



MARMARA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ



HAVA DURUMUNA GÖRE DEĞİŞEN **ARAÇ HIZ SINIRI UYGULAMASI**

EMİR MUSTAFA EFE

YÜKSEK LİSANS TEZİ

Fen Bilimleri Enstitüsü

Teknoloji Fakültesi

Elektrik-Elektronik Mühendisliği Anabilim Dalı (Türkçe)

Tezli Yüksek Lisans

DANIŞMAN

Dr. Öğr. Üyesi Veysel Gökhan BÖCEKÇİ

İSTANBUL, 2022

TEŞEKKÜR

Öğrenim hayatımda ve tez çalışmamda bilgi ve tecrübelerinden yararlandığım tüm öğretmenlerime, özellikle Sayın Dr. Öğr. Üyesi Veysel Gökhan BÖCEKÇİ'ye, maddi ve manevi tüm destekleri için biricik annem Münüre EFE'ye teşekkürü borç bilirim.

Eylül 2022

Emir Mustafa EFE

İÇİNDEKİLER

ÖZET.....	i
ABSTRACT	iii
SEMBOLLER	vi
KISALTMALAR	vii
ŞEKİLLER LİSTESİ.....	viii
TABLO LİSTESİ	xi
1 GİRİŞ.....	1
1.1 Aşırı Hızın Tanımı ve Yönetimi.....	2
1.2 Literatür Araştırmaları.....	9
1.2.1 Araç ve Hız Tespiti Uygulamaları.....	9
1.2.2 Derin Öğrenme Yöntemleri.....	10
2 GENEL BİLGİLER.....	13
2.1 Evrişimli Sinir Ağları (Convolution Neural Networks)	13
2.1.1 Evrişimli Sinir Ağları Mimarisi.....	14
2.2 Morfolojik Dönüşümler.....	21
2.2.1 Aşınma ve Genişleme.....	21
2.2.2 Açma ve Kapama	22
2.3 Kalman Filtresi	25
3 MATERYAL ve YÖNTEM.....	27
3.1 Hava Durumu Tespiti İçin Model Oluşturulması	30
3.1.1 Veri Toplanması	30
3.1.2 Veri Modeli Oluşturulması.....	34
3.1.3 Veri Modelinin Değerlendirilmesi	36
3.2 Araç Tespiti ve Takibi.....	41
3.2.1 İlgi Bölgesi (ROI) Belirleme	45
3.2.2 Yoğunluk Zaman Aralığı Belirleme.....	47
3.2.3 Arka Plan Çıkarımı.....	48
3.2.4 Morfolojik Dönüşümler.....	51
3.3 Hız Hesabı ve Hız Sınırı Kontrolü	52
4 BULGULAR	56
4.1 Gerçekleştirilen Sistem.....	56
4.2 Sonuçların Değerlendirilmesi.....	58
4.2.1 Farklı Hava Durumlarının Sonuçlara Etkisi	63
4.2.2 Farklı Derin Öğrenme Parametrelerinin Sonuçlara Etkisi.....	64

4.2.3	Farklı Görüntü İşleme Parametrelerinin Sonuçlara Etkisi	65
5	TARTIŞMA ve SONUÇ	68
KAYNAKLAR.....		70
ÖZGEÇMİŞ		76



ÖZET

HAVA DURUMUNA GÖRE DEĞİŞEN ARAÇ HIZ SINIRI UYGULAMASI

Son yıllarda, tüm dünyada trafiğe kayıtlı toplam taşıt sayısı hızla artmaktadır. Taşıt sayısındaki artışa gelişmekte olan araç hız kabiliyetleri de eklenince, trafik yoğunluğu ve kaza riskinin oldukça arttığı belirlenmiştir. Artan kaza riskini en aza indirebilmek için dünya genelinde birçok çalışma yapılmıştır. Dünya Sağlık Örgütü (DSÖ), Dünya Sağlık Gününü ilk defa 2004 yılında karayolları güvenliği konusuna ayırmıştır. Karayolları ulaştırma türlerini ve bağlantılarını araştırmak amacı ile ülkelerin bakanlarına doğrudan rapor veren Ulaştırma Bakanları Avrupa Konferansı (ECMT) 1 Ocak 2004 tarihinde kurulmuştur. 2006 yılında DSÖ ve ECMT'nin ortaklaşa hazırladığı rapor olan Hız Yönetimi raporu önemli çalışmaların başında gelir.

OECD ve ECMT üyesi olan ülkemizde de benzer çalışmalar yapılmaktadır. Geçmişteki çalışmalardan ve raporlardan yola çıkılarak oluşturulan Hız Yönetimi Karar Organları Ve Uygulayıcılar İçin Karayolu Güvenliği El Kitabı, Karayolları Trafik Güvenliği Strateji Belgesi (2021-2030) ve Karayolu Trafik Güvenlik Eylem Planı (2021-2023) çalışmaları ülkemiz için bu konunun ne kadar önemli olduğunu göstermektedir.

Birçok Avrupa ülkesinde yapılan hız ve kaza önleyici çalışma ve raporların en önemli ortak çıktıları denetleme sistemlerinde gelişen teknolojilerin kullanım oranının artırılması gerektiğidir. Karayolları denetiminin elektronik sistemler ile uygulanması, saha elemanı ve aracı kullanılmasını azaltacağından hem daha pratik hem de daha ekonomik olacaktır. Bu bağlamda birçok ülkede çeşitli elektronik denetleme sistemlerinin kullanılması yaygınlaşmaktadır. Bu elektronik denetleme sistemleri ile hız sınırı ihlali kontrolü, şerit ihlali kontrolü, emniyet kemeri kullanılması kontrolü vb. yapılabilmektedir.

Birçok Avrupa ve OECD üyesi ülkesinde hız sınırlarının belirlenmesinde hava koşulları parametre olarak kullanılmaktadır. Ancak bu parametreye uygun geliştirilen geliştirilmeye muhtaç uygulamalarda hız sınırı değişimleri manuel olarak gerçekleştirilmektedir. Ülkemizde ise elektronik denetim sistemlerinde uygulanan hız limitleri, kullanılan karayoluna ve kullanılan araç tipine bağlı olarak belirlenmektedir, değişen mevsimsel şartlara adapte bir hız sınırı uygulaması yapılmamaktadır. Aynı yolda aynı araç ile kış şartlarında veya yoğun yağış altında hava durumu göz ardı edilerek belirlenen hız sınırı ile gidilebilir olması kaza riskini oldukça

arttırmaktadır. Kaza riskinin azaltılması için mevsimsel şartlara adapte hız sınırlarının belirlenmesi ve bu sınırlara uygun elektronik denetlemenin sağlanması gerekmektedir.

Bu çalışmada, birçok alan uygulamalarında tercih edilen derin öğrenme ve görüntü işleme yöntemleri kullanılarak, elektronik denetleme sistemlerinde kullanılan hız sınırlarının mevsimsel şartlara adapte olarak uygulanabilirliği gösterilmiştir. Böylece herhangi bir elektronik denetleme sistemi kamerasından alınan anlık görüntüden, bölgesel ve mevsimsel parametreler ile yapılandırılmış derin öğrenme modelleri kullanılarak, mevsimsel şartlara adapte motorlu araç hız sınırı belirlenebilmektedir. Derin öğrenme algoritmasında kullanılmak üzere yaklaşık 6 ay boyunca Fransa, Türkiye vb. konumlardan çevrim içi olarak erişilebilen birçok kameradan dört mevsim şartlarını içeren, 1280x720 çözünürlükte ve renkli görseller ile veri seti oluşturulmuştur. Elde edilen yaklaşık 14GB büyüklüğündeki veri setinden anlamlı görseller seçilerek her kamera için ayrı modeller oluşturulmuştur. (Veri tabanı erişim linki: <https://mega.nz/folder/DE4gHYqS#-sU2k0eiHuYOnhUCet9WQ>)

Oluşturulan derin öğrenme modelleri sonucunda belirlenen adapte hız sınırlarına uygun elektronik denetleme, görüntü işleme yöntemleri kullanılarak elde edilen hız kestirimi ile gerçekleştirilmiştir. Hız kestirimi işlemi belirli mesafeden geçen araçların geçiş zamanları kullanılarak hesaplanmış ve Kalman filtresi ve test aracı ile geçişler yapılarak test ve kalibre edilmiştir. Yapılan denetleme sırasında her araç özgün bir kimlik değeri ile adlandırılmış ve sınır aşımı durumunda özgün kimlik numaralarını, ortalama hızlarını ve anlık ekran görüntülerini içerecek şekilde kaydedilmiştir. Araçların tespit edilen hızlarının doğrulanması işlemi için Kalman filtresi kullanılmıştır ayrıca manuel test aracı ile testler yapılarak gerçekleştirilmiştir. Test aracı ile yapılan testlerden, örnek olarak sistem hız çıktısı 60.38 km/h olan 60km/h sabit hızla yapılan geçiş için Kalman filtresi hız çıktısının 61.74 km/h olduğu görülmüştür. Test aracı ile geçişler sonucunda araç hızı tespitindeki hata oranının ortalama %2,54 olduğu belirlenmiştir. Ek olarak bu çalışma, ilgili karayolundan geçen araç sayısı bilgisini de içereceğinden periyodik trafik yoğunluğu verisini sağlayabilmektedir.

Sistem kameralarından alınan görsellerden oluşan veri tabanları kullanılarak oluşturulan modeller ile hava durumu tespiti işleminde başarımlar oranları %99.51-99.85 arasındadır. Araç tespiti, sayımı ve hızlarının belirlenmesinde ise başarımlar oranlarının %76.46-94.67 arasında olduğu belirlenmiştir.

ABSTRACT

VEHICLE SPEED LIMIT APPLICATION ACCORDING TO WEATHER CONDITIONS

In the recent years, the total number of vehicles registered to traffic has been increasing rapidly all over the world. While the developing vehicle technologies increase with the number of vehicles, it has been determined that the traffic density and the risk of accidents increases considerably. Many studies have been carried out around the world in order to minimize the increased risk of accidents. The World Health Organization (WHO) dedicated World Health Day to the subject of road safety for the first time in 2004. The European Conference of Transport Ministers (ECMT), which reports directly to the ministers of the countries, was established on 1 January 2004 with the aim of researching road transport modes and their connections. The Speed Management report, which was prepared jointly by WHO and ECMT in 2006, is one of the most important studies in this area.

Similar studies are carried out in Turkey, which is a member of OECD and ECMT. The Highway Safety Handbook for Speed Management Decision Bodies and Practitioners, Highway Traffic Safety Strategy Document (2021-2030) and Highway Traffic Safety Action Plan (2021-2023), which were created based on past studies and reports, are indicators of how important this issue is becoming.

The most important output of speed and accident prevention studies and reports made in many European countries is the need to increase the use of developing technologies in inspection systems. Implementation of highway inspection with electronic systems will be more practical and economical as it will reduce the use of field personnel and vehicles. In this context, the use of various electronic monitoring systems is becoming more widespread in many countries. Speed limit violation control, lane violation control, seat belt usage control etc. can be done with these electronic monitoring systems.

Weather conditions are used as a parameter in setting speed limits in many European and OECD member countries. However, speed limit changes are performed manually in applications that need improvement developed in accordance with this parameter. In Türkiye, the speed limits applied in electronic control systems are determined depending on the highway used and the type of vehicle used, a speed limit application is not applied to adapt to changing seasonal conditions.

The fact that it is possible to travel with the same vehicle on the same road in winter conditions or under heavy rain with the speed limit determined by ignoring the weather conditions increases the risk of accident considerably. In order to reduce the risk of accidents, it is necessary to determine speed limits adapted to seasonal conditions and to provide electronic control in accordance with these limits.

In this study, it has been determined that the speed limits used in electronic inspection systems can be applied by adapting to seasonal conditions by using deep learning and image processing methods used in many field applications. Thus, motor vehicle speed limit adapted to seasonal conditions can be determined from the snapshot taken from any electronic surveillance system camera, using deep learning models configured with regional and seasonal parameters. In order to be used in the deep learning algorithm, a data set was created with images that will include 4-season conditions from many cameras that can be accessed online from locations such as France, Turkey, etc. for about 6 months. Separate models were created for each camera by selecting meaningful images from the 14GB data set obtained. (Database access link: <https://mega.nz/folder/DE4gHYqS#-sU2k0eiHuYOnhUCet9WQ>)

Electronic monitoring in accordance with the adapted speed limits determined as a result of the deep learning models created was carried out with the speed estimation obtained by using image processing methods. The speed estimation process was calculated using the transit times of the vehicles passing a certain distance and calibrated by making passes with the Kalman filter and the test car. During the monitoring, each vehicle was named with a unique identification value and recorded to include its unique identification numbers, average speeds and snapshots in case of exceeding the speed limit. The verification of the detected speeds of the vehicles was carried out by using the Kalman filter and by making passes with the manual test vehicle. From the tests carried out with the test car, for example, the Kalman filter speed output was 61.74 km/h for the transition made at a constant speed of 60km/h with a system speed output of 60.38 km/h. As a result of the tests with the test vehicle, it was determined that the error rate in determining the vehicle speed was 2.54% on average. In addition, this study can provide periodic traffic density data, as it will include information on the number of vehicles passing through the relevant highway.

With the models created using databases consisting of images taken from system cameras, the success rates in weather detection are between 99.51% and 99.85%. It was determined that the

success rates in vehicle detection, counting and speed determination were between 76.46-94.67%.



SEMBOLLER

A	: Morfolojik İşlemlerde Ham Görüntü
B	: Morfolojik İşlemlerde Yapılandırma Elemanı
B_x	: Kalman Filtresi Pozisyon Ölçüm Hatalarının Varyantı
d_{öklid}	: Öklid Uzaklığı
H	: Kalman Filtresi Ölçüm Matrisi
k	: Hız Kalibrasyon Sabiti
K_k	: Kalman Kazancı
l	: Mesafe Katsayısı
P_k	: Kalman Filtresi Hataların Koverans Matrisi
R	: Kalman Filtresi Kovaryans Matrisi
t_{ROI}	: İlgi Alanı İçerisinde Geçen Süre
t_{EnterTime}	: İlgi Alanına Giriş Anı
t_{ExitTime}	: İlgi Alanından Çıkış Anı
v_k	: Kalman Filtresi Ölçüm Gürültüsü
v_{yaklaşık}	: Araç Hız Tahmini Değeri
Z_k	: Kalman Filtresi Ölçüm Vektörü

KISALTMALAR

ECMT	: Ulaştırma Bakanları Avrupa Konferansı
CPU	: Merkezi İşlem Birimi
CNN	: Evrişimli Sinir Ağları
FPS	: Saniyedeki Kare Sayısı
GPU	: Grafik İşlem Birimi
DSÖ	: Dünya Sağlık Örgütü
İBBUYM	: İstanbul Büyükşehir Belediyesi Ulaşım Yönetim Merkezi
KGM	: Karayolları Genel Müdürlüğü
OECD	: Ekonomik İşbirliği ve Kalkınma Örgütü
RAM	: Rastgele Erişimli Hafıza
RELU	: Düzeltilmiş Doğrusal Birim Katmanı
ROI	: İlgi Bölgesi
SARTRE	: Avrupa'da Karayolu Trafik Riskine Karşı Sosyal Tutumlar
TÜİK	: Türkiye İstatistik Kurumu
YZ	: Yapay Zekâ

ŞEKİLLER LİSTESİ

Şekil-1 Fransa’da Satılan 150 km/s Üzerinde Hız Yapabilen Taşıt Yüzdesi	2
Şekil-2 Ülkemizdeki Motorlu Kara Taşıtları Sayısı, 2002-2021	3
Şekil-3 Gaz Emisyonu İngiltere (2005)	3
Şekil-4 Motor ve Lastik Gürültüsü	3
Şekil-5 Hız-Kaza Riski Grafiği	4
Şekil-6 Fransa A7 Otoyolu Yoğun Trafikte Değişken Hız Sınırı.....	8
Şekil-7 CNN Katman Mimarisi	14
Şekil-8 Örnek CNN Filtresi	15
Şekil-9 Örnek Resim Girdi	15
Şekil-10 Filtre Uygulaması.....	15
Şekil-11 Maksimum ve Ortalama Ortaklama	18
Şekil-12 Tam Bağlantı Katmanı	18
Şekil-13 Düzleştirme İşlemi	19
Şekil-14 Sınıflandırma İşlemi.....	20
Şekil-15 Yapılandırma Elemanı Örneği	21
Şekil-16 3x3 Aşınma İşlemi	22
Şekil-17 3x3 Genişleme İşlemi.....	22
Şekil-18 Açma İşlemi Örnekleri.....	23
Şekil-19 3×3 Yapı Elemanı ile Kapama İşlemi	23
Şekil-20 Akış Diyagramı	28
Şekil-21 Araç Yoğunluğu Grafiği Örneği	29
Şekil-22 Youtube ve İBBUYM Üzerinden Erişilen Canlı Kamera Görüntüleri	31
Şekil-23 SelfDb Uygulaması	32
Şekil-24 Kamera Görüntüleri Dizini.....	33
Şekil-25 İstanbul Çeşitli Hava Durumları	34

Şekil-26 Camera #4.model Tahmin Değerleri Isı Haritası	37
Şekil-27 Camera #4.model ROC Eğrisi.....	37
Şekil-28 Camera #12.model Tahmin Değerleri Isı Haritası	38
Şekil-29 Camera #12.model ROC Eğrisi.....	39
Şekil-30 Camera #13.model Tahmin Değerleri Isı Haritası	40
Şekil-31 Camera #13.model ROC Eğrisi.....	40
Şekil-32 Uygulama Arayüzü ve Kamera Listesi	41
Şekil-33 Farklı Kamera Video Görüntüleri	42
Şekil-34 Kamera Seçimi ve Process İşlemi Başlatılması	42
Şekil-35 Camera #4 için Açılan Process Ekranı.....	43
Şekil-36 Kaydedilen Veri Tabanları	43
Şekil-37 Normal Havada İşlem Çıktıları	44
Şekil-38 Normal Havada Hız Sınırı Aşımı Yapan Araç	44
Şekil-39 Yağmurlu Havada İşlem Çıktıları	44
Şekil-40 Yağmurlu Havada Hız Sınırı Aşımı Yapan Araçlar.....	45
Şekil-41 Karlı ve Sisli Havada Hız Sınırı Aşım Anları	45
Şekil-42 ROI Seçim Butonu	46
Şekil-43 ROI Belirleme Ekranı	46
Şekil-44 Kaydedilmiş ROI config Dosyaları.....	47
Şekil-45 Araç Yoğunluğu Zaman Aralığı Ayarı.....	47
Şekil-46 Araç Yoğunluğu Grafiği Örnekleri	48
Şekil-47 Arka Plan Çıkarım Yöntemi.....	48
Şekil-48 Tespit Edilmiş ve Numaralandırılmış Araç.....	48
Şekil-49 Arka Plan Çıkarımı ve Morfolojik Kapatma İşlemi	49
Şekil-50 Morfolojik Açma İşlemi ve Tespit Edilen Hareketli Bölge	49
Şekil-51 Tespit Edilen Hareketli Obje Sınırları.....	49
Şekil-52 ROI Sınırları İçerisinde mask Alanı.....	50

Şekil-53 ROI alanı ve Ana Görüntüde ROI Sınırları.....	50
Şekil-54 Arka Plan Çıkarımı	50
Şekil-55 Aynı Ekranda Birden Fazla Araç Tespiti	51
Şekil-56 Büyük Araçların Tespiti	51
Şekil-57 Morfolojik Dönüşümler Öncesi ve Sonrası.....	52
Şekil-58 Fransa’da Kullanılan Trafik Çizgileri ve Özellikleri	52
Şekil-59 Türkiye’de Yerleşim İçi Trafik Çizgileri ve Özellikleri	53
Şekil-60 Araç ROI Giriş ve Çıkış Anları.....	53
Şekil-61 Aynı Anda Birden Fazla Araç İçin ROI Giriş ve Çıkış Anları	54
Şekil-62 Normal ve Sisli Havada Hız Aşım Anları Örneği	55
Şekil-63 Araç Giriş-Çıkış ve Hız Veri Tabanı Örneği	55
Şekil-64 Düşük Görüntü Kalitesi Olan Kameralar İle Araç Tespiti	57
Şekil-65 Kalman Filtresi İle Araç Merkez Noktası Tahminleri.....	60
Şekil-66 Son Karedeki Kalman Filtresi Tahmini	60
Şekil-67 Camera #4 Kalman Tahmininin x-y Düzlemindeki Koordinatları.....	61
Şekil-68 Uygulama Hız Tahmini Sonucu.....	61
Şekil-69 Camera #4 Kalman Hız Hesabı Sonucu.....	61
Şekil-70 Test Aracı İle Uygulama Hız Tahmini Sonucu.....	62
Şekil-71 Camera #13 Kalman Tahmininin x-y Düzlemindeki Koordinatları.....	62
Şekil-72 Camera #13 Kalman Tahmininin x-y Düzlemindeki Koordinatları (Yaklaştırılmış)	62
Şekil-73 Test Aracı İle Kalman Hız Hesabı Sonucu	63
Şekil-74 Yoğun Yağış Altında Hatalı Araç ve Hız Tespiti.....	63
Şekil-75 Hatalı Görüntü İşleme Parametreleri Sonucu-1	66
Şekil-76 Hatalı Görüntü İşleme Parametreleri Sonucu-2	66
Şekil-77 Hatalı Görüntü İşleme Parametreleri Sonucu-3	67
Şekil-78 Hatalı Görüntü İşleme Parametreleri Sonucu-4	67

TABLO LİSTESİ

Tablo-1 Türkiye'de Araçların Uyması Gereken Yasal Hız Sınırları	5
Tablo-2 Filtre Örnekleri	15
Tablo-3 ReLU Aktivasyon Fonksiyonu	17
Tablo-4 Veri Setleri ve Demografik Yapıları.....	36
Tablo-5 Camera #4.model Sonuçları.....	37
Tablo-6 Camera #4.model F1 Skorları.....	38
Tablo-7 Camera #12.model Sonuçları.....	39
Tablo-8 Camera #12.model F1 Skorları.....	39
Tablo-9 Camera #13.model Sonuçları.....	40
Tablo-10 Camera #13.model F1 Skorları.....	41
Tablo-11 Kamera Listesi	57
Tablo-12 Çeşitli Hava Şartlarında Gerçekleştirilen Sayımlar	58
Tablo-13 Tasarlanan Uygulamanın Benzer Çalışmalar İle Karşılaştırılması.....	59
Tablo-14 Farklı CNN Modelleri ve Tahmin Sonuçları	64

1 GİRİŞ

Elektronik sistemlerdeki teknolojik gelişmelerle birlikte karayolları elektronik denetleme sistemlerinde kullanılmak üzere plaka tanıma, hız sınırı kontrolü, hatalı şerit değişimi kontrolü, araç kullanırken telefon kullanımı kontrolü vb. denetlemelerin yapılabilmesi mümkün hale gelmiştir. Trafik kontrol merkezlerinde elektronik denetleme sistemi ağlarına bağlı kameralar aracılığıyla trafik gerçek zamanlı olarak izlenebilmekte ve denetlenebilmektedir. Kontrol merkezlerinin mevcudiyetinin ve işlevselliğinin artmasıyla sürücülerin bilinçli ve denetimli bir şekilde kurallara uyarak daha güvenli seyahati sağlanmaya çalışılmaktadır.

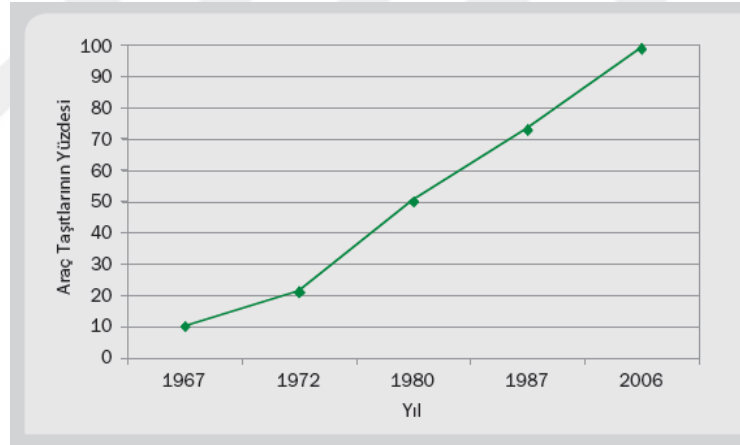
Karayollarında motorlu araçlar için elektronik denetleme sistemlerinde hız sınırı uygulaması yaygın olarak yapılmaktadır. Hız sınırı uygulama için bazı bölgelerde kamera yeterli olurken bazı bölgelerde ise eski teknoloji, altyapı yetersizliği, hatalı planlama vb. sebeple ek olarak hız algılayıcı cihazlar kullanılmaktadır. Bu cihazların varlığı hem maliyeti arttırmakta hem de bakım gerektirdiğinden iş gücü verimini düşürmektedir. Diğer bir husus birçok Avrupa ülkesinde olduğu gibi ülkemizde de karayollarında hız sınırları belirlenirken araç tipi ve kullanılan karayolu temel alınmasıdır. Elektronik denetleme sistemlerinde hız sınırlarının belirlenmesinde, araç tipi ve kullanılan karayolu kadar önemli bir parametre olan hava durumu göz önüne alınmamaktadır. Bu durumda denetleme sırasında hava durumuna adapte hız sınırı kullanılmadığından, mevsimsel şartlara uygun hızda araç kullanımı sürücü inisiyatifine ve saha denetleyici personel varlığına bağlı kalmaktadır.

Birçok Avrupa ülkesinde de görülen bu durumların önüne geçilebilmesi için bu çalışma ile birlikte trafik kontrol merkezlerinde kullanılabilecek, kamera görüntülerinden görüntü işleme yöntemleri kullanan ve mevsimsel şartlara adapte hız sınırı uygulaması yapabilen bir yazılım ortaya konulması hedeflenmektedir.

1.1 Aşırı Hızın Tanımı ve Yönetimi

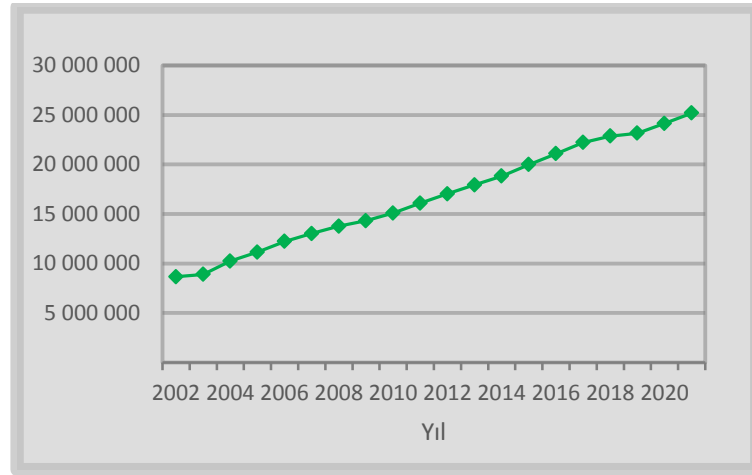
Hız birçok alanda olduğu gibi trafik akışı sırasında da genel manada olumlu bir kavramdır. Yolculuk süresinin azalması ve hareketliliğin artması en önemli olumlu etkilerindendir. Son yıllarda karayollarında ve araçlarda meydana gelen gelişmeler bu artışlarda önemli rol oynamıştır. İş, alışveriş ve eğlence yerlerine ulaşımın hızlanması sayesinde ekonomik kalkınmaya fayda sağladığı gibi yaşam kalitesini arttırmıştır. Ayrıca hastane ve ambulans ulaşımı gibi acil durumlarda hızın artması sonucunda ölüm oranları azalmıştır.

Gelişen teknoloji ve ihtiyaçlar kapsamında, ekonomik gelişmelerle de doğru orantılı olacak şekilde her geçen yıl trafiğe çıkan yeni araç sayısı artmaktadır. Yıllar önce çeşitli parametrelere göre konulan hız sınırları, güncel teknolojik araç hızlarına ve araç yoğunluğuna uygunluğunu zamanla kaybetmektedir. Şekil-1’de [1] görülebileceği şekilde araç teknolojilerindeki gelişimler sonucunda Fransa’daki çoğu araç güncel hız limitlerine rahatlıkla ulaşabilmektedir.



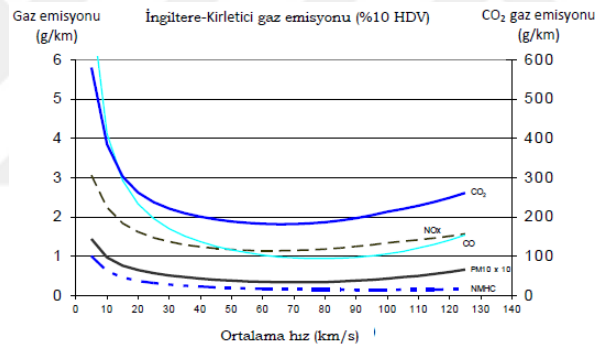
Şekil-1 Fransa’da Satılan 150 km/s Üzerinde Hız Yapabilen Taşıt Yüzdesi

Ülkemizde de bu durum diğer Avrupa ülkelerinden farklı değildir. TÜİK verilerine göre 2002-2021 yılları arasındaki toplam araç sayısı her geçen yıl artarak devam etmiştir. Bu artış beraberinde trafik yoğunluklarını, çevresel kirlilikleri, karayolları güvenlik problemlerini ve kaza risklerindeki artışı getirmektedir. Şekil-2’de [2] Türkiye’de motorlu kara taşıtlarının zamanla değişimi grafiksel olarak verilmiştir.

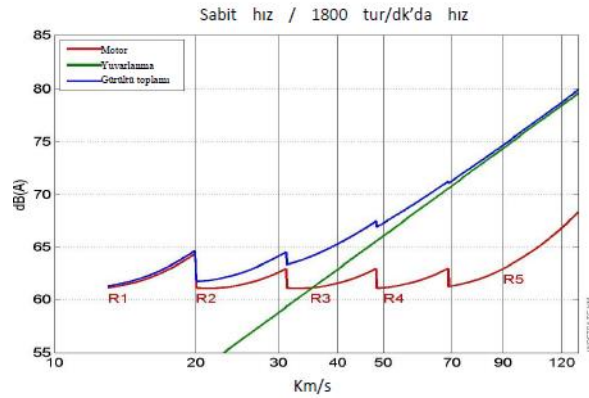


Şekil-2 Ülkemizdeki Motorlu Kara Taşıtları Sayısı, 2002-2021

Araç sayısındaki ve hızdaki artış beraberinde çevre ve gürültü kirliliğindeki artışı, karayolları güvenlik problemleri gibi olumsuz etkileri doğurmaktadır. Şekil-3’de [3] gaz emisyonu ile ortalama hız arasındaki değişim ve Şekil-4’de [3] hız değişimi ile motor ve lastik gürültüsündeki değişim grafiksel olarak verilmiştir.

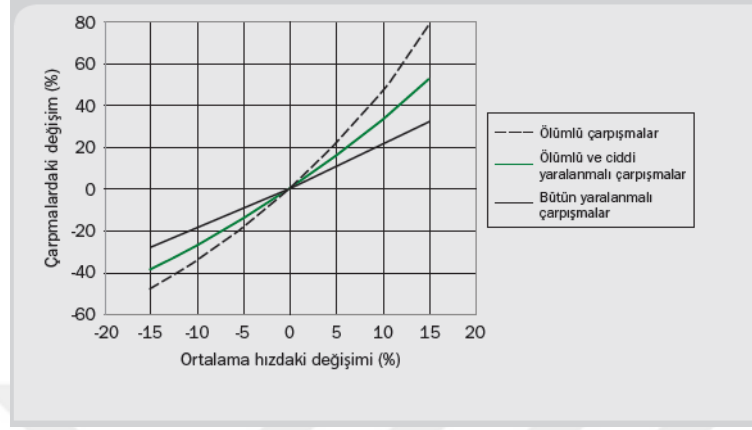


Şekil-3 Gaz Emisyonu İngiltere (2005)



Şekil-4 Motor ve Lastik Gürültüsü

Hızın olumsuz etkilerinin büyük bir çoğunluğu aşırı hızdan kaynaklanmaktadır. Aşırı hız; hız sınırının aşıldığı hızdır. Aşırı hız, ölümlü kazaların büyük çoğunluğunun sebebi olmakla beraber kazaların şiddetini belirleyen önemli bir faktördür. Şekil-5’de [1] görüldüğü gibi ortalama hızdaki değişimler arttıkça kaza riski de artmaktadır.



Şekil-5 Hız-Kaza Riski Grafiği

Aşırı hız sonuçları göz önüne alındığında kontrol edilmesi, denetlenmesi ve sınırlandırılması gereken bir durumdur. Hız yönetimi, yönetim birimlerince şartlara uygun şekilde tasarlanan hız sınırı uygulamalarıdır ve çevrenin hareketlilik ihtiyaçlarına aykırı olmamalıdır.

Geçmişte hız sınırı belirlemede en çok kullanılan yöntem V85 yöntemi olmuştur. Bu yöntemde tahmini hız sınırı, sürücülerin %85’nin serbest akış durumunda uyduğu düşük hız olarak belirlenmektedir. Bu yöntemde sürücülerin ilgili yolda uygun hızla seyahat ettikleri varsayılmaktadır. Bu yöntem günümüz şartlarına ve teknolojisine uygun değildir. Kullanılan hız sınırlarının uygulanabilirliğinin artırılması ve çevre ile uyumunun denetlenmesi için sürekli gözden geçirilmesi gerekmektedir.

Yetkili birimler aşırı hız ile mücadele politikalarını belirlerken ortaklaşa çalışmalı ve hızlı ve kalıcı çözümler üretmelidir. Örneğin Fransa 14 Haziran 2002 tarihinde ülke yollarının güvenliğini 3 ana hedeflerinden biri olarak ilan etmiş ve 5 yıllık planlarına dâhil etmiştir. 2004 yılı sonunda toplam hız kamerası adedi 400 olurken bu sayı 2007 sonunda 2000’e kadar çıkmıştır. Birçok alandan bakanların katılımıyla politika çalışmaları yapılmış ve sonuç olarak yıllık ortalama hız 5 km/h kadar düşmüş ve ölümlü kaza oranı %30 [1] kadar azalma görülmüştür.

Ülkemizde de aşırı hız önleme çalışmaları yapılmaktadır. Yol ve araç çeşitlerine göre belirlenen hız sınırları görevli personel, elektronik denetleme sistemleri vb. ile aşırı hız

belirlenen hız sınırlarına göre önlenmeye çalışılmaktadır. Tablo-1’de [4] Türkiye’deki yasal hız sınırları belirtilmiştir.

Tablo-1 Türkiye’de Araçların Uyması Gereken Yasal Hız Sınırları

TÜRKİYE’DE ARAÇLARIN UYMASI GEREKEN YASAL HIZ SINIRLARI					
ARAÇ CİNSİ	YERLEŞİM YERİ İÇİNDE (km/s)	YERLEŞİM YERİ DIŞINDA		OTOYOLLARDA	
		ŞEHİRLERARASI ÇİFT YÖNLÜ KARAYOLLARINDA (km/s)	BÖLÜNMÜŞ YOLLARDA (km/s)	KGM tarafından işletilen otoyollarda (km/s)	Yap-İşlet-Devret (YİD) modeliyle işletilen otoyollarda (km/s)
Otomobil (M1), (M1G)	50	90	110	130*	140*
Minibüs (M2)	50	80	90	100	
Otobüs (M2-M3)	50	80	90	100	
Kamyonet (N1), N1G)	50	80	85	95	
(Ek satır: RG-21/3/2012-28240)	50	85	100	110	
Panelvan (N1)					
Kamyon (N2-N3)	50	80	85	90	
Çekici (N2-N3)					
Motosiklet (L3)	50	80	90	100	
Motosiklet (L4, L5, L7)	50	70	80	80	
Motorlu bisiklet (L1, L2, L6)	30	45	45	Giremez	
Motorsuz bisiklet					
(Değişik satır: RG-17/4/2015-29329) Tehlikeli madde taşıyan araçlar (Belgelerinde aksine bir hüküm yoksa)	30	50	60	70	
(Ek satır: RG-17/4/2015-29329) Özel yük taşıma izin belgesi veya özel izin belgesi ile karayoluna çıkan araçlarda (Belgelerinde aksine bir hüküm yoksa)	30	50	50	60	
Lastik tekerlekli traktörler	20	30	40	Giremez	
Arızalı bir aracı çeken araçlar	20	20	30	40	
İş makineleri	20	20	20	Yolun yapım, bakım veya işletilmesinden sorumlu kuruluştan izin alınmadan giremez	
* Not: İçişleri Bakanlığının 02.06.2022 tarihli 23635644.249-31129 sayılı yazısına istinaden otomobiller için otoyollarda hız sınırı artırılmıştır.					

Ayrıca ülkemizde trafikteki can kayıplarının azaltılması, ciddi yaralanmaların önüne geçilmesi ve risklerin %50 oranında azaltılması hedefi için, 3 Şubat 2021 tarihinde 31384 sayılı resmi gazetede yayınlanan genelge ile Karayolu Trafik Güvenliği Strateji Belgesi (2021-2030) ve Karayolu Trafik Güvenliği Eylem Planı (2021-2023) çalışmalarını ilan edilmiştir [5], [6]. Bu planlarda ölümlü ve yaralanmalı kazaların önlemesi adına aşırı hızın düşürülmesi için en etkili yöntemin elektronik denetimler olduğu tespit edilmiş ve bu elektronik denetleme sistemlerinin arttırılması karar alınmıştır

Ülkeler ve yerel yönetimler ne kadar her parametreye uygun politikalar belirlemeye çalışsa ve denetimlerini sıkılaştırsa da, hız sınırlarına uyum nihayetinde sürücü psikoloji ve alışkanlıklarına bağlıdır. Sürücü yol süresini kısaltmaya, aşırı hız yapıp heyecan aramaya eğilimli olabilir. Sürücü hız sınırına uyması gerektiğini bilmesine rağmen hız sınırlarına uyum oranı, kuralı çiğnemesinin avantajlarına ve dezavantajlarına göre değişmektedir. 2004 yılında SARTRE [3] projesi kapsamında Avrupa ülkelerinde yapılan araştırmaya göre, sürücülerin %82'si aşırı hızın önemli bir kaza faktörü olduğunu bildiği ancak %70'inden fazlasının hız sınırlarını aştıklarını belirttikleri görülmektedir.

Hız sınırları belirlenirken yol hizalaması, yol kalitesi, çevresel faktörler vs. dikkate alınmaktadır. İlgili herhangi yol için hız sınırı belirlenirken uygulanması gereken somut bir yöntem olmamakla beraber hava koşullarına uygun hız sınırı uygulamaları çok sınırlıdır. Örnek olarak 2000 yılında Norveç'te şehir içi yollarda hız sınırları belirlenirken aşağıdaki unsurlar göz önüne alınmıştır [3].

- Tüm kullanıcılar için zaman maliyeti
- Kazaların maliyeti
- Tehlike hissine bağlı maliyet
- Trafikten kaynaklanan gürültü maliyeti
- Yerel ve küresel kirlilik maliyeti

Bu unsurlara göre belirlenen hız sınırları aşağıdaki gibidir.

- Bölgesel ana yollar: 60 km/s
- Yerel ana yollar: 50 km/s
- Dağıtıcı yollar: 50 km/s
- Giriş yolları: 30 km/s

- Şehir merkezindeki yollar: 30 km/s

OECD üyesi birçok ülkede aşağıdaki kıstaslara göre hız sınırı uygulamaları yapılmaktadır [3].

- Yol türü
- Araç ve yük türü
- Lastik tipi
- Sürücü sınıfı
- Hava koşulları

SARTRE ve benzeri çalışmaların gösterdiği gibi alışagelmış yöntemlerle belirlenmiş hız sınırları zaten sürücüler tarafından yüksek oranlarla aşılmaktadır. Buna ek olarak, dinamik ve güncel teknoloji ile desteklenmeyen bu hız sınırı uygulamaları hedeflediği kaza riskini azaltma başarısına uzak kalacaktır. Örneğin 120 km/h hız sınırı bulunan bir otoyolda yapılan elektronik denetlemede güneşli hava şartlarında ve yoğun yağış altında yasal olarak aynı hız sınırı geçerli olacaktır. Bu durumda kaza riskinin azaltılması %70 oranında aşırı hız yaptığı belirlenen sürücülerin inisiyatifine ve denetleyici personel varlığına bağlı kalacaktır.

Bu durumun önüne geçilmesi için birçok model geliştirilmiştir. Bu modeller ile farklı unsurlara göre karayolları güvenliği için uygun hız sınırları belirlenmek hedeflenmiştir. En çok kullanılan model LIMITS [3] programlarıdır. Bu programlar hız sınırı belirlenmesinde karar ağaçları kullanmaktadır. Yeni Zelanda’da NZLIMITS, Amerika’da USLIMITS, Avustralya’da QLIMITS, NLIMITS, VLIMITS vb. farklı versiyonları mevcuttur. Bu modellerde hız sınırı belirlenmesinde aşağıdaki parametreler göz önüne alınmıştır.

- Yol ve özellikleri
- Çevresel faktörler, yapılar
- Kullanım sayıları ve faaliyetleri
- Kaza geçmişi
- Kesişim bölgelerindeki hız sınırları

Avrupa Birliği’nde sadece Fransa hava koşullarına göre değişken hız sınırını uygulamaktadır. Yağmur ya da kar yağması durumunda veya yoğun trafik şartlarında otoyollar üzerindeki hız sınırı 130 km/s’den 110 km/s’ye, şehir içindeki yollarda 90 km/s’den 80 km/s’ye manuel olarak düşürülmektedir. Sis olması durumunda (görüş mesafesi 50 m’den az olduğu durumlarda) hız sınırı saatte 50 km’ye düşürülmektedir. Şekil-6’da [3] gösterildiği gibi bu yöntem manuel olarak gerçekleşmekte ve bölgesel kapsamdadır.



Şekil-6 Fransa A7 Otoyolu Yoğun Trafikte Değişken Hız Sınırı

Benzer yöntemlerin daha faydalı olabilmesi için; yolun, trafiğin ve hava koşullarının gerçek durumunu göz önünde bulundurması ve hız sınırlarının dinamik olarak değişebilir olması gerekmektedir. Nitekim bu yöntemleri kullanmak özellikle maliyet sebebiyle şimdiye kadar mümkün olmamıştır; fakat gelişen teknoloji ile personel ve donanım maliyetleri de düşünüldüğünde teşvik edilmeleri gerekmektedir.

Bu tez çalışmasında ülkemiz 2021-2030 Karayolu Trafik Güvenliği Stratejisi'nde [5] ve 2021-2023 Karayolu Trafik Eylem Planı'nda [6] açıklanan tespit, strateji ve ihtiyaçlara uygun, yapay zekâ teknolojilerindeki gelişme ile değişen hava koşullarını hız sınırlarının geçici belirlenmesinde parametre olarak kullanabilecek, dinamik ve yerel elektronik hız denetimi sistemi ortaya konulmuştur. Bu çalışmada az sayıda Avrupa ülkesinde manuel olarak yapılan denetimin otomatik olarak yapılabilmesi hedeflenmiş, ihlallerin tespiti ile birlikte denetlemenin amacına uygun olmasına çalışılmıştır. Eylem planlarındaki stratejilerden biri olan elektronik denetleme sistemleri ile daha az personel kullanarak 7/24 denetim sağlanmasına destek olunması hedeflenmiştir. Araç sayısı ve yoğunluğu uygulamaları ile farklı çalışmalarda kullanılmak üzere değerli veriler oluşturulmuştur. Hazırlanan bilgisayar programı sayesinde çok yüksek teknolojik özellikli ve maliyetli kameralara ihtiyaç duymadan, uzaktan erişilebilecek herhangi kamera görüntüsü sayesinde elektronik denetlemelerin yapılabilceğinin ortaya konulması hedeflenmiştir.

1.2 Literatür Araştırmaları

Bu bölümde, çalışmadan önce yapılmış benzer çalışmaların incelemeleri iki ana başlık altında gerçekleştirilmiştir.

1.2.1 Araç ve Hız Tespiti Uygulamaları

Partha Narayan C. ve arkadaşları tarafından yapılan çalışmada [7] görüntü işleme yöntemleri gece ve gündüz zaman aralıkları için farklı parametreler içerecek şekilde kullanılarak araç tespiti çalışması yapılmıştır. Bu çalışmada araç tespiti gündüz zaman aralığı için %95.08 gece zaman aralığı için %94.10 başarımları elde etmiştir.

Debojit B. ve arkadaşları tarafından yapılan çalışmada [8] araç tespiti ve sayımı için CNN mimarisi kullanılmıştır. Birden fazla veri tabanı için belirli saat aralıklarında yapılan sayımlarda (gündüz saat aralıkları) ortalama %96.55 başarımları elde edilmiştir. Bu çalışmada da belirtildiği şekilde, araç sayım işlemi ağır yağmurlu ve karlı hava durumları sırasında gerçekleştirilememiştir ve ayrıca araç sayımı şerit içerisinde geçen araçlara göre yapıldığından şerit değişimi yapan araçlar birden fazla defa sayılmış olmaktadır.

Jun-Wei H. ve arkadaşları tarafından yapılan çalışmada [9] araç sayımı sırasında görüntü işleme yöntemleri kullanarak gölge filtrelemesi ve lineerlik özelliği ile sayım verimi artırılmaya çalışılmıştır. Çalışma sonucunda araç sayım başarımları farklı kameralar için %92.8–94.8 aralığında olduğu görülmüştür.

Surendra G. ve arkadaşları tarafından yapılan çalışmada [10] görüntü işleme ve dinamik arka plan çıkarımı yöntemleri kullanarak araç tespiti, takibi ve sınıflandırılması yapılmıştır. Bu çalışma sonucunda 20 dakikalık, 15 fps (saniye başına görüntü) video görüntüsünde araçlar %90 oranında başarılı tespit ve takip edilmiş ve %70 başarımları oranında başarılı sınıflandırılmıştır.

Van Huy P. ve arkadaşları tarafından yapılan çalışmada [11] görüntü işleme yöntemleri kullanılarak araç sayımı gerçekleştirilmiştir. Çalışma sonucunda 20 dakikalık bir görüntüde bulunan araçlar %98 oranında başarılı olarak tespit edilip sayılmıştır.

Ayush S. ve arkadaşları tarafından yapılan çalışmada [12] akıllı trafik ışıkları kontrolü yapılabilmesi adına araç yoğunluğu/sayımı yapılmıştır. Araç yoğunluğu için sayım işlemleri görüntü işleme yöntemleri kullanılarak gerçekleştirilmiştir. Bu sistem sayesinde yoğunluk durumunda 180 saniye olan bekleme süresi 30 saniyeye çekilmiş ve toplam trafik akışında

%35 artış gözlemlenmiştir. Günbatımı ve düşük ışık şartlarında bu sistemin beklendiği şekilde çalışmadığı belirtilmiştir.

Danang Wahyu W. ve arkadaşları tarafından çalışmada [13] görüntü işleme yöntemleri ve morfolojik dönüşümler kullanılarak araç tespiti yapılmış ve yaklaşık araç hızı hesaplaması gerçekleştirilmiştir. Hız kestirimi başarımlı oranı %87.01 oranında gerçekleşmiştir ancak bu çalışmada yaratılan sistemin yüksek hızda hata oranının arttığı belirlenmiştir.

Toufiq P. ve arkadaşları tarafından yapılan çalışmada [14]görüntü işlemlerinde çokça kullanılan arka plan çıkarımı ve görüntüden özellik çıkarımı için yeni bir metot sunmaktadır.

Prem Kumar B. ve arkadaşları tarafından yapılan çalışmada[15] görüntü işleme yöntemlerinden Gaussian Mixture Model kullanılarak araç sayımı gerçekleştirilmiştir. Çalışma sonucunda üç farklı videoda yapılan sayım sonucunda %91.26 oranında başarılı sayım gerçekleştirilmiştir.

Sen-Ching C. ve arkadaşları tarafından yapılan çalışmada [16] görüntü işleme yöntemlerinde kullanılan arka plan çıkarım yöntemleri olan Frame Differencing, Approximated Median Filter, Kalman Filter, Median Filter ve Mixture of Gaussian yöntemleri incelemesi yapılmıştır. İncelemeler sırasında farklı hava şartları, yoğunluk ve ışık durumlarında araç tespitinde kullanılmak üzere uygun parametreler belirlenmeye çalışılmış ve yöntemler arası kıyaslamalar yapılmıştır. Yapılan kıyaslamalarda en iyi yöntemin Mixture of Gaussian yöntemi olduğu görülmüştür.

Nastaran E. ve arkadaşları tarafından yapılan çalışmada [17] farklı yoğunluk, hava durumu, yol cinsi, ışık şiddeti ve mekânlarda araç tespiti yaklaşık %94.02 başarılı olarak gerçekleştirilmiştir.

Gökçe Çiçek Ş. tarafından yapılan çalışmada [18] görüntü işleme yöntemleri kullanarak gerçek hız değerlerine oldukça yakın hız değerlerinin tespit edilmiştir. Düzgün yol çizgileri ve trafik akışının bulunmadığı şartlarda yapılacak hız kestirimlerine alternatif bir çalışma olmuştur.

1.2.2 Derin Öğrenme Yöntemleri

Bu bölümde görüntü sınıflandırma uygulamaları için CNN algoritması ile yapılan benzer çalışmalardan bahsedilmektedir.

Wei-Ta C. ve arkadaşları tarafından yapılan çalışmada [19] Avrupa'nın birçok şehrinde temin edilmiş yüzlerce görselden SVM mimarisi kullanılarak sıcaklık, nem ve hava durumu tahmini yapılmıştır. Çalışmanın sonunda kameralardan elde edilen görüntülerden oluşturulan veri tabanı kullanıldığında %76.6 ortalama ile hava durumunun doğru tahmin edildiği görülmüştür. Ancak çalışmada kullanılan 5 adet (güneşli, bulutlu, karlı, yağmurlu ve sisli) hava durumundan yağmurlu ve bulutlu hava durumları için performansın %76.6'lık genel ortalamanın altında olduğu görülmüştür.

Boukaye T. ve arkadaşları tarafından yapılan çalışmada [20] salgın patojenlerin sınıflandırılmasında CNN mimarisi kullanılmıştır. Veri setinde 200'er adet vibrio kolera ve plasmodium falciparum patojenlerinin görseli mevcuttur. Bu çalışma sonucunda ilgili patojenlerin sınıflandırması %94 oranında başarılı gerçekleştirilmiştir.

Xingcheng L. ve arkadaşları tarafından yapılan çalışmada [21] yaklaşık 90.000 görsel ve 998 araç sınıfı bulunduran veri tabanı için araç tespiti algoritmaları oluşturulmuştur. Uygulama sonucunda araç tespiti için %92.25 başarımlı oranı elde edilmiştir.

Linkai C. ve arkadaşları tarafından yapılan çalışmada [22] kamera yayınlarından alınan 5.000 adet görsel ile oluşturulan veri tabanı kullanılmıştır. Faster R-CNN algoritması kullanılarak oluşturulan model sonucunda çeşitli tip araçlar için araç tespiti ortalama %75.7 başarımlı oranı ile gerçekleştirilmiştir.

Zheng Z. ve arkadaşları tarafından yapılan çalışmada [23] 4000 adet görselden oluşan veri tabanı ile Çoklu Çekirdek Öğrenmesi (MKL) kullanılarak güneşli, yağmurlu, karlı ve sisli hava durumları sınıflandırması uygulanmıştır. Genel ortalama olarak %71.39 başarımlı oranı elde edilmiştir. Bu modelde yağmurlu görsellerin %32'lik kısmı hatalı olarak karlı olarak sınıflandırılmıştır.

Xiaohu Z. ve arkadaşları tarafından yapılan çalışmada [24] üç farklı veri tabanından alınan sesler kullanılarak oluşturulan D-CNN algoritmasıyla ses sınıflandırmasında %81.9, %68.1 ve %87.1 başarımlı oranları elde edilmiştir.

Gongfa L. ve arkadaşları tarafından yapılan çalışmada [25] 8 farklı el hareketi için oluşturulan 2000 adet görsel içeren veri tabanı kullanılmıştır. Bu veri tabanından CNN algoritması kullanılarak oluşturulan model ile el hareketlerinin sınıflandırılması %98.52 oranında başarımlı oranı gerçekleştirilmiştir.

Uday S. ve arkadaşları tarafından yapılan çalışmada [26] antaknoz hastalığına yakalanmış mango yapraklarının sınıflandırılmasında CNN algoritması kullanılmış ve %97.13 başarımlı oranı alınmıştır.

Martin R. ve arkadaşları tarafından yapılan çalışmada [27] yaklaşık 500.000 adet görselden oluşan veri tabanı kullanarak tasarlanan SVM modeli ile açık, hafif yağışlı ve ağır yağışlı hava durumlarına tahmin gerçekleştirilmiş ve başarımlı oranı %85.19 olarak elde edilmiştir.

Hilal T. ve arkadaşları tarafından yapılan çalışmada [28] FCRN kullanılarak hazırlanan model ile uydu görüntülerinden araç sayımı çalışması %90.51 başarımlı oranı ile gerçekleştirilmiştir.

Muhammed Taha Z. tarafından yapılan [29] çalışmasında Faster R-CNN kullanılarak oluşturulan model ile uydu görüntülerindeki havaalanı tespiti %99.52 oranında başarılı şekilde gerçekleştirilirken, SSD algoritması ile bu oran %76.61 olmuştur.

2 GENEL BİLGİLER

Bu bölümde, çalışmada kullanılan yöntemlerin teknik bilgileri üç ana başlık altında aktarılmıştır.

2.1 Evrişimli Sinir Ağları (Convolution Neural Networks)

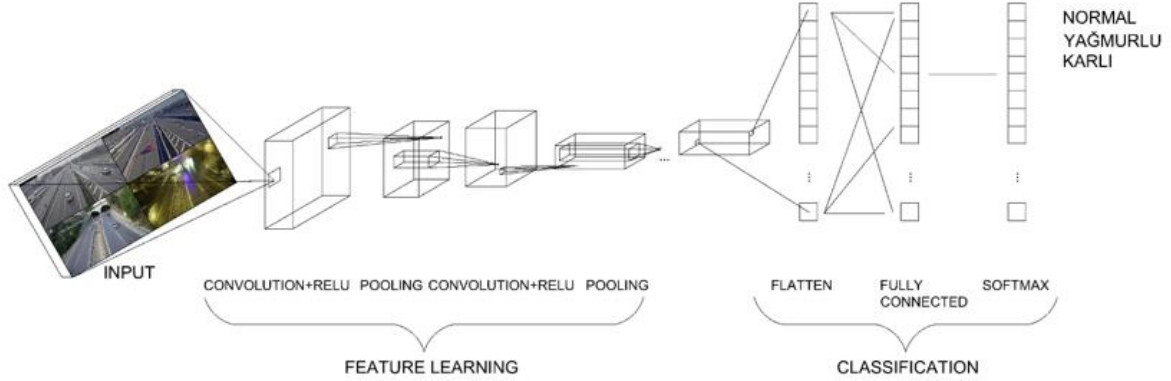
Yapay zekâ son yıllar sıklıkla karşımıza çıkan bir kavramdır. Bir makinenin, tıpkı bir insanın ilk anından son anına kadar geçen sürede çevresinde gerçekleşen olaylara, olgulara verdiği tepkiler gibi çevresinde veya girdi olarak verilenlere gösterdiği tepki o makinenin yapay zekâyâ sahip olduğunun göstergesidir. Bir makinenin zeki olup olmadığı sorusu ilk olarak 1950 yılında Alan Turing tarafından tasarlanan ve “Computing Machinery and Intelligence” makalesinde [30] sunulan Imitation Game adındaki test ile sorgulanmıştır.

Bilgisayarların hayatlarımıza girişinden bu yana daha yüksek sesle duyulan, başlarda insanları ürkütebilen ancak son zamanlarda heyecan yaratan Yapay Zekâ (YZ) kavramı, insanın düşünce ve karar yeteneklerini modellemek amacıyla birçok gelişim yaşamıştır. İlk olarak 1943 yılında McCulloch ve Pitts tarafından insan sinir sisteminden feyz alınarak beyin fonksiyonlarının mantıksal modelinin ortaya konulduğu A Logical Calculus of The Ideas Immanent in Nervous Activity çalışması [31] ile başlayan bu gelişim serüveni, 2012 yılında yapılan ImageNet Large Scale Visual Recognition Challenge yarışmasında Krizhevsky ve Sutskever tarafından ortaya konulan ImageNet Classification with Deep Convolutional Neural Networks çalışması [32] ile temel seviyesine ulaşmıştır.

Derin öğrenme işlemlerinde kullanılan yöntemler bu tarihlerden itibaren çok büyük değişikliğe uğramamasına rağmen, bu işlemlerde kullanılabilecek veri setlerinin zamanla dijitalleşen dünyada daha rahat temin edilebilmesinden dolayı derin öğrenme kavramının popülerliği artmıştır. Bu büyük veri setlerine uygun donanımsal gelişmelerle birlikte, artık günümüzde hemen hemen her alanda derin öğrenme yöntemleri kullanılır olmuştur.

2.1.1 Evrişimli Sinir Ağları Mimarisi

Evrişimli Sinir Ağları (CNN) mimarisi genellikle aşağıdaki katmanlardan oluşmaktadır. Şekil-7’de CNN katman mimarisi görsel diyagram olarak verilmiştir.



Şekil-7 CNN Katman Mimarisi

2.1.1.1 Evrişim Katmanı (Convolutional Layer)

Girdinin özelliklerini belirlemek için kullanılan evrişim katmanı, evrişim işlemlerini gerçekleştiren filtrelerin, girdide yatay ve dikey olarak tarandığı adımdır. Örneğin girdi olarak 1 ve 0’lardan oluşan Şekil-9’daki resmin Şekil-8’de verilen CNN filtresi ile tarandığında elde edilecek çıktı özellik haritası Şekil-10’daki [33] gibidir. Burada filtre, örnek resmin sol üst köşesinden başlayarak gezdirilirken iki matris arasındaki değerler çarpılır ve sonuçlar toplanır. Toplanan sonuçlar özellik haritasında saklanır. Bu işlem görseldeki her satır ve sütundaki elemanlar için tekrarlanır. Şekil-10’da örnek CNN filtresini görseldeki son işlemi gösterilmektedir. Özellik haritasının (3,3) elemanı olan 4 değeri aşağıdaki işlem sonucunda elde edilmektedir.

$$(1 \times 1) + (1 \times 0) + (1 \times 1) + (1 \times 0) + (1 \times 1) + (0 \times 0) + (1 \times 1) + (0 \times 0) + (0 \times 1) = 4$$

Aşağıda görsel olarak verilen evrişim işleminin matematiksel ifadesi [34] Denklem (1)’de gösterildiği gibidir.

$$S(i, j) = (I * K)(i * j) = \sum_m \sum_n I(i + m, j + n) K(m + n) \quad (1)$$

Denklemde S evrişim işlemi çıktısını, I girdiyi, K çekirdeği (kernel) ve $*$ evrişim işlemini ifade etmektedir.

1	0	1
0	1	0
1	0	1

Şekil-8 Örnek CNN Filtresi

1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

Şekil-9 Örnek Resim Girdi

1	1	1	0	0
0	1	1	1	0
0	0	1 _{x1}	1 _{x0}	1 _{x1}
0	0	1 _{x0}	1 _{x1}	0 _{x0}
0	1	1 _{x1}	0 _{x0}	0 _{x1}

Resim

4	3	4
2	4	3
2	3	4

Özellik Haritası

Şekil-10 Filtre Uygulaması

Tablo-2’de [29] yaygın olarak kullanılan filtreler ve matematiksel ifadeleri verilmektedir.

Tablo-2 Filtre Örnekleri

Operasyon	Filtre	Sonuç Görüntü
Tanımlama	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	
Kenar Tespiti	$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$	
	$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$	

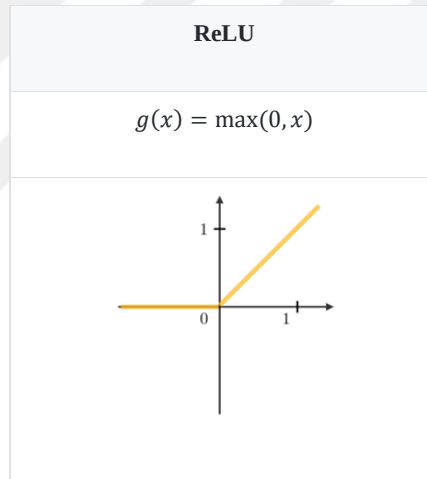
	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$	
Keskinleştirme	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	
Bulanıklaştırma	$\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	
Gaussian Bulanıklaştırma 3 x 3	$\frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$	
Gaussian Bulanıklaştırma 5 x 5	$\frac{1}{256} \begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}$	
Keskinlik Azaltım Maskesi 5 x 5	$\frac{-1}{256} \begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}$	

▪ Düzeltilmiş Doğrusal Birim Katmanı (ReLU)

Evrişim katmanlarında sonra gelen bu katman ile nöron çıktıların diğer katmanlara hazır hale gelmesi için doğrultulduğu katmandır. Evrişim katmanından sonra gelen verilerdeki negatif değerler sıfıra çekilir ve öğrenme sırasında kullanılmaz. Bu sebepten ötürü ReLU aktivasyon fonksiyonu çok katmanlı sinir ağlarında sıklıkla tercih edilmektedir. Bu katmanda yapılan işlemler sayesinde derin öğrenme daha da hızlanır. Aşağıdaki Tablo-3’de [34] yaygın olarak kullanılan aktivasyon fonksiyonlarına örnekler bulunmaktadır. Denklem (2)’de ReLU fonksiyonun matematiksel ifadesi verilmiştir.

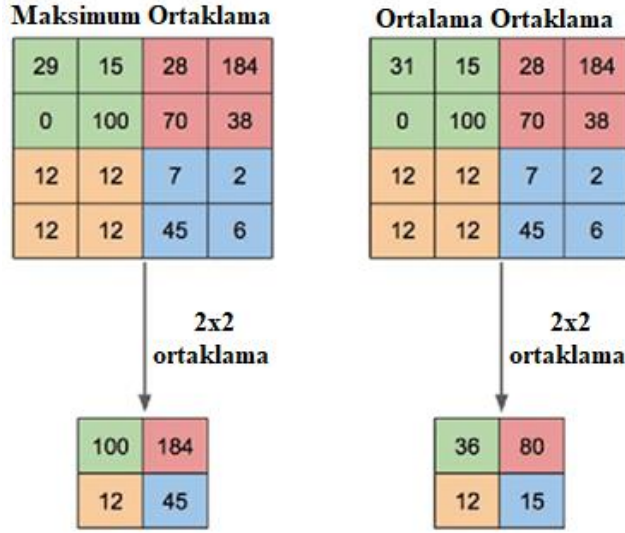
$$g(x) = \max(0, x) = \begin{cases} x, & x \geq 0 \\ 0, & x < 0 \end{cases} \quad (2)$$

Tablo-3 ReLU Aktivasyon Fonksiyonu



2.1.1.2 Ortaklama Katmanı (Pooling Layer)

Ortaklama katmanı ağ mimarisinde hesaplamaların yapıldığı katmandır. Bu katmanda oluşturulan modelin başarısı hesaplanıp, boyutsallık azaltılmak istenir. Böylece gereken güç ve belirlenen ancak kullanılmayacak özelliklerin filtrelenmesi sağlanır. Maksimum ve ortalama ortaklama olarak iki farklı ortaklama yöntemi bulunmaktadır. Aşağıda Şekil-11’de [35] maksimum ve ortalama ortaklama işlemleri gösterilmiştir.

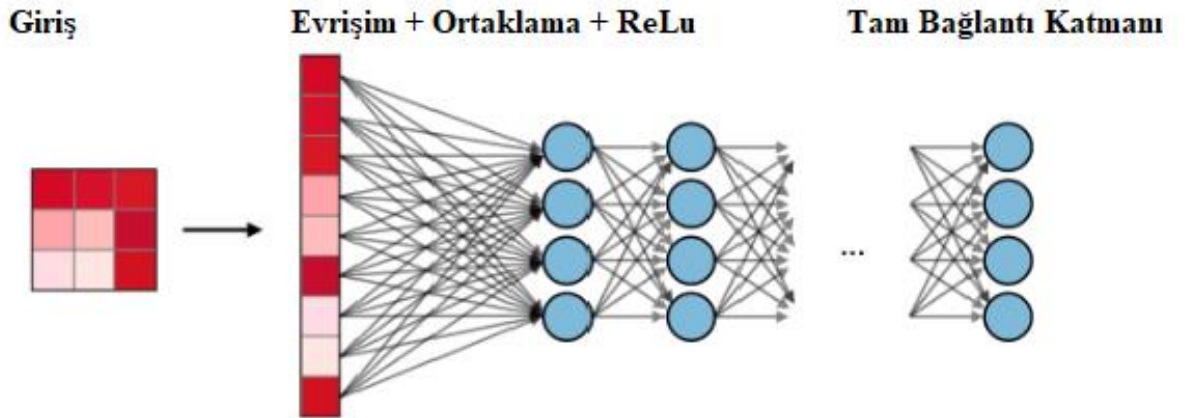


Şekil-11 Maksimum ve Ortalama Ortaklama

Şekil-11'deki [34] maksimum ortaklama işleminde modeldeki her grubun en yüksek piksel değeri 2x2 filtreden geçirilip çıkarılmaktadır. Ortalama ortaklama işleminde ise her grubun 2x2 filtreden geçirilmeden önce ortalamaları alınıp çıkarılmaktadır. Her iki işlemde de 2 adım kaydırma yapılmıştır.

2.1.1.3 Tam Bağlantı Katmanı (Fully-Connected Layer)

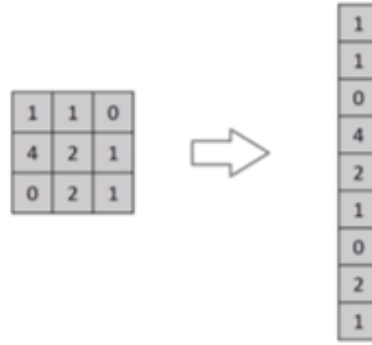
Bu katmanda birkaç kez evrişim katmanından ve ortaklama katmanından geçen ve matris halinde olan girdi görsel, sınıflandırma katmanına uygun olacak şekilde düz bir vektör haline getirilir. Şekil-12'de [34] evrişim işlemleri sonrasında çıkan nöronların tam bağlantı katmanında tek ve düz vektör olarak sınıflandırma katmanına hazırlanma süreci gösterilmektedir.



Şekil-12 Tam Bağlantı Katmanı

▪ Düzleştirme İşlemi

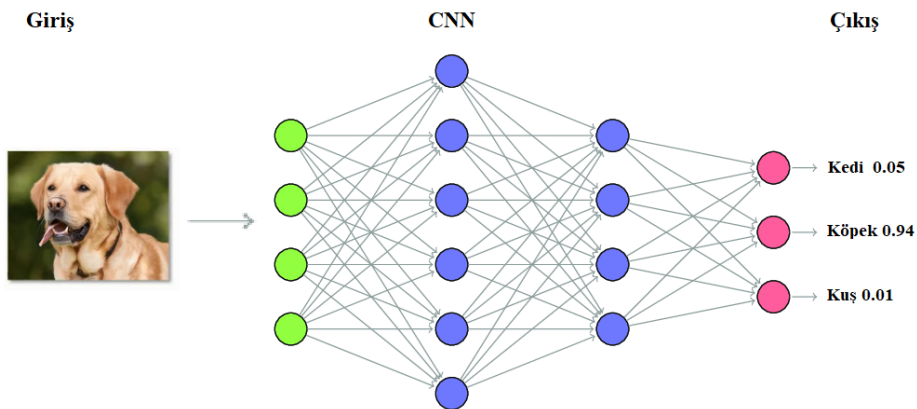
Bu katmanda son katman olan tam bağlantı katmanına girecek veriler hazırlanır. Önceki katmanlarda yapılan işlemler tamamlandıktan sonra elde edilen matris tam bağlantı katmanda kullanılabilmesi için düzleştirilmesi gerekir. Bu amaç için veriler tek boyutlu dizi haline bu katmanda çevrilir. Şekil-13’de [33] ortaklama işleminden sonra elde edilen verinin tam bağlantı katmanında girdi olarak kullanılabilmesi için yapılan işlem gösterilmektedir.



Şekil-13 Düzleştirme İşlemi

2.1.1.4 Sınıflandırma Katmanı (Classification Layer)

Bu katman tam bağlantı katmanından sonra gelerek nöronlardan gelen verileri sınıflandırma yapılacak nesne sayısı kadar mantıklı değerlere çevirir. Örneğin bu katmana giren 1024 nöron, çıkış değeri olarak 3 farklı nesne için 1024x3 boyutunda ağırlık matrisi olarak çıkmaktadır. Bu katmanda en yaygın olarak kullanılan sınıflandırıcı softmax’dır. Softmax tahmin edilecek her nesne için 0-1 arasında değerler üretir, 1’e en yakın değeri olan nesne modelin çıktısı olarak tahmin edilir. Şekil-14’de sınıflandırma işlemi için örnek çıktı gösterilmektedir.



▪ **Softmax**

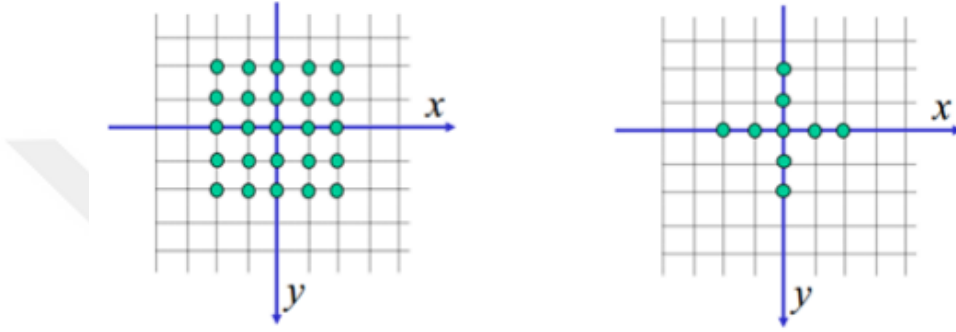
Genellikle n adet farklı olayın olasılık dağılımının hesaplandığı sinir ağlarının son katmanında kullanılır. En önemli avantajı birden çok sınıflı işleyebilir olmasıdır. Matematiksel ifadesi Denklem (3)'de [34] görülmektedir.

$$Softmax(x_i) = \frac{e^{x_i}}{\sum_{j=1}^n e^{x_j}} \quad (3)$$

Bu çalışmada CNN algoritması kullanılarak kamera görüntülerinden oluşan veri tabanı ile derin öğrenme gerçekleştirilmiştir. Derin öğrenme algoritma yöntemi olarak CNN seçilmesinin en önemli nedeni CNN algoritmasının veri tabanı ön işlemlerinin diğer yöntemlere göre çok daha az ve öğrenme işlemlerinin çok daha hızlı olmasıdır. Derin öğrenme çıktısı olarak hava durumu Softmax algoritması sonucu olarak alınmıştır. Model çıktısı hava durumu tespitinde kullanılacak olup belirli periyodlarla model sorgusu gerçekleştirilmiştir. Model çıktısına göre hava durumu tespit edildikten sonra uygulamada kullanılacak olan görüntü işleme parametreleri güncellenip hız limiti belirlenmiştir. Uygun parametreler ve hız limitine göre araç hız denetimi uygulanmıştır.

2.2 Morfolojik Dönüşümler

Morfolojik görüntü işleme, bir görüntüdeki özelliklerinin şekli ve geometrisiyle ilgili doğrusal olmayan işlemlere dayalı analiz yöntemidir. Genellikle ikili görüntülerde ayrılmak istenilen alanların/cisimlerin ilgi bölgesi dışında kalan nesnelerden temizlenmesinde kullanılır. Bu işlemler sırasında girdi görsel üzerinde yapılandırma elemanı gezdirilmektedir. Görselin piksel değeri ve yapılandırma elemanının değerine göre çeşitli matematiksel işlemler yapılmaktadır. Şekil-15’de [36] yapılandırma elemanı örnekleri mevcuttur.



Şekil-15 Yapılandırma Elemanı Örneği

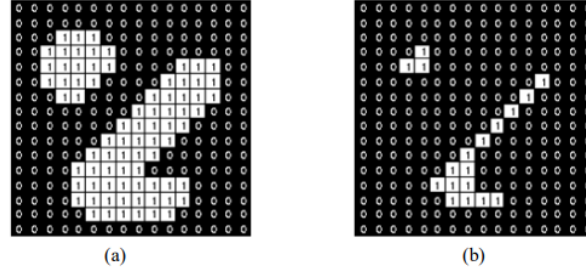
2.2.1 Aşınma ve Genişleme

Aşınma: Aşınma görüntüdeki nesneleri küçültmek ve aşındırmak için kullanılır. Yapılandırma elemanı görsel üzerinde piksel-piksel gezdirilerek kıyaslama yapılır. Yapılandırma elemanının merkez değeri görselde 1 değerine denk geldiği konumda yapılandırma elemanın diğer 1 değerleri altında görsele ait 0 değeri varsa, yapılandırma elemanının diğer 1 değerleri altındaki görsel pikselleri 0’a çevrilir. Aşınma zirveleri azalttığından ve görseldeki 0 pikseli bölgelerin genişliğini büyüttüğünden istenmeyen gürültüyü kaldırmak için kullanılmaktadır.

Aşağıda Aşınma işleminin matematiksel ifadesi Denklem (4)’de [36] belirtilmiştir. A ham görüntüyü B ise yapılandırma elemanını ifade etmektedir.

$$\varepsilon(A) = \bigcap_{b \in B} A - b = A \ominus B \quad (4)$$

Şekil-16'da [36] verilen (a) görüntüsü üzerinde Aşınma işlemi uygulandığında elde edilen (b) görüntüsü verilmiştir.



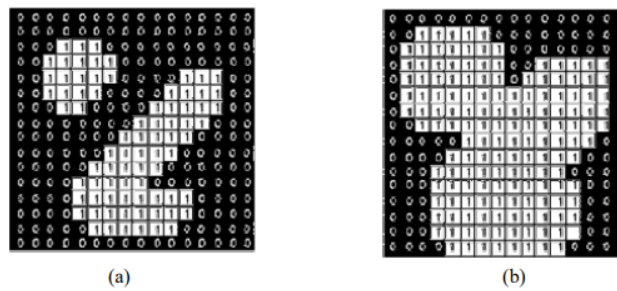
Şekil-16 3x3 Aşınma İşlemi

Genişleme: Genişleme görüntüdeki nesneleri büyültmek ve genişletmek için kullanılır. Yapılandırma elemanı görsel üzerinde piksel-piksel gezdirilir. Yapılandırma elemanı merkezi görselde 1 değerli piksele denk geldiğinde, diğer değerleri altında kalan görsel piksel değerleri 1 olarak değiştirilir. Genişleme görseldeki 1 piksellli bölgelerin genişliğini genişletir, böylece olumsuz gürültüleri giderebilir, ancak istenmeyen gürültüleri de artırabilir.

Aşağıda Denklem (5)'de [36] Genişleme işleminin matematiksel ifadesi belirtilmiştir. A ham görüntüyü B ise yapılandırma elemanını ifade etmektedir.

$$\delta(A) = \bigcap_{b \in B} A + b = A \oplus B \quad (5)$$

Şekil-17'de [36] verilen (a) görüntüsü üzerinde Aşınma işlemi uygulandığında elde edilen (b) görüntüsü verilmiştir.



Şekil-17 3x3 Genişleme İşlemi

2.2.2 Açma ve Kapama

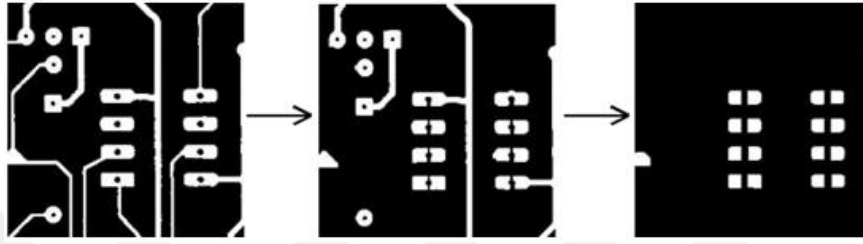
Açma: Orijinal görüntüye ilk önce Aşınma ardından Genişleme işlemi yapılır. Yapılandırma elemanı görselde piksel-piksel gezdirilirken, görüntüdeki daha az değerli pikselleri eleyip

diğer bölgelerin aynen kalmasını sağlar. Böylece görseldeki ince çizgiler, benekler ve istenmeyen gürültüler elenmiş olur.

Aşağıda Denklem (6)'da [36] Açma işleminin matematiksel ifadesi belirtilmiştir. A ham görüntüyü B ise yapılandırma elemanını ifade etmektedir.

$$A \circ B = (A \ominus B) \oplus B \quad (6)$$

Şekil-18'deki [36] 3 adet görselde sırasıyla; orijinal görsel, 5x5 Açma işlemi uygulanmış görsel, 9x9 Açma işlemi uygulanmış görsel.



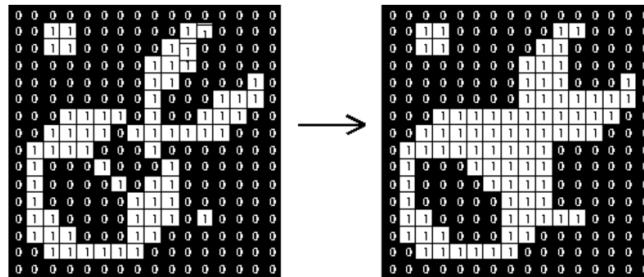
Şekil-18 Açma İşlemi Örnekleri

Kapama: Orijinal görüntüye ilk önce Genişleme ardından Aşınma işlemi yapılır. Kapama işleminde boşlukların doldurulması hedeflenmektedir. Yapılandırma elemanı görüntüde piksel-piksel gezdirilirken, eleman merkezine görselin 1 değerli pikseli denk geldiği durumda elemanın diğer değerleri altında kalan az değerli piksellerin 1'e çevrilmesini sağlar

Aşağıda Denklem (7)'de [36] Kapama işleminin matematiksel ifadesi belirtilmiştir. A ham görüntüyü B ise yapılandırma elemanını ifade etmektedir.

$$A \cdot B = (A \oplus B) \ominus B \quad (7)$$

Şekil-19'daki [36] görselde Kapama işlemi sonucunda nesnedeki gürültülerin ve boşlukların kapandığı görülebilmektedir.



Şekil-19 3x3 Yapı Elemanı ile Kapama İşlemi



2.3 Kalman Filtresi

Kalman filtresi [37] 1960 yılında R. E. Kalman tarafından gürültülü ölçüm değerlerinden faydalı sinyalin çıkarılması amacı ile yayımlanmıştır. R. E. Kalman bu yaklaşımı durum uzayı formatında doğrusal dinamik sistemlere dayalı durumları tahmin etmek için kullanmıştır. Daha genel ifadeyle Kalman filtresi bir modelin bir önceki durumdaki verilerine göre bir sonraki durumunu tahmin edebilen bir algoritmadır. Bu algoritmanın filtre olarak adlandırılmasındaki en büyük neden ölçümlerdeki gerçek değer ile gürültüyü gerçekleştirdiği tahminlere göre optimize ederek ayrıştırabiliyor olmasıdır.

Radar, lazer ve sonar gibi sensörlerin kullanıldığı sistemlerde Kalman filtresi genellikle sadece hedefin pozisyonu (POM Kalman Filtresi) ile ilgilenmektedir. Genel manada hareketli objenin takibi için Z_k ölçüm vektörü, H (ölçüm matrisi) ve R (kovaryans matrisi) değerleri aşağıdaki Denklem (8) ve Denklem (9)'daki [38] gibi ifade edilmektedir.

$$Z_k = H_{x_{tk}} + v_k \quad (8)$$

Bu denklemde belirtilen ölçüm modelinde Z_k ölçüm vektörünü, H ölçüm matrisini ve v_k ise R kovaryans matrisi ile ölçüm gürültüsünü ifade etmektedir.

$$H = (1 \ 0), R = (B_x) \quad (9)$$

Burada B_x değeri pozisyon ölçüm hatalarının bir varyantıdır. Filtrede $k-1$ anındaki durumdan k anına kadar olan değişimi aşağıdaki Denklem (10) ve Denklem (11)'de verildiği gibi tanımlanabilmektedir.

$$\tilde{x}_k = \Phi \hat{x}_{k-1} \quad (10)$$

$$\hat{x}_k = \tilde{x}_k + K_k(Z_k - H\tilde{x}_k) \quad (11)$$

Bu denklemlerde Kalman tahmini işleminin $\hat{\cdot}$ işareti ile Kalman kestirimi ise $\tilde{\cdot}$ işareti gösterilmektedir. Φ geçiş matrisi, Z_k ölçüm vektörü, H ölçüm matrisidir. Bu bağlamda K_k hatayı en aza indirgeyen Kalman kazancını ifade etmektedir. Buradaki K_k 'nın matematiksel ifadesi Denklem (12)'de aşağıdaki gibidir.

$$K_k = \tilde{P}_k H^T (H \tilde{P}_k H^T + R) \quad (12)$$

Buradaki P_k hataların koverans matrisidir. Matematiksel olarak ifadesi Denklem (13) ve Denklem (14)'de paylaşıldığı gibidir.

$$\tilde{P}_k = \Phi \hat{P}_{k-1} \Phi^T + Q \quad (13)$$

$$\hat{P}_k = \tilde{P}_k - K_k H \tilde{P}_k \quad (14)$$

Buradaki Q işlem gürültüsünü, K_k hatayı en aza indirgeyen Kalman kazancını, H ölçüm matrisini, P_k hataların koverans matrisini, Φ geçiş matrisini ifade etmektedir

3 MATERYAL ve YÖNTEM

Çalışmada oluşturulan tüm derin öğrenme ve görüntü işleme işlemleri Intel® Core™ i7-9750H 2.60GHz CPU [39] işlemciye ve 16.0 GB RAM'e sahip bilgisayar ile CPU üzerinden koşturulmuştur. Çalışmada erişilen kameralar ve çalışma parametrelerinin girilmesi ve ayarların yapılabilmesi adına arayüz tasarlanmıştır. Bu arayüz tasarlanırken python [40] yazılım dili kullanılmış ve pyqt5 [41] kütüphanesi ile inşa edilmiştir. Diğer yardımcı kütüphaneler ise pafy [42], joblib [43] ve os [44] kütüphaneleridir.

Hava durumuna göre değişen hız sınırı uygulaması başlatılmadan önce yapılması gereken 3 adet ön işlem mevcuttur. CNN modelinin oluşturulması, ilgi bölgesi (ROI) seçimi ve yoğunluğu zaman aralığı seçimi. Aralık zamanı varsayılan olarak 1 saat ayarlıdır. Şekil-20'de programın akış diyagramı verilmiştir.

Uygulama başlatılmadan önce programda kullanılan her kamera için özgün CNN modelleri hazırlanmıştır. Her kamera için ayrı olarak SelfDb uygulaması ile elde edilen görseller bu modellerin oluşturulmasına kullanılmıştır. İlgi bölgesi seçimi kullanılacak kamera için daha önce yapılmamışsa gerçekleştirilmelidir. Kullanıcı tarafından köşe noktaları işaretlenerek oluşturulan dikdörtgen alan için *.config* uzantılı dosya oluşturulmaktadır. Daha önce ilgi bölgesi seçimi yapılmışsa tekrar seçim yapılmasına gerek kalmadan otomatik yükleme gerçekleşmektedir. İlgi bölgesi değiştirilmek istendiğinde uygulama arayüzündeki buton ile işlem tekrarlanabilmektedir. Araç yoğunluğu zaman aralığı seçimi program arayüzü üzerindeki buton ile yapılabilmektedir.

The flowchart illustrates the proposed system architecture for traffic accident detection and classification. The process begins with data collection from various sources, including cameras (Kameralardan), vehicle movement (Araç Yürüşü), CCTV, and other sensors. The data is then processed through feature extraction and model training. The system includes decision points for handling errors and detecting faults. Key components and steps include:

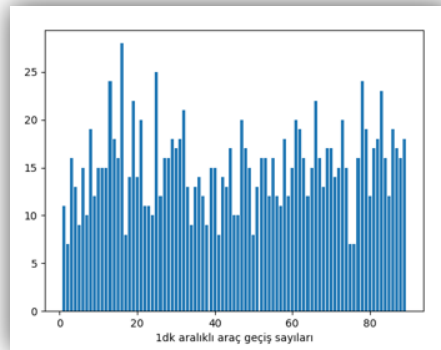
- Data Collection:** Kameralardan, Araç Yürüşü, CCTV, and other sensors provide input to the system.
- Feature Extraction:** The collected data is processed to extract relevant features.
- Model Training:** The extracted features are used to train the system's models.
- Real-time Detection:** The trained models are used to detect and classify traffic accidents in real-time.
- Decision Points:**
 - Hata Durumuna Giren Trafik Durumu Büyük mü?** This decision point checks if the detected traffic situation is a significant error. If YES, it leads to further processing; if NO, it leads to a different path.
 - Bu araca girilen araçlar arasında bir arıza var mı?** This decision point checks for faults among vehicles entering a specific area. If YES, it leads to a fault detection output; if NO, it leads to a fault-free output.
- Outputs:** The system generates various outputs, including accident detection reports, fault detection reports, and real-time monitoring dashboards.

28

Ön işlemler tamamlandıktan sonra program arayüzündeki listeden kamera seçimi yapılır. Seçilen kameranın anlık görüntüsü dilendiği zaman *play* butonu ile arayüzden görüntülenebilir. Hava durumuna göre değişen hız sınırı ile araç tespiti ve takibi işleminin (dokümanın devamında *process* işlemi olarak adlandırılacaktır) başlaması *process* tuşuna basıldığında gerçekleşir.

Tespit edilen aracın merkez noktası, ilgi bölgesinin giriş bölgesinden çıktığı an *Enter Time* sütununa kaydedilir, benzer durum ilgi bölgesinin çıkış bölgesinde gerçekleştiğinde *Exit Time* sütununda yapılmaktadır. İlgi bölgesi içerisinde birden fazla araç bulunması durumu veri tabanına yapılan kayıt işlemler için sorun teşkil etmemektedir. Ancak veri tabanında her araç için çıkış ve giriş değerlerinin bulunması durumunda tahmini hız değeri hesaplanabilmektedir. Anlık internet kesintileri, görüntü akışında gerçekleşen anlık duraksamalar veya çeşitli sebeplerden dolayı araçların giriş ya da çıkış zaman bilgisi bulunmayabilmektedir. Bu durumda araçların tahmini hız değerleri hesaplanamamaktadır fakat araç sayısında ya da araç yoğunluğu hesabında bu durumun olumsuz etkisi bulunmamaktadır.

Process işlemi sırasında klavyeden “q” tuşuna basıldığında anlık araç yoğunluğu grafiği kaydedilir ve ekranda gösterilir. Bu işlem sırasında *process* işlemi devam edebilmektedir. Araç yoğunluk grafiğinin kaydedilmesi *process* işlemi klavyeden “Esc” tuşuna basılarak sonlandırıldığında da gerçekleşmektedir. Şekil-21’de örnek araç yoğunluk grafiği mevcuttur.



Şekil-21 Araç Yoğunluğu Grafiği Örneği

Process işlemi başlatıldığı anda belirlenen hız sınırı, belirli periyotlarla CNN modeli üzerinden hava durumu tahmini gerçekleştirilerek güncellenmektedir. Hava durumuna göre belirlenen bu hız sınırına göre denetim her araç için hesaplanan tahmini hızlar kullanılarak

gerçekleştirilmektedir. Güncel hız sınırını aşan araçların ROI alanından çıkış anını ve hesaplanan hız değerlerini içeren anlık ekran görüntüleri ihlal dizinine kaydedilmektedir.

Yukarıda anlatılan işlemler, aşağıdaki başlıklarda ayrıntılı olarak verilmiştir.

3.1 Hava Durumu Tespiti İçin Model Oluşturulması

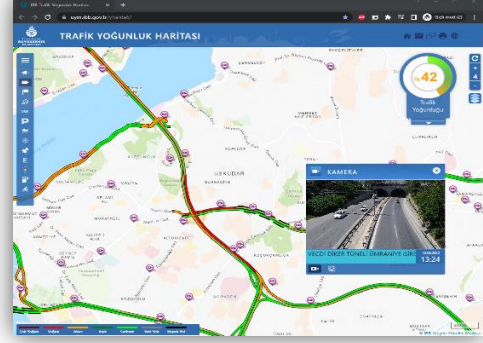
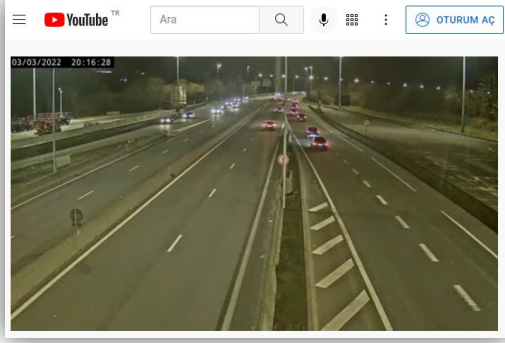
Hava durumu modeli oluşturulmak üzere, bu çalışmada kullanılacak kameralardan belirli aralıklarla, çeşitli mevsim ve yağış şartları içeren ekran görüntüleri alınmıştır. Kamera seçimlerinde çevrim içi yayın yapabilen mevcut altyapılar kullanılmıştır.

3.1.1 Veri Toplanması

Veri toplanması işlemi için python yazılım dili ile bu çalışmaya özel hazırlanan SelfDb uygulaması kullanılmış ve işlem otomatize edilmiştir.

Çalışmada kullanılacak kamera görüntüleri erişim linkleri ile birlikte tespit edildikten sonra her biri için özel olan dizinde SelfDb uygulaması üzerinden, saatte en fazla 10 olmak üzere 24 saat boyunca ekran görüntüleri kaydedilmiştir. Belirlenen kameralar arasında Fransa/Lyon şehrinde bulunan çevreyolu kameraları, Türkiye/İstanbul şehrinde bulunan birçok merkezi ve otoyol kamerası vb. bulunmaktadır. Birçok ülke ve şehirden, çevrim içi olarak erişilen bu kameraların teknik olarak herhangi bir özelliklerine gerek duyulmamaktadır ancak görüntü kalitesi, sabit kamera yapısı ve erişim istikrarı bu kameraları seçilmelerinde bir adım öne çıkarmaktadır.

Şekil-22'deki gibi YouTube [45] vb. çevrim içi platformlarda bulunan canlı yayın kameraları gibi kameraların yanı sıra, İstanbul Büyükşehir Belediyesi Ulaşım Yönetim Merkezi [46] vb. gibi şehir trafik yoğunluğu bilgisi için sunulan haritalarda paylaşılan kamera görüntüleri de bu sistemde kullanılabilir. İstanbul şehriden alınan kamera görüntüleri İBBUYM tarafından sağlanan trafik yoğunluk haritasından alınmıştır.



Şekil-22 Youtube ve İBBUYM Üzerinden Erişilen Canlı Kamera Görüntüleri

3.1.1.1 SelfDb Uygulaması

Bu çalışma için özel hazırlanan ve belirlenen kameralara erişim linklerini girdi olarak alan bu yazılım uygulaması ile tercih edilen aralıklar ve parametreler ile kamera görüntüleri ilgili dizine kaydedilmektedir. Bu çalışmada kamera görüntüleri için her saat en fazla 10 adet olacak şekilde 24 saat boyunca 5 dakika aralıklarla ve 6 ay boyunca, dört farklı mevsim şartlarını içerecek şekilde ekran görüntüleri kaydedilmiştir. Ekran görüntüleri 1280x720 çözünürlükte ve renkli görüntülerdir. Aşağıdaki Şekil-23’de SelfDb uygulamasının çalışma anına ait ekran görüntüsü bulunmaktadır.

```
camera #4.py - Kızıyol
Starting main.py at C:\Users\Emir\Desktop\camera #4
Started at 2022/03/31 17:00:01
C:\Users\Emir\Desktop\camera #4\03-31-17 dir created!
17.00.jpg saved!
17.05.jpg saved!
17.10.jpg saved!
17.15.jpg saved!
17.20.jpg saved!
17.25.jpg saved!
17.30.jpg saved!
17.35.jpg saved!
17.40.jpg saved!
17.45.jpg saved!
10 adet foto var!
Next Jobs: [Every 1 hour at 00:00:00 do run() (last run: 2022-03-31 16:45:02, next run: 2022-03-31 17:00:00)]
Starting main.py at C:\Users\Emir\Desktop\camera #4
Started at 2022/03/31 18:00:03
C:\Users\Emir\Desktop\camera #4\03-31-18 dir created!
18.00.jpg saved!
18.05.jpg saved!
18.10.jpg saved!
18.15.jpg saved!
18.20.jpg saved!
18.25.jpg saved!
18.30.jpg saved!
18.35.jpg saved!
18.40.jpg saved!
18.45.jpg saved!
10 adet foto var!
Next Jobs: [Every 1 hour at 00:00:00 do run() (last run: 2022-03-31 17:45:02, next run: 2022-03-31 18:00:00)]
Starting main.py at C:\Users\Emir\Desktop\camera #4
Started at 2022/03/31 19:00:01
C:\Users\Emir\Desktop\camera #4\03-31-19 dir created!
19.00.jpg saved!
19.05.jpg saved!
19.10.jpg saved!
19.15.jpg saved!
19.20.jpg saved!
19.25.jpg saved!
19.30.jpg saved!
19.35.jpg saved!
19.40.jpg saved!
19.45.jpg saved!
10 adet foto var!
Next Jobs: [Every 1 hour at 00:00:00 do run() (last run: 2022-03-31 18:45:02, next run: 2022-03-31 19:00:00)]
Starting main.py at C:\Users\Emir\Desktop\camera #4
Started at 2022/03/31 20:00:02
C:\Users\Emir\Desktop\camera #4\03-31-20 dir created!
20.00.jpg saved!
20.05.jpg saved!
20.10.jpg saved!
20.15.jpg saved!
20.20.jpg saved!
```

Şekil-23 SelfDb Uygulaması

Görüntü işleme yöntemlerinde kullanılan python kütüphanesi olan openCV [47] kütüphanesi ile gelen her görüntü çerçevesi 1280x720 formatına dönüştürülmektedir. Kamera görüntülerinin kaydedilmesinde parametreler kullanılmıştır. Her 60 dakikada uygulamanın çalıştığı kontrolü için *range_dk*, her 5 dakikada kamera anlık görüntüsü almak ve kaydetmek için *per_dk*, her saat başına en fazla 10 adet görsel sınırı için de *range_file* parametreleri tanımlanmıştır.

Aşağıdaki Şekil-24’de ay-gün-saat formatında her saat 10’ar adet anlık ve işlenmemiş kamera görüntüsü içeren dosya dizinleri görülmektedir. Bu işlem yaklaşık 6 ay boyunca kamera bağlantısının sağlanabildiği her saat için gerçekleştirilmiştir.

Ad	Değiştirme tarihi	Tür
Camera #13.py	8.02.2022 21:57	Python File
main.py	13.11.2021 13:25	Python File
02-08-22	8.02.2022 22:00	Dosya klasörü
02-08-21	8.02.2022 21:00	Dosya klasörü
02-08-20	8.02.2022 20:00	Dosya klasörü
02-08-19	8.02.2022 19:01	Dosya klasörü
02-03-23	3.02.2022 23:00	Dosya klasörü
02-03-22	3.02.2022 22:00	Dosya klasörü
02-03-21	3.02.2022 21:00	Dosya klasörü
02-03-20	3.02.2022 20:00	Dosya klasörü
02-03-19	3.02.2022 19:00	Dosya klasörü
02-03-18	3.02.2022 18:00	Dosya klasörü
02-03-17	3.02.2022 17:34	Dosya klasörü
01-29-00	29.01.2022 00:00	Dosya klasörü
01-28-23	28.01.2022 23:00	Dosya klasörü
01-28-22	28.01.2022 22:00	Dosya klasörü
01-28-21	28.01.2022 21:00	Dosya klasörü
01-28-20	28.01.2022 20:00	Dosya klasörü
01-28-19	28.01.2022 19:00	Dosya klasörü
01-28-18	28.01.2022 18:00	Dosya klasörü
01-28-17	28.01.2022 17:00	Dosya klasörü
01-28-16	28.01.2022 16:00	Dosya klasörü
01-28-15	28.01.2022 15:00	Dosya klasörü

Şekil-24 Kamera Görüntüleri Dizini

İşlem görmeden, SelfDb uygulaması ile kaydedilmiş bu görüntü havuzundan CNN algoritmasında kullanılmak üzere mantıklı ve yararlı görüntüler gözle seçilip giriş verisi olmaları sağlanmıştır. Şekil-25’de İstanbul Büyükşehir Belediyesi Ulaşım Yönetim Merkezi tarafından sağlanan trafik yoğunluk haritasından alınmış ekran görüntüleri bulunmaktadır. (a)’da normal hava durumu, (b)’de yağmur yağışlı hava durumu, (c)’de karlı hava durumu ve (d)’de sisli hava durumu için kullanılabilecek örnek kamera görüntüleri mevcuttur.



(a) İstanbul Normal Hava Durumu



(b) İstanbul Yağmurlu Hava Durumu



(c) İstanbul Karlı Hava Durum



(d) İstanbul Sisli Hava Durumu

Şekil-25 İstanbul Çeşitli Hava Durumları

3.1.2 Veri Modeli Oluşturulması

Yapılan bu çalışmada her kamera için ayrı elde edilen görüntüler kullanılarak 4 farklı çıkış verebilen modeller elde edilmiştir. Veri modelleri oluşturulurken python yazılım dili kullanılmıştır. SelfDb ile oluşturulan veri tabanlarının modellerin oluşturulması sırasında matematiksel işlemler için numpy [48] kütüphanesi, anlık zaman bilgisi için datetime [49] kütüphanesi, görseller üzerinde çalışılması için openCV kütüphanesi, işletim sistemi üzerinde gerçekleştirilen emirler için os kütüphanesi, veri tabanları içerisindeki görsellerin karıştırılması için random [50] kütüphanesi, görsellerden elde edilen verilerin içeri ve dışarı aktarılması için pickle [51] kütüphanesi, derin öğrenme ve model değerlendirmeleri işlemleri için tensorflow [52], sklearn [53] ve keras [54] kütüphaneleri, model değerlendirmelerinin görselleştirmeleri için seaborn [55] ve matplotlib [56] kütüphaneleri kullanılmıştır.

Tensorflow kütüphanesi CPU üzerinden ya da GPU üzerinden çalışabilmektedir. Çalışmada oluşturulan tüm derin öğrenme modelleri Intel® Core™ i7-9750H 2.60GHz CPU işlemciye ve 16.0 GB RAM'e sahip bilgisayar ile CPU üzerinden oluşturulmuştur.

Model oluşturulurken ilk olarak kullanılacak sınıflar belirtilmektedir. Ardından ilgili eğitim dizini altında bulunan tüm görseller openCV kütüphanesi ile okunarak 150x150 formatına küçültülmektedir. Ardından elde edilen eğitim verisi, görsel verilerini içeren X ve ilgili görsellerin sınıfını içeren y değişkenlerine aktarılmaktadır. Bu değişkenler tekrar kullanılabilirlik üzere pickle kütüphanesi ile dışarı aktarılmaktadır. Elde edilen $X.pickle$ ve $y.pickle$ dosyaları model eğitimi ve değerlendirmesinde kullanılmak üzere X_{train} , X_{test} , y_{train} , y_{test} olarak parçalara ayrılmaktadır. Veri tabanında bulunan görsellerin %70'i eğitimde, %30'u ise model değerlendirmesinde kullanılmaktadır.

Kategorik olarak elde edilen X_{train} ve y_{train} eğitim verileri derin öğrenme algoritmasında veri olarak sağlanarak, 2 adet evrişim katmanından oluşan ve aşağıda kod parçası verilen algoritma ile eğitim gerçekleştirilmiştir. Son olarak 10 tur eğitime tabii tutulan model istenilen dizine kaydedilmektedir.

SelfDb uygulaması ile elde edilmiş, 4 farklı hava durumuna ait görsellerden oluşan veri tabanları ile oluşturulan modeller ve veri tabanlarının demografik yapıları Tablo-4'de belirtilmiştir. Şekil-25'de örnekleri gösterilen görseller *Camera #13.model* modeli oluşturulurken kullanılan görsellere örnektir.

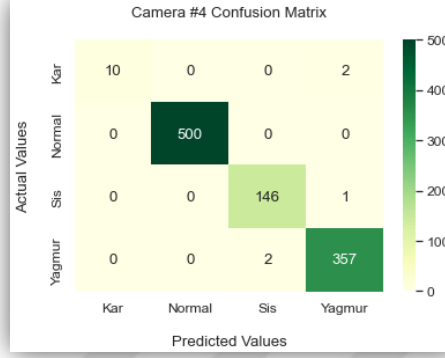
Tablo-4 Veri Setleri ve Demografik Yapıları

<i>CNN Modeli</i>	<i>Veri Seti Sınıfları</i>	<i>Örnek Uzantısı</i>	<i>Örnek Çözünürlüğü</i>	<i>Örnek Adetleri</i>	<i>İşlem Çözünürlüğü</i>	<i>Başarım Oranı</i>
Camera #4.model	Kar	.jpg	1280x720	38	150x150	%99.51
	Normal	.jpg	1280x720	1645	150x150	
	Sis	.jpg	1280x720	468	150x150	
	Yağmur	.jpg	1280x720	1241	150x150	
Camera #12.model	Kar	.jpg	1280x720	756	150x150	%99.85
	Normal	.jpg	1280x720	4185	150x150	
	Sis	.jpg	1280x720	35	150x150	
	Yağmur	.jpg	1280x720	1800	150x150	
Camera #13.model	Kar	.jpg	1280x720	1846	150x150	%99.60
	Normal	.jpg	1280x720	3500	150x150	
	Sis	.jpg	1280x720	66	150x150	
	Yağmur	.jpg	1280x720	2000	150x150	

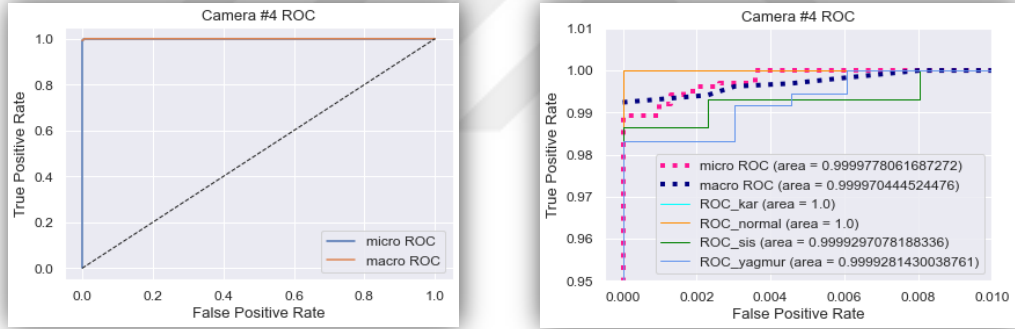
3.1.3 Veri Modelinin Değerlendirilmesi

Oluşturulan modellerin başarımlarının değerlendirilmesinde model verileri oluşturulurken ilgili kamera görüntülerini içeren X_{test} ve y_{test} verileri kullanılmıştır. Bu veriler kullanılarak elde edilen başarımlar %99.51 ile %99.85 arasında değişmektedir.

Tablo-4'de görülebildiği üzere, sırasıyla *Camera #4.model*, *Camera #12.model* ve *Camera #13.model* modellerinin başarımlar değerlendirilmeleri %99.51, %99.85 ve %99.60 olarak belirlenmiştir. Başarımlar değerlendirilmeleri sonrasında seaborn kütüphanesi kullanılarak oluşturulmuş başarımlar istatistik görselleri Şekil-26 ve Şekil-27'da *Camera #4.model* için verilmiştir.



Şekil-26 *Camera #4.model* Tahmin Değerleri Isı Haritası



Şekil-27 *Camera #4.model* ROC Eğrisi

Tablo-5'de *Camera #4.model* çıktıları için elde edilen başarımlar oranları ve değerleri verilmiştir.

Tablo-5 *Camera #4.model* Sonuçları

	kar	normal	sis	yagmur
FP (False Positive)	0	0	2	3
FN (False Negative)	2	0	1	2
TP (True Positive)	10	500	146	357
TN (True Negative)	1006	518	869	656
TPR (Sensitivity/Recall)	0.83333333	1	0.99319728	0.99442897
TNR (Specificity)	1	1	0.99770379	0.99544765
PPV (Precision)	1	1	0.98648649	0.99166667

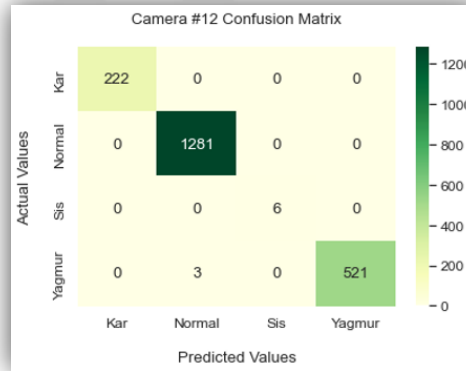
NPV (Negative Predictive)	0.99801587	1	0.99885057	0.99696049
FPR (False Positive Rate)	0	0	0.00229621	0.00455235
FNR (False Negative Rate)	0.16666667	0	0.00680272	0.00557103
FDR (False Discovery Rate)	0	0	0.01351351	0.00833333
ACC (Accuracy)	0.99803536	1	0.99705305	0.99508841

Tablo-6’da *Camera #4.model* için F1score değerleri belirtilmektedir. Tasarlanan model çok sınıflı bir model olduğundan, her sınıfın F1 skorunun ortalamalarından elde edilen F1weighted değeri ağırlıklı model skoru olarak kullanılmaktadır.

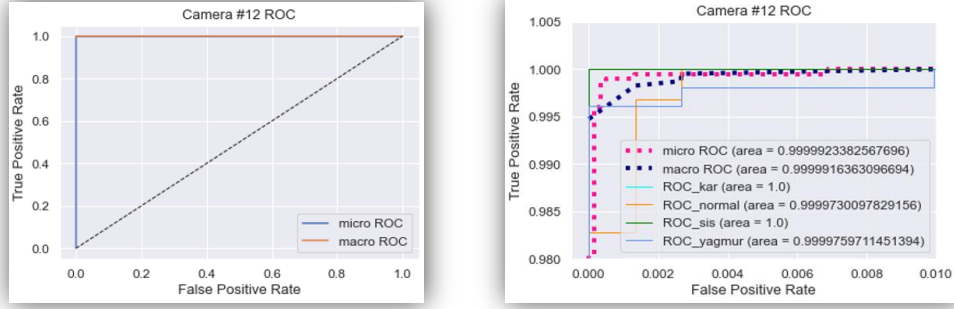
Tablo-6 *Camera #4.model F1 Skorları*

Puan	
F1_{micro}	0.9950884086444007
F1_{macro}	0.9729918286611905
F1_{weighted}	0.9950075173932358

Başarım değerlendirmeleri sonrasında seaborn kütüphanesi kullanılarak oluşturulmuş başarımlar istatistik görselleri Şekil-31 ve şekil-32’de *Camera #12.model* için verilmiştir.



Şekil-28 *Camera #12.model Tahmin Değerleri Isı Haritası*



Şekil-29 Camera #12.model ROC Eğrisi

Tablo-7’de Camera #12.model çıkışları için elde edilen başarımlar oranları ve değerleri verilmiştir.

Tablo-7 Camera #12.model Sonuçları

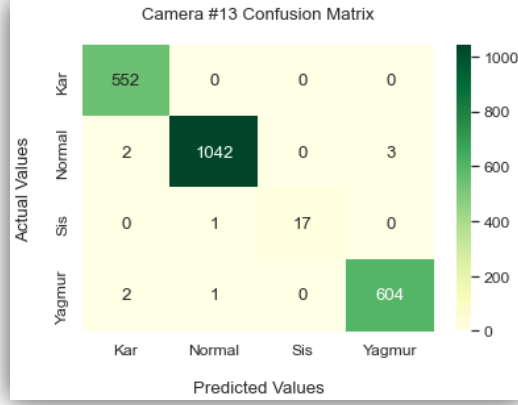
	Kar	Normal	Sis	Yagmur
FP (False Positive)	0	3	0	0
FN (False Negative)	0	0	0	3
TP (True Positive)	222	1281	6	521
TN (True Negative)	1811	749	2027	1509
TPR (Sensitivity/Recall)	1	1	1	0.99427481
TNR (Specificity)	1	0.99601064	1	1
PPV (Precision)	1	0.99766355	1	1
NPV (Negative Predictive)	1	1	1	0.99801587
FPR (False Positive Rate)	0	0.00398936	0	0
FNR (False Negative Rate)	0	0	0	0.00572519
FDR (False Discovery Rate)	0	0.00233645	0	0
ACC (Accuracy)	1	0.99852435	1	0.99852435

Tablo-8’de Camera #12.model için F1score değerleri belirtilmektedir. Tasarlanan model çok sınıflı bir model olduğundan, her sınıfın F1 skorunun ortalamalarından elde edilen $F1_{weighted}$ değeri ağırlıklı model skoru olarak kullanılmaktadır.

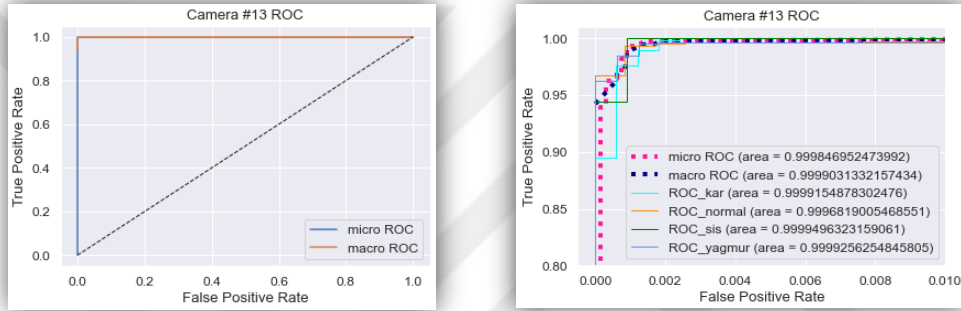
Tablo-8 Camera #12.model F1 Skorları

Puan	
$F1_{micro}$	0.9985243482538121
$F1_{macro}$	0.998989898989899
$F1_{weighted}$	0.9985230930476484

Başarım değerlendirmeleri sonrasında seaborn kütüphanesi kullanılarak oluşturulmuş başarımlar istatistik görselleri Şekil-30 ve Şekil-31’de Camera #13.model için verilmiştir.



Şekil-30 Camera #13.model Tahmin Değerleri Isı Haritası



Şekil-31 Camera #13.model ROC Eğrisi

Tablo-9'da Camera #13.model çıkışları için elde edilen başarımlar oranları ve değerleri verilmiştir.

Tablo-9 Camera #13.model Sonuçları

	Kar	Normal	Sis	Yagmur
FP (False Positive)	4	2	0	3
FN (False Negative)	0	5	1	3
TP (True Positive)	552	1042	17	604
TN (True Negative)	1668	1175	2206	1614
TPR (Sensitivity/Recall)	1	0.99522445	0.94444444	0.99505766
TNR (Specificity)	0.99760766	0.99830076	1	0.99814471
PPV (Precision)	0.99280576	0.99808429	1	0.99505766
NPV (Negative Predictive)	1	0.99576271	0.9995469	0.99814471
FPR (False Positive Rate)	0.00239234	0.00169924	0	0.00185529
FNR (False Negative Rate)	0	0.00477555	0.05555556	0.00494234
FDR (False Discovery Rate)	0.00719424	0.00191571	0	0.00494234
ACC (Accuracy)	0.99820144	0.99685252	0.99955036	0.99730216

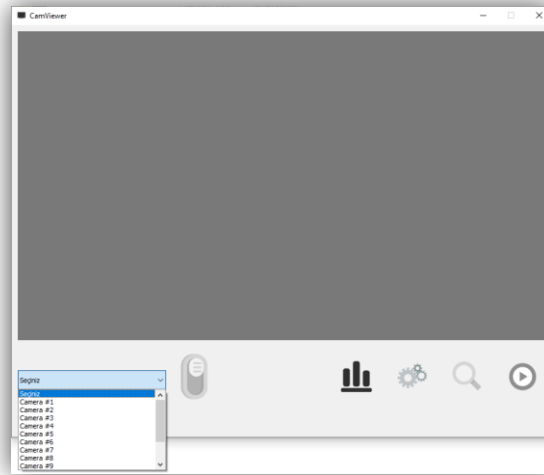
Tablo-10’da Camera #13.model için F1score değerleri belirtilmektedir. Tasarlanan model çok sınıflı bir model olduğundan, her sınıfın F1 skorunun ortalamalarından elde edilen F1weighted değeri ağırlıklı model skoru olarak kullanılmaktadır.

Tablo-10 Camera #13.model F1 Skorları

Puan	
F1_{micro}	0.9959532374100719
F1_{macro}	0.9898821108039307
F1_{weighted}	0.995947802599603

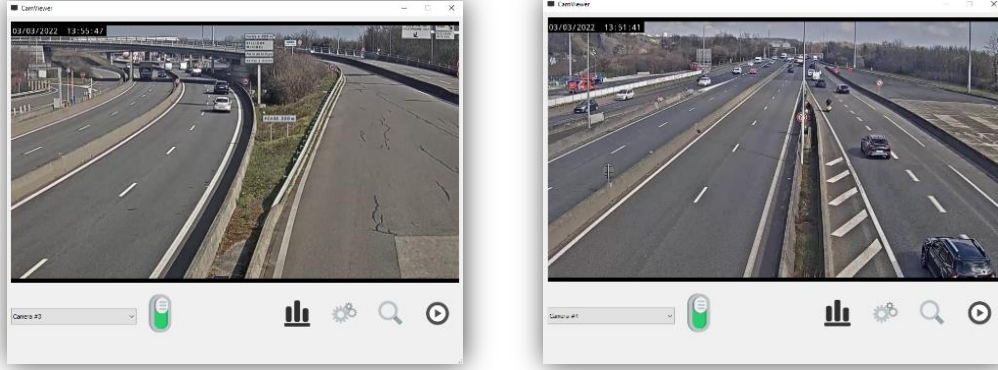
3.2 Araç Tespiti ve Takibi

Araç takibi, sabit kamera görüntüsünden arka plan çıkarımı ve morfolojik dönüşümler ile elde edilecek eşik değerinden büyük piksel alanlarının numaralandırılması ve takibi ile gerçekleştirilmektedir. Arayüzün orta alanı video ekranı olarak kullanılmaktadır. İlk açılışta ekran alanı gri renkte görülmektedir. Sol alt kısımda bulunan açılır listeden kamera seçilmesi sonrasında arayüz alt kısmında bulunan mesaj çubuğunda ilgili kameraya bağlanıldığı bilgisi görülmektedir. Şekil-32’de uygulama arayüzü gösterilmektedir.



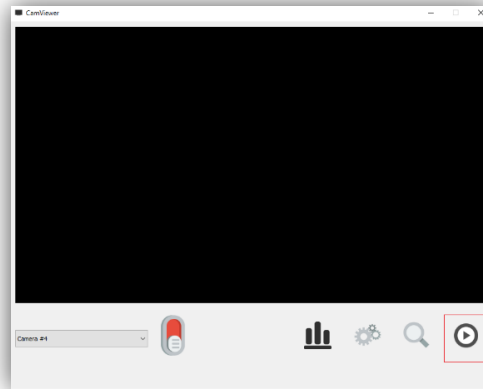
Şekil-32 Uygulama Arayüzü ve Kamera Listesi

Bağlanma başarılı olduktan sonra açılır listenin hemen yanındaki video oynat anahtarı, *process* butonu ve ROI seçim butonu aktif hale gelir. Şekil-33’de farklı kamera görüntüleri bulunmaktadır.



Şekil-33 Farklı Kamera Video Görüntüleri

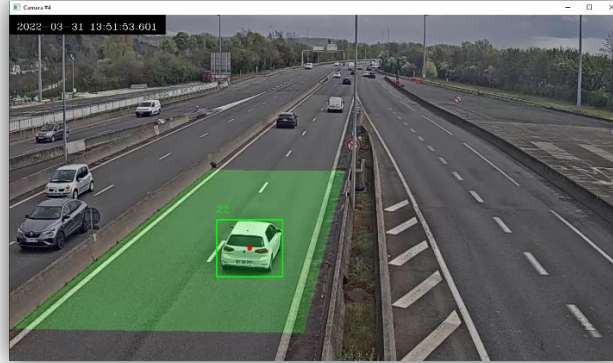
Video oynat anahtarı yeşil ikona gelecek şekilde değiştirildiğinde arayüz ekranında ilgili kameranın anlık video görüntüsü ekranda oynatılır ancak *process* işlemi bu durumda başlamamaktadır. Bu ekran anlık görüntünün izlenebilmesi için kullanılmaktadır. Kamera listesinden başka kameraya geçilmesi ya da video oynat anahtarının kırmızı konuma alınması sonrasında arayüzde bulunan video ekranı siyah renge döner ve kamera videosu kapatılır. Yeni seçilen kamera videosunun arayüz ekranında başlatılması için oynat anahtarı tekrar yeşil konuma alınmalıdır. Şekil-34’de kamera seçimi sonrası uygulama arayüzü gösterilmektedir.



Şekil-34 Kamera Seçimi ve Process İşlemi Başlatılması

Kamera seçimi, ROI alanı ve yoğunluk aralığı parametreleri sağlanıp, kamera video görüntüsü bağlantısı gerçekleştirilebildiği durumda *process* tuşuna basılırsa *process* işleminin

gerçekleştiği yeni bir video ekranı açılır. Aksi halde hata veya uyarı mesajları açılır pencerede gösterilir. Şekil-35’de *process* işleminin gerçekleştiği ekran görüntüsü verilmiştir.



Şekil-35 Camera #4 için Açılan Process Ekranı

Process ekranından bağımsız şekilde uygulama arayüzü üzerindeki video ekranında diğer kameralar izlenebilmektedir. *Process* işlemi sırasında her kamera için kamera isminin ayırt edilebileceği ayrı ekran açılmaktadır. Açılan ekranda klavyeden “q” tuşuna basıldığında yoğunluk zaman aralığı parametresine uygun şekilde anlık yoğunluk grafiği üretilir, kaydedilir ve ekranda gösterilir. Bu işlem zarfında *process* işlemi devam etmektedir. Yine açılan *process* ekranında klavyenin “Esc” tuşuna basıldığında *process* işlemi bitirilir, açılan *process* ekranı kapatılır ve tespit edilen araçların zaman ve hız bilgilerini içeren veri tabanı kaydedilir. Veri tabanları isimlendirilirken kamera numarası ve saat damgası kullanılmaktadır. Şekil-36’da kaydedilen veri tabanları verilmiştir.

GUI > results > csv			
Ad	Değiştirme tarihi	Tür	Boyut
11.58.26_Camera #12.csv	13.06.2022 11:58	Microsoft Excel Vir...	1 KB
10.08.42_Camera #13.csv	13.06.2022 10:08	Microsoft Excel Vir...	1 KB
10.08.05_Camera #12.csv	13.06.2022 10:08	Microsoft Excel Vir...	1 KB
10.07.54_Camera #13.csv	13.06.2022 10:07	Microsoft Excel Vir...	1 KB
10.07.51_Camera #13.csv	13.06.2022 10:07	Microsoft Excel Vir...	2 KB
10.07.37_Camera #12.csv	13.06.2022 10:07	Microsoft Excel Vir...	1 KB
08.58.27_Camera #13.csv	13.06.2022 08:58	Microsoft Excel Vir...	1 KB
08.58.12_Camera #12.csv	13.06.2022 08:58	Microsoft Excel Vir...	1 KB
08.53.27_Camera #4.csv	13.06.2022 08:53	Microsoft Excel Vir...	1 KB
08.53.08_Camera #13.csv	13.06.2022 08:53	Microsoft Excel Vir...	1 KB
14.21.29_Camera #4.csv	10.06.2022 14:21	Microsoft Excel Vir...	1 KB
13.48.57_Camera #4.csv	10.06.2022 13:48	Microsoft Excel Vir...	1 KB

Şekil-36 Kaydedilen Veri Tabanları

Aşağıdaki bölümlerde, örnek olması adına her hava durumu için gerçekleştirilen *process* işlemi çıktıları mevcuttur.

▪ Normal Hava Durumu

Normal hava durumunda *process* işlemi çıktıları örneği Şekil-37’deki gibidir.

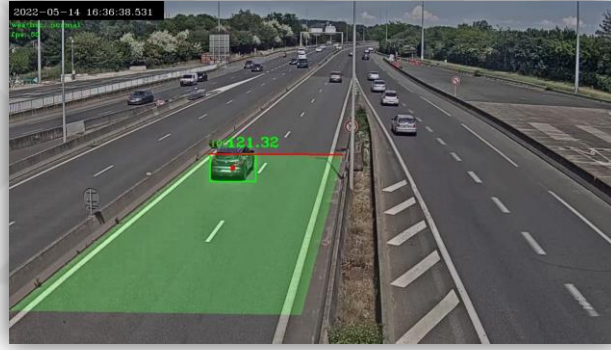
```
Loading weather model..
Model loaded..
Connecting Pafy...
Pafy Connected...

Camera #4 started
Hava Durumu: normal
Mesafe: 25.86m
Hız Sınırı: 120km/h
Yoğunluk_aralığı: 10sn
```

```
1.aracı ortalama hızı: 77.32 km/h
2.aracı ortalama hızı: 57.97 km/h
3.aracı ortalama hızı: 60.72 km/h
4.aracı ortalama hızı: 79.90 km/h
5.aracı ortalama hızı: 53.90 km/h
6.aracı ortalama hızı: 60.26 km/h
7.aracı ortalama hızı: 80.88 km/h
8.aracı ortalama hızı: 53.00 km/h
9.aracı ortalama hızı: 77.48 km/h
```

Şekil-37 Normal Havada İşlem Çıktıları

Şekil-38’de normal hava durumunda belirlenen 120km/h hız sınırını aşan araçlara ait anlık görüntü örneği mevcuttur.



Şekil-38 Normal Havada Hız Sınırı Aşımı Yapan Araç

▪ Yağmurlu Hava Durumu

Yağmurlu hava durumunda *process* işlemi çıktıları örneği Şekil-39’deki gibidir.

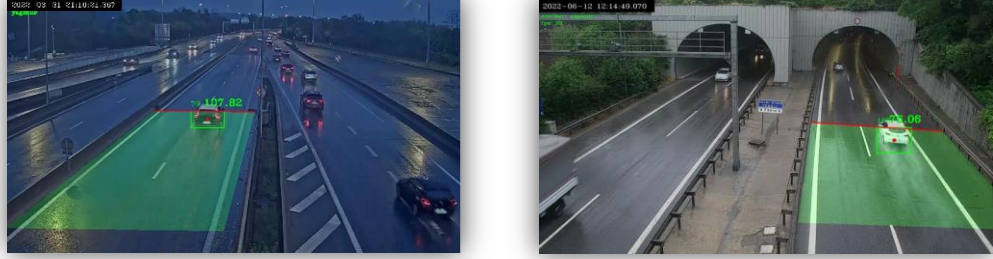
```
Loading weather model..
Model loaded..
Connecting Pafy...
Pafy Connected...

Camera #4 started
Hava Durumu: yağmur
Mesafe: 25.86m
Hız Sınırı: 100km/h
Yoğunluk_aralığı: 10sn
```

```
1.aracı hızı: 67.40 km/h
2.aracı hızı: 99.58 km/h
!!!İHLAL!!! @Camera #4: 3. araç hız sınırını 121.71 km/h hızla aştı
4.aracı hızı: 85.94 km/h
5.aracı hızı: 73.66 km/h
6.aracı hızı: 88.27 km/h
7.aracı hızı: 79.39 km/h
!!!İHLAL!!! @Camera #4: 8. araç hız sınırını 102.37 km/h hızla aştı
9.aracı hızı: 98.82 km/h
10.aracı hızı: 98.75 km/h
11.aracı hızı: 86.64 km/h
12.aracı hızı: 85.57 km/h
13.aracı hızı: 89.99 km/h
14.aracı hızı: 83.96 km/h
```

Şekil-39 Yağmurlu Havada İşlem Çıktıları

Şekil-40’de yağmurlu hava durumunda Camera #4 için belirlenen 100km/h hız sınırını ve Camera #12 için belirlenen 70km/h hız sınırını aşan araçlara ait anlık görüntü örnekleri mevcuttur.



Şekil-40 Yağmurlu Havada Hız Sınırı Aşımı Yapan Araçlar

- **Karlı ve Sisli Hava Durumu**

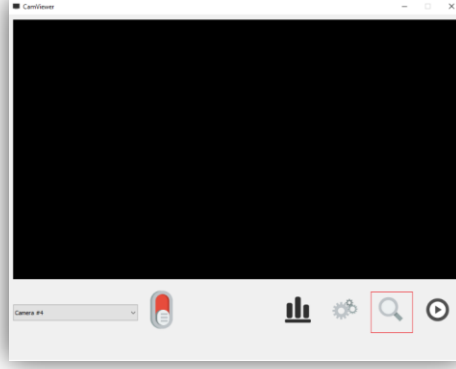
Şekil-41’de karlı ve sisli hava durumunda belirlenen 70km/h hız sınırını aşan araçlara ait anlık görüntü örnekleri mevcuttur.



Şekil-41 Karlı ve Sisli Havada Hız Sınırı Aşım Anları

3.2.1 İlgi Bölgesi (ROI) Belirleme

Arayüzde bulunan açılır listeden kamera seçimi yapılır. Sol alttan kameraya bağlanıldığı bilgisi kısa süreli görülür. Şekil-42’de gösterilen ROI seçimi butonuna tıklanır ve ROI seçim işlemi başlatılır. Ancak kamera seçimi sonrasında ilgili kamera için daha önce *config* dosyası oluşturulmadığı durumda *process* tuşuna basılırsa ROI seçimi yapılması gerektiği açılır pencere ile kullanıcıya belirtilerek otomatik olarak ROI seçimi işlemi başlatılmaktadır.



Şekil-42 ROI Seçim Butonu

Seçilen kameradan anlık tek kare görüntü yeni açılan ROI seçim ekranında görülür. Açılan ekranın sol üst köşesinde yapılacak işlem hakkında açıklayıcı bilgiler mevcuttur. Şekil-43’da açılan ROI seçim ekranı örneği görülmektedir.



Şekil-43 ROI Belirleme Ekranı

ROI seçim ekranı sınırları içerisinde imleç yardımıyla ilgi bölgesi köşe noktaları sol tık ile işaretlenir. Hatalı işaretleme olduğunda sağ tık ile geri alınabilmektedir. İlgi bölgesi köşe noktaları belirlendikten sonra ROI seçim ekranında klavyeden “s” tuşuna basılarak ilgi bölgesi sınırları *Camera #X** adıyla *.config* uzantılı şekilde kaydedilmektedir. Ancak “s” tuşuna basılmadan “Esc” tuşuna basılırsa işlem iptal edilerek çıkış sağlanmaktadır. Şekil-44’de daha önceden kaydedilmiş ve *process* işlemi sırasında otomatik olarak yüklenen config dosyaları mevcuttur.

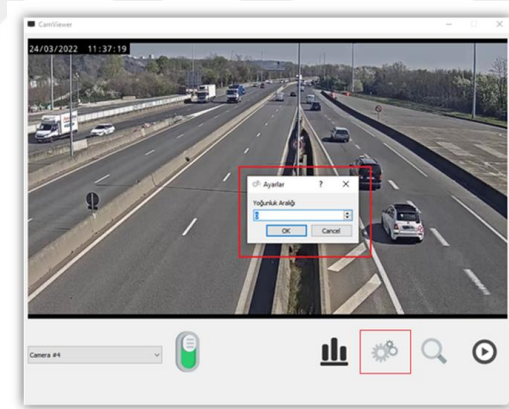
* Kamera numarasına göre değişmektedir.

GUI > data > configs			
Ad	Değiştirme tarihi	Tür	Boyut
Camera #1.config	18.12.2021 17:21	CONFIG Dosyası	1 KB
Camera #2.config	18.12.2021 01:07	CONFIG Dosyası	1 KB
Camera #3.config	6.12.2021 19:43	CONFIG Dosyası	1 KB
Camera #4.config	6.12.2021 20:07	CONFIG Dosyası	1 KB
Camera #5.config	18.12.2021 01:16	CONFIG Dosyası	1 KB
Camera #7.config	18.12.2021 01:13	CONFIG Dosyası	1 KB
Camera #8.config	22.11.2021 18:42	CONFIG Dosyası	1 KB
Camera #9.config	18.12.2021 01:13	CONFIG Dosyası	1 KB
Camera #10.config	11.11.2021 23:17	CONFIG Dosyası	1 KB
Camera #11.config	18.12.2021 01:17	CONFIG Dosyası	1 KB
Camera #12.config	13.12.2021 18:56	CONFIG Dosyası	1 KB
Camera #13.config	13.12.2021 18:55	CONFIG Dosyası	1 KB

Şekil-44 Kaydedilmiş ROI config Dosyaları

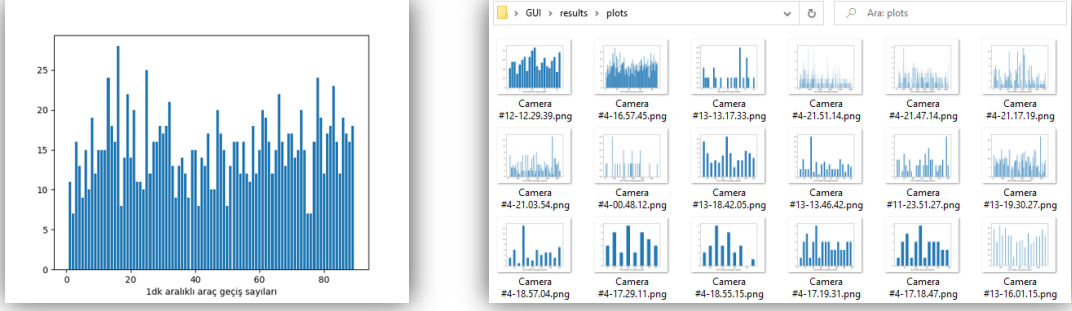
3.2.2 Yoğunluk Zaman Aralığı Belirleme

Araç yoğunluğu zaman aralığı seçimi program arayüzü üzerindeki buton ile yapılabilmektedir. Kamera listesinden kamera seçilmesine gerek kalmadan program başladığı anda aktif olan bu buton ile araç yoğunluğu zaman aralığı her kamera için ayarlanabilmektedir. Arayüz çalıştırıldıktan sonraki her *process* işlemi için yeniden ayarlama yapmaya gerek kalmadan tek ayarlama ile işlem tamamlanmaktadır. Araç yoğunluğu zamanı varsayılan olarak 1 saat olacak şekilde ayarlıdır. Şekil-45’de araç yoğunluğu zaman ayarı ekranı gösterilmiştir.



Şekil-45 Araç Yoğunluğu Zaman Aralığı Ayarı

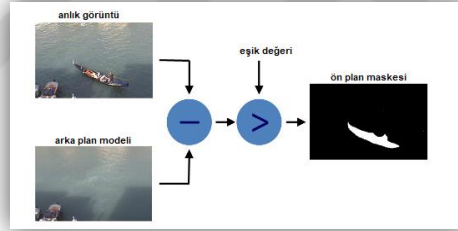
Araç yoğunluğu için zaman ayarı yapıldıktan sonra *process* işlemi başlatılır. *Process* işlemi sırasında klavyeden “q” tuşuna basıldığında anlık araç yoğunluğu grafiği kaydedilip ve ekranda gösterilmektedir. Şekil-46’da yoğunluk grafiklerine örnekler verilmiştir.



Şekil-46 Araç Yoğunluğu Grafiği Örnekleri

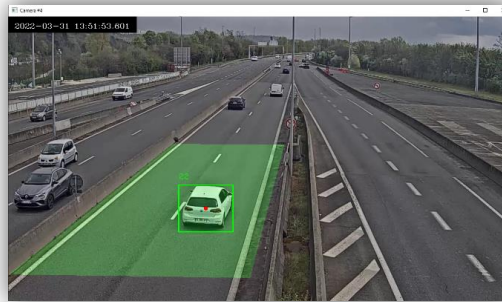
3.2.3 Arka Plan Çıkarımı

Sabit kamera görüntüsündeki hareket eden objelerin tespiti için en çok kullanılan yöntem arka plan çıkarımı işlemidir. Bu işlem ile anlık görüntü ile arka plan modeli arasındaki fark alınır. Elde edilen fark önceden belirlenen bir eşik değerinden fazla olduğu durumlarda tespit edilmek istenilen büyüklükteki objeler ile küçük piksel grupları ayrılmış olur. Şekil-47’de [57] arka plan çıkarımı işlemi görsel olarak da anlatılmaktadır.



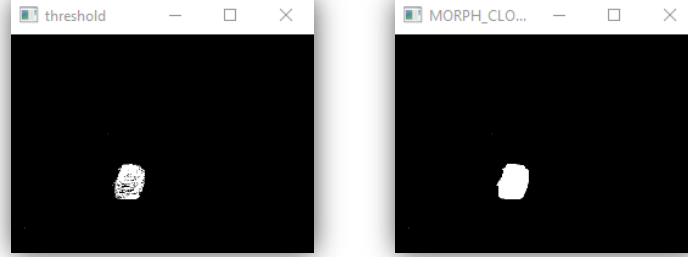
Şekil-47 Arka Plan Çıkarım Yöntemi

Şekil-48’de arka plan çıkarımı işlemleri sonucu tespit edilmiş ve numaralandırılmış araç örnek görseli bulunmaktadır.

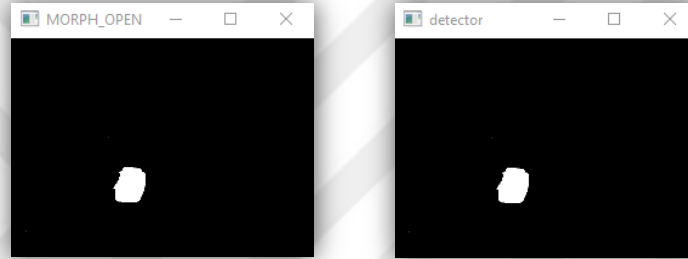


Şekil-48 Tespit Edilmiş ve Numaralandırılmış Araç

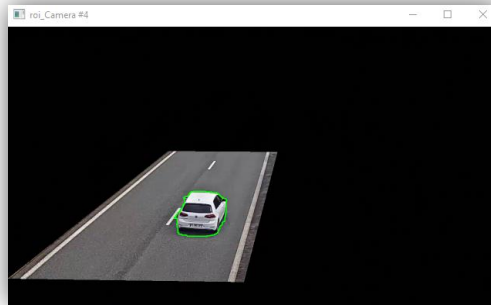
Sırasıyla Şekil-49, Şekil-50, Şekil-51’de arka plan çıkarımı sonrası araç, morfolojik kapama işlemi sonrası araç, morfolojik açma işlemi sonrası araç, arka plan çıkarımı işlemi sonrası araç ve bu işlemler sonrası sınırları ile tespit edilen hareketli araç görselleri bulunmaktadır.



Şekil-49 Arka Plan Çıkarımı ve Morfolojik Kapatma İşlemi

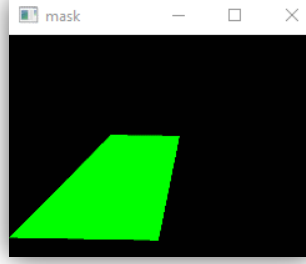


Şekil-50 Morfolojik Açma İşlemi ve Tespit Edilen Hareketli Bölge

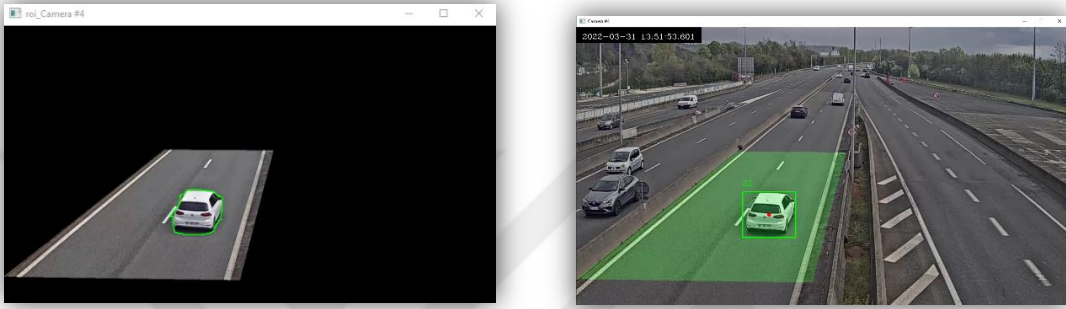


Şekil-51 Tespit Edilen Hareketli Obje Sınırları

Kamera listesinden seçilen kamera görüntüsü, yayın kaynağındaki fps değerine göre kare kare okunarak her kare 1280x720 boyutlarına çevrildikten sonra config dosyası içerisindeki ROI sınırlarına göre ilgi bölgesi kırpılır. İlgi bölgesinin ana görüntüde anlaşılabilmesi için *addWeighted* modülü ile ana görüntüye eklenir. Şekil-52 ve Şekil-53’da yukarıda açıklanan işlemler sonucunda elde edilen görseller bulunmaktadır.



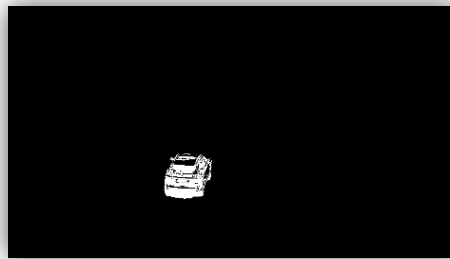
Şekil-52 ROI Sınırları İçerisinde mask Alanı



Şekil-53 ROI alanı ve Ana Görüntüde ROI Sınırları

Kırpılan ilgi bölgesi içerisinde arka plan çıkarım işlemi gerçekleşir. Arka plan çıkarım işlemi ve parametreleri ilgili kamera ya ait hava durumu modeli çıktısına göre dinamik şekilde değişmektedir. Arka plan çıkarımı için openCV modülleri olan `createBackgroundSubtractorMOG2` ve `bgsegm.createBackgroundSubtractorMOG` [57] kullanılmaktadır.

Şekil-54’de ana görüntüden arka plan çıkarımı işlemi sonrası elde edilen görsel bulunmaktadır.



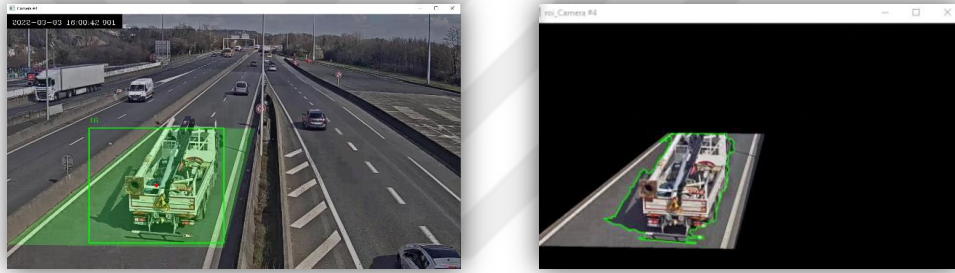
Şekil-54 Arka Plan Çıkarımı

Şekil-55’de aynı anda birden fazla araç tespiti ve numaralandırmasının sorunsuz yapılabildiği görülmektedir.



Şekil-55 Aynı Ekranda Birden Fazla Araç Tespiti

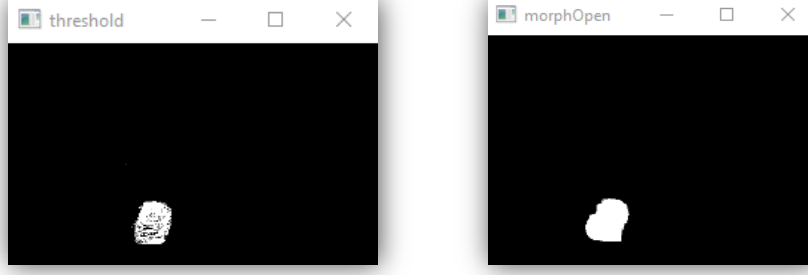
Ancak Şekil-56’da görüldüğü üzere geniş ve uzun araçların ilgi alanından geçişi sırasında bu araçlar tüm ilgi alanını kaplayabilmektedir.



Şekil-56 Büyük Araçların Tespiti

3.2.4 Morfolojik Dönüşümler

Arka planı çıkarılmasından sonra elde edilen görüntüde istenmeyen alanların çıkarılması için morfolojik dönüşümler kullanılmaktadır. Bu dönüşümlerin cinsi ve parametreleri her kamera için hava durumuna göre dinamik şekilde değişmektedir. Morfolojik dönüşümler için openCV modülü olan *morphologyEx* [47] kullanılmıştır. Şekil-57’de arka planı çıkarılmış görüntünün morfolojik dönüşümler öncesi ve sonrası sırasıyla verilmiştir.



Şekil-57 Morfolojik Dönüşümler Öncesi ve Sonrası

3.3 Hız Hesabı ve Hız Sınırı Kontrolü

Hız hesabı yapılırken ROI mesafesi Öklid uzaklığı kullanılarak hesaplanmaktadır. N boyutlu Öklid uzayında noktalar arası Öklid uzaklığı [58] Denklem (15)'de verilmiştir.

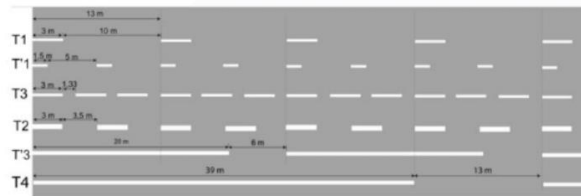
$$d_{öklid} = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \dots + (p_n - q_n)^2} = \sqrt{\sum_{i=1}^n (p_i - q_i)^2} \quad (15)$$

Bu denklemde p_n ve q_n , Öklid uzaklığı hesabı yapılan koordinatların x ve y düzlemdeki noktalarını ifade etmektedir. Tasarlanan sistemde Öklid uzaklığı hesabı, 2 boyutlu düzlem için *Camera #X.config* dosyasından okunan ROI giriş çizgisi koordinatları ve ROI çıkış çizgisi koordinatları için otomatik olarak gerçekleştirilmektedir. Yaklaşık mesafe eşitliği Denklem (16)'da verilmiştir.

$$d_{yaklaşık} = d_{öklid} * l \text{ (km)} \quad (16)$$

Bu denklemde hesaplanan $d_{öklid}$ Öklid uzaklığı, l katsayı ile çarpılarak km cinsinden gerçek değerlere yaklaşımları sağlanmıştır.

Fransa'dan çevrimiçi erişilen kamerada ilgi alanı mesafesi otomatik hesaplanırken yol üzerinde bulunan şerit çizgileri ve ölçülerinden faydalanılmıştır. Bunun için [59] çalışmasında belirtilen ve Şekil-58'de gösterilen T'1 işaretleme yöntemi temel alınmıştır.



Şekil-58 Fransa'da Kullanılan Trafik Çizgileri ve Özellikleri

Ayrıca Şekil-59'de verildiği üzere, İstanbul kameraları olan Camera #12 ve Camera #13 için Karayolları Genel Müdürlüğü (KGM) tarafından yayınlanan kılavuza [60] göre ilgili

kameradaki yol üzerinde bulunan şerit çizgisi ve aralıkları 3'er metre olacak şekilde sistem kalibre edilmiştir.

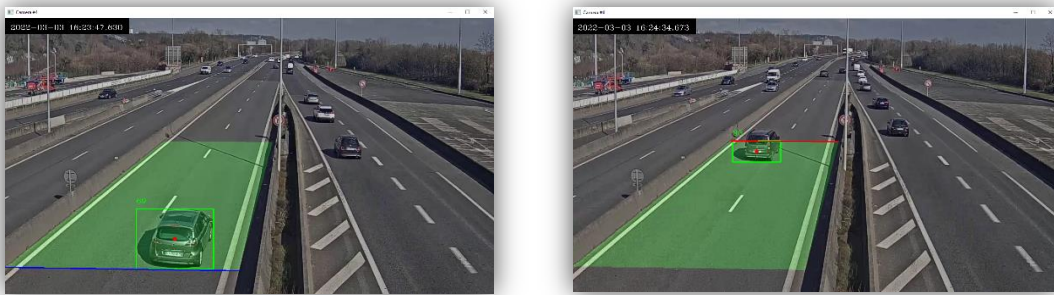
YERLEŞİM YERİ İÇİ	Devamlı Çizgi	0.10		Bisiklet yolu sınır çizgilerinde
		0.25		Taahhüt yolundaki bisiklet yolu ayırıcı çizgilerinde
	Kesikli Çizgi	0.10		Bisiklet yolu şerit çizgilerinde
				Bisiklet yolu garaj ve bahçe giriş çıkışı kesimlerinde
				Şerit çizgilerinde
		0.14		Kavşak kesimlerinde (Kenar çizgisinin devamında)
				Sinyalize kavşaklarda yaya geçidi sınır çizgilerinde
		0.25		Sağa ve sola dönüş şeritlerinin ayrımında
				Ana yoldan ayrıma yada katılmalarda
		0.50		Bisiklet yolunun ana yola kesiştiği kesimlerde

Şekil-59 Türkiye’de Yerleşim İçi Trafik Çizgileri ve Özellikleri

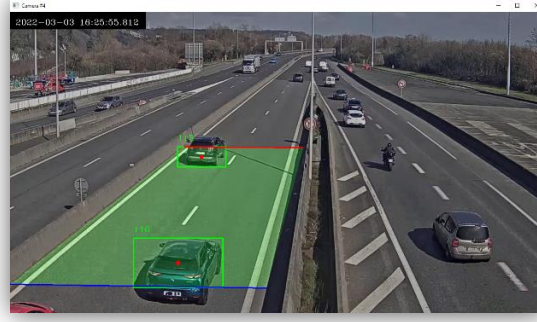
Her araç numarası için, aynı anda birden fazla aracın ROI sınırları içinde olması sorun teşkil etmeyecek şekilde, ROI sınırları içerisine giriş zamanı *Enter Time* ve çıkış zamanı *Exit Time* bilgileri gerçek zaman olarak kaydedilmektedir. Çıkış zamanı ve giriş zamanı arasındaki fark ile ilgili aracın ROI sınırları içerisinde geçirdiği süre Denklem (17)’de verildiği gibi hesaplanabilmektedir.

$$t_{ROI} = t_{ExitTime} - t_{EnterTime} (sn) \quad (17)$$

Burada $t_{ExitTime}$ aracın ilgi bölgesinden çıkış anını, $t_{EnterTime}$ ise aracın ilgi bölgesine giriş anını temsil etmektedir. Şekil-60 ve Şekil-61’de araçların ROI alanına giriş çıkış anlarının görselleri mevcuttur.



Şekil-60 Araç ROI Giriş ve Çıkış Anları



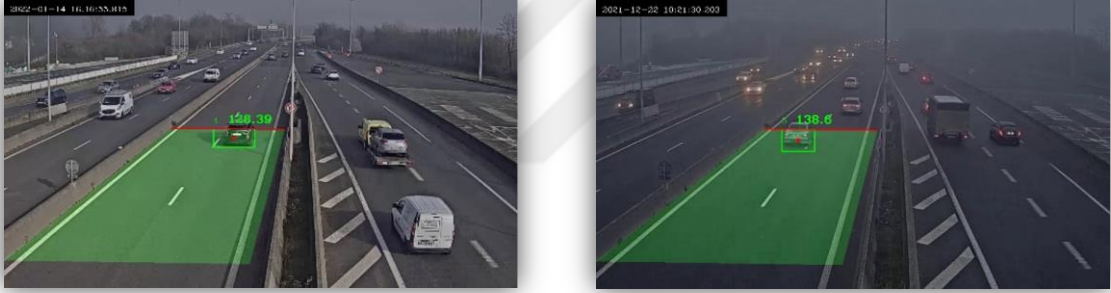
Şekil-61 Aynı Anda Birden Fazla Araç İçin ROI Giriş ve Çıkış Anları

Farklı numaralar ile numaralandırılan her araç için hız tahminleri $d_{yaklaşık}$ ve t_{ROI} değerleri kullanılarak Denklem (18)'de [58] verildiği gibi hesaplanmaktadır. k değeri kameraya göre değişen hız kalibrasyonu sabitidir.

$$v_{yaklaşık} = \frac{d_{yaklaşık}}{t_{ROI}} * 3.6 * k \text{ (km/h)} \quad (18)$$

Hesaplanan $v_{yaklaşık}$ değeri CNN modelinin periyotlarla tahmin ettiği hava durumuna göre güncellenen hız sınırını aştığı durumda, aşım yapan numaralı aracın ROI sınırlarından çıkış anı $v_{yaklaşık}$ değerini de içerecek şekilde anlık görüntü alınarak kaydedilmektedir.

Aşağıdaki Şekil-62'de çeşitli hava durumlarında yapılan hız sınırı aşımaları sırasında alınan anlık görüntüler bulunmaktadır. Ayrıca Şekil-63'da işlem sonucunda oluşan csv veri tabanlarına örnek bulunmaktadır.



Şekil-62 Normal ve Sisli Havada Hız Aşım Anları Örneği

	Enter Time	Exit Time	Speed
1	2022-03-03 15:59:35.287960+03:00	2022-03-03 15:59:36.419264+03:00	77.32
2	2022-03-03 15:59:41.521123+03:00	2022-03-03 15:59:42.980225+03:00	57.97
3	2022-03-03 15:59:49.566171+03:00	2022-03-03 15:59:51.138204+03:00	60.72
4	2022-03-03 15:59:54.171337+03:00	2022-03-03 15:59:55.352609+03:00	79.9
5	2022-03-03 16:00:13.211616+03:00	2022-03-03 16:00:14.900577+03:00	53.9
6	2022-03-03 16:00:17.018273+03:00	2022-03-03 16:00:18.478375+03:00	60.26
7	2022-03-03 16:00:20.359216+03:00	2022-03-03 16:00:21.485526+03:00	80.88
8	2022-03-03 16:00:21.289644+03:00	2022-03-03 16:00:22.829696+03:00	53
9	2022-03-03 16:00:24.558636+03:00	2022-03-03 16:00:25.735906+03:00	77.48

Şekil-63 Araç Giriş-Çıkış ve Hız Veri Tabanı Örneği

4 BULGULAR

Bu bölümde, tasarlanan sistemin sonuçları ve bu sonuçların değerlendirmeleri aşağıdaki iki başlık altında verilmiştir.

4.1 Gerçekleştirilen Sistem

Hava durumuna göre değişen hız sınırı uygulamasında, herhangi bir çevrim içi kameradan alınan görüntü iki senkronize işleme tabii tutulmaktadır.

İlk işlem olan hava durumunun tespiti için daha önce ilgili kameradan elde edilen çeşitli mevsimsel şartları içeren görsellerden CNN hava durumu modeli oluşturulmuştur. Oluşturulan ve sistemde kullanılan modellerin başarımları oranları %99.51 - %99.85 arasındadır. Bu modellere göre ilgili kameradan alınan anlık görüntüler kullanılarak belirli periyotlarla hava durumu tahmini yapılmaktadır. Bu tahmin sonuçlarına göre hız limiti ve görüntü işleme parametreleri otomatik olarak değiştirilmektedir.

İkinci işlem ise görüntü işleme ile hız ölçümlerini kapsamaktadır. Bu işlem için ilk olarak her kamera için ayrı ayrı belirlenen ilgi alanlarından işlem yapılan ilgili kamera için yaklaşık mesafe hesaplanmaktadır. Sonrasında yukarıdaki bölümlerde ayrıntılı olarak açıklandığı şekliyle görüntü işleme yöntemleri kullanılarak hareketli araç ve hız tespitleri gerçekleştirilmektedir. Oluşturulan sistemin araç tespiti, sayımı ve hızının hesaplamasında başarımları oranları %76.46 - %94.67 arasındadır. Araç tespiti ile birlikte araç sayımı ve belirli periyotlarla araç yoğunluğu bilgileri elde edilebilmektedir. İşlem sonucunda her aracın giriş çıkış zaman bilgilerinin ve hesaplanan hız değerlerini içeren csv dosyası oluşturulmaktadır. Ayrıca hava durumuna göre değişen hız sınırlarının aşımı durumunda ilgili aracın anlık görüntüsü belirlenen dizine kaydedilmektedir.

Şekil-64’de kamera görüntü kalitesi çok kötü olan herhangi bir kameradan tasarlanan sistemin araç tespiti ve sayımı için çalışabileceği görülmektedir, ilgili kamera görüntülerinde hava durumu tespiti için gerekli ekran kalitesi bulunmamaktadır.



Şekil-64 Düşük Görüntü Kalitesi Olan Kameralar İle Araç Tespiti

Tablo-11’de sistemde kullanılması planlanan kameraların listesi bulunmaktadır. 6 aylık geliştirme süresi boyunca görülen erişim problemleri, kameranın fiziksel stabilizasyonu, görüntü kalitesi ve yağış altında görüntü açısının kapanması gibi problemlerden ötürü 3 adet kamera ile sistem geliştirmesine devam edilmiştir. Bu kameraların isimleri aşağıdaki gibidir.

- Camera #4
- Camera #12
- Camera #13

Tablo-11 Kamera Listesi

Kamera İsmi	Kamera Konumu	Çözünürlük	fps	Erişim Adresi	Kullanım Durumu
Camera #1	Apex Twon/Kuzey Carolina	1280x720	30	https://www.youtube.com/watch?v=vWaFYoZ5qyM	X
Camera #2	Fort Erie/Kanada	1280x720	30	https://www.youtube.com/watch?v=XoNZZJyRpUc	X
Camera #3	Lyon/Fransa	1280x720	60	https://www.youtube.com/watch?v=EBhCrTPdBI	X
Camera #4	Lyon/Fransa	1280x720	60	https://www.youtube.com/watch?v=yIoY-edup8w	✓
Camera #5	İstanbul/Türkiye	640x352	20	https://uym.ibb.gov.tr/yharita6/	X
Camera #6	İstanbul/Türkiye	640x352	20	https://uym.ibb.gov.tr/yharita6/	X
Camera #7	İstanbul/Türkiye	640x352	20	https://uym.ibb.gov.tr/yharita6/	X
Camera #8	İstanbul/Türkiye	640x352	20	https://uym.ibb.gov.tr/yharita6/	X
Camera #9	İstanbul/Türkiye	640x352	20	https://uym.ibb.gov.tr/yharita6/	X
Camera #10	İstanbul/Türkiye	640x352	20	https://uym.ibb.gov.tr/yharita6/	X
Camera #11	İstanbul/Türkiye	640x352	12	https://uym.ibb.gov.tr/yharita6/	X
Camera #12	İstanbul/Türkiye	1920x1080	30	https://uym.ibb.gov.tr/yharita6/	✓
Camera #13	İstanbul/Türkiye	1920x1080	30	https://uym.ibb.gov.tr/yharita6/	✓

4.2 Sonuçların Değerlendirilmesi

Her kamera için ayrı oluşturulan hava durumu modelleri ile gerçekleştirilen işlemler sonucunda günün belirli saatlerinde araç sayımı ve hava durumu tespiti sonuçları aşağıdaki Tablo-12’de verildiği şekilde gözlemlenmiştir.

Tablo-12 Çeşitli Hava Şartlarında Gerçekleştirilen Sayımlar

Kamera	Hava Durumu	Ölçüm Zaman Aralığı	Toplam Araç Sayımı	Hatalı Araç Sayımı	Araç Tespiti Başarım Onayı
Camera #4	Normal	21:05 – 21:47	558	93	%83.33
Camera #4	Normal	20:10 – 21:03	1000	94	%90.60
Camera #4	Normal	21:36 – 22:00	113	19	%83.19
Camera #4	Normal	15:28 – 16:57	1333	181	%86.42
Camera #4	Yağmur	21:05 – 21:10	66	15	%77.27
Camera #4	Yağmur	07:10 – 07:15	207	45	%78.26
Camera #12	Normal	19:43 – 19:58	450	24	%94.67
Camera #12	Yağmur	12:08 – 12:29	581	35	%93.98
Camera #12	Yağmur	11:52 – 11:58	165	33	%80.00
Camera #13	Normal	12:02 – 12:44	2256	531	%76.46
Camera #13	Normal	20:21 – 20:27	254	38	%85.04
Camera #13	Normal	13:10 – 13:16	298	63	%78.86
Camera #13	Normal	13:36 – 13:44	400	74	%81.50
Camera #13	Yağmur	13:13 – 13:23	196	41	%79.08
Camera #13	Yağmur	13:23 – 13:29	243	41	%83.13

Tablo-12’de görülebildiği üzere trafiğin yoğun olduğu saatlerde dahi sistem araç tespitinde tatmin edici başarımlar göstermektedir. Örneğin Camera #13 ile yapılan, İstanbul’da günün en yoğun saatlerinden biri olan öğle vaktinde yaklaşık 42 dakikalık canlı gözlemde, saniyede yaklaşık 0.9 araç ile 2256 araç sayımı gerçekleştirilmiştir. Sistem bu gözlemde her aracı eksiksiz tespit etmesine karşın bazı (uzun ve geniş) araçların geçişi sırasında araç numarasını hatalı şekilde arttırdığından ya da canlı kamera görüntüsündeki atlamalardan kaynaklı hatalar ile birlikte dahi en az %76.46 başarımlar göstermiştir. Keza Camera #4 ile yağmurlu havada gerçekleştirilen 5 dakikalık gözlemde, yoğun trafikte dahi başarımlar oranı %78.26 ile 207 adet araç sayımı gerçekleştirilmiştir.

Aşağıdaki Tablo-13’de tasarlanan uygulamanın benzer çalışmalar ile karşılaştırılması mevcuttur.

Tablo-13 Tasarlanan Uygulamanın Benzer Çalışmalar İle Karşılaştırılması

	Araç Tespiti ve Sayımı - Hız Kestirimi	Hava Durumu Tespiti – Görsel Sınıflandırma
Partha C. ve arkadaşları	95.08 - 94.1	-
Debojit B. ve arkadaşları	96.55	-
Jun-Wei H. ve arkadaşları	92.8 - 94.8	-
Surendra G. ve arkadaşları	90	-
Van Huy P. ve arkadaşları	98*	-
Prem B. ve arkadaşları	91.26	-
Nastaran E. ve arkadaşları	94.02	-
Hilal T. ve arkadaşları	90.51	-
Danang W. ve arkadaşları	87.01	-
Wei-Ta C. ve arkadaşları	-	76.6
Martin R. ve arkadaşları	-	85.19
Zheng Z. ve arkadaşları	-	71.39
Boukaye T. ve arkadaşları	-	94
Xingcheng L. ve arkadaşları	-	92.25
Linkai C. ve arkadaşları	-	75.7
Muhammed Taha Z. tarafından	-	99.52
Gongfa L. ve arkadaşları	-	98.52
Uday S. ve arkadaşları	-	97.13
Hedeflenen Çalışma	76.46 - 94.67	99.51 – 99.85*

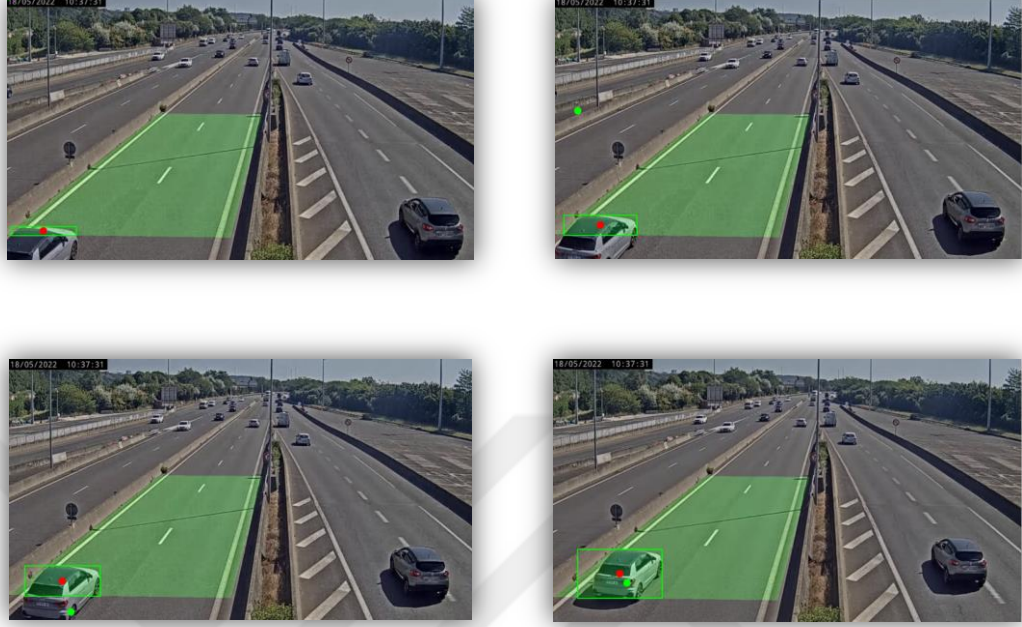
En iyi performanslar kalın yazılmış ve “” ile işaretlenmiştir*

Yapılan çalışmanın çıktılarından biri olan hız tahmininin doğruluğunu gösterebilmek adına 2 farklı yönteme başvurulmuştur. Birinci yöntem olarak her kameradaki hız ölçümlerinde araç geçiş parametrelerinin Kalman filtresi [61] ile tahmin edilerek doğrulanması ve ardından hız hesabının sağlanmasının yapılmasıdır. İkinci yöntem ise kamera görüntüsünün bulunduğu ilgili yoldan test aracı ile geçiş yapılması ve gerçek hız değeri ile uygulama tarafından hesaplanan hız değerinin karşılaştırılmasıdır. Bu bağlamda hız tahmini sağlanması için yapılan her iki yöntem sonuçları aşağıdaki gibidir.

- **Kalman Filtresi ile Doğrulama**

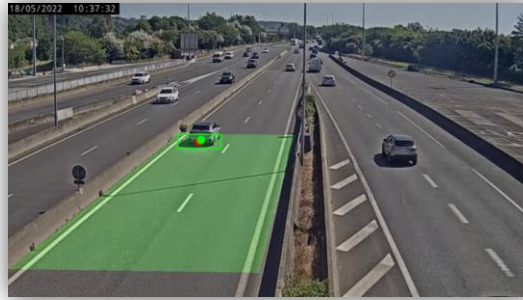
Tasarlanan sistem ile yapılan hız tahmininin doğrulanması adına her kare için, bir önceki karedeki araç merkez noktaları arasındaki birim mesafe farkı hesaplanmaktadır. Aynı hesap kalman filtresinin Denklem (11) kullanılarak tespit edilen aracın bir sonraki karedeki merkez noktası tahminleri için de gerçekleştirilmiştir. Elde edilen iki kare arasındaki mesafenin iki kare arasındaki zaman farklarına bölünmesiyle, her kare için ayrı ayrı hızlar hesaplanmıştır. Kalman filtresi tahminlerine (0,0) noktasından başlayıp bir sonraki karedeki araç merkez noktasına göre tahminlerine devam etmektedir. Aşağıdaki Şekil-65’deki kırmızı noktalar

aracın tespit edilen merkez noktasını gösterirken yeşil noktalar ise Kalman filtresi tahminini göstermektedir. Görülebildiği üzere ilk birkaç kareden sonra Kalman filtresi merkez noktaya yakın tahminlerde bulunmuştur.



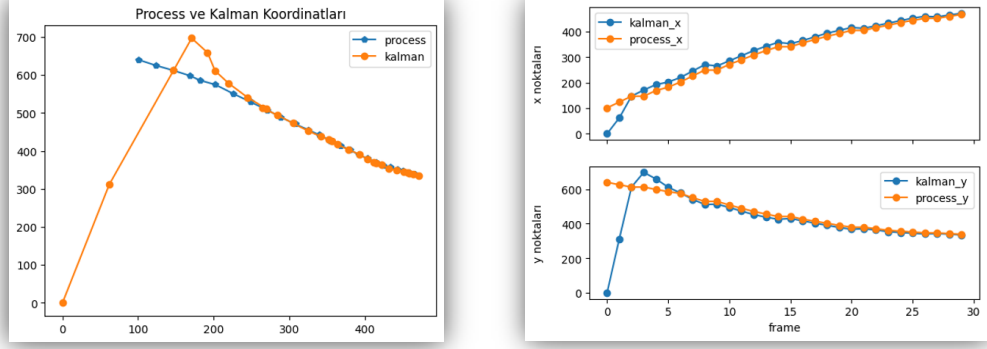
Şekil-65 Kalman Filtresi İle Araç Merkez Noktası Tahminleri

Son karedeki araç merkez noktası *process* ve Kalman tahminleri durumları Şekil-66’da verilmiştir.



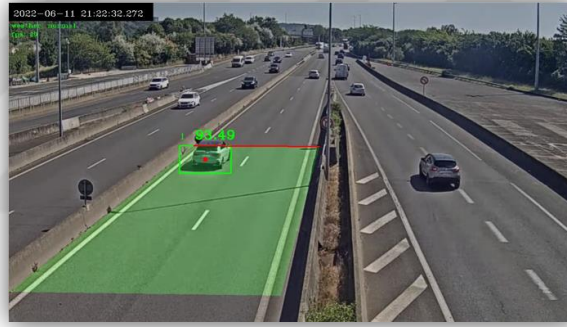
Şekil-66 Son Karedeki Kalman Filtresi Tahmini

Şekil-67’de ise her kare için, araç merkez noktasının ve Kalman filtresi ile tahmin edilen merkez noktanın x ve y koordinatlarındaki değişimleri grafik üzerinden gösterilmiştir.

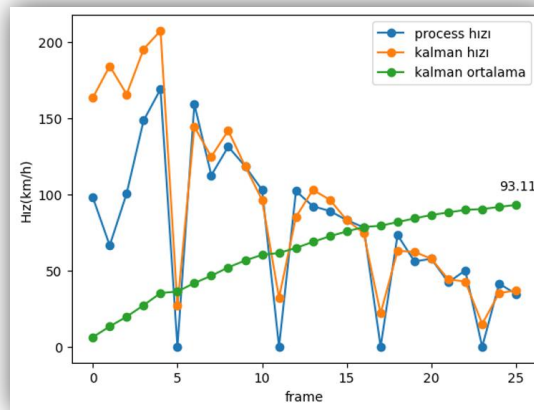


Şekil-67 Camera #4 Kalman Tahmininin x-y Düzlemindeki Koordinatları

Şekil-68 ve Şekil-69’de *process* işlemi sonrasında elde edilen hız tahmini ile Kalman filtresi ile elde edilen her karedeki hız değerlerinin ortalaması bulunmaktadır. *Process* işlemi sonrasında ilgili araç için hız tahmini 93.49 km/h iken kalman filtresi ile ilgili aracın ortalama hızı 93.11 km/h olarak hesaplanmıştır.



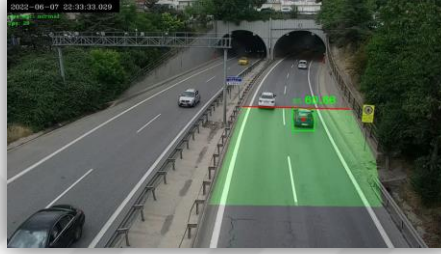
Şekil-68 Uygulama Hız Tahmini Sonucu



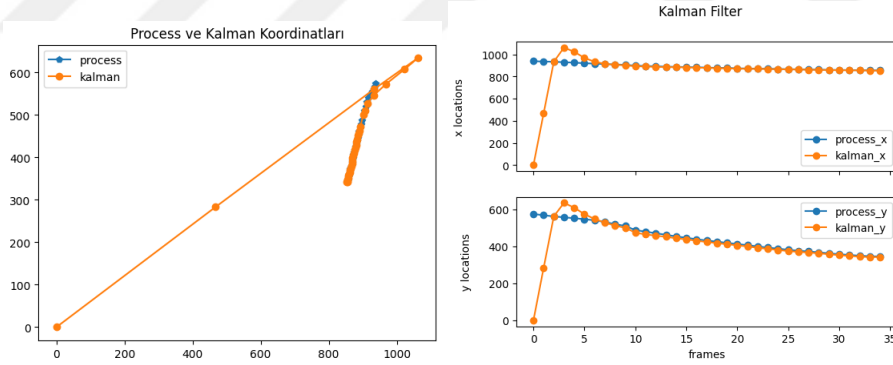
Şekil-69 Camera #4 Kalman Hız Hesabı Sonucu

- **Test Aracı ile Doğrulama**

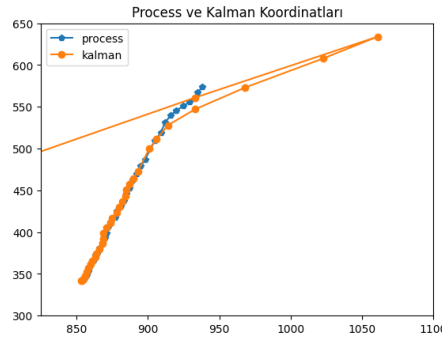
Planlamalar sonucunda Camera #13'ün canlı yayın yaptığı yol üzerinden sabit 60 km/h hızla geçiş gerçekleştirilmiştir. Bu geçiş sırasındaki canlı video, fps değeri değişmeyecek şekilde kaydedilmiş ve tasarlanan uygulamada *process* işleminin video kaynağı olarak kullanılmıştır. Bu bağlamda ilgili kamera için hız hesabı kalibrasyonu sağlanmıştır. Yapılan bu kalibrasyon sonrasında ilgili kameralardan tekrar geçişler gerçekleştirilip canlı ortamda hız tahmini yapılmıştır. Bu tahminlerin gerçek değerlere yakın ve kalibrasyona uygun olduğu belirlenmiştir. Şekil-70, Şekil-71 ve Şekil-72'de test aracı ile geçiş sırasında *process* işlemi sonucunda elde edilen hız değerinin 60.68 km/h olduğu görülmektedir.



Şekil-70 Test Aracı ile Uygulama Hız Tahmini Sonucu

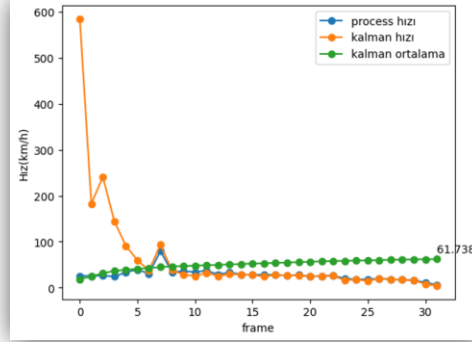


Şekil-71 Camera #13 Kalman Tahmininin x-y Düzlemindeki Koordinatları



Şekil-72 Camera #13 Kalman Tahmininin x-y Düzlemindeki Koordinatları (Yaklaştırılmış)

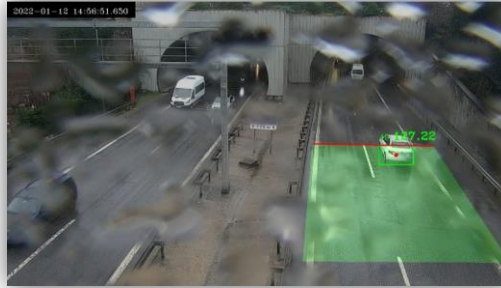
Şekil-73’da test aracı ile geçiş anında kalman filtresi tahminlerinden elde edilen ortalama hız değeri bulunmaktadır. *Process* işlemi sonucunda hesaplanan hız değeri 60.68 km/h iken aynı araç için kalman ortalama hızı 61.74 km/h olmuştur.



Şekil-73 Test Aracı İle Kalman Hız Hesabı Sonucu

4.2.1 Farklı Hava Durumlarının Sonuçlara Etkisi

Tablo-12’de görüldüğü üzere yağmurlu, karlı ve sisli hava şartlarının araç sayımında ve hız hesabında kullanılan görüntü işleme adımlarını olumsuz etkilediği belirlenmiştir. Şekil-74’de görüldüğü gibi yoğun yağış altında araç ve doğru hız tespiti zorlaşmaktadır.



Şekil-74 Yoğun Yağış Altında Hatalı Araç ve Hız Tespiti

4.2.2 Farklı Derin Öğrenme Parametrelerinin Sonuçlara Etkisi

Tablo-14’de;

- 2 evrişim katmanına,
- [8-16-32] nörona,
- [(3,3) - (5,5)] ortaklama katmanlarına,
- [8-16-32] grup boyutlarına,
- [8-16-32] dense değerlerine sahip,

7 tur öğrenime sokulmuş toplam 54 adet CNN modeli başarımları ve 10’ar adet normal, karlı, yağmurlu ve sisli hava durumlarını içeren görsellere oluşturdukları tahminleri bulunmaktadır.

Tablo-14 Farklı CNN Modelleri ve Tahmin Sonuçları

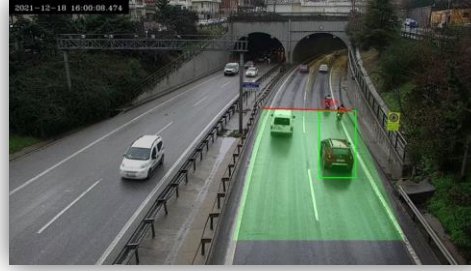
		Val_Accuracy	kar	normal	yağmur	sis
1	2cov_8nodes_8batch_0.3valid_7epochs_8dense_(3, 3)size.model	0.5094	0	0	40	0
2	2cov_8nodes_8batch_0.3valid_7epochs_8dense_(5, 5)size.model	0.9929	10	10	10	10
3	2cov_8nodes_8batch_0.3valid_7epochs_16dense_(3, 3)size.model	0.9976	10	10	10	10
4	2cov_8nodes_8batch_0.3valid_7epochs_16dense_(5, 5)size.model	0.9882	10	10	10	10
5	2cov_8nodes_8batch_0.3valid_7epochs_32dense_(3, 3)size.model	0.9976	10	10	10	10
6	2cov_8nodes_8batch_0.3valid_7epochs_32dense_(5, 5)size.model	0.9882	10	10	10	10
7	2cov_8nodes_16batch_0.3valid_7epochs_8dense_(3, 3)size.model	0.9764	0	10	20	10
8	2cov_8nodes_16batch_0.3valid_7epochs_8dense_(5, 5)size.model	0.7429	0	30	10	0
9	2cov_8nodes_16batch_0.3valid_7epochs_16dense_(3, 3)size.model	0.9976	10	10	10	10
10	2cov_8nodes_16batch_0.3valid_7epochs_16dense_(5, 5)size.model	0.9929	10	10	10	10
11	2cov_8nodes_16batch_0.3valid_7epochs_32dense_(3, 3)size.model	0.9953	10	10	10	10
12	2cov_8nodes_16batch_0.3valid_7epochs_32dense_(5, 5)size.model	0.9976	10	10	10	10
13	2cov_8nodes_32batch_0.3valid_7epochs_8dense_(3, 3)size.model	0.9976	10	10	10	10
14	2cov_8nodes_32batch_0.3valid_7epochs_8dense_(5, 5)size.model	0.9764	0	19	11	10
15	2cov_8nodes_32batch_0.3valid_7epochs_16dense_(3, 3)size.model	0.9976	10	10	10	10
16	2cov_8nodes_32batch_0.3valid_7epochs_16dense_(5, 5)size.model	0.9976	10	10	10	10
17	2cov_8nodes_32batch_0.3valid_7epochs_32dense_(3, 3)size.model	0.9906	10	10	10	10
18	2cov_8nodes_32batch_0.3valid_7epochs_32dense_(5, 5)size.model	0.9976	10	10	10	10
19	2cov_32nodes_8batch_0.3valid_7epochs_8dense_(3, 3)size.model	0.9976	10	10	10	10
20	2cov_32nodes_8batch_0.3valid_7epochs_8dense_(5, 5)size.model	0.9976	10	10	10	10
21	2cov_32nodes_8batch_0.3valid_7epochs_16dense_(3, 3)size.model	0.9906	10	10	10	10
22	2cov_32nodes_8batch_0.3valid_7epochs_16dense_(5, 5)size.model	0.9953	10	10	10	10
23	2cov_32nodes_8batch_0.3valid_7epochs_32dense_(3, 3)size.model	0.9976	10	10	10	10

24	2cov_32nodes_8batch_0.3valid_7epochs_32dense_(5, 5)size.model	0.9976	10	10	10	10
25	2cov_32nodes_16batch_0.3valid_7epochs_8dense_(3, 3)size.model	0.9788	10	10	10	10
26	2cov_32nodes_16batch_0.3valid_7epochs_8dense_(5, 5)size.model	0.9717	0	10	14	16
27	2cov_32nodes_16batch_0.3valid_7epochs_16dense_(3, 3)size.model	0.9976	10	10	10	10
28	2cov_32nodes_16batch_0.3valid_7epochs_16dense_(5, 5)size.model	0.9929	10	10	10	10
29	2cov_32nodes_16batch_0.3valid_7epochs_32dense_(3, 3)size.model	0.9953	10	10	10	10
30	2cov_32nodes_16batch_0.3valid_7epochs_32dense_(5, 5)size.model	0.9953	10	10	10	10
31	2cov_32nodes_32batch_0.3valid_7epochs_8dense_(3, 3)size.model	0.9976	10	10	10	10
32	2cov_32nodes_32batch_0.3valid_7epochs_8dense_(5, 5)size.model	0.7712	8	14	18	0
33	2cov_32nodes_32batch_0.3valid_7epochs_16dense_(3, 3)size.model	0.9976	10	10	10	10
34	2cov_32nodes_32batch_0.3valid_7epochs_16dense_(5, 5)size.model	0.9976	10	10	10	10
35	2cov_32nodes_32batch_0.3valid_7epochs_32dense_(3, 3)size.model	0.9741	10	10	10	10
36	2cov_32nodes_32batch_0.3valid_7epochs_32dense_(5, 5)size.model	0.9976	10	10	10	10
37	2cov_64nodes_8batch_0.3valid_7epochs_8dense_(3, 3)size.model	0.5094	0	0	40	0
38	2cov_64nodes_8batch_0.3valid_7epochs_8dense_(5, 5)size.model	0.7594	10	10	10	10
39	2cov_64nodes_8batch_0.3valid_7epochs_16dense_(3, 3)size.model	0.9929	10	10	10	10
40	2cov_64nodes_8batch_0.3valid_7epochs_16dense_(5, 5)size.model	0.9646	10	10	10	10
41	2cov_64nodes_8batch_0.3valid_7epochs_32dense_(3, 3)size.model	0.9741	10	10	10	10
42	2cov_64nodes_8batch_0.3valid_7epochs_32dense_(5, 5)size.model	0.9976	10	10	10	10
43	2cov_64nodes_16batch_0.3valid_7epochs_8dense_(3, 3)size.model	0.9976	10	10	10	10
44	2cov_64nodes_16batch_0.3valid_7epochs_8dense_(5, 5)size.model	0.7500	0	10	30	0
45	2cov_64nodes_16batch_0.3valid_7epochs_16dense_(3, 3)size.model	0.9882	10	10	10	10
46	2cov_64nodes_16batch_0.3valid_7epochs_16dense_(5, 5)size.model	0.9788	0	10	20	10
47	2cov_64nodes_16batch_0.3valid_7epochs_32dense_(3, 3)size.model	0.9835	10	13	10	7
48	2cov_64nodes_16batch_0.3valid_7epochs_32dense_(5, 5)size.model	0.9976	10	10	10	10
49	2cov_64nodes_32batch_0.3valid_7epochs_8dense_(3, 3)size.model	0.9788	0	10	10	20
50	2cov_64nodes_32batch_0.3valid_7epochs_8dense_(5, 5)size.model	0.5094	0	0	40	0
51	2cov_64nodes_32batch_0.3valid_7epochs_16dense_(3, 3)size.model	0.9953	10	10	10	10
52	2cov_64nodes_32batch_0.3valid_7epochs_16dense_(5, 5)size.model	0.9788	0	10	20	10
53	2cov_64nodes_32batch_0.3valid_7epochs_32dense_(3, 3)size.model	0.9953	10	10	10	10
54	2cov_64nodes_32batch_0.3valid_7epochs_32dense_(5, 5)size.model	0.9976	10	10	10	10

4.2.3 Farklı Görüntü İşleme Parametrelerinin Sonuçlara Etkisi

Şekil-75’de motosiklet ve yağmurlu zemindeki silueti hacim olarak küçük olduğundan araç olarak algılanmayıp görüntü işleme parametresinde diğer aracın bir parçası olarak algılanmaktadır. Bu durumda ilgili aracın hız hesabı hatalı olacaktır. Bunun önüne

geçebilmek için morfolojik açma işleminde kullanılan yapılandırma elemanının ikinci değeri ilgili kameranın yağmurlu hava şartları için arttırılmıştır.



Şekil-75 Hatalı Görüntü İşleme Parametreleri Sonucu-1

Şekil-76'da görüldüğü gibi morfolojik dönüşüm ve arka plan çıkarımı parametreleri uygun ayarlanmadığı durumda, özellikle akşam saatlerinde çevredeki ışıklar asfalt yol üzerinde yansımalara sebep olmaktadır. Bu durumda görüntü işleme algoritmamız ilgili ışık hareketlerini araç olarak algılayabilmektedir. Bu durumun önüne geçilmesi için her kameraya uygun parametreler girilmesi gerekmektedir.



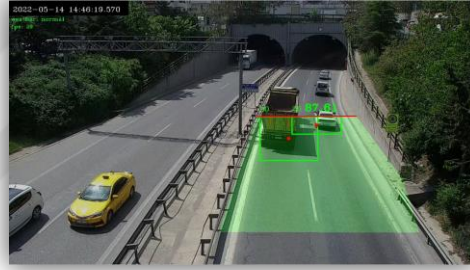
Şekil-76 Hatalı Görüntü İşleme Parametreleri Sonucu-2

Şekil-77'de büyük araçların geçişleri sırasında morfolojik dönüşüm parametrelerinin uygun seçilmemesi sonucunda oluşan duruma örnek gösterilmektedir. Morfolojik kapatma işlemi sırasında kullanılan yapılandırma elemanının ilk değeri artırılarak ilgili durumun önüne geçilmiş ve tespit edilen bütünleşik bölgenin orta noktasının kırmızı sınır çizgisinden çıkması sonrasında daha doğru bir hız hesabı elde edilebilmiştir.



Şekil-77 Hatalı Görüntü İşleme Parametreleri Sonucu-3

Şekil-78’de takip algoritmasında kullanılan minimum değer değişkeninin kamera açısı ve büyük araç geçişlerine uygun girilmediği durumda, birbirine yakın geçen araçların aynı numara ile numaralandırması sonucu araç sayısında ve dolayısıyla araç hız hesabında karşılaşılan hataya örnek gösterilmektedir. Bu durumun önüne her kameraya özel minimum takip değeri girilerek geçilmiştir.



Şekil-78 Hatalı Görüntü İşleme Parametreleri Sonucu-4

5 TARTIŞMA ve SONUÇ

Bu çalışmada zamanla gelişen teknoloji ile sayısı ve kullanım alanları hızla artan elektronik denetleme sistemlerinin, kaza riskini oldukça artıran hava durumu parametresine uygun geliştirilebileceği gösterilmiştir. Hali hazırda internet ortamından erişilebilen herhangi bir kameradan alınan görüntülerden oluşturulan derin öğrenme modelleri ile bu işlemin ek maliyet çıkarmadan yapılabileceği ortaya konulmuştur. Şehirlerarası ya da şehir içi trafiğinde fark etmeksizin kullanılabilecek bu çalışma ile kaza riskinin azaltılmasında elektronik denetleme caydırıcılığı artırılabilir.

Geliştirilen uygulama ile hava durumu tespiti için otomatik veri toplanması sağlanmış, toplanan verilerin %70'i ile yüksek başarımlı CNN tahmin modelleri oluşturulmuştur. Oluşturulan CNN modellerine ait başarımların ölçülmesi için toplanan verilerin kalan %30'luk kısmı kullanılmıştır. Tasarlanan uygulama ile hava durumu tespiti sonrası otomatik hız sınırları belirlenmiş ve bu sınırlara uygun şekilde denetim gerçekleştirilmiştir. Araç tespiti, araçların eşsiz numaralandırılması ve araç hızının kestirimi yine bu uygulama ile görüntü işleme yöntemleri kullanılarak gerçekleştirilmiştir, hız sınırı aşımaları durumunda ise ilgili araçların anlık görüntüleri belirlenen dizine kaydedilmiştir. Uygulama çıktılarından, her araç için özgün olan hız kestirimlerinin iki farklı yöntem ile sağlanması yapılmıştır. Birinci yöntem olarak Kalman filtresi kullanılmış ve ortalama Kalman hızı ile tespit edilen hız kestirimi karşılaştırılmıştır. İkinci yöntem içinse test aracı ile mevcut kameradan sabit hızlarla geçişler gerçekleştirilmiştir. Bu geçişlere ait uygulama hız kestirimleri sabit geçiş hızları ile karşılaştırılmıştır.

Geliştirilen uygulamanın ek olarak sağladığı araç sayısı ve gün içi trafik yoğunlukları verileri, sonraki çalışmalarda ya da ilgili kameranin bulunacağı yol hakkındaki ulaştırma planlarında kullanılabilecektir. Trafik akışı planlaması, trafik işaret ve işaretçileri planlaması, yol hız sınırları güncellemesi, yol bakım/onarım ve geliştirmesi ve benzer özellikte veya mevkide bulunacak yeni yollar için tahmini geçiş sayılarının planlanması vb. bu çalışmadan elde edilen veriler ışığında planlanabilir.

Sistemin en önemli parçası olan hava durumu modellerine ait başarımlarının %99.51-99.85 aralığında olması bu çalışmanın amacına uygun olarak gerçekleştirildiğinin en önemli dayanağıdır. CPU üzerinden anlık görüntüde görüntü işleme yöntemleri kullanılarak yapılan araç ve araç hızı tespitleri, hava durumu, kamera açısı, kamera erişilebilirliği, ışık şiddeti ve

açısı gibi parametrelere bağlı olduğundan hava durumu tespitinde elde edilen yüksek başarımların araç ve araç hızı tespitlerinde yakalanamaması doğaldır. Çalışma sırasında yaşanan anlık görüntü atlamaları araç sayısı ve tespit edilen aracın hızının tespiti konusunda en büyük sorun olmuştur. Bu bağlamda uygulamanın araç ve araç hızı tespitleri başarımlarının %76.46-94.67 aralığında olduğu görülmüştür. Elde edilen bu oranların sabit kamera görüntüsü erişimi, sabit internet erişimi vb. şartlar altında literatürdeki benzer çalışmaların başarımlarını yakaladığı görülmektedir

Hedeflenen sistemin ek maliyet oluşturmada tasarlanabileceğinin gösterilebilmesi adına araç tespiti ve hız hesabı CPU üzerinden görüntü işleme yöntemleri kullanılarak gerçekleştirilmiştir. Ancak araç tespitinde yüksek hızlı GPU içeren bilgisayarlar ile günümüzde yaygınca kullanılan nesne algılama modelleri kullanılırsa, hava durumu tespitinde olduğu gibi araç ve araç hızı tespitinde de yüksek başarımların yakalanabileceği öngörülmektedir. Bu çalışmanın devamı niteliğinde, araç ve araç hızı tespitinde nesne tespiti modelleri kullanılarak geliştirilen yeni bir uygulama tasarlanması planlanmaktadır.

KAYNAKLAR

- [1] Emniyet Genel Müdürlüğü Trafik Araştırma Merkezi Müdürlüğü, *Hız Yönetimi Karar Organları Ve Uygulayıcılar İçin Karayolu Güvenliği El Kitabı*, 2nd ed. Ankara: Uzerler Matbaası, Ankara, 2011. Accessed: Jan. 23, 2022. [Online]. Available: <http://www.trafik.gov.tr/kurumlar/trafik.gov.tr/Arsiv/SiteAssets/Yayinlar/Kitaplar/hizyonetimi.pdf>
- [2] TÜİK, “Motorlu Kara Taşıtları, Kasım 2021,” Ankara, Dec. 2021. Accessed: Jan. 24, 2022. [Online]. Available: <https://data.tuik.gov.tr/Bulten/Index?p=Motorlu-Kara-Tasitlari-Kasim-2021-37433>
- [3] OECD/ECMT Transport Research Centre, “Speed Management report,” 2006, Accessed: Jan. 04, 2022. [Online]. Available: <https://www.itf-oecd.org/sites/default/files/docs/06speed.pdf>
- [4] Ulaştırma ve Altyapı Bakanlığı Karayolları Genel Müdürlüğü, “Ulaştırma ve Altyapı Bakanlığı Karayolları Genel Müdürlüğü.” <https://www.kgm.gov.tr/Sayfalar/KGM/SiteTr/Trafik/HizSinirlari.aspx> (accessed Aug. 11, 2022).
- [5] T.C. Cumhurbaşkanlığı, “Karayolu Trafik Güvenliği Strateji Belgesi (2021-2030),” *Resmi Gazete*, Ankara, Feb. 03, 2021. Accessed: Jan. 25, 2022. [Online]. Available: http://www.trafik.gov.tr/kurumlar/trafik.gov.tr/01-Haberler/03-2021/2021_2030-Karayolu-Trafik-Guvenligi-Strateji-Belgesi.pdf
- [6] T.C. Cumhurbaşkanlığı, “Karayolu Trafik Güvenliği Eylem Planı (2021-2023),” *Resmi Gazete*, Ankara, Feb. 03, 2021. Accessed: Jan. 25, 2022. [Online]. Available: http://www.trafik.gov.tr/kurumlar/trafik.gov.tr/01-Haberler/03-2021/2021_2023-Karayolu-Trafik-Guvenligi-Eylem-Plani.pdf
- [7] P. N. Chowdhury, T. Chandra Ray, and J. Uddin, “A Vehicle Detection Technique for Traffic Management using Image Processing,” *International Conference on Computer, Communication, Chemical, Material and Electronic Engineering, IC4ME2 2018*, Sep. 2018, doi: 10.1109/IC4ME2.2018.8465599.
- [8] D. Biswas, H. Su, C. Wang, J. Blankenship, and A. Stevanovic, “An automatic car counting system using overfeat framework,” *Sensors (Switzerland)*, vol. 17, no. 7, Jul. 2017, doi: 10.3390/S17071535.

- [9] J. W. Hsieh, S. H. Yu, Y. S. Chen, and W. F. Hu, "Automatic Traffic Surveillance System for Vehicle Tracking and Classification," *IEEE Intelligent Transportation Systems Magazine*, vol. 7, no. 2, pp. 175–187, Jun. 2006, doi: 10.1109/TITS.2006.874722.
- [10] S. Gupte, O. Masoud, R. F. K. Martin, and N. P. Papanikolopoulos, "Detection and Classification of Vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 3, no. 1, pp. 37–47, Mar. 2002, doi: 10.1109/6979.994794.
- [11] H. van Pham and B. R. Lee, "Front-view car detection and counting with occlusion in dense traffic flow," *International Journal of Control, Automation and Systems*, vol. 13, no. 5, pp. 1150–1160, Oct. 2015, doi: 10.1007/S12555-014-0229-7.
- [12] A. Kanungo, A. Sharma, and C. Singla, "Smart traffic lights switching and traffic density calculation using video processing," *2014 Recent Advances in Engineering and Computational Sciences, RAECS 2014*, 2014, doi: 10.1109/RAECS.2014.6799542.
- [13] D. W. Wicaksono and B. Setiyono, "Speed Estimation On Moving Vehicle Based On Digital Image Processing," *International Journal of Computing Science and Applied Mathematics*, vol. 3, no. 1, p. 21, Mar. 2017, doi: 10.12962/J24775401.V3I1.2117.
- [14] T. Parag, A. Elgammal, and A. Mittal, "A framework for feature selection for background subtraction," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 1916–1923, 2006, doi: 10.1109/CVPR.2006.24.
- [15] P. K. Bhaskar and S. P. Yong, "Image processing based vehicle detection and tracking method," *2014 International Conference on Computer and Information Sciences, ICCOINS 2014 - A Conference of World Engineering, Science and Technology Congress, ESTCON 2014 - Proceedings*, Jul. 2014, doi: 10.1109/ICCOINS.2014.6868357.
- [16] S. S. Cheung and C. Kamath, "Robust techniques for background subtraction in urban traffic video," <https://doi.org/10.1117/12.526886>, vol. 5308, pp. 881–892, Jan. 2004, doi: 10.1117/12.526886.
- [17] N. Y. Ershadi, J. M. Menéndez, and D. Jiménez, "Robust vehicle detection in different weather conditions: Using MIPM," *PLoS ONE*, vol. 13, no. 3, Mar. 2018, doi: 10.1371/JOURNAL.PONE.0191355.
- [18] G. Ç. Şen, "Hareketli görüntüde hız kestirimi," *YILDIZ TEKNİK ÜNİVERSİTESİ*, İstanbul, 2009. Accessed: Jan. 04, 2022. [Online]. Available: <http://dSPACE.yildiz.edu.tr/xmlui/handle/1/7887>

- [19] W. T. Chu, X. Y. Zheng, and D. S. Ding, "Camera as weather sensor: Estimating weather information from single images," *Journal of Visual Communication and Image Representation*, vol. 46, pp. 233–249, Jul. 2017, doi: 10.1016/J.JVCIR.2017.04.002.
- [20] B. B. Traore, B. Kamsu-Foguem, and F. Tangara, "Deep convolution neural network for image recognition," *Ecological Informatics*, vol. 48, pp. 257–268, Nov. 2018, doi: 10.1016/J.ECOINF.2018.10.002.
- [21] X. Luo, R. Shen, J. Hu, J. Deng, L. Hu, and Q. Guan, "A Deep Convolution Neural Network Model for Vehicle Recognition and Face Recognition," *Procedia Computer Science*, vol. 107, pp. 715–720, Jan. 2017, doi: 10.1016/J.PROCS.2017.03.153.
- [22] L. Chen, F. Ye, Y. Ruan, H. Fan, and Q. Chen, "An algorithm for highway vehicle detection based on convolutional neural network," *Eurasip Journal on Image and Video Processing*, vol. 2018, no. 1, pp. 1–7, Dec. 2018, doi: 10.1186/S13640-018-0350-2/TABLES/2.
- [23] Z. Zhang, H. Ma, H. Fu, and C. Zhang, "Scene-free multi-class weather classification on single images," *Neurocomputing*, vol. 207, pp. 365–373, Sep. 2016, doi: 10.1016/J.NEUCOM.2016.05.015.
- [24] X. Zhang, Y. Zou, and W. Shi, "Dilated convolution neural network with LeakyReLU for environmental sound classification," *International Conference on Digital Signal Processing, DSP*, vol. 2017-August, Nov. 2017, doi: 10.1109/ICDSP.2017.8096153.
- [25] G. Li *et al.*, "Hand gesture recognition based on convolution neural network," *Cluster Computing*, vol. 22, pp. 2719–2729, Mar. 2019, doi: 10.1007/S10586-017-1435-X.
- [26] U. P. Singh, S. S. Chouhan, S. Jain, and S. Jain, "Multilayer Convolution Neural Network for the Classification of Mango Leaves Infected by Anthracnose Disease," *IEEE Access*, vol. 7, pp. 43721–43729, 2019, doi: 10.1109/ACCESS.2019.2907383.
- [27] M. Roser and F. Moosmann, "Classification of weather situations on single color images," *IEEE Intelligent Vehicles Symposium, Proceedings*, pp. 798–803, 2008, doi: 10.1109/IVS.2008.4621205.
- [28] H. Tayara, K. G. Soo, and K. T. Chong, "Vehicle Detection and Counting in High-Resolution Aerial Images Using Convolutional Regression Neural Network," *IEEE Access*, vol. 6, pp. 2220–2230, Dec. 2017, doi: 10.1109/ACCESS.2017.2782260.
- [29] M. T. Zeren, "İnsansız hava araçları ve uydu görüntülerinden elde edilen veri seti ile havaalanlarının tespiti yaplmasında SSD ve Faster R-CNN algoritmalarının karşılaştırılması," Apr. 2020, Accessed: Jan. 04, 2022. [Online]. Available: <https://acikbilim.yok.gov.tr/handle/20.500.12812/709375>

- [30] A. Turing, “Computing Machinery and Intelligence (1950),” *The Essential Turing*, Sep. 2004, doi: 10.1093/OSO/9780198250791.003.0017.
- [31] W. S. McCulloch and W. Pitts, “A logical calculus of the ideas immanent in nervous activity,” *The bulletin of mathematical biophysics* 1943 5:4, vol. 5, no. 4, pp. 115–133, Dec. 1943, doi: 10.1007/BF02478259.
- [32] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks,” *Advances in Neural Information Processing Systems*, vol. 25, 2012.
- [33] Ergin Tuncer, “Convolutional Neural Network (ConvNet yada CNN) nedir, nasıl çalışır?” <https://medium.com/@tuncerergin/convolutional-neural-network-convnet-yada-cnn-nedir-nasil-calisir-97a0f5d34cad> (accessed Jan. 21, 2022).
- [34] “CS 230 - Evrişimli Sinir Ağları El Kitabı.” <https://stanford.edu/~shervine/l/tr/teaching/cs-230/cheatsheet-convolutional-neural-networks> (accessed Jan. 21, 2022).
- [35] Doğan Özgür, “CNN (Convolutional Neural Networks) Nedir?” <https://teknoloji.org/cnn-convolutional-neural-networks-nedir/> (accessed Jan. 21, 2022).
- [36] Perihanoğlu and G. Miray, “Dijital Görüntü İşleme Teknikleri Kullanılarak Görüntülerden Detay Çıkarımı,” Feb. 2015, Accessed: Jan. 23, 2022. [Online]. Available: <https://polen.itu.edu.tr/handle/11527/13574>
- [37] R. E. Kalman, “A new approach to linear filtering and prediction problems,” *Journal of Fluids Engineering, Transactions of the ASME*, vol. 82, no. 1, pp. 35–45, 1960, doi: 10.1115/1.3662552.
- [38] K. Saho, “Kalman Filter for Moving Object Tracking: Performance Analysis and Filter Design,” *Kalman Filters - Theory for Advanced Applications*, Feb. 2018, doi: 10.5772/INTECHOPEN.71731.
- [39] “Intel® Core™ i7-9750H İşlemci.” <https://www.intel.com.tr/content/www/tr/tr/products/sku/191045/intel-core-i79750h-processor-12m-cache-up-to-4-50-ghz/specifications.html> (accessed May 17, 2022).
- [40] “Python.” <https://www.python.org/> (accessed Mar. 30, 2022).
- [41] “PyQt v5.15 Reference Guide.” <https://www.riverbankcomputing.com/static/Docs/PyQt5/introduction.html#pyqt5-components> (accessed May 17, 2022).

- [42] “Pafy Documentation — pafy 0.5.1 documentation.” <https://pythonhosted.org/pafy/> (accessed May 17, 2022).
- [43] “Joblib: running Python functions as pipeline jobs — joblib 1.2.0.dev0 documentation.” <https://joblib.readthedocs.io/en/latest/> (accessed May 17, 2022).
- [44] “os — Miscellaneous operating system interfaces — Python 3.10.4 documentation.” <https://docs.python.org/3/library/os.html> (accessed May 17, 2022).
- [45] “YouTube.” <https://www.youtube.com/> (accessed Mar. 30, 2022).
- [46] “İBB Trafik Yoğunluk Haritası.” <https://uym.ibt.gov.tr/yharita6/> (accessed Mar. 30, 2022).
- [47] “OpenCV.” <https://opencv.org/> (accessed Mar. 30, 2022).
- [48] “NumPy.” <https://numpy.org/> (accessed May 17, 2022).
- [49] “GitHub - DateTime.” <https://github.com/zopecfoundation/DateTime> (accessed May 17, 2022).
- [50] “random Python 3.10.4 documentation.” <https://docs.python.org/3/library/random.html> (accessed May 17, 2022).
- [51] “pickle — Python 3.10.4 documentation.” <https://docs.python.org/3/library/pickle.html> (accessed May 17, 2022).
- [52] “TensorFlow.” <https://www.tensorflow.org/> (accessed May 17, 2022).
- [53] “scikit-learn: machine learning in Python — scikit-learn 1.1.0 documentation.” <https://scikit-learn.org/stable/> (accessed May 17, 2022).
- [54] “Keras: the Python deep learning API.” <https://keras.io/> (accessed May 17, 2022).
- [55] “seaborn: statistical data visualization — seaborn 0.11.2 documentation.” <https://seaborn.pydata.org/> (accessed May 17, 2022).
- [56] “Matplotlib — Visualization with Python.” <https://matplotlib.org/> (accessed May 17, 2022).
- [57] “OpenCV: How to Use Background Subtraction Methods.” https://docs.opencv.org/4.x/d1/dc5/tutorial_background_subtraction.html (accessed Apr. 30, 2022).
- [58] N. Chintalacheruvu, V. Muthukumar, N. Chintalacheruvu, and V. Muthukumar, “Video Based Vehicle Detection and its Application in Intelligent Transportation Systems,” *Journal of Transportation Technologies*, vol. 2, no. 4, pp. 305–314, Oct. 2012, doi: 10.4236/JTTS.2012.24033.
- [59] R. Marc, G. Dominique, and P. Evangeline, “Generator of road marking textures and associated ground truth Applied to the evaluation of road marking detection,” *IEEE*

Conference on Intelligent Transportation Systems, Proceedings, ITSC, pp. 933–938, 2012, doi: 10.1109/ITSC.2012.6338773.

- [60] Trafik Güvenliği Dairesi Başkanlığı/Trafik Güvenliği İşaretleme Şubesi Müdürlüğü, “Karayolu Trafik İşaretleme Standartları 1,” 2020. Accessed: Jun. 12, 2022. [Online]. Available: <https://www.kgm.gov.tr/SiteCollectionDocuments/KGMdocuments/Trafik/IsaretlerElKitabi/KarayoluTrafikIsaretlemeStandartlari1.pdf>
- [61] “Kalman filter, predict the trajectory of an Object - Pysource.” <https://pysource.com/2021/11/02/kalman-filter-predict-the-trajectory-of-an-object/> (accessed Jun. 14, 2022).



ÖZGEÇMİŞ

Kişisel Bilgiler

Adı-Soyadı Emir Mustafa EFE

Eğitim Bilgileri

Lise Malatya Anadolu Lisesi

Lisans İstanbul Üniversitesi Elektrik-Elektronik Mühendisliği

Çalıştığı Kurumlar

2016 - 2017 BOTECH Görüntü Güvenlik Sistemleri San ve Tic A.Ş.

2017 - TÜBİTAK/Bilişim ve Bilgi Güvenliği İleri Teknolojiler
Araştırma Merkezi

