

MOT17 Video Analizi: Takip ve Sayım Teknik Raporu

Mehmet Özer - ozermehmet911@gmail.com

Trio Mobil AI intern Case Study - 2

1. Özeti

Bu çalışmada MOT17 veri seti üzerinde kişi takibi ve sayımı gerçekleştirdim. Projenin hedefi, GPU gerektirmeyen hafif bir sistem kurmak ve özellikle kalabalık sahnelerde sık karşılaşılan occlusion problemine çözüm üretmektı.

YOLOv8n ile ByteTrack mantığını temel alan bir tracker geliştirdim. Kalman Filter ile hareket tahmini ve Hungarian Algorithm ile nesne ve iz eşleştirme yaparak, sanal çizgi üzerinden giriş/çıkış sayımı gerçekleştirdim.

Testler sonucunda düşük yoğunluklu sahnelerde (MOT17-04) %97.6 Precision ve 0.856 F1 skoru elde ettim. Kalabalık sahnelerde ise ID değişme sorunlarıyla karşılaştım ve bunların nedenlerini analiz ettim.

2. Teknik Yaklaşım

Sistem üç modülden oluşan bir pipeline mimarisine sahiptir:

2.1. Tespit Modülü (Detection)

CPU'da makul sürede çalışabilmesi için **YOLOv8 Nano (yolov8n)** modelini tercih ettim. Hafif ve CPU'da hızlı çalışıyor.

- Confidence Threshold (0.35):** Standart 0.50 yerine 0.35 seçtiğimde, kalabalıkta arkada kalan veya kısmen görünen kişilerin tespit edilme başarısı arttı.
- Class Filter:** Sadece "Person" sınıfı işlenmektedir.

2.2. Takip Modülü (ByteTrack)

Hazır bir kütüphane (DeepSORT vb.) kullanmak yerine, problem için bir **ByteTracker** sınıfı geliştirilmiştir.

- Hareket Tahmini (Kalman Filtresi):** Sadece anlık konuma güvenmek yerine, kişilerin hız ve yönünü de modelliyyorum. Constant Velocity modeli kullanıyorum. Örnek olarak; Bir kişi başka birisinin arkasında kaldığında 3-4 frame tespit edilmeyebilir. Ama sistem bu sağa gidiyordu, şimdi şurallarda olmalı diyerek tahmin yapıyor ve iz kaybını önlemeye çalışıyor.
- Veri Eşleştirme (Hungarian Algorithm):** Her frame'de tespit edilen bounding box'ları mevcut tracklerle eşleştirirken greedy (açgözlü) yöntem yerine global hatayı minimize eden Hungarian algoritmasını kullandım. Maliyet matrisi: $1 - \text{IoU} \rightarrow \text{IoU}$ yüksekse (kutular çakışıyorsa) maliyet düşük olur, eşleşme her zaman öncelikli

- **Çift Aşamalı Eşleştirme:** Önce yüksek confidence'lı detectionlar eşleştirilir. Eşleşmeyen trackler için düşük confidence'lı detectionlara bakılır. Bu sayede, silik/bulanık da olsa görünen kişiler tracker tarafından yakalanabiliyor ve ID switch azalıyor.

2.3. Sayım Modülü

Videolara sanal bir çizgi tanımladım (sequence ye göre dikey veya yatay).

- **Geometri:** Kutunun merkezi yerine, zemine temas eden alt-orta noktayı referans aldım. Bu sayede kamera açısından dolayı oluşan perspektif etkisini ve buna bağlı konum hatalarını azalttım.
- **Vektörel Analiz:** Kişinin önceki konumu ($t-1$) ile şimdiki konumu (t) arasındaki vektörün sanal çizgisi kesip kesmediğini ve hangi yönde geçtiğini hesaplıyorum.

3. Performans Analizi ve Sonuçlar

MOT17 veri setinden 3 sekans üzerinde testler yapıldı.

Sequence	Precision	Recall	F1	ID Switches	Fragments	Entry	Exit	Total
MOT17-09	96.0%	65.0%	0.775	206	145	9	3	12
MOT17-02	77.2%	48.3%	0.594	82	95	5	4	9
MOT17-04	97.6%	76.2%	0.856	101	384	12	8	20

Tablo 1: Farklı senaryolarda sistemin performans metrikleri.

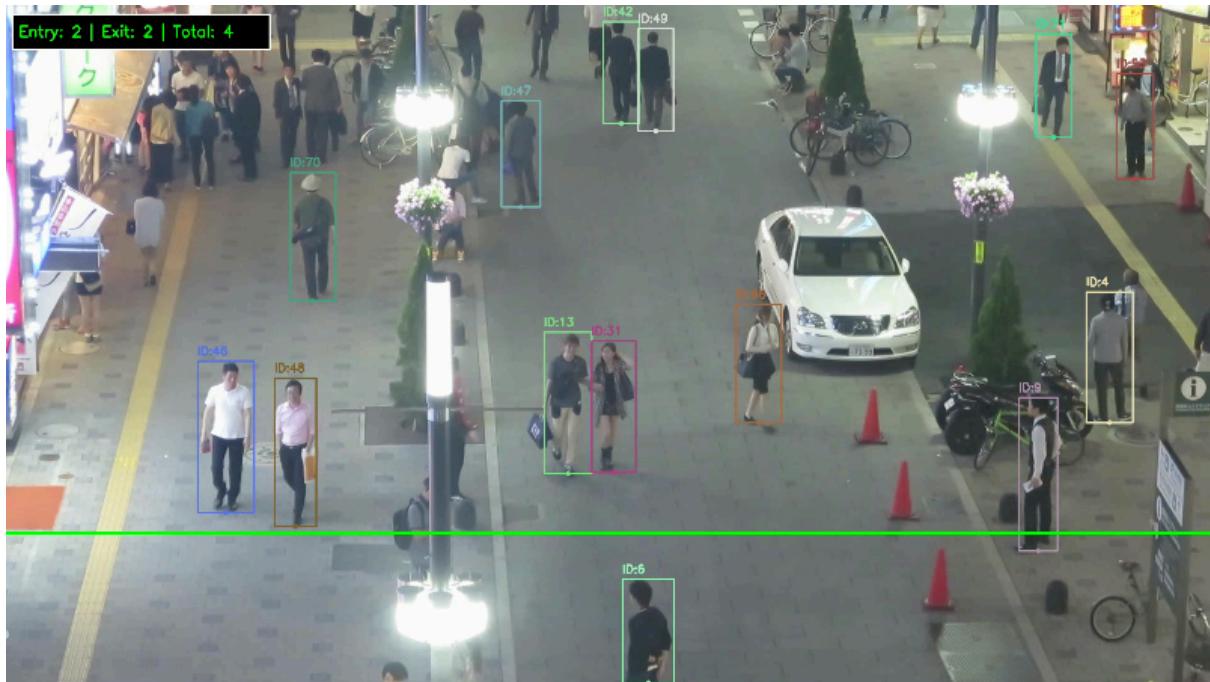
Yorum:

1. **En İyi Performans (MOT17-04):** Gece çekimi olmasına rağmen yolov8n başarılı. Neden? Kalabalık az, kamera açısı tepede olduğu için kişiler birbirini pek örtmüyor. Kalman filter güzel çalışıyor.
2. **En Zorlu Senaryo (MOT17-02):** Dar sokak + yan açı kamera. İnsanlar yan yana yürüyünce iç içe geçiyor, YOLO arkadaki kişileri tespit edemiyor. Detectionda kayıplar olunca tracking de çok doğru sonuçlar vermiyor.
3. **ID Switch Analizi (MOT17-09):** AVM girişi, sürekli hareket var. Bir kişi başkasının önünden geçtiğinde 5-10 frame kaybolabiliyor. Track buffer 100 frame'e çıkarmama rağmen, kişi geri geldiğinde Kalman tahminiyle yeni konum örtüşmüyordu -> sistem yeni ID atıyor.

4. Hata Analizi ve Görselleştirme

Sistemin nerelerde başarılı, nerelerde başarısız olduğunu anlamak için 3 farklı senaryo görselleştirdim.

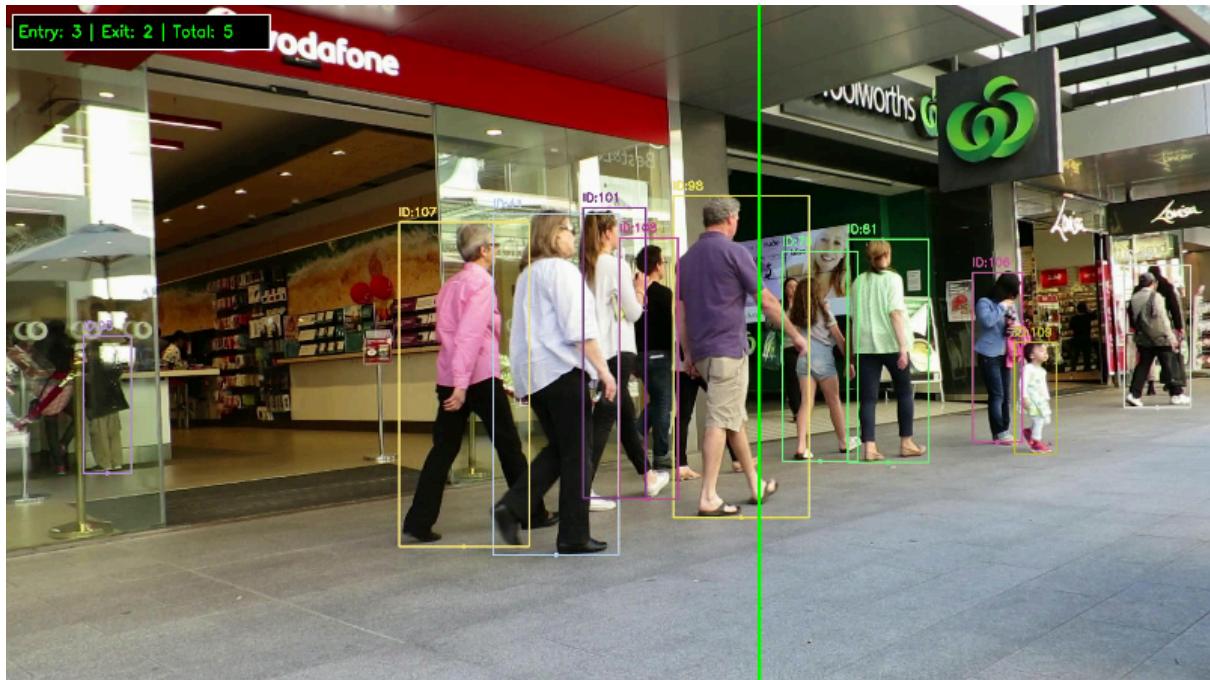
4.1. Başarılı Takip Örneği



Şekil 1: MOT17-04 Sekansı.

Bu videoda kamera tepe açıda, kalabalık az, kişiler birbirinden yeterince uzak. Tracker başarılı çalışıyor: - ID:46 ve ID:48 neredeyse tüm video boyunca aynı kaldı - Yeşil çizgi üzerindeki geçişler doğru sayılıdı - Precision %97.6 - çok az false positive var

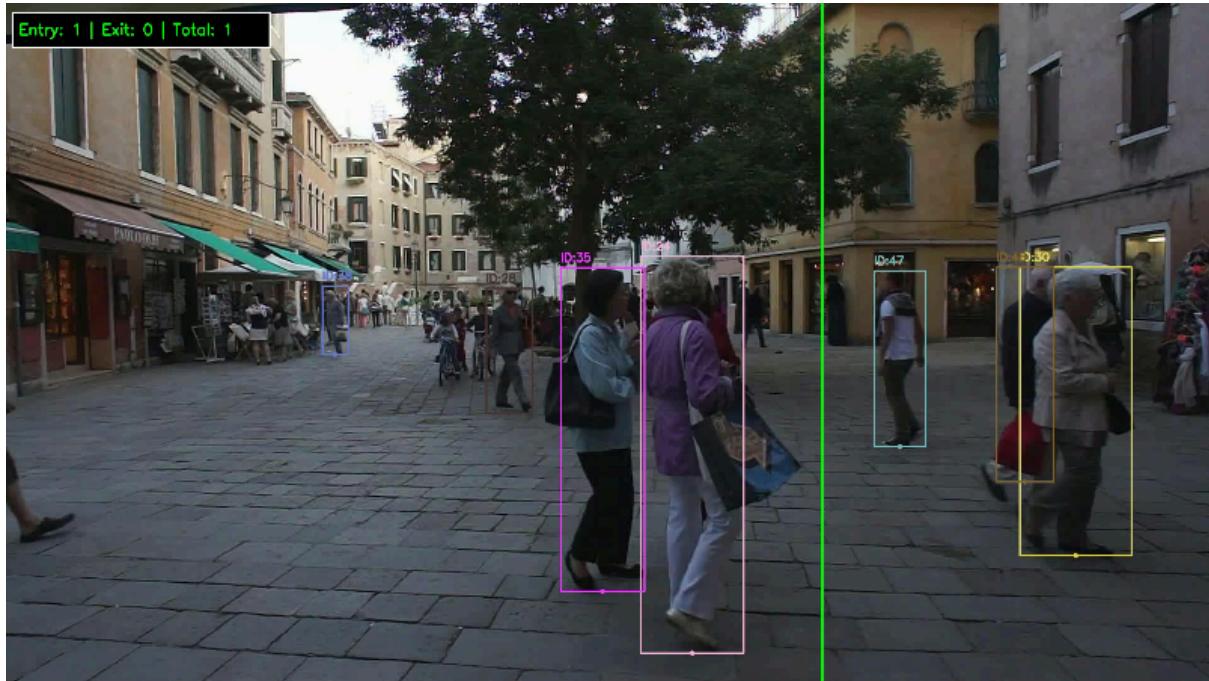
4.2. ID Switch Sorunu



Şekil 2: MOT17-09 Sekansı.

Bu videoda dikkatimi çeken şey: iki kişi yan yana yürürken biri öne geçtiğinde arka plandaki kişinin detection'ı tamamen kayboluyor. Track buffer 100 frame bekliyor ama kişi 80. frame'de geri geldiğinde: - Kalman'ın tahmin ettiği konum gerçek konum ile eşleşmiyor bu yüzden yeni ID atıyor. İşte tablodaki 206 ID switch'in sebebi bu. Fragmentation da buradan geliyor (aynı kişi 3-4 farklı track olarak parçalanıyor).

4.3. Örtüşme Kaynaklı Tespit Kaybı



Şekil 3: MOT17-02 Sekansı.

Kamera yan açıdan bakıyor, hafif bir titreme var, insanlar iç içe geçmiş durumda. İnsanlar yan yana yürürken üst üste biniyorlar. Tracker aynı kişiyi birden fazla kez sayabiliyor veya ID karıştırabiliyor. Özellikle sağ tarafta yürüyen kişiler birbirinin önünden geçenken kutular örtüşüyor, bu da ID switche yol açıyor.

5. İyileştirme Önerileri

Eğer proje için daha fazla kaynak (GPU, zaman) olsaydı neler denerdim:

- DeepSORT (Görünüş Tabanlı Takip):** Mevcut sistem sadece konum (IoU) kullanmaktadır. Sisteme bir Re-ID (Re-Identification) ağı eklenebilir, kişilerin kıyafet renkleri ve özellikleri vektörel olarak saklanabilir. Böylece bir kişi kaybolup geri gelse bile bu kişi ID:45'in vektörüyle benziyor denilerek eski ID geri verilebilir. Bu, ID Switch hatasını ~%50 oranında azaltabilir.
- Adaptif Parametreler:** Şu an tüm videolarda aynı parametreleri kullanıyorum. Sahne yoğunluğuna göre **track_buffer** ve **confidence_threshold** değerlerinin dinamik olarak değişmesi sağlanabilir.
- Kamera Hareketi Telafisi (CMC):** MOT17-02 sekansında kamera hafif hareket ediyor. Bu Kalman filter'i yanıltabilir çünkü kişi duruyorken bile konumu değişiyor gibi görünüyor. Optical Flow kullanılarak kamera hareketi hesaplanıp, nesne hızından düşülebilir.

6. Sonuç

Bu projede CPU da makul sürede çalışan bir sistem kurdum. Tracking mantığını daha iyi kavradım.

En çok MOT17-02'de zorlandım, yan kamera + kalabalık kombinasyonu tespit ve takibi zorlaştırmıştı. MOT17-04'te 0.856 F1 güzel bir sonuç, düşük yoğunluklu sahnelerde sistem iyi çalışıyor. Kalabalık sahnelerde ise ID switch problemi belirgin, DeepSORT gibi bir algoritma ile bu sorunda çözülebilir.