GU J Sci, Part C, 11(1): 68-80 (2023)

Gazi University

Journal of Science

PART C: DESIGN AND TECHNOLOGY

http://dergipark.gov.tr/gujsc

Determining Empty Parking Spaces and Giving Directions to The User with The Mobile Application

Ahmet Basri ALTAY [1] Ayşe DEMİRHAN[1,*]

[1]*Gazi University, Faculty of Technology, Department of Electrical and Electronics Engineering, 06500, Yenimahalle/ANKARA*

Article Info:

Research article
Received: 09.09.2022
Revision: 14.10.2022
Accepted: 02.12.2022

Highlights

- *Parking Space Detection*
- *Object Detection with Background Removal Algorithms*
- *Giving Directions To Empty Parking Spaces With Mobile Application*

Keywords

Parking Space
OpenCV
Kotlin
Mobile Application
Background Subtraction
Haar Cascade

Graphical/Tabular Abstract

In this study, a mobile application has been developed using Kotlin where users can easily find empty parking spaces and get directions to the selected one.

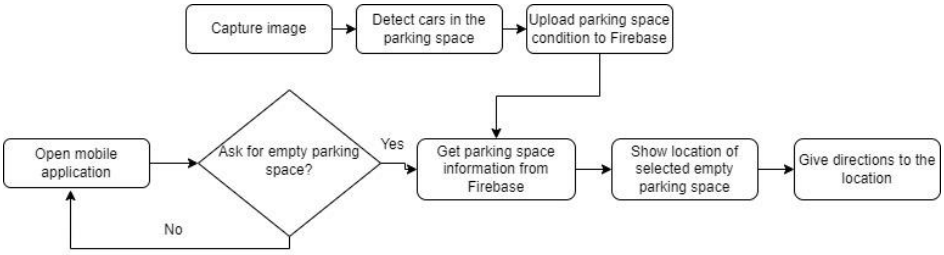


Figure A. Summary workflow block diagram

Purpose: Purpose of this study is to develop a mobile application that gives information of the empty parking spaces and directions to the selected empty parking space among available ones. Cameras that are nearby the different parking spaces are used for this purpose. Images obtained from the cameras are processed to detect the places where the vehicles can park and the information of the empty parking spaces are reported to the user. Then, the user is provided with the opportunity to choose the appropriate one from the empty parking spaces and get directions to that point.

Theory and Methods: Haar cascade classifier was used to detect empty parking spaces and background deletion algorithms are also used to detect parked cars. Obtained data was uploaded to Firebase service, which is a mobile and web application development platform. When the user wants to find a parking space in the mobile application, the necessary data is retrieved from the Firebase platform, and the location of the empty parking spaces is displayed to the user, and the user can get directions to the location that belongs to the selected empty parking space.

Results: The proposed system in the study was tested with images taken from different parking regions and 100% success was achieved. The test of the system was made with the videos taken within the scope of the study. As there is currently no access to street cameras, currently testing of the software has been done by phone video recording of various streets and parking lots. In this way, 25 videos were shot and tried, and in all of these trials, occupied and empty parking spaces were successfully detected. Since the proposed system works based on object detection, it is not affected by weather conditions as long as it does not block the view of the camera. While the system works with full efficiency in daylight, it will not work efficiently if there is a lighting problem in the parking area when it gets dark.

Conclusion: Developed mobile application has the capacity to prevent unnecessary fuel and time loss by eliminating the problem of drivers searching for parking spaces. This application also has the capacity to help reduce traffic congestion by removing vehicles that create congestion in traffic to search for parking spaces. One of the steps that can be taken for the development of the application is to mark the parking place as full in the database if the user marks the parking place they want to go, in order to prevent the application from working more healthily and to eliminate the possibility of filling the parking space they plan to go to if the application becomes widespread. Thus, another user will not be offered the same place and the probability of that parking lot to get full will decrease.

*Corresponding author, e-mail: ayseoguz@gazi.edu.tr
DOI: 10.29109/gujsc.1173113

GU J Sci, Part C, 11(1): 68-80 (2023)

Gazi Üniversitesi

Fen Bilimleri Dergisi

PART C: TASARIM VE TEKNOLOJİ

http://dergipark.gov.tr/gujsc

Determining Empty Parking Spaces and Giving Directions to The User with The Mobile Application

Ahmet Basri ALTAY [2] Ayşe DEMİRHAN[1,*]

[1]*Gazi University, Faculty of Technology, Department of Electrical and Electronics Engineering, 06500, Yenimahalle/ANKARA*

Makale Bilgisi

Araştırma makalesi
Başvuru: 09.09.2022
Düzeltme: 14.10.2022
Kabul: 02.12.2022

Keywords

Parking Space
OpenCV
Kotlin
Mobile Application
Background Subtraction
Haar Cascade

Anahtar Kelimeler

Park Yeri
OpenCV
Kotlin
Mobil Uygulama
Arka Plan Çıkarma
Haar Cascade

Abstract

Today, the population density is increasing day by day and the number of vehicles in cities is increasing accordingly. For this reason, finding a parking space has become very difficult for drivers. Drivers staying in traffic to find a parking space both compress traffic and cause unnecessary fuel costs. In this study, an application has been developed where users can easily find a parking space and choose among the empty parking spaces. Cameras on the street, cameras placed for security or satellite images can be used for this study. By processing the obtained images, the places where the vehicles can park are determined and the information of the empty ones is sent to the user. Then, the user is provided with the opportunity to choose the appropriate one from the empty parking spaces and get directions to that point. Two methods were used for image processing and detection of empty parking spaces. The first of these methods detects vehicles using the background deletion algorithms, the second method uses the Haar cascade classifier. After these determinations are made, information is obtained whether the parking spaces are full or empty. The mobile application was developed with the Kotlin software language on the Android Studio platform.

Boş Park Yerlerinin Tespiti ve Kullanıcıya Mobil Uygulama İle Yol Tarifi Verilmesi

Öz

Günümüzde nüfus yoğunluğu her geçen gün hızla artmakta ve buna bağlı olarak şehirlerdeki araçların sayısı da artış göstermektedir. Bu nedenle park yeri bulmak sürücüler için oldukça zorlu bir hal almıştır. Sürücülerin park yeri bulmak için trafikte kalması hem trafiği daha da sıkıştırmakta hem de gereksiz yakıt masrafına sebep olmaktadır. Bu çalışmada kullanıcıların kolayca park yeri bulabilecekleri ve boş park yerleri içerisinden seçim yapabilecekleri bir uygulama geliştirilmiştir. Sokakta bulunan kameralardan, güvenlik için yerleştirilmiş kameralardan ya da uydu görüntülerinden bu çalışma için yararlanılabilir. Elde edilen görüntüler işlenerek araçların park edebileceği yerler tespit edilmekte ve kullanıcıya boş olanların bilgisi iletilmektedir. Ardından kullanıcıya boş park yerlerinden kendisine uygun olanı seçme ve o noktaya yol tarifi alma imkânı sağlanmaktadır. Görüntünün işlenmesi ile boş park yerlerinin tespiti için iki yöntemden yararlanılmıştır. Bu yöntemlerden ilki arka plan silme algoritmaları kullanılarak, ikincisi ise Haar cascade sınıflandırıcısı kullanılarak araçların tespit edilmesidir. Bu tespitlerin yapılmasının ardından park yerlerinin dolu veya boş olduğu bilgisi elde edilmektedir. Mobil uygulama, Kotlin yazılım dili ile Android Studio platformu kullanılarak geliştirilmiştir.

1. GİRİŞ (INTRODUCTION)

Özellikle büyük şehirlerde park yeri bulmak sürücülerin trafikte yaşadığı büyük bir problem haline gelmiştir [1]. 2017'de yayınlanan bir rapora göre Amerika'da park yeri aramanın maliyeti yıllık 73 milyar dolardır [2]. Günümüzde hızlı nüfus artışına bağlı olarak artan araç sayıları nedeniyle şehirlerde oluşan park yeri problemlerini çözmek için hem yatay hem de dikey alanın daha etkin kullanılması amacıyla teknolojik gelişmeler ışığında çeşitli otopark sistemleri geliştirilmiştir [3]. Ayrıca geliştirilen birçok değişik

yöntemle boş park yerleri tespit edilmeye çalışılmaktadır. Boş park yerlerinin tespiti için çeşitli sensörler kullanılabilmektedir. Bu sensörler ve kullandıkları yöntemler Tablo 1'de verilmiştir [4].

Tablo 1. *Boş park yerlerinin tespitinde kullanılan sensörler [4]*

Sensör	***Yöntem***
Kızılötesi	*Işık*
Elektromanyetik	*Elektromanyetik*
Piezoelektrik	*Mekanik, titreşim*
Mikrodalga radar	*Mikrodalga*
Manyetometre	*Manyetik*
RFID	*Radyo dalgaları*
Kamera	*Görüntü işleme*

Bu çalışmada kamera görüntülerinin işlenmesi yoluyla boş park yerlerinin tespiti üzerinde çalışılmıştır. Bu yöntemin seçilmesinin sebepleri; diğer sistemlere göre daha ekonomik olması, kötü hava koşullarından daha az etkileniyor olması ve günümüzde hali hazırda birçok yere yerleştirilmiş olan kameraların kullanılabilecek olmasıdır.

Kameraları ve görüntü işlemeyi kullanarak boş park yeri bulma ve nesne tespiti için yapılmış benzer çalışmalar literatürde bulunmaktadır. Bu çalışmaların çoğunda nesne tespiti derin öğrenme yöntemleri kullanılarak yapılmıştır. Gökçe ve Sonugür hareketli nesnelerin sınıflandırılması için kullanılan yöntemleri karşılaştırmış ve hangi yöntemin ne gibi üstünlükleri olduğunu belirlemeye çalışmışlardır [5]. Şenel ve Tokat ise görüntü işleme yöntemleri ile ortamdaki insanların tespit edilmesi ve insan yoğunluğunun hesaplanması üzerine çalışmışlardır. Benzer şekilde birçok çalışma Haar cascade uygulaması ile yüz ve araç tespiti gibi işlemler için görüntü işlemeden faydalanmıştır [6]. Bu alandaki çalışmalar genel olarak 3 başlıkta sınıflandırılabilir. 1) Bir sinir ağı eğiterek nesne tespiti ve park yerinin bu yolla bulunması, 2) Arka plan silme teknikleri ile nesnelerin tespiti ve maske yöntemleri ile park yerinin doluluğu ya da boşluğunun bulunması, 3) Eğitilmiş hazır ağları Haar cascade yapısı şeklinde kullanarak boş park yerlerinin bulunmasıdır.

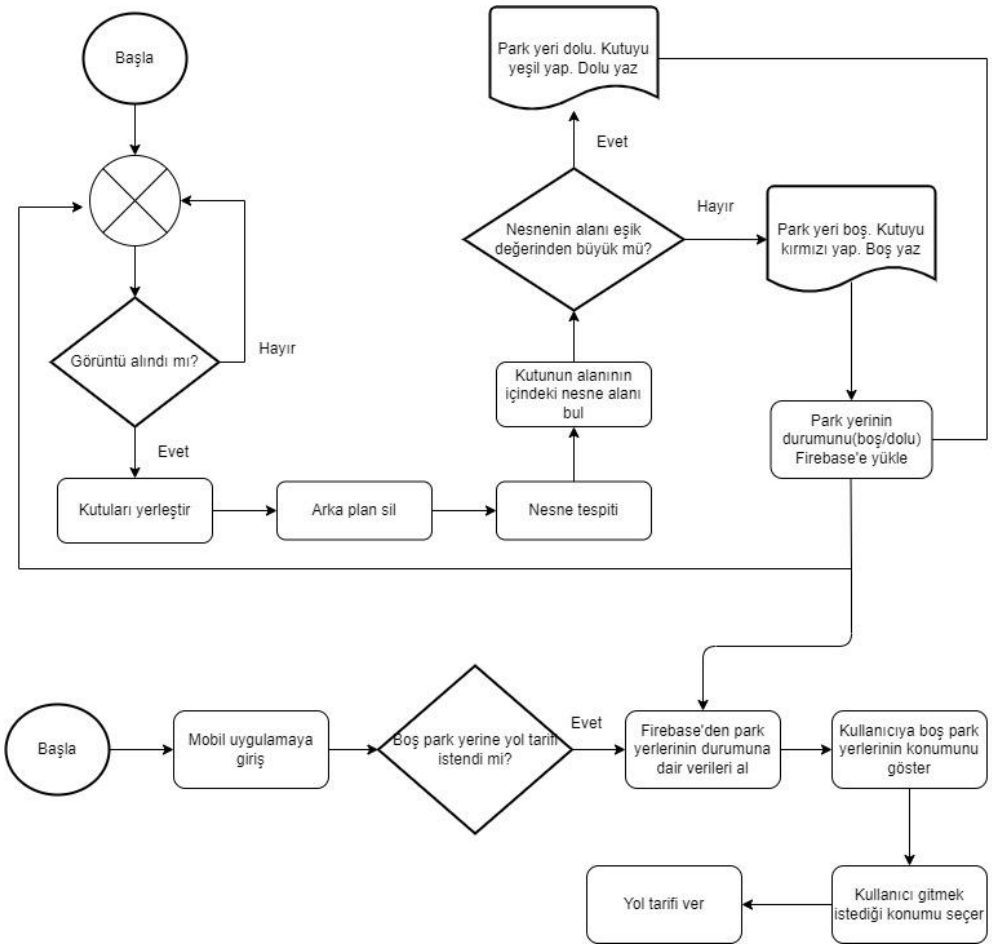
Nesne tespiti ve boş park yeri bulunması konusunda pek çok görüntü işleme çalışması yapılmıştır. Bu çalışmalardan büyük çoğunluğunda derin öğrenme yöntemleri kullanılırken [7]-[11] bazılarında ise yalnızca görüntü işleme yöntemleri kullanılarak park yeri durumunun tespiti sağlanmıştır. Dikbayır ve Bülbül evrişimli sinir ağları ve YOLO algoritmasını kullanarak bir araç tespit uygulaması geliştirilmişlerdir [9]. Amato ve diğerleri akıllı bir kamera üzerinde çalışan bir evrişimli sinir ağı sınıflandırıcısını kullanarak gerçek zamanlı otopark doluluk tespiti yapan bir çalışma gerçekleştirmiştir [10]. Wu ve diğerleri jeomanyetik alan değişimlerine dayalı park alanlarındaki araçların varlığını tespit eden akıllı bir park sistemi sunmuştur [12]. Kayış ve diğerleri ise bir web uygulaması ile kapalı alanlarda binaları modellemiş ve bir mobil uygulama ile modellenen binaları 2B düzlemde gözlemlemiştir [13]. Görüntü işleme çalışmalarının dışında boş park yerlerinin tespiti için elektronik sensörler de kullanılmaktadır [12]. Ancak sensör kullanılan yöntemde her park yeri için ayrı bir sensöre ihtiyaç duyulduğu ve sensörlerin maliyeti fazla olduğu için görüntü işleme ile bu işlemi gerçekleştirmek daha avantajlıdır. Görüntü işleme yöntemleri ile görüntülerde kenar tespiti ve arka plan silme fonksiyonu ile nesne tespiti ve takibi yapılabilmektedir [14]-[16]. Görüntü işleme yöntemleri ile nesne tespiti ve takibi yapabilen farklı birçok çalışma ile bu çalışmaların kıyaslandığı yayınlar da mevcuttur [4],[17]. Chowdhury ve diğerleri araç hırsızlığını önlemek amacıyla kapalı otoparklar için bir mobil uygulama geliştirmişlerdir. Bu uygulama aracılığıyla otoparka giren araç bilgileri görüntü işleme yöntemleri ile alındıktan sonra otoparktan çıkış için araç sahibinin mobil uygulama aracılığıyla onay vermesi sağlanmıştır [18]. Lopez ve diğerleri web bağlantılı bir kameradan canlı görüntüleri işleyerek belirli bir park yerinin her bir alanında bir aracın bulunup bulunmadığını tespit eden ve bu bilgiyi bir mobil uygulama aracılığıyla kullanıcıya sunan bir çalışma gerçekleştirmişlerdir [19].

Bu çalışmada boş park yerlerinin tespiti için Haar cascade sınıflandırıcı kullanılmıştır. Aynı zamanda arka plan silme algoritmalarından da nesne tespiti için faydalanılmıştır. İki farklı yöntemin kullanılmasının nedeni ikisinin de birbirine karşı avantajları bulunmasıdır. Bu sebeple iki yönteme de park yerlerinin boşluğu veya doluluğunun tespitini sağlamak amacıyla yer verilmiştir. Yapılan bu tespitten sonra ise elde edilen veriler bir mobil ve web uygulaması geliştirme platformu olan Firebase servisine yüklenmiştir. Mobil uygulamada kullanıcı park yeri bulmak istediği zaman Firebase platformundan gerekli olan veriler çekilerek kullanıcıya boş park yerlerinin konumu gösterilmekte ve kullanıcı içlerinden istediğini belirleyerek o konuma yol tarifi alabilmektedir.

Önerilen sistem literatürdeki benzer çalışmalarla karşılaştırıldığında harici bir sensör kullanmayarak sadece gerçek zamanlı kamera görüntüleri üzerinde çalışma, gerçek zamanlı sonuç üretme, park yeri doluluk durumunu anlık olarak veri tabanında tutma, boş park yerlerini harita üzerinde gösterme ve seçilen boş park yerine yol tarifi verme konularında üstünlükleri vardır.

2. MATERYAL VE METOTLAR (MATERIALS AND METHODS)

Bu çalışmada görüntü işleme algoritması OpenCV kütüphaneleri kullanarak PyCharm IDE'si üzerinde geliştirilmiştir. OpenCV nesne tanıma, yüz tespiti, videodaki hareketleri algılama, şablon ve şekil eşleştirme ve kamera kalibrasyonu gibi görüntü işleme ile ilgili birçok fonksiyonu optimize edilmiş şekilde içinde bulundurmaktadır [20, 21]. Çalışmadaki hem görüntü işleme hem de mobil uygulama işlem adımlarını gösteren blok diyagram Şekil 1'de verilmiştir.



Şekil 1. *İş akış blok diyagramı*

Mobil uygulamayı geliştirmek için Android Studio IDE'si üzerinden Kotlin dili kullanılmıştır. Kotlin ile Java'ya nazaran daha kısa ve sade kodlama yapmak mümkündür. Kullanıcılara daha hızlı çalışma imkânı sağlamaktadır. Kotlin ile JVM, Android, tarayıcı, web ve gömülü sistem uygulamaları geliştirilebilmektedir. Android Studio anında çalıştırma özelliği ile emülatör üzerinde çalışan uygulamanın

hızlı bir şekilde yansıtılmasını sağlayarak düzenleme, derleme ve çalıştırma süreçlerini hızlandırmaktadır. Kodlarda yapılan her adımı uygulama üzerinden görebilme imkânı sağlar [22].

Çalışmada veritabanı yönetimi için Firebase platformu kullanılmıştır. Uygulamanın gerçek zamanlı veritabanı yönetimi, kullanıcı giriş yetkilendirmesi ve depolama özelliklerinden yararlanılmıştır.

PyCharm'da görüntü işlenerek elde edilen park yerlerinin durumları ve koordinatları Firebase'e yüklendikten sonra mobil uygulamadan bu verilere erişim sağlanmıştır. Benzer şekilde Kotlin ile de kullanıcı bilgileri Firebase'e yüklenmiş ve kullanıcı giriş yaparken kontrol edilmesi sağlanmıştır.

2.1. PyCharm'da Yapılan İşlemler (Transactions Performed In PyCharm)

Bu çalışma için herhangi bir sokakta kamera görüntüsü alınan araçlar ve park yeri olarak uygun olabilecek bölgeleri gösteren görüntüler işlenmiştir. Bu görüntülere bir örnek Şekil 2'de verilmiştir. İşlenen görüntüler ile park yerlerinin dolu veya boş olma durumu tespit edilmiştir. Bu işlem PyCharm IDE'sinde OpenCV kütüphanesi ve Pyhton dili kullanılarak gerçekleştirilmiştir. Park yerlerinin dolu veya boş olma durumunu tespit etmek için arka plan çıkarma ve Haar cascade sınıflandırıcısı olmak üzere iki temel yol izlenmiştir.

Şekil 2. *Örnek park yeri görüntüsü*

Park Alanlarının Tespiti (Detection of Parking Spaces)

Park yeri olarak kullanılabilecek bölgelere manuel olarak dikdörtgen kutular yerleştirilmiştir. Şekil 3'te 2 adet park yeri için görüntüye yerleştirilmiş 2 kutu görülmektedir. Bu kutuların konumları araçların park edebileceği noktalara göre ayarlanmıştır. Kutular park yeri olarak kullanılacak bölgenin görüntüsüne göre önceden belirlenmektedir. Kutuların sayısı ve konumu park yeri alanının büyüklüğüne ve yerleşimine göre kullanılabilecek park alanı sayısınca ayarlanmaktadır. Sistem, görüntülerine ulaşılabilen herhangi bir kameranın görüş alanı içerisinde bulunan ve önceden kutu yerleştirme işlemi yapılmış her noktada kullanılabilir özelliktedir. Kutu büyüklüğü kameranın görüş açısından standart bir binek aracının sığabileceği boyutlarda belirlenmektedir. Park yerlerinin etrafındaki araç trafiğinin park yeri alanını gösteren kutu bölgesini etkilemediğinden uygulamanın çalışması üzerinde bir etkisi yoktur.

Şekil 3. *Kutuların Görüntüye Yerleştirilmesi*

Arka Plan Çıkarma (Background Subtraction)

Kameradan alınan görüntülerde peş peşe devamlı bir görüntü akışı söz konusudur. Akan görüntülerde sabit olan kısımlar ve hareket eden nesneler mevcuttur. Görüntü içinde bulunan nesnelerin konumu değiştiği takdirde önceki ve sonraki konumdaki piksel değerleri değişir. Piksellerde oluşan bu farklılıklar sayesinde arka planın tespiti yapılarak hareket halindeki nesnelerin ayrımının yapılması mümkündür.

Şekil 4'te Şekil 2'de verilen sokak görüntüsünün arka plan silme yöntemi uygulanmış hali gösterilmektedir.

Şekil 4. *Görüntüden arka planı silme*

Arka plan silme işleminde ön plan ve arka planı bölütlemek için Gauss karışımı (Mixture of Gaussian) [23], adaptif Gauss karışımı [24], k-en yakın komşu [25], istatistiksel arka plan görüntüsü tahmini ve piksel başına Bayes bölütlemesi [26] algoritmaları kullanılabilmektedir. Arka plan silme işlemi için bu çalışmada k-en yakın komşu algoritması kullanılmıştır. K-en yakın komşu algoritması sınıflandırılacak veri örneğini mevcut eğitim örnekleriyle karşılaştırarak ona en yakın örnekleri bulan, parametrik olmayan denetimli bir sınıflandırıcıdır. Algoritmanın bir sonraki adımı, k-en yakın eğitim örneklerinin etiketleri arasında en fazla tekrar eden değeri sınıflandırılacak veri örneğinin sınıf etiketi olarak atamaktır. K değeri, sınıflandırma için bulunan özellik uzayındaki en yakın eğitim örneklerinin sayısı olan algoritmanın anahtar parametresidir. K-en yakın komşu algoritmasının tahmin performansı, eğitim verilerinin boyutu ile artar. K-en yakın komşu algoritmasının temel dezavantajı, yüksek hesaplama gereksinimlerine sahip olmasıdır fakat kararlı bir performans göstermesi nedeniyle çeşitli görüntü işleme görevlerinde tercih edilmektedir [27].

Haar Cascade Sınıflandırıcısı (Haar Cascade Classifier)

Haar cascade, Viola ve Jones tarafından yayımlanan ve görüntülerdeki nesnelerin tespiti için önerilen etkili bir yöntemdir [28]. Makine öğrenimine dayalı bir yaklaşım olan Haar cascade yapıları görüntüyü tarayarak genellikle önceden belirlenmiş olan nesneleri tespit etmeye çalışır [29]. Bu çalışmada araç tespit işlemleri için eğitilmiş bir Haar cascade algoritması kullanılmıştır [30]. Bu algoritma ile görüntü taranmış ve bu tarama araç tespitinde kullanılmıştır. Haar cascade sınıflandırıcısı kullanılmasının nedeni nesne tespitinde arka plan silme yöntemine göre daha başarılı bir şekilde çalışmasıdır. Bunun sebebi ise arka plan silme yönteminin hareket eden nesnelerin tespitine daha uygun olmasıdır. Haar cascade sınıflandırıcıda görüntünün işlenmesi zaman almaktadır ve bu sebeple anlık veri alma işleminde gecikmeler yaşanabilmektedir. Fakat arka plan silme yöntemi ile görüntü işleme daha hızlı gerçekleştirilebilir. Bu nedenle çalışmada iki sistemden de yararlanılmıştır.

Nesnelerin bu iki yöntemle tespitinden sonra park yerinin durumu belirlenmiştir. Bunun için görüntünün üzerinde olası park yerlerine kutular yerleştirilmiştir. Bu kutuların özelliği içine beyaz piksel girdiği takdirde bu durumu tespit etmesidir. Burada üzerinde çalışılan görüntü Şekil 2'de verilen örnek görüntüdür. Beyaz pikseller ise görüntüde bir araç veya nesnenin var olması sonucu oluşmaktadır. Eğer kutunun alanının içinde belirli bir eşik değerin üzerinde beyaz piksel varsa park yerinin durumu dolu, eşik değerinin altında beyaz piksel mevcut ise park yeri boş olarak tespit edilmektedir. Bu nedenle iki araçlık yere park etme durumunda sistemde her iki park yeri de dolu olarak görünecektir.

Kutuların işleyişini daha iyi anlayabilmek için Şekil 5 ve Şekil 6 incelenebilir. Şekil 5'te nesne tespit işlemi sonucunda görüntüde bulunan bölgeler beyaz kareler ile gösterilmiştir. Şekil 6'da ise görüntüde tespit edilen nesne bölgeleri üzerine kutu alanları bindirildiğinde elde edilen sonuç yer almaktadır. Eğer tespit edilen nesne kutunun alanı içinde değilse bu nesne göz ardı edilmekte ve park yerlerini etkilememektedir. Park yeri, kutu alanı içinde yer alan nesne bölgelerinin toplam alanı deneysel olarak belirlenen bir eşik değerinin üzerindeyse dolu değilse boş olarak etiketlenmektedir. Bu çalışmada eşik değeri olarak %70 kullanılmıştır. %70'in altında doluluk oranı olduğunda bölge içerisindeki nesne araç olarak tanınmamaktadır.

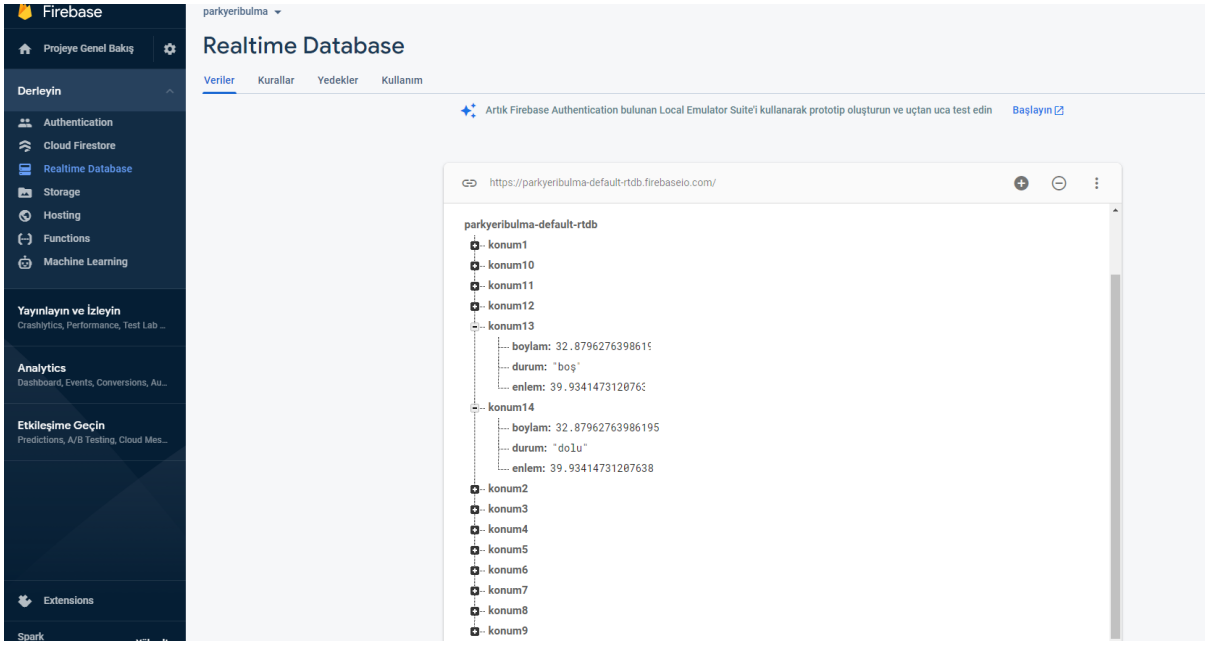
Şekil 5. *Nesne tespit işlemi sonucu*

Şekil 6. *Nesne bölgeleri ve kutu alanları*

Bu işlemler sonucunda park yerlerinin durumu dolu veya boş şeklinde belirlenmektedir (Şekil 7). Bu görüntüde kırmızı ve yeşil ile işaretlenmiş kareler kutulardır. Kutu park yeri boş ise kırmızı, dolu ise yeşil olarak gösterilmiştir. Mavi ile gösterilen olan kutular ise nesne tespitinden elde edilmiş olan karelerdir.

Şekil 7. *Park yerlerinin durumu*

Bu kısımda araçların tespiti yapılırken elde edilen sonuçlar eş zamanlı olarak Firebase'de bulunan Realtime Database'e yüklenmektedir. Bu verilerle beraber görüntünün işlendiği yerin konum bilgileri de Firebase'e iletilmektedir. Şekil 8 (a)'da Firebase ortamında oluşturulan veritabanı görülmektedir. Burada önemli olan nokta park yerinin durumunda herhangi bir değişiklik olduğu anda bunun Firebase'e iletilecek olmasıdır. Bu sayede kullanıcılar uygulamadan anlık olarak doğru bir şekilde park yerlerinin bilgisini alabilecektir. Şekil 8 (a) ve Şekil 8 (b) kıyaslanarak Firebase'de park yerinin durumunun nasıl değiştiği görülebilir. Şekil 8 (a)'da görülen "konum 14" Şekil 2'de verilen örnek görüntüdeki beyaz aracın bulunduğu konumu temsil etmektedir ve "dolu" etiketine sahiptir. Şekil 2'deki araç bölgeden ayrıldığında Şekil 8 (b)'de "konum 14"ün etiketi "boş" şeklinde güncellenmiştir.



(a)

(b)

Şekil 8. *(a)Firebase ve Realtime veritabanı, (b) Park yeri durumu değişiminin veritabanına yansıması*

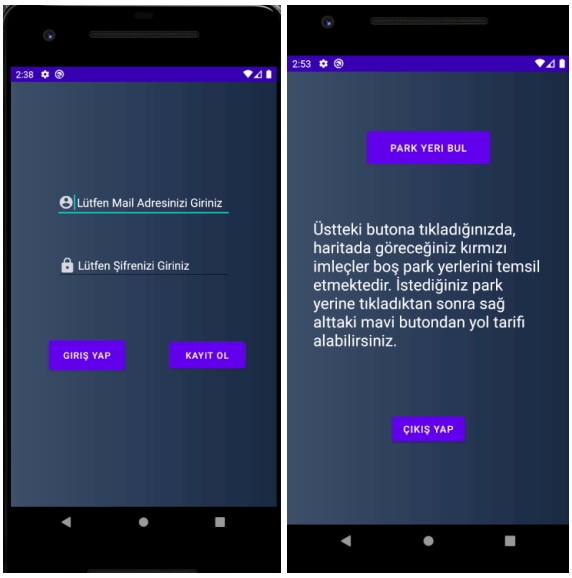
2.2. Android Studio'da Yapılan İşlemler (Transactions Performed in Android Studio)

Bu çalışmanın mobil uygulama kısmı Kotlin dili kullanılarak gerçekleştirilmiştir. Uygulama geliştirilirken Android Studio'ya ait olan emülatör kullanılmıştır. Kullanılan emülatör sayesinde uygulama anlık olarak kontrol edilerek geliştirilme imkânına sahip olunmuştur.

Uygulama ilk defa açıldığında bir kullanıcı adı ve şifre ile giriş yapılması ya da kayıt olunması gerekmektedir. Bu arayüz Şekil 9 (a)'da görülmektedir.

Girilen kullanıcı bilgileri Firebase'e gönderilmekte ve orada tutulmaktadır. Bu sayede kullanıcının bir sonraki girişinde şifre ve kullanıcı ismi kontrol edilebilmektedir. Giriş yapıldıktan sonra ise Şekil 9 (b)'de görülen arayüz ile karşılaşılmaktadır. Bu arayüzde kullanıcıya uygulamanın kullanımı hakkında bir bilgilendirme verilmektedir. Bilgilendirmede kullanıcıya bir sonraki adımda ne yapması gerektiği anlatılmaktadır. Bu bilgilendirme metni dışında ekranda iki adet buton bulunmaktadır. "PARK YERI BUL" butonu kullanıcıyı bir sonraki arayüze geçirmekte "ÇIKIŞ YAP" butonu ise kullanıcının çıkış yapmasını sağlamaktadır. Çıkış butonuna basılmadığı sürece uygulama kapatılsa dahi sonraki girişlerde kullanıcı tanınmakta ve tekrar mail adresi ve şifre bilgisi girmesine gerek kalmamaktadır. Eğer kullanıcı yeni bir

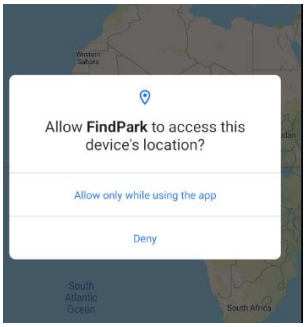
hesap açmak ya da başka bir sebeple çıkış yapmak isterse çıkış butonunu kullanarak uygulamadan ayrılabilir ve bir daha giriş yapmak istediğinde kullanıcı bilgilerini girerek giriş yapabilir.



(a) (b)

Şekil 9. *(a) Kullanıcı girişi ve kayıt için arayüz (b) Kullanıcı arayüzü*

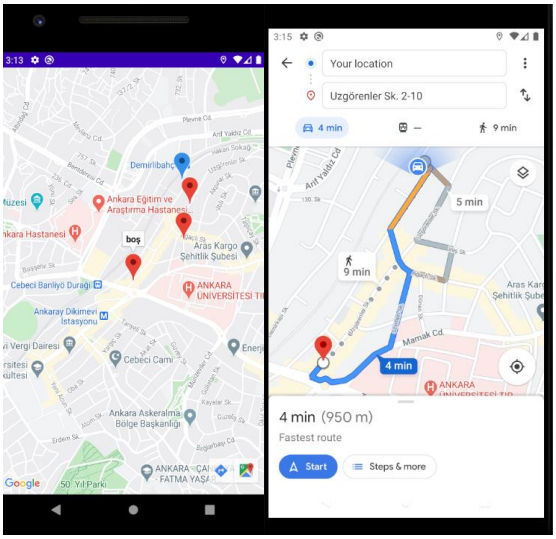
Kullanıcı "PARK YERI BUL" butonuna tıkladığı zaman uygulama boş park yerlerini görüntüleyeceği arayüze geçiş yapmaktadır. Park yerlerinin konumları ve durumları Firebase'de tutulmaktadır. Park yeri bulma butonuna basıldığı an Firebase'den veriler alınmakta ve boş olan park yerleri için haritada imleçler eklenmektedir. Eğer uygulamanın ilk açılışı ise kullanıcıdan konumunun kullanılması amacıyla izin istenmektedir. Daha önce bu izin verildiyse bu ekranla tekrar karşılaşılmamaktadır. Şekil 10'de konum izni isteme ekranı yer almaktadır.



Şekil 10. *Kullanıcıdan konum bilgileri için izin alınma ekranı*

Eğer uygulamanın cihazın konumuna erişmesine izin verilirse harita görünümü açılmakta ve boş olan park yerleri ve kullanıcının konumu haritada görülmektedir. Şekil 11 (a)'da boş park yerlerini haritada gösteren örnek bir görüntü yer almaktadır. Kullanıcı boş park yerlerini gösteren herhangi bir imlece tıkladıktan sonra sağ altta yer alan mavi butona tıklarsa seçtiği boş park yeri konumuna yol tarifi alabilmektedir.

Araç hareketliliği ve trafik yoğunluğu önerilen sistemin çalışmasını etkilememektedir. Kullanıcı uygulamaya giriş yaptığı zaman park yerinin durumunu mevcut şartlarda dolu ya da boş olarak görmektedir. Kullanıcıyı boş park yerine yönlendiren yol tarifi ekran görüntüsü Şekil 11 (b)'te verilmiştir.



(a) (b)

Şekil 11. *(a) Kullanıcının konumunun ve boş park yerlerinin görüldüğü arayüz, (b) Yol tarifi ekranı*

3. 3. BULGULAR VE TARTIŞMA (RESULTS AND DISCUSSION)

Bu çalışma ile kamera görüntülerine erişilebilen ve Bölüm 2.1'de verilen kutu yerleştirme işlemi yapılmış çeşitli bölgelerde araçların park etmesi için uygun olan alanların tespiti yapılarak park yeri durumlarının boş ya da dolu olduğu bilgisi oluşturulmuştur. Önerilen sistem farklı noktalara yerleştirilecek çok sayıda kameradan aynı anda görüntü alarak yol tarifi verme kapasitesine sahiptir. Sistem bir ağa bağlı olarak görüntü aktarımı yapacağı için kapsadığı alan yerleştirilecek kamera sayısına bağlı olarak ağ bağlantısı olan yerlerde arttırılabilir. Bu bilgilerden kendini güncelleyebilen bir veritabanı oluşturulmuştur. Geliştirilen mobil uygulama ile verilerin gerektiği zaman kullanıcıya harita üzerinden gösterilmesi ve kullanıcının kolaylıkla park yerlerinin konumuna dair bilgi sahibi olması sağlanmıştır. Kullanıcıya park yerlerinin yol tarifini sunarak park yerine ulaşması kolaylaştırılmıştır.

Çalışmada önerilen sistem farklı bölgelerden alınan görüntüler ile denenmiş ve %100 başarı elde edilmiştir. Sistemin testi çalışma kapsamında çekilen videolar ile yapılmıştır. Sokak kameralarına hali hazırda erişim bulunmadığı için şu anda yazılımın testleri çeşitli sokak ve park yerlerinin telefon ile video kaydına alınması ile yapılmıştır. Bu şekilde 25 adet video çekilip denenmiştir ve bu denemelerin hepsinde dolu ve boş park yerleri başarılı ile tespit edilmiştir.

Şekil 12'de farklı park yerleri için sistemden elde edilen park yeri tespit sonuçları verilmiştir. Şekilde içi dolu kırmızı dikdörtgen alanlar boş park yerlerini ve içi dolu yeşil dikdörtgen alanlar dolu park yerlerini göstermektedir. Sonuç görüntülerinde görülen diğer küçük dikdörtgenler ise görüntüde tespit edilen nesneleri göstermekte ve sistem tarafından park alanını gösteren kutu bölgeleri içinde araç ifade edecek belirli bir alandan küçük oldukları için dikkate alınmamaktadırlar.

Önerilen sistem nesne tespiti temelli çalıştığından kameranın görüşünü kapatmadığı sürece hava koşullarından etkilenmemektedir. Sistem gün ışığında tam verimle çalışmaktayken hava kararlığında eğer park alanında aydınlatma sorunu varsa verimli çalışmayacaktır. Sokak lambası ya da park yerinin aydınlatması yeterli ise sistem sorunsuz çalışmaktadır. Sistemin bir diğer dezavantajı da sisteme dâhil edilecek yeni park yerleri için ilk etapta kutu bölgelerinin manuel olarak belirlenmesi gerekliliğidir.

Şekil 12. *Farklı park yerleri için sistemden elde edilen park yeri tespit sonuçları*

Uygulamada anlık olarak veri aktarımının güncellenme süresi toplamda yaklaşık 2.5 saniye sürmektedir. Firebase'in güncellenmesi 1 saniye, görüntünün işlenip Firebase'e bilgiyi iletmesi ise 1,5 saniye sürmektedir.

4. SONUÇ VE ÖNERİLER (CONCLUSION AND RECOMMENDATIONS)

Bu çalışmada sürücülerin park yeri arama sorununu ortadan kaldırarak gereksiz yakıt ve zaman kaybı önleme kapasitesine sahip bir uygulama geliştirilmiştir. Bu uygulama park yeri aramak için trafikte yoğunluk yaratmakta olan araçları trafikten çıkararak trafik yoğunluğunu azaltmaya da yardım olma kapasitesine sahiptir.

Uygulamanın geliştirilmesi için atılabilecek adımlardan biri uygulamanın yaygınlaşması durumunda hem uygulamanın daha sağlıklı çalışması hem de kullanıcıların gitmeyi planladıkları park yerinin dolma ihtimalini ortadan kaldırmak için kullanıcının gitmek istediği park yerini işaretlemesi durumunda o park yerinin veritabanında dolu olarak işaretlenmesidir. Böylece başka bir kullanıcıya aynı yer önerilmeyecek ve o park yerinin dolma ihtimali düşecektir.

Teknolojinin ilerlemesi ile uydulardan alınan görüntülerin bu alanda kullanılması mümkün olsa da günümüzde anlık uydu görüntüleri daha ziyade askeri alanda kullanılmaktadır ve anlık olarak alınan

görüntülerin netliği yeterli olmayabilmektedir. Ancak ilerleyen günlerde uydu görüntüleri üzerinde de çalışma imkânı olacağı düşünülmektedir.

Görüntü işleme yöntemleri ile park yeri bulma uygulaması günümüzde yaygınlaşan otonom araçlar için de büyük fayda sağlama kapasitesine sahiptir. Uygulama otonom araca gideceği yerde önceden park yeri bulabilme ve rezerve edebilme imkânı sağlayabilmesi bakımından büyük kolaylık getirecektir.

KAYNAKLAR (REFERENCES)

[1] Önder, H. G., Kaplan, H. (2017). Ankara'da park et-devam et sisteminin modellenmesi ile yolculuk değişimine bağlı emisyon azaltımının ölçülmesi. Gazi University Journal of Science Part C: Design and Technology, 5, 139-152.

[2] İnternet: Inrix, "Searching for Parking Costs Americans $73 Billion a Year", inrix.com/press-releases/parking-pain-us/, Son Erişim Tarihi: 08.09.2022.

[3] Dudaklı, N., Baykasoğlu, A. (2020). Tam otomatik otopark sistemlerinde operasyonel planlama ve kontrol problemleri üzerine bir inceleme. Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi, 35, 2239-2254.

[4] Akıncı, F. C., Karakaya, M. (2017). Şehirlerin dijital dönüşümü: görüntü işleme yöntemlerinin boş park yerlerinin tespitinde kullanılması. TBD 34. Ulusal Bilişim Sempozyumu, Ankara, Türkiye, 20-21.

[5] Gökçe, B., Sonugür, G. (2018) İnsansız kara araçlarından kamera ile görüntülenen hareketli nesnelerin sınıflandırılması amacıyla geliştirilen görüntü işleme tabanlı yöntemlerin karşılaştırılması. Afyon Kocatepe Üniversitesi Fen ve Mühendislik Bilimleri Dergisi, 015901, 1118-1129.

[6] Şenel, F. A., Tokat, S. (2012) Görüntü işleme teknikleri kullanılarak bir ortamın insan yoğunluğunun hesaplanması. ELECO'2012 Elektrik - Elektronik ve Bilgisayar Mühendisliği Sempozyumu, Bursa, Türkiye, 613-617.

[7] Savaş, B. K., İlkin, S., Becerikli, Y. (2016) The realization of face detection and fullness detection in medium by using Haar cascade classifiers. 24th Signal Processing and Communication Application Conference (SIU), Zonguldak, Türkiye, 2217-2220.

[8] Daş, R., Polat, B., Tuna, G. (2019) Derin öğrenme ile resim ve videolarda nesnelerin tanınması ve takibi. Fırat Üniversitesi Mühendislik Bilimleri Dergisi, 31, 571-581.

[9] Dikbayır, H., Bülbül, H. (2020) Derin öğrenme yöntemleri kullanarak gerçek zamanlı araç tespiti. TÜBAV Bilim Dergisi, 13, 1-14.

[10] Amato, G., Carrara, F., Falchi, F., Gennaro, C., Meghini, C., Vario, C. (2016) Car parking occupancy detection using smart camera networks and deep learning. 21th IEEE Symposium on Computers and Communication, Messina, Italy, 1212-1217.

[11] Almeida, P. R. L., Oliveira, L. S., Britto, Jr. A. S., Silva, Jr. E. J., Koerich, A. L. (2015) PKLot–A robust dataset for parking lot classification. Expert Systems with Applications, 42, 4937-4949.

[12] Wu, X., Wang, Y., Chen, H., Shu, L. (2015) A parking management system based on background difference detecting algorithm. 11th International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness, Taipei, Taiwan, 246-250.

[13] Kayış, O., Çakmak, Y., Utku, S. (2018) Mobil cihazlar kullanılarak kapalı alanlarda navigasyon sistemi. Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi, 24, 238-245.

[14] Karasulu, B. (2013) Videolardaki hareketli nesnelerin tespit ve takibi için uyarlanabilir arkaplan çıkarımı yaklaşımı tabanlı bir sistem. Uludağ Üniversitesi Mühendislik-Mimarlık Fakültesi Dergisi, 18, 93-110.

[15] Karaköse, M., Baygın, M., Aydın, İ., Sarımaden, A., Akın, E. (2016) Endüstriyel sistemlerde arkaplan çıkarımı tabanlı hareketli nesne tespiti ve sayılması için yeni bir yaklaşım. Muş Alparslan Üniversitesi Fen Bilimleri Dergisi, 4, 373-381.

[16] Solak, S., Altınışık, U. (2018) Görüntü işleme teknikleri ve kümeleme yöntemleri kullanılarak fındık meyvesinin tespit ve sınıflandırılması. Sakarya Üniversitesi Fen Bilimleri Enstitüsü Dergisi, 22, 56-65.

[17] Hanbay, K., Üzen, H. (2017) Nesne tespit ve takip metotları: kapsamlı bir derleme. Türk Doğa ve Fen Dergisi, 6, 40-49.

[18] Chowdhury, T., Rahman, S. A., Islam, F., Mallick, R. (2019) Automated Car Parking System with Increased Security by Digital Image Processing, Brac University, Department of Computer Science and Engineering.

[19] Lopez, M., Griffin, T., Ellis, K., Enem, A., Duhan, C (2019) Parking lot occupancy tracking through image processing. EPiC Series in Computing, 58, 265–270.

[20] Kuyumcu, B. (2018) OpenCV Görüntü İşleme ve Yapay Öğrenme, Level Kitap, İstanbul,Türkiye.

[21] Eldem, A., Eldem, H., Palalı, A. (2017) Görüntü işleme teknikleriyle yüz algılama sistemi geliştirme. Bitlis Eren Üniversitesi Fen Bilimleri Dergisi, 6, 44-48.

[22] İnternet: Webtekno/Konaray O, "Adım Adım Google Android Studio: Nedir, Nasıl Kullanılır?", https://www.webtekno.com/google-Android-stuido-indir-h92271.html, Son Erişim Tarihi: 02.02.2022.

[23] KaewTraKulPong, P., Bowden, R. (2002) An Improved Adaptive Background Mixture Model for Real-Time Tracking with Shadow Detection. Editors: Remagnino, P., Jones, G. A., Paragios, N., Regazzoni, C. S., Video-Based Surveillance Systems, Springer, Boston, Massachusetts, USA, 135-144.

[24] Zivkovic, Z. (2004) Improved adaptive Gaussian mixture model for background subtraction. 17th International Conference on Pattern Recognition, Cambridge, UK, 28-31.

[25] Zivkovic, Z., Heijden, F van der. (2006) Efficient adaptive density estimation per image pixel for the task of background subtraction. Pattern Recognition Letters, 27, 773-780.

[26] Godbehere, A., Matsukawa, A., Goldberg, K. (2012) Visual tracking of human visitors under variable-lighting conditions for a responsive audio art installation. American Control Conference, Montreal, Canada, 4305-4312.

[27] Simaremare, H., Melwanda, H., Abdillah, A. (2017) The comparison of vehicle speed accuracy using video based mixture of Gaussian 2 method and k-nearest neighbor method. Seminar Nasional Teknologi Informasi Komunikasi dan Industri, Endonezya, 1-5.

[28] Viola, P., Jones, M. J. (2004) Robust real-time face detection. International Journal of Computer Vision, 57, 137–154.

[29] Dandıl, E., Özkul, İ. (2019) Futbol Maçları İçin Bilgisayarlı Görü Destekli Gol Karar Sistemi(GolKaSis): Bir Prototip Çalışma. Gazi University Journal of Science Part C: Design and Technology, 7, 213-224.

[30] İnternet: OpenCV, "CascadeClassifier", https://docs.OpenCV.org/3.4/db/d28/tutorial_cascade_classifier.html, Son Erişim Tarihi: 08.09.2022.