

T.C.
FIRAT ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ

GÖRÜNTÜ ÜZERİNDE ARAÇ LOGOSUNUN TESPİTİ VE
LOGONUN İÇBOYAMA İŞLEMİ İLE KALDIRILMASI

Mert İNCİDELEN
Okan YAR

Tez Danışmanı:
Doç. Dr. Burhan ERGEN

BİTİRME TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

ELÂZİĞ
2020

T.C.
FIRAT ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ

**GÖRÜNTÜ ÜZERİNDE ARAÇ LOGOSUNUN TESPİTİ VE
LOGONUN İÇBOYAMA İŞLEMİ İLE KALDIRILMASI**

Mert İNCİDELEN

Okan YAR

BİTİRME TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

Bu bitirme tezi ... / ... / 2020 tarihinde, aşağıda belirtilen jüri tarafından oybirliği/oyçokluğu ile başarılı/başarısız olarak değerlendirilmiştir.

Danışman

Üye

Üye

Doç. Dr. Burhan ERGEN

ÖZGÜNLÜK BİLDİRİMİ

Bu çalışmada, başka kaynaklardan yapılan tüm alıntılar, ilgili kaynaklar referans gösterilerek açıkça belirtildiğini, alıntılar dışındaki bölümlerin, özellikle projenin ana konusunu oluşturan teorik çalışmaların ve yazılım/donanımın benim tarafımdan yapıldığını bildiririm.

Fırat Üniversitesi

02/07/2020

Bilgisayar Mühendisliği

23100 Elâzığ

Mert İNCİDELEN

Okan YAR

BENZERLİK BİLDİRİMİ

02.07.2020

Turnitin Orijinallik Raporu

Turnitin Orijinallik Raporu

MertOkan Burhan Ergen tarafından

Btirme2020 (Özgünlük İnceleme) den



- 02-Tem-2020 13:48 +03' de işleme kondu
- NUMARA: 1352565833
- Kelime Sayısı: 7376

Benzerlik Endeksi

%4

Kaynağa göre Benzerlik

Internet Sources:

%3

Yayınlar:

%1

Öğrenci Ödevleri:

N/A

kaynaklar:

- 1 1% match (08-May-2019 tarihli internet)
<https://polen.itu.edu.tr/bitstream/11527/12993/1/10078761.pdf>
- 2 < 1% match (25-Haz-2020 tarihli internet)
<https://openaccess.iku.edu.tr/bitstream/handle/11413/1611/Hilal%20%C5%9Eavl%C4%B1.pdf?isAllowed=y&sequence=1>
- 3 < 1% match (07-Oca-2020 tarihli internet)
<https://www.yumpu.com/tr/document/view/32830680/uzaktan-algalama-verileri-ile-kaya-aizgisi-dealiminin-zamansal-olarak->
- 4 < 1% match (08-May-2019 tarihli internet)
<https://polen.itu.edu.tr/bitstream/11527/12376/1/10031547.pdf>
- 5 < 1% match (10-Eki-2019 tarihli internet)
<http://muhammetbaykara.com/hakkimda/>
- 6 < 1% match (26-May-2019 tarihli internet)
<http://www.openaccess.hacettepe.edu.tr:8080/xmlui/bitstream/handle/11655/2237/eef30c30-bb50-4c59-8261-afa881d6d58e.pdf?isAllowed=y&sequence=1>
- 7 < 1% match (24-Haz-2015 tarihli internet)
<http://www.slideshare.net/ERPCommittee/kurumsal-zmler-rehberi-2013>
- 8 < 1% match (05-Haz-2020 tarihli internet)
<https://es.scribd.com/document/388900276/gulsum-hasiloglu>
- 9 < 1% match (yayınlar)
[Merve Polat, Hussein Mahmood Abdo Mohammed, Emin Argun Oral, Ibrahim Yucel Ozbek, "Aircraft Detection from Satellite Images Using ATA-Plane Data Set", 2019 27th Signal Processing and Communications Applications Conference \(SIU\), 2019](#)

TEŞEKKÜR

Unutulmayacak anılara sahip olduğumuz ve büyük kazanımlar edindiğimiz dört yıllık eğitim serüvenimizi başarı ile tamamlamanın heyecanını duyduğumuz bitirme tezi sürecimizde bizden desteklerini, deneyimlerini ve tavsiyelerini hiçbir zaman esirgemeyen değerli yol göstericilerimiz Sayın Doç. Dr. Burhan ERGEN Hocamıza ve Sayın Meryem TAŞKESEN'e sabırları ve ilgileri için yürekten teşekkürlerimizi sunarız.

İÇİNDEKİLER

ÖZGÜNLÜK BİLDİRİMİ	I
BENZERLİK BİLDİRİMİ	II
TEŞEKKÜR	III
İÇİNDEKİLER	IV
ŞEKİLLER	VI
TABLOLAR	VIII
KISALTMALAR	IX
ÖZET	X
ABSTRACT.....	XI
1. GİRİŞ.....	1
1.1. Çalışmanın Amacı.....	1
1.2. Çalışmada İncelenen Yöntemler.....	2
1.3. Çalışma İçeriği	2
2. SAYISAL GÖRÜNTÜ VE GÖRÜNTÜ İŞLEME	4
2.1. Görüntü İşleme.....	5
2.2. Görüntü İşleme Uygulama Alanları.....	6
3. GÖRÜNTÜ ÜZERİNDE İSTENİLEN NESNENİN BULUNMASI.....	8
3.1. Makine Öğrenimi İle Nesne Tespit Yöntemleri.....	8
3.1.1. Haar-Benzeri Öznitelikler Bazlı Nesne Tespiti.....	9
3.1.2. Histogram of Oriented Gradients (HOG) İle Nesne Tespiti	11
3.2. Derin Öğrenme İle Nesne Tespit Yöntemleri	13
3.2.1. Bölgesel Tabanlı Evrişimsel Sinir Ağı (R-CNN) İle Nesne Tespiti	14
3.2.2. You Only Look Once (YOLO) İle Nesne Tespiti	17
4. GÖRÜNTÜ ÜZERİNDE BELİRLENMİŞ BÖLGELERİN İÇBOYAMA İŞLEMİ İLE TAMAMLANMASI.....	19
4.1. Geleneksel İçboyama Yöntemleri.....	19
4.1.1. Hızlı Yürüyüş Tekniği (FMM) Bazlı İçboyama	20
4.1.2. Kısmi Diferansiyel Denklemler (PDE) Bazlı İçboyama	22
4.1.3. Örnek Bazlı İçboyama	24
5. UYGULAMA.....	26
5.1. Uygulamanın Tanımı.....	26
5.2. Görüntü Üzerinde Araç Logosunun Tespiti.....	27
5.2.1. Eğitim Süreci.....	28
5.2.2. Logonun Tespiti	32

5.3. Araç Logosunun İçboyama İşlemi İle Kaldırılması	33
5.3.1. Belirlenen Bölge İçin Maskelenmiş Görüntünün Çıkarılması	33
5.3.2. Maskelenmiş Görüntü Kullanılarak İçboyama İşleminin Gerçekleştirilmesi	35
5.4. Uygulama Sonuçları	36
6. SONUÇ	40
ÖZGEÇMİŞ	41
KAYNAKLAR	43

ŞEKİLLER

ŞEKİL 2.1. Analog Sinyalden Sayısal Sinyalin Oluşturulması	4
ŞEKİL 2.2. Görüntünün Satır ve Sütunlarında Bulunan Pikseller	5
ŞEKİL 2.3. Görüntü Üzerindeki Pikseller ve Sayısal Karşılıkları	5
ŞEKİL 2.4. a) Orijinal Görüntü b) Yumuşatılmış Görüntü (Gaussian Smoothing)	6
ŞEKİL 2.5. Görüntü İşlemenin Diğer Disiplinler Arası İlişkisi	6
ŞEKİL 3.1. Görüntü Üzerinde Nesnelerin Tespit Edilmesi	8
ŞEKİL 3.2. Haar-Benzeri Öznitelikler	9
ŞEKİL 3.3. Haar-Benzeri Öznitelik Saptamaları	10
ŞEKİL 3.4. Yüz Tanıma Eğitimi İçin Pozitif Görüntü Kümesi	10
ŞEKİL 3.5. Yüz Tanıma Eğitimi Sonuçları Kullanılarak Görüntüdeki Yüzün Tespiti	11
ŞEKİL 3.6. Histogram of Oriented Gradients Aşamaları	11
ŞEKİL 3.7. Hücrelere Bölünmüş Görüntüde Bir Hücresin Gradyan Büyüklükleri ve Yön Değerleri	12
ŞEKİL 3.8. Histogram of Oriented Gradients Algoritmasının Akış Şeması	13
ŞEKİL 3.9. Yapay Sinir Ağlarının Mimarisi	14
ŞEKİL 3.10. R-CNN Aşamaları	15
ŞEKİL 3.11. R-CNN Zaman Tüketimi Karşılaştırmaları	15
ŞEKİL 3.12. Evrimsel Sinir Ağı Tasviri	16
ŞEKİL 3.13. Konvolüsyon İşlemi	16
ŞEKİL 3.14. YOLO Algoritmasının İşleyişi	17
ŞEKİL 3.15. You Only Look Once Yönteminin Uygulanması	17
ŞEKİL 3.16. Tahmin Bölgesi ve Sınırlanması	18
ŞEKİL 4.1. Görüntü İçboyama Uygulamaları	19
ŞEKİL 4.2. Hızlı Yürüyüş Tekniği İle İçboyama İşleminde Boyanacak Piksel (p) ve Bilinen Komşu Piksellerden Oluşan Alan Gösterimi	20
ŞEKİL 4.3. Görüntü Üzerinde Hızlı Yürüyüş Tekniği Kullanılarak İçboyama İşleminin Uygulanması: a) Orijinal Görüntü, b) Maske, c) Sonuç Görüntü	21
ŞEKİL 4.4. Görüntü Üzerinde Büyük Bir Alanın Hızlı Yürüyüş Tekniği ile İçboyama Yöntemi Uygulanarak Doldurulması: a) Orijinal Görüntü, b) Maske Görüntü, c) Sonuç Görüntü	21
ŞEKİL 4.5. Aynı Parlaklıktaki Bölgelerin İzofot Eğrileri	22
ŞEKİL 4.6. İzofot Eğrilerinin Eksik Alan İçine Yayılımı (Sayılar, gri seviyeleri ifade etmektedir): a) İçboyama Uygulanacak Alan ve Çevresindeki İzofot Çizgileri, b) İçboyama Uygulanacak Alana İzofot Çizgilerinin Yayılımı	22

ŞEKİL 4.7. Görüntü Üzerinde Navier-Stokes Denklemleri ile İçboyama Tekniği Kullanılarak İçboyama İşleminin Uygulanması: a)Orijinal Görüntü, b)Maske Görüntü, c)Sonuç Görüntü.....	23
ŞEKİL 4.8. Görüntü Üzerinde Büyük Bir Alanın Navier-Stokes Denklemleri ile İçboyama Uygulanarak Doldurulması: a) Orijinal Görüntü, b) Maske Görüntü, c) Sonuç Görüntü	24
ŞEKİL 4.9. Örnek Tabanlı İçboyama İşlemi Aşamaları: a) Orijinal Görüntü, b) Sınır Üzerinde ($\partial\Omega$) Yüksek Öncelikli Doldurulacak p Merkezli Alan, c) Doldurulacak Alan İçin Uygun Yamaların (Ψ_q' , Ψ_q'') Belirlenmesi, d) Uygun Yamanın Bölgeye Uygulanması.....	24
ŞEKİL 4.10. Görüntü Üzerinde Büyük Bir Alanın, Örnek Tabanlı İçboyama Yöntemi Uygulanarak Doldurulması: a) Doldurulmak İstenilen Beyaz Alan, b) Bölgenin Maskelenmesi, c) Örnek Bazlı İçboyama İşlemi Uygulanmış Sonuç Görüntü	25
ŞEKİL 5.1. Uygulamanın Grafıksel Arayüzü.....	26
ŞEKİL 5.2. Uygulamanın İşleyiş Şeması	27
ŞEKİL 5.3. Araç Logolarının Bulunduğu Örnek Pozitif Görüntüler	28
ŞEKİL 5.4. Araç Logolarının Bulunmadığı Örnek Negatif Görüntüler	29
ŞEKİL 5.5. Pozitif Görüntüleri 50x50 Boyutunda Yeniden Ölçeklendiren Metot	29
ŞEKİL 5.6. Pozitif Görüntü Verilerin Bilgilerini Listeleyen Metot	30
ŞEKİL 5.7. Negatif Görüntüleri 100x100 Boyutunda Yeniden Ölçeklendiren Metot ve Negatif Görüntü Verilerin Bilgilerini Listeleyen Metot	30
ŞEKİL 5.8. Eğitimi Başlatmak İçin Gereken Komut ve Parametreler	31
ŞEKİL 5.9. Eğitim Sonucunda Elde Edilen .xml Uzantılı Eğitim Çıktıları	31
ŞEKİL 5.10. Logonun Bulunduğu Bölgeyi Kapsayan Çerçevenin Bilgilerini Veren Kod Parçası	32
ŞEKİL 5.11. Logoyu İçine Alan Çerçevenin Orijinal Görüntü Üzerine Çizdirilmesi.....	32
ŞEKİL 5.12. Logonun Tespiti	33
ŞEKİL 5.13. Logonun Tespitini Gerçekleştirerek, Tespit Edilen Bölge İçin Maskelenmiş Görüntü Oluşturan Metot	34
ŞEKİL 5.14. a) Orijinal Görüntü b) Logonun Çerçeve İçine Alınmış Hali c) Maskelenmiş Görüntü	35
ŞEKİL 5.15. İçboyama İşleminin Gerçekleştirilmesi	35
ŞEKİL 5.16. a) Orijinal Görüntü, b) Logonun Çerçeve İçine Alınmış Hali, c) Maskelenmiş Görüntü, d) İçboyama Uygulanan Sonuç Görüntü	36
ŞEKİL 5.17. Uygulamanın Proje Dosyası İçeriği.....	36
ŞEKİL 5.18. Farklı Marka Araçlar İçin Uygulamanın Sonuçları	37

TABLÖLAR

TABLÖ 5.1. Orijinal ve Sonuç Görüntülerin Karşılaştırması İçin MSE ve PSNR Değerleri ...**38**

TABLÖ 5.2. Uygulamanın Her Araç Sınıfı İçin Başarı Değerlendirmesi**39**

KISALTMALAR

CNN: Convolutional Neural Networks

R-CNN: Region-based Convolutional Neural Networks

YOLO: You Only Look Once

FMM: Fast Marching Method

PDE: Partial Differential Equations

HOG: Histogram of Oriented Gradients

YSA: Yapay Sinir Ağları

RRN: Recurrent Neural Network

RBM: Restricted Boltzmann Machine

DBN: Deep Belief Networks

LSTM: Long Short-Term Memory

JPEG: Joint Photographic Experts Group

PNG: Portable Network Graphics

BMP: Bitmap

RGB: Red-Green-Blue

CMD: Command

MSE: Mean-Squared Error

PSNR: Peak Signal to Noise Ratio

ÖZET

Teknolojinin gelişmesi ve yaygınlaşmasıyla birlikte, teknoloji hayatımızın değişmez bir parçası olmuştur. Görüntü kayıt cihazları da gelişen teknolojilerden nasibini alarak büyük ilerlemeler katetmiştir. Akıllı telefonların da yaşamımıza girmesi ile, dijital görüntüler hayatımızda önemli bir yer edinmiştir. Her gün devasa boyutlarda görüntü verileri üretilmekte ve bu verileri anlamlandırma gereksinimleri ortaya çıkmaktadır. Hayatımızın hemen her alanında görüntülerin işlenmesi ve analizi ihtiyacı doğmuş, bu bakımdan çeşitli sistemler ve uygulamalar ortaya konulmuştur. Tüm bu sebepler doğrultusunda görüntü işleme giderek ilgi gören bir araştırma alanı olma statüsünü korumaktadır. Bu çalışmada görüntü işlemenin temel kavramlarından bahsedilerek, görüntü üzerinde nesne tespit yöntemleri ve içboyama yöntemleri incelenmiş ve incelenen yöntemlerin işleyiş prensipleri açıklanmaya çalışılmıştır. İncelenen yöntemlerden seçilen yollarla, nesne tanıma ve içboyama işlemleri birleştirilerek görüntüdeki aracın logosunu tespit eden ve tespit edilen bölgeye içboyama işlemi uygulayarak görüntü üzerinden logonun kaldırılmasını amaçlayan bir uygulama ortaya konulmuştur. Uygulamanın çalışma aşamaları ve kaynak kodları verilerek, elde edilen sonuçlar değerlendirilmiştir.

ANAHTAR KELİMELE: Görüntü İşleme, Nesne Tanıma, İçboyama, Makine Öğrenmesi, Derin Öğrenme

ABSTRACT

With the development and spread of technology, technology has become an indispensable part of our lives. Image recorders have also made significant progress by taking advantage of developing technologies. With the introduction of smartphones into our lives, digital images have taken an important place. Image data is produced in enormous dimensions every day, and there is a need to make sense of this data. The need for processing and analyzing images in almost every part of our lives has arisen, and in this regard, various systems and applications have been introduced. For all these reasons, image processing maintains its status as an increasingly popular research area. In this study, by mentioning the basic concepts of image processing, object detection methods, and inpainting methods on the image and the working principles of the studied methods, they are tried to be explained. By combining the methods of object recognition and inpainting in the ways chosen from the methods examined, An application that detects the logo of the vehicle in the image and aims to remove the logo on the image by applying inpainting on the determined area has been developed. Working steps and source codes of the application are given, and the results are evaluated.

KEYWORDS: Image Processing, Object Detection, Inpainting, Machine Learning, Deep Learning

1. GİRİŞ

İnsanlığın gerçek dünya algısını resmetme çabası 40000 yıl öncesine kadar dayanmaktadır. Bilinen en eski resim en az 40000 yıl öncesine ait olduğu düşünülen Endonezya'da Borneo Adası'ndaki bir mağarada bulunan duvarlara işlenmiş bir avlanma tasviridir. [1] Geçmişten günümüze insanlar, farklı yöntem ve çabalar ile soyut düşünebilme yetisini kullanarak gerçek dünyaya karşı sahip olduğu soyut algıyı somut resme dönüştürmüştür. Önceleri yaşama karşı mistik inanışlar ve benzeri nedenler dolayısıyla ortaya çıkan bu uğraşı, yakın tarihe bakıldığında bir sanat uğraşısı haline gelmiştir. Gerçek dünyayı resmetme uğraşısı teknolojinin getirdikleri ile birlikte bambaşka bir boyut kazanmıştır. Görüntü kayıt cihazları ve bilgisayarların yaşamımıza girmesiyle sayısal görüntü kavramı ortaya çıkmış ve yeni araştırma alanları doğmasına sebep olmuştur. Bu araştırma alanlarından biri de görüntü işleme alanıdır. Görüntü işleme hayatımızın hemen her alanında karşılık bulmaktadır. Sağlık uygulamalarından savunma sanayisine, endüstriyel uygulamalardan eğlence sektörüne birçok alanda görüntü işleme uygulamalarından faydalanılmaktadır. Bu durum görüntü işleme alanını giderek popüleritesi artan bir araştırma alanı yapmıştır. Görüntü işleme, aynı zamanda çok çeşit konuyu da bünyesinde barındırmaktadır. Bunlardan biri görüntünün analiz edilerek görüntü üzerinde nesnenin tanınması hususudur. Nesnenin tanınması ve değerlendirilmesi konusunda birçok bilim insanı tarafından araştırma çalışmaları yayınlanmış, teknikler ve algoritmalar önerilmiştir. Görüntü işlemenin kapsadığı bir diğer konu ise görüntü üzerinde belirli bölgenin içboyama ile doldurulması işlemidir. İçboyama için bilim insanları tarafından düşünülmüş ve hayata geçirilmiş birçok yöntem bulunmaktadır. Bu çalışmada, içboyama ve nesne tanıma alanlarında şimdiye kadar ortaya konulmuş bu yöntemlerden bazıları ele alınarak incelenmekte ve incelenen yöntemlerden seçilenler ile bir uygulama gerçekleştirilmektedir. Uygulama, nesne tanıma tekniği ile araç üzerindeki logoyu tespit etmekte ve tespit edilen logo içboyama işlemi uygulanarak kaldırılmaktadır. Uygulamanın çalışma prensipleri ve sonuçları bu çalışmanın içinde verilmiş ve değerlendirilmiştir.

1.1. Çalışmanın Amacı

Nesne tespiti ve içboyama işlemleri günümüzde birçok uygulamada farklı amaçlar ile kullanılmaktadır. Her gün kullandığımız akıllı telefonlardaki uygulamalarda bulunan yüz tanıma işlevlerinde, parmak izi tanıma sistemlerinde, plaka tanıma sistemlerinde, sağlık alanındaki tıbbi görüntülerindeki hastalık teşhislerinde, uydu görüntülerinde ve daha birçok alanda nesne tespit yöntemleri kullanılarak görüntülerin analizi gerçekleştirilmektedir. Yine aynı şekilde içboyama

işlemi de günlük yaşamda birçok uygulamada karşılık bulmaktadır. Eski görüntülerdeki leke ve kayıpların giderilmesi ve onarımında, resim düzenleme uygulamalarında, görüntüdeki istenmeyen kısımların kapatılmasında ve birçok alanda yaygın olarak kullanılmaktadır. Bu çalışmamızda nesne tespit yöntemleri ve içboyama yöntemleri incelenmiş ve işleyiş prensipleri açıklanmaya çalışılmıştır. İncelenen yöntemlerden seçilen yollarla bir uygulama ortaya konulmuştur. Uygulama, görüntü üzerinde bulunan araçların logolarını tespit ederek kaldırmayı amaçlamaktadır. Logonun kaldırılması içboyama işlemi ile gerçekleştirilmektedir. Uygulama ile belirli sınıflardaki araç markalarının logoları tanınarak içboyama işlemi ile görüntü üzerinden kaldırılmaktadır.

1.2. Çalışmada İncelenen Yöntemler

Çalışmada nesne tanıma ve içboyama yöntemleri incelenmiş ve açıklanmıştır. Nesne tanıma yöntemleri makine öğrenimi ile tespit yöntemleri ve derin öğrenme ile nesne tespit yöntemleri şeklinde iki alt başlıkta incelenmiştir. Makine öğrenimi ile nesne tespit yöntemlerinden Haar-Benzeri Öznitelikler Bazlı Nesne Tespiti ve Histogram of Oriented Gradients İle Nesne Tespiti yöntemleri ele alınmıştır. Derin öğrenme ile nesne tespit yöntemlerinden Bölgesel Tabanlı Evrimsel Sinir Ağı (R-CNN) ile Nesne Tespiti ve You Only Look Once (YOLO) İle Nesne Tespiti yöntemleri ele alınmıştır. İçboyama yöntemleri geleneksel içboyama yöntemleri başlığı altında incelenmiştir. Geleneksel içboyama yöntemleri başlığında Hızlı Yürüyüş Tekniği (FMM) Bazlı İçboyama, Kısmi Diferansiyel Denklemler (PDE) Bazlı İçboyama ve Örnek Bazlı İçboyama Yöntemleri ele alınmıştır.

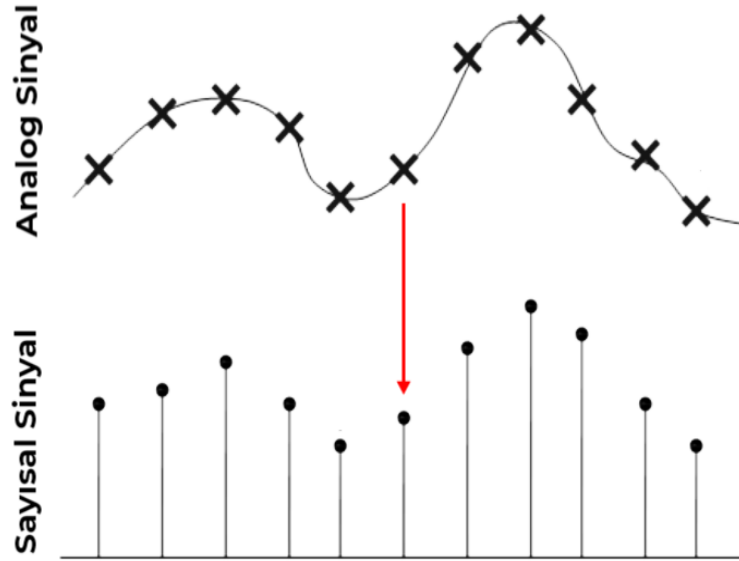
1.3. Çalışma İçeriği

Çalışmada sayısal görüntü ve sayısal görüntü işleme kavramları açıklanmıştır. Görüntü işleme uygulama alanlarından bahsedilerek örnekler verilmiştir. Sonraki ana başlıkta nesne tanıma yöntemleri, makine öğrenimi ile tespit yöntemleri ve derin öğrenme ile nesne tespit yöntemleri şeklinde iki alt başlık altında incelenmiş ve açıklanmıştır. Bu alt başlıklar altında 1.2. bölümde belirtilen nesne tanıma yöntemleri incelenmiştir. Bir sonraki ana başlıkta ise iç boyama yöntemleri geleneksel içboyama yöntemleri alt başlığı altında incelenmiş ve açıklanmıştır. Bu alt başlıklar altında 1.2. bölümde belirtilen içboyama yöntemleri incelenmiştir. Uygulama ana başlığı altında, önceki ana başlıklarda incelenen yöntemlerden seçilen yollarla ortaya konulmuş uygulama hakkında bilgiler yer almaktadır. Bu ana başlıkta uygulamanın tanımı yapılmış ve kullanılan yöntemler sunulmuştur. Görüntü üzerinde araç logosunun tespiti aşamaları açıklanmış, tespit edilen logonun içboyama yöntemi ile kaldırılma işlemi tarif edilmiştir. Uygulamanın

işlevlerinden bahsedilerek, giriş görüntüleri için üretilen sonuç görüntüleri sunulmuştur. Giriş ve sonuç verileri karşılaştırılarak uygulamanın başarı değeri değerlendirilmiştir.

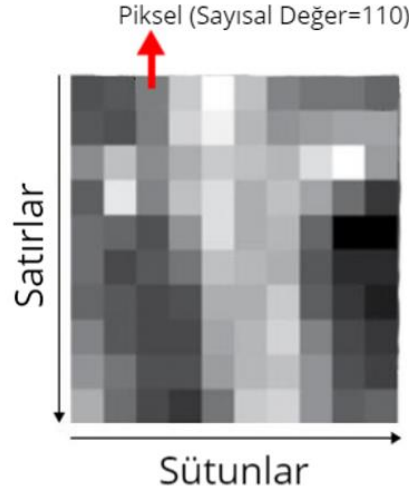
2. SAYISAL GÖRÜNTÜ VE GÖRÜNTÜ İŞLEME

Bilgisayarların günlük yaşamımıza girmesi ve yaygınlaşmasıyla birlikte sayısal görüntü kavramı ortaya çıkmıştır. Çok çeşitli görüntü kayıt cihazları ile gerçek yaşamdaki görüntü bilgisayar ortamına taşınabilmiş, aynı zamanda bilgisayarlar üzerinde de sayısal görüntüler oluşturulabilmiştir. Günlük yaşamdaki bilgi sonsuzdur ve bilgisayarlar tarafından işlenemezler. Bu nedenle analog bir sinyalin bilgisayar tarafından işlenebilmesi için belirli örnek aralıklarıyla değerleri alınarak sayısal sinyale dönüştürülmesi gerekir. Böylelikle günlük yaşam bilgisi, sayısal değerler ile bilgisayar ortamında temsil edilebilir bir nitelik kazanır. Sayısal değerlere sahip veriler bu haliyle bilgisayar ortamlarında işlenebilmektedir. Şekil 2.1.'de analog bir sinyalden örnekler alarak sayısal bir sinyalin elde edilmesi gösterilmiştir.



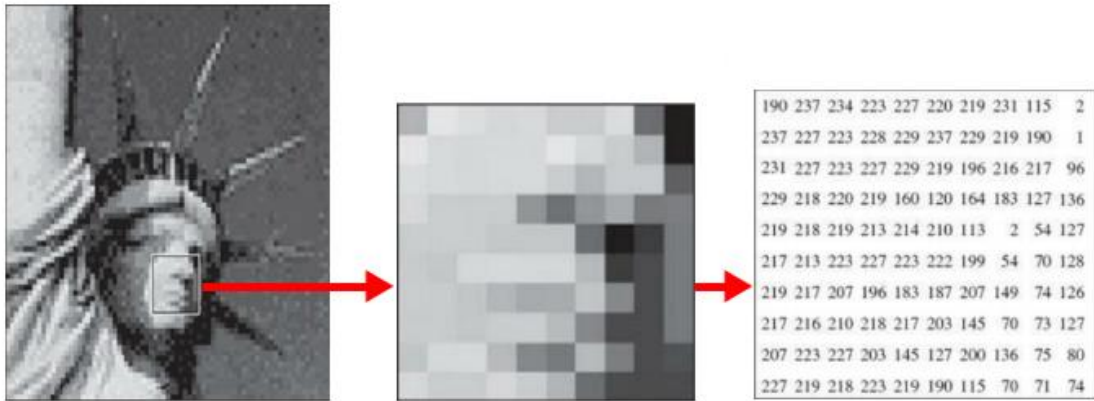
ŞEKİL 2.1. Analog Sinyalden Sayısal Sinyalin Oluşturulması

Bilgisayar ortamında bulunan bir sayısal görüntü, temel anlamda iki boyutlu matristen oluşmakta ve piksellerden meydana gelmektedir. Sayısal bir görüntü, $f(x,y)$ şeklinde bir fonksiyon olarak düşünülebilir. Burada x ve y değerleri iki boyutlu görüntü matris üzerindeki satır ve sütun koordinatlarını ifade eder. Her x ve y değeri için iki boyutlu düzlemde bir noktada piksel verisi bulunmaktadır. Pikseller, iki boyutlu görüntü matrisi üzerinde bir koordinat değerine sahiptir. Piksel verileri belirli bir anlam taşıyan sayısal değerlere sahiptirler. Bu veriler iki boyutlu düzlemde bir araya gelerek sayısal görüntü bütünü oluştururlar. [2] Şekil 2.2.'de bir sayısal görüntünün satır ve sütunlarındaki pikseller gösterilmiştir.



ŞEKİL 2.2. Görüntünün Satır ve Sütunlarında Bulunan Pikseller [3]

Bir sayısal görüntü, birçok piksel verisinin bir araya gelmesiyle oluşmaktadır. Sayısal görüntüdeki piksellerde parlaklık değerleri bilgisi tutulmaktadır. [2] [3] Şekil 2.3.'te görüntü üzerindeki bir alanda bulunan pikseller ve sayısal karşılıkları gösterilmiştir. Bu şekilde bir görüntü bilgisayar ortamında sayısal olarak temsil edilebilmektedir.

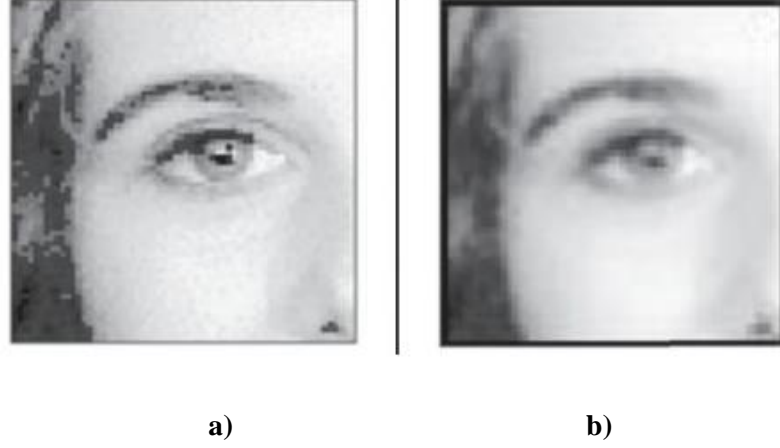


ŞEKİL 2.3. Görüntü Üzerindeki Pikseller ve Sayısal Karşılıkları [4]

2.1. Görüntü İşleme

Teknolojinin gelişmesi ve yaygınlaşmasıyla birlikte her gün devasa düzeyde sayısal görüntü verisi ortaya çıkmaktadır. Bu durum birçok alanda, görüntü üzerinde bazı işlemlerin gerçekleştirilmesiyle işlenmesi gereksinimini ortaya çıkarmıştır. Bir sayısal görüntü, iyileştirme, sınıflandırma, çıkarımında bulunma ve benzer hedefler doğrultusunda bilgisayar ortamında çeşitli

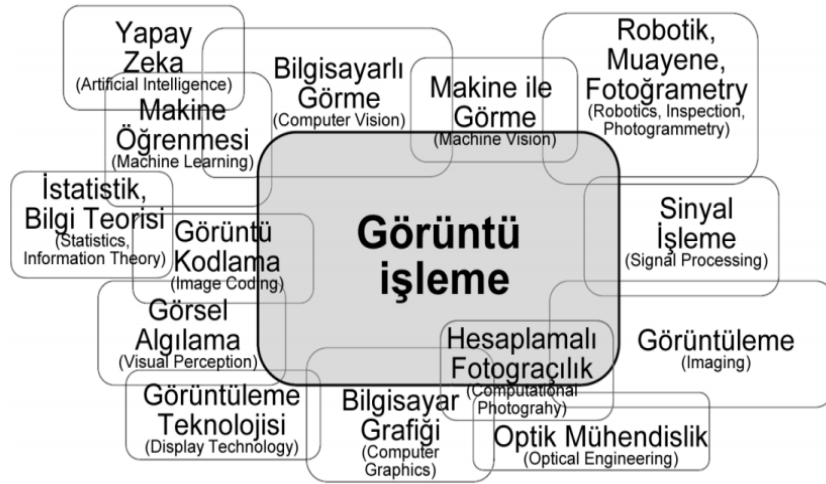
işlemler uygulanarak işlenebilmektedir. Bu işlemlerin tümü, sayısal görüntünün işlenmesi olarak ele alınmaktadır. Sayısal görüntünün işlenmesi hususu, özellikle bilgisayar bilimi ve mühendislik alanında önemli bir araştırma alanı olma statüsünü korumaktadır. [5] Şekil 2.4.'te bir görüntü ve işlenmiş hali gösterilmiştir.



ŞEKİL 2.4. a) Orijinal Görüntü b) Yumuşatılmış Görüntü (Gaussian Smoothing) [6]

2.2. Görüntü İşleme Uygulama Alanları

Görüntü işleme, günümüzde giderek popülerliği artan bir araştırma alanıdır. Birçok disiplin ile doğrudan ya da dolaylı olarak ilişkilidir. Görüntü işleme alanının makine görüşü, yapay zekâ, işaret işleme, makine öğrenmesi, derin öğrenme ve daha birçok farklı disiplin ve araştırma alanıyla dolaylı ve doğrudan ilişkisi vardır. [7] Şekil 2.5.'te görüntü işleme ve diğer disiplinler arası ilişki gösterilmiştir.



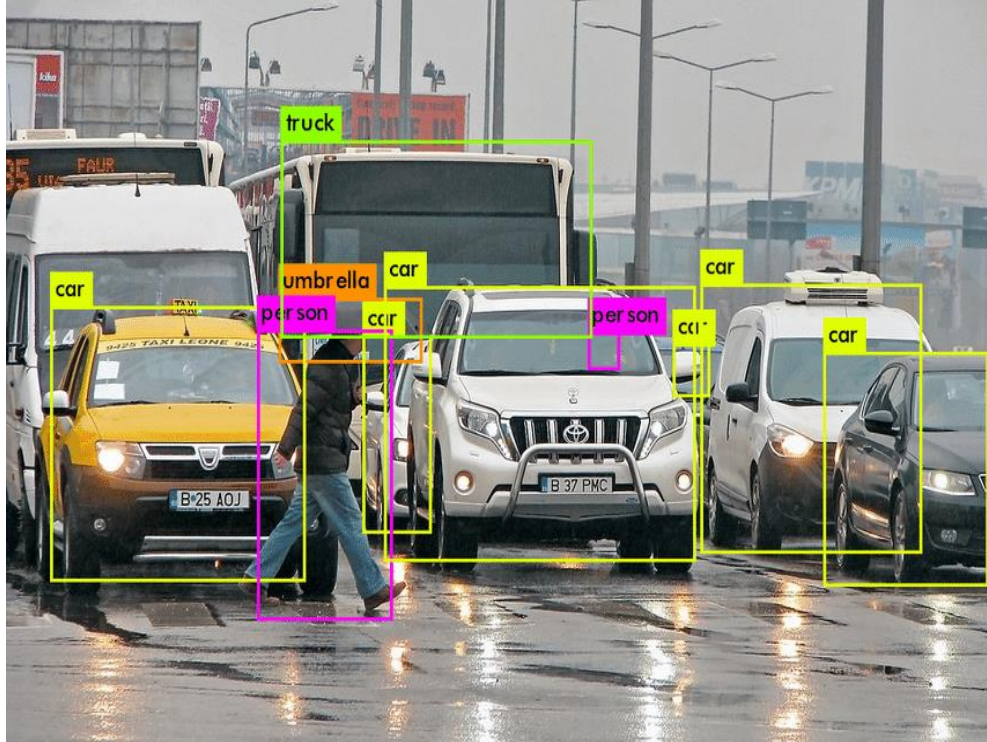
ŞEKİL 2.5. Görüntü İşlemenin Diğer Disiplinler Arası İlişkisi [7]

Gelişen ve yaygınlaşan teknoloji ve günümüz dünyasının ihtiyaçları doğrultusunda, görüntü işleme birçok alanda karşılık bulmaktadır. Sağlık, güvenlik, endüstriyel uygulamalarda görüntü işleme yöntemleri kullanılmaktadır. Görüntü işlemenin kullanıldığı bazı uygulamalar şu şekilde sıralanabilir: [8] [9] [10]

- Askeri uygulamalar,
- Yeraltı araştırmaları,
- Parmak izi işleme, yüz tanıma ve çeşitli güvenlik sistemleri,
- Nesne sınıflandırma ve tanıma,
- Tıbbi görüntüleme uygulamaları,
- Fotoğraf düzenleme yazılımları, görüntülerin onarımı,
- Uzay araştırmaları,
- Makine görüşü

3. GÖRÜNTÜ ÜZERİNDE İSTENİLEN NESNENİN BULUNMASI

Nesne tespiti, görüntüdeki ya da videodaki nesneleri sınıflandırma, tanımlama ve anlamlandırma işlemidir. Aynı zamanda görüntü işleme alanının da önemli bir araştırma konusudur. Günümüzde görüntüdeki nesnenin tespiti uygulamaları eğlence, güvenlik, savunma, sağlık gibi birçok alanda kullanılmaktadır. Bu uygulamalara örnek olarak plaka tanıma sistemleri, yüz tanıma sistemleri, parmak izi algılayıcıları, tıbbi görüntülemede teşhis sistemleri gösterilebilir.



ŞEKİL 3.1. Görüntü Üzerinde Nesnelerin Tespit Edilmesi [11]

Görüntüde hedef nesnenin tespit edilmesi işlemi için birçok yöntem önerilmiştir. Bu yöntemlerden bazıları sonraki alt başlıklarda incelenmiştir.

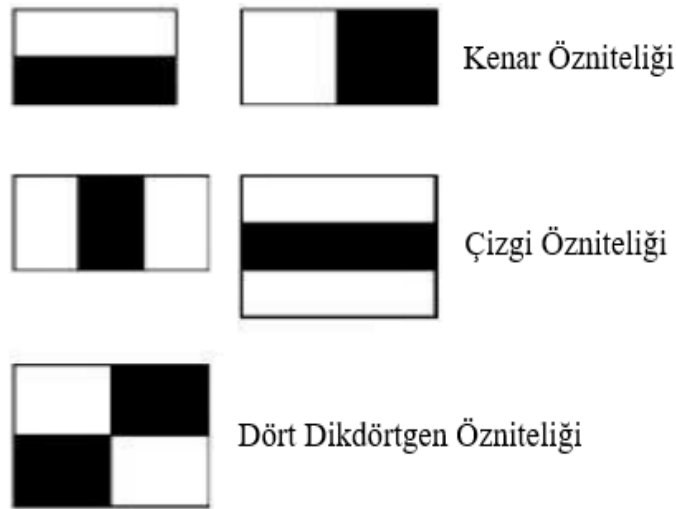
3.1. Makine Öğrenimi İle Nesne Tespit Yöntemleri

Makine öğrenmesi, bilgisayarların var olan verilerden çıkarımlarıyla belirli algoritma ve yöntemleri kullanarak öğrenme ve öğrenimleri işleme süreçlerini kapsamaktadır. Günümüzde ilgi gören bir araştırma alanı olma statüsünü korumaktadır. Makine öğrenmesi kavramı ilk kez 1959 yılında Arthur Samuel tarafından bilgisayarlara öğrenme becerisi kazandırma çabaları olarak kullanılmıştır. [12] Makine öğrenmesi, birçok alanda olduğu gibi nesnelerin tespitinde de

kullanılmaktadır. Nesnenin tespit edilebilmesi için nesneye ait görüntüler ile bir makine öğrenmesi süreci gerçekleştirilir. Bu başlıkta Haar-Benzeri Öznitelikler Bazlı Nesne Tespiti ve Histogram of Oriented Gradients (HOG) ile Nesne Tespiti yöntemleri incelenmiştir. Bu yöntemlerin çalışma prensipleri açıklanmıştır.

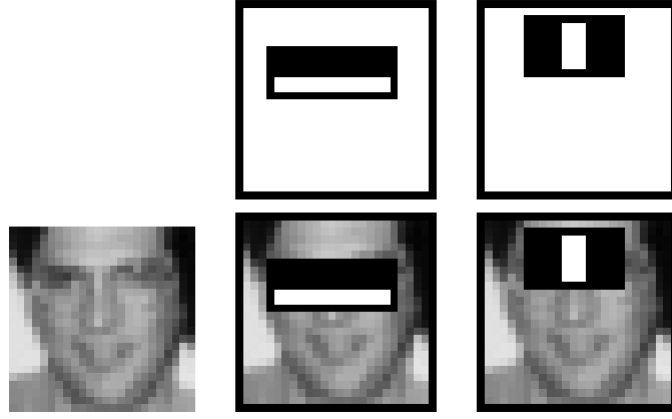
3.1.1. Haar-Benzeri Öznitelikler Bazlı Nesne Tespiti

Haar-Benzeri Öznitelikler (Haar-Like Features) bazlı basamaklı sınıflandırıcılar kullanarak nesnenin tespiti yöntemi Paul Viola ve Michael Jones tarafından 2001 yılında önerilmiştir. Yöntem, Haar-Benzeri Öznitelikler ile nesne tespiti gerçekleştirir. Haar-Benzeri Öznitelikler, görüntü üzerinde nesnenin tespitinin yapılması için kullanılan ayırt edici özniteliklerdir. Bu öznitelikler kullanılarak dikdörtgen bölgeler arasındaki yoğunluk farkına göre çıkarımlar gerçekleştirilir. [13] [14] Şekil 3.2’de kenar özniteliği, çizgi özniteliği ve dört dikdörtgen öznitelikleri gösterilmiştir.



ŞEKİL 3.2. Haar-Benzeri Öznitelikler [15]

Yöntemde, belirli boyutlardaki (genelde 24x24) çerçeveler halinde görüntü analiz edilerek Haar-Benzeri Öznitelikler saptanır. Bu saptamalar belirli hesaplamalar gerçekleştirerek yapılır. [16] Şekil 3.3’te Haar-Benzeri Öznitelik saptamaları gösterilmiştir.



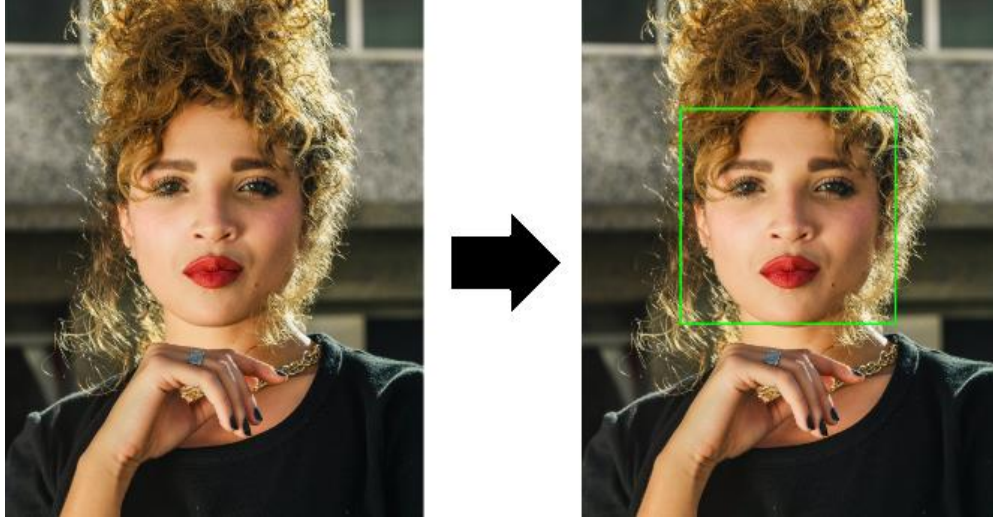
ŞEKİL 3.3. Haar-Benzeri Öznitelik Saptamaları [16]

Küçük alt çerçeveler halinde görüntünün analizi düşünüldüğünde, görüntü üzerinde çok fazla sayıda öznitelik çıkarımının bulunacağı söz edilebilir. Bu durumda hesaplamalar için eşik değerlerin belirlenmesi ve en iyi çıkarımların gerçekleşmesi hedeflenir. Hedef nesnenin bulunduğu pozitif görüntüler ve hedef nesnenin görüntü içeriğinde hiçbir şekilde bulunmadığı negatif görüntüler ile bir sınıflandırma eğitimi gerçekleştirilir. [16] Şekil 3.4'te yüz tanıma amacıyla gerçekleştirilecek eğitim için farklı insan yüzlerinin açıkça görüldüğü ve sınırlandırılmış pozitif görüntü kümesi örneği yer almaktadır.



ŞEKİL 3.4. Yüz Tanıma Eğitimi İçin Pozitif Görüntü Kümesi [16]

Pozitif görüntüler ve negatif görüntülerden oluşan veri kümeleriyle gerçekleştirilen eğitim sonucundaki hesaplamalarla elde edilmiş değerler ile görüntü üzerinde hedeflenen nesnenin tespiti gerçekleştirilebilir. İnsan yüzlerinin bulunduğu pozitif görüntüler ile insan yüzlerinin bulunmadığı negatif görüntüler kullanılarak gerçekleştirilmiş bir eğitimin sonuçları kullanılarak görüntü üzerinde yüzün tespit edilmesi işlemi Şekil 3.5'te gösterilmiştir.



ŞEKİL 3.5. Yüz Tanıma Eğitimi Sonuçları Kullanılarak Görüntüdeki Yüzün Tespiti

3.1.2. Histogram of Oriented Gradients (HOG) İle Nesne Tespiti

Yöntem, ilk olarak Amnon Shashua ve arkadaşları ile Dalal Navneet ve arkadaşları tarafından önerilmiştir. Amnon Shashua ve arkadaşları, yöntemi trafikteki yayaların tanınmasına yönelik bir uygulamada önermiştir. [17] Dalal Navneet ve arkadaşları bu uygulamada kullanılan yöntemi çok daha karmaşık insan tespit problemleri için geliştirmişlerdir. [18] Histogram of Oriented Gradients ile nesne tanıma günümüzde nesne tanıma yöntemleri arasında popülerliğini korumaktadır. [19] Yöntem genel anlamda üç aşamadan oluşmaktadır. Bu aşamalar Şekil 3.6.'da gösterilmiştir.



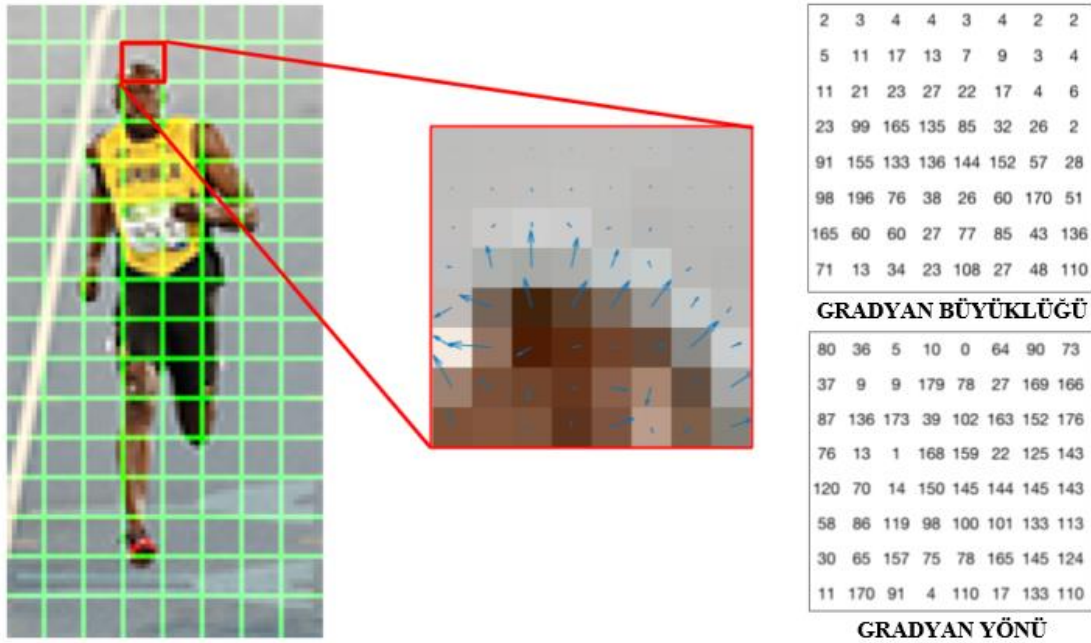
ŞEKİL 3.6. Histogram of Oriented Gradients Aşamaları

HOG Algoritması ile öncelikle yatay ve dikey gradyanlar hesaplanır. Gradyan büyüklüğü ($|G|$), Denklem 3.1. kullanılarak ve gradyan yönü (θ), Denklem 3.2. kullanılarak ilgili değerler ile hesaplanır. [20]

$$|G| = \sqrt{I_x^2 + I_y^2} \quad (3.1)$$

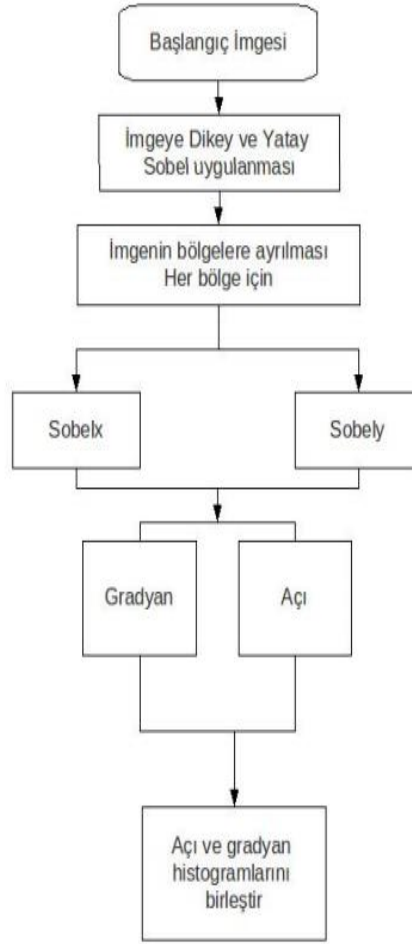
$$\theta = \arctan \frac{I_x}{I_y} \quad (3.2)$$

Bu denklemlerde I imgeyi, x ve y ise matrisin satır ve sütun değerlerini ifade etmektedir. Gradyan hesaplamaları yapıldıktan sonra, görüntü ızgaralara ayrılarak hücreler halinde gradyanların histogramları hesaplanır. Her hücredeki piksellerin gradyan büyüklükleri ve gradyan yönlerinin değer karşılıkları bulunmaktadır. [20] Görüntünün hücelere ayrılmış hali, bir hücrenin gradyan büyüklüğü ve yönü şekil 3.7’de gösterilmiştir.



ŞEKİL 3.7. Hücelere Bölünmüş Görüntüde Bir Hücrenin Gradyan Büyüklükleri ve Yön Değerleri [21]

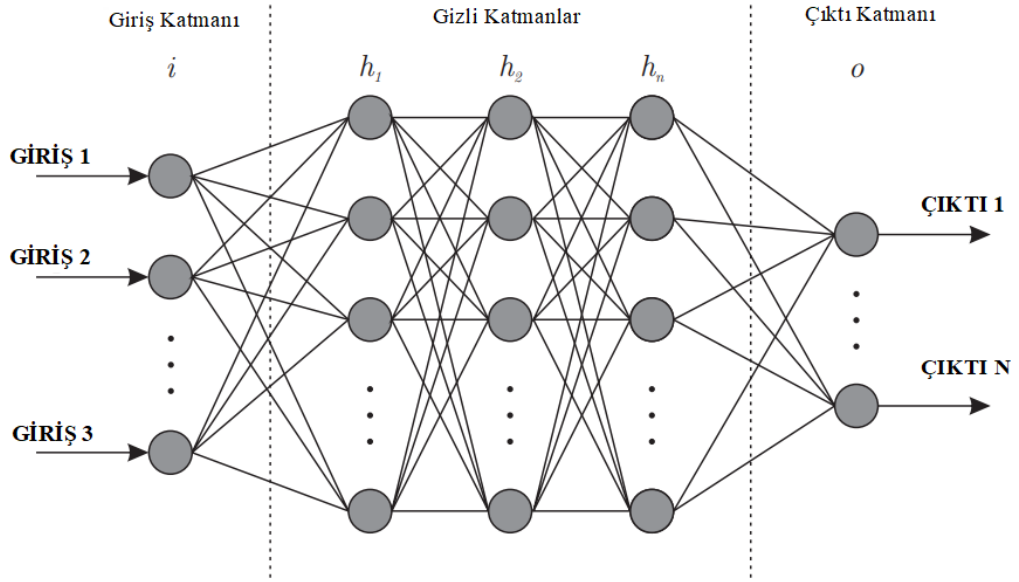
Histogram hesaplamalarının ardından son olarak bloklar halinde bir normalizasyon işlemi gerçekleştirilir. [20] Şekil 3.8.’de Histogram of Oriented Gradients algoritmasının işleyişi için bir akış şeması verilmiştir.



ŞEKİL 3.8. Histogram of Oriented Gradients Algoritmasının Akış Şeması [19]

3.2. Derin Öğrenme İle Nesne Tespit Yöntemleri

Derin öğrenme, son yıllarda oldukça popüler olan bir araştırma alanıdır. Görüntü işleme ve ses işleme başta olmak üzere birçok alanda derin öğrenme yöntemleri kullanılmaktadır. Derin öğrenme, gerçek dünyadan var olan veriler ile bir yapay zekanın eğitilmesi olarak ifade edilebilir. Derin öğrenmede yapay sinir ağları (YSA) kullanılmaktadır. Yapay sinir ağlarındaki düğüm hücreleri, insan beynindeki nöronlar olarak düşünülebilir. Bu hücreler ağırlıklandırılmış bağlantılar ile birbirilerine bağlıdır. Yapay sinir ağlarındaki hücreler üç katmanda yer alırlar. Bu katmanlar sırasıyla giriş katmanı, derinliği değişkenlik gösteren gizli katmanlar ve çıktı katmanıdır. [22] [23] Bir yapay sinir ağının yapısı, katmanları ve hücreler ile bağlantıları Şekil 3.9.'da gösterilmiştir.

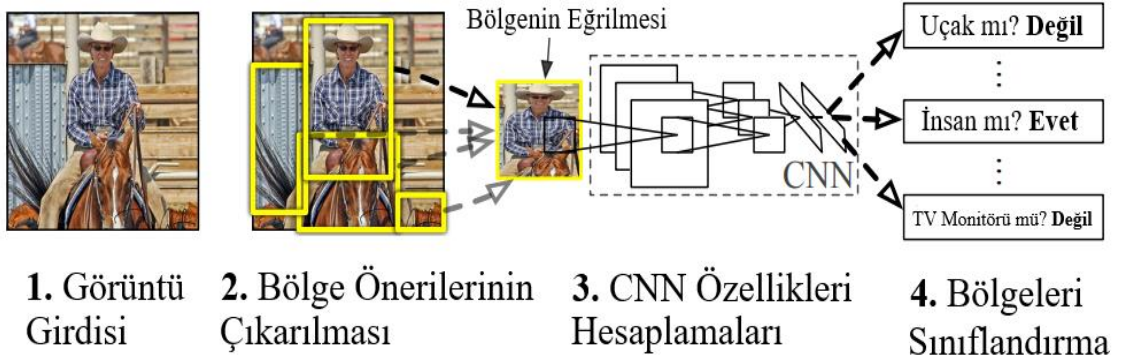


ŞEKİL 3.9. Yapay Sinir Ağlarının Mimarisi [23]

Yapay sinir ağlarında birbiri ardına her seviyenin giriş verisi bir önceki seviyenin çıktı verisinden oluşmaktadır. Her hücre bağlantısının bir ağırlık değeri bulunmaktadır. Yapay sinir ağları, derin öğrenmenin en önemli bileşenidir. Derin öğrenme alanında birçok mimari bulunmaktadır. Bunlar, Evrişimsel Sinir Ağları (CNN), Tekrarlayan Sinir Ağları (RRN), Uzun Kısa Vadeli Hafıza Ağları (LSTM), Sınırlı Boltzmann Makineleri (RBM), Derin İnanç Ağları (DBN) olarak sıralanabilir. [22] Aşağıdaki alt başlıklarda derin öğrenme yöntemlerinden evrişimsel sinir ağları temelli R-CNN ve YOLO nesne tespit yöntemleri incelenmiştir.

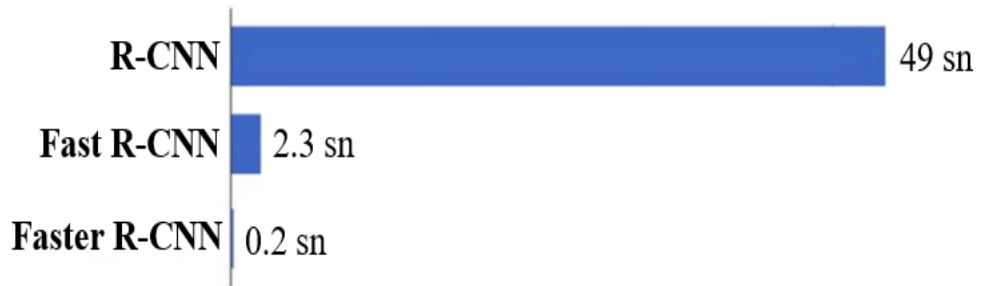
3.2.1. Bölgesel Tabanlı Evrişimsel Sinir Ağı (R-CNN) İle Nesne Tespiti

Bölgesel Tabanlı Evrişimsel Sinir Ağı (Regions with Convolutional Neural Networks) ile Nesne Tespiti 2014 yılında Ross Girshick ve arkadaşları tarafından ortaya konulmuştur. Yöntemde evrişimsel sinir ağları (CNN) derin öğrenme mimarisi kullanılmaktadır. Bölgesel Tabanlı Evrişimsel Sinir Ağı ile Nesne Tespitinin farklı yöntemleri vardır. Bu yöntemler geleneksel R-CNN, Fast R-CNN ve Faster R-CNN olarak sıralanabilir. R-CNN yönteminde görsel, bölge tavsiyelerine ayrılır. Bölge tavsiyeleri tespit için aday bölgelerin veri kümesini oluşturur. Bölgelere ayrı ayrı CNN hesaplamaları uygulanarak görüntü sınıflandırması gerçekleştirilir. [24] [25] Şekil 3.10'da yöntemin aşamaları gösterilmiştir.



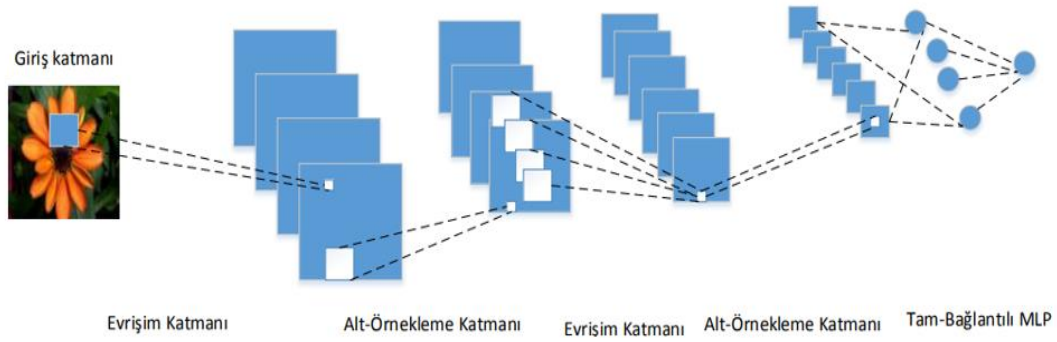
ŞEKİL 3.10. R-CNN Aşamaları [25]

Şekil 3.10.'da görüleceği üzere bölge önerileri eğrilerek kare haliyle işlem görür. Bu kareler, 64x64 boyutundadır. Bölge önerilerinin bulunması aşamasında yaklaşık 2000 adet bölge önerisi çıkarılır. 2000 adet 64x64 (4096) boyutlu bölge tavsiyesi bulunacağından CNN hesaplamaları uzun zaman tüketimi gerektirir. Daha az zaman tüketimi için Fast R-CNN ve Faster R-CNN yöntemleri önerilmiştir. [24] [25] Şekil 3.11'de R-CNN yöntemlerin zaman tüketimi karşılaştırmaları gösterilmiştir. Zaman tüketim değerleri için real-time uygulamalarda R-CNN ve Fast-CNN yöntemlerinin uygun olmadığı çıkarımında bulunulabilir.



ŞEKİL 3.11. R-CNN Zaman Tüketimi Karşılaştırmaları

Evrişimsel sinir ağının (CNN) yapısı Şekil 3.12.'de gösterilmiştir. Görüntü alt örneklemeleri alınarak evrişim katmanında konvülasyon işlemleri uygulanır. Bu işlemler ile evrişim katmanında çıktı öznitelik haritaları elde edilerek öznitelik çıkarımları gerçekleştirilir. [26]

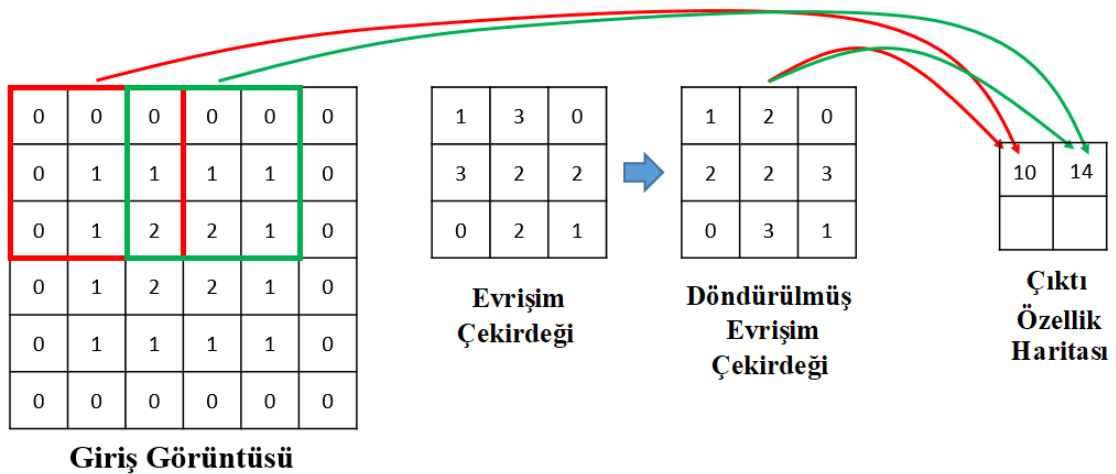


ŞEKİL 3.12. Evrişimsel Sinir Ağı Tasviri [27]

Evrişim katmanında iki boyutlu giriş görüntüsü görüntüden daha küçük çerçeveler halinde, filtreler ile ayrı ayrı konvolüsyon işleminden geçirilerek yeni çıktı matrisleri elde edilir. Bu matrisler çıktı özellik haritalarını oluşturur. [26] Evrişim katmanındaki işlemin denklemi, ‘I’ görüntüyü, ‘K’ çekirdeği ve ‘*’ işlemi konvolüsyon işlemi ifade etmek üzere Denklem 3.3’teki gibi gösterilebilir.

$$S(i, j) = (I * K)(i, j) = \sum_x^m \sum_y^n I(x, y) K(i - x, j - y) \quad (3.3)$$

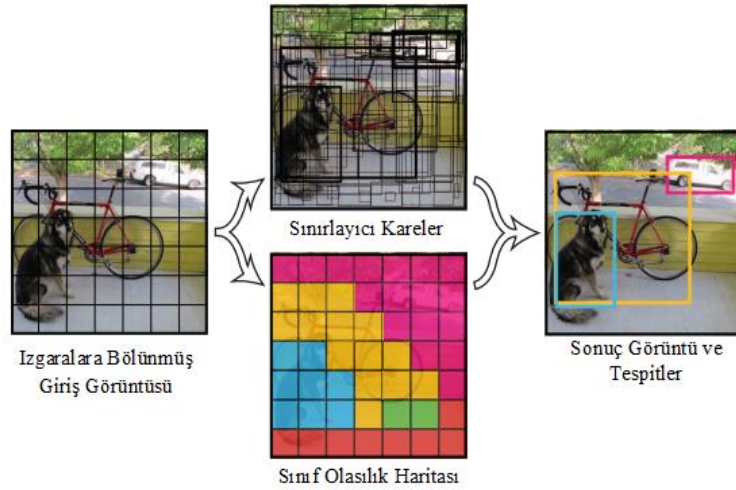
Denklem 3.3 ile gösterilen giriş görüntüsü ve x/y eksenlerinde simetriği alınarak döndürülmüş evrişim çekirdeği ile evrişim katmanında gerçekleştirilen konvolüsyon işlemi Şekil 3.13.’te örneklenmiştir.



ŞEKİL 3.13. Konvolüsyon İşlemi

3.2.2. You Only Look Once (YOLO) İle Nesne Tespiti

You Only Look Once (YOLO) İle Nesne Tespiti Joseph Redmon ve arkadaşları tarafından 2016 yılında yayınlanan bir makale ile önerilmiştir. Yöntemde derin öğrenme mimarilerinden evrimsel sinir ağı (CNN) yapısı kullanılmaktadır. Bu yöntemin en belirgin özelliği çok hızlı çalışması ve gerçek zamanlı uygulamalar için kullanışlı olmasıdır. İsmi, görüntüdeki nesnelerin tespiti için tam bir görüntüye yalnızca bir defa bakılıyor olmasından almaktadır. Algoritma ile görüntü ızgaralar halinde ayrılır (genelde 19x19 adet kutu) ve sınırlanan kutuların nesne olasılıkları tahmin edilir. [28] Şekil 3.14.'de giriş görüntüsünün ızgaralara bölünmüş hali, sınırlayıcı kutular ve sınıflandırma olasılık haritası gösterilmiştir.



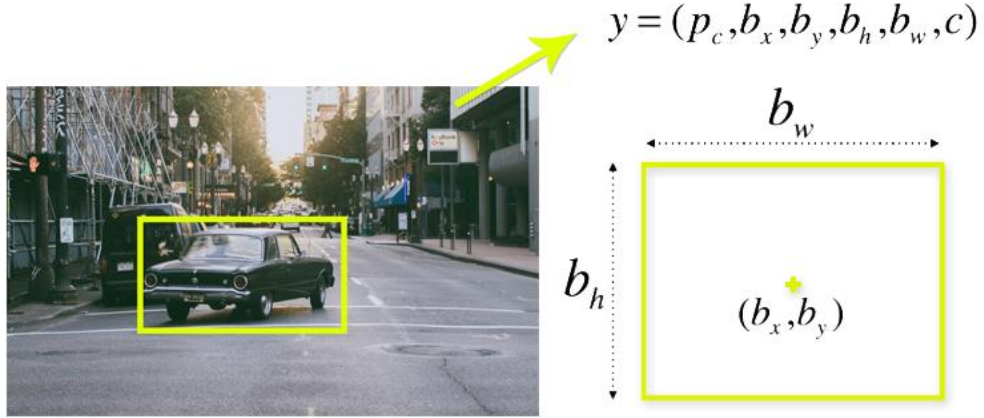
ŞEKİL 3.14. YOLO Algoritmasının İşleyişi [28]

Şekil 3.12.'de görülebileceği üzere giriş görüntüsü ızgaralara ayrılarak sınıf olasılık haritası çıkarılır. Evrimsel sinir ağında tam bir görüntü değerlendirilerek, sınırlayıcı karelerin sınıflandırma olasılıkları tahmin edilir ve ağırlıklandırılır. Bir CNN'e her ızgarada 5 adet kutu bulunan 19x19 ızgara gönderilir. [28] Şekil 3.15.'te YOLO yönteminin uygulanması gösterilmiştir.



ŞEKİL 3.15. You Only Look Once Yönteminin Uygulanması [28]

Şekil 3.16’da görsel üzerinde tahmin bölgesi gösterilmiştir. p_c sınırlanan bölgenin nesne içerme olasılığını ifade etmektedir. b_x , b_y sınırlı bölgenin merkezinin x ve y koordinatlarını ifade etmektedir. b_h sınırlanan bölgenin yüksekliğini, b_w sınırlanan bölgenin genişliğini ifade eder. c değeri bir sınıfa (araba, insan vb.) karşılık gelen değere sahiptir. [29]

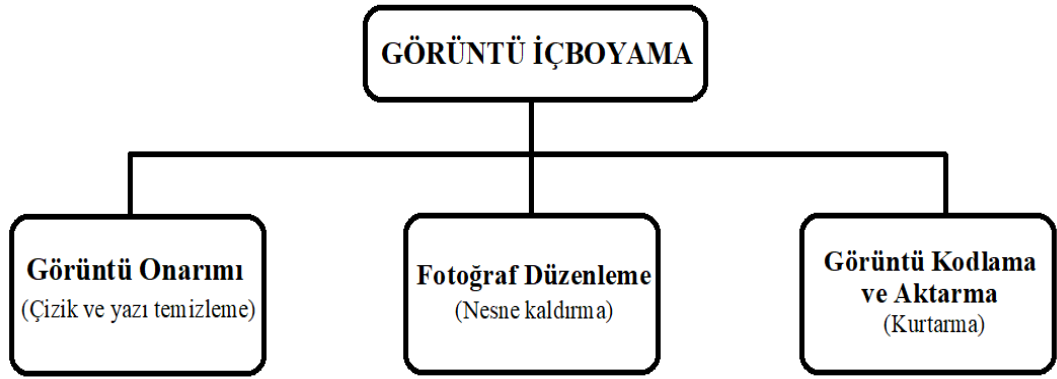


ŞEKİL 3.16. Tahmin Bölgesi ve Sınırlanması [29]

Tahmin bölgelerinin üst üste gelmesi durumunda tahmin bölgelerinden p_c değeri maksimum olmayan sınırlanmış bölgeler elenir. [29] Şekil 3.16.’te sınırlanan alanlar ve alanlar için sınıflandırmalarla birlikte p_c olasılık değerleri gösterilmiştir.

4. GÖRÜNTÜ ÜZERİNDE BELİRLENMİŞ BÖLGELERİN İÇBOYAMA İŞLEMİ İLE TAMAMLANMASI

Bir görüntü üzerinde belirlenmiş olan bir bölgenin, ilgili bölgeye uyumlu bir biçimde tamamlanması işlemine içboyama denilmektedir. İçboyama işlemi ile görüntünün bütünlüğüne uygun olarak bir doldurma işleminin gerçekleştirilmesi hedeflenir. İçboyama yaygın olarak eskimiş görüntülerdeki eksik ya da yıpranmış bölgelerin onarılmasında, görüntülerdeki hatalı ve kayıp bölgelerin giderilmesinde, görüntüde istenmeyen kısımların çıkarılmasında, görüntünün leke ve benzeri bozulmalardan arındırılarak iyileştirilmesinde kullanılmaktadır. Günümüzdeki birçok fotoğraf düzenleme uygulamalarında da içboyama yöntemleri kullanılmaktadır. Şekil 4.1.'de görüntü içboyama uygulamaları gösterilmiştir. [30]



ŞEKİL 4.1. Görüntü İçboyama Uygulamaları [30]

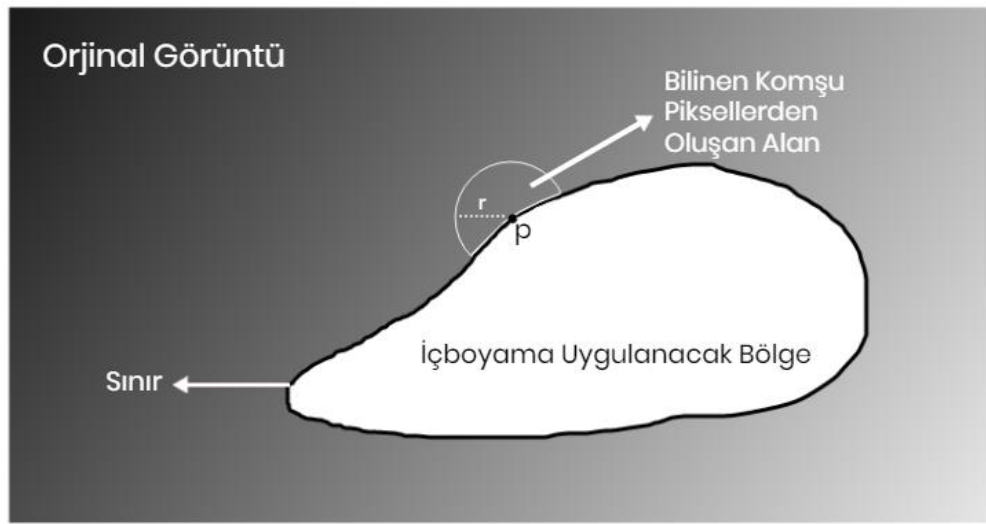
İçboyama işlemi için birçok yaklaşım önerilmiştir. Bu yaklaşımlarla görüntüdeki belirlenen bölgelerin görüntüye uyumlu olarak doldurulması işlemi birçok yöntem ile gerçekleştirilebilmektedir.

4.1. Geleneksel İçboyama Yöntemleri

İçboyama işlemi için kullanılan geleneksel yöntemler, doldurulmak istenilen bölgeye bilinen görüntü verisini kullanarak ve yayarak doldurma yaklaşımı ile içboyama işlemini gerçekleştirmektedir. [30] Aşağıdaki alt başlıklarda geleneksel yöntemlerden hızlı yürüyüş tekniği bazlı içboyama, kısmi diferansiyel denklemler bazlı içboyama ve örnek bazlı içboyama yöntemleri incelenmiştir.

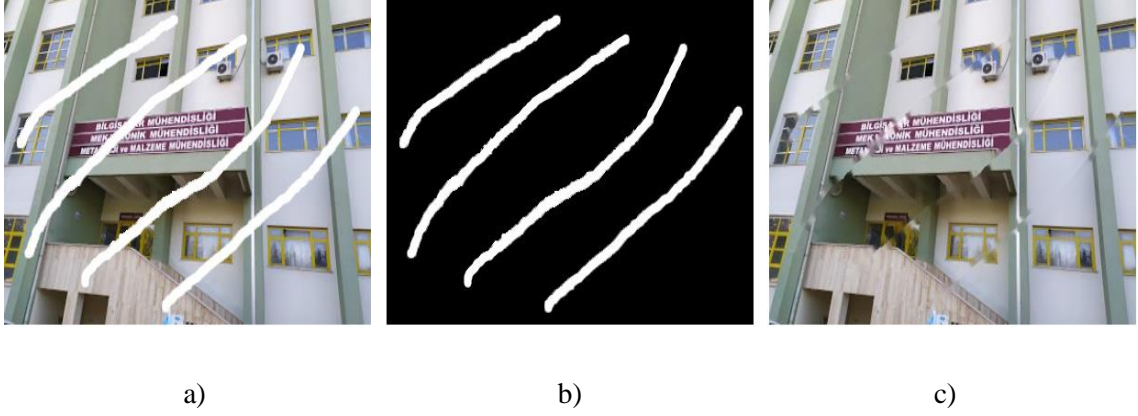
4.1.1. Hızlı Yürüyüş Tekniği (FMM) Bazlı İçboyama

Hızlı Yürüyüş Tekniğine dayalı içboyama, 2004 yılında Alexandru Telea tarafından yayımlanan “An Image Inpainting Technique Based on the Fast Marching Method” isimli makalede önerilmiştir. [31] Bu teknikte, içboyama uygulanacak bölgenin, çevresindeki bilgiye dayalı doldurulması yaklaşımı uygulanmaktadır. İçboyama işlemi, işlemin uygulanacağı bölgenin sınırlarında bulunan ve bilinen komşu piksellere göre gerçekleştirilmektedir. Şekil 4.2.’de boyanması gereken p noktasının çevresindeki, bilinen piksellerden oluşan r uzaklığına sahip alan gösterilmiştir.



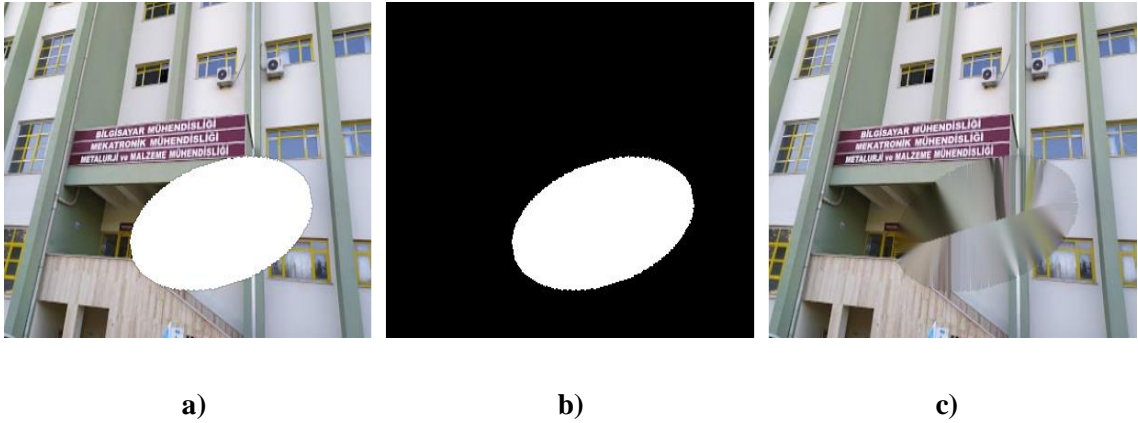
ŞEKİL 4.2. Hızlı Yürüyüş Tekniği ile İçboyama İşleminde Boyanacak Piksel (p) ve Bilinen Komşu Piksellerden Oluşan Alan Gösterimi

Alan içinde kalan bilinen piksellerin normalleştirilmiş ağırlıklı toplamalarına göre, boyanması gereken pikselin değeri hesaplanır. İlgili noktanın değeri, hesaplanan bu değer ile değiştirilerek boyanır. Bu işlem Hızlı Yürüyüş Yöntemi ile sonraki boyanması gereken noktadan devam eder. Böylece bilinen piksellerin yakınında bulunan boyanması gereken piksellerden başlanarak en dıştan en içe doğru boyama işlemi gerçekleştirilmiş olur. Bilinen komşu piksellerden oluşan alanın yarıçap değeri büyüdükçe değerlendirilecek alan büyüyecektir. Bu sebeple r değeri büyüdükçe doldurulması gereken noktadan uzak bir alan değerlendirilecek ve böylelikle yarıçap büyüklüğü ile doğru oranlı olarak boyanacak nokta için bulunacak değer doğruluğundan uzaklaşılacaktır. [31] [32] Şekil 4.3.’te Hızlı Yürüyüş Tekniği ile gerçekleştirilmiş bir içboyama örneği gösterilmiştir.



ŞEKİL 4.3. Görüntü Üzerinde Hızlı Yürüyüş Tekniği Kullanılarak İçboyama İşleminin Uygulanması: **a)** Orijinal Görüntü, **b)** Maske, **c)** Sonuç Görüntü

Hızlı Yürüyüş Tekniği ile İçboyama Yöntemi, boyanacak bölgenin çevresinde bulunan bölgelerdeki bilgiye dayalı tamamlama yaklaşımı ile içboyama işlemi gerçekleştirmekte ve boyanması gereken büyük alanlar için görüntü bütünlüğünden uzak bulanık sonuçlar vermektedir. Bu nedenle, bu yöntem ile küçük alanlar için hızlı ve başarılı sonuçlar elde edilebilmektedir. Şekil 4.4.'te görüntü üzerindeki büyük bir alanın içboyama işlemi Hızlı Yürüyüş Tekniği uygulanarak gerçekleştirilmiştir.

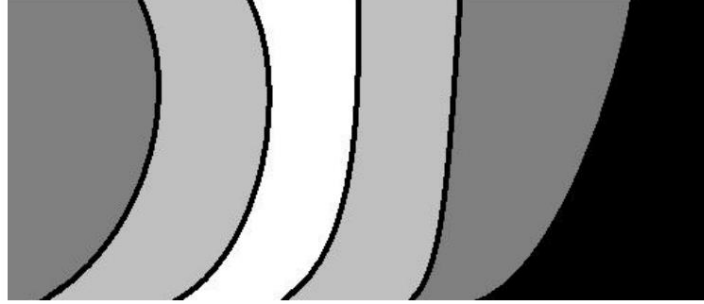


ŞEKİL 4.4. Görüntü Üzerinde Büyük Bir Alanın Hızlı Yürüyüş Tekniği ile İçboyama Yöntemi Uygulanarak Doldurulması: **a)** Orijinal Görüntü, **b)** Maske Görüntü, **c)** Sonuç Görüntü

Şekil 4.4.'te görülebileceği üzere içboyama uygulanan alan büyük seçildiğinden bu yöntem için bulanık bir sonuç elde edilmiştir.

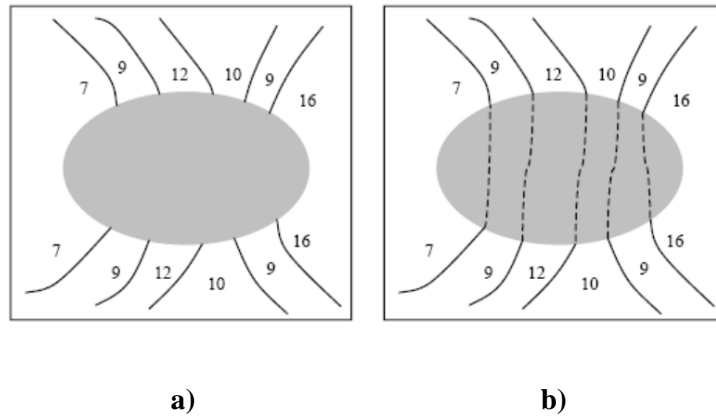
4.1.2. Kısmi Diferansiyel Denklemler (PDE) Bazlı İçboyama

Kısmi Diferansiyel Denklemler Bazlı İçboyama yöntemleri, içboyama uygulanacak bölgenin çevresindeki bilgiye dayalı olarak boyamanın gerçekleştirilmesi yaklaşımı ile kısmi diferansiyel denklemler kullanılarak eksik bölgeyi tamamlamaktadır. Görüntüdeki aynı parlaklığa sahip bölgeler izofot eğrileri ile belirlenir. Şekil 4.5.'te bir görüntüde aynı parlaklıktaki alanlar için izofot eğrileri gösterilmiştir.



ŞEKİL 4.5. Aynı Parlaklıktaki Bölgelerin İzofot Eğrileri [33]

İçboyama işlemi gerçekleştirilirken, içboyama uygulanacak alanın çevresindeki bilgi izofotlar boyunca eksik alan içine yayılır. Eksik bölge bu şekilde çevresel verilere dayanarak tamamlanmış olur. [33] Şekil 4.6'da izofot eğrilerinin içboyama uygulanacak eksik alan içine yayılımı gösterilmiştir.



ŞEKİL 4.6. İzofot Eğrilerinin Eksik Alan İçine Yayılımı (Sayılar, gri seviyeleri ifade etmektedir): **a)** İçboyama Uygulanacak Alan ve Çevresindeki İzofot Çizgileri, **b)** İçboyama Uygulanacak Alana İzofot Çizgilerinin Yayılımı [33]

Bertalmio ve arkadaşları, 2001 yılında yayımladıkları “Navier-Stokes, Fluid Dynamics, and Image and Video Inpainting” başlıklı makalede Navier-Stokes Denklemleri kullanarak görüntü ve video üzerinde içboyama yöntemi önermişlerdir. Bu yöntem kısmi diferansiyel denklemler bazlı bir içboyama tekniğidir. Navier-Stokes Denklemleri, fizikçi ve mühendis Claude-Louis Navier ve İrlandalı fizikçi ve matematikçi George G. Stokes tarafından akışkanların hareketlerini tanımlamak üzere ortaya konulmuştur. Navier-Stokes Denklemleri, doğrusal olmayan kısmi diferansiyel denklemlerdir. Bertalmio ve arkadaşları tarafından önerilen Navier-Stokes Denklemleri İle İçboyama Yöntemi esasen akışkanlar dinamiği ve uygulamalarına dayanır. Eksik bölge, içboyama uygulanacak bölgenin, çevresindeki bilgiye dayalı olarak tamamlanır. İzofotların içboyama uygulanacak bölgeye yayılımı, Navier-Stokes Denklemleri kullanılarak akışkanlar dinamiğine göre gerçekleştirilir. Bu şekilde, alanın çevresindeki bilgi ile içboyama uygulanacak bölge doldurulmuş olur. [34] [35] Şekil 4.7.’de Navier-Stokes Denklemleri ile İçboyama tekniğine dayalı bir içboyama örneği gösterilmiştir.



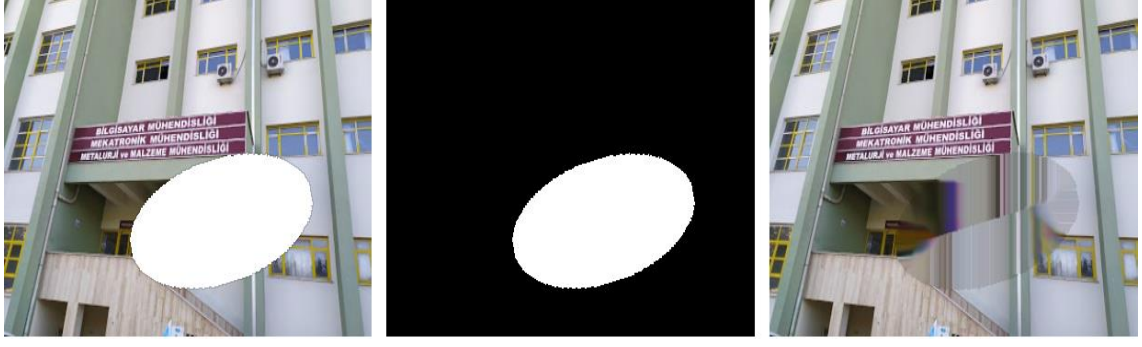
a)

b)

c)

ŞEKİL 4.7. Görüntü Üzerinde Navier-Stokes Denklemleri ile İçboyama Tekniği Kullanılarak İçboyama İşleminin Uygulanması: **a)** Orijinal Görüntü, **b)** Maske Görüntü, **c)** Sonuç Görüntü

Navier-Stokes Denklemleri ile İçboyama hızlı bir yöntem olmasına karşın boyanması gereken büyük alanlar için sağlıklı sonuçlar vermemektedir. Şekil 4.8.’de görüntü üzerindeki büyük bir alanın içboyama işlemi Navier-Stokes Denklemleri ile İçboyama uygulanarak gerçekleştirilmiştir. Şekilde görülebileceği gibi içboyama uygulanan alan için bulanık bir sonuç elde edilmiştir.



a)

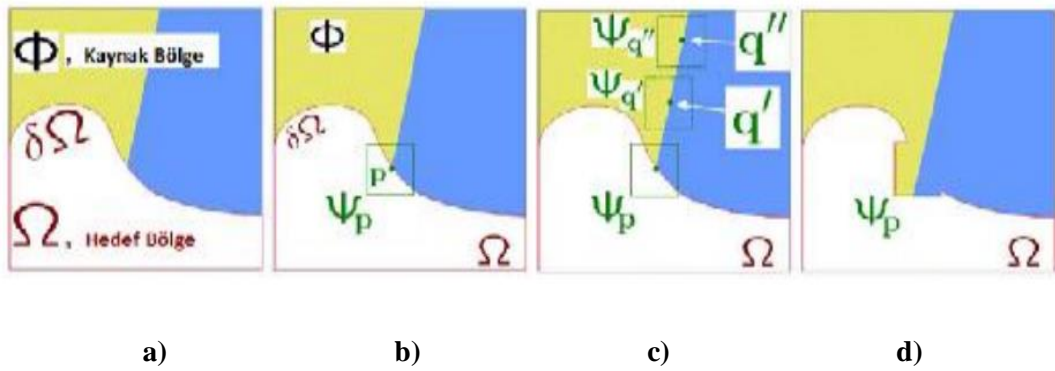
b)

c)

ŞEKİL 4.8. Görüntü Üzerinde Büyük Bir Alanın Navier-Stokes Denklemleri ile İçboyama Uygulanarak Doldurulması: **a)** Orijinal Görüntü, **b)** Maske Görüntü, **c)** Sonuç Görüntü

4.1.3. Örnek Bazlı İçboyama

Örnek Bazlı İçboyama Yöntemi, boyanacak bölgede her aşamada boyanması gereken daha küçük alanın tespit edilerek tespit edilen alan için uygun bir boyama işleminin gerçekleştirilmesi ve boyanacak bölgenin tümü ile dolduruluncaya kadar işlemin tekrarlanması yaklaşımı izlenmektedir. Örnek Bazlı İçboyama Yöntemi temel anlamda iki aşamalıdır. Bunlardan ilki önceliğin atanması, ikincisi ise öncelikli olan ilgili bölgeyi tamamlamak için en uygun parçanın belirlenmesi olarak sıralanabilir. Böylelikle boyanmak istenilen alan için, alan çevresindeki bilinen görüntüden parçalar seçilerek içboyama işlemi gerçekleştirilir. [36] Şekil 4.9.'da örnek tabanlı içboyama işleminin aşamaları gösterilmiştir.



a)

b)

c)

d)

ŞEKİL 4.9. Örnek Tabanlı İçboyama İşlemi Aşamaları: **a)** Orijinal Görüntü, **b)** Sınır Üzerinde ($\partial\Omega$) Yüksek Öncelikli Doldurulacak p Merkezli Alan, **c)** Doldurulacak Alan İçin Uygun Yamaların ($\Psi_{q'}$, $\Psi_{q''}$) Belirlenmesi, **d)** Uygun Yamanın Bölgeye Uygulanması [37]

Kısmi diferansiyel denklemler ile içboyama gibi küçük alanlara uygulandığında sağlıklı sonuçlar verebilen diğer bazı yöntemlerde büyük bölgelerin boyanması işlemlerinde gerçeklikten ve bütünlükten uzak görüntü sorunu ortaya çıkabilmektedir. Bu yöntem ile büyük bölgelerin doldurulması ile daha doğru sonuçlar elde edilmektedir. Bu bakımdan büyük bölgeler için elverişli bir yöntem olarak nitelendirilebilir. [36] Şekil 4.10.'da görüntü üzerinde belirlenen büyük bir alan için Örnek Tabanlı İçboyama Yöntemi ile alanın doldurulması işlemi gerçekleştirilmiştir. Görüntüdeki içboyama uygulanan alan, görüntünün bütünlüğüne uygun olarak doldurulmuştur.



a)

b)

c)

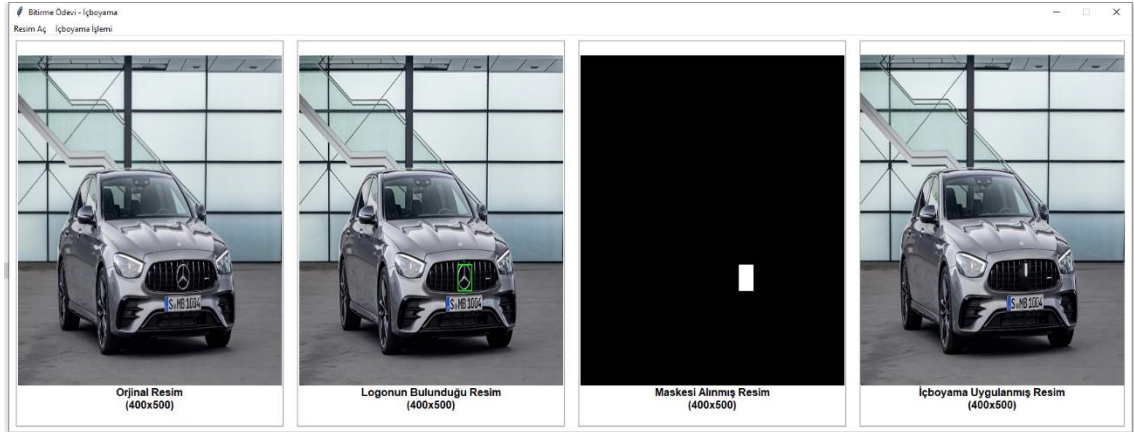
ŞEKİL 4.10. Görüntü Üzerinde Büyük Bir Alanın, Örnek Tabanlı İçboyama Yöntemi Uygulanarak Doldurulması: **a)** Doldurulmak İstenilen Beyaz Alan, **b)** Bölgenin Maskelenmesi, **c)** Örnek Bazlı İçboyama İşlemi Uygulanmış Sonuç Görüntü [36]

5. UYGULAMA

Önceki ana başlıklarda görüntü işleme ve sayısal görüntü kavramları ele alınmış, görüntüde istenilen nesnenin bulunması ve görüntüde belirlenen bölgenin içboyama uygulanarak doldurulması işlemleri için bazı yöntemler incelenmiştir. İncelenen bu tekniklerden Haar-Benzeri Öz nitelikler Bazlı Nesne Tespit Yöntemi ve Hızlı Yürüyüş Tekniği Bazlı İçboyama Yönteminden faydalanılarak bir uygulama geliştirilmiştir. Geliştirilen uygulamanın amaçları, işleyişi, yöntemleri ve sonuçları ile birlikte uygulamada kullanılan geliştirme dili ve geliştirme ortamı, veri kümeleri sonraki alt başlıklarda açıklanmıştır.

5.1. Uygulamanın Tanımı

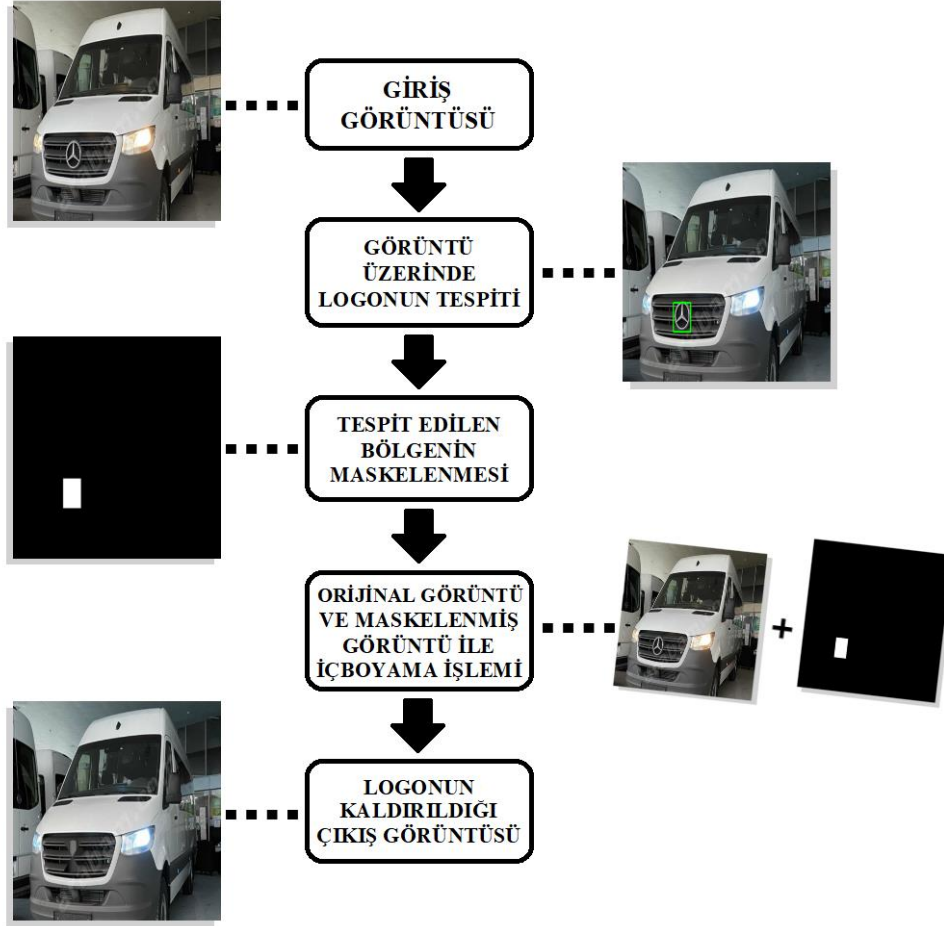
Görüntü üzerinde aracın logosunu tespit eden ve logoyu görüntüden kaldıran bir uygulama geliştirilmiştir. Uygulama için, görüntü işleme uygulamalarında yaygın olarak tercih edilen Python Programlama Dili ve PyCharm Geliştirme Ortamı tercih edilmiştir. Uygulamada içboyama ve nesne tespiti için OpenCV Kütüphanesinden faydalanılmıştır. Uygulamada Citroen, Honda, Mercedes, Opel ve Renault marka araçların logoları tespit edilebilmektedir. Program, herhangi bir boyuttaki görüntü verisini giriş olarak almaktadır. JPEG, PNG, BMP formatlarında kodlanmış giriş görüntülerini desteklemektedir. Programın grafiksel arayüzünden bir görünüm Şekil 5.1.'de gösterilmiştir.



ŞEKİL 5.1. Uygulamanın Grafiksel Arayüzü

Giriş görüntüsü alındıktan sonra uygulamanın çalışma prensibi genel anlamda 3 aşamadan oluşmaktadır. İlk aşamada görüntü, nesne tespit yöntemi ile analiz edilerek, görüntü içeriğindeki makine öğrenmesi ile öğretilmiş araç logolarının tespiti sağlanmaktadır. Tespit işleminin ardından tespit edilen bölgenin bilgileri alınarak ilgili bölge için bir maskeleme işlemi uygulanmaktadır.

Maskelenmiş görüntü çıkartıldıktan sonra orijinal görüntü ile birlikte işlenerek içboyama işlemi gerçekleştirilmektedir. Böylelikle görüntüde bulunan ilgili marka araçların üzerindeki logo, içboyama işlemi uygulanarak kaldırılmaktadır. Uygulamanın işleyiş şeması Şekil 5.2.'de gösterilmiştir.



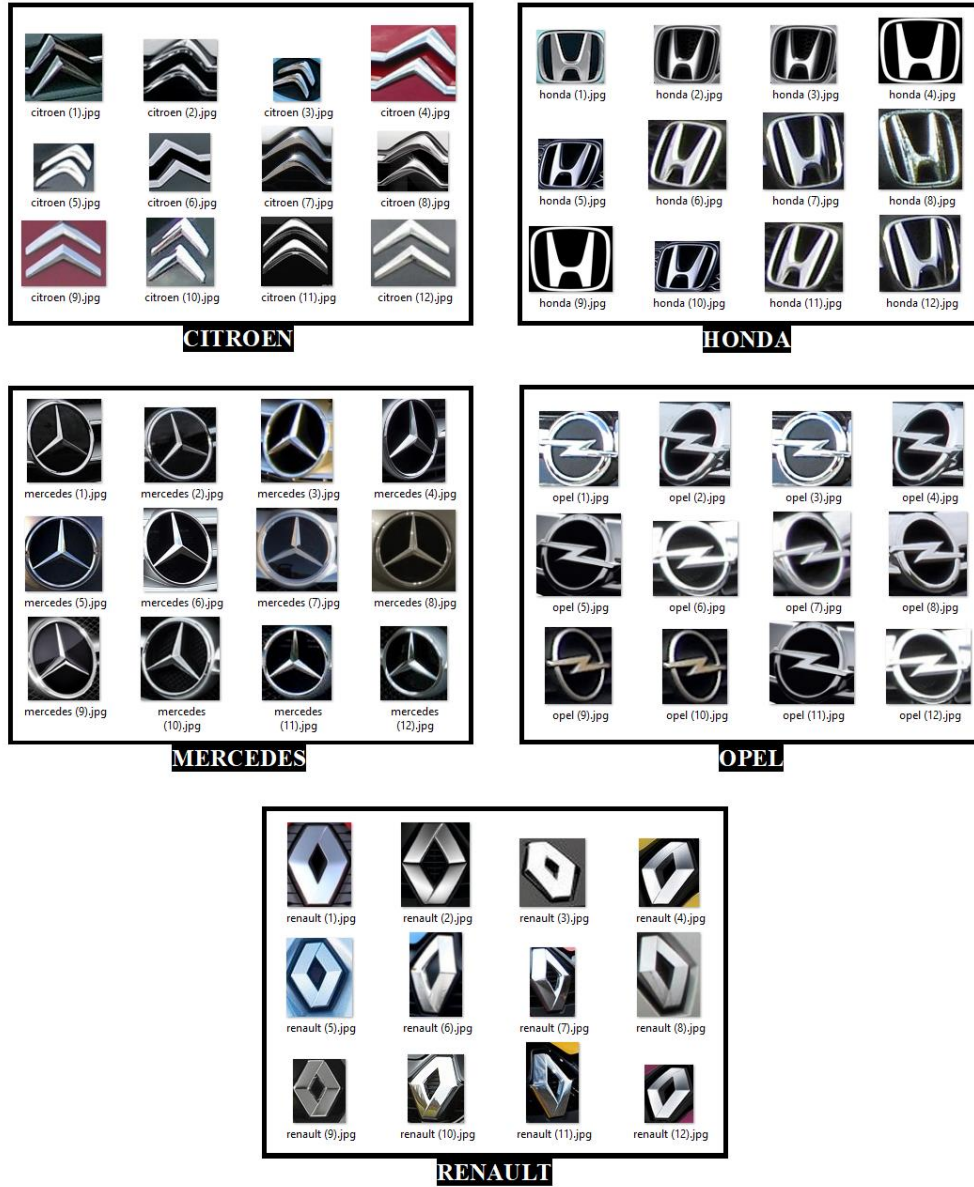
ŞEKİL 5.2. Uygulamanın İşleyiş Şeması

5.2. Görüntü Üzerinde Araç Logosunun Tespiti

Görüntü üzerinde araç logosunun tespit edilebilmesi için, her araç markası için veri kümeleri kullanılarak bir makine öğrenmesi gerçekleştirilmiştir. Eğitim sürecinin ardından, görüntüler üzerinde logoların bulunduğu bölgeler tespit edilerek tespit edilen bölgelerin, bölgeyi sınırlayan çerçeveler içine alınması sağlanmıştır. Sonraki alt başlıklarda uygulamanın eğitim süreci ve görüntü üzerinde logonun tespit edilme işlemi açıklanmıştır.

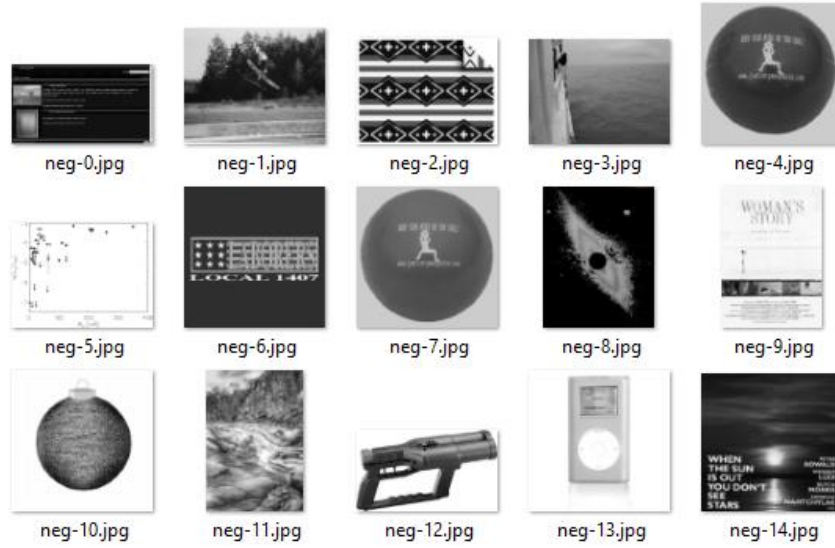
5.2.1. Eğitim Süreci

Uygulamada araç logolarının tespitinde 3.1.1. başlıkta incelenen Haar-Benzeri Öznitelikler Bazlı Nesne Tespit Yöntemi kullanılmıştır. Bu yöntemde, eğitim aşamasında her araç markası için araç logolarının bulunduğu pozitif görüntülerden oluşan veri kümeleri ve araç logolarının yer almadığı negatif görüntülerden oluşan veri kümeleri kullanılmıştır. Şekil 5.3.'te araç logolarının tanınması için gerçekleştirilen eğitimde kullanılan pozitif görüntülerden örnek görüntüler gösterilmiştir.



ŞEKİL 5.3. Araç Logolarının Bulunduğu Örnek Pozitif Görüntüler

Her araç markası için ayrı ayrı öğrenme işlemi gerçekleştirilmiştir. Böylelikle araç markaları sınıflandırılarak eğitim tamamlanmıştır. Şekil 5.3.'te gösterilen pozitif görüntü kümeleri için, araç logolarının bulunmadığı örnek negatif görüntüler Şekil 5.4.'te gösterilmiştir.



ŞEKİL 5.4. Araç Logolarının Bulunmadığı Örnek Negatif Görüntüler

Eğitim işleminde pozitif ve negatif görüntülerin belirli bir boyut bilgisi girilerek eğitim gerçekleştirileceğinden, kolaylıkla belirli boyut bilgilerinin girilebilmesi için tüm pozitif ve negatif görüntüler yeniden ölçeklendirilerek aynı boyutlara sahip görüntü verileri elde edilmiştir. Şekil 5.5.'te, verilen yolda bulunan pozitif görüntüleri 50x50 boyutunda yeniden ölçeklendirerek “pos” isimli dosyaya yazan metot gösterilmiştir.

```
def create_pos(dataset_path):
    datasetList=os.listdir(dataset_path)
    num=0
    for file in datasetList:
        img=cv.imread(dataset_path+"/"+file)
        img=cv.resize(img,(50,50))
        cv.imwrite("pos/"+str(num)+".jpg",img)
        num+=1
```

ŞEKİL 5.5. Pozitif Görüntüleri 50x50 Boyutunda Yeniden Ölçeklendiren Metot

Ölçeklendirme işlemi gerçekleştirildikten sonra Şekil 5.6.'da verilen metot ile, eğitim aşamasında kullanılacak pozitif veriler için verilerin bilgilerini içeren bir “info.dat” isimli dosya oluşturmaktadır. Bu metotta, “pos” klasöründen okunan pozitif görüntü verilerinin, “info.dat”

içine ayrı satırlarda olmak üzere verilerin bilgileri yazılma işlemi gerçekleştirilmektedir. Bu işlemi gerçekleştiren metot Şekil 5.6.'da gösterilmiştir.

```
def create_pos_info():
    posList=os.listdir("pos")
    for pos in posList:
        line="pos/"+pos+" 1 0 0 50 50\n"
        with open("info.dat","a") as f:
            f.write(line)
```

ŞEKİL 5.6. Pozitif Görüntü Verilerin Bilgilerini Listeleyen Metot

Şekil 5.6. ile gösterilen metot ile oluşturulan “info.dat” dosyasında her satırdaki bilgi içeriğinde boşluklarla ayrılmak üzere sırasıyla verinin yolu, veride kaç adet logonun bulunduğu bilgisi, logo için pencere başlangıç x ve y noktaları, görüntünün genişlik ve yükseklik değerleri yer almaktadır. Kullanılan pozitif görüntü verilerinin her biri, bir adet logo içerdiğinden nesne adet değeri “1” olarak girilmektedir. Pozitif görüntüler yalnızca logoyu içerdiğinden logonun bulunduğu pencere başlangıç noktası için değerler “0” ve “0” olarak girilmiştir. Pozitif görüntüler 50x50 boyutunda bulunduğundan boyut bilgisi için genişlik ve yükseklik değerleri “50” ve “50” olarak girilmiştir. Yeniden ölçeklendirme işlemi negatif görüntü verileri için de 100x100 boyutunda gerçekleştirilmektedir. Negatif görüntüler için yeniden ölçeklendirme işlemi yaparak, görüntü verilerini “neg” isimli dosyaya yazdıran metot ve yeniden boyutlandırılmış verilerin yol bilgisini içeren “bg.txt” dosyasını oluşturan metot Şekil 5.7.'de gösterilmiştir.

```
def create_neg(bg_paths):
    datasetList=os.listdir(bg_paths)
    num=0
    for file in datasetList:
        img=cv.imread(bg_paths+"/"+file,cv.IMREAD_GRAYSCALE)
        img=cv.resize(img,(100,100))
        cv.imwrite("neg/neg-"+str(num)+".jpg",img)
        num+=1

def create_bg():
    negFileList=os.listdir("neg")
    for negFile in negFileList:
        line="neg/"+negFile+"\n"
        with open("bg.txt","a") as f:
            f.write(line)
```

ŞEKİL 5.7. Negatif Görüntüleri 100x100 Boyutunda Yeniden Ölçeklendiren Metot ve Negatif Görüntü Verilerin Bilgilerini Listeleyen Metot

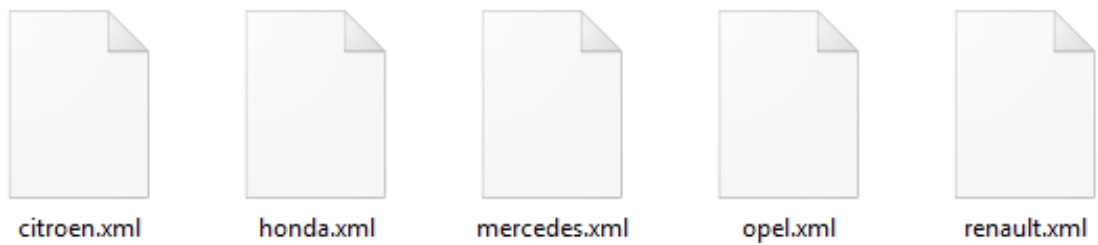
OpenCV Kütüphanesinde bulunan “opencv_createsamples.exe” ve “opencv_traincascade.exe” programları ile eğitim işlemi gerçekleştirilebilmektedir. Eğitimi başlatmak için gereken komut ve parametreler, Şekil 5.8.’de gösterildiği şekilde komut ve parametreler, CMD ekranından girilerek ya da .bat dosyası halinde yürütülerek eğitim işlemi başlatılabilir.

```
opencv_createsamples.exe -info info.dat
-vec positives.vec -num 700 -w 20 -h 20

opencv_traincascade.exe -data data
-vec positives.vec -bg bg.txt -numPos 1400 -numNeg 700 -numStages 10 -w 20 -h 20
```

ŞEKİL 5.8. Eğitimi Başlatmak İçin Gereken Komut ve Parametreler

Pozitif görüntülerin eğitimde kullanılabilmesi için OpenCV kütüphanesinde bulunan “opencv_createsamples.exe” yürütülerek pozitif görüntülerin bilgisinin bulunduğu “info.dat” parametre olarak gönderilir. Daha sonra OpenCV Kütüphanesinde bulunan “opencv_traincascade.exe” ile eğitim gerçekleştirilir. “-data data” komutu, eğitim çıktılarının kaydedileceği klasör bilgisini gönderir. “-vec positives.vec” komutu ile bir önceki pozitif görüntüler üzerinde gerçekleştirilen işlemin çıktısı eğitim için giriş verisi olarak gönderilir. “-bg bg.txt” komutu ile Şekil 5.7.’de gösterilen işlemde oluşturulan negatif görüntülerin yol bilgilerini içeren “bg.txt” dosyası parametre olarak gönderilir. Pozitif ve negatif görüntülerin adet bilgileri ve eğitim aşama sayısı gönderilerek eğitim işlemine başlanır. Eğitimin sonunda .xml uzantılı eğitim çıktıları alınır. Aynı ayrı, araç markaları için pozitif ve negatif görüntüler kullanarak eğitimin çıktıları sonucunda elde edilen .xml uzantılı eğitim verileri Şekil 5.9’da gösterilmiştir.



ŞEKİL 5.9. Eğitim Sonucunda Elde Edilen .xml Uzantılı Eğitim Çıktıları

Eğitimin çıktıları elde edildikten sonra logonun tespiti, Cascade Sınıflandırıcısı kullanarak gerçekleştirilebilir.

5.2.2. Logonun Tespiti

Eğitim sürecinin ardından her araç sınıfı için ayrı ayrı xml formatında eğitim çıktıları elde edilmiştir. xml formatındaki eğitim verileri kullanılarak görüntüdeki logonun tespiti işlemi gerçekleştirilebilmektedir. Şekil 5.10.'da "renault.jpg" isimli görüntü dosyasında Renault marka araç için logoyu tespit eden ve logoyu kapsayan bir çerçevenin bilgilerini veren kod parçası gösterilmiştir.

```
import cv2
img = cv2.imread('renault.jpg')
cascade=cv2.CascadeClassifier('renault.xml')
grayColors=cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
logo=cascade.detectMultiScale(grayColors,1.1,3)
```

ŞEKİL 5.10. Logonun Bulunduğu Bölgeyi Kapsayan Çerçevenin Bilgilerini Veren Kod Parçası

Şekil 5.10.'d yer alan kod parçasında 3. satırda bulunan "cascade" değişkenine eğitim sonucunda elde edilen xml formatındaki eğitim bilgileri alınmıştır. Bu işlem için OpenCV Kütüphanesinde bulunan ve eğitim bilgilerini işleyen Cascade Sınıflandırıcı metodu kullanılmıştır. Sonraki satırda okunan "renault.jpg" görüntü dosyasının renk skalası RGB formatından gri formata dönüştürülerek "grayColors" değişkenine atılmıştır. Gri renk formatında elde edilen bu görüntüde logonun konumunun tespiti, eğitim verileri kullanılarak "detectMultiScale" metoduyla iki boyutlu görüntü üzerinde, logoyu kapsayan bir çerçeve için x ve y başlangıç koordinat noktası ile çerçeve genişlik ve uzunluk değerleri döndürülerek gerçekleştirilmektedir. Döndürülen bu değerler "logo" değişkenine atılmıştır. Şekil 5.11.'de yer alan kod parçasında, tespit edilen logoyu kapsayan bir çerçevenin görüntü üzerindeki koordinat başlangıç noktası değerlerini ve çerçeve yüksekliği, çerçeve genişliği değerlerini kullanarak görüntü üzerinde çerçeveyi çizdiren bir kod gösterilmiştir.

```
cv2.rectangle(img,(x,y),(x + w,y + h),(0,255,0),2)
```

ŞEKİL 5.11. Logoyu İçine Alan Çerçevenin Orijinal Görüntü Üzerine Çizdirilmesi

"Rectangle" isimli metot 5 parametre almaktadır. İlk parametre çizimin yapılacağı görüntü verisidir. İkinci parametre olarak çerçevenin başlangıç konum bilgisi girilmektedir. Üçüncü parametreye girilen bilgiler ile, x ve y başlangıç değerlerine uzunluk ve genişlik değerleri eklenerek çerçevenin sınırları belirlenir. Dördüncü parametre çerçeve için renk bilgisini içermektedir. Şekil 5.11.'de ilgili parametre değeri olarak girilen "(0,255,0)", yeşil rengini vermektedir. Son parametre, çerçeve stili için gerekli değeri almaktadır. Şekil 5.11.'de bu

parametre için girilen “2” değeri, sınırları olan içi boş bir çerçeve elde etmek için kullanılmaktadır. Şekil 5.10. ve Şekil 5.11.’de verilen kodların yürütülmesi sonucunda Şekil 5.12.’de yer alan çıktı görüntüsü elde edilmiştir.



ŞEKİL 5.12. Logonun Tespiti

Logonun bulunduğu bölgenin tespiti gerçekleştirildikten sonra araç logosunun orijinal görüntü üzerinden kaldırılması için içboyama işlemine geçilir.

5.3. Araç Logosunun İçboyama İşlemi İle Kaldırılması

Uygulamada içboyama işlemi için 4.1.1. başlıkta incelenen Hızlı Yürüyüş Tekniği Bazlı İçboyama yöntemi kullanılmıştır. İçboyama işlemi için iki aşamalı bir yol izlenmiştir. Öncelikle logonun bulunduğu boyanması gereken bölgenin belirlenmesi için bir maske görüntüsü yaratılmaktadır. Ardından orijinal görüntü ve maskelenmiş görüntü kullanılarak yöntem uygulanmaktadır.

5.3.1. Belirlenen Bölge İçin Maskelenmiş Görüntünün Çıkarılması

Görüntü üzerinde tespit edilen logonun bulunduğu bölgeyi sınırlayan çerçeve bilgisi çıkarılarak bir maskeleme görüntüsü yaratılmaktadır. Orijinal görüntü, maskelenmiş görüntü ile Hızlı Yürüyüş Tekniği Bazlı İçboyama yöntemi kullanılarak işlenip, logonun kaldırılması işlemi gerçekleştirilmiş olur. 5.2.1. başlıkta açıklanan eğitim sürecinden elde edilen eğitim verilerinden faydalanılarak görüntü üzerinde logonun bulunduğu bölgeyi içine alan çerçevenin bilgileri

kullanılarak maskenin çıkarılması işlemi için “researchLogoMask” isimli bir metot yazılmıştır. Bu metot için kodlar Şekil 5.13.’te verilmiştir.

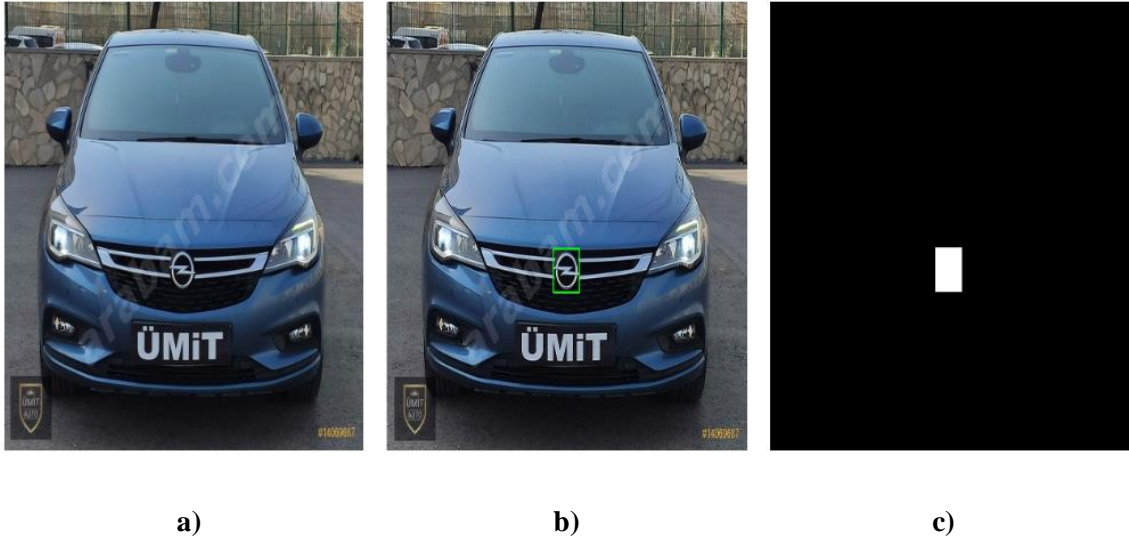
```

1  import cv2
2  import os
3
4  def researchLogoMask(img):
5      classList=os.listdir("class_xml")
6      grayColors = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
7      mask = grayColors.copy()
8      foundLogo = img.copy()
9      x, y, w, h = cv2.boundingRect(mask)
10     cv2.rectangle(mask,(x,y),(x + w,y + h),(0,0,0),-1)
11     isFound=False
12     for clas_xml in classList:
13         cascad = cv2.CascadeClassifier("class_xml/"+clas_xml)
14         logo = cascad.detectMultiScale(grayColors,1.1,3)
15         for (x,y,w,h) in logo:
16             cv2.rectangle(mask,(x,y),(x + w,y + h),(255,255,255),-1)
17             cv2.rectangle(foundLogo,(x,y),(x + w,y + h),(0,255,0),2)
18             isFound=True
19
20     if (isFound==True):
21         cv2.imwrite("output/foundLogo.jpg",foundLogo)
22         cv2.imwrite("output/mask.jpg",mask)
23         return mask,foundLogo
24     else:
25         return None,None

```

ŞEKİL 5.13. Logonun Tespitini Gerçekleştirerek, Tespit Edilen Bölge İçin Maskelenmiş Görüntü Oluşturan Metot

Şekil 5.13.’te yer alan metot, giriş değeri olarak bir görüntü almaktadır. Eğitim sonuçlarında elde edilen araç sınıfları 5 numaralı satırda “classList” isimli listeye alınmıştır. 12 numaralı satırda yer alan döngü ile her araç sınıfı için ayrı ayrı karşılaştırma gerçekleştirilmiş olur. 14 numaralı satırda, 5.2.2. başlıkta açıklanan şekilde, görüntü üzerinde ilgili logonun tespitinin gerçekleşmesi halinde tespit edilen bölgeyi içine alan çerçevenin bilgileri “logo” dizisi içine atılmaktadır. Tespit edilen logo sayısının birden fazla olması halinde her çerçevenin bilgisi ayrı ayrı elemanlar olarak diziye alınır. 15 numaralı satırda, tespit edilen her bölge için çerçeve bilgileri kullanılmak üzere bir döngü oluşturulur. 16 numaralı satırda tümüyle siyah olan “mask” görüntüsü için logonun bulunduğu bölgeyi sınırlayan çerçeve bilgileri kullanarak, ilgili metoda 4. parametre olarak “(255,255,255)” değeri (beyaz renk bilgisi) gönderilmiş ve son parametre değeri “-1” girilerek çerçevenin içinde kalan bölgenin beyaz renk pikselleri ile doldurulması sağlanmıştır. Böylelikle bir maskelenmiş görüntü elde edilmiştir. Şekil 5.14.’te, bahsedilen metot kullanılarak orijinal görüntüden elde edilen logonun çerçevelendiği görüntü ve maskelenmiş görüntü gösterilmiştir.



ŞEKİL 5.14. a) Orijinal Görüntü b) Logonun Çerçeve İçine Alınmış Hali c) Maskelenmiş Görüntü

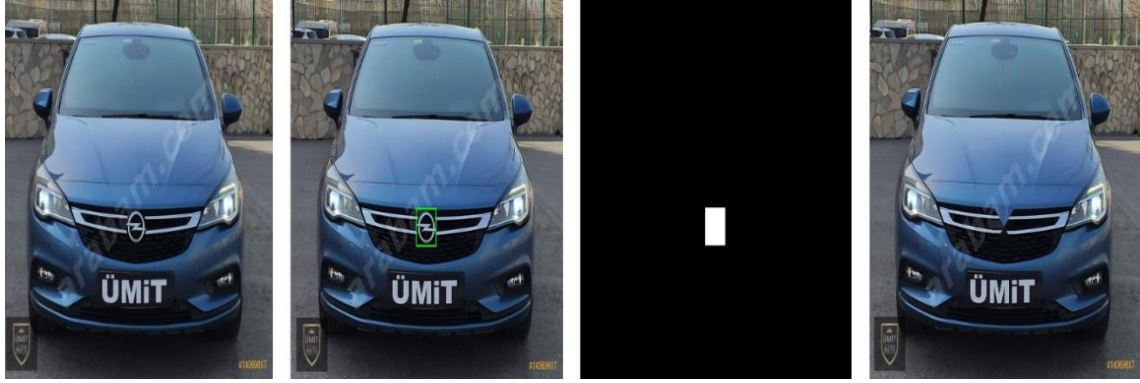
5.3.2. Maskelenmiş Görüntü Kullanılarak İçboyama İşleminin Gerçekleştirilmesi

Araç logosunun tespit edildiği bölge için bir maskelenmiş görüntü elde edildikten sonra, orijinal görüntü maskelenmiş görüntü ile yöntem tarafından işlenir. Hızlı Yürüyüş Tekniğine Dayalı İçboyama işlemi için OpenCV kütüphanesinde yer alan “inpaint” metodu kullanılmıştır. Metotta, 4.1.1. başlıkta incelenen Alexandru Telea tarafından önerilmiş olan Hızlı Yürüyüş Tekniği Bazlı İçboyama işlemi yürütülmektedir. Şekil 5.15.’te içboyama işleminin gerçekleştirilmesi için kod parçası verilmiştir.

```
img = cv2.inpaint(img, mask, 1, cv2.INPAINT_TELEA)
cv2.imwrite("output/output.jpg",img)
```

ŞEKİL 5.15. İçboyama İşleminin Gerçekleştirilmesi

Şekil 5.1.’de yer alan “inpaint” metodu 4 parametre almaktadır. İlk parametre için orijinal görüntü girilmiştir. İkinci parametre olarak elde edilen maskelenmiş görüntü gönderilmiştir. Üçüncü parametre 4.1.1. başlıkta anlatılan yöntemde, bilinen alan için bir yarıçap değeri alır. Dördüncü parametre, içboyama işlemi için kullanılacak algoritmayı belirtmektedir. Şekil 5.16.’da girilen orijinal görüntü ve maskelenmiş görüntü değerleri için elde edilen sonuç gösterilmiştir.



a)

b)

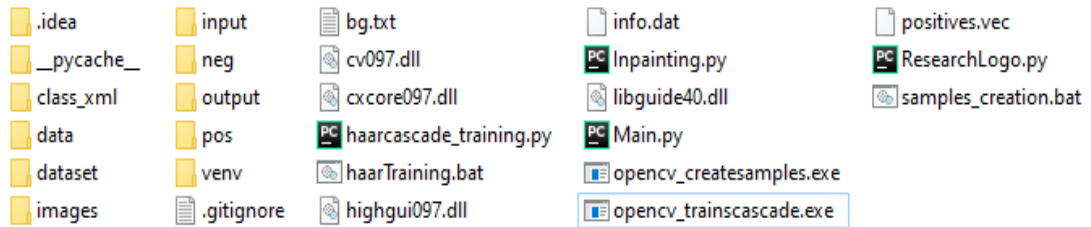
c)

d)

ŞEKİL 5.16. a) Orijinal Görüntü, b) Logonun Çerçeve İçine Alınmış Hali, c) Maskelenmiş Görüntü, d) İçboyama Uygulanan Sonuç Görüntü





















5.4. Uygulama Sonuçları

Uygulama, Citroen, Honda, Mercedes, Opel ve Renault araç markalarına ait logoları tanıyabilmektedir. Bu işlem için öncelikle belirlenen araç markalarına ait logo görüntüleri ile eğitim gerçekleştirilmiştir. Daha sonra nesne tespit yöntemi kullanılarak bu eğitimden elde edilen sonuçlarla aracın logosu orijinal görüntü üzerinde tespit edilebilmiştir. Tespit edilen bölgeye içboyama işlemi uygulanarak logonun orijinal görüntüden kaldırılması sağlanmaktadır. Uygulama için Python Programlama Dili ve Open CV Kütüphanesi kullanılmıştır. Çalışmanın içeriği Şekil 5.17’de gösterilmiştir.



ŞEKİL 5.17. Uygulamanın Proje Dosyası İçeriği

Gerçekleştirilen uygulama için 5 farklı araç markasına ait görüntüler uygulamaya giriş parametresi olarak gönderilmiş ve Şekil 5.18’de gösterilen sonuçlar elde edilmiştir.

	ORİJİNAL GÖRÜNTÜ	SONUÇ GÖRÜNTÜ		ORİJİNAL GÖRÜNTÜ	SONUÇ GÖRÜNTÜ
1			2		
3			4		
5			6		
7			8		
9			10		

ŞEKİL 5.18. Farklı Marka Araçlar İçin Uygulamanın Sonuçları

10 adet orijinal görüntüden elde edilen sonuç görüntülerle karşılaştırma sonucunda elde edilen Ortalama Karesel Hata (MSE) ve Tepe Sinyali Gürültü Oranı (PSNR) değerleri FMM Bazlı İçboyama Yöntemi ve PDE Bazlı İçboyama Yöntemi için hesaplanarak Tablo 5.1.'de gösterilmiştir.

TABLO 5.1. Orijinal ve Sonuç Görüntülerin Karşılaştırması İçin MSE ve PSNR Değerleri

GÖRÜNTÜ	FMM Bazlı İçboyama		PDE Bazlı İçboyama	
	MSE	PSNR	MSE	PSNR
Orijinal Görüntü 1 Sonuç Görüntü 1	79.500444	29.127108	81.720277	29.007505
Orijinal Görüntü 2 Sonuç Görüntü 2	53.058561	30.883249	49.333376	31.199395
Orijinal Görüntü 3 Sonuç Görüntü 3	22.800571	34.551346	22.282226	34.651218
Orijinal Görüntü 4 Sonuç Görüntü 4	44.742082	31.623642	48.237383	31.296966
Orijinal Görüntü 5 Sonuç Görüntü 5	7.845488	39.184604	7.570816	39.339377
Orijinal Görüntü 6 Sonuç Görüntü 6	20.291321	35.057700	16.396702	35.983239
Orijinal Görüntü 7 Sonuç Görüntü 7	10.602747	37.876620	10.592203	37.880941
Orijinal Görüntü 8 Sonuç Görüntü 8	44.742082	33.276754	26.475251	33.902403
Orijinal Görüntü 9 Sonuç Görüntü 9	55.858852	30.659884	60.069350	30.344274
Orijinal Görüntü 10 Sonuç Görüntü 10	25.204563	34.116012	28.381827	33.600400
	Ort. PSNR: 33.6356919		Ort. PSNR: 33.7205718	

10 adet örnek giriş görüntüsünün sonuç görüntüleriyle karşılaştırılması sonucunda Hızlı Yürüyüş Yöntemi ile İçboyama işleminin ortalama PSNR değeri 33.6356919 ve Kısmi Diferansiyel Denklemler Bazlı İçboyama işleminin ortalama PSNR değeri 33.7205718 olarak bulunmuştur. Başarı oranının saptanması amacıyla, her marka için 20 adet rastgele görüntü seçilerek uygulama çıktıları alınmıştır. Tablo 5.2’de her araç sınıfı için logonun başarı ile tespit edilerek içboyama işleminin gerçekleştirildiği başarılı sonuç sayısı ve logonun tespit edilemeyerek içboyama işleminin gerçekleştirilemediği başarısız sonuç sayısı oranlanarak ayrı ayrı ve toplam başarı yüzdeleri hesaplanmıştır.

TABLO 5.2. Uygulamanın Her Araç Sınıfı İçin Başarı Değerlendirmesi

MARKA	BAŞARILI SONUÇ	BAŞARISIZ SONUÇ	TOPLAM	BAŞARI YÜZDESİ
Citroen	8	12	20	%40
Honda	16	4	20	%80
Mercedes	17	3	20	%85
Opel	15	5	20	%75
Renault	15	5	20	%80
GENEL BAŞARI YÜZDESİ: %71				

Tablo 5.2.'de görülebileceği üzere toplam 100 orijinal görüntü için uygulamanın logoyu tespit etme genel başarı yüzdesi %71 olarak hesaplanmıştır. En düşük başarı oranı %40 ile Citroen araç sınıfı için gözlemlenmiş, en başarılı sonuçlar %85 başarı oranı ile Mercedes araç sınıfı ile elde edilmiştir. Uygulamada giriş görüntüsünün çözünürlüğü arttıkça başarı oranı artmaktadır.

6. SONUÇ

Çalışmada sayısal görüntü ve görüntü işleme kavramları üzerinde durulmuş ve görüntü işleminin uygulama alanlarından bahsedilmiştir. Görüntü üzerinde nesnenin tanınması, görüntü üzerinde içboyama işleminin gerçekleştirilmesi için yöntemler incelenmiştir. Makine öğrenmesi kullanılarak nesne tespit yöntemlerinden Haar Benzeri Öznitelikler Bazlı Nesne Tespiti ve HOG ile Nesne Tespiti ele alınmıştır. Derin öğrenme yapısı açıklanmaya çalışılarak derin öğrenme kullanılarak nesne tespit yöntemlerinden R-CNN ve YOLO ile nesne tespit yöntemleri incelenmiştir. Görüntü üzerinde içboyama işlemi için geleneksel yöntemler olan FMM Bazlı İçboyama, PDE Bazlı İçboyama ve Örnek Bazlı İçboyama yöntemleri incelenmiştir. İncelenen yöntemlerin ardından, nesne tanıma için Haar Benzeri Öznitelikler ve içboyama işlemi için Hızlı Yürüyüş Tekniği kullanılarak; görüntü üzerinde aracın logosunu tespit ederek görüntüden logoyu kaldıran bir uygulama ortaya konulmuştur. Uygulamanın amacından, çalışma işleyişinden ve kullanılan yöntemlerinden bahsedilmiştir. Uygulamada giriş görüntü verileri için sonuçların karşılaştırılması ile Hızlı Yürüyüş Tekniği Bazlı İçboyama işleminde ortalama PSNR değeri 33.6356919 ve Kısmi Diferansiyel Denklemler Bazlı İçboyama işleminde ortalama PSNR değeri 33.7205718 olarak elde edilmiştir. Uygulamada 100 adet örnek giriş görüntüsü için uygulamanın logoyu başarı ile tespit ederek içboyama işleminin gerçekleştirilmesinin başarı yüzdesi %71 olarak bulunmuştur.

ÖZGEÇMİŞ

Kişisel Bilgiler:

Adı Soyadı: Mert İNCİDELEN

Doğum Yeri: Merkez/ELÂZİĞ

Doğum Tarihi: 04/07/1997

İletişim Bilgileri:

Adres: Hilalkent Mah., Defne 2 Apt., No:5, Merkez/ELÂZİĞ

Telefon: 0 531 267 97 23

E-Posta: mincidelen@gmail.com – mert@incidelen.com

Kişisel Web Sitesi: incidelen.com – bilgisayar kutuphanesi.com

Eğitim Bilgileri:

Lise: Elâzığ Mehmet Koloğlu Anadolu Lisesi

Lisans: Fırat Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği Bölümü

Yabancı Dil:

İngilizce: Orta Seviye

Yetkinlikler:

Java, Python, C, ABAP, HTML, CSS, JavaScript, WordPress, OpenCard, SAP Teknolojileri, Bilgisayar Ağları, Linux Sistemler, MySQL ve MsSQL

Deneyimler:

Detaysoft - Detay Danışmanlık Bilgisayar Hizmetleri Sanayi ve Dış Tic. A.Ş.

WittyCommerce - MYE Bilgisayar Yazılım Tic. A.Ş.

Türkiye Vodafone Vakfı ve Habitat Derneği (Yarını Kodlayanlar Projesi, Gönüllü Eğitimci)

Organizasyonlar:

13. 14. ve 15. Bilgisayar Mühendisliği Öğrencileri Kongresi (Üniversite Temsilcisi, Resmi Katılımcı)

ALKÜ IEEE (Kurucu Bşk.Yrd.)

Sertifikalar:

Pedagojik Formasyon Eğitimi Sertifikası (Fırat Üniversitesi Eğitim Fakültesi)

Introduction to Cyber Security (Cisco Networking Academy)

Wireless Networks Training (Cisco Networking Academy)



Kişisel Bilgiler:

Adı Soyadı: Okan YAR

Doğum Yeri: Ovacık/TUNCELİ

Doğum Tarihi: 01/01/1995

İletişim Bilgileri:

Adres: İzzetpaşa Mah. Söngü Sok. No:1 Merkez/ELÂZİĞ

Telefon: 0 531 625 45 56

E-Posta: okanyar.ce@gmail.com

Eğitim Bilgileri:

Lise: Elâzığ Cumhuriyet Lisesi

Lisans: Fırat Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği Bölümü

Yabancı Dil:

İngilizce: Orta Seviye

Yetkinlikler:

Java, Python, Android mobil geliştirme, SQL İlişkisel Veritabanı.Nesneye Yönelik Programlama (OOP), HTML, JavaScript, Php, CSS, UML, Photoshop

Deneyimler:

Ares İnovasyon Arge Yazılım Hizmetleri/ODTÜ Teknokent

Korona Yazılım/ODTÜ Teknokent

Organizasyonlar:

14. ve 15. Bilgisayar Mühendisliği Öğrencileri Kongresi (Üniversite Temsilcisi, Resmi Katılımcı)

Sertifikalar:

Introduction to Cyber Security (Cisco Networking Academy)

Wireless Networks Training (Cisco Networking Academy)

Uygulamalar:

Kelimecik Android Mobil Oyun Uygulaması (Şirket Uygulaması Oluşturma-Geliştirme)

Halipratik Mobil Uygulaması (Şirket Uygulaması Oluşturma-Geliştirme)



KAYNAKLAR

- [1] Aubert M., Brumm A., Ramli M., Sutikna T., Saptomo E.W., Hakim B., Morwood M.J., Bergh G.D., Kinsley L., Dosseto A., «Pleistocene cave art from Sulawesi, Indonesia,» *Nature*, no. 514, p. 223–227, 2014.
- [2] Gonzalez R.C., Woods R.E., Digital Image Processing, Third Edition, Prentice Hall, 2008.
- [3] Liu J.G., Mason P.J., Image Processing and GIS for Remote Sensing: Techniques and Applications, Wiley-Blackwell, 2016.
- [4] Wolf P.R., Dewitt B.A., Wilkinson B.E., Elements of Photogrammetry with Application in GIS, Fourth Edition, McGraw-Hill Education, 2013.
- [5] University of Tartu, «1. Introduction to image processing | Digital Image Processing,» [Çevrimiçi]. Available: <https://sisu.ut.ee/imageprocessing/book/1>. [Erişildi: 10 06 2020].
- [6] Nixon M., Aguado A., Feature Extraction and Image Processing for Computer Vision Fourth Edition, Academic Press, 2019.
- [7] İbrahim Çayiroğlu, «Goruntu_Isleme_Ders_Notlari-1.Hafta,» [Çevrimiçi]. Available: http://www.ibrahimcayiroglu.com/Dokumanlar/GoruntuIsleme/Goruntu_Isleme_Ders_Notlari-1.Hafta.pdf. [Erişildi: 11 06 2020].
- [8] Chanda B., Majumdar D.D., Digital Image Processing and Analysis Second Edition, PHI Learning, 2011.
- [9] Esakkirajan S., Veerakumar T., Jayaraman S., Digital Image Processing, Mc Graw Hill India, 2009.
- [10] Geoff D., Digital Image Processing for Medical Applications, Cambridge University Press, 2009.
- [11] Potdar K., Pai C., Akolkar S., «A Convolutional Neural Network based Live Object Recognition System as Blind Aid,» 2018.
- [12] Alzubi J., Nayyar A., Kumar A., «Machine Learning from Theory to Algorithms: An Overview,» *IOP Publishing*, 2018.
- [13] Viola P., Jones M., «Robust Real-time Object Detection,» *International Journal of Computer Vision*, 2001.
- [14] Mita T., Kaneko T., Hori O., «Joint Haar-like Features for Face Detection,» *IEEE International Conference on Computer Vision*, no. 2, pp. 1619 - 1626, 2005.
- [15] Goyal K., Agarwal K., Kumar R., «Face detection and tracking: Using OpenCV,» *International conference of Electronics, Communication and Aerospace Technology 2017*, pp. 474-478, 2017.

- [16] Viola P., Jones M., «Rapid Object Detection using a Boosted Cascade of Simple Features,» *Accepted Conference On Computer Vision And Pattern Recognition*, 2001.
- [17] Shashua A., Gdalyahu Y., Hayon G., «Pedestrian Detection for Driving Assistance Systems: Single-Frame Classification and System Level Performance,» *In Proceedings of IEEE Intelligent Vehicles Symposium*, 2004.
- [18] Dalal N., Triggs B., «Histograms of Oriented Gradients for Human Detection,» *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2005)*, 2005.
- [19] Peker M., Altun H., Karakaya F., «HOG Temelli Bir Yöntem ile Ölçek ve Yönden Bağımsız Gerçek Zamanlı Nesne Tanıma,» *Otomatik Kontrol Türk Milli Komitesi'nin Otomatik Kontrol Ulusal Toplantısı*, 2012.
- [20] Karakaya F., Altun H., Çavuşlu M.A., «Gerçek Zamanlı Nesne Tanıma Uygulamaları için HOG Algoritmasının FPGA Tabanlı Gömülü Sistem Uyarlaması,» 2009.
- [21] «Histogram of Oriented Gradients | Learn OpenCV,» [Çevrimiçi]. Available: <https://www.learnopencv.com/histogram-of-oriented-gradients/>. [Erişildi: 04 06 2020].
- [22] Şeker A., Diri B., Balık H.H., «Derin Öğrenme Yöntemleri ve Uygulamaları Hakkında Bir İnceleme,» *Gazi Mühendislik Bilimleri Dergisi*, 2017.
- [23] Bre F., Gimenez J.M., Fachinotti V.D., «Prediction of wind pressure coefficients on building surfaces using Artificial Neural Networks,» *Energy and Buildings*, 2017.
- [24] «RCNN , YOLO-OpenCV ile Nesne Tanıma - Operations Management Türkiye - Medium,» [Çevrimiçi]. Available: <https://medium.com/operations-management-t%C3%BCrkiye/rcnn-yolo-opencv-ile-nesne-tan%C4%B1ma-fa3856e69fb3>. [Erişildi: 04 06 2020].
- [25] Girshick R., Donahue J., Darrell T., Malik J., «Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation,» *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014.
- [26] Tan Z., «Derin Öğrenme Yardımıyla Araç Sınıflandırma,» *Fırat Üniversitesi Fen Bilimleri Enstitüsü Yüksek Lisans Tezi*, 2019.
- [27] Cengil E., Çınar A., «Görüntü Sınıflandırma İçin Yeni Bir Yaklaşım: Evrimsel Sinir Ağları,» *European Journal of Technic*, cilt 6, no. 2, 2016.
- [28] Redmon J., Divvala S., Girshick R., Farhadi A., «You Only Look Once: Unified, Real-Time Object Detection,» *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [29] Appsilon Data Science, «YOLO Algorithm and YOLO Object Detection: An Introduction,» [Çevrimiçi]. Available: <https://appsilon.com/object-detection-yolo-algorithm/>. [Erişildi: 29 06 2020].
- [30] Harrouss E.O., Almaadeed N., Al-Maadeed S., Akbari Y., «Image Inpainting: A Review,» *Springer Nature*, 2019.

- [31] Telea A., «An Image Inpainting Technique Based on the Fast Marching Method,» *Journal of Graphics Tools*, cilt 9, no. 1, p. 28, 2004.
- [32] OpenCV, «OpenCV: Image Inpainting,» [Çevrimiçi]. Available: https://docs.opencv.org/master/df/d3d/tutorial_py_inpainting.html. [Erişildi: 05 06 2020].
- [33] Au W., Takei R., «Image Inpainting With the Navier-Stokes Equations,» *Simon Fraser University*, 2001.
- [34] El-Glaly Y.N., «Development of PDE-based Digital Inpainting,» *Ain shams University Faculty of Computer & Information Sciences Computer Science Department*, 2002.
- [35] Bertalmio M., Bertozzi A.L., Sapiro G., «Navier-Stokes, Fluid Dynamics, and Image and Video Inpainting,» *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2001.
- [36] Thanki B.B., «Overview of An Image Inpainting Techniques,» *International Journal For Technological*, no. 2, pp. 388-391, 2015.
- [37] Nabyev V.V., A. Taşçı, «İstenmeyen Cisimlerin Resimden Kaldırılması,» *Türkiye Bilişim Vakfı Bilgisayar Bilimleri ve Mühendisliği Dergisi*, no. 4, 2016.