1. Problem Tanımı

Marketlerde çalınan ürünlerin bedeli çoğu zaman çalışanlara ödetilir. Çalışanlar zaten iş saatleri içerisinde yeterince yoğun bir tempoda çalışmalarına rağmen bu tarz olayların yaşanabilmesi sebebiyle kaygılı bir durumda manuel olarak saatlerce güvenlik kamerası kayıtlarını izlemek zorunda kalmaktadırlar. Ayrıca ürün hırsızlığı etik olarak doğru değildir ve toplumda bir adaletsizlik yaratmaktadır. Bu proje, güvenlik kameralarını yapay zekâ destekli bir şekilde anlık olarak takip edip, güvenlik sistemlerinin otonom bir şekilde yönetilmesini sağlamayı amaçlamaktadır.

2. Gereksinim Analizi

Bu sistem, marketlerdeki güvenlik kameralarından gelen verileri yapay zekâ destekli bir şekilde anlık olarak işleyip hırsızlık davranışını tespit etmeyi amaçlamaktadır.
Sistemin temel hedefi; çalışanları sürekli güvenlik kamera kayıtlarını izleme işi ile uğraştırmamak, marketin güvenliğini otonom bir şekilde sağlamak ve müşterilerin etik davranışlara göre davranmasını sağlamaktır.
Sistem; görüntü işleme, olay kaydı ve yönetim paneli bileşenlerinden oluşan modüler bir yapıya sahip olacaktır.

2.1. Fonksiyonel Gereksinimler

- Sistem, güvenlik kameralarından gelen verileri anlık olarak işleyebilmelidir.
- Sistem, kişi ve ürünleri algılayabilmelidir.
- Sistem, kişi ve ürün birbirine çok yakın durumdayken ürünün kaybolması durumunda şüpheli olay oluşturabilmelidir.
- Sistem, oluşturulan olayı Backend API'ına gönderebilmelidir.
- Sistem, şüpheli olay durumunda market çalışanlarını uyarabilmelidir.
- Sistem, olay geçmişini yönetim panelinde kaydeden bir şekilde çalışmalıdır.
- Sistem, tespit edilen olayları zaman damgası ile kaydetmelidir.
- Sistem, şüpheli olay anında ilgili görüntüyü snapshot olarak kaydedebilmelidir.

2.2. Fonksiyonel Olmayan Gereksinimler

- Sistem, 500ms altında görüntü işleme süresine sahip olmalıdır.
- Sistem, en az %75 oranında kişi algılama doğruluğu yetisine sahip olmalıdır.
- Sistem, minimum %90 API Uptime oranı hedeflemektedir.
- Sistem, yüksek çözünürlüklü kamera girişini desteklemelidir.

2.3. Kullanıcılar

- Market çalışanları
- Güvenlik görevlisi
- Sistem yöneticisi

2.4. Sistem Kapsamı

- Kamera
- Yapay Zekâ
- Backend
- Dashboard/Log Kayıtları
- Veri Tabanı/Kayıt Sistemi

3. Sistem Tasarımı
3.1. Sistem Bileşenleri

- Kamera
  Sistemden anlık görüntü verisi almayı sağlayan fiziksel bileşen

- Görüntü İşleme ve Yapay Zekâ Modülü
  Kamera görüntüsüne göre YOLO tabanlı insan ve ürün tespiti yapmayı sağlayan yapı

- Olay Yönetim Modülü (Event Engine)
  Ürünün kaybolma durumunu inceleyen ve kurallara göre şüpheli olay üreten ve yöneten yapı

- Backend API
  Yapay zekâ tarafından işlenen verileri kaydeden ve yönetim paneline aktaran yapı

- Dashboard Arayüzü
  Kayıtların tutulmasını sağlayan ve şüpheli durumların incelenebilmesini sağlayan yapı

- Veri Tabanı / Log Sistemi
  Olay kayıtlarının tutulduğu yapı

- Model Yapılandırma / Ayar Modülü
  Sistem için kritik ayarların arayüz üzerinden kolay ayarlanabilmesini sağlayan yapı

3.2. Sistem Mimarisi

- Kamera
  Görüntü işleme modülüne veri akışı sağlar

- Yapay Zekâ
  Kişiyi ve ürünü algılar

- Olay Yönetim Modülü
  Durumu analiz eder

- Backend API
  Olayı alır, veri tabanına kaydeder ve dashboard'a sunar

- Dashboard
  Olay kayıtlarını ve istatistikleri kullanıcıya gösterir

- Veri Tabanı / Log Sistemi
  Olay ve geçmiş kayıtlarının tutulduğu yapı

3.2.1. Katmanlı Mimari
3.2.1.1. Sunum Katmanı

- Dashboard Arayüzü

  Bu katman, sistemi inceleyen market çalışanları, güvenlik görevlileri ve yöneticiler için şüpheli olay geçmişini gösterir.

3.2.1.2. Uygulama / İş Mantığı Katmanı

- Görüntü İşleme ve Yapay Zekâ Modülü
- Olay Yönetim Modülü (Event Engine)
- Backend API

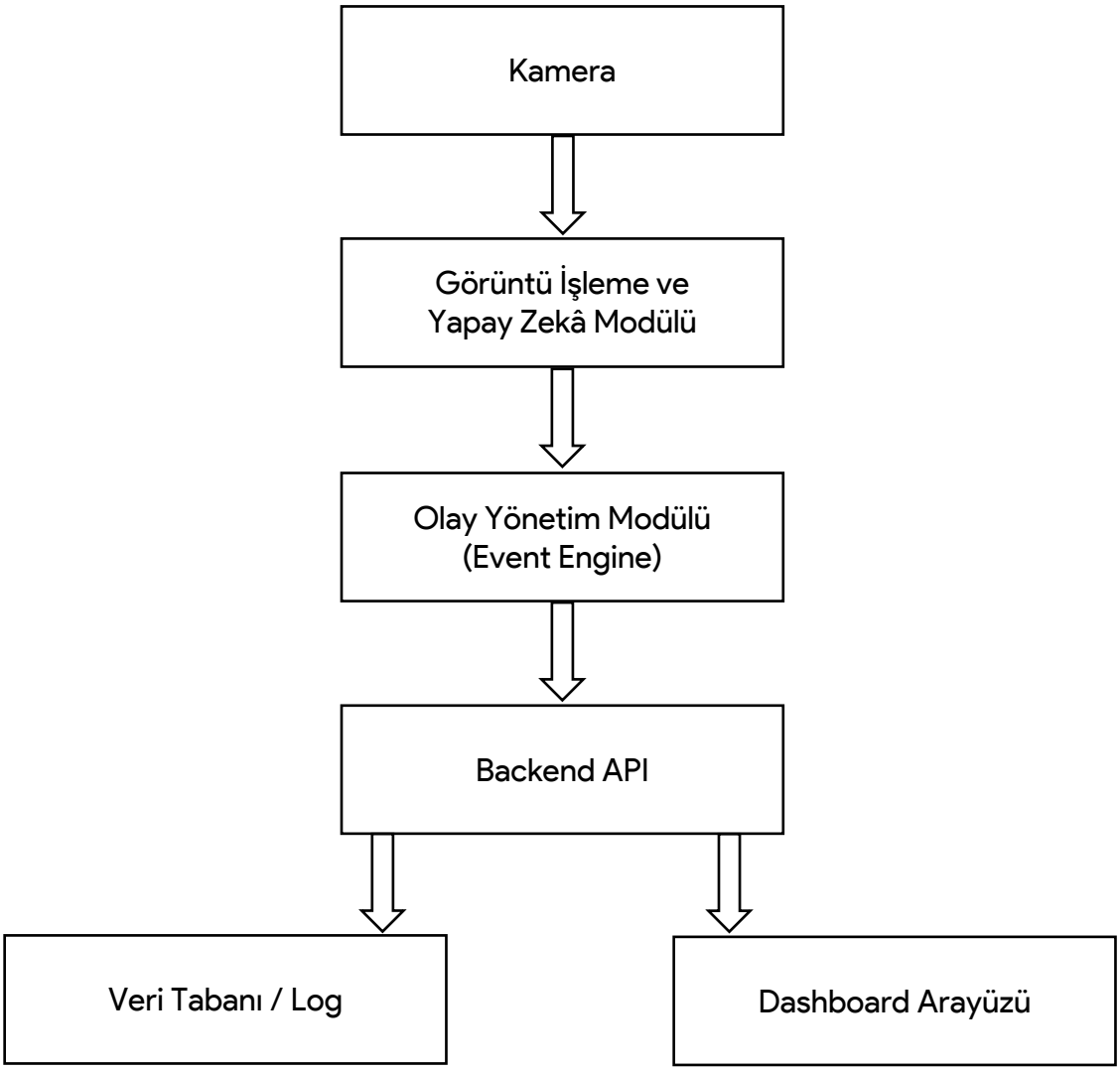
  Bu katman, sistemin beynini oluşturur. Kamera görüntülerinin işlenmesi ve anlık olarak takip edilmesi, kişi ve ürün tespiti yapılması, kayıtların (logların) tutulması ve üretilen şüpheli olayların sisteme kaydedilmesi bu katmanda gerçekleştirilir.

3.2.1.3. Veri Katmanı

- Veri tabanı / Log

  Bu katman, sistemde oluşturulan tüm gerekli verilerin kaydedilip saklandığı kısımdır. Tüm olay geçmişi ve raporlama gibi işlemler burada kaydedilen veriler kullanılarak yapılır.

3.2.2. Mimari Diyagram



3.2.3. Sistem Yerleşimi ve Çalışma Ortamı

Bu sistem, tek bir fiziksel bilgisayar üzerinde çalışacak şekilde tasarlanmıştır. Güvenlik kamerası bu bilgisayara (kablo ile ya da uzaktan) bağlıdır. Kamera üzerinden gelen veriler, bilgisayar üzerinde çalışan Görüntü İşleme ve Yapay Zekâ Modülü tarafından işlenir.

Olay Yönetim Modülü ve Backend API ile şüpheli olaylar arayüz üzerinden anlık olarak görüntülenebilir hâldedir ve aynı zamanda yerel veri tabanına (SQLite ile) da kaydedilir.

4. Scrum Planlaması

4.1. Product Backlog

ID	User Story	Task	Öncelik	Tahmini Süre
#1	Kamera görüntülerini gerçek zamanlı izleyebilme	Kamera+Yazılım bağlantısının yapılması	Yüksek	1 gün
#2	Sistemin kişi ve ürünleri algılayabilmesi	Görüntü işleme modülünün kurulması ve test edilmesi	Yüksek	2 gün
#3	Ürün, kişinin elindeyken kaybolursa bilgilendirilmek	Mesafe algoritması ve kaybolma mantığının geliştirilmesi	Yüksek	2 gün
#4	Tespit edilen olayların kayıtlarının tutulması	Event Engine tasarımı	Orta	1 gün
#5	Tespit edilen olay üretilebilmesi	Backend API	Orta	1 gün
#6	Geçmiş olayların incelenebilmesi	Dashboard tasarımı, Olay listeleme, Log kayıtları	Orta	2 gün
#7	Sistemin doğru çalışabilmesi	Testler, optimizasyon, hata ayıklama	Orta	2 gün

4.2. User Stories

- **US-1: Gerçek zamanlı görüntü işleme**
  Bir güvenlik görevlisi olarak, kamera görüntülerinin anlık olarak takip edilebilmesini istiyorum.

- **US-2: Kişi ve ürün algılama**
  Sistem olarak, kameradaki kişileri ve ürünleri doğru algılayabilmeliyim, bu sayede hatasız çalışır gereksiz kayıt tutmam.

- **US-3: Ürün kaybolma tespiti**
  Bir güvenlik görevlisi olarak, kişinin elindeki ürün kaybolduğunda müdahale edebilmek için bilgilendirilmek istiyorum.

- **US-4: Olay kaydı**
  Bir yönetici olarak, tüm şüpheli durumların kayıtlarının tutulmasını istiyorum.

4.3. Sprint Planlaması

4.3.1. Sprint 1 (Gün 1-3)

- **Sprint Goal**
  Sistemin gerçek zamanlı görüntü alması ve yapay zekâ modeli ile işlemesi.

- **Sprint Tasks**
  OpenCV ile görüntü alma
  Kamera + Python-YOLO bağlantısı
  YOLOv8n modelini çalıştırma
  Kişi ve ürünün algılandığını doğrulama, gerekli testlerin yapılması
  FPS testi ve optimizasyon

- **Beklenen Çıktı**
  Gerçek zamanlı verinin işlenerek optimizasyonu yüksek bir şekilde hedef takibi yapılabilmesi.

4.3.2. Sprint 2 (Gün 4-6)

- **Sprint Goal**
  Ürün kaybolma mantığının doğru çalışması.

- **Sprint Tasks**
  İnsan-ürün mesafe ölçüm sistemi

Threshold belirleme
Kaybolma = şüpheli olay flow'u
Olay nesnesinin oluşturulması

- **Beklenen Çıktı**
  Şüpheli olayları kurallara göre üretebilen çalışan bir Event Engine.

4.3.3. Sprint 3 (Gün 7-9)

- **Sprint Goal**
  Olayların veri tabanına kaydedilmesi.

- **Sprint Tasks**
  SQLite ile backend oluşturma
  /event/send endpoint geliştirme
  Veri tabanı/log dosyası oluşturma
  Olayların backend'e gönderilmesi

- **Beklenen Çıktı**
  Olayların Backend API üzerinden alınarak SQLite ve CSV loglara kaydedilmesi.

4.3.4. Sprint 4 (Gün 10-11)

- **Sprint Goal**
  Yetkililerin şüpheli olayları inceleyebilmesi.

- **Sprint Tasks**
  Dashboard ile olay listeleme
  Tüm sistemin test edilmesi
  Küçük optimizasyonlar

- **Beklenen Çıktı**
  Tam çalışan bir Minimum Viable Product (MVP) prototipi.

5. Algoritmik Yapı ve Akış Diyagramı

5.1. Algoritmik Yapı

- Sistem başlatılır
- Kamera görüntüsü ana makineye sürekli iletilir
- Ana makinede her kare için kamera görüntüsü işlenir ve yapay zekâ modülü çalıştırılır
- Model çıktısında kişi ve ürün ayrımı yapılır

- Algılanan kişi ve ürünler kareler arasında takip edilir
- Kişi/ürün mesafe hesaplanır
- Ürün, kişiye yakın durumdayken belirlenen süre boyunca tespit edilemezse şüpheli durum oluşur
- Şüpheli durumu yaratan kişi sistem arayüzünde kırmızı renkli bir kutucuk içinde (riskli kişi olarak) gösterilir
- Şüpheli duruma ait bilgiler yetkililere bildirilmek üzere Backend API'a gönderilir
- Şüpheli durum olayı, log kayıtlara eklenir ve veri tabanına kaydedilir
- Sistem çalıştığı sürece bu adımlar her yeni kare için tekrarlanır

5.2. Akış Diyagramı