

KARAYOLU BAKIŞ AÇILI GÖRÜNTÜ İLE DERİN ÖĞRENME TABANLI ARAÇ ALGILAMA-SAYMA, TRAFİK YOĞUNLUĞU HESABI YAPILMASI VE GOOGLE HARİTALARA ANLIK VERİ SUNMA SİSTEMİ

DEEP LEARNING BASED VEHICLE DETECTION, VEHICLE COUNTING, TRAFFIC DENSITY CALCULATION AND INSTANT DATA DELIVERY SYSTEM TO GOOGLE MAPS WITH HIGHWAYS IMAGE.

Abdullah Sökülmez*

Ankara, abdullah.sokulmez@trt.net.tr

Özet

Karayollarında kullanılan geleneksel araç yoğunluk hesaplama yöntemleri maliyet kurulum ve zaman gerektirmektedir. Bu faktörlerin sonucu olarak bu sistemler tüm karayollarında yaygın değildir. Bu nedenle çoğu karayolunda araç sayısı, yoğunluk gibi bilgilere erişilememektedir. Karayolu video görüntüleri ile akıllı araç tespiti ve sayımı gün geçtikçe daha önemli hale gelmektedir. Bu yöntem maliyet ve kurulum gerektirmeyen hali hazırda yol perspektifine monte edilmiş karayolları kamera verilerini kullanarak araç tespiti yapabilmektedir. Ancak hem kamera açısı sebebiyle hem de araçların kendine öz boyut farklılıkları, araç tespitini ve doğruluğunu doğrudan etkileyen bir sorun olmaya devam etmektedir. Bu makalede hem bu sorunu ele almak hem de karayolu üzerinde araç yoğunluğunun hesaplanarak internet erişiminin bulunduğu herhangi bir konumda bilgi alınabilmesini amaçlayan bir sistem önerilmiştir. Önerilen araç algılama, sayma ve trafik yoğunluğunu hesaplama sisteminde, öncelikle karayolu görüntüsünde sınırları belirlenmiş bir alan elde edilir. Ardından, tespit edilecek aracı ve konumunu tespit etmek için YOLOv3 ağı kullanılır. Son olarak nesne takibinde nesnenin sürekli görüntü dizileri arasında ki konumunu tahmin etmek için Deep SORT ağı kullanılmaktadır. Araç tespitinin doğruluğunu ve sürekliliğini artırmak amaçlı yapılan tüm bunların sonucunda görüntüsü alınan karayolunu kullanan araçlar tespit edilir. Ayrıca görüntü üzerinde sınırları belirlenmiş alanda bulunan araçların tespiti ile yol yoğunluğu hesaplanır. Programda tüm bu veriler hem çalışma ekranında hem de html dosyası oluşturularak Google Haritalar üzerinde yüzde yoğunluğa göre renklendirilerek gösterilmektedir.

Deneyisel sonuçlar, önerilen çalışmanın kullanılmasının, küçük araç nesnelerinin tespiti için yüksek algılama doğruluğunun sağlandığını göstermektedir. Dahası bu makalede sunulan çalışma, hem araç tespitinde hem de trafik yoğunluğunun anlık gösteriminde oldukça iyi performans gösteriyor. Bu makale, karayolu görüntüleri üzerinden verisel çıkarımlar yapmak ve bunları ileride farklı alanlarda kullanmak açısından pratik bir öneme sahiptir.

Anahtar Kelimeler: Görüntü işleme, Algoritma, Akıllı araç tespiti-izlemesi,yoğunluk.

Abstract

Traditional vehicle density calculation methods used on highways require cost setup and time. As a result of these factors these systems are not all common. For this reason, information such as the number of vehicles and density cannot be accessed on most highways. With highway video images, smart vehicle detection and counting is becoming more and more important day by day. While this method does not require cost and installation, it can detect vehicles using highway camera data already mounted in the road perspective. However, both due to the camera angle and the self-dimension differences of the vehicles, it continues to be a problem that directly affects the vehicle detection and accuracy. In this article, a program is proposed that aims to both address this problem and to obtain information in any location where internet access is available by calculating the vehicle density on the highway. In the proposed vehicle detection, counting and traffic density calculation system, first of all, a defined area in the highway image is obtained. The YOLOv3 network is then used to locate the vehicle to be detected and its location. Finally, Deep SORT network is used to estimate the position of the object between continuous image sequences in object tracking. As a result of all these done to increase the accuracy and continuity of the vehicle detection, the vehicles using the road, the image of which is taken, are detected. In addition, the road density is calculated by detecting the vehicles in the area whose boundaries are determined on the image. In the program, all these data are displayed both on the work screen and by creating an html file and coloring them according to percentage density on Google Maps.

Experimental results show that using the proposed study provides high detection accuracy for the detection of small vehicle objects. Moreover, the study presented in this article performs quite well both in vehicle detection and in instant display of traffic density. This article is of practical importance in making data inferences from road images and using them in different areas in the future.

Keywords: Image processing, Algorithm, Intelligent vehicle detection-tracking, density.

1. GİRİŞ

Son yüzyıldan günümüze dayanan nüfusun hızla artışı, kentleşme, kaynaklara duyulan ihtiyaç vb. faktörler ulaştırma sektörüne direkt olarak etki etmiş ve bilhassa 20. yy ikinci yarısından itibaren ulaştırma sistemleri üzerine yapılan çalışmaları artırmıştır.

Ulaşım ihtiyacının artışı ile Japonya, ABD ve Avrupa ülkelerinin önderliğinde teknolojik yol ağları ve üstün standartlara sahip yollar hizmete geçirilmiştir. Her gelişmenin yanında getirdiği gibi bu gelişmede bir sorun olarak, araç yoğunluğunun artarak sıkışık trafik oluşturmasını, enerji tüketiminde artışı, trafik kaza oranlarının artmasını, trafik sıkışıklığı ve uzun kuyrukları, park yeri sorunu, uzun seyahat süreleri gibi sorunları ortaya çıkarmıştır.

Ulaşım sistemlerinin avantajlarının yanında getirdiği bu dezavantajları ortadan kaldırmak veya azaltmak amacıyla Akıllı Ulaşım Sistemleri (AUS) kavramı türemiştir.

AUS günümüzde güncel veri bankalarını kullanan, birçok ulaştırma modu ile entegre edilmiş, elektrik-elektronik, iletişim, ve bilişim yapıları üzerine kurulmuş bir yapıdır[1], [2] .

AUS karayollarının anlık kamera görüntüleri ile izlenmesi, araç tespiti ve yoğunluk istatistikleri, trafiğin akıllı olarak yönetilmesi ve karayolunun kontrolü açısından büyük öneme sahiptir. Trafik gözetleme kamera sistemlerinin yaygın olarak kullanılması trafik analizi konusunda büyük bir veritabanı elde edilmesini sağlamaktadır. Genel olarak, yüksek bir görüş açısına sahip kameralar, bir yolun uzak mesafe yüzeylerine kadar görüş açısına sahiptirler[3]. Bu geniş görüş açısı görünen yol yüzeyini artırmakta ve kameralar üzerinden büyük alanda araç tespiti, sayımı ve bu verilerin işlenmesi ile birlikte farklı sistemlerin kurulmasını kolaylaştırmaktadır. Karayolları kameralarından alınan görüntüler derin öğrenme yöntemleri ile işlenebilir ve görüntü üzerinden çıkarımlar yapılabilir. Bu yöntemte görüntü işleme adı verilmektedir.

Görüntü işleme, görsel bir veriyi dijital forma dönüştürme ve veriden gelişmiş görüntü bilgileri almak amacıyla bazı işlemlerin gerçekleştirilmesi işlevidir. 1950’li yılların başlarında bilgisayar teknolojisinin gelişiminin bir sonucu olarak görüntü işleme teknolojisi meydana gelmiştir[4].

Görüntü işlemenin kullanıldığı alanlardan bazıları; akıllı ulaşım sistemleri, güvenlik sistemleri, biyomedikal görüntüleme sistemleri olmak üzere neredeyse görsel verinin olduğu her alanda uygulanabilmektedir.

Bu çalışmalarda veritabanı genellikle karayolu trafiği görüntü kaynakları olmakta ve bu veriler bilgisayar ortamında işlenmektedirler. Yol üzerinde bulunan araçların tespit edilmesi, hareket halinde olanların hız denetimi, araçlar arası takip mesafelerinin ve şerit tespitinin yapılması gibi sistemler kavşak ve karayollarında mevcuttur. Buna ek olarak, otopark gibi araçların durağan halde bulunduğu yerlerde park müsaitlik durumunun tespitinde de kullanılmaktadır.[5]

Bautista ve ark. yaptığı çalışmada düşük çözünürlüğe sahip karayolu görüntülerinde araç tespit etmek amacıyla görüntü işleme yapılarak bir model önerisinde bulunmuşlardır [6]. Elde edilen veriler, önerilen sistemin %94.72 sınıflandırma başarısına sahip olduğunu göstermiştir.

Bir diğer çalışmada ise, Çin’de kayıt altına alınan karayolu trafik videoları üzerinde, araç plaka tanıma ve görsel uyarı sistemine sahip, derin öğrenme yaklaşımını kullanan bir model önerilmiştir[7]. Çalışmada kullanılan veriler, haftanın tüm günü kayıt yapan 30 fps kare hızına sahip kameralar kullanılarak elde edilmiştir. Modelin sınıflandırma doğruluk oranı, %98.9 olarak bulunmuştur. Yapılan bir başka çalışmada ise, servis araçlarının en uygun rota belirlemesini sağlamak amacıyla genetik algoritma kullanılmıştır. Genetik algoritma bir görüntü işleme ve yapay zekâ yöntemidir[8]. Bu çalışma ile servis araçlarının trafikte bulunma sürelerini azaltarak ve en yoğun zamanlarda trafiğin hafifletilmesi amaçlanmıştır.

[9]’da yapılan çalışmada ise, otoyol trafik akışını tahmin etmeye çalışan bir model sunulmuştur. Çalışmada, highway system of China otoyol veri setleri kullanılmıştır. Tasarlanan model, sabit olan bir gözlem noktasında, gelecekteki zaman için oluşacak trafik yoğunluğunu tahmin etmektedir.

[10]’da ki çalışmada kısa vadeli karayolu trafik yoğunluğunu tahmin etmeyi amaçlamışlardır.

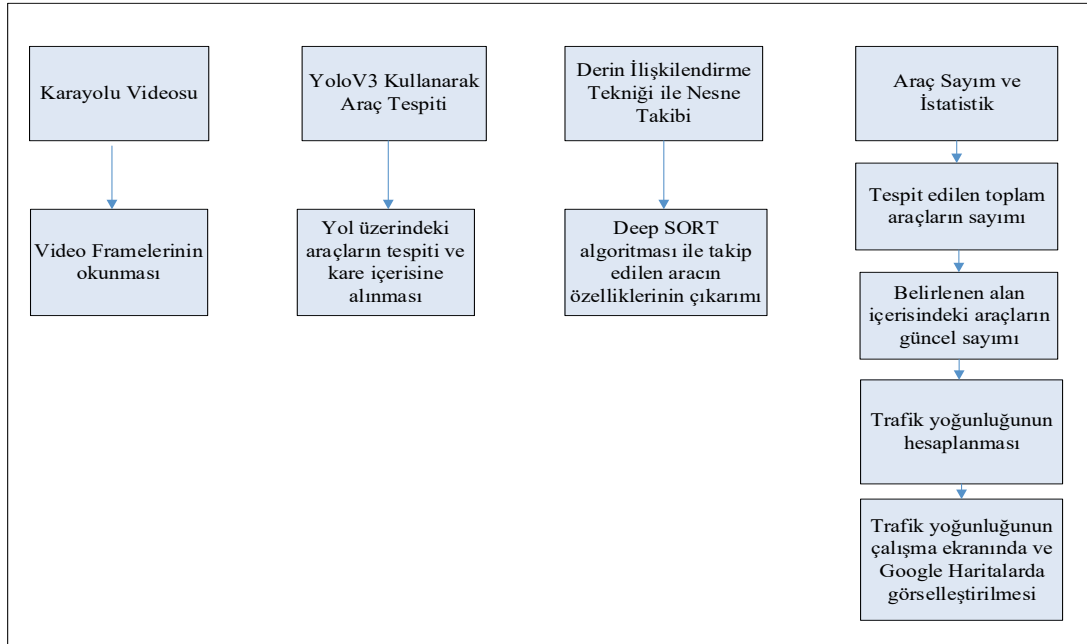
[11]’de yapılan çalışmada, trafik sıklığının otomatik tespitini sağlayan, böylece ekonomik ve çevresel kazançlar elde etmeyi planladıkları bir model önerilmiştir. Burada kullanılan veri seti, Londra SCOOT trafik merkezinden elde edilmiştir.

Bu makale görüntü işleme ve yapayzeka yöntemlerinden olan YOLOv3 algoritmasını Python programlama dilinde kullanarak oluşturulan, nesne algılama, nesne takibi ve hareket algılama sistemini anlatmaktadır. İstanbul Unkapanı köprüsü olduğu varsayılan kamera görüntüsü üzerinde, karayolunu kullanan toplam araç sayısı hesaplanmış, ayrıca karayolu görüntüsü içerisinde sınırları belirlenmiş alanda bulunan araçların miktarı ile trafik yoğunluğu hesaplanmıştır. Ayrıca video ile eş zamanlı çalışan ve Google Haritalara yönlendiren bir html uzantı dosyası oluşturulmuştur. Uzantı kullanılarak harita üzerinde Unkapanı Köprüsünün yoğunluk değeri (seyrekten yoğunla doğru) yeşil, sarı, turuncu ve kırmızı renkleriyle 20 saniye aralıklarla güncellenerek gösterilmiştir.

2. METOTLAR

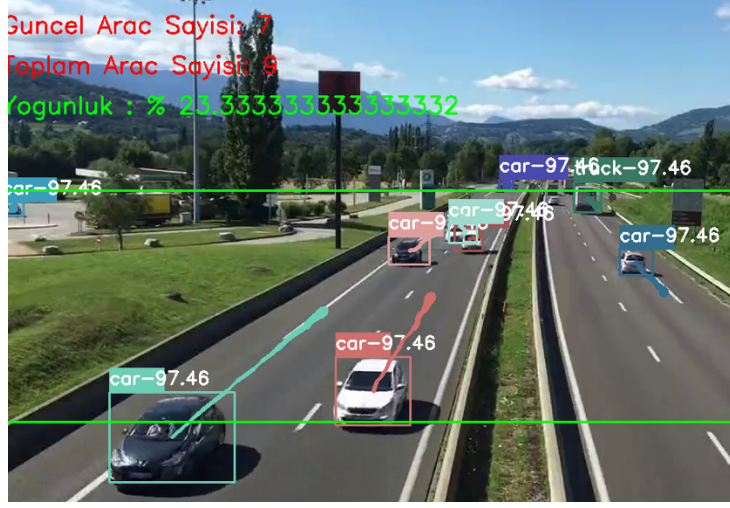
2.1 Sistem Mimarisi

Bu bölümde, araç algılama, sayma, yol yoğunluğunun bulunup Google haritalarda anlık olarak görüntüleme sisteminin ana yapısını açıklanmaktadır. Sistem yapısı şekil 1’de şematize edilmiştir.



Şekil 1. Sistem ana yapısı

Sistem mimarisi veri olarak karayolu videosunun framelerinin okunmasıyla başlar. Python programlama dili, Opencv görüntü işleme kütüphanesi kullanılarak görüntü işleme yapılır. Görüntü işleme aşamasında YOLOv3 (You Look Only Once)[12] derin öğrenme ve nesne algılama algoritması ve kullanılmıştır. Bu algoritma, küçük nesnelerin algılanmasını geliştirebilir ve nesnenin keskin ölçek değişikliğine uğraması durumlarında algılanamama sorununu çözebilmektedir[3]. Ayrıca nesne takibi, görüntü işleme sürecinde en önemli kısımlardan biridir. Bu makalede nesne takibini en yüksek doğrulukla gerçekleştirmek için sistem mimarisine Simple Online and Real-time Tracking with Deep Association Metric (Deep SORT) algoritması eklenmiştir. Deep SORT algoritması, tespit edilen nesne çerçevesinin özelliklerini çıkarır ve aynı nesne ile farklı video frameleri arasında bağlantı sağlamak için bu özellikleri eşleştirir. Son olarak toplam ve güncel araç sayısı hesaplanır. Güncel araç sayı verisi kullanılarak trafik yoğunluğu hesaplanır. Hesaplanan yoğunluk bilgisi programın çalışma ekranı ile birlikte bir html uzantılı dosya oluşturularak Google Haritalara aktarılır. Şekil 2 programın çalışma ekranını göstermektedir.

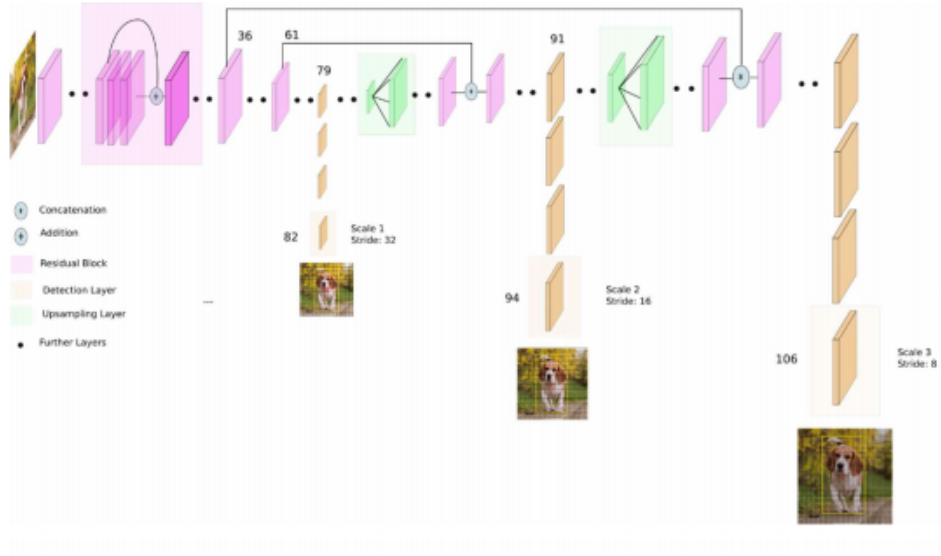


Şekil 2. Program çalışma ekranı

2.2 YOLOv3 kullanarak araç algılama

Bu bölüm, bu makalede kullanılan nesne algılama yöntemlerini açıklamaktadır. Karayolu araç algılama çerçevesinin uygulanmasında YOLOv3 algoritması kullanıldı. YOLOv3 algoritması, YOLO algoritmalarının ilk iki neslinin temel fikrini sürdürür. Evrişimli sinir ağı, giriş görüntüsünün özelliğini çıkarmak için kullanılır. 13×13 gibi özellik haritasının boyutuna göre, giriş görüntüsü 13×13 ızgaraya bölünür.

Nesne etiket kutusunun merkezi bir ızgara biçimindedir ve ızgara birimi nesneyi tahmin etmekle sorumludur. YOLOv3 algoritması Darknet-53 ağ yapısını benimsemiştir. Bu yapı, tam evrişim yöntemini benimser ve doğrudan bağlı evrişimli sinir ağının önceki versiyonunu bir önceki yapıyla değiştirir. [12]. Daha önceki yapının doğrudan öğrenilmesi, görüntü özelliği bilgilerinin bütünlüğünü sağlar, eğitimin karmaşıklığını basitleştirir ve ağın genel algılama doğruluğunu geliştirir. YOLOv3'te, her ızgara biriminde bir nesne için farklı ölçeklerde üç tarayıcı ızgara olmaktadır. Şekil 3.de görüldüğü üzere YOLOv3 çalışmasında istenilen nesne ile en büyük oranda örtüşen alana sahip aday kutu, nihai tahmin sonucu olacaktır. Ayrıca YOLOv3 küçük nesneleri tespit etmek için sıkı özellikler kullanılır ve büyük nesneleri algılamak için ise derin özellikler kullanılır; böylece ağ boyut değişimi olan nesneleri tanıyabilir.[3], [12]



Şekil 3. YOLOv3 sinir ağ çalışma prensibi

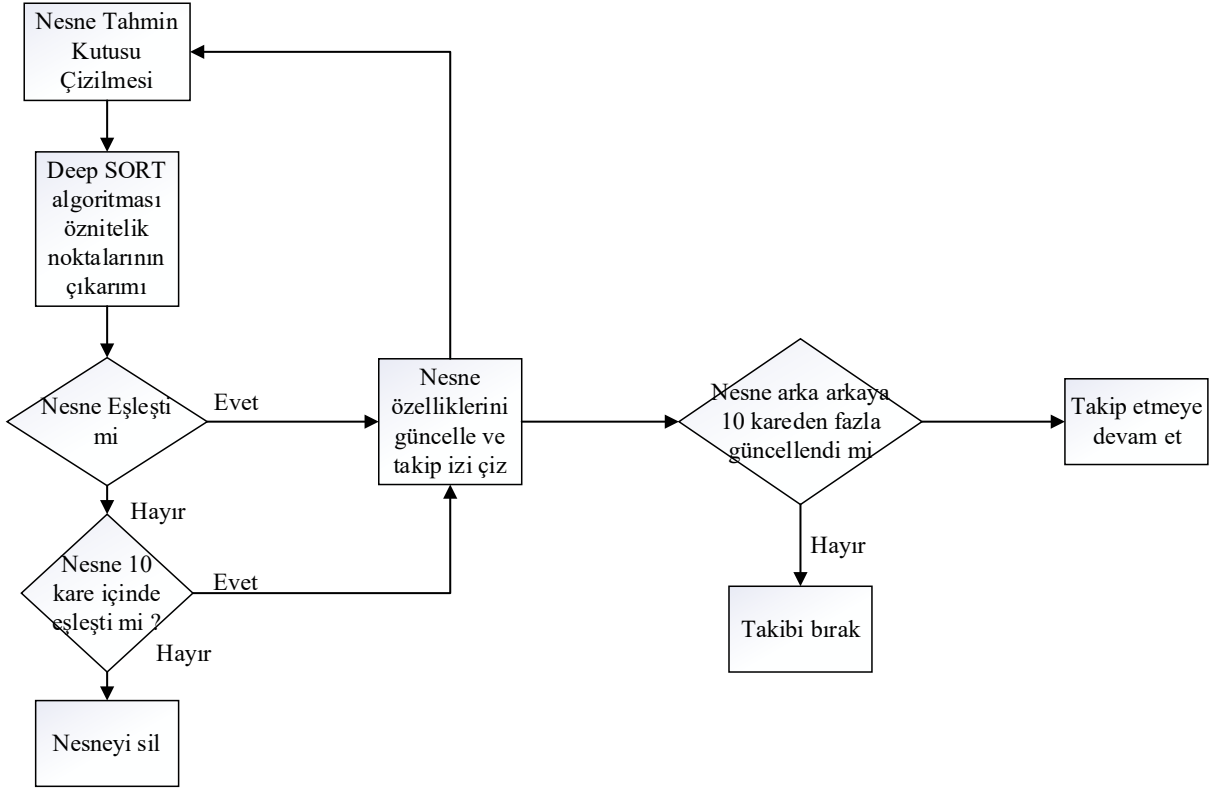
2.3 Simple Online and Real-time Tracking with Deep Association Metric (Deep SORT)-(Derin İlişkilendirme Metriği) İle Nesne Takibi

Nesne takibinde nesnenin sürekli görüntü dizileri (frame) arasında ki konumunu tahmin etmek büyük sorundur. Görüntü dizileri arasındaki hızlı geçişler, birden çok nesnenin görüntü üzerinde birbirine çok yakın durmaları gibi durumlar sıklıkla maruz kalınan bu büyük zorluklardandır. Bu zorluklar bir nesnenin hem takip edilmesi hem de konumu hakkında yanlış tahminlerde bulunulmasına sebep olabilir.

Deep SORT [13], araçları takip ederken frame arası tikanıklığı ortadan kaldırmak için derin evrişimli sinir ağının görünüm özelliklerini kullanmaktadır.

Araçları aşamalı olarak takip etmek için basit bir sistemdir ve videonun her karesinde Kalman filtresini[14] kullanmaktadır. Şekil 4'te Deep SORT algoritmasının akış diyagramı gösterilmiştir.

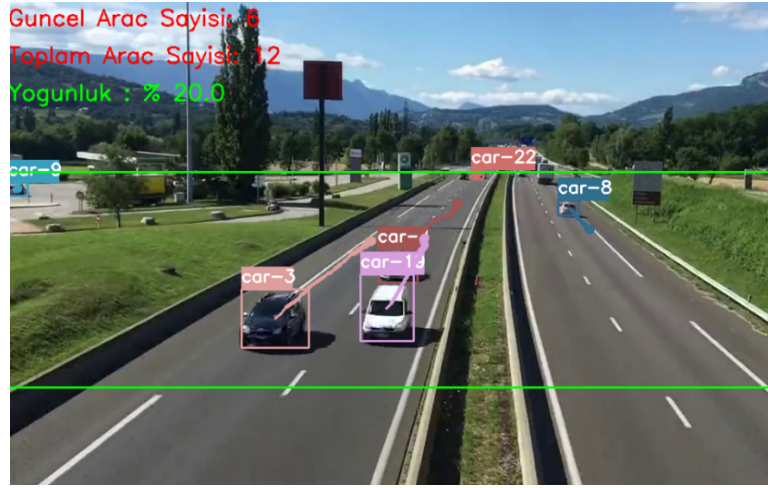
Tespit edilen her nesne için çizilen nesne tahmin kutusu ile başlayan algoritma nesne öznelik çıkarımı ile devam eder. Bir sonraki framede çıkarılan özneliklerin nesne ile eşleşme durumu sorgulanır. Eğer eşleşme devam ediyorsa özellikler güncellenir ve 10 kez bu döngünün gerçekleşip gerçekleşmediği sorgulanır. Döngünün 10 kez gerçekleşmesi durumunda nesne takibi devam etmektedir. Nesnenin bir sonraki frame de öznelik eşleşmesine ulaşılmaması veya ulaşılmaması durumunun 10 kez tekrarı olmaması durumunda ise nesnenin silinmesi veya takibinin bırakılması sonucuna varılır.



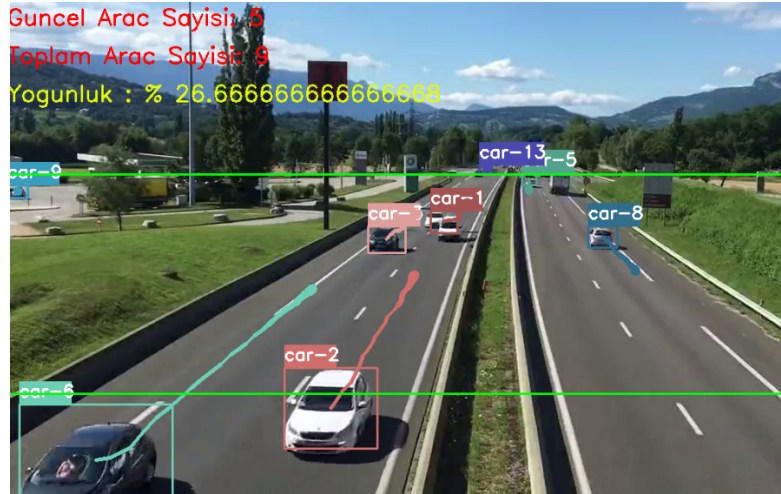
Şekil 4. Deep SORT algoritması akış diyagramı.

2.4 Sistemin Çalışması

Sistem çalışmaya başlatıldığında şekil 1’de ki ekran açılmaktadır. Başlangıç ekranında ilk 20 saniye yoğunluk hesaplaması yapılır. Yolu kullanan toplam araç ve yoğunluk hesabı yapılacak olan alan içerisinde ki araç sayısı video ekranında gösterilir. Toplam araç sayısı bir yolun yıl bazında yoğunluğu ve kullanım sıklığı bakımından bazı bilgiler sunabilir. Karayolları bu bilgiler ışığında yol projeleri yapabilir. Sınırlanan alana araç boyutlarının mesafeye olan değişimine göre yapılan tahminde 30 araç sığıldığı varsayılmıştır ve yoğunluk yüzdesi bu varsayıma göre hesaplanır. Şekil 3 de görüldüğü gibi 20 saniye sonunda sınırlandırılmış alanda bulunan aracın sayısına göre yüzde yoğunluk hesabı yapılır ve yazdırılır. Şekil 5’de yoğunluk yazısı yoğunluğun yüzde değerine göre farklı renkler almaktadır. Eğer yoğunluk 100 de 25ten az ise yeşil, 25 ile 50 arasında ise sarı, 50 ile 70 arasında ise turuncu, 70 ve üzeri ise kırmızı renkte gösterilmektedir.



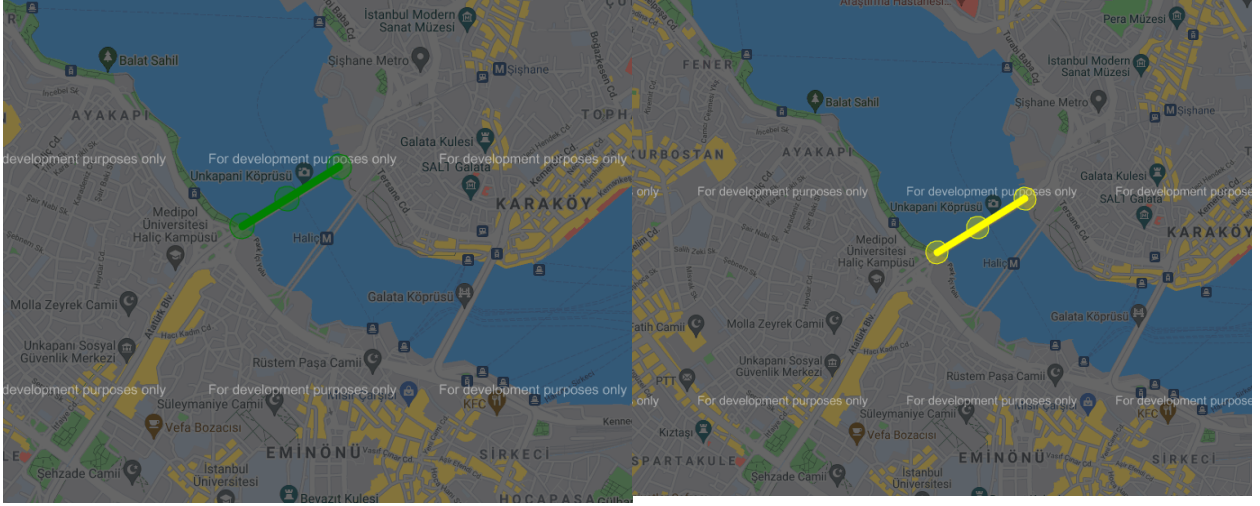
(a)



(b)

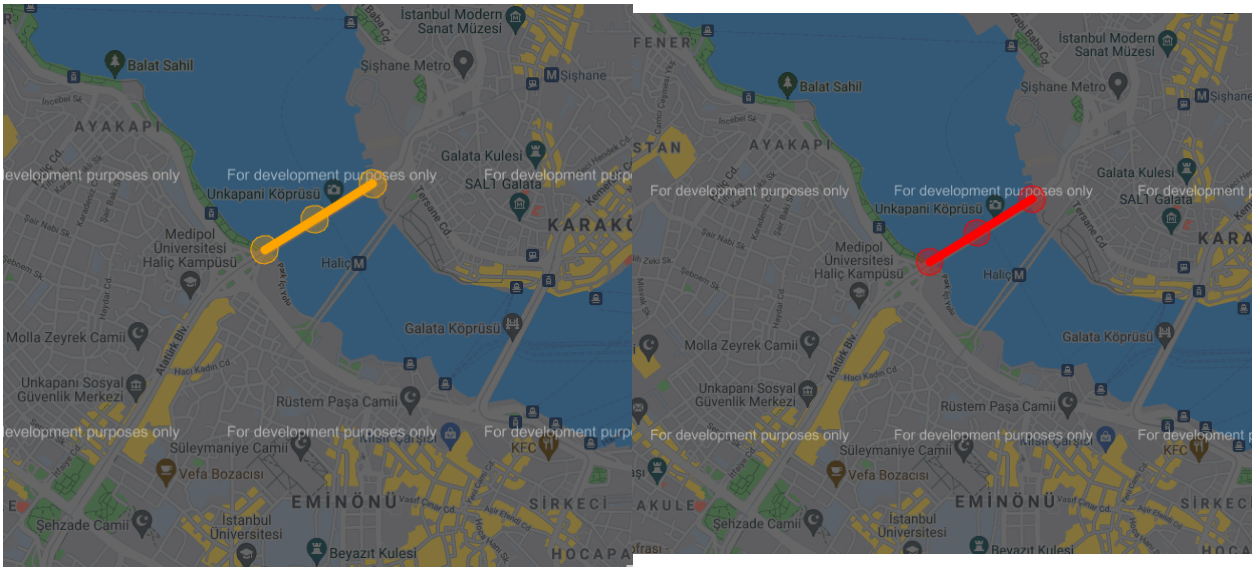
Şekil 5. Programın çalışma ekranı üzerinde toplam araç, güncel araç ve yoğunluk hesaplaması yapması. (a) %25'den az yoğunluk. (b) %25-%50' arası yoğunluk.

Ayrıca makalede anlatılan programda gmplot kütüphanesi kullanılarak yoğunluk harita üzerine harita.html dosyası oluşturularaktan çalışma ekranında olduğu gibi her 20 saniye de güncellenerekten gösterilir. Programda kullanılan karayolu video görüntüsünün İstanbul Unkapanı köprüsü olduğu varsayılaraktan haritada başlangıç, orta ve bitiş enlem ve boylamları sırasıyla; (41.023081,28.962607), (41.024057, 28.964736) ve (41.025207, 28,967228) olarak işaretlenmiştir. Html uzantısı üzerinde haritanın her 20sn de bir güncellenmesi amacıyla Google Chrome'da açılarak Easy Auto Refresh eklentisi kullanılmıştır. Şekil 6 harita üzerinde varsayılan konumu ve trafik yoğunluğunun renklendirilmesini göstermektedir.



(a)

(b)



(c)

(d)

Şekil 6. Programın harita üzerinde yoğunluk renklendirmesi yapması. (a) %25'den az yoğunluk. (b) %50'den az yoğunluk. (c) % 70'den az yoğunluk. (d) %70 ve üzeri yoğunluk.

3. SİSTEMİN EĞİTİLMESİ

Eğitim için Nepal'de çekilen karayolu görüntülerini içeren verisetimizi[15] kullandık.

Verisetimiz eğitim ve test amaçlı olarak ikiye bölünmüştür. Bu bölünmede %80 eğitim seti ve %20 test seti olarak ayırım yapılmıştır ve toplamda 1000 görsele ile çalışılmıştır. Eğitim ve test seti için seçilen görseller verisetinden rastgele seçilmiştir.

Modelin iyi eğitilmesi için eğitim seti oranı yüksek tutulmalıdır. Bu çalışmada eğitim setinde 800 görüntü bulunmaktadır ve içerisinde örnek taşıt türü olarak otomobil (car) ve kamyon (truck) bulunmaktadır. Test setinde ise 200 görsel örnek bulunmaktadır ve modeli test etmek için yeterlidir. Eğitim aşaması Google Colaboratory kullanılarak gerçekleştirilmiştir.

3.1. Başarı Oranının Test Edilmesi

Eğitilen modelimizin başarı oranının test edilmesinde kullanılan ve nesne algılama için başarımlı ölçütü olan Mean Average Precision(map) kullanılmıştır.

Map, sınıflandırılmış her nesnenin average accuracy' (ap)-(ortalama doğruluğu) ortalamasını temsil eder. Map başarı ölçüm sistemi içerisinde Recall (Geri çağırma), Presicion (Hassasiyet) ve IoU (Birleşim üzerinde kesişim) referans değerleri bulunur. Recall değeri, tüm doğru tespitlerin ne kadar iyi doğruluk ile bulunduğunu hesaplar. Recall hesaplama formülü denklem 1.de verilmiştir.

$$\text{Recall} = \frac{TP}{TP+FN} \quad (1)$$

Burada TP (True Positive) yani nesnenin doğru sınıflandırılarak tespit edilmesi, FN (False Negative) ise tespiti yapılmak istenen nesneyi o sınıfa dahil etmemektir. Yani görselde bulunan otomobile, otomobil değil denilme değeridir.

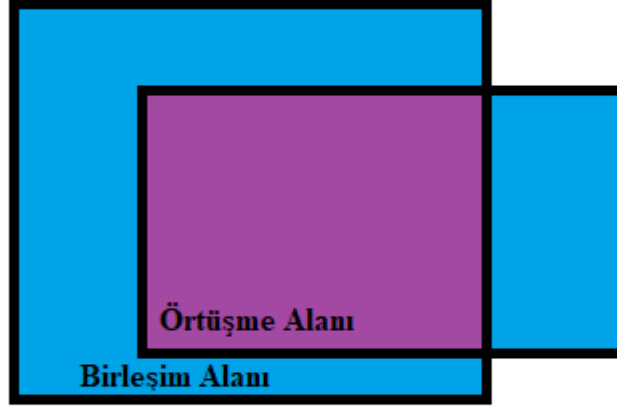
Presicion değeri tahmin değerlerinin ne kadar doğru olduğunu ölçer. Yani tahminlerimizin yüzde doğruluğudur. Hesaplama formülü denklem 2 de verilmiştir.

$$\text{Presicion} = \frac{TP}{TP+FP} \quad (2)$$

Burada FP (False Positive) değeri nesne olmayana nesne denilmesi değeridir.

IoU değeri ise, 2 sınır arasındaki çakışmayı ölçen bir ölçüdür. Bu sınırlardan birisi tahmin edilen nesnenin sınırları diğeri ise nesnenin gerçek sınırlarıdır. IoU değeri ile tahmin edilen sınırların gerçek sınırlar ile ne kadar örtüştüğü ölçülmektedir. Denklem 3 ve şekil 6 IoU ölçüm ve hesaplamasını açıklamaktadır.

$$\text{IoU} = \frac{(\text{Örtüşme Alanı})}{(\text{Birleşim Alanı})} \quad (3)$$



Şekil 7. IoU Hesaplama Alanı

Map değeri bu faktörler kullanılarak bütün kategorilerin ap değerinin ortalanması ile elde edilmektedir. Tablo 1 modelimizin map değerlerini sunmaktadır.

Tablo 1. Modelin Doğruluk değerleri

Prametreler	Car(otomobil)	Truck(Kamyon)	Presicion	Recall	Ortalama IoU	map
Sonuçlar	%90.05	%91.2	0.92	0.92	%67.21	%90.52

4. SONUÇLAR

Bu çalışma , bir karayolu perspektifinden gözetlenmiş 3 dakikalık video sahnesi için araç algılama, izleme ve sayma verileri elde ederek trafik yoğunluğunu hesaplayan bir yöntemi önerdi. YOLOv3 nesne algılama algoritması, karayolu üzerinde nesne olarak otomobil ve kamyon nesnelere dayalı sınırları kesinleştirilmiş alan üzerinde araç algılama modelini elde etti. Küçük nesne algılama sorununu ve nesnenin çok ölçekli varyasyonunu tespit edebilmek üzere hem YOLOv3 hem de Deep SORT algoritmaları kullanıldı. Araçların görüntüdeki konumu, nesne algılama sonucu ile Deep SORT özellik çıkarma algoritması ile tahmin edilmiştir.

Modelin başarı ölçü oranı map yöntemi ile test edilmiş ve otomobil tespitinde %90.05 kamyon tespitinde ise %91.2 başarı oranı ile yüksek başarı elde etmiştir. Model gerçek hayata uyarlanabilir derecede başarılıdır ve daha fazla eğitim sonucunda başarı oranı artırılabilir.

Araç trafik yoğunluğunun geleneksel olarak donanımla izleme yöntemine karşın, makalede sunulan yöntem hem düşük maliyetlidir hem de doğruluk oranı yüksektir. Ayrıca donanımsal izlemeye göre herhangi bir inşaat ve kurulum gerektirmez. Yalnızca var olan yol güvenlik kameralarından alınan verilerin kullanılması sağlanır.

Sunulan çalışma karayollarında bulunan mobese kameraları ile entegre edilebilmesi durumunda daha çok yol üzerinde daha çok trafik bilgisi edilerek hem yoğunluk hem de farklı tespitler yapılabilir. Ayrıca uygulama geliştirilerek anormal park olayları, kaza olayları ve hız tespiti alanlarında kullanılabilir.

Bu şekilde, devamlı suretle mobese izleme takibi yapan polislerin yükü azalabilir ayrıca kaza tespitinde otomatik bilgilendirme yapılarak sağlık görevlileri yönlendirilebilir. Ek olarak harita görüntüleme ile internet erişiminin olduğu her yerden anlık trafik yoğunluğu, kaza durumu öğrenilebilir.

5. Kaynaklar

- [1] Y. Lin, P. Wang, and M. Ma, "Intelligent Transportation System(ITS): Concept, Challenge and Opportunity," in 2017 IEEE 3rd International Conference on Big Data Security on Cloud (BigDataSecurity), IEEE International Conference on High Performance and Smart Computing, (HPSC) and IEEE International Conference on Intelligent Data and Security (IDS), May 2017, pp. 167–172, doi: 10.1109/BigDataSecurity.2017.50.
- [2] H. T. Yaman, "Akıllı Ulaşım Sistemleri tanımı ve kapsamı," Akıllı Ulaşım ve Güvenlik Sistemleri," p. 54, 2014.
- [3] H. Song, H. Liang, H. Li, Z. Dai, and X. Yun, "Vision-based vehicle detection and counting system using deep learning in highway scenes," Eur. Transp. Res. Rev., vol. 11, no. 1, p. 51, Dec. 2019, doi: 10.1186/s12544-019-0390-4.
- [4] T. Acharya, "Image Processing: Principles and Applications," J. Electron. Imaging, vol. 15, no. 3, p. 039901, Jul. 2006, doi: 10.1117/1.2348895.
- [5] N. Yusnita, R., Norbaya, F., Basharuiddin, "Intelligent parking space detection system based on image processing," vol. 3, pp. 232–235, 2012.
- [6] C. M. Bautista, C. A. Dy, M. I. Manalac, R. A. Orbe, and M. Cordel, "Convolutional neural network for vehicle detection in low resolution traffic videos," in 2016 IEEE Region 10 Symposium (TENSYP), May 2016, pp. 277–281, doi: 10.1109/TENCONSpring.2016.7519418.
- [7] D. Zang, Z. Chai, J. Zhang, D. Zhang, and J. Cheng, "Vehicle license plate recognition using visual attention model and deep learning," J. Electron. Imaging, vol. 24, no. 3, p. 033001, May 2015, doi: 10.1117/1.JEI.24.3.033001.
- [8] M. Özdemir, E., Oral, "Genetik Algoritma ve Görüntü İşleme Teknikleri Kullanarak Servis Araçlarının Şehir Trafğinde İzlediği Yolun Kısaltılması," 1. Uluslararası Yerel Yönetimler Üniversitesi ve Sanayi İşbirliği Sempozyumu, 2003.
- [9] W. Huang, G. Song, H. Hong, and K. Xie, "Deep Architecture for Traffic Flow Prediction: Deep Belief Networks With Multitask Learning," IEEE Trans. Intell. Transp. Syst., vol. 15, no. 5, pp. 2191–2201, Oct. 2014, doi: 10.1109/TITS.2014.2311123.
- [10] N. G. Polson and V. O. Sokolov, "Deep learning for short-term traffic flow prediction," Transp. Res. Part C Emerg. Technol., vol. 79, pp. 1–17, Jun. 2017, doi: 10.1016/j.trc.2017.02.024.
- [11] L. Zhu, F. Guo, R. Krishnan, and J. W. Polak, "A Deep Learning Approach for Traffic Incident Detection in Urban Networks," in 2018 21st International Conference on Intelligent Transportation Systems (ITSC), Nov. 2018, pp. 1011–1016, doi: 10.1109/ITSC.2018.8569402.
- [12] A. Redmon, J., & Farhadi, "Yolov3: An incremental improvement," Redmon, J., Farhadi, A., 2018.
- [13] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," IEEE, 2017, doi: 10.1002/0471221546.
- [14] Simon Haykin, Kalman Filtering and Neural Networks. John Wiley & Sons, Inc., 2001.
- [15] "https://www.kaggle.com/sdevkota007/vehicles-nepal," 2021. .