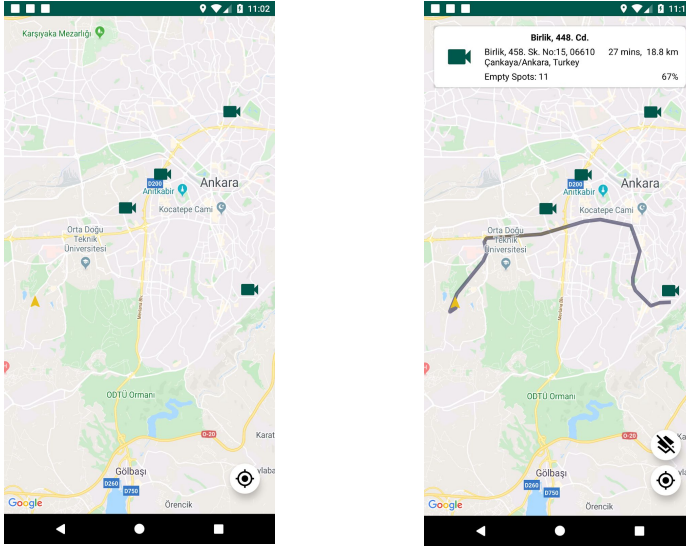
1. Akış Diyagramı



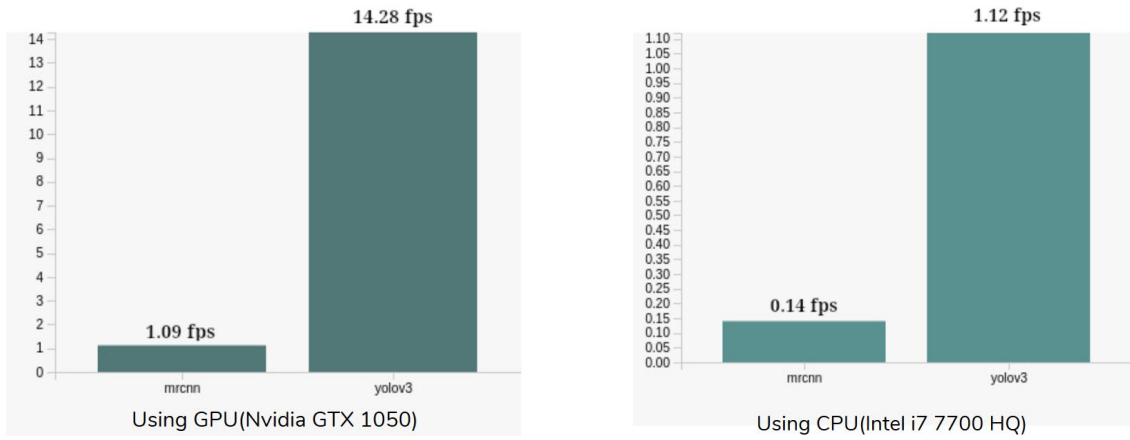
Diyagramın yüksek çözünürlüklü özgün hali verilen adreste bulunabilir:

<https://drive.google.com/file/d/1T5pMK0uCkz0D7AiXIHPsLYh1RNss61jb/view?usp=sharing>

2. Mobil Uygulama GUI



3. GPU vs CPU Algoritma Karşılaştırması - Grafik



4. GPU vs CPU Algoritma Karşılaştırması - Tablo

Algoritma	GPU	CPU
YoloV3	14.28 fps	1.12 fps
Mask R-CNN	1.09 fps	0.14 fps

5. YoloV3 vs Mask R-CNN Doğruluk Karşılaştırması

