

**TÜRKİYE KARAYOLLARINDAKİ TRAFİK İŞARETLERİNİN
OTOMATİK TANINMASI**

**AUTOMATIC RECOGNITION OF TRAFFIC SIGNS IN
TURKEY**

HAMDİ YALIN YALIÇ

Hacettepe Üniversitesi

Lisansüstü Eğitim – Öğretim ve Sınav Yönetmeliğinin

Bilgisayar Mühendisliği Anabilim Dalı İçin Öngördüğü

YÜKSEK LİSANS TEZİ

olarak hazırlanmıştır.

2010

Fen Bilimleri Enstitüsü Müdürlüğü'ne,

Bu çalışma jürimiz tarafından **BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI**
'nda YÜKSEK LİSANS TEZİ olarak kabul edilmiştir.

Başkan :.....
Doç. Dr. Atila YILMAZ

Üye (Danışman) :.....
Dr. Ahmet Burak CAN

Üye :.....
Yard. Doç. Dr. Mustafa EGE

Üye :.....
Yard. Doç. Dr. Kayhan İMRE

Üye :.....
Dr. Ayça TARHAN

ONAY

Bu tez/...../..... tarihinde Enstitü Yönetim Kurulunca kabul edilmiştir.

Prof. Dr. Adil DENİZLİ
Fen Bilimleri Enstitüsü Müdürü

TÜRKİYE KARAYOLLARINDAKİ TRAFİK İŞARETLERİNİN OTOMATİK TANINMASI

HAMDİ YALIN YALIÇ

ÖZ

İlerleyen bilgi teknolojileriyle birlikte, akıllı taşıtların ve sürücü destek sistemlerinin günlük hayatımızda daha çok yer edinmesi kaçınılmazdır. Bu tür taşıtlar trafik işaretleri yardımıyla yol hakkında pek çok bilgi edinebilirler. Bu bilgiyi kullanarak sürücüyü uyarabilecekleri gibi, aracın kontrolünü sağlayarak daha konforlu bir sürüş ortamı da yaratabilirler. Ayrıca trafik işaretlerinin zamanında ve yerinde tespiti, güvenli bir sürüş sağlanmasında da en önemli etkidir. Trafik işaretlerinin gerçek zamanlı ve otomatik olarak tanınması, kazaların azalmasına yardımcı olacaktır.

Bu tez kapsamında, Türkiye Karayollarındaki trafik işaretlerinin otomatik tanınması ve eşleştirilmesi için geliştirilen bir yaklaşım anlatılmaktadır. Trafik işaretinin bir sahne içerisinden tespiti için renk bilgileri kullanılmış, çeşitli görüntü işleme teknikleri uygulanarak işaretin bulunduğu bölge elde edilmiştir. Trafik işaretinin tespit edildiği bölgeden çıkarılan niteliklerle, veritabanındaki işaretlerin nitelikleri arasında bir eşleme yapılarak işaretin türü tespit edilmektedir. Geliştirilen yöntemde çeşitli çevrelerde, farklı boyutlarda ve görüş açılarında olan işaretler tanınabilmektedir. Yöntemin hız başarımı gelecekte bir gerçek zamanlı sistem gerçekleştirimine olanak sağlayacak düzeydedir.

Anahtar Kelimeler: Trafik işareti tanıma, video ve görüntü işleme, nitelik çıkarma.

Danışman: Dr. Ahmet Burak CAN, Hacettepe Üniversitesi, Bilgisayar Mühendisliği Bölümü

AUTOMATIC RECOGNITION OF TRAFFIC SIGNS IN TURKEY

HAMDİ YALIN YALIÇ

ABSTRACT

Intelligent vehicles and driver support systems will take place more in our daily life with the development of information technology. This kind of vehicles can obtain a lot of information about roads from the traffic signs. Such information can be used to warn drivers, regulate traffic or control movements of the vehicle, as a result, more comfortable driving could be provided. Moreover, recognition of traffic signs at the right time and place is an important factor to ensure the safe journey. Automatic recognition of traffic signs in real-time will help to reduce accidents.

This thesis introduces an approach to detect and match traffic signs in Turkish Highways automatically. In the detection phase, color information is used to find a sign in the scene and the region of sign is extracted by using image processing techniques. The features extracted from the regions of signs are matched with the features of traffic signs in the database to determine the type of signs. With the proposed solution, signs were recognized in various scenes, different scales, and viewing angles. Speed performance of the approach is sufficient to implement a real-time system in the future.

Keywords: Traffic sign recognition, video and image processing, feature extraction.

Advisor: Dr. Ahmet Burak CAN, Hacettepe University, Department of Computer Science and Engineering.

TEŞEKKÜR

Tez konusunun belirlenmesini sađlayan, tez metninin yazılmasına ve tez alıřmasının hazırlanmasına yardımcı olan Sayın Dr. Ahmet Burak Can'a,

Tez metnini inceleyerek biçim ve ierik bakımından son halini almasına yardımcı olan Sayın Do. Dr. Atila Yılmaz'a,

Tez metnini inceleyerek biçim ve ierik bakımından son halini almasına yardımcı olan Sayın Yrd. Do. Dr. Mustafa Ege'ye,

Tez metnini inceleyerek biçim ve ierik bakımından son halini almasına yardımcı olan Sayın Yrd. Do. Dr. Kayhan İmre'ye,

Tez metnini inceleyerek biçim ve ierik bakımından son halini almasına yardımcı olan Sayın Dr. Ebru Sezer'e,

Tez metnini inceleyerek biçim ve ierik bakımından son halini almasına yardımcı olan Sayın Dr. Aya Tarhan'a,

Tez alıřmalarında verdiđi desteklerden dolayı Ali Seydi Keeli'ye ve Aydın Kaya'ya,

Yüksek lisans eđitimi boyunca burs sađlayarak maddi ve manevi destek olan TÜBİTAK'a,

Bana her konuda destek olan canımdan ok sevdiđim Özüm Iřıtman'a canı gönülden teşekkür ederim.

İÇİNDEKİLER

	<u>Sayfa</u>
ÖZ	i
ABSTRACT	ii
TEŞEKKÜR.....	iii
İÇİNDEKİLER.....	iv
ŞEKİLLER DİZİNİ.....	vi
ÇİZELGELER DİZİNİ.....	ix
SİMGELER ve KISALTMALAR DİZİNİ	x
SÖZLÜK.....	xi
1. GİRİŞ.....	1
2. GENEL BİLGİLER.....	4
2.1. Renk Özniteliği	4
2.1.1. Renk Uzayları	4
2.1.1.1. RGB Renk Uzayı	4
2.1.1.2. HSV Renk Uzayı	6
2.1.1.3. CIE XYZ Renk Uzayı	7
2.1.1.4. CIE L*a*b Renk Uzayı	9
2.2. İkili Görüntü (<i>Binary Image</i>) ve Morfolojik (<i>Morphological</i>) İşlemler.....	10
2.2.1 Genişletme (<i>Dilation</i>)	10
2.2.2. Boşluk Doldurma (<i>Hole Filling</i>).....	12
2.3. Şablon Eşleştirme (<i>Normalized Cross Correlation</i>).....	14
2.3.1 Karşıtlık Esnetme.....	14
2.3.1 Çapraz Korelasyon (<i>Cross-Correlation</i>)	15
3. NESNE TAKİBİ.....	16
3.1. Nitelik Çıkarım Yöntemleri	16
3.1.1. SIFT Yöntemi.....	17
3.1.1.1. Ölçek Uzayında En Uç Değer Tespiti.....	17
3.1.1.2. Anahtar Nokta Konumlandırma.....	20
3.1.1.3. Yönelim Atama	21
3.1.1.4. Anahtar Nokta Tanımlayıcısı	22
3.2. Nitelik Eşleştirme Yöntemleri	23
3.2.1. Öklid Uzaklığı.....	24
4. TRAFİK İŞARETLERİNİN TANINMASINA YÖNELİK YAPILMIŞ ÇALIŞMALAR	25

5. TRAFİK İŞARETLERİNİN OTOMATİK TANINMASI VE EŞLEŞTİRİLMESİ.....	30
5.1. Sistemin Genel Görünümü	30
5.1.1. Veriler	32
5.1.2. Yazılım ve Donanım Araçları.....	32
5.2. Yöntem Adımları ve Uygulamaları	34
5.2.1. Çerçeve İçindeki Trafik İşaretlerinin Tespit Edilmesi	34
5.2.1.1. Renk Eşiği Uygulama Adımında RGB Renk Uzayının Kullanımı.....	35
5.2.1.2. Renk Eşiği Uygulama Adımında HSV Renk Uzayının Kullanımı	38
5.2.1.3. Gürültülerin Yok Edilmesi	38
5.2.1.4. Genişletme İşlemi	39
5.2.1.5. Farklı Renk Eşiklemelerinin Birleştirilmesi Ve İşaret Adaylarının Belirlenmesi	40
5.2.1.6. İşaret Renginin Tespit Edilmesi	44
5.2.2. Trafik İşaretlerinin Eşleştirilmesi.....	45
5.2.2.1. Sıft Kullanarak Eşleştirme.....	46
5.2.2.1.1. İşaretin Dışının Temizlenmesi	47
5.2.2.1.2. SIFT Nitelik Çıkarımı ve Anahtar Nokta Tanımlayıcıların Eşleştirilmesi	49
5.2.2.2. Normalleştirilmiş Korelasyon Kullanarak Eşleştirme	51
5.2.3. Tanınan İşaretlerin Sahneler Arası Denetimi.....	52
5.2.4. Tanınan İşaretlerin Geri Bildirimi	53
5.3. Başarım.....	54
6. SONUÇ	58
KAYNAKLAR.....	60

ŞEKİLLER DİZİNİ

Şekil 2.1	RGB renk küpü [10].....	5
Şekil 2.2a	HSV renk uzayının konik gösterimi [11].....	7
Şekil 2.2b	HSV renk uzayının silindir biçimli gösterimi [11].....	7
Şekil 2.3	RGB renkleri kullanılarak görüntülenebilen CIE renk gamutu [12].....	8
Şekil 2.4	CIE L*a*b renk uzayı modeli.....	9
Şekil 2.5a	İkili görüntü [10].....	11
Şekil 2.5b	Genişletme işlemi sonucunda elde edilen görüntü [10].....	11
Şekil 2.5c	Genişletme işleminde kullanılan yapısal eleman [10].....	11
Şekil 2.6a	A kümesi [10].....	12
Şekil 2.6b	A'nın tümleyeni [12].....	12
Şekil 2.6c	B yapısal elemanı [10]	12
Şekil 2.6d	Sınırın içindeki başlangıç noktası [10].....	12
Şekil 2.6e-h	Eşitlik 2.8'in çeşitli adımları [10].....	12
Şekil 2.6i	Varılan sonuç 2.3a ve 2.3b'nin birleşimi [10].....	12
Şekil 2.7a	İkili görüntü [10]	13
Şekil 2.7b	Genişletme işlemi sonucunda elde edilen görüntü [10].....	13
Şekil 2.8a	Şablon görüntü.....	15
Şekil 2.8b	Hedef görüntü.....	15
Şekil 2.8c	Çapraz korelasyon kullanılmasıyla elde edilen eşleştirme sonucu... 15	
Şekil 3.1	DoG uygulama örneği [18].....	18
Şekil 3.2	Aday anahtar noktaların belirlenmesi için piksellerin komşuları ile karşılaştırılması [18].....	18
Şekil 3.3	Anahtar nokta tanımlayıcılarının oluşturulması [18].....	23
Şekil 4.1	Hız levhalarındaki rakamlardan elde edilen matrislere örnekler [16]	25
Şekil 4.2a	Görüntü elde edilmesi [7].....	26
Şekil 4.2b	Renk bölütlemesi [7].....	26
Şekil 4.2c	Şekil algılama [7].....	26
Şekil 4.2d	Ölçekleme [7].....	26
Şekil 4.2e	Sınıflandırma [7].....	26
Şekil 4.2f	Çıktı [7].....	26
Şekil 4.3a	İlgi penceresinin şeması [29].....	27
Şekil 4.3b	İlgi penceresi ile tespit edilen yönelim elemanlarına örnek [29].....	27
Şekil 4.4a	Haar dalgacığına örnek [30].....	28

Şekil 4.4b	AdaBoost eğitimi sonucu elde edilen ve nitelik çıkarmakta kullanılan Haar dalgacıkları [30].....	28
Şekil 4.5	5 farklı türdeki trafik işaretinin alt-parçalara ayrılması [31].....	28
Şekil 5.1	Yazılımın kullanıcı arayüzü.....	30
Şekil 5.2	Yazılımın genel görünümü.....	31
Şekil 5.3	Trafik işareti tespitinin genel görünümü.....	34
Şekil 5.4a	RGB renk uzayındaki görüntü.....	36
Şekil 5.4b	Görüntünün kırmızı renk kanalı.....	36
Şekil 5.4c	Görüntünün mavi renk kanalı.....	36
Şekil 5.5a	Kırmızı renk eşikleme sonrası görüntü.....	37
Şekil 5.5b	Mavi renk eşikleme sonrası görüntü.....	37
Şekil 5.6	HSV renk uzayında renk özü değerinin renklere göre değişimi.....	38
Şekil 5.7a	Kırmızı işaretin olduğu görüntüde alan yok etme.....	39
Şekil 5.7b	Mavi işaretin olduğu görüntüde alan yok etme.....	39
Şekil 5.8	Görüntü içinde parçalara ayrılmış işaretler.....	40
Şekil 5.9a	İşaretin genişletme öncesindeki durumu ve sarmalayan kutu içine alınması.....	40
Şekil 5.9b	İşaretin genişletme sonrasındaki durumu ve kutu içine alınması.....	40
Şekil 5.9c	Genişletme işleminde kullanılan yapısal eleman.....	40
Şekil 5.10a	Kırmızı renkli işaretlerin tespitinde kullanılan görüntü.....	41
Şekil 5.10b	Mavi renkli işaretlerin tespitinde kullanılan görüntü.....	41
Şekil 5.10c	(a) ve (b)'nin VEYA işlecine tabi tutulması.....	41
Şekil 5.11	İçerisinde trafik işareti tespit edilen bazı sahneler.....	43
Şekil 5.12	Kırmızı ve mavi renkleri içeren işaretler.....	45
Şekil 5.13	Sahne içinde tespit edilmiş trafik işaretleri.....	46
Şekil 5.14a	İşaretin orijinal görüntüsü	48
Şekil 5.14b	Renk eşikleme işleminden sonra elde edilen ikili görüntü.....	48
Şekil 5.14c	Boşluk doldurma işleminden sonraki görüntü.....	48
Şekil 5.14d	Orijinal görüntüye (c) maskesi uygulanarak işaretin dışının temizlenmesi.....	48
Şekil 5.15	Dışı temizlenmiş bazı işaretler.....	49
Şekil 5.16	Tespit edilen işaretler ile veri kümesindeki en fazla benzerliğe sahip işaretlerin nitelik eşleşmeleri.....	50
Şekil 5.17	Aynı levhanın dönüşümüyle elde edilen işaretler.....	51
Şekil 5.18a	Tespit edilen işaret.....	52
Şekil 5.18b	Şablon işaretlere uyacak şekilde yeniden boyutlandırılması.....	52

Şekil 5.18c	Şablon kümesindeki işaretler ile karşılaştırılarak korelasyonun hesaplanması.....	52
Şekil 5.19	Taninan trafik işaretinin geri bildirimi.....	53
Şekil 5.20	Tanınamayan işaretlere örnekler.....	56

ÇİZELGELER DİZİNİ

Çizelge 4.1	Yapılan çalışmaların başarımları bilgileri.....	29
Çizelge 5.1	Olasılık matrisi.....	54
Çizelge 5.2	İşaret tanımlama kullanılan yöntemin olasılık matrisi.....	55
Çizelge 5.3	Belirli hızlarda giden taşıttaki sürücünün düşünme süresi içerisinde taşıttın kat ettiği mesafe [37].....	55
Çizelge 5.4	Hızı belirleyen temel adımların ortalama hesaplama süreleri.....	57

SİMGELER ve KISALTMALAR DİZİNİ

RGB	Red Green Blue
HSV	Hue Saturation Value
CMY	Cyan Magenta Yellow
YUV	Luminance Chrominance1 Chrominance1
YIQ	Luminance In-phase Quadrature
BPP	Bits Per Pixel
CAD	Computer-aided Design
CIE	International Commission on Illumination
SIFT	Scale-Invariant Feature Transform
DoG	Difference of Gaussian
KLT	Kanade-Lucas-Tomasi
DSA	Driving Standards Agency

SÖZLÜK

Hue	Renk Özü
Saturation	Doygunluk
Value	Parlaklık
Background	Arka plan
Foreground	Ön plan
Calibration	Ölçümleme
Intensity	Yoğunluk
Keypoint	Anahtar Nokta
Orientation	Yönelim
Descriptor	Tanımlayıcı
Feature	Nitelik
Scale-Space	Ölçek Uzayı
Extrema	En Uç Değer
Octave	Oktav
Convolution	Evrişim
Correlation	Korelasyon
Noise	Gürültü
Dilation	Genişletme
Bounding Box	Sarmalayan Kutu
Operator	İşleç
Sensitivity	Duyarlılık
Specificity	Özgüllük
Accuracy	Doğruluk
Contingency	Olasılık

1. GİRİŞ

Trafik işaretlerinin tehlikeleri uyararak, trafiği tanzim etmek ve yol hakkında bilgi vermek gibi farklı işlevleri vardır. Bu işlevler, işaretlerin belirli renklere ve biçimlere sahip olmalarıyla ilişkilidir. Bu farklılıklar işaretlerin ayrıştırılmasında ve tanınmasında etkilidir.

Trafik işaretlerinin tam zamanında ve yerinde doğru bir şekilde tespit edilmesi sürücülerin kendilerine ve taşıt içindeki yolculara, güvenli bir yolculuk sağlamaları için oldukça önemlidir. Fakat sürücünün dikkatinin dağılması, görüş açısının yetersiz olması veya hava şartları gibi sebeplerden dolayı işaretin zamanında fark edilmesi güçleşmektedir. Zamanında fark edilmeyen bir işaret ise ölümcül kazalara yol açabilir. Örneğin “Dur” veya “Yol Ver” levhalarının sürücü tarafından fark edilememesi, kazaların önde gelen sebeplerinden biridir. Bazı işaretlerin gözden kaçırılması sadece taşıt içindekilerin değil, karşıdan karşıya geçen yayaların da hayatını tehlikeye atmaktadır.

Taşıtlara yerleştirilecek ve trafik işaretlerini otomatik tanıyacak bir sistem, güvenli bir sürüş sağlamaya yardımcı olacaktır. Örneğin, Azami hız levhalarının tanınması, taşıt, hız sınırının üzerinde seyir halinde ise sürücüyü uyararak, hızını düşürmesini sağlayabilir. Böylece sürücü yoldaki işaretlere ve bunları konumlandırmaya harcayacağı dikkati, yola ve yoldaki taşıtlara verebilir.

İşaretleri zamanında tanıyıp sürücüyü ikaz etmelerinin yanında, önümüzdeki yıllarda daha da yaygınlaşması beklenen akıllı taşıtlarda, bu tür sistemlerin kullanılması planlanmaktadır. Günümüzde, akıllı taşıtlarda kullanılan sürücü destek sistemleri, yol kılavuz çizgilerinin tespiti, diğer taşıtların ve taşıt yolundaki yayaların tespiti üzerinde durmaktadır. Trafik işaretlerinin tanınmasına yönelik sistemlerin gelişmesiyle, destek sistemleri taşıtların sürüşüne yönelik konulara da trafik işaretlerine dayanarak karar verebilir. Örneğin mesafe kontrollü hız sabitleyici sisteme (ACC – *Adaptive Cruise Control*) [1] sahip olan taşıtlar, sizin belirlediğiniz hıza ulaştıktan sonra öndeki daha yavaş giden araca yaklaştığında fren yapar ve öndeki aracın hızında yola devam eder. Buradaki hız sınırının sürücü tarafından belirlenmesi yerine, yoldaki azami hız levhalarında belirtilen hızlara göre ayarlanması daha güvenli olacaktır.

Trafik işaretlerindeki hız sınırlarına göre taşıtın hızının kontrol edilmesinin yanında, girilmez yollara girmeyi veya dönülmesi yasak yönlere dönmeyi engellemek, yaya geçitlerinde, hız kesici kasislerde, virajlarda ve kavşaklarda yavaşlamak gibi taşıt hareketleri, akıllı taşıtlar tarafından kontrol edilebilir.

Trafik işaretlerinin tanınmasının taşıt destek sistemlerinde kullanılmasının dışında diğer bir uygulama alanı otoyol bakımının sağlanmasıdır. Günümüzde bu işlem sorumlu memur tarafından yapılmakta, taşıttan kaydedilmiş video görüntüsünü izleyerek trafik işaretlerinin durumunu ve konumunu denetlemektedir. Bu çok zahmetli ve yorucu bir iştir çünkü bu görüntüler saatler süren kayıtlardır ve işaretler farklı aralıklarla belirlemektedir. Sorumlu kişinin büyük dikkat göstermesi gerekir. Bu nedenle, otoyollardaki demirbaşların konumlandırılmasına yönelik AUTOCAT[2] isimli bir (ESPRIT) Avrupa teknoloji projesi mevcuttur.

Bu tür sistemler, otoyol demirbaşlarının denetlenmesinin yanında şehir içi ve mahalle içi trafik işaretlerinin denetlenmesi için de kullanılabilir. Bu işlem çevre koşulları sebebiyle otoyollardakinden daha zordur. Çünkü işaretler her zaman taşıtın hareket yönüne dik olmayabilir, işaretlerde bozulma ve yıpranma gerçekleşmiş olabilir ve en önemlisi aynı renkte ve biçimde birçok farklı nesne yollarda bulunabilir.

Görüntü işleme alanındaki gelişmeler ve donanım hızlarının artarken maliyetlerinin azalması ile özellikle son on yılda trafik işaretlerinin tanınmasına yönelik çalışmalar hızlanmıştır. Trafik işaretlerinin tanınması için geliştirilen sistemlerde üç ana yaklaşım uygulanmıştır: renk temelli tanıma [3,4], şekil temelli tanıma [5] ve renk ile şekil bilgilerinden yararlanılan sinir ağları kullanılarak yapılan tanıma [6,7,8,9]. İşaretin tanınması için geliştirilen yöntemlerde genelde iki temel ve birbirinden bağımsız adım bulunmaktadır: işaretin tespiti ve işaretin eşleştirilmesi (sınıflandırılması). Renk, görüntülerde çok baskın olan bir özneliktir ve görüntü hakkında çok sayıda anahtar bilgi taşımaktadır. Bu sebepten dolayı trafik işaretinin sahnenin geri kalanından ayrıştırılmasında çoğunlukla renk bilgisi kullanılmaktadır. Çünkü bütün bir sahnedeki renk niteliklerinin çıkarılması, şekil niteliklerinin çıkarılmasına göre oldukça hızlıdır.

Yapılan ilk çalışmalarda işaret tanıma sistemleri iki kameradan oluşmaktaydı. Kameralardan bir tanesi geniş-açılı görüntüyü kullanarak ve renk, yoğunluk ve şekil bilgilerinden yararlanarak trafik işaretinin konumunu tespit etmekte, diğer kamera ise belirlenen konumdaki işaretin daha kaliteli görüntüsünü çekmekte idi. İşareti sınıflandırma aşamasında ise görüntü eşleştirme yöntemi kullanılmıştır.

Trafik işaretinin sınıflandırılması adımı sinir ağlarının kullanımı da, oldukça yaygın bir yöntemdir. İşaretlerden elde edilen renk (kırmızı, mavi, sarı, vb.), şekil (üçgen, daire, dörtgen, vb.) gibi nitelikler kullanılarak farklı kategoriler oluşturulmuş, bu kategoriler kullanılarak yaratılan sinir ağları kullanılarak tespit edilen işaretin sınıflandırılması sağlanmıştır.

Bu tez kapsamında yapılan çalışmada, Türkiye Karayollarındaki trafik işaretlerinin otomatik tanınması ve eşleştirilmesi için bir yazılım tasarlanmış ve gerçekleştirilmiştir. Tasarlanan yazılım kullanıcıdan herhangi bir geribildirim almaksızın ve otomatik bir şekilde, kamera görüntülerinden, trafik işaretlerinin tanınmasını ve eşleştirilmesini sağlamaktadır.

Tezin 2. bölümünde çalışma kapsamında kullanılan renk uzayları hakkında bilgi verilmiş, ikili görüntüler üzerinde kullanılan morfolojik işlemler ve eşleştirme adımı kullanılan yöntemlerden biri olan normalleştirilmiş korelasyon hakkında bilgi verilmiştir.

3. bölümde, tez çalışması kapsamında nesne tanımda ana yaklaşım olarak kullanılan nitelik çıkarma ve nitelik eşleştirme yöntemi anlatılmıştır. 4. bölümde trafik işaretlerinin tanınması konusunda literatürde yer alan çalışmalardan bahsedilmiş, ilgili çalışmaların işleyişleri kısaca açıklanmıştır. 5. bölümde tez kapsamında yapılan çalışmalar açıklanmıştır. Bu kapsamda ilk olarak video görüntüsünden elde edilen çerçeve içerisindeki trafik işaretlerinin tespitinde kullanılan adımlar anlatılmıştır. Daha sonra işaretlerin eşleştirilmesi için geliştirilen yöntemler açıklanmıştır. 6. bölümde ise çalışmanın sonuçları irdelenmiş ve ileriki aşamalarda yapılabilecek olan gelişmelerden bahsedilmiştir.

2. GENEL BİLGİLER

2.1. Renk Özniteliği

Renk özniteliği, görüntü işleme alanında en çok kullanılan görsel özniteliktir. Rengin üç-boyutlu yapısı, bir resmin tek boyutlu gri değerlerine oranla çok üstün bir ayırt etme gücü sağlamaktadır. Uygun bir renk tanımlaması seçmeden önce, kullanılacak renk uzayı belirlenmelidir.

2.1.1. Renk Uzayları

Renkli görüntüyü ifade etmek için, saklama ortamları ve uygulamaların gereksinimlerine uygun olacak şekilde çeşitli renk uzayları kullanılmaktadır. Kullanılacak uzayın seçimi, yapılacak işleme göre değişmektedir. Bir rengi, bir uzayda temsil etmek için temel renk bileşenleri, parlaklık, renk tonu, doygunluk gibi öğeler kullanılır. RGB, HSV, CMY, YUV, YIQ en çok kullanılan renk uzaylarına örnek verilebilir[10].

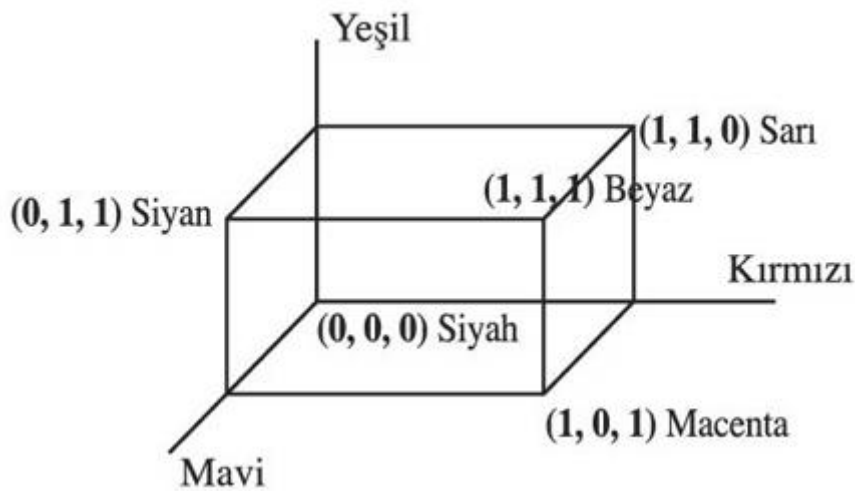
2.1.1.1. RGB Renk Uzayı

RGB renk uzayı, kırmızı, mavi ve yeşil ışığın çeşitli oranlarda birleşerek diğer renkleri meydana getirdiği toplamalı bir renk modelidir. RGB kısaltması üç temel öğe olan kırmızı (*red*), yeşil (*green*) ve mavi (*blue*) rengin baş harflerinden oluşur. RGB renk uzayı serisindeki diğer tüm renkler, belli miktarlarda kırmızı, mavi ve yeşil ışığın birleşmesinden oluşur. RGB'nin toplamalı bir renk uzayı oluşunun nedeni, iki farklı frekansa sahip ışık bir araya geldiğinde bu olayı gözlemleyen kişi için bu renklerin toplamından oluşan yeni bir renk olarak algılanmasıdır. Kırmızı, mavi ve yeşil renklerin seçilme sebebi ise, her birinin insan gözündeki renge duyarlı algılayıcı türlerinden birine karşılık gelmesidir. Kırmızı, mavi ve yeşil renklerin uygun karışımlarıyla insanın algıladığı renklerin hepsini oluşturmak mümkündür. Örneğin; saf kırmızı ve yeşilin birleşiminden sarı; kırmızı ve mavinin birleşiminden macenta ve üçünün birleşiminden de beyaz renk elde edilir. RGB renk uzayının genel kullanım alanlarından biri, bilgisayar grafiğinin ekran üstünde görüntülenmesidir. Ekran üstündeki her bir piksel, bilgisayar belleğinde kırmızı, mavi ve yeşil için ayrı değerler halinde tutulur. Her piksel, kırmızı, mavi ve yeşil bileşenlerin her biri için 8 bit olmak üzere toplam 24 bitlik bilgi ile temsil edilir. Bu sistemle yaklaşık 16.7 milyon farklı renk elde edilebilmektedir ve bu insanın ayırt

edebileceği renk sayısından çok daha fazladır. Yazılı olarak, RGB değerleri, sırasıyla kırmızı, mavi ve yeşili temsil eden üç tamsayı (24 bpp, 0-255 arasında) ile ifade edilir. Örneğin;

- (0, 0, 0) siyah;
- (255, 255, 255) beyaz;
- (255, 0, 0) kırmızı;
- (0, 255, 0) yeşil;
- (0, 0, 255) mavi;
- (255, 255, 0) sarı;
- (0, 255, 255) siyan;
- (255, 0, 255) macenta.

RGB renk uzayı genellikle bir küp ile gösterilir. Her renk (kırmızı, mavi, yeşil), üç boyutlu uzayda birbirine dik olan koordinat eksenlerinden birine atanır. Renk küpünün her ekseninde, renkler, rengin hiç bulunmadığı durumdan, tam doygun renge doğru uzanır. Renk küpü katıdır, yani küpün üzerindeki herhangi bir nokta r, g, b üçlüsüyle ifade edilebilir. Siyahtan $(0,0,0)$ beyaza $(1,1,1)$ doğru olan köşegen tüm gri renkleri temsil eder, bu köşegen üzerindeki her noktada kırmızı, yeşil ve mavi bileşenleri eşit miktardadır. Kırmızı, mavi, yeşil renklerin 0-255 arasında değişmesi, üç boyutlu renk küpünün doğrusal olarak ölçeklendirilmesidir. Örnek bir renk küpü aşağıdaki şekilde görülmektedir (Şekil 2.1), bazı anahtar renkler ve koordinatları küpün üzerinde gösterilmiştir.



Şekil 2.1: RGB renk küpü [10]

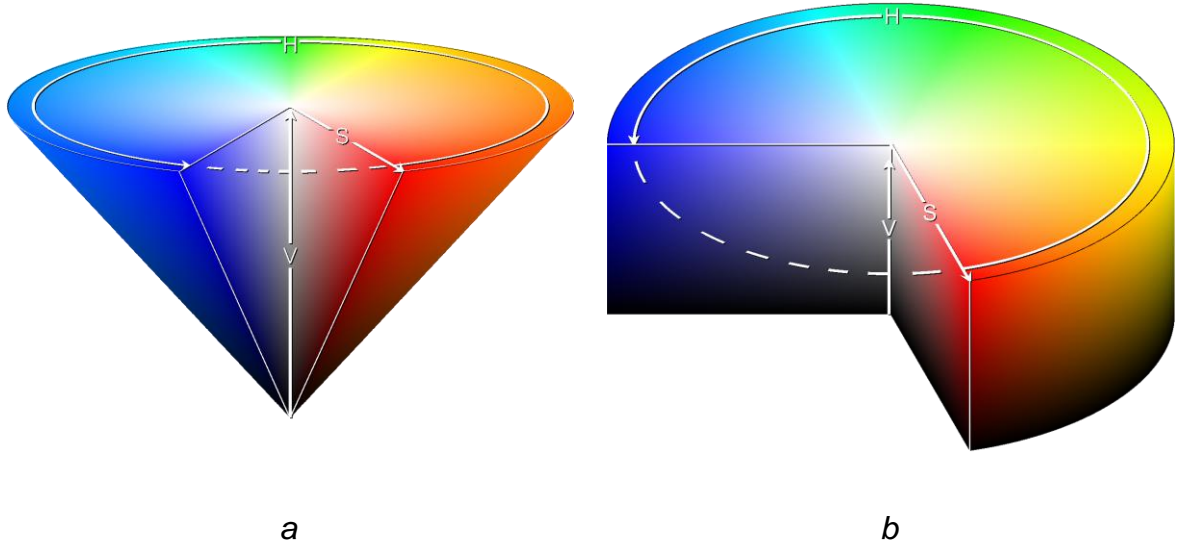
2.1.1.2. HSV Renk Uzayı

HSV (Hue, Saturation, Value) veya HSB (Hue, Saturation, Brightness) renk uzayı, renkleri sırasıyla renk özü, doygunluk ve parlaklık olarak tanımlar [11].

- Renk özü (H), rengin baskın dalga uzunluğunu belirler, örneğin sarı, mavi, yeşil, vb.
 - Açısal bir değerdir 0° - 360° , bazı uygulamalarda ise 0-100 arası olağanlaştırılır.
- Doygunluk (S), rengin canlılığını belirler. Yüksek doygunluk canlı renklere neden olurken, düşük olasılık rengin gri tonlarına yaklaşmasına neden olur.
 - 0-100 arasında değişir.
- Parlaklık (V) ise rengin aydınlığını yani içindeki beyaz oranını belirler.
 - 0-100 arasından değişir.

HSV renk uzayı 1978 yılında Alvy Ray Smith tarafından tanımlandı. Amacı RGB uzayına göre insan görü düzeneğine daha yakın bir yapı oluşturmaktır. HSV, RGB renk uzayından doğrusal olmayan bir dönüşüm ile elde edilir. HSV ayrıca aygıt bağımlıdır. Yani bu uzayda tanımlı bir renk, rengi üreten aygıt cihazına göre değişim gösterebilir. Aygıt bağımsız renk gösterimi için CIE L^*a^*b veya CIE L^*u^*v kullanılmalıdır.

HSV uzayı, ilk tanımlandığı zamanlarda konik bir biçime sahipti (Şekil 2.2a). Ancak sonraki yıllarda, gerçek zamanlı geçerli koordinat denetimi için zamanın bilgisayarları yeterli olmadığından silindir biçimine dönüştürüldü (Şekil 2.2b). Konik biçimde, aydınlık düzeyi azaldıkça koninin genişliği azalır, dolayısıyla, insan görüşüne uygun olarak, düşük aydınlıkta algılanabilen farklı doygunluk düzeyleri de azalır. Diğer yandan, silindir biçimi ile sıfır aydınlık düzeyinde bile yüksek doygunluk düzeyleri tanımlanabilir ve böylece geçersiz renkler elde edilebilir. Dolayısıyla görüntü işleme uygulamalarında konik biçimi tercih edilirken, renk seçimi görevlerinde silindir biçimi kullanılma eğilimini gösterir.



Şekil 2.2: (a) HSV renk uzayının konik gösterimi. (b) HSV renk uzayının silindirik biçimli gösterimi. [11]

RGB renk uzayından HSV renk uzayına dönüşümde, $r, g, b \in [0,1]$, $\max = \{ r, g, b \}$ kümesinin en büyüğü, $\min = \{ r, g, b \}$ kümesinin en küçüğü ve $h \in [0, 360]$ olarak kabul edilir. $\{ h, s, v \}$ değerleri aşağıdaki eşitlikler kullanılarak hesaplanır.

$$h = \begin{cases} 0 & \text{if } \max = \min \\ (60^\circ \times \frac{g-b}{\max - \min} + 360^\circ) \bmod 360^\circ, & \text{if } \max = r \\ 60^\circ \times \frac{b-r}{\max - \min} + 120^\circ, & \text{if } \max = g \\ 60^\circ \times \frac{r-g}{\max - \min} + 240^\circ, & \text{if } \max = b \end{cases} \quad (2.1)$$

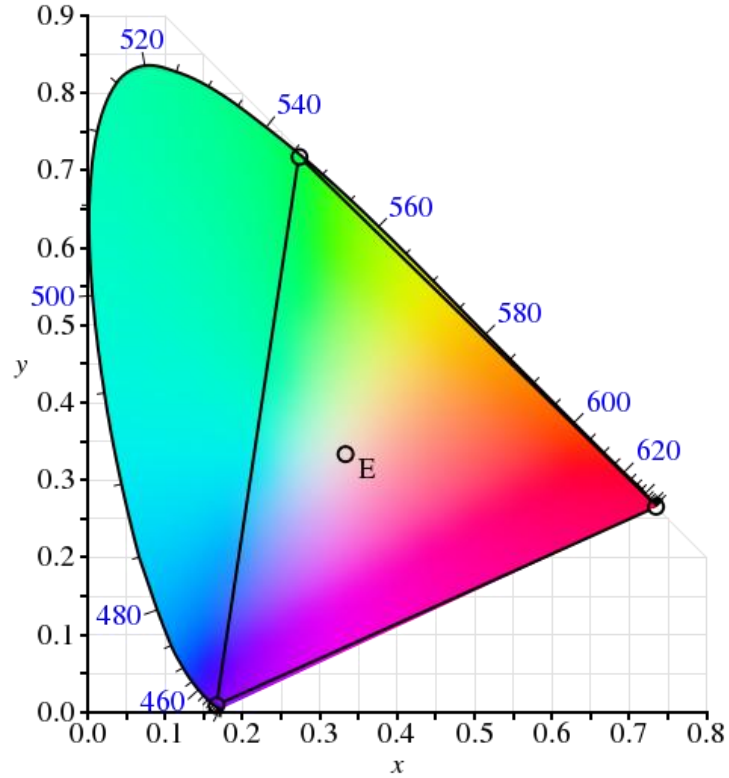
$$s = \begin{cases} 0, & \text{if } \max = 0 \\ \frac{\max - \min}{\max} = 1 - \frac{\min}{\max}, & \text{otherwise} \end{cases} \quad (2.2)$$

$$v = \max \quad (2.3)$$

2.1.1.3. CIE XYZ Renk Uzayı

Renk algılama çalışmalarında matematiksel olarak yapılan ilk tanımlama CIE (*International Commission on Illumination*) tarafından tanımlanan “CIE 1931 color space” olarak da bilinen CIE XYZ renk modelidir. CIE XYZ (1931), tüm renk ölçüm işlemlerinin temelini oluşturur. Bu sistemde renkler Şekil 2.3'teki gibi pozitif

değerler ve Y harfi ile temsil edilen parlaklık (*luminance*) değerleriyle tanımlanabilir.



Şekil 2.3: RGB renkleri kullanılarak görüntülenebilen CIE renk gamutu [12]

Sonuç olarak, X-Y-Z değerleri görülebilir değerler değildir. Bununla birlikte renksellik (*chromaticity*) şeması genellikle doğrusal bir değere sahip değildir çünkü iki parlaklık değeri arasındaki birim vektörün değeri, insan gözü ile daima görünebilir bir renk değerine sahip olmayabilir. Bu sistemde renk Yxy olarak tanımlanır ve adlandırılır. Üçüncü koordinat olan z, tanımlanabilir fakat gerekli değildir. Ayrıca, z değeri tanımlanmak istenirse Eş. 2.4, Eş. 2.5, Eş. 2.6 eşitlikleri kullanılabilir.

$$x = \frac{X}{X + Y + Z} \quad (2.4)$$

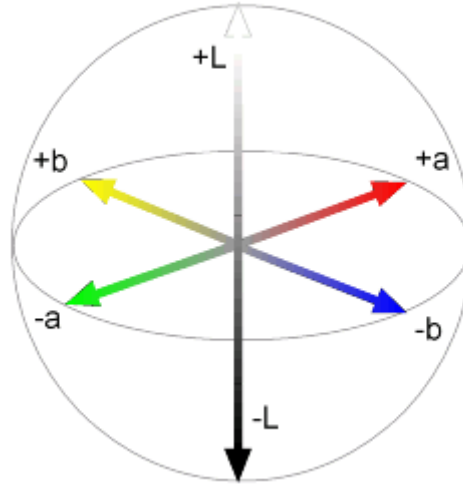
$$y = \frac{Y}{X + Y + Z} \quad (2.5)$$

$$z = \frac{Z}{X + Y + Z} = 1 - x - y \quad (2.6)$$

İnsan gözü kırmızı, yeşil, mavi olarak bilinen kısa, orta ve uzun dalga boylarına duyarlı hücrelere sahiptir. Bu renk hissini tanımlanabilmesi için üç parametrenin yeterli olması anlamına gelmektedir. XYZ renk uzayı doğrudan insan gözünün ölçümünü esas aldığı için diğer renk uzaylarından farklıdır ve diğer renk uzaylarına da temel oluşturur. RGB renk uzayından XYZ renk uzayına dönüşüm için öncelikle sRGB ya da *Adobe* RGB gibi aygıtla özgü belirlenen mutlak renk uzayına dönüşüm yapılmalıdır. Bu ayarlama aygıt bağımlıdır. Fakat dönüşüm sonucu elde edilen veri aygıttan bağımsızdır.

2.1.1.4. CIE L*a*b Renk Uzayı

CIE'nin asıl amacı, boya, mürekkep gibi malzeme üreticileri için renk iletişim standartlarının tekrarlanabilir bir sistemini oluşturmaktır. Renk eşleşmesi için evrensel bir şablon sağlamak bu standartların en önemli fonksiyonudur. Bu şablonun kaynağı ise standart gözlemci ve XYZ renk uzayı olarak belirlenir; ancak, XYZ uzayının ölçümsüz doğası bu standartların kolayca elde edilmesini zorlaştırmıştır [13].



Şekil 2.4: CIE L*a*b renk uzayı modeli.

L*a*b* renk modeli Şekil 2.4'te gösterildiği gibi dikey sarı-mavi ve yeşil-kırmızı eksenlerine dayanan dörtgensel koordinatlar kullanır. Sonuç olarak CIE, CIE L*a*b* ve CIE L*u*v* olarak adlandırılan daha düzenli renk ölçütleri geliştirmiştir. L*a*b* renk uzayının iyi dengelenmiş yapısı, bir rengin aynı zamanda hem yeşil hem kırmızı veya hem mavi hem sarı olamayacağı kuramı üzerine kurulmuştur.

Bunun sonucunda, kırmızı/yeşil ve sarı/mavi sıfatlarını tarif etmek için basit değerler kullanılabilir. CIE $L^*a^*b^*$ 'da bir rengi gösterirken, L^* aydınlığı (*lightness*), a^* kırmızı/yeşil değerini ve b^* sarı/mavi değerini gösterir.

Günümüzde en çok kullanılan ve temel alınan renk evreni CIE $L^*a^*b^*$ evrendir. Masaüstü yayıncılıkta bilgisayarlar ve programlar $L^*a^*b^*$ sistemini temel alırlar.

2.2. İkili Görüntü (*Binary Image*) ve Morfolojik (*Morphological*) İşlemler

Aynı zamanda *bi-level* veya *two-level* olarak da adlandırılan ikili görüntü, her piksel için sadece iki olası değer bulunduğuy sayısal görüntüdür. Genelde ikili resimler için kullanılan değerler 1 ve 0'dır. 1 değerine sahip olan pikseller ön plandaki (*foreground*) nesnelere belirtirken değeri 0 olan pikseller ise arka plana (*background*) ait noktaları belirtmektedir. İkili görüntüler, görüntü işlemede maskeler olarak kullanılmaktadır. Bunun dışında bölütleme ve iyileştirme işlemleri sırasında da kullanılmaktadırlar. İkili görüntü işleme teknikleri, yazılı belgelerin analizi, endüstriyel üretim ve tıbbi görüntü işleme alanlarında da yaygın olarak kullanılmaktadır.

Morfolojik görüntü işleme, matematiksel morfolojinin görüntü işlemeye uyarlanmasıdır. İlk olarak 1964 yılında Georges Mathheron ve Jean Serra [14] tarafından ortaya atılmıştır. Geometrik şekillerin analizi ve işlenmesi için geliştirilmiş bir kuram ve tekniktir. Küme teorisi, latis teorisi ve topolojiye dayanmaktadır. İlk başta, ikili görüntüler için tasarlanmışsa da sonradan gri seviye görüntüler için de genişletilmiştir. Morfolojik görüntü işlemede kullanılan temel işlemler aşındırma, genişletme, açma ve kapama işlemleridir.

Aşındırma ve genişletme işlemleri, morfolojik görüntü işlemede kullanılan iki temel işlemdir. Diğer morfolojik işlemlerin birçoğuy, bu iki temel işlem kullanılarak gerçekleştirilmektedir. Bu iki işlemden aşındırma işlemi ikili görüntünün inceltilmesi için kullanılırken, genişletme işlemi ikili görüntüyü kalınlaştırma amacıyla uygulanır.

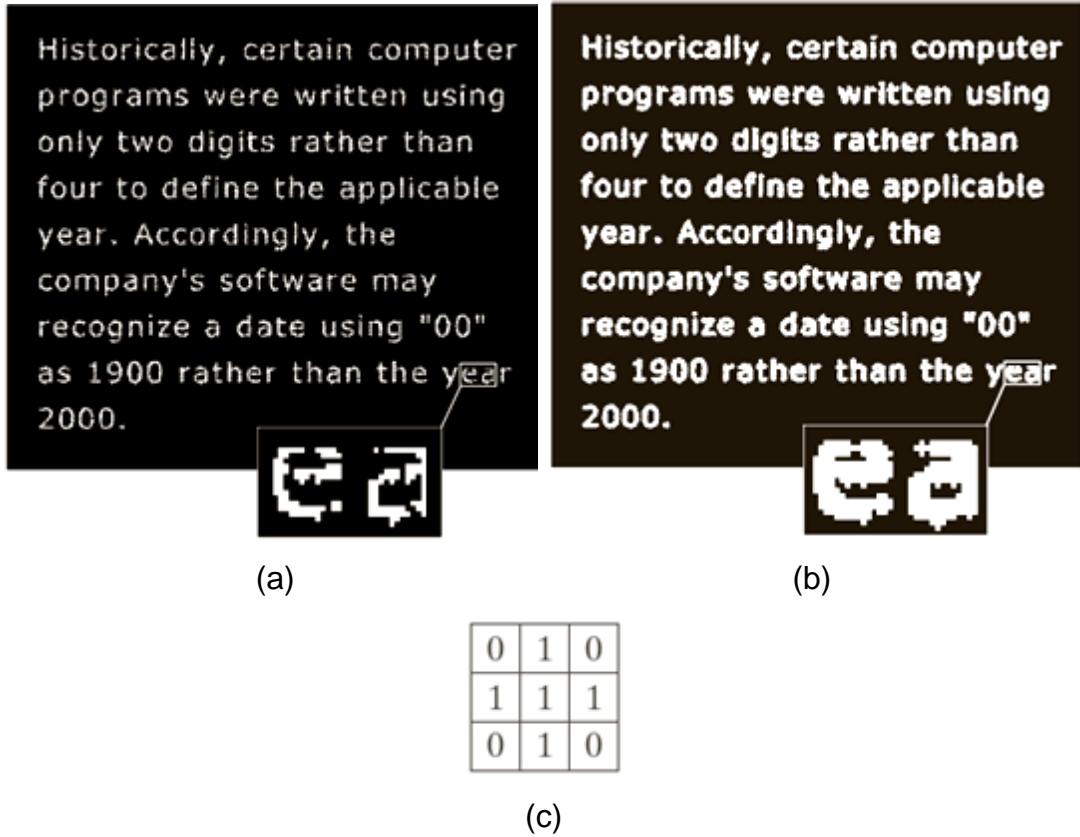
2.2.1. Genişletme (*Dilation*)

Genişletme işlemi, ikili bir görüntü üzerinde yer alan nesnelere genişletilmesi ve kalınlaştırılması işlemidir [10]. Genişletme işlemi ile ikili görüntü üzerinde yer alan

nesnelerin boyutlarının büyütülmesi ve bu nesnelere içerisinde yer alan deliklerin kapatılması sağlanabilir. Genişletme işlemi sırasında gerçekleştirilen kalınlaşma ve daralma işlemleri, bir yapısal eleman (*structuring element*) tarafından denetlenmektedir. Bu işlemi gerçekleştirirken kullanılan yapısal elemanlar ise bir ve sıfırlardan oluşan matrislerdir.

İkili genişletme işlemi gerçekleştirilirken bir yapısal eleman ikili görüntü üzerinde gezdirilir. Eğer yapısal elemanın merkezi, ikili görüntü üzerinde değeri 1 olan bir nokta üzerindeyse yapısal eleman ve ikili görüntü mantıksal VEYA işlemine tabi tutulur. Bu işlem yapısal elemanın merkezinin üzerinden geçtiği ve değeri 1 olan tüm noktalar için tekrarlanır. Yapısal elemanın merkezi şeklin ağırlık merkezi olabileceği gibi isteğe göre belirlenmiş bir değer de olabilir. Genişletme işleminin matematiksel ifadesi Eşitlik 2.7'de belirtilmiştir. Eşitlikte B yapısal elemanın yansımasını, $(B)_z$ ise yansımanın görüntü üzerindeki ötelenmesini belirtmektedir.

$$A \oplus B = \{z | (B)_z \cap A \neq \emptyset\} \quad (2.7)$$

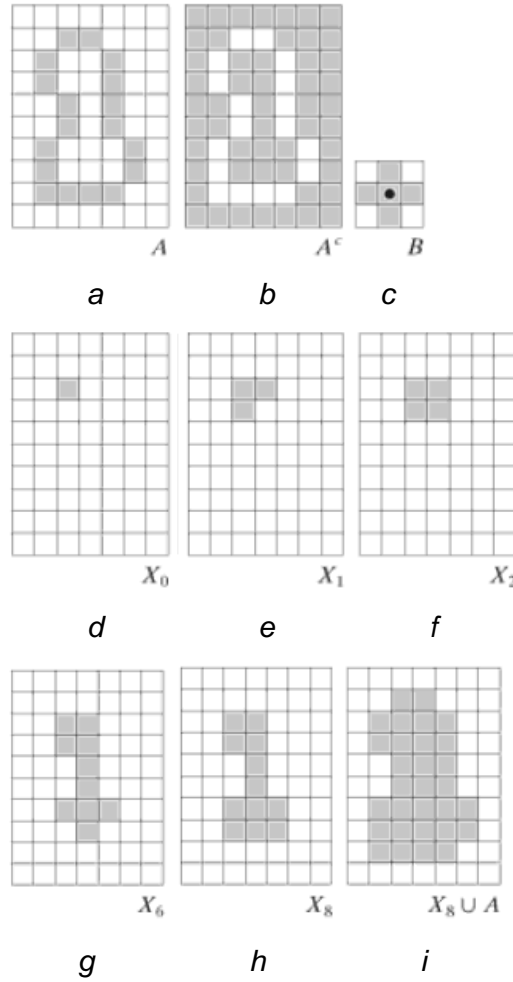


Şekil 2.5: (a) İkili görüntü, (b) Genişletme işlemi sonucunda elde edilen görüntü, (c) Genişletme işleminde kullanılan yapısal eleman [10]

Yukarıda belirtilen matematiksel gösterimde A ikili görüntüyü B ise yapısal elemanı belirtmektedir. Şekil 2.5a'da ikili bir görüntü gösterilmektedir. Şekil 2.5c'de ise örnek bir yapısal eleman gösterilmektedir. Şekil 2.5b'de ise ikili görüntü üzerine genişletme işlemi yapıldıktan sonraki hali yer almaktadır.

2.2.2. Boşluk Doldurma (*Hole Filling*)

Görüntülerde boşluk, arka plan bölgesinin, birbiriyle bağlantılı ön plan pikselleri tarafından etrafının sarıldığı kesimler şeklinde tanımlanabilir. Görüntülerdeki bu boşlukları doldurmaya yönelik kullanılan algoritma ise genişletme, tümlleme ve kesişim işlemlerini temel alır.



Şekil 2.6: (a) A kümesi, (b) A 'nın tümleyeni, (c) B yapısal elemanı, (d) Sınır içindeki başlangıç noktası, (e)-(h) Eşitlik 2.8'in çeşitli adımları, (i) Varılan sonuç (a) ve (h)'nin birleşimi [10]

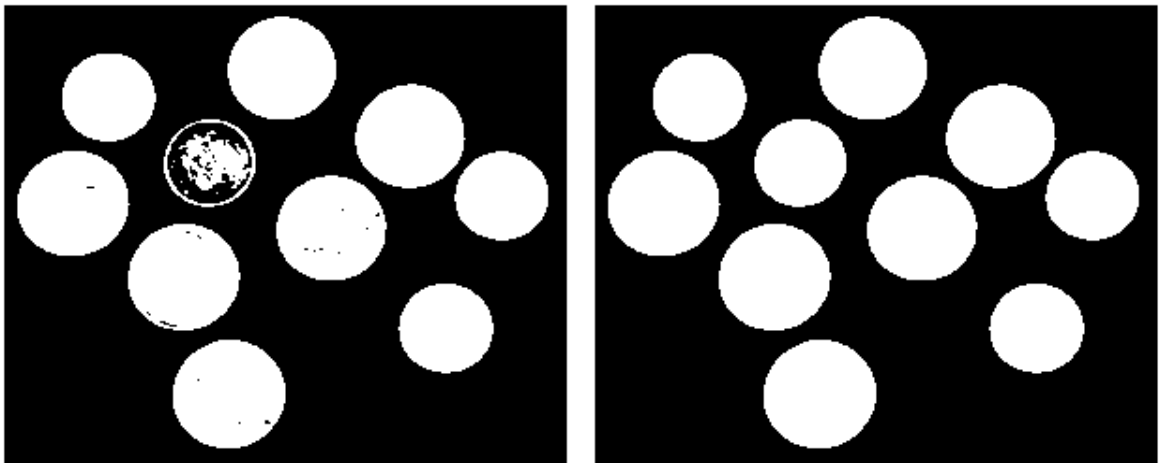
Elemanları 8-bağlantılı sınırlardan oluşan bir A kümesi olduğunu ve bu sınırların arka plan bölgesini çevrelediğini varsayalım. Elde edilen kısım boşluk olarak adlandırılır. Her bir boşluk içinde bir nokta verildiğinde, amacımız bütün boşlukları 1 ile doldurmaktır.

Öncelikle 0'lardan oluşan bir dizi oluşturulur (A'yı kapsayan dizi ile aynı boyutta), ancak X_0 'da verilen noktaya karşılık gelen konumdaki her boşluğa 1 atanır. Sonra sıradaki eşitlikte verilen matematiksel işlem tekrarlanarak ile tüm boşluklar 1 ile doldurulur.

$$X_k = (X_{k-1} \oplus B) \cap A^c \quad k=1,2,3,\dots \quad (2.8)$$

B, simetrik yapısal elemandır (Şekil 2.6c). Algoritma, k yineleme sonra, $X_k = X_{k-1}$ olduğu durumda sonlandırılır. X_k kümesi, tüm doldurulmuş boşlukları kapsar. X_k ve A kümesinin birleşimi tüm doldurulmuş boşlukları ve onların sınırlarını kapsar.

Eşitlik 2.8'deki genişletme işlemi, işaretlenmemiş olarak bırakıldığında, tüm alanı doldurur. Ancak her adımdaki A_c ile kesişim işlemi sayesinde, elde edilen sonuç, ilgi alanının içi ile sınırlandırılır. Şekil 2.6, Eşitlik 2.8'in ileri adımlarını göstermektedir. Şekildeki örnekte sadece 1 boşluk olmasına rağmen bu kavram rahatlıkla sonlu sayıda boşluğa uygulanabilir. Şekil 2.7'de ise boşluk doldurma işlemi uygulanmış görüntü verilmiştir.



Şekil 2.7: (a) İkili görüntü, (b) Genişletme işlemi sonucunda elde edilen görüntü,

2.3. Şablon Eşleştirme (*Template Matching*)

İki görüntüyü karşılaştırırken kullanılan en basit ve hızlı yöntemlerden biri normalleştirilmiş korelasyon ölçüsüdür (*Normalized Correlation – NC*) [15]. Bu yöntem temel olarak, gri tonlamalı iki görüntünün üst üste çakıştırılıp, piksel tabanlı karşılaştırılması esasına dayanır. Bu işlem sonucunda elde edilen değere korelasyon ölçüsü adı verilir. A ve B olarak tanımlanan iki görüntü olduğunu varsayalım. Bunların korelasyon ölçüsü Eşitlik 2.9'daki formül kullanılarak hesaplanır.

$$K = \frac{\sum_x \sum_y ((A[x, y] - \bar{A})(B[x, y] - \bar{B}))}{\sqrt{\sum_x \sum_y (A[x, y] - \bar{A})^2} \sqrt{\sum_x \sum_y (B[x, y] - \bar{B})^2}} \quad (2.9)$$

Korelasyon işlemi aktarmaya, dönmeye ve ölçeğe duyarlıdır. Diğer bir deyişle ilgin (*affine*) ve bakış açısı (*perspective*) dönüşümlerine duyarlıdır.

2.3.1. Karşıtlık Esnetme

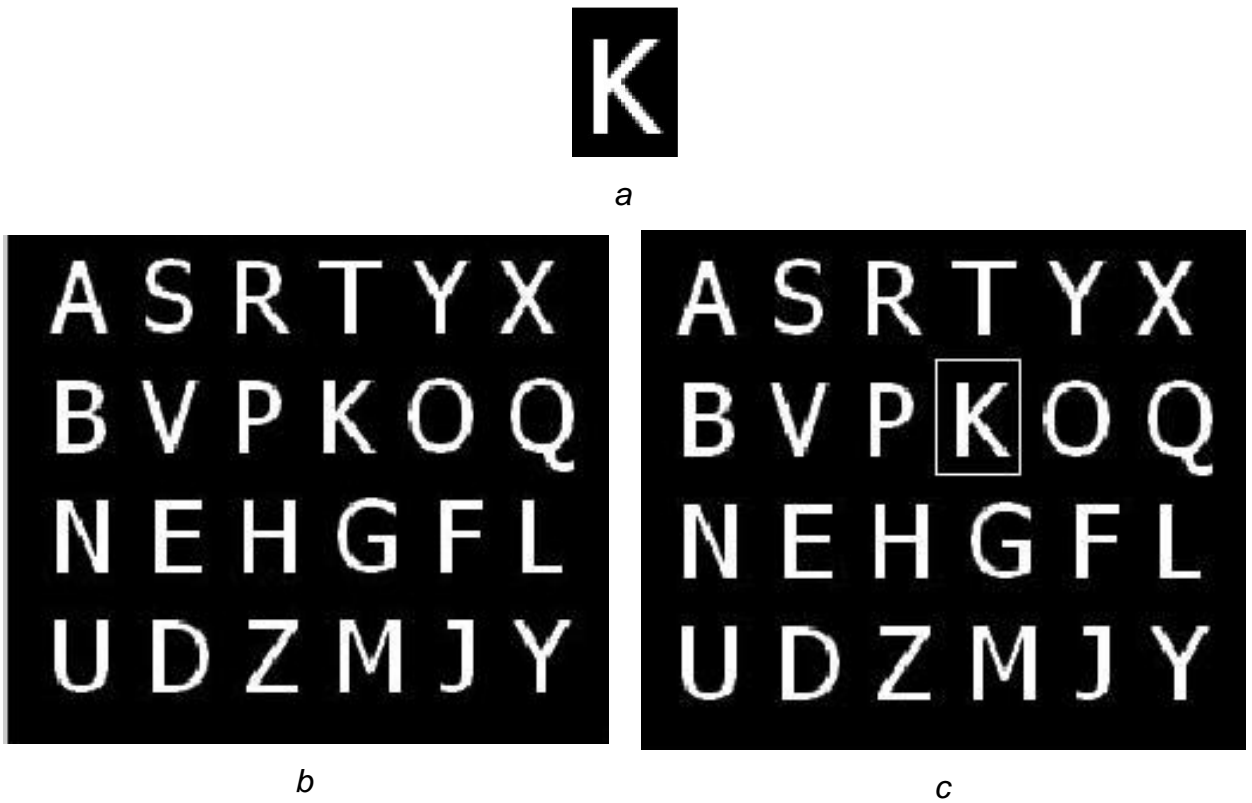
Daha başarılı sonuç elde edebilmek adına, görüntülerin normalleştirildikten sonra korelasyon ölçülerinin hesaplanması gerekmektedir. Görüntü işlemede normalleştirme, piksel yoğunluk değerlerinin aralığının değiştirilmesi işlemidir. Fotoğraflardaki parlamalardan kaynaklı düşük karşıtlığı düzeltmek gibi uygulama alanları vardır. Karşıtlık esnetme (*contrast stretching*) olarak da adlandırılabilir.

Farklı uygulama alanlarında, devingen olarak karşıtlık esnetmenin amacı, görüntüyü ya da diğer türlerdeki sinyalleri, daha tanıdık ve normal hale getirmektir. Devingen aralıktaki bir veri kümesinin, sinyallerin veya görüntülerin tutarlılığını arttırmak ve kesikliği azaltmak amaçlanmıştır. Örneğin gazetelerde, tüm görüntülerin benzer gri tonlamalı aralığı paylaşması için çaba sarf edilir.

Normalleştirme doğrusal bir süreçtir. Örneğin görüntüdeki yoğunluğun 50 ile 180 arasında değiştiğini ve istenen aralığın 0 ile 255 olduğunu varsayalım. Bu durumda her piksel değerinden 50 çıkartılıp aralık 0-130 haline getirilir. Sonra 255/130 ile çarpımı ile aralık 0-255'e çekilmiş olur. Görüntü işleme yazılımlarında otomatik normalleştirme, görüntünün dosya biçiminde tanımlanan sayı sisteminin tam devingen aralığa normalleştirilmesi ile sağlanır.

2.3.2. Çapraz Korelasyon (*Cross-Correlation*)

Çapraz korelasyon [16], boyut olarak küçük bir görüntünün (şablon), boyut olarak daha büyük bir görüntü (hedef) içinde tüm x,y koordinat değerleri için korelasyon ölçüsünün hesaplanması ve en yüksek sonucu veren alanın tespit edilmesi işlemidir. Böylece şablon görüntü, hedef görüntünün içinde bulunmuş olur (Şekil 2.8). Görüntünün tamamında tarama yaptığı için hesaplama karmaşıklığı yüksektir.



Şekil 2.8: (a) Şablon görüntü. (b) Hedef görüntü. (c) Çapraz korelasyon kullanılmasıyla elde edilen eşleştirme sonucu.

3. NESNE TAKİBİ

Nesne takibi ele alınması zor bir problemdir. Nesnelerin takibinde oluşabilecek sıkıntılar; belirsiz nesne hareketleri, nesne ve olay yerinin görünüm örüntüsündeki değişiklikler, belirsiz nesne yapıları, nesne-nesne ya da nesne-olay yeri engellemeleri ve kamera hareketlerinden kaynaklanabilir [17]. Takip için, nesne tespitinin ve eşleştirilmesinin yapılması gereklidir. Tespit için nokta algılayıcılar (*point detectors*), arka plan çıkarımı (*background subtraction*), bölütleme (*segmentation*), denetimli öğrenme (*supervised learning*) kullanılabilir. Tez kapsamında, takip için nokta algılayıcılar kullanılacaktır. Bunun için görüntülerdeki ayırt edici nitelikler (*keypoints*) çıkarılacak, tanımlayıcıları (*descriptors*) oluşturulacak ve bunlar arasındaki benzerlikler karşılaştırılacaktır.

Kesim 3.1'de nitelik çıkarımı için kullanılan yöntem, kesim 3.2'de ise, çıkarılan niteliklerin eşleştirilmesinde kullanılan yöntem ve uzaklık örneklenecektir.

3.1. Nitelik Çıkarım Yöntemi

Nitelik çıkarımı ve tanımlayıcı oluşturma için SIFT (*scale invariant feature transform*) [18] tanımlayıcıları, Harris köşe algılayıcısı [19], KLT [20], GLOH [21] gibi yöntemler kullanılmaktadır. Harris köşe algılayıcıları, Moravec köşe belirleme yönteminin [22] geliştirilmesi sonucu elde edilmiştir. Moravec'in yöntemi dönüşten bağımsız değildir (*anisotropic*). Harris yönteminde bunun yerine kaydırma merkezi etrafında analitik genişletme yapılarak, olabilecek tüm küçük kaydırmalara imkân sağlanmıştır. Harris yönteminin en önemli dezavantajı, köşe belirleme için eşik değeri kullanması ve bu eşik değerinin ışıklandırma durumuna bağlı olarak değişiklik göstermesidir. KLT (Kanade-Lucas-Tomasi) nitelik takipçisinde ise çıkarılan niteliklerin sonraki sahnelerde takip edilebilir olmasına dikkat edilir. Nitelik, takip eden sahnelerde kaybolduğunda, yöntemin başarımı düşmektedir. Bu yüzden çıkarılan nitelikler, eşleştirme amaçlı kullanılmamaktadır.

SIFT yöntemi kullanılarak çıkarılan niteliklerin, görüntüdeki döngüsel ve boyutsal farklılıklardan, parlaklık değişimlerinden ve gürültüden etkilenmemesinden dolayı tez kapsamında yapılan çalışmalarda, SIFT tanımlayıcıları kullanılmıştır. Bu başlığın alt kesimlerinde SIFT yöntemi detaylı olarak açıklanmıştır.

3.1.1. SIFT Yöntemi

Görüntülerin eşleştirilmesi, bilgisayarla görmede en temel sorunların başında gelmektedir. Lowe [18]'un ortaya koyduğu bu algoritma nesne tanıma, üç boyutlu modelleme, panoramik resimler oluşturma, görüntü çakıştırma, video ile nesne takibi, görüntü edinme uygulamalarda kullanılabilir.

Algoritmanın adımları aşağıdaki gibi özetlenebilir:

- 1- **Ölçek Uzayında En Uç Değer Tespiti (*Scale-Space Extrema Detection*):** Hesaplamanın ilk aşaması bütün ölçekler ve görüntü konumları için yapılır. Gauss'lar farkı (difference-of-Gaussian, DoG) kullanılarak, dönüş ve ölçekten bağımsız ilgi noktalarının tespiti etkin bir şekilde gerçekleştirilir.
- 2- **Anahtar Nokta Konumlandırma (*Keypoint Localization*):** Bütün aday noktalara, ölçek ve konumlarını belirleyecek ayrıntılı bir model oluşturulur. Anahtar noktalar, kararlılıklarının ölçümlerine dayanarak seçilir.
- 3- **Yönelim Atama (*Orientation Assignment*):** Her anahtar noktaya, yerel görüntü eğim yönlerine göre bir ya da daha fazla yönelim atanır. Daha sonra uygulanacak tüm işlemler; niteliklere atanmış yönelim, ölçek ve konum değerlerine bağlı olarak dönüştürülmüş görüntü verisi kullanılarak yapılacaktır.
- 4- **Anahtar Nokta Tanımlayıcısı (*Keypoint Descriptor*):** Her anahtar noktanın etrafındaki alandan, seçilen ölçekte, yerel görüntü eğimleri ölçülür. Bu değerler, yerel şekil bozulmalarına ve aydınlanmada değişikliğe karşı büyük oranda duyarsız sunumlara (*representation*) dönüştürülür.

3.1.1.1. Ölçek Uzayında En Uç Değer Tespiti

Bu adım, ilgi noktalarının belirlendiği kesimdir. Bunun için görüntüye farklı ölçeklerde Gauss süzgeci uygulanır ve Gauss ile bulanıklaştırılmış görüntülerin farkları alınır. Anahtar noktalar, çoklu ölçeklerde ortaya çıkan en fazla ve en az değere sahip noktalar olarak belirlenir.

Girdi görüntüsü $I(x,y)$, değişken ölçekli Gauss filtresi $G(x,y,\sigma)$ olarak tanımlanırsa (Eş. 3.2) ve Gauss uygulaması sonrası oluşan ölçek uzayındaki resmi işlev olarak $L(x,y,\sigma)$ ile ifade edilebilir (Eş. 3.1).

$$L(x, y, \sigma) = I(x, y) * G(x, y, \sigma) \quad (3.1)$$

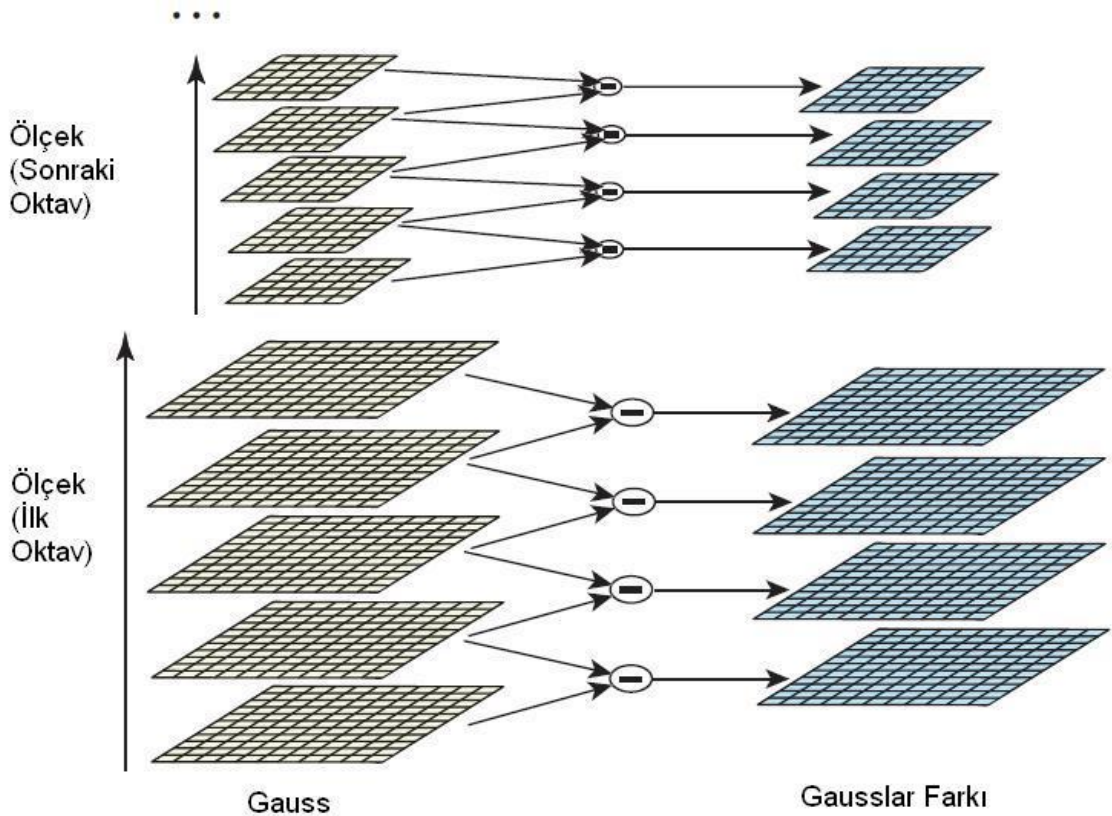
'*' işleci, x ve y'de evrişim (*convolution*) işlemini belirtir.

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2} \quad (3.2)$$

Tutarlı anahtar noktaları konumları, k sabit çarpım değeriyle ayrılmış ölçeklerin Gauss'lar farkı ile $D(x,y,\sigma)$, etkin bir şekilde hesaplanır (Eş. 3.3).

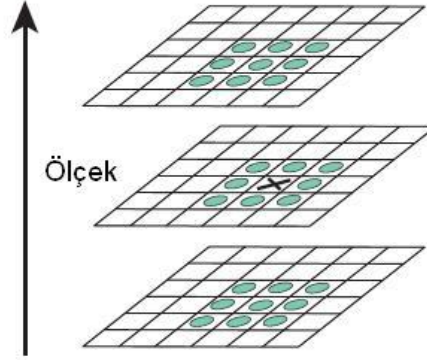
$$D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) \quad (3.3)$$

$$= L(x, y, k\sigma) - L(x, y, \sigma)$$



Şekil 3.1: DoG uygulama örneği [18].

Şekil 3.1’de DoG uygulaması örneklenmiştir. Ölçek uzayının her oktavi (*octave*) için, başlangıçtaki görüntü tekrarlı olarak Gauss ile evriştirilerek solda görünen ölçek uzayı grupları oluşturulur. Sıralı Gauss görüntüleri birbirlerinden çıkarılarak sağdaki DoG görüntüleri oluşturulur. Her oktavdan sonra, Gauss görüntüsü yarısına örneklenerek işlemler tekrarlanır.



Şekil 3.2: Aday anahtar noktaların belirlenmesi için piksellerin komşuları ile karşılaştırılması [18].

DoG görüntüleri elde edildikten sonra, anahtar noktalar yerel en az/en fazla değerler alınarak tanımlanır. Bu işlem, DoG görüntülerindeki her pikselin sekiz komşuluğu ve komşu ölçeklerdeki dokuz komşulukları değerlendirilerek yapılır. Eğer piksel değeri, karşılaştırıldığı piksellerin değerlerinden en büyüğü ya da en küçüğü ise aday anahtar nokta olarak seçilir. Karşılaştırma şekil 3.2’de gösterilmiştir.

Aday anahtar nokta sayısı arttıkça işlem başarımı da azalmaktadır. Ayrıca çok sayıda bulunan nokta sanılanın aksine, resim bozulmaya uğradığında kaybolma ihtimalleri artacağından, güvenilirliğin düşmesine sebep olmaktadır. Özetle, bu işlemlerin sonucunda, DoG işlevleri uygulandığında fazla miktarda anahtar nokta adayı belirlenmekte ve hepsinin hesaplanması maliyetli bir iş olmaktadır. Bu yüzden en güvenilir ve kullanışlı alt kümeyi belirleyerek daha tutarlı sonuçlar elde edilebilir.

3.1.1.2. Anahtar Nokta Konumlandırma

Anahtar nokta adayları, komşu pikseller ile karşılaştırılıp seçildiğinde sonraki adım; konum, ölçek ve başçıl eğriliğin (*principal curvature*) oranları ile yakınındaki veriler ile ayrıntılı uyumunu sağlamaktır. Bu bilgi, düşük karşıtlık değerine sahip (gürültüye karşı duyarlı) ve kenarlarda kötü bir şekilde konumlanmış olan noktaların elenmesinde kullanılacaktır.

Bu yaklaşımın ilk gerçekleştiriminde Lowe [18], basitçe anahtar noktalarını merkezi örneklem noktasının konum ve ölçeğine göre konumlandırmıştır. Daha sonra Brown ve Lowe [23] farklı bir yöntem geliştirmişlerdir. Yeni yöntemde, en büyük değer aradeğerlenmiş konumu hesaplanmaktadır. Bu aradeğerleme, DoG ölçek uzayının, $D(x,y,\sigma)$, karesel Taylor açılımı kullanılarak yapılmıştır (Eş. 3.4).

$$D(x) = D + \frac{\partial D^T}{\partial x} x + \frac{1}{2} x^T \frac{\partial^2 D}{\partial x^2} x \quad (3.4)$$

D ve türevleri örneklem noktasında değerlendirilir ve $x = (x,y,\sigma)^T$ bu noktadan uzaklığın değeridir. En uç değer konumu, \hat{x} , bu işlevin x 'e göre türevinin sıfıra eşitlenmesi ile bulunur (Eş. 3.5).

$$\hat{x} = - \frac{\partial^2 D^{-1}}{\partial x^2} \frac{\partial D}{\partial x} \quad (3.5)$$

Eğer \hat{x} uzaklığı bütün boyutlarda 0.5'ten büyükse, en uç değer diğer bir aday noktaya yakın olduğunu gösterir. Bu durumda, aday nokta değiştirilir ve ara değerlendirme bu noktaya uygulanır. Aksi durumda uzaklık, en uç değer konumunun aradeğerleme tahmini için aday noktasına eklenir.

Düşük zıtlıklı anahtar noktaları elemek için ikinci dereceden Taylor açılımı, $D(x)$, \hat{x} uzaklığında hesaplanır. Eğer bu değer 0.03'den daha azsa, aday nokta elenir. Diğer koşulda, son konumu $y + \hat{x}$ ve ölçeği σ olarak nokta saklanır. (y , σ ölçeğinde noktanın asıl konumudur)

Tutarlılık için sadece düşük zıtlık değeri olan noktaları elemek yeterli değildir. Kenarlar yeterince belirli olmasa da, DoG işlevi buna güçlü tepki vermektedir.

DoG işlevinde zayıf tanımlanmış tepe noktaları için kenarın yatayındaki başçıl eğrilik, üzerindeki daha geniş olacaktır. Bu başçıl eğriliklerin niceliklerini

bulmak için ikinci dereceden Hessian matrisinin, H, özdeğerlerini hesaplanması gerekmektedir (Eş. 3.6).

$$H = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix} \quad (3.6)$$

H'nin özdeğerleri, D'nin başçıl eğrilikleri ile orantılıdır. α en büyük, β en küçük genlikli öz değer olarak düşünüldüğünde; H'nin izi ile bunların toplamları (Eşitlik 3.7), H'nin determinanı ile de çarpımları bulunur (Eşitlik 3.8).

$$Tr(H) = D_{xx} + D_{yy} = \alpha + \beta \quad (3.7)$$

$$Det(H) = D_{xx}D_{yy} - (D_{xy})^2 = \alpha\beta \quad (3.8)$$

r, en büyük ve en küçük genliğin oranıdır. $r = \frac{\alpha}{\beta}$ olduğundan, α yerine $r\beta$ yazılabilir (Eş. 3.9).

$$\frac{Tr(H)^2}{Det(H)} = \frac{(\alpha+\beta)^2}{\alpha\beta} = \frac{(r\beta+\beta)^2}{r\beta^2} = \frac{(r+1)^2}{r} \quad (3.9)$$

$\frac{(r+1)^2}{r}$ değeri, iki özdeğer birbirine eşit olduklarında en küçüktür ve r değeri ile artmaktadır. Sonuçta, başçıl eğriliklerin oranının belli bir eşik değerinin (r) altında olup olmadığını denetlemek yeterlidir (Eşitlik 3.10).

$$\frac{Tr(H)^2}{Det(H)} < \frac{(r+1)^2}{r} \quad (3.10)$$

Bu yöntem etkin bir biçimde hesaplanabilir (Her bir anahtar noktayı denemek için 20'den az kayan noktalı işlem yapılmaktadır).

3.1.1.3. Yönelim Atama

Yerel görüntü özelliklerine göre her anahtar noktaya tutarlı bir yönelim atayıp, anahtar noktanın tanımlayıcılarını bu yönetime bağlı olarak belirleyerek, görüntünün döndürülmesine karşı büyük oranda dayanıklılık elde edilebilir.

İlk önce Gauss ile yumuşatılmış, σ ölçeğindeki görüntü $L(x,y,\sigma)$ seçilir. Böylece bütün hesaplamalar ölçekten bağımsız bir mantıkla yapılır. σ ölçeğindeki görüntü

örneklemi $L(x,y)$, için eğim genliği $m(x,y)$, ve yönelimi $\theta(x,y)$, piksel farkları kullanılarak önceden hesaplanır:

$$m(x, y) = \sqrt{(L(x + 1, y) - L(x - 1, y))^2 + (L(x, y + 1) - L(x, y - 1))^2} \quad (3.11)$$

$$\theta(x, y) = \tan^{-1}((L(x, y + 1) - L(x, y - 1))/(L(x + 1, y) - L(x - 1, y))) \quad (3.12)$$

Eğim için genlik ve yön hesaplamaları, Gauss ile yumuşatılmış görüntüde anahtar noktasına komşu olan alanlar içindeki her piksel için hesaplanır. Her biri on dereceyi kapsayan, 36 bölümlük bir yönelim histogramı oluşturulur.

Histograma eklenen her örneklem, eğim genliği ve Gauss-ağırlıklı dairesel bir pencere ile (anahtar noktasının 1.5 katı σ değerine sahip) ağırlıklandırılır. Histogramdaki tepe değerler baskın yönelimlerdir. Histogram tamamlandığında en tepe nokta ve ona %80 oranında yaklaşan diğer tepe noktalar, anahtar noktasına atanır. Çoklu yönelim atamaları sonucunda, her bir yönelim için, aynı konumda ve ölçekte yeni bir anahtar nokta yaratılır.

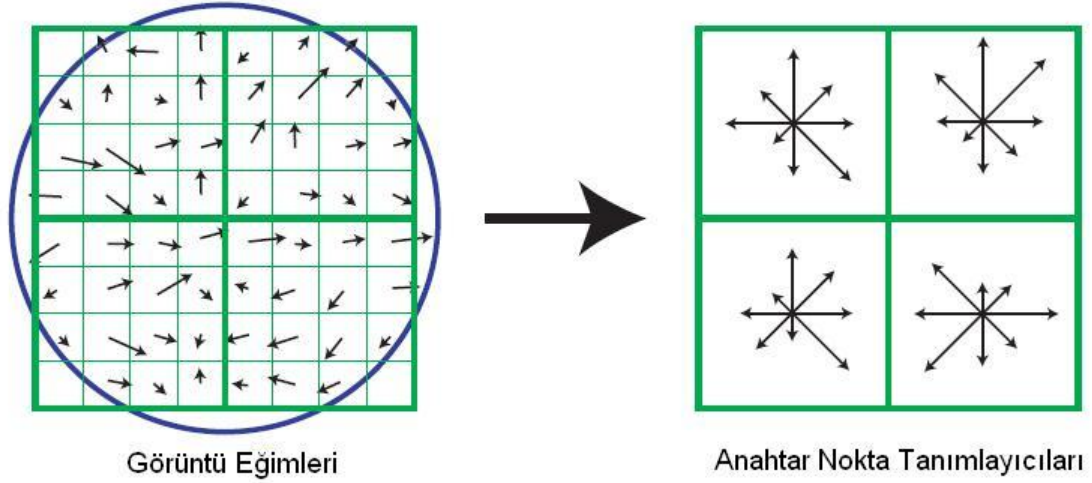
3.1.1.4. Anahtar Nokta Tanımlayıcısı

Önceki işlemlerde, her anahtar noktasına bir görüntü konumu, ölçeği ve yönelimi atanmıştır. Bu adımlar sonrasında görüntünün konum, ölçek ve dönüşlere karşı dayanıklı olmasını sağlanmıştır. Sonraki adımda ise yerel görüntü alanları için yüksek oranda ayırt edici, aynı zamanda aydınlanma ve üç boyutlu bakış açısındaki değişimlere karşı dayanıklı tanımlayıcılar hesaplanacaktır.

Bu adım, yönelim atama adımına benzemektedir. Nitelik tanımlayıcısı (4x4) komşuluklardaki yönelim histogramlarının kümelerinden hesaplanır. Yönelim histogramları, anahtar noktalarının yönelimlerine ve anahtar noktasının en yakın ölçeğindeki Gauss görüntüsünden gelen yönelim bilgisine bağlıdır. Her bir histogram sekiz bölüme sahiptir. Her tanımlayıcı, anahtar noktasının etrafından elde edilen 4x4'lük bir histogram dizisine sahiptir. SIFT nitelik vektörü (4 x 4 x 8) 128 elemandan oluşur. Bu vektör, anahtar noktasının aydınlanma değişikliklerine karşı dayanıklı olabilmesi için normalleştirilir.

Şekil 3.3'te; 8x8 örneklem kümesinden hesaplanmış 2x2 tanımlayıcı dizisi gösterilmektedir. Yukarıdaki tanımlanmış adımlar kapsamında 16x16 örneklem

kümesinden elde edilmiş 4x4 tanımlayıcı dizisi kullanılmıştır. Solda, her bir örneklem noktasının görüntü eğim genliği hesaplandıktan sonra, çember ile belirtilmiş Gauss penceresi ile ağırlıklandırılması; sağda ise, tanımlayıcı dizisinin elemanları olan yönelim histogramları gösterilmiştir.



Şekil 3.3: Anahtar nokta tanımlayıcılarının oluşturulması [18].

3.2. Nitelik Eşleştirme Yöntemleri

Nitelik eşleştirmeleri, matematikteki benzerlik (similarity) ilkesi temel alınarak yapılır. Nitelik vektörleri genellikle, aralarındaki uzaklıklar karşılaştırılarak eşleştirilirler. Vektörlerin boyutları, eşleştirmedeki hız başarımını doğrudan etkiler. Bu yüzden boyutların mümkün olduğunca azaltılması beklenir [24].

Eşleştirme stratejileri üç türlü ele alınabilir [25]: Eşik değer tabanlı eşleştirmede, iki nitelik arasındaki uzaklık belli bir eşik değerinin altında ise eşleşme gerçekleştirilir. Bu yaklaşımda bir niteliğin birden çok eşleşmesi olabilir. En yakın komşuluk (*nearest neighbour*) tabanlı eşleştirmede, bir niteliğin en yakın komşusu olan nitelik bir eşik değerinin altında ise eşleşme gerçekleştirilir. Bir diğer eşleştirme yaklaşımı ise, birinci ve ikinci komşuluk değerlerini kullanılarak gerçekleştirilir. Eşik değeri ve en yakın ikinci komşuluk değerinin çarpımı, birinci en yakın komşuluk değerinden daha fazla değilse eşleştirme gerçekleştirilir.

Üstte bahsedildiği üzere, iki farklı görüntüdeki iki anahtar noktanın aynı olup olmadığını tespit için, iki nitelik vektörü arasından Öklid uzaklığı ölçümü yapılabilir.

Bu uzaklık bir eşik değeri ile denetlenir. Eşik değerinin değiştirilmesi, yanlış doğru (*false positive*) ve yanlış hatalı (*false negative*) ölçümlerin yapılmasında etkilidir.

Yukarıda bahsedilen eşleştirme yaklaşımlarında kullanılacak uzaklık Kesim 3.2.1'de örneklenmiştir.

3.2.1. Öklid Uzaklığı

Öklid uzaklığı, iki nokta arasında cetvel ile hesaplanabilecek 'sıradan' uzaklıktır. Bu tanım, Pisagor teoreminin tekrarlı uygulaması ile ispatlanabilir [25].

$P = (p_1, p_2, \dots, p_n)$ ve $Q = (q_1, q_2, \dots, q_n)$ noktaları arasındaki Öklid uzaklığı;

$$\sqrt{\sum_{i=1}^n (p_i - q_i)^2} = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \dots + (p_n - q_n)^2} \quad (3.15)$$

Eş 3.15'deki gibi tanımlanabilir.

4. TRAFİK İŞARETLERİNİN TANINMASINA YÖNELİK YAPILMIŞ ÇALIŞMALAR

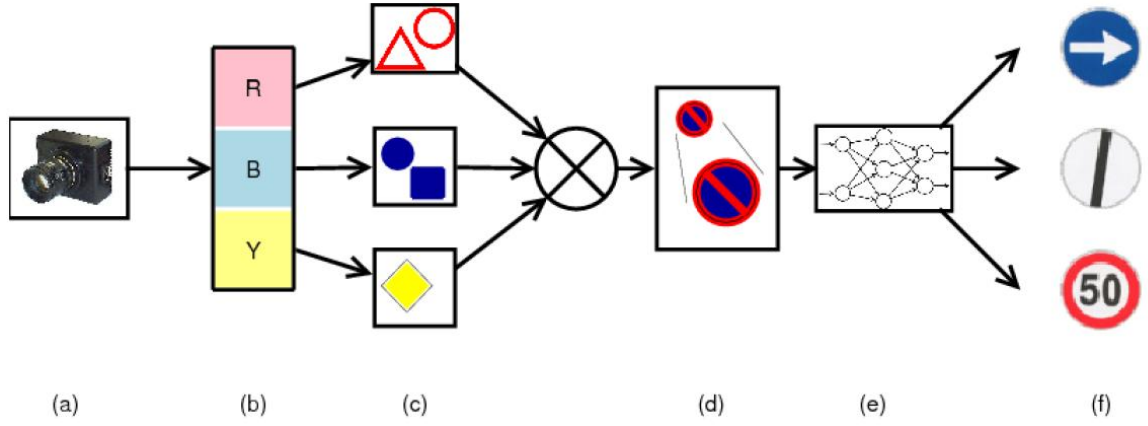
Miura [5] tarafından yapılan çalışmada sistem iki kameradan oluşmaktaydı. Kameralardan biri geniş-açılı görüntüdeki, renk, yoğunluk ve şekil bilgilerinden yararlanarak trafik işaretinin konumunu tespit etmekte, diğeri ise belirlenen konumdaki işaretin daha kaliteli görüntüsünü çekmekte idi. YUV renk uzayı kullanılarak işaretlerin rengine sahip nesnelere ortaya çıkarılmıştır. Sonrasında elde edilen görüntüden kenar bilgileri çıkartılarak işaretler tespit edilmiştir. İşareti sınıflandırma aşamasında ise normalleştirilmiş korelasyon tabanlı örüntü eşleştirme kullanılmıştır. İşaretleri tespit etmede %97,2'lik bir başarı elde edilmesine rağmen, eşleştirmede %46,5'lik bir başarı elde edilmiştir. Bir görüntüyü işleme süresi ise yaklaşık 500 ms.dir.

Torrensen [6] tarafından geliştirilen yöntemde sadece azami hız işaretlerinin tanınması amaçlanmıştır. RGB renk uzayının kullanıldığı çalışmada, görüntü renk-tabanlı süzgeçlerden geçirildikten sonra, 7 farklı boyuttaki dairenin sahne üzerinde gezdirilmesi ile trafik işaretinin konumu tespit edilmiştir. İşaretteki rakamlar çıkartılarak Şekil 4.1'de gösterilen 7x5 boyutunda ikili değerlerden oluşan matrislerle ifade edilmiştir. Sınıflandırma, geri yayılım algoritması ile eğitilen, ileri beslemeli sinir ağı kullanılarak yapılmıştır. Hız levhaları %87 başarıyla sınıflandırılmış, saniyede 8 görüntü işleme hızına ulaşılmıştır.



Şekil 4.1: Hız levhalarındaki rakamlardan elde edilen matrislere örnekler [6]

Trafik işaretinin sınıflandırılması adımı sinir ağlarının kullanımına sıklıkla rastlanmaktadır. Broggi [7] tarafından geliştirilen sistemde, RGB renk uzayı kullanılmış, farklı şekillerdeki (üçgen, ters üçgen, daire, eşkenar dörtgen, vb.) işaretler örüntü eşleştirme yöntemiyle tespit edilmiş, renk ve şekil bilgilerine göre belirlenen 7 farklı kategoride LWN++ [26] açık kaynak kodlu sinir ağı kullanılarak sınıflandırma gerçekleştirilmiştir. Geliştirilen sistemin akış çizelgesi Şekil 4.2'de gösterilmiştir.



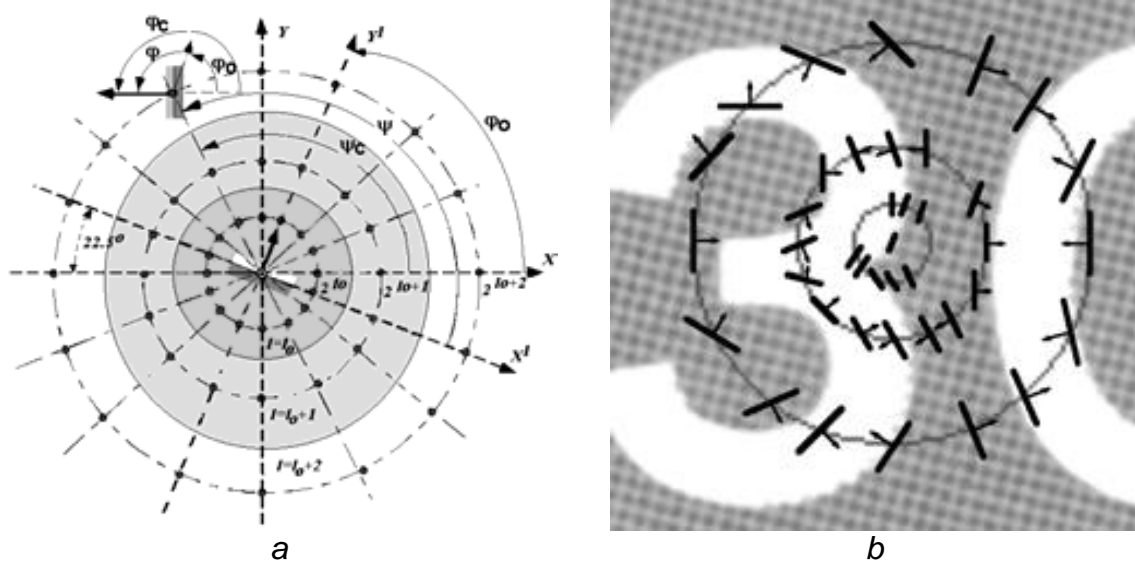
Şekil 4.2: (a) Görüntü elde edilmesi, (b) Renk bölütlemesi, (c) Şekil algılama, (d) Ölçekleme, (e) Sınıflandırma, (f) Çıktı [7].

Fang [8] tarafından geliştirilen yöntemde, HSI renk uzayı kullanılmış, işaretlerin renk ve şekil niteliklerinin çıkartılarak, sahne içindeki konumunun saptanması da, işaretlerin sınıflandırılması da iki farklı sinir ağı tarafından gerçekleştirilmiştir. Tespit edilen işaretlerin bir sonraki çerçevede hangi konumda ve boyutta olduğunu hesaplayarak, işleme süresini azaltmak amaçlanmıştır ve bu hesaplamada Kalman süzgeci [27] kullanılmıştır. Bu şekilde işleme süresi 300 ms.'ye indirilmiştir.

Escalera [9] tarafından geliştirilen yöntemde HSI renk uzayı kullanılmış, işaret renk bilgisi kullanılarak tespit edildikten sonra, sinir ağları kullanılarak %85'lik başarımla sınıflandırılmıştır. Sinir ağlarının kullanıldığı çalışmalarda, başarımın yüksek ancak işleme hızının düşük olduğu gözlemlenmektedir.

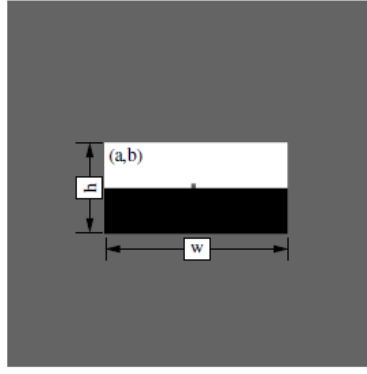
Gao [3] tarafından geliştirilen yöntemde, CIECAM97 [28] adı verilen renk görünüm modeli kullanılmıştır. Bu renk modeli CIE (International Commission on Illumination) tarafından geliştirilmiştir. Bu renk modelinde, renklerin, farklı ışıklandırma koşulları altında ve farklı arka planlarda, insanların algıladığı biçimde algılanması amaçlanmıştır. Görüntü önce RGB renk uzayından CIE XYZ uzayına dönüştürülür. Daha sonra kameranın bir dizi görüntü yardımıyla ayarlanması sonucu elde edilen veriler kullanılarak CIECAM97 modeline dönüştürülür. Bu modeldeki renksel parlaklık (*chroma*) ve renk tonu (*hue*) değerleri kullanılarak trafik işareti tespit edilir. Eşleştirme adımında ise davranışsal görünüm modeli (BMV – *Behavioural Model of Vision*) [29] kullanılmıştır. Bu davranış modelinde, görüntü üzerine oturtulmuş ilgi penceresi üzerinde, işaretin içerisindeki kenarların

bu pencere ile kesiştiği noktadaki yönelimler hesaplanmaktadır ve bunlar kullanılarak eşleştirme yapılmaktadır (Şekil 4.3). Bu çalışmada %89'luk bir başarımlar elde edilmiştir ve tanıma süresi 0,35 ile 0,6 saniye arasında değişmektedir. Daha sonraki çalışmalarında [4] BMV modelini trafik işaretleri üzerinde geliştirerek FOSTS (*foveal system for traffic signs*) adını verdikleri modeli kullanmışlardır. Bu çalışmalarında %95'lik başarımlar elde etmişlerdir ve tanıma süresi 0,2 ile 0,7 arasında değişmektedir.

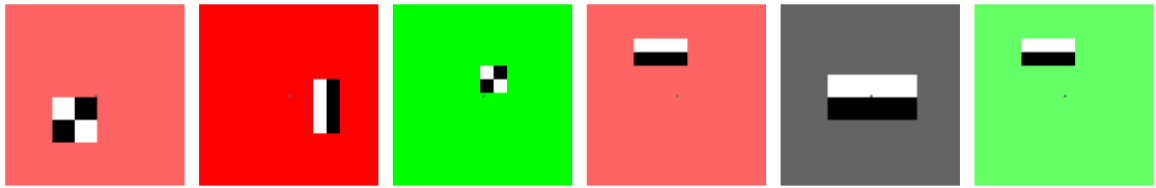


Şekil 4.3 (a) İlgili penceresinin şeması; farklı çözünürlükteki alanlar tonlamayla ayrılmıştır. İlgili penceresinin içerik düğümleri merkezden yayılan 16 doğru ile eş-merkezli 3 dairenin kesişim noktalarına yerleştirilmiştir. (b) İlgili penceresi ile tespit edilen yönelim elemanlarına örnek [29]

Bahlmann [30] tarafından gerçekleştirilen çalışmada, makine öğrenme algoritması olan *AdaBoost* kullanılarak Haar dalgacıkları (Şekil 4.4b) ile çıkarılan nitelikler yardımıyla işaretler tespit edilmiştir. Bu yöntemde renk ve şekil bilgileri birlikte kullanılmıştır. Fakat renk ve şekil bilgileri ile ilgili değişken belirleme gereksinimi ortadan kaldırılmıştır. Sistemin eğitimle bu bilgileri öğrenmesi sağlanmıştır. Sınıflandırma işlemi Gauss Olasılık dağılımının kullanıldığı Bayes üretken modeli ile gerçekleştirilmiştir. 23 trafik işaretinin tanınmaya çalışıldığı çalışmada, saniyede 10 çerçeve işlenerek %85 başarı yakalanmıştır.



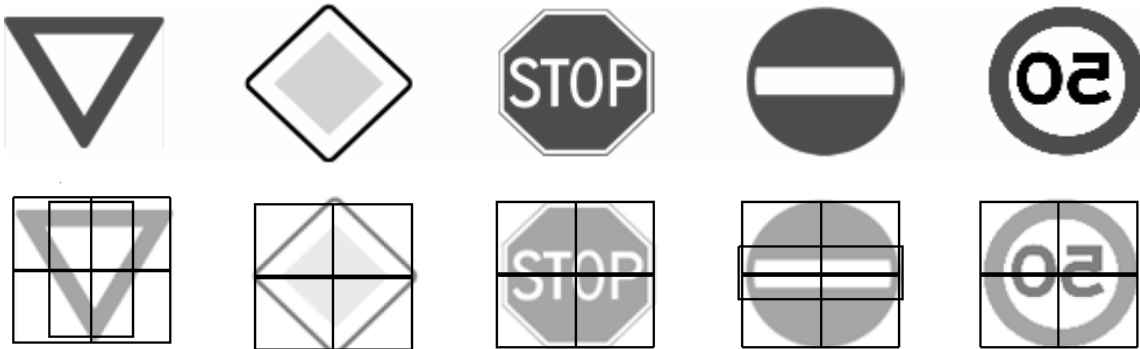
a



b

Şekil 4.4: (a) Haar dalgacına örnek, (b) AdaBoost eğitimi sonucu elde edilen ve nitelik çıkarmakta kullanılan Haar dalgacıları [30].

Alefs [31] tarafından geliştirilen yöntemde ise tanıma işleminde, kenar yönelim histogramları kullanılmıştır. İşaretlerden çıkarılan kenar yönelim nitelikleri, öğrenme aşamasından geçirildikten sonra eşleştirme sağlanmıştır. Kenar yönelim histogramları hesaplanırken işaret belirli sayıda parçalara ayrılarak (Şekil 4.5), bu alt parçalarda hesaplama yapılmış ve daha ayrıştırıcı sonuçlar elde edilmiştir. 10 trafik işaretini tanımaya yönelik geliştirilen sistem, %90 başarı ile saniyede 7 çerçeve işleyebilmektedir.



Şekil 4.5: 5 farklı türdeki trafik işaretinin alt-parçalara ayrılması [31].

Trafik işaretlerinin tanınmasına yönelik yapılmış çalışmaların başarımlarına, Çizelge 4.1’de yer verilmiştir.

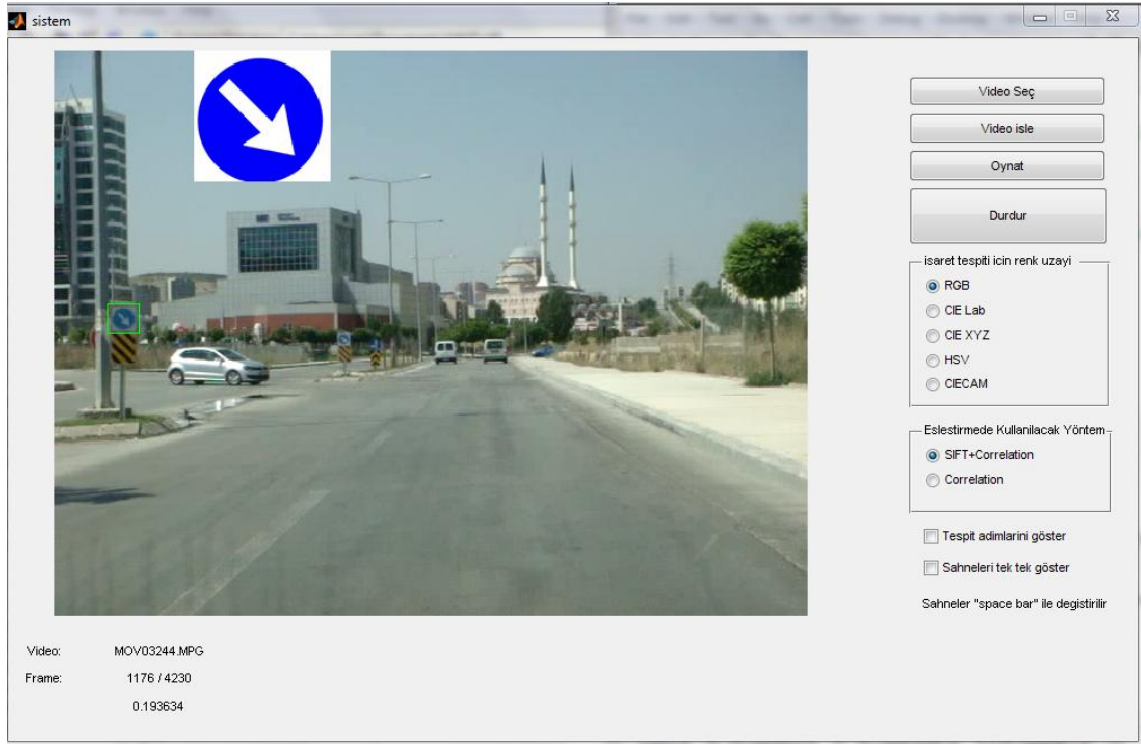
Çizelge 4.1: Yapılan çalışmaların başarımları.

Çalışmayı gerçekleştiren	Tanınabilen işaretler	İşleme hızı (saniye)	Tanım başarımları (%)
Miura [5]	Hız işaretleri	0,500	47
Torrens [6]	Hız işaretleri	0,125	90
Broggi [7]	İtalya’daki işaretler	0,100	-
Fang [8]	Tayvan’daki işaretler	0,300	-
Escalera [9]	14 tane trafik işareti	-	85
Gao [3]	İngiltere’deki işaretler	0,350 – 0,600	89
Bahlmann [30]	23 tane trafik işareti	0,100	85
Alefs [31]	10 tane trafik işareti	0,142	90

5. TRAFİK İŞARETLERİNİN OTOMATİK TANINMASI VE EŞLEŞTİRİLMESİ

5.1. Sistemin Genel Görünümü

Tez kapsamında, Türkiye karayollarındaki trafik işaretlerinin otomatik tanınması ve eşleştirilmesi için bir yöntem geliştirilmesi amaçlanmış, geliştirilen yöntemin denenmesi için de bir yazılım gerçekleştirilmiştir (Şekil 5.1).

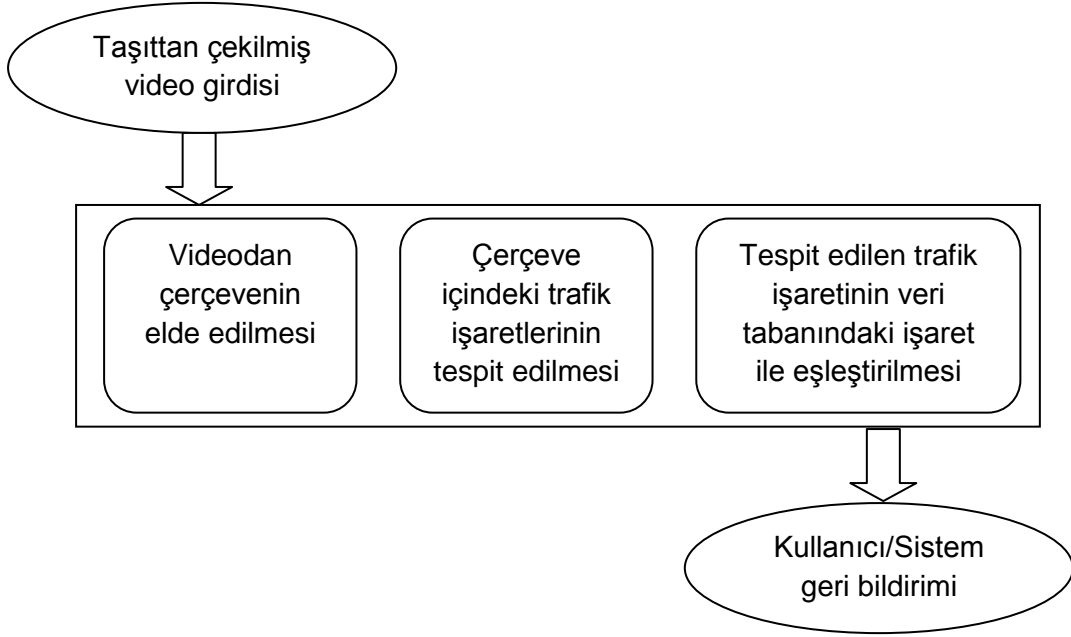


Şekil 5.1: Yazılımın kullanıcı arayüzü.

Şekil 5.2’de çizimi verilen genel görünümde; video girdisi, geliştirilen yöntemlerin temel adımları ve kullanıcı/sistem geri bildirimini gösterilmiştir. (Kesim 5.1.1.’de girdilerin yapısı ve kullanılan işaret veritabanı anlatılmıştır). Tez kapsamında yapılan çalışmaları, birbirinden bağımsız iki temel adımda ele alabiliriz:

- 1- Çerçeve içindeki trafik işaretlerinin tespit edilmesi,
- 2- Tespit edilen trafik işaretinin veri tabanındaki işaret ile eşleştirilmesi

Aşağıda, adımların amaçları ve kullanılan yöntemler özetlenecektir. Bu temel adımların her biri için farklı yaklaşımlar da uygulanabilir.



Şekil 5.2: Yazılımın genel görünümü.

Trafik işaretini doğru bir şekilde eşleştirebilmek için, çerçeve içindeki konumunun tam olarak saptanması gerekmektedir. Trafik işaretlerinin yapısı incelendiğinde, taşıt sürücülerinin dikkatini çekmesi için belirli renklerde tasarlandığı söylenebilir.

Bu sebepten, çevredeki nesnelere ayırt edebilmek için öncelikle işaretlerin renk niteliği kullanılmıştır. Tanınması amaçlanan işaretlerde, işaretin dışını sarmalayan renkler kırmızı ve mavidir. Çerçeve kırmızı ve mavi renk süzgeçlerinden geçirildikten sonra, ikili görüntü morfolojik işlem adımlarından geçirilerek, işaret olarak sınıflandırılıp sınıflandırılmayacağına karar verilir.

İşaret tespit edildikten sonra veri tabanındaki uygun karşılığına eşleştirmek gerekmektedir. Tez kapsamında yapılan çalışmada, işaretin rengine göre, iki farklı eşleştirme yöntemi kullanılmıştır. Birinci yöntemde, işaretten nitelikler elde edilerek, bu niteliklerin eşleştirilmesiyle aralarındaki ilişki anlamlandırılmıştır. Nitelik çıkarım ve tanımlayıcı yöntemi olarak SIFT kullanılmıştır. İkinci olarak ise şablon eşleştirme yöntemi kullanılmıştır.

Tezin bundan sonraki kesimlerinde, yukarıda belirtilen iki ana adımın alt adımları anlatılacak ve geliştirilen yöntemlere ait sonuç ve başarımlar değerlendirilmeleri yapılacaktır.

5.1.1. Veriler

Tez çalışması kapsamında kullanılan video görüntüleri, bu alanda Türkiye’de bir çalışma yapılmadığı için tez yazarı tarafından, Ankara ili içerisindeki yollarda, taşıt içinden kamera yardımı ile kaydedilmiştir. Videolar, RGB biçiminde ve 640x480 çözünürlüğünde, saniyede 30 kare sıklığında kaydedilmiştir.

Görüntüler taşıt içerisindeki kamera ile kaydedildiğinden, taşıt camının görüntüye olumsuz yönde etkisi olmaktadır. Bu tür sistemler için kullanılan görüntüler, çoğunlukla aracın dışına yerleştirilmiş bir kamera yardımıyla kaydedildiğinden, görüntüler daha nettir. Buna ek olarak, kameranın taşıt içine sabitlenememesinden kaynaklanan titreşimler, görüntülerde az da olsa bulanıklığa sebep olmuştur.

Tanınması amaçlanan trafik işaretleri kümesi, Türkiye’de trafik işaretlerinden sorumlu kuruluş olan Karayolları Genel Müdürlüğü’nden alınmıştır. Elde edilen CAD çiziminden trafik işaretleri ayrı ayrı kesilmiş ve tanınacak 52 işareti içeren bir veri kümesi oluşturulmuştur. Türkiye’deki trafik işaretleri Tehlike Uyarı İşaretleri, Trafik Tanzim İşaretleri, Bilgi İşaretleri, Durma ve Park etme İşaretleri gibi alt kategorilere ayrılmakta, değişik nitelikte ve çok sayıda trafik işareti bulunmaktadır. Tez kapsamında yapılan çalışmada, karayollarında en çok kullanılan trafik işaretlerinin tanınması amaçlanmıştır. Fakat Türkiye Karayolları Genel Müdürlüğü’nde, hangi işaretin ne sıklıkla kullanıldığına ilişkin bir istatistiksel verinin tutulmadığı öğrenilmiştir.

Trafik işaretlerinin yollara yerleştirilmesinden sorumlu olan kuruluş İl Belediyeleri’dir. Ankara Belediyesi’nde, aylara göre dikilen trafik işaretlerinin bilgisi olduğu öğrenilmiştir. Haziran 2009, Aralık 2009 ve Ocak 2010 aylarındaki dikilen işaret sayıları Belediye yetkililerince sağlanmıştır. Bu aylardaki verilerden genelleme yapılarak işaretlerin kullanım sıklıkları belirlenmiştir. Tez çalışmasında gerçekleştirilen sistemde tanınabilen işaretleri EK-1’de, bu işaretlerin yollardaki kullanım sıklıkları EK-2’deki çizelgede gösterilmiştir.

5.1.2. Yazılım ve Donanım Araçları

Tez kapsamında yapılan çalışmanın geliştirme ortamı olarak MATLAB kullanılmıştır [32]. MATLAB, yoğun hesaplama içeren işlem adımlarını, C, C++, Java gibi geleneksel yazılım geliştirme dillerinden daha hızlı geliştirmeye olanak

sağlayan yüksek düzeyli bir dil ve etkileşimli bir ortamdır. Bunlara ek olarak gelişmiş bir görüntü işleme kütüphanesine sahip olması, MATLAB ortamının seçilmesindeki en önemli etkidir.

Nitelik çıkarma amacıyla Lowe [33]'un SIFT gerçekleştirimi kullanılmıştır.

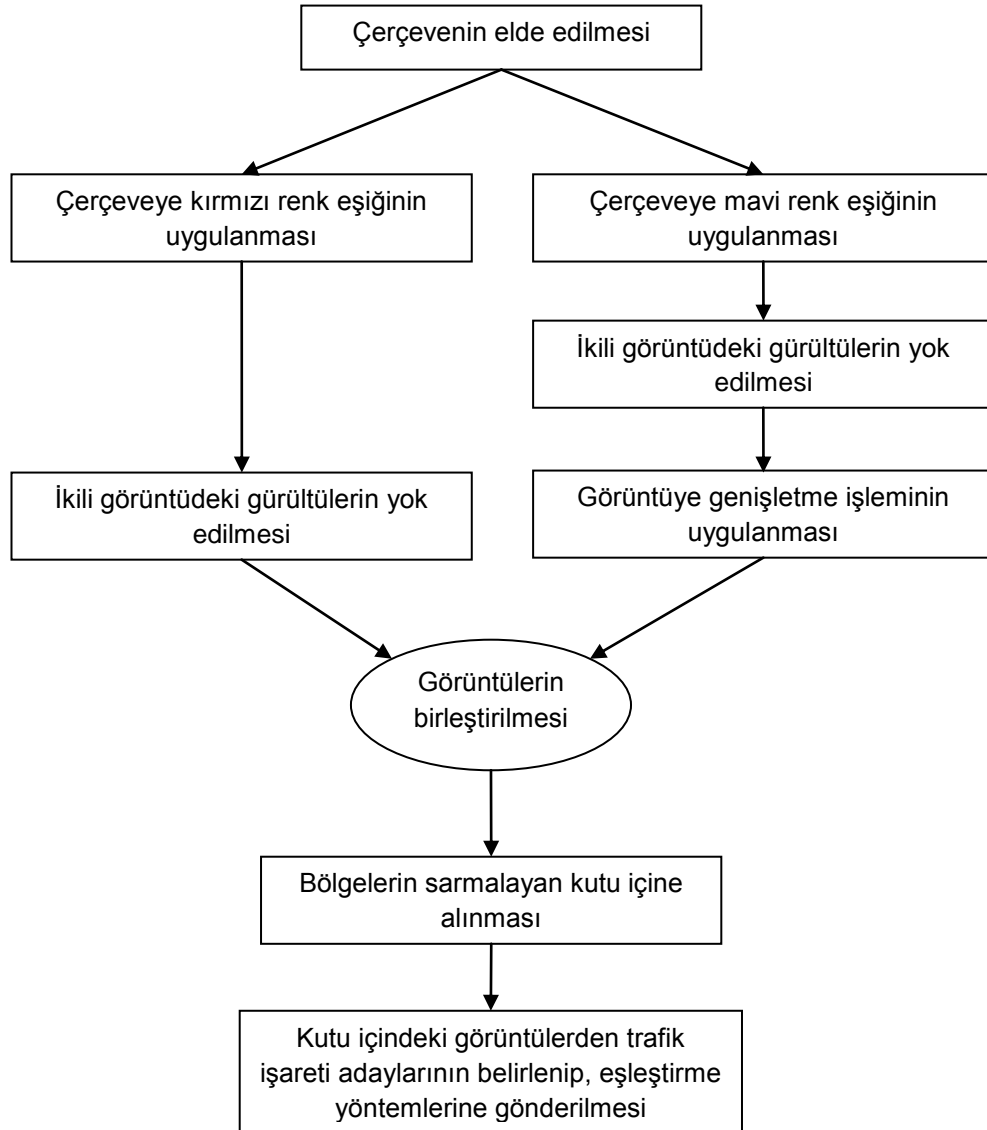
Yöntemlerin çalıştırıldığı bilgisayar sisteminde: Core 2 Quad 3.00 Ghz işlemci, 3 GB Ana bellek, GTX 280 ekran kartı, bulunmaktadır.

Trafik işaretlerinin bulunduğu görüntüler Sony marka, DSC-W120 model dijital kamera kullanılarak kaydedilmiştir.

5.2. Yöntem Adımları ve Uygulamaları

5.2.1 Çerçeve İçindeki Trafik İşaretlerinin Tespit Edilmesi

Şekil 5.3'de çizimi verilen genel sistem görünümünde; eşleştirme uygulanacak trafik işaretlerinin çerçeve içerisinden çıkarılmasında uygulanan adımlar belirtilmiştir. Sonraki kesimlerde uygulanan adımların gerekçeleri ve detayları anlatılacaktır.



Şekil 5.3: Trafik işareti tespitinin genel görünümü.

5.2.1.1. Renk Eşığı Uygulama Adımında RGB Renk Uzayının Kullanımı

Trafik işaretlerinin yapısı ve nitelikleri incelendiğinde, renk niteliğinin, en önemli ayırt edici özellik olduğu söylenebilir. İşaretin çevresindeki nesnelere ayrılmasında ilk adım olarak renk eşığı uygulanmıştır. Renk eşığı, ilk olarak RGB renk uzayında uygulanmıştır. Bunun en önemli sebebi kameradan elde edilen görüntünün RGB biçiminde olmasıdır. Böylece, görüntüyü örneğin HSV renk uzayına dönüştürmek gibi hesaplama süresini arttıracak işlemlerden kaçınılmıştır. Diğer bir sebebi ise tanınması amaçlanan işaretlerin kırmızı ve mavi baskın renklerine sahip olmalarıdır. RGB biçimindeki görüntüden kolaylıkla elde edilen kırmızı ve mavi renk kanallarına uygulanacak eşikleme işlemleri ile işaretlerin belirlenmesi sağlanmıştır. İçerisinde trafik işareti bulunan çerçevenin RGB renk uzayındaki görüntüsü, kırmızı renk kanalı ve mavi renk kanalı Şekil 5.4'de gösterilmiştir.

Renk eşikleme işlemi uygulanırken, trafik işaretlerinde kullanılan mavi ve kırmızı renklerin RGB renk kodlarına sahip piksellerin yakalanması amaçlanmıştır. Ancak yollardaki trafik işaretlerinin tam olarak standartlara uymaması, hava ve ışık şartlarındaki değişiklikler göz önüne alındığında, bir miktar hata payı bırakılmıştır. Bu değerler, veri kümesindeki görüntülerden elde edilen trafik işaretleri kullanılarak belirlenmiştir. Kırmızı renk eşikleme işlemi için kısıtlar aşağıda listelenmiştir (R: kırmızı kanaldaki piksel değeri, G: yeşil kanaldaki piksel değeri, B: mavi kanaldaki piksel değeri ve $\{ R, G, B \} \in [0, 255]$):

- $R > 1.3 * G$
- $R > 1.3 * B$
- $R > 100$

Şekil 5.4a'deki çerçeve üzerinde uygulanan kırmızı renk eşikleme işlemi sonucunda ikili görüntü elde edilmiştir (Şekil 5.5a). Benzer şekilde, renkli görüntüye mavi renk eşikleme işlemi de aşağıdaki kısıtlara göre uygulanmıştır.

- $B > 2 * R$
- $B > G$
- $B > 20$



a



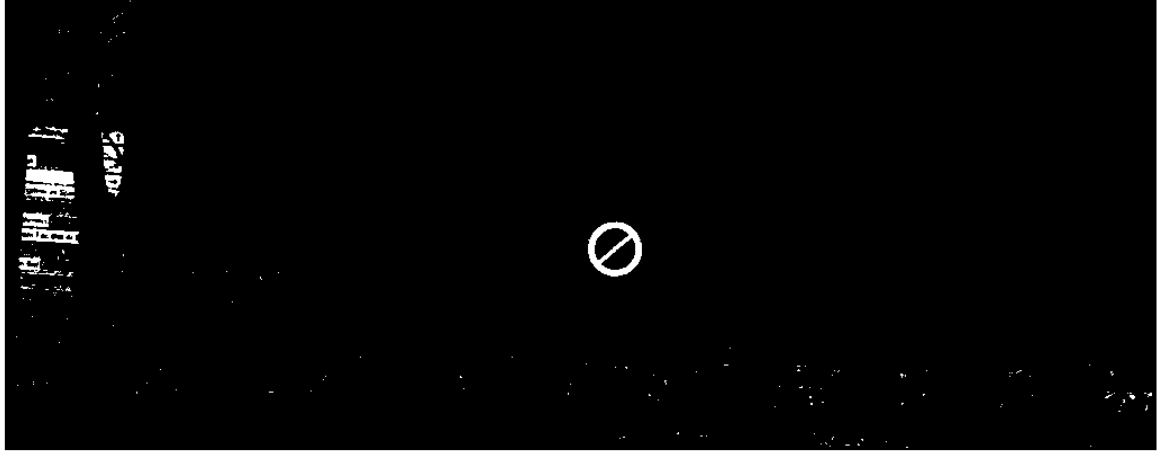
b



c

Şekil 5.4: (a) RGB renk uzayındaki görüntü. (b) Görüntünün kırmızı renk kanalı. (c) Görüntünün mavi renk kanalı.

Bu işlem sonucunda da mavi işaretin belirttiği ikili görüntü elde edilmiştir (Şekil 5.5b).



a



b

Şekil 5.5: (a) Kırmızı renk eşikleme sonrası görüntü. (b) Mavi renk eşikleme sonrası görüntü.

Bu çalışmada belirlenen eşik değerleri elimizdeki verilerin kaydedildiği kameraya özgüdür. Farklı bir kamerada çekilen videolarda farklı eşik değerleri belirlenmesi gerekebilir. Bu durum renk tabanlı çalışmalarda karşılaşılan genel bir problemdir. Bu amaçla, cihazdan bağımsız renk temsili için tanımlanan CIE L^*a^*b renk uzayı üzerine çalışılmış, fakat bu renk uzayının kullanılabilmesi için gerekli olan kameraya özgü bazı ölçümlere (*calibration*) değerlerine sahip olunamadığı için kullanılamamıştır.

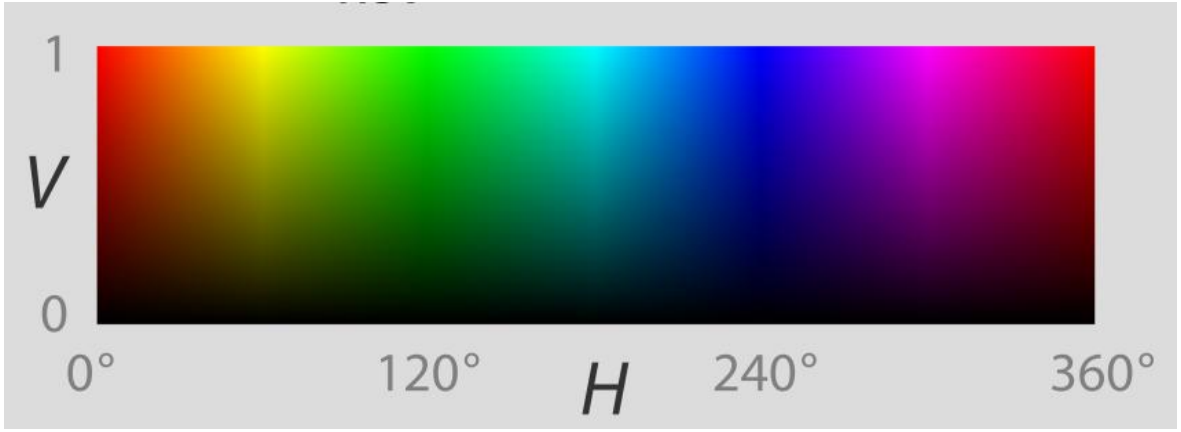
5.2.1.2. Renk Eşığı Uygulama Adımında HSV Renk Uzayının Kullanımı

Kırmızı ve mavi trafik işaretlerinin tespit edilmesinde HSV renk uzayının renk özü (*Hue*) değeri kullanılmıştır. Renk özü değerinin renklere göre değişimi Şekil 5.6'te verilmiştir. Kırmızı renk eşikleme işlemi için kıstaslar aşağıda listelenmiştir ($H \in [0,360]$):

- $H > 329$ ve $H < 29$

Benzer şekilde, HSV biçimindeki görüntüye mavi renk eşikleme işlemi de aşağıdaki kıstaslara göre uygulanmıştır:

- $199 < H < 270$



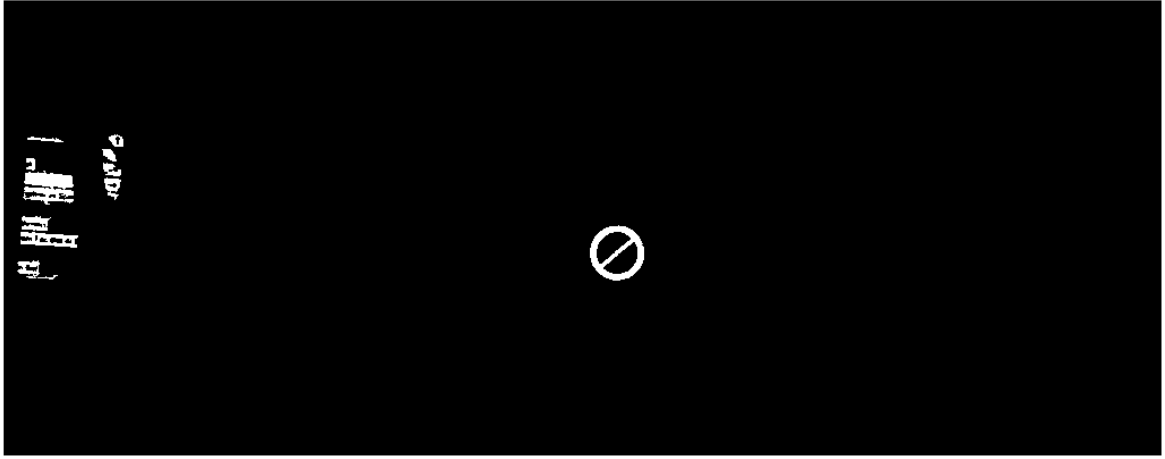
Şekil 5.6: HSV renk uzayında renk özü (*Hue*) değerinin renklere göre değişimi.

Yaptığımız çalışmalarda HSV renk uzayı kullanarak trafik işaretinin sahne içinde bulunduğu alanın tespitinde yeterince başarımlı elde edilememiştir. Bu nedenle, trafik işaretlerinin sahne içinden tespiti amacıyla RGB renk uzayı kullanılmıştır.

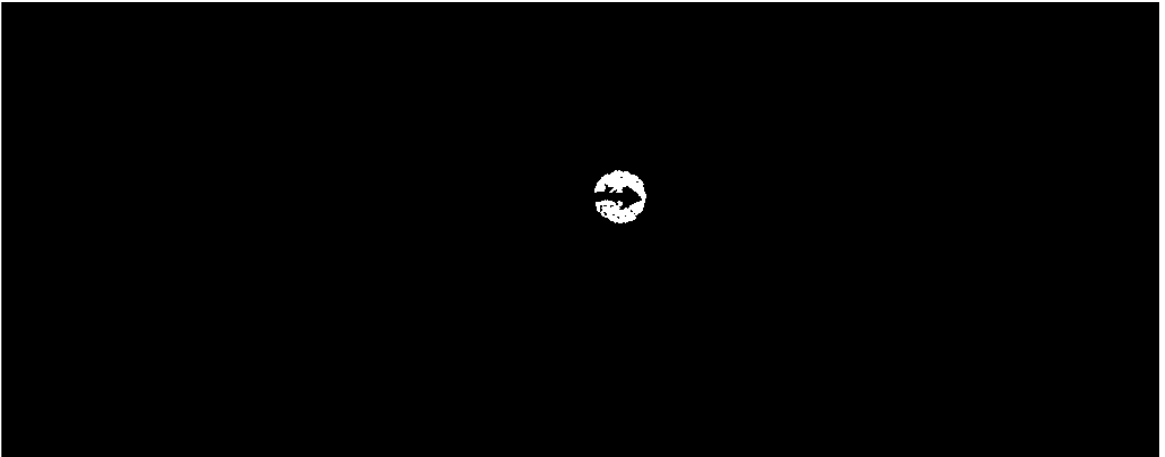
5.2.1.3. Gürültülerin Yok Edilmesi

Renk eşikleme işleminden sonra elde edilen görüntülerde, trafik işareti dışında birçok nesnenin ve gürültünün süzgeçten geçtiği gözlemlenmiştir. İşaret adaylarının konumlarının saptanması için bu tür gürültülerden kurtulmak gerekmektedir. Buna çözüm olarak ikili görüntülerdeki düşük alanlı bölgeleri yok etmeye yönelik açma işlemi uygulanmıştır.

Alan açma işleminde 4-bağlı komşuluğa sahip bölgeler tespit edilir, bu bölgelerin alanları hesaplanır ve belirlenen eşik değerinden düşük alana sahip bölgeler sıfıra eşitlenir. Yapılan çalışmalarda, 200 piksel alanının altındaki bölgelerin silinmesi, işaretin korunarak gürültülerin yok edilmesi için uygun bir değer olarak saptanmıştır. Şekil 5.7'te gürültüleri yok edilmiş kırmızı ve mavi işaretleri içeren ikili görüntüler verilmiştir. Şekil 5.7a'da görüldüğü üzere bu işlem sonucu elde edilen her alan bir trafik işaretine ilişkin değildir. İşaret olmayan alanların elemesi, sonraki adımlarda yapılacaktır.



a



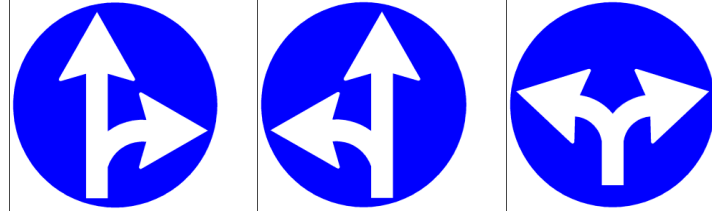
b

Şekil 5.7: (a) Kırmızı işaretin olduğu görüntüde alan yok etme. (b) Mavi işaretin olduğu görüntüde alan yok etme.

5.2.1.4. Genişletme İşlemi

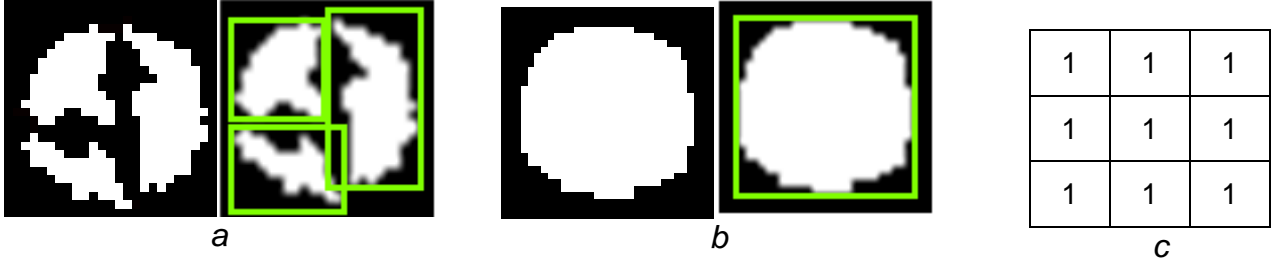
Görüntüdeki gürültülerin yok edilmesinin ardından, işaret adayının belirlenmesi için sarmalayan kutuya alınması öngörülmüştür. Fakat mavi işaretlerin (çoğunlukla

Şekil 5.8'deki işaretler) uzakta ve küçük olduğu durumlarda, tek bir sarmalayan kutu ile temsil edilememesi sorunu ortaya çıkmıştır. Bunun sebebi işaretlerdeki okların uç kısımlarının çok dışta olması ve o bölgedeki mavi rengin, görüntüde neredeyse hiç gözükmemesidir. Bu gibi durumlarda Şekil 5.9a'da gösterildiği gibi işaret birden fazla kutu ile temsil edilmekte ve bu alt kutuların asıl işaretle eşleştirilmesi mümkün olmamaktadır.



Şekil 5.8: Görüntü içinde parçalara ayrılan işaretler

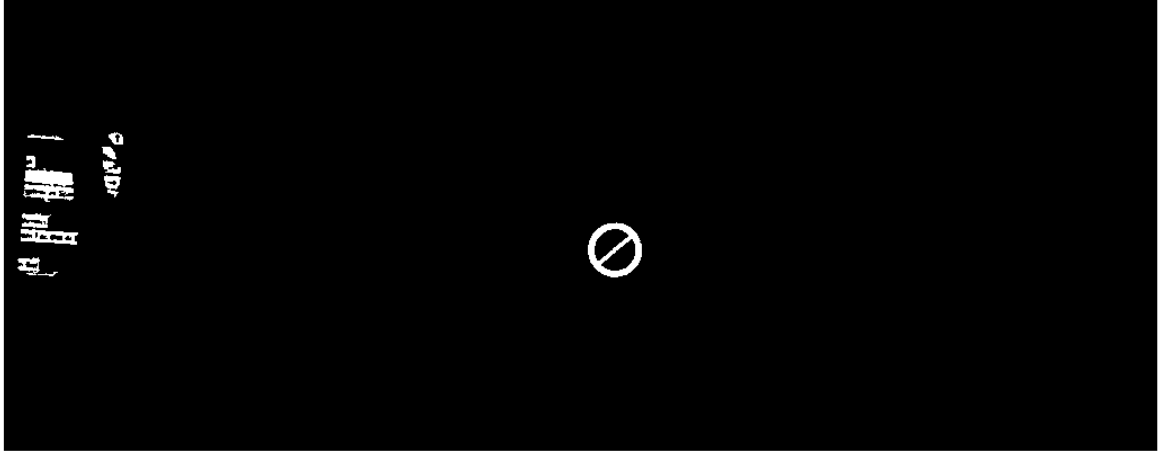
Görüntü içindeki işaretlerin parçalanmasına çözüm olarak, aralardaki boşlukların doldurulması amacıyla 4 defa morfolojik genişletme işlemi uygulanmıştır. Sonuç olarak işaretlerin Şekil 5.6b'deki gibi tek parça haline gelmeleri sağlanmıştır.



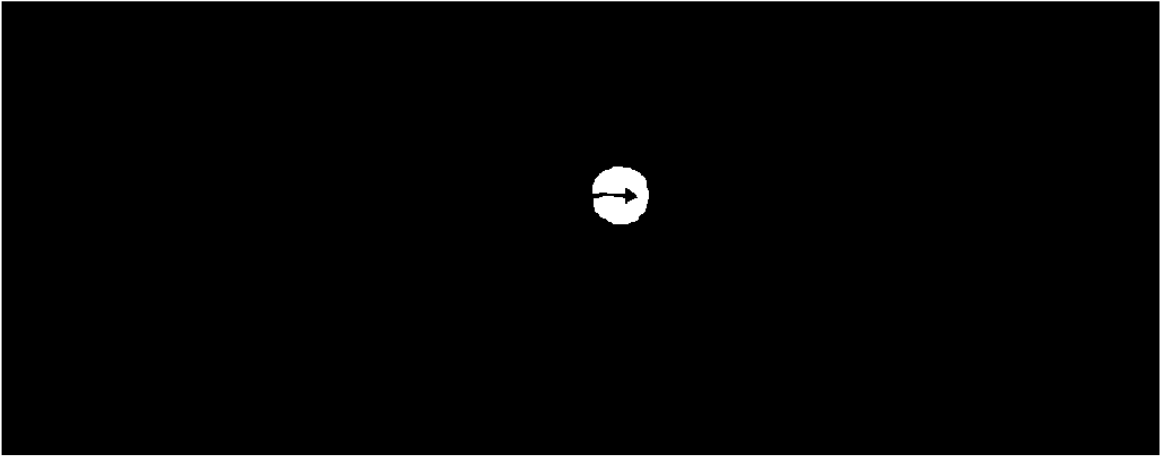
Şekil 5.9: (a) İşaretin genişletme öncesindeki durumu ve sarmalayan kutu içine alınması. (b) İşaretin genişletme sonrasındaki durumu ve kutu içine alınması. (c) Genişletme işleminde kullanılan yapısal eleman

5.2.1.5. Farklı Renk Eşiklemelerinin Birleştirilmesi ve İşaret Adaylarının Belirlenmesi

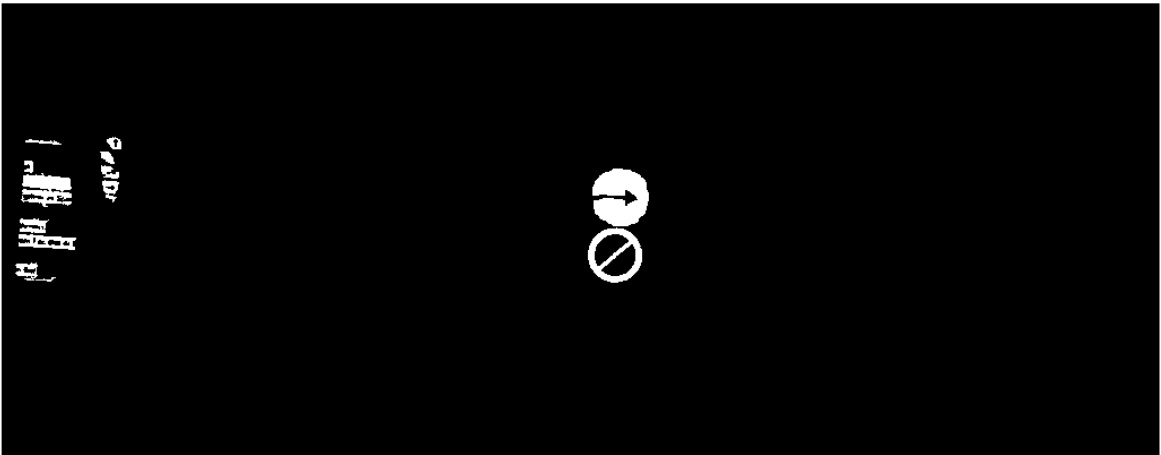
Görüntüdeki mavi ve kırmızı işaretlerin sarmalayan kutu içine alınması ve işaret adayları olarak belirlenmesi gerekmektedir. Fakat kırmızı ve mavi renk eşiklemelerinden gelen iki farklı ikili görüntüde (Şekil 5.10a ve Şekil 5.10b) kutu içine alma işlemi, yazılımın çalışma süresini arttıran bir etkidir. Bundan dolayı bu görüntülerin birleştirildikten sonra işaretlerin kutu içine alınması daha hızlı bir



a



b



c

Şekil 5.10: (a) Kırmızı renkli işaretlerin tespitinde kullanılan görüntü. (b) Mavi renkli işaretlerin tespitinde kullanılan görüntü. (c) (a) ve (b)'nin VEYA işlecine tabi tutulması.

çözüm olarak belirlenmiştir. Bu ikili görüntüleri birleştirmek için VEYA işleci kullanılmıştır (Şekil 5.10c).

Elde edilen görüntüde trafik işaretleri dışında, yakın renklere sahip, başka nesnelere de bulunabilir. Bunlar da sarmalayan kutu içine alınacaktır. Fakat yollarda bulunan işaretlerin geometrik şekil olarak bazı koşullara uyması gerekmektedir. Trafik işaretleri genelde üçgen, kare ve daire şeklinde olmaktadır. Bu koşullara uymayan şekillerin trafik işareti olamayacağı kesindir. Böylece, trafik işareti olmayan şekillerin eşleştirme yöntemlerine gönderilerek, işleme süresini uzatmanın ve hatalı eşleştirme yapmanın önüne geçilmiştir.

Sarmalayan kutu, ikili görüntüdeki bir bölgeyi kapsayan en küçük dikdörtgendir. Bu dikdörtgenin görüntü içindeki konumu (x,y koordinatları cinsinde), ve dikdörtgenin en, boy bilgileri (piksel cinsinde) elde edilebilmektedir. Bu bilgiler kullanılarak aşağıdaki koşullara uyan kutular, muhtemel bir trafik işareti varsayıp sahnedeki kesilerek eşleştirme yöntemine gönderilmiştir (Şekil 5.11):

1. En / boy oranı 0.8 ile 1.25 arasında olanlar,
2. Eni ve boyu 15 pikselden büyük olanlar.

Trafik işaretlerinin şekilleri itibariyle, sarmalayan kutu içine alındıklarında, bu kutunun kare biçiminde olması beklenir (en/boy oranı: 1). Ancak kameranın tam olarak işaret yüzeyine dik gelmemesi halinde veya işaretin az da olsa sağa/sola dönmüş olabileceği durumlarda, en/boy oranı 1'den uzaklaşır. Veri kümesindeki işaretler de incelendikten sonra, bu durumdaki işaretleri de tespit edebilmek için bu oran esnetilerek yukarıdaki 1. koşuldaki değerler belirlenmiştir.

Trafik işaretinin çok uzakta olduğu durumlarda, işaret belirli bir boyutun altında kalır. Bu durum işaretin eşleştirilmesini zorlaştırır ya da hatalı sonuçlara yol açar. Bu sebepten 2. koşuldaki değerler belirlenmiş ve bu değerlerin altındaki nesnelere ihmal edilmesi kararlaştırılmıştır.



Şekil 5.11: İçerisinde trafik işareti tespit edilen bazı sahneler



Şekil 5.11 (devamı): İçerisinde trafik işareti tespit edilen bazı sahneler

5.2.1.6. İşaret Renginin Tespit Edilmesi

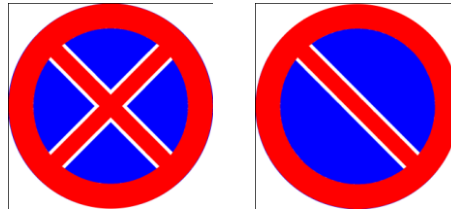
Tez kapsamında yapılan çalışmada tanınması amaçlanan işaretlerin renk nitelikleri incelendiğinde, kırmızı rengin baskın olduğu işaretler, mavi rengin baskın olduğu işaretler ve kırmızı ile mavi rengin birlikte baskın olduğu işaretler şeklinde gruplara ayrıldığı gözlemlenmektedir. Bu nitelik kullanılarak işaret eşleştirmesi yaparken taranan şablon kümesi azaltılabilir. Bu yüzden çerçeve içerisinde çıkarılan işaretlerin baskın renklerinin tespit edilmesi hem yazılımın hızını, hem de başarımını arttırmaktadır.

Görüntülerdeki renklerin dağılımlarını incelemekte çoğunlukla renk histogramları kullanılır. Renk histogramları bir görüntüdeki renk değerlerinin sayısını gösteren grafiklerdir. Trafik işaretlerindeki baskın rengin tespit edilmesinde, renk histogramlarından daha hızlı sonuç veren bir yöntem geliştirilmesi uygun

görülmüştür. Çünkü tespit edilmesi gereken renkler sadece kırmızı ve mavidir. Görüntülerde RGB renk uzayının kullanılmasındaki diğer bir avantajda, RGB görüntüden kırmızı ve mavi kanalların kolayca çıkarılabilmesidir. Bu sayede bu renk kanallarını oluşturan piksel değerlerinin ortalamaları alınmıştır. Hangi renk kanalının ortalama değeri daha yüksekse, işaretteki baskın rengin o olduğu sonucu çıkarılabilir. Veri kümesindeki görüntülerden çıkarılan işaretlerde yapılan testlerde de bu yaklaşım başarılı ve hızlı sonuçlar vermiştir.

Rengine göre ayrılması gereken diğer bir grup da iki işareten oluşan, kırmızı ve mavi rengin birlikte baskın olduğu işaret grubudur (Şekil 5.12). Bu işaretlerdeki renk kanallarının ortalama değerlerinin birbirine yakınlığı saptanmış ve Eş. 5.1 kullanılarak işaretin bu gruba dahil olup olmadığı denetlenmiştir.

$$| \text{KırmızıKanalOrtalama} - \text{MaviKanalOrtalama} | < 15 \quad (5.1)$$



Şekil 5.12: Kırmızı ve mavi renkleri içeren işaretler.

5.2.2. Trafik İşaretlerinin Eşleştirilmesi

Bu aşamada, çerçeve içinden tespit edilen trafik işaretlerinin, tanınması amaçlanan işaretler kümesinden uygun olan işaret ile eşleştirilmesi açıklanacaktır. Eşleştirme, temel olarak trafik işaretlerinin niteliklerinin çıkartılarak, bu niteliklerin karşılaştırılması yöntemine dayanmaktadır. İşaretlerdeki niteliklerin çıkarılmasında SIFT kullanılmıştır.

İşaretlerin niteliklerinin çıkarılmasında SIFT kullanılmasının en önemli avantajları, ölçeklemeye, yönelime, ilgin bozulmalara ve kısmen ışıklandırma farklılıklarına karşı değişmez olmasıdır. Tespit edilen işaretlerdeki bu tür etkenler, eşleştirme sonuçlarını olumsuz etkilememektedir.

Ancak mavi trafik işaretlerinde SIFT metodu doğru sonuçlar vermemektedir. Örneğin tek ok simgesinden oluşan işaretlerde, SIFT metodu kullanıldığında ayırt

edici nitelikler çıkarılamamaktadır. Çünkü bu yöntem yönelimden etkilenmediği için, bu işaretleri aynı olarak algılamaktadır. Bu yüzden mavi işaretlerin eşleştirilmesi için normalleştirilmiş korelasyon yöntemi kullanılmıştır.

5.2.2.1. SIFT Kullanarak Eşleştirme

Eşleştirme başarımının yanında diğer önemli unsur ise hesaplama süresidir. SIFT nitelik çıkarımı bütün çevreye uygulandığında hesaplama süresi oldukça fazladır (yaklaşık 1 sn). Ancak tespit edilen trafik işaretine ilişkin alan görüntünün içinden kesilerek, sadece işareti içeren sarmalayan kutuya (Şekil 5.13) SIFT nitelik çıkarım işlemi uygulandığı zaman hesaplama süresi azalmaktadır.



Şekil 5.13: Sahne içinde tespit edilmiş trafik işaretleri.

İşaretlerin eşleştirilmesine ön hazırlık olarak, tanınması amaçlanan veritabanındaki 46 trafik işaretinin nitelik çıkarım işlemleri gerçekleştirilmiş ve anahtar nokta tanımlayıcıları çıkarılmıştır. Bu işlem gerçekleştirilirken veritabanındaki 46 trafik işaretinin üç farklı çözünürlükteki görüntülerinden de nitelikler çıkartılmıştır.

- 1172x1172: Trafik işaretlerinin CAD dosyasından kesilmiş yüksek çözünürlükteki veri kümesi,

- 110x110: Kaydedilen görüntülerde tespit edilen trafik işaretlerinin ortalama boyutuna sahip veri kümesi,
- 50x50: Düşük çözünürlükteki veri kümesi.

SIFT yöntemi ölçekten bağımsız olduğu için, eşleştirme adımındaki başarımda ve eşleştirme hızında bir değişim yoktur. Yazılımda ortalama değer olan 110x110 veri kümesinin nitelikleri kullanılmaktadır. Yazılımın içerisinde tanımlayıcıların bilgileri matrisler halinde tutulmaktadır. Böylece her eşleştirme yinelemesinde, veritabanındaki 46 trafik işaretinin tekrar tekrar nitelik çıkarım işlemine tabi tutulması engellenmiş ve eşleştirme hızı artırılmıştır.

Kutu içine alınan işaretlerin baskın renk bilgileri bir önceki adımdan elde edilebildiği için, bu bilgi kullanılarak tarama kümesi daraltılmış, böylece hem yazılımın başarımı artırılmış, hem de hesaplama süresi azaltılmıştır. Kırmızı işaretler 34 işaretin bulunduğu kümeden, mavi işaretler 16 işaretin bulunduğu kümeden, kırmızı ve mavi rengin aynı oranda olduğu işaretler ise 2 işaretin bulunduğu kümeden eşleştirilmeye çalışılmıştır.

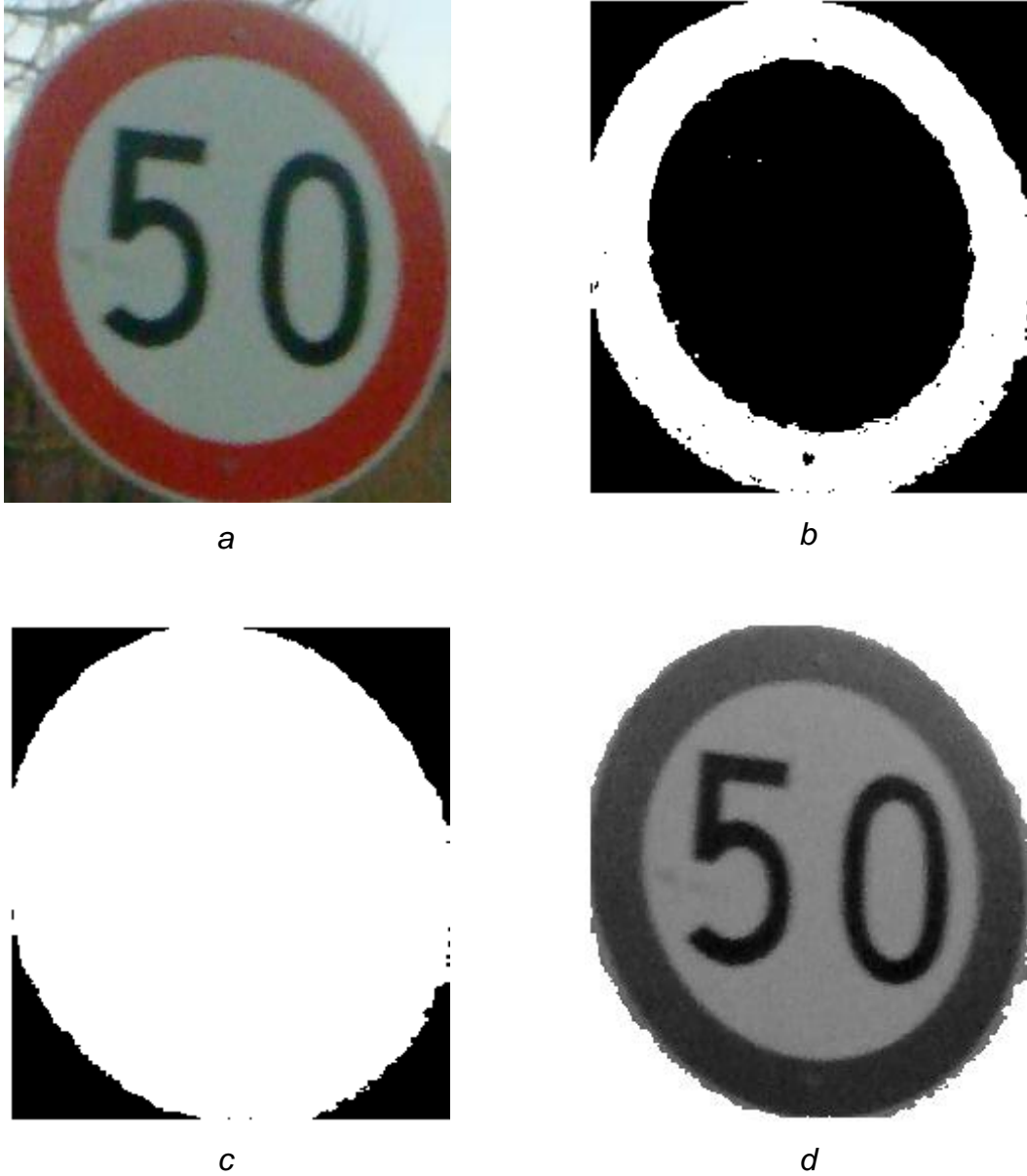
5.2.2.1.1. İşaretlerin Dışının Temizlenmesi

Tanınması amaçlanan trafik işaretlerinin 4 tanesi kare, geri kalanları ise üçgen ve daireden oluşmaktadır. Video görüntüsü içinden, sarmalayan kutu içine alınarak kesilen trafik işaretlerinde, kare şeklinde olanlar haricinde, işaretin dış yüzeyinde çevredeki nesnelere kalıntılar yer almaktadır. İşaret dışındaki bütün kalıntılar, nitelik çıkarım işlemi etkilemekte, anahtar noktalar eşleştirilirken yanlış sonuçlara yol açmaktadır. Eşleştirmenin başarımını arttırmak için kutu içindeki işaretlerin, tarama yapılacak şablon görüntülerdeki işaretlere benzerliğini arttırmak gerekir. Bunu sağlamak için, işaretlerin dış yüzeyinin temizlenmesi ve kutu içerisinde işaretten başka çevre kalıntısı kalmaması sağlanmıştır.

Kutudaki görüntüye kırmızı renk eşikleme işlemi uygulanmıştır. Bütün kırmızı trafik işaretlerinin dış yüzeyi tamamıyla kırmızı olduğundan, içi boş bir kapalı eğri elde edilmiştir (Şekil 5.14b). Bu kapalı eğrinin içi doldurulmuştur (Şekil 5.14c). Son olarak da işaret etrafında kırmızı renk eşikinden geçen kalıntılar kalmışsa, içerdiği piksel sayısı belli bir eşik değerinin altında olan alanları yok etme yöntemiyle bu

kalıntılar yok edilir. Çalışmada kullandığımız görüntülerde eşik değeri olarak 50 değerinin uygun olduğu tespit edilmiştir.

Morfolojik işlemler sonucu elde edilen bu ikili görüntü, kutudaki ilk görüntüye maske olarak uygulanmış ve sadece işareti içeren görüntü elde edilmiştir (Şekil 5.14d). Şekil 5.15'te dışı temizlenmiş bazı işaretlere örnekler verilmiştir.



Şekil 5.14: (a) İşaretin orijinal görüntüsü, (b) Renk eşikleme işleminden sonra elde edilen ikili görüntü, (c) Boşluk doldurma işleminden sonraki görüntü, (d) Orijinal görüntüye (c) maskesi uygulanarak işaretin dışının temizlenmesi.



Şekil 5.15: Dışı temizlenmiş bazı işaretler.

5.2.2.1.2. SIFT Nitelik Çıkarımı ve Anahtar Nokta Tanımlayıcıların Eşleştirilmesi

Temizlenen işaretle SIFT nitelik çıkarım işlemi gerçekleştirilmiş ve anahtar nokta tanımlayıcıları çıkartılmıştır. Veri kümesinde tutulan şablon işaretlerin nitelikleri ile eşleştirilirken, eşik değer tabanlı eşleştirme kullanılmıştır. İki nitelik arasındaki uzaklık değeri, diğer nitelikler ile arasındaki uzaklıkların altında bir değere sahip ise eşleşme gerçekleştirilir (Eş. 5.2). 5.2'deki ifadede tespit edilen işaretin nitelik kümesi D , veritabanındaki bir işaretin nitelik kümesi V ile gösterilmiştir.

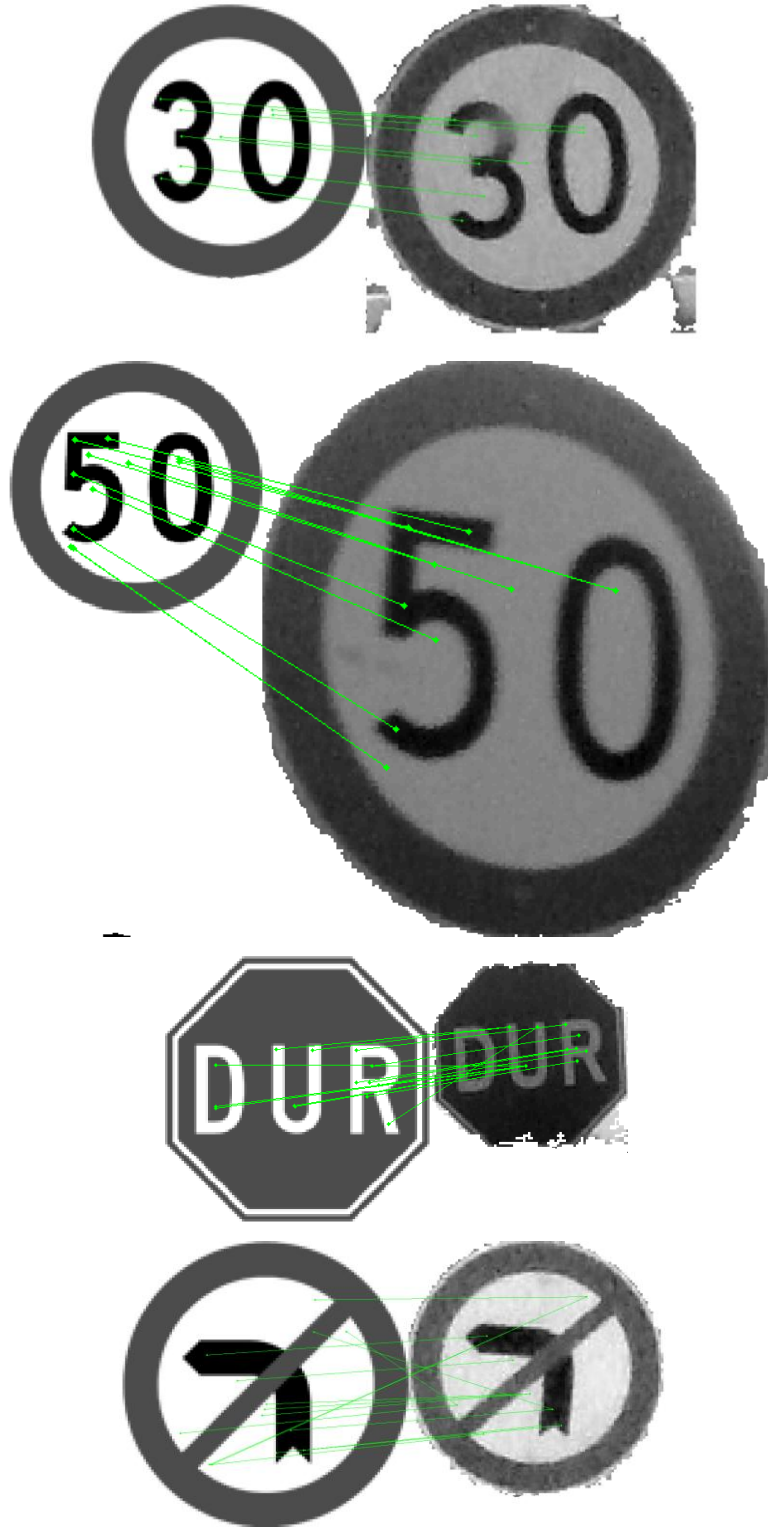
$$\text{ÖklidUzaklığı}(D_i, V_j) < \forall \text{ÖklidUzaklığı}(D_i, V_k) \mid V_j \neq V_k, V_k \in V \Rightarrow D_i = V_j \quad (5.2)$$

İki görüntü arasındaki niteliklerin eşleşme sonucu, görüntülerin ne kadar sayıda benzer niteliğe sahip olduğunu gösterir. Şablon veri kümesindeki bütün işaretler için bu benzerlik değeri hesaplandıktan sonra, görüntüden kesilen alan en yüksek sonucu veren işaret ile eşleştirilmiştir (Şekil 5.16'da benzer niteliklerin eşleşmesine ait gösterimde, veritabanında saklanan işaretler aynı boyutta olmasına rağmen, sahne içinden alınan görüntüler farklı boyutlardadır).

Kutu içine alınan görüntünün işaret olmama olasılığı da vardır. Trafik işareti ile aynı renklere ve boyutlara sahip çevredeki nesnelere de işaret eşleştirme adımına gelebilir. Fakat bu şekilde seçilen alanların veri kümesindeki işaretler ile eşleştirme sonucunda benzerlik değeri çok düşük çıkmaktadır. Bu durumda, seçilen alanın hiçbir işarete benzemediği ve trafik işareti olmadığı sonucuna varılır.

Tespit edilen nesne trafik işareti ise, eşleştirme sonucunda 7 ile 22 arasında benzer nitelik olduğu saptanmıştır. Bu sayılar doğru işaretle eşleşip eşleşmediği konusunda tam bir bilgi vermemektedir. Ancak nesnenin trafik işareti olduğu

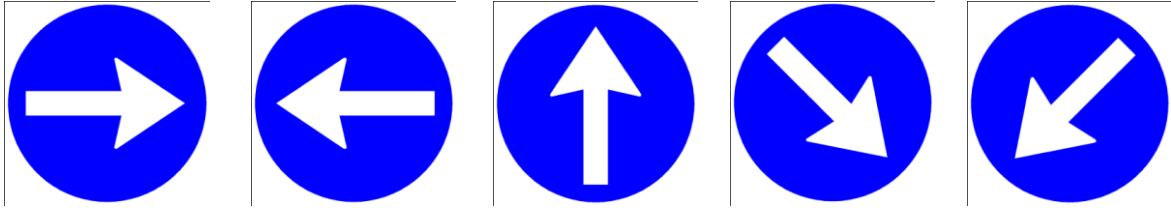
anlaşılmaktadır. Bu yüzden benzerlik sayısı 5'ten düşük olan nesnelere trafik işareti olarak sınıflandırılmamış ve eşleştirilmemiştir.



Şekil 5.16: Tespit edilen işaretler ile veri kümesindeki en fazla benzerliğe sahip işaretlerin nitelik eşleşmeleri.

5.2.2.2. Normalleştirilmiş Korelasyon Kullanarak Eşleştirme

Mavi trafik işaretlerinin nitelikleri incelendiğinde, 14 işaretten 10 tanesinin ok simgesi içerdiğini görebiliriz. Özellikle Şekil 5.17'deki mecburi yön levhaları birbirleriyle tamamen aynı işaret olup; bu levhanın döndürülmesiyle elde edilmişlerdir. Bu yüzden SIFT algoritması kullanılan nitelik çıkarma yöntemi bu işaretlerde başarılı olmamış, yanlış eşleşmelere sebep olmuştur. Mavi işaretlerin eşleştirilmesinde normalleştirilmiş korelasyon kullanılmıştır. Mavi ve kırmızı renkleri birlikte içeren Şekil 5.12'deki iki işaret de, ayırt edici niteliğe sahip olmadıkları için korelasyon yöntemi kullanılarak eşleştirilmiştir.



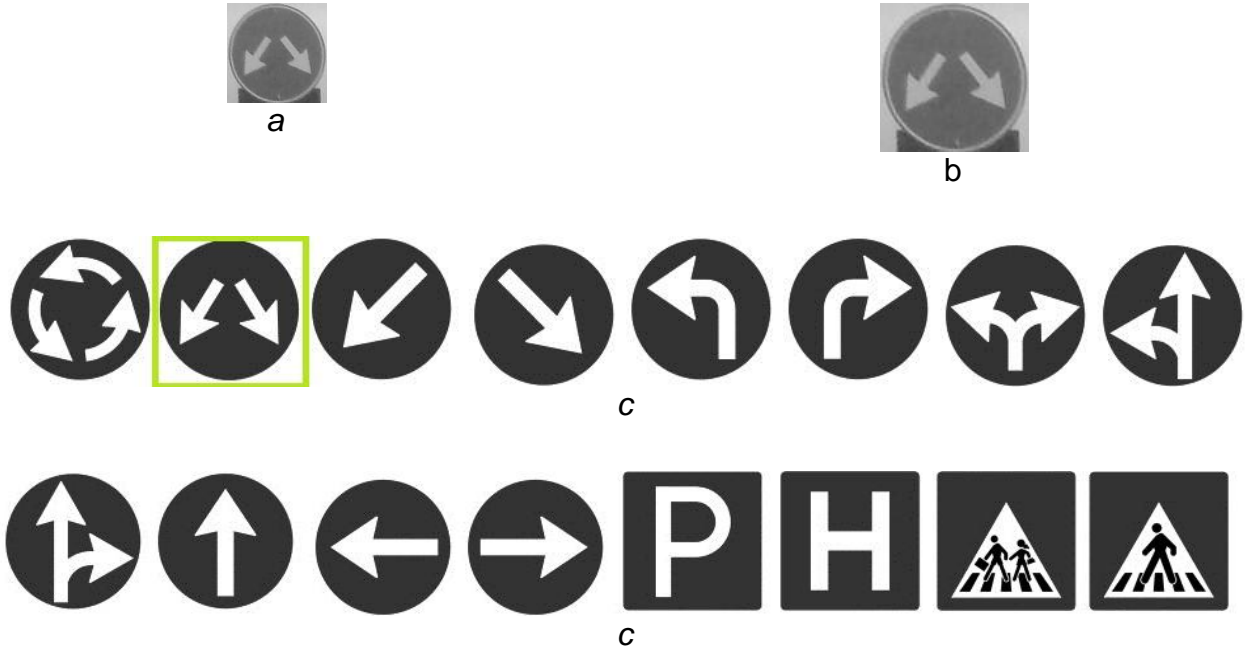
Şekil 5.17: Aynı levhanın dönüşümüyle elde edilen işaretler.

Korelasyon yöntemi görüntülerin üst üste çakıştırılmasıyla hesaplandığı için, tespit edilen trafik işaretinin boyutları önemlidir. Eşleştirilecek şablon kümesindeki işaretlerin boyutlarına getirilmeli ve sonrasında korelasyon uygulanmalıdır. Bu aşamada, şablon kümesindeki işaretlerin ayrı görüntü dosyalarında tutulması yerine, tek bir dosyada tutulması (Şekil 5.18c) ve bu dosya içerisinde çapraz korelasyon uygulanması eşleştirme süresini hızlandırmıştır.

Eşleşme yapılacak veritabanındaki hedef görüntüde korelasyon ölçüleri hesaplanırken, çapraz-korelasyon yönteminde bazı değişikliklere gidilmiştir. Çapraz korelasyonda hedef görüntüdeki bütün x,y koordinatları için çakıştırma yapılmaktadır. Fakat işaretlerin eşleştirilmesi için görüntü içinde sadece işaretlerin bulunduğu bölgelerde korelasyon hesaplanması yeterlidir.

Hedef görüntünün boyutları, çerçeve içerisinde tespit edilen trafik işaretlerinin boyutlarının ortalama değeri alınarak hesaplanmıştır. Öncelikle sahne içinden seçilen işaret, veritabanındaki işaretin boyutuna göre yeniden boyutlandırılır. En ve boy uzunlukları, hedef görüntünün en ve boy uzunluğuna eşitlenir (Şekil 5.18b). Böylece işaretlerin üst üste gelmesi sağlanır. Sonra işaretlerin üzerinde

kaydırılarak korelasyon ölçüleri hesaplanır (Şekil 5.18c). En yüksek korelasyon değerine sahip işaret ile eşleştirme gerçekleştirilir.



Şekil 5.18: (a) Tespit edilen işaret, (b) Şablon işaretlere uyacak şekilde yeniden boyutlandırılması, (c) Şablon kümesindeki işaretler ile karşılaştırılarak korelasyonun hesaplanması.

5.2.3. Tanınan İşaretlerin Sahneler Arası Denetimi

Trafik işaretleri, sahnede ilk tespit edildiği andan itibaren kameraya yakınlaştıkça farklı açılarda görüntülenmektedir. Bu durum bazı sahnelerde yanlış eşleştirme ile sonuçlanabilmektedir. Hatalı eşleşmeleri azaltmak için üst üste 2 sahnede tanınan işaret doğru olarak kabul edilerek kullanıcıya bildirim yapılmaktadır. Sahneler arası denetimi sağlamak için bir önceki sahnede eşleşen işaretin bilgisi tutulmakta ve işlenen sahnedeki eşleşme sonucu ile karşılaştırılmaktadır. Bu sayede yanlış eşleşmelerin sayısı azaltılmaktadır. Sahneler arası denetim olmadan, özellikle mavi trafik işaretlerinde SIFT yöntemi kullanılmadığından, sarmalayan kutu içerisindeki mavi renk içeren alanların elenmesi eşleme adımında kadar gerçekleştirilemiyordu. Sahneler arası denetim ile işaret olmayan bu alanların ya da nesnelerin elenmesi sağlanmıştır. Kırmızı renk içeren ve trafik işareti olmayan alanlar SIFT yöntemiyle çıkarılan nitelik sayılarının az olması sonucu elenebilmesine rağmen bazı işaretler yanlış eşlenebiliyordu. Fakat sahneler arası

denetim ile kırmızı renk içeren işaretlerin tanınmasında da başarı artırılmış, yanlış eşleme sayısı azaltılmıştır.

5.2.4. Tanınan İşaretlerin Geri Bildirimi

Kırmızı ve mavi ağırlıklı trafik işaretleri için eşleştirme ve denetim işlemi yapıldıktan sonra, eğer çerçeveden seçilen alanın trafik işareti olduğuna karar verilirse, işaretin türü yazılım arayüzü ile kullanıcıya gösterilir (Şekil 5.19).



Şekil 5.19: Tanınan trafik işaretinin geri bildirimi

5.3. Başarım

Geliştirilen yazılım, Türkiye karayollarında kaydedilmiş video görüntüleri üzerinde test edilmiştir. Veritabanındaki 46 farklı trafik işaretinin de bulunduğu bu görüntülerde, aynı sahne içinde birden fazla işaretin bulunduğu görüntüler de mevcuttur.

Toplamda 164 işaretin bulunduğu görüntülerde, RGB renk uzayı kullanılan yöntemde, 154 tanesi trafik işareti olarak tespit edilerek eşleştirme adımına gönderilmiş, %94'lük bir başarımla yakalanmıştır. Trafik işareti olmamasına rağmen işaret adayı olarak belirlenen alanların elenmesi eşleştirme adımında ve sahneler arası denetimde gerçekleştirilmiştir. Trafik işaretinin tespit edilmesinde HSV renk uzayı da denenmiş fakat %86'lık bir başarımla yakalanmıştır. Doğru tespit başarımının RGB renk uzayına göre düşük olmasının yanında, işaret dışındaki alanların elenmesinde de HSV renk uzayı başarısız olmaktadır. İşaret tespitinde insan gözünün renk algısına daha yakın sonuç veren CIE L*a*b renk uzayı üzerinde çalışılmış, ancak kullanılan kameraya özgü ölçümlere değerlerine sahip olunamadığı için, bu renk uzayı gerçekleştirilememiştir.

Eşleştirme adımındaki tanıma başarımını değerlendirmek için Çizelge 5.2'deki olasılık matrisi (*contingency matrix*) kullanılacaktır. Öncelikle Çizelge 5.1'deki hücrelerin anlamları açıklanacaktır:

- DP (doğru pozitif): İşareti doğru tanıdı.
- YP (yanlış pozitif): İşareti yanlış tanıdı.
- YN (yanlış negatif): İşareti yanlışlıkla işaret değil gibi algıladı.
- DN (doğru negatif): İşaret olmadığını doğru anladı.

Çizelge 5.1: Olasılık matrisi

			<i>Toplam</i>
	DP	YP	DP+YP
	YN	DN	YN+DN
<i>Toplam</i>	DP+YN	YP+DN	DP+YP+YN+DN

Çizelge 5.2: İşaret tanımada kullanılan yöntemin olasılık matrisi

			<i>Toplam</i>
	139	15	154
	10	362	372
<i>Toplam</i>	149	377	526

Geliştirilen yöntemin kaydedilen video görüntüleri kullanılarak denenmesi sonucunda elde edilen değerler aşağıdaki gibidir:

- a) Duyarlılık (*sensitivity*): $DP / (DP + YN) = \% 93$
- b) Özgüllük (*specificity*): $DN / (DN + YP) = \% 96$
- c) Genel doğruluk (*accuracy*): $(DP + DN) / (DP + DN + YP + YN) = \% 95$

Özgüllük değerinin %100 olması, hiçbir trafik işaretinin yanlış bir şekilde tanınmadığı anlamına gelmektedir. Duyarlılık değerinin %100 olması, tüm trafik işareti olmayan nesnelerin doğru bir şekilde reddedildiği anlamına gelmektedir.

Çizelge 5.3: Belirli hızlarda giden taşıttaki sürücünün düşünme süresi içerisinde taşıtın kat ettiği mesafe [37]

Taşıt hızı (km/h)	Düşünme mesafesi (metre)
32	6
48	9
64	12
80	15
96	18
112	21

Geliştirilen yöntem ile hareket halindeki taşıt içerisinde kaydedilen görüntülerdeki trafik işaretleri, yaklaşık 50 metre mesafeden doğru bir şekilde tanınabilmektedir. Sürüş standartları biriminin (DSA) yaptığı araştırmaya göre [37], sürücünün belirli hızlarda giden taşıt içerisinde, düşünüp karar verdiğinde taşıtın kat ettiği mesafe Çizelge 5.3'te verilmiştir. Bu bilgiler dikkate alındığında, trafik işaretlerinin tanındığı

mesafe, sürücüye trafik işaretini yorumlayıp karar vermesi için yeterli zamanı sağlayabilecek düzeydedir.

Trafik işaretlerinin tanınmasında karşılaşılan zorluklar şunlardır:

1. **Işıklandırma koşulları:** Işıklandırma, özellikle günün belirli saatlerinde değişiklik göstermektedir ve dış etmen olduğu için kontrol edilememektedir. Özellikle yansımalarda ve gölgede kalmış işaretlerin tanınmasında sorun yaşanmıştır (Şekil 5.20a).
2. **Diğer nesnelerin işaretleri kapatması:** İşaretlerin önündeki ağaç, direk, araba, vb. gibi nesneler, trafik işaretlerinin kısmi olarak kapanmasına yol açmaktadır (Şekil 5.20b). Bu durum işaretin tek bir sarmalayan kutu içine alınmasını zorlaştırmıştır. Kırmızı işaretlerde, işaretin bir kısmı gözüküyorsa ama tek parça halinde görüntülenebiliyorsa, nitelik çıkarım yöntemi en fazla benzerlik ile doğru eşleştirme yapabilmektedir. Ancak mavi işaretlerde başarımlar düşmektedir.
3. **Uygun olmayan biçimdeki işaretler:** Karayollarında, resmi kurumlardan alınan şablonlara uygun olmayan çok sayıda işaret mevcuttur (Şekil 5.20c). Kural dışı bu işaretlerin bütün olasılıkları için bir model geliştirmek imkânsızdır. Benzer sorun renk konusunda da yaşanmaktadır. Özellikle işaretlerdeki mavi renklerin, siyaha yaklaşan koyulukta olanlarını tespit etmekte sorun yaşanmıştır (Şekil 5.20d). Ayrıca eskimiş ve zarar görmüş levhaların tanınmasında başarımlar düşmektedir.



Şekil 5.20: Tanınamayan işaretlere örnekler

Yazılımın bir görüntüyü işleme hızı, sahne içerisindeki trafik işareti sayısına bağlı olarak 0.04 ile 0.15 saniye arasında değişmektedir. Bu şekilde sistem, saniyede yaklaşık 10 sahne işleyebilecek duruma getirilmiştir. Hızı belirleyen temel adımlar ve bu adımların test verilerinden elde edilen ortalama süreleri Çizelge 5.4'te verilmiştir.

Çizelge 5.4: Hızı belirleyen temel adımların ortalama hesaplama süreleri

Sahneden trafik işaretinin tespiti		~0.0750 sn
Kırmızı trafik işareti	SIFT nitelik çıkarımı	~0.0080 sn
	Nitelikleri eşleştirme ve işareti tanıma	~0.0068 sn
Mavi trafik işareti	Korelasyonları hesaplama ve işareti tanıma	~0.0160 sn

6. SONUÇ

Tez kapsamında geliştirilen yazılım, taşıt içinden çekilmiş video görüntülerini kullanarak, sahne içindeki trafik işaretlerini tespit edebilmekte, yüksek bir başarımla veritabanındaki 46 işaret arasından doğru eşleştirebilmektedir. MATLAB ortamında geliştirilen sistemin hızı, saniyede 10 çerçeve işleyebilecek durumda olup, bu tür sistemler için önemli bir özellik olan gerçek-zamanlı tanımayı da destekleyebilir.

Trafik işaretlerinin sahne içinden tespitinde renk-temelli yaklaşım uygun görülmüştür. CIE L*a*b renk uzayının gerçekleştirilmesi amaçlanmış, ancak kullanılan kameraya özgü ölçümlene değerlerine sahip olunamadığı için, bu renk uzayı kullanılamamıştır. RGB renk uzayı kullanılarak yapılan çalışmalarda, HSV uzayına göre daha yüksek başarımla sağlanmış ve bu nedenle RGB renk uzayının kullanılmasına karar verilmiştir. Bu sayede sahne içindeki işaretlerin konumu %94 başarımla tespit edilmiştir. İşaretlerin eşleştirilmesinde ise, renk bilgileri kullanılarak tarama yapılacak küme daraltılmıştır. Eşleştirme adımında, işaretlerin niteliğine göre birden fazla yöntem kullanılması sağlanmıştır. Çerçeveler arası bilgi aktarımı ile eşleştirme başarımla artırılmış ve %95 genel doğruluk oranına erişilmiştir.

Kırmızı ağırlıklı renk içeren işaretler için SIFT yöntemiyle nitelikler çıkarılmış ve veritabanındaki kırmızı ağırlıklı işaretlerle eşleştirilmiştir. Bu sayede ölçklemeden ve dönüşümden bağımsız olarak doğru işaret ile eşleştirilme sağlanmıştır. Trafik işaretlerindeki bozulmalardan dolayı işaretin bir kısmının sarmalayan kutu içine alınması veya aynı renkteki bitişik cisimlerden dolayı işaretin geniş bir kutu içine alınması gibi durumlarda da, SIFT yönteminin başarılı sonuç verdiği gözlemlenmiştir. Özellikle karayollarında sıklıkla rastlanan, eğri biçimde konumlandırılmış işaretlerin aynı başarımla tanınması bu yöntem ile mümkündür.

Dönme açısı ile birlikte anlamı değişen işaretlerden oluşan ve benzer niteliklere sahip olan mavi renkteki trafik işaretleri ile kırmızı/mavi renkleri içeren iki işaret, şablon-eşleştirme yöntemi kullanılarak eşleştirilmiştir. SIFT yönteminin dönme açısından bağımsız olması nedeniyle, mavi işaretlerde bu yöntemin uygun olmadığı anlaşılmıştır.

Gelecekteki alıřmalarda gerekleřtirdiĐimiz yazılıma ek olarak, iřaretin ilk tespit edildiĐi erevede eřleřtirme yapılmadan, sonraki erevelerde iřaretin tahmini yeni konumuna gre tarama yaparak tespit alanı dřrlebilir ve yaklařan iřaretin daha kaliteli grnts ile eřleřtirme bařarımı ve sistemin hızı arttırılabilir. Ayrıca, tasarladığımız yntemin OpenCV [34] gibi ortamda gerekleřtirilmesi halinde hız bařarımı saniyede 10 kareden daha yukarıya ekilebilecektir.

İřaretin sahne iinde tespitinde ise farklı ıřıklandırma ve hava řartlarında da, insan gznn rengi algıladıĐı řekilde algılanmasını saĐlayan CIECAM97 [35] renk grnm modeli kullanılarak, trafik iřareti daha kolay ayırt edilebilir. Bu renk uzayında renk eřikleme iřlemi uygulanabilirse, geriye oĐunlukla trafik iřareti kalacaĐından, grltlerin yok edilmesine ynelik morfolojik iřlemlerin sresi azalır ve sistem hızı arttırılır. Bu renk grnm modelinin kullanılması, yaĐmurlu ve karlı hava řartlarında, hatta gece kořullarında bile iřaretlerin tanınmasına olanak saĐlayabilir. Fakat, bu amala kullanılacak kameralara iliřkin bazı lmleme (*calibration*) bilgilerine sahip olunması ve modelin bu bilgilere gre kurulması gerekir.







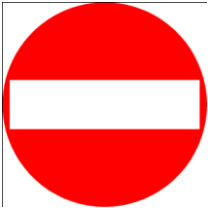











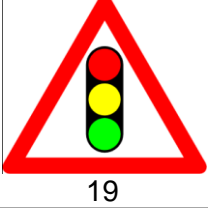






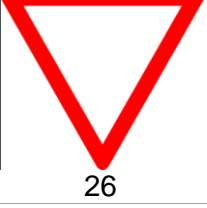




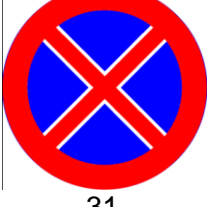
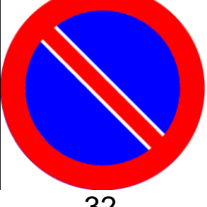

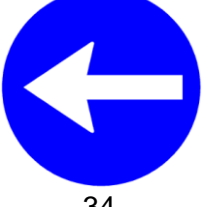



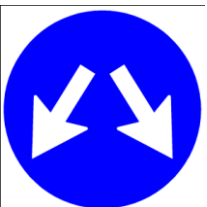
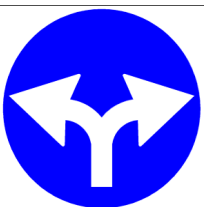
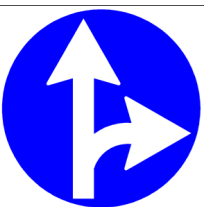
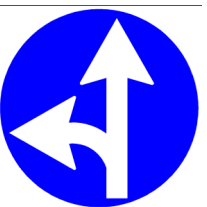
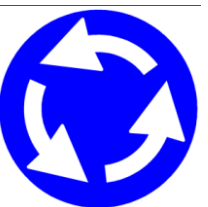
KAYNAKLAR

- [1] Internet: http://en.wikipedia.org/wiki/Adaptive_cruise_control
- [2] Arnoul, P., Viala, M., Guerin, J.P., Mergy, M., 1996, Traffic signs localisation for highways inventory from a video camera on board a moving collection van, Intelligent Vehicles Symposium, IEEE September.
- [3] Gao, X., Shetsova, N., Hong, K., Batty, S., Podladchikova, L., Golovan, A., Shaposhnikov, D., Gusakova, V., 2002, Vision models based identification of traffic signs.
- [4] Gao, X., Shetsova, N., Hong, K., Podladchikova, L., Shaposhnikov, D., 2006, Recognition of traffic signs based on their colour and shape features extracted using human vision models, J. Vis. Commun. Image R. 17. pp. 675-685.
- [5] Miura, J., Kanda, T, Shirai, Y., 2000, An active vision system for real-time traffic sign recognition, Proc. 2000 IEEE Int. Conf. on Intelligent Transportation Systems, pp 52-57.
- [6] Torresen, J., Jorgen, W.B., Sekanina, L., 2004, Efficient recognition of speed limit signs.
- [7] Broggi, A., Cerri, P., Medici, P., Porta, P.P., Ghisio, G., 2007, Real time road signs recognition, Proceedings of the 2007 IEEE Intelligent Vehicles Symposium, pp. 981-986.
- [8] Fang, C.Y., Chen, S.W., Fuh, C.S., 2003, Road-sign detection and tracking, IEEE Transactions Vehicular Technology, Vol. 52, No. 5, pp. 1329-1341.
- [9] Escalera, A., Armingol, J.M., Mata, M. 2003, Traffic sign recognition and analysis for intelligent vehicles, Image and Vision Computing 21, pp. 247-258.
- [10] Gonzalez, R. C., Woods, R. E., 2008, Digital Image Processing 3rd Edition, Prentice Hall.
- [11] Gonzalez, R. C., Woods, R. E., 2002, Digital Image Processing 2nd Edition, Prentice Hall.
- [12] Internet: http://en.wikipedia.org/wiki/CIE_1931_color_space
- [13] Yılmaz, İ., 2002, Renk sistemleri, renk uzayları ve dönüşümler, Selçuk Üniversitesi Jeodezi ve Fotogrametri Mühendisliği Öğretiminde 30. Yıl Sempozyumu.
- [14] Serra, J., 1988, Image Analysis and Mathematical Morphology, Volume 2: Theoretical Advances, ISBN 0-12-637241-1
- [15] Lewis, J.P., 1995, Fast normalized cross-correlation, Vision Interface.

- [16] Briechle, K., Hanebeck, U.D., 2001, Template matching using fast normalized cross correlation, Proceedings of SPIE.
- [17] Yilmaz, A., Javed, O., Shah, M., 2006, Object tracking: A survey. ACM Comput. Surv. 38, 4 (Dec. 2006), 13.
- [18] Lowe, D.G., 2004, Distinctive image features from scale-invariant keypoints, International Journal of Computer Vision, 60, 2, pp. 91-110.
- [19] Harris, C., Stephens, M., 1988, A combined corner and edge detector, Proceedings of the 4th Alvey Vision Conference: pp 147-151
- [20] Tomasi, C., Kanade, T., 1991, Detection and tracking of point features. Technical Report CMU-CS-91-132, Carnegie Mellon University, Pittsburgh, PA.
- [21] Mikolajczyk, K., Schmid, C., 2005, A performance evaluation of local descriptors, IEEE Transactions on Pattern Analysis and Machine Intelligence, 10, 27, pp 1615-1630.
- [22] Moravec, H., 1980, Obstacle avoidance and navigation in the real world by a seeing robot rover, Tech Report CMU-RI-TR-3, Carnegie-Mellon University, Robotics Institute.
- [23] Brown, M., Lowe, D.G., 2002, Invariant features from interest point groups. In British Machine Vision Conference, Cardiff, Wales, pp. 656-665.
- [24] Bay, H., Ess, A., Tuytelaars, T., Van Gool, L., 2008, SURF: Speeded up robust features, Computer Vision and Image Understanding (CVIU), Vol. 110, No. 3, pp. 346-359.
- [25] Internet: http://en.wikipedia.org/wiki/Euclidean_distance
- [26] Internet: <http://lwneuralnetplus.sourceforge.net>
- [27] Internet: <http://www.cs.unc.edu/~welch/kalman/>
- [28] Gao, X., Passmore, P., Hong, K., Podladchikova, L., Shaposhnikov, D., 2008, Colour vision model-based approach for segmentation of traffic signs, EURASIP Journal on Image and Video Processing, ID:386705.
- [29] Rybak, I. A., Guskova, V. I., Golovan, A. V., Podladchikova, L. N., Shevtsova, N. A., 2005, Attention-guided recognition based on “what” and “where” representations: A behavioral model. Neurobiology of Attention (Eds. Itti, L., Rees, G. and Tsotsos, J.). Elsevier Acad. Press, pp. 663-670.
- [30] Bahlmann, C., Zhu, Y., Ramesh, V., Pellkofer, M., Koehler, T., 2005, A system for traffic sign detection, tracking and recognition using color, shape and motion information.

- [31] Alefs, B., Eschemann, G., Ramoser, H., Beleznai, C., 2007, Road sign detection from edge orientation histograms, Proceedings of the 2007 IEEE Intelligent Vehicles Symposium, pp. 993-998.
- [32] Internet:MATLAB, <http://www.mathworks.com>
- [33] Internet: <http://www.cs.ubc.ca/~lowe/keypoints/>
- [34] Internet: <http://sourceforge.net/projects/opencvlibrary/>
- [35] Internet: <http://scien.stanford.edu/pages/labsite/1998/psych221/projects/98/ciecam/matlab/imp.html>
- [36] Gonzalez, R. C., Woods, E., 2004, Digital Image Processing Using Matlab, Prentice Hall.
- [37] Internet: <http://www.dsa.gov.uk/>

EK-1 YAZILIM TARAFINDAN TANINAN TRAFİK İŞARETLERİ

 1	 2	 3	 4	 5	 6
 7	 8	 9	 10	 11	 12
 13	 14	 15	 16	 17	 18
 19	 20	 21	 22	 23	 24
 25	 26	 27	 28	 29	 30
 31	 32	 33	 34	 35	 36
 37	 38	 39	 40	 41	 42



43



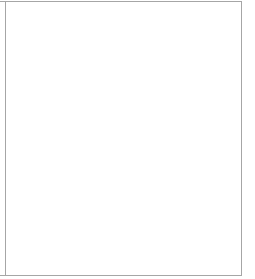
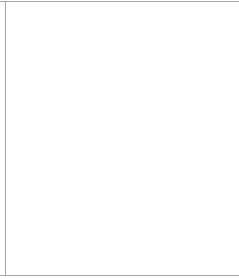
44



45



46



EK-2 TRAFİK İŞARETLERİNİN KULLANIM SIKLIKLARI

İşaret no	İşaretin adı	Kullanım sıklığı (%)
1	DUR	0,47
2	AZAMİ HIZ SINIRLAMASI (20)	-
3	AZAMİ HIZ SINIRLAMASI (30)	2,94
4	AZAMİ HIZ SINIRLAMASI (50)	2,26
5	AZAMİ HIZ SINIRLAMASI (70)	-
6	AZAMİ HIZ SINIRLAMASI (90)	-
7	GİRİŞİ OLMAYAN YOL	3,93
8	SAĞA DÖNÜLMEZ	1,71
9	SOLA DÖNÜLMEZ	3,71
10	U DÖNÜŞÜ YAPILMAZ	7,60
11	SESLİ İKAZ CİHAZLARININ KULLANIMI YASAKTIR	-
12	SAĞA TEHLİKELİ VİRAJ	2,13
13	SOLA TEHLİKELİ VİRAJ	2,52
14	SAĞA TEHLİKELİ DEVAMLİ VİRAJLAR	0,51
15	SOLA TEHLİKELİ DEVAMLİ VİRAJLAR	0,43
16	KASISLI YOL	1,02
17	YAYA GEÇİDİ	4,27
18	YOLDA ÇALIŞMA	0,34
19	İŞIKLI İŞARET CİHAZI	2,05
20	DİKKAT	0,73
21	KONTROLSÜZ KAVŞAK	-
22	ANA YOL- TALİ YOL KAVŞAĞI	2,94
23	ANA YOL- TALİ YOL KAVŞAĞI	1,62
24	ANA YOL- TALİ YOL KAVŞAĞI	0,73
25	ADA ETRAFINDA DÖNÜNÜZ	0,26
26	YOL VER	7,04
27	İKİ YÖNLÜ TRAFİK	0,55
28	SAĞDAN DARALAN KAPLAMA	0,21
29	SOLDAN DARALAN KAPLAMA	0,09
30	HER İKİ TARAFTAN DARALAN KAPLAMA	0,09
31	DURAKLAMAK VE PARK ETMEK YASAKTIR	3,63
32	PARK ETMEK YASAKTIR	12,76

33	SAĞA MECBURİ YÖN	8,41
34	SOLA MECBURİ YÖN	
35	İLERİ MECBURİ YÖN	
36	SAĞDAN GİDİNİZ	
37	SOLDAN GİDİNİZ	
38	HER İKİ YANDAN GİDİNİZ	1,54
39	SAĞA VE SOLA MECBURİ YÖN	0,68
40	İLERİ VE SAĞA MECBURİ YÖN	3,03
41	İLERİ VE SOLA MECBURİ YÖN	7,30
42	DÖNEL KAVŞAK	0,26
43	YAYA GEÇİDİ	-
44	OKUL GEÇİDİ	-
45	HASTANE	-
46	PARK YERİ	1,92

ÖZGEÇMİŞ

Adı Soyadı : Hamdi Yalın Yalıç

Doğum Yeri : Ankara

Doğum Yılı : 1984

Medeni Hali : Bekar

Eğitim ve Akademik Durumu:

Lise 1996-2002 Milli Piyango Anadolu Lisesi

Lisans 2002-2007 Çankaya Üniversitesi Bilgisayar Mühendisliği Bölümü

Yabancı Dil: İngilizce

İş Tecrübesi:

2007-... Hacettepe Üniversitesi Bilgisayar Mühendisliği Bölümü
Araştırma Görevlisi