



**T.C.  
YALOVA ÜNİVERSİTESİ  
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ**

**BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI  
BİLGİSAYAR MÜHENDİSLİĞİ PROGRAMI**

**TRAFİKTEKİ ARAÇ KUYRUKLANMALARININ DERİN ÖĞRENME İLE  
GERÇEK ZAMANLI TESPİTİ**

**YÜKSEK LİSANS TEZİ**

**AHSEN BATTAL**

**DANIŞMAN: DOÇ. DR. ADEM TUNCER**

**YALOVA  
EKİM 2024**



T.C.  
YALOVA ÜNİVERSİTESİ  
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI  
BİLGİSAYAR MÜHENDİSLİĞİ PROGRAMI

TRAFİKTEKİ ARAÇ KUYRUKLANMALARININ DERİN ÖĞRENME İLE GERÇEK  
ZAMANLI TESPİTİ

YÜKSEK LİSANS TEZİ

AHSEN BATTAL  
218102004

DANIŞMAN: DOÇ. DR. ADEM TUNCER

YALOVA  
EKİM 2024

## ETİK BEYAN

Yalova Üniversitesi Lisansüstü Eğitim Enstitüsü Tez Yazım Kuralları'na uygun olarak hazırladığım “Trafikteki Araç Kuyruklanmalarının Derin Öğrenme ile Gerçek Zamanlı Tespiti” başlıklı bu tez çalışmada; tez içinde sunduğum verileri, bilgileri ve dokümanları akademik ve etik kurallar çerçevesinde elde ettiğimi, tüm bilgi, belge, değerlendirme ve sonuçları bilimsel etik ve ahlak kurallarına uygun olarak sunduğumu, tez çalışmada yararlandığım eserlerin tümüne uygun atıfta bulunarak kaynak gösterdiğimi, kullanılan verilerde herhangi bir değişiklik yapmadığımı, bu tezde sunduğum çalışmanın özgün olduğunu bildirir, aksinin tespiti halinde doğabilecek her türlü hukuki sorumluluğu kabul ettiğimi taahhüt ve beyan ederim.

Ahsen BATTAL



## ÖNSÖZ

Lisans ve yüksek lisans eğitimim boyunca benden yardımlarını esirgemeyen, akademik anlamda bana her zaman destek olan ve doğru yolu gösteren sayın danışmanım Doç. Dr. Adem TUNCER'e teşekkür ederim. Her zaman yanımda oldukları ve benden desteklerini hiçbir zaman esirgemedikleri için aileme, arkadaşlarıma ve çalışma arkadaşlarıma teşekkür ederim.

Ekim – 2024

Ahsen BATTAL



## İÇİNDEKİLER

ETİK BEYAN .....	i
ÖNSÖZ.....	ii
İÇİNDEKİLER.....	iii
KISALTMALAR LİSTESİ.....	iv
TABLolar LİSTESİ .....	v
ŞEKİLLER LİSTESİ.....	vi
ÖZET.....	vii
ABSTRACT .....	viii
1. GİRİŞ .....	1
1.1. Tezin Amacı .....	2
1.2. Literatür Taraması .....	2
2. DERİN ÖĞRENME.....	9
2.1. Evrişimsel Sinir Ağı .....	9
2.2. YOLO .....	11
2.3. Hiper Parametreler.....	13
2.3.1 Devir sayısı .....	13
2.3.2 Parti boyutu.....	13
2.3.3 Öğrenme oranı .....	14
2.4. SORT .....	14
3. MATERYAL VE METOT.....	15
3.1. Veri Seti.....	15
3.2. Verilerin Etiketlenmesi.....	16
3.3. Araç Sınıf Tespiti.....	18
3.4. Araç Sayımı .....	18
3.5. Hız Tespiti .....	19
3.6. Kuyruklanma Durum Tespiti.....	21
3.7. Değerlendirme Kriterleri .....	22
4. DENEYSEL ÇALIŞMALAR .....	25
5. SONUÇLAR VE ÖNERİLER .....	35
KAYNAKLAR.....	37
ÖZGEÇMİŞ .....	41

## KISALTMALAR LİSTESİ

### KISALTMALAR

AUS	: Akıllı Ulaşım Sistemleri
CCTV	: Closed-Circuit Television
CNN	: Evrişimsel Sinir Ağları (Convolutional Neural Network)
Conv	: Evrişimsel Katmanı
FN	: Yanlış Negatif (False Negative)
FrRCNN	: Daha Hızlı Bölge CNN (Faster Region CNN)
FP	: Yanlış Pozitif (False Positive)
IOU	: Intersection over Union
LSTM	: Uzun Kısa Vadeli Bellek (Long- Short Term Memory)
mAP	: Ortalama Hassasiyet (Mean Average Precision)
RoI	: İlgi Bölgesi (Region of Interest)
SPPF	: Uzamsal Piramit Havuzlama-Hızlı (Spatial Pyramid Pooling-Fast)
SORT	: Simple Online and Realtime Tracking
SSD	: Single Shot Multibox Detector
TN	: Doğru Negatif (True Negative)
TP	: Doğru Pozitif (True Positive)
YOLO	: You Only Look Once

## TABLULAR LİSTESİ

Tablo 3.1. Veri seti hakkında bilgi .....	15
Tablo 3.2. Veri setindeki sınıf etiketleri sayısı.....	15
Tablo 3.3. Kuyruklanma koşulları.....	22
Tablo 4.1. Veri setinin eğitim aşamasında kullanılan parametreler .....	25
Tablo 4.2. Veri setinin eğitim aşaması sonucunda elde edilen araç sınıf tespiti performans sonuçları .....	26
Tablo 4.3. Karmaşıklık matrisi sonucunda hesaplanan performans değerleri .....	28
Tablo 4.4. Araç sayım aşaması ile ilgili sonuçlar .....	29
Tablo 4.5. Hız tespiti ile ilgili sonuçlar .....	30



## ŞEKİLLER LİSTESİ

Şekil 2.1. Önerilen sistemin blok diyagramı .....	9
Şekil 2.2. Evrimsel sinir ağı mimarisi .....	10
Şekil 2.3. Maksimum havuzlama örneği .....	10
Şekil 2.4. Ortalama havuzlama örneği .....	10
Şekil 2.5. Düzleştirme örneği .....	11
Şekil 2.6. YOLO algoritmasında nesne için çizilen sınırlayıcı kutular .....	11
Şekil 2.7. YOLOv5 mimarisi [27] .....	12
Şekil 3.1. GitHub veri setinden örnekler .....	16
Şekil 3.2. Kaggle veri setinden örnekler .....	16
Şekil 3.3. Veri etiketleme diyagramı .....	16
Şekil 3.4. Veri etiketleme uygulaması .....	17
Şekil 3.5. Etiketleme sonucu oluşan txt dosyası .....	17
Şekil 3.6. Eğitim ve sınıflandırma diyagramı .....	18
Şekil 3.7. Araç sayım diyagramı .....	19
Şekil 3.8. Hız tespiti diyagramı .....	19
Şekil 3.9. Hız için gerekli olan bilgilerin gösterilmesi .....	20
Şekil 3.10. Kuyruklanma durum tespiti .....	21
Şekil 3.11. Karmaşıklık matrisi .....	22
Şekil 4.1. Araç veri setinin eğitimi sonucunda oluşan modelin kesinlik-duyarlılık grafiği .....	25
Şekil 4.2. Araç ve sınıf tespiti test aşamasından örnekler .....	26
Şekil 4.3. Araç sınıf tespitine ait karmaşıklık matrisleri .....	27
Şekil 4.4. Şerit bazlı araç sayımı .....	29
Şekil 4.5. Gerçekte 50 km/sa ile giden aracın tahmini hız tespiti .....	31
Şekil 4.6. Gerçekte 70 km/sa ile giden aracın tahmini hız tespiti .....	31
Şekil 4.7. Gerçekte 30 km/sa ile giden aracın tahmini hız tespiti .....	32
Şekil 4.8. Gerçekte 50 km/sa ile giden aracın tahmini hız tespiti .....	32
Şekil 4.9. “Kuyruklanma yok” durum tespiti .....	33
Şekil 4.10. “Hafif kuyruklanma”, “Kuyruklanma yok” ve “Kuyruklanma var” durum tespiti .....	34
Şekil 4.11. Az araç geçişi için “Kuyruklanma yok” durum tespiti .....	34

# TRAFİKTEKİ ARAÇ KUYRUKLANMALARININ DERİN ÖĞRENME İLE GERÇEK ZAMANLI TESPİTİ

## ÖZET

Günümüzde hızla artan araç sayısı ve trafik yoğunluğuna bağlı olarak, trafik yönetimi ve kuyruklanma tespiti önemli bir konu haline gelmiştir. Özellikle nüfus yoğunluğunun fazla olduğu şehirlerde trafik yoğunluğundaki artış, trafikteki insanların günlük yaşamında olumsuz etkiye sebep olmaktadır. Günlük yaşamı olumsuz etkilemesinin yanı sıra trafik yoğunluğu insanların gideceği yerlere gidiş-geliş sürelerinin uzamasına, araçların daha fazla yakıt tüketmesine ve bunun sonucunda enerji kaynaklarının hızlı tükenmesi gibi sorunlara da neden olmaktadır. Bu nedenle trafik yönetimi ve kuyruklanma tespiti, trafik yönetiminde önemli bir rol oynamaktadır. Trafikte oluşan olumsuz durumların daha kolay çözümlenebilmesi için son yıllarda yapay zeka destekli sistemler önerilmeye başlamıştır. Yapay zeka temelli bu sistemler ile trafik akışının izlenmesi, trafikteki yoğunluğun belirlenmesi ve bu yoğunluğa bağlı olarak kuyruklanmanın tespit edilip etkili bir şekilde yönetilmesi trafikteki olumsuz etkilerin azalmasına yardımcı olmaktadır.

Bu tez çalışmasında, trafik bileşenleri analizinin gerçekleştirilmesi amaçlanmıştır. Çalışma dört aşamadan oluşmaktadır. İlk aşamada görsel trafik veri setleri kullanılarak derin öğrenme algoritmalarından YOLO ile araç tespiti ve sınıflandırılma işlemi gerçekleştirilmektedir. Modeli otomobil, motosiklet, kamyon, otobüs ve bisiklet gibi beş farklı araç türünü tespit etmekte ve sınıflandırmaktadır. Tespit modelinin araç sınıf tahmininde ulaştığı kesinlik değeri %86, duyarlılık değeri %87 ve ortalama hassasiyet değeri %93 olarak hesaplanmıştır. İkinci aşamada, ilgili bölgeden geçen araçların SORT algoritması ile araç sayımı ve takibi gerçekleştirilmiştir. Üçüncü aşamada, sayımı yapılan araçların hızları tespit edilmiştir. Son aşamada ise, araçların tespit edilen hız bilgileri kullanılarak kuyruklanma durum tespiti gerçekleştirilmiştir. Kuyruklanma durum tespiti işleminde, araçların ortalama hızları kullanılmıştır. Araç sınıf tespiti, araç sayımı, araç hız tespiti ve kuyruklanma durum tespiti için hem kendi çekimimiz olan videolar hem de internet kaynaklarından elde edilen videolar kullanılarak kullanılan algoritmaların performansları değerlendirilmiştir.

**Anahtar Kelimeler:** Kuyruklanma, Trafikte araç tespiti, Derin öğrenme, YOLO, SORT

# REAL-TIME DETECTION OF VEHICLE QUEUES IN TRAFFIC WITH DEEP LEARNING

## ABSTRACT

Nowadays, due to the rapidly increasing number of vehicles and traffic density, traffic management and queuing detection have become an important issue. Especially in cities with high population density, the increase in traffic density has a negative impact on the daily life of people in traffic. In addition to adversely affecting daily life, traffic density causes problems such as prolonged journey times to and from destinations, increased fuel consumption of vehicles and consequently rapid depletion of energy resources. Therefore, traffic management and queuing detection play an important role in traffic management. In recent years, artificial intelligence supported systems have started to be proposed in order to solve the negative situations in traffic more easily. With these artificial intelligence-based systems, monitoring the traffic flow, determining the density in traffic and detecting and effectively managing queuing depending on this density helps to reduce the negative effects in traffic.

In this thesis, it is aimed at performing traffic component analysis. The study consists of four stages. In the first stage, vehicle detection and classification are performed with YOLO, one of the deep learning algorithms, using visual traffic data sets. The model detects and classifies five different vehicle types such as car, motorbike, truck, bus and bicycle. The precision value achieved by the detection model in vehicle class prediction was calculated as 86%, the recall value as 87% and the mean average precision value as 93%. In the second stage, vehicle counting and tracking of the vehicles passing through the relevant region was performed with the SORT algorithm. In the third stage, the speeds of the counted vehicles were determined. In the last stage, queuing detection was performed using the detected speed information of the vehicles. In the queuing detection process, the average speed of the vehicles was used. The performances of the algorithms used for vehicle class detection, vehicle counting, vehicle speed detection and queuing detection were evaluated using both our own videos and videos obtained from internet sources.

**Keywords:** Queuing, Vehicle detection in traffic, Deep learning, YOLO, SORT

## 1. GİRİŞ

Günümüzde artan nüfusla birlikte karayolu trafiğinin artması sonucunda trafikte kuyruklanma, sıkışıklık, ulaşımda gecikmeler gibi pek çok sorun ortaya çıkmaktadır. Bu tarz durumlarda kullanılabilir en iyi yaklaşımlardan biri Akıllı Ulaşım Sistemleridir (AUS). AUS, seyahat sürelerinin azaltılması, trafik güvenliğinin artırılması, mevcut yol kapasitelerinin verimli kullanılması gibi amaçlar doğrultusunda geliştirilen kullanıcı, araç, altyapı ve merkez arasında çok yönlü veri alışverişi ile izleme, ölçme, analiz ve kontrol mekanizmalarını içeren bilgi iletişim temelli sistemlerdir. AUS, trafikteki araçların oluşturduğu kuyruklanmaların uzunluklarını, araçların şeritte kalma ve seyahat sürelerini veya otobüslerin durağa varma süreleri gibi çeşitli tahminler için de kullanılabilir.

Nüfusun artması ve kentleşmenin yaygınlaşmasıyla birlikte, trafik yönetimi ve güvenliği günümüzde büyük bir önem arz etmektedir. Ekonomik büyüme ve şehirleşme ile her yıl daha fazla araç yollara çıkmakta ve bu durum trafik kazaları ve trafik sıkışıklığı gibi sorunları beraberinde getirmektedir. Trafik kazalarının önlenmesi ya da trafik sıkışıklığının kontrol altına alınabilmesi amacıyla trafikteki araç türü tespiti, araç sayımı, araç hız tespiti ve araçların kuyruk uzunluğunun otomatik olarak belirlenmesi gibi çalışmalar hızla artmaktadır. Araç sınıf tespiti, video görüntüleri ya da fotoğraflarda bulunan araçların türlerinin (otomobil, kamyon, otobüs, motosiklet vb.) belirlenmesini ifade etmektedir. Bu tespit, farklı araç türlerinin trafik üzerindeki etkilerinin analiz edilmesi ve hedeflenen planların uygulamasında önemli bir yere sahiptir. Araç sayımı, video görüntüleri ya da fotoğraflarda bulunan toplam araç sayısının bulunmasını içermektedir. Araç sayımı sayesinde trafik akışının zaman içindeki değişimleri takip edilmekte ve sayıma bağlı olarak yoğun saatlerde gerekli önlemler alınabilmektedir. Araç hız tespitinde ise şeritlerden geçen araçların hızları hesaplanmaktadır. Hesaplanmış hızlar, trafikte kuyruklanma durumunun ölçülmesinde büyük rol oynamaktadır. Trafikte oluşan kuyruk uzunluğu trafik sıkışıklığının boyutunu ve kuyruklanmanın meydana geldiği noktaları belirlemek için kullanılmaktadır. Uzun araç kuyukları, trafik akışının ciddi şekilde yavaşladığı veya durma noktasına geldiği yerlerde oluşmaktadır. Kuyruklanmanın sürekli olarak izlenmesi, trafik ışıkları ve kavşakların yönetimi gibi trafik kontrolünde daha etkili bir şekilde uygulanmasına yardımcı olmaktadır. Bu kapsamda, araçların otomatik olarak tespit edilmesi, türlerinin belirlenmesi ve belli bir bölgeden geçen araç sayısının hesaplanması gibi çalışmalar, trafik yönetim süreçlerinin optimize edilmesinde ve trafik güvenliğinin artırılmasında kritik bir rol oynamaktadır. Araçların otomatik olarak tespit edilmesi dışında ilgili bölgeden geçen araç sayısı, hızlarının tespiti ve kuyruklanmanın tahmin edilmesi gibi faktörler de trafik yönetiminde

önemli bir etkendir. Araç hız tespitinin yapılıp izlenmesi durumunda ilgili bölgedeki trafik sıkışıklığı azaltılabilir ve buna bağlı olarak seyahat süreleri kısaltılabilir. Kuyruklanmanın tespit edilmesi durumunda da kuyruklanma noktaları tespit edilip bölgelerdeki trafik akışının düzenlenmesi sağlanabilir. Özellikle trafikte kuyruklanma, şehirlerin en büyük problemlerinden biri haline gelebilmektedir. Trafik yoğunluğunun sürekli olarak takip edilmesi ve bunların analiz edilmesi, kuyruklanma noktalarının belirlenmesine ve bu noktalarda gerekli tedbirlerin alınmasına olanak sağlamaktadır. Bu tedbirlerin alınmasında geleneksel yaklaşımlara ek olarak son yıllarda görüntü işleme ve yapay zeka temelli derin öğrenme algoritmaları gibi teknolojiler hızla kullanılmaya başlanmıştır. Literatürde farklı araç tespit algoritmalarını içeren pek çok çalışma bulunmaktadır. Araç türü tespitinde tek bir modelin kullanıldığı çalışmalar olduğu gibi, birden fazla modelin kullanıldığı ve performanslarının karşılaştırıldığı çalışmalar da bulunmaktadır. Bu çalışmaların sonuçları, trafik yönetimi alanında yeni yöntemlerin ve teknolojilerin geliştirilmesine katkı sağlamaktadır.

### **1.1. Tezin Amacı**

Bu tez çalışmasında, trafikteki araç kuyruklanma durumunun tespit edilmesi için dört aşamalı bir sistem önerilmektedir; ilk aşamada, ilgili trafik videolarında akan araçların derin öğrenme yöntemi ile belirlenip türlerinin tespit edilmesi, ikinci aşamada, araçların şerit bazlı olarak sayımının gerçekleştirilmesi, üçüncü aşamada, şerit bazlı sayımı yapılan araçların hızlarının tespit edilmesi ve son aşamada ise hızları ve sayıları belirlenen araçların trafikte oluşturmuş olduğu kuyruklanma durumunun tespit edilmesi gerçekleştirilmiştir.

Literatürdeki kuyruklanma çalışmalarında genellikle videolardaki araçların hareket etmesi ya da sabit şekilde duruyor olması dikkate alınmış ve kuyruklanma olup olmadığı sadece bu bilgilere göre belirlenmiştir. Bu tez çalışmasında ise, literatürdeki çalışmalarından farklı olarak araçların türleri ve hızları da kuyruklanma durumunun tahmin edilmesi işlemine dahil edilmiştir. Böylece araç türleri ve hızları gibi gerçek veriler ile daha gerçekçi kuyruklanma durum tespitinin gerçekleştirilmesi amaçlanmıştır.

### **1.2. Literatür Taraması**

Büyük şehirlerde trafik yoğunluğu yaygın olarak görülen bir problemdir. Bu problemin en temel sebebi ise, genellikle hızlı kentleşme ve şehirlerin gelişmesiyle gelen nüfus ve araç sayısının artmasıdır. Trafik yoğunluğu ve beraberinde getirdiği sıkışıklık, şehirlerin altyapı ve ulaşım sistemlerinin yetersiz kalmasıyla daha da artmaktadır. Trafikteki araç tespiti ile sayımı

ve kuyruklanma tespitine yönelik literatürde gerçekleştirilmiş olan farklı çalışmalar bulunmaktadır.

Araç tespiti sürecinde, literatürde kullanılmış olan geleneksel görüntü işleme yöntemlerinin yanı sıra yapay zeka tabanlı derin öğrenme algoritmaları da önemli rol oynamaktadır. Görüntü işleme ve derin öğrenme tabanlı teknikler, trafikteki araçları tanımlamak ve izlemek gibi görevlerde kullanılarak trafik yönetimine önemli katkılar sağlamaktadır. Literatürde, araç tespiti konusunda çeşitli nesne algılama algoritmaları kullanılarak yapılan çalışmaların bazılarında yalnızca tek bir nesne algılama modelinin kullanılırken bazılarında ise birden fazla nesne algılama modelinin kullanıldığı ve bu modellerin karşılaştırıldığı uygulamalar bulunmaktadır.

Trafik videolarında araçların tespit doğruluğunu artırmak için Xiaomeng ve diğ. [1] tarafından YOLOv5 (You Only Look Once) tabanlı bir araç tespit modeli önerilmiştir. Modelin kesinlik, duyarlılık ve ortalama hassasiyet (mean average precision, mAP) değerleri sırasıyla %93,7, %94,2 ve %93,9 olarak bulunmuştur. Bir diğer çalışmada [2] ise araç sınıfı tespiti aşamasında YOLOv3 modeli iyileştirilmiş ve bu modelin performansı YOLOv2 ve YOLOv3 modelleri ile karşılaştırılmıştır. Bir başka çalışmada [3], YOLOv3 modelini temel alan geliştirilmiş bir araç tespit yöntemi önerilmiş ve model Single Shot Multibox Detector (SSD) ve YOLOv2 modelleri ile karşılaştırılmıştır. Araçları tespit edip sınıflandırmanın yanı sıra belirli bir alandan geçen araç sayısını belirlemek, trafik yönetimi ve güvenliği için trafik yoğunluğunu ve sıklığını belirlemede kritik bir öneme sahiptir. Literatürde araç sayma sistemleri üzerine çeşitli çalışmalar bulunmaktadır. Battal ve diğ. [4] tarafından gerçekleştirilen çalışmada, trafik videoları kullanılarak videodaki araçların YOLOv5 ile tespitini ve sınıflandırılması, ardından ilgili bölgeden geçen araç sayısının hesaplanması amaçlanmıştır. Araç sınıf tespitinde elde edilen modelin doğruluk oranı %88 iken model araç sayımında kullanılan sistemin doğruluk oranı %85'tir. Anil ve diğ. [5], UA-DETRAC veri setindeki videoları kullanarak YOLOv3 ile araç sınıfını (otomobil, otobüs, kamyon, motosiklet ve bisiklet) tespit etmiş ve Basit Çevrimiçi ve Gerçek Zamanlı Takip (Simple Online and Realtime Tracking, SORT) algoritması ile araç takibi gerçekleştirmiştir. Araç sayımına ilişkin başka bir çalışma da Algiriyage ve diğ. [6] tarafından önerilmiştir. Çalışmada, Closed-Circuit Television (CCTV) veri seti kullanılarak beş farklı araç tespiti için YOLOv4 ve araç sayımı için SORT algoritması kullanılmıştır. Farklı bir çalışmada [7], araç tespiti için YOLOv4 modeli, sayım için ise DeepSORT algoritması kullanılmıştır. Dört farklı araç sınıfının (otomobil, motosiklet, otobüs ve kamyon) tespiti için kullanılan modelin doğruluk oranının %82,08 olduğu bildirilmiştir. Bu çalışmalar, otomotiv

sektöründe sürücü destek sistemlerinin geliştirilmesine ve trafik güvenliğine yönelik önemli katkılarda bulunmaktadır. Nesne tespit algoritmaları arasında YOLO gibi yöntemler öne çıkarken, videolardaki nesnelere tespit etmek için arka plan çıkarma tekniğini kullanan birçok çalışma da bulunmaktadır. Seenoupong ve diğ. [8], arka plan çıkarma tekniği uygulayarak ve sanal bir algılama bölgesi kullanarak araç sayımı gerçekleştirmiştir. Başka bir çalışmada [9], arka plan modelleme ve Kalman filtresini içeren bir bilgisayarlı görme yaklaşımı kullanılmıştır. Video analizi ve nesne tespiti teknikleri, sadece araçların tespit edilmesi ve sınıflandırılmasını değil, aynı zamanda araçların hızlarını da doğru bir şekilde ölçmeyi mümkün kılmaktadır. Araçların hızlarının doğru bir şekilde belirlenmesi, trafik yönetimi, güvenliği ve planlaması açısından kritik bir öneme sahiptir. Bu bağlamda, nesne tespit algoritmaları ile birleştirilen hız ölçüm teknikleri, trafik akışını daha etkili bir şekilde analiz etmeyi ve potansiyel tehlikeleri önceden tespit etmeyi sağlamaktadır. Sharma ve diğ. [10], yaptıkları çalışmada ilk olarak araçların tespitini, daha sonra araç hızı tespitini ve son olarak da hızlı araçların plakalarını tespit etmeyi gerçekleştirmişlerdir. Uygulanan sistemin performansını analiz etmek için dört farklı video veri seti kullanılarak bir karşılaştırma yapılmıştır. Modelin, %89,02'lik duyarlılık ve %91,9'luk kesinlik değeri elde ettiği belirtilmiştir. Thadagoppula ve Upadhyaya [11], aracın hızını hesaplamak için güvenlik kameralarından gelen canlı video akışını kullanmıştır. Aracın hızı her yarım saniyede bir güncellendiği için kameranın görüş alanında aracın hızlanması ve yavaşlaması takip edilmiştir. Hesaplanan hızın, gerçek zamanlı olarak yalnızca %3'lük bir hata gösterdiği ifade edilmiştir. Rahman ve diğ. [12], YOLO algoritmasını kullanarak araç türlerini tespit etmişlerdir. Tespit edilen araçların İlgi Bölgesi'nden (RoI) geçişini takip ederek hızını hesaplamak amacıyla SORT algoritması kullanılmıştır. 5, 10 ve 15 metre RoI'de en düşük hata değerlerinin sırasıyla %4,79, %4,38 ve %2,96 olduğu belirtilmiştir. Sonuçların her biri 10 metre yükseklikte drone görüntüleri için elde edilmiştir. Waregaonkar ve diğ. [13], hız hesaplaması yapmışlar ve aracın hızını kat ettiği mesafeye dayalı olarak matematiksel bir formül kullanarak hesaplamışlardır. Hız tahminindeki hatanın %9 olduğu belirtilmiştir. Kumar ve diğ. [14], aracın hızını tahmin etmek için çift eksenli ve üç eksenli referans çizgileri kavramı kullanmıştır. Önerilen yöntemin hız tahmini doğruluğunun %94,4'e ulaştığı belirtilmiştir. Ayrıca, trafikteki araçların sayısını tespit etmek ve trafik yönetim sistemini geliştirmek için bir ilgi alanı sayma algoritması ve araçların takibi için de bir arka plan çıkarma yönteminin kullanıldığı ifade edilmiştir.

Trafikte kuyruklanma tespiti ile ilgili ise video görüntülerinden kuyruklanma ve mobil sensörler kullanılarak kuyruklanma tespiti gibi çeşitli yaklaşımlar bulunmaktadır. Umair ve diğ. [15],

düşük çözünürlüklü trafik videoları üzerinden şehir içi trafik senaryosunda araç kuyruk uzunluğunun tahmin edilmesi için Evrimsel Sinir Ağı (CNN) tabanlı bir yaklaşım önermişlerdir. Kuyruk uzunluğu, ışıklarda bekleyen toplam araç sayısına göre tahmin edilmiştir. Kuyruk uzunluğu tahmininde YOLOv3, YOLOv4, YOLOv5, SSD, ResNet101 ve Inceptionv3'ü içeren araç algılama modelleri deneysel olarak karşılaştırılmıştır. Deneysel sonuçlara göre kuyruk uzunluğunun tahmini için temel model olarak YOLOv4 modeli seçilmiştir. Önceden eğitilmiş YOLOv4 modeli kullanılarak, araç sayısı ve araç sayısına dayalı kuyruk uzunluğu tahmini için sırasıyla %73 ve %88 ortalama doğruluk elde edildiği belirtilmiştir. Modelin ince ayarı (fine-tuning) yapıldıktan ve çıktı sınıfları yalnızca tespit edilmek istenen araç sınıflarına göre düzenlendikten sonra, sırasıyla ortalama %83 ve %93 doğruluk elde edildiği belirtilmiştir. YOLOv4 modeli ile harici bir kamera ayarı veya kalibrasyon işlemi olmadan sadece düşük çözünürlüklü video görüntülerinden kuyruk uzunluğunun başarılı bir şekilde tespit edildiği ifade edilmiştir.

Okaishi ve diğ. [16], kuyruk uzunluğu tahmini için iki aşamalı bir sistem önermişlerdir. Sistemin ilk aşamasında, ilgili şeritlerde hareketin olup olmadığını tespit etmek için kareler farkı yöntemi kullanılmıştır. Herhangi bir hareket olmaması durumunda ikinci aşama olarak bu bölgelerdeki araçlar SSD algoritması kullanılarak tespit edilmiştir. Kuyruk uzunluğunun tahmini ölçümü için sınırlayıcı kutuların (bounding boxes) yükseklikleri toplanmıştır. Başka bir çalışmada [17] kuyruklanma tespiti, nesne takibi ya da arka plan çıkarım işlemi olmadan araçların düşük seviyeli özelliklerinin (aracın köşeleri) tespitine dayanmaktadır. Kuyruk uzunluğunu ölçmek için kuyruktaki ilk aracın başlangıç köşesi ve son aracın bitiş köşesinin koordinatları tespit edilmiştir. Bu koordinatlar ve Öklid mesafesi kullanılarak kuyruk uzunluğu hesaplanmıştır. Zhu ve diğ. [18], hareketli pencere yöntemi ve kenar tespiti yönteminin kombinasyonunu kullanarak kuyruk uzunluğu tespiti yapmıştır. Hareketli pencerenin durumu 3 farklı durum ile gözlemlenmiştir; araç kuyrukta, araç kuyruğa girmeye hazır ve kuyrukta araç yok. Araç kuyruğa girmeye hazır durumu hareketli pencerenin bir kısmının boşluk ve kalan kısmının araç içermesidir. Yani bu hareketli pencere alanı kuyruğa dahil edilmektedir. Kuyruğun başlangıcı ve sonu kenar tespiti ile belirlenerek Öklid mesafesi ile kuyruk uzunluğu tespit edilmiştir. Cai ve diğ. [19], gri renkli videoları kullanmışlardır. Kuyruk uzunluğu tespiti için yol önce 3,5 metrelik alanlara bölünmüş, daha sonra boş ve dolu alanlar bu alandaki özelliklerine (ışık, şekil, boyut) bakılarak tespit edilmiştir. İlgili alandaki görüntü entropisi (piksel değerleri arasındaki çeşitlilik) hesaplanması sonucu 0,8 değerinden büyükse kenar tespiti yapılmış ve bu alan kuyruğa dahil edilmiştir. Ardışık olarak eklenen her bir dolu alan

için kuyruk uzunluğu 3,5 metre artırılarak kuyruk uzunluğu hesaplanmıştır. Shirazi ve Morris [20] tarafından önerilen yöntemdeki kuyruk analizi, kuyruktaki araç sayısının bekleme sürelerinin ve kuyruk uzunluğunun tahmin edilmesini içermektedir. Çalışmada yazarlar tarafından belirli bir şerit belirlemesi yapılmamış bunun yerine Longest Common Subsequence tekniğiyle şeritler belirlenmiştir. Belirlenen her şeritte araçların bekleme veya hareket etme durumu durağan veya hareket özelliklerinden tespit edilmiştir. Araçların bekleme durumu tespit edildiğinde, bu araç kuyruk uzunluğunun tahmini ölçümünde kullanılmıştır. Kuyruk uzunluğu, kuyruktaki ilk aracın ön köşesinin ve son aracın arka köşesinin piksel koordinatlarının Öklid mesafesi kullanılmasıyla hesaplanmıştır. Qi ve diğ. [21] tarafından gerçekleştirilen çalışmada, kavşaklardaki araç kuyruk uzunluğu videolar yerine fotoğraflar kullanılarak gerçekleştirilmiştir. Görüntü ön işleme aşamasında, önce görüntünün parlaklığı otomatik olarak düzenlenmiş ve sonra şeritlerin çizgileri kenar tespit yöntemi ve Olasılıksal Hough Dönüşümü (Probabilistic Hough Transform) yöntemiyle tespit edilmiştir. Daha sonra görüntüdeki şerit çizgileri arasında arka plan ve ön plan çıkarımı uygulanmıştır. Ön plan görselinde araçlar beyaz renkte gösterilmiştir. Kuyruk başındaki beyaz pikselin ve kuyruk sonundaki siyah pikselin koordinat bilgileri kuyruk uzunluğu ölçümünde kullanılmıştır. Kuyruklanma ile ilgili literatür taramasında görüntü işleme tekniklerinin kullanılarak kuyruklanma tespitinin yanı sıra sayısal veri seti kullanılarak kuyruklanma tespitinin yapıldığı çalışmalar da bulunmaktadır. Lee ve diğ. çalışmalarında [22] kümülatif hesaplama hatalarını en aza indirmek üzere CNN ile Uzun Kısa Vadeli Bellek (Long- Short Term Memory, LSTM) yöntemlerini birleştirerek entegre şerit tabanlı kuyruk uzunluğu tahmin modeli geliştirilmişlerdir. Şerit kullanım oranı tahmini için şerit üzerinde kısa zamanda (anlık) değişimler ve uzun süredeki (haftalık, aylık vs.) değişimler LSTM'de kullanılmıştır. Geliştirilen entegre derin öğrenme modelinin tek tek şeritlerdeki kuyruk uzunluklarını tahmin etmede yüksek performans gösterdiği ancak hem izole sinyalizasyon kavşaklar hem de alan trafiği için sinyal gruplarının süresini, başlangıç zamanını ve sırasını birkaç saniye veya dakika olarak ayarlama yetersiz kaldığı yazarlar tarafından belirtilmiştir. Gerçek zamanlı trafik verileri kullanılan ve tahmin döngülerine dayanan bir çalışmada [23], bir sonraki döngüde kuyruk uzunluklarını tahmin etmek için LSTM kullanılmıştır. Bunun yanı sıra aşırı öğrenmeyi engellemek ve hiper parametre seçimi için sıra model tabanlı optimizasyon tekniği kullanılmıştır. Trafik kontrol sistemi veri seti ile gerçekleştirilen deneylerde araçların sadece düz hareket yönüne yönelik kuyruk uzunluğu tahmin edilmeye çalışılmıştır. Zhao ve diğ. [24], her bir şeritte yer alan prob araçlar kullanarak bayes tabanlı farklı yaklaşımlar geliştirmişlerdir. Yolun trafik kapasitesi, yoğunluğu ile yoldaki kuyruk uzunluğunun hesaplanmasında ve tahmininde, bulunduğu yol ortamından veri toplamak üzere çeşitli

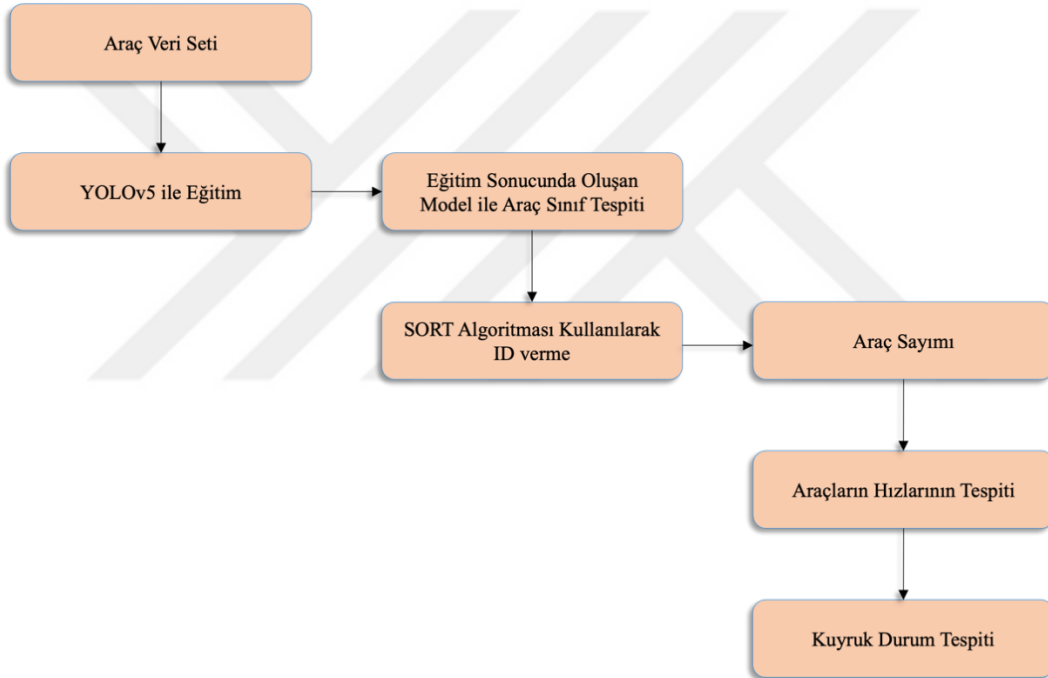
teknolojik donanımlara sahip olan prob araçlar kuyruk uzunluğu, trafik hacmi tahmini gibi alanlarda sıklıkla kullanılmaktadır. Prob araçların karayolundaki diğer araçlara göre oranını ifade eden penetrasyonun ve penetrasyon hızının tahminine dayalı bu yaklaşımlarla kuyruk uzunlukları düz devam eden ve sağa dönüş yapılan şeritlerde hesaplanmıştır. Yazarlara göre geliştirilen modelin yüksek bir tahmin doğruluğuna ulaştığı ve çalışmanın trafik sinyal kontrolünde kullanılabileceği belirtilmiştir.

Tezin 2. bölümünde YOLOv5 mimarisinde kullanılan CNN, YOLO, SORT ve hiper parametreler hakkında bilgiler sunulmuştur. 3. bölümde, tezde kullanılan veri seti, veri setinin etiketlenmesi, tezin dört aşamasından (araç sınıf tespiti, araç sayımı, araçların tahmini hız tespiti ve kuyruk durum tespiti) ve değerlendirme kriterlerinden, 4. bölümde elde edilen test sonuçlarından ve 5. bölümde ise sonuçlar ve önerilerden bahsedilmiştir.



## 2. DERİN ÖĞRENME

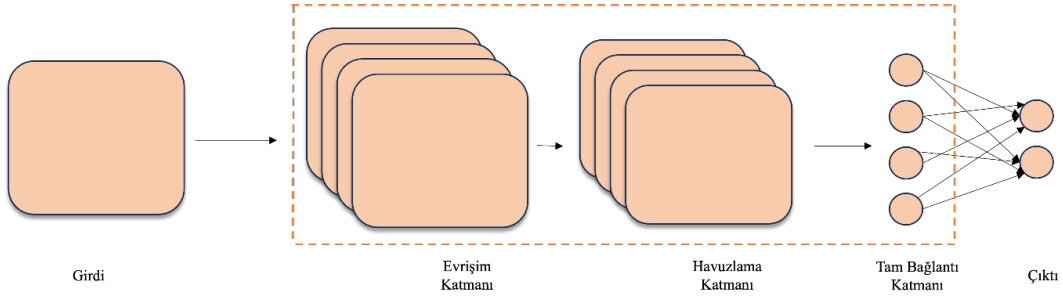
Trafikteki araç kuyruklanma durumunun derin öğrenme ile gerçek zamanlı tespit sisteminin blok diyagramı Şekil 2.1’de gösterilmektedir. Blok diyagramındaki gösterilen ilk aşamada araç veri seti içerisinde bulunan araba, motosiklet, kamyon, otobüs ve bisiklet görsellerini içeren veri seti kullanılarak YOLOv5 [25] algoritması kullanılarak eğitim işlemi gerçekleştirilmiştir. Eğitim sonucunda elde edilen model ile videoda ilgili bölgelerdeki araçların sınıf tespiti gerçekleştirilmiştir. İlgili bölgedeki araçların tespit işleminden sonra SORT algoritması kullanılarak tespit edilen araçlara ID verilip araç sayım işlemi yapılmıştır. Araç sayımı yapıldıktan sonra son aşamada araçların hızlarının tespiti yapıлып, hız sonuçlarına göre kuyruklanma durumunun tespiti gerçekleştirilmektedir.



Şekil 2.1. Önerilen sistemin blok diyagramı

### 2.1. Evrişimsel Sinir Ağı

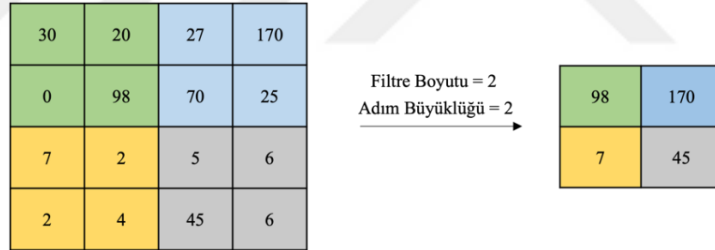
CNN, görüntü işleme ve sınıflandırma gibi çalışmalarda sıklıkla kullanılan bir yapay sinir ağı modelidir. Girdi olarak bir görüntüyü alır ve görüntüdeki nesnelere birbirinden ayırmaktadır. Şekil 2.2’de örnek bir CNN mimarisi gösterilmiştir. Mimari incelendiğinde bir CNN’nin, girdi, evrişim katmanı, havuzlama katmanı, tam bağlantı katman ve çıktı katmanını içerdiği görülmektedir.



Şekil 2.2. Evrişimsel sinir ağı mimarisi

Evrişim Katmanı, CNN'nin önemli katmanlarından biridir. Girdi olarak verilen görselin özelliklerini çıkarmak için kullanılmaktadır.

Havuzlama Katmanı, görsellerden çıkarılan özellik haritalarının boyutlarını azaltmayı amaçlamaktadır. Bu katmanda maksimum havuzlama ve ortalama havuzlama gibi farklı havuzlama yöntemleri bulunmaktadır.  $n \times n$  lik bir filtre, görsel üzerinde gezinirken gezinme işlemi sırasında belirlenen havuzlama teknikleri uygulanmaktadır. Eğer kullanılan teknik maksimum havuzlama yöntemi ise filtrenin kapsadığı alandaki en büyük değer ele alınmaktadır. Şekil 2.3'te filtre boyutu iki ve adım büyüklüğü iki olarak belirlenen bir maksimum havuzlama yöntemi gösterilmektedir.



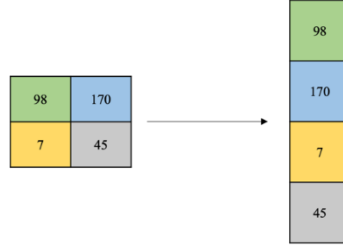
Şekil 2.3. Maksimum havuzlama örneği

Eğer kullanılan teknik ortalama havuzlama yöntemi ise  $n \times n$  lik filtrenin kapsadığı alandaki değerlerin ortalaması alınmaktadır. Şekil 2.4'te filtre boyutu iki ve adım büyüklüğü iki olarak belirlenen bir ortalama havuzlama yöntemi gösterilmektedir.



Şekil 2.4. Ortalama havuzlama örneği

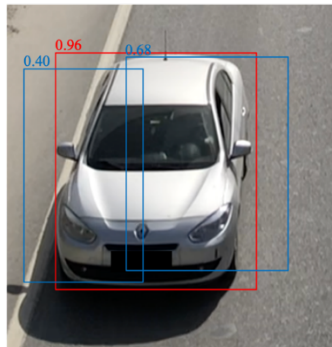
Havuzlama katmanından sonraki bir diğer aşama ise tam bağlantı katmanıdır. Matris formatındaki girdinin önce tek boyutlu bir vektöre dönüştürülüp sonrasında tam bağlantı katmanına iletilmektedir. Vektör formatına dönüştürülen girdi yapay sinir ağı yolu kullanılarak öğrenme işleminde kullanılmaktadır. Şekil 2.5’te 2 boyutlu bir matrisin düzleştirme işlemi gösterilmektedir.



Şekil 2.5. Düzleştirme örneği

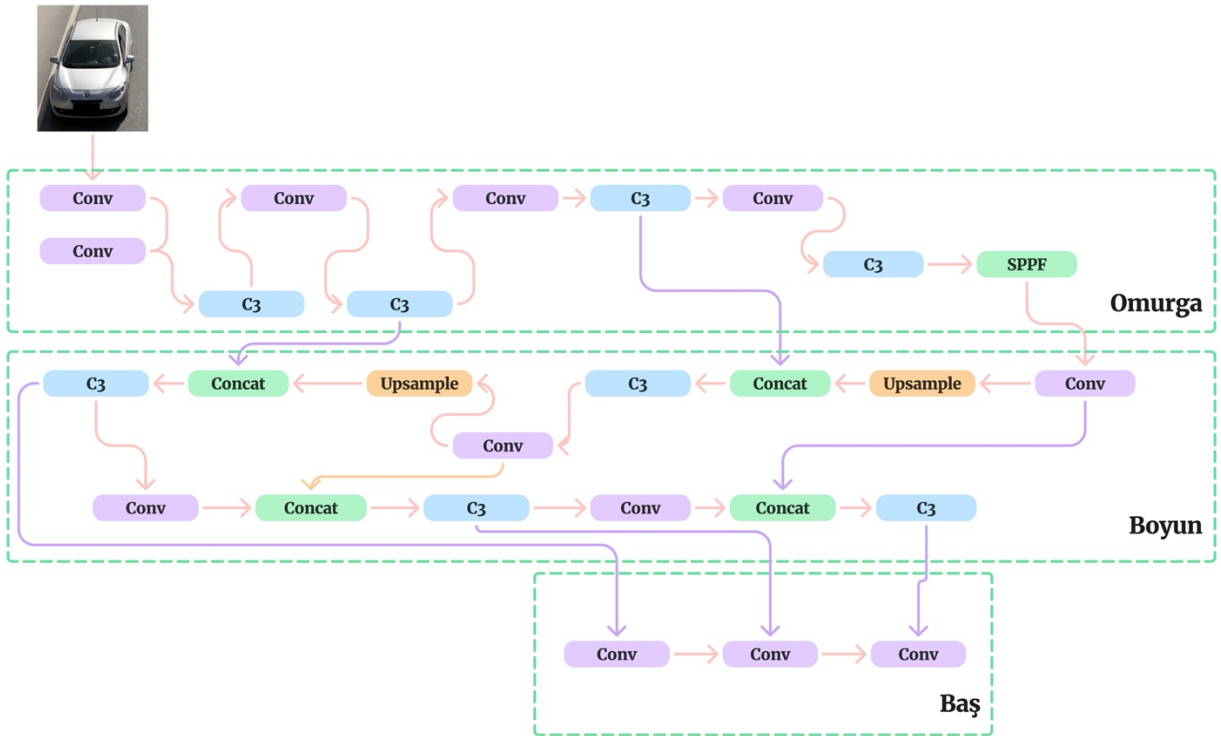
## 2.2. YOLO

“You Only Look Once” ifadesinin kısaltması olan YOLO, 2016 yılında Redmon ve diğ. [26] tarafından geliştirilmiştir. Sürücüsüz arabalar, gözetleme sistemleri ve görüntü tanıma gibi çeşitli uygulamalarda yaygın olarak kullanılan gerçek zamanlı bir nesne algılama algoritmasıdır. Algoritma, sınırlayıcı kutuları ve sınıf olasılıklarının tahmin aşamasında tek bir sinir ağı kullanmaktadır. Girdi olarak verilen görüntü  $S \times S$  boyutunda ızgaraya bölünür ve her bir ızgaranın merkezi tespit edilir. Daha sonra ızgaralarda tespit edilen nesnelere göre bir sınırlayıcı kutu çizilir. Eğer ızgaranın merkezi çizilen sınırlayıcı kutu içerisinde ise bu ana sınırlayıcı kutuya dahil edilmektedir. Bu dahil etme işlemi için güven puanı kullanılmaktadır. Her bir ızgara hücresi, sınırlayıcı kutu ve bu kutuların güven puanlarını tespit eder. Şekil 2.6’da bir nesne için çizilen sınırlayıcı kutular gösterilmiştir. Güven puanı, modelin geçerli ızgara içerisinde nesne olup olmadığından ne kadar emin olduğunu gösterir. Eğer nesne varsa o nesnenin gerçekten o nesne olup olmadığını ve etrafındaki kutunun koordinatlarından ne kadar doğru olduğu gösterir.



Şekil 2.6. YOLO algoritmasında nesne için çizilen sınırlayıcı kutular

Araç sınıf tespiti aşamasında, YOLO'nun beşinci versiyonu olan YOLOv5 kullanılmıştır. YOLOv5 mimarisi, omurga, boyun ve baş olmak üzere üç bölümden oluşmaktadır. YOLOv5 mimarisi Şekil 2.7'de gösterilmektedir. Kullanılan modelin konfigürasyon dosyası ve mimarisi incelendiğinde omurga bölümünde, giriş görüntüsünden özellik haritaları üreten bir özellik çıkarıcı bulunduğu görülmektedir. Bu kısımda her katman, bir önceki katmandan aldığı girdiyi işleyerek özellik çıkarımı yapmaktadır. İlk olarak, 64 filtrelilik, 6×6 boyutunda çekirdeğe sahip ve 2 adımlama ile çalışan bir evrişimsel katman bulunmaktadır. Ardından, 128 filtrelilik ve 3×3 çekirdek boyutuna sahip bir başka evrişimsel katmanı gelmektedir. Üçüncü katmanda, C3 modülü kullanılarak 128 filtre ile üç kez tekrar edilen bir yapı mevcuttur. Daha sonra, 256 filtrelilik ve 3×3 çekirdek boyutunda bir evrişimsel katmanı ile 256 filtrelilik C3 modülünün altı kez tekrarlandığı bir yapı bulunur. Beşinci katmanda, 512 filtrelilik ve 3×3 çekirdek boyutunda bir evrişimsel katmanı yer almaktadır. Bu adımı, 512 filtrelilik C3 modülünün dokuz kez tekrarlandığı bir yapı takip etmektedir. Yedinci katmanda, 1024 filtrelilik ve 3×3 çekirdek boyutunda bir evrişimsel katmanı bulunur ve ardından 1024 filtrelilik C3 modülü üç kez tekrar edilmektedir. Son olarak, 1024 filtrelilik ve 5×5 çekirdek boyutuna sahip Uzamsal Piramit Havuzlama-Hızlı (Spatial Pyramid Pooling – Fast, SPPF) modülü yer almaktadır. Bu omurga yapısı, YOLOv5'in farklı ölçeklerdeki nesnelere etkin bir şekilde algılamasını sağlayan yüksek seviyeli özellikler çıkarmak üzere tasarlanmıştır.



Şekil 2.7. YOLOv5 mimarisi [27]

Boyun yapısı ise omurgadan gelen özellik haritalarını işleyerek nesne tespitini gerçekleştiren katmanları içermektedir. Boyun kısmı, çeşitli evrişimsel katmanlar, yukarı örnekleme (upsampling) ve birleştirme (concat) işlemleri ile özellik haritalarını işlemekte ve nesnelere tespit etmektedir. İlk olarak, 512 filtrelili ve  $1 \times 1$  çekirdek boyutunda bir evrişimsel katmanı uygulamaktadır ve ardından yukarı örnekleme yapmaktadır. Bu özellik haritaları, omurganın P4 katmanı ile birleştirilmektedir ve 512 filtrelili C3 modülü ile işlenmektedir. Aynı şekilde, 256 filtrelili konvolüsyon katmanı ve yukarı örnekleme işlemi ile bu özellikler, omurganın P3 katmanı ile birleştirilmektedir ve 256 filtrelili C3 modülü ile işlenmektedir. Bu katman küçük nesnelere tespit etmek için kullanılmaktadır. Orta büyüklükteki nesnelere tespit etmek için, 256 filtrelili  $3 \times 3$  çekirdek boyutunda evrişimsel katmanı kullanılmaktadır ve bu özellikler, başlığın P4 katmanı ile birleştirilmektedir. Ardından 512 filtrelili C3 modülü ile işlenmektedir. Büyük nesnelere tespit etmek için ise 512 filtrelili  $3 \times 3$  çekirdek boyutunda evrişimsel katman kullanılmaktadır ve bu özellikler, başlığın P5 katmanı ile birleştirilerek 1024 filtrelili C3 modülü ile işlenmektedir. Son olarak, baş kısmında ise tespit katmanı P3, P4 ve P5 ölçeklerinden gelen özellik haritalarını kullanarak nesnelere tespit etmektedir.

### **2.3. Hiper Parametreler**

Derin öğrenmede hiper parametreler, bir modelin öğrenme sürecini etkileyen önemli faktörlerdir. Bu parametreler eğitime başlamadan önce belirlenip modelin eğitimi sırasında gerekli durumlarda güncellenebilmektedir. Hiper parametreler modelin katman sayısı, devir sayısı, parti boyutu, öğrenme oranı gibi kısımları kapsamaktadır. Doğru hiper parametre değerleri modelin iyi öğrenmesinde katkı sağlarken yanlış hiper parametre değerleri modelin öğrenmesini zorlaştırmaktadır.

#### **2.3.1 Devir sayısı**

Devir (Epoch) sayısı, eğitim aşamasında kullanılan bir hiper parametredir. Bu parametre bir veri seti üzerinde kaç kez çalışılacağını göstermektedir.

#### **2.3.2 Parti boyutu**

Parti boyutu (Batch size), model eğitime başlamadan önce tanımlanan bir başka önemli parametredir. Bu parametre modelin eğitim aşamasında girilen parti boyutu kadar verinin aynı anda eğitimde kullanılacağı anlamına gelmektedir. Bu değer genelde ikinin üssel değerleri olarak seçilmektedir.

### 2.3.3 Öğrenme oranı

Öğrenme oranı (Learning rate) modelin parametrelerinin ne kadar hızlı güncelleneceğini tanımlamaktadır [28]. Makine öğrenmesinde ve derin öğrenmede optimizasyon aşamasında kullanılmaktadır. Öğrenme oranı küçük olduğunda model yavaş öğrenir fakat düzgün bir yakınsama sağlamaktadır. Öğrenme oranı büyük olduğunda ise öğrenmeyi hızlandırır ancak yakınsamayabilir [28].

### 2.4. SORT

SORT, çoklu nesne takibinde kullanılan bir algoritmadır [29]. Özellikle hız gerektiren gerçek zamanlı uygulamalarda tercih edilmektedir. Takip aşamasında Kalman filtresi [30] ve Macar algoritması [31] gibi tekniklerin kombinasyonu kullanılmaktadır. Bu sistem nesneyi tespit etme, tespit edilen nesnenin sonraki video kareleri arasında tespit etmeye devam etmesi, video kareleri arasında tespit etmeye devam ederken verilen araç ID'lerinin korunması ve nesnenin takibinin sürekli olması olmak üzere dört aşamadan oluşmaktadır [29].

SORT ile nesne takibi için önce nesne tespit etme aşamasının gerçekleştirilmesi gerekmektedir. Nesne tespiti için, YOLO veya Daha Hızlı Bölge CNN (Faster Region CNN, FrRCNN) [32] gibi nesne tespit modelleri kullanılmaktadır. Her bir nesnenin kareler arasındaki yer değiştirmesi diğer nesnelere ve kamera hareketlerinden bağımsız bir şekilde hesaplanmaktadır. Algılamaların mevcut hedeflere atanmasında, hedefin sınırlayıcı kutu geometrisi var olan çerçevedeki konumu hesaplanarak tahmin edilir. Atama maliyeti matrisi, her tespit ile mevcut hedeflerden tahmin edilen tüm sınırlayıcı kutular arasındaki IOU (Intersection-Over-Union) mesafesi olarak hesaplanır [29]. Buradaki atama işlemi Macar algoritması kullanılarak gerçekleştirilir. Ek olarak, hedef çakışmasına yönelik algılamaların  $IOU_{min}$  değerinden az olduğu atamaları reddetmek için bir minimum IOU uygulanır [29]. Nesnelerin ilgili görüntüye girip çıkması durumunda ID oluşturulması veya yok edilmesi gerekmektedir.

### 3. MATERYAL VE METOT

#### 3.1. Veri Seti

Tez çalışmasında kullanılan veri seti, birbirinden farklı üç veri setinin birleşiminden oluşmaktadır. Kullanılan veri setleri GitHub [33] ve Kaggle'dan [34, 35] alınmıştır. Oluşturulan veri seti içerisinde beş farklı sınıf için (otomobil, motosiklet, kamyon, otobüs ve bisiklet) etiketlenmiş görseller bulunmaktadır. GitHub'taki görsellerin etiket dosyaları mevcut iken Kaggle'dan alınan dosyaların etiket dosyaları bulunmamaktadır. GitHub'ta bulunan veri setlerinin içerisindeki görsellerin bir kısmı ve Kaggle'daki veri setlerinin içerisindeki görsellerin bir kısmı kullanılarak Araç veri seti oluşturulmuştur. Tablo 3.1'de modelin eğitimi için kullanılan görsellerin sayıları hakkında bilgi verilmektedir.

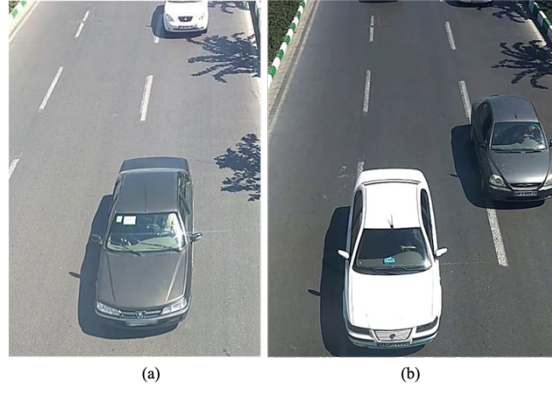
Tablo 3.1. Veri seti hakkında bilgi

	Veri Sayısı
Eğitim	1593
Test	309

Tablo 3.2'de araç veri seti içerisindeki her bir sınıfa ait olan etiket sayıları belirtilmiştir. Diğer bir deyişle 1902 görsel içeren veri seti içerisinde toplam 2565 adet otomobil, 382 adet motosiklet, 459 adet kamyon, 411 adet otobüs ve 84 adet bisiklet etiketi bulunmaktadır. Şekil 3.1'de GitHub'tan alınan veri setinden örnekler gösterilmiştir.

Tablo 3.2. Veri setindeki sınıf etiketleri sayısı

	Etiket Sayısı
Otomobil	2565
Motosiklet	382
Kamyon	459
Otobüs	411
Bisiklet	84
Toplam	3901



Şekil 3.1. GitHub veri setinden örnekler

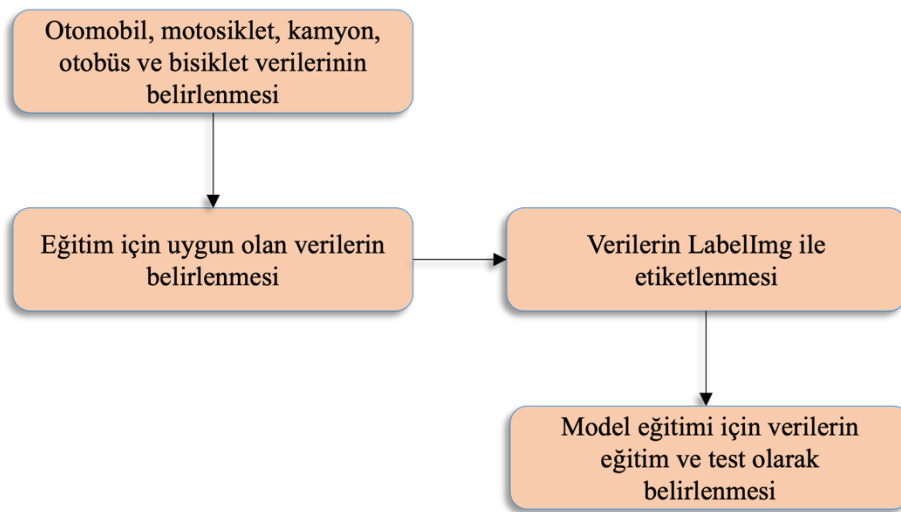
Şekil 3.2’de ise Kaggle’ dan alınan veri setinden örnekler gösterilmiştir.



Şekil 3.2. Kaggle veri setinden örnekler

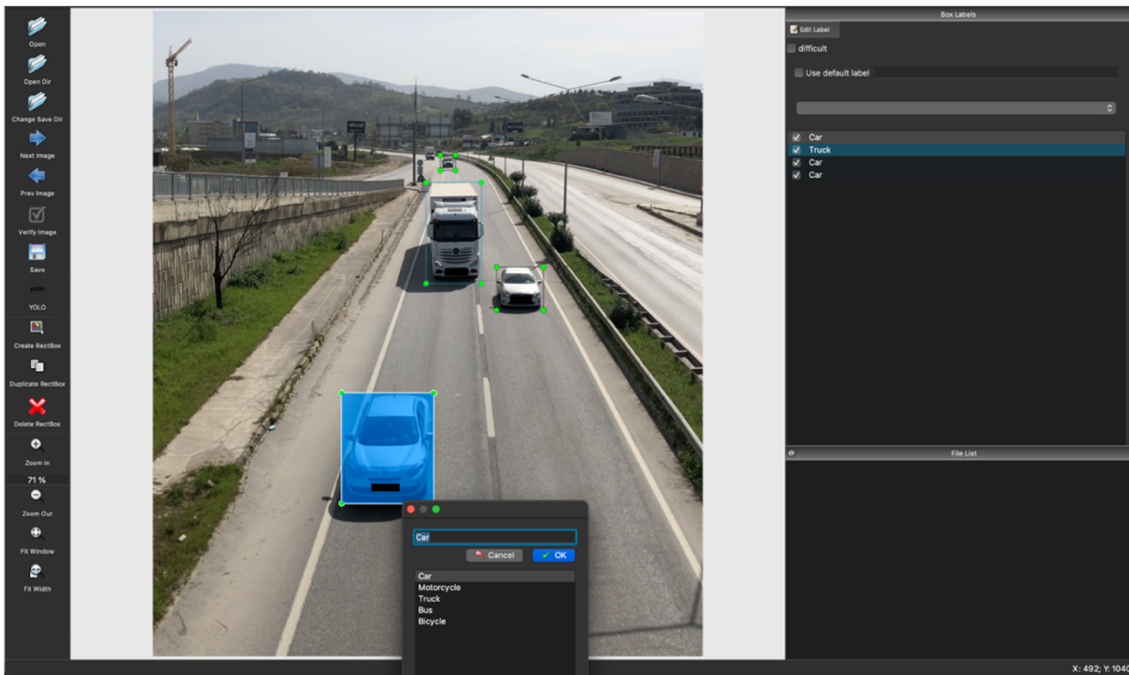
### 3.2. Verilerin Etiketlenmesi

Trafik akış videolarındaki otomobil, motosiklet, kamyon, otobüs ve bisiklet sınıflarını etiketlemek için gerekli olan adımlar Şekil 3.3’te gösterilmiştir.



Şekil 3.3. Veri etiketleme diyagramı

Veri seti içerisindeki GitHub'tan alınan görsellerin etiket dosyası mevcut iken daha sonra eklenen diğer veri setlerindeki görsellerin etiket dosyaları bulunmamaktadır. Bu görsellerin etiketleme işlemleri için nesne tespiti gibi çalışmalarda tercih edilen LabelImg [36] isimli açık kaynaklı etiketleme aracı kullanılmıştır. Görseller üzerinde ilgili nesnelerin olduğu bölgenin işaretlenip sınıf etiketi eklenmesini sağlamaktadır. Etiketleme işleminden sonra çıktı xml veya txt olmak üzere iki formattan birinde kaydedilmektedir ve oluşturulan bu dosyalar derin öğrenme modellerinde kullanılmaktadır. LabelImg etiketleme uygulaması ekranı Şekil 3.4'te görülmektedir. YOLO ile eğitim yapılacağından dolayı uygulamada etiketleme sonucu oluşturulan dosya türü YOLO modelinde kullanılan txt formatında kaydedilmektedir.



Şekil 3.4. Veri etiketleme uygulaması

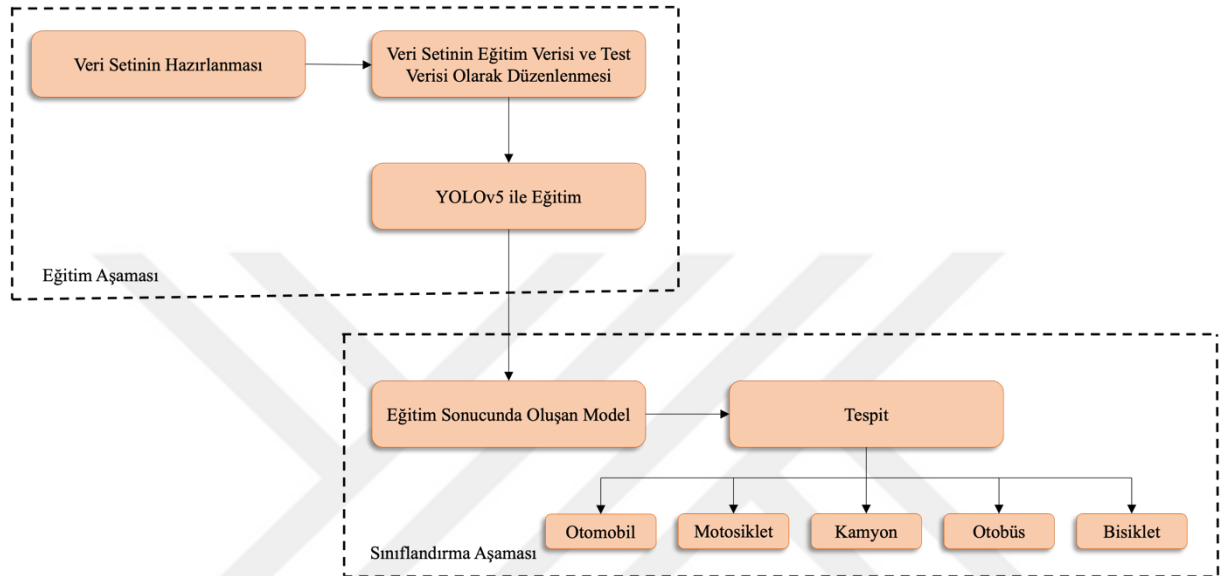
Etiketleme sonucunda oluşan txt formatındaki dosyadan bir örnek Şekil 3.5'te gösterilmektedir. Dosyada beş farklı değer bulunmaktadır. İlk değer nesnenin sınıfını, ikinci değer merkez noktasının x koordinatını, üçüncü değer merkez noktasının y koordinatını, dördüncü değer genişlik değerini ve son değer ise yükseklik değerini ifade etmektedir.

```
0 0.1993055555555557 0.058203125 0.1097222222222222 0.04453125
```

Şekil 3.5. Etiketleme sonucu oluşan txt dosyası

### 3.3. Araç Sınıf Tespiti

Videoda bulunan aracın sınıf tespiti için ilk adım eğitim aşamasıdır. Eğitim aşamasında YOLOv5 kullanılmıştır. YOLOv5 ile eğitim ve sınıflandırma işleminin adımları Şekil 3.6'da gösterilmiştir. Diyagram incelendiğinde, eğitim aşamasında araç sınıf tespiti için gerekli olan veriler ayarlanmaktadır ve oluşturulan araç veri seti eğitim verisi ve test verisi olmak üzere ikiye ayrılmaktadır. Bu veri seti daha sonra YOLOv5 ile eğitilmektedir.

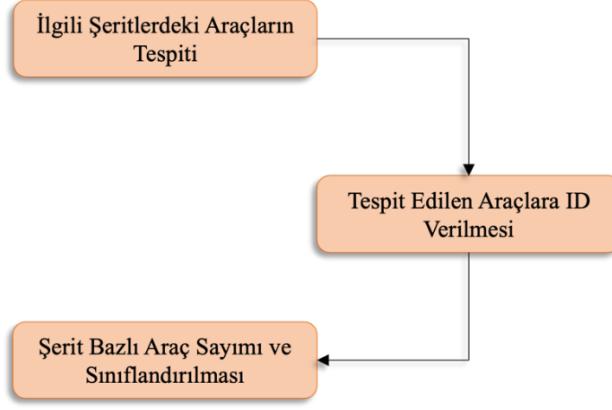


Şekil 3.6. Eğitim ve sınıflandırma diyagramı

Sınıflandırma aşamasında ise, eğitim sonucunda oluşan model, test için belirlenen videolar kullanılarak test edilmektedir. Test işlemi sonucunda videodaki nesnelere otomobil, motosiklet, kamyon, otobüs ve bisiklet olarak sınıflandırma yapılmaktadır.

### 3.4. Araç Sayımı

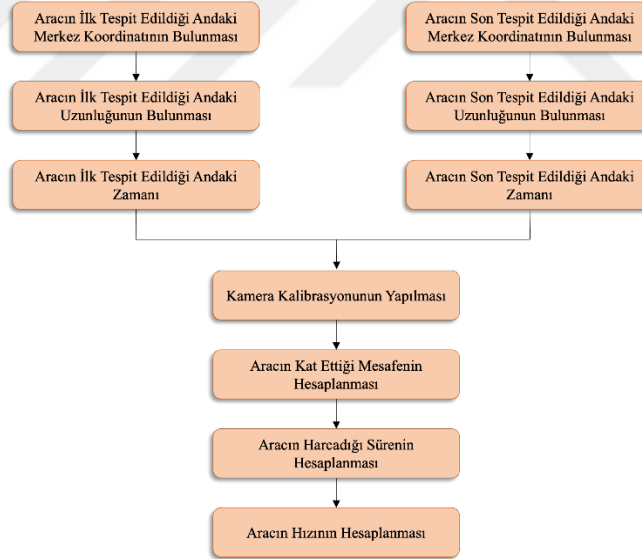
Şerit bazlı araç sayımı aşamasında araç sınıf tespitinin ardından tespit edilen her bir araç için SORT algoritması kullanılarak ID verilmektedir ve her bir şeritteki toplam araç sayısı ayrı ayrı hesaplanmaktadır. Eğer tespit edilen araç kendi bulunduğu şeritten başka bir şeride geçerse (şerit değişikliği durumu) bu durumda tespit edilmekte ve tespit edildikten sonra şeritlerdeki araç sayısı güncellenmektedir. Araç sayımı için izlenen adımlar Şekil 3.7'de gösterilmiştir.



Şekil 3.7. Araç sayım diyagramı

### 3.5. Hız Tespiti

Araç sınıf tespiti ve araç sayım işlemi yapıldıktan sonraki bir diğer aşama ise tespit edilen araçların hız tespiti tahmininin gerçekleştirilmesidir. Şekil 3.8’deki diyagrama bakıldığında ilk olarak videodaki aracın ilgili bölgeye girdiği andaki merkez koordinatı, uzunluğu ve videoda kaçınıcı saniyede yer aldığı bilgisi bulunmaktadır. Ardından aracın ilgili bölgeden çıktığı andaki merkez koordinatı, uzunluğu ve videoda kaçınıcı saniyede yer aldığı bilgisi bulunmaktadır.



Şekil 3.8. Hız tespiti diyagramı

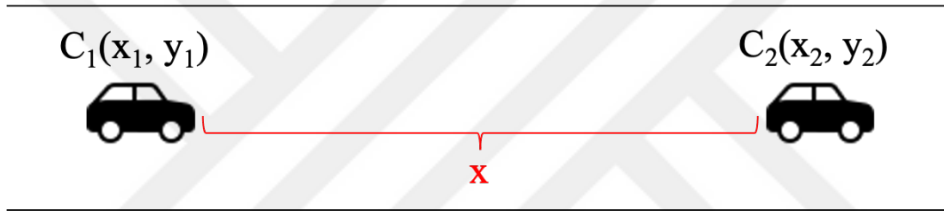
İlgili değerler bulunduktan sonra kamera kalibrasyonu yapılmakta ve Öklid mesafesi, aracın toplam hareket süresi ve tahmini hız değerleri hesaplanmaktadır. Test aşamasında kullanılan videoların kamera bilgileri yer almadığı için kamera kalibrasyonu yapılmalıdır. Kamera kalibrasyonu yapılırken [11] aracın tahmini gerçek uzunluğunun [37] ve videodaki piksel bazlı uzunluğunun bilinmesi gerekmektedir. İlgili değerler bulunduktan sonra Denklem (3.1)’deki

işlem gerçekleştirilmektedir. Denklem sonucunda elde edilen kamera kalibrasyonunun birimi metre/piksel olarak bulunmaktadır.

$$k = \frac{\text{Aracın gerçek uzunluğu (metre)}}{\text{Aracın piksel bazlı uzunluğu (piksel)}} \quad (3.1)$$

Motosiklet ve bisikletin kuyruklanmaya herhangi bir etkisi olmadığı için tahmini uzunluğu sıfır olarak alınmaktadır. Bu yüzden motosiklet ve bisiklet sınıfı şerit sayımına dahil edilirken hız hesaplama ve kuyruklanmaya dahil edilmemektedir. Hızları doğrudan sıfır olarak hesaplanmakta ve kuyruklanmada ortalama hıza dahil edilmemektedir.

Kamera kalibrasyonu yapıldıktan sonra aracın video boyunca aldığı mesafeyi bulmak için Şekil 3.9'da gösterildiği gibi aracın ilk tespit edildiği merkez koordinatı  $C_1(x_1, y_1)$  ile son tespit edildiği andaki merkez koordinatı  $C_2(x_2, y_2)$  belirlenmektedir. Ardından  $C_1$  ve  $C_2$  arasındaki Öklid mesafesi piksel cinsinden hesaplanmaktadır.



Şekil 3.9. Hız için gerekli olan bilgilerin gösterilmesi

Öklid mesafesinin formülü Denklem (3.2)'de gösterilmiştir.

$$X (\text{piksel}) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (3.2)$$

Aracın kat ettiği mesafeyi bulmak için kullanılan Öklid mesafesinin birimi piksel olduğu için Denklem (3.3)'te alınan mesafenin metre birimine dönüştürülme işlemi yapılmaktadır.

$$X (\text{metre}) = k \times X (\text{piksel}) \quad (3.3)$$

Aracın kat ettiği mesafeyi metre cinsinden hesaplandıktan sonraki aşama ise videoda tespit edilen aracın tespit edildiği andan itibaren kaç saniye boyunca hareket ettiğinin bulunmasıdır. Bu sürenin hesaplanması Denklem (3.4)'te gösterilmiştir.  $t_1$  aracın videoda ilk tespit edildiği andaki zamanı,  $t_2$  ise aracın videoda son tespit edildiği andaki zamanı olarak kaydedilmektedir.

$$t = t_2 - t_1 \quad (3.4)$$

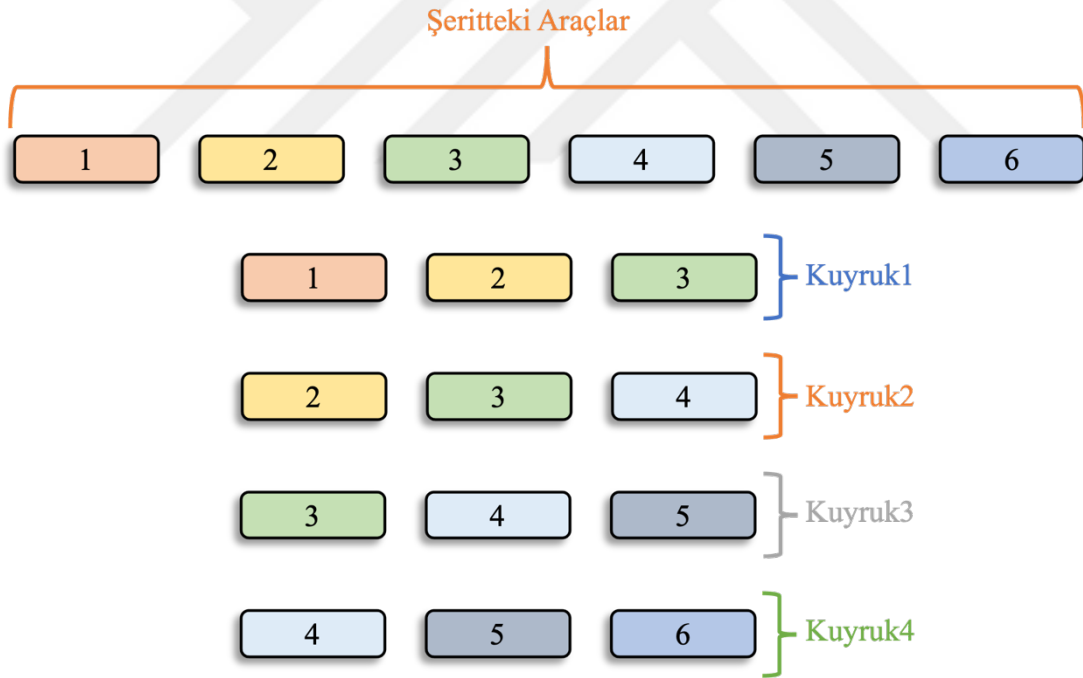
Son aşama olan tahmini hız hesaplamada kısmında ise bulunan tüm değerler formülde yerine yerleştirilmektedir.  $(X/t)$ 'ni birimi m/s olarak hesaplanmaktadır.  $X$ , aracın kat ettiği mesafe,  $t$

ise aracın hareket süresidir. Aracın tahmini hızının km/sa olarak hesaplanması için çıkan sonuç 3,6 ile çarpılmaktadır. Kullanılan formül Denklem (3.5)'te gösterilmiştir.

$$V = (X/t) \times 3.6 \quad (3.5)$$

### 3.6. Kuyruklanma Durum Tespiti

Kuyruklanma durum tespiti yapılırken değerlendirme kriteri olarak tespit edilen araçların ortalama hızları dikkate alınmaktadır. Bu aşamada ilgili bölgeden geçen araçların hızları kaydedilmektedir. Ardından ardışık üç aracın hızlarının ortalaması alınarak kuyruklanma durum tespiti yapılmaktadır. Şeride yeni bir araç girdiği anda şeride ilk giren araç kuyruktan çıkartılarak ve yeni giren araç kuyruğa eklenerek tekrar ortalama hız hesaplaması yapılmaktadır. Şekil 3.10'da kuyruklanma tespiti ile ilgili bir örnek gösterilmektedir. İlk olarak tespit edilen 1, 2 ve 3 numaralı araçların ortalama hızları hesaplanmaktadır. Daha sonra 1 numaralı araç kuyruktan çıkarılıp 2, 3 ve 4 numaralı araçların ortalama hızları hesaplanmaktadır ve hesaplamalar bu şekilde devam etmektedir.



Şekil 3.10. Kuyruklanma durum tespiti

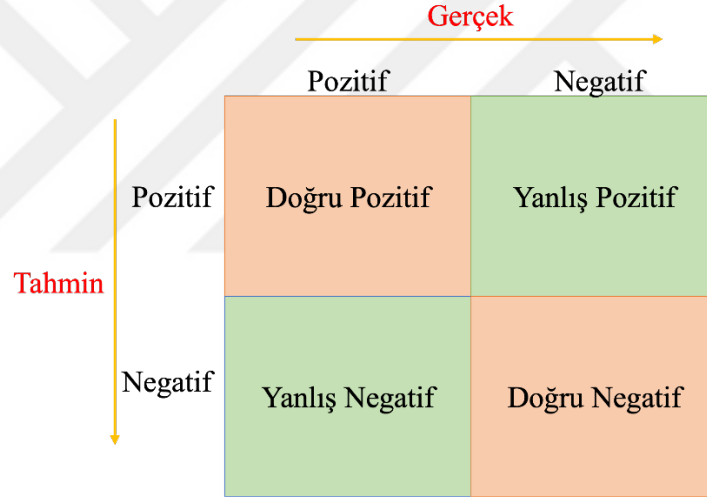
Tablo 3.3'te gösterildiği gibi ortalama hız değeri 0-20 km/sa aralığında ise "Kuyruklanma var", 21-55 km/sa aralığında ise "Hafif kuyruklanma" ve 55 km/sa üzerinde ise "Kuyruklanma yok" şeklinde üç farklı kuyruklanma durum tespiti yapılmaktadır. Kuyruklanma durum tespiti için belirlenen ortalama hız koşulları trafikte test edilerek belirlenmiştir.

Tablo 3.3. Kuyruklanma koşulları

Kuyruklanma Türü	Ortalama Hız Koşulu (km/sa)
Kuyruklanma var	$0 \leq \text{Hız} \leq 20$
Hafif kuyruklanma	$21 \leq \text{Hız} \leq 55$
Kuyruklanma yok	$56 \leq \text{Hız}$

### 3.7. Değerlendirme Kriterleri

Eğitim işlemi sonucunda elde edilen modelin değerlendirme aşamasında kesinlik (precision), duyarlılık (recall), F1 skor ve ortalama hassasiyet (mean average precision, mAP) değerleri kullanılmaktadır. Şekil 3.11’de değerlendirme kriterlerinde kullanılan karmaşıklık matrisi verilmektedir. Karmaşıklık matrisi makine öğrenmesi ve sınıflandırma yapılırken kullanılan bir tablodur. Bu tablo, bir modelin performans değerlendirmesinde etkili olmaktadır.



Şekil 3.11. Karmaşıklık matrisi

Karmaşıklık matrisi incelendiğinde dört farklı (Doğru pozitif, yanlış negatif, yanlış pozitif, gerçek negatif) değer bulunmaktadır.

Doğru pozitif (True Positive, TP): Gerçekte pozitif olan bir durumun pozitif olarak tahmin edilmesidir. Yanlış Negatif (False Negatif, FN): Mevcutta var olan pozitif durumun negatif olarak tahmin edilmesidir. Yanlış Pozitif (False Positive, FP): Mevcutta var olan negatif durumun pozitif olarak tahmin edilmesidir. Doğru Negatif (True Negative, TN): Gerçekte negatif olan durumun negatif olarak tahmin edilmesidir.

Kesinlik, doğru pozitiflerin toplam pozitif tahminlere oranını göstermektedir (Denklem (3.6)).

$$Kesinlik = \frac{Gerçek\ pozitif}{Gerçek\ pozitif + Yanlış\ pozitif} \quad (3.6)$$

Duyarlılık, doğru pozitiflerin toplam gerçek pozitiflere oranını göstermektedir (Denklem (3.7)).

$$Duyarlılık = \frac{Gerçek\ pozitif}{Gerçek\ pozitif + Yanlış\ negatif} \quad (3.7)$$

F1 skor, kesinlik ve duyarlılığın harmonik ortalamasını göstermektedir (Denklem (3.8)).

$$F1\ Skor = 2 \times \left( \frac{Kesinlik \times Duyarlılık}{Kesinlik + Duyarlılık} \right) \quad (3.8)$$

mAP, nesne algılama algoritmalarının performansını değerlendirmek için kullanılan temel bir ölçüttür. Bu ölçüt, bir algılama modelinin nesnelere doğru bir şekilde tespit edip sınıflandırma başarısını değerlendirmek için tercih edilir.

YOLOv5 modeli ile araç tespiti, şerit bazlı araç sayımı ve hız hesaplamasında bulunan değerlerin doğruluk değerlendirmesi [38] Denklem (3.9) ve Denklem (3.10)'da gösterilen formül ile yapılmaktadır. İlk olarak, modelin tahmin ettiği araç sayısı ile videodaki gerçek araç sayısının farkı alınmaktadır ve videodaki gerçek araç sayısına bölünmektedir.

$$Hata\ (\%) = \frac{|Tahmin - Gerçek|}{Gerçek} \times 100 \quad (3.9)$$

Bölme işleminden sonra da elde edilen sayı 100 ile çarpılarak hata oranı bulunmaktadır. Doğruluk oranı bulunurken de elde edilen hata oranı 100'den çıkarılmaktadır.

$$Doğruluk\ (\%100) = 100 - Hata\ (\%) \quad (3.10)$$

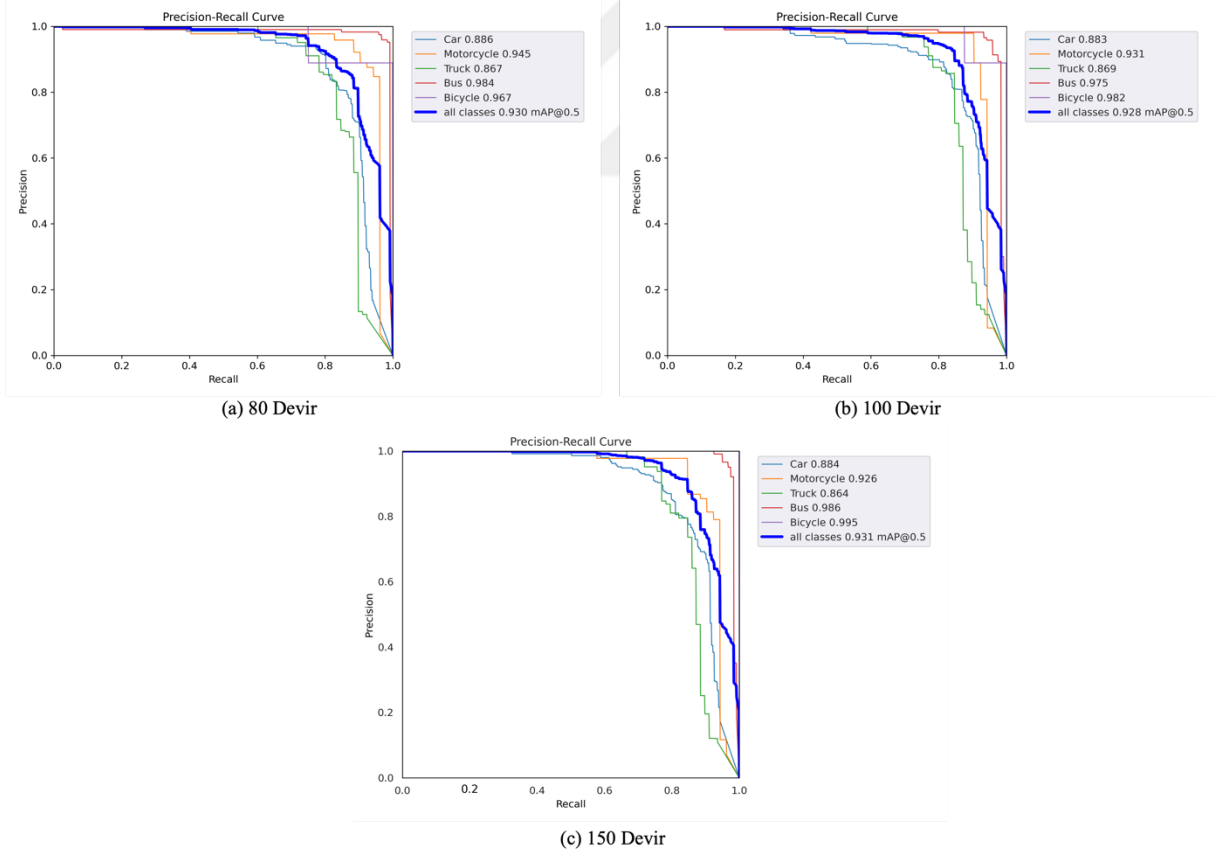


#### 4. DENEYSEL ÇALIŞMALAR

Oluşturulan araç veri seti üzerinde YOLOv5'in m modeli kullanılarak üç farklı eğitim işlemi gerçekleştirilmiştir. Eğitim aşaması sırasıyla 80, 100 ve 150 devir değerlerinde gerçekleştirilmiştir. Eğitim aşamasında kullanılan parametreler Tablo 4.1'de gösterilmektedir. Şekil 4.1'de eğitim sonucunda oluşan her bir modelin kesinlik-duyarlılık grafiği gösterilmektedir.

Tablo 4.1. Veri setinin eğitim aşamasında kullanılan parametreler

Devir Sayısı	Öğrenme Oranı	Parti Boyutu
80	0.01	16
100	0.01	16
150	0.01	16



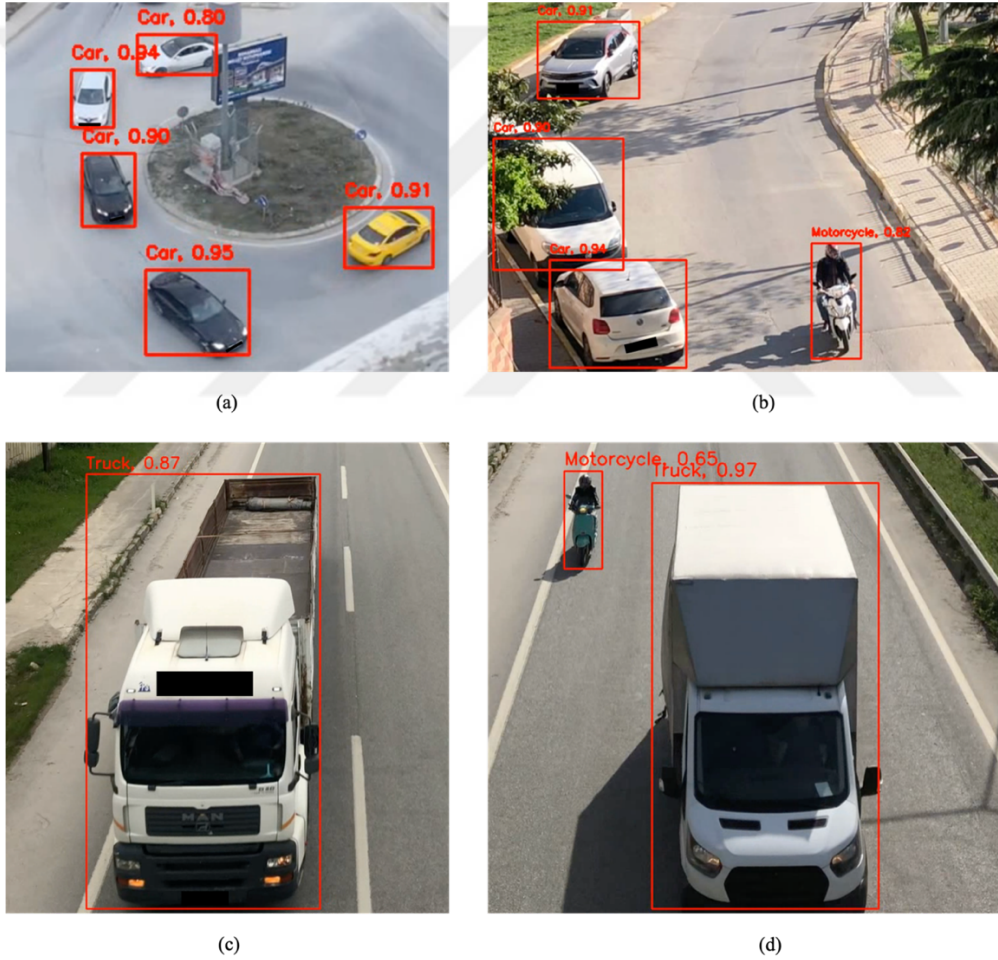
Şekil 4.1. Araç veri setinin eğitimi sonucunda oluşan modelin kesinlik-duyarlılık grafiği

Eğitim sonucunda elde edilen kesinlik, duyarlılık ve mAP@50 puan değerleri ise Tablo 4.2'de gösterilmektedir.

Tablo 4.2. Veri setinin eğitim aşaması sonucunda elde edilen araç sınıf tespiti performans sonuçları

Devir Sayısı	Kesinlik	Duyarlılık	mAP@50
80	%86	%91	%93
100	%94	%85	%92
150	%86	%87	%93

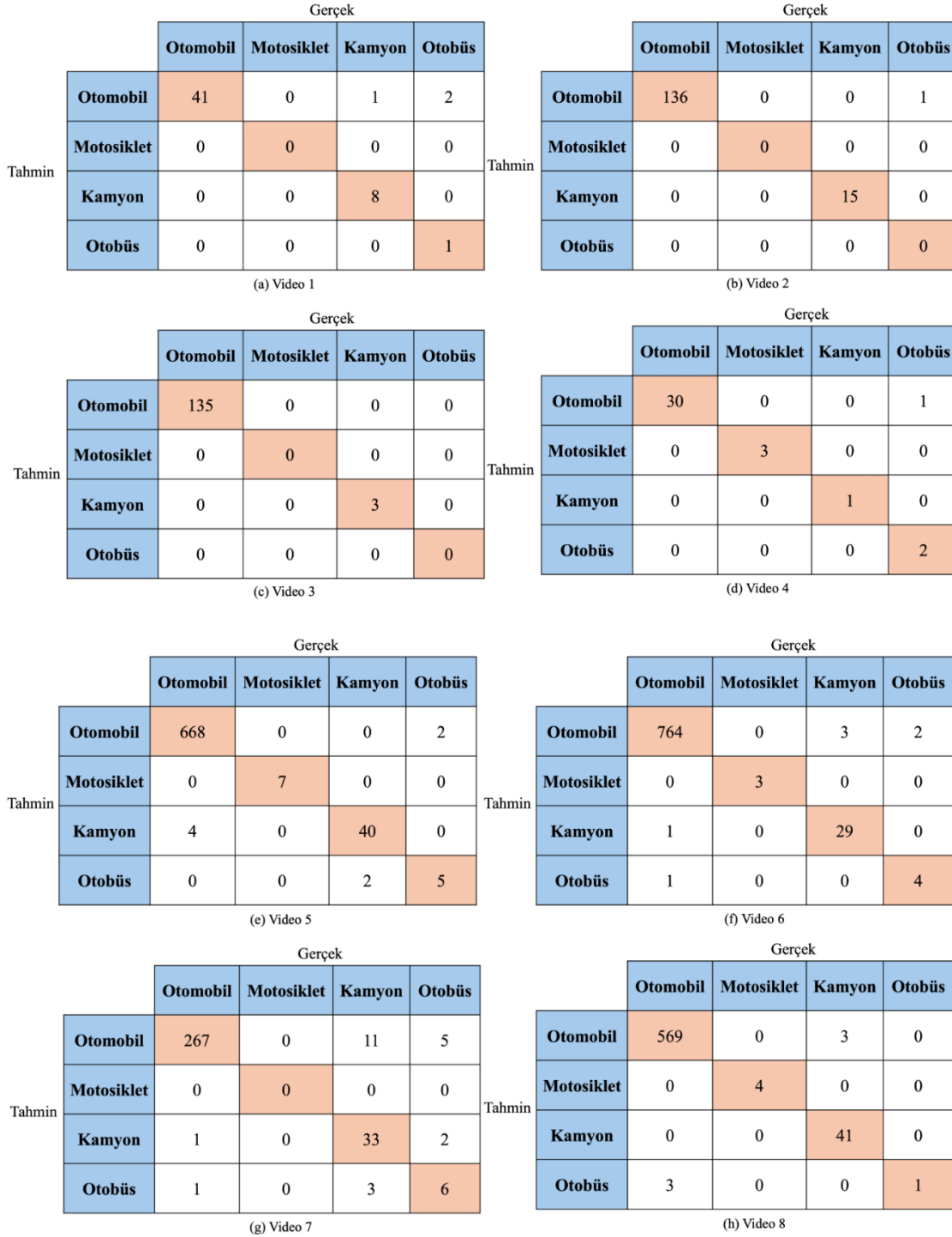
Grafikler ve modellerin testleri incelendiğinde en iyi tespit yapan modelin 150 devirde olduğu tespit edilmiştir. Şekil 4.2’de 150 devirde edilen modelin test aşamasından örnekler gösterilmektedir.



Şekil 4.2. Araç ve sınıf tespiti test aşamasından örnekler

Şekil 4.3'te sekiz farklı video üzerinde yapılan testler sonucunda modelin karmaşıklık matrisleri gösterilmektedir. Araç sınıf tespitinde kullanılan model bisiklet sınıfını tespit edebilmektedir fakat kuyruklanma tespiti yapılırken bisikletin herhangi bir etkisi olmadığı için videolarda

kuyruklanma tespiti aşamasında bisiklet sınıfı göz ardı edilmiştir. Bundan dolayı karmaşıklık matrisinde bisiklet sınıfı görülmektedir.



Şekil 4.3. Araç sınıf tespitine ait karmaşıklık matrisleri

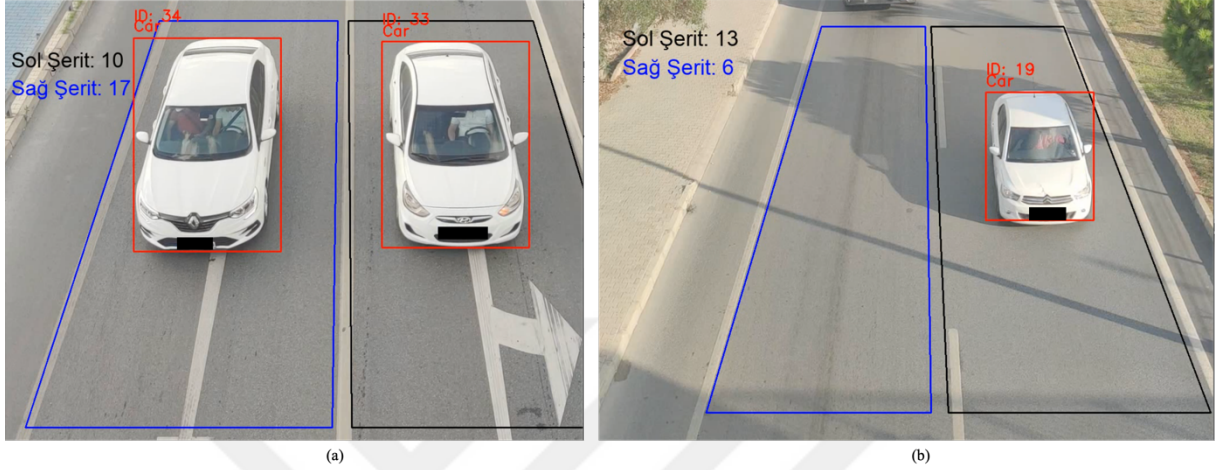
Tablo 4.3'te, Şekil 4.3'te gösterilen sekiz farklı video üzerinde yapılan testler sonucunda oluşan karmaşıklık matrislerinin araç tespit ve sınıflandırmada kesinlik, duyarlılık ve F1 skor değerleri verilmiştir. Tabloda sadece videoda bulunan sınıfların kesinlik, duyarlılık ve F1 skor değerleri

gösterilmiştir. Bu değerler Bölüm 3.7’deki “Değerlendirme Kriterleri” başlığı altındaki Denklem (3.6), Denklem (3.7) ve Denklem (3.8)’deki formüllerin kullanılması sonucunda elde edilen sonuçlardır.

Tablo 4.3. Karmaşıklık matrisi sonucunda hesaplanan performans değerleri

Video	Sınıf	Kesinlik	Duyarlılık	F1 Skor
Video 1	Otomobil	%93	%100	%96
	Kamyon	%100	%88	%94
	Otobüs	%100	%33	%50
Video 2	Otomobil	%99	%100	%100
	Kamyon	%100	%100	%100
Video 3	Otomobil	%100	%100	%100
	Kamyon	%100	%100	%100
Video 4	Otomobil	%96	%100	%98
	Motosiklet	%100	%100	%100
	Kamyon	%100	%100	%100
	Otobüs	%100	%66	%80
Video 5	Otomobil	%99	%99	%99
	Motosiklet	%100	%100	%100
	Kamyon	%91	%95	%93
	Otobüs	%71	%71	%71
Video 6	Otomobil	%99	%99	%99
	Motosiklet	%100	%100	%100
	Kamyon	%96	%90	%93
	Otobüs	%80	%66	%72
Video 7	Otomobil	%94	%99	%96
	Kamyon	%91	%70	%79
	Otobüs	%60	%46	%52
Video 8	Otomobil	%99	%99	%99
	Motosiklet	%100	%100	%100
	Kamyon	%100	%93	%96
	Otobüs	%25	%100	%40

Şerit bazlı araç sayım aşamasında ilgili bölgelerden (şeritlerden) geçen araçların sınıfları belirlendikten sonra sayım işlemi gerçekleştirilmiştir. İlgili şeritteki araçlar tespit edildikten sonra SORT algoritması kullanılarak her bir araca benzersiz bir ID verilmektedir. Şekil 4.4'te şerit bazlı araç sayımına örnekler gösterilmektedir. Test aşamasında hem kendi çekimimiz olan videolar hem de internet kaynaklarından elde edilen videolar kullanılmıştır.



Şekil 4.4. Şerit bazlı araç sayımı

Tablo 4.4'te test videolarından elde edilen sonuçlar gösterilmektedir. Tabloda “Videodaki Toplam Araç Sayısı” tüm şeritlerden geçen araçların toplam sayısını, “Tahmin Edilen Araç Sayısı” test aşamasında tüm şeritlerde tahmin edilen araçların toplam sayısını göstermektedir. “Doğruluk Değeri” ise Bölüm 3.7'deki “Değerlendirme Kriterleri” başlığı altındaki Denklem (3.9) ve Denklem (3.10)'un kullanılması sonucunda hesaplanan doğruluk değerlerini göstermektedir. Örneğin, Video1'de gerçekte videoda toplam 33 araç geçmiştir ve test aşamasında 34 aracın geçtiği tespit edilmiştir. Denklem (3.9)'da hata oranı %3,03 olarak hesaplanmıştır. Doğruluk değeri ise Denklem (3.10)'daki formül kullanılarak %96,97 olarak hesaplanmıştır.

Tablo 4.4. Araç sayım aşaması ile ilgili sonuçlar

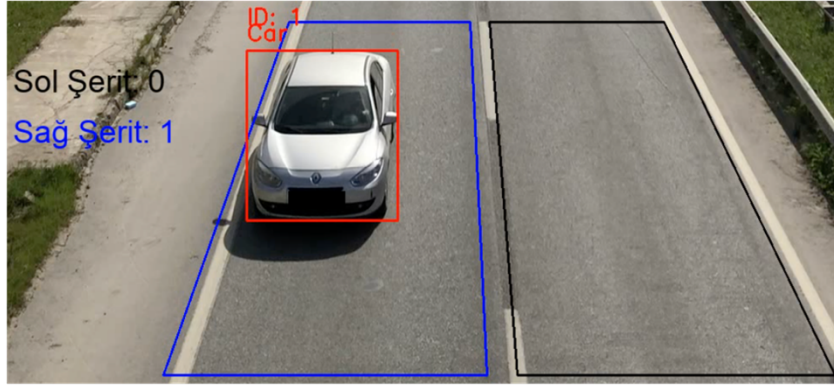
Video	Videodaki Toplam Araç Sayısı	Tahmin Edilen Araç Sayısı	Doğruluk Değeri (%)
Video 1	33	34	%96,96
Video 2	41	41	%100
Video 3	52	48	%92,30
Video 4	29	28	%96,55

Hız tespit aşamasındaki testlerde hızları bilinen araçların yer aldığı videolar kullanılmıştır. Videolar, kamera açısına göre aracın hem yatay yönde hareket ettiği hem de dikey yönde hareket ettiği görüntüleri içermektedir. Hız tespitinde, ilgili bölgeden geçen aracın ilk tespit edildiği andaki merkez koordinatı ile ilgili bölgeden çıktığı andaki son merkez koordinatı, harcadığı süre, ilk tespit edildiği andaki uzunluğu (piksel olarak), Öklid mesafesi hesaplandıktan sonra tahmini hız bulunmakta ve video bittiği anda tüm bilgiler txt dosyasına kaydedilmektedir. Hız hesaplamasında ilk olarak kamera kalibrasyonu (Denklem (3.1)) ardından merkez koordinatları arasındaki Öklid mesafesi hesaplanmaktadır (Denklem (3.2)). Öklid mesafesi kamera kalibrasyonu ile çarpılarak birimi metreye dönüştürülmektedir (Denklem (3.3)). Son olarak aracın video boyunca harcamış olduğu süre hesaplanarak (Denklem (3.4)) kat edilen yola bölünüp, 3,6 ile çarpılarak aracın hızı km/sa olarak tespit edilmiştir (Denklem (3.5)).

Tablo 4.5. Hız tespiti ile ilgili sonuçlar

Video	Gerçek Hız	Tahmini Hız	Doğruluk Değeri
Video 1	30 km/sa	30,83 km/sa	%97,23
Video 2	50 km/sa	46,51 km/sa	%93,02
Video 3	70 km/sa	70,60 km/sa	%99,14
Video 4	100 km/sa	89,11km/sa	%89,11
Video 5	40 km/sa	39,24 km/sa	%98,10
Video 6	50 km/sa	53,10 km/sa	%98,80
Video 7	10 km/sa	9,87 km/sa	%98,70

Şekil 4.5, Şekil 4.6, Şekil 4.7 ve Şekil 4.8’de videoya çekmiş olduğumuz araçların hız tespiti sonucu gösterilmiştir. Şekiller (a) ve (b) olarak ayrılmış ve (a) olanlarda aracın tespit edilme durumu, (b) olanlarda ise aracın ilk tespit edildiği andaki ve son andaki merkez koordinatı bilgisi, kat etmiş olduğu mesafe, ilk tespit edildiği andaki uzunluğu (piksel olarak) ve hangi şeritte olduğu bilgileriyle birlikte tespit edilen tahmini hızı gösterilmektedir. Şekil 4.5’teki aracın gerçekteki hızı 50 km/sa iken hesaplanan tahmini hız 46,51 km/sa olarak bulunmuştur.



(a)

```

*****
ID'lerin İlk ve Son Merkez Koordinatları ve Öklid Mesafesi:
Şerit: Sağ: ID 1, Sınıf İsmi: Car,
Merkez Koordinat (Başlangıç) = (484.5, 1317.5), Merkez Koordinat (Bitiş) = (426.0, 1569.5),
Aldığı Mesafe =258.70, Uzunluk : 159, Hız = 46.51 km/sa

Şerit: Sol: ID 2, Sınıf İsmi: Car,
Merkez Koordinat (Başlangıç) = (811.0, 1357.5), Merkez Koordinat (Bitiş) = (861.5, 1504.0),
Aldığı Mesafe =154.96, Uzunluk : 187, Hız = 80.55 km/sa

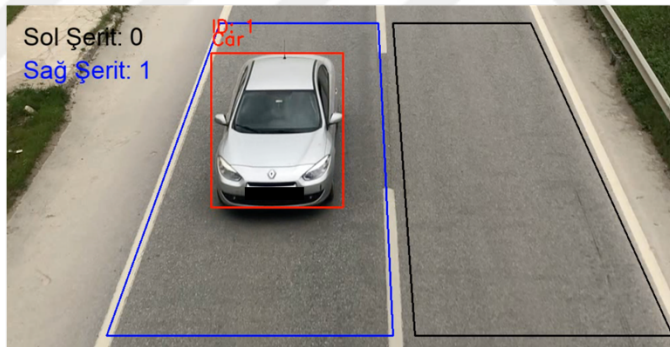
*****

```

(b)

Şekil 4.5. Gerçekte 50 km/sa ile giden aracın tahmini hız tespiti

Şekil 4.6'da aracın gerçekteki hızı 70 km/sa iken hesaplanan tahmini hız 70,60 km/sa olarak bulunmuştur.



(a)

```

*****
ID'lerin İlk ve Son Merkez Koordinatları ve Öklid Mesafesi:
Şerit: Sağ: ID 1, Sınıf İsmi: Car,
Merkez Koordinat (Başlangıç) = (497.0, 1001.0), Merkez Koordinat (Bitiş) = (427.0, 1382.5),
Aldığı Mesafe =387.87, Uzunluk : 178, Hız = 70.60 km/sa

*****

```

(b)

Şekil 4.6. Gerçekte 70 km/sa ile giden aracın tahmini hız tespiti

Şekil 4.7'deki aracın gerçekteki hızı 30 km/sa iken hesaplanan tahmini hız 30,83 km/sa olarak bulunmuştur.



(a)

```
*****  
ID'lerin İlk ve Son Merkez Koordinatları ve Öklid Mesafesi:  
Şerit: Diğer: ID 1, Sınıf İsmi: Car,  
Merkez Koordinatı (Başlangıç) = (398.0, 354.0), Merkez Koordinatı (Bitiş) = (571.5, 354.5),  
Aldığı Mesafe =173.50, Uzunluk : 152, Hız = 30.83 km/sa
```

```
*****
```

(b)

Şekil 4.7. Gerçekte 30 km/sa ile giden aracın tahmini hız tespiti

Şekil 4.8’de ise aracın gerçekteki hızı 50 km/sa iken hesaplanan tahmini hız 53,10 km/sa olarak bulunmuştur.



(a)

```
*****  
ID'lerin İlk ve Son Merkez Koordinatları ve Öklid Mesafesi:  
Şerit: Diğer: ID 1, Sınıf İsmi: Car,  
Merkez Koordinatı (Başlangıç) = (530.0, 335.5), Merkez Koordinatı (Bitiş) = (381.5, 332.0),  
Aldığı Mesafe =148.54, Uzunluk : 136, Hız = 53.10 km/sa
```

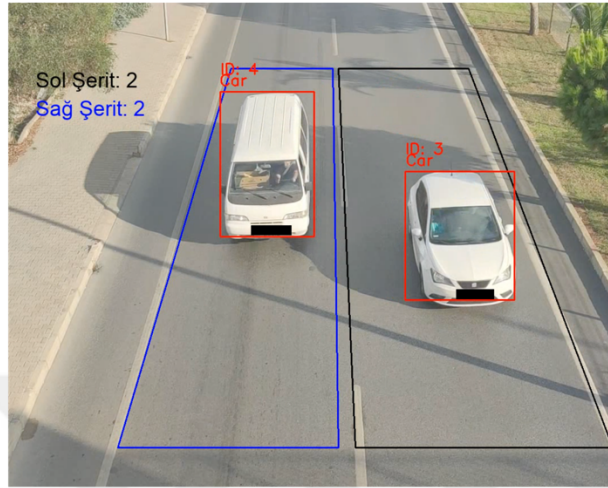
```
*****
```

(b)

Şekil 4.8. Gerçekte 50 km/sa ile giden aracın tahmini hız tespiti

İlgili şeritteki aracın tespiti, sınıfı, hangi şeritte olduğu ve hızı tahmin edildikten sonra, kuyruklanma durumu tespiti ilgili şeritten geçen son üç aracın ortalama hızına bakılarak yapılmaktadır. Kuyruklanma durumu tespitinin belirlenmesinde Bölüm 3.6’daki “Kuyruklanma Durum Tespiti” başlığı altındaki Tablo 3.4’teki kuyruklanma koşulları kullanılmıştır. Bu aşamada hem kendi çekimimiz olan videolar hem de internet kaynaklarından elde edilen videolar kullanılmıştır. Şekil 4.9’da kuyruklanma durumu ile ilgili bir örnek gösterilmiştir. Video trafik olağan akışında devam etmekte ve kuyruklanma veya hafif

kuyruklanma durumu bulunmamaktadır. Video bittikten sonra ortalama hız, şerit bilgileri ve kuyruklanma durumu hakkındaki bilgiler .txt formatlı dosyaya yazdırılmaktadır. Şekil 4.9 (b)'de ilgili videonun çıktısına ait bir örnek gösterilmektedir. Sonuçlar incelendiğinde sol şeritte ve sağ şeritte kuyruklanma olmadığı tespit edilmiştir.



(a)

```

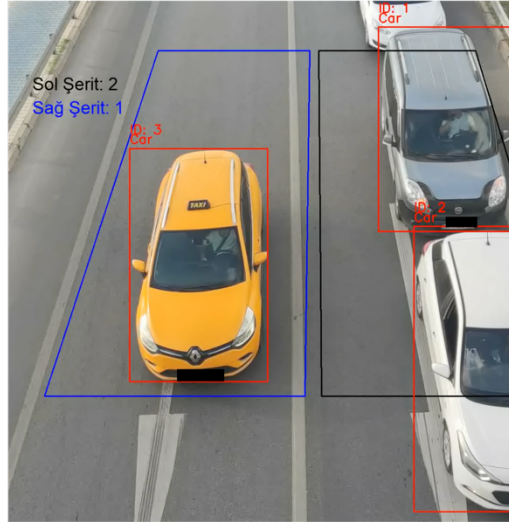
Şerit: Sol, Listede 27 - 28 - 29. araçların ortalama hızı: 45.55 km/sa
Şerit: Sol, Kuyruklanma yok.
Şerit: Sol, Listede 28 - 29 - 30. araçların ortalama hızı: 44.01 km/sa
Şerit: Sol, Kuyruklanma yok.
Şerit: Sağ, Listede 1 - 2 - 3. araçların ortalama hızı: 59.39 km/sa
Şerit: Sağ, Kuyruklanma yok.
Şerit: Sağ, Listede 2 - 3 - 4. araçların ortalama hızı: 58.83 km/sa
Şerit: Sağ, Kuyruklanma yok.
Şerit: Sağ, Listede 3 - 4 - 5. araçların ortalama hızı: 58.70 km/sa
Şerit: Sağ, Kuyruklanma yok.
Şerit: Sağ, Listede 4 - 5 - 6. araçların ortalama hızı: 59.93 km/sa
Şerit: Sağ, Kuyruklanma yok.
Şerit: Sağ, Listede 5 - 6 - 7. araçların ortalama hızı: 48.78 km/sa
Şerit: Sağ, Kuyruklanma yok.

```

(b)

Şekil 4.9. “Kuyruklanma yok” durum tespiti

Şekil 4.10’da kuyruklanma durumu ile ilgili başka bir örnek gösterilmiştir. Videoda gerçekte sol şeritte hafif kuyruklanma durumu varken sağ şeritte ise kuyruklanma durumu bulunmamaktadır. Şekil 4.10 (b)’de kuyruklanma durumunun tespitine yönelik sonuçlar gösterilmektedir. Sonuçlar incelendiğinde sağ şeritte hafif kuyruklanma ve kuyruklanma yok, sol şeritte ise kuyruklanma olduğu tespit edilmiştir. Sağ şeritte sadece bir tane aracın görünmesine rağmen hafif kuyruklanma olarak tespit edilmesinin sebebi aracın hızının azalmasıdır. Çünkü ilgili araçtan önce ve sonra sağ şeritten giden araçların olduğu bilinmektedir.



(a)

```

Şerit: Sol, Listede 6 - 7 - 8. araçların ortalama hızı: 15.60 km/sa
Şerit: Sol, Kuyruklanma var!!!
Şerit: Sol, Listede 7 - 8 - 9. araçların ortalama hızı: 17.07 km/sa
Şerit: Sol, Kuyruklanma var!!!
Şerit: Sol, Listede 8 - 9 - 10. araçların ortalama hızı: 16.35 km/sa
Şerit: Sol, Kuyruklanma var!!!
Şerit: Sağ, Listede 1 - 2 - 3. araçların ortalama hızı: 46.18 km/sa
Şerit: Sağ, Kuyruklanma yok.
Şerit: Sağ, Listede 2 - 3 - 4. araçların ortalama hızı: 44.51 km/sa
Şerit: Sağ, Kuyruklanma yok.
Şerit: Sağ, Listede 3 - 4 - 5. araçların ortalama hızı: 34.21 km/sa
Şerit: Sağ, Hafif kuyruklanma!
Şerit: Sağ, Listede 4 - 5 - 6. araçların ortalama hızı: 29.32 km/sa
Şerit: Sağ, Hafif kuyruklanma!

```

(b)

Şekil 4.10. “Hafif kuyruklanma”, “Kuyruklanma yok” ve “Kuyruklanma var” durum tespiti

Kuyruklanma durum tespiti test aşamasında eğer yeterli veri yoksa diğer bir ifade ile şeritten üçten daha az araç geçmişse veya hiç araç geçmemişse video bittikten sonra “Kuyruklanma yok” uyarısı verilmektedir.

```

*****
ID'lerin İlk ve Son Merkez Koordinatları ve Öklid Mesafesi:
Şerit: Sağ: ID 1, Sınıf İsmi: Car,
Merkez Koordinat (Başlangıç) = (497.0, 1001.0), Merkez Koordinat (Bitiş) = (427.0, 1382.5),
Aldığı Mesafe =387.87, Uzunluk : 178, Hız = 70.60 km/sa
*****

Sol Şerit Hızları:

Sağ Şerit Hızları:
70.60084807400756

*****
Ortalama Hız ve Kuyruklanma Durumu:
Şerit: Sol, Kuyruklanma yok!
Şerit: Sağ, Kuyruklanma yok!

```

Şekil 4.11. Az araç geçişi için “Kuyruklanma yok” durum tespiti

## 5. SONUÇLAR VE ÖNERİLER

Yapay zeka alanındaki hızlı gelişmeler, trafikle ilgili alanlarda da çeşitli çalışmaların yapılmasına ve bu alandaki problemlerin çözümüne katkı sağlamıştır. Trafikteki araçların tespiti ve sınıflandırılması, araçların sayımı, araçların hızlarının tespiti, trafik yoğunluğuna bağlı olarak kuyruklanmaların olup olmadığının tespitine yönelik çalışmalar yapay zeka yöntemleri ile başarılı bir şekilde gerçekleştirilebilir hale gelmiştir. Sensörlerden ve kameralardan gelen gerçek zamanlı verilerin işlenerek trafik akışının düzenlenmesi ve buna bağlı olarak olası problemlerin üstesinden gelenebilmesi yapay zeka tabanlı çalışmalara daha da fazla ağırlık verilmesi gerektiğini ortaya koymaktadır.

Tez çalışmasında trafikteki araç kuyruklanmalarının derin öğrenme ile video görüntüleri üzerinden gerçek zamanlı tespitine yönelik bir yaklaşım sunulmaktadır. Bu yaklaşımda araçların hızlarına bağlı olarak trafikteki kuyruklanma durum tespiti yapılmaktadır. Kuyruklanma durum tespitine yönelik olan çalışma dört aşamadan oluşmaktadır. İlk aşama, ilgili trafik videolarında akan araçların tespit edilmesi, ikinci aşama, araçların şerit bazlı olarak sayımının gerçekleştirilmesi, üçüncü aşama, şerit bazlı sayımı yapılan araçların hızlarının tespit edilmesi ve son aşama ise hızları ve sayıları belirlenen araçların trafikte oluşturmuş olduğu kuyruklanma durumunun tespit edilmesini içermektedir. Kuyruklanma durum tespiti için; “Hafif kuyruklanma”, “Kuyruklanma yok” ve “Kuyruklanma var” sınıfları belirlenmiştir. Bu durumlar akan trafikteki araçların ortalama hızlarına göre belirlenmektedir. Tez çalışmasında video görüntülerindeki araç tespiti ve sınıflandırılması için YOLOv5 algoritması ve hareket halindeki araçların sayımı ve takibi için SORT algoritması kullanılmıştır. Araç kuyruklanma durumu modelinin performans başarısının ölçülmesi için hem kendi çekimimiz olan videolar hem de internet kaynaklarından elde edilen videolar kullanılmıştır. Deneysel sonuçlar dikkate alındığında modelin, video görüntüleri üzerinde araçların tespitini ve sınıflandırılmasını, araç hızlarının tespitini ve hızlara bağlı olarak kuyruklanma durumu tespitini başarılı bir şekilde yapabildiği görülmektedir.

Elde edilen bulgular trafik yönetimi ve iyileştirme çalışmaları için önemli veriler sunmaktadır. Araç sınıfı, sayısı, hızı ve kuyruklanma durumu verileri trafik sıkışıklığının nedenlerini ve çözüm yollarını belirlemek için kritik öneme sahiptir. Bu veriler ilgili uzmanlar tarafından kullanılabilir ve trafik akışının daha verimli hale getirilmesi için stratejik planlamalarda değerlendirilebilir.



## KAYNAKLAR

- [1] L. Xiaomeng, F. Jun, and C. Peng, "Vehicle detection in traffic monitoring scenes based on improved YOLOv5s," *2022 International Conference on Computer Engineering and Artificial Intelligence (ICCEAI)*, 2022, pp. 467-471.
- [2] A. Wang, "Vehicle recognition algorithm based on improved YOLOv3," *2020 IEEE 9th Joint International Information Technology and Artificial Intelligence Conference (ITAIC)*, 2020, pp. 2301-2305.
- [3] X. Sun, Q. Huang, Y. Li, and Y. Huang, "An improved vehicle detection algorithm based on YOLOv3," *2019 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCLOUD/SocialCom/SustainCom)*, 2019, pp. 1445-1450.
- [4] A. Battal, Y. E. Avcı, and A. Tuncer, "Vehicle detection and counting in traffic videos using deep learning," *ICENTE '23*, Konya, 2023, pp. 272.
- [5] J. M. Anil, L. Mathews, R. Renji, R. M. Jose, and S. Thomas, "Vehicle counting based on convolution neural network," *Proceedings of the 7th International Conference on Intelligent Computing and Control Systems (ICICCS-2023)*, 2023, pp. 695-699.
- [6] N. Algiriyage, R. Prrasanna, E. E. H. Doyle, K. Stock, D. Johnston, M. Punchihewa, and S. Jayawardhana, "Towards real-time traffic flow estimation using YOLO and SORT from surveillance video footage," *AI and Intelligent Systems for Crises and Risks Proceedings of the 18th ISCRAM Conference*, Blacksburg, VA, USA, May 2021, pp. 40-48.
- [7] M. A. B. Zuraimi, and F. H. K. Zaman, "Vehicle detection and tracking using YOLO and DeepSORT," *2021 IEEE 11th IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE)*, 2021, pp. 23-29.
- [8] N. Seenouvong, U. Watchareeruetai, C. Nuthong, and K. Khongsomboon, "A computer vision based vehicle detection and counting system," *2016 8th International Conference on Knowledge and Smart Technology (KST)*, 2016, pp. 224-227.
- [9] V. Shaweddy, and W. Wahyono, "Vehicle counting framework for intelligent traffic monitoring system," *2019 5th International Conference on Science and Technology (ICST)*, Yogyakarta, Indonesia, 2019, pp. 1-5.
- [10] D. Sharma, S. Sharma, and V. Bhatnagar, "Automated vehicle speed estimation and license plate detection for smart cities development," *2022 IEEE World Conference on Applied Intelligence and Computing (AIC)*, 2022, pp. 378-383.
- [11] P. K. Thadagoppula, and V. Upadhyaya, "Speed detection using image processing," *2016 International Conference on Computer, Control, Informatics and its Applications (IC3INA)*, 2016, pp. 11-16.
- [12] T. Rahman, M. A. L. Siregar, A. Kurniawan, S. Juniastuti, and E. M. Yuniarno, "Vehicle speed calculation from drone video based on deep learning," *2020 International*

*Conference on Computer Engineering, Network and Intelligent Multimedia (CENIM 2020)*, 2020, pp. 229-233.

- [13] R. M. Waregaonkar, and S. M. Deokar, S. S. Patankar, and J. V. Kulkarni, "Development of prototype for vehicle speed measurement," *2017 2nd IEEE International Conference On Recent Trends in Electronics Information & Communication Technology (RTEICT)*, India, 2017, pp. 945-949.
- [14] K. Kumar, M. T. Talluri, B. Krishna, and V. Karthikeyan, "A novel approach for speed estimation vehicle detection counting," *2022 IEEE Students Conference on Engineering and Systems (SCES)*, Prayagraj, India, 2022, pp. 1-5.
- [15] M. Umair, M. U. Farooq, R. H. Raza, Q. Chen, B. Abdulhai, "Efficient video-based vehicle queue length estimation using computer vision and deep learning for an urban traffic scenario," *Processes*, vol. 9, pp. 1786, 2021.
- [16] W. A. Okaishi, A. Zaarane, I. Slimani, I. Atouf, M. Benrabh, "A vehicular queue length measurement system in real-time based on SSD network," *Transport and Telecommunication Journal*, vol. 22, no. 1, pp. 29-38, 2021.
- [17] A. Albiol, A. Albiol, and J. M. Mossi, "Video-based traffic queue length estimation", *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, Barcelona, Spain, 2011, pp. 1928-1932.
- [18] J. Zhu, Y. Ai, and B. Tian, "Real-time vehicle queue estimation of large-scale traffic scene", *In 2017 2nd International Conference on Image, Vision and Computing (ICIVC)*, Chengdu, China, 2017, pp. 1160-1165.
- [19] Y. Cai, W. Zhang, and H. Wang, "Measurement of vehicle queue length based on video processing in intelligent traffic signal control system", *In 2010 International Conference on Measuring Technology and Mechatronics Automation*, Changsha, China, 2010, pp. 615-618.
- [20] M. S. Shirazi, and B. Morris, "Vision-based vehicle queue analysis at junctions", *In 2015 12th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, Karlsruhe, Germany, 2015, pp. 1-6.
- [21] Z. Qi, M. Li, C. Liu, M. Zhao, and M. Long, "A measurement method for vehicle queue length of intersection based on image processing", *In 2018 Eighth International Conference on Image Processing Theory, Tools and Applications (IPTA)*, Xi'an, China, 2018, pp. 1-6.
- [22] S. Lee, K. Xie, D. Ngoduy, and M. Keyvan-Ekbatani, "An advanced deep learning approach to real-time estimation of lane-based queue lengths at a signalized junction," *Transportation Research Part c: Emerging Technologies*, 109, pp. 117-136, 2019.
- [23] R. Rahman, and S. Hasan, "Real-time signal queue length prediction using long short-term memory neural network," *Neural Computing and Applications*, vol. 33, pp. 3311-3324, 2021.

- [24] Y. Zhao, J. Zheng, W. Wong, X. Wang, Y. Meng, and H. X. Liu, "Various methods for queue length and traffic volume estimation using probe vehicle trajectories," *Transportation Research Part C: Emerging Technologies*, 107, pp. 70-91, 2016.
- [25] Ultralytics, YOLOv5, GitHub, <https://github.com/ultralytics/yolov5?tab=readme-ov-file>, (Erişim Tarihi: 10 Şubat 2023).
- [26] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: unified, real-time object detection," *In Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779-788.
- [27] H. Liu, "SF-YOLOv5: A lightweight small object detection algorithm based on improved feature fusion mode," *Sensors*, vol. 22, no. 15, pp. 2-14, 2022.
- [28] A. Gaspar, D. Oliva, E. Cuevas, D. Zaldívar, M. Pérez, and G. Pajares, "Hyperparameter optimization in a convolutional neural network using metaheuristic algorithms," *In Metaheuristics in machine learning: Theory and applications*, pp. 37-59, 2021.
- [29] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," *In 2016 IEEE International Conference on Image Processing (ICIP)*, Phoenix, AZ, USA, pp. 3464-3468.
- [30] R. E. Kalman, "A new approach to linear filtering and prediction problems", *Journal of Basic Engineering*, vol. 82, pp. 35-45, 1960.
- [31] H. W. Kuhn, "The Hungarian method for the assignment problem", *Naval Research Logistics Quarterly*, vol. 2, pp. 83-97, 1955.
- [32] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *in Advances in Neural Information Processing Systems*, 2015.
- [33] M. Boneh, Vehicle Detection, GitHub, <https://github.com/MaryamBoneh/Vehicle-Detection/tree/main/Dataset>, (Erişim Tarihi: 12 Şubat 2023).
- [34] L. Soetanto, Vehicle Images Dataset, Kaggle, <https://www.kaggle.com/datasets/lyensoetanto/vehicle-images-dataset>, (Erişim Tarihi: 21 Şubat 2023).
- [35] M. TÁRNOK, Kaggle, <https://www.kaggle.com/datasets/mrtontrnok/5-vehichles-for-multicategory-classification>, (Erişim Tarihi: 6 Haziran 2024).
- [36] Human Signal, labelImg, GitHub, <https://github.com/HumanSignal/labelImg>, (Erişim Tarihi: 18 Haziran 2024).
- [37] C. J. Lin, S. Y. Jeng, and H. W. Lion, "A real-time vehicle counting, speed estimation, and classification system based on virtual detection zone and YOLO," *Mathematical Problems in Engineering*, vol. 2021, no. 1, pp. 1-10, 2021.
- [38] A. M. Santos, C. J. A. Bastos-Filho, A. M. A. Maciel, and E. Lima, "Counting vehicle with high-precision in brazilian roads using YOLOv3 and Deep SORT," *2020 33rd SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*, 2020, pp. 69-76.



## **ÖZGEÇMİŞ**

2016 yılında girdiği Yalova Üniversitesi Bilgisayar Mühendisliği Bölümünden 2021 yılında mezun oldu. 2021 yılında Yalova Üniversitesi Bilgisayar Mühendisliği Anabilim Dalında yüksek lisans yapmaya başladı. 2024 yılından beri Alanya Alaaddin Keykubat Üniversitesi Bilgisayar Mühendisliği Bölümünde araştırma görevlisi olarak çalışmaktadır.

## **TEZDEN TÜRETİLEN YAYIN VE ESERLER**

- [1] A. Battal, Y. E. Avcı, A. Tuncer, “Vehicle detection and counting in traffic videos using deep learning,” ICENTE’23, Konya, 2023, pp. 272.

## **DİĞER YAYIN VE ESERLER**

- [1] A. Battal, A. Tuncer, “Detection of face mask wearing condition for COVID-19 using Mask R-CNN,” El-Cezerî Journal of Science Engineering, vol. 9, no. 2, pp. 1051-1060, 2022.
- [2] T. Uygunoğlu, A. Tuncer, A. Battal, “Detection of alkali silica reaction cracks on concrete from drone images using image processing methods, “Tokyo 7th International Innovate Studies of Contemporary Scientific Research Congress,” Tokyo, 2023, pp. 408-414.