# BENZETİM ORTAMINDA VE GERÇEK ZAMANLI OLARAK TRAFİK İŞARETLERİNİN TESPİTİ

# DETECTION OF TRAFFIC SIGNS IN SIMULATION AND REAL TIME ENVIRONMENT

**JALE NUR MERTOĞLU**

**Dr. MEHMET DEMİRER**
**Tez Danışmanı**

Hacettepe Üniversitesi
Lisansüstü Eğitim – Öğretim ve Sınav Yönetmeliğinin
Elektrik ve Elektronik Mühendisliği Anabilim Dalı İçin Öngördüğü
YÜKSEK LİSANS TEZİ olarak hazırlanmıştır.

2017

**Jale Nur Mertoğlu** 'nun hazırladığı "**Benzetim Ortamında ve Gerçek Zamanlı Olarak Trafik İşaretlerinin Tespiti** "adlı bu çalışma aşağıdaki jüri tarafından **ELEKTRİK ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM Dalı**'nda **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Prof. Dr. Selçuk GEÇİM
Başkan .....

Dr. Mehmet DEMİRER
Danışman .....

Doç. Dr. Harun ARTUNER
Üye .....

Doç. Dr. Umut SEZEN
Üye .....

Doç. Dr. Ali Ziya ALKAR
Üye .....

Bu tez Hacettepe Üniversitesi Fen Bilimleri Enstitüsü tarafından YÜKSEK LİSANS TEZİ olarak kabul edilmiştir.

Prof. Dr. Menemşe GÜMÜŞDERELİOĞLU
Fen Bilimleri Enstitüsü Müdürü

***Canım aileme...***

## YAYINLAMA VE FİKRİ MÜLKİYET HAKLARI BEYANI

Enstitü tarafından onaylanan lisansüstü tezimin/raporumun tamamını veya herhangi bir kısmını, basılı (kağıt) ve elektronik formatta arşivleme ve aşağıda verilen koşullarla kullanıma açma iznini Hacettepe üniversitesine verdiğimi bildiririm. Bu izinle Üniversiteye verilen kullanım hakları dışındaki tüm fikri mülkiyet haklarım bende kalacak, tezimin tamamının ya da bir bölümünün gelecekteki çalışmalarda (makale, kitap, lisans ve patent vb.) kullanım hakları bana ait olacaktır.

Tezin kendi orijinal çalışmam olduğunu, başkalarının haklarını ihlal etmediğimi ve tezimin tek yetkili sahibi olduğumu beyan ve taahhüt ederim. Tezimde yer alan telif hakkı bulunan ve sahiplerinden yazılı izin alınarak kullanması zorunlu metinlerin yazılı izin alarak kullandığımı ve istenildiğinde suretlerini Üniversiteye teslim etmeyi taahhüt ederim.

☑ **Tezimin/Raporumun tamamı dünya çapında erişime açılabilir ve bir kısmı veya tamamının fotokopisi alınabilir.**
(Bu seçenekle teziniz arama motorlarında indekslenebilecek, daha sonra tezinizin erişim statüsünün değiştirilmesini talep etseniz ve kütüphane bu talebinizi yerine getirse bile, tezinin arama motorlarının önbelleklerinde kalmaya devam edebilecektir.)

☐ **Tezimin/Raporumun ..... tarihine kadar erişime açılmasını ve fotokopi alınmasını (İç Kapak, Özet, İçindekiler ve Kaynakça hariç) istemiyorum.**
(Bu sürenin sonunda uzatma için başvuruda bulunmadığım taktirde, tezimin/raporumun tamamı her yerden erişime açılabilir, kaynak gösterilmek şartıyla bir kısmı ve ya tamamının fotokopisi alınabilir)

☐ **Tezimin/Raporumun ..... tarihine kadar erişime açılmasını istemiyorum, ancak kaynak gösterilmek şartıyla bir kısmı veya tamamının fotokopisinin alınmasını onaylıyorum.**

☐ **Serbest Seçenek/Yazarın Seçimi**

22 / 05 / 2017

*(İmza)*

*Öğrencinin Adı Soyadı*
Jale NERTOĞLU

# ETİK

Hacettepe Üniversitesi Fen Bilimleri Enstitüsü, tez yazım kurallarına uygun olarak hazırladığım bu tez çalışmasında;

- Tez içindeki bütün bilgi ve belgeleri akademik kurallar çerçevesinde elde ettiğimi,
- Görsel, işitsel ve yazılı tüm bilgi ve sonuçları bilimsel ahlak kurallarına uygun olarak sunduğumu,
- Başkalarının eserlerinden yararlanılması durumunda ilgili eserlere bilimsel normlara uygun olarak atıfta bulunduğumu,
- Atıfta bulunduğum eserlerin tümünü kaynak olarak gösterdiğimi,
- Kullanılan verilerde herhangi bir tahrifat yapmadığımı,
- Ve bu tezin herhangi bir bölümünü bu üniversite veya başka bir üniversitede başka bir tez çalışması olarak sunmadığımı

Beyan ederim.

22/05/2017

Jale Nur MERTOĞLU

# ÖZET

# BENZETİM ORTAMINDA VE GERÇEK ZAMANLI OLARAK TRAFİK İŞARETLERİNİN TESPİTİ

**JALE NUR MERTOĞLU**

**Yüksek Lisans, Elektrik Elektronik Mühendisliği Bölümü**

**Tez Danışmanı: Dr. Mehmet DEMİRER**

**Nisan 2017, 105 sayfa**

Trafik işaretlerinin sürüş esnasında tespiti, sürücünün güvenliğini sağlaması açısından ve araçlarda oluşabilecek maddi hasarları önlemesi açısından önem arz etmektedir. Bu çalışma kapsamında ise bütün trafik işaretlerini tanıyabilecek ve algılayabilecek bir sistem önce benzetim ortamında sunulmuş daha sonra uygulama olarak hem benzetim ortamında hem gerçek zamanlı olarak kasis işareti levhasının tespiti gerçekleştirilmiştir. Uygulamada kasis işaretinin belirli bir mesafede tespit edilmesi neticesinde kullanıcıya sesli uyarı verdirilerek belirli bir süre önce sürücünün hızını azaltması beklenmektedir. Büyük otomobil üretici firmalarının bu sistemler ile donattıkları araçlar yüksek fiyatlarda alıcı bulabilmektedir. Bu tez çalışması ile hedeflenen, bu sistemlerin çok ucuz maliyetler ile ülkemiz sınırlarının içerisinde üretilebilmesidir. Ayrıca hali hazırda araçlar içerisinde kullanılan sistemler şu an için sadece hız işareti tabelalarının içerisindeki sayısal değerin algılanması prensibi ile çalışmaktadır. Bu tez kapsamında ise şekil tabanlı olarak bütün trafik işaretlerini tanıyabilecek bir sistem önerilmiş ve sonuçları irdelenmiştir. Algoritmalar sistemin gerçek zamanlı performansı da düşünülerek geliştirilmiştir ve çeşitli testler ile sistemin başarımı sınanmıştır.

**Anahtar Kelimeler:** Trafik işareti tespiti, kasis işareti tespiti, şekil tabanlı tespit

# ABSTRACT

## DETECTION OF TRAFFIC SIGNS IN SIMULATION AND REAL TIME ENVIRONMENTS

**JALE NUR MERTOĞLU**

**Master of Science, Department of Electrical Electronics Engineering**

**Supervisor: Dr. Mehmet DEMİRER**

**April 2017, 105 pages**

Detection of traffic signs during driving is important in terms of ensuring the safety of the driver and preventing material damage that may occur in vehicles. In the scope of this study, a system that can recognize and detect all traffic signals was presented in the simulation environment first and then the application of kasis marking plate was performed both in real time and in simulation environment. In practice, it is expected to reduce the speed of the driver for a certain period of time by giving an audible warning to the user in the event that the beep sign is detected at a certain distance. The cars that big carmakers make with these systems find buyers at high prices. The aim of this thesis is to produce these systems within the boundaries of our country with very cheap costs. In addition, the systems already in use in the vehicles currently operate only with the principle of perceiving the numerical value in the speed markers. In this thesis, a system which can recognize all traffic signs with shape based is proposed and its results are examined. Algorithms have been developed with real-time performance of the system in mind, and the performance of the system has been tested with various tests.

**Keywords:** Traffic sign detection, speed bump detection, shape based detection

# TEŞEKKÜR

Bu tezin çalışmasında sağladığı katkılardan ötürü tez danışmanım Dr. Mehmet Demirer'e teşekkürlerimi sunarım. Benden sevgilerini esirgemeyen aileme ve nişanlım Tuna Orhanlı'ya verdikleri desteklerden dolayı sonsuz teşekkürlerimi sunarım.

# İÇİNDEKİLER TABLOSU

Sayfa

# ŞEKİLLER

Sayfa

# ÇİZELGELER

Sayfa

# SİMGELER VE KISALTMALAR

## Kısaltmalar

| | |
|---|---|
| fps | Frame Per Second |
| sn | Saniye |
| dk | Dakika |
| km/s | Kilometre / Saat |

# 1. GİRİŞ

Trafik işareti tanıma sistemleri günümüzde sürücü güvenliğinin sağlanması açısından çok önemlidir. Bu trafik işaretleri yollardaki araç akışını düzenledikleri gibi aynı zamanda sürücüleri tehlikeli durumlar için uyarmak için de geliştirilmişlerdir. Her ne kadar bu trafik işaretleri sürücüler ve yayalar için hayati öneme sahip olsa da çok sayıda trafik işaretinin sürüş esnasında dikkat ile takip edilmesi birtakım sürüş problemlerine yol açabilmektedir. Yapılan araştırmaların sonucuna göre İngiltere'de sürücülerden bütün trafik işaretlerini tam anlamıyla doğru bilenlerin sayısı çok azdır, ayrıca sürücülerin neredeyse % 46'sı trafik işaretlerini takip etmeye çalıştıklarında dikkatlerinin dağıldığını bildirmişlerdir. Bir diğer ilginç taraf ise İngiltere Ulaşım Bakanlığı'nın verilerine göre İngiltere de 9000 adet yanlış yorumlanan trafik işareti bulunmaktadır [1]. Yapılan trafik kazalarının bazılarından sonra sürücüler kazaya sebebiyet olarak trafik işaretlerini takip etmek zorunda kaldıklarını bildirmişlerdir. Sürücünün bu aşamada otomatik olarak çeşitli trafik işaretlerini tanıyabilen bir sisteme gereksinimi ortaya çıkmaktadır. Bu gereksinimi karşılamak üzere yollarda sürücülere çeşitli elektronik bileşenli destek sağlayan sistemler ortaya çıkmıştır. Bu sistemler çeşitli bilgisayarlı görü algoritmaları kullanarak sürücüye yardım etmeyi amaçlar. Bu tip sürücü yardım sistemlerine örnek olarak, trafik işareti tanıma sistemi, şerit değiştirme uyarı sistemi, ani fren uyarı sistemi, otomatik park sistemi, yorgunluk belirleyici sistem verilebilir [2]. Bu sistemlerin hepsi bilgisayarlı görüntü algoritmalarını kullanarak sürücüye yardım etmektedir fakat bu sistemlerin takıldığı araçlar genelde lüks sınıfındadır ve araçların satış maliyetleri çok yüksektir. Sistemler detaylı incelendiğinde genelde yüksek çözünürlüklü ve saniyede yüksek sayıda çerçeve alabilen bir endüstriyel kamera, yüksek hafızaya ve yüksek çalışma hızına sahip görüntü çipleri ile desteklenmiş bir ana çip, çeşitli bağlantı aparatları ve uygun bir bilgisayarlı görü algoritması karşımıza çıkmaktadır [3]. Günümüzde artık lüks ve orta sınıf birçok araçta belirli hız limitlerini kullanıcıya sesli olarak bildiren yardımcı sistemler ile karşılaşılmaktadır. Bu sistemlerin birçoğu hız limiti işaretinin içerisindeki rakamları tanımaya yöneliktir ve bütün trafik işaretlerinin her çeşidini tanıyıp sürücüye uyarı verecek bir sistem henüz bulunmamaktadır. Bunun nedenleri arasında başta hafıza kısıtı ve kullanılacak işlemcinin maliyeti gelir. Ayrıca yanlış alarm (False Alarm) olarak nitelendirilen durumları varlığı da kabul edilemez sonuçlara yol açabilmektedir. Örneğin araç içi kamera ile 1 saat boyunca alınmış bir video (24 fps) görüntüsünü

düşünürsek toplam sürede 86400 adet çerçeve elde edilir. Ve her üç dakikada bir trafik işareti varlığı kabul edilirse ve bir işaretin 40 çerçeveyi kapladığı düşünülürse işareti anlatan toplam çerçeve sayısı 1 saat boyunca 800 olmaktadır. Bu bağlamda 800 adet işaret olan çerçeve varsa 85600 adet işaret olmayan çerçeve bulunmaktadır. Bu oran ise yanlış alarm verilme ihtimalini çok kuvvetlendirmektedir [4]. Kullanılan görüntü sensörlerinin maliyetleri arttıkça belirleme ve algılama işlemlerinin performansları artmaktadır fakat bu seferde algoritmanın işlem yükü çok artmakta ve işlemcinin gücü yetmeyebilmektedir.

Bu tezde ise önce bütün trafik işaretlerini tanıyabilecek bir sistem üzerinde yoğunlaşılmıştır daha sonra uygulama olarak sadece trafik işaretlerinden kasis işareti seçilmiştir. Sadece bu işaretin sürüş esnasında gerçek zamanlı olarak tespit edilmesi ve sürücüye uyarı verdirilmesi hedeflenmiştir. Kasis işaretleri Türkiye şartlarında otoban ve şehirlerarası yollar aracı her türlü yolda fazlaca miktarda bulunmaktadır. Genelde aşina olunan yollarda seyir halinde olan sürücüler kasis tabelasına bakma ihitiyacı hissetmeden kasisten önce yavaşlayabilmektedirler. Yola aşina olmayan bir sürücü ise ilk defa geçeceği bir yolda kasis işaretini anlık olarak farkedemeyip ya kasisten çok şiddetli bir geçiş yapacaktır ya da çok ani bir fren yaparak anlık yavaşlama gerçekleştirecektir. Bu tip durumlar hem sürücü için, hem araç içerisindeki yolcu için hemde yollardaki yayalar için ciddi anlamda bir risk teşkil etmektedir. Ayrıca yapılan bu ani frenler veya kasislerden şiddetli geçişlerde araçlarda bulunan mekanik sistemler önemli ölçüde hasar görmektedir. Eğer sürücüye kasis işaretine belirli bir mesafe varken sesli uyarı verdirilebilirse bu sayede sürücü ani frenlerden veya sert geçişlerden kurtulabilmektedir.

Trafik işaretlerini tanımak için literatürde yapılan çalışmalar incelendiğinde üç farklı yöntem göze çarpmaktadır. Bunlardan ilki Renk Tabanlı Yöntemler, ikincisi Şekil Tabanlı Yötemler ve son olarak Makine Öğrenmesi Tabanlı Yöntemlerdir [4]. Renk tabanlı Yöntemler genelde yazarların farklı renk uzaylarında çalışması ve farklı eşik değeri belirleme yöntemleri kullanması ile kendi içerisinde çeşitlenmektedir. Bu yönteme, Lopez ve Fuentes'in yaptığı çalışma örnek olarak verilebilir. Bu çalışmada yazarlar alınan resmi CIE Lab renk uzayına taşırlar. Bu renk uzayının bileşenlerinden L açıklık-koyuluk değerini, a kırmızı-yeşil değerini, b ise sarı-mavi değerini, göstermektedir. LAB renk uzayı, gözümüzün görebildiği renklerden de fazlasını kapsayan bir renk uzayıdır. Bu çalışma ile renk tabanlı algoritmaların performansı ölçülmüştür. Çalışmada saniyede 20 çerçeve alabilen bir kamera ile görüntüler

toplanmıştır ama görüntülerin işlenmesi gerçek zamanlı olarak yapılmamıştır. Bu şekilde veritabanı üzerinden işaret belirleme yapılmış ve % 97 lik bir başarım elde edilmiştir [5]. Şekil tabanlı tespit yöntemlerinde ise alınan görüntülerdeki tabela renkerinden bağımsız olarak çalışılır ve tabelaların şekilsel karakteristik özellikleri incelenir. Şekil tabanlı algoritmalar renklerdeki ayırt ediciliği kullanmadıkları için performansları düşük olarak düşünülebilmektedir. Ancak şekil tabanlı algoritmaların ışık koşullarından renk tabanlı algoritmalar gibi etkilenmemesi büyük bir avantajdır. Son yöntem olarak Makine Öğrenmesi yöntemlerinde ise alınan görüntüler genelde filtrelendikten sonra sınıflandırılırlar. Bu yöntemlerin başlıcaları Sinir Ağları, Derin Öğrenme ve Destek Vektör Makinalarıdır [6, 7, 8]. Makine öğrenme yöntemlerin de benzetim ortamında tespit ve algılama oranları yüksektir fakat veritabanı oluşturmanın her bir işaret için çok güç olması ve algoritmaların işlem yüklerinin fazla olması gerçek zamanlı çalışmalarda bir kısıt oluşturduğu açıktır [9].

Tezin ana amacı bütün trafik işaretlerini tanıyabilecek bir sistemin tasarlanmasıdır. Bunun için önce belirleme işlemi ve algılama işlemi şekil tabanlı olarak planlanmış ve uygulanmıştır. Uygulama olarak en son amaç ise düşük maliyetli ve ülkemiz şartlarında rahatlıkla üretilebilen gerçek zamanlı bir kasis işareti tanıma sisteminin tasarlanmasıdır. Bu sistemlerin gerçek zamanlı çalışabilmesi için öncelikle benzetim ortamında bir takım performans ve işlem yükü testlerinin gerçekleştirilmiş olması gerekmektedir. Ayrıca benzetim ortamında geliştirilen algoritmalar performans ve işlem yükü açısından incelendiğinde gerçek zamanlı algoritmada hangi bilgisayarlı görü yöntemlerinin kullanılacağına karar verilmiştir. Bu bağlamda ilk olarak bütün işaretleri şekillerine bakarak sınıflandıracak bir yöntem incelenmiştir. Trafik işaretlerinin üçgen, daire, kare vs olmasının belirlenmesinin ardından işaretlerin iç kısmının belirlenmesi ve algılanması işlemi gerçekleştirilmiştir. Bu yöntemler kapsamlıca anlatıldıktan sonra iki adet özgün benzetim ortamında kasis işareti tanıma sistemi daha sonra gerçek zamanlı kasis işareti tanıma sistemi uygulama olarak yapılmış ve sonuçları paylaşılmıştır. Bu yöntemler ayrı ayrı kapsamlı bir şekilde ele alınmıştır performans sonuçları hem şekiller ile hem de istatistiki veriler ile sunulmuştur.

Bölüm 2'de benzetim ortamında bütün trafik işaretlerini tanıyacak bir sistem üzerinde bir takım algoritmalar geliştirilmiştir. Bu bölümde yapılan belirleme ve algılama işlemi tamamen şekillerin belirli ayırıcı özniteliklerinin çıkarılması ve bu öznitelikler üzerinden bir kural tabanlı yapı oluşturulması prensibine dayanmaktadır. Bu bölümde

önce trafik işaretlerinin dış kısmındaki üçgen daire kare vs. şekillerin belirlenmesi gerçekleştirilmiş daha sonra içteki işareti tanıyabilecek bir yöntem önerilmiştir.

Bölüm 3'de ise Bölüm 2'de önerilen yöntemin bir uygulaması ele alınmıştır. Bu bağlamda benzetim ortamında tasarlanan kasis işareti belirleme sistemi kapsamlıca ele alınmıştır ve sonuçları hem şekiller ile hem de istatistiki olarak verilmiştir. Bu bölümde kırmızı renk bulma algoritması ve Otsu dinamik eşik değeri yönteminin kullanılması konuları özellikle vurgulanmıştır. Bölüm 4'de yine benzetim ortamında geliştirilen bir algoritma ele alınmıştır. Bu algoritma ilk benzetim yönteminden tamamen farklı bir yöntem ile geliştirilmiştir. Bölüm 5'de ise benzetim ortamında geliştirilen yöntemlerde edinilen bilgiler harmanlanarak gerçek zamanlı olarak çalışabilen bir sistem geliştirilmiş ve başarım testleri yapılmıştır. Bölüm 6'da ise sonuçlar ve yorumlar bulunmaktadır.

# 2. TRAFİK İŞARETLERİNİN TANINMASI VE ALGILANMASI

Tezin ilk bölümünde bütün trafik işaretlerini tanıyabilen bir yöntem üzerinde durulmuştur. Bu yöntem benzetim ortamında çalışabilecek şekilde geliştirilmiştir. Geliştirilen algoritma gerçek zamanlı performans beklentileri göz ardı edilerek Matlab ortamında sınanmıştır. Türkiye Cumhuriyeti Karayolları yönetmeliklerine göre trafik işaretleri Tehlike Uyarı İşaretleri, Trafik Tanzim İşaretleri, Bilgi İşaretleri, Durma ve Parketme İşaretleri ve Yatay işaretleme olarak sınıflandırılmıştır. Bu tezde özellikle üzerinde yoğunlaşılan bölüm Tehlike Uyarı İşaretleri ve Trafik Tanzim İşaretleridir çünkü bu iki sınıfta bulunan işaretler günlük hayatta sürekli karşılaşılan kırmızı üçgen ve kırmızı daire içerisinde bulunan farklı uyarı levhalarıdır. Bu bilgiler doğrultusunda şekilleri doğru sınıflandıran bir yöntem üzerinde durulmuştur.

## 2.1. Giriş

Bu bölümde bütün üçgen levhaları simgeleyecek bir veritabanı oluşturulmuş ve çalışmalar bu veritabanı üzerinden yürütülmüştür. Oluşturulan veritabanı 90 adet resimden oluşmaktadır. Her bir resim yine tezin ilk bölümlerinde kullanılan aynı cep telefonu kamerası ile çekilmiştir. Veritabanında çeşitli şekillerde üçgen levha içi trafik işaretleri vardır. Bunların bazıları aşağıda listelenmiştir.

-Vahşi hayvanlar geçebilir,
-Sağa tehlikeli viraj,
-Sola tehlikeli viraj,
-Okul geçidi,
-Tehlikeli eğim iniş,
-Tehlikeli eğim çıkış,
-Dikkat,
-Kontrolsüz kavşak vb.

Bu trafik işaretlerin hepsi kırmızı üçgen işaretçi içerisindedir ve her bir işaret siyah iç işaretçi ile ifade edilmektedir.

## 2.2. Kırmızı Üçgen İçi Trafik İşaretlerinin Belirlenmesi

Akıcılığın sağlanması ve yöntemin kolay anlaşılabilir olması için Şekil 1'de dört adet kırmızı üçgen içi trafik işaretleri örnek teşkil etmesi açısından verilmiştir.


a)


b)


c)


d)

Şekil 1. Dört Farklı Orijinal Resim

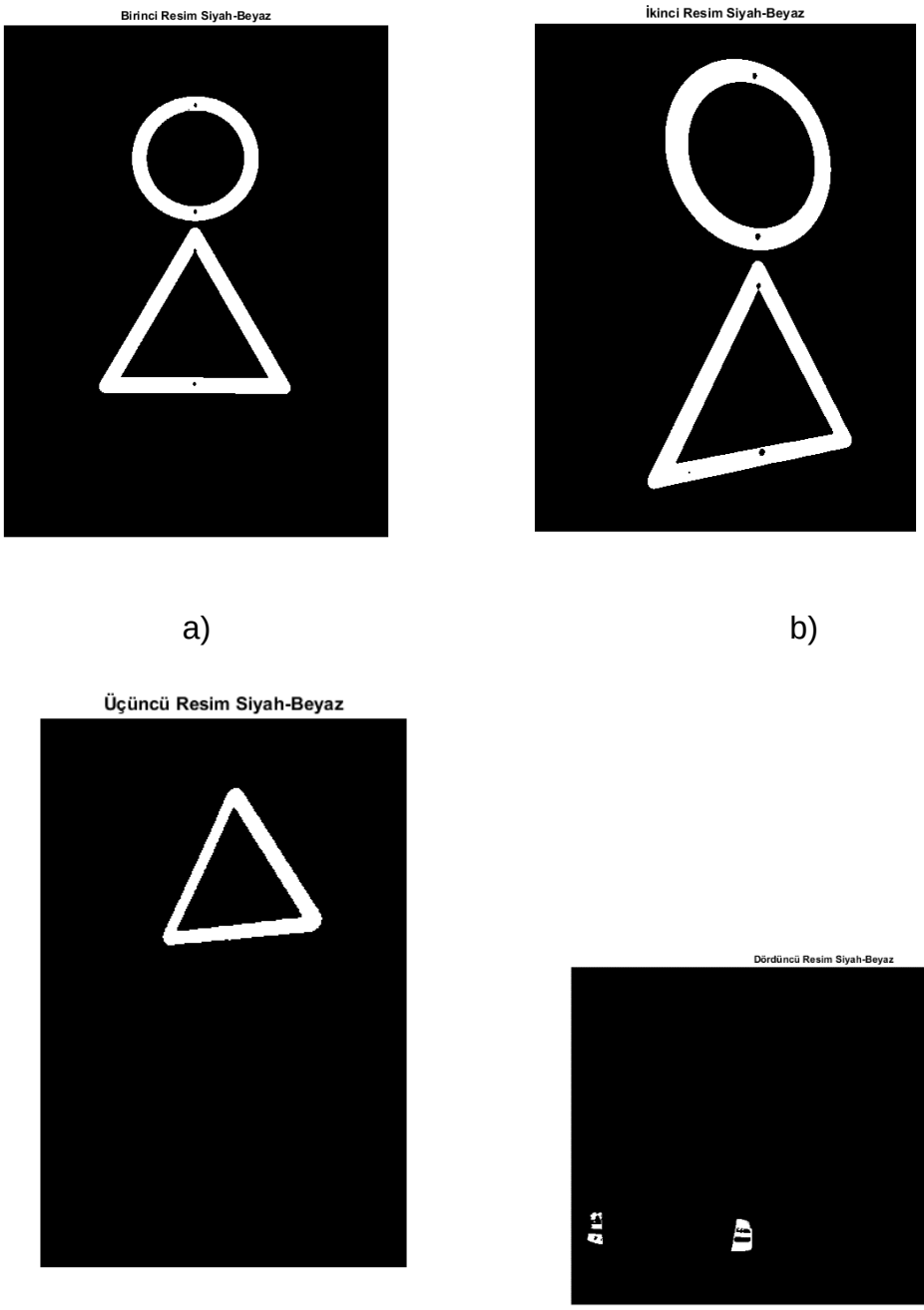
Trafik işareti tanıma sistemini iki ana ayrı sistem olarak düşünürsek ilk kısım dış işaretin belirlenmesi ikinci kısım ise iç işaretçinin algılanmasıdır. Belirleme safhasını kırmızı rengin ön plana çıkarılması ve üçgen veya daire şeklin tespit edilmesi oluştururken, algılama safhasını ise belirlenen kırmızı üçgen levha içerisindeki iç işaretçinin algılanması oluşturur.

Bu kırmızı üçgen veya daire bölümün belirlenmesi için geliştirilen yöntemler önce tek bir resim üzerinde denemiş daha sonra veritabanındaki bütün resimler için

çalışabilecek hale dönüştürülmüştür. İlk olarak renk uzayları etraflıca ele alınmış ve her bir renk uzayının özellikleri incelenmiştir. Literatürde trafik işareti tanıma için çeşitli renk uzaylarında (RGB, YCbCr, HSV) uygun yöntemin belirlenebilmesi için çeşitli çalışmalar gerçekleştirilmiştir [10]. Uygun renk uzayına doğru karar verilebilmek hem tanıma hem de algılama performansına doğrudan etki etmektedir ve işlem yükünü azaltmaktadır [11].

Bu resimler üzerinde ilk uygulanacak işlem kırmızı rengin ön plana çıkarılması ve daha sonra ikili görüntü sistemine dönüşümdür. Resimler ileriki bölümlerde tekrar kapsamlıca ele alndığı gibi kırmızı altuzay ile gri altuzay farkının alınması ve Otsu Otomatik Eşik Değerleme ile ikili sisteme dönüşüme tabii tutulmuştur. Çeşitli otomatik eşik değeri bulma yöntemlerinden Otsu yöntemi bu aşamada kullanılmıştır ve basit olarak Otsu metodu, gri seviye görüntüler üzerinde uygulanabilen bir eşik değeri tespit yöntemidir. Bu yöntem kullanılırken görüntünün siyah ve beyaz olmak üzere iki renk sınıfından oluştuğu varsayımı yapılır. Daha sonra tüm eşik değerleri için bu iki renk sınıfının sınıf içi varyans değeri hesaplanır. Bu değerin en küçük olmasını sağlayan eşik değeri, bulunabilen en iyi eşik değeridir. Tek boyutta yapılan bu çalışma çok boyutta yapılan Fisher Lineer Ayırıcı Fonksiyon yönteminin tek boyuta indirgenmesi olarak düşünülebilir [12].

Sınıf içi varyans değeri minimum değerinde iken sınıflar arası varyans değeri maximum değerinde olur. Yöntem gri seviye görüntüler üzerinde çalışır ve sadece renklerin görüntü üzerinde kaçar defa bulunduğuna bakar. Bu yüzden önce görüntünün renk histogramı hesaplanır ve tüm işlemler histogram dizisi üzerinde yapılır [13]. Veri setinde bulunan 90 tane farklı koşullarda çekilmiş resimlerin hepsinde Otsu Yöntemi başarı ile eşik değerlerini tespit etmiştir [14]. Hiç birisinde kırmızı üçgen daire yok olmamıştır ve amaçlanan duruma ulaşılmıştır. Bir sonraki aşamada bahsedilen bu fark resmi siyah beyaz resme dönüştürülmüştür. İşlem yükünü azaltmak için bulunan üçgen levhanın etrafına bir kare çizilecek şekilde sınırlar belirlenmiş ve siyah beyaz ana resim bu sınırlardan kesilmiştir. İkili sayı sistemine dönüşüm tamamlandıktan sonra istenmeyen gürültülerin atılması için alan filtresi ve eliptiklik filtresi kullanılmıştır. Tabelanın en yakında ve en uzakta bulunduğu aralıklar belirlenmiş ve bu aralıkların dışında kalan bütün nesneler gürültü olarak değerlendirilmiş ve atılmıştır. Şekil 2'de dört resim için bulunan siyah beyaz dönüşüm resimleri ve alan ile eliptiklik filtresi uygulanmış sonuçlar verilmiştir.



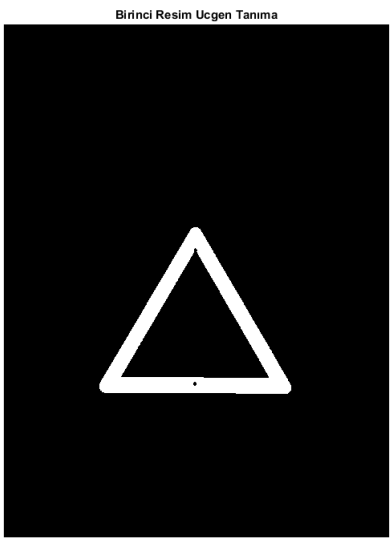

Şekil 2. Dört Farklı Resim için Siyah-Beyaz Dönüşümler

Geliştirilen yöntemin bundan sonraki kısmı şekillerin başarı ile tanınması için anlatılmıştır.

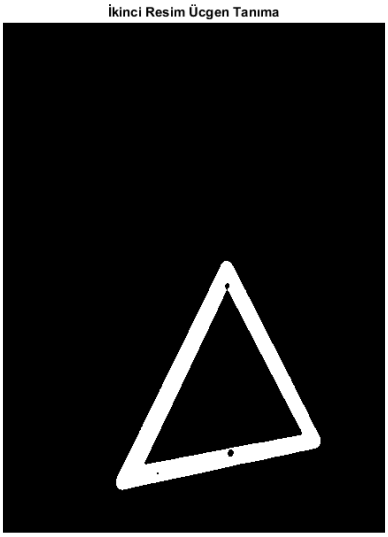
## 2.3. Trafik İşaretlerinin Algılanması Ve Sınıflandırma

Bu bölümde artık nesnelerin geometrik şekilleri incelenecektir. Bu işlem için nesnelerin geometrik boyutlarını hesaba katarak şekillerini belirleyebilen bir yöntem geliştirilmiştir. Şekil faktörü, nümerik olarak bir partikül veya nesnenin şeklini

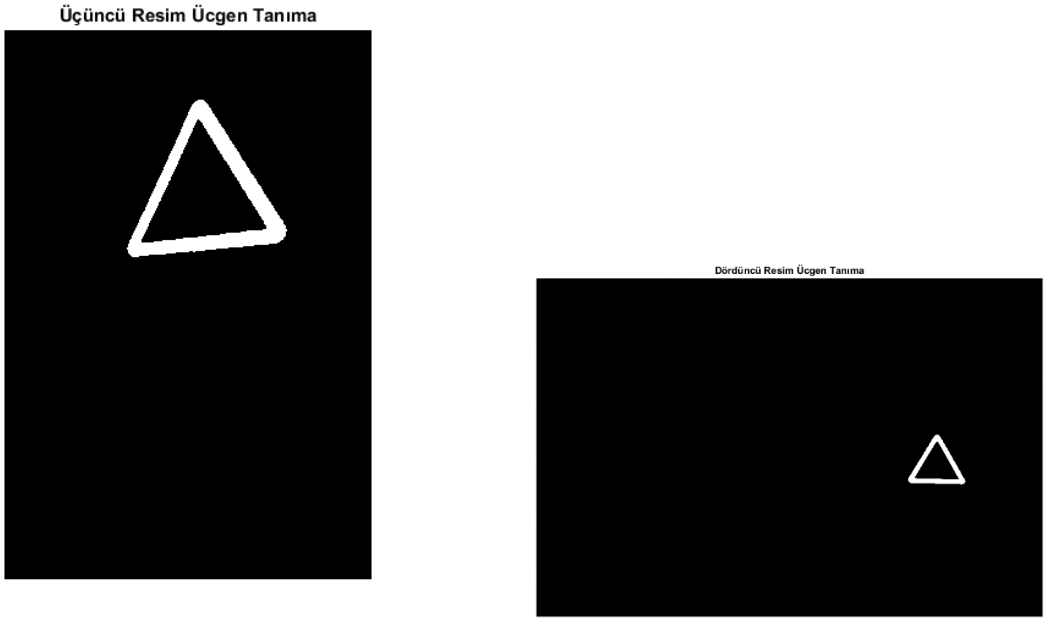
büyüklüğünden bağımsız olarak belirlemek için kullanılır. Burada kritik olan nokta bu yöntemin belirlenen nesnelerin boyutlarından bağımsız olarak çalışabilmesidir. Bu kritik nokta üçgen levhanın şeklini belirlemekte kilit rol oynamaktadır çünkü alınan resimlerin bazıları yakından bazıları uzaktan çekilmiş olabilmektedir. Dolayısı ile uzaklıktan yani büyüklükten etkilenmeyen bir algoritmanın gerekliliği açıktır. Bu bilgiler ışığında bulunan nesneler bir döngü içerisinde teker teker kontrol edilmiştir ve her bir nesne için en-boy oranı, dairesellik ve üçgen şekil faktörü değerleri tek tek hesaplanmıştır. Bulunan üçgen şekil faktörü 0.20 ile 0.25 aralığında çıkmıştır ve bu aralık ciddi derecede dar olduğu için gelen resimde kasis etrafındaki üçgen kadar kesin olmadıkça üçgene benzeyen bütün şekiller elenmektedir. Ayrıca yöntemin uzaklıktan da bağımsız olması önemli bir artıdır. Belirlenen eşik değeri uygulandığında bulunan nesnelerden sadece üçgen olanlar belirlenmiş diğerleri elenmiştir.




a)




b)




c) d)

Şekil 3. Dört Farklı Resimde Tespit Edilen Üçgenler

Üçgen şekiller belirlendikten sonra bir sonraki aşama üçgen içerisindeki birbirine bağlı siyah örüntülerin hangi trafik işaretine ait olduğunun algılanmasıdır. Bu işlem için yeni bir yöntem önerilmiştir. Bu önerilen yöntem üçgen işareti içerisindeki bütün trafik işaretlerini başarı ile tanıyabilmektedir. Sistemin başarı ile çalışabilmesi için tek gereken şart bazı değerlerin sisteme öğretilmesidir. Bu öğretim ise önden bilgilerin algoritmaya verilmesi ile gerçekleştirilir.

Üç farklı parametre algoritmaya girdi olarak verilir ve sistem kural tabanlı bir yapı ile gelen her bir resmi değerlendirmeye tabi tutar. Bu değerlendirme için önceden belirlenmiş eşik değerleri sisteme girdi olarak verilir. Bu belirleyici özelliklerin ilki kareselliktir ve bir nesnenin ne kadar kareye benzediğinin geometrik bir ölçüsüdür. Bu geometrik karesellik ölçüsü nesnenin etrafına çizilen çevreleyen kutunun en ve boy uzunluklarının hesaplanması prensibine dayanır.

$$A_r = \frac{d_{\min}}{d_{\max}} \qquad \ldots\ldots\ldots\ldots\ldots\ldots(1)$$

Eşitlik 1'deki oran aslında basit olarak en-boy oranı olarak düşünülebilir. Bu oran tam 1 veya 1 civarında olduğunda nesnenin kare olduğu çıkarımı yapılabilir. Oran 0'a yaklaştığında ise nesnenin uzun kenarı ile kısa kenarı arasındaki fark artacağı için karesel şekle olan benzerlik azalacaktır.

Bir diğer ayırt edici değişken ise nesnenin daireselliğidir ve tanımı Eşitlik 2'de verilmiştir. Bu şekil faktörü değeri de bulunan nesnenin etrafına çizilen dış çizimin daireye ne kadar benzediğini belirlemek için kullanılır.

$$f_{circ} = \frac{4\pi A}{P^2} \quad \text{……..……(2)}$$

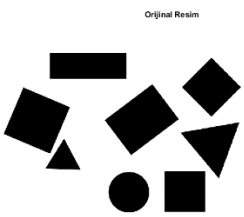
Eşitlikte A değeri nesnenin dış çiziminin alanını, P değeri ise çevresini göstermektedir. Bu oran 1'e yaklaştığında nesnenin daireselliğini göstermektedir. 0'a yaklaşması ise düz çizgiye ne kadar benzediğinin bir ölçüsüdür.

Son ayırt edici parametre ise nesnenin üçgenselliğidir ve tanımı Eşitlik 3'de verilmiştir.

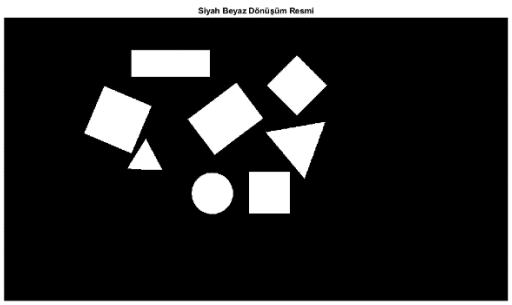
$$f_{triangular} = \frac{AF}{AB} \quad \text{…….(3)}$$

Bu oranda bulunan AF nesneyi dolduran toplam alanı AB ise etrafına çizilen çevreleyen kutunun alanıdır. Bu oran da 0.5'e yaklaştıkça nesnenin üçgene olan benzerliği artmaktadır. Bu bilgilerin üçü bir arada kullanılarak her nesneye özgü bir 3 elemanlı vektör oluşturulur ve kural tabanlı "eğer" yapısı ile algılama ve sınıflandırma gerçekleştirilir.

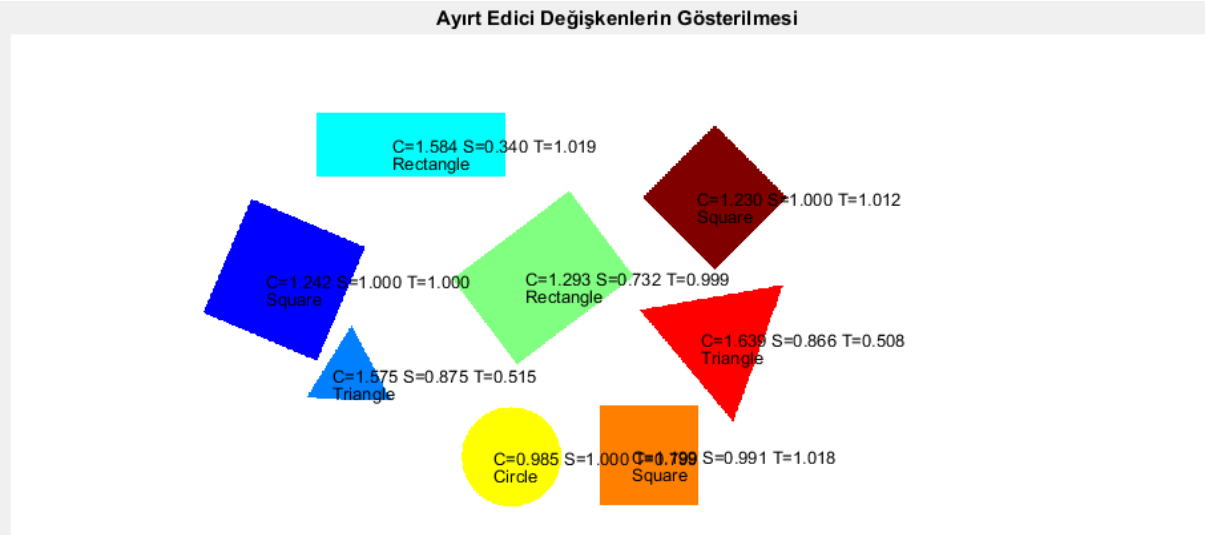
Bu sınıflandırmanın kolay anlaşılabilmesi için Şekil 4'te her bir geometrik şeklin şekil faktörü değerleri verilmiştir. Şekil 4.a'da orijinal resim, Şekil 4.b'de siyah beyaza renklerin ters çevrildiği resim ve Şekil 4.c'de ise belirlenen değerler gösterilmiştir.




a)




b)




c)

Şekil 4. a. Orijinal Resim b. Siyah-Beyaz Resim c. Ayırt Edici Değişkenlerin Değerleri

Daha sonra yine önceki bölümlerde anlatıldığı gibi buluna üçgen şekilden kesilmiş ve aynı boyutlarda oluşturulan bir maske boş matrise yapıştırılmıştır. Bu sayede çalışmalar artık sadece üçgen alan için gerçekleştirilecektir. Şekil 5'de bu dört resim gösterilmektedir.

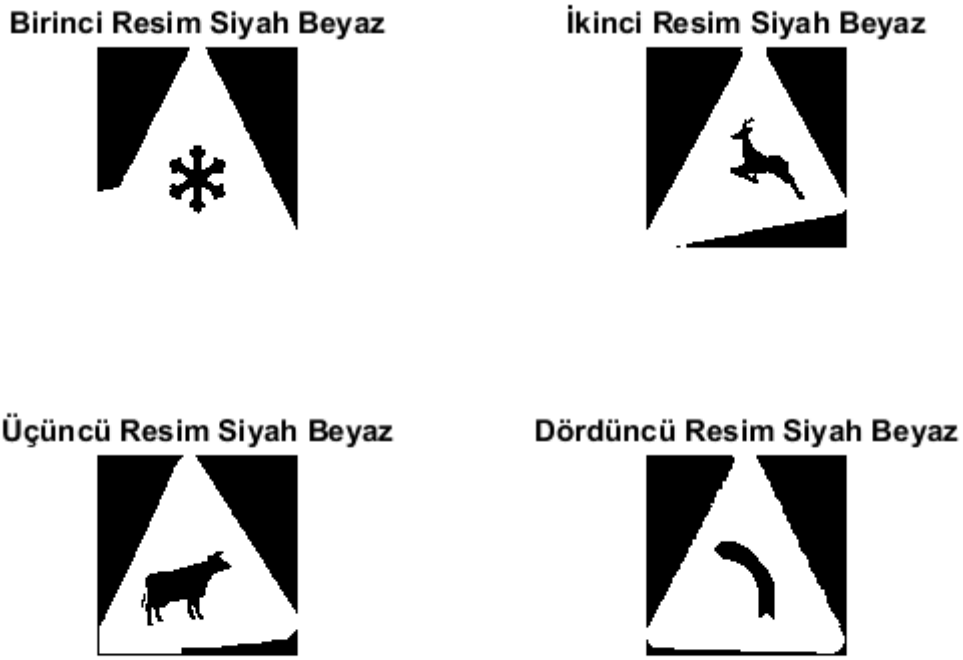

a)


b)


Şekil 5. Dört Farklı Resimde Üçgen Bölümlerin Çıkarılması

Daha sonra resim ileriki bölümlerde anlatılacağı üzere 100x100 boyutuna yeniden boyutlandırılmıştır. Bu sayede piksel ölçümü yapılırken uzaklık ve yakınlık bilgisinin algoritmanın gürbüzlüğüne olan etkisi en aza indirilmiştir.

Son aşamada ise algılama işlemine geçilmiştir ve bu algılama işlemi yine bu üç değişken üzerinden yürütülmüştür. İlk olarak 100x100 boyutuna yeniden ayarlanmış resimler önce kırmızı, yeşil ve mavi altuzaylara ayrılır. Daha sonra bu altuzaylardan yeşil ve mavi altuzaylar atılır kırmızı altuzay ile işleme devam edilir. Bu kırmızı altuzay yine Otsu Otomatik Eşik Değerleme yöntemi ile siyah beyaza dinamik eşik değerleri ile çevrilir. Burada önemli olan nokta bu işlemde eşik değerlerinin aralığının yeniden belirlenmesidir. Siyah beyaza dönüşüm sonrası elde edilen resimler Şekil 6'da gösterilmektedir. Bu aşamada Otsu yönteminin kullanılmasının nedeni sadece belirli resimler için değil algoritmanın bütün resimler için çalışmasının beklenmesidir. Bu aşamada artık algılama işlemi sınıflandırma prensibi ile gerçekleştirilmişir. Burada temel amaç bir veritabanı ile kıyaslama yapmadan sadece eldeki verilerden türetilen değişkenler ile sınıflandırma yapabilmektir. Bu kıyaslama nesnelerin dört farklı ayırt edici değişkenleri en başta elde edilerek gerçekleştirilmiştir. 100x100 piksel resimlerin toplam piksel sayısı 10000'dir. Alan filtresi uygulayarak üçgen dışı siyah alanlar atılmış ve sadece üçgen içine odaklanılmıştır.




Şekil 6. Dört Farklı Resimde Çıkarılan Üçgenlerin Siyah-Beyaz Dönüşümleri

Üçgen içerisindeki şekiller yaklaşık 370 ile 10500 piksel arasında tespit edilmiştir. Bu piksel değerleri alt ve üst değer olarak alındığında geri kalan bağlı örüntüler atılmıştır. Üçgen içerisindeki siyah şekillerin kullanılan ayırt edici özellikleri listelenmiştir.

Çizelge 1. Siyah Şekillerin Ayırt Edici Özellikleri

| İşaretler / Ayırt Edici Özellikler | Siyah Piksel Sayısı | Karesellik | Dairesellik | Üçgensellik |
|---|---|---|---|---|
| **Buzlanma Tehlikesi** | 597 | 1.05 | 1.07 | 0.98 |
| **Vahşi Hayvanlar Geçebilir** | 610 | 0.78 | 1.29 | 0.91 |
| **Ehli Hayvanlar Geçebilir** | 987 | 0.81 | 1.24 | 0.95 |
| **Sola Tehlikeli Viraj** | 440 | 0.47 | 1.60 | 0.88 |

Bu çizelgedeki değerler ile bir kural tabanlı yapı oluşturulduğunda dört trafik işareti de başarı ile sınıflandırılmıştır. Örneğin Siyah piksel sayısı 550 – 650 arasında, karesellik, dairesellik ve üçgensellik yaklaşık 1 civarında ise bulunan işaret "Buzlanma Tehlikesi" şeklinde sınıflandırma yapılmıştır. Bu yöntem ile veritabanındaki 90 adet resimden 71 tanesinde başarı ile trafik işaretleri tespit edilmiştir. Bu 90 resmin içerisinde toplam 12 farklı üçgen içi trafik işareti bulunmaktadır. 71 tanesinde başarı ile işaretin bulunması Çizelge 1'deki dört ayırt edici parametrenin hepsi bu 12 farklı işaret için bulunmuş yani tablo satırları genişletilmiştir. Bulunan işaretler Şekil 7'de gösterilmektedir.


a)


b)



c) d)

Şekil 7. Dört Farklı Resimde Bulunan İşaretler

Geliştirilen algoritma ile istenilen trafik işaretinin doğru parametreler yardımı ile tespit edilebildiği görülmektedir. Bu geliştirilen algoritma sadece üçgen levhalar ile bu tezde denenmiş olsa da üçgen işaretinin parametrelerinin yerine daire işaretlerinde parametreleri girilirse bu seferde daire işaretlerin içlerindeki trafik işaretleri tespit edilebilir. Bu işlemin doğruluğu Şekil 4'de görülebilmektedir. Yöntemin en önemli eksikliği dinamik olarak parametreleri kendinin öğrenmeyip kullanıcıdan girmesini beklemesidir. Bir diğer önemli eksiklik ise yöntem gerçek zamanlı performans kıstasları göz önünde bulundurulmayarak geliştirilmiştir. Bir sonraki bölümde sonuçlara yer verilmiştir.

# 3. OTOMATİK RENK EŞİK DEĞERİ TABANLI KASİS İŞARETİ TANIMA SİSTEMİ

Çalışmanın bu bölümünde benzetim tabanlı olarak Matlab arayüz yazılımı üzerinde kasis işaretinin tespiti ve algılanması gerçekleştirilmiştir. Bu bölümün ilk aşamasını renge göre bölütleme yani tabela üzerindeki kırmızı bölümün ön plana çıkartılması ve üçgen şeklin tespiti oluşturmaktadır. Daha sonra ise levha içerisindeki kırmızı bölümde bulunan kasis işareti morfolojik sinyal işleme teknikleri kullanılarak algılanmıştır. Kırmızı rengin belirlenmesi aşamasında özgün ve bu tez kapsamında geliştirilen bir yöntem kullanılmıştır ve bu siyah beyaz resme dönüştürme işleminde literatürde sıklıkla kullanılan Otsu Otomatik Eşik Değeri Belirleme yöntemi ile bulunan dinamik eşik değerleri kullanılmıştır. Siyah beyaza dönüşümü sağlanmış resim üzerinde biribirine bağlı örüntüler belirlenmiş ve bu örüntülerin belirleyici özellikleri irdelenmiştir. Elde edilen bu belirleyici özellikler sayesinde kural tabanlı bir yapı oluşturulmuş ve algılama aşaması tamamlanmıştır. Bu bölümde geliştirilen algoritma daha önceden oluşturulan veritabanı üzerinde denenmiş ve sonuçları raporlanmıştır. Daha sonra geliştirilen algoritmanın gürbüzlüğünü sınamak için yapay olarak çeşitli zor durumlar oluşturulmuş ve algoritma bu şartlarda da denemiştir. Bu bölümde yapılan çalışmaların hepsi benzetim ortamında gerçekleştirilmiş olup ilerde yapılmış olan gerçek zamanlı çalışmalara temel teşkil etmesi açısından çok önemlidir.

## 3.1. Kasis İşareti Belirleme

Kasis işareti tanıma sistemini iki ana ayrı sistem olarak düşünürsek ilk kısım işaretin belirlenmesi ikinci kısım ise işaretin algılanmasıdır. Sistemin tamamının blok diyagramı Ek-1'de verilmiştir. Belirleme safhasını kırmızı rengin ön plana çıkarılması ve üçgen şeklin tespit edilmesi oluştururken, algılama safhasını ise belirlenen kırmızı üçgen levha içerisindeki işaretin kasis olup olmadığının algılanması oluşturur. Bu kırmızı üçgen bölümün belirlenmesi için geliştirilen yöntemler önce tek bir resim üzerinde denemiş daha sonra veritabanındaki bütün resimler için çalışabilecek hale dönüştürülmüştür. İlk olarak renk uzayları etraflıca ele alınmış ve her bir renk uzayının özellikleri incelenmiştir. Literatürde trafik işareti tanıma için çeşitli renk uzaylarında (RGB, YCbCr, HSV) uygun yöntemin belirlenebilmesi için çeşitli çalışmalar gerçekleştirilmiştir [10]. Uygun renk uzayına doğru karar verilebilmek hem

tanıma hem de algılama performansına doğrudan etki etmektedir ve işlem yükünü azaltmaktadır [11]. Algoritmanın bütün resimler için çalışabilmesi sabit bir eşik değeri kullanımı yerine dinamik bir eşik değeri kullanımı gerekliliğini ortaya koymuştur.

### 3.1.1. Renk Uzayı Dönüşümü ve Kırmızı Üçgen Çıkarımı

Tabelanın dış sınırını oluşturan kırmızı üçgen alanın çıkarılması için kırmızı bölgeyi bölütlemek üzere bir yöntem geliştirilmiştir. Bu yöntemin uygulanmasının ilk aşamasını orijinal çerçeve olan kırmızı yeşil mavi altuzaylı resimden sadece kırmızı alt uzayın seçilmesi oluşturur. İkinci aşamada ise orijinal çerçeve gri tonlamalı resme hazır fonksiyonlar kullanılarak dönüştürülmüştür. Son olarak ise, bu iki kırmızı ve gri tonlamalı resimlerin birbirleri ile olan farkı alınmıştır ve sonuçta kırmızı rengin başarı ile bölütlendiği gözlemlenmiştir. Şekil 8.a orijinal resmi göstermek üzere Şekil 8.b, 8.c ve 8.d'de bu sonuçlar paylaşılmıştır.


a) b)



c) d)

Şekil 8. a. Orijinal Çerçeve, b. Gri Tonlamalı Resim c. Kırmızı Alt Uzay Resim, d. İki Resmin Farkı

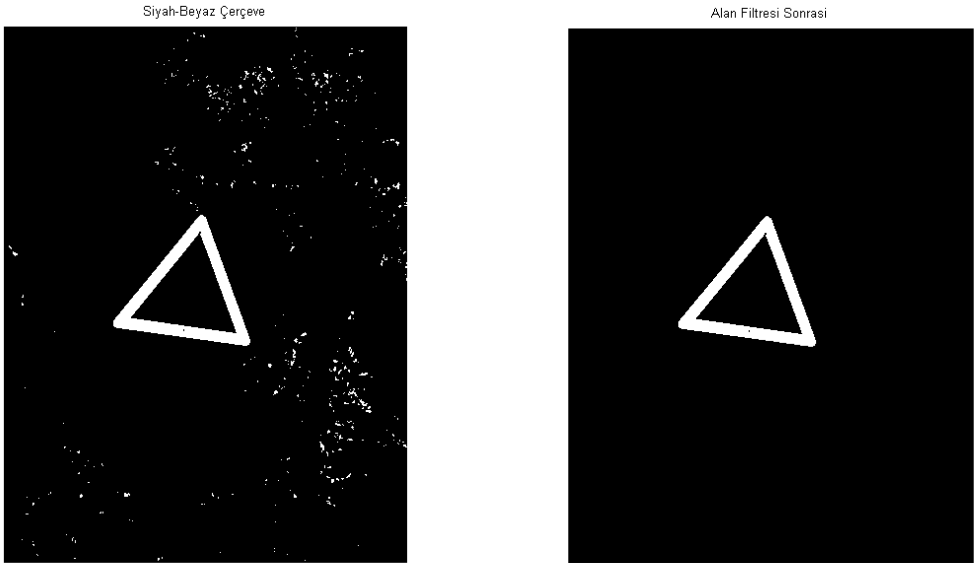
Kırmızı rengin ön plana çıkarıldığı resimde çeşitli görüntü işleme ve bilgisayarlı görü işlemlerinin gerçekleştirilmesi için resmin siyah beyaz resim formatına dönüştürülmesi gerekmektedir. Bu dönüşüm için gereken eşik değerinin ise sabit olması yöntemin sadece tek resimde çalışmasına yol açmaktadır. Dinamik eşik değerlerinin bulunabilmesi adına 20 adet resim için en uygun siyah beyaz dönüşüm eşik değerleri bir tabloya kaydedilmiştir. Bu eşik değerlerinin belirlenmesinde ilk olarak resimlerin renk histogram bilgileri kullanılmıştır. Daha sonra bu histogram bilgisine bakarak yapılan analizler sonucu en iyi şartlardaki eşik değeri ve en kötü şartlardaki eşik değerleri belirlenmiştir. Son olarak bulunan bu alt ve üst eşik değerleri bir fonksiyona girdi olarak verilecek ve gelen her yeni resim için bu aralıkta yeni bir uygun eşik değeri bulunacaktır. Çeşitli otomatik eşik değeri bulma yöntemlerinden Otsu yöntemi yine bu aşamada kullanılmıştır ve Bölüm 2'de detaylandırıldığı için burada kapsamlıca ele alınmamaıştır [12,13,14].

Bir sonraki aşamada bahsedilen bu fark resmi siyah beyaz resme dönüştürülmüştür. İşlem yükünü azaltmak için bulunan üçgen levhanın etrafına bir kare çizilecek şekilde sınırlar belirlenmiş ve siyah beyaz ana resim bu sınırlardan kesilmiştir. Bu işlemlerin detayı ve elde edilen bu kırmızı üçgen levhayı bulunduran

resme çeşitli alan ve eliptiklik filtrelerinin uygulanması bir sonraki bölümde anlatılmıştır

### 3.1.2. Morfolojik Sinyal İşleme

Alınan çerçeveler siyah beyaza dönüştürüldükten sonra her resimde gürültüler oluşmaktadır. Bu gürültüler seçilen dinamik eşik değerine göre bazı resimlerde büyük ölçüde bazı resimlerde ise daha küçük ölçüde olmaktadır. Bu gürültüleri filtrelemek için gürültülerin karakteristik özellikleri ile üçgen levhanın karakteristik özellikleri karşılaştırılmış ve filtreleme yapılmıştır. Kullanılan filtrenin temel özelliği gürültüyü oluşturan bağlı örüntünün piksel sayısı cinsinden alanı ile üçgen bölümün piksel sayısı cinsinden alanlarının karşılaştırılmasıdır. Bu karşılaştırma sonucu bir alan eşik değeri kabaca belirlenmiş ve bu eşik değerin sayıca altında alan değerine sahip bütün bağlı örüntüler gürültü olarak değerlendirilmiş ve filtrelenmiştir. Örüntülerin piksel cinsinden alan değerine göre filtreleme yapılması bilgisayarlı görü çalışmalarında sıklıkla tercih edilmektedir. Bu sürecin çıktıları Şekil 9.a ve 9.b'de gösterilmektedir.




a) b)

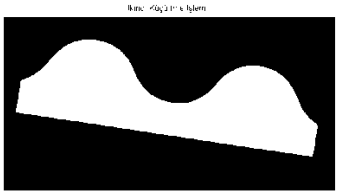
Şekil 9. Siyah Beyaz Resim, b. Filtre Sonrası Resim

Alan filtresi uygulandıktan sonra kalan bağlı örüntülerin bazıları üçgen şekli bazıları ise alan olarak yüksek değere sahip kırmızı gürültü objeleridir. Bunları da filtreleyerek sadece kırmızı üçgeni ortaya çıkarmak için bir başka filtreleme tekniği uygulanmıştır.

Bu teknik ise bağlı örüntülerin eliptiklik değerlerinin belirli bir eşik değeri ile kıyaslanması mantığı ile çalışır. Eliptiklik değeri nesnenin daireye olan yakınlığını belirleyen bir ölçüdür ve 0 ile 1 arasında ondalıklı değerler alır. Bu filtreleme sonucu bulunan nesne bütün resimlerde üçgen bölümdür ve ana resimden bu bölüm çıkartılarak işlem yükü azaltılmak istenmektedir. Üzerinde çalışılan orijinal çerçeve boyutundan her azalma işlem yükünü hafiflettiği için performansı artırmaktadır. Şekil 9.b' de bulunan üçgen levha gösterilmektedir. Bu üçgen levha etrafına çizilen bir çevreleyen kutu sayesinde üçgeni çevreleyen dikdörtgenin koordinatları belirlenmiştir. Bu koordinatlardan kesme işlemi gerçekleştirilmiş ve boyut 1201x1201 piksel boyutuna indirilmiştir (Şekil 10.a). Bu 1201x1201 piksel boyutundaki resimde kasis bölümü ön plana çıkarmak için siyah rengin bölütlenmesi gerekmektedir. Bu işlem için resim önce kırmızı, yeşil, mavi altuzaya ayrılmış ve içlerinden sadece kırmızı alt uzay alınmıştır. Bu kırmızı alt uzay ise siyah beyaza ortalama bir eşik değeri ile dönüştürülmüş ve sadece kasis simgesinin olduğu bölümün ön plana çıkarılması hedeflenmiştir. Kırmızı üçgenin içinde sadece siyah ve beyaz renklerinin olması sayesinde tek bir dönüşüm ile siyah renkli kasis bölümü kolaylıkla ortaya çıkarılmıştır. Yine merkezi bulunan kasis işaretinin etrafına çizdirilen dikdörtgen yardımı ile matris boyutu 271x524 piksel boyutuna düşürülmüştür. Bir sonraki bölümde ise bulunan siyah simgenin gerçekten kasis olup olmadığını algılamak için ele alınan yöntem etraflıca ele alınmıştır (Şekil 10.b, Şekil 10.c ve Şekil 10.d).


a)


b)

c) d)

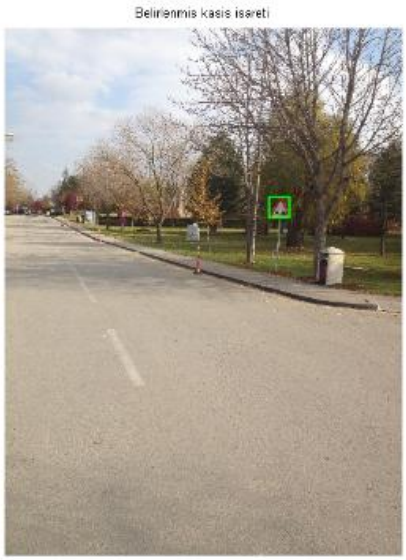
Şekil 10. Orijinal Resim, b. Kırmızı Alt Uzay Küçültülmüş Resim, c. Siyah Beyaz Küçültülmüş Resim, d. Belirlenmiş Kasis İşareti

### 3.2. Kasis İşareti Algılama

Bir önceki bölümde yapılan belirleme işleminden sonra ortaya çıkan siyah bölümün hangi işareti temsil ettiği sorusu ortaya çıkmıştır. Yine ilgili bağlı örüntünün karakteristik özellikleri incelenmiştir fakat piksel sayısı cinsinden alan değeri başlı başına nesnenin kasis olup olmadığını belirlemek için yeterli olmamaktadır. Alınan çerçevelerin çekim mesafesine ve çekim açısına göre nesneler daha büyük veya daha küçük gözükmektedir. Bu şekilde çok uzaktan alınmış bir kasis işareti görüntüsünde sadece alan filtresi ile algılama yapmak yetersiz kalacaktır. Bu aşamada bu alan filtresi ile eş zamanlı olarak kullanılabilecek ikinci bir parametre olan eliptiklik değeri kullanılmıştır. Bu iki parametrenin bir kural tabanlı yapı sayesinde birlikte kullanılması ile çift tabanlı gürbüz bir karar verme yapısı üretilmiştir.

Veritabanındaki 90 adet resim üzerinde bu geliştirilen algoritma sınamıştır ve tatmin edici sonuçlar benzetim ortamında elde edilmiştir. 90 adet resmin 80 tanesinde başarılı bir şekilde kasis işareti tespit edilmiştir ve kasis işaretleri daha rahat görülebilmesi açısından yeşil çerçeve içerisine alınmıştır (Şekil 11).


a)




b)


c)


d)

Şekil 11. a. Kötü hava şartları durumu, b. Kasis işaretinin uzakta bulunduğu durum, c. Eğik açı ile çekildiği ve eski kasis tabelası, d. Kırmızı bir arabanın gürültü teşkil ettiği durum

Performans tetsleri sadece tespiti kolay durumlardaki kasis işaretlerini belirlemek üzere yapılmamıştır, her türlü zor durum düşünülerek algoritma geliştirilmiştir. Buna örnek olarak Şekil 11.a'da kötü hava şartlarında çekilmiş bir görüntüde kasis belirlenmişken, Şekil 11.b'de kasis işaretinin çok uzakta olduğu bir durumda tespit

yapılmıştır, Şekil 11.c eski tipte bir kasis tabelasını gösterirken son görüntü Şekil 11.d'de ise kırmızı bir arabanın filtrelenmesi zor bir gürültü teşkil etiiği durum görülmektedir.

### 3.3. Performans Çözümleme Sonuçları

Trafik işareti belirleme çalışmalarının gerçek zamanlı olarak gerçekleştirilmesi oluşabilecek ani kazaların önlenmesi ve araçlarda oluşabilecek mekanik hasarların giderilmesi açısından çok önemlidir. Kullanıcıya kasis işaretinin yanına gelmeden önce uyarı verilebilirse kullanıcının fren mesafesi artırılabilecektir. Bu bağlamda araç dikiz aynasına yerleştirilen bir kameradan sürüş videosu toplanacaktır ve çeşitli algoritmalar yardımıyla kasis işareti belirlenecektir. Çalışmanın sadece kasis tabelası üzerinden yürütülmesi ilerde bütün tabelaların tespit edilebilmesi için bir fikir verecektir ve sistemin performansının belirlenmesi açısından önem arz etmektedir. Dolayısıyla, bu çalışmada gerçek zamanlı yapılacak bu sistemin bir ön çalışması yapılmıştır ve tasarlanan algoritma veri setinde bulunan 90 tane resmin 80 tanesinde başarı ile kasis tabelalarını tespit edebilmiştir. Algoritmanın performansının belirlenebilmesi için bu 90 tane resim farklı gürültü koşulları altında çekilmiştir. Tespit edilemeyen 10 resimde ise genellikle kasis tabelasının önünde bir nesne bulunmaktadır veya tabela üzerine yapıştırılan bir reklam etiketi tanıma işlemine engel olmuştur. 90 resim üzerinde yapılan bu analizde %88.8'lik bir başarım elde edilmiştir.

Bu tez kapsamında kasis işaretini tanıyan ve algılayabilen bir sistem benzetim ortamında etraflıca incelenmiş ve sistemin çalışmasını sağlayan algoritma ele alınmıştır. Sistem ilk olarak kasis işaretinin içerisinde bulunduğu kırmızı üçgen çerçeveyi belirler ve ardından kırmızı üçgen levha içerisindeki kasis işaretini algılar. Sistemin tasarlanma amacı ileride gerçekleştirilecek gerçek zamanlı çalışabilen, düşük maliyetli gömülü bir sistemin ön analizlerinin yapılabilmesi ve uygun algoritmaların belirlenebilmesidir. Sistemin benzetim ortamındaki performansı yukarıda verilen analiz sonuçları ile gösterilmiştir. Sıradaki bölümde ise geliştirilen algoritmanın performansını zor koşullarda çözümleyebilmek için çeşitli yapay zorluklar oluşturulmuştur.

Araca yerleştirilen kamera sistemlerinden görüntüler her zaman düz bir şekilde alınamayabilmektedir. Eğer tasarlanan sistem sadece resmin karşıdan çekildiği

durumda başarılı olması hesaplanırsa gerçek zamanlı çalışmada problemler çıkabilmektedir. Örneğin kameranın 5, 10 veya 20 derecelik bir eğikliği sonucunda elde edilen görüntüler de beklenilenden 5, 10 veya 20 derece eğik şekilde gelecektir. Bu durum yapay olarak "java" dilinde yazılan basit bir resimleri belirli açılarda çevirme kodu ile gerçeklenmiştir.

Bu sistemin performansını belirleyebilmek için cep telefonu kamerası ile çekilmiş 90 adet resim ve bunların 20, 40, 60 ve 80 derece yapay olarak döndürülmesi ile oluşturulmuş 360 adet resim kullanılmıştır. Toplamda 450 adet resimden oluşan veritabanı üzerinde 400 adet resimde kasis işareti başarı ile belirlenmiş ve farklı açılarda elde edilen resimler üzerinde algoritmanın gürbüzlüğü sınanmıştır.

### 3.4. Yapay Gürültü Üretilen Ortamlarda Algoritmanın Sınanması

Yine algoritmanın ilk kısmını kasis işaretinin belirlenmesi ikinci kısmını ise algılama oluşturmaktadır. Bu bölümde yeni bir algoritma geliştirilmemiş önceki bölümde geliştirilen algoritmanın çeşitli derecelerde eğiklik durumundaki performansı incelenmiştir.

Bizim kullandığımız eliptiklik ve alan filtresi bütün gürültülü durumlar ile baş edemeyebilmektedir. Kasis işaretin belirli bir piksel olarak alan değeri ve 0-1 arasında bir eliptiklik değeri vardır ama özellikle alan filtresi değerleri bazı durumlarda değişiklik gösterebilmektedir. Bu durumlar aşağıda listelenmiştir:

1) Kasis işaretini çeken kameraların belirli derecelerde rüzgar vb. nedenlerle eğilmiş olması,

2) Çekilen tabelanın fiziksel olarak bir darbe sonucu veya rüzgar, fırtına sonucu eğilmiş olması,

3) Çekilen tabela görüntüsü konumlanmasına göre (yakınlık, uzaklık) alan değerlerinin değişmesi,

Çalışmanın yeni olan bölümünde birinci ve ikinci madde simule edilmiştir. Veri setimizde toplam 90 adet resim vardır. Her bir resim 20, 40, 60 ve 80 derece olarak bir şekilde MATLAB ortamında döndürülmüştür. Dolayısı ile yeni elde edilen 360 tane resim türetilmiştir. Çalışmasının amacı bu aykırı durumlarda da yazılan algoritmanın ne denli başarılı olduğunu ölçülmesidir. Bu çalışma yapılmadan önce 90 adet resmin

80 tanesinde kasis işareti başarılı ile belirlenmiştir. Bu çalışma da ise 360 adet resmin 320' sinde yine belirlenmiştir. Gerçekleştirilen çalışmanın sonuçları Şekil 12'de sırası ile rastgele seçilmiş 20, 40, 60 ve 80 derece için bir resim üzerinden gösterilmiştir.


a) b)

c) d)

Şekil 12 .a. 20 derece, b. 40 derece, c. 60 derece, d. 80 derece [15]

Algoritmanın başarısı %88.8 olarak belirlenmiştir ve algoritmanın gürbüz olduğu ispatlanmıştır. Algoritma her bir derece için başarılı olarak çalışmıştır.

### 3.5. Benzetim Ortamında Elde Edilen Sonuçların Yorumlanması

Tezin ilk bölümünde yapılan çalışmalar tamamen Matlab programı üzerinde benzetim ortamı çalışmalarıdır. Gerçekleştirilecek her türlü gerçek zamanlı çalışmanın ilk

aşamasını yapılan benzetim çalışmaları oluşturmaktadır. Bu açıdan bakıldığında ilk olarak adım adım kasis işaretinin nasıl tespit edilebileceği detaylı olarak ele alınmış ve birtakım analizler yürütülmüştür. Bu sürecin ilk aşamasını kasis işaretini belirleme ikinci aşamasını ise kasis işaretini tanıma oluşturmaktadır. Belirleme ve algılama (Detection and Recognition) aşamaları cep telefonu kamerası ile çekilen 90 adet resim üzerinden yürütülmüş ve 80 tanesinde kasis işareti başarı ile tespit edilebilmiştir. Kalan 10 resimde ise algoritmanın başarısızlığından ziyade kasis işaretinin bir bölümünün yoldan geçen yayalar veya işaretler üzerinde yapıştırılan reklam etiketleri dolayısı ile tespiti gerçekleştirilememiştir.

Daha sonra aynı algoritma aynı veritabanındaki resimlerin belirli açılarda eğik pozisyona getirilmesi ile oluşan resimler üzerinde tekrar koşturulmuş ve sonuçlar toplanmıştır. Elde edilen sonuçlar 360 resim üzerinden incelemiş ilk durumdaki başarı oranı tekrar elde edilmiştir. Buradan görülebilmektedir ki tasarlanan sistem kasis işaretini çeken kameraların belirli derecelerde rüzgar vb. nedenlerle eğilmiş olması, çekilen tabelanın fiziksel olarak bir darbe sonucu veya rüzgar, fırtına sonucu eğilmiş olması veya çekilen tabela görüntüsü konumlanmasına göre (yakınlık, uzaklık) alan değerlerinin değişmesi gibi durumlardan etkilenmemektedir.

# 4. BENZETİM ORTAMINDA ALTERNATİF BİR KASİS İŞARETİ TANIMA SİSTEMİ

Bu bölümde, benzetim ortamında gerçekleştirilen bir kasis tanıma algoritması sunulmaktadır. Bu algoritma daha önce sunulan ilgili algoritmadan tamamen farklıdır ve bu yayımlanan yönteme göre bir takım artıları ve eksileri bulunmaktadır [16]. Benzetim ortamında ilk gerçekleştirilen algoritmada kasis çerçevesinin dışındaki kırmızı üçgen levha bir takım renk bölütleme yöntemi ile tespit edilmekteydi ve bulunan resimler alt ve üst değerler yardımı ile Otsu otomatik eşik değeri bulma fonksiyonu içerisine gönderilmekteydi. Bulunan eşik değeri yardımı ile resimler siyah beyaza dönüştürülmekteydi ve veri tabanındaki çoğu resim için başarılı sonuçlar üretilebilmişti. İlgili çalışmada Otsu fonksiyonuna sokulan resim orijinal resmin gri skalaya dönüştürülmüş hali ile orijinal resmin kırmızı alt uzay resminin farkından oluşturulmaktaydı. Bu algoritmada ise tamamen farklı bir yöntem kullanılmıştır. Yine ana amaç belli bir rengi ön plana çıkartmak olsa da kullanılan yöntem farklı adımlardan oluşmaktadır. Aşağıdaki bölümlerde bu algoritmanın her aşaması kapsamlı bir şekilde ele alınmıştır ve sonuçlar şekillerle verilmiştir. Sistemin tamamının blok diyagramı Ek-2'de verilmiştir

Algoritmanın ilk aşamasını, veritabanındaki resimlerin otomatik olarak Matlab çalışma ortamına alınması oluşturmaktadır. Bu işlemi gerçekleştirebilmek için hazır Matlab fonksiyonları kullanılarak belirli bir bilgisayar konumundan n adet ".JPEG" uzantılı görüntüyü matris formunda aktaracak bir kod parçası yazılmıştır. Bu bölümde yine tezin ilk bölümünde kullanılan 90 resimlik standart cep telefonu kamerası ile çekilmiş kasis tabelası resimlerinden oluşturulan veritabanı değiştirilmeden kullanılmıştır.

## 4.1. Resimlerin çalışma ortamına alınması

Şekil 13'de veritabanında bulunan 90 adet resmin sadece iki tanesi sunum yükünü azaltmak adına gösterilmektedir. Bu resimlerden çalışmanın sonunda kasis işaretinin belirlenmesi ve algılanması gerçekleştirilecektir.

a) b)

Şekil 13. Veritabanından örnek iki adet resim

## 4.2. Kırmızı Üçgen Bölümün Çıkarılması

Kasis işaretinin belirlenebilmesi için yine ilk olarak uygun renk uzayı seçimi ve belirli rengi bölütleyebilmek için birtakım çalışmaların yapılması gerekmektedir. Kırmızı renkli üçgen bölüm ön plana çıkartılıp resmin geri kalanı atıldığında odaklanılacak bölüm geriye kalan kasis işaretidir. Dolayısı ile aşağıda bu kırmızı bölümü ön plana çıkartıp daha sonra yapılacak bölütlemenin ilgili aşamaları verilmektedir.

### 4.2.1. Renk Bölütleme ve Kırmızı Rengin Ön Plana Çıkarılması

İlk yöntemde Otsu dinamik eşik değeri yöntemi ile her bir resim için kırmızı alan çıkarılıyor daha sonra morfolojik sinyal işleme teknikleri ile kasis işareti tanınıyordu. Şimdi ise bu yöntemden farklı bir yöntem kullanılarak aynı sonuca yakın sonuçlar türetilmeye çalışılmıştır. Bu yeni yöntemde ana amaç veri seti içerisindeki bütün resimlerde üçgen levha üzerindeki piksellerin sırası ile R (kırmızı), G (yeşil), B (mavi) alt uzay değerlerine göre uygun analizlerin gerçekleştirilmesidir. Şekil 14'de ilgili bölümlerdeki piksellerin alt uzay değerleri gösterilmiştir. Veri seti üzerindeki bütün resimleri için aynı süreç tekrarlanmıştır. Ve kabaca her bir alt uzay için aralıklar tablolara not edilmiştir.


a)


b)


b)


d)

Şekil 14. Orijinal (İşlenmemiş) dört resim üzerinde alt uzay değerlerinin işaretlenmesi

Bu değerlere bakılarak aşağıdaki gibi Çizelge 2 oluşturulmuştur.

Çizelge 2. Belirlenen alt uzay değer aralıkları

| | **Kırmızı Alt Uzay Değeri** | **Yeşil Alt Uzay Değeri** | **Mavi Alt Uzay Değeri** |
|---|---|---|---|
| **1. Resim** | 250 - 255 | 35 – 50 | 45 – 70 |
| **2. Resim** | 130 – 165 | 25 – 35 | 30 – 40 |
| **3. Resim** | 160 – 170 | 60 – 85 | 100 – 125 |
| **4. Resim** | 135 – 150 | 35 – 45 | 35 – 45 |

Çizelge 1'e bakıldığında örneğin birinci resim için kırmızı rengin hangi üç renk aralığının karışımından elde edilebileceği tespit edilmiştir. Tablodaki renk aralıkları her bir alt uzay için seçilebilecek değer aralığını simgelemektedir. Mesela seçilebilecek (252, 45, 60) değeri birinci resimdeki tona yakın bir kırmızı renk üretecektir. Bu mantık çerçevesinde bir filtreleme tekniği oluşturulmuştur. Literatür çalışmaları ele alındığında bu yöntemi kullanan bir renk bölütleme çalışması bulunamamıştır. Bu ise yapılan çalışmanın özgünlüğünü ortaya koymaktadır. Bu bölümde resimlerin renk tonları için bir normalizasyon yapılabilmektedir. Normalizasyonun tez boyunca kullanılmamasının iki temel nedeni bulunamkatdır. İlk olarak normalizasyon da bulunacak alt ve üst limit değerler herbir ışık açısı için değişebilmektedir. İkinci neden ise normalizasyon formülasyonunda paydada bulunan değerin işlem yükü getirmesidir. Çünkü bütün çalışmalar gerçek zamanlı algoritmanın hızlı olması hedeflenerek yapılmıştır. Gerçek zamanlı çalışmada normalizasyon için yapılacak bölme işlemi işlemci için fazladan yük getirecek ve hesapsal olarak ağır olacaktır. Aşağıda kullanılan teknik sunum yükünü azaltmak için sadece birinci resim üzerinden özetlenmiştir:

- Verilen resim kırmızı, yeşil, mavi alt uzaylarına ayırılır
- İlk olarak kırmızı alt uzayda resim içinde verilen aralığa uygun piksellere "0" uymayanlara "1" değeri atanır.
- Çizelgedeki dört aralık içinde bu işlem uygulanır ve dört farklı 1,0 değerlerinden oluşan yeni resimler türetilir.
- Türetilen bu dört 1,0 değerlerinden oluşan resimler "VE" işlemine tabi tutulur.

Bu sürecin amacı gelen her bir resimdeki kırmızı üçgen levha kırmızının farklı tonlarında olabileceği için bu aralıklardan birine denk düşeceği beklenmektedir ve o

yüzden her bir resim bu dört aralık ile sınanmakta ve sonuç “VE” işlemine tabii tutulmaktadır. Şekil 15’de dört aralık ile sınama işlemi verilmiştir. Ayrıca aynı şekil grubunun son alt şeklinde çıkan “VE” işlemi sonucu da görülebilmektedir. Burada dikkat edilmesi gereken hususlar aşağıdaki gibidir:

- Diyelim ki sadece birinci resim için kırmızı rengi ön plana çıkartalım.
- İlk olarak bakıldığında birinci resimde kırmızı renginin ton aralığı Tablo 1’den de görülebileceği üzere 250 – 255 arasındadır. (Şekil 14.a)
- Şekil 14.a’daki resmin hepsinin diğer üç resimdeki kırmızı renk aralığı ile örtüşüp örtüşmediğine sırası ile bakılır. Buna göre 250-255 arası tabloda sadece birinci resim kırmızı alt uzayı ile örtüşmektedir. 250-255 arası olan birinci resim renk tonu 2. resim kırmızı alt uzay aralığı olan 130-165, 3. resim kırmızı alt uzay aralığı olan 160-170 ve 4. resim kırmızı alt uzay aralığı olan 135-150 ile örtüşmemektedir. Ve örtüşen çıktılara mantıksal “0” yani beyaz renk, örtüşmeyen çıktılara ise “1” yani siyah renk atanmıştır.
- Bu çerçeveden bakıldığında birinci resimdeki piksellerin her biri sırası ile dört resim için tablolanan kırmızı alt uzay aralıkları ile teker teker kontrol edilmiştir ve sonuçlar kaydedilmiştir. Bu ilgili sonuçlar Şekil 15.a, 15.b, 15.c ve 15.d’de görülebilmektedir.
- Tablodan da rahatlıkla anlaşılabileceği üzere sadece birinci aralık için örtüşme olmuştur ve Şekil 15.a’da kırmızı bölüm mantıksal “0” değerini yani beyaz rengi alırken aynı kırmızı bölüm diğer aralıklar ile kontrol edildiğinde kırmızı bölüm mantıksal “1” yani siyah rengi almıştır.
- Son aşamada ise aynı resmin dört aralıkla örtüşmeleri tespit edildikten sonra elde edilen dört farklı “1” ve “0” lardan oluşan resimlerin mantıksal “VE” işlemin tabi tutulması gelir ve bu işlemin sonucu Şekil 15.e’de gösterilmektedir.


Birinci Aralik

a)


Ikinci Aralik

b)


Ucuncu Aralik

c)


Dorduncu Aralik

d)


e)

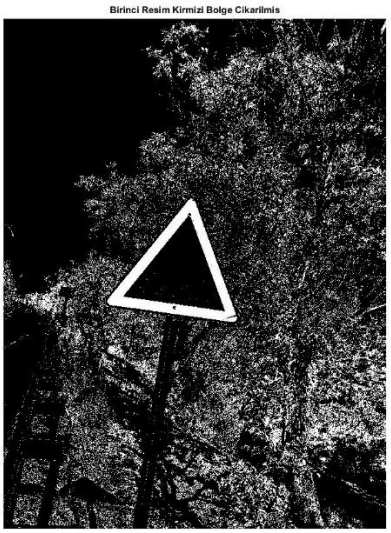
Şekil 15. Birinci resim kırmızı alt uzay için dört aralığın teker teker kontrol edilmesi ve çıkan sonuçların “VE” işlemine tabi tutulması sonucu çıkan resim

Daha sonra aynı işlemler aynı resmin yeşil ve mavi alt uzayları için de uygulanmış ve Şekil 16’daki şekiller elde edilmiştir. Yine aynı resim için Çizelge 2’in bu sefer üçüncü ve dördüncü sütunundaki değerler sınanmıştır.

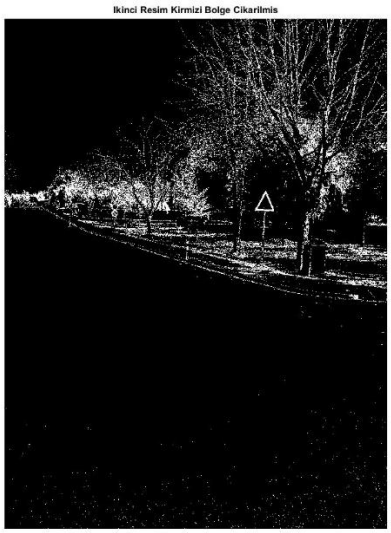

a) b)

Şekil 16. Birinci resim için yeşil ve mavi alt uzay için “VE” işlemleri sonuçları

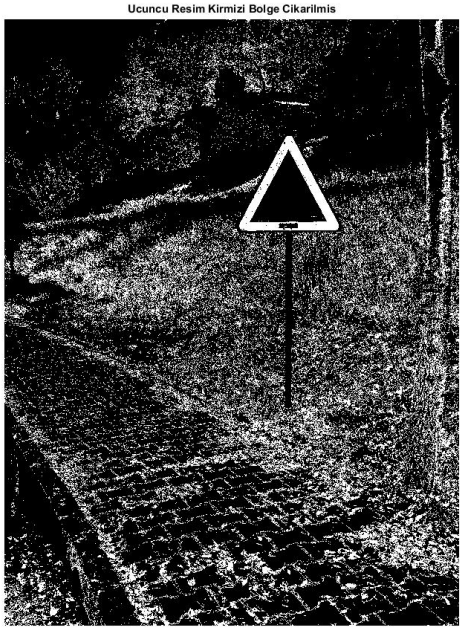
Sıradaki işlem bu bulunan üç alt uzay eşik değerlenmiş resmin (Şekil 15.e, Şekil 16.a ve Şekil 16.b) mantıksal “VEYA” işlemine tabii tutulmasıdır. Bu işlemin sonucu Şekil 17’da dört resim için gösterilmektedir.



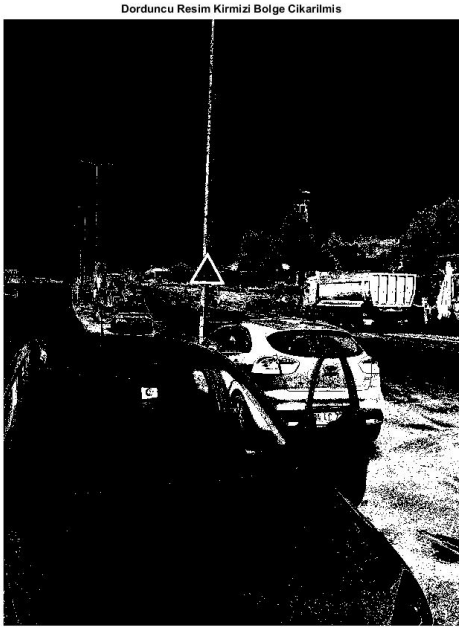

a)




b)




c)




d)

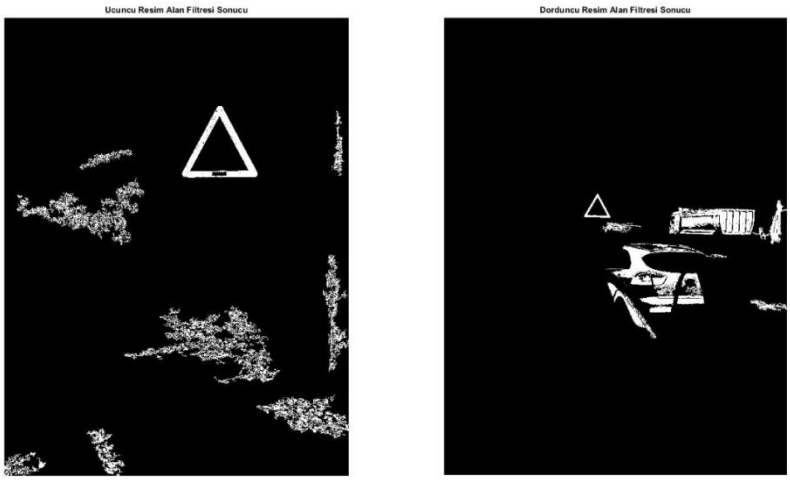
Şekil 17. Birinci resim b. ikinci resim c. üçüncü resim d. dördüncü resim için ön plana çıkarılan kırmızı bölgeler

Bu bulunan renk bölütleme yöntemi özgün olmakla beraber bir takım artıları ve eksileri bulunmaktadır. Öncelikle oluşturulan Çizelge 2'deki değerler veri tabanındaki dört resim için oluşturulmuş ve bütün veritabanı için kullanılmıştır. Veritabanındaki bütün resimlerde başarı ile çalışması bütün resimlerin genelde bu ışık seviyelerinde, aynı telefon kamerası ile çekilmesindendir. Veritabanı farklılaştıkça bu eşik değeri tablosunun genişletilmesi gerekebilmektedir. Bu eksisinin yanında çok önemli artıları da bulunmaktadır. İlk önemli getirisi siyah beyaza dönüştürme işleminin gerekmemesidir. Çünkü işlemler alt uzaylarda yapıldığı için tek kanallıdır ve bu tek kanal üzerinde mantıksal atamalar yapıldığı için elde edilen resimler 1 ve 0' lardan oluşmaktadır. Böylece türetilen resimler doğası gereği siyah beyazdır. İkinci avantajı ise dinamik olarak bir eşik değeri bulma fonksiyonu kullanılmamıştır ve bu fonksiyonun getirdiği fazladan iş yükü ortadan kaldırılmıştır. Ayrıca resimlerin literatürdeki gibi HSC veya YCbCr gibi diğer uzaylara dönüştürülmesine gerek kalmadan işlemler zaten görüntünün alındığı RGB uzayında gerçekleştirilmiştir. Bu da işlem yükünü oldukça azaltmaktadır. Bir sonraki bölümde üçgen geometrik şeklinin tanınması kapsamlı bir şekilde ele alınmıştır.

### 4.3. Kırmızı Bölgenin Şeklinin Belirlenmesi

Veritabanı içerisinde kırmızı rengin ön plana çıkarılması işlemi tamamlandıktan sonra bir sonraki işlem bu kırmızı bölgelerin içerisinden üçgen olanların alınması kalanların gürültü olarak değerlendirip atılmasıdır. Yöntemde çok küçük nesneler tesadüf eseri üçgen olarak dizilmiş olabilir. Bu gürültü manasına gelen nesnelerin de bir alan filtresinden geçirilmesi sonucu çok küçük üçgen şekilli gürültüler atılmıştır. Bu çalışmada bulunan şekilleri çevreleyen en küçük dikdörtgenler, şekillerin etrafına şekli kapsayacak şekilde çizilmiş ve bu dikdörtgenlerin çeşitli ayırt edici özellikleri incelenerek bir sınıflandırma yapılmıştır. Önceki bölümün sonunda bahsedildiği gibi 1, ve 0'lardan oluşan mantıksal resim elde edilmiştir. Artık bu bölümde yapılan çalışmalar bu siyah beyaz resimler üzerinden gerçekleştirilmiştir. Şekil 17'deki dört resim çıktısı da dikkatle incelendiğinde birbirine bağlı örüntü (connected pattern) olarak kasis harici çok fazla nesne bulunmamaktadır. Sadece Şekil 17.d'de kırmızı arabanın arkası birbirine bağlı örüntü olarak bir nesne gibi davranabilmektedir. Siyah zemin üzerindeki geri kalan nesnelerin bağlı örüntü özellikleri incelendiğinde belirli piksel değerinin altındaki birbirine bağlı örüntüler gürültü olarak nitelendirilebilmektedir. Bu işlemin iki ana faydası vardır. İlki çok sayıda gürültünün

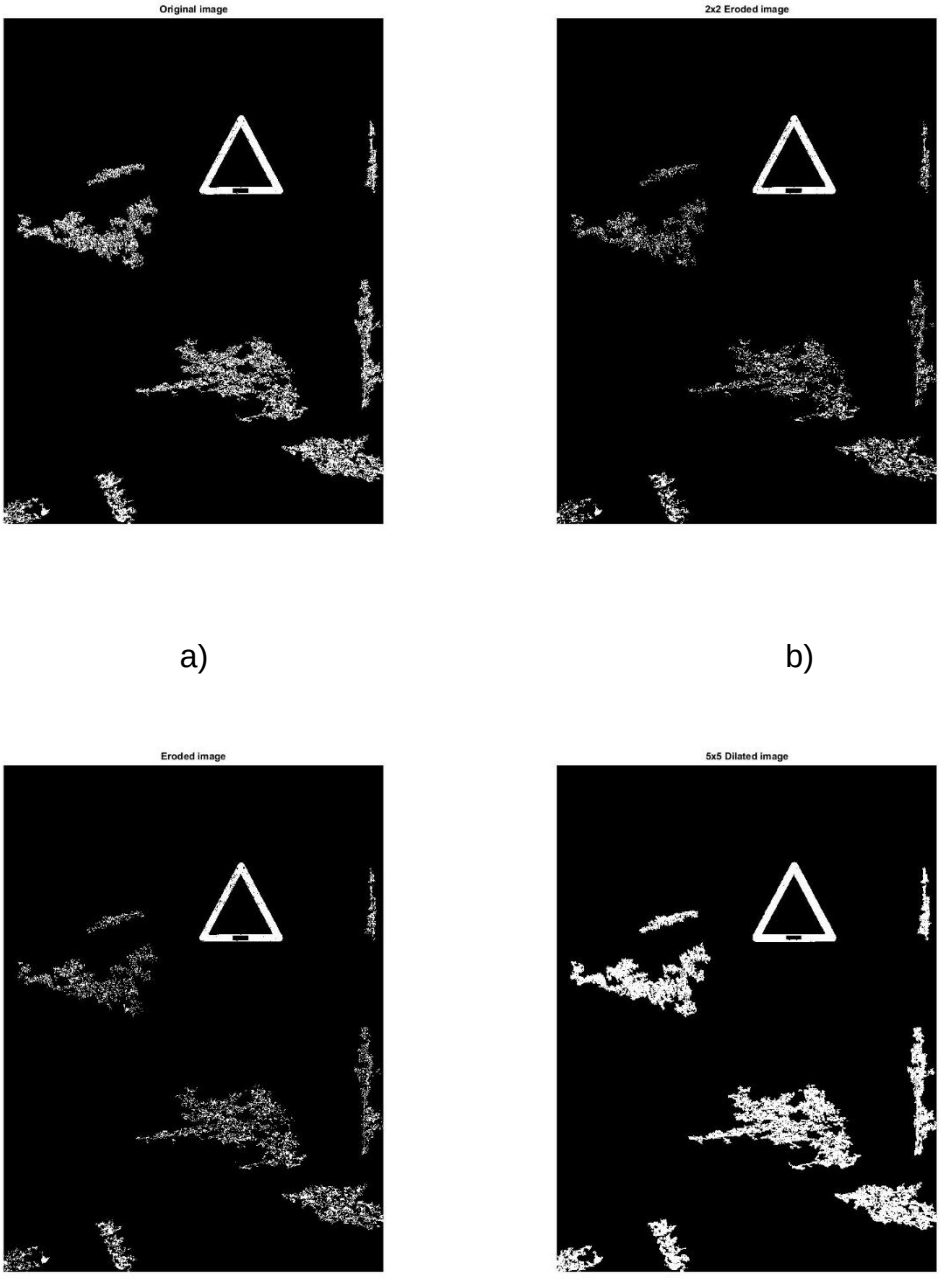
kasis olup olmadığını denetlemek çok işlem yükü getireceği için bu işlem yükünün fazlası azaltılmış olur. Diğer bir avantajı ise; bu işlem uygulanmaz ise tesadüf eseri üçgen şeklinde dizilmiş en az dört piksel bile üçgen şeklinin oluşumuna neden olabilir ve birazdan ele alınacak üçgen belirleme algoritması bu örüntüyü üçgen olarak değerlendirebilir. Bu alan filtresinin gürültüleri yok edip üçgen işareti yok etmemesi için veri tabanındaki bütün resimler tek tek incelenmiştir ve üçgen levhaları oluşturan bağlı örüntülerin piksel sayıları not edilmiştir. Kabaca bir rakam vermek gerekirse Şekil 17.b ve Şekil 17.d'deki gibi uzaktaki nesneler için 3000 piksel gibi rakam çıkarken yakındaki üçgen levhalar için (Şekil 17.a ve Şekil 17.c) bu değer yaklaşık 15000 piksel olarak tespit edilmiştir. Bu bilgiler ışığında kullanılacak alan filtresi için eşik değeri tabanlı bir yöntem izlenmiştir. Bu bölüm Matlab ortamında gerçekleştirildiğinde algoritmanın siyah beyaz resim üzerinde başarı ile çalıştığı gözlemlenmiştir. Bu işlem yapıldıktan sonra elde edilen yeni siyah beyaz resimler Şekil 18'de verilmiştir. Şekil 18'e dikkat edildiğinde halen birtakım istenmeyen gürültülerin bütün resimler için kaldığı görülmektedir. Bu aşamada yapılması gereken işlem yeni bir yöntem geliştirilmesidir. Tezin bir önceki bölümünde morfolojik bir sinyal işleme tekniği olan eliptiklik filtresi kullanılmıştı ve tatmin edici sonuçlar elde edilmişti. Her ne kadar sonuçlar tatmin edici olsa da kullanılan filtre her bir nesne için çalışmakta ve çeşitli matematiksel ve geometrik hesaplar yapmaktaydı. Benzetim ortamında işlem yükü çok ciddi bir problem olmasa bile gerçek zamanlı bir çalışma için daha farklı alternatifler arayışına gidilmiştir. Bu çerçeveden bakıldığında geliştirilmesi gereken yöntemin hem bu üçgen olmayan nesneleri yok etmesi hem de bulunan nesnenin üçgen olup olmadığını kontrol etmesi gerekmektedir. yapılacak ilk işlem bulunan her bir bağlı örüntünün dışına çevreleyen sınırlarını çizmektir. Bunun için Matlab içerisinde bulunan hazır bir fonksiyon kullanılmıştır. Bu hazır fonksiyonun normal şartlar altında varsayılan olarak nesnelerin içerisindeki delikler için de dış sınır çizimi yaptığı belirlenmiştir.




a) b)

Şekil 18. Rastgele seçilmiş birinci b. ikinci resim için alan filtresi sonucu

Bulunacak üçgen levhanın üçgen olan kısmı içerisindeki noktasal deliklerin işlem yükü getirmemesi için aşındırma ve genişletme işlemleri gerçekleştirilmiştir. Morfolojik filtreler genelde iki temel işlemden türetilmiştir. Bunlar aşındırma (erosion) ve genişletme (dilation) işlemleridir. Aşındırma ikili bir görüntüde bulunan nesnelerin boyutunu seçilen yapısal elemente bağlı olarak küçültürken, genişletme nesnenin alanını artırır. Bu işlemlerden erosion işlemi birbirine ince bir gürültü ile bağlanmış iki veya daha fazla nesneyi birbirinden ayırmak için kullanılırken, dilation işlemi ise aynı nesnenin bir gürültü ile ince bir şekilde bölünerek ayrı iki nesne gibi görünmesini engellemek için kullanılır. Bu işlemlerin haricinde morfolojik görüntü işleme altında bu fonksiyonlardan türetilmiş çok sayıda fonksiyon bulunmaktadır. En sık kullanılan diğer iki morfolojik işlemden birincisi morfolojik açmadır. Açma işlemi görüntü üzerine önce erosion ardından dilation işlemi uygulanması ile gerçekleştirilir. Aşındırma ile küçük parçalar yok edildikten sonra dilation ile görüntü tekrar genişletilerek küçük parçaların kaybolması sağlanır. Kapama işlemi ise açma işleminde uygulanan adımların tersten uygulanmasıdır.

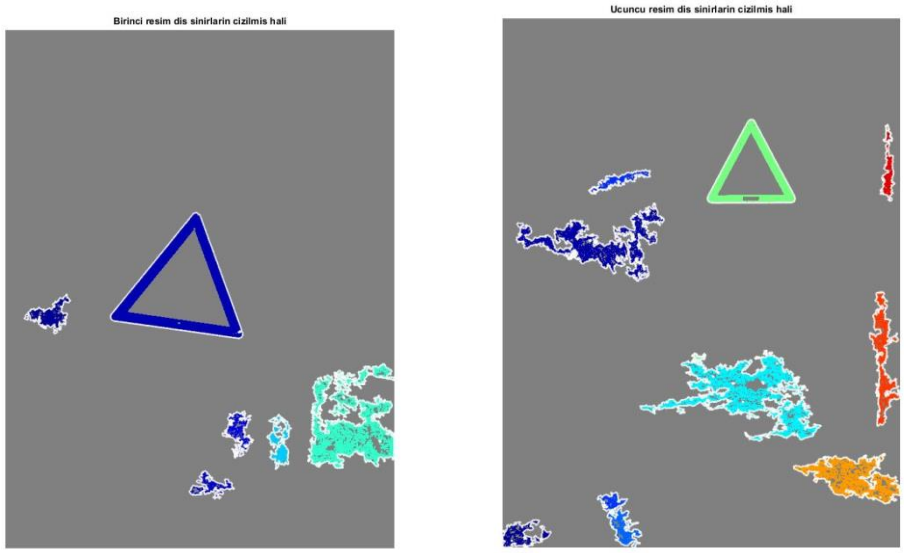



Şekil 19. Morfolojik Açma İşlemi

Örnek olarak Şekil 18.a‘ya bakılırsa eğer, kasisi içeren üçgen nesnenin alt kenarında bir etiketten dolayı oluşan bir açıklık bulunmaktadır. Bu açıklık biraz önce anlatıldığı gibi bağlı örüntünün dış sınırını çizerken yeni bir nesne gibi davranacaktır ve fazladan işlem yükü getirecektir. Bu morfolojik açma işleminin sonucu sadece Şekil 18.a üzerinden gösterilmiştir. Şekil 19.d dikkatle incelendiğinde Şekil 19.a’da beyaz üçgen

içerisinde olan siyah nokta pikseller artık burada kalmamıştır. Dolayısı ile morfolojik açma işleminin bu durumda uygun çözüm olduğu belirlenmiştir.

Dış sınır çizimi için hazırlanan örüntüler bahsedilen hazır Matlab fonksiyonuna sokulduğunda hem dış sınırın (x,y) koordinatlarını hem de kaç adet nesne olduğunun sayısını üretmektedir. Bu hazır fonksiyon içerisine Şekil 19.d gönderildiğinde birinci resim için Şekil 20.a üçüncü resim için Şekil 20.b'deki gibi çıktılar üretmektedir.




a) b)

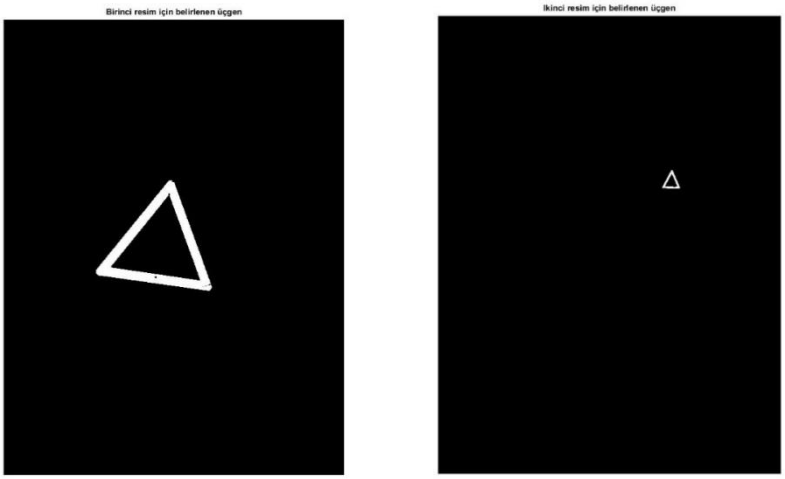
Şekil 20. Rastgele seçilmiş birinci b. ikinci resim için dış sınır belirleme işlemi

Bir sonraki bölümde dış şekil çizimi yapılan nesnelerin geometrik şekillerinin ne olduğunun belirlenmesi ile ilgili birtakım çalışmalar yapılmış ve sonuçları gösterilmiştir.

### 4.4. Üçgen Şeklin Tespit edilmesi

Bu bölümde artık dış çizimi yapılmış nesnelerin geometrik şekilleri belirlenecektir. Bu işlem için nesnelerin geometrik boyutlarını hesaba katarak şekillerini belirleyebilen bir yöntem geliştirilmiştir. Şekil faktörü, nümerik olarak bir partikül veya nesnenin şeklini büyüklüğünden bağımsız olarak belirlemek için kullanılır. Burada kritik olan nokta bu yöntemin belirlenen nesnelerin boyutlarından bağımsız olarak çalışabilmesidir. Bu kritik nokta üçgen levhanın şeklini belirlemekte kilit rol oynamaktadır çünkü alınan resimlerin bazıları yakından bazıları uzaktan çekilmiş olabilmektedir. Dolayısı ile uzaklıktan yani büyüklükten etkilenmeyen bir algoritmanın gerekliliği açıktır. Bu

bilgiler ışığında bulunan nesneler bir döngü içerisinde teker teker kontrol edilmiştir ve her bir nesne için en-boy oranı, dairesellik ve üçgen şekil faktörü değerleri tek tek hesaplanmıştır. Bulunan üçgen şekil faktörü 0.20 ile 0.25 aralığında çıkmıştır ve bu aralık ciddi derecede dar olduğu için gelen resimde kasis etrafındaki üçgen kadar kesin olmadıkça üçgene benzeyen bütün şekiller elenmektedir. Ayrıca yöntemin uzaklıktan da bağımsız olması önemli bir artıdır. Belirlenen eşik değeri uygulandığında bulunan nesnelerden sadece üçgen olanlar belirlenmiş diğerleri elenmiştir. Örneğin orijinal resim olan Şekil 8.a için bulunan 40 adet nesnenin 39 u bu eşik değeri vasıtası ile elenmiş sadece aradığımız üçgen elde kalmıştır. Şekil 21.a ve 21.b'de bulunan üçgenler siyah beyaz resim olarak görülebilmektedir.

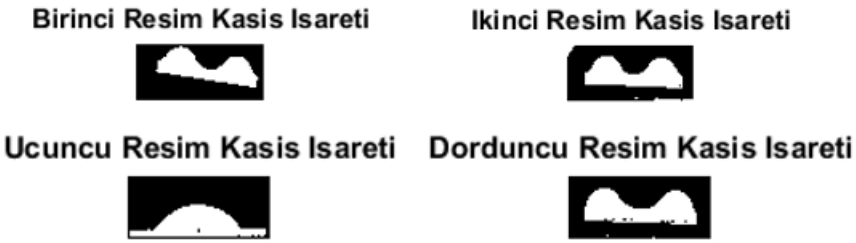



a) b)

Şekil 21. Rastgele seçilmiş birinci resim b. ikinci resim için üçgen tespit algoritması sonuçları

Bir sonraki aşamada bulunan üçgen etrafına çizilen çevreleyen dikdörtgenin koordinatları üzerinden resmin kesilmesi ve daha sonra yeniden boyutlandırılması gerçekleştirilmiştir. Eski boyutlar dört resim içinde standart olarak 3264x2448 iken sonraki boyut dört resim içinde farklıdır. Örnek olarak ilk resim için 793x834 iken ikinci resim için 133x126 dir. Bir sonraki aşamada ise kasis işaretinin tanınması sağlanacaktır. Bu işlemde uygulaması basit ve çok işlem yükü olmayan bir metot izlenmiştir. İzlenen metodun en önemli özelliklerinden biri yine uzaklıktan bağımsız çalışabilmesidir. Bu uzaklıktan bağımsızlığı sağlayan ise yeniden boyutlandırma işlemidir. Yeniden boyutlandırma sayesinde resim nerden ve hangi açıdan çekilirse çekilsin aynı boyuta taşınacaktır. Bu yeni boyuta işlem yükünü düşürmek maksadı ile

100x100 piksel olarak karar verilmiştir. Bu piksel değeri ile hem boyut çok azalmıştır hem de üçgen işareti içerisinde ki kasis simgesinde fazlaca bir bozulma meydana gelmemiştir. Bu aşamadan sonra yapılacak işlem üçgen şekiller içerisinde ki kasis işaretini oluşturan simgedeki beyaz piksellerin sayılmasıdır. Burada kritik noktalardan birisi uygun renk uzayı dönüşümü ile sadece kasis simgesini ön plana çıkarmak geri kalan arka planı zıt bir renk ile ifade etmektir. Önceki bölümde yapıldığı gibi yine renk bölütleme işlemi yapılmıştır. Ama bu sefer kırmızı rengi ön plana çıkarmak yerine siyah rengi ön plana çıkartmak gerekmektedir. Bu işlemin detayları verilmemiştir ancak sayım işlemi sonucu bulunan kasis işaretleri Şekil 22'de gösterilmektedir.




Şekil 22. Dört resim için bulunan kasis işaretleri

Veritabanındaki 90 adet resim için bu 4 adet resimden elde edilen üç farklı alt uzay için ilgili eşik değerleri kullanılmıştır. Daha önceden de belirtildiği gibi örnek olarak alınan resim sayısı dörtten daha fazlaya çıkarılırsa denenen yöntemin başarımı artacaktır. Bu koşullar altında 90 resim üzerinde 76 tanesinde başarı ile kasis işareti tespit edilmiştir. Tezin ilk bölümündeki sonuçlar dikkatle incelenirse orada 90 resim üzerinden 80'i üzerinde başarım elde edilmişti. 10 resim de kasis işareti bir şekilde çekim açısı nedeni ile kapatılmıştır ve algoritma başarı ile çalışmamıştır. Bu bölümde yine aynı 10 resim için sistem çalışmamıştır. Sistemin çalışmadığı diğer dört resim ise tabloda oluşturulan eşik değeri aralıklarına düşmeyecek şekilde farklı bir kırmızı tonudur. Daha önceden belirtildiği gibi eşik değerleri tablosu genişletildiğinde çalışmanın performansı artmaktadır.

### 4.5. Benzetim Ortamında Elde Edilen Sonuçların Yorumlanması

Bu bölümde yapılan benzetim ortamı çalışmasında 90 adet resimden 76 tanesinde başarı ile kasis işareti tespit edilebilmiştir. İlk yönteme göre dört resim için başarısız sonuç elde edilmiştir. Veritabanındaki birçok resimlerde renk bölütlemenin başarı ile çalışması bütün resimlerin genelde bu ışık seviyelerinde, aynı telefon kamerası ile

çekilmesindendir. Veritabanı farklılaştıkça bu eşik değeri tablosunun genişletilmesi gerekebilmektedir. Bu eksisinin yanında çok önemli artıları da bulunmaktadır. İlk önemli getirisi siyah beyaza dönüştürme işleminin gerekmemesidir. Çünkü işlemler alt uzaylarda yapıldığı için tek kanallıdır ve bu tek kanal üzerinde mantıksal atamalar yapıldığı için elde edilen resimler 1 ve 0' lardan oluşmaktadır. Ayrıca gerçek zamanlı çalışmalarda 3 kanal ile çalışmanın getirdiği işlem yükü bir kanal üzerinde çalışmanın vermiş olduğu işlem yükünden çok daha fazladır. Böylece türetilen resimler doğası gereği siyah beyazdır. İkinci avantajı ise dinamik olarak bir eşik değeri bulma fonksiyonu kullanılmamıştır ve bu fonksiyonun getirdiği fazladan iş yükü ortadan kaldırılmıştır. Ayrıca resimlerin literatürdeki gibi HSC veya YCbCr gibi diğer uzaylara dönüştürülmesine gerek kalmadan işlemler zaten görüntünün alındığı RGB uzayında gerçekleştirilmiştir. Bu da işlem yükünü oldukça azaltmaktadır.

# 5. GERÇEK ZAMANLI KASİS İŞARETİ TANIMA SİSTEMİNİN GELİŞTİRİLMESİ

Gerçek zamanlı çalışabilen bir kasis işareti tanıma sistemi gerçekleştirebilmek için "Opencv" adlı Intel'in geliştirdiği açık kaynak kodlu bir kütüphane kullanılmıştır. Opencv Bilgisayar ortamında görüyü işlemeyi sağlayan bir kütüphanedir. Intel tarafından geliştirilen openCV, BSD (Berkeley Software Distribution) tarafından lisanslanmış olup, Windows, Linux, MacOS X gibi birçok platformda çalıştırılabilir, ayrıca C, C++, python ve Java gibi dillerde kütüphanesi bulunmaktadır. Opencv kütüphanesi, ücretsiz olması, işletim sistemi bağımsız olması, gerçek zamanlı uygulamalarda kullanılabilir olması ve 500'ün üzerinde fonksiyon desteği sayesinde görüntü işleme ile ilgilenen kişiler için hızlı ve rahat bir geliştirme süreci sağlar. Ayrıca eğitim kurumlarında açık kaynak kodlu ders aracı olarak kullanılabilir [17].
Bu kütüphane kullanılarak gerçekleştirilen kodlar Matlab'a nazaran 500~600 kat daha hızlı çalışabilmektedir. Gerçek zamanlı görüntü işleme çalışmaları için Opencv'yi popüler kılan en önemli nokta ise bu çalışma performansıdır. Literatüre bakıldığında gerçek zamanlı çalışan çoğu tanıma veya algılama sisteminin Opencv Kütüphanesi kullanılarak gerçekleştirildiği göze çarpmaktadır [18,19,20].

Bu çalışmalar ve diğer birçok çalışma ele alındığında dikkati çeken önemli bir husus vardır. Yapılan çoğu görüntü veya video işleme çalışması Opencv ile yapılmıştır fakat sistemlerin gerçek zamanlı olarak üzerinde koştuğu cihazlar çok geliştirilememektedir. Dolayısı ile bu yazılımlar ile kameralardan alınan görüntüler statik olarak depolanır ve daha sonra Matlab, Opencv vs. yazılımlar ile çeşitli algoritmalar geliştirilir. Gerçek zamanlı çalışan bir sistem tasarlamanın zorlukları aşağıda listelenmiştir.

1- Alınan görüntülerin kalitesi yüksek olduğunda görüntü işleme algoritması iyi çalışmakta fakat kullanılan işlemcinin hafıza problemi ortaya çıkmaktadır.

2- Eğer bir tanıma veya algılama işlemi yapılacaksa ve yeni gelen görüntü bir veritabanı ile kıyaslanacaksa oluşturulacak veritabanına önceden girilmelidir. Bu da işlemci için ekstra hafıza gerektirmektedir.

3- Alınan görüntülerin saniyedeki toplanma hızı (frame per second) artırıldığında performans artmaktadır fakat hafıza yükü artmaktadır. Eğer çerçeve toplama hızı düşürülürse bu seferde önemli bilgiler atılan çerçeveler ile kaybolabilmektedir.

Bu problem göz önünde bulundurulduğunda halen araç içerisine yerleştirilen ve tüm trafik işaretlerini tanıyan bir sistemin neden olmadığı açıklanmaktadır. Aynı zamanda bu kadar işlemi yapabilmek için kullanacak sistem hem çok pahalı olacak hem de çok büyük hafızalara gereksinim duyacaktır.

Bu tez çalışmasında ise bu kısıtlar göz önünde bulundurulduğunda sadece ilk aşama olarak bir kasis işareti tanıma çalışması gerçekleştirilecektir. Denenecek yöntem bir veritabanı ile kıyaslama yapılmadan sadece gelen görüntü üzerinde aşağıdaki işlemlerin sırası ile yapılması ile gerçekleştirilecektir. Çünkü şablon karşılaştırma yöntemleri çok başarılı bir şekilde nesne tanıma görevlerinde kullanılsa da hem hafızada tutulması gereken verilerin çok sayıda olması hem de eldeki resmin veritabanındaki bütün resimler ile teker teker karşılaştırılmak zorunda olması çok fazla iş yükü getirmektedir. Bu amaçla şablon veya taslak resim ile veritabanındaki diğer resimleri karşılaştırma temelli algoritmalarda bu gerçek zamanlı bölümde kaçınılmıştır.

Aşağıda sırası ile tasarlanan gerçek zamanlı sistemin görevleri listelenmiştir ve sistemin tamamının blok diyagramı Ek-3'de verilmiştir:

1- Görüntünün alınması

2- Renk uzaylarına bölünmesi ve uygun renk uzayına karar verilmesi

3- Kırmızı bölgenin ön plana çıkarılması

4- Otomatik eşik değeri yöntemi ile siyah beyaz resme dönüşüm işlemlerinin gerçekleştirilmesi

5- Kırmızı bölge içerisinde piksel bazında alan filtresinin uygulanması ve her türlü gürültünün bertaraf edilmesi

6- Hafıza probleminin gidermek için önce ilgili bölgeler alınacak ve geri kalan bölgeler resimden atılacak ve dolayısı ile yeni resim boyutları düşürülerek elde edilecektir.

7- Elde edilen resim içerisindeki kasis işaretinin algılamak için tablolama yöntemi ile elde edilmiş alan değerleri ve kasis işaretini anlatan simgenin piksel sayıları kullanılacaktır.

8- If-else yapısı ile kırmızı üçgen yapısı içindeki işaretin kasis veya başka bir işaret olup olmadığı belirlenecektir.

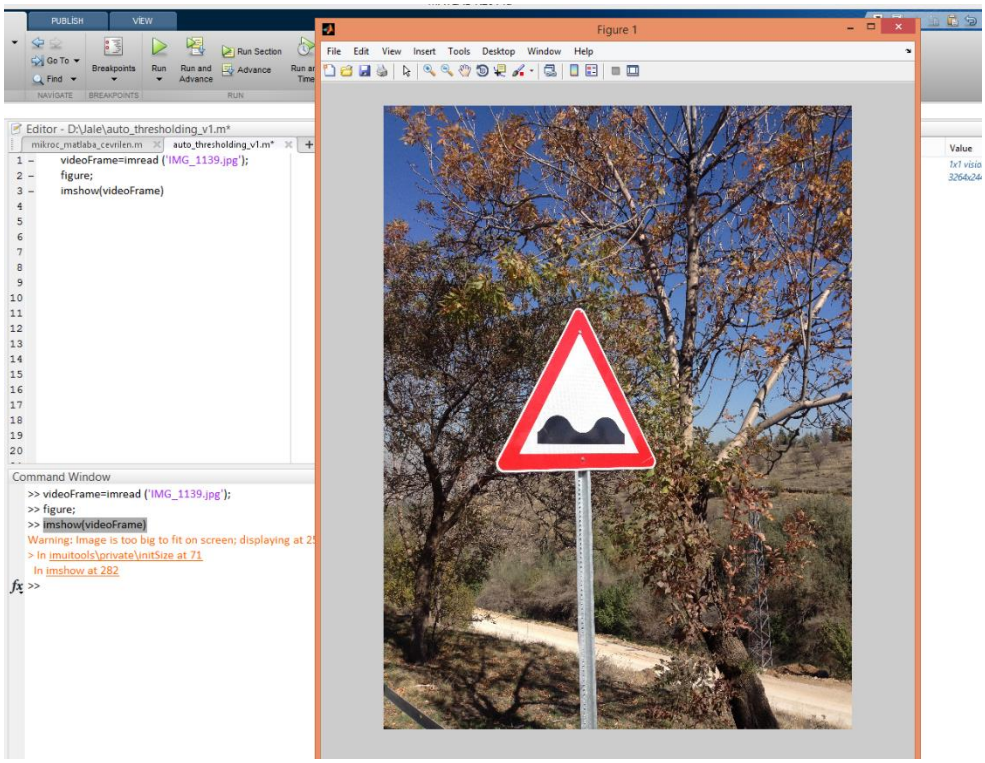
9- Eğer işaret kasis ise sürücüye sesli uyarı verilecektir.

Çevrimdışı analiz yapmak için uygun olan Matlab yazılımı ile bu çalışmalar tezin ilk iki bölümünde başarı ile gerçekleştirilmiş olsa da gerçek zamanlı olarak denendiğinde Matlab analiz yapamamıştır. Zaten raporlanan diğer çalışmalar incelendiğinde Matlab ile yapılan gerçek zamanlı çalışmalardan elde edilen düşük performanslı sonuçlar görülebilmektedir [21].

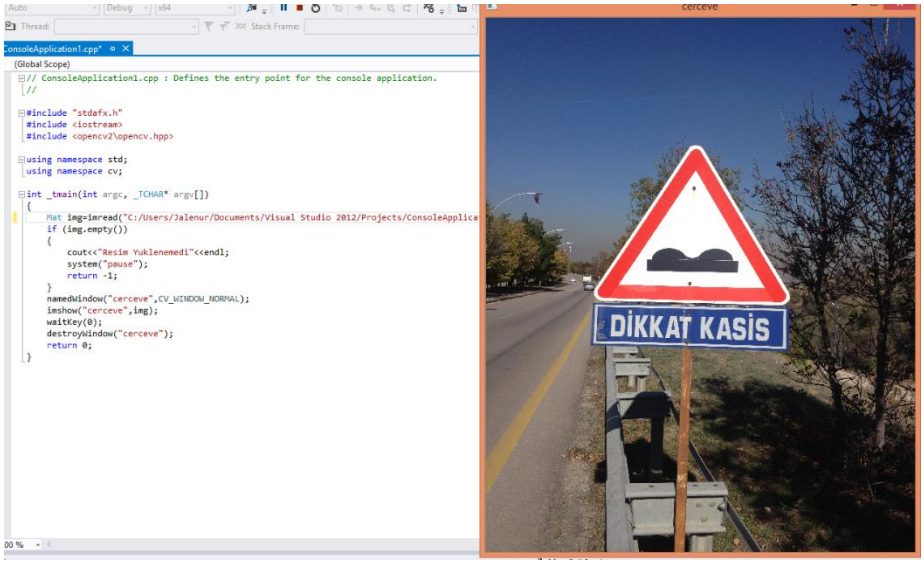
## 5.1. OpenCV Açık Kaynak Kodlu Görüntü İşleme Kütüphanesi

Bu tez çalışması kapsamında yukarıda anlatılan bütün işlemler Opencv açık kaynak kodlu kütüphane ile ayrı ayrı gerçekleştirilecektir. Opencv açık kaynak kodlu kütüphanesi her ne kadar hız ve efektif çalışma yönünden birtakım avantajlara sahip olsada kütüphane içerisinden basit işlemler için bile çok sayıda temel seviyede C++ kodunun yazılması gerekliliği önemli bir dezavantajdır. Buradaki zorluğa örnek verecek olursak Matlab'da sadece bilgisayarda bulunan bir resmi almak için yazılan kod ve aynı işlemin Opencv'de yapılması için yazılan kod Ek-4'de verilmiştir. Ekteki kod C++ ile yazılmıştır ve derleyici olarak Visual Studio 2012 kullanılmıştır. Bu aşamaların gerçekleştirilmesi için c programlama dili bilgisi C++ 'a aktarılmıştır ve Opencv kütüphanesi ile C++ 'ın bütünleştirilmesi için çaba sarf edilmiştir.

Yukarıdaki resim alma kodundan da görüleceği üzere en genel manada Matlab ve Opencv aynı çıktıları vermektedir. Buradan çıkarılacak sonuç ise çok emek harcayarak yazılan C++ kodlarının Java arayüzü ile programlanan Matlab'a göre çok hızlı çalışmasıdır.
Aşağıda iki farklı yazılım ile alınan resimlerin program çıktıları Şekil 23 ve Şekil 24'de görülmektedir.




Şekil 23. Matlab ile görüntü alma çıktısı




Şekil 24. Opencv ile görüntü alma çıktısı

İlerleyen bölümlerde yapılan işlemler adım adım detaylı olarak ele alınmış ve çıktıları her aşamada ürettirilen şekiller ile paylaşılmıştır. Opencv kütüphanesinin kullanımı ile yeterli sayıda kaynak bulunamaması yeni kullanıcıları Opencv yerine daha yavaş çalışan fakat birçok fonksiyonu hazır olarak barındıran Matlab programlama ortamına yönlendirmektedir. Bu çalışma içerisinde yapılan işlemlerin özü olarak

nitelendirilebilecek yazılan bazı fonksiyonların temel teşkil etmesi açısından tez içerisinde kısa kısa paylaşılması uygun görülmüştür.

## 5.2. OpenCV Kasis İşareti Belirleme Çalışmaları

Yapılan çalışmalar, ilk olarak tek bir resim çerçevesi üzerinde gerçekleştirilmiştir. Gösterilen sonuçlar tek bir resim üzerinden olacaktır. Yapılan her çalışma altbaşlıklarda adım adım gösterilmektedir. Bu bölümün bitiminde ilgili algoritmanın anlatımı tamamlanmış olup sonuçlar bölümünde istatistiksel olarak algoritmanın performansı analiz edilmiştir. Kütüphanenin çalışması için gereken "header" dosyaları aşağıda gösterilmiştir.

### 5.2.1. Kullanılan Başlık Dosyalarının Tanıtılması

Bu başlık dosyaların C ve C++ dilleri için özelleşmiş olup kullanılacak bazı temel fonksiyonları içermektedirler. Bu başlık dosyaları kodu daha verimli yazmak için gerekmektedirler.

Ek-5'de bu tez kapsamında kullanılan başlık dosyaları listelenmiştir. Başlık dosyaları hakkında detaylı bilgiye Opencv resmi yardım sayfasından erişilebilmektedir.

### 5.2.2. Resmin Çalışma Alanına Kaydedilmesi ve Gösterilmesi:

Opencv kütüphanesinde resim alma işlemi standartlaşmış bazı komutlar kullanarak gerçekleştirilir. Bu komutlar kullanılırken dikkat edilmesi gereken dosya yolunun doğru belirtilmesi ve resmi tutacak değişkenin doğru değişken tipinde tanımlanmasıdır. Opencv resimleri tutmak için "Mat" adlı bir değişken tipi kullanmaktadır. Bu değişken tipi kullanılarak gerçekleştirilen kasis işareti okuma işleminin kodu Ek-6'da ve program çıktısı Şekil 25'de gösterilmektedir.


Şekil 25. Resmin Kaydedilmesi ve Görüntülenmesi

Bir sonraki bölümlerde ilk olarak resmin üzerinde gerçekleştirilen renk bölütleme çalışmaları paylaşılmıştır.

### 5.2.3. Renk Bölütleme Çalışmaları

Bu bölümde alt başlıklar halinde renk bölütleme çalışmalarının aşamaları gösterilmektedir. Yine yeni kullanıcılara rahatlık sağlaması açısından bazı önemli bölümler için kısa kod parçaları paylaşılmıştır.

#### 5.2.3.1. Resmin Altuzaylara ayrılması

Bu aşamada resim alt renk uzaylarına ayrılmıştır. Bu aşamanın amacı resmin içinde bulunan kasis işaretinin etrafındaki kırmızı üçgen bölümü ön plana çıkarmaktır. Resimler “Iphone” (Apple Inc.) telefonun kamerası ile çekilmiştir ve 8 bitlik RGB uzayında 3 kanal olarak kayıt altına alınmıştır. Ek 4’deki kod parçası resmi üç alt uzay parçaya ayırır ve her bir parçayı görüntülemektedir. Şekil 26’da kırmızı yeşil ve mavi altuzaylar ayrı ayrı gösterilmiştir.


a) b)


c)

Şekil 26. a. Mavi, b. Yeşil c. Kırmızı Altuzay Resimler

Altuzayda resimler oluşturulduktan sonra bir sonraki aşama bu resimleri basit matematiksel fonksiyonlar kullanarak birtakım işlemlere tabi tutmak ve sonucunda kırmızı renkli bölümleri ön plana çıkarmaktır.

### 5.2.3.2. Fark Resmi Oluşturulması:

Bu bölümde üstte katmanlarına ayrılan resimden sadece kırmızı altuzay resmi alınacaktır ve gri skala resim ile farkına bakılacaktır. Bunun için elimizde olan resim gri skala resme dönüştürülmelidir. Bu işlem için yazılmış çevrim fonksiyonu kullanılmıştır. İlgili kod parçası Ek-8'de ve program çıktısı aşağıda görülmektedir. Şekil 27'de gri resim ve bunların farkı olan resim gösterilmektedir. Görüldüğü gibi kırmızı üçgen levhanın kırmızı bölümü başarı ile ön plana çıkarılmıştır.


a) b)

Şekil 27. a. Gri Skala b. Kırmızı Alt Uzay Fark Resmi

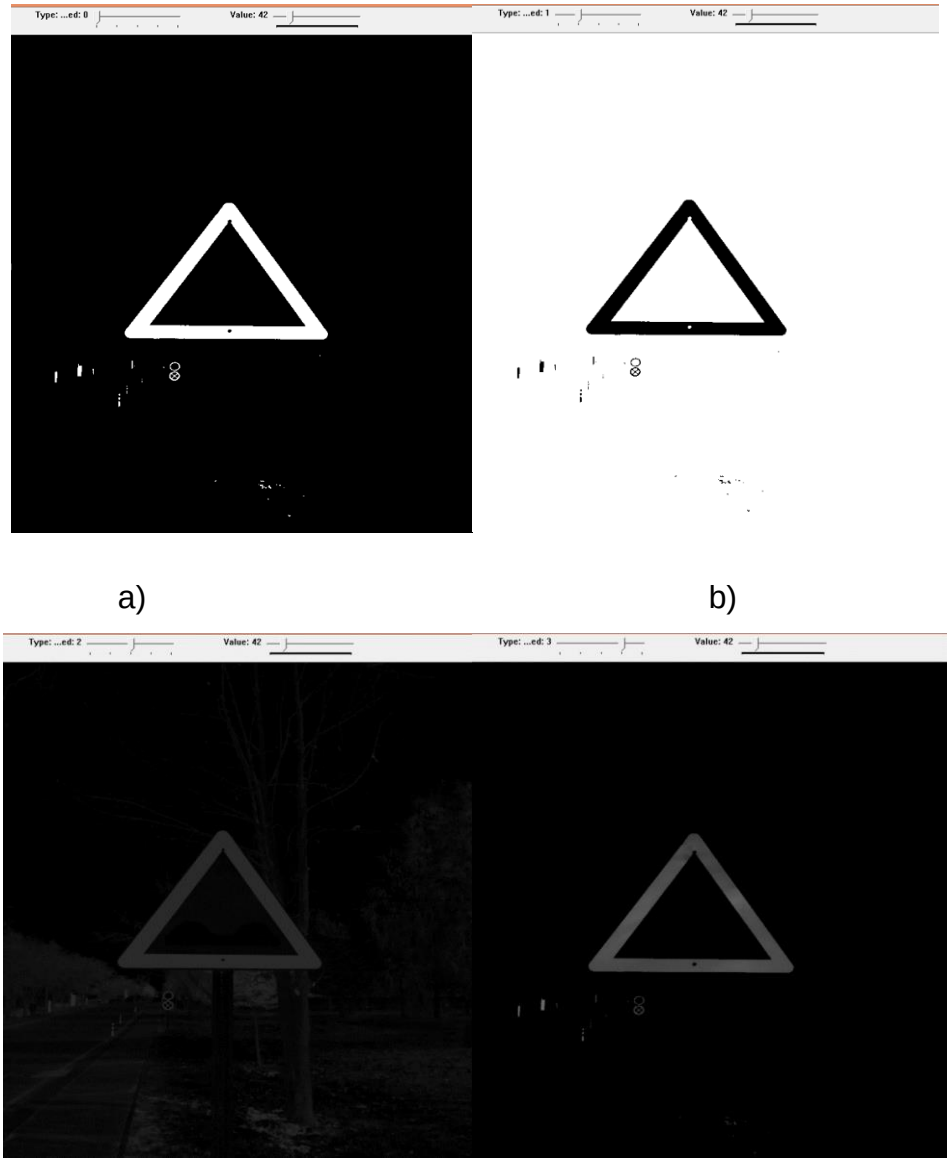
Bu aşamada fark resmini bir eşik değeri yardımıyla siyah beyaz resme dönüştürmek gerekir. Bunun için de birtakım çalışmalar yapılmıştır.

### 5.2.3.3. Sabit Eşik Değer ile Siyah Beyaza Dönüşüm

Opencv içerisinde siyah beyaza dönüşüm için 4 farklı yöntem bulunmaktadır. Bunların içerisinde 0-255 arası eşik değerleri ile ayarlamalar yapılabilmektedir. Bu

işlemi görsel olarak anlayabilmek için Ek-9'daki kod yazılmıştır ve sonuçları Şekil 28'de paylaşılmıştır.

İlgili çalışmaların sonucu sırası ile Şekil 28.a, 28.b, 28.c ve 28.d de görülebilmektedir.




a) b)

c) d)

Şekil 28. Aynı eşik değerinde dört adet eşik değerleme yönteminin incelenmesi

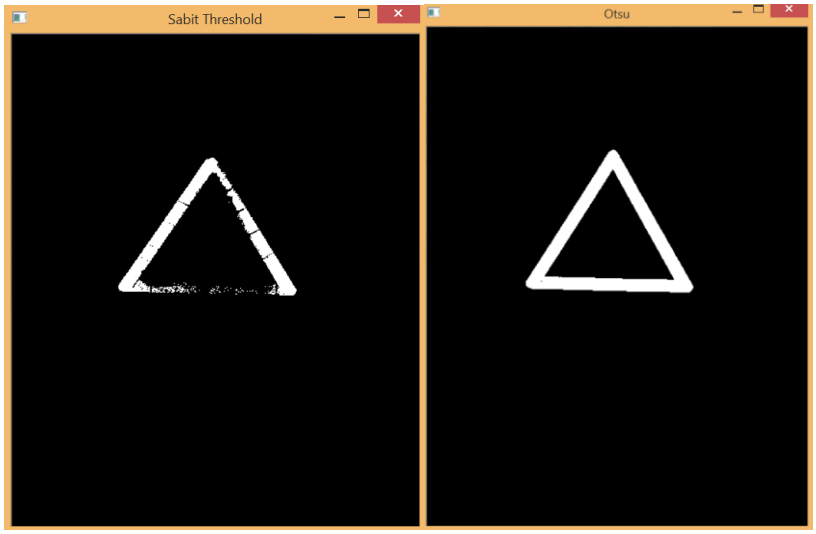
Yapılacak çalışmalar için otomatik eşik değeri belirlemede en uygun yöntemin "Tip 0" olduğu görülebilmektedir. Aslında ne kadar "Tip 1" de uygun gözükse de "Tip 0" kullanılmıştır çünkü çalışmada lazım olan kırmızı üçgen bölümün beyaz renk ile ön plana çıkartılmış olmasıdır. "Tip 1" zaten "Tip 0" 'ın renk olarak ters çevrilmiş halidir.

#### 5.2.3.4. Sabit Eşik Değer ile Siyah Beyaza Dönüşüm

Elde edilen bu fark resmi tek kanaldı ve 8 bit boyutundaydı. Yani resim piksel yoğunlukları 0-255 arasındadır. Şimdiki amaç ise bu resimdeki her bir pikseli 0 ve 1 e eşlemektir. Bu işlemi yapmak için literatürde farklı yöntemler bulunmaktadır. Bunlardan biri sabit bir eşik değeri (threshold) eşliğinde bu eşleme işleminin yapılmasıdır ve 5.2.3.3 bölümünde anlatılmıştır. Bir diğer yöntem ise her bir gelen yeni resim için ortam ışığına vb. faktörlere bağlı olarak dinamik olarak uygun eşik değerinin bulunmasıdır.

Bu çalışma kapsamında iki ana yöntemde incelenmiştir ve çıktılarına bakılarak en uygun yöntemin hangisi olduğuna karar verilmiştir.

Ek-10'da bu yöntemlerin uygulanması için gerekli kod parçası verilmiştir. Şekil 29'da ilk olarak sabit eşik değeri kullanımının sonucu kıyaslama yapılabilmesi için farklı bir resimde tekrar verilmiştir. Diğer resimde ise dinamik eşik değeri kullanmanın sonucu paylaşılmıştır. Matlab'da yapılan çalışmaların sonucu ile paralel olarak yine Otsu Dinamik Eşik Değeri Belirleme yöntemi en iyi sonucu vermiştir.




a) b)

Şekil 29. a. Sabit Eşik Değeri Kullanımı b. Dinamik (Otsu) Eşik Değeri Kullanımı

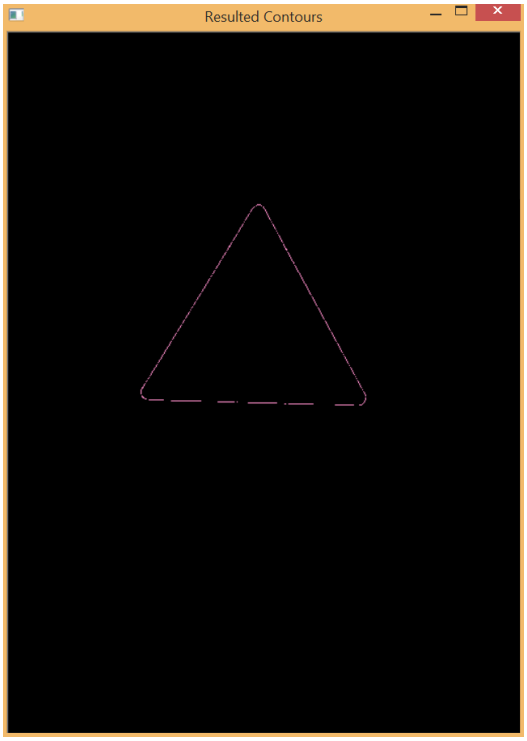
Böylece renk bölütleme ve siyah beyaza dönüşüm çalışmaları bütün resimleri için teker teker denenmiş ve kırmızı altuzay resim ile gri resmin farkının alınıp Otsu Dinamik Eşik Değeri Yöntemi ile siyah beyaza dönüştürülmesi uygun yöntem olarak belirlenmiştir.

### 5.2.4. Üçgen Şeklinin Tespit Çalışmaları

Bir önceki bölümde anlatılan işlemler yapıldığında elde kalan tek kanallı ve içerisinde üçgen levha ile büyüklü küçüklü birçok başka nesneyi barındıran siyah-beyaz bir resimdir. Bu resme birtakım işlemler uygulanarak ilgili üçgen şekli diğerlerinden ayrılması kalan şekillerin ise filtrelenerek atılması gerekmektedir.

#### 5.2.4.1. Levha Dış Şeklinin (Contour) Belirlenmesi:

Bu bölümde ön plana çıkarılan kırmızı üçgen levhanın etrafının kenarlarının belirlenmesidir. Bu işlem kodun çok önemli bir bölümünü teşkil etmektedir. Çünkü bulunan şeklin üçgen mi olup olmadığı bu bölüm sayesinde bilinmektedir. İlgili bölümün kod parçası Ek-11'de ve program çıktısı aşağıda paylaşılmıştır.



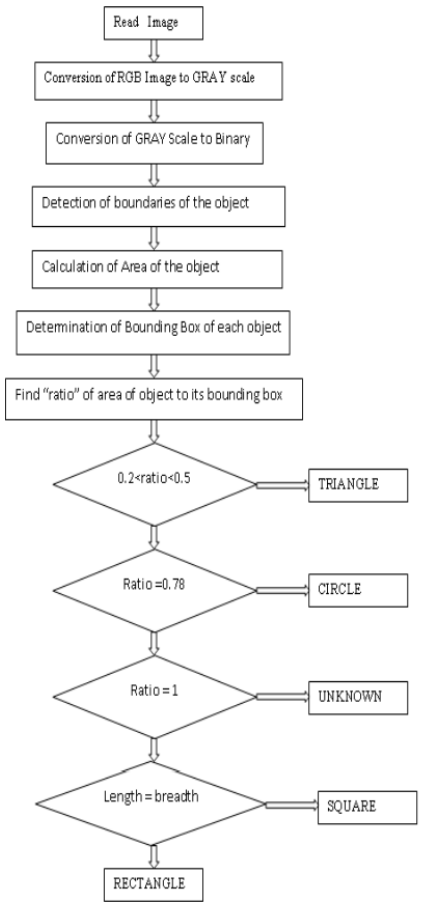

Şekil 30. Bulunan Dış Kenar Çizimi

Burada dikkat edilmesi gereken bazı hususlar bulunmaktadır. Öncelikle literatürde çok sayıda dış çizim bulma algoritması bulunmaktadır. Opencv içerisindeki dahili dış çizim bulma algoritması içerisinde pek değişiklik yapılmasa da kullanılan hiyerarşik yapıda çeşitli değişiklikler performansı artırıcı yönde yapılmıştır.

Dış çizim bulma algoritmalarının performanslı çalışabilmeleri için aşağıdaki şartların sağlanması önemlidir ve bu işlemler gerçekleştirilmiştir.

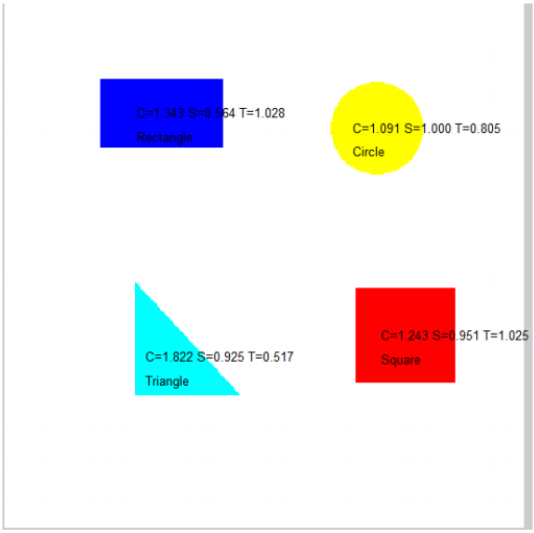
1) Doğru dış çizimlerin elde edilebilmesi için ikili sayı sistemindeki resimler ile çalışılmalıdır. Zaten alınan fark resmi dinamik eşik değerleri yardımı ile siyah ve beyaza dönüştürülmektedir.
2) Dış çizim bulma algoritması arka plandan nesne çıkartma algoritmaları ile aynı mantıkta çalışmaktadır. Dolayısı ile dış çizimi bulunacak nesnenin beyaz arka planın siyah olması gerekmektedir. Yine dinamik eşik değeri yöntemi ile siyah beyaza çevrilen resimler bu şartı sağlamaktadırlar.

Bilgisayarlı görü çalışmalarından en önemlilerinden biri değişik geometrik şekillerin tespit edilmesi ve sınıflandırılmasıdır. Bu konuda literatürde yapılmış çok sayıda çalışma mevcuttur. Benzetim ortamında yapılan çalışmalarda çalışma performansı ve işlem yükü gibi kısıtlar göz önünde fazla bulundurulmadığından farklı şekil tanıma algoritmaları duruma göre rahatlıkla denenebilmektedir. Chhya ve arkadaşlarının yaptığı çalışmada bazı temel geometrik şekillerin nasıl tespit edileceği ile ilgili yöntemler önerilmektedir. Şekillerin tespit edilmesi için önerilen algoritmalardan biri akış diyagramı olarak Şekil 31'de gösterilmektedir [22].



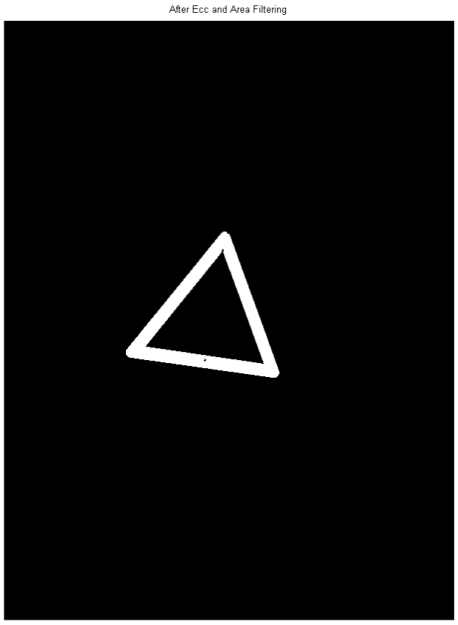

Şekil 31. Şekil Tespiti için Akış Diyagramı [22]

Önerilen algoritma gerçekten de benzetim ortamında bu tez kapsamında denenmiştir ve başarı ile çalıştığı görülmektedir. Bulunan benzetim sonuçları Şekil 32'de paylaşılmıştır.




Şekil 32. Geometrik Şekil Tanıma Algoritmasının Başarımı

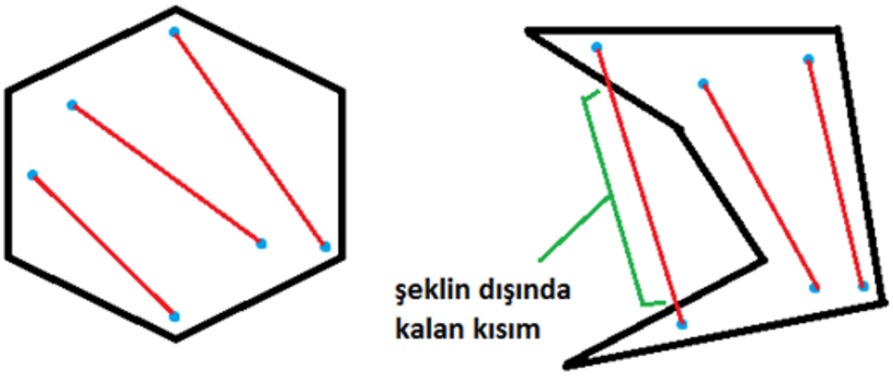
Bu çalışmanın sonuçlanması için önerilen algoritmanın işlem yükü açısından incelenmesi çalışmada verilmemiştir. Ama kasis işareti tanıma için aynı metodun bir benzeri önceki bu tez çalışmalarında mevcuttur ve gerçek zamanlı çalışma için işlem yükü açısından uygun olmadığı yayınlanan tebliğde belirtilmiştir [16]. Şekil 33'de Matlab ortamında bulunan üçgen levha gösterilmiştir. Bu işlemin getirdiği işlem yükü Matlab tarafından olması gerekenden fazla olarak hesaplanmıştır ve gerçek zamanlı olarak çalışmaya uygun olmadığı raporlanmıştır.




Şekil 33. Matlab ortamında üçgen şeklin tespit edilmesi

Ayrıca geometrik şekil bulma algoritmalarından bir diğeri de bulunan şekillerin etrafına çizilen doğruların kesişim noktalarındaki açıların tespit edilmesidir. Mesela eşkenar üçgen bir şeklin tespiti için çizilen üç adet doğrunun kesişim noktalarındaki açıya bakılabilir ve bu açının üç yerde de $60^0$ civarında olması gerekir. Yine bu yöntem de benzetim ortamında başarı ile çalışmaktadır ama gerçek zamanlı performansı düşüktür.

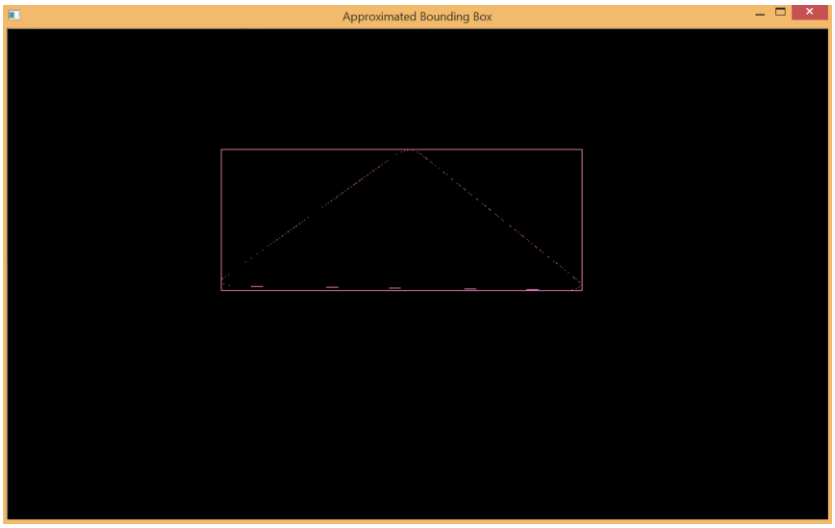
Önceden yapılan çalışmalarda kasis işaretini çevreleyen üçgen levha rahatlıkla bazı renk tabanlı görüntü işleme algoritmaları ile çıkartılmıştı. Daha sonra Matlab içerisinde bulunan bazı hazır şekil tanıma algoritmaları ile bulunan şeklin üçgen olup olmadığı belirlenebilmişti. Artık Opencv de işlem yükü düşük ama doğruluğu yüksek bir yöntem denenecektir. Aşağıda bu yöntemin adımlarından yöntem ile ilgili koddan ve en son olarak sonuçlarından bahsedilecektir. Öncelikle yapılan işlemi özetlemek gerekirse ilk olarak bulunan dış çerçevenin etrafına çokgen şekli çizilir Daha sonra çizilen bu çokgeni oluşturan doğrular sayılır. Bu şekilde doğru sayısı üç çıkar ise bulunan şeklin üçgen olduğuna hüküm verilebilir. Ayrıca algoritmanın gürbüzlüğünü artırmak için bulunan şeklin konkav mı konveks mi olduğuna da bakılmıştır. Fikir vermesi açısından Şekil 34.a ve 34.b'de konkav ve konveks şekiller gösterilmiştir.




Şekil 34. a. Konveks b. Konkav

Üçgen şeklinin bu tanımlamalara göre konveks olduğu açıktır ve istenmeyen nesnelerin gürültü olarak belirlenebilmesi için filtre parametresi olarak kullanılmıştır. Ayrıca yine işlem yükünü artırmayacak bir başka filtrenin de kullanılması algoritmanın hızlı çalışması için gereklidir. Bu filtre ise alan boyutuna göre davranan bir filtredir. Üçgen olan nesnenin belirli bir mesafeden bile görüntüsü alınsa alanın belirli bir alt piksel değerinin üzerindedir. Dolayısı ile bu alt değer filtreleme de kullanılmıştır. Algoritma alt değerin altında bulduğu üç köşeli nesnelerin hepsini almamaktadır. Eğer bu filtreleme kullanılmasaydı ilerleyen aşamalarda gürültüler giderilse bile işlem yükü çok artmış olacaktır. Aşağıdaki şekilde bu yöntemlerin kullanılması sonucu elde edilmiş üçgen şeklin etrafına çevreleyen kare çizilmiş ve gösterilmiştir. Burada akla neden üçgenin etrafına çizilmesi gereken çokgenin üçgen değil de kare olduğu sorusu gelebilir. Bunun nedeni opencv içerisinde hazır poligon (çokgen) çizim fonksiyonu bulunmamaktadır. Ancak çevreleyen kutu çizimi hazır fonksiyon olarak bulunmaktadır.

Şekil 35'de bahsedilen bu sonuç gösterilmiştir.




Şekil 35. Belirlenen üçgenin etrafına çizilen çevreleyen kutu

Bahsedilen kod bölümü Ek-12'de paylaşılmıştır.

Ek-13'de ise çokgenin çizimi için yazılan kod parçası paylaşılmıştır.

### 5.2.5. Kasis İşaretinin Algılanması

Çalışmanın bu bölümü üç ana başlıktan oluşmaktadır ve her bir ana başlık kendi içerisinde gerçek zamanlı çalışma için kritik noktalar barındırmaktadır.

Raporun ilk bölümünde resmin boyutunun gerçek zamanlı olarak çalışmaya uygun hale getirilmesi incelenecektir. Sonuçlar paylaşılacaktır ve yapılan bu işlemin işlem yükünü ne kadar hafiflettiği ile ilgili somut veriler gösterilecektir. Raporun ikinci bölümünde ise yine bir boyut küçültme işlemine benzer olan, resmi belirli bölgeden kesme işlemi anlatılacaktır. Bu bölüm farklı görüntü işleme teknikleri hususunda bilgi sahibi olmayı gerektirmektedir. İlgilenilen alanın nasıl tespit edildiği, maske matris kullanılması, resim kesme ve resim üzerine maske matrisin oturtulması etraflıca incelenecektir. Raporun son aşaması ise üçgen levhanın içerisindeki işaretin kasis olup olmadığının tespit edilmesine yönelik bir bölümdür. Bu bölümde gerçek zamanlı çalışmaya uygun bir yöntem incelenmiştir ve sonuçları paylaşılacaktır.

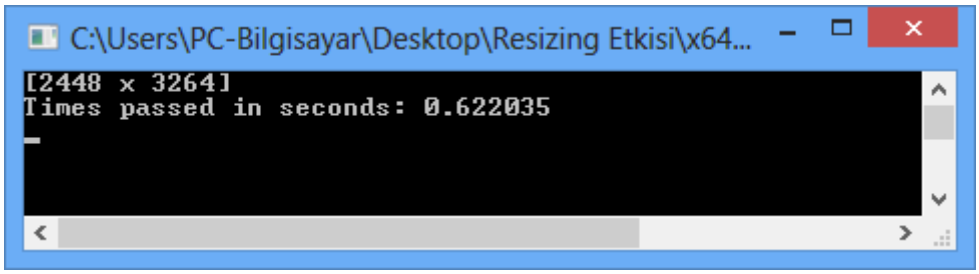
#### 5.2.5.1. Yeniden Boyutlandırma

Her ne kadar çalışma gerçek zamanlı çalışması için tasarlanmış olsa da ilk aşamada çoğu test ve çalışma durağan resimler yani fotoğraflar üzerinden gerçekleştirilmiştir. Çalışma için oluşturulan veri seti yaklaşık 90 resimden oluşmaktadır ve her birisi mobil telefon kamerası ile elde edilmiştir. Standart olarak elde edilen resimler

(herhangi bir filtreleme yazılımı kullanmadan) 2448x3264 boyutundadır ve gerçek zamanlı çalışma için bu boyut uygun değildir. Bu boyutun istenilen boyuta istenildiği gibi düşürülmesi gerekmektedir. Bu elde edilecek yeni boyutun bulunması ise rastgele olmayıp belirli bir kural dahilindedir. Öyle ki boyut gerçek zamanlı çalışma için olabildiğince düşürülmelidir fakat resimdeki önemli bilgi taşıyan özelliklerinde kaybolmaması gerekmektedir. Dolayısı ile aşağıda 3 farklı yeniden boyutlandırma işlemleri ve sonuçları bu çerçevede incelenmiştir. Bu işlemin etkisini görebilmek için veriseti içerisinde bol miktarda gürültü barındıran bir resim seçilmiştir. Bu sayede yeniden boyutlandırma işleminin işlem yüküne getirdiği saniye olarak olumlu etki ölçülecektir bir diğer avantaj ise geliştirilen sistemin ne derece gürbüz olduğu belirlenmiştir. Çünkü seçilen resim Karayolları'nın kullandığı eski tip kasis işaretlerindendir.

İlk olarak gerçek boyutlarda çalışıldığında ki orijinal resim ve elde edilen yeniden boyutlandırılmış resim gösterilmiştir (Şekil 36.a ve Şekil 36.b.) İlk olarak orijinal boyut (2448x3264 piksel) çalışmasının sonucu verilmiştir.


a) b)



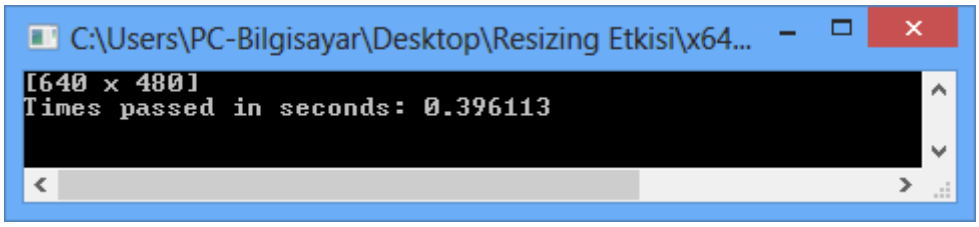

c)

Şekil 36. a. 2448x3264 boyutunda orijinal resim b. Bu boyutta üçgen levhanın dış çerçevesinin belirlenmesi c. Boyut (piksel) ve işlem yükü (sn)

İkinci olarak ise resim 640x480 boyutunda indirgenmiştir ve sonuçları ele alınmıştır. Boyut olarak 640x480 seçilmiştir ve elde edilen çıktılar Şekil 37.a ve 37.b'de gösterilmiştir.

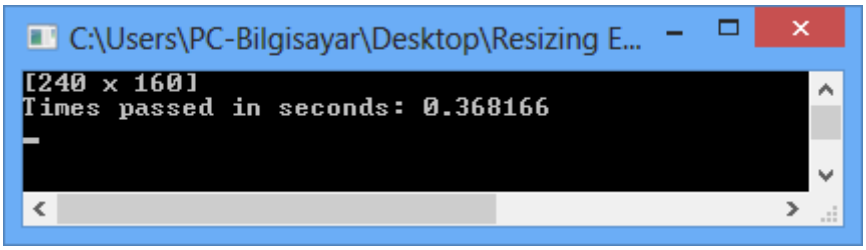

Şekil 37. a. 640x480 boyutunda orijinal resim b. Bu boyutta üçgen levhanın dış çerçevesinin belirlenmesi

Burada çok önemli bir bilgi elde edilmiştir; resmin boyutu yaklaşık 0.25 katına gelmesine rağmen görüntüde gözün algılayabildiği bir kayıp söz konusu değildir dolayısı ile çalışmak için uygundur. Bu boyut indirgeme işleminin performansa olan etkisi gözlemleyebilmek için Şekil 38 incelenebilir.




Şekil 38. Boyut (piksel) ve işlem yükü (sn)

Son olarak boyut olarak 240x160 seçilmiştir ve sonuçlar incelenmiştir. (Şekil 39.a, Şekil 39.b, Şekil 39.c)


Şekil 39. a. 240x160 boyutunda orijinal resim b. Bu boyutta üçgen levhanın dış çerçevesinin belirlenmesi c. Boyut (piksel) ve işlem yükü (sn)

Bu noktada ilgi çekici sonuçlar bulunmuştur ve uygun boyuta karar verilmiştir. 640x480 piksel boyutundan 240x120 piksel boyutuna düşüldüğünde görüldüğü gibi üçgen levhanın kenar bölgelerinde çözünürlük azalmasından dolayı bozulmalar

(artifact) olmuştur ve bu bozulmalar dış çizim bulunması aşamasında yanlış sonuçlar bulunmasına yol açacaktır. Bir diğer husus ise boyutun belli bir seviyeden daha aşağı düşürülmesiyle doğrusal olarak işlem yükünün de azalmadığıdır. Çünkü orijinal boyutta yaklaşık 0.62 saniye olan işlem yükü, resim dörtte bir boyutuna düştüğünde yaklaşık yarısına düşmekteydi ama son boyut indirgemede resim yaklaşık yarısına düşmekte iken işlem yükü süresinde 0.1'lik bir azalma olmuştur ve bu sonuç tatmin edici değildir. Çalışmak için 640x480 piksel görüntü boyutu uygun olarak seçilmiştir.

#### 5.2.5.2. Resmin kesilmesi

Bu bölüm yine gerçek zamanlı çalışmalar hedef alınarak oluşturulmuştur çünkü benzetim ortamında çalışıldığında bu bölümü yapmanın algoritmanın doğruluğunu artırmayacağı açıktır. Burada amaç işlem yükünü minimum seviyede tutabilmektir. Geride kalan bölümlerde son bölümünde üçgen şeklin etrafına dış çizim (contour) yapılmıştı ve daha sonra bu dış çizimin gerçekten de üçgen olup olmadığına bakılmıştı. Şimdi ise gerçekten üçgen olduğu bilinen bir dış çizimin orijinal resimden çıkarılıp yeni bir resim olarak kayıt altına alınması gerçekleştirilecektir. Bu aşama çok önemlidir çünkü üçgen levha bulunduktan sonra bir sonraki aşama kasis işaretinin bulunmasıdır ve kasis işaretinin orijinal resimde taranması ve bulunması çok fazladan işlem yükü getirmektedir. Bunun yerine kasis işareti yeni oluşturulmuş küçük boyutlu ve sadece üçgen levhayı içeren bir yeni resimde aranacaktır.

Bu işlem üç aşamada gerçekleştirilir. Bunların ilki orijinal resimde ilgili dış çizimin etrafına çevreleyen dikdörtgen çizilmesidir. Daha sonra bu çevreleyen dikdörtgenin boyutu hesaplanır ve bu boyutta bir boş matris oluşturulur. Orijinal resimdeki ilgili bölüm bu bölüm üzerine oturtulur ve yeni resim elde edilir. Şekil 40'da bu işlemin basamakları gösterilmiştir.


Şekil 40. a. Orijinal Resimden çıkarılacak bölüm b. Kesilmiş Bölüm

Kasis işareti artık Şekil 40.a'da değil Şekil 40.b'de aranacaktır. Bu işlemlerin getirdiği performans avantajları açıktır.

### 5.2.5.3. Kasis İşaretinin Belirlenmesi için Önerilen Yöntemin İncelenmesi

Bu bölüm için birtakım testler yapılmış olup mesafenin etkisi ile resimlerdeki siyah piksellerin sayısının fazlaca değişmeye neden olması çalışma için biraz daha zaman ayrılmasına zemin oluşturmuştur.

Çalışmanın bu bölümünde dış çerçevesi bulunan kasis işaretini barındıran üçgen levhanın iç kısmına odaklanılmıştır. Üçgen içerisinde bulunan kasis işaretinin tanınması için bir yöntem denenmiştir ve sonuçları paylaşılmıştır. Bu işlem dört aşamada gerçekleştirilmektedir. İlk aşamada bulunan ve çıkarılan üçgen levha üzerinde yeniden boyutlandırma işlemi gerçekleştirilmiştir.

i) Yeniden Boyutlandırma:

Bu işlemin amacı kasis işareti uzakta da olsa yakında da olsa aynı boyuta getirebilmektir. Bu işlem için en kolay yol olan linear interpolasyon yöntemini kullanan yeniden boyutlandırma kullanılmıştır. Bulunan her bir üçgen 100x100'lük yeni resimler haline getirilmiştir. Ayrıca bu işlem sayesinde işlem yükünün fazlası da azaltılmıştır. Yapılan işlemlerin kolay anlaşılabilmesi için Şekil 41.a'da orijinal resim gösterilmiştir. Ayrıca Şekil 41.b ve 41.c'de çıkarılmış üçgen levha ve üzerine yeniden boyutlandırma işlemi uygulanmış üçgen levha gösterilmektedir. Yeniden

boyutlandırma öncesi resim 171x157 piksel iken yeniden boyutlandırma sonrası 100x100 piksel boyutuna getirilmiştir.

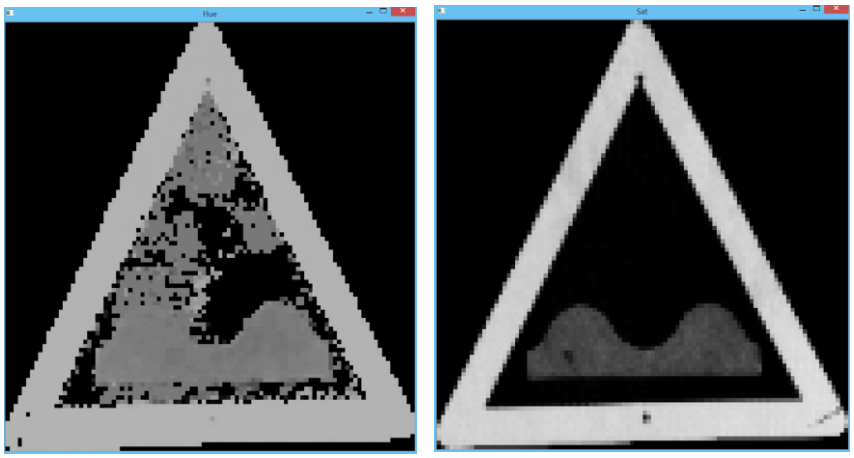

a)


b) c)

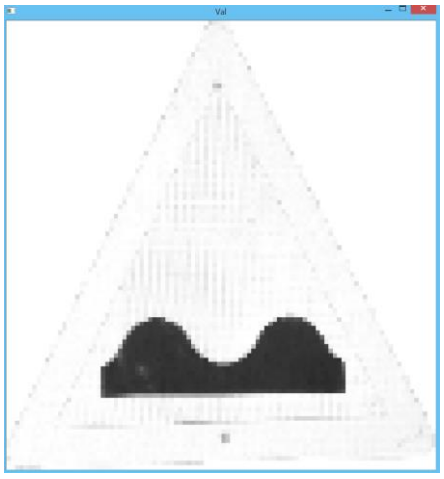
Şekil 41. a. Orijinal Çerçeve b. çıkarılmış üçgen levha c. yeniden boyutlandırılmış üçgen levha

ii) Renk Uzayı Dönüşümü

Bir sonraki aşamada yeniden boyutlandırılmış resim için siyah bölümler ön plana çıkarılacaktır ki kasis işareti rahat tanınabilsin. Bu işlemin ilk aşaması siyah bölümü ön plana çıkarmak için kullanılan renk uzayının seçilmesidir. Bütün renk uzayları ve alt segmentleri denenmiştir. Ve uygun olarak Hue, Saturation ve Value (HSV) renk uzayına karar verilmiştir. Bu tip çalışmalar için RGB uzayının uygun olmadığı çeşitli çalışmalarda raporlanmıştır. Şekil 42'de HSV uzayına çevrilmiş resmin Hue, Saturation ve Value kanalları ayrı ayrı gösterilmiştir.



a) b)



c)

Şekil 42. Sırası ile a. Hue, b. Saturation c. Value Kanalları

**HSV** (**H**ue, **S**aturation, **V**alue) veya **HSB** (**H**ue, **S**aturation, **B**rightness) renk uzayı, renkleri sırasıyla renk özü, doygunluk ve parlaklık olarak tanımlar [23].

- Renk özü, rengin baskın dalga uzunluğunu belirler, örneğin sarı, mavi, yeşil, vb.
  - Açısal bir değerdir 0° - 360°, bazı uygulamalarda ise 0-100 arası olağanlaştırılır.

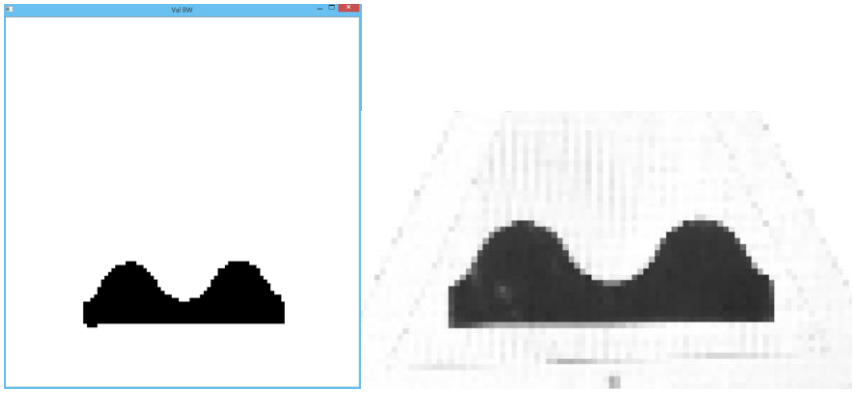
- Doygunluk, rengin "canlılığını" belirler. Yüksek doygunluk canlı renklere neden olurken, düşük olasılık rengin gri tonlarına yaklaşmasına neden olur.
  - 0-100 arasında değişir.

- Parlaklık ise rengin aydınlığını yani içindeki beyaz oranını belirler.
  - 0-100 arasından değişir.

Şekil 42'den görüldüğü üzere en uygun uzay olarak HSV en uygun alt uzay olarak Value alt uzayı seçilmiştir. Çünkü siyah kasis en fazla bu alt uzayda ön plandadır.

iii) Kasis işaretinin belirlenmesi

Bu aşamada ise ön plana çıkarılan kasisi işareti gerçekten de kasis işareti mi yoksa farklı bir şekil mi ona bakılmıştır. Bunun için kasis işaretinin varlığına tekabül eden siyah pikseller toplam piksellerin içinde sayılmıştır. Resimdeki toplam piksel sayısı 100x100=10000 pikseldir. Bu piksellerin bir kısmı siyah bir kısmı da beyazdır ve siyahlar kasis işaretine denk gelmektedir. Bu siyahların sayısı eldeki veri tabanı için sayıldığında 300 ile 900 piksel arasında çıkmıştır. Buradan şu sonuç çıkarılmıştır. Eldeki siyah beyaz resimdeki siyahların sayısı 300-900 piksel arasında ise kasis olarak sınıflandırılabilir. Şekil 43'de ilgili sınıflandırma ve sonucu gösterilmiştir. İlk resimde Value altuzayının siyah beyaza dönüşmüş hali ikinci resimde ise bulunan kasis işareti gösterilmektedir.



Şekil 43. Siyah Beyaza dönüşmüş Value Altuzayı ve Bulunan Kasis işareti

## 5.3. Gerçek Zamanlı Ortamında Elde Edilen Sonuçların Yorumlanması

Tasarlanan gerçek zamanlı sistem tezin ilk bölümünde tasarlanan benzetim ortamındaki sistem ile aynı istatistiki başarıya sahiptir. Ama sistem gerçek zamanlı çalışacak şekilde tasarlanmış olsa da veritabanındaki resimler üzerinde çalışma yapıldığı için aslında tam anlamıyla gerçek zamanlı bir çalışma olmamaktadır. Bu bağlamda araç içerisine yerleştirilen bir Logitech Web Camera ve Windows işletim sistemi olan bir dizüstü bilgisayar ile tasarlanan sistemin gerçek zamanlı performansı ölçülmüştür. Sistem yolda araç ile düşük hızda seyrederken Hacettepe Üniversitesi Kampüsü içerisindeki birçok kasis tabelasını rahatlıkla tanımıştır. Hatta kampüs içerisinde bulunan eski tipteki kasis işaretlerini de rahatlıkla tanıyıp kullanıcıya sesli

uyarı vermiştir. Çalışmanın eski tipte bir web kamerası ile yapılıyor olması ve Windows işletim sistemi tabanlı bir diz üstü bilgisayar da işlemci üzerinde çalışılması büyük dezavantajlardandır.

Bilinmektedir ki, görüntü işleme, sinyal işleme, veri madenciliği vb. gibi birçok çalışma Windows işletim sistemi üzerinde yapılmamaktadır. Algoritma ve uygulama geliştirme, network kurma ve işletme gibi görevlerde Linux tabanlı işletim daha sistemlerinin hızlı olduğu açıktır. Buna şöyle bir örnek verilebilir. Yeni kurulmuş bir Windows sistem açıldığı anda 700 MB-1 GB arası hafıza tüketirken yeni kurulmuş bir Ubuntu'da 250 MB gibi rakamlar görülebilir [24]. Sonuç olarak bu çalışmanın Linux tabanlı bir diz üstü bilgisayarda çok daha hızlı çalışmasını beklemek mantıklıdır.

İkinci kısıt olan web kamerası ise çok önemli bir rol oynamaktadır. Çünkü alınan görüntüler saniyede 20 çerçeve olarak maksimum kaydedilebilmektedir. Ve hareketli nesneler üzerinde yapılan çalışmalarda sıklıkla karşılaşılan, hareketten dolayı bozulma kavramı sıklıkla çalışmaları etkilemiştir. Kameranın getirdiği bir diğer kısıt ise ışık seviyelerindeki ani değişimler sonucu odaklanma sürecinin çok uzun olmasıdır. Bu süreç yaşanırken denk gelebilecek bir kasis işareti kaçırılabilecektir.

Son olarak kullanılan donanım elemanı olarak bu tez çalışması kapsamında bir adet Logitech marka standart web kamerası ve orta donanım seviyesine sahip tam şarj süresinde Lenovo dizüstü bilgisayar kullanılmıştır. Eğer endüstriyel manada görüntü işlemek için kullanılan saniyede alabildiği çerçeve sayısı 100-120 civarı olan bir kamera kullanılırsa çalışmanın performansının çok yüksek olacağı beklenmektedir.

# 6. SONUÇLAR

Trafik işaretlerinin günümüz araçları tarafından algılanıp sürücüye çeşitli uyarılar vermesi fikri ışığında bu tez içerisindeki çalışmalar gerçekleştirilmiştir. Yapılan araştırmaların sonucuna göre İngiltere'de sürücülerden bütün trafik işaretlerini tam anlamıyla doğru bilenlerin sayısı çok azdır, ayrıca sürücülerin neredeyse %46'sı trafik işaretlerini takip etmeye çalıştıklarında dikkatlerinin dağıldığını bildirmişlerdir. Bir diğer ilginç taraf ise Ulaşım Bakanlığı'nın verilerine göre İngiltere de 9000 adet yanlış yorumlanan trafik işareti bulunmaktadır [1]. Yapılan trafik kazalarının bazılarından sonra sürücüler kazaya sebebiyet olarak trafik işaretlerini takip etmek zorunda kaldıklarını bildirmişlerdir. Bu bilgiler ışığında önem sırasına göre birçok trafik işaretinin bir elektronik sistem tarafından algılanması fikri ortaya atılmıştır ve son yıllarda çalışmalar ciddi ölçüde ivme kazanmıştır. Kamera teknolojisinin hızla ilerlemesi ve sayısal görüntü verilerini büyük bir hızla işleyebilen işlemcilerin yaygınlaşması bu bilgisayarlı görü çalışmalarının da aynı hızda ilerlemesini sağlamıştır.

Çalışmalar ticari olarak incelendiğinde ilk olarak 2008 yılında Mobileye ve Continental firmalarının ortaya çıkardığı trafik işareti tanıma sistemi, BMW-7 Serisi araçlara takılmıştır bu araçlardaki sistem hız limiti, çocuk uyarısı ve çeşitli yönlere dönüş ile ilgili bilgileri sesli olarak sürücüye bildirmektedir. Günümüzde lüks arabaların çoğunda bu sistemin bir türevi bulunmaktadır. Araçlarda ticari olarak bulunan trafik işareti tanıma sistemlerinin birçoğu hız limiti işaretinin içerisindeki rakamları tanımaya yöneliktir ve bütün trafik işaretlerinin her çeşidini tanıyıp sürücüye uyarı verecek bir sistem henüz bulunmamaktadır. Bunun nedenleri arasında başta hafıza kısıtı ve kullanılacak işlemcinin maliyeti gelir.

Gerçek zamanlı çalışan bir bilgisayarlı görü sistemi tasarlamanın zorlukları aşağıda listelenmiştir.

1- Alınan görüntülerin kalitesi yüksek olduğunda görüntü işleme algoritması iyi çalışmakta fakat kullanılan işlemcinin hafıza problemi ortaya çıkmaktadır.

2- Eğer bir tanıma veya algoritma işlemi yapılacaksa ve yeni gelen görüntü bir veritabanı ile kıyaslanacaksa oluşturulacak veritabanına eğitim seti önceden girilmelidir. Bu da işlemci için ekstra hafıza gerektirmektedir.

3- Alınan görüntülerin saniyedeki toplanma hızı artırıldığında performans artmaktadır fakat hafıza yükü de aynı oranda artmaktadır. Eğer çerçeve toplama hızı düşürülürse bu seferde önemli bilgiler atılan çerçeveler ile kaybolabilmektedir.

Bu problemler göz önünde bulundurulduğunda halen araç içerisine yerleştirilen ve tüm trafik işaretlerini tanıyan bir sistemin neden olmadığı açıktır. Aynı zamanda bu kadar işlemi yapabilmek için kullanacak sistem hem çok pahalı olacak hem de çok büyük hafızalara gereksinim duyacaktır.

Bu tez çalışmasında ise bu kısıtlar göz önünde bulundurulduğunda sadece ilk aşama olarak bir kasis işareti tanıma çalışması gerçekleştirilmiştir. Denenen yöntem bir veritabanı ile kıyaslama yapılmadan sadece gelen görüntü üzerinde aşağıdaki işlemlerin sırası ile yapılmasını içermektedir. Çünkü şablon karşılaştırma yöntemleri çok başarılı bir şekilde nesne tanıma görevlerinde kullanılsa da hem hafızada tutulması gereken verilerin çok sayıda olması hem de eldeki resmin veritabanındaki bütün resimler ile teker teker karşılaştırılmak zorunda olması çok fazla iş yükü getirmektedir. Bu amaçla şablon veya taslak resim ile veritabanındaki diğer resimleri karşılaştırma temelli algoritmalarda bu gerçek zamanlı bölümde kaçınılmıştır.

Tez kapsamı özet olarak ele alınırsa ilk olarak benzetim ortamında kasis işaretini tanıyabilen bir sistem Matlab programı aracılığıyla gerçekleştirilmiştir. Bu sistem sabit oluşturulmuş bir veritabanı üzerinden çalıştırılmıştır. Çalışmada ön plana çıkan bölümler kırmızı rengin ortaya çıkarılması için fark almaya dayalı bir yöntem ve Otsu Dinamik Eşik Değer Yöntemi'nin kullanılarak her resim için en iyi siyah beyaz dönüşüm eşik değerinin belirlenmesidir. Buradan ayrıca öğrenilmiştir ki birçok gerçek zamanlı uygulamada da Otsu Dinamik Eşik Değer belirleme yöntemi kullanılmaktadır. Ayrıca bu bölüm morfolojik sinyal işleme tekniklerinin uygulanması ve sonuçlarının gözlemlenmesi açısından da önemlidir. Her bir nesne resim içerisinde bir bağlı örüntü olarak düşünülürse, bu bağlı örüntülerin alan, eliptiklik, çevre, merkez noktası koordinatları vs. morfolojik özellikleri bir araya geldiğinde özel olarak bir nesneyi tanımlar. Dolayısı ile bu fikir çerçevesinde bir if-else yapısı kurulmuştur ve tatmin edici sonuçlar elde edilmiştir. Çalışmanın sadece kasis tabelası üzerinden yürütülmesi ilerde bütün tabelaların tespit edilebilmesi için bir fikir verecektir ve sistemin performansının belirlenmesi açısından önem arz etmektedir. Dolayısıyla, bu çalışmada gerçek zamanlı yapılacak bu sistemin bir ön çalışması yapılmıştır ve

tasarlanan algoritma veri setinde bulunan 90 tane resmin 80 tanesinde başarı ile kasis tabelalarını tespit edebilmiştir. Algoritmanın performansının belirlenebilmesi için bu 90 tane resim faklı gürültü koşulları altında çekilmiştir. Tespit edilemeyen 10 resimde ise genellikle kasis tabelasının önünde bir nesne bulunmaktadır veya tabela üzerine yapıştırılan bir reklam etiketi tanıma işlemine engel olmuştur. 90 resim üzerinde yapılan bu analizde %88.8'lik bir başarım elde edilmiştir. Çalışma 20 adet üçgen şeklini içeren ama içlerinde kasis olmayan levhalar üzerinde de denenmiştir ve algorita sınıflandırma yaparken hiç birisinde olmadığı halde kasis işaretini bulmamıştır. Bu da sistemin gürbüzlüğünün başka bir göstergesidir. Çünkü kasis işareti yok iken var demek beraberinde ciddi problemli durumlar oluşturmaktadır.

İkinci ana bölümde ise yine aynı kasis işareti belirleme problemi bu sefer yine tamamen özgün bir algoritma ile Matlab arayüz ortamında belirlenmiştir. İlk yöntemden tamamen farklı bir yöntem kullanılmıştır ve sonuçları irdelenmiştir. Bu bölümden ise vurgulanması gereken nokta herhangi bir rengi ön plana çıkartmak için kullanılabilecek tablolama tabanlı bir renk bölütleme algoritmasının geliştirilmiş olmasıdır. Ayrıca daha sonra bulunan kırmızı üçgen içerisinde kasis işareti piksel sayma yöntemi ile belirlenmiştir. Bu yöntemin özgün olmakla beraber bir takım artıları ve eksileri bulunmaktadır. Öncelikle oluşturulan Çizelge 1'deki değerler veri tabanındaki dört resim için oluşturulmuş ve bütün veritabanı için kullanılmıştır. Veritabanındaki bütün resimlerde başarı ile çalışması bütün resimlerin genelde bu ışık seviyelerinde, aynı telefon kamerası ile çekilmesindendir. Veritabanı farklılaştıkça bu eşik değeri tablosunun genişletilmesi gerekebilmektedir. Bu eksisinin yanında çok önemli artıları da bulunmaktadır. İlk önemli getirisi siyah beyaza dönüştürme işleminin gerekmemesidir. Çünkü işlemler alt uzaylarda yapıldığı için tek kanallıdır ve bu tek kanal üzerinde mantıksal atamalar yapıldığı için elde edilen resimler 1 ve 0'lardan oluşmaktadır. İkinci avantajı ise dinamik olarak bir eşik değeri bulma fonksiyonu kullanılmamıştır ve bu fonksiyonun getirdiği fazladan iş yükü ortadan kaldırılmıştır. Ayrıca resimlerin literatürdeki gibi HSC veya YCbCr gibi diğer uzaylara dönüştürülmesine gerek kalmadan işlemler zaten görüntünün alındığı RGB uzayında gerçekleştirilmiştir. Bu da işlem yükünü oldukça azaltmaktadır. Tasarlanan algoritma veri setinde bulunan 90 tane resmin yine 80 tanesinde başarı ile kasis tabelalarını tespit edebilmiştir. 90 resim üzerinde yapılan bu analizde %88.8'lik bir başarım elde edilmiştir. Bu bölümde yine 20 adet üçgen şeklini içeren ama içlerinde kasis olmayan

levhalar üzerinde de denenmiştir ve algorita sınıflandırma yaparken hiç birisinde olmadığı halde kasis işaretini bulmamıştır.

Tezin son bölümünde ise çalışmalar benzetim ortamında elde edilen bilgiler ışığında gerçek zamanlı çalışma ortamına alınmıştır. Gerçek zamanlı bilgisayarlı görü çalışanların sıklıkla tercih ettiği OpenCV açık kaynak kodlu kütüphanesi kullanılarak Microsoft Visual Studio 2012 ortamında kodlar düzenlenmiş ve derlenmiştir. Bu bölümde çeşitli zamansal başarım analizler yapılmış ve sonuçları bölüm içerisinde paylaşılmıştır. Özellikler alınan görüntülerin boyutlarının düşürülmesi ve gereksiz bölümlerinin atılması çok önemli sonuçlar içermektedir. Çünkü boyut küçültme ile iş yükü azalması arasındaki ilişkinin linear olmadığı tespit edilmiştir. Tasarlanan gerçek zamanlı sistem tezin ilk bölümünde tasarlanan benzetim ortamındaki sistem ile aynı istatistiki başarıya sahiptir. Araç içerisine yerleştirilen kamera ile yapılan gerçek zamanlı testlerde sonuç tatmin edici çıksada algılamanın doğru yapılması için tabelaya yakınlaşmış olmak zorunda kalınması sistemin uygulanabilirliğini düşürmektedir.

Sistem özet olarak kasis işareti içeren levhalarda başarı ile işareti her üç yöntemde de tespit edebilmiştir. Ayrıca sistem üçgen levhanın içerisinde kasis olmadığı durumlarda da kasis vardır diye uyarı vermemiştir. Bu da sistemin ayrı bir başarısıdır.

# KAYNAKLAR


[1] http://www.dailymail.co.uk/news/article-2285668/Car-crash-Blame-road-signs-Third-drivers-accident-near-miss-baffling-signs.html

[2] Ian Riches (2014-10-24). "Strategy Analytics: Automotive Ethernet: Market Growth Outlook | Keynote Speech 2014 IEEE SA: Ethernet & IP @ Automotive Technology Day" (PDF). IEEE. Retrieved 2014-11-23.

[3] Bengler, Klaus, et al. "Three decades of driver assistance systems: Review and future perspectives." IEEE Intelligent Transportation Systems Magazine 6.4 (2014): 6-22.

[4] Brkic, Karla. "An overview of traffic sign detection methods." Department of Electronics, Microelectronics, Computer and Intelligent Systems Faculty of Electrical Engineering and Computing Unska 3 (2010): 10000.

[5] Lopez, Luis David, and Olac Fuentes. "Color-based road sign detection and tracking." International Conference Image Analysis and Recognition. Springer Berlin Heidelberg, 2007.

[6] Mathias, Markus, et al. "Traffic sign recognition—How far are we from the solution?." Neural Networks (IJCNN), The 2013 International Joint Conference on. IEEE, 2013.

[7] Ciresan, Dan, et al. "Multi-column deep neural network for traffic sign classification." Neural Networks 32 (2012): 333-338.

[8] Sermanet, Pierre, and Yann LeCun. "Traffic sign recognition with multi-scale convolutional networks." Neural Networks (IJCNN), The 2011 International Joint Conference on. IEEE, 2011.

[9] Broggi, Alberto, et al. "Real time road signs recognition." 2007 IEEE Intelligent Vehicles Symposium. IEEE, 2007.

[10] Fleyeh, Hasan. "Color detection and segmentation for road and traffic signs." Cybernetics and Intelligent Systems, 2004 IEEE Conference on. Vol. 2. IEEE, 2004.

[11] Brkić, Karla, Axel Pinz, and Siniša Šegvić. "Traffic sign detection as a component of an automated traffic infrastructure inventory system." 33rd annual Workshop of the Austrian Association for Pattern Recognition (OAGM/AAPR). 2009.

[12] Scholkopft, Bernhard, and Klaus-Robert Mullert. "Fisher discriminant analysis with kernels." Neural networks for signal processing IX 1.1 (1999): 1.

[13] Öz, Sinan. "Fast 3D reconstruction from medical image series based on thresholding method." Biomedical Engineering Meeting (BIYOMUT), 2010 15th National. IEEE, 2010.

[14] Otsu, Nobuyuki. "A threshold selection method from gray-level histograms." Automatica 11.285-296 (1975): 23-27.

[15] Mertoğlu, Jale Nur, Demirer, Mehmet, Orhanlı, Tuna "Speed Bump Detection System by Autothresholding Method in Difficult Vision Conditions" 31st International Conference on Computers and Their Applications (CATA 2016), April 4–6, Las Vegas, Nevada, USA, 2016.

[16] Mertoğlu, Jale Nur, Orhanlı, Tuna "Speed bump detection system by autothresholding method." 2015 23nd Signal Processing and Communications Applications Conference (SIU). IEEE, 2015.

[17] Erişti, Ezgi. "Görüntü İşlemede Yeni Bir Soluk, OPENCV." Akademik Bilişim (2010).

[18] Lei, Zhang, Zhang Xue-fei, and Liu Yin-ping. "Research of the real-time detection of traffic flow based on OpenCV." Computer Science and Software Engineering, 2008 International Conference on. Vol. 2. IEEE, 2008.

[19] Gurav, Ruchi Manish, and Premanand K. Kadbe. "Real time finger tracking and contour detection for gesture recognition using OpenCV." Industrial Instrumentation and Control (ICIC), 2015 International Conference on. IEEE, 2015.

[20] Li, Da, Bodong Liang, and Weigang Zhang. "Real-time moving vehicle detection, tracking, and counting system implemented with OpenCV." 2014 4th IEEE International Conference on Information Science and Technology. IEEE, 2014.

[21] Dean, Huda Noor, and K. V. T. Jabir. "Real time detection and recognition of indian traffic signs using Matlab." Int J Sci Eng Res 4.5 (2013).

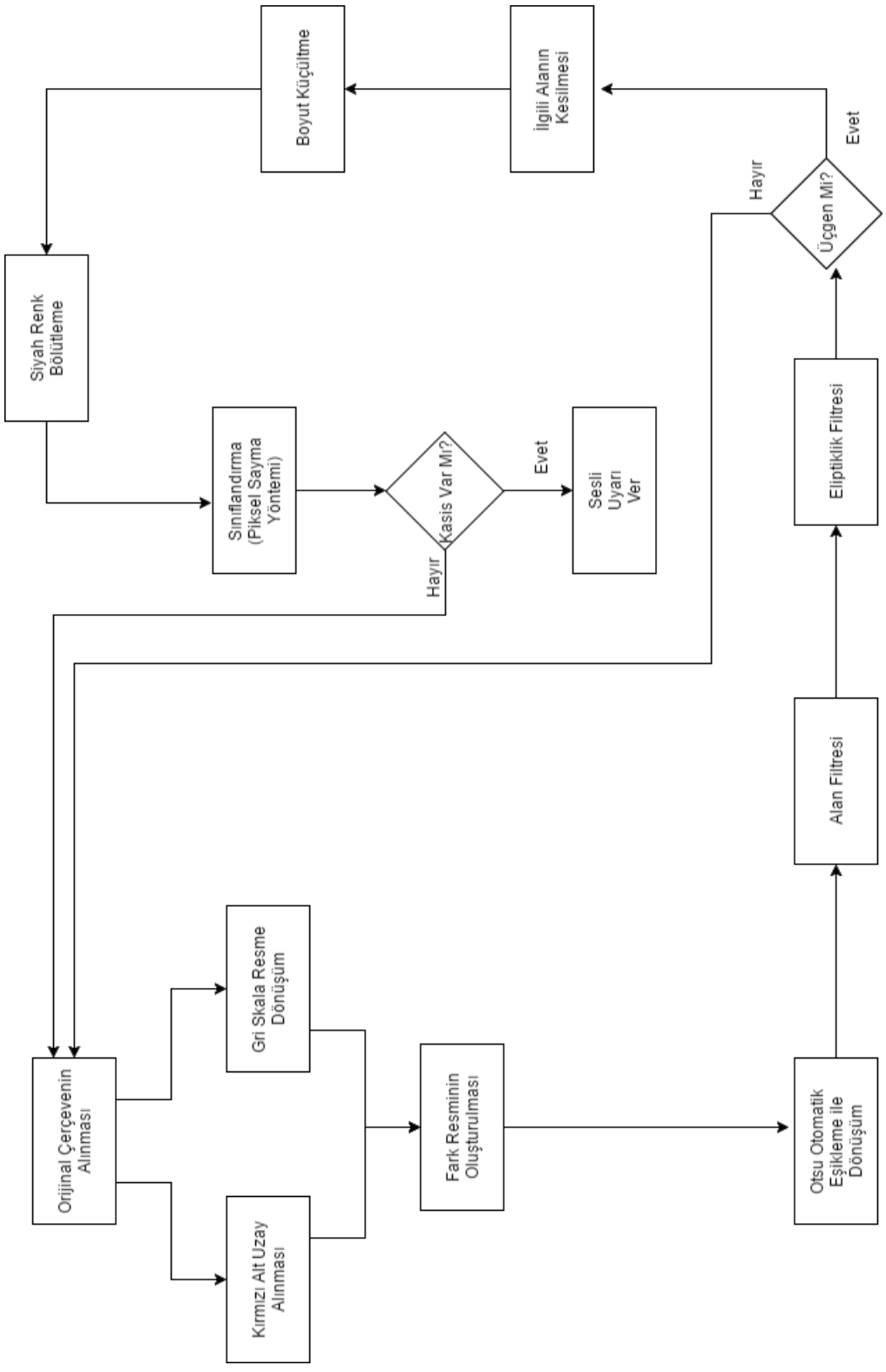
[22] Chhaya, Shambhavi Vijay, Sachin Khera, and S. Pradeep Kumar. "Basic Geometric Shape And Primary Colour Detection Using Image Processing on MATLAB." IJRET: International Journal of Research in Engineering and Technology 4: 2319-1163.

[23] Reinhard, Erik, et al. "Color transfer between images." IEEE Computer graphics and applications 21.5 (2001): 34-41.

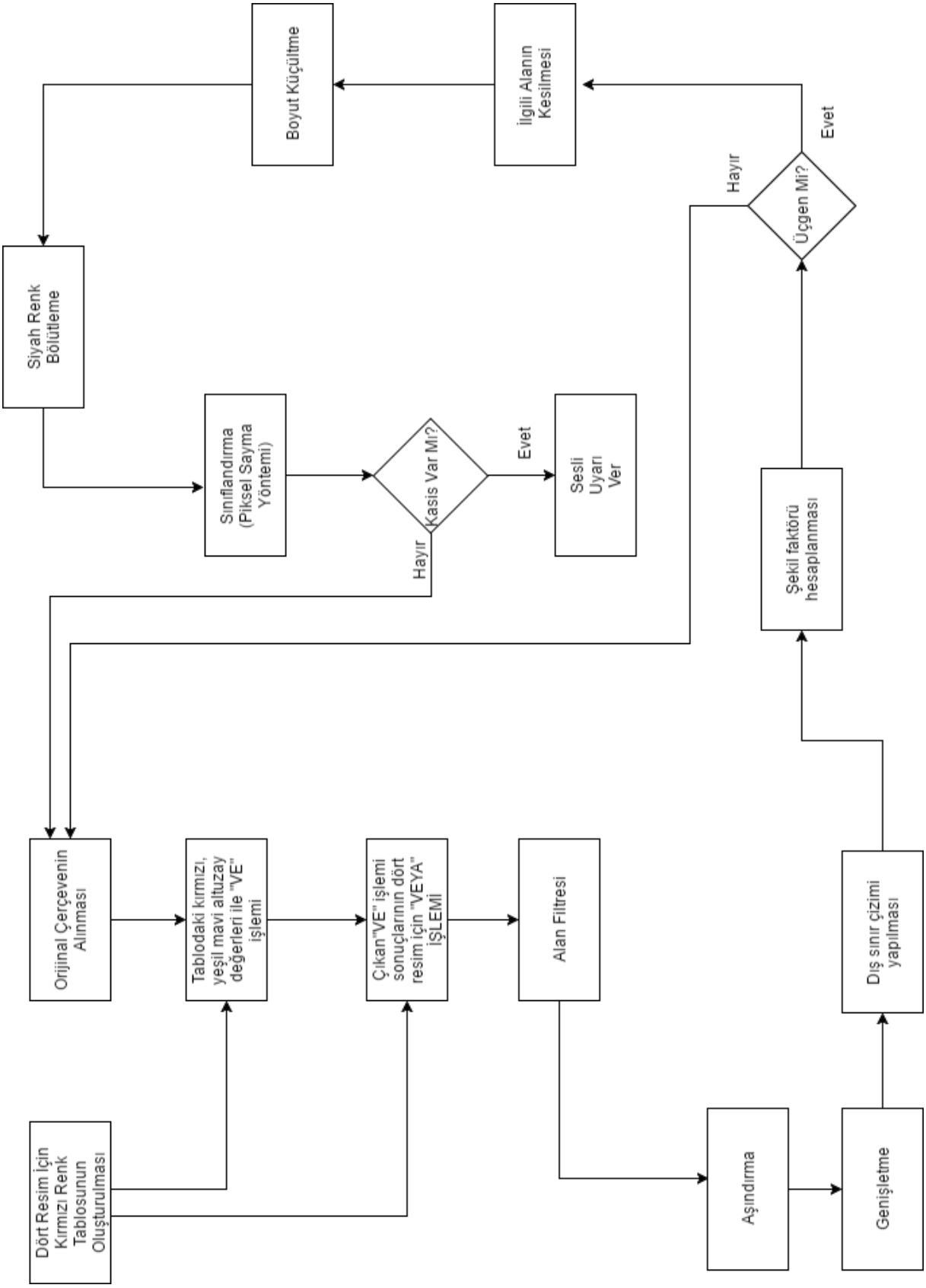
[24] http://ieee.bilkent.edu.tr/teknoloji101/?p=626

# EKLER

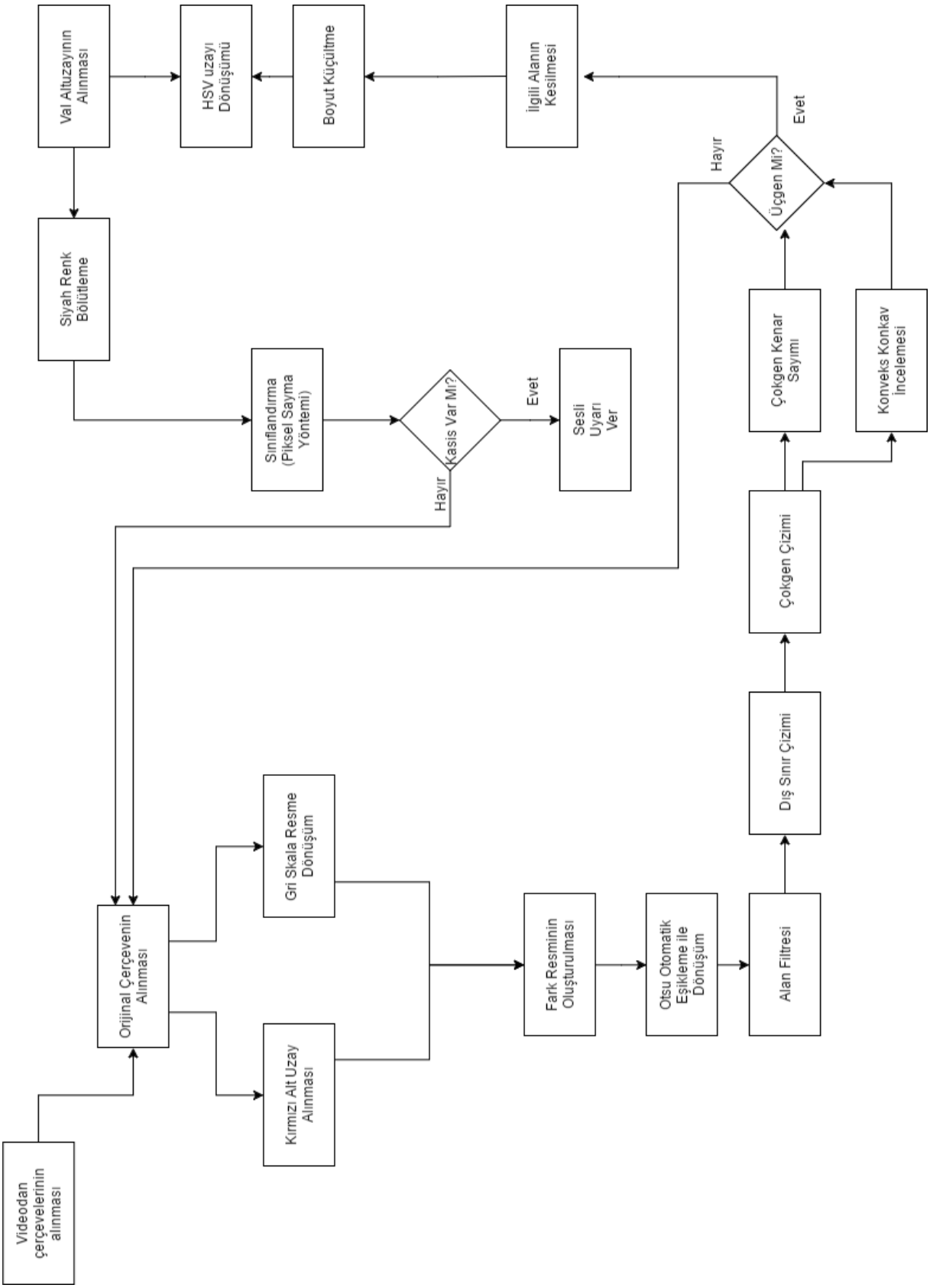
EK-1


Boyut Küçültme
İlgili Alanın Kesilmesi
Hayır
Üçgen Mi?
Evet
Siyah Renk Bölütleme
Sınıflandırma (Piksel Sayma Yöntemi)
Kasis Var Mı?
Evet
Sesli Uyarı Ver
Hayır
Eliptiklik Filtresi
Alan Filtresi
Gri Skala Resme Dönüşüm
Orijinal Çerçevenin Alınması
Fark Resminin Oluşturulması
Kırmızı Alt Uzay Alınması
Otsu Otomatik Eşikleme ile Dönüşüm

EK-2


Boyut Küçültme
İlgili Alanın Kesilmesi
Hayır
Üçgen Mi?
Evet
Siyah Renk Bölütleme
Sınıflandırma (Piksel Sayma Yöntemi)
Kasis Var Mı?
Evet
Sesli Uyarı Ver
Hayır
Şekil faktörü hesaplanması
Orijinal Çerçevenin Alınması
Tablodaki kırmızı, yeşil mavi altuzay değerleri ile "VE" işlemi
Çıkan"VE" işlemi sonuçlarının dört resim için "VEYA" İŞLEMİ
Alan Filtresi
Dış sınır çizimi yapılması
Dört Resim İçin Kırmızı Renk Tablosunun Oluşturulması
Aşındırma
Genişletme

EK-3


Videodan çerçevelerinin alınması
Orijinal Çerçevenin Alınması
Gri Skala Resme Dönüşüm
Kırmızı Alt Uzay Alınması
Fark Resminin Oluşturulması
Otsu Otomatik Eşikleme ile Dönüşüm
Alan Filtresi
Dış Sınır Çizimi
Çokgen Çizimi
Çokgen Kenar Sayımı
Konveks Konkav İncelemesi
Üçgen Mi?
Hayır
Evet
İlgili Alanın Kesilmesi
Boyut Küçültme
HSV uzayı Dönüşümü
Val Altuzayının Alınması
Siyah Renk Bölütleme
Sınıflandırma (Piksel Sayma Yöntemi)
Kasis Var Mı?
Evet
Sesli Uyarı Ver
Hayır

EK-4

Matlab’da görüntünün alınması:

videoFrame=imread ('IMG_1139.jpg');  komutu kullanırken,

c++ ile yazılmış Opencv kodu ile görüntünün alınması:

```
#include "stdafx.h"
#include <iostream>
#include <opencv2\opencv.hpp>
using namespace std;
using namespace cv;
int _tmain(int argc, _TCHAR* argv[])
{
	Mat img=imread("C:/Users/Jalenur/Documents/Visual Studio 2012/Projects/ConsoleApplication1/fotograf93.jpg");
	if (img.empty())
	{
		cout<<"Resim Yuklenemedi"<<endl;
		system("pause");
		return -1;
	}
	namedWindow("cerceve",CV_WINDOW_NORMAL);
	imshow("cerceve",img);
	waitKey(0);
	destroyWindow("cerceve");
	return 0;
}
```

EK-5

Opencv için başlık dosyalarının tanıtılması:

```
#include "stdafx.h"
#include <iostream>
#include "opencv2/imgproc/imgproc.hpp"
#include "opencv2/highgui/highgui.hpp"
#include <stdlib.h>
#include <stdio.h>
#include "opencv2/features2d/features2d.hpp"
```

EK-6

Görüntünün okunması ile ilgili kod parçası:

```
Mat src;
src=imread("C:/Users/Jale/Dropbox/Kodlar/ConsoleApplication30/x64/Debug/fotograf126.jpg");
namedWindow("Orjinal Resim",CV_WINDOW_NORMAL );
imshow("Orjinal Resim ", src);
```

EK-7

Alt uzaylara ayırma ile ilgili kod parçası:

```
Mat channel[2];
Mat jale_kirmizi( src.rows, src.cols, CV_8UC1 );
vector<Mat> layers;
split(src, layers);
namedWindow("Kirmizi",CV_WINDOW_NORMAL );
imshow("Kirmizi", layers[0]);
namedWindow("Yesil",CV_WINDOW_NORMAL );
```

```
imshow("Yesil", layers[1]);
namedWindow("Mavi",CV_WINDOW_NORMAL );
imshow("Mavi ", layers[2]);
```

EK-8

Fark resmi oluşturmak için yazılan kod parçası:

```
Mat jale_gri;
cvtColor(src,jale_gri,CV_BGR2GRAY);
namedWindow(“Gri”,CV_WINDOW_NORMAL );
imshow(“Gri “, jale_gri);
Mat fark_resmi;
fark_resmi=layers[2]-jale_gri;
namedWindow(“Fark Resmi”,CV_WINDOW_NORMAL );
imshow(“Fark Resmi”, fark_resmi);
```

EK-9

```
#include "stdafx.h"
#include "opencv2/imgproc/imgproc.hpp"
#include "opencv2/highgui/highgui.hpp"
#include <stdlib.h>
#include <stdio.h>
using namespace cv;
int threshold_value = 0;
int threshold_type = 3;
int const max_value = 255;
int const max_type = 4;
int const max_BINARY_value = 255;
Mat src, src_gray, dst;
```

```
const char* window_name = "Threshold Demo";

const char* trackbar_type = "Type: \n 0: Binary \n 1: Binary Inverted \n 2: Truncate \n
3: To Zero \n 4: To Zero Inverted";

const char* trackbar_value = "Value";

void Threshold_Demo( int, void* );

int main( int, char** argv )

{

  src=imread("C:/Users/Jalenur/Documents/Visual Studio
2012/Projects/ConsoleApplication23/x64/Debug/fotograf125.jpg");

  Mat channel[2];

  Mat jale_kirmizi( src.rows, src.cols, CV_8UC1 );

  vector<Mat> layers;

  split(src, layers);

  Mat jale_gri;

  cvtColor(src,jale_gri,CV_BGR2GRAY);

  Mat fark_resmi;

  fark_resmi=layers[2]-jale_gri;

  namedWindow("Fark Resmi",CV_WINDOW_NORMAL );

  imshow("Fark Resmi", fark_resmi);

  cvtColor( fark_resmi, src_gray, COLOR_RGB2GRAY );

  namedWindow( window_name, WINDOW_NORMAL );

  createTrackbar( trackbar_type,

                  window_name, &threshold_type,

                  max_type, Threshold_Demo );

  createTrackbar( trackbar_value,

                  window_name, &threshold_value,

                  max_value, Threshold_Demo );

  Threshold_Demo( 0, 0 );

  for(;;)

    {
```

```
      int c;
      c = waitKey( 20 );
      if( (char)c == 27 )
    { break; }
    }
}
void Threshold_Demo( int, void* )
{
  /* 0: Binary
     1: Binary Inverted
     2: Threshold Truncated
     3: Threshold to Zero
     4: Threshold to Zero Inverted
   */
  threshold( src_gray, dst, threshold_value, max_BINARY_value,threshold_type );
  imshow( window_name, dst );
}
```

EK-10

```
Mat thresh0;
threshold(fark_resmi,thresh0, 70, 255, CV_THRESH_BINARY);
namedWindow("Sabit Eşik Değeri",CV_WINDOW_NORMAL );
imshow("Otsu", thresh1);
Mat thresh1;
threshold(fark_resmi,thresh1, 0, 255, CV_THRESH_BINARY | CV_THRESH_OTSU);
namedWindow("Otsu Eşik Değeri",CV_WINDOW_NORMAL );
imshow("Otsu", thresh1);
```

```
vector<vector<Point> > contours;
 cv::findContours(thresh1.clone(), contours, CV_RETR_EXTERNAL, CV_CHAIN_APPROX_SIMPLE);
 //---------------------------------------------------------------------
 /// Draw contours
 vector<Vec4i> hierarchy;
 RNG rng(12345);
 Mat drawing = Mat::zeros( thresh1.clone().size(), CV_8UC3 );
  for( int i = 0; i< contours.size(); i++ )
  {
     Scalar color = Scal
ar( rng.uniform(0, 255), rng.uniform(0,255), rng.uniform(0,255) );
     drawContours( drawing, contours, i, color, 2, 8, hierarchy, 0, Point() );
  }
       namedWindow("Resulted Contours",CV_WINDOW_NORMAL);
       imshow( "Resulted Contours", drawing );
```

EK-12

```
vector<Point> approx;
  for (int i = 0; i < contours.size(); i++)
        {
    approxPolyDP(contours[i], approx, arcLength(contours[i], true) * 0.02, true);
    if (fabs(contourArea(contours[i])) < 5000 || !cv::isContourConvex(approx))
       continue;
        if (approx.size() == 3)
            {
                cout<<"Ucgen"<<endl;
        }
        else cout<<"Ucgen Yok"<<endl;
```

EK-13

```
vector<vector<Point>> contours;
cv::findContours(thresh1.clone(), contours, CV_RETR_EXTERNAL,
CV_CHAIN_APPROX_SIMPLE);
  cout<<endl<<contours.size()<<endl;
  vector<Vec4i> hierarchy;
  Mat drawing = Mat::zeros( thresh1.clone().size(), CV_8UC3 );
    for( int i = 0; i< contours.size(); i++ )
    {
       drawContours( drawing, contours, i, 150, 2, 8, hierarchy, 0, Point() );
    }
        namedWindow("Resulted Contours",CV_WINDOW_NORMAL);
        imshow( "Resulted Contours", drawing );
        RNG rng(12345);
  vector<vector<Point> > contours_poly( contours.size() );
```

```
  vector<Rect> boundRect( contours.size() );
  vector<Point2f>center( contours.size() );
  vector<float>radius( contours.size() );
  for( int i = 0; i < contours.size(); i++ )
     { approxPolyDP( Mat(contours[i]), contours_poly[i], 3, true );
       boundRect[i] = boundingRect( Mat(contours_poly[i]) );
       minEnclosingCircle( (Mat)contours_poly[i], center[i], radius[i] );
     }
  /// Draw polygonal contour + bonding rects + circles
  Mat drawing3 = Mat::zeros( thresh1.size(), CV_8UC3 );
  for( int i = 0; i< contours.size(); i++ )
     {
       Scalar color = Scalar( rng.uniform(0, 255), rng.uniform(0,255),
rng.uniform(0,255) );
       drawContours( drawing3, contours_poly, i, color, 1, 8, vector<Vec4i>(), 0, Point()
);
       rectangle( drawing3, boundRect[i].tl(), boundRect[i].br(), color, 2, 8, 0 );
     }
  namedWindow( "Approximated Bounding Box", CV_WINDOW_NORMAL);
  imshow( "Approximated Bounding Box", drawing3 );
```

# ÖZGEÇMİŞ

**Kimlik Bilgileri**


Adı Soyadı : Jale Nur Mertoğlu

Doğum Yeri : Malatya

Medeni Hali : Bekar

E-posta : jalenurmertoglu@gmail.com

Adresi : Şehit İlhan Tan Kışlası Ümitköy / Ankara


**Eğitim**

Lisans : 2006-2011 Erciyes Üniversitesi, Elektrik-Elektronik Mühendisliği

**Yabancı Dil ve Düzeyi**

İngilizce : İyi

**İş Deneyimi**

TSK , (2012-Devam Ediyor)

**Deneyim Alanları**

Görüntü İşleme, Bilgisayarlı Görü, Savunma Sistemleri, Radar Sistemleri

**Tezden Üretilmiş Projeler ve Bütçesi**

-

**Tezden Üretilmiş Yayınlar**

**-**

**Tezden Üretilmiş Tebliğ ve/veya Poster Sunumu İle Katıldığı Toplantılar**

1- Mertoğlu, Jale Nur, Tuna Orhanlı. "Speed bump detection system by autothresholding method." Signal Processing and Communications Applications Conference (SIU), 2015 23th. IEEE, Malatya, Türkiye, 2015.

2- Mertoğlu, Jale Nur, Demirer Mehmet, Tuna Orhanlı. " Speed Bump Detection System by Autothresholding Method in Difficult Vision Conditions" 31st International Conference on Computers and Their Applications (CATA 2016), Las Vegas, Nevada, USA, 2016

# CURRICULUM VITAE

## Credentials

Name, Surname : Jale Nur Mertoğlu

Place of Birth : Malatya

Martial Status : Single

E-mail : jalenurmertoglu@gmail.com

Address : Şehit İlhan Tan Kışlası Ümitköy / Ankara

## Eğitim

BSc. : 2006-2011 Erciyes University Department of Electrical Electronics Engineering

## Foreign Language

English : Advanced

## Work Experience

TSK , (2012-Continues)

## Areas of Experiences

Image Processign, Computer Vision, Defence Systems, Radar Systems

## Projects and Budgets

-

## Publications

-

**Oral and Poster Presentations**

1- Mertoğlu, Jale Nur, Tuna Orhanlı. "Speed bump detection system by autothresholding method." Signal Processing and Communications Applications Conference (SIU), 2015 23th. IEEE, Malatya, Turkey, 2015.

2- Mertoğlu, Jale Nur, Demirer Mehmet, Tuna Orhanlı. " Speed Bump Detection System by Autothresholding Method in Difficult Vision Conditions" 31st International Conference on Computers and Their Applications (CATA 2016), Las Vegas, Nevada, USA, 2016

# HACETTEPE ÜNİVERSİTESİ
# FEN BİLİMLERİ ENSTİTÜSÜ
# YÜKSEK LİSANS/DOKTORA TEZ ÇALIŞMASI ORJİNALLİK RAPORU

**HACETTEPE ÜNİVERSİTESİ**
**FEN BİLİMLER ENSTİTÜSÜ**
**ELEKTRİK VE ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI BAŞKANLIĞI'NA**

Tarih: 24/04/2017

Tez Başlığı / Konusu: BENZETİM ORTAMINDA VE GERÇEK ZAMANLI OLARAK TRAFİK İŞARETLERİNİN TESPİTİ

Yukarıda başlığı/konusu gösterilen tez çalışmamın a) Kapak sayfası, b) Giriş, c) Ana bölümler d) Sonuç kısımlarından oluşan toplam 74 sayfalık kısmına ilişkin, 23/04/2017 tarihinde şahsım/tez danışmanım tarafından *Turnitin* adlı intihal tespit programından aşağıda belirtilen filtrelemeler uygulanarak alınmış olan orijinallik raporuna göre, tezimin benzerlik oranı % 9 'dur.

Uygulanan filtrelemeler:
1- Kaynakça hariç
2- Alıntılar ~~hariç~~/dâhil
3- 5 kelimeden daha az örtüşme içeren metin kısımları hariç

Hacettepe Üniversitesi Fen Bilimleri Enstitüsü Tez Çalışması Orjinallik Raporu Alınması ve Kullanılması Uygulama Esasları'nı inceledim ve bu Uygulama Esasları'nda belirtilen azami benzerlik oranlarına göre tez çalışmamın herhangi bir intihal içermediğini; aksinin tespit edileceği muhtemel durumda doğabilecek her türlü hukuki sorumluluğu kabul ettiğimi ve yukarıda vermiş olduğum bilgilerin doğru olduğunu beyan ederim.

Gereğini saygılarımla arz ederim.

Tarih ve İmza
22/05/2017

**Adı Soyadı:** Jale Nur Mertoğlu
**Öğrenci No:** N12223979
**Anabilim Dalı:** Elektrik-Elektronik Mühendisliği
**Programı:** Elektrik-Elektronik Mühendisliği
**Statüsü:** ☒ Y.Lisans ☐ Doktora ☐ Bütünleşik Dr.

## DANIŞMAN ONAYI

UYGUNDUR.

Dr. Mehmet DEMİRER
(Unvan, Ad Soyad, İmza)