

# Piksel Tabanlı Hız Tahmini ve Deneysel Uzaklık Ayarlaması ile Güncellenen Araç Hız Tespiti

## Özet

Bu makale, akıllı ulaşım sistemleri için kritik bir görev olan araç hız tespiti konusunda, geleneksel homografi tabanlı gerçek dünya koordinatlarına dönüştürme yönteminden, piksel tabanlı hız tahmini ve deneysel uzaklık ayarlamasına dayalı yeni bir metodolojiye geçişi sunmaktadır. Homografi tabanlı yaklaşımlar, doğruluk potansiyeli sunsa da, genellikle karmaşık ve sahneye özgü kamera kalibrasyonu gerektirmeleri nedeniyle pratik uygulamalarda zorluklar çıkarabilmektedir. Önerilen bu çalışma, kalibrasyon ihtiyacını ortadan kaldırmayı hedefleyerek, nesne tespiti için YOLOv8 modelini <sup>1</sup>, araç takibi için ByteTrack algoritmasını <sup>2</sup> ve hız tahmini için ardışık karelerdeki araçların piksel koordinatlarındaki değişimi kullanmaktadır. Gerçek dünya hızına geçiş, başlangıçta varsayılan bir ortalama araç genişliği kullanılarak tahmin edilmekte, ardından sınırlayıcı kutunun köşegen uzunluğundaki değişim oranına dayalı tamamen deneysel bir uzaklık ayarlaması ile bu tahmin iyileştirilmeye çalışılmaktadır. Yapılan deneysel değerlendirmeler, önerilen yöntemin homografi tabanlı yaklaşıma kıyasla daha yüksek bir işlem hızı (yaklaşık 40 FPS) sunduğunu, ancak hız tahmin doğruluğunda beklenen bir düşüşle (yaklaşık 6.5 km/sa RMSE) sonuçlandığını göstermektedir. Deneysel ayarlamasının belirli senaryolarda sınırlı bir iyileştirme sağladığı gözlemlenmiştir. Bu çalışma, kalibrasyonsuz hız tahmini için potansiyel bir alternatif sunarken, piksel tabanlı tahminin ve önerilen deneysel ayarlamasının avantajlarını, doğruluk-hız ödünleşimini ve mevcut sınırlamalarını detaylı bir şekilde ortaya koymaktadır.

## Giriş

Araç hız tespiti, trafik akışının yönetilmesi, yol güvenliğinin artırılması ve otonom sürüş sistemlerinin geliştirilmesi gibi birçok akıllı ulaşım sistemi (ITS) uygulamasının temel taşlarından biridir.<sup>2</sup> Araçların hızlarının doğru ve verimli bir şekilde belirlenmesi, trafik sıkışıklığının önlenmesi, kaza risklerinin azaltılması ve daha akıllı şehir altyapılarının oluşturulması için kritik bilgiler sağlar.<sup>5</sup>

Geçmişteki çalışmalarımızda, araç hızlarını tahmin etmek için homografi tabanlı bir yaklaşım benimsenmiştir. Bu yöntem, kamera görüntüsündeki piksel koordinatlarını, bir homografi matrisi aracılığıyla gerçek dünya düzlemindeki koordinatlara dönüştürme prensibine dayanır.<sup>1</sup> Bu dönüşüm sayesinde, görüntüdeki pikseller başına sabit bir gerçek dünya mesafesi değeri elde edilir ve araçların ardışık karelerdeki piksel hareketleri doğrudan gerçek dünya hızlarına çevrilebilir.<sup>1</sup> Bu yaklaşım, geometrik olarak doğru sonuçlar verme potansiyeline sahiptir. Ancak, homografi tabanlı yöntemlerin en önemli ve zorlayıcı gereksinimi, hassas bir kamera kalibrasyonudur.<sup>2</sup> Kalibrasyon süreci, özel ekipman veya belirli sahne geometrisi bilgisi

gerektirebilir, zaman alıcıdır ve çevresel değişikliklere (örn. kamera titreşimi) karşı hassastır. Ayrıca, bu yöntemler genellikle yolun düz olduğu varsayımına dayanır ki bu varsayım her zaman geçerli olmayabilir ve kalibrasyonun doğruluğunu etkileyebilir.<sup>1</sup>

Bu pratik zorlukların ve sınırlamaların üstesinden gelmek, daha basit, daha esnek ve potansiyel olarak daha hızlı bir çözüm sunmak amacıyla, metodolojimizi piksel tabanlı hız tahminine kaydirdık. Bu yeni yaklaşım, kamera kalibrasyonu yapma ihtiyacını ortadan kaldırarak, araç hızını doğrudan ardışık video karelerindeki piksel koordinatlarının değişiminden tahmin etmeyi amaçlamaktadır. Tespit edilen araçların takibi için gelişmiş algoritmalar kullanılırken<sup>3</sup>, nesne tespiti için yüksek performanslı YOLOv8 modeli tercih edilmiştir.<sup>2</sup> Piksel cinsinden hesaplanan hızı gerçek dünya birimlerine (örn. km/sa) dönüştürmek için ise iki aşamalı bir yaklaşım geliştirilmiştir: İlk olarak, ortalama bir araç genişliği varsayımı kullanılarak kaba bir ölçekleme yapılır; ikinci olarak, aracın sınırlayıcı kutusunun köşegen uzunluğundaki değişim oranına dayalı tamamen deneysel bir mekanizma ile bu ilk hız tahmini ayarlanmaya çalışılır. Bu deneysel ayarlama, aracın kameraya yaklaşıp uzaklaşmasıyla değişen görünür boyutundan faydalanarak hız tahminini iyileştirmeyi hedefler.

Bu makalenin temel katkısı, kalibrasyon gerektirmeyen bir hız tespit yöntemini detaylandırmak ve bu yöntemin performansını, özellikle de deneysel uzaklık ayarlamasının etkinliğini değerlendirmektir. Çalışma, homografi tabanlı yöntemlerin pratik zorluklarına bir alternatif sunarken, önerilen piksel tabanlı yaklaşımın doğasında var olan doğruluk ve hız arasındaki ödünleşmeyi (trade-off) vurgulamaktadır. Makalenin devamında, ilgili literatürdeki çalışmalar özetlenecek, önerilen metodolojinin her bir adımı ayrıntılı olarak açıklanacak, elde edilen deneysel sonuçlar sunulacak, bu sonuçlar tartışılacak ve gelecek çalışmalar için önerilerde bulunulacaktır.

## İlgili Çalışmalar

Araç hız tespiti, bilgisayarlı görü ve akıllı ulaşım sistemleri alanlarında uzun yıllardır aktif bir araştırma konusudur ve literatürde çok çeşitli yöntemler önerilmiştir. Bu bölümde, önerilen piksel tabanlı yaklaşımın bağlamını oluşturmak ve literatürdeki yerini belirlemek amacıyla başlıca yöntemler incelenmektedir.

- **Homografi Tabanlı Yöntemler:** Bu yöntemler, kamera görüntüsünü, genellikle bir kuşbakışı (bird's-eye view) görünümüne dönüştürmek için bir homografi matrisi kullanır.<sup>1</sup> Bu dönüşüm, görüntü düzlemindeki pikseller ile gerçek dünya düzlemi arasında bir eşleme sağlar, böylece piksel başına sabit bir gerçek dünya mesafesi oranı elde edilir.<sup>1</sup> Araçların ardışık karelerdeki piksel hareketleri bu oran kullanılarak gerçek dünya hızına çevrilir. Derin öğrenme tabanlı homografi ağları, kalibrasyon sürecini otomatikleştirmek veya basitleştirmek için önerilmiştir.<sup>2</sup> Ancak, bu yöntemlerin temel dezavantajı, doğru bir homografi matrisi elde etmek için genellikle hassas kamera kalibrasyonu veya bilinen dünya noktalarına ihtiyaç duymalarıdır.<sup>2</sup> Ayrıca, genellikle düz bir zemin varsayımına dayanırlar ve bu varsayımın ihlali doğruluk sorunlarına yol açabilir.<sup>1</sup> Bu kalibrasyon zorlukları, önerilen çalışmanın temel motivasyonlarından biridir.

- **Optik Akış Tabanlı Yöntemler:** Optik akış, ardışık video kareleri arasındaki piksellerin veya özellik noktalarının görünür hareketini temsil eder.<sup>10</sup> Lucas-Kanade veya Farneback gibi klasik algoritmalar veya derin öğrenme tabanlı yaklaşımlar, bu hareket vektörlerini tahmin etmek için kullanılır.<sup>10</sup> Bu vektörler, araçların hareketini izlemek ve hızlarını tahmin etmek için kullanılabilir; örneğin, aracın ağırlık merkezinin kareler arasındaki yer değiştirmesi hesaplanarak hız bulunabilir.<sup>10</sup> Optik akış, yoğun hareket bilgisi sağlama potansiyeline sahip olsa da, genellikle gürültüye karşı hassastır, hesaplama maliyeti yüksektir ve doğru ölçekleme için yine kamera kalibrasyonuna veya ek bilgilere ihtiyaç duyabilir.<sup>10</sup> Önerilen yöntem, optik akış gibi piksel hareketini kullansa da, nesne tespiti ve takibine dayandığı için potansiyel olarak daha az hesaplama yoğunudur ve farklı bir ölçekleme yaklaşımı benimser.
- **Kalman Filtresi Tabanlı Yöntemler:** Kalman filtreleri, zaman içinde değişen sistemlerin durumunu (örneğin, bir aracın konumu ve hızı) gürültülü ölçümlerden tahmin etmek için yaygın olarak kullanılan istatistiksel bir araçtır.<sup>12</sup> Araç hız tespiti bağlamında, Kalman filtreleri genellikle araç takip algoritmalarının (SORT, DeepSORT, ByteTrack gibi) bir parçası olarak kullanılır.<sup>2</sup> Bu filtreler, önceki durum tahminini ve mevcut ölçümü (tespit edilen konum) birleştirerek aracın bir sonraki konumunu ve hızını tahmin eder, böylece ölçüm gürültüsünü filtreler ve kısa süreli oklüzyonlarda bile takibi sürdürmeye yardımcı olur.<sup>5</sup> Ayrıca, farklı sensörlerden (örn. ivmeölçer, GPS) gelen verileri birleştirmek için de kullanılabilirler.<sup>12</sup> Kalman filtreleri gürültüyü azaltmada etkili olsa da, performansları büyük ölçüde kullanılan hareket modelinin doğruluğuna bağlıdır ve tek başlarına hız tespiti için tam bir çözüm sunmazlar, genellikle bir takip sisteminin parçasıdırlar.
- **Derin Öğrenme Tabanlı Yöntemler (Doğrudan Regresyon):** Son yıllarda, derin sinir ağları (özellikle Evrişimli Sinir Ağları - CNN'ler ve Tekrarlayan Sinir Ağları - RNN'ler) doğrudan video karelerinden veya kare dizilerinden araç hızını tahmin etmek için kullanılmaktadır. Bu yaklaşımlar, genellikle büyük veri setleri üzerinde eğitilerek görüntüdeki görsel ipuçlarından veya zamansal değişimlerden hız ile ilgili karmaşık özellikleri öğrenirler. Bu yöntemler, potansiyel olarak kalibrasyon ihtiyacını ortadan kaldırabilir ve uçtan uca bir çözüm sunabilir. Ancak, genellikle büyük miktarda etiketli veriye ihtiyaç duyarlar ve modelin yorumlanabilirliği zor olabilir.
- **LiDAR ve Radar Füzyonu Tabanlı Yöntemler:** LiDAR (Lazerle Tespit ve Menzil Belirleme) ve Radar (Radyo ile Tespit ve Menzil Belirleme) sensörleri, araç hızını yüksek doğrulukla ölçmek için kullanılabilir.<sup>11</sup> LiDAR, nesnelere olan mesafeyi hassas bir şekilde ölçerken, Radar Doppler etkisi sayesinde doğrudan hız bilgisini sağlayabilir. Bu iki sensörden gelen verilerin birleştirilmesi (füzyon), özellikle otonom araç uygulamalarında, güvenilir ve doğru hız tahminleri sağlayabilir.<sup>16</sup> Ancak bu yöntemler, kamera tabanlı sistemlere göre önemli ölçüde daha yüksek maliyet ve donanım karmaşıklığı gerektirir.<sup>11</sup>

- **Önerilen Yaklaşımın Konumlandırılması:** Bu çalışmada sunulan piksel tabanlı hız tahmini ve deneysel uzaklık ayarlaması yöntemi, yukarıda belirtilen yaklaşımlardan farklılaşmaktadır. Homografi tabanlı yöntemlerin aksine, karmaşık kamera kalibrasyonu gerektirmez. Optik akış yöntemleri gibi piksel hareketini kullansa da, nesne tespiti ve takibine odaklanarak potansiyel olarak daha az hesaplama maliyeti sunar. Kalman filtresi, kullanılan takip algoritmasının (ByteTrack) zaten bir parçasıdır. LiDAR/Radar gibi pahalı sensörlere ihtiyaç duymaz. Yöntemin asıl yeniliği, piksel tabanlı hız tahminine gerçek dünya ölçeği kazandırmak için *varsayılan araç genişliği* ve *deneysel köşegen değişimi ayarlaması* gibi basitleştirilmiş ve kalibrasyonsuz yaklaşımları birleştirmesidir. Bu yaklaşım, basitlik ve uygulama kolaylığı potansiyeli sunarken, doğruluk konusunda diğer daha karmaşık veya kalibrasyonlu yöntemlere göre ödünleşimler içermesi beklenmektedir.

### Önerilen Yöntem

Bu bölümde, araç hız tespiti için geliştirilen güncellenmiş metodolojinin bileşenleri ve işleyişi ayrıntılı olarak açıklanmaktadır. Sistem, bir video akışındaki her kare için sırasıyla nesne tespiti, araç takibi, piksel tabanlı hız hesaplama ve gerçek dünya hızı tahmini adımlarını uygular.

### Genel Bakış

Önerilen sistemin iş akışı şu adımlardan oluşur:

1. Video akışından bir kare alınır.
2. YOLOv8 modeli kullanılarak karedeki araçlar tespit edilir ve sınırlayıcı kutuları belirlenir.
3. ByteTrack algoritması kullanılarak mevcut karedeki tespitler önceki karelerdeki araçlarla ilişkilendirilir ve her araca benzersiz bir kimlik atanarak takibi sağlanır.
4. Takip edilen her araç için, mevcut ve bir önceki karedeki referans noktaları (alt orta nokta) arasındaki piksel cinsinden yer değiştirme hesaplanır.
5. Piksel yer değiştirmesi ve kareler arası zaman kullanılarak aracın hızı piksel/saniye cinsinden bulunur.
6. Varsayılan bir ortalama araç genişliği ve aracın sınırlayıcı kutusunun piksel cinsinden genişliği kullanılarak anlık bir piksel-metre dönüşüm oranı tahmin edilir.
7. Piksel hızı, bu dönüşüm oranı kullanılarak ilk gerçek dünya hızı tahminine (m/s veya km/sa) dönüştürülür.
8. Sınırlayıcı kutunun köşegen uzunluğunun ardışık karelerdeki değişim oranı hesaplanır.
9. Bu değişim oranına göre, önceden belirlenmiş eşik değerleri ve ayarlama faktörleri kullanılarak ilk gerçek dünya hızı tahmini deneysel olarak ayarlanır ve nihai hız tahmini elde edilir.

### YOLOv8 Nesne Tespit Modeli

Video karelerindeki araçları (otomobil, kamyon, otobüs, motosiklet vb.) tespit etmek için YOLOv8 (You Only Look Once version 8) nesne tespit modeli kullanılmıştır.<sup>1</sup> YOLOv8, özellikle gerçek zamanlı uygulamalar için uygun olan yüksek tespit hızı, iyi doğruluk oranı ve farklı ölçeklerde (örn. YOLOv8n, YOLOv8s, YOLOv8m, YOLOv8l, YOLOv8x) sunulan model boyutları nedeniyle tercih edilmiştir.<sup>2</sup> Bu çalışmada, doğruluk ve hız arasında iyi bir denge sunan

YOLOv8x modeli kullanılmıştır.<sup>2</sup> Model, girdi olarak aldığı her video karesi için, tespit ettiği her araç nesnesine ait bir sınırlayıcı kutu (bounding box) koordinatları (genellikle sol üst köşe  $(x_{min}, y_{min})$  ve sağ alt köşe  $(x_{max}, y_{max})$  veya merkez  $(x_c, y_c)$ , genişlik  $w$  ve yükseklik  $h$ ), bir güven skoru ve bir sınıf etiketi (örn. 'araba', 'kamyon') üretir.

Takip ve hız hesaplamalarında referans noktası olarak, her sınırlayıcı kutunun alt orta noktasının piksel koordinatları

$(c_x, c_y)$  hesaplanır

$c_x = x_{min} + w = x_{min} + x_{max}$

$c_y = y_{max}$

Bu noktanın seçilme nedeni, genellikle aracın yolla temas ettiği noktaya daha yakın olması ve dikey perspektif etkilerini kısmen azaltma potansiyelidir.<sup>1</sup> Ancak, sınırlayıcı kutunun alt kenarının titremesi durumunda bu nokta da kararsızlık gösterebilir. Alternatif olarak, kutunun geometrik merkezi  $(x_c, y_c)$  de kullanılabilir.

### Araç Takibi (ByteTrack)

Tespit edilen araçların video boyunca sürekliliğini sağlamak ve her birine benzersiz bir kimlik atamak için ByteTrack takip algoritması kullanılmıştır.<sup>2</sup> ByteTrack, SORT<sup>4</sup> ve DeepSORT<sup>4</sup> gibi önceki popüler takip algoritmalarının üzerine inşa edilmiş ve özellikle oklüzyon durumlarında ve düşük güven skorlu tespitlerde daha iyi performans göstermeyi hedefleyen bir yöntemdir.<sup>3</sup>

ByteTrack'in temel çalışma prensibi şöyledir<sup>5</sup>:

- Yüksek Güvenli Tespitlerle İlişkilendirme:** Önceki karelerdeki takip edilen nesnelerin (tracklet) mevcut karedeki konumları Kalman filtresi kullanılarak tahmin edilir. Bu tahmin edilen konumlar, mevcut karedeki yüksek güven skoruna sahip tespit kutuları ile eşleştirilir. Eşleştirme genellikle IoU (Intersection over Union) veya görünüm özellikleri (Re-ID) kullanılarak yapılır.
- Düşük Güvenli Tespitlerle İlişkilendirme:** İlk eşleştirmeden sonra eşleşmeyen kalan takip edilen nesneler (genellikle oklüzyon nedeniyle tespiti zorlaşanlar) ve eşleşmeyen kalan yüksek güvenli tespitler, bu sefer mevcut karedeki *düşük* güven skoruna sahip tespit kutuları ile eşleştirilmeye çalışılır. Bu adım, düşük kaliteli veya kısmen görünen tespitlerin de takip sürecine dahil edilmesini sağlar.
- Yeni Takip Başlatma:** Eşleşmeyen kalan yüksek güven skorlu tespitler, yeni araçlar olarak kabul edilir ve yeni takip kimlikleri başlatılır.
- Kalman Filtresi Güncelleme:** Eşleşen her takip edilen nesne için Kalman filtresinin durumu (konum, hız vb.) güncellenir. ByteTrack, durum tahmini için standart bir lineer hareket modeli ve Kalman filtresi kullanır.<sup>2</sup>

ByteTrack'in düşük güvenli tespitleri de kullanması, onu özellikle yoğun trafik veya nesnelerin birbirini kapattığı durumlar için daha dayanıklı hale getirir.<sup>3</sup> Bu çalışmada ByteTrack'in standart

uygulaması kullanılmıştır.

### **Perspektif Dönüşümü (Homografi) - Kullanılmayan Yöntem**

Tekrar belirtmek gerekirse, orijinal metodolojide kullanılan ve kamera görüntüsünü kuşbakışı görünüme çeviren homografi tabanlı perspektif dönüşümü <sup>1</sup>, güncellenen bu yöntemde *kullanılmamaktadır*. Bu değişikliğin temel amacı, sistemin kurulumunu ve uygulanmasını basitleştirmek ve zorlu kamera kalibrasyonu adımını ortadan kaldırmaktır.<sup>2</sup>

### **Hız Hesaplama ve Açı Analizi**

Takip algoritması tarafından sağlanan, her bir aracın ardışık karelerdeki (cxpixel,cypixel) koordinatları kullanılarak hız ve hareket açısı hesaplanır.

- **Piksel Tabanlı Hız Hesaplama:**

Bir aracın  $t$  anındaki(karen $t$ ) konumu  $(cxn,cyn)$ ve bir önceki  $t-\Delta t$  anındaki(karen $t-1$ ) konumu

$(cxn-1,cyn-1)$  olsun. Kareler arası zaman farkı  $\Delta t = 1/\text{FPS}$ 'dir, burada FPS (Frames Per Second) videonun kare hızıdır. İki kare arasındaki piksel cinsinden yer değiştirme  $\Delta d_{\text{pixel}}$  Öklid mesafesi ile hesaplanır:

$\Delta d_{\text{pixel}} = \sqrt{(cxn - cxn-1)^2 + (cyn - cyn-1)^2}$  Aracın piksel tabanlı hızı  $V_{\text{pixel}}$  (piksel/saniye) şu şekilde bulunur:

$$V_{\text{pixel}} = \Delta t \Delta d_{\text{pixel}} = \Delta d_{\text{pixel}} \times \text{FPS}$$

Bu değer, aracın görüntü düzlemindeki hareket hızını verir ancak gerçek dünya hızını doğrudan yansıtmaz.<sup>1</sup>

- **Gerçek Dünya Hızının Tahmini (Varsayılan Araç Genişliği ile):**

Piksel hızını gerçek dünya hızına ( $V_{\text{real}}$ , örn. km/sa) dönüştürmek için bir ölçek faktörüne ihtiyaç vardır. Bu çalışmada, ilk tahmin için basitleştirilmiş bir yaklaşım olarak, ortalama bir binek aracın standart genişliği varsayılır (örn. ASSUMED\_CAR\_WIDTH\_METERS = 1.8 metre).<sup>18</sup> Aracın mevcut karedeki sınırlayıcı kutusunun piksel cinsinden genişliği  $w_{\text{pixel}}$  kullanılarak anlık bir piksel-metre oranı (PPM) tahmin edilir:

$$\text{PPM}_{\text{estimated}} = \frac{\text{ASSUMED\_CAR\_WIDTH\_METERS}}{w_{\text{pixel}}}$$

Bu tahmini PPM değeri kullanılarak piksel hızı gerçek dünya hızına dönüştürülür:

$$V_{\text{real\_initial}}(\text{m/s}) = \text{PPM}_{\text{estimated}} V_{\text{pixel}} = w_{\text{pixel}} V_{\text{pixel}} \times \text{ASSUMED\_CAR\_WIDTH\_METERS}$$

Sonuç genellikle km/sa'e çevrilir ( $V_{\text{real\_initial}}(\text{km/h}) = V_{\text{real\_initial}}(\text{m/s}) \times 3.6$ ).

Bu yöntemin en kritik zayıflığı, tüm araçların (kamyonlar, motosikletler dahil) aynı genişliğe sahip olduğunun varsayılması ve perspektif etkisinin (uzaktaki araçların piksellerde daha küçük görünmesi) göz ardı edilmesidir.<sup>10</sup> Bu durum, özellikle farklı boyutlardaki araçlar ve farklı uzaklıklar için önemli hız tahmin hatalarına yol açar.

- **Hareket Açısı Hesaplama:**

Aracın anlık hareket yönünü belirlemek için, ardışık karelerdeki yer değiştirme vektörünün yatay eksenle yaptığı açı  $\theta$  hesaplanır:

$$\theta = \text{atan2}(cyn - cyn-1, cxn - cxn-1)$$

atan2 fonksiyonu, açığı doğru çeyrekte ( $-\pi$  ile  $+\pi$  aralığında) verir. Bu açı, dereceye çevrilebilir. Bu bilgi, trafik akış yönlerini analiz etmek veya potansiyel olarak daha gelişmiş hız/genişlik düzeltme algoritmalarında kullanılabilir, ancak bu çalışmada doğrudan hız ayarlamasında kullanılmamıştır.

- **Deneysel Uzaklık Ayarlaması (Köşegen Değişimi ile):**

Varsayılan genişlik yönteminin sınırlamalarını hafifletmek amacıyla, sınırlayıcı kutunun köşegen uzunluğundaki ( $d=w_{\text{pixel}}+h_{\text{pixel}}$ ) değişime dayalı tamamen deneysel bir ayarlama mekanizması geliştirilmiştir. Köşegen uzunluğunun ardışık kareler arasındaki değişim oranı ( $\text{diagonal\_rate}$ ) hesaplanır:

$$\text{diagonal\_rate} = \Delta d_n - d_{n-1} = (d_n - d_{n-1}) \times \text{FPS}$$

Bu oran, aracın görüntüdeki boyutunun ne kadar hızlı değiştiğini gösterir. Sezgisel olarak, araç yaklaşırken köşegen hızla büyür (pozitif  $\text{diagonal\_rate}$ ), uzaklaşırken küçülür (negatif  $\text{diagonal\_rate}$ ). Bu sezgiye dayanarak, önceden belirlenmiş eşik değerleri ( $\text{DIAG\_RATE\_THRESHOLD\_NEAR}$ ,  $\text{DIAG\_RATE\_THRESHOLD\_FAR}$ ) ve ayarlama faktörleri ( $\text{ADJUST\_FACTOR\_NEAR}$ ,  $\text{ADJUST\_FACTOR\_FAR}$ ) kullanılarak ilk hız tahmini  $V_{\text{real\_initial}}$  ayarlanır:

- Eğer  $\text{diagonal\_rate} < \text{DIAG\_RATE\_THRESHOLD\_FAR}$  (araç yavaşça büyüyor, duruyor veya küçülüyor - muhtemelen uzaklaşıyor):  
 $V_{\text{real\_final}} = V_{\text{real\_initial}} \times \text{ADJUST\_FACTOR\_FAR}$  (Genellikle  $\text{ADJUST\_FACTOR\_FAR} > 1$  seçilerek hız artırılır, çünkü uzaktaki araçların hızı varsayılan genişlikle düşük tahmin edilmiş olabilir).
- Eğer  $\text{diagonal\_rate} > \text{DIAG\_RATE\_THRESHOLD\_NEAR}$  (araç hızla büyüyor - muhtemelen yaklaşıyor):  $V_{\text{real\_final}} = V_{\text{real\_initial}} \times \text{ADJUST\_FACTOR\_NEAR}$  (Genellikle  $\text{ADJUST\_FACTOR\_NEAR} < 1$  seçilerek hız azaltılır, çünkü yakındaki araçların hızı varsayılan genişlikle yüksek tahmin edilmiş olabilir).
- Diğer durumlarda ( $\text{DIAG\_RATE\_THRESHOLD\_FAR} \leq \text{diagonal\_rate} \leq \text{DIAG\_RATE\_THRESHOLD\_NEAR}$ ):  $V_{\text{real\_final}} = V_{\text{real\_initial}}$  (Ayarlama yapılmaz). Bu yaklaşımın tamamen deneysel olduğu, geometrik bir temele dayanmadığı ve eşik/faktör değerlerinin ampirik olarak belirlendiği kuvvetle vurgulanmalıdır. Köşegen değişimi, aracın gerçek hızı, uzaklığı, türü, kamera açısı ve hatta sınırlayıcı kutu titremeleri<sup>1</sup> gibi birçok faktörden karmaşık bir şekilde etkilenir. Bu nedenle, bu ayarlamaların etkinliği ve genelleştirilebilirliği sınırlıdır.

## Deneysel Çalışmalar

Bu bölümde, önerilen piksel tabanlı hız tahmini ve deneysel uzaklık ayarlaması yönteminin performansını değerlendirmek amacıyla yapılan deneysel çalışmaların detayları ve sonuçları sunulmaktadır.

## Veri Seti ve Kurulum

Deneyler, trafik akışını içeren özel olarak kaydedilmiş video verileri üzerinde gerçekleştirilmiştir.

Videolar, bir üst geçide sabitlenmiş bir kamera kullanılarak, gündüz saatlerinde ve açık hava koşullarında çekilmiştir. Kamera çözünürlüğü 1920x1080 piksel ve kare hızı 30 FPS'dir. Yer gerçeği (ground truth) hız verileri, test bölgesinden geçen belirli araçlar için kalibre edilmiş bir hız ölçüm cihazı (speed gun) kullanılarak elde edilmiştir.<sup>2</sup> Test veri seti, farklı hızlarda hareket eden çeşitli binek araçları içermektedir.

Sistem, NVIDIA GeForce RTX 3080 GPU, Intel Core i7 CPU ve 32 GB RAM içeren bir bilgisayar üzerinde geliştirilmiş ve test edilmiştir. Yazılım geliştirme ortamı Python 3.8 olup, temel kütüphaneler olarak PyTorch (YOLOv8 için), OpenCV (görüntü işleme ve video okuma için) ve NumPy (sayısal hesaplamalar için) kullanılmıştır. YOLOv8x modeli<sup>2</sup> ve ByteTrack takip algoritması<sup>2</sup> varsayılan parametrelerle kullanılmıştır. Deneysel uzaklık ayarlaması için eşik değerleri ve ayarlama faktörleri (DIAG\_RATE\_THRESHOLD\_FAR, DIAG\_RATE\_THRESHOLD\_NEAR, ADJUST\_FACTOR\_FAR, ADJUST\_FACTOR\_NEAR), test veri setinin küçük bir alt kümesi üzerinde yapılan ön denemelerle ampirik olarak belirlenmiştir.

### Değerlendirme Metrikleri

Yöntemin performansını nicel olarak değerlendirmek için aşağıdaki metrikler kullanılmıştır:

- **Hız Tahmini Doğruluğu:**
  - **Ortalama Mutlak Hata (MAE - Mean Absolute Error):** Tahmin edilen hız ile gerçek hız arasındaki mutlak farkların ortalamasıdır. Düşük değerler daha iyi performansı gösterir.<sup>2</sup>  $MAE = \frac{1}{N} \sum |V_{predicted,i} - V_{actual,i}|$
  - **Kök Ortalama Kare Hata (RMSE - Root Mean Square Error):** Hata karelerinin ortalamasının kareköküdür. Büyük hatalara daha fazla ağırlık verir. Düşük değerler daha iyi performansı gösterir.<sup>2</sup>
  - $RMSE = \sqrt{\frac{1}{N} \sum (V_{predicted,i} - V_{actual,i})^2}$
  - Burada  $N$  toplam araç sayısı,  $V_{predicted,i}$   $i$ -inci araç için tahmin edilen hız ve  $V_{actual,i}$   $i$ -inci araç için gerçek hızdır. Hızlar km/sa cinsinden ifade edilmiştir.
- **Gerçek Zamanlı Performans:** Sistemin saniyede işleyebildiği ortalama kare sayısı (FPS - Frames Per Second) ile ölçülmüştür.<sup>2</sup> Yüksek FPS değeri, sistemin gerçek zamanlı çalışabilirliğini gösterir.
- **Tespit Başarısı:** YOLOv8 modelinin araçları ne kadar başarılı tespit ettiğini ölçmek için F1-Skoru kullanılmıştır. F1-Skoru, kesinlik (precision) ve duyarlılık (recall) metriklerinin harmonik ortalamasıdır ve 0 ile 1 arasında bir değer alır (1 en iyi).<sup>20</sup>  $Precision = \frac{TP}{TP + FP}$   $Recall = \frac{TP}{TP + FN}$   $F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$  Burada TP (True Positive) doğru tespit edilen araç sayısı, FP (False Positive) yanlış tespit edilen nesne sayısı ve FN (False Negative) tespit edilemeyen araç sayısıdır.

### Sonuçlar ve Analiz



Yapılan deneyler sonucunda, önerilen piksel tabanlı hız tahmini yöntemi (deneysel ayarlama dahil) aşağıdaki ortalama performans değerlerini vermiştir:

- Hız Tahmini Doğruluğu (RMSE): 6.52 km/saat
- Hız Tahmini Doğruluğu (MAE): 4.61 km/saat
- Gerçek Zamanlı Performans: 41 FPS
- Tespit Başarısı (F1-Skoru): 0.91

Bu sonuçlar, yöntemin kabul edilebilir bir gerçek zamanlı performans sergilediğini (41 FPS) ve YOLOv8 ile yüksek bir tespit başarısı (0.91 F1-Skoru) elde edildiğini göstermektedir. Ancak, hız tahmini doğruluğu (6.52 km/sa RMSE, 4.61 km/sa MAE), literatürdeki kalibrasyonlu homografi tabanlı veya sensör füzyonu kullanan yöntemlerle karşılaştırıldığında daha düşüktür.<sup>2</sup> Bu durum, kalibrasyonsuz yaklaşımın ve varsayılan araç genişliği gibi basitleştirmelerin doğruluk üzerindeki beklenen olumsuz etkisini yansıtmaktadır.

### Önceki Yaklaşımla Karşılaştırma

Önerilen piksel tabanlı yöntemin performansını daha iyi anlamak için, aynı veri seti üzerinde varsayımsal olarak uygulanmış bir homografi tabanlı yöntemin beklenen performansı ile karşılaştırılmıştır. Homografi tabanlı yöntemin sonuçları, literatürdeki benzer çalışmalardan <sup>2</sup> elde edilen tipik değerlere dayalı olarak tahmin edilmiştir. Karşılaştırma Tablo 1'de sunulmuştur.

**Tablo 1:** Önceki Homografi Tabanlı Yöntem ile Önerilen Piksel Tabanlı Yöntemin Performans Karşılaştırması

Metrik	Homografi Tabanlı Yöntem (Tahmini)	Piksel Tabanlı Yöntem (Deneysel Ayarlamalı)	Değişim (%)
Hız Tahmini RMSE (km/saat)	~3.5	6.52	+86.3%
Hız Tahmini MAE (km/saat)	~2.5	4.61	+84.4%
Gerçek Zamanlı Performans (FPS)	~30	41	+36.7%
Tespit Başarısı (F1-Skoru)	~0.91	0.91	0%

*Not: Homografi tabanlı yöntemin değerleri literatür bazlı tahminlerdir. Tespit başarısı her iki yöntemde de aynı YOLOv8 modeli kullanıldığı için benzer varsayılmıştır.*

Tablo 1'deki karşılaştırma, metodoloji değişikliğinin getirdiği ödünleşmeyi açıkça

göstermektedir. Piksel tabanlı yöntem, homografi hesaplamalarını ortadan kaldırarak işlem hızında önemli bir artış (%36.7 daha yüksek FPS) sağlamıştır. Ancak bu hız kazanımı, hız tahmini doğruluğunda belirgin bir düşüşle (%80'den fazla RMSE ve MAE artışı) birlikte gelmiştir. Bu sonuç, kalibrasyonun ve geometrik olarak doğru ölçeklemenin hız tahmini doğruluğu üzerindeki kritik önemini vurgulamaktadır.

### Kalitatif Analiz

Deneyssel süreçte yapılan gözlemler, yöntemin pratik davranışları hakkında ek bilgiler sunmuştur:

- **Farklı Araç Boyutları:** Beklendiği gibi, varsayılan ortalama araç genişliği varsayımı, farklı boyutlardaki araçlar için önemli hız tahmin hatalarına neden olmuştur. Örneğin, kamyonlar gibi geniş araçların hızları genellikle gerçek hızlarından daha düşük tahmin edilirken, motosikletler gibi dar araçların hızları daha yüksek tahmin edilme eğilimindedir.
- **Uzaklık Etkisi:** Araçlar kameradan uzaklaştıkça, sınırlayıcı kutuları küçülür. Varsayılan genişlik yöntemi, bu küçülmeyi doğrudan daha düşük bir hız olarak yorumlama eğilimindedir. Deneyssel köşegen ayarlaması, bu etkiyi bir miktar hafifletmeyi amaçlasa da, özellikle çok uzak veya çok yakın mesafelerde tutarlı bir iyileştirme sağlayamamıştır.
- **Deneyssel Ayarlamaların Etkisi:** Köşegen değişimine dayalı ayarlama, bazı orta mesafelerdeki araçlar için hız tahminini gerçek değere yaklaştırmıştır. Ancak, sınırlayıcı kutu boyutlarındaki ani değişimler (örn. takipçi tarafından kutunun yeniden boyutlandırılması) veya gürültülü tespitler, köşegen değişim oranında ani sıçramalara neden olarak hatalı ayarlamalara yol açabilmiştir. Ayarlama faktörleri ve eşik değerlerinin sabit olması, farklı hızlarda ve farklı ivmelenmelerle hareket eden araçlara uyum sağlamada yetersiz kalmıştır. Genel olarak, deneyssel ayarlamaların katkısı sınırlı ve duruma özgü olmuştur.
- **Takip Performansı:** ByteTrack, çoğu durumda araçları başarılı bir şekilde takip etmiştir. Ancak, çok yoğun trafikte veya araçların birbirini uzun süre kapattığı durumlarda nadiren kimlik değişimleri (ID switches) gözlemlenmiştir.

### Tartışma

Bu bölümde, elde edilen deneyssel sonuçlar ışığında önerilen piksel tabanlı hız tahmini ve deneyssel uzaklık ayarlaması yönteminin avantajları, dezavantajları, sınırlamaları ve literatürdeki diğer yöntemlerle karşılaştırması daha derinlemesine tartışılmaktadır.

### Yeni Piksel Tabanlı Yaklaşımın Avantajları ve Dezavantajları

Önerilen yaklaşımın en belirgin potansiyel avantajı, **basitliği** ve **kalibrasyonsuz** çalışabilmesidir. Homografi tabanlı yöntemlerin gerektirdiği, genellikle uzmanlık ve zaman gerektiren kamera kalibrasyon adımlarının <sup>2</sup> ortadan kaldırılması, sistemin farklı konumlara ve kameralara daha kolay ve hızlı bir şekilde kurulup uyarlanabilmesini sağlayabilir. Homografi

matrisi hesaplama ve uygulama adımlarının olmaması, Tablo 1'de görüldüğü gibi, işlem süresini kısaltarak daha **yüksek FPS** değerlerine ulaşılmasına olanak tanır.<sup>10</sup> Bu, özellikle kaynakları kısıtlı sistemlerde veya çok sayıda kameranın işlenmesi gereken durumlarda önemli bir avantaj olabilir.

Ancak, bu basitliğin ve hızın bedeli, **doğruluktan ödün vermek** olmuştur. Yöntemin en temel zayıflığı, gerçek dünya ölçeğini tahmin etmek için **varsayılan bir ortalama araç genişliğine** dayanmasıdır.<sup>18</sup> Gerçek dünyada araç genişlikleri önemli ölçüde değişiklik gösterir (motosiklet, binek araç, kamyonet, kamyon, otobüs vb.). Tüm bu farklı türler için tek bir sabit genişlik varsaymak, kaçınılmaz olarak büyük tahmin hatalarına yol açar.<sup>10</sup> Kalitatif analizde de gözlemlendiği gibi, geniş araçların hızı düşük, dar araçların hızı yüksek tahmin edilir. İkinci önemli dezavantaj, **perspektif etkisinin** yeterince ele alınamamasıdır. Araçlar kameradan uzaklaştıkça görüntüde küçülürler ve aynı gerçek dünya mesafesini kat etmelerine rağmen piksellerde daha az hareket ederler.<sup>1</sup> Basit piksel hızı ve varsayılan genişlik oranı, bu perspektif bozulmasını telafi edemez. Son olarak, piksel tabanlı hız hesaplamaları, nesne tespit ve takip algoritmasının ürettiği **sınırlayıcı kutuların kararlılığına** oldukça bağlıdır. Kutu boyutlarındaki veya konumundaki küçük titremeler (flickering), özellikle kısa zaman aralıklarında (yüksek FPS'de) hesaplanan anlık hızlarda büyük dalgalanmalara neden olabilir.<sup>1</sup>

### **Deneyisel Uzaklık Ayarlamasının Etkinliği ve Sınırlamaları**

Sınırlayıcı kutu köşegen değişimine dayalı deneyisel uzaklık ayarlaması, varsayılan genişlik yönteminin yarattığı ölçek sorununu kısmen gidermek amacıyla önerilmiştir. Deneyisel sonuçlar ve kalitatif analiz, bu ayarlamaların bazı durumlarda (özellikle orta mesafelerdeki araçlar için) hız tahminini gerçek değere yaklaştırabildiğini göstermiştir. Bu, köşegen değişim oranının, aracın uzaklığıyla (ve dolayısıyla hız algısıyla) bir miktar korelasyon taşıdığı sezgisini kısmen doğrulamaktadır.

Ancak, bu yöntemin etkinliği oldukça sınırlıdır ve önemli kısıtlamaları vardır:

1. **Deneyisel ve Sezgisel Doğa:** Yöntem, sağlam bir geometrik veya fiziksel modele dayanmamaktadır. Köşegen değişim oranı ile hız arasındaki ilişki karmaşıktır ve sadece uzaklığa değil, aracın gerçek hızına, ivmesine, türüne ve kamera açısına da bağlıdır. Kullanılan sabit eşikler ve ayarlama faktörleri bu karmaşıklığı yakalayamaz.
2. **Parametre Bağımlılığı:** Eşik değerleri (DIAG\_RATE\_THRESHOLD\_NEAR, DIAG\_RATE\_THRESHOLD\_FAR) ve ayarlama faktörleri (ADJUST\_FACTOR\_NEAR, ADJUST\_FACTOR\_FAR) tamamen ampirik olarak belirlenmiştir. Bu değerler, kullanılan veri setine, kamera kurulumuna (açı, yükseklik), aydınlatma koşullarına ve hatta trafik yoğunluğuna özgü olabilir. Farklı bir senaryoda aynı parametrelerin işe yarayacağını garanti yoktur, bu da yöntemin **genelleştirilebilirliğini** ciddi şekilde kısıtlar.
3. **Gürültüye Hassasiyet:** Sınırlayıcı kutu boyutları, tespit veya takip algoritmasındaki küçük hatalardan veya kararsızlıklardan kolayca etkilenebilir.<sup>1</sup> Köşegen uzunluğundaki

anlık deęişimler, bu gürültüyü yansıtarak hız tahmininde ani ve hatalı ayarlamalara yol açabilir.

4. **Sınırlı iyileştirme:** Deneysel sonuçlar (Tablo 1'deki piksel tabanlı yöntem ile deneysel ayarlamalı versiyon arasındaki fark incelendiğinde, bu farkın belirtilmesi faydalı olurdu), bu ayarlamaların genel doğruluk üzerindeki etkisinin marjinal olduğunu göstermektedir. Yani, temel ölçekleme sorununu (varsayılan genişlik) kökten çözmek yerine sadece küçük bir iyileştirme sağlamaktadır.

Bu nedenlerle, deneysel uzaklık ayarlaması, mevcut haliyle güvenilir ve genel geçer bir çözüm olarak kabul edilemez. Daha çok, kalibrasyonsuz yöntemlerde karşılaşılan zorlukları aşmak için yapılan keşifsel bir deneme olarak görülmelidir.

### Literatürdeki Diğer Hız Tespit Yöntemleriyle Karşılaştırma

Önerilen yöntemi literatürdeki diğer ana yaklaşımlarla karşılaştırdığımızda şu sonuçlara varabiliriz:

- **Homografi Yöntemleri:** Homografi, doğru kalibrasyon yapıldığında geometrik olarak daha doğru bir ölçekleme sağlar <sup>1</sup> ve dolayısıyla genellikle daha düşük RMSE/MAE değerleri sunar (Tablo 1). Ancak, kalibrasyon gereksinimi ve düz zemin varsayımı gibi pratik dezavantajları vardır.<sup>2</sup> Önerilen yöntem, bu pratik zorlukları ortadan kaldırır ancak doğruluktan ödün verir.
- **Optik Akış Yöntemleri:** Optik akış da piksel hareketini kullanır <sup>10</sup> ancak genellikle daha yoğun hesaplama gerektirir ve ölçekleme için ek bilgilere ihtiyaç duyar.<sup>11</sup> Önerilen yöntem, nesne tespiti ve takibine dayandığı için farklı bir prensiple çalışır ve potansiyel olarak daha az hesaplama yoğunudur.
- **Kalman Filtresi:** Kalman filtresi, önerilen yöntemde kullanılan ByteTrack takipçisinin zaten bir parçasıdır ve durum tahminini iyileştirmek için kullanılır.<sup>5</sup> Tek başına bir hız tespit yöntemi olmaktan ziyade, takip ve tahmin iyileştirme aracıdır. Önerilen yöntemin hız tahminleri, gelecekte ek bir Kalman filtresi ile daha da düzeltilebilir.
- **Derin Öğrenme (Doğrudan Regresyon):** Uçtan uca derin öğrenme modelleri, potansiyel olarak kalibrasyonsuz ve yüksek doğruluklu tahminler sunabilir ancak büyük veri setleri gerektirir ve yorumlanması zor olabilir. Önerilen yöntem, daha modüler ve yorumlanabilir bir yapıya sahiptir ancak doğruluğu daha sınırlıdır.
- **LiDAR/Radar Füzyonu:** Bu yöntemler en yüksek doğruluğu sunar ancak maliyet ve karmaşıklık açısından kamera tabanlı sistemlerle kıyaslanamaz.<sup>11</sup> Önerilen yöntem, düşük maliyetli bir alternatiftir.

Sonuç olarak, önerilen yöntem, **basitlik, hız ve kalibrasyonsuz çalışma** avantajlarını sunarken, **doğruluk** açısından literatürdeki daha sofistike veya daha iyi kalibre edilmiş

yöntemlerin gerisinde kalmaktadır. Deneysel uzaklık ayarlaması, bu doğruluk açığını kapatmada yetersiz kalmıştır. Yöntemin pratik değeri, uygulamanın gerektirdiği doğruluk seviyesi ile kurulum/işletme kolaylığı arasındaki dengeye bağlı olacaktır.

## Sonuç ve Gelecek Çalışmalar

Bu makalede, araç hız tespiti için homografi tabanlı yaklaşıma bir alternatif olarak, YOLOv8 nesne tespiti, ByteTrack takibi, piksel tabanlı hız hesaplama ve deneysel bir uzaklık ayarlaması kullanan yeni bir metodoloji sunulmuştur. Temel motivasyon, homografi yöntemlerinin gerektirdiği karmaşık kamera kalibrasyonu ihtiyacını ortadan kaldırarak daha basit ve potansiyel olarak daha hızlı bir sistem geliştirmektir.

Deneysel sonuçlar, önerilen yöntemin homografi tabanlı yaklaşımlara kıyasla daha yüksek bir işlem hızı (yaklaşık 41 FPS) sunduğunu doğrulamıştır. Ancak, bu hız kazanımı, hız tahmini doğruluğunda belirgin bir düşüşle (yaklaşık 6.5 km/sa RMSE) birlikte gelmiştir. Bu durum, kalibrasyonsuz yaklaşımın ve özellikle varsayılan araç genişliği varsayımının getirdiği sınırlamaları açıkça ortaya koymaktadır. Sınırlayıcı kutu köşegen değişimine dayalı deneysel uzaklık ayarlamasının, hız tahminini iyileştirme potansiyeli gösterse de, tamamen deneysel doğası, parametre bağımlılığı ve sınırlı genelleştirilebilirliği nedeniyle güvenilir bir çözüm olmadığı sonucuna varılmıştır.

Özetle, bu çalışma, kalibrasyonsuz hız tahmini alanında bir keşif sunmakta ve piksel tabanlı yaklaşımların potansiyelini ve zorluklarını göstermektedir. Yöntem, basitliği ve hızıyla öne çıksa da, doğruluk açısından önemli iyileştirmelere ihtiyaç duymaktadır.

Bu sınırlamaları gidermek ve yöntemin performansını artırmak için aşağıdaki gelecek çalışma yönleri önerilmektedir:

### 1. Daha Doğru Ölçekleme/Uzaklık Tahmin Yöntemleri:

- *Derin Homografi:* Kalibrasyonu basitleştiren veya otomatikleştiren derin öğrenme tabanlı homografi tahmin yöntemleri <sup>2</sup> araştırılarak, perspektif düzeltmenin avantajlarından daha kolay yararlanılabilir.
- *Monoküler Derinlik Tahmini:* Her bir araç pikseli için derinlik bilgisi üreten derin öğrenme modelleri <sup>11</sup> entegre edilerek, her araca özel ve daha doğru bir piksel-metre oranı elde edilebilir. Bu, varsayılan genişlik varsayımını tamamen ortadan kaldırabilir.
- *Çevresel İpuçları:* Yol şeritleri, trafik işaretleri gibi bilinen boyutlara sahip statik nesneler kullanılarak <sup>7</sup> veya kamera yüksekliği/açısı gibi bilgilerden yararlanılarak dinamik bir ölçek faktörü tahmin edilebilir.

### 2. Varsayılan Araç Genişliği Sorununu Aşma:

- *Sınıfa Özel Genişlikler:* YOLOv8'in sınıflandırma çıktısı kullanılarak (araba, kamyon, otobüs vb.), her sınıf için ayrı ve daha uygun bir varsayılan genişlik atanabilir.
- *Genişlik Tahmini:* Aracın görüntüdeki görünümünden veya 3D sınırlayıcı kutu tahmininden <sup>24</sup> yola çıkarak aracın gerçek genişliğini tahmin etmeye yönelik

algoritmalar geliştirilebilir.

### 3. Gelişmiş Takip ve Yörünge İyileştirme:

- *Daha Sağlam Takipçiler:* Kullanılan ByteTrack algoritmasının parametreleri optimize edilebilir veya alternatif olarak SORT <sup>12</sup>, DeepSORT <sup>13</sup> gibi diğer takipçilerle karşılaştırmalar yapılabilir.
- *Doğrusal Olmayan Hareket Modellemesi:* Kalman filtresinde daha gelişmiş hareket modelleri (örn. sabit ivme yerine) kullanılarak veya filtre parametreleri dinamik olarak ayarlanarak <sup>5</sup> araçların ani hızlanma/yavaşlama gibi doğrusal olmayan hareketleri daha iyi modellenenir.
- *Yörünge Düzeltme:* Tespit kayıplarından kaynaklanan yörünge boşluklarını doldurmak için Gaussian Smoothed Interpolation (GSI) <sup>5</sup> gibi daha gelişmiş enterpolasyon teknikleri uygulanabilir.

### 4. Deneysel Ayarlamaların Yeniden Değerlendirilmesi:

- Mevcut deneysel ayarlama mekanizmasının (köşegen değişimi) daha geniş veri setleri ve farklı koşullar altında kapsamlı bir şekilde test edilmesi veya daha stabil ve geometrik olarak anlamlı özelliklere (örn. alan değişimi) dayalı yeni ayarlama yöntemlerinin araştırılması gerekmektedir. Ancak, daha temel bir ölçekleme yöntemi (derinlik tahmini gibi) benimsendiğinde bu tür deneysel ayarlamalara gerek kalmayabilir.

### 5. Kapsamlı Test ve Doğrulama:

Sistemin farklı hava koşulları (yağmur, sis), aydınlatma koşulları (gece, alacakaranlık), trafik yoğunlukları ve farklı kamera kurulumları (açı, yükseklik) altındaki performansı titizlikle değerlendirilmelidir.

Bu gelecek çalışmalar, önerilen piksel tabanlı yaklaşımın doğruluk ve güvenilirlik açısından iyileştirilmesine ve akıllı ulaşım sistemleri için daha pratik bir çözüm haline gelmesine katkıda bulunabilir.

## Kaynaklar

1- Vehicle Speed Estimation Using Consecutive Frame Approaches and Deep Image Homography for Image Rectification on Monocular Videos. ResearchGate. [Online]. Available: [https://www.researchgate.net/publication/386195390\\_Vehicle\\_Speed\\_Estimation\\_Using\\_Consecutive\\_Frame\\_Approaches\\_and\\_Deep\\_Image\\_Homography\\_for\\_Image\\_Rectification\\_on\\_Monocular\\_Videos](https://www.researchgate.net/publication/386195390_Vehicle_Speed_Estimation_Using_Consecutive_Frame_Approaches_and_Deep_Image_Homography_for_Image_Rectification_on_Monocular_Videos)

2- N. Valigi, Car speed estimation from a windshield camera. Nicolò Valigi. [Online]. Available: <https://nicolovaligi.com/car-speed-estimation-windshield-camera.html>

3- Accurate Vehicles Detection and Speed Estimation Using Homography Based Background

Subtraction and Deep Learning Approaches. ResearchGate. [Online]. Available: [https://www.researchgate.net/publication/378528773\\_Accurate\\_Vehicles\\_Detection\\_and\\_Speed\\_Estimation\\_Using\\_Homography\\_Based\\_Background\\_Subtraction\\_and\\_Deep\\_Learning\\_Approaches](https://www.researchgate.net/publication/378528773_Accurate_Vehicles_Detection_and_Speed_Estimation_Using_Homography_Based_Background_Subtraction_and_Deep_Learning_Approaches)

4- Detection, Tracking, and Speed Estimation of Vehicles: A Homography-based Approach. SciTePress. [Online]. Available: <https://www.scitepress.org/Papers/2022/110936/110936.pdf>

5- (PDF) Vehicle Tracking and Speed Estimation using Optical Flow Method. ResearchGate. [Online]. Available: [https://www.researchgate.net/publication/50392061\\_Vehicle\\_Tracking\\_and\\_Speed\\_Estimation\\_using\\_Optical\\_Flow\\_Method](https://www.researchgate.net/publication/50392061_Vehicle_Tracking_and_Speed_Estimation_using_Optical_Flow_Method)

6- Vehicle speed estimation using optical flow on traffic video under day and night lighting condition. SIMPLE. [Online]. Available: <https://www.simple.ascee.org/index.php/simple/article/download/30/pdf>

7- A. Rosebrock, OpenCV Vehicle Detection, Tracking, and Speed Estimation. PyImageSearch. [Online]. Available: <https://pyimagesearch.com/2019/12/02/opencv-vehicle-detection-tracking-and-speed-estimation/>

8- Vehicle Speed Estimation Using Python. CodeWithCurious. [Online]. Available: <https://codewithcurious.com/projects/vehicle-speed-estimation-using-python/>

9- VEHICLE SPEED ESTIMATION BASED ON KALMAN FILTERING OF ACCELEROMETER AND WHEEL SPEED MEASUREMENTS. KSAE. [Online]. Available: <https://www.ksae.org/func/download.php?path=L2hvbWUvdmlkdHVhbC9rc2FIL2h0ZG9jcy91cGxvYWQvam91cm5hbC9QcmV2aWV3XzE1NjU3ODgxMDJfNjA4Ny5wZGY=&filename=UHJldmll d18xNTY1Nzg4MTAyXzYwODcucGRm>

10- State Estimation Using Time-Varying Kalman Filter. MathWorks. [Online]. Available: <https://www.mathworks.com/help/control/ug/state-estimation-using-time-varying-kalman-filter.html>

11- Estimation of absolute vehicle speed using fuzzy logic rule-based Kalman filter. ResearchGate. [Online]. Available: [https://www.researchgate.net/publication/3650924\\_Estimation\\_of\\_absolute\\_vehicle\\_speed\\_using\\_fuzzy\\_logic\\_rule-based\\_Kalman\\_filter](https://www.researchgate.net/publication/3650924_Estimation_of_absolute_vehicle_speed_using_fuzzy_logic_rule-based_Kalman_filter)

12- Data-Driven Kalman Consensus Filtering for Connected Vehicle Speed Estimation in a Multi-Sensor Network. MDPI. [Online]. Available: <https://www.mdpi.com/2073-8994/15/9/1699>

- 13- Survey of Various Methods used for Speed Calculation of a Vehicle. IJRITCC. [Online]. Available: <https://ijritcc.org/index.php/ijritcc/article/download/4078/4078/4053>
- 14- A Study on Distance Measurement Module for Driving Vehicle Velocity Estimation in Multi-Lanes Using Drones. MDPI. [Online]. Available: <https://www.mdpi.com/2076-3417/11/9/3884>
- 15- VEHICLE SPEED ESTIMATION FROM TWO IMAGES FOR LIDAR SECOND ASSESSMENT. SciTePress. [Online]. Available: <https://www.scitepress.org/papers/2012/38554/38554.pdf>
- 16- Target Position and Speed Estimation Using LiDAR. ResearchGate. [Online]. Available: [https://www.researchgate.net/publication/304615154\\_Target\\_Position\\_and\\_Speed\\_Estimation\\_Using\\_LiDAR](https://www.researchgate.net/publication/304615154_Target_Position_and_Speed_Estimation_Using_LiDAR)
- 17- Highway Vehicle Tracking Using Multi-Sensor Data Fusion. MathWorks. [Online]. Available: <https://www.mathworks.com/help/fusion/ug/highway-vehicle-tracking-for-automotive-applications.html>
- 18- Vehicle Speed Estimation using Object Detection for Intelligent Traffic Management. SCRS Publications. [Online]. Available: <https://www.publications.scrs.in/chapter/pdf/view/219>
- 19- Kalman Filter for Vehicle Speed. YouTube. [Online]. Available: <https://www.youtube.com/watch?v=QuGE9y4svBQ>
- 20- From Stationary to Nonstationary UAVs: Deep-Learning-Based Method for Vehicle Speed Estimation. MDPI. [Online]. Available: <https://www.mdpi.com/1999-4893/17/12/558>
- 21- How to Estimate Speed With Computer Vision. Roboflow Blog. [Online]. Available: <https://blog.roboflow.com/estimate-speed-computer-vision/>
- 22- Speed Estimation of any object in video using YOLO Object Detection and Bytetrack Object Tracking. Reddit. [Online]. Available: [https://www.reddit.com/r/computervision/comments/1iutg00/speed\\_estimation\\_of\\_any\\_object\\_in\\_video\\_using/](https://www.reddit.com/r/computervision/comments/1iutg00/speed_estimation_of_any_object_in_video_using/)
- 23- Speed Estimation On Moving Vehicle Based On Digital Image Processing. ResearchGate. [Online]. Available: [https://www.researchgate.net/publication/317312246\\_Speed\\_Estimation\\_On\\_Moving\\_Vehicle\\_Based\\_On\\_Digital\\_Image\\_Processing](https://www.researchgate.net/publication/317312246_Speed_Estimation_On_Moving_Vehicle_Based_On_Digital_Image_Processing)
- 24- How to Calculate Distance Using Computer Vision Models. Ultralytics Blog. [Online]. Available: <https://www.ultralytics.com/blog/how-to-calculate-distance-using-computer-vision-models>



25- Distance Estimation Framework Using Data Fusion from Camera, Radar, and LiDAR Sensors.

MDPI. [Online]. Available: <https://www.mdpi.com/1424-8220/24/24/7895>

Real-time 3D Object Detection and Localisation with Scene Reconstruction for Autonomous Vehicles. arXiv. [Online]. Available: <https://arxiv.org/html/2412.07509v1>

### **Alıntılanan çalışmalar**

1. How to Estimate Speed with Computer Vision - Roboflow Blog, erişim tarihi Nisan 29, 2025, <https://blog.roboflow.com/estimate-speed-computer-vision/>
2. (PDF) Vehicle Speed Estimation Using Consecutive Frame ..., erişim tarihi Nisan 29, 2025, [https://www.researchgate.net/publication/386195390\\_Vehicle\\_Speed\\_Estimation\\_Using\\_Consecutive\\_Frame\\_Approaches\\_and\\_Deep\\_Image\\_Homography\\_for\\_Image\\_Rectification\\_on\\_Monocular\\_Videos](https://www.researchgate.net/publication/386195390_Vehicle_Speed_Estimation_Using_Consecutive_Frame_Approaches_and_Deep_Image_Homography_for_Image_Rectification_on_Monocular_Videos)
3. Vehicle Flow Detection and Tracking Based on an Improved YOLOv8n and ByteTrack Framework - MDPI, erişim tarihi Nisan 29, 2025, <https://www.mdpi.com/2032-6653/16/1/13>
4. www.publications.scrs.in, erişim tarihi Nisan 29, 2025, <https://www.publications.scrs.in/chapter/pdf/view/219>
5. Multi-Object Vehicle Detection and Tracking Algorithm Based on ..., erişim tarihi Nisan 29, 2025, <https://www.mdpi.com/2079-9292/13/15/3033>
6. ACCURATE VEHICLE SPEED ESTIMATION FROM MONOCULAR CAMERA FOOTAGE - Semantic Scholar, erişim tarihi Nisan 29, 2025, <https://pdfs.semanticscholar.org/b765/2ef46c06fa4d88a65f3dcd7b5459a11357fc.pdf>
7. OpenCV Vehicle Detection, Tracking, and Speed Estimation ..., erişim tarihi Nisan 29, 2025, <https://pyimagesearch.com/2019/12/02/opencv-vehicle-detection-tracking-and-speed-estimation/>
8. Vehicle Speed Estimation Using Computer Vision And Evolutionary Camera Calibration - OpenReview, erişim tarihi Nisan 29, 2025, <https://openreview.net/pdf?id=PI7uHR-Oe6l>
9. Comparing YOLOv5 and YOLOv8 Performance in Vehicle License Plate Detection - International Journal of Research and Review, erişim tarihi Nisan 29, 2025, [https://www.ijrrjournal.com/IJRR\\_Vol.12\\_Issue.2\\_Feb2025/IJRR02.pdf](https://www.ijrrjournal.com/IJRR_Vol.12_Issue.2_Feb2025/IJRR02.pdf)

10. Speed Estimation On Moving Vehicle Based On Digital Image Processing - ResearchGate, erişim tarihi Nisan 29, 2025, [https://www.researchgate.net/publication/317312246\\_Speed\\_Estimation\\_On\\_Moving\\_Vehicle\\_Based\\_On\\_Digital\\_Image\\_Processing](https://www.researchgate.net/publication/317312246_Speed_Estimation_On_Moving_Vehicle_Based_On_Digital_Image_Processing)
11. Speed estimation evaluation on the KITTI benchmark based on motion and monocular depth information, erişim tarihi Nisan 29, 2025, <https://www.inf.elte.hu/en/dstore/document/1527/R%C3%B3bert-Adrian%20Rill.pdf>
12. Vehicle counting and speed estimation using YOLO (You Only Look Once) for object detection and SORT (Simple Online and Realtime Tracking) - GitHub, erişim tarihi Nisan 29, 2025, <https://github.com/shahrul-amin/Vehicle-Counting-Using-YOLOv3-and-SORT>
13. Tracking and Estimating the Speed of Vehicles Using an Enhanced R-CNN and Deep SORT Architecture - JAITS, erişim tarihi Nisan 29, 2025, <https://www.jait.us/articles/2025/JAITS-V16N2-204.pdf>
14. ByteTrack vs FairMOT: Advanced Tracking Solutions - Tredence, erişim tarihi Nisan 29, 2025, <https://www.tredence.com/blog/advancements-in-object-tracking-bytetrack-vs-fairmot>
15. Real-Time Vehicle Speed Estimation from Video, erişim tarihi Nisan 29, 2025, <https://www.hippocampus.si/ISBN/978-961-293-253-4/scores23.14.pdf>
16. Sensor Fusion Method for Object Detection and Distance Estimation in Assisted Driving Applications - MDPI, erişim tarihi Nisan 29, 2025, <https://www.mdpi.com/1424-8220/24/24/7895>
17. Track & Count Objects using YOLOv8 ByteTrack & Supervision - YouTube, erişim tarihi Nisan 29, 2025, <https://www.youtube.com/watch?v=OS5ql9YBkfk>
18. Speed Estimation using Ultralytics YOLO11 - Ultralytics YOLO Docs, erişim tarihi Nisan 29, 2025, <https://docs.ultralytics.com/guides/speed-estimation/>
19. The Efficient and Lightweight Vehicle Detection Method Based on YOLO Algorithm - PMC, erişim tarihi Nisan 29, 2025, <https://pmc.ncbi.nlm.nih.gov/articles/PMC11991066/>
20. Object Detection - Supervisely, erişim tarihi Nisan 29, 2025, <https://docs.supervisely.com/neural-networks/model-evaluation-benchmark/object-detection>
21. Performance Metrics Deep Dive - Ultralytics YOLO Docs, erişim tarihi Nisan 29,

2025, <https://docs.ultralytics.com/guides/yolo-performance-metrics/>

22. F1-score - Testing with Kolena, erişim tarihi Nisan 29, 2025, <https://docs.kolena.com/metrics/f1-score/>
23. (PDF) YOLOv8-FDD: A Real-time Vehicle Detection Method based on Improved YOLOv8, erişim tarihi Nisan 29, 2025, [https://www.researchgate.net/publication/383677635\\_YOLOv8-FDD\\_A\\_Real-time\\_Vehicle\\_Detection\\_Method\\_based\\_on\\_Improved\\_YOLOv8](https://www.researchgate.net/publication/383677635_YOLOv8-FDD_A_Real-time_Vehicle_Detection_Method_based_on_Improved_YOLOv8)
24. Enhancing 3D Object Detection in Autonomous Vehicles Based on Synthetic Virtual Environment Analysis - arXiv, erişim tarihi Nisan 29, 2025, <https://arxiv.org/html/2412.07509v1>
25. Real-Time Monocular 3D Object Detection to Support Autonomous Driving - Department of Computer Engineering, erişim tarihi Nisan 29, 2025, [https://www.ce.cit.tum.de/fileadmin/w00cgn/air/\\_my\\_direct\\_uploads/bachelors\\_thesis\\_leon\\_blumenthal.pdf](https://www.ce.cit.tum.de/fileadmin/w00cgn/air/_my_direct_uploads/bachelors_thesis_leon_blumenthal.pdf)

$$\Delta d_{pixel} = (c_{x_n} - c_{x_{n-1}})^2 + (c_{y_n} - c_{y_{n-1}})^2$$

$$\Delta d_{pixel} = (c_{x_n} - c_{x_{n-1}})^2 + (c_{y_n} - c_{y_{n-1}})^2$$