



**Makale / Research Paper**

**Elektrikli ve Otonom Araçlarda Makine Öğrenmesi Kullanarak Trafik Levhaları Tanıma ve Simülasyon Uygulaması**

Fatma Nur ORTATAŞ<sup>1a</sup>, Emrah ÇETİN<sup>1b\*</sup>

<sup>1</sup>Yozgat Bozok Üniversitesi, Mühendislik Fakültesi, Elektrik-Elektronik Mühendisliği Bölümü, 66200 Yozgat/TÜRKİYE, [emrah.cetin@yobu.edu.tr](mailto:emrah.cetin@yobu.edu.tr)

Received/Geliş: 25.01.2021

Accepted/Kabul: 16.06.2021

**Öz:** Trafik kazaları büyük oranda sürücü kusurlarından ve trafik kurallarına uyulmamasından kaynaklanmaktadır. Son yıllarda yapılan çalışmalar ile elektrikli ve otonom araç teknolojilerindeki gelişmeler çok hızlı bir şekilde ilerlemektedir. Geliştirilen sistemler ile sürücü kaynaklı kazaların önüne geçilmesi amaçlanmaktadır. Otonom araçlardaki en temel problem ise gerçek zamanlı olarak hareket ederken trafik işaret ve işaretçilerinin tanımlanmasıdır. Otonom araçlarda kullanılan sensör ve kamera gibi algılayıcılardan gelen verilerin kurulan algoritmalar ile anlamlı sonuçlara dönüştürülmesi sağlanmaktadır. Bu sayede aracın sürücüdün bağımsız olarak trafik kurallarına uygun şekilde hareket etmesi hedeflenmektedir. Bu çalışma kapsamında Türkiye’de kullanılan trafik işaret ve işaretçileriyle ilgili veriler toplanarak on adet trafik işaret ve işaretçisine ait farklı veri setleri oluşturulmuştur. Haarcascade makine öğrenmesi algoritması ile veri seti dâhilinde eğitimleri gerçekleştirilmiştir. Haarcascade yöntemi ile eğitimi gerçekleştirilen levhalar üç boyutlu simülasyon ortamında oluşturulan parkur üzerinde tanımlanmıştır. Parkur üzerinde otonom aracın davranışları simüle edilerek eğitim verileri test edilmiştir. Bu veriler ile simülasyon ortamında otonom aracın hareket kontrolü başarılı bir şekilde gerçekleştirilmiştir.

**Anahtar Kelimeler:** Haarcascade; otonom araç; levha tanıma; makine öğrenmesi.

**Traffic Sign Recognition and the Application of Simulation Using Machine Learning in Electric and Autonomous Vehicles**

**Abstract:** Traffic accidents are substantially caused by driver faults and breaking the rules. With the studies conducted in recent years, developments in electric and autonomous vehicle technologies are progressing very rapidly. By the systems developed, it is aimed to prevent the driver-induced accidents. The main problem in autonomous vehicles is the recognition of traffic signs and markers while running in real time. The data from the sensors and cameras used in autonomous vehicles are transformed into meaningful results with established algorithms. In this way, it is aimed that the vehicle will act in accordance with the traffic rules, independent of the driver. Within the scope of this study, the data of traffic signs and markers used in Turkey were collected and different data sets of six signs were created. Using these data sets, trainings were carried out via Haarcascade machine learning algorithm. The traffic signs trained via the Haarcascade method were defined on the track created in a three-dimensional simulation environment. Training data were tested by simulating the behavior of the autonomous vehicle on the track. In the simulation environment, the motion control of the autonomous vehicle was successfully performed by this data.

**Keywords:** Haarcascade; autonomous vehicle; recognition of traffic signs; machine learning.

**1. Giriş**

Elektrikli araçlar bundan bir asır önce otomobillerin ilk çıktığı yıllarda da üretilmekteydi. Ancak fosil yakıtların cazibesinden dolayı gelişim imkânı bulamayarak geri planda kalmıştır. Son yıllarda ise hem fosil yakıtların cazibesini yitirmesi hem de teknolojinin geldiği seviye elektrikli araçları

*Bu makaleye atf yapmak için*

Ortataş, F. N., Çetin E. “Elektrikli ve Otonom Araçlarda Makine Öğrenmesi Kullanarak Trafik Levhaları Tanıma ve Simülasyon Uygulaması” El-Cezerî Fen ve Mühendislik Dergisi 2021, 8 (3); 1081-1092.

*How to cite this article*

Ortataş, F. N., Çetin E. “Traffic Sign Recognition and the Application of Simulation Using Machine Learning in Electric and Autonomous Vehicles” El-Cezerî Journal of Science and Engineering, 2021, 8(3); 1081-1092.

ORCID ID: <sup>a</sup>0000-0001-7897-9958, <sup>b</sup>0000-0002-7023-6604

yeniden gündeme getirmiştir. Çevreci yapısı bu gelişimin en önemli itici unsurlarından biridir. İçten yanmalı motorların yaydığı karbon salınımı araç sayısındaki artışın da etkisiyle artık kabul edilebilir düzeyi aşması bütün dünyada daha temiz otomobillere olan arayışı hızlandırmıştır. Teknoloji bu şekilde evrilirken, son birkaç yılda bunun üzerine sürücüsüz araç konsepti de eklenmiştir. Elektrikli araçların otonom araçlar için uygun yapıda olması da bunun başlıca nedenleri arasındadır. Çünkü elektrikli araçta mekanik kontrol minimum seviyededir. Bir elektrikli araçta aracın hareketi, frenleme işlemi, şarjı ve direksiyon kontrolü gibi neredeyse bütün aksiyonları elektrik sinyalleriyle yapılabilmektedir. Bu da yazılıma dayalı kontrolü kolaylaştırmaktadır[1]. Yapay zekâ birçok alanda kullanılmaktadır [2]. Bunlardan birisi de elektrikli araca sağlanan çeşitli duyarlar (sensörler) vasıtasıyla aracın artık hissedebilir bir yapıda olması sağlanmıştır. Bu sayede araçlar sürücüsüz olarak geliştirilebilmektedir.

Günümüzde hızla ilerleyen teknoloji ve akıllı sistemlerdeki gelişmeler dâhilinde araçlardaki duyarlar daha etkin bir şekilde kullanılabilme imkânına kavuşmuştur. Sensörlerden aldığı verileri değerlendirip öğrenebilen, öğrendikçe gelişen otonom araçlar üzerindeki çalışmalar da hız kazanmıştır. Aynı zamanda akıllı sistemlerin elektrikli ve otonom araçlarda kullanımı da yaygınlaşmaya başlamıştır. Bir yandan teknolojisini geliştirirken, diğer yandan da otonom araçlar üzerindeki çalışma ve gelişimlerin standartlaştırma çalışmaları yapılmaktadır. Araç otomasyonunun standartlaşması için Uluslararası Otomotiv Mühendisleri Derneği (OMD- Society Of Automotive Engineers- SAE) 2018 yılında bir makale yayınlamıştır. Bu makalede 6 seviyeden oluşan bir sınıflandırma tanımlanmıştır. Bu sınıflar aşağıdaki gibidir [1].

- Seviye 0. Otomasyon yok. Sürücü tüm sürüş görevlerini yerine getirir.
- Seviye 1. Sürücü yardımı. Araç sürücü tarafından kontrol edilir ancak bazı sürüş destek özellikleri araç tasarımına dâhil edilebilir.
- Seviye 2. Kısmi otomasyon. Araç, hızlanma ve direksiyon kullanımı gibi otomatik fonksiyonları bir araya getirir ancak sürücünün sürüş durumuna bağlı kalması ve ortamı her zaman izlemesi gerekir.
- Seviye 3. Koşullu otomasyon. Sürücü bulunması zorunlu ancak ortamı sürekli izlemesi zorunlu değildir. Sürücünün, aracın önceden vereceği uyarılara karşı aracı her an kontrol etmek için hazır bulunması gerekir.
- Seviye 4. Yüksek otomasyon. Araç, belirli koşullar altında tüm sürüş fonksiyonlarını yerine getirebilir. Sürücünün aracı kontrol etme seçeneği olabilir.
- Seviye 5. Tam otomasyon. Araç her koşulda tüm sürüş fonksiyonlarını yerine getirebilir. Sürücünün aracı kontrol etme seçeneği olabilir.

Günümüzde kullanılan araçlar bu sınıflandırma için 2 veya 3. seviyede bulunmaktadır [2]. Otonom araçlar için seviye 4 ve seviye 5 tanımlanmaktadır. Bu araçlara olan ilgi her geçen gün daha da artmaktadır. Otonom teknolojinin gelişim potansiyeli açıkça görülmektedir. Otonom araçların kullanılmaya başlanmasıyla ulaşım ve taşımacılıkta yaşanan birçok problem ve aksamaların ortadan kalkacağı ve güvenliğin artacağı ön görülmektedir [3].

Elektrikli ve otonom bir aracın çevresinden bilgi almanın ilk yolu kamera ve sensörlerdir. Günümüzde kullanılan gelişmiş teknoloji ile üretilen sensörlerin otonom araçlarda kullanımı da verimli ve aktif bir şekilde gerçekleşmektedir. Otonom aracın temel görevi trafik işaret ve işaretçilerine uyararak şerit takibi yapabilmesi, bireysel bir araç ise yolcu, taşımacılık için kullanılıyorsa ürün güvenliğini sağlayarak gerekli kamera ve sensörlerden gelen verileri değerlendirerek güvenli bir sürüşün gerçekleşmesidir. Bu amaç için otonom araçlarda kullanılacak olan yazılımların en önemli kısımları nesne tanıma algoritmalarıdır. Son yıllarda bilgisayarlı görü ve makine öğrenmesi gibi alanlarda önemli çalışmalar yapılmıştır [4]. Otonom araç için en temel

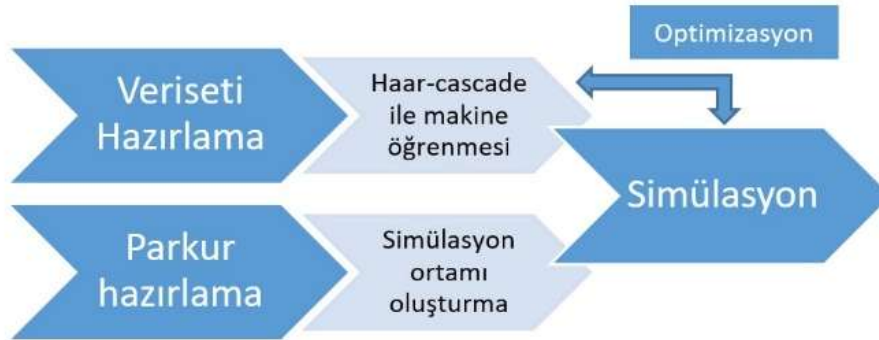
özellik trafik işaret ve işaretçilerinin algılanması, sınıflandırılması ve takip edilmesi ve uygun davranışı gerçekleştirmesidir.

Trafik ve yol işaretleri çevreden farklı olan ve araç sürücüsü tarafından kolaylıkla fark edilen şekil ve renge sahiptir. Trafik işaretçilerinde yoğun olarak mavi, kırmızı ve sarı renk kullanılmaktadır. Kullanılan bu levhaların renkleri anlamlarına göre sınıflandırılır. Mavi renk zorunluluk, kırmızı renk uyarı ve sarı renk ise tehlike olduğunu belirtir [5]. Trafik işaret ve işaretçilerinin algılanmasında birçok yöntem uygulanmaktadır. Bu algılama yapıları farklı veri yapılarına, eğitim metotlarına ve algılama işlevlerine sahip birçok türeve sahiptir. Çoğu yaklaşım için trafik işaretçilerinin iki önemli fiziksel özelliği olan renk ve şekli kullanılır. Renk tabanlı algılama yöntemi trafik işaret ve işaretçilerinde mevcut olan mavi, kırmızı ve sarı renkleri ve yoğunluk algılama tekniği kullanılır. Bu yöntem ile görüntünün renk kanallarına ayrılması sağlanır ve ayrılan bu renk piksellerinin birbiriyle bağlantılı olacak şekilde en büyük alanı oluşturulur. Yoğunluk bilgisi ise görüntünün HSV renk uzayına dönüştürülmesiyle elde edilir. Şekil tabanlı algılama da ise trafik işaretlerinin ortak standart şekilleri olarak üçgen, daire, dörtgen ve sekizgen kullanılmaktadır. Şekil tabanlı algılamada bu özellikler ön plana çıkarılarak algılama yapılır. Bu özellikler için ise standart şekillerin sınır çizgileri, yapı ve dokuları ile kilit noktaları kullanılır. Bu özellikleri algılamak için ise genellikle Hough dönüşümü uygulanır. Hough dönüşümü en temel anlamda görüntüdeki çizgilerin tanımlanması için kullanılır. Son yıllarda önemli gelişmelerin yaşandığı makine öğrenmesi ile makine öğrenmeye dayalı algılama yöntemleri geliştirilmiştir. Trafik işaret ve işaretçilerini algılamak için AdaBoost, Support Vector Machine (SVM), Neural Networks (NN) gibi makine öğrenme yöntemleri kullanılmaktadır. Bu yöntemlerin farklı eğitim şekilleri ve farklı tespit süreçleri vardır. Bu tespit süreçlerinde ise farklı özelliklere sahip birçok girdinin türevi mevcuttur. AdaBoost ve Cascade tabanlı algılama yapıları yüz tanıma, araç tanıma, plaka tanımlama gibi birçok nesne tanımlama probleminde başarılı olmuştur [6]. Bu yapı aynı zamanda trafik işaret ve işaretçilerinin algılanmasında da başarıyla uygulanmıştır. AdaBoost tabanlı öğrenme metodu ile Cascade yapısı birleştirildiğinde bazı özellikleri seçebilir ve nesnelere algılayabilir hale gelir. AdaBoost tabanlı algılama yöntemleri için özellik seçimi çok önemlidir ve bu algılama yapısı oldukça hızlıdır. Haar benzeri özellikleri farklı tespit problemlerinde oldukça önemli özelliklerdir. Bu özellikler görüntüdeki dikdörtgensel piksellerin toplamının farkı olarak ifade edilebilir. Burada kastedilen dikdörtgensel piksel ifadesi konuma (x, y) sahip pikseller için kullanılır. Ayrıca bahsi geçen toplamların farkı ifadesi ise Haar özellik çıkarımı için kullanılan bir ifadedir ve piksellerin sayısı ve yoğunluk hesaplamaları sonucunda yapılan işlemleri ifade etmektedir. Kenar, çizgi ve dört kare özellikleri olmak üzere 3 bölgeye ayrılabilir. Cascade yapısı AdaBoost tabanlı algılamadaki en önemli yapıdır. SVM ve Histograms of Oriented Gradients (HOG) tespit yapısı öncelikle yayaların tespiti için kullanılmıştır. Bu algılama yapısı ise nesnelere ifade etmek için HOG benzeri özellikleri kullanır [7]. HOG benzeri özellikleri, bir görüntüyü küçük kare hücrelere ayırır ve bu hücrede yönlendirilmiş gradyanların bir histogramını hesaplar ve görüntüyü normalleştirerek sunar. Burada bahsedilen normalleştirme işlemi uygulanan gradyanların orijinal görüntü üzerine uygulanmasıdır. Daha açık bir ifadeyle görüntü üzerinde hesaplanan histogramlara ait yönlendirmeler gerçek görüntü ile birleştirilerek sunulur. SVM tabanlı algılama yöntemleri için HOG benzeri özelliklerinin kullanımı çok önemlidir. Trafik işaret ve işaretçilerinin algılanmasında bu yöntem başarıyla kullanılmıştır [8]. Convolutional Neural Network (CNN) tabanlı algılama yöntemleri evrimsel ağ yoluyla özellikleri öğrenir. Derin öğrenme yöntemlerinin geliştirilmesiyle CNN trafik işaret ve işaretçilerinin tespiti için kullanılmıştır. Bu nesnelere tanımlamak için bir CNN sınıflandırıcısı kullanılır. CNN tabanlı algılama ağlarının çoğu nesneyi algılamak için oldukça yavaştır. Bu ağlardan You Only Look One (YOLO) oldukça hızlı performansla sahiptir [9]. YOLO sinir ağı temel olarak bir videodan tek bir görüntü çıkararak işe başlar bir sonraki adımda ise çıkarılan bu görüntüyü yeniden boyutlandırır ve bu YOLO ağına girişi temsil eder. YOLO ağı katmanlardan oluşur. Her bir katmanda farklı yapılar mevcuttur. YOLO ağı bir görüntüyü giriş olarak ve üzerinde sınır kutusu çizerek koordinat bilgisini alır, kutu içerisinde nesne olma olasılığını

değerlendirir ve kutu içerisinde bulunan nesnelerin herhangi bir sınıfa ait olup olmadığını kontrol ederek çalışır [10].

Tüm bu algılama yöntemlerinin uygulanması ve öğrenme yapabilmesi için trafik işaret ve işaretçilerinin bulunduğu veri setlerine ihtiyaç vardır. Açık kaynak olarak bulunabilecek veri setleri Almanya, Belçika, Çin, Amerika, İsviçre, Fransa gibi ülkelerdeki trafik işaretleri için mevcuttur. Ancak Türkiye için bu çalışmaların çok yeni olmasından dolayı bir veri seti bulunamamıştır.

Bu çalışmada trafik işaret ve işaretçilerini tanımlamak için Haarcascade algoritması kullanılmıştır. Literatürde bu yapının yüz tanıma, yaya tanıma ve plaka tanıma için yapılmış örneklerinin olmasına rağmen levha tanıma uygulaması bulunmamaktadır. Bu kapsamda ilk olarak Türkiye Cumhuriyeti Karayolları Genel Müdürlüğüne ait trafik işaret ve işaretçilerden oluşan bir veri seti hazırlanmıştır. Daha sonra bu veri setinin Haarcascade algoritması ile makine öğrenmesi gerçekleştirilmiştir. Bu öğrenmenin doğruluğunu kanıtlayabilmek için de bir simülasyon ortamı hazırlanmıştır. VectorZero RoadRunner programı ile üç boyutlu tanımlanan bir parkur üzerinde trafik işaret ve işaretçilerinin bulunduğu bir yol tanımlanmıştır. Simülasyon ortamı içerisinde trafik akışında bir sürücünün karşısına çıkabilecek ve Haarcascade yapısı ile öğrenmesi gerçekleştirilen muhtemel levhalar yerleştirilmiştir. Oluşturulan bu parkur, Unreal Engine platformuna eklenmiş ve bu parkuru kullanabilecek kamerası olan bir otonom araç tanımlanmıştır. Simülasyon çıktılarına göre de optimizasyon çalışmaları gerçekleştirilmiştir. Bu çalışma kapsamındaki otonom aracın simülasyonuna ait geliştirme akış şeması Şekil 1’de verilmiştir. Sonuçta simülasyondaki otonom aracın trafik işaret ve işaretçilerini büyük oranda bir doğrulukla tanımladığı ve bu levhalara uygun şekilde hareket etmesi sağlanmıştır.



Şekil 1. Otonom araç simülasyonu geliştirme akış şeması

## 2. Veri Seti Hazırlama

Makine öğrenmesi ve derin öğrenme işlemlerinin gerçekleştirilmesi için ilk gereksinim veri setidir. Veri seti, kullanılacağı alana bağlı olarak matematiksel ifadelerden veya görsellerden oluşabilmektedir. Bu çalışma kapsamında gerçek zamanlı olarak trafik işaret ve işaretçilerinin tespiti gerçekleştirileceği için veri seti bunların görüntülerinden oluşmaktadır. Dijital işlemler için bir görüntü matristen oluşmaktadır. Bu matris 0-255 arasındaki değerleri içermektedir. Bu veri seti ile trafik işaret ve işaretçilerinin bir fotoğraf, video veya gerçek zamanlı bir görüntüde bulunabilmesi için uygulanacak Haarcascade algoritması ile makine öğrenmesi gerçekleştirilmiştir. Bu işlem için ilk olarak kullanılacak verinin toplanması gerekmektedir. Veri toplama işlemi manuel fotoğraf çekimleri ile elde edilmiştir. Görüntülerin toplanması esnasında dikkat edilmesi gereken 3 önemli husus vardır.

1. *Farklı konumlar*, makine öğrenmesinde kullanılacak bir fotoğraf, video veya gerçek zamanlı bir görüntüde bir nesnenin bulunması için veri setinde kullanılacak o nesneye ait

görüntülerin farklı açılardan çekilmiş görüntülerden oluşması gerekir. Bu otonom bir aracın trafik işaret ve işaretçilerinin tespitinde yolun sağ veya sol kısmında bulunmasından veya farklı yüksekliklerde bulunmasından kaynaklanan hatalı nesne tespitini minimize etmede önem arz etmektedir.

2. *Farklı aydınlatma şartları ve seviyeleri*, dikkate alınarak çekilmiş olan fotoğraflardan oluşan veri seti ile gerçekleştirilen makine öğrenmesi günün farklı saatlerinde muhtemel karşılaşılabilecek olan ışık seviyelerindeki görüntülerde nesne tespitini kolaylaştıracaktır.
3. *Farklı mesafeler* gözetilerek çekilmiş görüntülerden oluşan bir veri seti ile gerçekleştirilen makine öğrenmesi sayesinde araç tarafından farklı uzaklıklardan trafik işaret ve işaretçilerinin tespiti sağlanacaktır.

Bu hususlara dikkat edilmesi sonucunda gerçekleştirilen makine öğrenmesi çok daha verimli olacaktır. Verilerin toplama işlemi tamamlandıktan sonra temizlenmesi gerekmektedir. Bu işlem için makine öğrenmesi gerçekleştirildikten sonra her bir görüntünün bulunması istenen nesne görüntüden kırılarak ayrılır. Haarcascade algoritması ile makine öğrenmesinin gerçekleştirilmesinde görüntülerin etiketlenme işlemi yoktur. Etiketleme işlemi görüntü üzerinde istenen nesneye veya kişiye ait anlamlı piksellerin bulunduğu kısımların işaretlenmesini sağlamaktadır. Bu işlem Haarcascade ile makine öğrenmesinin gerçekleştirilmesinde, tespiti istenen nesnenin fotoğraflardan kırılarak ayırt edilmesi ile sağlanmaktadır.

Veri setinin oluşturulmasındaki son işlem ise görüntülerin yeniden boyutlandırılması işlemidir.

*Pozitif görüntüler*, Haarcascade algoritmasının uygulanmasında kullanılacak olan tespitinin gerçekleştirilmesi istenen nesnelerin görüntülerine verilen isimdir. Kullanılacak tüm pozitif görüntülerin boyutları birbirine yakın olacak şekilde yeniden boyutlandırılması gerekmektedir. Veri seti hazırlama aşamaları Şekil 2’ de görülmektedir.



Şekil 2. Veri seti hazırlama aşamaları



Şekil 3. Pozitif görüntülerin temizlenmesi ve yeniden boyutlandırılması

Şekil 3’ te ise temizlenen ve yeniden boyutlandırılan görüntü örnekleri verilmiştir.

*Negatif görüntüler*, tespit edilmesi istenilen nesnenin içerisinde bulunmasının muhtemel olduğu fotoğrafları içerir. Bunlar nesnenin içerisinde bulunmayan görüntülerdir. Bu pozitif fotoğrafların her biri farklı sayılarda negatif resimlere uygulanarak kullanılmıştır. Böylelikle mevcut pozitif görüntü sayıları negatif örneklere uygulanarak çok daha fazla veri üretilmiştir.

Tablo 1’de simülasyon çalışmasında kullanılan levhalara ait veri seti bilgileri gösterilmektedir.

**Tablo 1.** Simülasyon çalışmalarında kullanılan veri seti içeriği

Trafik İşaret ve İşaretçi İsmi	Görüntü Sayısı
1. Trafik Lambası	120
2. Dur Levhası	100
3. Park Levhası	110
4. Sola Dönüş Yasak Levhası	100
5. Durak Levhası	125
6. Girilmez Levhası	118
7. İleri ve Sağa Mecburi Yön Levhası	100
8. Park Yasak Levhası	90
9. Trafiğe Kapalı Yol Levhası	100
10. İleri ve Sola Mecburi Yön Levhası	95

Veri setini oluşturan pozitif görüntülerin boyutları yaklaşık olarak 95\*95 ile 110\*110 arasında değişkenlik gösterecek şekilde yeniden boyutlandırma yapılmıştır. Bir levhaya ait farklı sayılara sahip veri setleri ile makine öğrenmesi gerçekleştirilmiştir. Sonuçlar simülasyon ortamında ayrı ayrı test edilmiştir. Bu test sonucunda başarı sağlayan verilerin sayısı ve içerik bilgisi Tablo-1’ de verilmiştir.

Simülasyon ortamında aracın otonomi hareketi ile nesne tespitlerinin farklı açılardan sağlanarak tespit edilen nesne üzerinden bilgi aktarılması ve belirli bir mesafeden bu işlemlerin gerçekleştirilmesi yapılan çalışma için başarı kriterleridir.

#### 4. Haarcascade Sınıflandırıcısı

P. Viola ve M. Jones tarafından 2001 yılında geliştirilen ilk gerçek zamanlı nesne algılama yapısıdır [11]. Çeşitli nesne sınıflarının tespiti için eğitilebilir. Genellikle yüz tespiti için kullanılmıştır. Bu algoritmanın temel prensibi belirli bir giriş görüntüsünde tespit edilecek olan nesneye ait alt pencere tespitinin sağlanmasıdır. Algılanacak olan bu alt pencereler yardımıyla tanıma işlemi gerçekleştirilmektedir. Temel olarak bir görüntüde bulunan nesnelerin farklı renk dağılımları, kontrast oranları ve doygunluk seviyeleri gibi görüntüye ait bilgilerden faydalanılarak görüntü alt pencerelere ayrılır. Bu farklı özelliklere Haar özellikleri denir. Bu özellikler nesne tespitini sağlayan sayısal görüntü özellikleridir. Ayrılan bu alt pencereler nesneye ait farklı bilgiler taşıyacağı için bu pencereler temelde ayrı özelliklere sahip olsa da bütünde nesneyi ifade etmektedir. Haar özelliklerinden faydalanılarak nesne tanıma işlemi gerçekleştirilmektedir [12,13]. Şekil 4’ de Haarcascade sınıflandırıcısı için Haar özelliklerinin tespitini sağlayan öznitelikler gösterilmektedir. Haarcascade algoritmasını oluşturan adımlar aşağıda verilmiştir.

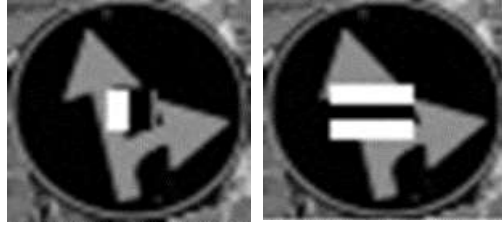
- a. **Haar Özellik Çıkarımı:** Haar- Cascade algoritmasının eğitilmesi için gerekli pozitif görüntüler ve negatif görüntülere ihtiyaç vardır. Bu görüntüler üzerinden özellik çıkarma işlemi gerçekleştirilecektir.

Öznitelik	Özellik Resimleri			
<b>Kenar öznitelikleri</b>				
<b>Satır Öznitelikleri</b>				
<b>Merkez-Çevre Öznitelikleri</b>				

**Şekil 4.** Haarcascade sınıflandırıcısında kullanılan Haar öznitelikleri



Bu işlem ise uygulanacak öznelikler ile çıkarılacak her özellik, beyaz dikdörtgen altındaki piksellerin toplamının siyah dikdörtgen altındaki piksellerin toplamından çıkarılmasıyla elde edilen tek bir değerdir. Tablo-2' de verilen özneliklerin uygulanması ile giriş görüntüsü üzerindeki özellik çıkarma işlemi gerçekleştirilir. Şekil-5' te Haar özelliklerinin uygulanması görülmektedir.

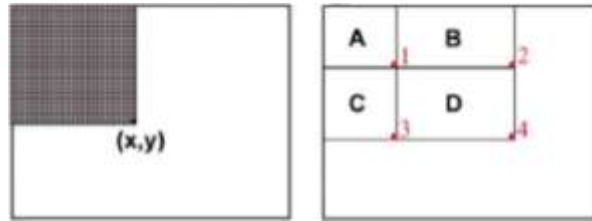


Şekil 5. Haar özellikleri ve uygulanması

- b. İntegral Görüntünün Oluşturulması:** Haar özelliklerin çıkarılması için gereken hesaplamaların hızlı yapılması için integral görüntü kullanılır. İntegral görüntünün  $(x, y)$  konumundaki değeri, bu konuma ait noktaların üstünde ve solunda kalan piksel değerlerinin toplamı ile bulunur. Bunu ifade eden denklem (1)' de verilmiştir.

$$ii(x,y) = \sum_{x' \leq x \text{ ve } y' \leq y} i(x',y') \quad (1)$$

Denklem (1)'de eşitliğin sol kısmında kalan  $ii(x, y)$  integral görüntüyü,  $i(x',y')$  ise kullanılan orijinal giriş görüntüsünü ifade etmektedir. İntegral görüntünün oluşturulması için uygulanacak özneliğe göre hesaplamalarda kullanılacak olan referans sayısı değişmektedir [14].



Şekil 6. İntegral görüntü oluşumu

Şekil 5'te verilen satır özneliklerinin uygulandığı bir görüntünün integral görüntüsü oluşturulurken siyah ve beyaz 3 dikdörtgensel toplam içerdiği için 8 referansla hesaplanabilmektedir. Şekil 6' da verilen  $(x, y)$  konumundaki görüntünün integral görüntüsü oluşturulurken A dörtgeninin içindeki piksel toplamları, integral görüntüsünün 1 konumundaki değeridir. 2 konumundaki değer A+B dörtgenine, 3 konumundaki değer A+C, 4 konumundaki değer ise A+B+C+D dörtgenine aittir. D dörtgenin piksel toplamı ise  $4+1-(2+3)$  şeklinde hesaplanır.

- c. Sınıflandırıcının Eğitilmesi:** Orijinal görüntü üzerinde özneliklerin kaydırılarak tüm resmin taranması sağlanır. Bu tarama sonucunda elde edilen binlerce özelliğin birçoğu ilişkisizdir. Burada özellik sayısı eğitimde kullanılan görüntünün boyutuyla ve oluşturulan alt pencere boyutuyla doğrudan ilişkilidir. Oluşan binlerce alt pencerenin küçük bir kısmı etkili bir sınıflandırıcının oluşturulmasında kullanılır. Bu işlemin gerçekleştirilmesi için AdaBoost öğrenme algoritmasından faydalanılır ve her bir özelliği eğitim için kullanılan pozitif ve negatif görüntüler üzerinde uygulayarak her özellik için sınıflandırma gerçekleştirecek eşik değeri tespit edilir. Böylelikle minimum hata oranlarına sahip özellikler seçilerek pozitif ve negatif görüntülerin doğru şekilde sınıflandırılması sağlanır. Bu tür sınıflandırıcılara zayıf sınıflandırıcılar ismi verilir. Zayıf sınıflandırıcılar tek başına görüntü üzerinde sınıflandırma gerçekleştiremez. Etkili bir sınıflandırıcı birkaç zayıf sınıflandırıcının birleşmiş halinden

oluşur. Her bir zayıf sınıflandırıcı eklendiğinde doğru sınıflandırma yapılan verilerin ağırlıkları artırılırken yanlış olan örneklerin ağırlıkları azaltılır. Böylelikle istenen doğru tespit ve yanlış alarm oranına erişildiğinde zayıf sınıflandırıcı ekleme durumu durdurulur.

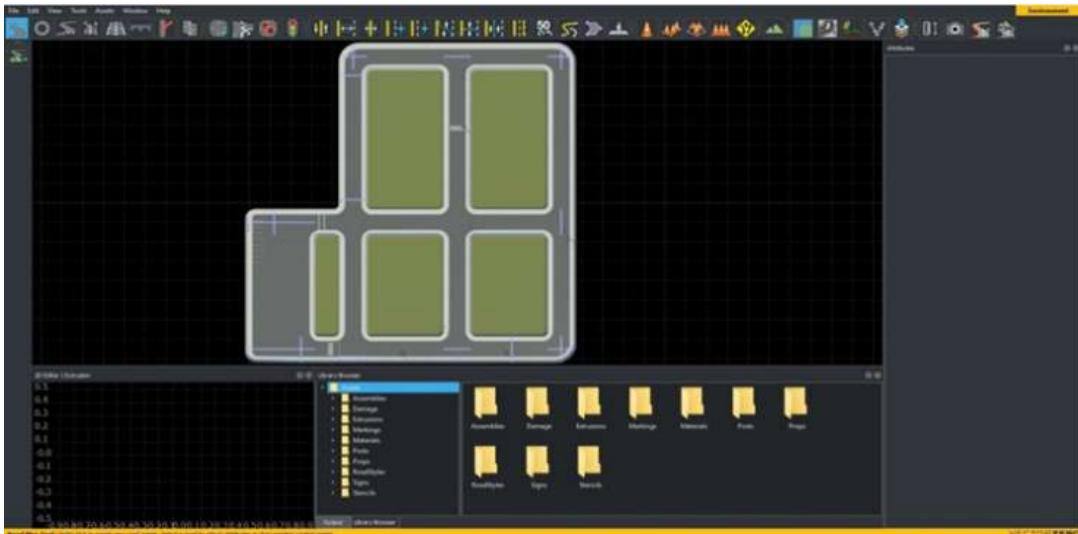
- d. Cascade Sınıflandırıcısı:** Bir giriş görüntüsü üzerinde istenen nesnenin bulunduğu kısım görüntüde küçük bir kısmı kapsamaktadır. Bu nedenle sınıflandırıcılar kullanılarak aranan nesnenin bulunmadığı bölgeler görüntü üzerinden çıkarılarak nihai hedef olan nesnenin bulunduğu kısmın tespitinin sağlanmasıdır. Böylelikle çıkarılan görüntüler algoritma boyunca tekrar işlenmez ve tespit edilecek nesnenin bulunabileceği alanlara odaklanarak devam edilir. Bu nedenle sınıflandırıcının eğitilmesi aşamasında belirlenen özelliklerin tümünü bir pencereye uygulamak yerine özellikleri sınıflandırıcının farklı aşamalarında gruplandırarak uygulanması sağlanır. Bu nedenle eğitimin ilk aşamalarında çok daha az sayıda özellik içerir ve ilerleyen aşamalarda tespit edilen özellik sayısı artar ve nihai sonuç olarak tespit işlemi gerçekleşir. Şekil 7’ de Cascade sınıflandırıcı işlem süreci görülmektedir.



Şekil 7. Cascade sınıflandırıcı işlem basamakları

## 5. Simülasyon Ortamı

Çalışmanın gerçekleştirilmesi için kullanılacak olan simülasyon ortamı VectorZero Road-Runner programı ile Şekil 8’ de verilen parkur çizilmiştir. Haarcascade sınıflandırıcı eğitimi gerçekleştirilen Tablo-1’de verilen trafik işaretçilerinin parkur üzerinde farklı yerlere yerleştirilmesiyle oluşturulmuştur. Parkur boyunca aracın şerit takibinin sağlanabilmesi için yol kenarlarında beyaz şeritler kullanılmıştır. Parkurda yol, bariyer ve trafik işaretçileri kullanılmıştır. Kullanılan tüm trafik işaret ve işaretçileri, bariyer ve yol arasına yerleştirilmiştir. Çizilen parkur, Unreal Engine programına aktarılarak araç eklemiş ve gerçekleştirilen nihai simülasyon ortamı oluşturulmuştur. Oluşturulan simülasyon ortamı Şekil 9’da görülmektedir.



Şekil 8. VectorZero RoadRunner ortamında çizilen parkur

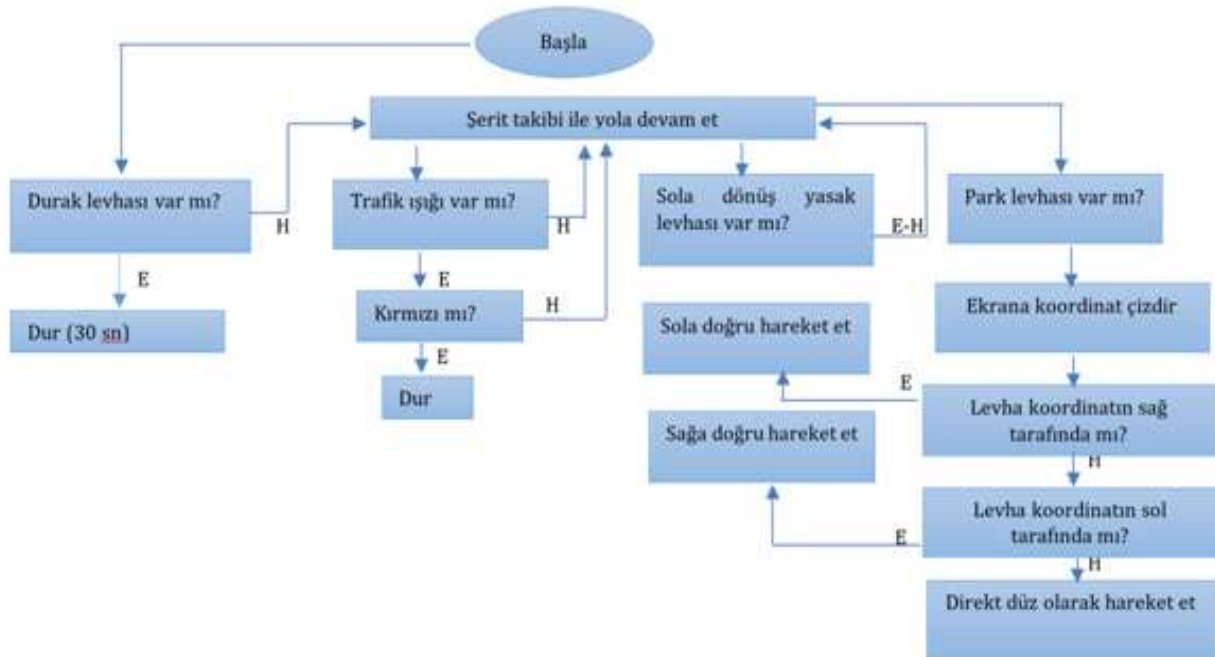




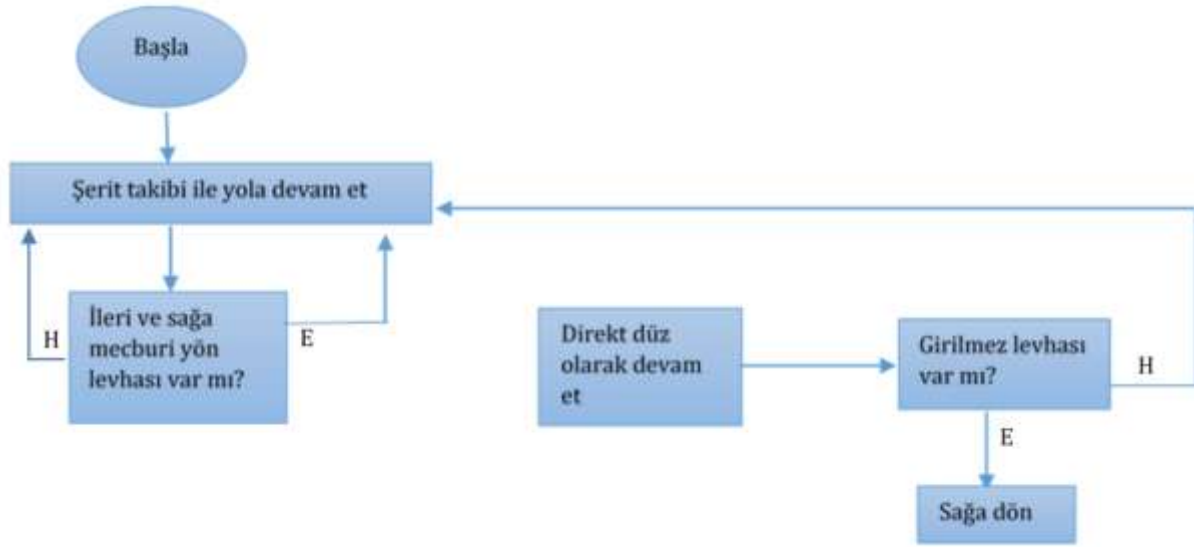
(a) Araç üzeri kamera görüntüsü (b) Parkur üstten görünüm  
Şekil 9. Unreal Engine programına aktarılan parkur ve araç görüntüsü

## 6. Simülasyon

Bu çalışma kapsamında yapılan Haarcascade sınıflandırıcı eğitimlerinin testi için simülasyon ortamı oluşturulmuştur. Bu test için kullanılacak yazılımlar Python dilinde yazılmıştır. Parkurda aracın başladığı ilk konumdan itibaren sırasıyla durak levhası, sola dönüş yasak, trafik lambası, durak levhası, ileri ve sağa mecburi yön, girilmez ve park levhaları ile karşılaşması gerekmektedir. Aracın ilk hareketi ile başlayacak olan simülasyon boyunca yapılan işlemlerin akış diyagramı verilmiştir. Şekil 10'da durak levhası, trafik ışığı, park, girilmez, ileri ve sağa mecburi yön ve sola girilmez levhasına ait akış diyagramı gösterilmektedir. Simülasyon ortamında bulunan aracın üzerindeki kamera ile yol üzerindeki şeritler görüntüden yazılım aracılığıyla ayrılarak aracın şeritler arasında ilerlemesi sağlanır. Simülasyon üzerinde gerçekleşen bu işlem Şekil 11' de görülmektedir. Gerçek zamanlı görüntü aktarımı ile sürekli gelen görüntü üzerinde Haarcascade sınıflandırıcı ile eğitilmiş trafik işaretçilerine ait görüntüler aranmaya başlar ve bulunduğu durumlarda trafik işareti üzerine bir çerçeve çizilir Şekil 12' de görülmektedir. Simülasyon çalışmasında bulunan her bir levhaya ait trafik kurallarına uygun bir şekilde hareket edilmesi geliştirilen algoritma ile sağlanır.

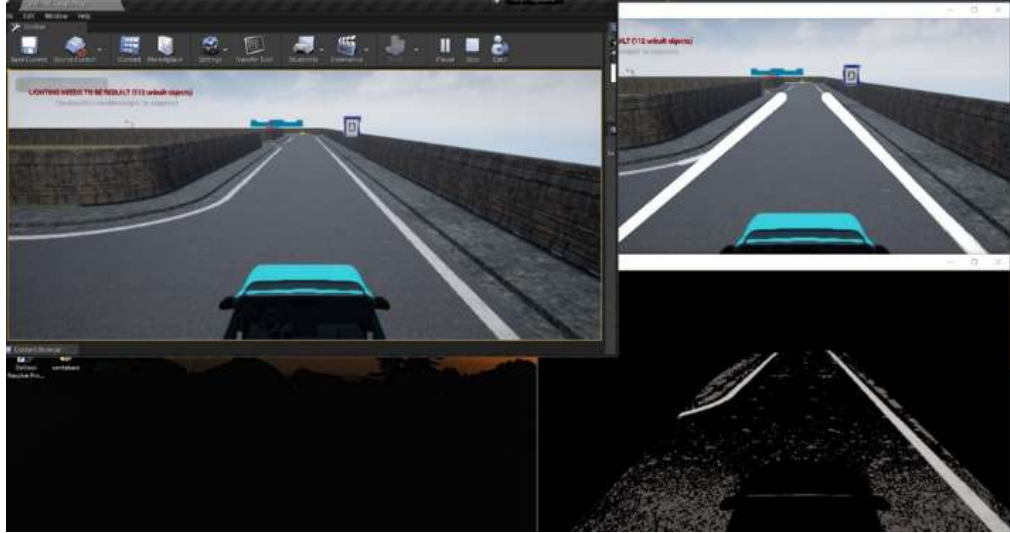


a) Durak, sola dönüş yasak, park levhası ve trafik ışığı algoritması



b) İleri ve sağa mecburi yön ve girilmez levhası algoritması

Şekil 10. Durak, park, sola dönüş yasak, ileri ve sağa mecburi yön, girilmez levhası ve trafik ışığı akış diyagramı



Şekil 11. Simülasyon ortamında şerit takibi görüntüsü



a) Park levhası

b) Durak levhası

Şekil 12. Haarcascade sınıflandırıcı ile bulunan trafik levha örnekleri

## 7. Sonuçlar

Elektrikli ve otonom bir aracın trafik akışını bozmayacak şekilde kurallara uygun hareket edebilmesini sağlayacak olan trafik işaret ve işaretçilerinin tespit edilmesi gerekmektedir. Bu işlem için en önemli kısım trafik işaret ve işaretçilerinin araç tarafından tespitinin sağlanması ve tespit edilen levhaya uygun şekilde hareket etmesini sağlayacak algoritmanın geliştirilmesidir. Bu makalede geliştirilen algoritmanın en temel yapısını ise levhaların tespitini sağlayacak olan sınıflandırma işlemi sağlamaktadır. Haarcascade sınıflandırıcısı eğitilerek geliştirilen algoritmanın Python üzerinden simülasyon ortamına entegre edilmesi sağlanmıştır. Bu sayede simülasyon ortamında bulunan aracın algoritmanın çalıştırılmasıyla otonomi hareketi başarılı bir şekilde gerçekleştirilmiştir. Gerçekleştirilen model eğitimlerinin her birine ait doğruluk yüzde değerleri gerçek ortam fotoğraflarında test edilerek hesaplanmış ve bu veriler Tablo 2’de verilmiştir. Şekil 13’te görüldüğü üzere simülasyon ortamında test edilen model eğitimleri gerçek ortamda da test edilmiş ve algoritmanın gerçek ortamdaki başarısı ortaya konulmuştur.

**Tablo 2.** Model eğitimlerine ait doğruluk yüzde tablosu

Trafik İşaret ve İşaretçileri İsmi	Doğruluk Yüzdesi(%)
1. Trafik Lambası	69,23
2. Dur Levhası	78,94
3. Park Levhası	78,57
4. Sola Dönüş Yasak Levhası	70,83
5. Durak Levhası	72,24
6. Girilmez Levhası	80,00
7. İleri ve Sağa Mecburi Yön Levhası	76,00
8. Park Yasak Levhası	76,92
9. Trafığe Kapalı Yol Levhası	69,23
10. İleri ve Sola Mecburi Yön Levhası	90,00



**Şekil 13.** Haarcascade sınıflandırıcısının uygulama örnekleri

## Yazarların Katkıları

FNO ve EÇ çalışmaları yapmış ve makalenin yazımını gerçekleştirmişlerdir. Yazarlar makalenin son halini okudu ve onayladı.

## Çıkar Çatışması

Yazarlar, çıkar çatışması olmadığını beyan eder.

## Kaynaklar

- [1]. Yakıştıran, Ö, Yaz M. "Bir Aracın Hareket Yönüne Konumlanmasının Bulanık Mantık İle Gerçekleştirilmesi" El-Cezerî Fen ve Mühendislik Dergisi 2020, 7(1):1-8.
- [2]. Aylak, B. L, Oral, O., Yazıcı, K. "Using Artificial Intelligence and Machine Learning Applications in Logistics," El-Cezerî Journal of Science and Engineering, 2021, 8(1): 74-93.
- [3]. Litman T. "Autonomous Vehicle Implementation Predictions Implications for Transport Planning" Traffic Technology International Workshop, 1-38, (2020).
- [4]. Nennioglu A.K, Köroglu T. "Otonom Araçlarda Hareket Planlanması", Adana Bilim ve Teknoloji Üniversitesi Fen Bilimleri Dergisi, 2018, 1(2): 20-29.
- [5]. Gomez V, "Object detection for autonomous driving using deep learning". PhD Thesis, Universitat At Politecnica De Catalunya, Automatica Robotica I Visio, Barcelona, (2015).
- [6]. Bingöl M.S, Kaymak Ç, Uçar A. "Derin Öğrenme Kullanılarak Otonom Araçların İnsan Sürüşünden Öğrenmesi". Fırat Üniversitesi Mühendislik Bilimleri Dergisi, 2019, 31(1): 177-185.
- [7]. Fleyeh H. "Color detection and segmentation for road and traffic signs". IEEE Conference on Cybernetics and Intelligent Systems, Singapore, 809-814, (2004).
- [8]. Houben S, Stallkamp J, Salmen J, Schlipsing M, Igel C. "Detection of traffic signs in real-world images: The German traffic sign detection benchmark". The 2013 International Joint Conference on Neural Networks, Dallas, USA, 1-8, (2013).
- [9]. Mehdi F, Sedigheh G. "Traffic Road Sign and Classification", Majlesi Journal Of Electrical Engineering, 2012, 4 (23): 54- 62.
- [10]. Liu C, Li S, Chang F, Wang Y. "Machine Vision Based Traffic Sign Detection Methods: Review, Analyses and Perspectives". IEEE Access, 2019, 7(1): 86578-86596.
- [11]. Karaimer H. C, Baştanlar Y. "Haar-Benzeri Öznitelikler ve Basamaklı Hızlandırma Kullanarak Tüm yönlü Kameralar ile Binek Araç Tespiti", IEEE Conference on Signal Processing and Communications Applications, Trabzon, Türkiye, 301-304, (2014).
- [12]. Ćorović A, Ilić V, Đurić S, Marijan M, Pavković B. "The Real-Time Detection of Traffic Participants Using YOLO Algorithm". 2018 26th Telecommunications Forum (TELFOR), Belgrade, 1-4, (2018).
- [13]. De Feo F, De Luca P. "Face Detection with a portable architecture in Real Time", Preprints, 2019, (1): 1-8.
- [14]. Kaplan A. "Gerçek ve Yarı Zamanlı Yüz Tespit Etme". Yüksek Lisans Tezi, Fırat Üniversitesi, Elâzığ, Türkiye, (2018).
- [15]. Viola P, Jones M. "Rapid object detection using a boosted cascade of simple features". Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Kauai, USA, I-I, (2001).
- [16]. Balasubramanian M, Palanivel S, Ramalingam V. "Video-based person recognition using fovea intensity comparison code". Behaviour & Information Technology, 2011, 30(6): 747-760.