

**SÜRÜCÜ GÜVENLİĞİ İÇİN GERÇEK ZAMANLI
ŞERİTTEN AYRILMA VE YSA TABANLI ÖNDEN
ÇARPMA UYARI SİSTEMİ**

**2017
YÜKSEK LİSANS TEZİ
ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ**

Kadir İLERİ

**SÜRÜCÜ GÜVENLİĞİ İÇİN GERÇEK ZAMANLI ŞERİTTEN AYRILMA
VE YSA TABANLI ÖNDEN ÇARPMA UYARI SİSTEMİ**

Kadir İLERİ

Karabük Üniversitesi

Fen Bilimleri Enstitüsü

Elektrik-Elektronik Mühendisliği Anabilim Dalında

Yüksek Lisans Tezi

Olarak Hazırlanmıştır

KARABÜK

Ocak 2017

Kadir İLERİ tarafından hazırlanan “SÜRÜCÜ GÜVENLİĞİ İÇİN GERÇEK ZAMANLI ŞERİTTEN AYRILMA VE YSA TABANLI ÖNDEN ÇARPMA UYARI SİSTEMİ” başlıklı bu tezin Yüksek Lisans Tezi olarak uygun olduğunu onaylarım.

Yrd. Doç. Dr. Mustafa GÖKDAĞ

Tez Danışmanı, Elektrik-Elektronik Mühendisliği Anabilim Dalı

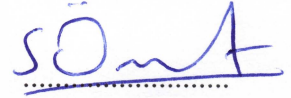


Bu çalışma, jürimiz tarafından oy birliği ile Elektrik-Elektronik Mühendisliği Anabilim Dalında Yüksek Lisans tezi olarak kabul edilmiştir. 04/01/2017

Ünvanı, Adı SOYADI (Kurumu)

İmzası

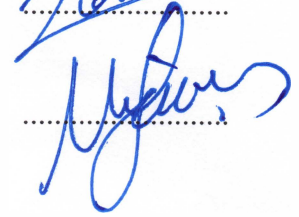
Başkan : Yrd. Doç. Dr. Serdar ÖZYURT (YBÜ)



Üye : Yrd. Doç. Dr. Mustafa GÖKDAĞ (KBÜ)



Üye : Yrd. Doç. Dr. M. Tahir GÜNEŞER (KBÜ)

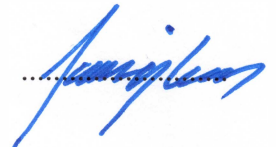


20/01/2017

KBÜ Fen Bilimleri Enstitüsü Yönetim Kurulu, bu tez ile, Yüksek Lisans derecesini onamıştır.

Prof. Dr. Nevin AYTEMİZ

Fen Bilimleri Enstitüsü Müdürü



“Bu tezdeki tüm bilgilerin akademik kurallara ve etik ilkelere uygun olarak elde edildiğini ve sunulduğunu; ayrıca bu kuralların ve ilkelerin gerektirdiği şekilde, bu çalışmadan kaynaklanmayan bütün atıfları yaptığımı beyan ederim.”

Kadir İLERİ



ÖZET

Yüksek Lisans Tezi

SÜRÜCÜ GÜVENLİĞİ İÇİN GERÇEK ZAMANLI ŞERİTTEN AYRILMA VE YSA TABANLI ÖNDEN ÇARPMA UYARI SİSTEMİ

Kadir İLERİ

Karabük Üniversitesi

Fen Bilimleri Enstitüsü

Elektrik-Elektronik Mühendisliği Anabilim Dalı

Tez Danışmanı:

Yrd. Doç. Dr. Mustafa GÖKDAĞ

Ocak 2017, 75 sayfa

Bu çalışmada, sürücüler için olası kazaları önlemek amacıyla gerçek zamanlı video tabanlı erken uyarı sistemi geliştirilmiştir. Bu erken uyarı sistemi iki kısımdan oluşmaktadır. Birinci kısımda, aracın şeritten ayrılıp ayrılmadığı tespit edilir. Eğer şeritten ayrılıyor ise hangi şerit çizgisinden ayrıldığı sürücüye sesli olarak uyarı verilir. İkinci kısımda ise önden çarpmayı engellemek için öndeki aracın mesafesi ile yola giren herhangi renkli bir cisim hakkında sürücüye sesli olarak uyarı verilir. Bu erken uyarı sistemi iki adet ARM geliştirme kartı üzerinde çalıştırılmış ve performans karşılaştırması yapılmıştır.

Şeritten ayrılma uyarı sisteminde, ilk olarak sistemin daha hızlı çalışması için şeridin olduğu kısım maskelenerek şeridin olmadığı kısımdan ayrılmıştır. Sonraki işlemler şeridin olduğu kısım üzerinden gerçekleştirilmiştir. Perspektif etkisinden dolayı şerit çizgileri görüntünün alt kısmında daha geniş üst kısmında ise daha dar olarak

gözükür. Bu perspektif etkisini yok etmek için görüntüye ters perspektif dönüşümü uygulanmıştır. Ters perspektif görüntüsü gri ölçekli görüntüye dönüştürüldükten sonra gürültü azaltma işlemi yapılmış ve şerit çizgilerini tespit etmek için basit bir filtre uygulanmıştır. Canny kenar belirleme yöntemiyle şerit çizgilerinin kenarları belirlenmiştir. Daha sonra Hough dönüşümü yapılarak şerit çizgilerinin açısı belirlenmiş ve bu açılar göz önüne alınarak şeritten ayrılmanın olup olmadığına karar verilmiştir. Eğer şeritten ayrılma söz konusuysa sürücüyü sesli olarak uyarı verilmiştir.

Önden çarpma uyarı sisteminde, önce kameradan alınan görüntü daha ince tespit edilen şerit çizgileri yardımıyla maskelenmiştir. Böylece muhtemel öndeki araç veya renkli cisim bu şerit çizgileri arasında aranmıştır. Bu maskelenmiş görüntü gri ölçekli görüntüye dönüştürülerek gürültü azaltma işlemi yapılmıştır. Öndeki aracı belirlemek için altında kalan gölgeden yararlanılmıştır. Bu gölge Otsu eşikleme yöntemiyle siyah-beyaz (ikili) görüntüye dönüştürülüp tespit edilmiştir. Gölge yardımıyla araç tespiti ve renk maskeleyme yöntemiyle renkli cisim tespiti gerçekleştirilmiştir. Bu aracın/cismin mesafesi eğri uydurma yöntemi ve yapay sinir ağları (YSA) ile tahmin edilerek sonuçlar karşılaştırılmıştır. Eğri uydurma için 10. dereceden polinom ile mesafe ortalama %8,7563, iki nörondan oluşan tek katmanlı YSA ile ortalama %0,8701 hata oranıyla elde edilmiştir.

Anahtar Sözcükler : Mesafe tahmini, şerit belirleme, gerçek zamanlı, Hough dönüşümü, ters perspektif dönüşümü, yapay sinir ağları.

Bilim Kodu : 905.1.067

ABSTRACT

M. Sc. Thesis

REAL TIME LANE DEPARTURE AND ANN BASED FORWARD COLLISION WARNING SYSTEM FOR DRIVER SAFETY

Kadir İLERİ

Karabük University

Graduate School of Natural and Applied Sciences

Department of Electrical-Electronics Engineering

Thesis Advisor:

Assist. Prof. Dr. Mustafa GÖKDAĞ

Jan 2017, 75 pages

In this study, real time video based early warning system is developed to avoid potential accidents for drivers. This early warning system consists of two parts. In the first part, driver is warned by voice message while driver is making lane departure. In the second part, driver is warned by voice message about distance of front vehicle and colourful objects to avoid front collision. This early warning system has been implemented on two ARM boards and their performances have been compared.

In the first part, input image has been separated into lane part and non-lane part of road to improve performance. Following steps were applied to lane part of road. Lane marks' width appears wider in lower part of image than upper part of image because of perspective effect. Inverse perspective mapping has been applied to the image to eliminate this perspective effect. After converting to grayscale image, it has reduced noise of grayscale image and a basic filter has applied to detect lane marks.

Lane marks' edges has been found by using Canny edge detector. Next, Lane marks' angles have been determined by applying Hough transform and it has been decided whether driver is making lane departure or not by considering these determined angles. Finally, driver is warned by voice message while driver is making lane departure.

In second part, input image captured from camera masked by the help of lane marks which found in first part. Thus, front vehicle has been searched between own lane marks. The masked image has been converted to grayscale image and its noise has been reduced. It has been used shadow underneath vehicle to detect front vehicle and this shadow has been detected by using Otsu method. Location of vehicle has been determined by the help of the shadow and the distances of front vehicle and colourful object have been estimated by curve fitting method. Finally, driver has been warned by voice message about the estimated distances.

Key Words : Distance estimation, lane detection, real-time, Hough transform, inverse perspective mapping, artificial neural network.

Science Code : 905.1.067

TEŐEKKÜR

Bu tez alıőmasının planlanmasında, araőtırılmasında, yürütülmesinde ve oluşumunda ilgi ve desteęini esirgemeyen, engin bilgi ve tecrübelerinden yararlandığım, yönlendirme ve bilgilendirmeleriyle alıőmamı bilimsel temeller ışığında őekillendiren sayın hocam Yrd. Do. Dr. Mustafa GÖKDAĞ'a sonsuz teşekkürlerimi sunarım.



İÇİNDEKİLER

Sayfa

KABUL.....	ii
ÖZET.....	iv
ABSTRACT.....	vi
TEŞEKKÜR.....	viii
İÇİNDEKİLER.....	ix
ŞEKİLLER DİZİNİ.....	xii
ÇİZELGELER DİZİNİ.....	xv
SİMGELER VE KISALTMALAR DİZİNİ.....	xvi
BÖLÜM 1.....	1
GİRİŞ.....	1
1.1. LİTERATÜR TARAMASI.....	3
1.1.1. Şerit Belirleme.....	3
1.1.2. Araç Tespit Etme.....	9
BÖLÜM 2.....	15
ŞERİTTEN AYRILMA UYARI SİSTEMİ.....	15
2.1. İLGİLİ ALAN BELİRLEME.....	17
2.2. TERS PERSPEKTİF DÖNÜŞÜMÜ.....	18
2.3. GRİ ÖLÇEKLİ GÖRÜNTÜ.....	20
2.4. GÜRÜLTÜ GİDERME.....	21
2.4.1. Medyan (Ortanca) Filtresi.....	21
2.4.2. Ortalama Filtresi.....	22
2.4.3. Gauss Filtresi.....	22
2.5. ŞERİT ÇİZGİSİ FİLTRELEME.....	23
2.6. KENAR BELİRLEME.....	26
2.6.1. Robert Cross Kenar Belirleme Operatörü.....	26

	<u>Sayfa</u>
2.6.2. Sobel Kenar Belirleme Operatörü	27
2.6.3. Prewitt Kenar Belirleme Operatörü	28
2.6.4. Canny Kenar Belirleme	29
2.7. HOUGH DÖNÜŞÜMÜ	35
2.8. BELİRLENEN ŞERİT ÇİZGİLERİNİ GİRİŞ GÖRÜNTÜSÜNE TAŞIMA	39
2.9. ŞERİTTEN AYRILMA UYARI SİSTEMİ	40
BÖLÜM 3	42
ÖNDEN ÇARPMA UYARI SİSTEMİ.....	42
3.1. İLGİLİ ALAN BELİRLEME.....	43
3.2. GRİ ÖLÇEKLİ GÖRÜNTÜ	45
3.3. GÜRÜLTÜ GİDERME.....	45
3.4. GÖRÜNTÜ EŞİKLEME.....	46
3.4.1. Global Eşikleme Yöntemi	46
3.4.2. Otsu Eşikleme Yöntemi.....	47
3.5. ARAÇ SINIRLAYICI KUTUSU BULMA	50
3.5.1. Aracın Sol, Sağ Ve Alt Sınırlarını Tespit Etme.....	50
3.5.2. Aracın Üst Sınırını Tespit Etme	51
3.6. RENKLİ NESNE TESPİT ETME	55
3.6.1. Renk Uzayı Dönüşümü.....	55
3.6.2. Renkli Görüntü Eşikleme	56
3.6.3. Nesnenin Alt Sınırını Tespit Etme.....	57
3.7. MESAFE TAHMİNİ.....	57
3.7.1. Eğri Uydurma Yöntemi İle Mesafe Tahmini.....	58
3.7.2. Yapay Sınır Ağları Yöntemi İle Mesafe Tahmini	60
BÖLÜM 4	65
DENEYSEL SONUÇLAR.....	65
BÖLÜM 5	69
SONUÇ VE ÖNERİLER	69

	<u>Sayfa</u>
KAYNAKLAR	70
ÖZGEÇMİŞ	75



ŞEKİLLER DİZİNİ

Sayfa

Şekil 1.1. Köşeleri belirlemek için kullanılan dört farklı maske çeşidi [37].....	13
Şekil 2.1. Şeritten ayrılma uyarı sisteminin akış şeması.....	16
Şekil 2.2. Belirlenen ilgili alan.....	17
Şekil 2.3. Kameranın araca yerleştirildiği konum.....	17
Şekil 2.4. Giriş görüntüsü üzerindeki perspektif etkileri, a) sabit olmayan şerit çizgisi genişliği, b) sabit olmayan şerit çizgisi uzunluğu [44].....	18
Şekil 2.5. Koordinat sistemleri, a) görüntü koordinat sistemi, b) gerçek dünya koordinat sistemi.....	19
Şekil 2.6. İlgili alan görüntüsü ve onun TPD görüntüsü, a) ilgili alan görüntüsü, b) TPD görüntüsü.....	20
Şekil 2.7. Gri ölçekli görüntü.....	21
Şekil 2.8. Medyan filtresinin uygulanışı.....	22
Şekil 2.9. Gürültüsü azaltılmış görüntü.....	23
Şekil 2.10. Şerit çizgisi filtreleme, a) gürültüsü azaltılmış görüntü, b) filtrelenmiş görüntü.....	25
Şekil 2.11. Satır analiz grafiği.....	25
Şekil 2.12. Robert Cross maskeleri, a) sol çapraz Robert Cross maskesi, b) sağ çapraz Robert Cross maskesi.....	26
Şekil 2.13. Maske altında kalan görüntü pikselleri.....	26
Şekil 2.14. Sobel maskeleri, a) yatay sobel maskesi, b) dikey sobel maskesi.....	27
Şekil 2.15. Maske altında kalan görüntü pikselleri.....	27
Şekil 2.16. Prewitt maskeleri, a) yatay prewitt maskesi, b) dikey prewitt maskesi.....	28
Şekil 2.17. Maske altında kalan görüntü pikselleri.....	28
Şekil 2.18. Canny kenar belirleme yöntemi için kullanılan örnek giriş görüntü [49].	29
Şekil 2.19. Yumuşatma sonucunda oluşan görüntü [49].	30
Şekil 2.20. Gradyan genliği görüntüsü [49].....	31
Şekil 2.21. Maksimum olmayanları yok etme gösterimi [49].....	32
Şekil 2.22. Sadece yerel maksimumu olan piksellerin korunduğu görüntü [49].....	32

Şekil 2.23. Çift eşikleme görüntüsü [49].	33
Şekil 2.24. Histerez ile kenar izleme görüntüsü [49].	34
Şekil 2.25. Örnek görüntü üzerinde Canny kenar belirleme, a) giriş görüntüsü, b) sonuç görüntüsü [49].	34
Şekil 2.26. Canny kenar belirleme, a) giriş görüntüsü, b) sonuç görüntüsü.	35
Şekil 2.27. Kartezyen koordinat sisteminde bir doğru.	36
Şekil 2.28. Kutupsal koordinat sisteminde bir doğru.	36
Şekil 2.29. Örnek bir noktanın iki koordinat sisteminde gösterimi, a) Kartezyen koordinat sisteminde, b) Kutupsal koordinat sisteminde [50].	37
Şekil 2.30. Örnek noktaların iki koordinat sisteminde gösterimi, a) Kartezyen koordinat sisteminde, b) Kutupsal koordinat sisteminde [50].	37
Şekil 2.31. Tespit edilen çizgi [50].	38
Şekil 2.32. Hough dönüşümü, a) giriş görüntüsü, b) sonuç görüntü.	38
Şekil 2.33. Tespit edilen sol ve sağ şerit çizgileri.	39
Şekil 2.34. Belirlenen şerit çizgileri.	40
Şekil 2.35. Şeritten ayrılma durumları, a) sol şerit çizgisinden ayrılma durumu, b) ideal durum, c) sağ şerit çizgisinden ayrılma durumu.	40
Şekil 2.36. Grafıksel olarak şeritten ayrılma durumları, a) sol şerit çizgisinden ayrılma durumu, b) ideal durum, c) sağ şerit çizgisinden ayrılma durumu.	41
Şekil 3.1. Önden çarpma uyarı sisteminin akış şeması.	43
Şekil 3.2. Örnek ilgili alanlar, (a) önde aracın olduğu, (b) önde renkli nesnenin olduğu, (c) öndeki aracın olmadığı.	44
Şekil 3.3. Gri ölçekli görüntü.	45
Şekil 3.4. Gürültüsü azaltılmış görüntü.	45
Şekil 3.5. Farklı eşik değerleri için çıktı görüntüleri.	47
Şekil 3.6. Tespit edilen araç gölgesi.	50
Şekil 3.7. Tespit edilen alt sınır.	51
Şekil 3.8. Tespit edilen sağ ve sol sınırlar.	51
Şekil 3.9. Piksel derinliği.	52
Şekil 3.10. Ölçüm sonucunda elde edilen piksel derinlikleri ve bu derinliklere karşılık gelen yükseklik değerleri.	53
Şekil 3.11. Elde edilen eğri denklemin grafiği.	55
Şekil 3.12. HSV renk uzayına dönüştürülmüş görüntü.	56

	<u>Sayfa</u>
Şekil 3.13. Eşiklenmiş renkli görüntü.....	56
Şekil 3.14. Alt sınırı tespit edilmiş görüntü.	57
Şekil 3.15. Ölçüm sonucunda elde edilen piksel derinlikleri ve bu derinliklere karşılık gelen mesafe değerleri.....	58
Şekil 3.16. Elde edilen eğri denkleminin grafiği.....	59
Şekil 3.17. Yapay sinir hücresinin yapısı.....	61
Şekil 3.18. Yapay sinir ağının yapısı.	62
Şekil 3.19. Uygulamada kullanılan yapay sinir ağının yapısı.....	64
Şekil 4.1. Elde edilen sonuç görüntüleri.	67



ÇİZELGELER DİZİNİ

Sayfa

Çizelge 1.1. Yıllara göre kaza, yaralı ve ölü sayıları istatistikleri [2].....	1
Çizelge 3.1. Elde edilen katyası değerleri.....	54
Çizelge 3.2. Elde edilen katyası değerleri.....	59
Çizelge 3.3. Eğitim sonucunda elde edilen ağırlık değerleri.	64
Çizelge 3.4. Eğri uydurma ve yapay sinir ağırları ile elde edilen mesafe hesaplama sonuçları.....	64
Çizelge 4.1. Algoritmanın tüm bileşenlerinin ortalama işlem süreleri.	66

SİMGELER VE KISALTMALAR DİZİNİ

KISALTMALAR

TPD : Ters Perspektif Dönüşümü

T : Eşik Değeri

P : Fonksiyon Olasılığı

I : En Büyük Piksel Değeri

H : Eğri ve Veri Noktaları Arasındaki Hata

YSA : Yapay Sinir Ağları

BÖLÜM 1

GİRİŞ

Dünyadaki ölümlerin başlıca nedenlerinden birisi trafik kazalarıdır. Artan nüfus ve trafik yoğunluğunun bir sonucu olarak trafik kazaları hem ölümcül insan kayıplarına neden olmakta hem de ülke ekonomisine zarar verip sosyal sorunlara yol açmaktadır. Bu kazalar birçok sebepten ve birçok farklı yolla meydana gelebilir.

Dünyada her yıl 1 170 000 insan trafik kazaları sonucunda hayatını kaybetmektedir. Bunun %70 gibi büyük bir çoğunluğu gelişmekte olan ülkelerde meydana gelmektedir. Ayrıca her yıl 10 000 000'dan fazla insan da sakat kalmakta veya yaralanmaktadır [1]. Türkiye'deki durum ise Çizelge 1.1'de gösterildiği üzere 2015 yılında 1 313 359 adet kaza olmuştur. Bu kazalardan 1 130 348 adedinde hasar meydana gelmiş, 3 831 kişi hayatını kaybetmiş ve 304 421 kişi ise yaralanmıştır. Ölümlü yaralanmalı kazaların %66,6'sı gündüz meydana gelmiştir [2].

Çizelge 1.1. Yıllara göre kaza, yaralı ve ölü sayıları istatistikleri [2].

Yıl	Toplam kaza sayısı	Maddi hasarlı kaza sayısı	Ölümlü, yaralanmalı kaza sayısı	Ölü sayısı (kaza yerindeki ölümleri kapsar)	Yaralı sayısı
2007	825 561	718 567	106 994	5 007	189 057
2008	950 120	845 908	104 212	4 236	184 468
2009	1 053 346	942 225	111 121	4 324	201 380
2010	1 106 201	989 397	116 804	4 045	211 496
2011	1 228 928	1 097 083	131 845	3 835	238 074
2012	1 296 634	1 143 082	153 552	3 750	268 079
2013	1 207 354	1 046 048	161 306	3 685	274 829
2014	1 199 010	1 030 498	168 512	3 524	285 059
2015	1 313 359	1 130 348	183 011	3 831	304 421

Son zamanlara kadar kazaları önlemek ya da etkilerini en aza indirmek için emniyet kemeri takma zorunluluğu, araba üreticilerini araçlarda hava yastığı olmasına zorlama gibi çeşitli yaklaşımlar gerçekleştirilmiştir. Bu yaklaşımlar kazaların etkisini azalttığı gibi gelişmiş teknolojiyi kullanarak daha fazlasını yapmak mümkün hale gelmiştir.

Sürücü uyarı sistemleri kaza olmadan önce sürücüyü uyararak kazanın olmasını önlemektedir. Bu çalışmada, sürücü şeritten ayrıldığı zaman sürücüye sesli uyarı veren ve önden çarpmayı önlemek için öndeki aracın ve renkli herhangi bir nesnenin mesafesi hakkında yine sürücüye sesli uyarı veren gerçek zamanlı erken uyarı sistemi gerçekleştirilmiştir.

Şeritten ayrılma uyarı sistemi bölümünü gerçekleştirmek için öncelikle aracın bulunduğu şeridin tespit edilmesi gerekir daha sonra tespit edilen şeridin bilgileri kullanılarak şeritten ayrılma söz konusu ise sürücüye uyarı verilir. Şerit tespiti için model temelli ve özellik temelli olmak üzere iki ana yaklaşım vardır. Model temelli yaklaşımda şerit çizgilerini belirlemek için yönlendirilebilir filtre kullanılır. Özellik temelli yöntemde şerit belirleme için Hough dönüşümü kullanılır. Hough dönüşümü düz çizgilerin belirlenmesinde kullanılan hızlı ve doğru çalışan klasik bir yöntemdir [3]. Bu nedenle çalışmamızda şerit çizgilerini belirlemek için Hough dönüşümü tercih edilmiştir. Bu çizgilerin bilgileri kullanılarak şeritten ayrılma olup olmadığı tespit edilmiştir.

Önden çarpma uyarı sistemi bölümünde ise, öndeki aracın mesafesi belirlenip sürücüye uyarı verilebilmesi için öncelikle öndeki aracın tespit edilmesi gerekir. Araç tespit etmek için çeşitli yöntemler mevcuttur. Bu yöntemler hareket temelli, görünüş temelli ve çift kamera temelli olmak üzere üç farklı gruba ayrılabilir. Hareket temelli yaklaşımda iki görüntü karesi arasındaki farktan yararlanılarak hareket halindeki bir araç tespit edilebilir [4]. Çift kamera temelli yaklaşımda araç tespit için iki kameradan aynı anda alınan görüntüler birbirinden çıkarılarak elde edilen derinlik haritası kullanılır [5]. Bu yöntem çift kamera gerektirdiği için pahalıdır ve işlem yükü de fazladır. Görünüş temelli yaklaşımda ise araç ışıkları, yapı özelliği, renk, simetri, köşeler, aracın altında kalan gölge, yatay ve dikey kenarlar

gibi belirli dış görünüş özelliklerine göre tespit etme işlemi yapılır [6]. Bu yöntem için tek kamera yeterlidir. Sistemimiz gerçek zamanlı olarak çalışacağından daha az işlem yükü gerektiren görünüş temelli yaklaşım kullanılmış ve aracın altında kalan gölgeden yararlanılmıştır. Ayrıca öndeki herhangi bir renkli nesnenin tespiti için renk temelli maskeleyme yöntemi kullanılmıştır. Tespit edilen aracın ve renkli nesnenin mesafesi eğri uydurma yöntemi ve yapay sinir ağları ile hesaplanıp sonuçlar karşılaştırılmıştır.

1.1. LİTERATÜR TARAMASI

Bu çalışma, şeritten ayrılma uyarı sistemi ve önden çarpma uyarı sistemi olmak üzere iki bölümden oluşmaktadır. Bu nedenle literatür taraması şerit belirleme ve araç tespit etme olmak üzere iki başlık altında ele alınmıştır.

1.1.1. Şerit Belirleme

Yol şeritlerini tespit etmek için yapılan bir çalışmada Anik Saha, Dipanjan Das Roy, Tauhidul Alam ve Kaushik Deb, akıllı araçları kullanarak bir algoritma geliştirmeyi amaçlamıştır. Bu algorithmada ilk olarak renkli yol sahnesi görüntüsü gri ölçekli görüntüye dönüştürülür ve gri ölçekli görüntü üzerinde bağıntılı bileşenleri (yol yüzeyi) sınıflandırmak için flood-fill algoritmasını kullanılır. Algoritma tarafından elde edilen en büyük bağıntılı bileşen ile yol bölgesi bulunmuş olur. Sonra yolun dış kısmı gibi istenmeyen bölge tespit edilip çıkartılır. Elde edilen yol bölgesine beyaz şeritleri tespit etmek için filtre uygulanır. Fakat bu şeritleri tespit etme algoritmasının beyaz renk haricindeki şeritleri tespit edememe gibi bir dezavantajı vardır [7].

Bu çalışmada Michael Smart ve Steven L. Waslander kuşbakışı görüntüsünde çift filtreleme yöntemi kullanarak şerit belirleme işlemini gerçekleştirmişlerdir. Bu yöntemde ilk olarak kameradan alınan görüntü işlem yükünü azaltmak için gri ölçekli görüntüye dönüştürülür. Perspektif etkileri yok etmek için gri ölçekli görüntünün ters perspektif dönüşümü alınır. Böylece kuşbakışı görünüm elde edilir. Olası şerit çizgilerini tespit etmek kenarları belirlemede RANSAC algoritmasını kullanır. Belirlenen çizgilerin şerit çizgisi mi yoksa gürültüden kaynaklı başka

çizgiler mi olduğunu ayırt etmek için çift filtreleme yaklaşımı kullanılır. Bu yaklaşım iki çizgi arası önceden belirlenen değerden (25 cm) küçük ise şerit çizgisi olarak kabul eder, büyük ise bu çizgileri şerit çizgisine ait bir çizgi olmadığı için yok eder. Yolun dışında gürültüden dolayı ikili (25cm) çizgiler var ise şerit olarak algılanır. Bu da hataya neden olur. Ayrıca şerit tespit etmeyi iyileştirmek için çözünürlüğü yüksek bir kamera kullanmışlardır. Bu doğruluğu arttırırken sistemin yavaşlamasına sebep olur [8].

Bu çalışmada Amjad Jaleel Humaidi ve Mohammed Abdulraheem Fadhel şerit belirlemede kullanılan Hough dönüşümü ve renk eşikleme yöntemlerini karşılaştırmışlardır. Hough dönüşümü için görüntü önce gri ölçekli görüntüye dönüştürülür. Ardından kenar belirleme işlemi yapılır. Hough dönüşümü yöntemi ile şerit çizgilerinin değişkenleri tespit edilir. Renk eşikleme yöntemi şerit belirleme için önce görüntü RGB renk uzayından HSV renk uzayına dönüştürülür. HSV renk uzayında renk filtreleme işlemi yapmak daha doğru sonuçlar üretir. Şerit çizgileri beyaz renkte olduğundan beyaz rengi filtrelemek için uygun bir eşik değeri belirlenir. Bu eşik değerine göre filtreleme sonucu şerit tespit edilmiş olur. Hough dönüşümü gri ölçekli görüntünün sadece beyaz ve siyah renk özelliğini kullanırken renk değişimi renkli görüntünün renk yoğunluğunu kullanır. Bu yüzden Hough dönüşümün renk eşikleme yönteminden daha hızlı çalıştığı belirlenmiştir. Ayrıca renk eşikleme yönteminde çevresel koşullar eşik değeri değiştirebileceği için eşik değerinin tekrar tekrar belirlenmesi gerekebilir. Bu sebeple Hough dönüşümü daha etkili bir yöntemdir [9].

Sunil Kumar Vishwakarma, Akash ve Divakar Singh Yadav bu çalışmalarında şerit tespit etmede kullanılan Sobel kenar belirleme ve Canny kenar belirleme yöntemlerini analiz etmişlerdir. Bu iki yöntemi OpenCV görüntü işleme kütüphanesini kullanarak gerçekleştirdikleri program ile aracın önüne yerleştirdikleri kameradan alınan video üzerinde test etmişlerdir. Elde ettikleri test sonuçlarını karşılaştırmışlardır. Bu iki yöntemin performans analizini standart bilgisayarlı görü değerlendirme yöntemlerini kullanarak gerçekleştirmişlerdir. Bu analiz sonucunda Canny ile kenar belirlemenin Sobel ile kenar belirlemeden daha iyi sonuç verdiğini gözlemlemişlerdir [10].

Xue Li, Qingxiang Wu, Yu Kou, Lei Hou ve Heng Yang SPIKING (başaklama) sinir ağı kullanarak şerit belirleme yöntemi sunmuşlardır. İlk önce şeritlerin olduğu ilgili alan belirlenir. Kameradan alınan görüntünün üstündeki 1/3 kısım genelde gökyüzüne denk gelir. Bu kısımda şerit çizgileri olmayacağı için görüntüden çıkarılır. Seçilen ilgili alan görüntüsü daha sonraki işlemlerin hızla çalışması için gri ölçekli dönüştürülür. Ardından gri ölçekli görüntü histogram üzerinde germe işlemi yapılarak ve renkler arasındaki kontrastın artırılarak soluk şerit çizgilerin daha belirgin hale getirilmesi sağlanır. Gürültüleri azaltmak için medyan filtresi uygulanır. Şerit çizgilerinin kenarlarını belirlemek için SPIKING (başaklama) sinir ağı kullanılır. Gürültüsü giderilmiş görüntü yapay sinir ağlarına sokulur ve renk değişimlerin olduğu yerlerde ağ cevap verir. Böylece kenarlar tespit edilmiş olur. Son olarak çizgilerin parametrelerini belirlemek için Hough dönüşümü kullanılır. Bu yöntem diğer kenar belirleme yöntemlerine (Roberts, Sobel, Prewitt and Canny) göre nispeten daha iyi sonuç vermektedir [11].

Chien-Cheng Tseng, Hsu-Yung Cheng ve Bor-Shenn Jeng, geometri bilgisi ve modifiye edilmiş Hough dönüşümü kullanarak bir şerit belirleme algoritması sunmuşlardır. Bu algoritmada ilk olarak, kameradan yakalanan görüntü kameranın geometri bilgisini kullanarak yol olan kısım ve yol olmayan kısım olarak ikiye ayrılır ve renkli yol görüntüsü siyah-beyaz görüntüye dönüştürülür. Sonra şeritleri tespit etmek için yol geometri bilgisi göz önünde tutularak modifiye edilmiş Hough dönüşümü kullanılır. Modifiye edilmiş Hough dönüşümü parametre uzayında komple bir arama algoritması olduğundan çok zaman alıcıdır. Ayrıca şeritler yol olmayan kısımda keşiştiği zaman algoritma şeritleri belirleyemez [12].

Şeritlerin pozisyonlarını gerçek zamanlı olarak belirleyebilen bir sistem geliştiren Xiaodong Miao, Shunming Li ve Huan Shen, bu sistemi tek kamera kullanarak gerçekleştirmişlerdir. Bu sistemde kullanılan algoritma beş adımdan oluşur. İlk olarak kameradan alınan görüntüdeki tüm kenarları bulmak için kenar belirleme işlemi yapılır. Kenar belirleme işlemi için Canny kenar belirleme yaklaşımı kullanılmıştır. Elde edilen kenar görüntüsünde şerit çizgisi olma potansiyeli yüksek olan bölgeleri belirlemek için 3x3 boyutundaki belirli maskeler yardımı ile tüm görüntüde tarama yapılır. Bu tarama işlemi sonucunda elde edilen görüntüde şerit

çizgisi olmayan çizgilerin çoğu yok edilmiş olsa da tamamı yok olmamıştır. Daha sonra gelen potansiyel şerit çizgilerinin tahminini güçlendiren eşleşmiş kısımları birleştirmek için bağlayıcı koşul yaklaşımı kullanılır. Böylece şerit çizgileri belirlenmiş olur. Son olarak şeritlerin pozisyonlarını belirlemek için k-ortalama kümesi algoritması kullanılır [13].

M. Dhana Lakshmi ve B.J. Rani Deepika, yoldaki beyaz ve sarı renkteki şeritleri tespit etmek için özgün bir algoritma sunmuşlardır. Bu şerit belirleme algoritmasını gerçek zamanlı olarak uygulamışlardır. Şeritlerin hem renk bilgisini hem de kenar yönlerini kullandıkları için güçlü ve efektif bir şerit tespit etme algoritmasıdır. Algoritmanın renk kısmı sarı ve beyaz renkteki şeritleri tanımlar daha sonra bunu yok edilen sınırlar içindeki kenar yönleri takip ederek şeritler tespit edilir. Yol yüzeyine göre sabit olan kameranın yüksekliğinden dolayı görüntünün yol olan kısmı ile yol olmayan kısmı kolayca ayrılır. Böylece şeritlerin belirlenmesi daha verimli hale getirilir [14].

Hongying Zhao, Zhu Teng, Hong-Hyun Kim ve Dong-Joong Kang, tavlı parçacık filtresiyle çoklu görüntüleri birleştiren tavlı parçacık filtre algoritması temelli şerit belirleme ve izleme yöntemi sunmuşlardır. Her şerit çizgisi için ayrı ayrı kutu belirlenir. Bu kutu şerit çizgisinin konumuna göre yatay ekseninde hareket ettirilerek şerit çizgilerinin bu kutular içinde kalması sağlanır. Bu kutuların içindeki çizgiler ise renkler arası keskin geçişlerin olduğu duruma göre belirlenir. Geleneksel parçacık filtresine yerine tavlı parçacık filtresi kullanılarak zaman maliyeti büyük oranda azaltılmıştır [15].

Young Uk Yim ve Se-Young Oh üç özellik temelli şerit belirleme algoritması geliştirmişlerdir. Kullanılan özellikler başlama pozisyonu, yönü ve yoğunluk değeridir. Başlangıç adımında, kenar bilgisini almak için Sobel operatörü uygulanır. Şerit sınırları üç özelliği kapsayan bir vektör olarak temsil edilir. Giriş görüntüsü ve bir önceki şerit model vektörü dikkate alınarak o anki şerit vektörü hesaplanır. Sağ ve sol sınırların her biri için iki pencere kullanılır. Her yatay çizgide N piksel olduğunu farz edilirse, N adet şerit vektör adayı oluşturulur. Ağırlıklı mesafe ölçümü kullanılarak bir önceki şerit vektöründen en kısa mesafe baz alınarak en iyi aday

seçilir. Dengeleme için her bir özelliğe farklı bir ağırlık atanır. Sonra yeni şerit vektörünü tahmin etmek için şerit çıkarım sistemi kullanılır. Eğer yol genişliği aniden değişiyorsa, o anki hesaplanan vektör atılır ve bir önceki vektör o anki vektör olarak alınır [16].

Stephan Sehestedt, Sarath Kodagoda, Alen Alempijevic ve Gamini Dissanayake, kentsel çevre için şerit belirleme yaklaşımı sunmuşlardır. Karmaşık yol geometrisi, yıpranma ve aşınma sebebiyle şerit çizgileri net bir şekilde görünmediğinden dolayı şerit çizgilerini tespit etmek için zayıf model kullanılır. Ters perspektif görüntüsüne aşağıdan yukarıya parçacık filtresi uygulanır. Bu filtre çoklu şerit çizgilerini izlemek için ayarlanır [17].

Kentsel yollardaki şerit çizgilerini tespit etmek için gerçek zamanlı bir yaklaşım sunan Mohamed Aly, ilk olarak perspektif etkisinden kaçınmak için ters perspektif dönüşümü yaparak yol görüntüsünün kuşbakışı görüntüsünü oluştururlar. Bu kuşbakışı görüntüsü, yönlendirilmiş iki boyutlu Gauss maskesi ile filtrelenir. Filtre özellikle karanlık arka plandaki belirli genişliğe sahip parlak çizgileri belirlemek için uyarlanır. Böylece şerit çizgileri yüksek cevaplar üretir. Filtrelenmiş görüntüden belli bir eşik değerinin altında kalan kısımlar atılarak ve geriye kalan %q yüzdeler dilimi seçilerek sadece yüksek cevap veren çizgiler elde tutulur. Sonra düz çizgiler basitleştirilmiş Hough dönüşümü uygulanarak belirlenir. Böylece şerit çizgilerinin görüntü üzerindeki yeri belirlenmiş olur [18].

ZuWhan Kim soluk, eğri ve bölünmüş şerit çizgileri gibi zorlu koşulların üstesinden gelebilen bir şerit belirleme ve izleme algoritması geliştirmiştir. Başlangıç adımı olarak, şerit olmayan çizgileri yok etmek için gradyan ve yoğunluk tümsek belirleyicisi kullanılır. Belirlenen şeritlerden örnekler alınarak bu örneklerle yapay sinir ağı uygulanır. Tespit edilen şerit çizgilerinin pikselleri kübik eğri kullanılarak gruplanır. Rastgele şerit bölümlerinden varsayım oluşturulur. RANSAC algoritması bu varsayımı doğrular. Şerit izleme içinse parçacık filtresi kullanılır [19].

Şerit belirleme için Hsu-Yung Cheng, Chih-Chang Yu, Chien-Cheng Tseng, Kuo-Chin Fan, Jenq-Neng Hwang ve Bor-Shenn Jeng hiyerarşik bir algoritma öne

sürmüşlerdir. Bu algoritma özellik renk çıkarımına dayanarak yüksek boyutlu özellik noktalarını çıkarır. Bu özellik noktaları kullanılarak yolun yapısal (şerit çizgisi içeren yol) olup olmadığına (şerit çizgisi içermeyen yol) karar verilir. Sonra özellik noktalarına ilişkili bileşenler uygulanır. Bu çıkarılan özellik noktalarındaki ilişkiye bakılarak yapısal yoldaki şerit çizgilerini tespit edilir. Eğer yol yapısal olmaya yol ise şerit çizgisi olarak yolun sağ ve sol sınırları şerit çizgisi olarak ele alınır ve bulunur [20].

Abdulahkam. AM. Assidiq, Othman O. Khalifa, Md. Rafiqul Islam ve Sheroz Khan, gerçek zamanlı bir şerit belirleme algoritması sunmuşlardır. Başlangıçta, renkli görüntü gri ölçekli görüntüye dönüştürülür. Sonra görüntünün gölgeli kısımları yok edilerek gürültüden arındırılır. Kenar tespiti için modifiye edilmiş Canny kenar belirleme yöntemi kullanılır. Bu siyah-beyaz görüntüye Hough dönüşümü uygulanır. Hough dönüşümü sonucunda elde edilen kenar görüntüsü kullanılarak yatay çizgili şerit sınırı tarama işlemi yapılır. Tarama alt görüntü sınırındaki Hough çizgilerinin kesiştiği bölümden başlar. Tarama işlemi sonucunda kenarla karşılık gelen veri noktaları bulunur. Sonra elde edilen veri noktaları üzerinde hiperbol eş uydurma uygulanarak kenar çiftleri bulunur. Bu kenar çiftleri dikkate alınarak şerit çizgisi belirlenir [21].

Amol Borkar, Monson Hayes ve Mark T. Smith, Hough dönüşümüne ve yinelemeli eşlemeli filtreye dayanan bir şerit belirleme algoritmasını sunmuşlardır. Algoritmadaki ilk adım renkli görüntüyü gri ölçekli görüntüye dönüştürmek ve sonra bu görüntü üzerinde ters perspektif dönüşümü uygulamaktır. Siyah-beyaz görüntü oluşturmak için ters perspektif görüntüsüne uyarlamalı eşik uygulanır. Siyah-beyaz görüntü her biri bir şerit çizgisini içerecek şekilde tam ortadan ikiye bölünür. Bu görüntülere ayrı ayrı Hough dönüşümü uygulanır. Her bir şerit çizgisinin yaklaşık olarak merkezini bulmak için şerit boyunca her bir numuneye bir boyutlu eşlemeli filtre uygulanır. Her iki şerit çizgisinin merkezi yaklaşık olarak bulduktan sonra şerit tespiti için veri noktalarına RANSAC algoritması uygulanır. Böylece şerit çizgileri belirlenmiş olur. Şerit çizgilerini takip etmek için ise Kalman filtresi kullanılmıştır. [22].

1.1.2. Araç Tespit Etme

Shiva Kamkar ve Reza Safabakhsh çalışmalarında araç tespit etme ve tespit edilen araçları tiplere göre sınıflama işlemi yapılmıştır. Araç tespit etmek için aktif temelli model yaklaşımı kullanılmıştır. Bu model de daha önceden elde edilmiş farklı araç tiplerine ait eğitim görüntüleri kullanılarak bir taslak oluşturulur. Bu taslak ile tüm görüntü üzerinde tarama işlemi yapılır ve aday araç bölgeleri tespit edilir. Daha sonra bu aday araç bölgelerinin simetrikliği kontrol edilerek araç olup olmadığına karar verilir. Tespit edilen aracın uzunluğuna ve kapladığı alana bakılarak sınıflama işlemi yapılır. Bu sınıflama işlemi ile araçlar küçük (araba), orta (minibüs) ve büyük (otobüs, kamyon) olmak üzere üç kategoriye ayrılır. Bu metotta aracı tespit etmek için tüm görüntüyü taramak gerektiğinden yavaş çalışır [23].

Bu çalışmada Yao Lin, Ma Fang ve Duan Shihong üçlü-çerçeve farkı tabanlı hareketli araç tespit etme algoritması sunmuşlardır. Bu yöntemde iki çerçeve arasındaki fark kullanıldığından arka plan rengine yakın olan araçları (genellikle koyu renkli araçlar) tespit etmek zordur. Bu problemin etkisini azaltmak ve daha efektif araç tespit etmek için görüntüye lineer nokta işlemi uygulanır. Bu lineer nokta işlemi, giriş görüntüsü gri ölçekli görüntüye dönüştürüldükten sonra kontrast germe işlemi ile gerçekleştirilir. Genellikle bir görüntü dizisindeki hareketli alanı belirlemek için ikili çerçeve farkı kullanılır. Bu çalışmada ise üçlü çerçeve farkı kullanılmıştır. Burada kullanılan çerçeveler; n . çerçeve, $(n+1)$. çerçeve ve $(n+2)$. çerçevedir. $((n+1) - n)$ fark görüntüsü ile $((n+2) - (n+1))$ fark görüntüsünün çarpımı ile üçlü-çerçeve farklı görüntüsü elde edilir. Elde edilen bu görüntü ikili (siyah-beyaz) görüntüye dönüştürülür. Küçük gürültüleri (tespit edilen araç üzerinde oluşan delikler ve çevresindeki noktasal gürültüler) yok etmek için morfolojik işlem uygulanır. Bu yöntem harekete dayalı olarak araç tespit ettiği için kameranın sabit konumda olması gerekmektedir. Dolayısıyla kameranın araç üstünde olduğu uygulamalarda kullanılamaz [24].

Bu çalışmada Munkhjargal Gochoo, Damdinsuren Bayanduuren, Uyangaa Khuchit, Galbadrakh Battur, Tan-Hsu Tan, Sy-Yen Kuo ve Shih-Chia Huang sabit bir kameradan alınan görüntü üzerinde özgün bir morfolojik filtre uygulayarak araç

tespit etmişlerdir. Kameradan alınan görüntü ilk olarak belirlenen bir eşik değerine göre ikili (siyah-beyaz) dönüştürülür. Bu işlemden sonra aracın muhtemel sınırlayıcı kutusu belirlenir. Sınırlayıcı kutunun içinde araç olup olmadığını belirlemek için kutunun içindeki beyaz piksel sayısı bulunur. Bu sayı, tüm piksel sayısının yarısından fazla ise sınırlayıcı kutu içinde araç olduğu kabul edilir. Böylece bu sınırlayıcı kutu tüm görüntüyü tarayarak araç olan alanlar tespit edilir. Bu yöntem diğer morfolojik filtreler (yayma, aşınma, açma, kapama, medyan) ile karşılaştırılmıştır ve arka planda gürültüye daha az duyarlı olduğu saptanmıştır. Tüm görüntü önceden belirlenen sınırlayıcı kutu ile tarandığı için yavaştır. Ayrıca belirlenen sınırlayıcı kutunun boyutuna ve kutunun içinde araç olup olmadığına karar veren eşik değerine göre araç tespit etme doğruluğu düşüktür [25].

K.V. Arya, Shailendra Tiwari ve Saurabh Behwal yaptıkları çalışmada sabit bir konuma yerleştirilmiş kameradan alınan görüntü öncelikle gri ölçekli görüntüye dönüştürülmüştür. Sonra bu görüntü medyan filtresinden geçirilerek noktasal gürültüler yok edilmiştir. Hareketli nesnelere belirlemek için art arda gelen iki fotoğraf karesi arasındaki fark bulunmuştur. Belirlenen eşik değerine göre bu fark görüntüsüne eşikleme işlemi uygulanmıştır. Tespit edilen nesnenin kenarları daha net olarak belirlemek için morfolojik işlemlerden bir tanesi olan aşındırma işlemi uygulanmıştır. Nesnenin içindeki delikleri yok etmek için ise kapama işlemi uygulanmıştır. Son olarak tespit edilen nesnenin araç olup olmadığını belirlemek için Blob analizi yapılmıştır. Bu analiz diğer tespit edilen hareketli nesnelere (kuş, yaya, vs.) yok eder. Belirledikleri eşik değerine (0,4) göre, Blob alanının sınırlayıcı kutu alanına oranı bu eşik değerinden büyük eşit ise araç vardır diye kabul edilir. Bu yöntem kameranın sabit bir konumda olduğu uygulamalarda kullanılabilir [26].

Jin-Woo Park and Byung Cheol Song bu çalışmalarında gece vaktinde aracın arkasındaki ışıklardan yararlanarak araç tespit etme işlemi gerçekleştirmişlerdir. İlk olarak aracın önüne yerleştirilen kameradan alınan görüntüye kontrast germe işlemi uygulanır. Bu işlem araç ışıklarının daha belirgin hale gelmesini sağlar. Öndeki aracın arka ışık çifti belirlenerek araç tespit edilir. Işıkları belirlemek için renk filtreleme yöntemi yerine cascade sınıflandırıcı kullanmışlardır. Bu yöntemde önceden elde edilmiş örnek görüntüler ile eğitim yapılır. Eğitim sonucunda elde

edilen verileri ile tüm resim taranır eşleşme olan kısımlarda araç tespit edilmiş olur. Fakat veri kümesi gece vakti alınmış görüntüler ile eğitildiğinden dolayı algoritma gündüz vakti çalışmaz [27].

Bu çalışmada Hao Yu, Yule Yuan, Yueting Guo ve Yong Zhao araç belirleme işlemi iki kısımda ele almışlardır. İlk olarak araç olması muhtemel alanlar belirlenir. Sonra bu alanların, doğrulama yöntemiyle araç olup olmadığına karar verilir. Muhtemel alanlar belirlenirken görüntü gri ölçekli görüntüye dönüştürülür. Sonra, önceden belirlenen bir eşik değerine göre eşikleme işlemi yapılır ve aracın altında kalan gölge tespit edilir. Bu gölge yardımı ile araç sınırlayıcı kutusu belirlenir. Sınırlayıcı kutunun içinde simetri kontrolü yapılır. Böylece araç olup olmadığı kararlaştırılır. Tespit edilen bu araçlar, önceden alınmış pozitif (araç olan) ve negatif (araç olmayan) görüntüler ile eğitilmiş bir veri kümesi ile karşılaştırılarak doğrulama işlemi gerçekleştirilir [28].

Nilakorn Seenoung, Ukrit Watchareeruetai, Chaiwat Nuthong, Khamphong Khongsomboon ve Noboru Ohnishi bu çalışmalarında yoldan geçen araçları tespit etme ve bu araçları türüne (küçük boyut; bisiklet, motosiklet, orta boyut; araba, minibüs, büyük boyut; kamyon, otobüs) göre sınıflama işlemi gerçekleştirmişlerdir. Bu sistem; arka plan çıkarma, araç tespiti, araç özellik çıkarma ve araç sınıflama olmak üzere dört ana adımdan oluşur. Arka plan çıkarma işlemi Gauss karışım modeli kullanarak yapılır. Her bir fotoğraf karesindeki piksellerin yoğunluğunu tahmin etmeye yarayan bu model Gauss dağılımının toplamı ile ifade edilir. Arka planı çıkarılmış görüntü üzerinde morfolojik işlemler uygulanır ve gürültüler giderilir. Kamera sabit olduğundan yolun olduğu ilgili alan kolayca belirlenir. Bu ilgili alan üç ayrı bölüme ayrılır. Bölgelerin uzaklığına göre tahmini araç boyutu (yükseklik, genişlik) belirlenir. Bu boyutlara göre her bölgede farklı sınırlayıcı kutu kullanılarak tarama yapılır. Tarama işlemiyle araçlar tespit edilir. Araç sınıflama işlemi için tespit edilen her bir aracın yükseklik, genişlik, alan ve yükseklik/genişlik oranı belirlenir [29].

S. M. Smith ve J. M. Brady tarafından önerilen hareket temelli bir sistem hareket tespiti için özellik temelli optik akış kullanır. İlk olarak Harris [30] köşe belirleme

metodu ile görüntüdeki köşeler belirlenir. Sonra belirlenen özellikler optik akış alanları oluşturmak için birkaç görüntü karesi boyunca takip edilir. Akış değişimlerine göre alanlar bölümlere ayrılarak gruplanır. Son olarak araç sınırlayıcı kutusu ve bölümlere ayrılmış grupların alan ağırlık merkezleri hesaplanır ve araç olarak varsayılırlar. Sistem yüksek hesaplama zamanı gerektirir ve bu nedenle gerçek zamanlı performans elde etmek için özel donanım hızlandırıcı kullanmak gerekir [31].

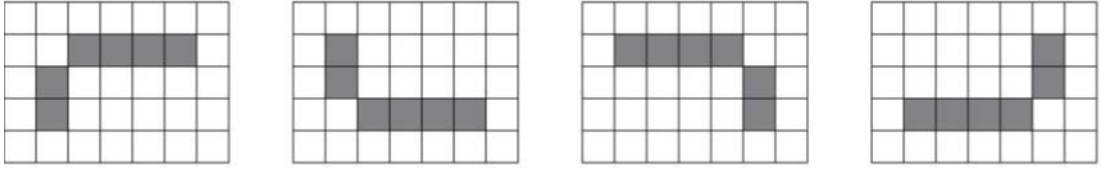
Cao Yanpeng, Alasdair Renfrew ve Peter Cook tarafından önerilen araç belirleme sisteminde optik akış alanı tahmin edilir. Bu uygulamada Üç Boyutlu Darbe Çifti Sınır Ağı [32] kullanarak akışları geliştiren bir teknik önermişlerdir ve ana araç ile önceki aracın bağıl hız farkına bağlı olarak hareket temelli tespit doğruluğunu göstermişlerdir. Küçük bağıl hıza sahip araçlar (< 10 km/h) düşük tespit oranlarına (%69,1) sahip olduğundan dolayı ayrıca araç belirleme için görünüş temelli teknikten de (araç altında kalan gölge) yararlanmışlardır [33].

Christos Tzomakas ve Werner von Seelen aracın altında gölgeden yararlanarak aracı tespit etmeye çalışmışlardır. Yoldaki gölgeli alanı belirlemek için bir eşik değerine ihtiyaç vardır. Bu eşik değerini belirlemek için görüntünün yol olan kısmının histogramı değerlendirilir. Sonra gölgenin bulunduğu pozisyona göre aracın pozisyonu belirlenir. Fakat doğru tespit yapmak için belirlenen gölgesinin bir araca mı ait yoksa gürültüden kaynaklı bir gölge mi ayırt edilmesi gerekir. Bunun için eşik değeri seçimi büyük önem taşımaktadır. Ayrıca çevresel etkilerden dolayı parklaklık değişebileceğinden eşik değeri seçimi zorlaşacaktır. [34].

Li-sheng Jin, Bai-yuan Gu, Rong-ben Wang, lie Guo, Yi-bing Zhao ve Lin-hui Li araç tespiti için iki özellik kullanmışlardır. İlk özellik olarak altındaki gölgeyi aracın olası yerini tahmin etmek için kullanmışlardır. Daha sonra aracın tam yerini belirlemek için yatay ve dikey kenarlardan yararlanmışlardır. Yatay ve dikey kenarlardan yararlanmanın bir dezavantajı arka plandaki nesnelere kaynaklanan kenarlar ile aracın kenarlarının ayırt edilmesi zordur [35].

Simetri araçların görsel karakteristiklerinden biridir. Araçlar arkadan bakıldığında tam ortadan dikey olarak simetrikdir. T. Zielke, M. Brauckmann ve W. V. Seelen bu simetri özelliğini kullanarak bir araç belirleme sistemi ortaya koymuşlardır. İlk olarak simetri eksenini belirler ve daha sonra belirlenen bu simetri eksenine göre simetri oranı bulunur. Bu simetri oranı belirlenen eşik değerinin üzerinde ise araç belirlenmiş olur. Simetri yaklaşımı kullanışlı olmasına rağmen fazla işlem yükü gerektirir [36].

Genel şekil itibarıyla araçlar dört köşeye sahiptir. M. Bertozzi, S. Broggi ve A. Castelluccio bu özelliği kullanarak araç belirleme yöntemi sunmuşlardır. Önce araç tespiti için görüntüdeki olası tüm köşeler Şekil 1.1'deki dört farklı maske kullanılarak bulunur. Tespit edilen tüm köşeler tiplerine ve yerlerine göre gruplandırılır. Son olarak gruplandırılmış her bir köşenin özellikleri Destek Vektör Makineleri (Support Vector Machines) sınıflandırıcısına giriş olarak verilir. Böylece bu köşelerin araca ait olup olmadığı belirlenir [37].



Şekil 1.1. Köşeleri belirlemek için kullanılan dört farklı maske çeşidi [37].

Çoğu araç yoldan ve arka planda farklı olarak renk değerlerine sahiptir. Dolayısıyla araçları arka plandan ayırmak için renk bilgisi kullanılabilir. Bu yöntem Tsai Luo-Wei, Hsieh Jun-Wei ve Fan Kuo-Chin tarafından kullanılarak araç tespit etme işlemi uygulanmıştır. Önemli araç renklerini bulmak için renk dönüşüm modeli sunmuşlar ve muhtemel araç yerlerini tespit etmişlerdir. Fakat renk temelli araç tespit etme parlaklık değişimine ve aracın yansıtma özelliklerine göre çok hassastır. Bu özellikler renk temelli araç belirleme işlemini oldukça zorlaştırırlar [38].

Gece araç tespit etmek için diğer özellikler belirsiz olduğundan aracın park lambaları kullanılır. Yen-Lin Chen, Yuan-Hsin Chen, Chao-Jung Chen ve Bing-Fei Wu, araçlara ait olan far veya park lambaları gibi parlak nesnelere tespit ederek araçları

belirleyen bir sistem sunmuşlardır. Bu sistem gece vakti çalışmak üzere geliştirilmiştir. Gündüz vakti çalışmaz [39].

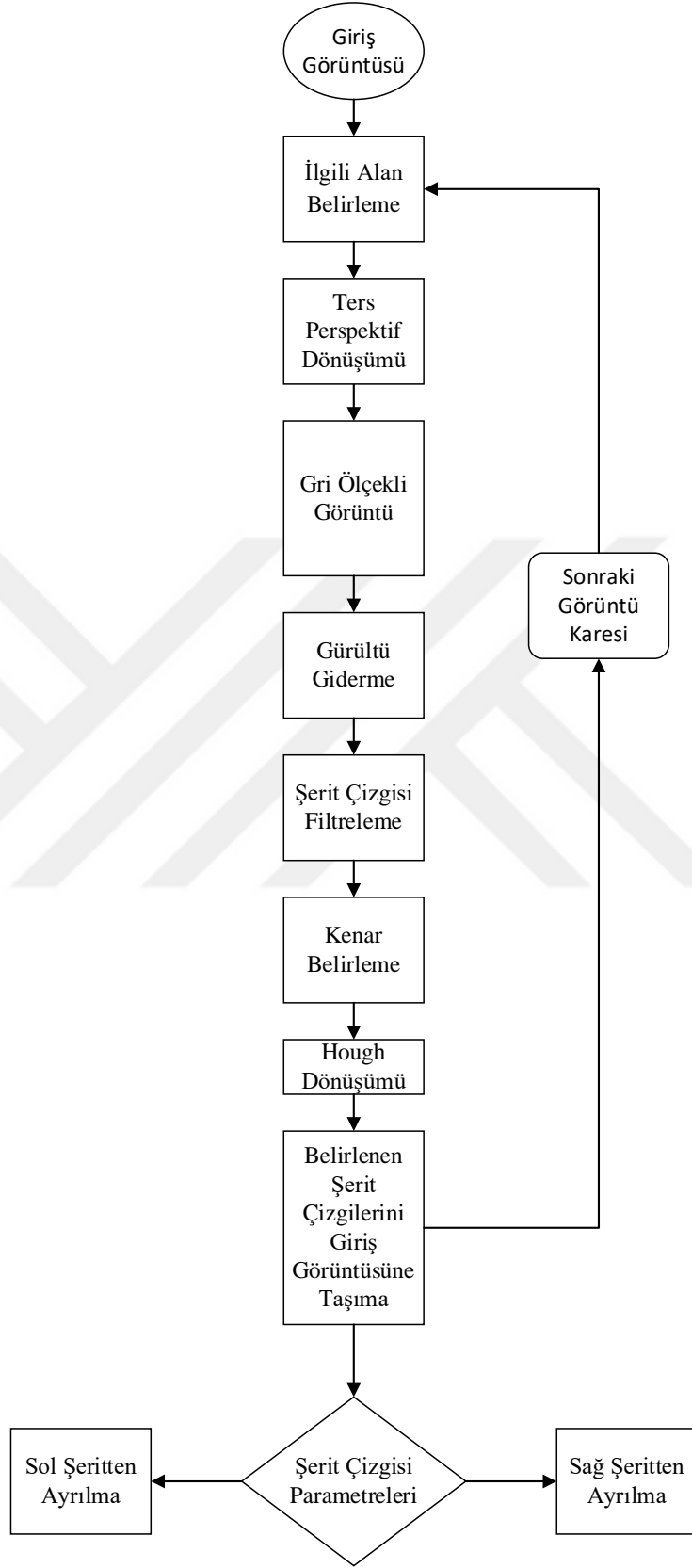
Chen Yen-Lin, Chen Yuan-Hsin, Chen Chao-Jung ve Wu Bing-Fei çift kamera temelli yaklaşım kullanarak araç tespit etme yöntemi sunmuşlardır. Bu yöntemde iki kameradan aynı anda alınan görüntüler birbirinden çıkarılarak derinlik haritası bulunur. Derinlik haritası görüntüsünde kameranın olduğu ilgili araca en yakın araç farklı renkte gözüktür. Bu özellik kullanılarak araç belirlenir. Fakat çift kameralı yöntemler hem pahalı hem de işlem yükü fazla olduğu için yavaş çalışır [5].



BÖLÜM 2

ŞERİTTEN AYRILMA UYARI SİSTEMİ

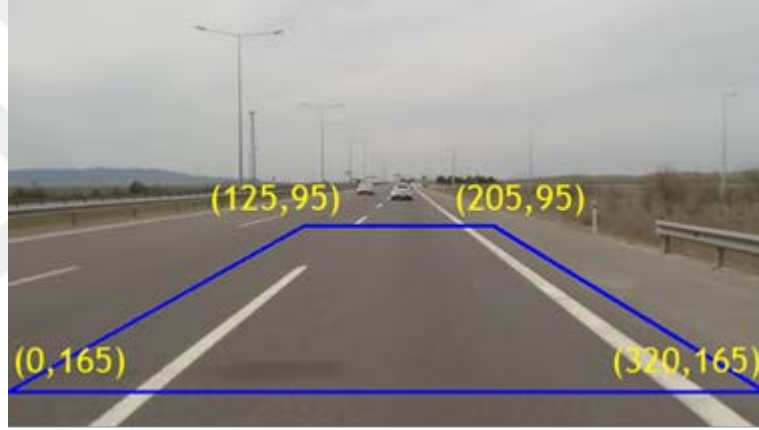
Bu bölümde şeritten ayrılma uyarı sistemi geliştirilmiştir. Bu sistemde, sürücü şeritten ayrıldığı zaman (istemi dışında, dalgalılık vs.) sesli olarak uyarı verilir ve olası kazalar önlemiş olur. Böyle bir sistemin ana konusu şerit belirlemedir. Şerit belirlendikten sonra şerit çizgilerinin durumlarına bakılarak hangi taraftan ayrıldığı tespit edilir ve sürücü uyarılır. Şerit belirleme işlemine geçmeden önce kameradan alınan 640x360 çözünürlüğündeki giriş görüntüsü şerit belirleme algoritmasının gerçek zamanlı bir uygulama için yeterli hızda olabilmesini sağlamak amacıyla 320x180 çözünürlüğüne dönüştürülmüştür. Şerit belirleme algoritmasının tüm adımları bu çözünürlük üzerinden yapılmıştır. Şeritten ayrılma uyarı sisteminin akış şeması Şekil 2.1’de gösterilmiştir.



Şekil 2.1. Şeritten ayrılma uyarı sisteminin akış şeması.

2.1. İLGİLİ ALAN BELİRLEME

İlgili alan, görüntü üzerinde şeridin olma potansiyelinin yüksek olduğu bölgedir. Kameradan giriş görüntüsü alındığında ilk olarak ilgili alan tanımlanır (Şekil 2.2). İlgili alan kameranın araca yerleştirildiği yükseklik ve eğim açısı değerleri dikkate alınarak belirlenir (Şekil 2.3). Şerit belirleme algoritması bu ilgili alan üzerinden uygulanarak hem işlem yükü büyük ölçüde azaltılır hem de şerit ile ilgisi olmayan kısımlar atılarak giriş görüntüsü gürültüden arındırılır. Böylece daha doğru sonuç elde etmemiz sağlanır. Ayrıca hafıza kullanımı da kayda değer bir oranda azalmış olur.



Şekil 2.2. Belirlenen ilgili alan.

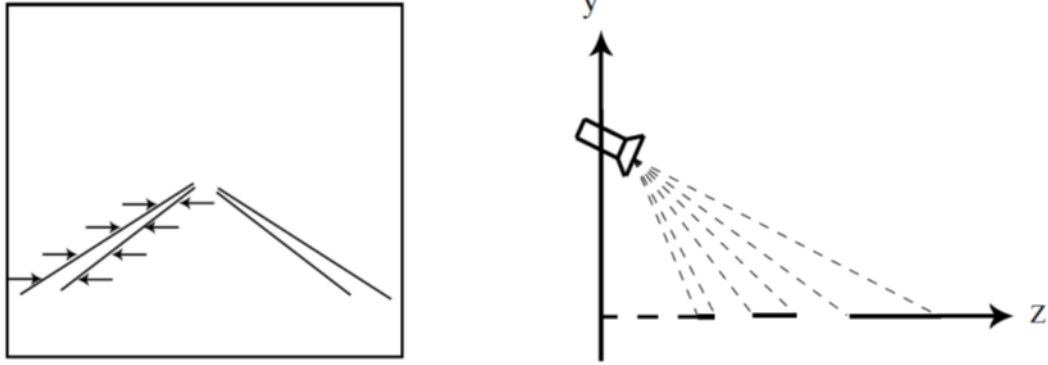


Şekil 2.3. Kameranın araca yerleştirildiği konum.

2.2. TERS PERSPEKTİF DÖNÜŞÜMÜ

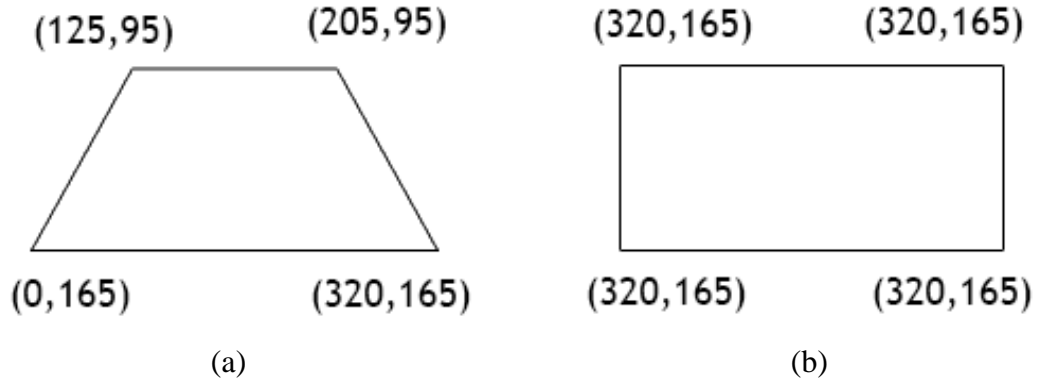
Son birkaç on yıl içinde, ters perspektif dönüşümü (TPD), özellikle akıllı ulaşım sistemleri alanında olmak üzere birçok probleme başarılı bir şekilde uygulanmıştır ve hala robotik, bilgisayarlı görü ve akıllı ulaşım sistemlerinin ilgi konusudur. TPD kullanarak yol tespiti ile ilgili literatürde birçok örnek mevcuttur [40–43].

Perspektif görüntünün sabit olmayan şerit çizgisi genişliği ve uzunluğu olmak üzere iki olumsuz etkisi vardır (Şekil 2.4). Bu tür perspektif etkilerini yok etmek şerit belirleme işlemini büyük ölçüde kolaylaştıracaktır. Genellikle, doğrudan giriş görüntüsü üzerinde şerit tespit etmek bahsi geçen bu iki olumsuz etkiden dolayı zor olacaktır. Dolayısıyla önce giriş görüntüsünün ters perspektif dönüşümü alınır daha sonra şerit tespit etme işlemi yapılır.



Şekil 2.4. Giriş görüntüsü üzerindeki perspektif etkileri, a) sabit olmayan şerit çizgisi genişliği, b) sabit olmayan şerit çizgisi uzunluğu [44].

Şekil 2.5'te sol taraftaki ikizkenar yamuk şekli görüntü düzlemini gösterir ve sağ taraftaki dikdörtgen alan ise gerçek dünya düzlemini gösterir.



Şekil 2.5. Koordinat sistemleri, a) görüntü koordinat sistemi, b) gerçek dünya koordinat sistemi.

TPD, noktaları görüntü düzleminden gerçek dünya koordinatlarına dönüştürmek için 3×3 'lük eşyazımlılık (homography) matrisi kullanır ve bu dönüştürme işlemi Denklem 2.1 ile gerçekleştirilebilir [45].

$$\begin{bmatrix} x^* \\ y^* \\ k \end{bmatrix} = H \begin{bmatrix} i \\ j \\ 1 \end{bmatrix} \quad (2.1)$$

Burada x^* ve y^* normalize edilmemiş gerçek dünya koordinatları, k normalizasyon faktörü, H 3×3 'lük eşyazımlılık matrisi, i ve j ise görüntü koordinat sistemindeki pikselin satır ve sütun indisleridir. Gerçek dünya koordinatlarını elde etmek için x^* ve y^* , k tarafından normalize edilmesi gerekir.

TPD, hesaplama zamanını azaltmak amacıyla tüm görüntü yerine Bölüm 2.1'de belirlenen ilgili alana uygulanır. İlgili alan görüntüsü ve onun TPD görüntüsü Şekil 2.6'da gösterilmiştir. Burada iki şerit çizgisinin, giriş görüntüsünde uzak bir noktada kesişeceği görünse de TPD görüntüsünde paralel olduğu görülebilir.



(a)



(b)

Şekil 2.6. İlgili alan görüntüsü ve onun TPD görüntüsü, a) ilgili alan görüntüsü, b) TPD görüntüsü

2.3. GRİ ÖLÇEKLİ GÖRÜNTÜ

Bu adımda, uygulanacak yöntemi hızlandırmak, daha az hesaba dayalı yapıp hesap karmaşıklığını azaltmak ve çevre koşullarına daha az duyarlı hale getirmek için kameradan alınan renkli giriş görüntüsü gri ölçekli görüntüye dönüştürülür.

Gri ölçekli görüntülerde her bir piksel değerinin yoğunluğu 0 ile 255 arasında bir değer ile temsil edilir. Burada 0 değeri siyaha, 255 değeri ise beyaza karşılık gelmektedir. Renkli görüntülerde ise her bir piksel değeri kırmızı, yeşil ve mavi olmak üzere üç bileşen ile ifade edilerek her bileşen 0 ile 255 arasında değer alır. Renkli bir görüntüyü gri ölçekli bir görüntüye dönüştürme işlemi Denklem 2.2 ile yapılabilir [46].

$$I(x, y) = 0.333 * R(x, y) + 0.5 * G(x, y) + 0.1666 * B(x, y) \quad (2.2)$$

Burada $R(x, y)$, $G(x, y)$, $B(x, y)$ sırasıyla renkli görüntünün (x, y) koordinatındaki pikselin sırasıyla kırmızı, yeşil ve mavi renk yoğunlukları, $I(x, y)$ ise gri ölçekli

görüntünün (x,y) koordinatındaki yeni pikselin yoğunluğudur. Şekil 2.6b'deki TPD görüntüsünün gri ölçekli görüntüye dönüştürülmüş hali Şekil 2.7'de gösterilmiştir.



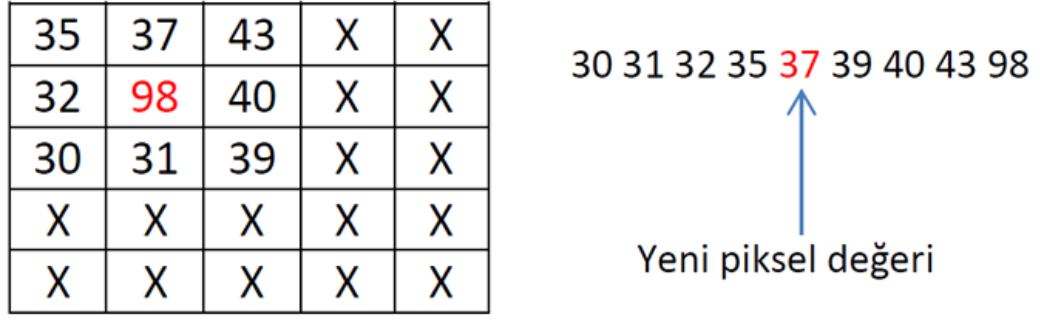
Şekil 2.7. Gri ölçekli görüntü.

2.4. GÜRÜLTÜ GİDERME

Gürültü tüm sistemler için bir problem olduğu gibi bilgisayarlı görü içinde bir problemdir. Geliştirilen algoritmaların ya gürültü toleransının olması gerekir ya da gürültüden arındırılmış olması gerekir. Gürültü giderme için kullanılan çeşitli filtreler mevcuttur ve bu filtreler aşağıda açıklanmıştır.

2.4.1. Medyan (Ortanca) Filtresi

Medyan filtresi görüntüdeki veya seçili bölgedeki piksellerin parlaklıklarını harmanlar ve istenmeyen parazitleri azaltır. Ortanca değeri kullanarak filtreleme işlemini gerçekleştirir. Özellikle, tuz-biber gürültüsü durumunda yani resim üzerinde siyah beyaz noktalardan oluşan gürültüleri ortadan kaldırmada etkilidir. Şekil 2.8'de medyan filtresinin uygulanışı gösterilmiştir. $(2,2)$ koordinatındaki 98 yoğunluk değerine sahip piksel, medyan filtresi uygulandıktan sonra 37 yoğunluk değerini alır.



Şekil 2.8. Medyan filtresinin uygulanışı.

2.4.2. Ortalama Filtresi

Ortalama filtresi, komşu pikseller arasındaki büyük parlaklık değeri farklarını aritmetik, geometrik veya harmonik ortalama kullanarak azaltıp görüntü üzerindeki istenmeyen gürültüleri yok eder. Bu filtre, görüntüyü belirlenen maske matrisin boyutlarında komşuluk olarak tarayacak, her bir matrisin ortalamasını alarak bu değeri matrisin orta elemanı olarak seçecektir.

2.4.3. Gauss Filtresi

Gauss filtresi Ortalama filtresine benzemektedir. Gauss filtresi uygulanan görüntüyü belirlenen maske boyutunda bir matris ile tarayarak her bir adım için ağırlıklı ortalamayı hesaplar. Bu sebeple Gauss filtresi Ortalama filtresine göre daha yumuşak bir bulanıklaştırma sağlar. Bu nedenle uygulamamızda Gauss filtresi tercih edilmiştir. Gauss filtresi görüntüye Denklem 2.3 ile uygulanabilir.

$$I' = G * I \quad (2.3)$$

Burada I' gürültüden arındırılmış görüntü, G 5x5'lik Gauss filtre maskesi ve I ise gri ölçekli görüntüdür. G Gauss filtre maskesini oluşturmak için Denklem 2.4'te verilen iki boyutlu Gauss fonksiyonundan yararlanır.

$$g(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}} \quad (2.4)$$

Burada x ve y maskenin her bir deęerinin satır ve sütun indisleri, σ Gauss fonksiyonunun keskinliğini ya da yumuşaklığını belirleyen Gauss dağılımının standart sapma deęeri, $\frac{1}{2\pi\sigma^2}$ ise normalizasyon katsayısıdır. $\sigma = 1$ alındığında 5×5 boyutundaki Gauss filtre maskesi Denklem 2.5'teki gibi elde edilir.

$$G = \frac{1}{273} \begin{bmatrix} 1 & 4 & 7 & 4 & 1 \\ 4 & 16 & 26 & 16 & 4 \\ 7 & 26 & 41 & 26 & 7 \\ 4 & 16 & 26 & 16 & 4 \\ 1 & 4 & 7 & 4 & 1 \end{bmatrix} \quad (2.5)$$

Bölüm 2.3'te elde ettiğimiz gri ölçekli görüntüye 5×5 'lik Gauss filtresi uygulayarak arka plan gürültüleri giderilir. Gauss filtresi sonucunda elde edilen görüntü Şekil 2.9'da gösterilmiştir.



Şekil 2.9. Gürültüsü azaltılmış görüntü.

2.5. ŞERİT ÇİZGİSİ FİLTRELEME

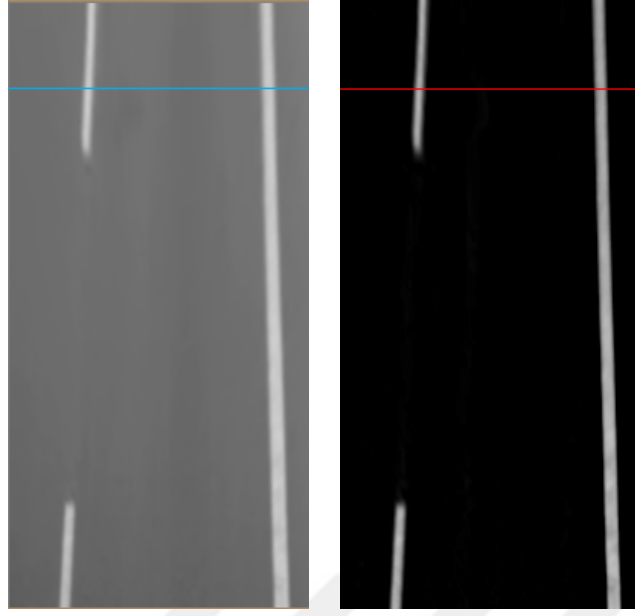
Şerit çizgileri görüntünün üst kısmında eğimli olsa da alt kısmında yaklaşık olarak düzdür. Ayrıca perspektif etkisinden dolayı görüntünün üst kısmına doğru çıktıkça şerit çizgisinin genişliği azalır ve kesikli şerit çizgisi göz önüne alındığında boyu da kısalmır. Bölüm 2.2'de görüntüyü perspektif etkisinden kurtarmak için TPD

uygulanmıştı. Böylece görüntünün alt ve üst kısmında şerit genişliği ile boyunun sabit kalması sağlanmıştı. Bu sayede, şerit çizgilerini filtrelemek için şerit çizgileri tek boyutlu bir uzayda kabul edilerek görüntünün her bir satırına, daha koyu alanlar tarafından çevrelenmiş yüksek yoğunluklu değerleri bulmaya yarayan spesifik bir şerit çizgisi filtreleme yöntemi uygulanır [47].

Her bir satırın yoğunluk değerleri $(x_i)_{i=1}^W$ ile ifade edilen görüntüye birbirinden bağımsız olarak Denklem 2.6'daki filtre uygulanmış ve filtrelenmiş veriler dizisi ise $(y_i)_{i=1}^W$ ile tanımlanmıştır [47].

$$y_i = 2x_i - (x_{i-\tau} + x_{i+\tau}) - |x_{i-\tau} - x_{i+\tau}| \quad (2.6)$$

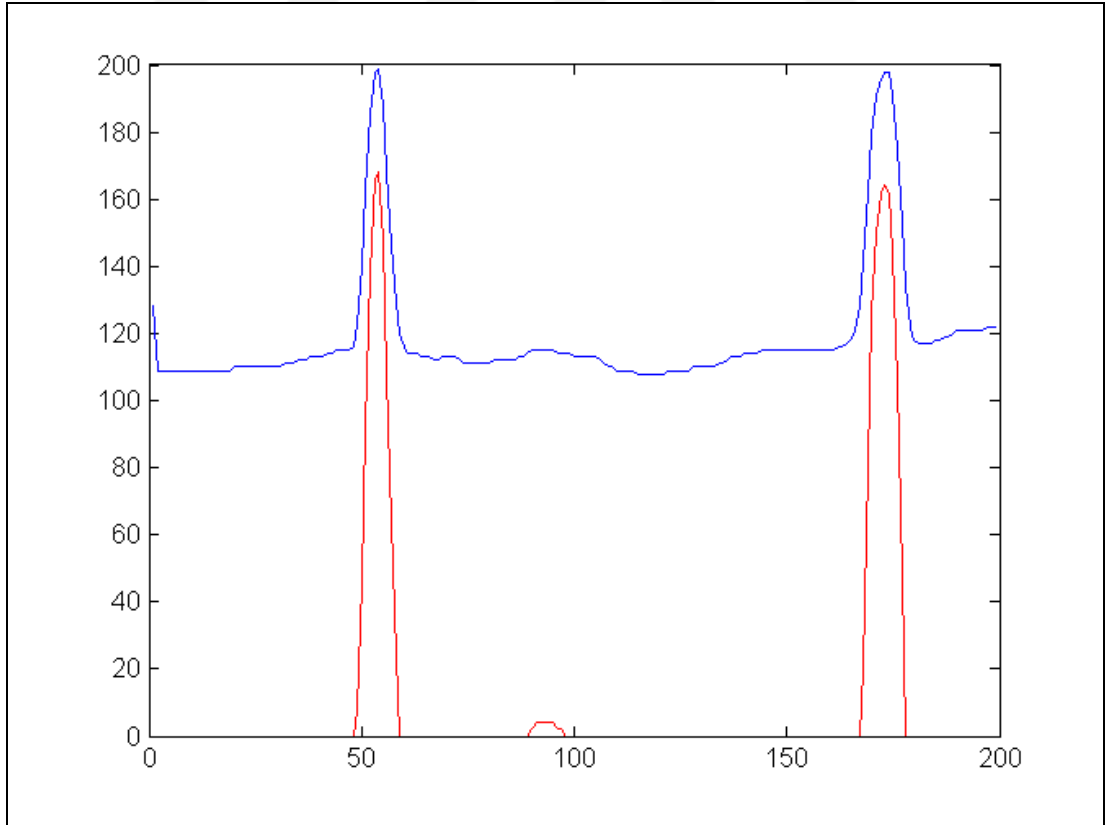
Burada τ tahmini şerit çizgisi genişliğinin yarısıdır. Bu filtre, sağ ve sol τ mesafesindeki komşu piksellerinden daha büyük x_i değerleri sayesinde pozisyonlar için yüksek cevaplar üretir. Denklem 2.6'daki son terim, benzer sağ ve sol komşular sayesinde pozisyonlara daha yüksek cevap verilebilsin diye sağ ve sol komşuların farkının yüksek olduğu durumları azaltır. Böylece hata azaltılır. Şekil 2.10'da gürültüsü azaltılmış görüntü ve onun filtrelenmiş görüntüsü gösterilmiştir. Ayrıca Şekil 2.10'daki görüntülerin bir satırının analizi Şekil 2.11'de gösterilmiştir.



(a)

(b)

Şekil 2.10. Şerit çizgisi filtreleme, a) gürültüsü azaltılmış görüntü, b) filtrelenmiş görüntü.



Şekil 2.11. Satır analiz grafiği.

2.6. KENAR BELİRLEME

Nesnelerin tanımlanması için sınırlarının belirlenmesi gerektiğinden kenar bulma, görüntü işleme ve görüntü analiz sistemleri için en önemli aşamalardan biri olmuştur. Görüntü içinde belirgin geçişler nesne kenarlarını gösterdiğinden, kenar bulma algoritmaları bu geçişleri bulmayı amaçlamaktadır. Görüntüye ilişkin kenarların belirlenmesi birçok durumda kullanışlıdır. Nesne tanıma problemi buna bir örnek olarak verilebilir. Kenar belirleme yöntemlerinden en çok kullanılanları Roberts Cross, Sobel ve Prewitt'tir. Daha gelişmiş bir teknik ise Canny kenar belirleme yöntemidir.

2.6.1. Robert Cross Kenar Belirleme Operatörü

Bu yöntemde iki adet konvolüsyon maskesi kullanılır. Daha çok çapraz kenarları belirlemek için kullanılan bir yöntemdir. Robert Cross maskeleri Şekil 2.12'de gösterilmiştir.

1	0
0	-1

(a)

0	-1
1	0

(b)

Şekil 2.12. Robert Cross maskeleri, a) sol çapraz Robert Cross maskesi, b) sağ çapraz Robert Cross maskesi.

P ₁	P ₂
P ₃	P ₄

Şekil 2.13. Maske altında kalan görüntü pikselleri.

Önce gradyanlar (maske uygulandıktan sonraki yoğunluk değerleri) Denklem 2.7 ile herhangi bir görüntü pikseli için hesaplanır. Daha sonra Denklem 2.8 ile gradyan genlikleri hesaplanarak kenarlar bulunur.

$$\begin{aligned} G_x &= (P_1 - P_4) \\ G_y &= (P_3 - P_2) \end{aligned} \quad (2.7)$$

$$g = \sqrt{G_x^2 + G_y^2} \quad (2.8)$$

2.6.2. Sobel Kenar Belirleme Operatörü

Bu yöntemde iki adet konvolüsyon maskesi kullanılır. Bunlardan birisi yatay kenarları bulmaya yarararken diğeri ise dikey kenarları bulmaya yarar. Bu maskeler görüntü içerisinde ışık yoğunluk değişiminin ani olduğu yerleri belirlememize yardımcı olur. Bu yatay ve dikey Sobel maskeleri Şekil 2.14'te gösterilmiştir.

-1	-2	-1
0	0	0
1	2	1

(a)

-1	0	1
-2	0	2
-1	0	1

(b)

Şekil 2.14. Sobel maskeleri, a) yatay sobel maskesi, b) dikey sobel maskesi.

P₁	P₂	P₃
P₄	P₅	P₆
P₇	P₈	P₉

Şekil 2.15. Maske altında kalan görüntü pikselleri.

Önce gradyanlar Denklem 2.9 ile herhangi bir görüntü pikseli için hesaplanır. Daha sonra Denklem 2.8 ile gradyan genlikleri hesaplanarak kenarlar bulunur.

$$\begin{aligned}
G_x &= (P_7 + 2P_8 + P_9) - (P_1 + 2P_2 + P_3) \\
G_y &= (P_3 + 2P_6 + P_9) - (P_1 + 2P_4 + P_7)
\end{aligned}
\tag{2.9}$$

2.6.3. Prewitt Kenar Belirleme Operatörü

Bu yöntemde iki adet konvolüsyon maskesi kullanılır. Bunlardan birisi yatay kenarları bulmaya yararken diğeri dikey kenarları bulmaya yarar. Sobel operatörü ile aynı işlevi yerine getirir. Sobele göre gürültüye daha fazla duyarlı fakat uygulamaya koyması daha basittir. Prewitt maskeleri Şekil 2.16’da gösterilmiştir.

-1	-1	-1
0	0	0
1	1	1

(a)

-1	0	1
-1	0	1
-1	0	1

(b)

Şekil 2.16. Prewitt maskeleri, a) yatay prewitt maskesi, b) dikey prewitt maskesi.

P₁	P₂	P₃
P₄	P₅	P₆
P₇	P₈	P₉

Şekil 2.17. Maske altında kalan görüntü pikselleri.

Önce gradyanlar Denklem 2.10 ile herhangi bir görüntü pikseli için hesaplanır. Daha sonra Denklem 2.8 ile gradyan genlikleri hesaplanarak kenarlar bulunmuş olur.

$$\begin{aligned}
G_x &= (P_7 + P_8 + P_9) - (P_1 + P_2 + P_3) \\
G_y &= (P_3 + P_6 + P_9) - (P_1 + P_4 + P_7)
\end{aligned}
\tag{2.10}$$

2.6.4. Canny Kenar Belirleme

Canny kenar belirleme algoritması 1986 yılında John F. Canny tarafından geliştirilmiştir [48]. Günümüzde ise kenar belirleme yöntemleri arasında bir standart haline gelmiş ve birçok araştırmada yaygın olarak kullanılmaktadır.

Canny kenar belirleme algoritması beş ayrı adımdan oluşur [49]. Bu beş adım Şekil 2.18'deki örnek görüntü üzerinde uygulanacaktır.



Şekil 2.18. Canny kenar belirleme yöntemi için kullanılan örnek giriş görüntü [49].

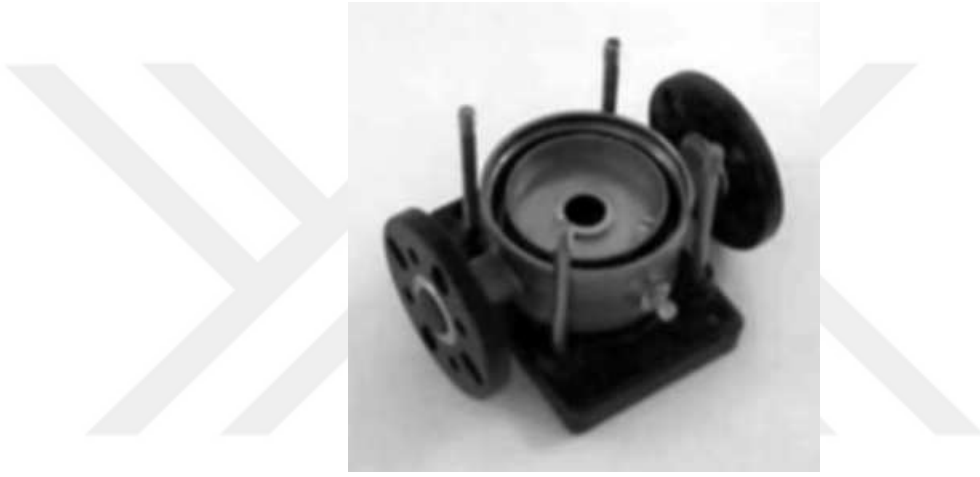
Yumuşatma

Yumuşatma, gürültüleri yok etmek için görüntünün bulanıklaştırılmasıdır. Kaçınılmaz olarak bir kameradan alınmış görüntü bir miktar gürültü içerir. Bu gürültü kenarların belirlenmesinde hatalara neden olabilir. Bu nedenle gürültünün azaltılması gerekmektedir. Gürültünün azaltılması için yumuşatılması için görüntüye Gauss filtresi uygulanır. Bu işlem için standart konvolüsyon yöntemi kullanılır. Gauss maskesi çoğunlukla görüntüden küçüktür. Bu maske görüntü üzerinde kaydırılarak işlem yapılır ve her bir işlemde maskenin merkezine karşılık gelen piksel güncellenir. Gauss maskesinin genişliği büyüdükçe, gürültüye karşı olan hassasiyet azalır.

Gauss filtresi Bölüm 2.4.3'te detaylı olarak anlatılmıştır. Buna göre standart sapma değeri $\sigma = 1,4$ olan Gauss maskesi Denklem 2.11'teki gibi elde edilir.

$$G = \frac{1}{159} \begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix} \quad (2.11)$$

Yumuşatma sonucunda oluşan görüntü Şekil 2.19'da gösterilmiştir.



Şekil 2.19. Yumuşatma sonucunda oluşan görüntü [49].

Gradyanların Bulunması

Kenarlar, görüntünün gradyan genliklerinin büyük olduğu yerde olur. Gradyanın genliği aynı zamanda kenar gücü olarak da bilinir. Kısmi türev yaklaşımı kullanılarak bulunur. Yumuşatılmış görüntüdeki her bir piksele ait gradyanlar, Sobel operatörü uygulanarak bulunur. Şekil 2.14'te gösterilen Sobel maskelerin her biri yatay ve dikey yönde olmak üzere sırasıyla uygulanır ve gradyanlar Denklem 2.9 ile herhangi bir görüntü pikseli için hesaplanır. Daha sonra Denklem 2.8 ile gradyan genlikleri hesaplanarak kenarlar bulunur. Şekil 2.20'da gradyan genliği görüntüsünde, kenarlar açık bir şekilde gözükse de kenarların kalın olmasından dolayı kenarların tam olarak nerede olduğu belirlenememektedir. Bunu önlemek için kenar vektörünün yönü Denklem 2.12 kullanılarak hesaplanır.

$$\theta = \tan^{-1} \left(\frac{|G_x|}{|G_y|} \right) \quad (2.12)$$

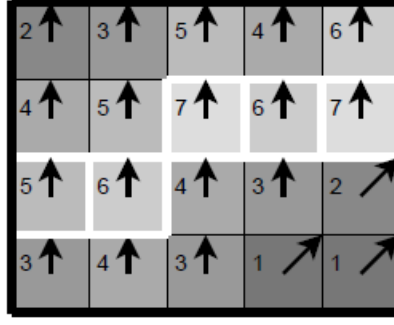


Şekil 2.20. Gradyan genliği görüntüsü [49].

Maksimum Olmayanları Yok Etme

Maksimum kenarları yok etme, bir kenar inceltme tekniğidir. Dikkat edilirse Şekil 2.20'deki gradyan genliği görüntüsünde kenarlar bulanıktır. Bu bölümün amacı gradyan genliği görüntüsündeki bulanık kenarları keskin kenarlara dönüştürmektir. Temel olarak bunu gerçekleştirmek için gradyan genliği görüntüsündeki tüm yerel maksimumları saklanıp diğer değerler silinir. Bu işlemi gerçekleştirmek için gradyan genliği görüntüsündeki her bir piksele uygulanacak algoritma sırasıyla aşağıdaki gibidir.

1. Gradyan yönü (θ) en yakın 45° ve katlarına yuvarlanır. Başka bir ifadeyle, 0° - $22,5^\circ$ ve $157,5^\circ$ - 180° arası 0° 'ye, $22,5^\circ$ - $67,5^\circ$ 45° 'ye, $67,5^\circ$ - $112,5^\circ$ arası 90° 'ye, $112,5^\circ$ - $157,5^\circ$ arası 135° 'ye yuvarlanır.
2. İlgili pikselin kenar gücüyle pozitif ve negatif gradyan yönündeki pikselin kenar gücü karşılaştırılır. Örneğin; ilgili pikselin gradyan yönü kuzey ($\theta=90^\circ$) ise, güney ve kuzey pikselleriyle karşılaştırılır.
3. Eğer ilgili pikselin kenar gücü en büyük ise, kenar gücü değerini korunur, değil ise yok edilir.



Şekil 2.21. Maksimum olmayanları yok etme gösterimi [49].

Şekil 2.21’deki örnekte yukarı olan okların olduğu pikseller bir üst (ok yönü – pozitif yön) ve bir alt (ok yönünün tam tersi – negatif yön) piksellerin kenar güçleriyle karşılaştırılır. Eğer ilgili pikselin gücü her ikisinden de büyük ise kenar gücü korunur ve kenar olarak tanımlanır. En büyük değerler yani kenar olarak tanımlanan pikseller beyaz sınır çizilerek gösterilmiştir. Gradyan yönleri ok ile kenar güçleri ise hücre içindeki değerler ile gösterilmiştir.

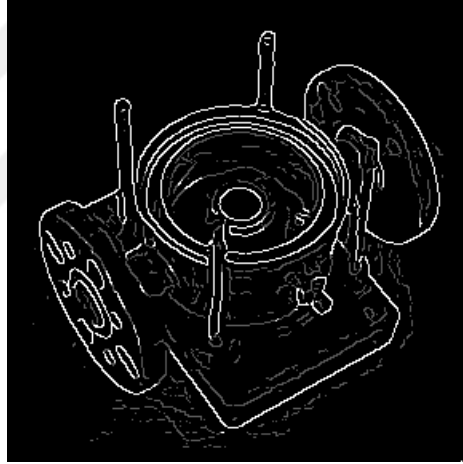
Maksimum olmayanları yok etme adımının görüntü üzerindeki etkisi Şekil 2.22’de gösterilmiştir.



Şekil 2.22. Sadece yerel maksimumu olan piksellerin korunduğu görüntü [49].

Çift Eşikleme

Maksimum olmayanları yok etme adımından sonra elde edilen görüntüde halen pürüzlü yüzeyler gibi gürültülerden kaynaklanan hatalı kenar belirlemeler olmuş olabilir. Bu gürültüyü yok etmek için çift eşikleme yöntemi kullanılır. Tek eşik değeri kullanılması durumunda eşğin altında kalan kenarlar siyaha çevrilerek yok edilebilirdi. Aynı şekilde kesikli çizgiler de kenar olarak belirlenebilirdi. Bunu önlemek için iki eşik kullanmak gerekir. Yüksek eşikten büyük olan pikseller güçlü, iki eşik arasında olan pikseller zayıf olarak nitelendirilir ve düşük eşikten küçük olan pikseller ise siyaha çevrilerek yok edilir. Şekil 2.23'de 20 ve 80 eşik değerlerinin Şekil 2.22'deki görüntü üzerindeki etkisi gösterilmiştir.



Şekil 2.23. Çift eşikleme görüntüsü [49].

Histerezis ile Kenar İzleme

Güçlü kenarlar, yüksek eşikten büyük olanlar, doğrudan kenar olarak belirlenerek son kenar görüntüsüne işlenir. Zayıf kenarlar, yüksek ve düşük eşik arasında olanlar, ise sadece güçlü kenar ile ilişkili olanlar kenar olarak kabul edilerek son kenar görüntüsüne işlenir. Kenar izleme BLOB (Binary Large Object) ile yapılır. BLOB 8-ilişkili komşuluğu kullanır. Eğer BLOB bir tane güçlü piksel içeriyorsa kenar olarak kabul edilir, hiç güçlü piksel yoksa yok edilir. Kenar izlemenin Şekil 2.23'deki çift eşikleme görüntüsü üzerindeki etkisi Şekil 2.24'de gösterilmiştir. Burada güçlü

kenarlar beyaz renk ile, güçlü kenarlara bağlı olan zayıf kenarlar mavi ile, diğer zayıf kenarlar ise kırmızı ile gösterilmiştir.

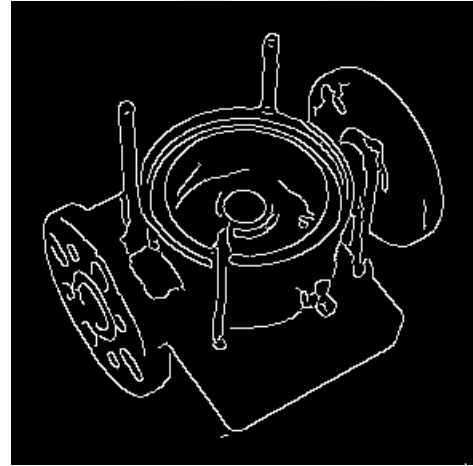


Şekil 2.24. Histerez ile kenar izleme görüntüsü [49].

Canny kenar belirleme yöntemi için ise son sonuç görüntüsü Şekil 2.25’de gösterilmiştir.



(a)

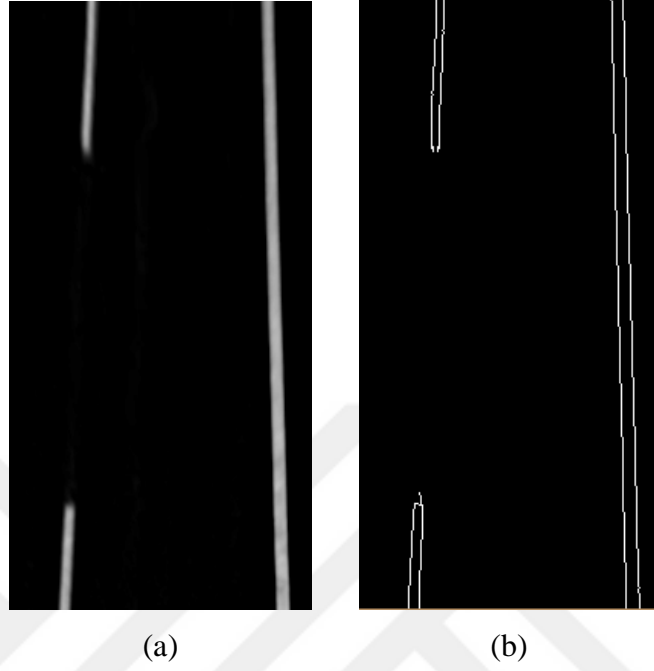


(b)

Şekil 2.25. Örnek görüntü üzerinde Canny kenar belirleme, a) giriş görüntüsü, b) sonuç görüntüsü [49].

Diğerlerine göre daha gelişmiş bir teknik olduğu için bu çalışmada Canny kenar belirleme tekniği tercih edilmiştir. Bölüm 2.5’te şerit çizgisi filtreleme sonucunda

elde edilen görüntüye Canny kenar belirleme yöntemi uygulandığında elde edilen sonuç görüntüsü Şekil 2.26'da gösterilmiştir.



Şekil 2.26. Canny kenar belirleme, a) giriş görüntüsü, b) sonuç görüntüsü.

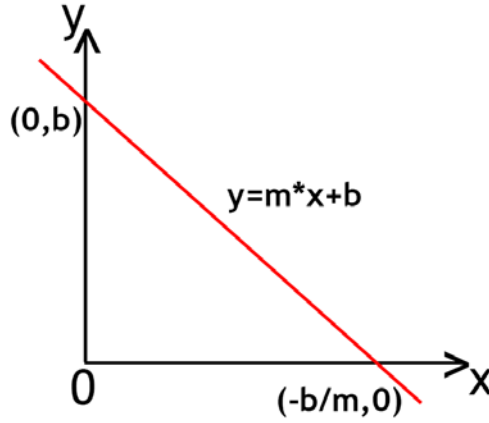
2.7. HOUGH DÖNÜŞÜMÜ

Hough dönüşümü, siyah-beyaz görüntülerde doğruların ve dairesel şekillerin tespiti için sıkça kullanılan bir tekniktir. Bu teknik 1962 yılında Paul Hough tarafından geliştirilmiştir ve ilgili nesnenin sınırlarını belirlemede etkili bir yaklaşım sunar. Yoldaki şerit çizgileri hemen hemen düz olduğundan, bu bölümdeki ana odak noktamız, siyah-beyaz görüntüde Hough Dönüşümü kullanarak doğruları bulmak olacaktır. Bölüm 2.5'te şerit çizgileri filtrelenmişti ve Bölüm 2.6'da ise bu şerit çizgilerinin kenarları tespit edilmişti. Dolayısıyla bu bölümde bulunan doğrular sağ ve sol şerit çizgileri olacaktır.

Bir doğru, iki boyutlu koordinat sisteminde birçok yol ile ifade edilebilir. Örneğin, Kartezyen koordinat sisteminde bir doğru Denklem 2.13 ile ifade edilir.

$$y = m * x + b \quad (2.13)$$

Burada m doğrunun eğimi ve b ise doğrunun y -eksenini kestiği noktadır (Şekil 2.27).

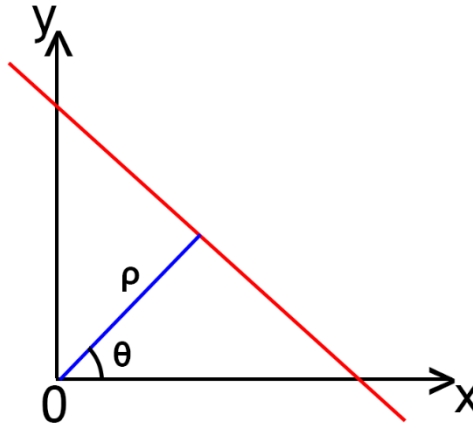


Şekil 2.27. Kartezyen koordinat sisteminde bir doğru.

Fakat Denklem 2.13'teki formülde m , dik doğrular için sonsuz olacağından Kartezyen koordinat sistemi yerine Kutupsal koordinat sistemi kullanılır. Kutupsal koordinat sisteminde bir doğru Denklem 2.14 ile ifade edilir.

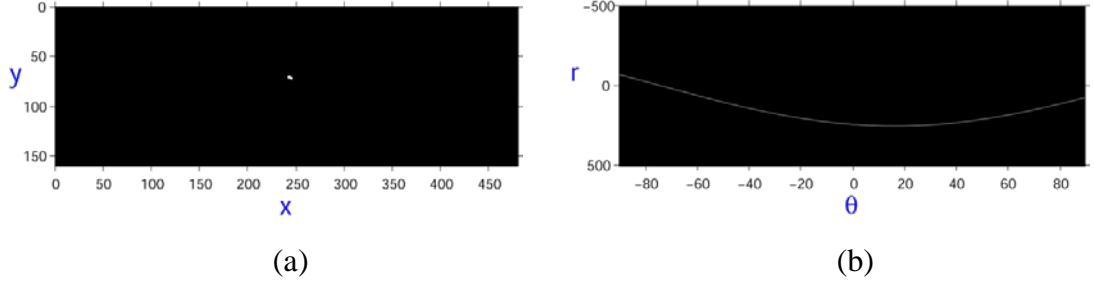
$$\rho = x * \cos(\theta) + y * \sin(\theta) \quad (2.14)$$

Burada ρ doğrunun orijinden olan uzaklığıdır. θ ise orijinden geçen doğrunun x düzlemiyle yaptığı açı olup 90° ile -90° arasında değer alır (Şekil 2.28). Bu iki sistem arasında dönüşüm kolaylıkla yapılabilir.



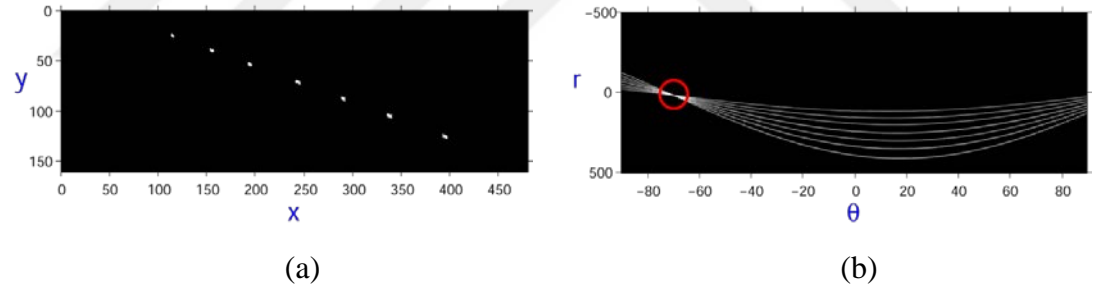
Şekil 2.28. Kutupsal koordinat sisteminde bir doğru.

Kartezyen koordinat sistemindeki her bir nokta (x_0, y_0) (Şekil 2.29a), Kutupsal koordinat sisteminde (sözde Hough uzayı) bir sinüs eğrisi verir (Şekil 2.29b).



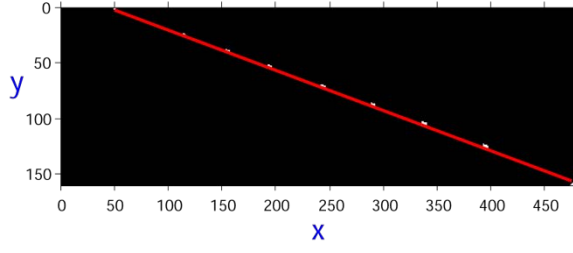
Şekil 2.29. Örnek bir noktanın iki koordinat sisteminde gösterimi, a) Kartezyen koordinat sisteminde, b) Kutupsal koordinat sisteminde [50].

Kartezyen koordinat sistemindeki aynı doğru üzerinde uzanan noktaların (Şekil 2.30a), Kutupsal koordinat sistemindeki sinüs eğrisi dönüşümleri bir noktada kesişirler (Şekil 2.30b). Kesiştikleri bu noktadaki uzaklık ve açı değeri, Kartezyen koordinat sistemindeki noktalardan geçen doğruyu verir.



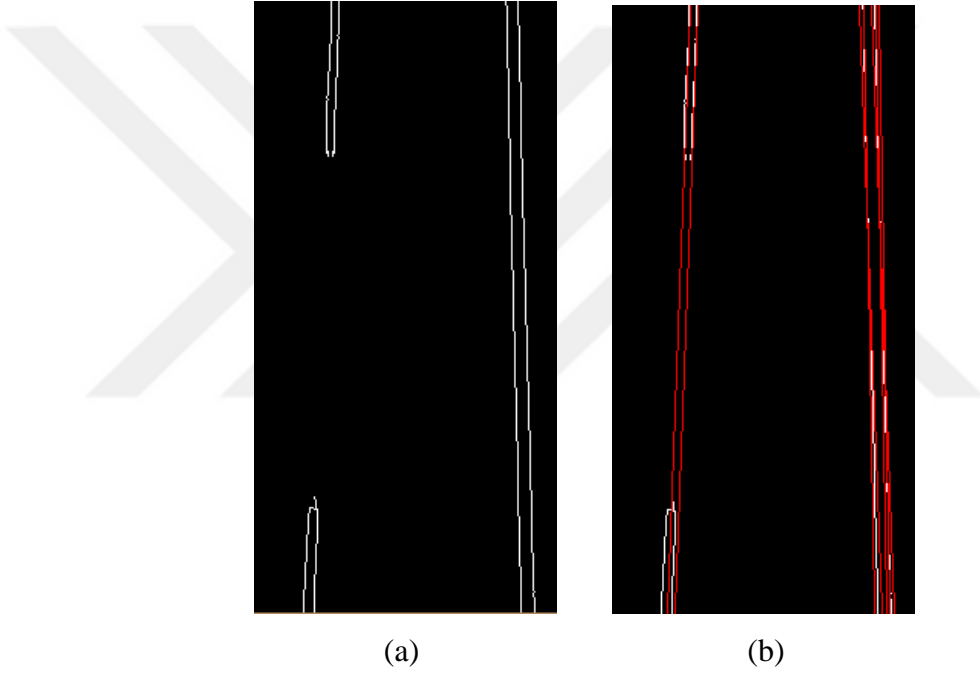
Şekil 2.30. Örnek noktaların iki koordinat sisteminde gösterimi, a) Kartezyen koordinat sisteminde, b) Kutupsal koordinat sisteminde [50].

Hough dönüşümü sayesinde giriş görüntüsündeki kesikli, yarıda kalmış ve hatta bölgesel olarak hasarlanmış çizgiler bile tespit edilebilmektedir. Çünkü hasarlanmamış kısımlar, çizginin varlığına işaret olarak kesişim noktası oluşturacaktır. Kutupsal koordinat sisteminde sinüsoidal işaretlerin kesiştiği açı kullanılarak Kartezyen koordinat sistemine geri dönülür. Böylece çizgi tespit edilmiş olur (Şekil 2.31).



Şekil 2.31. Tespit edilen çizgi [50].

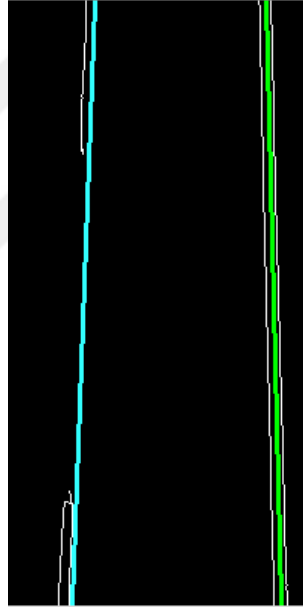
Bölüm 2.6.4'te Canny kenar belirleme sonucunda elde edilen görüntüye Hough dönüşümü uyguladığında elde edilen sonuç görüntüsü Şekil 2.32'de gösterilmiştir.



Şekil 2.32. Hough dönüşümü, a) giriş görüntüsü, b) sonuç görüntü.

Şekil 2.32b'de görüldüğü gibi Hough dönüşümü her bir şerit çizgisi için birden fazla sonuç üretebilir. Uygulamamızın sonraki bölümlerinde kullanmak üzere Hough dönüşümü sonucunda elde edilen çoklu çizgileri teke düşürüp her bir şerit çizgisini tek çizgi ile ifade etmemiz gerekir. Bunu gerçekleştirmek için aşağıdaki adımlar sırasıyla uygulanır.

1. Hough dönüşümü sonucunda elde edilen her bir çizginin sağ ve sol şerit çizgisinden hangisine ait olduğu tespit edilir ve bundan sonraki adımlar her bir şerit çizgisi için ayrı ayrı uygulanır.
2. İlgili şerit çizgisine ait çizgilerin ρ (doğrunun orijinden olan uzaklığı) değerlerinin aritmetik ortalaması alınır ve sonuç yeni oluşacak tek çizginin ρ değeridir.
3. İlgili şerit çizgisine ait çizgilerin θ (orijinden geçen doğrunun x düzlemiyle yaptığı açı) değerlerinin aritmetik ortalaması alınır ve sonuç yeni oluşacak tek çizginin θ değeridir.
4. Her bir şerit için bulunan p ve θ değerleri Kartezyen koordinat sistemine dönüştürülür. Böylece her şerit için çizgiler ayrı ayrı elde edilir (Şekil 2.33).

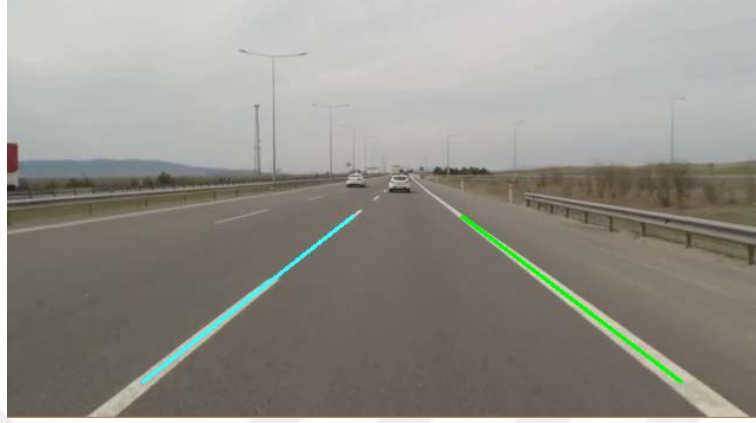


Şekil 2.33. Tespit edilen sol ve sağ şerit çizgileri.

2.8. BELİRLENEN ŞERİT ÇİZGİLERİNİ GİRİŞ GÖRÜNTÜSÜNE TAŞIMA

Bölüm 2.7’de elde ettiğimiz şerit çizgilerini giriş görüntüsü üzerine taşımak için oran orantı yöntemi kullanılmıştır. Bölüm 2.2’de TPD yaparken belirlenen sınırları baz alarak, TPD görüntüsündeki sol ve sağ şerit çizgilerin x ve y koordinatlarının giriş görüntüsü üzerindeki karşılıkları bulunmuştur. Böylece TPD görüntüsü üzerinde

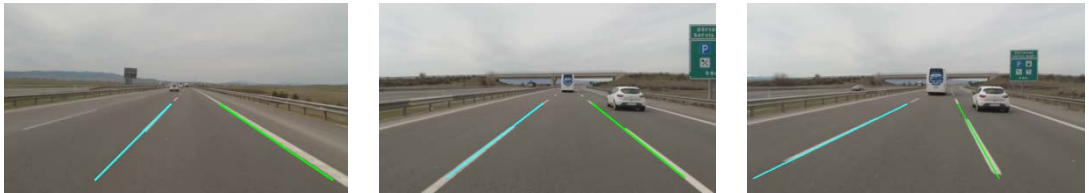
Hough dönüşümü kullanılarak bulunan Şekil 2.33'deki şerit çizgileri, giriş görüntüsüne taşınmış ve bu taşınan şerit çizgileri Şekil 2.34'de gösterilmiştir.



Şekil 2.34. Belirlenen şerit çizgileri.

2.9. ŞERİTTEN AYRILMA UYARI SİSTEMİ

Araç yolda normal bir seyir halindeyken, iki şerit çizgisi Şekil 2.35b'deki gibi görünür. Sağ ve sol şerit çizgileri dikey bir çizgiye göre birbirine paraleldir ve yatay eksenle yaptıkları açılar birbirine eşittir. Eğer araç şeritten ayrılıyor ise, şerit çizgilerinden bir tanesi diğerine göre daha dikey bir konum alır. Araç sol şerit çizgisinden ayrılıyor ise sol şerit çizgisi sağ şerit çizgisine göre daha dikey (Şekil 2.35a), eğer araç sağ şerit çizgisinden ayrılıyor ise sağ şerit çizgisi sol şerit çizgisine göre daha dikey olur (Şekil 2.35c). Bu özelliği kullanılarak, aracın sağ şerit çizgisinden mi yoksa sol şerit çizgisinden mi ayrıldığı kolayca tespit edilebilir.



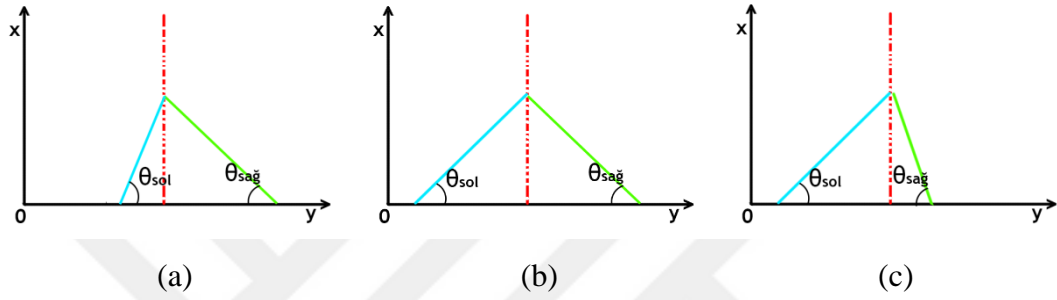
(a)

(b)

(c)

Şekil 2.35. Şeritten ayrılma durumları, a) sol şerit çizgisinden ayrılma durumu, b) ideal durum, c) sağ şerit çizgisinden ayrılma durumu.

Sağ ve sol şerit çizgilerinin birbirine göre dikey oluşları yatay eksen ile yaptıkları açılara bakılarak karar verilebilir. Hangi şerit çizgisinin yatay ile yaptığı açı daha büyükse, o diğerine göre daha diktir. Şekil 2.36b’de görüldüğü gibi açılarının birbirine eşit olduğu durum ($\theta_{sağ} = \theta_{sol}$) ideal durumdur ve herhangi bir şeritten ayrılma sözkonusu değildir. Eğer açılardan herhangi biri diğerine göre daha büyükse, büyük açının olduğu şerit çizgisinden ayrılma başlamış demektir (Şekil 2.36a ve Şekil 2.36c).

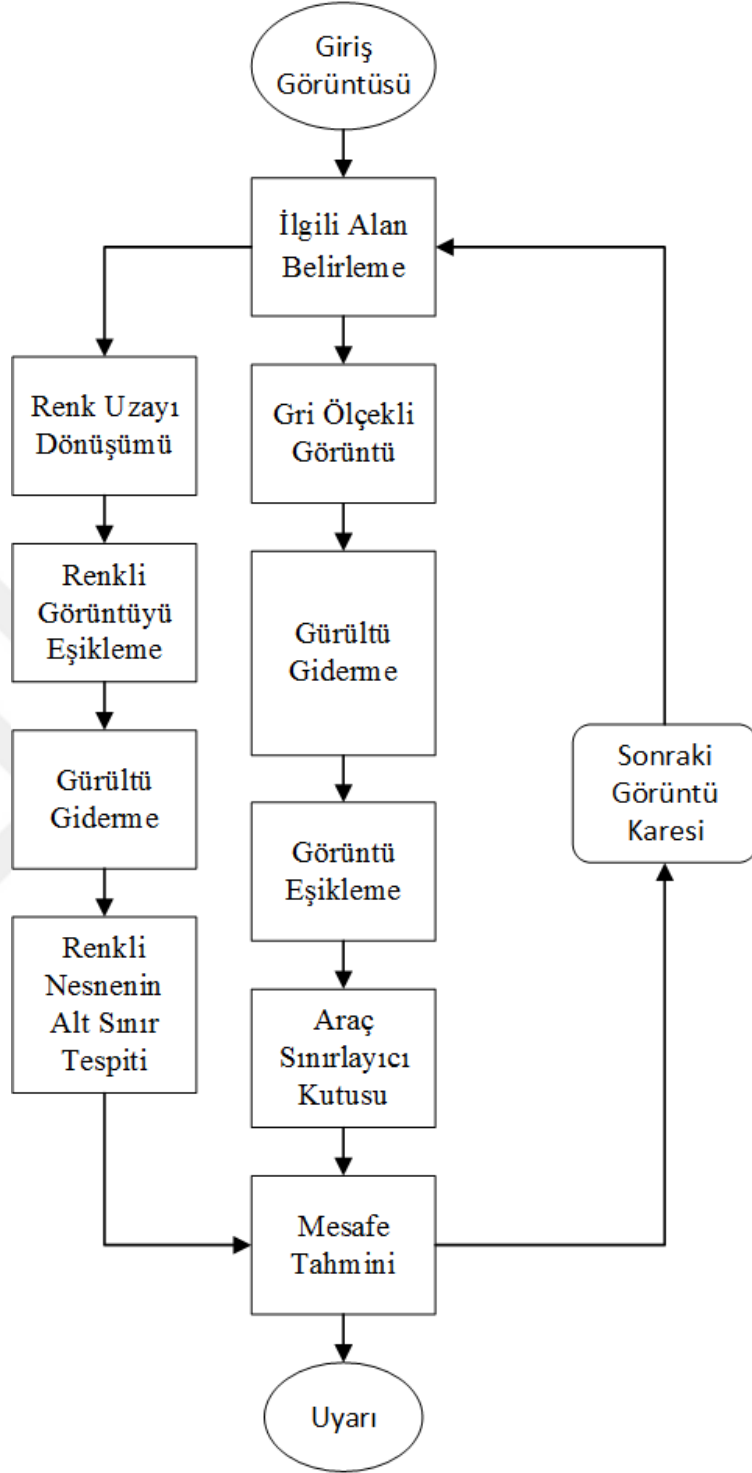


Şekil 2.36. Grafiksel olarak şeritten ayrılma durumları, a) sol şerit çizgisinden ayrılma durumu, b) ideal durum, c) sağ şerit çizgisinden ayrılma durumu.

BÖLÜM 3

ÖNDEN ÇARPMA UYARI SİSTEMİ

Bu bölümde önden çarpmayı engellemek için öndeki aracın ve yol üzerindeki renkli herhangi bir nesnenin mesafesi belirlenip sürücüye sesli uyarı veren bir sistem geliştirilmiştir. Bu sistem sayesinde sürücünün dikkatsizliğinden dolayı öndeki araca fazla yaklaşması ve herhangi bir renkli nesnenin yandan yola girmesi durumunda sürücüye sesli olarak uyarı verilerek olası bir kaza önlenmiş olur. Bunu yapabilmek için öncelikle öndeki aracın veya renkli nesnenin tespit edilmesi gerekir daha sonra tespit edilen bu nesnenin mesafesinin belirlenmesi gerekir. Önden çarpma uyarı sisteminin akış şeması Şekil 3.1’de gösterilmiştir.



Şekil 3.1. Önden çarpma uyarı sisteminin akış şeması.

3.1. İLGİLİ ALAN BELİRLEME

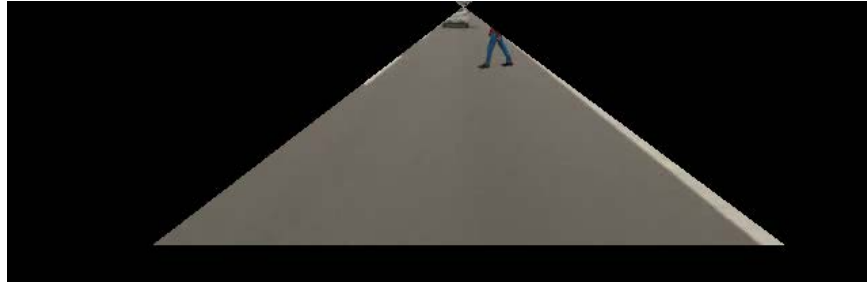
BÖLÜM 3'ün amacı öndeki aracın veya renkli nesnenin mesafesini tespit edip olası kazaları önlemek için sürücüye uyarı vermektir. Dolayısıyla, mesafesi bulunacak ilgili

nesnenin bizim aracımız ile aynı şeritte olması gerekir. İlk olarak, öndeki ilgili nesnenin olma ihtimalinin yüksek olduğu ilgili alan belirlenir.

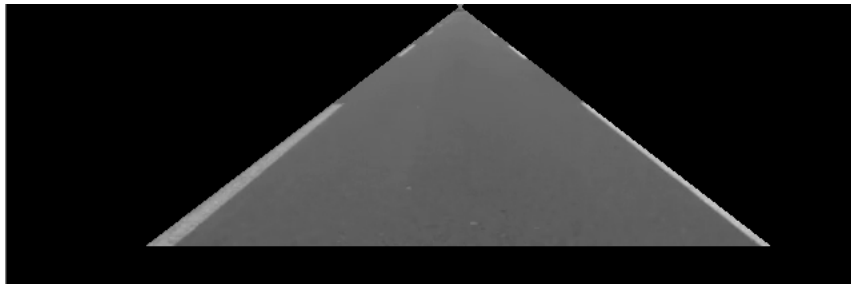
Bölüm 2.1’de şerit tespiti yaparken şerit çizgilerinin olma ihtimalinin yüksek olduğu kısım ilgili alan olarak seçilmişti ve bu işlem yükünü büyük ölçüde azaltmıştı. Araç veya renkli nesne tespit işlemine geçmeden önce Bölüm 2.8’de belirlenen şerit çizgilerinin dışında kalan kısım yok edilerek ilgili nesne bizim aracımız ile aynı şeritte belirlenmeye çalışılır. Böylece hem işlem yükü daha da azalır hem de sürücüye mesafesi hakkında bilgi verilecek doğru araç (kendi şeridimizdeki araç) veya renkli nesne tespit edilmiş olur. Önde aracın veya renkli nesnenin olduğu durumlar ile hiçbir nesnenin olmadığı örnek ilgili alanlar Şekil 3.2’de gösterilmiştir.



(a)



(b)

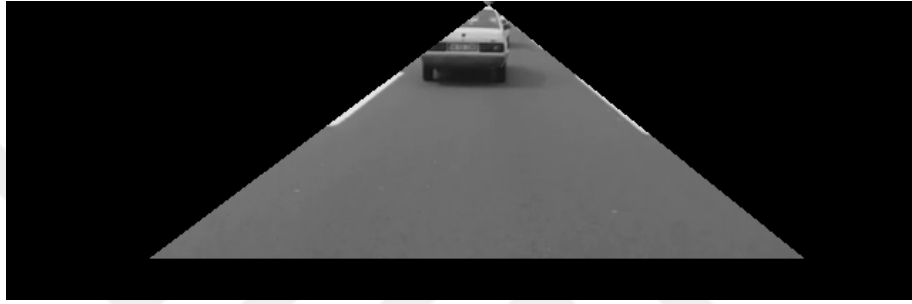


(c)

Şekil 3.2. Örnek ilgili alanlar, (a) önde aracın olduğu, (b) önde renkli nesnenin olduğu, (c) öndeki aracın olmadığı.

3.2. GRİ ÖLÇEKLİ GÖRÜNTÜ

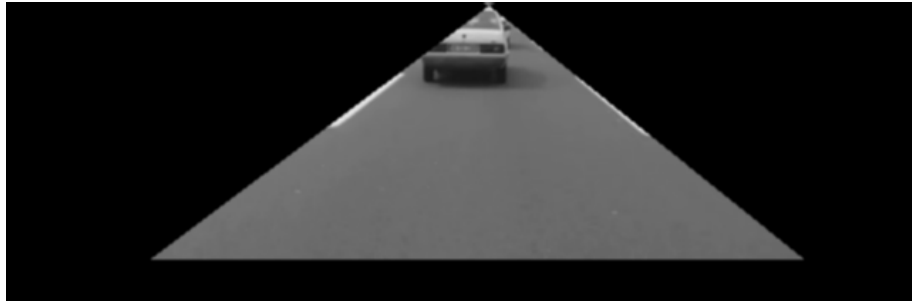
Bölüm 3.1’de ilgili alanı belirledikten sonra araç tespit etmeye geçmeden önce ilgili alan görüntüsü gri ölçekli görüntüye dönüştürülür. Böylece daha sonraki işlemler bu gri ölçekli görüntü üzerinden yapılacağından işlem yükü azalacaktır. Renkli görüntüyü gri ölçekli görüntüye dönüştürme işlemi Bölüm 2.3’te detaylı bir şekilde anlatılmıştır. Şekil 3.2a’daki renkli ilgili alan görüntüsünün gri ölçekli görüntüye dönüştürülmüş hali Şekil 3.3’te gösterilmiştir.



Şekil 3.3. Gri ölçekli görüntü.

3.3. GÜRÜLTÜ GİDERME

Gürültü azaltma yöntemleri Bölüm 2.4’te detaylı bir şekilde anlatılmıştır. Uygulamamıza Gauss gürültü azaltma yöntemi kullanılmıştır ve gürültü azaltma işlemi sonucunda elde edilen görüntü Şekil 3.4’de gösterilmiştir.



Şekil 3.4. Gürültüsü azaltılmış görüntü.

3.4. GÖRÜNTÜ EŞİKLEME

Görüntü bölütleme, bölme stratejisine (bir görüntüyü birçok parçaya ayırma) dayanan görüntü işleme tekniklerini kapsayan genel bir terimdir. Ayırma işleminin sonucunda oluşan her bir parça spesifik bir amaç (nesne tanıma, nesnenin boyutunu belirleme, vs.) için kullanılır. Görüntü bölütleme için; eşikleme, renk tabanlı yöntemler, yapı filtreleri, kümeleme gibi birçok yöntem kullanılır [51].






Görüntü bölütleme için kullanılan en yaygın yöntemlerden biri eşikleme yöntemidir. Eşikleme, gri ölçekli görüntüyü alır ve her bir pikseli, pikselin yoğunluk değeri sabit bir eşik değerinin altında ise siyah yapar üstünde ise beyaz yapar. Bunun sonucunda yeni oluşan siyah-beyaz görüntüde karanlık bölge ışıklı bölgeden ayrılmış olur. Bu bölümde aracı tespit etmek için aracın altında oluşan gölge kullanılacaktır. Önce aracın gölgesi tespit edilecek daha sonra bu tespit edilen gölgeden yararlanılarak aracın sınırlayıcı kutusu bulunacaktır. Aracın gölgesini belirlemek için görüntü eşikleme yöntemi kullanılacaktır.

3.4.1. Global Eşikleme Yöntemi

Eşikleme, gri ölçekli bir görüntünün siyah-beyaz görüntüye dönüştürülme işlemidir. Siyah-beyaz görüntüler, görüntü üzerindeki renklerin pek önemli olmadığı, görüntü üzerinde belirli şekillerin arandığı uygulamalarda işlem yükünü hafifletmek ve görüntü üzerinde mantıksal (0-1) işlemleri hızlı bir şekilde yapabilmek için çoğunlukla kullanılan görüntülerdir. Basitçe gri ölçekli bir görüntü üzerinde 0 ile 255 arasında seçilen bir T eşik değerine göre, siyah-beyaz görüntü Denklem 3.1'deki şekilde oluşturulur.

$$g(x, y) = \begin{cases} 1, & f(x, y) > T \\ 0, & f(x, y) \leq T \end{cases} \quad (3.1)$$

burada T eşik değeri, f(x,y) gri ölçekli görüntünün (x,y) pikselinin yoğunluk değeri ve g(x,y) ise bu piksele karşılık gelen siyah-beyaz görüntünün yeni değeridir. Çeşitli eşik değerleri için sonuçlar Şekil 3.5'te gösterilmiştir.

Giriş Görüntüsü	
	
Eşik Değeri	Çıktı Görüntüsü
T = 51	
T = 102	
T = 153	
T = 204	

Şekil 3.5. Farklı eşik değerleri için çıktı görüntüleri.

3.4.2. Otsu Eşikleme Yöntemi

Global eşikleme yönteminde T değerinin doğru seçilmesi son derece önemlidir. Eğer T değeri çok küçük seçilirse oluşacak yeni görüntüde pek çok piksel beyaz, büyük seçilirse de siyah olacağından görüntünün içerdiği bilgi büyük oranda azalacaktır. Birkaç görüntü için ideal eşik değeri deneme yoluyla bulunabilse de farklı ışık ortamlarında alınmış çok sayıda görüntü için bu söz konusu değildir. Bu sebeple giriş

görüntüsüne karşılık eşik değerini otomatik olarak hesaplayan bir algoritma gerekmektedir. Nobuyuki Otsu (1979) tarafından geliştirilen Otsu metodu ile eşik değeri görüntü histogramı üzerinden hesaplanmaktadır [52]. Bu algoritma, sınıf içi değişkenliğin çok küçük olduğu bir eşik değeri kullanarak görüntü histogramını iki sınıfa ayırır. Bu yolla, her bir sınıf olabildiğince yoğun olacaktır. Histogram hesaplanırken pikseller arasındaki uzaysal ilişki hesaba katılmaz ve böylece benzer piksel değerlerine sahip alanlar birleştirilir [51].

Bu algoritma piksellerin iki sınıfa ayrıldığını farz ederek, Denklem 3.2’de tanımlanan sınıf içi ağırlıklı varyansı minimize etmeye çalışır.

$$\sigma_w^2(t) = q_1(t)\sigma_1^2(t) + q_2(t)\sigma_2^2(t) \quad (3.2)$$

Burada t değişkeni 0 ile 255 arasında bir değere sahip olan eşik değeridir. Denklem 3.2’de varyans hesaplanırken önce her piksel değeri için bir fonksiyon olasılığı olan P elde edilir. Sonra histogram dağılımı hesaplanır ve olasılık dağılımını takip ettiğini garanti etmek için normalizasyon işlemi gerçekleştirilir. Daha sonra, piksel değerleri Denklem 3.3 ve Denklem 3.4’te tanımlanan sınıf olasılık fonksiyonları $q_1(t)$ ve $q_2(t)$ kullanılarak C_1 ve C_2 olmak üzere iki sınıfa ayrılır.

$$q_1(t) = \sum_{i=1}^t P(i) \quad (3.3)$$

$$q_2(t) = \sum_{i=t+1}^I P(i) \quad (3.4)$$

Bu piksellerden yoğunluk seviyesi 1 ve t arasında olanlar sınıf C_1 , yoğunluk seviyesi t+1 ile I (en büyük piksel değeri, genellikle 255) arasında olanlar ise sınıf C_2 ile temsil edilir. Sonra, sınıf C_1 ’in ortalaması $\mu_1(t)$ ile sınıf C_2 ’nin ortalaması $\mu_2(t)$ Denklem 3.5 ve Denklem 3.6 ile elde edilir.

$$\mu_1(t) = \sum_{i=1}^t \frac{iP(i)}{q_1(t)} \quad (3.5)$$

$$\mu_2(t) = \sum_{i=t+1}^I \frac{iP(i)}{q_2(t)} \quad (3.6)$$

Daha sonra, sınıf C_1 ve sınıf C_2 'in varyansları sırasıyla Denklem 3.7 ve Denklem 3.8 ile hesaplanır.

$$\sigma_1^2(t) = \sum_{i=1}^t [i - \mu_1(t)]^2 \frac{P(i)}{q_1(t)} \quad (3.7)$$

$$\sigma_2^2(t) = \sum_{i=t+1}^I [i - \mu_2(t)]^2 \frac{P(i)}{q_2(t)} \quad (3.8)$$

Denklem 3.7 ve Denklem 3.8 sırasıyla C_1 ve C_2 'nin ağırlıklı sınıf içi varyanslarını tanımlar. Bu değerler Otsu algoritmasının minimize etmeye çalıştığı değerlerdir. Bu varyanslar, her bir sınıfın ne kadar yoğun olduğunun ölçüsüdür. Yani, yöntem kötü bir eşik seçerse, sınıflardan birinin varyansı büyük olacaktır. Toplam varyans σ^2 , Denklem 3.2, Denklem 3.7 ve Denklem 3.8'i kullanarak Denklem 3.9'daki gibi tanımlanabilir.

$$\sigma^2 = \sigma_w^2(t) + \sigma_b^2(t) \quad (3.9)$$

$$\sigma_b^2(t) = q_1(t)q_2(t)[\mu_1(t) - \mu_2(t)]^2 \quad (3.10)$$

Burada $\sigma_w^2(t)$ sınıf içi varyansı ve $\sigma_b^2(t)$ ise sınıflar arası varyansı ifade eder. Algoritmanın amacı $\sigma_w^2(t)$ varyansını minimize etmek ve $\sigma_b^2(t)$ varyansını maksimize etmektir. Böylece Denklem 3.11'de belirtildiği gibi eşik değeri t^* , sınıflar arası varyans $\sigma_b^2(t)$ 'i maksimize eden t değeridir [53].

$$t^* = \operatorname{argmax} \sigma_b^2 \quad (3.11)$$

Bizim uygulamamızda değişen hava koşullarından dolayı parlaklık farklı zamanlarda kameradan alınan her bir görüntüde farklı olacaktır. Dolayısıyla uygulamamızda, her

bir görüntüde eşik değeri de farklı olacağı için otomatik eşik değeri belirleyen Otsu metodu kullanılmıştır. Şekil 3.4'teki görüntüye Otsu eşikleme yöntemi uygulanarak tespit edilen araç gölgesi Şekil 3.6'da gösterilmiştir.



Şekil 3.6. Tespit edilen araç gölgesi.

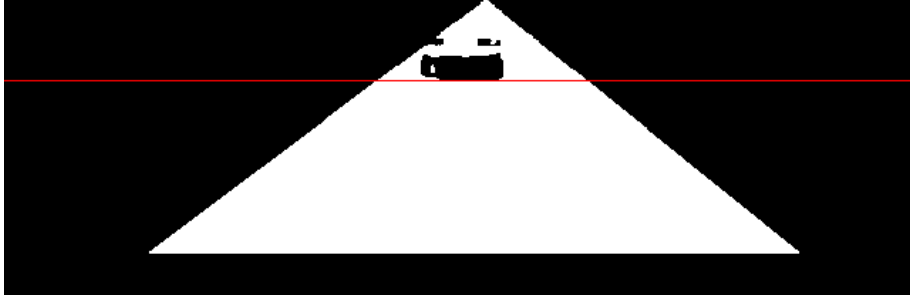
3.5. ARAÇ SINIRLAYICI KUTUSU BULMA

Bölüm 3.4.2'de görüntü, Otsu eşikleme yöntemiyle siyah-beyaz görüntüye dönüştürülmüş ve aracın gölgesi tespit edilmişti. Bu bölümün amacı ise tespit edilen gölgeyi kullanarak aracın sınırlayıcı kutusunu bulmaktır. Bu işlemi gerçekleştirmek için önce belirlenen gölgenin en alt, en sağ ve en sol sınırlarının bulunması gerekir. Sağ ve sol sınırları bulmak aracın genişliğini bulmak için yeterli olacaktır. Bu bilgilere ek olarak bir de aracın yüksekliğinin bulunması gerekmektedir. Aracın yüksekliği de bulununca aracın sınırlayıcı kutusu bulunmuş olur.

3.5.1. Aracın Sol, Sağ Ve Alt Sınırlarını Tespit Etme

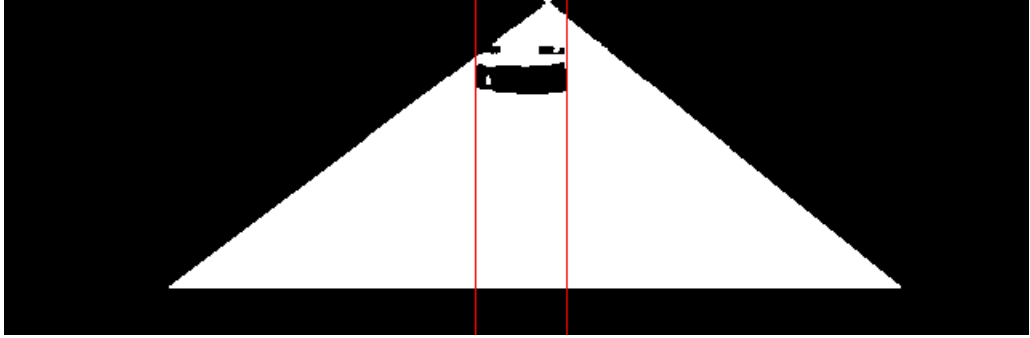
Bölüm 3.4.2'de tespit edilen gölgenin alt sınırını bulmak için ilgili alanın en alt satırından başlanarak üste doğru satır satır tarama yapılabilir ve karşılaşılan ilk siyah piksel gölgenin alt sınırını verir. Bu tarama işlemi yapılırken görüntünün piksel değerlerine ulaşmak için klasik yöntem olan görüntü matrisinin bir değerine ulaşırken kullanılan satır ve sütun değerleri kullanılır. Bu yöntem oldukça yavaş çalışmaktadır. Bunun yerine işaretçi kullanılarak tarama işlemi yapılmıştır. İşaretçi ile görüntünün herhangi bir satırının başlangıç adresine erişilir ve daha sonra ilgili piksele ulaşmak için adres değeri artırılır. Bu yöntem klasik matris tarama

yöntemine göre oldukça hızlıdır [54]. Şekil 3.7’de tespit edilen alt sınır gösterilmiştir.



Şekil 3.7. Tespit edilen alt sınır.

Sağ ve sol sınırları belirlemek için yine ilgili alan işaretçi kullanılarak taranır. Öncelikle sol sınırı bulmak için görüntünün sol tarafından başlanarak ilgili alan üzerinde tarama yapılır. Karşılaşılan ilk siyah piksel sol sınır olarak bulunur. Sonra sağ sınırı bulmak için ise sağdan başlanarak tarama yapılır. Karşılaşılan ilk siyah piksel sağ sınır olarak bulunur. Tespit edilen sol ve sağ sınırlar Şekil 3.8’de gösterilmiştir.



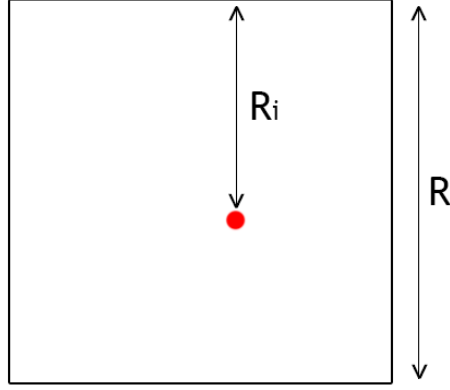
Şekil 3.8. Tespit edilen sağ ve sol sınırlar.

3.5.2. Aracın Üst Sınırını Tespit Etme

Bölüm 3.5.1’de aracın sağ, sol ve alt sınırını tespit edilmiştir. Aracın sınırlayıcı kutusunu bulabilmek için üst sınırının da tespit edilmesi gerekir. Üst sınırı tespit etmek için ise önce aracın yüksekliği hesaplanır ve bulunan yüksekliğin alt sınırdan çıkarılması ile üst sınır değeri elde edilir.

Piksel Derinliđi

Görüntüdeki bir nesnenin piksel derinliđi, görüntünün alt kenarından nesnenin bulunduđu koordinata olan toplam piksel sayısıdır. Şekil 3.9’da gösterilen piksel derinliđi Denklem 3.12 ile hesaplanabilir [55].

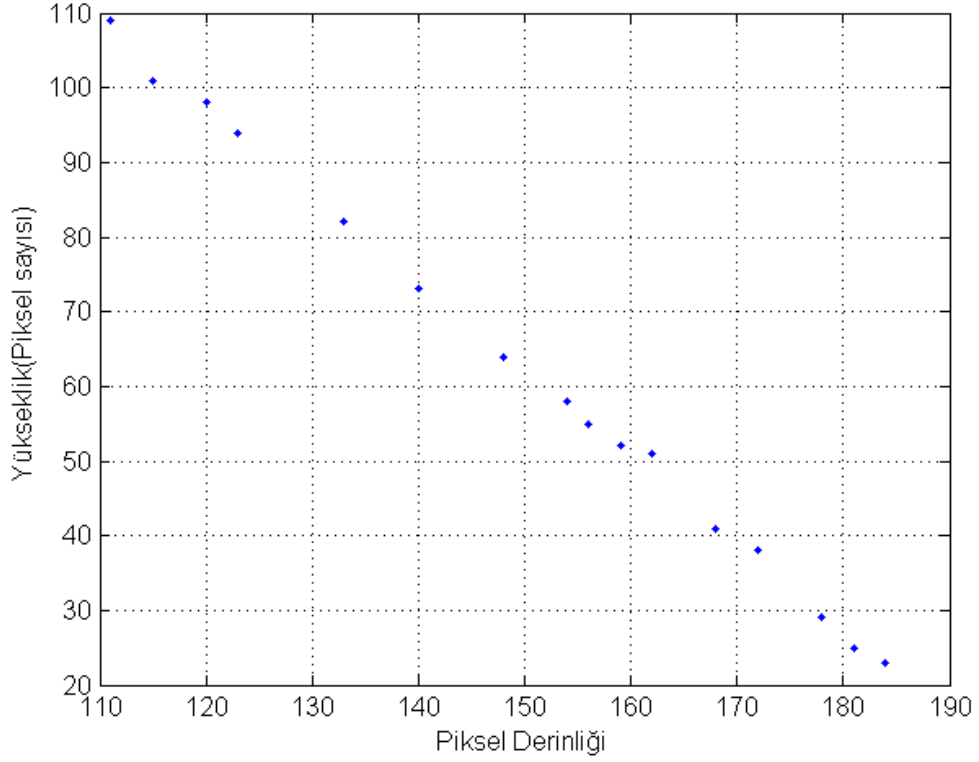


Şekil 3.9. Piksel derinliđi.

$$\text{Piksel Derinliđi} = R - R_i \quad (3.12)$$

Burada R görüntünün boyutu ve R_i ise görüntünün üst kenarından nesnenin bulunduđu koordinata olan piksel sayısıdır.

Araç yakında iken daha büyük bir yüksekliğe uzakta iken daha küçük bir yüksekliğe sahiptir. Dolayısıyla araç uzaklaştıkça aracın yüksekliđi de uzaklığa bađlı olarak belli bir oranda azalacaktır. Bu özellik kullanılarak aracın yüksekliğini hesaplamak için öncelikle aracın bulunduđu belli mesafelerde (piksel derinliklerinde) aracın yüksekliđi piksel sayısı cinsinden hesaplanmıştır. Sonuçlar Şekil 3.10’da gösterilmiştir.



Şekil 3.10. Ölçüm sonucunda elde edilen piksel derinlikleri ve bu derinliklere karşılık gelen yükseklik değerleri.

Şekil 3.10'daki değerler (piksel sayısı cinsinden yükseklik değerleri ve bu değerlere karşılık gelen piksel derinlikleri) kullanılarak, eğri uydurma yöntemi ile bir fonksiyon elde edilir. Daha sonra elde edilen fonksiyona aracın bulunduğu piksel derinliği gönderilerek aracın tahmini yüksekliği bulunur.

Eğri Uydurma

Eğri uydurma noktasal veri dizilerini en iyi ifade eden doğruyu ya da eğriyi bulmak için kullanılır. Eğri uydurma çoğunlukla eğri boyunca noktaları bulabilen bir denklem üretir. Bazı durumlarda ise eğri boyunca noktaları bulabilen bir denklem üretemez ve onun yerine verileri yumuşatarak yaklaşık bir eğri bulur [56].

Eğri uydurma; en küçük kareler, doğrusal olmayan ve yumuşatma olmak üzere üç temel kategoriye ayrılır. En yaygın kullanılanı ise en küçük kareler eğri uydurma yöntemidir.

Polinom fonksiyonunu kullanan eğri uydurma yöntemi Denklem 3.13 formundadır. Bu denklemde j polinom derecesidir.

$$f(x) = a_0 + a_1 * x + a_2 * x^2 + \dots + a_j * x^j \quad (3.13)$$

Veri noktalarının ($y_i : i=1,2,\dots,n$) doğrudan uzaklıkları hata olarak nitelendirilirse en uygun eğri, bu hataların minimum olduğu eğri olacaktır. Dolayısıyla bu hataları minimum yapan a_0, a_1, \dots, a_j katsayı değerleri bulunarak eğri denklemi elde edilmiş olur.

3. dereceden polinom eğri uydurma sonucunda Denklem 3.14'deki eğri denkleminin katsayıları elde edilmiş ve Çizelge 3.1'de gösterilmiştir. Bu katsayılar Denklem 3.14'de yerine yazılırsa eğri denklemi Denklem 3.15'teki gibi elde edilir. Bu eğri denkleminde aracın tespit edildiği piksel derinlik değeri yerine yazılırsa aracın yüksekliği hesaplanmış olur.

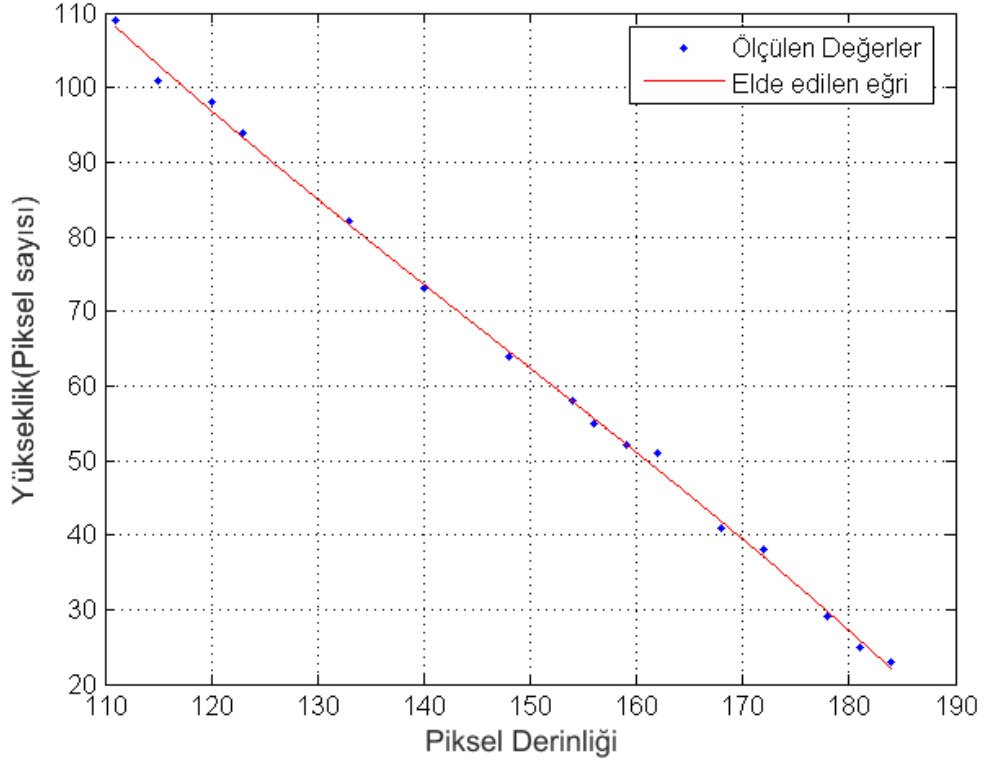
$$f(x) = a_0 + a_1 * x + a_2 * x^2 + a_3 * x^3 \quad (3.14)$$

Çizelge 3.1. Elde edilen katyası değerleri.

a0	a1	a2	a3
369	- 3,942	$1,916 * 10^{-2}$	$- 4,341 * 10^{-5}$

$$f(x) = 369 - 3,942 * x + 1,916 * 10^{-2} * x^2 - 4,341 * 10^{-5} * x^3 \quad (3.15)$$

Şekil 3.10'daki veri değerleri ile Denklem 3.15'teki bulunan eğri Şekil 3.11'de gösterilmiştir.



Şekil 3.11. Elde edilen eğri denklemin grafiği.

3.6. RENKLİ NESNE TESPİT ETME

3.6.1. Renk Uzayı Dönüşümü

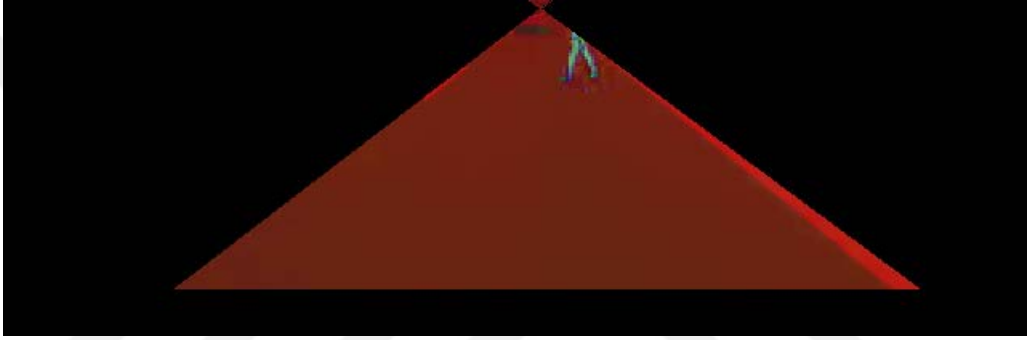
Renk uzayları bir görüntüdeki renkleri tanımlamak için kullanılan matematiksel modellerdir ve bütün renkleri temsil edecek şekilde oluşturulur.

RGB (Red, Green, Blue) renk uzayı kırmızı yeşil mavi ana renklerinden oluşur ve en sık kullanılan renk uzayıdır. RGB renk uzayında, tüm renklerin kodları bu üç temel renge dayalı olarak belirtilir. Dolayısıyla, bu ana renklerin karışımı diğer tüm renkleri belirler.

HSV (Hue, Saturation, Value) renk uzayı, renkleri sırasıyla renk özü, doygunluk ve parlaklık olarak tanımlar. Renk özü, rengin baskın dalga uzunluğunu belirler. Doymunluk, rengin canlılığını belirler. Yüksek doymunluk canlı renklere neden

olurken, düşük doygunluk rengin gri tonlarına yaklaşmasına neden olur. Parlaklık ise rengin aydınlığını yani içindeki beyaz oranını belirler.

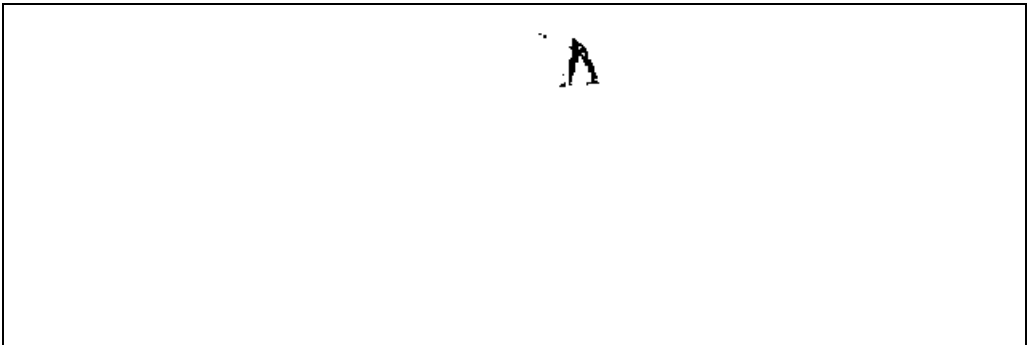
Buna göre herhangi bir görüntü işleme uygulamasında belirli renkteki bir nesne ayırt etmek istenildiğinde HSV renk uzayını kullanmak daha uygundur. Çünkü HSV renk uzayında RGB renk uzayının aksine sadece hue değeri kullanılarak renkler daha net ayırt edilebilir. Bölüm 3.1’de ilgili alanı belirledikten sonra renkli nesneyi tespit etmeye geçmeden önce ilgili alan görüntüsü RGB renk uzayından HSV renk uzayına dönüştürülür.



Şekil 3.12. HSV renk uzayına dönüştürülmüş görüntü.

3.6.2. Renkli Görüntü Eşikleme

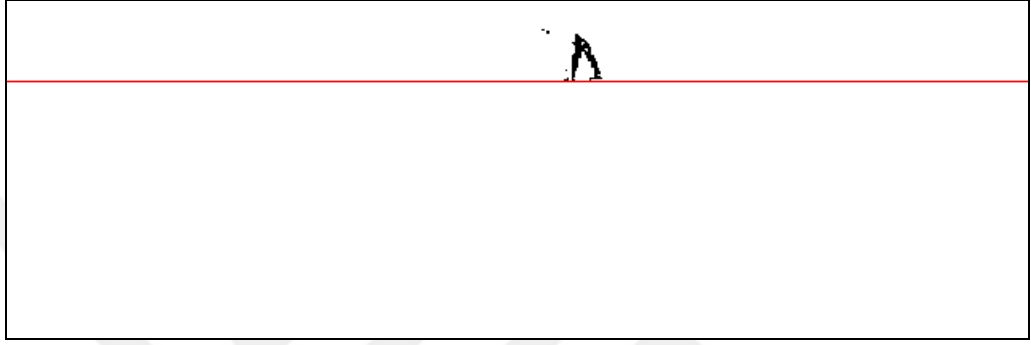
Bu bölümde Şekil 3.12’deki HSV renk uzayına dönüştürülmüş görüntüye gri ve soluk tondaki (yol yüzey rengine yakın renk) rengi göz ardı ederek sadece renkli kısımları filtreleyen uygun renk özü, doygunluk, parlaklık (hue, saturation ve value) eşik değerleri ile Şekil 3.13’deki görüntü elde edilir.



Şekil 3.13. Eşiklenmiş renkli görüntü.

3.6.3. Nesnenin Alt Sınırını Tespit Etme

Şekil 3.13'te elde edilen görüntüdeki nesnenin alt sınırını bulmak için ilgili alanın en alt satırından başlanarak üste doğru satır satır tarama yapılır ve karşılaşılan ilk siyah piksel nesnenin alt sınırını verir. Bu tarama işlemi işaretçi kullanılarak yapılmıştır. Şekil 3.14'te tespit edilen alt sınır gösterilmiştir.



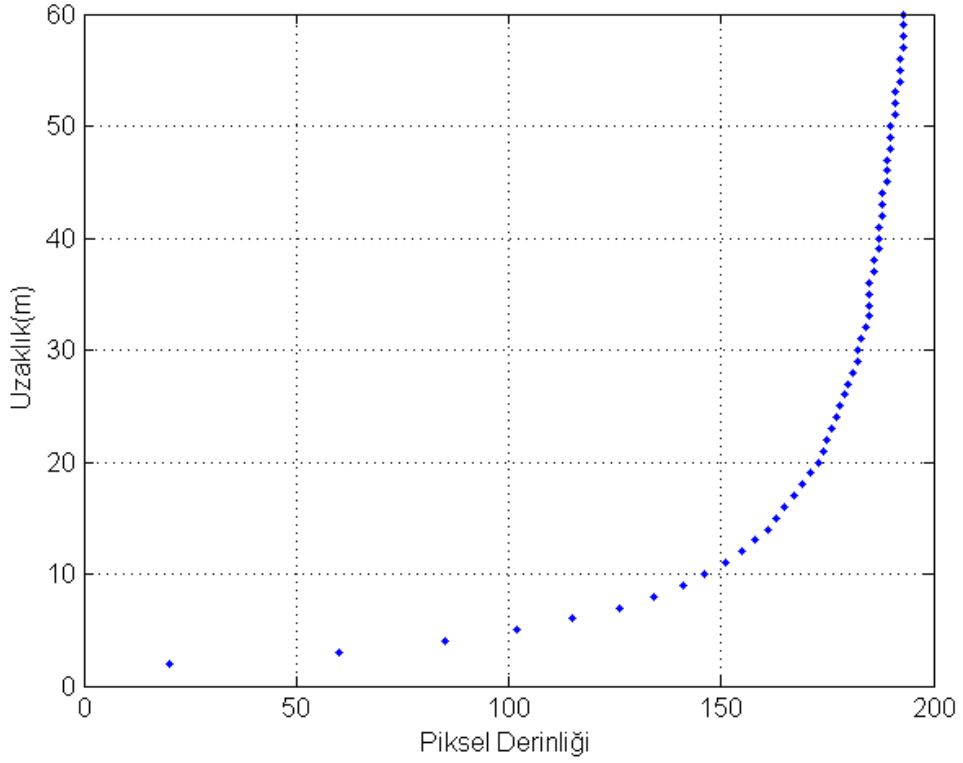
Şekil 3.14. Alt sınırı tespit edilmiş görüntü.

3.7. MESAFE TAHMİNİ

Mesafe tahmini için çift görü tabanlı ve tek görü tabanlı olmak üzere temelde iki yaklaşım kullanılır. Çift görü tabanlı yaklaşımda iki kamera kullanılır. Bu durum uygulamanın maliyetini önemli ölçüde artıracaktır ve iki kameranın kalibrasyonunun yapılması da büyük bir iş yükü getirecektir. Bir diğer büyük sorun da kullanılan algoritmanın iki kameradan alınan görüntüye ayrı ayrı uygulanacak olması ciddi bir performans kaybına neden olacaktır. Bu nedenle bu uygulamada tek kameranın kullanıldığı tek görü tabanlı yaklaşım tercih edilmiştir.

Uygulamada, Bölüm 3.5'te araç tespit edilmiş ve aracın bulunduğu koordinattaki piksel derinliği ile Bölüm 3.6.3'te tespit edilen renkli nesnenin bulunduğu piksel derinliği hesaplanmıştır (Piksel derinliği Bölüm 3.5.2'de detaylı olarak anlatılmıştır). Hesaplanan bu piksel derinliği kullanılarak öndeki aracın veya renkli nesnenin mesafesi hem eğri uydurma yöntemi hem de yapay sinir ağları yöntemi ile hesaplanmış ve sonuçlar Çizelge 3.4'te gösterilmiştir.

Eđri uydurma ve yapay sinir ađları ile mesafe hesaplama iřlemine gemeden 2 ile 60 metre arasında bir metre aralıklarla 2 ölçüm alınmıř ve her bir metreye karřılık gelen piksel derinlikleri hesaplanmıřtır (řekil 3.15). Bu deđerler eđri uydurma ve yapay sinir ađları yöntemlerinde kullanılarak mesafe hesaplama iřlemi gerekleřtirilmiřtir.



řekil 3.15. Ölüm sonucunda elde edilen piksel derinlikleri ve bu derinliklere karřılık gelen mesafe deđerleri.

3.7.1. Eđri Uydurma Yöntemi İle Mesafe Tahmini

Öndeki aracın veya renkli nesnenin mesafesini hesaplamak için gerek mesafe ve bu mesafelere denk gelen piksel derinlikleri kullanılarak (řekil 3.15), eđri uydurma yöntemi ile bir fonksiyon elde edilmiřtir. Sonra elde edilen fonksiyona aracın veya renkli nesnenin bulunduđu piksel derinliđi gönderilerek ilgili nesnenin gerek mesafesi bulunmuřtur.

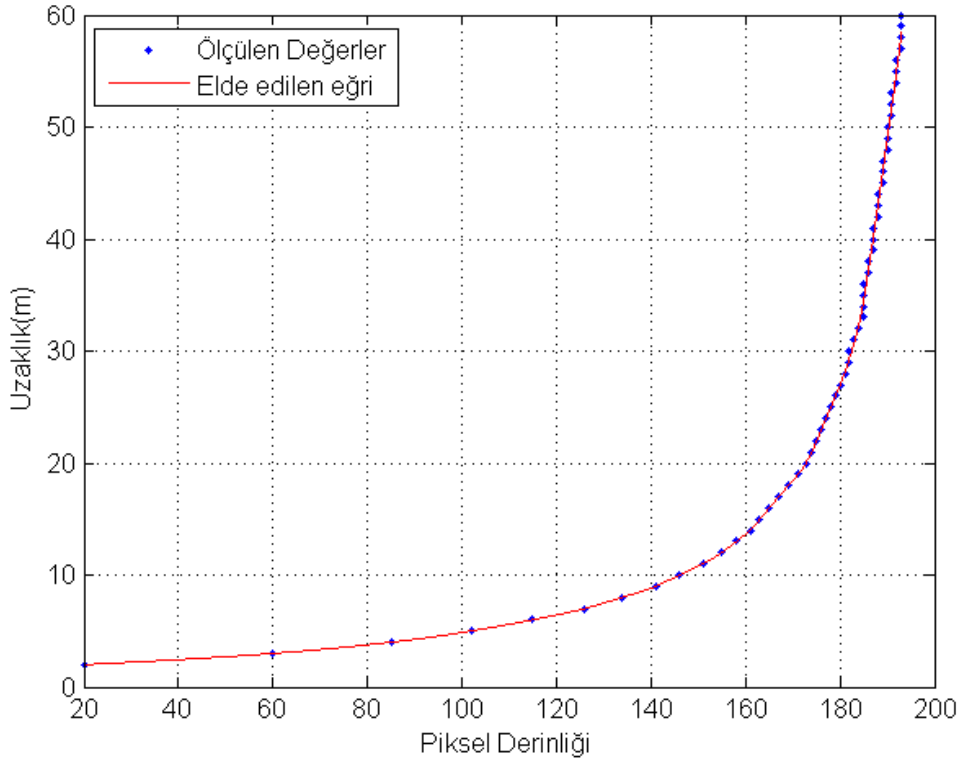
Polinom derecesi 10 olan eğri uydurma sonucunda Denklem 3.16'daki eğri denkleminin katsayıları elde edilmiş ve Çizelge 3.2'de gösterilmiştir.

$$f(x) = a_0 + a_1 * x + a_2 * x^2 + a_3 * x^3 + a_4 * x^4 + a_5 * x^5 + a_6 * x^6 + a_7 * x^7 + a_8 * x^8 + a_9 * x^9 + a_{10} * x^{10} \quad (3.16)$$

Çizelge 3.2. Elde edilen katyası değerleri.

a₀	a₁	a₂	a₃
- 5,9728	1,1063e1	- 6,2825e-2	1,9025e-3
a₄	a₅	a₆	a₇
- 3,2411e-5	2,9933e-7	- 1,0188e-9	- 6,3861e-12
a₈	a₉	a₁₀	
7,9867e-14	- 3,1193e-16	4,4376e-19	

Şekil 3.15'deki veri değerleri ile Denklem 3.16'daki bulunan eğri denklemini Şekil 3.16'da üst üste çizdirilmiştir.



Şekil 3.16. Elde edilen eğri denkleminin grafiği.

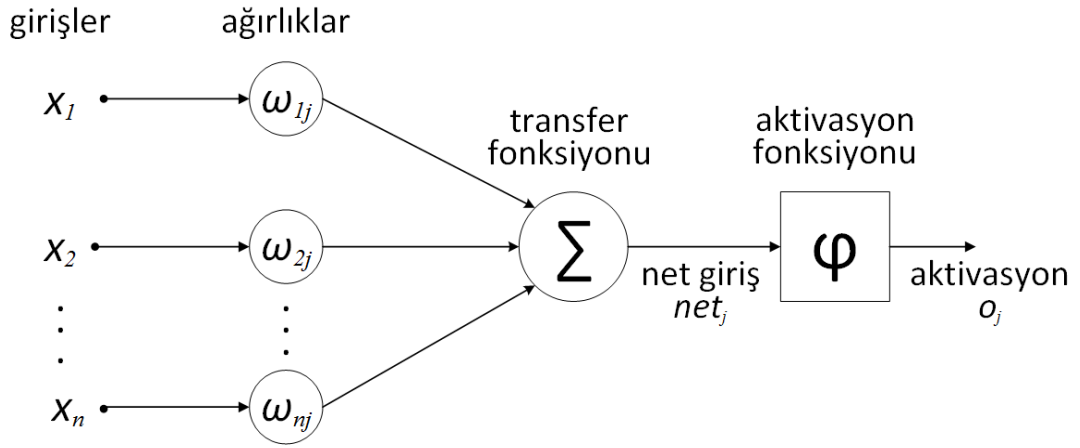
3.7.2. Yapay Sinir Ağları Yöntemi İle Mesafe Tahmini

Yapay sinir ağları, insan beyninin çalışma mekanizmasını taklit ederek beyin öğrenme, hatırlama ve genelleme yapma yoluyla yeni bilgiler türetebilme gibi temel işlevlerini gerçekleştirebilmek amacı ile geliştirilen bilgisayar sistemleridir.

Yapay sinir ağları; insan beyninden esinlenerek, öğrenme sürecinin matematiksel olarak modellenmesi uğraşı sonucu ortaya çıkmıştır ve ilk yapay sinir ağı modeli 1943 yılında bir sinir hekimi olan Warren McCulloch ve bir matematikçi olan Walter Pitts tarafından ortaya atılmıştır. Bu konu üzerindeki çalışmalar ilk olarak beyni oluşturan biyolojik üniteler olan nöronların modellenmesi ve bilgisayar sistemlerinde uygulanması ile başlamış, daha sonraları bilgisayar sistemlerinin gelişimine de paralel olarak birçok alanda kullanılır hale gelmiştir. Başlıca; sınıflandırma, modelleme ve tahmin uygulamaları olmak üzere, pek çok alanda yaygın olarak kullanılmaktadır. Bu amaçla geliştirilmiş ağlar genellikle; doğrusal olmayan sinyal işleme, doğrusal olmayan sistem modelleme, muhtemel fonksiyon kestirimleri, ilişkilendirme veya örüntü eşleştirme, zaman serileri analizleri, sinyal filtreleme, veri sıkıştırma, örüntü tanıma, optimizasyon ve kontrol fonksiyonlarını gerçekleştirmektedirler.

Yapay Sinir Hücresinin Yapısı

Bir yapay sinir hücresi; girdiler, ağırlıklar, toplama (transfer) fonksiyonu, aktivasyon fonksiyonu ve çıktı olmak üzere beş bölümden oluşur [57].

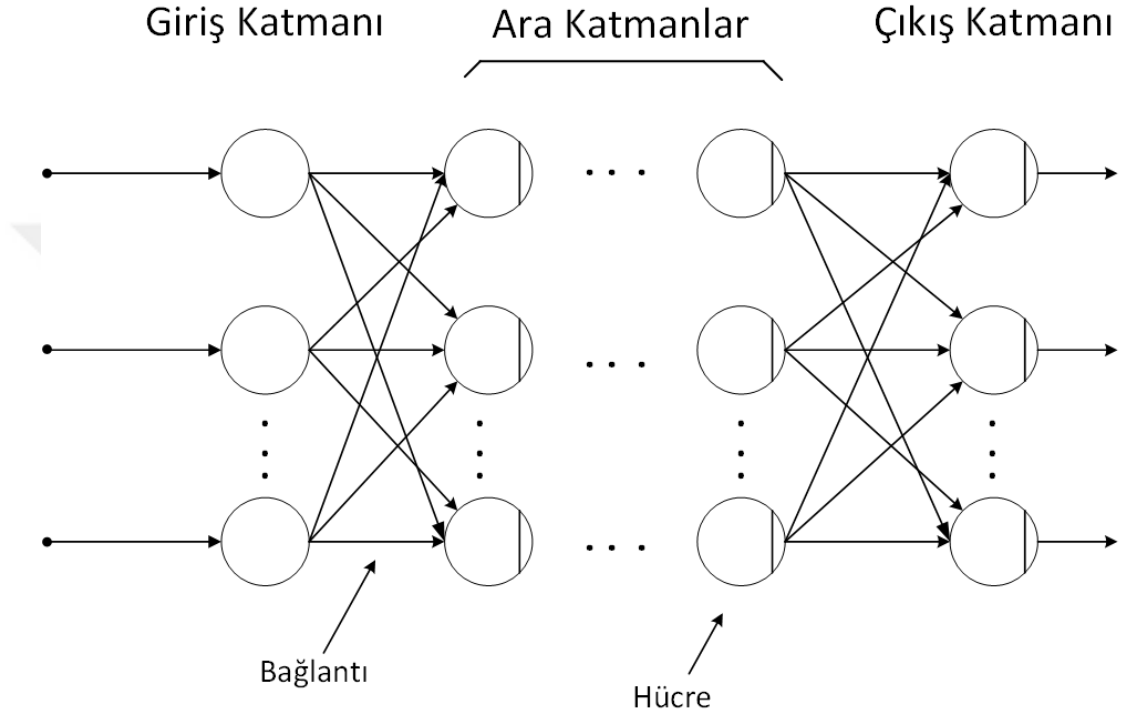


Şekil 3.17. Yapay sinir hücresinin yapısı.

- **Girdiler:** Girdiler yapay sinir hücresine gelen verilerdir. Bu veriler yapay sinir ağı yapısına bağlı olarak dış dünyadan gelebildiği gibi bir diğer yapay sinir hücresinden de gelebilir.
- **Ağırlıklar:** Girdilerin üretilecek çıktı üzerindeki etkisinin ayarlanabilmesi için yapay sinir hücresine gelen girdiler bağlantı ağırlıklarıyla çarpılarak nörona iletilirler.
- **Toplama (transfer) Fonksiyonu:** Toplama fonksiyonu bir yapay sinir hücresine ağırlıklarla çarpılarak gelen girdileri toplayarak o hücrenin net girdisini hesaplayan bir fonksiyondur.
- **Aktivasyon Fonksiyonu:** Aktivasyon fonksiyonu hücreye gelen net girdiyi işleyerek bu girdiye karşılık üretilecek çıktıyı belirleyen fonksiyondur. Ağı doğrusal olmayan yapıyı öğrenmesini sağlayan bileşenidir. Aktivasyon fonksiyonu için genellikle doğrusal olmayan bir fonksiyon seçilir ve günümüzde yaygın olarak sigmoid fonksiyonu tercih edilir.
- **Çıktı:** Aktivasyon sonucu oluşan değer hücrenin çıktı değeridir. Bu değer yapay sinir ağının çıktısı olarak dış dünyaya verilebileceği gibi bir başka hücreye girdi olarak ta verilebilir.

Yapay Sinir Ağının Yapısı

Yapay sinir ağları yapay sinir hücrelerinin birbirine bağlanması ile oluşan yapılardır. Yapay sinir ağları giriş katmanı, ara (gizli) katman ve çıkış katmanı olmak üzere üç ana katmandan oluşur.



Şekil 3.18. Yapay sinir ağının yapısı.

Giriş Katmanı: Yapay sinir ağına dış dünyadan gelen girdilerin geldiği katmandır.

Ara (Gizli) Katman: Giriş katmanı çıktılarının geldiği katmandır. Yapay sinir ağının tasarımı göre ara katman sayısı ve ara katmandaki hücre sayısı değişiklik gösterebilir. Birden fazla ara katman olduğu gibi hiç ara katman olmaya da bilir. Ara katmandaki hücre sayısı artması daha karmaşık problemlerin çözülmesini sağlarken diğer taraftan hesap karmaşıklığı ve süresinin artmasına yol açar. Bu nedenle hücre sayısının optimum olarak seçilmesi gerekir.

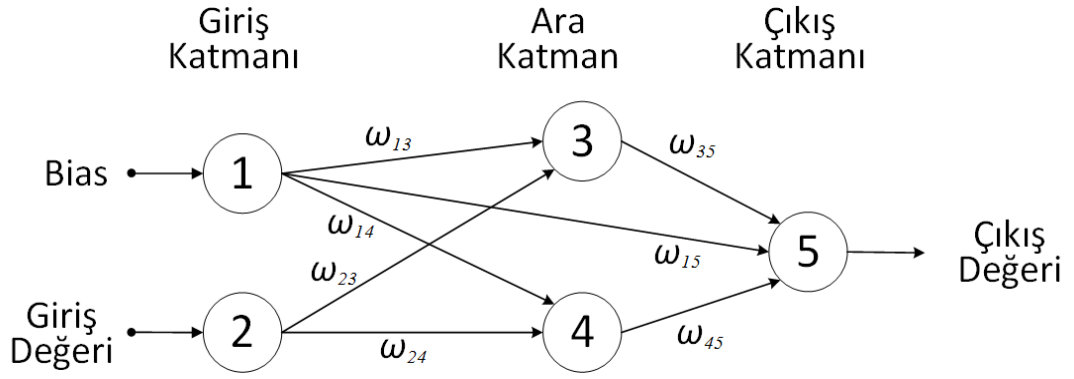
Çıkış Katmanı: Ara katman çıktılarını girdi olarak alan ürettiği çıktıları dış dünyaya aktaran katmandır.

Yapay Sinir Ağlarının Eğitilmesi

Bir yapay sinir ağlarının eğitimine başlamadan önce ağın yapısı belirlenir. Öncelikle ağın giriş sayısı, çıkış sayısı, ara katman sayısı ve ara katmanda bulunacak hücre sayısına karar verilir. Daha sonra toplama fonksiyonu, aktivasyon fonksiyonu, öğrenme katsayısı ve momentum katsayısı gibi parametreler belirlenir. Böylece ağın yapısı oluşturulmuş olur. Son olarak hücreler arasındaki ağırlıklara başlangıç değerleri rastgele olarak atanır.

Yapay sinir ağlarının eğitilme sürecinde dış ortamdan alınan veriler kullanılır. Bu verilerden girdi olanları aktivasyon fonksiyonundan geçirilerek bir tepki çıkışı üretilir. Bu tepki çıkışı ile dış ortamdan alınan verilerden çıktı olan ile karşılaştırılarak hata bulunur. Sonra yapay sinir ağının ağırlıkları yenilenecek hata azaltılıp gerçek çıkışa yaklaşılmaya çalışılır. Bu işlem hata istenilen bir eğik değerinin altına düşene kadar devam eder ve hata istenilen seviyeye düştüğü anda o anki ağırlık değerleri yapay sinir ağının ağırlıkları olarak belirlenmiş olur. Bu süreç öğrenme adı verilir. Daha sonra dış ortamdan alınan ve eğitimde kullanılmayan veriler ile ağ test edilir.

Öndeki aracın veya renkli nesnenin mesafesini hesaplamak için gerçek mesafe ve bu mesafelere denk gelen piksel derinlikleri kullanılarak (Şekil 3.15) Şekil 3.19'daki ileri beslemeli yapay sinir ağına uygulanmıştır. Elde edilen piksel derinlikleri yapay sinir ağına giriş olarak verilmiştir ve bu piksel derinliğine karşılık gelen gerçek mesafe değerleri ile ağ eğitilmiştir.



Şekil 3.19. Uygulamada kullanılan yapay sinir ağının yapısı.

Eğitim sonucunda elde edilen ve ağın yapısını oluşturan ağırlık değerleri Çizelge 3.3'te gösterilmiştir.

Çizelge 3.3. Eğitim sonucunda elde edilen ağırlık değerleri.

ω_{13}	ω_{14}	ω_{23}	ω_{24}	ω_{15}	ω_{35}	ω_{45}
-4,2647	33,5617	4,2038	-37,5558	12,7034	10,0290	-16,1788

Önceki aracın mesafesi hem eğri uydurma yöntemi ile hem de yapay sinir ağları ile hesaplanmıştır. Yapay sinir ağları ile elde edilen değerler kayda değer şekilde eğri uydurma yöntemiyle elde edilen değerlerden gerçeğe daha yakın sonuçlar üretmiştir ve bu sonuçlar Çizelge 3.4'te gösterilmiştir.

Çizelge 3.4. Eğri uydurma ve yapay sinir ağları ile elde edilen mesafe hesaplama sonuçları.

Piksel Derinliği	Gerçek Mesafe (m)	Eğri Uydurma ile Mesafe Tahmini	YSA ile Mesafe Tahmini	% Hata (Eğri Uydurma ile)	% Hata (YSA ile)
161	14	12,7734	14,3734	8,7614	-2,6671
173	20	18,1064	20,2833	9,4680	-1,4165
180	27	24,2526	26,7246	10,1756	1,0200
184	32	30,0043	33,0161	6,2366	-3,1753
187	40	36,0568	39,9016	9,8580	0,2460
190	49	44,2036	48,8431	9,7886	0,3202
192	55	51,1468	55,3415	7,0058	-0,6209

BÖLÜM 4

DENEYSEL SONUÇLAR

Bu çalışmada gerçekleştirilen erken uyarı sistemi şeritten ayrılma ve önden çarpma durumlarını içermektedir. Kameradan alınan giriş görüntüsü 640x360 piksel çözünürlüğüne sahiptir. Bu giriş görüntüsü şeritten ayrılma erken uyarı sistemi için 320x180 piksel çözünürlüğüne dönüştürülmüştür. Böylece algoritmanın daha performanslı çalışması sağlanmıştır. Önden çarpma uyarı sisteminde ise aracın mesafesinin daha doğru tahmin edilmesi gerektiği için giriş görüntüsü çözünürlüğü değiştirilmeden 640x360 piksel olarak gönderilmiştir.

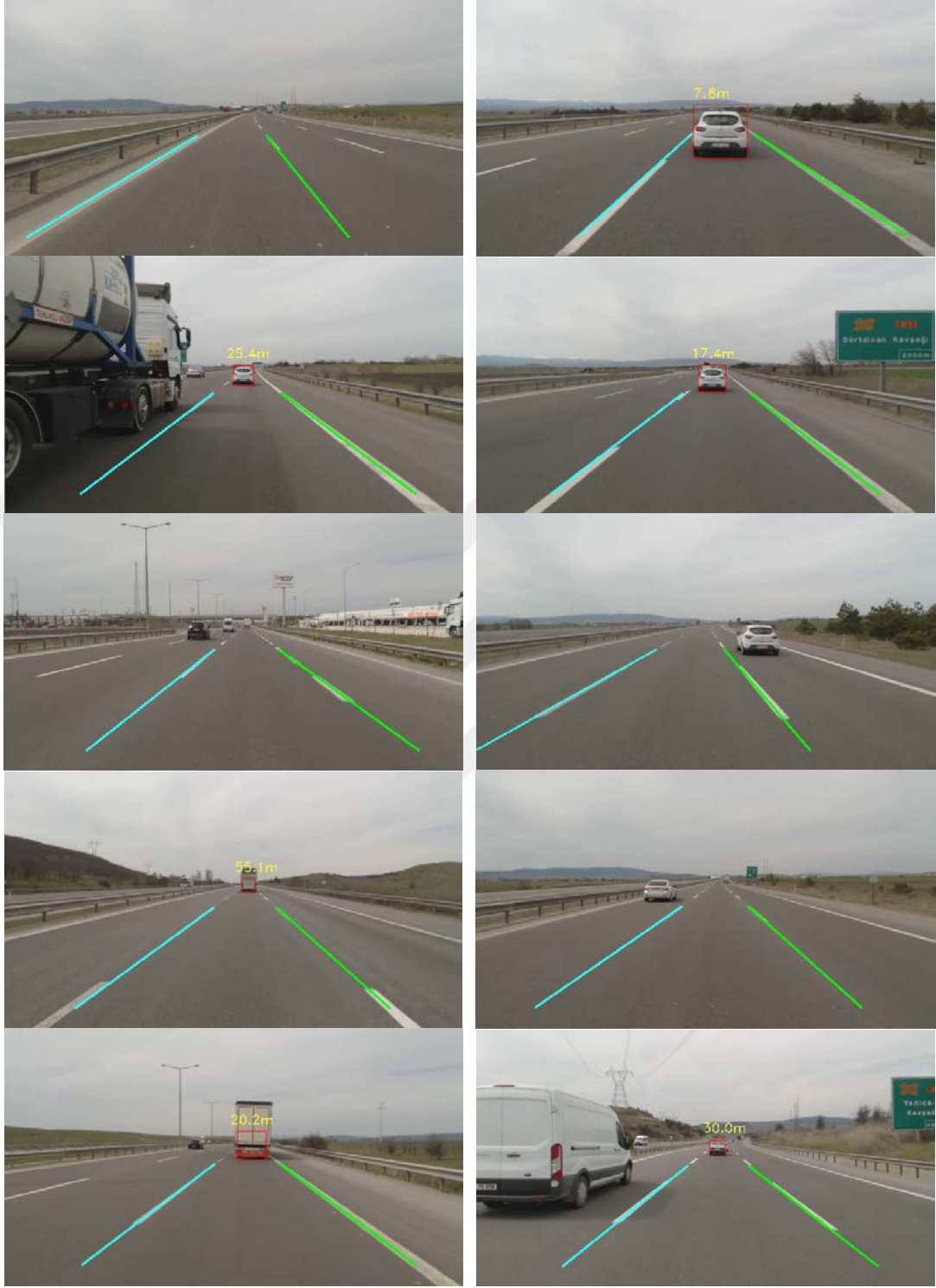
Geliştirilen erken uyarı sistemi 50km/s – 90km/s ve 90km/s – 120km/s hız aralıklarında test edilmiştir. Şeritten ayrılma uyarı sistemi düz ve kesikli şerit çizgilerini belirleyebilmektedir. Ayrıca renk temelli bir şerit belirleme işlemi yapılmadığı için tüm renkteki şerit çizgilerini de tespit edebilmektedir. Bu sistem sürücü sol ya da sağ şeritten ayrıldığı zaman sürücüye hangi şerit çizgisinden ayrıldığını sesli olarak uyarı vermektedir. Önden çarpma uyarı sistemi ise 7-60 metre arasında bir mesafede bulunan araçları tespit edebilmekte ve öndeki aracın mesafesi 15 metrenin altına düştüğü zaman sesli uyarı vermektedir.

Geliştirilen erken uyarı sistemi gerçek zamanlı olarak Odroid XU4 (2Ghz Cortex-A15 işlemci ve 2GB LPDDR3 RAM) ve Raspberry Pi 3 Model B (1,2GHz quad-core ARM Cortex-A53 işlemci ve 1GB LPDDR2 RAM) olmak üzere iki farklı ARM geliştirme kartı üzerinde gerçek zamanlı olarak çalıştırılmıştır. İşlem süresi Raspberry Pi 3 Model B için 65~85ms ve Odroid XU4 için 27~37ms olarak ölçülmüştür. Algoritmanın tüm bileşenlerinin ortalama işlem süreleri Çizelge 4.1’de detaylı olarak gösterilmiştir.

Çizelge 4.1. Algoritmanın tüm bileşenlerinin ortalama işlem süreleri.

Önerilen Algoritmanın Bileşenleri	Ortalama İşlem Süresi (ms)	
	Raspberry Pi 3	Odroid XU4
Giriş görüntüsü çözünürlük değiştirme	3,1	2,3
İlgili alan belirleme	0,14	0,1
Ters perspektif dönüşümü	7,65	2,2
Gri ölçekli görüntüye dönüştürme	0,54	0,1
Gauss gürültü giderme	1,4	0,33
Şerit çizgisi filtreleme	2,25	0,51
Canny kenar belirleme	2,5	0,56
Hough dönüşümü	14	5,37
Şerit çizgisi parametreleri ve uyarı mekanizması	0,01	0,01
Şerit Belirleme İçin Geçen Süre	31,59	11,48
İlgili alan belirleme	2,9	1,37
HSV renk uzayı dönüşümü	10,1	4,01
Renk temelli görüntü eşikleme	6,62	1,82
Araç sınırlayıcı kutusu bulma	1,3	0,19
Mesafe tahmini	0,04	0,01
Öndeki Cismi Belirlemek İçin Geçen Süre	20,96	7,4
İlgili alan belirleme	2,9	1,35
Gri ölçekli görüntüye dönüştürme	2,7	1,23
Gauss gürültü giderme	3,75	0,89
Otsu görüntü eşikleme	2,1	0,6
Araç sınırlayıcı kutusu bulma	0,7	0,16
Mesafe tahmini	0,04	0,01
Uyarı mekanizması	0,01	0,01
Öndeki Aracı Belirlemek İçin Geçen Süre	12,2	4,25
USB kameradan görüntü alınması	11	8,5
Toplam	75,75	31,63

Çalışmada geliştirilen sistemden elde edilen örnek sonuç görüntüleri Şekil 4.1’de gösterilmiştir.



Şekil 4.1. Elde edilen sonuç görüntüleri.



Şekil 4.1. (devam ediyor).

BÖLÜM 5

SONUÇ VE ÖNERİLER

Bu çalışmada olası kazaları önlemek için gerçek zamanlı video tabanlı erken uyarı sistemi geliştirilmiştir. Bu sistem, şeritten ayrılma uyarı sistemi ve önden çarpma uyarı sistemi olmak üzere iki kısımdan oluşmaktadır. Şeritten ayrılma uyarı sistemi için öncelikle şerit çizgileri belirlenmiştir. İlk olarak, şerit çizgilerini daha doğru tespit edebilmek için giriş görüntüsünün ters perspektif dönüşümü alınarak perspektif etkisi yok edilmiştir. Sonra şerit çizgileri için basit bir filtre uygulanmış ve bu çizgilerin kenarları Canny kenar belirleme yöntemiyle tespit edilmiştir. Daha sonra şerit çizgilerinin parametrelerini bulmak için Hough dönüşümü uygulanmıştır. Bulunan bu parametreler kullanılarak aracın şeritten ayrılıp ayrılmadığı tespit edilip sürücüye sesli uyarı verilmiştir. İkinci kısımda ise önden çarpmayı engellemek için öndeki aracın veya herhangi renkli bir nesnenin mesafesi tahmin edilip sürücüye sesli olarak uyarı verilmiştir. Mesafe tahminine geçmeden önce öndeki aracın veya renkli nesnenin tespit edilmesi gerekir. Bu tespit işlemi için öncelikle birinci kısımda bulunan şerit çizgileri yardımıyla giriş görüntüsü maskelenmiş ve öndeki ilgili nesne bu maskelenen görüntü içinde aranmıştır. Bu işlem yükünü oldukça azaltmıştır. Sonra aracın altında kalan gölgeden yararlanılarak araç ve renk filtreleme yöntemiyle renkli nesne tespit edilmiştir. Belirlenen aracın ve nesnenin mesafesi ise eğri uydurma yaklaşımı (ortalama hata %8,7563) ve YSA (ortalama hata %0,8701) kullanılarak tahmin edilip sonuçlar karşılaştırılmıştır.

Bu sistem gelecekte araca entegre edilerek daha kullanışlı hale getirilebilir. Şeritten ayrılma uyarı sistemi için sürücü sinyal vermeden şeritten ayrılıyorsa direksiyon titretme işlemi uygulanarak sürücü uyarılabilir ya da direksiyon otomatik yönlendirilerek aracın şeritte kalması sağlanabilir. Önden çarpmayı engellemek için ise kendi aracımızın hızı araçtan alınarak takip mesafesini korumuyorsa otomatik frenleme sistemiyle aracın hızı azaltılabilir.

KAYNAKLAR

1. İnternet: T.C. Ulaştırma, Denizcilik ve Haberleşme Bakanlığı, "T.C. Ulaştırma, Denizcilik ve Haberleşme Bakanlığı", http://www.ubak.gov.tr/BLSM_WIYS/KAIK/tr/Belgelik/20140312_103823_76347_1_76648.pdf (2016).
2. İnternet: TÜİK, "Türkiye İstatistik Kurumu", <http://www.tuik.gov.tr/UstMenu.do?metod=temelist> (2016).
3. Xie, B., "Lane Departure and Front Collision Warning System Using Monocular and stereo vision", *Worcester Polytechnic Institute*, (2013).
4. Hadi, R. A., Sulong, G., and George, L. E., "Vehicle Detection and Tracking Techniques: A Concise Review", *Signal & Image Processing: An International Journal (SIPIJ)*, 5 (1): 1–12 (2014).
5. Surgailis, T., Valinevicius, A., Markevicius, V., Navikas, D., and Andriukaitis, D., "Avoiding forward car collision using stereo vision system", *Elektronika Ir Elektrotechnika*, 18 (8): 37–40 (2012).
6. Teoh, S. S., "Development of a Robust Monocular-Based Vehicle Detection and Tracking System", *The University of Western Australia*, (2011).
7. Saha, A., Roy, D. Das, Alam, T., and Deb, K., "Automated Road Lane Detection for Intelligent", *Global Journal Of Computer Science And Technology*, 12 (6): (2012).
8. Smart, M. and Waslander, S. L., "Stereo Augmented Detection of Lane Marking Boundaries", *IEEE Conference On Intelligent Transportation Systems, Proceedings, ITSC*, 2015–Octob2491–2496 (2015).
9. Humaidi, A. J., "Performance Comparison for Lane Detection and Tracking with Two Different Techniques", *2016 Al-Sadeq International Conference On Multidisciplinary In IT And Communication Science And Applications (AIC-MITCSA)*, (2016).
10. Vishwakarma, S. K., Technology, I., Science, C., Yadav, D. S., and Science, C., "Analysis of Lane Detection Techniques using openCV", *2015 Annual IEEE India Conference (INDICON)*, 1–4 (2015).
11. Li, X., Wu, Q., Kou, Y., Hou, L., and Yang, H., "Lane Detection Based on Spiking Neural Network and Hough Transform", *2015 8th International Congress On Image And Signal Processing (CISP) 14-16 Oct. 2015*, 2 (Cisp): 531–535 (2015).
12. Chien-Cheng Tseng, Hsu-Yung Cheng, B.-S. J., "A Lane Detection Algorithm Using Geometry Information and Modified Hough Transform", *18th IPPR*

- Conference On Computer Vision, Graphics And Image Processing*, 1 (Cvqip): 796–802 (2005).
13. Miao, X., Li, S., and Shen, H., "On-board lane detection system for intelligent vehicle based on monocular vision", *International Journal On Smart Sensing*, 5 (4): 957–972 (2012).
 14. DhanaLakshmi, M. and Deepika, B. J. R., "A brawny multicolor lane coloration method to Indian scenarios", *IJRET International Journal Of Research In Engineering And Technology*, 202 – 206 (2012).
 15. Zhao, H., Teng, Z., Kim, H.-H., and Kang, D.-J., "Annealed Particle Filter Algorithm Used for Lane Detection and Tracking", *Journal Of Automation And Control Engineering*, 1 (1): 31–35 (2013).
 16. Yim, Y. U. and Oh, S. Y., "Three-feature based automatic lane detection algorithm (TFALDA) for autonomous driving", *IEEE Transactions On Intelligent Transportation Systems*, 4 (4): 219–225 (2003).
 17. Sehestedt, S., Kodagoda, S., Alempijevic, A., and Dissanayake, G., "Robust lane detection in urban environments", *Intelligent Robots And Systems, 2007. IROS 2007. IEEE/RSJ International Conference On*, (January 2016): 123–128 (2007).
 18. Aly, M., "Real time Detection of Lane Markers in Urban Streets", *IEEE Intelligent Vehicles Symposium*, 7–12 (2008).
 19. Kim, Z., "Robust lane detection and tracking in challenging scenarios", *IEEE Transactions On Intelligent Transportation Systems*, 9 (1): 16–26 (2008).
 20. Cheng, H., Yu, C., Tseng, C., Fan, K., Hwang, J., and Jeng, B., "Hierarchical lane detection for different types of roads", *2008 IEEE International Conference On Acoustics, Speech And Signal Processing*, 1349–1352 (2008).
 21. Assidiq, A. A., Khalifa, O. O., Islam, M. R., and Khan, S., "Real time lane detection for autonomous vehicles", *2008 International Conference On Computer And Communication Engineering*, 82–88 (2008).
 22. Amol Borkar, Monson Hayes, M. T. S., "ROBUST LANE DETECTION AND TRACKING WITH RANSAC AND KALMAN FILTER", *Proceedings Of The IEEE International Conference On Image Processing*, 3261–3264 (2009).
 23. Kamkar, S. and Safabakhsh, R., "Vehicle detection, counting and classification in various conditions", *IET Intelligent Transport Systems*, 10 (6): 406–413 (2016).
 24. Lin, Y., Fang, M., and Shihong, D., "An Object Reconstruction Algorithm for Moving Vehicle Detection Based on Three-Frame Differencing", *2015 IEEE 12th Intl Conf On Ubiquitous Intelligence And Computing And 2015 IEEE 12th Intl Conf On Autonomic And Trusted Computing And 2015 IEEE 15th Intl Conf On Scalable Computing And Communications And Its Associated Workshops (UIC-ATC-ScalCom)*, 1864–1868 (2015).

25. Gochoo, M., Bayanduuren, D., Khuchit, U., Battur, G., Tan, T. H., Kuo, S. Y., and Huang, S. C., "Design and application of novel morphological filter used in vehicle detection", *2016 IEEE/ACIS 15th International Conference On Computer And Information Science, ICIS 2016 - Proceedings*, (1): 3–7 (2016).
26. Arya, K. V, Tiwari, S., and Behwal, S., "Real-time Vehicle Detection and Tracking", *2016 13th International Conference On Electrical Engineering/Electronics, Computer, Telecommunications And Information Technology (ECTI-CON)*, 0–5 (2016).
27. Park, J. W. and Song, B. C., "Night-time vehicle detection using low exposure video enhancement and lamp detection", *International Conference On Electronics, Information, And Communications, ICEIC 2016*, (1): 3–4 (2016).
28. Yu, H., Yuan, Y., Guo, Y., and Zhao, Y., "Vision-Based Lane Marking Detection and Moving Vehicle Detection", *2016 8th International Conference On Intelligent Human-Machine Systems And Cybernetics (IHMSC)*, 1574–577 (2016).
29. Seenouvong, N., Watchareeruetai, U., Nuthong, C., Khongsomboon, K., and Ohnishi, N., "Vehicle Detection and Classification System Based on Virtual Detection Zone", *2016 13th Interbational Joint Conference On Computer Science And Software Engineering (JCSSE)*, (2016).
30. Harris, C. and Stephens, M., "A Combined Corner and Edge Detector", *Proceedings Of The Alvey Vision Conference 1988*, 147–151 (1988).
31. S. M. Smith, J. M. B., "ASSET-2:Real-Time Motion Segmentation and Shape Tracking", *IEEE Transactions On Pattern Analysis And Machine Intelligence*, 17 (8): 814–820 (1995).
32. Cao, Y., Renfrew, A., and Cook, P., "Novel Optical Flow Optimization Using Pulse-Coupled Neural Network", *International Symposium On Intelligent Signal Processing And Communication Systems*, 264–267 (2007).
33. Cao Yanpeng, Alasdair Renfrew, P. C., "Vehicle Motion Analysis Based on a Monocular Vision System", *RTIC 2008 And ITS United Kingdom Members' Conference*, 1–6 (2008).
34. Christos Tzomakas, W. von S., "Vehicle Detection in Traffic Scenes Using Shadows", *Technical Report, Institut Fur Neuroinformatik, Neuroinformatik, Ruhr-Universitat, Bochum, Germany*, (August): (1998).
35. Jin, L. S., Gu, B. Y., Wang, R. B., Guo, L., Zhao, Y. B., and Li, L. H., "Preceding vehicle detection based on multi-characteristics fusion", *2006 IEEE International Conference On Vehicular Electronics And Safety, ICVES*, (2): 356–360 (2006).
36. Zielke, T., Brauckmann, M., and Vonseelen, W., "Intensity and Edge-Based Symmetry Detection with an Application to Car-Following", *CVGIP: Image Understanding*, 58 (2): 177–190 (1993).

37. M. Bertozzi, S. Broggi, A. C., "A real-time oriented system for vehicle detection", *Journal Of Systems Architecture*, 317–325 (1997).
38. Tsai, L. W., Hsieh, J. W., and Fan, K. C., "Vehicle detection using normalized color and edge map", *IEEE Transactions On Image Processing*, 16 (3): 850–864 (2007).
39. Chen, Y. L., Chen, Y. H., Chen, C. J., and Wu, B. F., "Nighttime vehicle detection for driver assistance and autonomous vehicles", *Proceedings - International Conference On Pattern Recognition*, 1687–690 (2006).
40. Muad, A. M., Hussain, A., Samad, S. a., Mustafa, M. M., and Majlis, B. Y., "Implementation of inverse perspective mapping algorithm for the development of an automatic lane tracking system", *2004 IEEE Region 10 Conference TENCON 2004.*, A207–210 (2004).
41. McCall, J. C. and Trivedi, M. M., "Performance evaluation of a vision based lane tracker designed for driver assistance systems", *IEEE Intelligent Vehicles Symposium, Proceedings*, 2005153–158 (2005).
42. Dornaika, F., Alvarez, J. M., Sappa, A. D., and Lopez, A. M., "A new framework for stereo sensor pose through road segmentation and registration", *IEEE Transactions On Intelligent Transportation Systems*, 12 (4): 954–966 (2011).
43. Guo, C., Mita, S., and Mcallester, D., "Stereovision-based Road Boundary Detection for Intelligent Vehicles in Challenging Scenarios", *The 2009 IEEE/RSJ International Conference On Intelligent Robots And Systems*, 1723–1728 (2009).
44. Jiang, R., Terauchi, M., Klette, R., Wang, S., and Vaudrey, T., "Low-level image processing for lane detection and tracking", *Arts And Technology*, 30190–197 (2010).
45. Küçükyıldız, G. and Ocak, H., "Development and optimization of a DSP-based real-time lane detection algorithm on a mobile platform", *Turkish Journal Of Electrical Engineering & Computer Sciences*, 221484–1500 (2014).
46. Kumar, T. and Verma, K., "A Theory Based on Conversion of RGB image to Gray image", *International Journal Of Computer Applications*, 7 (2): 5–12 (2010).
47. Nieto, M., Arrospide Laborda, J., and Salgado, L., "Road environment modeling using robust perspective analysis and recursive Bayesian segmentation", *Machine Vision And Applications*, 22 (6): 927–945 (2011).
48. Canny, J., "A Computational Approach to Edge Detection", *IEEE Transactions On Pattern Analysis And Machine Intelligence*, PAMI-8 (6): 679–698 (1986).
49. Internet: Fei-Fie, "Canny Edge Detection", <http://www.cse.iitd.ernet.in/~pkalra/csl783/canny.pdf> (2016).
50. Jakub, H., "Road lane detection for Android", *Masaryk University*, (2014).

51. Balarini, J. P. and Nesmachnow, S., "A C ++ Implementation of Otsu ' s Image Segmentation Method", *Image Processing On Line(IPOL)*, 1–10 (2015).
52. Otsu, N., "A threshold selection method from gray-level histograms", *IEEE Transactions On Systems, Man, And Cybernetics*, 9 (1): 62–66 (1979).
53. Yang, X., Shen, X., Long, J., and Chen, H., "An Improved Median-based Otsu Image Thresholding Algorithm", *AASRI Procedia*, 3468–473 (2012).
54. Laganière, R. and Lagani, R., "OpenCV2 Computer Vision Application Programming Cookbook", (2011).
55. Tiwari, K., "Formulation Of A N-Degree Polynomial For Depth Estimation using a Single Image", *CoRR*, abs/1011.5(2010).
56. Internet: Software, S., "The KaleidaGraph Guide to Curve Fitting",<http://www.synergy.com/Tools/curvefitting.pdf> (2016).
57. Internet: Çayıroğlu, İ., "Yapa Sınır Ağları",<http://www.ibrahimcayiroglu.com/Dokumanlar/IleriAlgoritmaAnalizi/IleriAlgoritmaAnalizi-5.Hafta-YapaySinirAglari.pdf> (2016).

ÖZGEÇMİŞ

Kadir İLERİ 1990 yılında Fethiye’de doğdu ve 2004 yılında ilk ve orta öğrenimini Fethiye’de Güneşli İlköğretim Okulu’nda tamamladı. 2008 yılında Antalya Aksu Anadolu Öğretmen Lisesi’nden mezun oldu. Aynı yıl Kocaeli Üniversitesi, Elektronik ve Bilgisayar Eğitimi Bölümü’nde öğrenim görmeye başladı ve 2012 yılında buradan mezun oldu. 2013 yılında Mühendislik Tamamlama programı ile Karabük Üniversitesi, Bilgisayar Mühendisliği Bölümü’ne başladı ve 2014 yılında mezun oldu. 2013 yılında ÖYP (Öğretim Üyesi Yetiştirme Programı) kapsamında Araştırma Görevlisi olarak Karabük Üniversitesi, Mühendislik Fakültesi, Elektrik-Elektronik Mühendisliği Bölümü’ne atandı ve halen burada çalışmaya devam etmektedir.

ADRES BİLGİLERİ

Adres : Karabük Üniversitesi
Mühendislik Fakültesi
Balıklar Kayası Mevkii / KARABÜK

Tel : (506) 468 34 01

E-posta : kadirileri@karabuk.edu.tr