

# **INTERNET OF THINGS**

## **PROJE RAPORU**

### **Akıllı Ulaşım Takip Sistemi - OTObus**

**Hazırlayanlar**

**Sude Andurman**

**Zehra Burcu**

**Elif Sena Soysal**

# BSM 313 – Nesnelerin İnterneti ve Uygulamaları

## 1. Proje Özeti

**Amaç:** Bu projenin amacı; toplu taşıma araçlarında anlık yolcu sayısını görüntü işleme ile tespit etmek, araç içi hava kalitesini sensör ağlarıyla analiz etmek ve elde edilen verileri yapay zeka algoritmalarıyla işleyerek yoğunluk sebeplerini tahminleyen bütünlük bir izleme sistemi geliştirmektir.

**Özet Metni:** Akıllı şehir konseptinde ulaşım konforu ve verimliliği en önemli parametrelerdir. Bu çalışma kapsamında, Raspberry Pi 3 ve ESP8266 mikrodenetleyicileri kullanılarak hibrit bir gömülü sistem mimarisi tasarlanmıştır. Sistem üç ana modülden oluşmaktadır [1]:

1. Görüntü İşleme Modülü: Raspberry Pi 3 ve Kızılıötesi (NoIR) Kamera kullanılarak, MobileNetSSD derin öğrenme modeli ile yolcu giriş-çıkışları sayılmakta ve anlık doluluk oranı hesaplanmaktadır.[4][5]
2. Ortam Analiz Modülü: ESP8266, MQ-135 (Hava Kalitesi)[3] ve MH-Z14 (CO<sub>2</sub>) sensörleri[2] ile entegre edilerek araç içi hava kalitesi sürekli izlenmektedir. CO<sub>2</sub> seviyesindeki artış ile yolcu yoğunluğu arasındaki korelasyon analiz edilerek, konfor seviyesi grafiksel olarak mobil uygulamaya yansıtılmaktadır.
3. Tahminleyici Yapay Zeka (Predictive AI): Toplanan veriler (yolcu sayısı, saat, hava durumu vb.) ve dış kaynaklı veriler (maç günleri, konserler, tatiller) işlenerek, "Yolcu yoğunluğunun neden arttığını" (Örn: "Bugün derbi maçı var, yoğunluk %40 artabilir") açıklayan bir doğal dil işleme (NLP) modülü geliştirilmiştir.

Tüm veriler Google Firebase üzerinde toplanmakta ve Flutter tabanlı mobil uygulama üzerinden yöneticilere anlık grafikler, canlı kamera görüntüsü ve yapay zeka analiz raporları sunulmaktadır.

## 2. Problemin Tanımı ve Motivasyon

**Problemin Tanımı:** Mevcut toplu taşıma sistemlerinde sadece "kaç kişi var" sorusuna odaklanılmakta, ancak "yolculuk ne kadar sağlıklı?" ve "neden bu kadar kalabalık?" soruları yanıtız kalmaktadır.

1. Havasızlık ve Sağlık Riski: Kalabalık araçlarda CO<sub>2</sub> seviyesinin yükselmesi, yolcularda yorgunluk ve baygınlık hissine yol açmaktadır. Mevcut sistemler hava kalitesini ölçmemektedir.[6]
2. Veri Anlamlandırma Eksikliği: Bir otobüsün dolu olduğunu bilmek tek başına yeterli değildir. Operatörlerin, yoğunluğun anlık bir etkinlikten mi yoksa sistematik bir sorundan mı kaynaklandığını bilmesi (Root Cause Analysis) gerekmektedir.
3. Gece Görüş Eksikliği: Standart kameralar akşam seferlerinde yetersiz kalmakta, bu da sayım doğruluğunu düşürmektedir.

**Motivasyon:** Projemiz, yolcu sayımını ortam sensörleri ile birleştirerek bu problemlere çok boyutlu bir çözüm sunmaktadır.

1. Sağlık Odaklı Yaklaşım: MQ-135 ve MH-Z14 sensörleri ile araç içi hava kalitesi mobil uygulamada grafikleştirilerek, riskli durumlarda havalandırma uyarısı yapılmasına olanak tanır.
2. Akıllı Analiz: Geliştirilen metin tabanlı yapay zeka modülü, yoğunluk verisini dış dünyadaki olaylarla eşleştirerek operatöre "Bugün X etkinliği sebebiyle yoğunluk bekleniyor" şeklinde stratejik bilgi sağlar.
3. Donanım Uyumu: Kızılıtesi (Infrared) kamera kullanımı ile günün her saatinde yüksek doğrulukla sayımları yapılmaktadır.

### 3. Kullanılan IoT Teknolojileri

#### 3.1. Donanım Bileşenleri

- Raspberry Pi 3
- Raspberry Pi Infrared Camera Modul V2
- NodeMCU ESP8266
- Air Quality Sensor Board - MQ-135
- MH-Z14 CO2 Karbondioksit Sensörü
- DHT-22 Sıcaklık ve Nem sensörü
- 8 Character 7 Segment SPI Display
- 4 Pin Push Button
- microSDHC UHS-I Card
- Breadboard
- Direnç

#### 3.2. Yazılım ve Platformlar

- Google Firebase Realtime Database (Bulut Veritabanı)
- Flutter (Mobil Uygulama Geliştirme Kiti)
- Dart (Programlama Dili)
- Python (Programlama Dili)
- OpenCV (Görüntü İşleme Kütüphanesi)
- MobileNetSSD (Yapay Zeka Modeli)
- Flask (Web Sunucusu Framework'ü)
- FastAPI (Api Framework'ü)

## 4. Sistem Mimarisi

Bu proje, nesnelerin interneti (IoT), bulut tabanlı veri yönetimi, yapay zeka destekli analiz ve mobil uygulama teknolojilerini bir araya getiren **uçtan uca bir izleme ve kontrol sistemi** olarak tasarlanmıştır. Sistem; fiziksel ortamdan veri toplayan donanım birimleri ile bu verileri işleyen, analiz eden ve kullanıcıya sunan yazılım bileşenlerinin entegre çalışmasına dayanmaktadır.

### Donanım Katmanı (IoT Layer)

Sistemin donanım katmanında **ESP8266 (NodeMCU)** mikrodenetleyicisi yer almaktadır. Bu mikrodenetleyici, otobüs içi ortam koşullarını izlemek amacıyla sıcaklık, nem, karbondioksit ve hava kalitesi sensörlerinden periyodik olarak veri toplamaktadır. Toplanan veriler mikrodenetleyici üzerinde ön işleme tabi tutulmakta ve kablosuz ağ üzerinden bulut altyapısına iletilmektedir. Ayrıca sistem, fiziksel buton ve kesme (interrupt) mekanizmaları sayesinde internet bağlantısından bağımsız temel kontrol işlemlerini gerçekleştirebilmektedir.

### Bulut Katmanı (Cloud Layer)

Toplanan tüm veriler, Firebase Realtime Database üzerinde merkezi olarak depolanmaktadır. Sensörlerden gelen anlık veriler gerçek zamanlı olarak güncellenirken, geçmişe dönük veriler zaman damgalı şekilde kayıt altına alınmaktadır. Firebase altyapısı, donanım ile mobil uygulama arasında çift yönlü bir iletişim köprüsü görevi görerek, kullanıcı tarafından verilen kontrol komutlarının anlık olarak donanım birimlerine iletilmesini sağlamaktadır.

### Yapay Zeka ve Analiz Katmanı

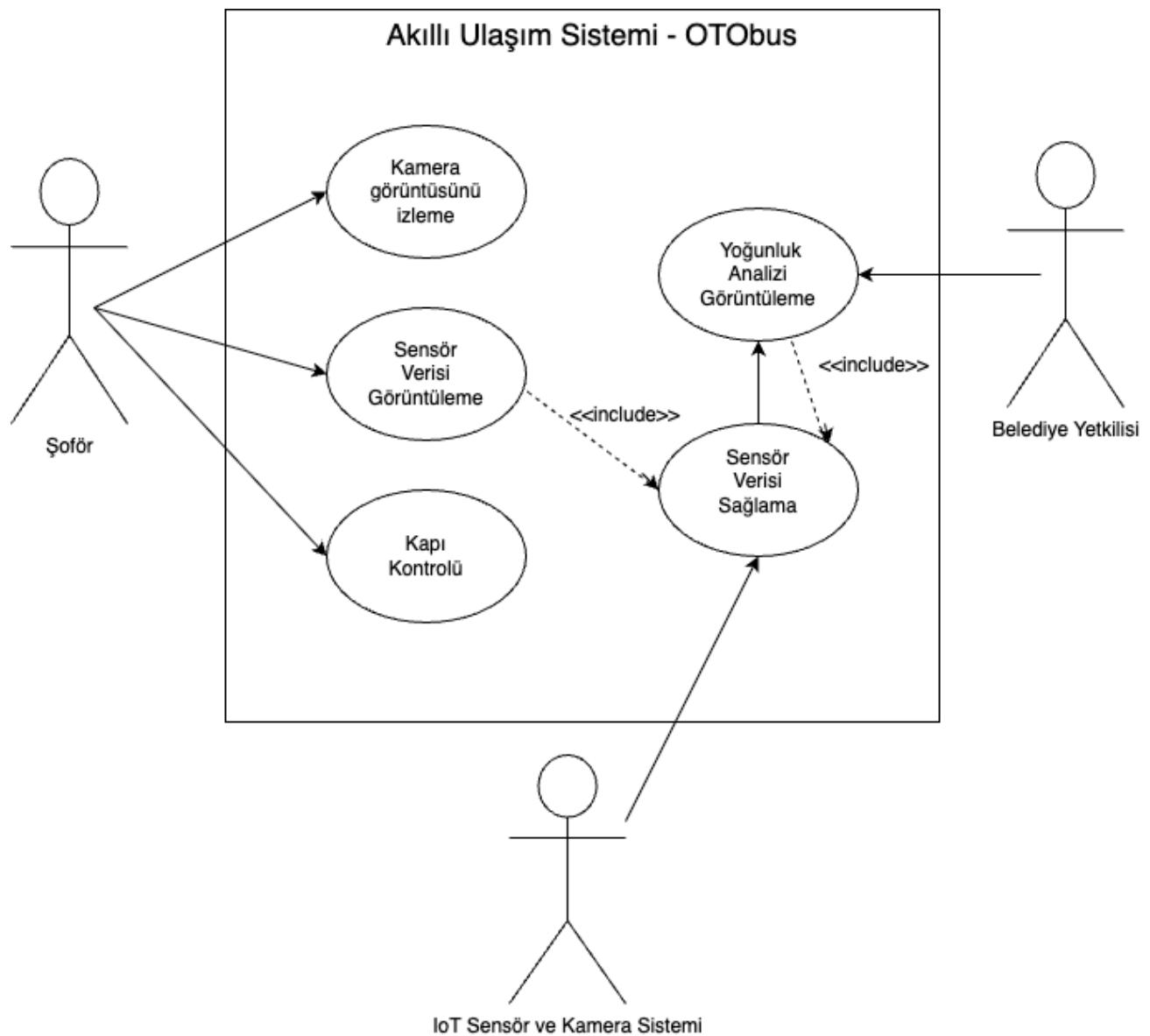
Sistem mimarisinde yer alan Python tabanlı yapay zeka sunucusu, Firebase üzerinde biriken geçmiş verileri kullanarak yoğunluk ve ortam durumu analizleri gerçekleştirmektedir. Bu katman, belirli zaman aralıklarında oluşan yoğunluk durumlarını tespit ederek karar destek süreçlerine katkı sağlamaayı amaçlamaktadır. Mobil uygulama, bu analiz sonuçlarını API servisleri aracılığıyla alarak kullanıcıya raporlar halinde sunmaktadır.

### Uygulama Katmanı (Application Layer)

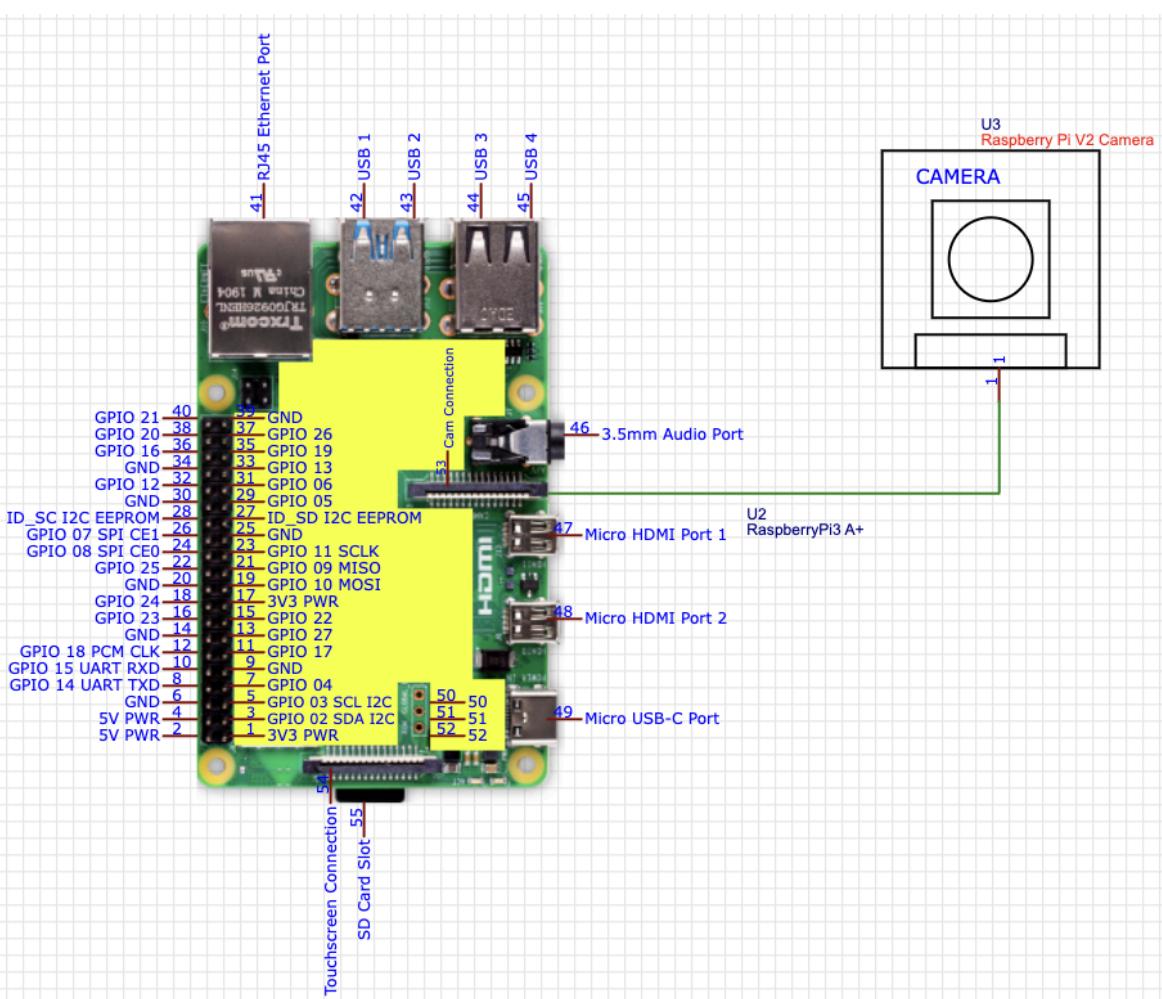
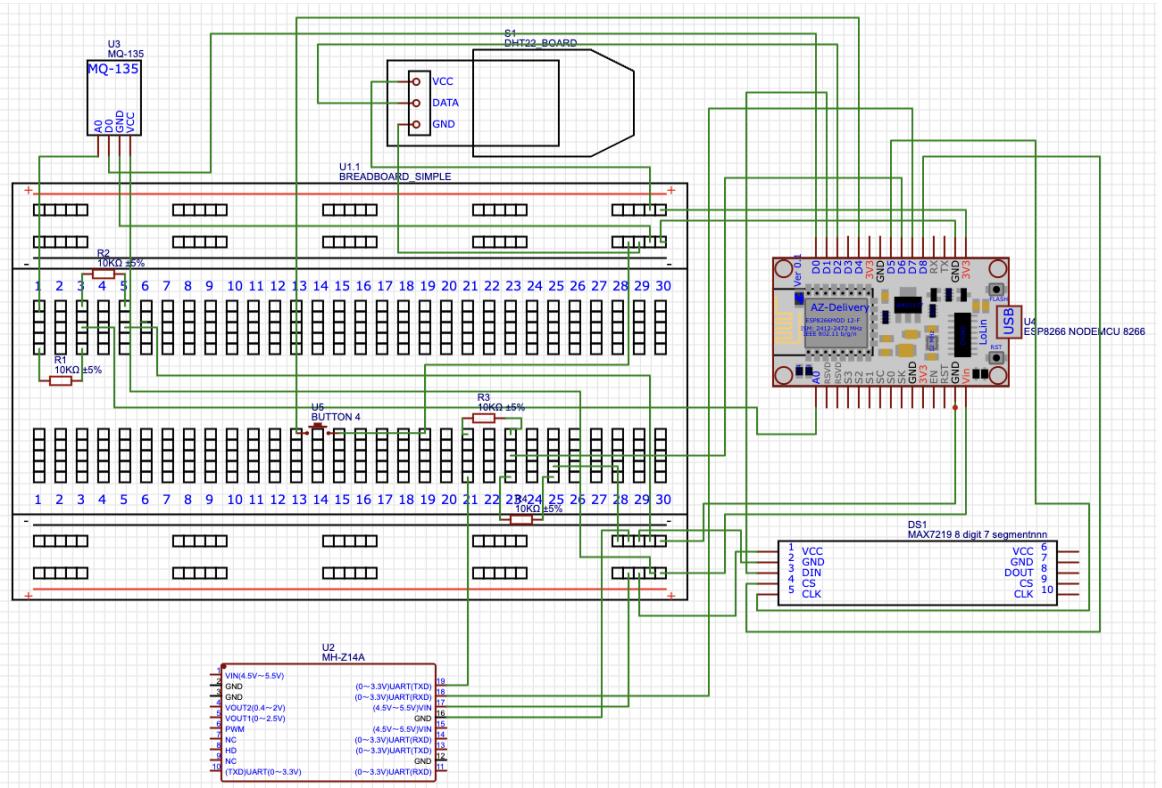
Kullanıcı etkileşiminin sağlandığı uygulama katmanı, Flutter framework'ü kullanılarak geliştirilmiştir. Mobil uygulama; Android ve iOS platformlarında çalışacak şekilde tasarlanmıştır. Kullanıcılar uygulama üzerinden anlık sensör verilerini görüntüleyebilmekte, geçmiş verileri grafiksel olarak inceleyebilmekte ve sistemin sunduğu kontrol işlevlerini uzaktan yönetebilmektedir. Ayrıca yapay zeka katmanından elde edilen analiz sonuçları bu arayüz üzerinden erişilebilir durumdadır.

Yukarıda açıklanan mimari yapı doğrultusunda, sistemin kullanıcı ve sistem bileşenleri ile olan etkileşimlerini daha net bir şekilde gösterebilmek amacıyla **Use Case diyagramı** oluşturulmuştur. Bu

diyagramda sistemin sunduğu temel işlevler ve bu işlevlerin ilgili aktörler tarafından nasıl kullanıldığı görselleştirilmektedir.



## 5. Elektronik Devre Tasarımı



## 6. Gerçekleme (Uygulama) Süreci

### 6.1. Yazılım Geliştirme Adımları

#### A. Gömülü Sistem ve Görüntü İşleme (Backend) Geliştirme:

- Ortam Kurulumu: Raspberry Pi üzerine "Raspberry Pi OS (Bookworm)" kurulması, OpenCV, Flask, Firebase-Admin kütüphanelerinin entegrasyonu.
- Algoritma Tasarımı:
  - Görüntü Alma: Kameradan görüntü akışının başlatılması.
  - Nesne Tespiti (Detection): MobileNetSSD modelinin yüklenmesi ve her karede (frame) insan sınıfınınfiltrelenmesi.
  - Sayımlığı (Counting): Sanal bir referans çizgisi belirlenmesi ve vektörel analiz ile (önceki konum vs şimdiki konum) geçiş yönünün (Giriş/Cıkış) sanal bir çizgi ile hesaplanması.
- Optimizasyon (Session Based Logic): İşlemciyi yormamak ve veriyi tutarlı tutmak için "Kapı Durumu"na bağlı bir algoritma geliştirildi. Kapı kapandığında sistemin uyku moduna geçmesi ve toplam veriyi veritabanına tek seferde yazması sağlandı.

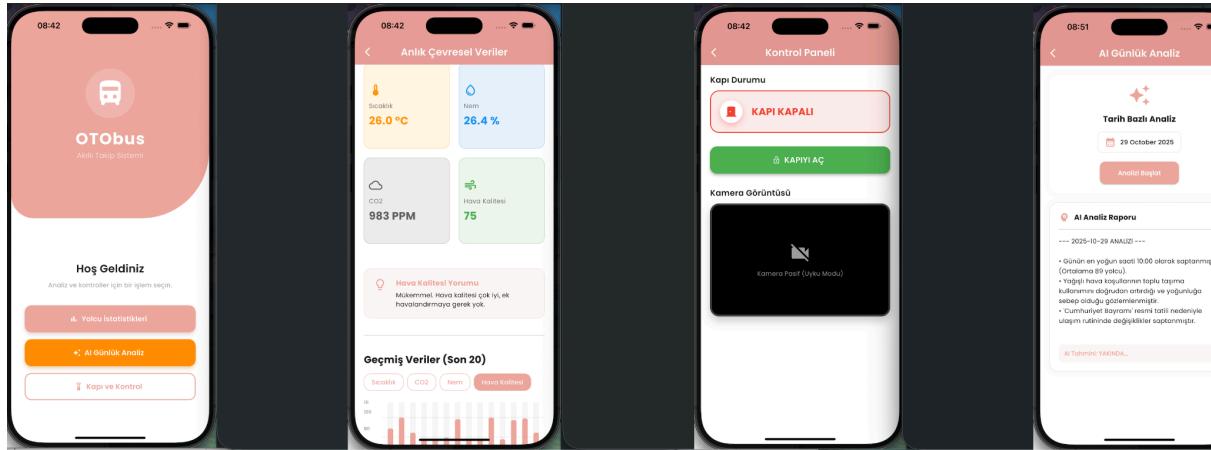
```
[pi@raspberrypi: ~ $ cd insan_sayma/
[pi@raspberrypi:~/insan_sayma $ python3 main.py
[INFO] Model yükleniyor...
 * Serving Flask app 'main'
 * Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on all addresses (0.0.0.0)
 * Running on http://127.0.0.1:5000
 * Running on http://172.20.10.2:5000
Press CTRL+C to quit
[INFO] CANLI YAYINLI SİSTEM BASLATILDI.
[BİLGİ] Kapı Kapandı! Hesaplamlalar yapılıyor...
Hesap: 4 + (0 - 0) = 4
[LOG] Durak verisi kaydedildi. Yeni Toplam: 4
172.20.10.3 - - [18/Dec/2025 12:47:57] "GET /video_feed HTTP/1.1" 200 -
[BİLGİ] Kapı Kapandı! Hesaplamlalar yapılıyor...
Hesap: 4 + (0 - 0) = 4
[LOG] Durak verisi kaydedildi. Yeni Toplam: 4
172.20.10.3 - - [18/Dec/2025 12:49:06] "GET /video_feed HTTP/1.1" 200 -
GİRİŞ (IN)! Anlık: 1
ÇIKIS (OUT)! Anlık: 1
GİRİŞ (IN)! Anlık: 2
[BİLGİ] Kapı Kapandı! Hesaplamlalar yapılıyor...
Hesap: 4 + (2 - 1) = 5
[LOG] Durak verisi kaydedildi. Yeni Toplam: 5
172.20.10.3 - - [18/Dec/2025 12:53:10] "GET /video_feed HTTP/1.1" 200 -
[BİLGİ] Kapı Kapandı! Hesaplamlalar yapılıyor...
Hesap: 5 + (0 - 0) = 5
[LOG] Durak verisi kaydedildi. Yeni Toplam: 5
172.20.10.3 - - [18/Dec/2025 12:56:46] "GET /video_feed HTTP/1.1" 200 -
[BİLGİ] Kapı Kapandı! Hesaplamlalar yapılıyor...
Hesap: 5 + (0 - 0) = 5
[LOG] Durak verisi kaydedildi. Yeni Toplam: 5
172.20.10.3 - - [18/Dec/2025 12:56:56] "GET /video_feed HTTP/1.1" 200 -
GİRİŞ (IN)! Anlık: 1
GİRİŞ (IN)! Anlık: 2
[BİLGİ] Kapı Kapandı! Hesaplamlalar yapılıyor...
Hesap: 5 + (2 - 0) = 7
[LOG] Durak verisi kaydedildi. Yeni Toplam: 7
172.20.10.3 - - [18/Dec/2025 12:59:56] "GET /video_feed HTTP/1.1" 200 -
ÇIKIS (OUT)! Anlık: 1
[BİLGİ] Kapı Kapandı! Hesaplamlalar yapılıyor...
Hesap: 7 + (0 - 1) = 6
[LOG] Durak verisi kaydedildi. Yeni Toplam: 6
172.20.10.3 - - [18/Dec/2025 13:01:34] "GET /video_feed HTTP/1.1" 200 -
GİRİŞ (IN)! Anlık: 1
ÇIKIS (OUT)! Anlık: 1
ÇIKIS (OUT)! Anlık: 2
[BİLGİ] Kapı Kapandı! Hesaplamlalar yapılıyor...
Hesap: 6 + (1 - 2) = 5
[LOG] Durak verisi kaydedildi. Yeni Toplam: 5
```

#### B. Mobil Uygulama ve Kullanıcı Arayüzü (Frontend):

Bu proje kapsamında geliştirilen mobil uygulama, kullanıcıların toplu taşıma araçlarına ait yolcu yoğunluğu verilerini kolay ve anlaşılır bir şekilde görüntüleyebilmesini amaçlamaktadır. Flutter framework'ü kullanılarak geliştirilen mobil uygulama, platform bağımsız bir yapı sunmakta ve modern kullanıcı arayüzü bileşenleri ile desteklenmektedir. Uygulama içerisinde kullanıcının tarih

seçimi yapabilmesi için takvim tabanlı bir arayüz tasarlanmıştır, seçilen tarihe ait yolcu yoğunluğu verileri ve günlük analiz özetleri kullanıcıya sunulmuştur. Mobil uygulama, FastAPI ile geliştirilen backend servisine HTTP istekleri göndererek gerekli verileri almakta ve bu verileri kullanıcı dostu bir formatta görselleştirmektedir. Böylece kullanıcılar, belirli bir güne ait yoğunluk durumunu ve buna ilişkin açıklayıcı yorumları mobil cihazları üzerinden hızlı bir şekilde görüntüleyebilmektedir.

Geliştirilen arayüz, sade ve anlaşılır bir tasarım anlayışıyla oluşturulmuş olup, uygulamanın ileride farklı toplu taşıma araçları ve ek özelliklerle genişletilmesine uygun bir yapı sunmaktadır.



### C.Yapay Zeka Entegrasyonu ve Akıllı Analiz:

Proje kapsamında geliştirilen sistemde, yolcu yoğunluğu verilerinin anlamlandırılması ve yorumlanması amacıyla yapay zeka destekli bir analiz yaklaşımı benimsenmiştir. Firebase üzerinde tutulan zaman damgali yolcu verileri, backend tarafında Python tabanlı bir servis aracılığıyla işlenmekte ve analiz edilmektedir. Yapay zeka entegrasyonu kapsamında, gün içerisindeki saatlik yoğunluk dağılımları, hafta içi-hafta sonu farkları, resmi tatil bilgileri ve temel istatistiksel ölçütler (ortalama, tepe saatler, dalgalanma düzeyi vb.) dikkate alınarak günlük özet analizler oluşturulmaktadır. Sistem, yalnızca sayısal değerler sunmak yerine, bu verileri yorumlayarak kullanıcıya açıklayıcı ve anlaşılır metinler halinde geri bildirim sağlamaktadır. Böylece ham veriler, kullanıcılar ve karar vericiler için anlamlı bilgilere dönüştürülmektedir. Geliştirilen bu yapay zeka destekli analiz yapısı, mevcut haliyle geçmiş verilerin yorumlanması odaklanmakta olup, ilerleyen süreçte daha uzun süreli veri setlerinin elde edilmesiyle birlikte ileriye yönelik yolcu yoğunluğu tahminleri üretebilecek şekilde genişletilmeye uygundur. Bu yönyle sistem, akıllı ulaşım uygulamaları için temel oluşturabilecek esnek ve ölçeklenebilir bir yapay zeka entegrasyonu sunmaktadır.

## 6.2.Karşılaşılan Problemler ve Çözümler

**SSH Erişim ve Imager Kurulum Sorunu:** Projenin başlangıcında Raspberry Pi'ye monitörsüz erişim sağlanamadı ve sorunun işletim sistemi kaynaklı olduğu düşünülerek SD karta defalarca işletim sistemi yeniden kuruldu. Hatta yeni SD kart almamıza sebep oldu.

**Düşük FPS (Kasma) Sorunu:** Çözemedik. Elimizdeki donanımın yetersiz olmasından dolayı bu sorun hala devam etmekte.

**Veri Tutarlılığı:** Giriş yaptığımız zaman çıkış verisini arttırmıyordu. Modelimizin yolcunun ekrandaki konumuna göre değilde vektörel bir sayım gerçekleştirmesini sağladık.

**Dependency Hell (Sürüm Çakışması):** Flutter'da kullanılan hazır MJPEG paketi, mobil applicationda kullandığımız güncel Google Fonts kütüphanesi ile uyumsuzluk yarattı. Çözüm olarak hazır paket kullanmak yerine, HTTP stream akışını byte-byte okuyan ve bunu ekrana resim olarak basan özel bir görüntüleme Widget'i (Custom MjpegView) manuel olarak kodlandı.

**Geçmiş Günler ve Hava Durumu Verisi Problemi:** Geçmiş günlerin özetini oluştururken o güne ait hava durumu verileri API ile alınmak istendi. Ancak OpenWeather gibi API'ler geçmiş verileri ücretsiz sunmamaktaydı. Bu problem Metostat kütüphanesi kullanılarak aşıldı.

**Yapay Zeka için Veri Eksikliği ve Test Zorluğu:** Yapay zeka modelini test etmek için belirli bir tarihe ait yeterli veri yoktu ayrıca Kaggle, UCI Machine Learning Repository gibi sentetik veri sağlayen site/uygulamalarda da modeli eğittiğimiz formatta veri yoktu. Bu yüzden üretken yapay zeka kullanılarak yapay veri elde edildi.

## 7. Büyük Veri ve Analitik Yaklaşım

Projemiz kapsamında geliştirilen IoT sistemi, tek bir araçta prototip olarak çalışsa da, bir şehrin tüm toplu taşıma araçlarına entegre edildiğinde devasa bir veri akışı oluşacaktır (Big Data).

### 7.1. Elde Edilecek Anlamlı Bilgiler ve Hayati Kolaylaştıran Çözümler

Milyonlarca satırlık yolcu ve sensör verisinin işlenmesiyle şu stratejik bilgiler elde edilebilir:

#### 1. Dinamik Sefer Planlaması ve Rota Optimizasyonu:

- **Senaryo:** Hangi durakta, haftanın hangi günü ve saatinde yoğunluk yaşadığı geçmiş verilerle analiz edilir.
- **Fayda:** "Sabit tarife" yerine "Talep Bazlı" sefer sistemine geçilir. Yoğun saatlerde otomatik olarak ek sefer konulur, boş giden otobüsler hattan çekilerek yakıt tasarrufu sağlanır.

#### 2. Halk Sağlığı Haritalaması ve Pandemi Yönetimi:

- **Senaryo:** Araç içi CO<sub>2</sub> ve hava kalitesi verileri, şehir genelindeki lokasyon verisiyle birleştirilir.
- **Fayda:** Şehrin hangi bölgelerindeki toplu taşımada hava kalitesinin düştüğü tespit edilir. Bu hatlara daha sık araç verilerek yolcu yoğunluğu düşürülür ve yolcuların sağlığı korunur.

#### 3. Tahminleyici Bakım (Predictive Maintenance):

- **Senaryo:** Motor sıcaklığı, titreşim veya sensörlerdeki anomalilikler yapay zeka ile izlenir.
- **Fayda:** Araç bozulmadan önce sistem arızayı tahmin eder (Örn: "Sensör X 48 saat içinde bozulabilir"). Böylece yolcular yolda kalmaz, operasyonel aksaklıklar önlenir, önceden arıza tespiti yapılabilir.

#### 4. Akıllı Şehir Planlaması:

- **Senaryo:** Durak bazlı iniş-biniş verileri analiz edilir.
- **Fayda:** İnsanların en çok hareket ettiği akslar belirlenerek yeni metro hatlarının veya durakların nereye yapılması gerekiğine veriye dayalı karar verilir.

#### 5. Yapay Zeka Destekli Hat Verimlilik Analizi ve Stratejik Karar Destek:

- Senaryo: Sistem, sadece anlık veriyi değil, geriye dönük geniş kapsamlı (örneğin 2 yıllık tarihsel verileri; mevsimsellik, hava durumu, okul dönemleri ve tatil günleri gibi parametrelerle birlikte işler.
- Fayda (Prescriptive Analytics): Yapay zeka, sadece "Şu an yoğunluk var" uyarısı vermekle kalmaz, belediye veya işletmeci için geleceğe yönelik "Reçete (Prescriptive)" niteliğinde kararlar üretir.

## 7.2. Önerilen Büyük Veri Altyapısı ve Teknolojiler

Projenin bir şehrde yayılarak binlerce otobüste uygulanması durumunda; saniyede oluşan devasa veri trafigini yönetmek ve anlık karar üretebilmek için ölçeklenebilir bir Büyük Veri Mimarisi gereklidir. Mevcut prototipte kullanılan Firebase yerine, endüstriyel ölçekte şu teknolojiler önerilmektedir:

Binlerce sensörden aynı anda gelen yoğun veri akışını kayıpsız karşılamak için Apache Kafka önerilmektedir [7]. Veriyi diske yazmadan RAM üzerinde işleyerek anormallik tespiti yapmak için Apache Spark Streaming teknolojisi tercih edilmiştir [8]. Zamana bağlı (Time-Series) verilerin yüksek hızda arşivlenmesi için ise Apache Cassandra gibi NoSQL veritabanları kullanılmalıdır [9]. Bu mimari, IoT tabanlı akıllı şehir uygulamalarında standart hale gelmiş ölçeklenebilir bir yaklaşımdır [10].

## 8. İş Fikri ve Business Model Canvas

### OTObus Business Model Canvas

Key Partners	Key Activities	Value Proposition	Customer Relationships	Customer Segments
<ul style="list-style-type: none"> <li>-Donanım ve Bileşen Tedarikçileri</li> <li>-Açık Kaynak Yazılım Toplulukları</li> <li>-Uygulama Mağazaları</li> <li>-Müşteriler</li> <li>-Bulut Servis Sağlayıcıları</li> </ul>	<ul style="list-style-type: none"> <li>-Görüntü İşleme Algoritmalarının Geliştirilmesi</li> <li>-Mobil Uygulama Tasarımı ve Geliştirme</li> <li>-Donanım ve Yazılım Entegrasyonu</li> <li>-Yapay Zeka Destekli Veri Analitiği ve Stratejik Raporlama</li> <li>-Sistem Testleri ve Optimizasyon</li> </ul>	<ul style="list-style-type: none"> <li>-Hibrit Veri Analizi</li> <li>-Veriye Dayalı Doluluk Yönetimi</li> <li>-Yapay Zeka Destekli Stratejik Analiz ve Özeti</li> <li>-Optimize Edilmiş Kaynak Kullanımı ve Verimlilik</li> <li>-Yolcu Sağlığı ve Konforu İyileştirme</li> </ul>	<ul style="list-style-type: none"> <li>-AI Destekli Stratejik Danışmanlık</li> <li>-Veriye Dayalı Güven İlişkisi</li> <li>-Co-Creation (Birlikte Gelişirme)</li> <li>-Self-Servis Yönetim Paneli</li> </ul>	<ul style="list-style-type: none"> <li>-Büyükşehir ve İlçe Belediyeleri (Ulaşım Daire Başkanlıklar)</li> <li>-Özel Halk Otobüsü Kooperatifleri ve İşletmecileri</li> <li>-Akıllı Şehir Proje Geliştiricileri ve Danışmanlık Firmaları</li> </ul>
Key Resources		Channels		
<ul style="list-style-type: none"> <li>-Raspberry Pi 3 ve Çevre Birimleri</li> <li>-Raspberry Pi Camera Module 2 NoIR</li> <li>-Sensörler</li> <li>-ESP8266 Mikrodenetleyici</li> <li>-Görüntü İşleme Kütüphaneleri</li> <li>-Flutter Framework</li> <li>-Özgün Algoritmalar</li> <li>-Ağ Altyapısı</li> <li>-Yazılım Geliştirme Yetkinliği</li> </ul>		<ul style="list-style-type: none"> <li>-Doğrudan Satış ve B2B Görüşmeler</li> <li>-Pilot Uygulamalar (Demo Süreşleri)</li> <li>-Resmi İhaileler</li> <li>-Sektörel Fuarlar</li> <li>-Saha Teknik Ekibi</li> <li>-Uygulama Mağazaları (App Store / Google Play)</li> <li>-Uzaktan Güncellemeye</li> <li>-Teknik Destek Hattı</li> </ul>		
Cost Structure		Revenue Streams		
<ul style="list-style-type: none"> <li>- Donanım Tedarik Maliyetleri (Birim Başı)</li> <li>- Yazılım Geliştirme, Ar-Ge ve AI Eğitim Giderleri</li> <li>-Bulut Sunucusu ve Veri Altyapı Ücretleri</li> <li>-Operasyonel ve Teknik Destek Giderleri</li> <li>- Pazarlama ve Satış Giderleri</li> </ul>		<ul style="list-style-type: none"> <li>-Donanım Satış ve Kurulum Bedeli (One-time Fee)</li> <li>-SaaS - Aylık/Yıllık Abonelik</li> <li>-Bakım ve Teknik Destek Sözleşmeleri</li> <li>- Lisanslama</li> </ul>		

## 9. Maliyet Analizi

<b>HARCAMALAR</b>	
Raspberry Pi 3 Model A+	1.350,00 TL
Raspberry Pi Infrared Camera Modul V2	923,61 TL
Air Quality Sensor Board - MQ-135	85,26 TL
MH-Z14 CO2 Karbondioksit Sensörü	1.629,31 TL
8 Character 7 Segment SPI Display	65,91 TL
4 Pin Push Button	2,04 TL
microSDHC UHS-I Card	200,00 TL
3D Baskı	580,00 TL
<b>TOPLAM</b>	<b>4.836,13 TL</b>

## 10. Sonuç ve Gelecek Çalışmalar

### 10.1 Sonuç

Bu proje çalışması kapsamında, toplu taşıma araçlarında yaşanan yoğunluk ve hava kalitesi problemlerine çözüm getiren, bütünlük ve düşük maliyetli bir Akıllı Ulaşım Yönetim Sistemi prototipi başarıyla geliştirilmiştir. Proje, sadece yolcu sayısını değil MQ-135 ve MH-Z14 sensörleri aracılığıyla ortamın CO2 seviyesini de analiz ederek literatürdeki benzer çalışmalarдан ayrılmıştır. Yolcu yoğunluğu ile hava kalitesi arasındaki ters orantı deneyisel olarak kanıtlanmış ve mobil uygulamada görselleştirilmiştir. Geliştirilen Flutter mobil uygulaması ve Firebase altyapısı sayesinde, veriler milisaniyeler mertebesinde gecikme ile operatöre sunulmuştur. Flask tabanlı MJPEG canlı yayın özelliği, operatörlere veriyi görsel olarak teyit etme imkanı sağlamıştır. Geliştirilen tahminleyici AI modülü, sensör verilerini dış kaynaklı verilerle birleştirerek yoğunluğun sebeplerini açıklayan anlamlı içgörüler üretmeyi başarmıştır.

### 10.2 Gelecek Çalışmalar

Bu proje kapsamında geliştirilen sistem, mevcut haliyle otobüs bazlı günlük yolcu yoğunluğu analizine ve açıklayıcı özetler üretmeye odaklanmaktadır. Ancak sistemin tasarlanan mimarisi, yalnızca otobüslerle sınırlı kalmayacak şekilde genişletilmeye uygunudur. İlerleyen çalışmalarında sistemin metro, tramvay ve benzeri diğer toplu taşıma araçlarına entegre edilmesi mümkün olup, farklı ulaşım türlerinden elde edilen verilerle daha kapsayıcı bir toplu taşıma analizi gerçekleştirilebilecektir. Ayrıca ilerleyen süreçte, en az bir veya iki yıllık gerçek yolcu verisinin toplanmasıyla birlikte kullanılan yapay zeka modeli yalnızca geçmiş verileri yorumlayan bir yapıdan çıkarılarak geleceğe yönelik yolcu yoğunluğu tahminleri üretebilen bir sisteme dönüştürülebilecektir. Bu sayede belirli gün ve saatler için yoğunluk öngörüleri yapılabilecek, beklenen yoğunluk artışlarına

göre ek sefer planlaması ya da düşük talep görülen zaman dilimlerinde sefer sayısının azaltılması gibi operasyonel kararlar desteklenebilecektir. Böylece sistem, yalnızca günlük analiz sunan bir uygulama olmanın ötesine geçerek toplu taşıma planlamasında karar vericilere uzun vadede katkı sağlayabilecek akıllı ve ölçülebilir bir çözüm haline gelecektir.

## 12. Kaynakça

- [1] Raspberry Pi Foundation, "Raspberry Pi 3 Model A+ Product Brief," [Online]. Available: <https://datasheets.raspberrypi.com/rpi3/raspberry-pi-3-a-plus-product-brief.pdf>.
- [2] Winsen Electronics, "Intelligent Infrared CO2 Module (Model: MH-Z14A) User's Manual," Version 1.4, Zhengzhou Winsen Electronics Technology Co., Ltd., 2015.
- [3] Hanwei Electronics Co., Ltd., "MQ-135 Gas Sensor Technical Data," Datasheet.
- [4] A. G. Howard et al., "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," *arXiv preprint arXiv:1704.04861*, 2017.
- [5] W. Liu et al., "SSD: Single Shot MultiBox Detector," in *European Conference on Computer Vision (ECCV)*, Springer, 2016, pp. 21–37.
- [6] S. Batterman, "Review of Air Quality in Public Transportation," in *Current Environmental Health Reports*, vol. 4, no. 2, 2017.
- [7] J. Kreps, N. Narkhede, and J. Rao, "Kafka: A Distributed Messaging System for Log Processing," in *Proceedings of the NetDB*, 2011.
- [8] M. Zaharia et al., "Discretized Streams: Fault-Tolerant Streaming Computation at Scale," in *Proceedings of the 24th ACM Symposium on Operating Systems Principles (SOSP)*, 2013, pp. 423–438.
- [9] A. Lakshman and P. Malik, "Cassandra: a decentralized structured storage system," *SIGOPS Operating Systems Review*, vol. 44, no. 2, pp. 35–40, Apr. 2010.
- [10] M. Marjani et al., "Big data analytics: Architecture, opportunities, and open research challenges," *IEEE Access*, vol. 5, pp. 5247–5261, 2017.