



**MARMARA ÜNİVERSİTESİ**  
**FEN BİLİMLERİ ENSTİTÜSÜ**



# **YAPAY GÖRÜ TEKNİKLERİ İLE ŞERİT ÇİZGİSİ ALGILAMA VE SINIFLANDIRMA**

HALE YENGİNER

**YÜKSEK LİSANS TEZİ**

**ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ**  
**(TÜRKÇE) ANABİLİM DALI**  
Elektrik-Elektronik Mühendisliği Programı

**DANIŞMAN**

Doç. Dr. Hayriye KORKMAZ

İSTANBUL, 2018



**MARMARA ÜNİVERSİTESİ**  
**FEN BİLİMLERİ ENSTİTÜSÜ**



# **YAPAY GÖRÜ TEKNİKLERİ İLE ŞERİT ÇİZGİSİ ALGILAMA VE SINIFLANDIRMA**

HALE YENGİNER

(523113015)

**YÜKSEK LİSANS TEZİ**

**ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ**  
**(TÜRKÇE) ANABİLİM DALI**

Elektrik-Elektronik Mühendisliği Programı

**DANIŞMAN**

Doç. Dr. Hayriye KORKMAZ

İSTANBUL, 2018

T.C.  
MARMARA ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ

Marmara Üniversitesi Fen Bilimleri Enstitüsü Yüksek Lisans Öğrencisi Hale YENGİNER' in "Yapay Görü Teknikleri İle Şerit Çizgisi Algılama Ve Sınıflandırma" başlıklı tez çalışması 10/01/2018 tarihinde savunulmuş ve jüri üyeleri tarafından başarılı bulunmuştur.

**Jüri Üyeleri**

Doç. Dr. Hayriye KORKMAZ (Danışman)

Marmara Üniversitesi.....(İmza)

Prof. Dr. Mehmet Melih İNAL (Üye)

Kocaeli Üniversitesi .....(İmza)

Yrd. Doç. Dr. Önder DEMİR (Üye)

Marmara Üniversitesi .....(İmza)

**ONAY**

Marmara Üniversitesi Fen Bilimleri Enstitüsü Yönetim Kurulu'nun 22/01/2018 tarih ve 2018/02-02 sayılı kararı ile Hale YENGİNER' in Elektrik-Elektronik Mühendisliği (Türkçe) Anabilim Dalı Elektrik-Elektronik Mühendisliği Programında Yüksek Lisans derecesi alması onanmıştır.

**Fen Bilimleri Enstitüsü Müdürü**

**Prof. Dr. Bülent EKİCİ**



## **TEŞEKKÜR**

Tez çalışmalarım süresince bilgisini, deneyimlerini benden esirgemeyen ve yardımcı olan değerli danışmanım Doç. Dr. Sn. Hayriye KORKMAZ' a,

Bilgi ve tecrübeleriyle çalışmalarımda yol gösteren Yrd. Doç. Dr. Sn. Gökhan ERDEMİR' e,

Çalışmalarım esnasında beni destekleyen ve sıkıntılarımı paylaşan değerli ablam Jale YENGİNER AKTAŞ' a, aileme ve arkadaşlarıma,

Teşekkürlerimi sunarım.

**Ocak, 2018**

**Hale YENGİNER**

# İÇİNDEKİLER

TEŞEKKÜR.....	i
İÇİNDEKİLER .....	ii
ÖZET .....	iv
ABSTRACT .....	v
SEMBOLLER.....	vi
KISALTMALAR.....	vii
ŞEKİL LİSTESİ.....	ix
TABLO LİSTESİ .....	xii
1. GİRİŞ.....	1
1.1. Amaç ve Gerekçeler .....	2
1.2. Benzer Çalışmalar .....	3
2. SANAL ENSTRÜMANTASYON TEMELLİ YAPAY GÖRÜ SİSTEMİ TASARIMI.....	10
2.1. LabVIEW Grafiksel Geliştirme Platformu .....	12
2.1.1. Ön panel (Front panel) .....	12
2.1.2. Blok diyagram (Block diagram).....	13
2.1.3. “Görme” kontrol kaleti.....	15
2.1.4. “Görme ve Hareket” fonksiyon paleti .....	16
2.1.4.1. “Görme Yardımcı” nesneleri .....	16
2.1.4.2. Görüntü işleme .....	18
2.2. Sistem Tasarımı.....	21
2.2.1. Şerit çizgisinin tespit edilmesi ve sınıflandırılmasına yönelik geliştirilen ön panel .....	21
2.2.2. Şerit çizgisinin tespit edilmesi ve sınıflandırılmasına yönelik geliştirilen blok diyagram .....	23
2.2.2.1. RGB-Gri skala dönüşümü .....	25

2.2.2.2.	Bölütleme.....	26
2.2.2.3.	Kenar tespiti.....	27
2.2.2.4.	Morfolojik işlemler.....	29
2.2.2.5.	Şerit çizgisi algılama algoritmaları .....	33
2.2.2.6.	Sınıflandırma .....	41
3.	DENEYSEL SONUÇLAR.....	43
4.	DEĞERLENDİRME VE ÖNERİLER.....	48
	KAYNAKLAR.....	54
	EK A: Sistem Blok Diyagramı - Sol Bölüm.....	58
	EK B: Sistem Blok Diyagramı - Sağ Bölüm .....	59
	EK C: Sistem Ön Paneli.....	60
	ÖZGEÇMİŞ .....	61

## ÖZET

# YAPAY GÖRÜ TEKNİKLERİ İLE ŞERİT ÇİZGİSİ ALGILAMA VE SINIFLANDIRMA

İleri Sürüş Destek Sistemleri (İSDS), trafikte hareket halindeki araçlarda, çarpışma veya şeritten ayrılma, kırmızı ışık ihlali gibi oluşabilecek tehlikeli durumların önceden tespiti ve önlenmesine yönelik geliştirilmiş, günümüzde üst sınıf araçlarda hali hazırda kullanılan sistemlerdir. Geliştirilen uygulamalar, güvenlik, ekonomik ve konfor olmak üzere 3 temel grupta yoğunlaşmaktadır. Bu tez çalışmasında, güvenlik destek sistemleri altında yer alan şerit çizgilerinin algılanması ve sınıflandırılması uygulaması ele alınmış ve gerçekleştirilmiştir.

Sistem, yapay görü teknikleri ile LabVIEW platformunda “Vision and Motion” fonksiyon paleti nesneleri kullanılarak tasarlanmıştır. Sentetik ve gerçek yol görüntülerinin bulunduğu veri setleri üzerinde çalışılmıştır. Farklı görüş açılarından elde edilmiş gerçek yol görüntüleri, aynı zamanda gölge, yansıma, silinmiş yol çizgisi, kötü hava ve farklı gün ışığı gibi olumsuz koşulları da barındırmaktadır.

Sistem, ön işlemler, bölütleme, çizgi algılama ve sınıflandırma olmak üzere dört temel adımdan oluşmaktadır. Ön işlemler adımı, Canny kenar belirleme algoritması ile genişletme (*dilate*) ve aşındırma (*erode*) operatörlerinin kullanıldığı matematiksel morfolojik işlemlerden faydalanılmıştır. Şerit çizgisinin algılanmasında Hough dönüşümü kullanılmıştır. Çizgi türünün sınıflandırılmasında ise, if – then – else yapısından oluşan basit bir kural tabanı ile düz ve kesik ayrımı yapılmıştır.

Çalışmalar süresince, 15 adet sentetik görüntünün yer aldığı Clipartfest veri seti ile 168 adet gerçek yol görüntüsünün yer aldığı Carnegie Mellon Üniversitesi Robotik Enstitüsü Görme ve Otonom Sistemler Merkezi’nden ve tarafımızdan farklı mobil cihaz kameraları kullanılarak elde edilen veri setlerinden faydalanılmıştır. Deneyler bu veri setlerinden rastgele seçilen 30 görüntü üzerinde gerçekleştirilmiştir. Gerçekleştirilen deneylerde şerit çizgisi algılama ve sınıflandırma başarısı % 77 olarak tespit edilmiştir.

## **ABSTRACT**

### **LANE DETECTION AND CLASSIFICATION BY USING MACHINE VISION TECHNIQUES**

The Advanced Driver Assistance Systems (ADAS) are currently used in upper class vehicles in traffic for collision, lane departure and red light warning, etc. Developed applications are focused on three main topics: safety, economic and comfort. In this thesis, the detection and classification of the lane lines under the safety sub-topic has been dealt and implemented.

The system has been developed in LabVIEW platform by using "Vision and Motion" function palette objects based on machine vision techniques. Dataset with synthetic and real road images were studied. Real road images were obtained from different points of view and also include adverse conditions such as shadow, reflection, bad weather, different daylights and deleted lane lines.

The system consists of four basic steps: preprocessing, segmentation, lane line detection and classification. In the preprocessing step, the Canny edge detection algorithm and mathematical morphological operations including dilate and erode operators was used respectively. The Hough Transform was used to detect the lane line locations in the image. In classification step, a simple rule base consisting if - then - else structure was used to distinguish between straight and dashed lane lines.

During the work, different datasets were used. These are obtained from Clipartfest, different mobile phone camera and Carnegie Mellon University Robotic Institute Vision and Autonomous Systems Center's. There are 15 synthetic images on Clipartfest and 168 real road images on other datasets. Experiments were performed on 30 images selected randomly from these datasets. In the experiments, the lane line detection and classification performance was determined as 77%.



## SEMBOLLER

$E_x[i,j]$	: Gradyanın X-yönündeki Kısmi Türevi
$E_y[i,j]$	: Gradyanın Y-yönündeki Kısmi Türevi
$f$	: Sayısal Görüntünün Işık Şiddeti
$f(x,y)$	: İşlem Görmemiş Görüntü
<b>GB</b>	: Giga Bayt
<b>GHz</b>	: Giga Hertz
$G(x)$	: X-yönündeki Gradyan Operatörü
$G(y)$	: Y-yönündeki Gradyan Operatörü
$G(x,y)$	: İki Boyutlu Gauss Filtresi
$I(i,j)$	: İşlem Görmüş Görüntü
<b>MHz</b>	: Mega Hertz
<b>MB</b>	: Mega Bayt
$M(i,j)$	: Gradyanın Büyüklüğü
$\oplus$	: Genişletme İşlemi
$\ominus$	: Aşındırma İşlemi
$\sigma$	: Standart Sapma
$r$	: Doğrusal Çizgi Mesafesi
$\theta$	: Doğrusal Çizgi Eğim Açısı
$\theta(i,j)$	: Görüntü Gradyanın Azimutu

## **KISALTMALAR**

<b>ADAS</b>	: Advanced Driver Assistance Systems
<b>ACO</b>	: Ant Colony Optimization
<b>ARHT</b>	: Adaptive Random Hough Transformation
<b>CC-TV</b>	: Closed Circuit Television
<b>DV</b>	: Digital Video
<b>DVM</b>	: Destek Vektör Makineleri
<b>FFT</b>	: Fast Fourier Transform
<b>FPGA</b>	: Field Programmable Gate Array
<b>GPS</b>	: Global Positioning System
<b>HSL</b>	: Hue Saturation Luminance
<b>IPM</b>	: Inverse Perspective Mapping
<b>İSDS</b>	: İleri Sürüş Destek Sistemleri
<b>LabVIEW</b>	: Laboratory Virtual Instrument Engineering Workbench
<b>LIDAR</b>	: Light Detection and Ranging
<b>MATLAB</b>	: Matrix Laboratory
<b>MP</b>	: Mega Piksel
<b>NMS</b>	: Non- maximum Suppression
<b>PC</b>	: Personal Computer
<b>RAM</b>	: Read Access Memory
<b>RANSAC</b>	: Random Sample Consensus Algorithm
<b>ROI</b>	: Region of Interest

<b>RGB</b>	: Red Green Blue
<b>VI</b>	: Virtual Instrumentation
<b>VMD</b>	: Vertical Mean Distribution Method
<b>YSA</b>	: Yapay Sinir Ağları
<b>2B</b>	: 2 Boyutlu
<b>3B</b>	: 3 Boyutlu



## ŞEKİL LİSTESİ

<b>Şekil 2.1</b> Deneylerde kullanılan veri setlerine ait örnek yol görüntüleri <b>a)</b> Mobil cihaz 1 kamera <b>b)</b> Mobil cihaz 2 kamera <b>c)</b> Carnegie Mellon Üniversitesi <b>d)</b> Sentetik.....	11
<b>Şekil 2.2</b> “Görme” nesne kutusunu da içeren LabVIEW platformu ön panel görüntüsü	13
<b>Şekil 2.3</b> “Görme ve Hareket” fonksiyon paletini de içeren LabVIEW platformu blok diyagram görüntüsü .....	14
<b>Şekil 2.4</b> “Görme” kontrol paleti.....	15
<b>Şekil 2.5</b> “Görme ve Hareket” fonksiyon paleti.....	16
<b>Şekil 2.6</b> “Görme Yardımcı” fonksiyon paleti .....	16
<b>Şekil 2.7</b> “Görüntü İşleme” fonksiyon paleti .....	19
<b>Şekil 2.8</b> Şerit çizgisinin tespit edilmesi ve sınıflandırılmasına yönelik geliştirilen örnek ön panel görüntüsü <b>a)</b> Sol şerit için işlem sonucu <b>b)</b> Canny filtesi çıkış görüntüleri <b>c)</b> Tespit edilen çizginin grafiksel gösterimi ve ROI, Hough dönüşümü ile Canny filtresi işlemlerine yönelik parametrelerin belirlenmesi <b>d)</b> ve <b>e)</b> Morfolojik işlemler çıkış görüntüleri .....	22
<b>Şekil 2.9</b> Sistem akış diyagramı .....	24
<b>Şekil 2.10</b> RGB- Gri skala dönüşümü öncesi ve sonrası gerçek yol görüntüsü.....	26
<b>Şekil 2.11</b> Bölütlenmiş yol görüntüsü ve bu aşamaya ait blok diyagram .....	27
<b>Şekil 2.12</b> Kenar belirleme işlemi öncesi ve sonrası gerçek yol görüntüsü ve bu aşamaya ait blok diyagram .....	29
<b>Şekil 2.13</b> Genişletme işleminin görüntü üzerindeki etkisi.....	30
<b>Şekil 2.14</b> Aşındırma işleminin görüntü üzerindeki etkisi .....	30
<b>Şekil 2.15</b> Silinmiş şerit çizgisi iyileştirme aşamaları <b>a)</b> Giriş görüntüsü <b>b)</b> Canny filtresi uygulanmış görüntü <b>c)</b> Genişletme işlemi etkisi <b>d)</b> Aşındırma işlemi etkisi .....	31

<b>Şekil 2.16</b> Morfolojik işlemler <b>a)</b> Genişletme ve aşındırma operatörleri uygulanmış örnek görüntüler <b>b)</b> Blok diyagram <b>c)</b> Genişletme ve aşındırma operatörlerinin bu sistemdeki çıkış görüntüleri.....	32
<b>Şekil 2.17</b> Hough Dönüşümü <b>a)</b> Elde edilen koordinat noktaları <b>b)</b> ve <b>c)</b> Muhtemel doğrusal bağlantı parçaları.....	33
<b>Şekil 2.18</b> Düz bir eğrinin parametrik tanımlanması .....	34
<b>Şekil 2.19</b> Hough yüzeyi .....	34
<b>Şekil 2.20</b> Şerit çizgisi tespiti blok diyagramı.....	35
<b>Şekil 2.21</b> Alan-tabanlı yaklaşım ile ROI seçimi [33] .....	36
<b>Şekil 2.22</b> ROI bölgesi sınırlarının belirlenmesi <b>a)</b> Sağ bölüm <b>b)</b> Sol bölüm .....	37
<b>Şekil 2.23</b> Aracın görüş alanı [33] .....	37
<b>Şekil 2.24</b> Şerit çizgisi konumu tespitinin iyileştirilmesine ait akış şeması.....	38
<b>Şekil 2.25</b> Algılanan şerit çizgisinin doğru konumunun elde edilmesi için eklenen iyileştirmelere yönelik blok diyagram .....	39
<b>Şekil 2.26</b> Işık şiddeti türev grafikleri <b>a)</b> Kesik şerit çizgisi <b>b)</b> Düz şerit çizgisi .....	41
<b>Şekil 2.27</b> Şerit çizgisi sınıflandırılması çıkış görüntüleri <b>a)</b> Sol şerit <b>b)</b> Sağ şerit .....	42
<b>Şekil 2.28</b> Şerit çizgisi sınıflandırılması blok diyagramı .....	42
<b>Şekil 3.1</b> Olumsuz koşullara sahip yol görüntüsü örnekleri.....	43
<b>Şekil 3.2</b> Clipartfest veri seti ile bulunan deney sonuçları <b>a)</b> Sentetik yol görüntüsü <b>b)</b> Sol şeridin algılanmasına ait sonuç görüntüsü <b>c)</b> Sağ şeridin algılanmasına ait sonuç görüntüsü .....	44
<b>Şekil 3.3</b> Carnegie Mellon Üniversitesi veri seti ile bulunan deney sonuçları <b>a)</b> Gerçek yol görüntüsü <b>b)</b> Sol şeridin algılanmasına ait sonuç görüntüsü <b>c)</b> Sağ şeridin algılanmasına ait sonuç görüntüsü.....	45

<b>Şekil 3.4</b> Mobil cihaz 1 veri seti ile bulunan deney sonuçları <b>a)</b> Gerçek yol görüntüsü <b>b)</b> Sol şeridin algılanmasına ait sonuç görüntüsü <b>c)</b> Sağ şeridin algılanmasına ait sonuç görüntüsü .....	46
<b>Şekil 3.5</b> Mobil cihaz 2 veri seti ile bulunan deney sonuçları <b>a)</b> Gerçek yol görüntüsü <b>b)</b> Sol şeridin algılanmasına ait sonuç görüntüsü <b>c)</b> Sağ şeridin algılanmasına ait sonuç görüntüsü .....	47
<b>Şekil 4.1</b> Carnegie Mellon Üniversitesi veri seti ile yanlış bulunan deney sonuçları <b>a)</b> Gerçek yol görüntüsü <b>b)</b> Sol şeridin algılanmasına ait sonuç görüntüsü <b>c)</b> Sağ şeridin algılanmasına ait sonuç görüntüsü.....	50
<b>Şekil 4.2</b> Mobil cihaz 2 veri seti ile yanlış bulunan deney sonuçları <b>a)</b> Gerçek yol görüntüsü <b>b)</b> Sol şeridin algılanmasına ait sonuç görüntüsü <b>c)</b> Sağ şeridin algılanmasına ait sonuç görüntüsü.....	51
<b>Şekil 4.3</b> Mobil cihaz 2 veri seti ile yanlış tespit edilen duruma ait detaylı görüntü .....	53

## TABLO LİSTESİ

<b>Tablo 1.1</b> Benzer Çalışmalar ve Karşılaştırma Tablosu .....	6
<b>Tablo 2.1</b> Dönüştürülebilecek Görüntü Türleri .....	25
<b>Tablo 2.2</b> Morfolojik İşlemlerde Kullanılabilecek Operatör Türleri ve İşlevleri .....	31
<b>Tablo 4.1</b> Şerit çizgisi konumu algılama başarı sonuçları .....	48
<b>Tablo 4.2</b> Şerit çizgisi sınıflandırma başarı sonuçları.....	49



## 1. GİRİŞ

İleri Sürüş Destek Sistemleri (İSDS, *Advanced Driver Assistance Systems-ADAS*), günümüzde üst sınıf araçlarda hali hazırda kullanılan ve trafikte hareket halindeki araçlar için oluşabilecek tehlikeli durumların önceden tespitine ve önlenmesine yönelik geliştirilmiş sistemlerdir. Bu sistemler, bir veya daha çok kamera ve çeşitli algılayıcılar (radar, ultrasonik) yardımı ile aracın bulunduğu konumdaki çevre koşulları (yaya, araç, yol, trafik işaretleri gibi) hakkında analiz imkânı sağlayan ve çözümler sunan yapay görü teknolojilerinin kullanıldığı sistemlerdir [1]. Örneğin; aracın ön camına yerleştirilen bir kameradan elde edilen görüntüler, şeritten ayrılma uyarısı, nesne algılama, mesafe ölçümü veya trafik işareti renk algılama gibi görevleri yerine getirmektedir. Sistemin çıktısı sürücüye yönelik bir uyarı olabileceği gibi; direksiyon hareketi veya frenleme gibi belirli bir fonksiyonu doğrudan kontrol edecek şekilde de olabilir.

İSDS çözümleri, genel olarak güvenlik, konfor ve ekonomik olmak üzere 3 alt grupta incelenmektedir [2-3]. Konfor destek sistemleri, sürücünün araç ile ergonomik etkileşimi ve kolay sürüş temeline dayanmaktadır. Bu sistemler sürücünün yorgunluğunu azaltmayı hedeflemektedir. Ekonomi alt grubunda bulunan destek sistemleri yakıt tüketimini düşürmeyi, verimliliği artırmayı ve güç tüketimini azaltmayı amaçlamaktadır. Güvenlik alt grubunda yer alan çözümler ise; sürücü, yolcu ve yayalar için daha güvenli bir sürüş ortamı sağlayan emniyet artırıcı uygulamaları içermektedir. Bu alt grupta yer alan sistemler, kavşak asistanı (*intersection assist*), ileri yaya çarpışma uyarı sistemi (*forward pedestrian collision warning*), gece görüş sistemi (*night vision*), çarpışma önleyici (*collision prevention assist*), ışık ve görüş sistemleri (*light and vision systems*), şerit algılama (*lane detection*) ve şerit takip sistemleri (*lane assist*), şeritten ayrılma uyarı sistemleri (*lane departure warning systems*), şerit değiştirme yardımcı sistemleri (*lane departure assist*) ve geri görüş kamerası (*rear assist*) olarak sıralanabilir [3-4]. Kaza oranlarını azaltmayı hedefleyen bu uygulamaların bir kısmının temelinde, bu tez çalışmasının da motivasyonunu oluşturan şerit çizgisi algılama ve şerit çizgisi türünün sınıflandırılması temaları işlenmektedir.



Bu çalışmada, Sanal Enstrümantasyon (*Virtual Instrumentation - VI*) temelli görüntü işleme teknikleri kullanılarak, gölgeli veya silinmiş-eskimiş şerit çizgisine sahip farklı kameralardan elde edilmiş yol görüntüleri için “Şerit Çizgisi Konum Tespiti ve Sınıflandırılmasına” yönelik bir yapay görü sistemi geliştirilmiştir.

Deneyle, farklı çevre koşullarına (farklı gün ışığında aydınlık seviyesindeki değişimler, gölge durumları, silinmiş-eskimiş şerit çizgisi vb.) ve farklı özelliklere sahip (boyut ve çözünürlük farklılıkları) gerçek ve sentetik görüntülerin bulunduğu iki ayrı veri seti ile tamamlanmıştır.

Her iki veri seti için de şerit çizgisi tespiti ve sınıflandırılması adımları tamamlanmıştır. Sistem tasarımı görüntünün gri skalada okunması, bölütlenmesi, sınır eğrilerinin çıkartılması, morfolojik işlemlerin uygulanması, şerit çizgilerinin tespit edilmesi ve sınıflandırılması olmak üzere 6 temel adımdan oluşmaktadır. Bu adımların tamamı, kullanılan platformda bulunan “Görme ve Hareket” fonksiyon paleti nesneleri ile gerçekleştirilmiştir. Şerit çizgi türünün sınıflandırılması için ise basit kural tabanlı bir algoritma oluşturulmuştur.

Bu tez çalışması, giriş, sanal enstrümantasyon temelli yapay görü sistemi tasarımında kullanılan materyal ve yöntemler, deneysel sonuçlar ile değerlendirme ve öneriler olmak üzere 4 bölümden oluşmaktadır. Giriş bölümünde tezin amacı ve literatürde yer alan benzer çalışmalar hakkında bilgi verilmiştir. İkinci bölümde LabVIEW (*Laboratory Virtual Instrument Engineering Workbench*) platformu ve yapay görme sistemi tasarımından bahsedilerek; yapılan çalışmanın detayları anlatılmıştır. Üçüncü bölümde, deneylerde elde edilen sonuçlar değerlendirilmiş ve son bölümde ise sistemin geliştirilmesi için öneriler sunulmuştur.

### **1.1. Amaç ve Gerekçeler**

Dünyada ölümle sonuçlanan trafik kazaları listesinde Türkiye üst sıralarda yer almaktadır. Kazaya sebebiyet verme açısından sürücü, yaya ve yolcu unsurları değerlendirildiğinde, sürücü kusurları %88,6 ile en başta gelmektedir [5]. Sürücüye ilişkin başlıca hata etmenleri; hız, alkol ve madde kullanımı, uyku ve yorgunluk ile emniyet kemeri, çocuk oto koltuğu ve kask takmama/kullanmama vb. olarak sıralanmaktadır. Türkiye’de ilgili mevzuat geliştirilmiş olmasına rağmen gerektiği gibi

uygulanmamaktadır [6]. Bu durum, ileri sürüş destek sistemlerine olan ihtiyacı vurgulamaktadır.

Günümüzde üst sınıf markaların çoğu aracında bu sistemler hali hazırda kullanılmaktadır. Ancak bu sistemlerin tüm araçlarda kullanımının yaygınlaşması için; geliştirilen teknolojilerin düşük maliyetli olması önem arz etmektedir.

Otonom ve akıllı araçlarda aracın ön panelinde bulunan iç dikiz aynasına monte edilen kamera ile alınan görüntüler güvenlik destek sistemlerinin vazgeçilmez unsurudur. Kamera yardımıyla gerçek zamanlı olarak alınan görüntülerden, şerit genişliği, şerit çizgisi takibi (yolun düz veya kıvrımlı olması durumuna göre), şerit çizgisi rengi, aracın şerit üzerindeki konumu ve şerit çizgisi sınıflandırma (düz veya kesik) gibi analizler yapılabilmektedir [7-8].

Bu çalışmada, İSDS' nin güvenlik alt grubunda yer alan; istem dışı şeritten ayrılma durumunda uyarı verme, şerit algılama ve şerit değiştirme yardımcısı gibi işlemlerin temelini oluşturan şerit çizgisi tespiti ve sınıflandırılması konusu ele alınmıştır. Böylelikle kaza oranlarının azaltılmasına yönelik önemli katkı sağlanması amaçlanmıştır. Ayrıca yakın gelecekte yaygınlaşması olası olan otonom araçlar için bu özellik, klasik araçlara göre bir kat daha önemli ve vazgeçilmezdir.

Şerit çizgisinin tespitinde ve sınıflandırılmasında, arayüz olarak sunulan ön panelin metin tabanlı programlama dillerine göre daha fazla görsellik barındırması, görüntü işlemede ileri düzey analizler yapılabilmesi (çeşitli gürültü giderme filtreleri ve kenar bulma algoritmaları) ve yapay görü teknikleri hakkında (*Select Region Of Interest, Locate Edges, Find Patterns, Analytic, Optical Character Recognition etc.*) geniş kapsamlı bir fonksiyon kütüphanesi sunması, diğer programlama dilleri ile uyumlu olması, veri analizine imkân vermesi ve sanal enstrümanlar ile tasarımı kolaylaştırması gibi unsurlar göz önünde bulundurularak LabVIEW platformu tercih edilmiştir [9].

## **1.2. Benzer Çalışmalar**

Literatürde şerit çizgisi algılama algoritmaları, model tabanlı (*model-based*) ve özellik tabanlı (*feature-based*) olmak üzere 2 temel başlık altında incelenmektedir [10]. Model tabanlı algoritmaların bozucu etkilere (sisli, yağmurlu, bulutlu vb. kötü hava koşulları,

günün farklı saatlerinde ışık şiddetinin farklılığı, silinmiş veya eskimiş yol çizgileri, yol üzerinde bulunan engeller) uyum sağlama başarısı özellik tabanlı algoritmalara göre daha düşüktür. Özellik tabanlı gerçekleştirilen algoritmalarda ise, silinmiş şerit çizgileri, güneşin doğuşu ve batışında olduğu gibi aydınlık seviyesindeki değişimler, yol kenarlarındaki ağaçların yol üzerine düşen gölgesi vb. problemlere çözümler üretilebilmektedir. Otonom araçlar için yeterli verimliliktedir şeklinde bir değerlendirme yapılmıştır. Ancak % cinsinden bir başarı değerlendirilmesi yapılmamıştır.

Qingquan Li ve arkadaşları [11] hazırladıkları çalışmada, 8 adet lazer tarayıcı ve 7 adet kamera kullanarak, trafikte güvenli sürüşün hedeflendiği otonom araç navigasyon sistemi tasarımı yapmışlardır. Birden fazla algılayıcı ve LIDAR (Light Detection and Ranging) kullanarak, çalışmalarını şerit çizgisine sahip olan veya olmayan (yapılandırılmış ve yapılandırılmamış) yollar üzerinde gerçek zamanlı olarak uygulama imkânı elde etmişlerdir. Lazer tarayıcılardan 3 adedi ve kameralardan 2 adedi yol genişliğini ve engeli algılamada kullanılmaktadır. Yapılan bu çalışmanın yol algılanması kısmında, top-hat dönüşümü, Prewitt dikey gradyan, uyarlanmış eşik değeri (*adaptive threshold*) ve aşamalı olasılıksal Hough dönüşümü (*Hough Transform*) metodlarını kullanmışlardır. Son adımda ise, yol koşullarının sürüş için uygun veya sürüş için uygun değil şeklinde bir sınıflandırma gerçekleştirmişlerdir.

Maurício Braga de Paula ve Cláudio Rosito Jung [12] “*Automatic Detection and Classification of Road Lane Markings Using Onboard Vehicular Cameras*” başlıklı çalışmalarında, araç üstü bir kamera (onboard vehicular camera) ile OpenCV arayüzünü kullanarak otomatik şerit çizgisi tespitini ve sınıflandırmasını tamamlamışlardır. Öncelikle, doğrusal parabolik modeli kullanarak şerit modelini tespit etmiş ve kamera parametrelerini ayarlamışlardır. Ardından uyarlamalı yumuşatma filtresi uygulayarak kenar tespitini gerçekleştirmişlerdir. Sınıflandırmada Bayes algoritmasını tercih ederek bu tez çalışmasında olduğu gibi kesikli, düz şeklinde bir sınıflandırma yapmışlar ve % 96 gibi yüksek bir başarı elde etmişlerdir. Ayrıca [12] nolu çalışmada, bu tezde gerçekleştirilenden farklı olarak kesikli/düz ve düz/kesikli şeklinde de sınıflandırma yapmışlardır. Bu tez çalışmasında öncelikli olarak kullanılan veri setinde yan yana şerit

izgileri ele alınmamıştır. Burada önerilen özüm, aracın bulunduėu řeridin saė veya solunda tek bir izgi olma varsayımı ile geliştirilmiştir. Dolayısı ile programda bulunacak aday izgi sayısı, 1 olarak girilmiştir. Öncelikle bu adedin en az 2'ye ıkarılması ve daha sonra da bu yapıya uygun kurallar oluşturulması gerekmektedir.

M.Daigavane ve Dr. P.R.Bajaj [13] alışmalarında, birden fazla kamera kullanarak olumsuz yol koşulları ve düşük düzeyde yansılardan etkilenmeyen řerit sınırı algılama işlemi yapmışlardır. Tasarımda, MATLAB (Matrix Laboratory) yazılımını ve Canny, Hough dönüşümü ve Rastgele Örnekleme Algoritmalarını (*Random Sample Consensus Algorithm-RANSAC*) kullanmışlardır.

Avishek Parajuli, Mehmet Celenk ve H. Bryan Riley [14] alışmalarını, kamera kullanarak beyaz renk dışında farklı bir renk ile boyanmış, yalnızca düz řerit izgilerini tespit ederek gerçekleştirmişlerdir. Canny Algoritması, Karınca Koloni Algoritması (*Ant Colony Optimization-ACO*) ve Hough dönüşümü tercih ederek hazırladıkları hibrid modelin verimli olduğunu belirtmişler ancak bir başarı yüzdesi vermemişlerdir.

Hunjae Yoo, Ukil Yang, ve Kwanghoon Sohn [15] “*Gradient-Enhancing Conversion for Illumination-Robust Lane Detection*” başlıklı alışmalarında, eşitli yansıma durumlarının bulunduğu yol koşullarını ele almışlardır. Bu tez alışmasında da uygulanmış olan Canny kenar bulma algoritması, Hough dönüşümü ve eğri uydurma (*curve fitting*) metotlarını kullanarak, % 96 lık bir başarı elde etmişlerdir. Ayrıca farklı yansıma ve yağışlı hava koşullarında alışmalarının yeterince iyi alışmadığını da söylemişlerdir.

özüm önerisi sunulan problem başlığı, kullanılan yöntemler, yazılım geliştirme ortamı, donanım bileşenleri ve başarı kategorilerini kapsayan Tablo 1.1' de özetlenen benzer alışmalar incelendiğinde, řerit izgisi algılama alışmalarında genellikle Hough dönüşümünün ve Canny kenar bulma algoritmasının kullanıldığı, yazılımsal olarak ise MATLAB programlama dilinin tercih edildiėi görülmektedir. Ayrıca gerçek zamanlı alışmalarda tek kameradan alınan görüntü yerine ok kameradan alınan ve bazı durumlarda da farklı algılayıcılar ile desteklenmiş görüntüler üzerinde alışılmıştır. Bu tezde ise aynı anda yalnızca tek kameradan alınmış boyutları ve görüş açıları farklı veri setleri ile alışılmıştır. Ayrıca kullanılan veri setleri farklı özelliklere sahip kameralar ile

oluşturulmuştur. Bu durum da başarının literatürdeki çalışmalar kadar yüksek olamayışını açıklamaktadır.

**Tablo 1.1** Benzer Çalışmalar ve Karşılaştırma Tablosu

<b>Kaynak</b>	<b>Çözüm önerisi sunulan problem başlığı</b>	<b>Kullanılan yöntem</b>	<b>Yazılım geliştirme</b>	<b>Donanım bileşenleri</b>	<b>Başarı Değerlendirmesi</b>
<b>[8]</b>	Gerçek zamanlı olarak düz şerit çizgisi algılama	CCD Kamera Parametrelerinin Düzeltilmesi (CCD Parameters Regulation), Hough Dönüşümü		Kamera: Basler A602f	Düz şerit çizgisi algılamada verimli, eğimli şerit çizgisinde uygun değil şeklinde bir değerlendirme yapılmıştır. Ancak % cinsinden bir değer verilmemiştir.
<b>[10]</b>	Yansıma değişiklikleri, kırık yol, gündüz/gece görüşü ve karmaşık şerit işaretlemeleri gibi zorlu durumlarda net olmayan şerit türlerinin akan trafikte gerçek zamanlı olarak algılanması	Dikey Ortalama Dağıtım Yöntemi (Vertical Mean Distribution Method-VMD), Canny Kenar Bulma Algoritması, K-ortalama Kümeleme Algoritması (K-means Clustering Algorithm) ve Rastgele Örnekleme Algoritması	MATLAB : R2010a	PC (Personel Computer): Intel Core Quad 2.67 GHz işlemci ve 2GB RAM  Kamera: 240*320 çözünürlük	Otonom araçlar için yeterli verimliliktedir şeklinde bir değerlendirme yapılmıştır. Ancak % cinsinden bir değer verilmemiştir.

[12]	Gerçek zamanlı otomatik şerit çizgisi tespiti ve sınıflandırılması	Doğrusal Parabolik Model, Bayes Algoritması	OpenCV	Araç üstüne yerleştirilmi ş kamera	%96
[15]	Çeşitli yansıma durumlarının bulunduğu yol koşullarında şerit çizgisi tespiti	Hough Dönüşümü, Eğri Uydurma		OV10630 algılayıcı, 1280 × 800 çözünürlük saniyede 15 frame	Maksimum %96 ve minimum %93 (Ayrıca yağışlı hava koşullarında yeterince iyi çalışmadığı belirtilmiştir.)
[16]	Farklı gün ışığı şartlarında (gece/gündüz görüşü) şerit çizgisi algılama	Gürbüz Algoritması (Robust Boosting Algorithm), Parça Filtre-(Particle Filter)		Kamera Algılayıcı GPS	Önerilen yöntem için yeterli verimlidir şeklinde bir değerlendirme yapılmıştır. Ancak % cinsinden bir değer verilmemiştir.
[17]	Çeşitli yol ve sürüş koşullarında şerit çizgisi algılama	Hough Dönüşümü	MATLAB	Kamera PC	Otonom araçlar için etkili şeklinde bir değerlendirme yapılmıştır. Ancak % cinsinden bir değer verilmemiştir.

[18]	Kent içi sokaklarda düz yolda renkli şeritleri algılama ve şerit çizgi türü sınıflandırma	Kalman Filtre (Kalman Filter), Parça filtre	MATLAB	Kamera: Standart DV-video, 25 Hz	Şerit çizgi algılama doğru olduğu sürece doğru sınıflandırma yapılabilmektedir şeklinde bir değerlendirme yapılmıştır. Ancak % cinsinden bir değer verilmemiştir.
[19]	Gerçek zamana yakın otomatik şerit sınır türleri (düz/kesikli/bitişik) tanımlama	Canny Kenar Bulma Algoritması, Kalman Filtresi, Hough Dönüşümü ve Fourier Analizi(Fourier Analysis)		Renkli kamera GPS PC: Pentium IV	Herhangi bir başarı değerlendirmesi yapılmamıştır.
[20]	Kent içi sokaklarda çeşitli koşullarda şerit algılama	Ters Perspektif Haritalama (Inverse Perspective Mapping-IPM), Gaussian Filtre(Gauss Filtre), RANSAC Eğri Uydurma Tekniği (RANSAC Spline FittingTechnique)	MATLAB : C++	Kamera: 50Hz, 640*480 çözünürlük, PC: Intel Core2, 2.4 GHz	% 100

[21]	Yol üzerinde bulunan şeridi algılama ve araç tipini sınıflandırma	Hough Dönüşümü C4.5 Makine Görme Algoritması(C4.5 Machine Learning Algorithm)		Kamera: CC-TV (Renkli) 320*240 çözünürlük	Oldukça verimli şeklinde bir değerlendirme yapılmıştır. Ancak % cinsinden bir değer verilmemiştir.
[22]	Kent içi sokaklarda bulutlu ve güneşli hava koşullarında şerit algılama	Ters Perspektif Haritalama, Kenar Bulma Algoritması Eğri Uydurma Algoritması (Fitting Lines Algorithm)	MATLAB	CMU VASC Image Database 256*240	% 97.2
[23]	Düz ve kıvrımlı şerit algılama	Adaptif Rastgele Hough Dönüşümü (Adaptive Random Hough Transformation-ARHT), Parça Filtre			Hızlı ve verimli şeklinde bir değerlendirme yapılmıştır. Ancak % cinsinden bir değer verilmemiştir.
[24]	Farklı gün ışığı koşullarında işaretli yol şeritleri ve sınırları algılama	Flood-fill Algoritması (Flood-fill Algorithm)		PC: Pentium-D 2.80 GHz, Kamera	Yüksek doğruluk şeklinde bir değerlendirme yapılmıştır. Ancak % cinsinden bir değer verilmemiştir.



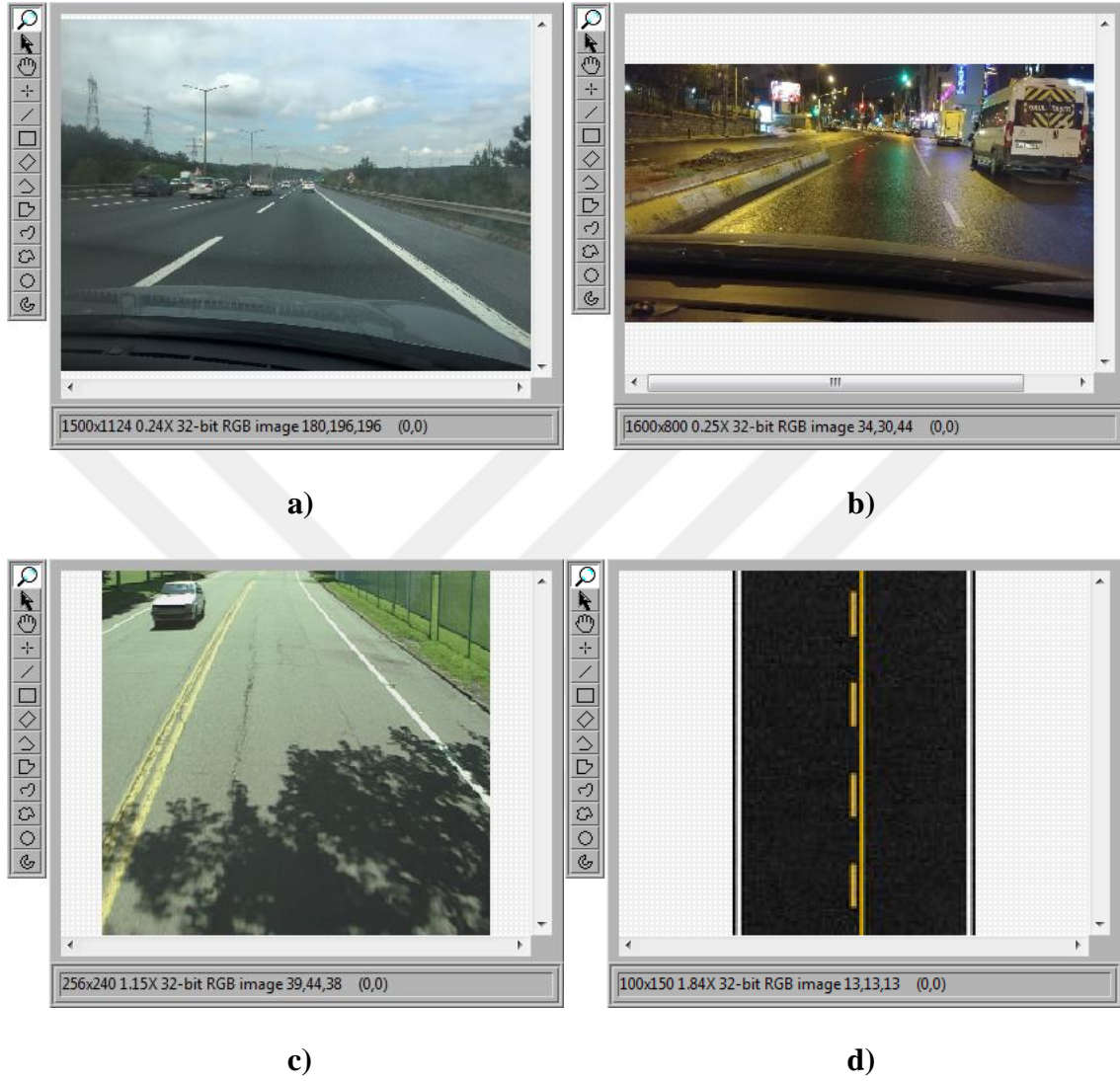
## 2. SANAL ENSTRÜMANTASYON TEMELLİ YAPAY GÖRÜ SİSTEMİ TASARIMI

Bu tez çalışmasında yapay görü sistemi geliştirme platformu olarak metin tabanlı bir programlama dili yerine (MATLAB, C++, OpenCV, ...) LabVIEW grafiksel geliştirme platformunun “Görme ve Hareket” fonksiyon paletinin (*Vision and Motion Function Palette*) kullanımı tercih edilmiştir. Sınır eğrilerinin çıkartılmasında Canny kenar bulma algoritması, görüntünün iyileştirilmesinde matematiksel morfolojik işlemlerden genişletme (*dilate*) ve aşındırma (*erode*) operatörleri ve şerit çizgisinin konumunun tespit edilmesinde Hough dönüşümü uygulanırken; çizgi türünün sınıflandırılmasında ise, if – then – else yapısından oluşan basit kural tabanlı bir algoritma tasarımı gerçekleştirilmiştir.

LabVIEW’ de görüntü işleme uygulamaları 2 farklı yöntem ile yapılabilir: 1) Hazır uygulama geliştirme şablonlarının bulunduğu “*Vision Assistant*” paketi kullanarak veya 2) Kişinin kendi algoritmasını tasarlayabildiği “Görme ve Hareket” fonksiyon paleti nesnelerini kullanarak. 1 nolu seçenek ile hazırlanan sistemler daha kolay oluşturulmaktadır ancak bu alanda yapılacak çalışmalar kısıtlıdır [25]. Kod yazmak yerine, sanal nesneler kullanılarak tasarımın daha kolay gerçekleştirilebildiği LabVIEW platformunda yukarıda bahsedilen nedenlerden dolayı 2 nolu seçenek tercih edilmiştir.

Çalışmada, 2 farklı veri setinden oluşan 168 adet gerçek yol görüntüsü ile 15 adet sentetik yol görüntüsünün bulunduğu veri seti birlikte kullanılmıştır. Başlangıç deneyleri, Clipartfest’ den elde edilen sentetik yol görüntüleri ile tamamlandıktan sonra, “Carnegie Mellon Üniversitesi Robotik Enstitüsü Görme ve Otonom Sistemler Merkezi” görüntü veri setinden elde edilen gerçek yol görüntüleri ile deneylere devam edilmiştir [26-27]. Ancak bu veri setinde yer alan görüntüler yalnızca düz şerit çizgisi barındırdığından tarafımızdan da farklı mobil cihaz kameralarından gerçek zamanlı yol görüntüleri elde edilerek deneyler tamamlanmıştır. Ayrıca gerçek yol görüntüleri günün farklı saatlerinde farklı gün ışığı koşulları, yansımalar, ağaç gölgeleri, silinmiş şerit çizgisi, kırık yol, kötü hava gibi olumsuz koşulları da barındırmaktadır. Ek olarak, bir

başka olumsuzluk da bu görüntülerin hepsinin farklı kamera ve görüş açılarından elde edilmiş olmasıdır. Şekil 2.1’ de örnek görüntülere yer verilmiştir.



**Şekil 2.1** Deneylerde kullanılan veri setlerine ait örnek yol görüntüleri **a)** Mobil cihaz 1 kamera **b)** Mobil cihaz 2 kamera **c)** Carnegie Mellon Üniversitesi **d)** Sentetik

Bu bölümde LabVIEW platformuna ait ön panel, blok diyagram, “Görme ve Hareket” fonksiyon paleti ile bu paletin alt paletlerinin ile içerdiği nesneler hakkında bilgiler verilmiştir. Ayrıca sistem tasarım aşamaları ile bu aşamalarda kullanılan yöntemler ve nesnelerden de detaylı olarak bahsedilmiştir.

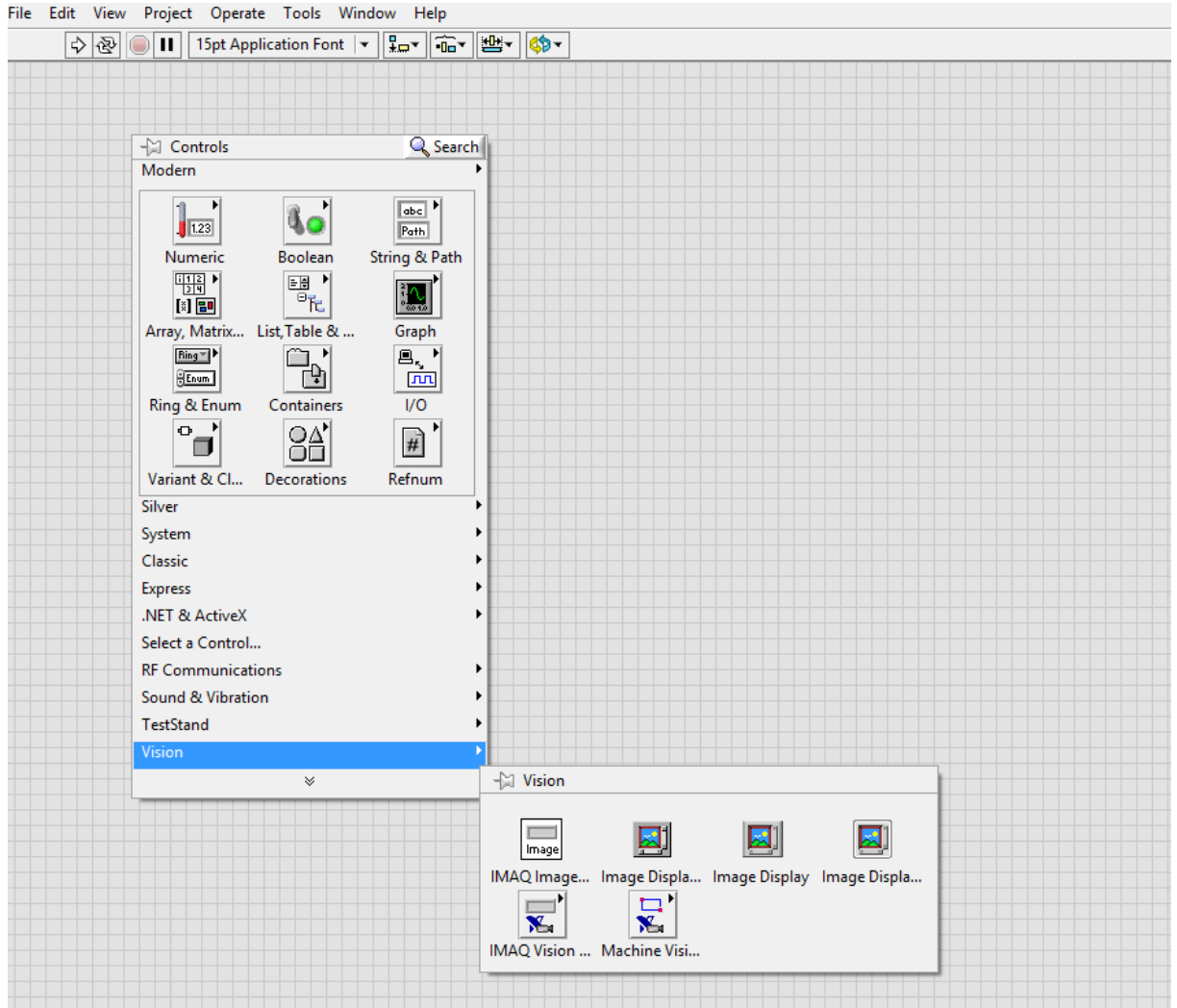
## 2.1. LabVIEW Grafiksel Geliştirme Platformu

LabVIEW, 1983 yılında National Instruments firması tarafından yazılım dillerindeki verimliliği arttırmak için ölçme ve enstrümantasyon odaklı olarak geliştirilmiş nesne-tabanlı bir programlama dilidir [28]. Aynı zamanda grafiksel sözcüğünden gelmekte olan G programlama dili olarak da isimlendirilmektedir. LabVIEW grafiksel geliştirme platformunda satırlarca metin tabanlı kod yazmak yerine birkaç sanal enstrüman (*Virtual Instruments*) ile akış diyagramları oluşturulabilmektedir.

LabVIEW, modüler yapıya sahip programlar oluşturma olanağına sahiptir. Uzun ve karmaşık programlar oluşturulurken program alt programlara ayrılabilir. Böylelikle tasarım basitleştirilebilmekte ve aynı zamanda hata ayıklama işlemi de kolaylaşmaktadır. Genel olarak bir uygulama, ön panel (*front panel*) ve blok diyagram (*block diagram*) olmak üzere 2 tasarım ortamından oluşmaktadır.

### 2.1.1. Ön panel (Front panel)

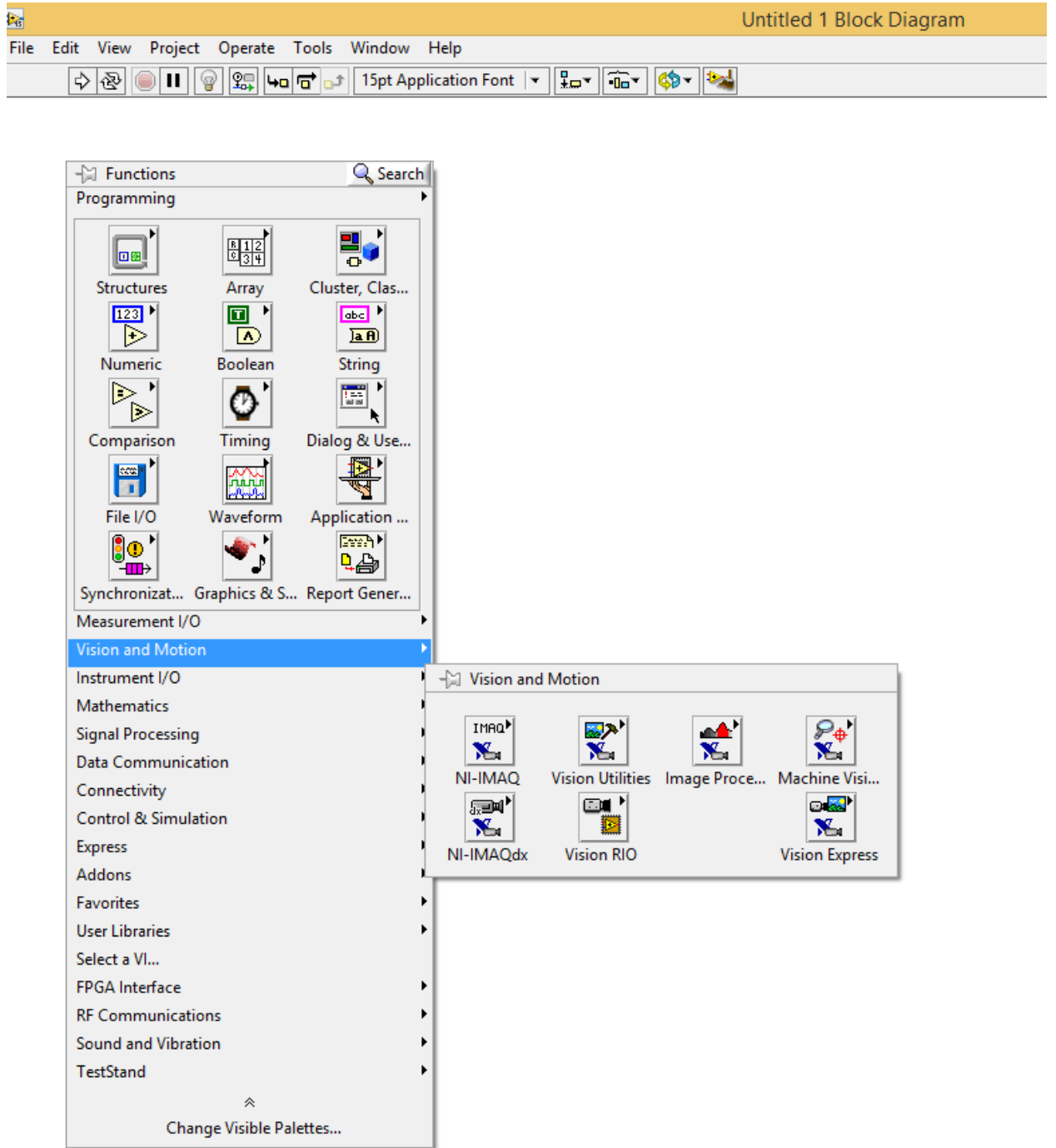
Ön panel, kullanıcının uygulama sonuçlarını gözlemleyebildiği ve uygulama üzerinde değişiklikler yapabildiği arayüz kısmıdır. Şekil 2.2' de görülen kontrol paletinde yer alan nesneleri barındırır. Bu alanda, kullanıcının izleme ve kontrol etme gibi görevler için ihtiyaç duyabileceği buton, kadran, anahtar ve diğer giriş nesneleri ile metin tabanlı mesaj kutusu, grafik, led ve gösterge gibi çeşitli verilerin izlenmesine yönelik kategoriler bulunmaktadır. Kontrol nesneleri ile sistem giriş parametreleri belirlenip, simüle edilirken; göstergeler ile çeşitli sistem çıktı ve bildirimlerin izlenmesine yönelik çıkış nesneleri simüle edilmektedir.



**Şekil 2.2** “Görme” nesne kutusunu da içeren LabVIEW platformu ön panel görüntüsü

### **2.1.2. Blok diyagram (Block diagram)**

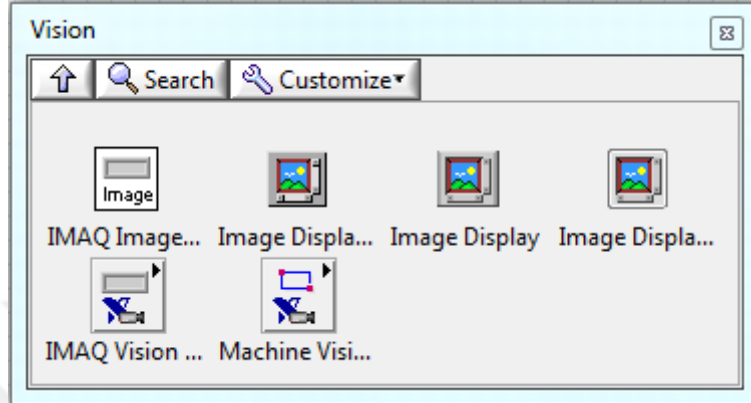
Asıl programın oluşturulduğu kısımdır. Şekil 2.3’ de görüldüğü gibi nesneleri, nesneler arasındaki iletişimi sağlamak için terminalleri ve kablo bağlantılarını içerir. Kullanıcı bu arayüz kısmında tasarımı gerçekleştirir ve sonuçları ön panel kısmından görüntüler.



Şekil 2.3 “Görme ve Hareket” fonksiyon paletini de içeren LabVIEW platformu blok diyagram görüntüsü

### 2.1.3. “Görme” kontrol kaleti

“Görme” kontrol paleti, LabVIEW 7.0 sürümünde veya daha sonraki sürümlerde kullanılabilir. Görme kontrol paletinde Şekil 2.4 ‘de görüldüğü gibi 6 temel nesne mevcuttur. Bu nesnelerin işlevleri aşağıda tanımlanmıştır.



Şekil 2.4 “Görme” kontrol paleti



**IMAQ Image.ctl** — Bu nesne, görüntü türünün tanımı olup veri tipini açıklar. Ön paneldeki görüntüye ait veri türünü temsil etmek için bu kontrol kullanılabilir.



**Image Display** — LabVIEW’ de ön panelde bulunan görüntüleri doğrudan görüntülemek ve ilgi alanı (*Region of Interest-ROI*) oluşturmak için bu nesne kullanılabilir. Klasik ve 3 boyutlu (3B) olmak üzere iki türü mevcuttur.



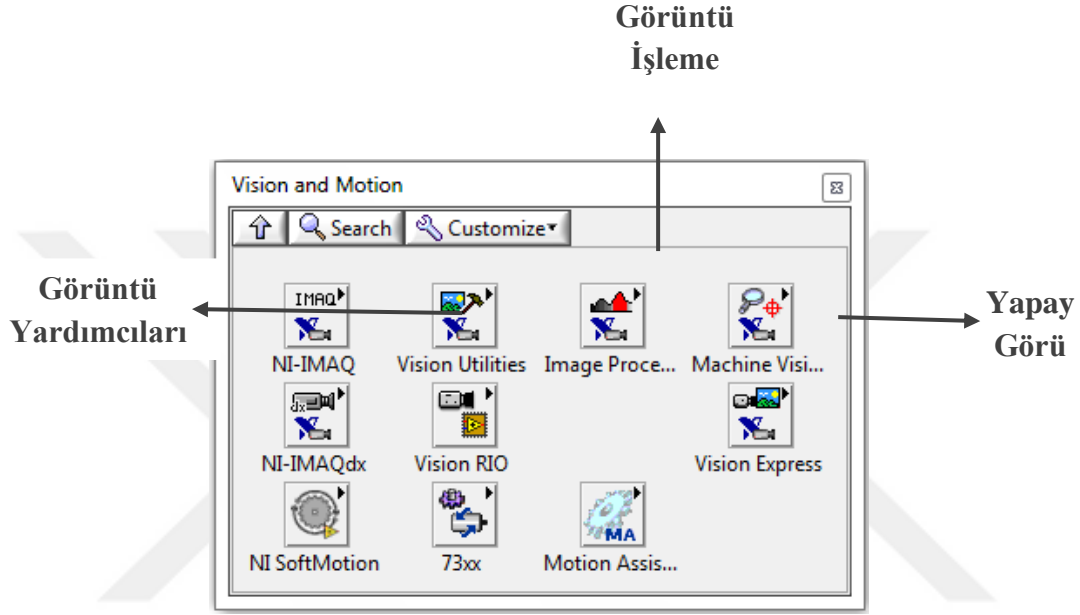
**IMAQ Vision Controls** — Kullanıcının programın içinde doğrudan kontrol işlevselliği elde etmesi için oluşturulmuş nesne türüdür.



**Machine Vision Controls** — Yapay Görü denetimlerinde işlevselliği elde etmek için oluşturulmuş nesnelerdir.

#### 2.1.4. “Görme ve Hareket” fonksiyon paleti

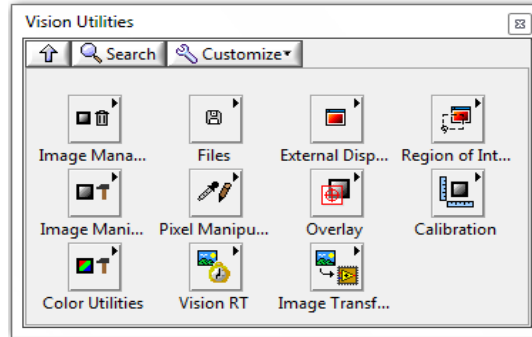
LabVIEW için “Görme ve Hareket” fonksiyon paleti Şekil 2.5’ te görüldüğü gibi üç ana fonksiyon ögesini barındırmaktadır. Temel olarak görüntü yardımcıları, görüntü işleme ve yapay görü fonksiyonlarından oluşmaktadır. Bu bölümde bu nesneler ve bunların alt nesnelerinden bahsedilmektedir.



Şekil 2.5 “Görme ve Hareket” fonksiyon paleti

##### 2.1.4.1. “Görme Yardımcı” nesneleri

Şekil 2.6’ da görülen “Görme Yardımcı” nesneleri, fonksiyonları işleme ve *IMAQ* *Vision* ekran görüntülerinde işlemlerin gerçekleştirilmesine olanak sağlar.



Şekil 2.6 “Görme Yardımcı” fonksiyon paleti



**Image Management** — Görüntüleri yönetmek için oluşturulmuş bir gruptur. Görüntünün özelliklerini (büyüklüğünü, dengesini veya kopyasını) oluşturmak, düzenlemek, ayarlamak ve okumak için kullanılır. Ayrıca gelişmiş özellikler kullanılarak, görüntünün sınır bölgesi tanımlanabilir ve görüntü işaretçisi verilerine erişilebilir.



**Files** — Görüntü dosyalarını okur. Dosya biçimi, kullanıcı tarafından bilinen standart bir format (BMP, TIFF, JPEG, JPEG2000, PNG ve AIPD) veya standart olmayan bir format olabilir. Her iki durum için de, okunan pikseller istenen görüntü dosyası tipine dönüştürülebilir. Ayrıca bir dosyada yer alan görüntü hakkında bilgi almak için oluşturulmuştur gruptur.



**External Display** — Görüntüleri veya ilgi alanını (ROI - Region of Interest) harici bir pencerede görüntüleme, görüntü tarayıcısı oluşturma ve kullanma olanağı sunar.



**Region of Interest** — Kullanıcının görüntü üzerinde bir ilgi alanı belirleyebilmesini sağlar. *IMAQ Construct ROI* nesnesi ile birlikte kullanılarak, görüntüyü belirtilen pencerede görüntüler ve ilgi alanı için araç seti sağlar. Oluşturucu penceresinde seçim tamamlandığında, görüntü ekranına belirtilen ilgi alanını döndürür. ROI nesnelerinin kullanılması özelliklerinin belirlenmesi hakkında detaylı bilgi Bölüm 2.2.2.5' te verilmiştir.



**Image Manipulation** — Görüntüyü düzenlemek için kullanılır. Görüntü yeniden örneklenebilir, döndürülebilir, kaydırılabilir ve içerisinden istenen kısımlar ayıklanabilir.





**Pixel Manipulation** — Görüntüdeki pikselleri ayrı ayrı okumaya veya düzenlemeye yarar. Görüntüdeki satır veya sütun boyunca olan piksel değerleri okunabilir veya değiştirilebilir. Belirli değere sahip pikseller doldurulabilir.



**Overlay** — Görüntüdeki piksel değerleri değiştirilmeden görüntüleme ortamında üzerine grafik, metin veya şekil eklenebilir. Sonuçlar incelenebilir.



**Calibration** — Görüntüyü uzaysal düzlemde kalibre etmek için kullanılır. Görüntüleme sistemindeki olası perspektif hataları veya doğrusal olmayan bozulmalar bu paletin nesneleri sayesinde düzeltilebilir. Piksel koordinatları gerçek-dünya koordinatlarına dönüştürülür.



**Color Utilities** — Görüntünün renk değerlerini ya da düzlemlerini değiştirmek için kullanılır. Renkli bir görüntüyü 2B bir diziye dönüştürülür veya tersini gerçekleştirilir. Ayrıca piksel değerleri okunabilir ve değiştirilebilir.



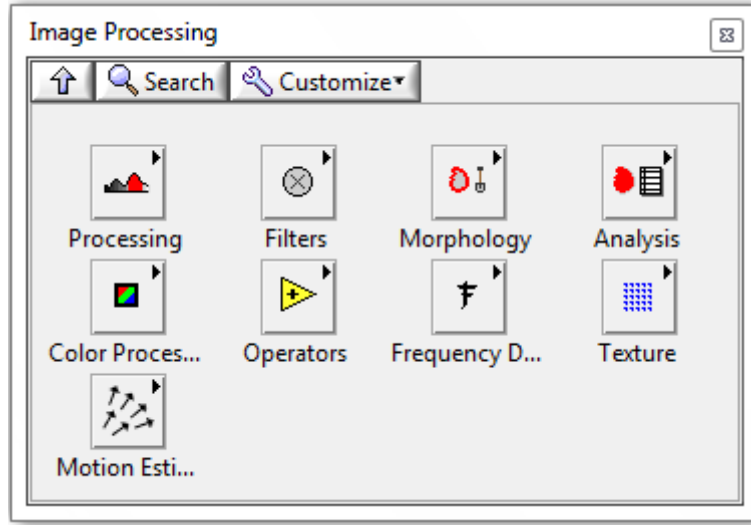
**Vision RT** — Gerçek zamanlı sistemler için ekran ayarlarını yapılandırmak ve değiştirmek için kullanılır.



**Image Transfer** — Görüntüleri ana bilgisayardan Alanda Programlanabilir Kapı Dizileri (*Field Programmable Gate Array –FPGA*)' ne aktarma veya FPGA' dan bilgisayara aktarmak için kullanılır.

#### 2.1.4.2. Görüntü işleme

*IMAQ Vision* içinde yer alan görüntü işleme fonksiyonları, analiz, filtreleme ve görüntü işleme amaçlı kullanılır. Şekil 2.7' de bu palette yer alan nesneler görülmektedir.



Şekil 2.7 “Görüntü İşleme” fonksiyon paleti



**Processing** — Gri tonlu ve ikili görüntüleri işlemek için oluşturulmuş bir gruptur. Çeşitli eşik seviyeleri uygulanarak gri tonlamalı bir görüntü, ikili görüntüye dönüştürülebilir. Ayrıca önceden tanımlanmış bir bölgede veya ROI’ de zıtlık (*contrast*) bilgilerini değiştirmek ve tersine çevirmek için kullanılabilir.



**Filters** — Görüntü iyileştirme amacıyla filtreleme yapılır. Bir görüntüyü düzeltmek (pürüzsüzleştirmek), gürültüyü çıkarmak, görüntüdeki kenarları keskinleştirmek veya vurgulamak için kullanılır. Bu işlemlerde önceden tanımlanmış veya oluşturulmuş konvolüsyon çekirdeği kullanılabilir. Bu çalışmada kullanılan filtre türleri ile ilgili ayrıntılı bilgi Bölüm 2.2.2.3’ te verilmiştir.



**Morphology** — Görüntü üzerinde morfolojik işlemleri gerçekleştirmek üzere oluşturulmuş bir gruptur. Bunlardan bazıları, gri tonlu veya ikili görüntülerde genişletme ve aşındırma gibi temel morfolojik işlemleri gerçekleştirmek için kullanılırken; bazıları da ikili görüntülerin kalitesini artırmak amacıyla farklı şekil özelliklerine göre görüntü parçacıklarının içinde yer alan boşlukları doldurma, görüntü sınırına dokunan parçacıkların çıkartılması ve istenmeyen parçacıkların kaldırılması için kullanılabilir. Yardımcı paletlerdeki nesnelerde ise, parçacıklar ayrılabilir, parçacıkların

taslağı bulunabilir veya dairesel parçacıklar algılanabilir. Morfoloji operatörleri ile ilgili ayrıntılı bilgilerden Bölüm 2.2.2.4 de bahsedilmektedir.



**Analysis** — Gri tonlu ve ikili görüntülerin içeriği analiz edilir. Görüntünün histogram bilgileri ve gri tonlama istatistiklerini hesaplamak için kullanılır. Görüntüdeki herhangi tek boyutlu görünüş boyunca piksel bilgileri ve istatistikleri alınabilir. İkili görüntüdeki parçacıklar algılanabilir ve büyüklükleri (uzunluk, çap vb. ) ölçülebilir.



**Color Processing** — Renkli görüntüleri analiz etme ve işleme için oluşturulmuş bir gruptur. Histogram hesaplama ve arama tabloları uygulama imkânı da sağlamaktadır. Görüntüyle ilgili parlaklık, zıtlık, gama bilgileri ve resmin eşik değeri değiştirilebilir. Bu grubun bazı nesneleri renk eşleştirme işlemi kullanılarak farklı bölgelerdeki veya farklı görüntülerdeki renk bilgileri ile karşılaştırılabilir.



**Operators** — Görüntü üzerinde temel aritmetiksel ve mantıksal işlemleri gerçekleştirmek için oluşturulmuş bir gruptur. Bu fonksiyona ait nesnelerden bir kısmı bir görüntü ile diğer bir görüntü veya bir görüntü ile sabit arasında toplama, çıkarma, çarpma, bölme için kullanılır. AND/NAND, OR/NOR, XOR/XNOR gibi mantıksal operatörlerden oluşan diğer bir kısmı ise, bir görüntü ve diğer görüntüler arasındaki piksel karşılaştırmaları yapmak için kullanılır. Ek olarak maskeleyme işlemi kullanılarak bir görüntüyü işlemek için bölge seçimine olanak verir.



**Frequency Domain** — Görüntüyü frekans düzleminde filtreleme imkânı sunar. Karmaşık görüntüdeki genlik ve faz değerlerini, gerçek ve sanal olarak ayrılabilir. Karmaşık bir görüntü ile farklı bir görüntü arasında toplama, çıkarma gibi temel aritmetik işlemler yapılabilir. Ayrıca, *Fast Fourier Transform (FFT)* kullanılarak, mekânsal düzlemden frekans düzlemine geçilebilir. Ardından ters *fourier* dönüşümü yapılarak daha az hesaplama yükü ile işlenmiş görüntü elde edilebilir.



**Texture** — Görüntü üzerindeki istenmeyen parçacıkları algılamak ve doku özelliklerini çıkarmak için kullanılır.



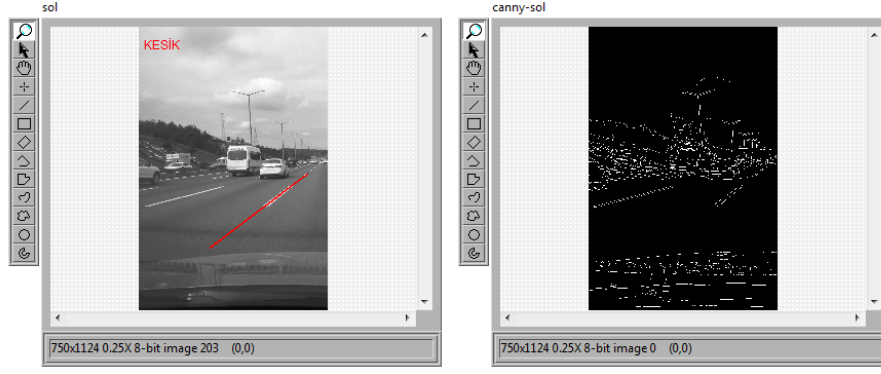
**Motion Estimation** — Görüntü dizisinde gözlemlenen hareketi tahmin etmek için kullanılır.

## 2.2. Sistem Tasarımı

Bu başlık altında, şerit çizgisinin konumunun algılanması, algılanan konumun doğruluğunun teyit edilmesi ve ardından sınıflandırılması süreçlerini kapsayan tasarımın aşamalarından bahsedilmektedir. Tasarım, görüntünün gri skalada okunması ve bölütlenmesi, ardından sağ ve sol şerit çizgilerini barındıran görüntü bölmeleri için paralel olarak yürütülen sınır eğrilerinin çıkartılması, matematiksel morfolojik işlemlerin uygulanması, şerit çizgilerinin tespit edilmesi ve gerçek koordinatların belirlenerek sınıflandırılması adımlarını içeren programın oluşturulduğu blok diyagram ve sonuçların görüntülendiği ön panel kısımlarından oluşmaktadır.

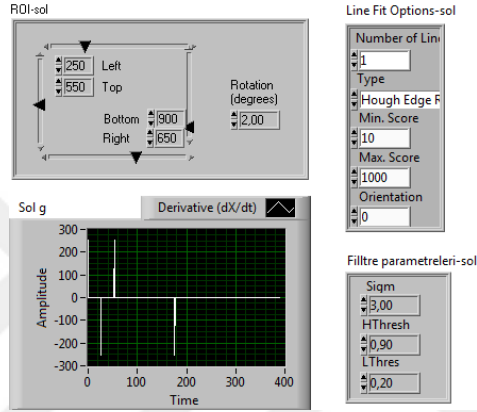
### 2.2.1. Şerit çizgisinin tespit edilmesi ve sınıflandırılmasına yönelik geliştirilen ön panel

Sistemde uygulanan adımların sonuçlarının görüntülendiği ve değişikliklerin yapılabildiği kısımdır. Şekil 2.8’ de görüldüğü gibi ön panel üzerinden, işlem uygulanacak görüntünün çağırılması, çağırılan görüntünün niteliklerine veya kullanıcının isteğine göre ROI’ de değişiklik yapılması, morfolojik işlemlerin operatör türlerinin değiştirilmesi, Hough dönüşümü ve Canny filtrelerine ait parametre değerlerinin düzenlenmesi ve tespit edilen şerit çizgisi bölgelerinin grafiksel olarak gözlemlenmesi mümkündür. Ön panel, sistem üzerinde sağ ve sol şerit çizgi bölgeleri ayrı olmak üzere paralel olarak yürütülen her bir adıma ait işlem sonuçlarının gözlemlenebileceği şekilde tasarlanmıştır.

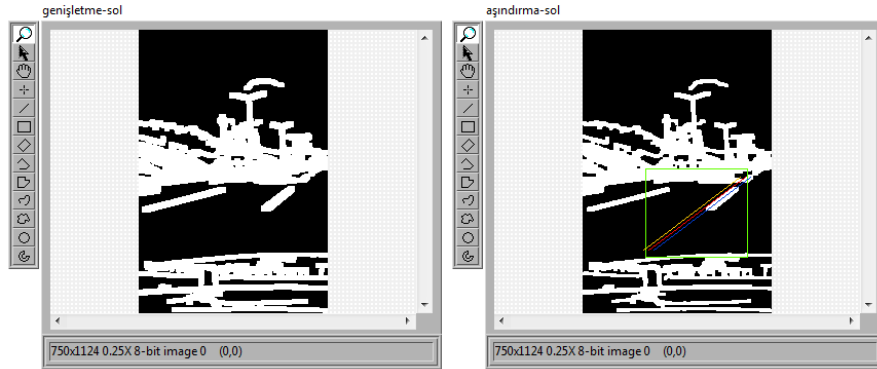


a)

b)



c)



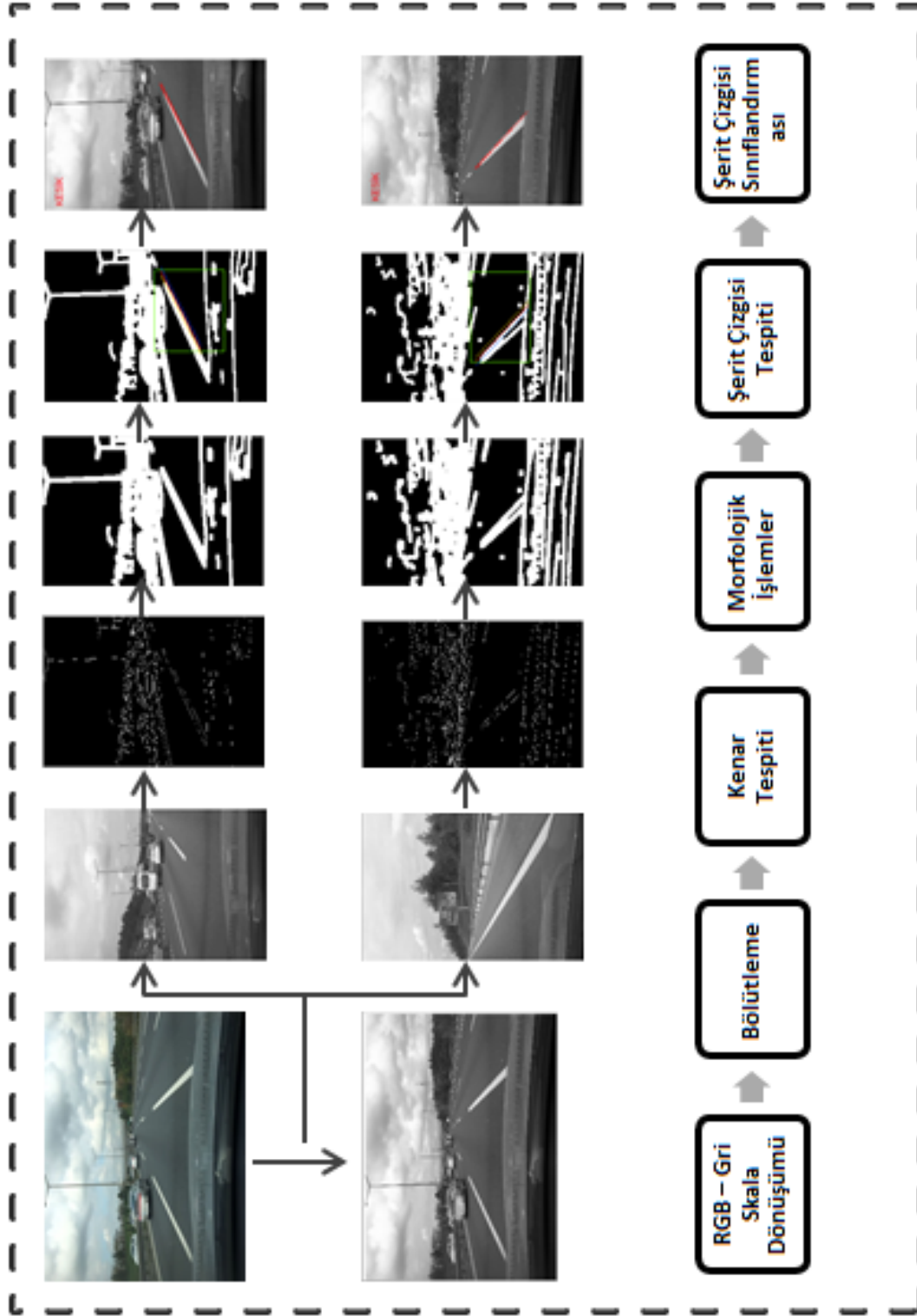
d)

e)

**Şekil 2.8** Şerit çizgisinin tespit edilmesi ve sınıflandırılmasına yönelik geliştirilen örnek ön panel görüntüsü **a)** Sol şerit için işlem sonucu **b)** Canny filtesi çıkış görüntüleri **c)** Tespit edilen çizginin grafiksel gösterimi ve ROI, Hough dönüşümü ile Canny filtesi işlemlerine yönelik parametrelerin belirlenmesi **d)** ve **e)** Morfolojik işlemler çıkış görüntüleri

### **2.2.2. Şerit çizgisinin tespit edilmesi ve sınıflandırılmasına yönelik geliştirilen blok diyagram**

Şerit çizgisinin tespit edilmesi ve sınıflandırılmasına yönelik olarak geliştirilen sistem Şekil 2.9' da görüldüğü gibi 6 adımdan oluşmaktadır. Bu adımlar: renkli görüntünün gri skalaya dönüştürülmesi, sol ve sağ şerit çizgi bölgelerini ayırarak; işlem süresinin kısaltılmasına yönelik olarak bölütlenmesi, kenar tespiti için sınır eğrilerinin çıkartılması, görüntüyü iyileştirmek, olası hatalı sonuçları önlemek için genişletme ve aşındırma morfolojik işlemlerinin uygulanması, şerit çizgilerinin tespit edilmesi amacıyla Hough dönüşümünün uygulanmasının ardından doğruluk oranının artırılması ve tespit edilen şerit çizgi türünün düz/kesik kararının verilmesi için basit kural tabanlı olarak hazırlanan sınıflandırılma basamaklarından oluşmaktadır.



Şekil 2.9 Sistem blok şeması

### 2.2.2.1. RGB-Gri skala dönüşümü

LabVIEW’de temel olarak karmaşık, renkli ve gri skalalı olmak üzere 3 tip görüntü işlenebilmektedir. *IMAQ Read File VI* nesnesi ile BMP, TIFF, JPEG, JPEG2000, PNG ve AIPD dosya türlerinden biri okunarak; *IMAQ Create* nesnesi ile Tablo 2.1’ de belirtilen görüntü türlerinden istenen herhangi birine dönüşüm gerçekleştirilebilmektedir.

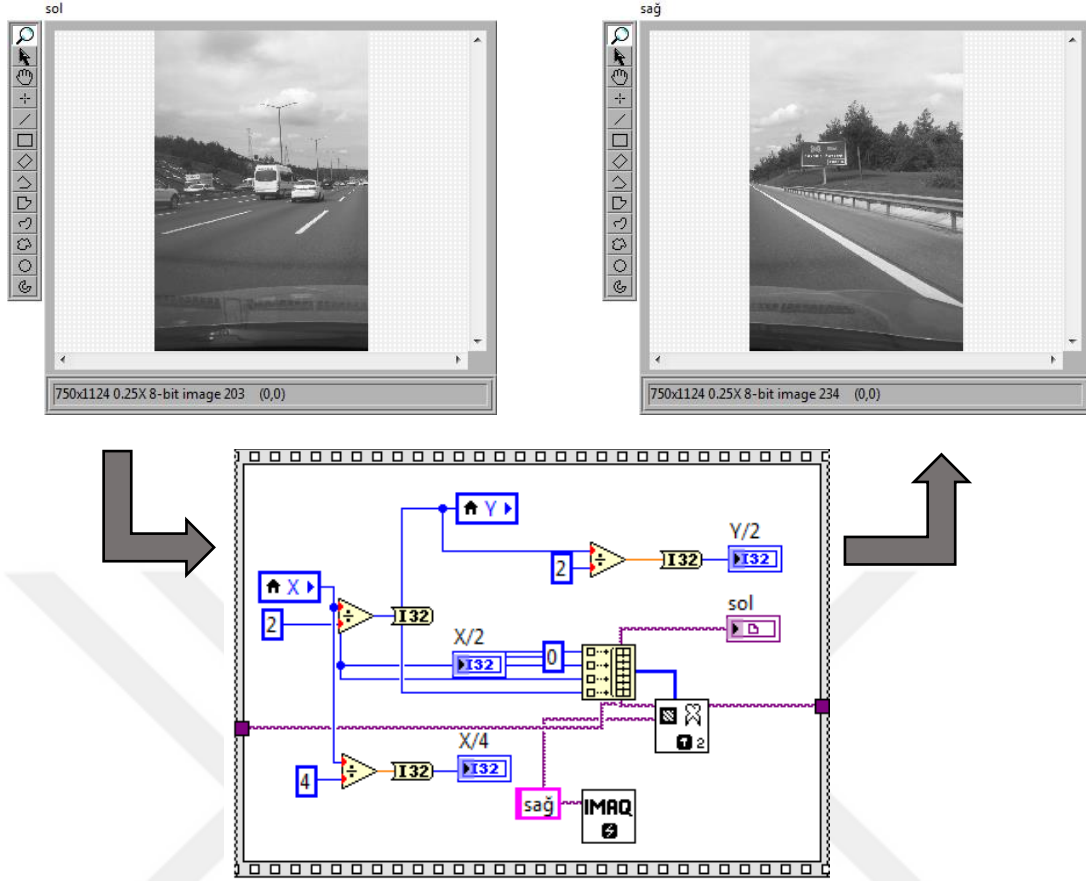
Gerçek ve sentetik yol görüntülerinin bulunduğu veri setlerinde yer alan renkli görüntüler Şekil 2.10’ da görüldüğü gibi RGB (Red Green Blue) olarak okunmakta ve blok diyagram üzerinden *IMAQ Create* nesnesi aracılığıyla 8-bit işaretli tamsayı gri skalalı görüntü formatına dönüştürülmektedir.

**Tablo 2.1** Dönüştürülebilecek Görüntü Türleri

Görüntü Türü	Bir Hücre için Bit Değeri
8-bit (İşaretsiz) Tamsayı Gri Skala	8-bit Gri Skala Değeri (Standart Monokrom)
16-bit (İşaretsiz) Tamsayı Gri Skala	16-bit Gri Skala Değeri
32-bit Kayan Nokta Gri Skala	32-bit Gri Skala Değeri
32-bit Karmaşık RGB Renkli	2x32-bit Kayan Nokta RGB değeri
32-bit (İşaretsiz) RGB Renkli	32-bit RGB Değeri (Alfa-kullanılmaz, Kırmızı, Yeşil, Mavi)
32-bit (İşaretsiz) HSL Renkli	32-bit HSL Değeri (Alfa-kullanılmaz, Renk, Doyum, Işık Şiddeti)
64-bit (İşaretsiz) RGB Renkli	64-bit RGB Değeri (Alfa-kullanılmaz, Kırmızı, Yeşil, Mavi)
16-bit (İşaretsiz) Gri Skala	16-bit Gri Skala Değeri (Standart Monokrom)







Şekil 2.11 Bölütlenmiş yol görüntüsü ve bu aşamaya ait blok diyagram

### 2.2.2.3. Kenar tespiti

Gri seviyesindeki yerel değişimler kenar olarak adlandırılmaktadır. Bu değişimlerin bulunmasında türev hesabı yapılmaktadır. Hesaplama sonuçlarında, gürültünün kenar olarak algılanması ve doğru konumun tespit edilmesi önemli hususlardır. Görüntü sınırı tespit eden algoritmalarından 1.derece türeve dayalı olanlar Roberts, Prewitt, Sobel, Canny, ve 2.derece türeve dayalı olanlar Marr-Hildreth olarak sıralanabilir.

### Canny kenar algılama

John F. Canny tarafından 1986 yılında geliştirilmiş, kenar bulmada etkin olan çok aşamalı bir algoritmadır. Düşük hata oranı ve kenar konumlarının doğru tespiti prensibiyle çalışmaktadır [29]. Birince aşamada belirlenen varyans için standart sapma ( $\sigma$ ) değerine göre, Eşitlik 2.1' de olduğu gibi Gaussian çekirdek ile konvolusyon

alınarak gürültü temizlenir. Varyans değerinin büyük seçilmesi gürültü giderimini arttırmakta; aynı zamanda görüntünün bulanıklaşmasına yol açmaktadır. Varyans değerinin küçük seçilmesi ise gürültü giderilmesini azaltmakta ancak kenar belirginliğini artırmaktadır. Birinci aşamanın ardından kısmi türev elde etmek için görüntü yatay ve dikey yönde Eşitlik 2.2 ve 2.3' teki gibi Eşitlik 2.4 ve Eşitlik 2.5' te görülen gradyan operatörleri ile filtrelendir. Ardından Eşitlik 2.6' daki gibi gradyan hesabı yapılır ve Eşitlik 2.7' teki gibi kenar yönünün ve büyüklüğünün azimut açısı hesaplanır. Ardından kenarlar Non- Maximum Suppression (NMS) işlemi ile inceltir. Son aşamada ise, ikili eşikleme yapılarak işlem tamamlanır [30].

$$G(x,y) = \exp[-(x^2+y^2) / 2\sigma^2] / 2\pi\sigma^2 \quad (2.1)$$

$$E_x[i,j] = (I[i+1,j] - I[i,j] + I[i+1,j+1] - I[i,j+1]) / 2 \quad (2.2)$$

$$E_y[i,j] = (I[i,j+1] - I[i,j] + I[i+1,j+1] - I[i+1,j]) / 2 \quad (2.3)$$

$$G_x \begin{pmatrix} -1 & 1 \\ -1 & 1 \end{pmatrix} \quad (2.4)$$

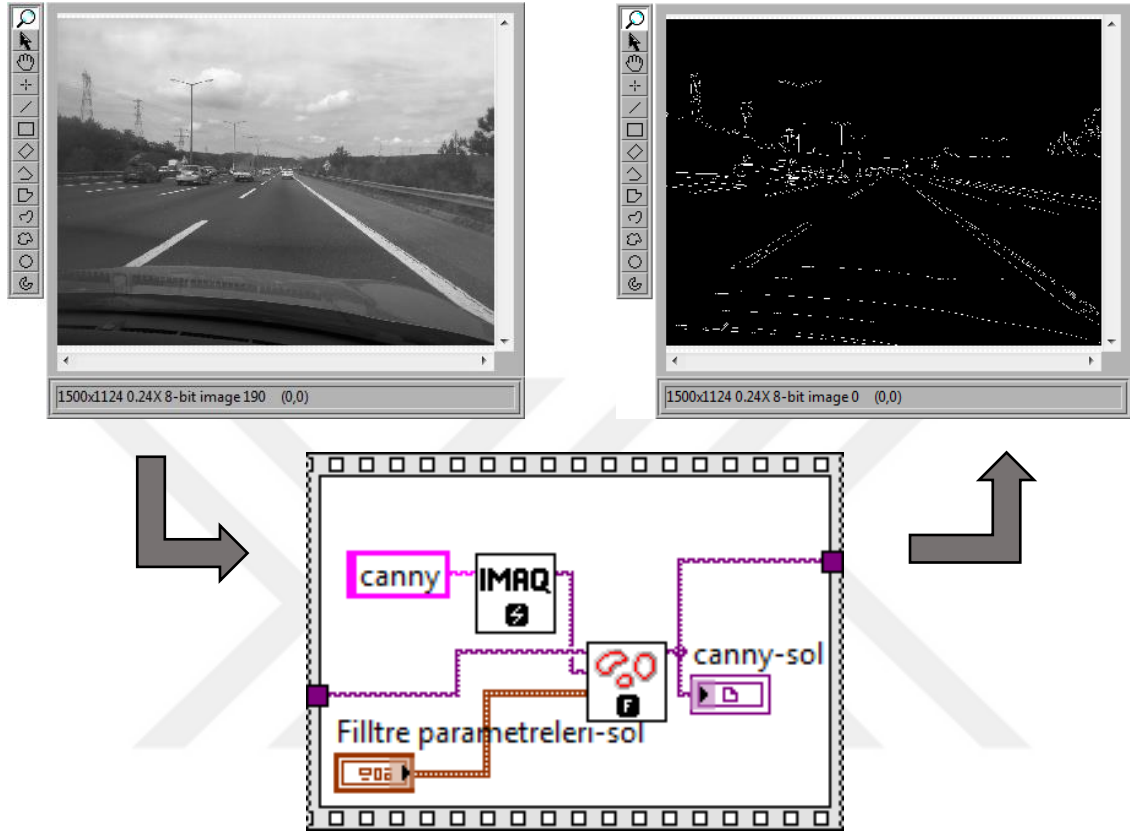
$$G_y \begin{pmatrix} -1 & 1 \\ -1 & 1 \end{pmatrix} \quad (2.5)$$

$$\|M(i,j)\| = \sqrt{E_x[i,j]^2 + E_y[i,j]^2} \quad (2.6)$$

$$\theta(i,j) = \arctan (E_y[i,j] / E_x[i,j]) \quad (2.7)$$

Bu tez çalışmasında kullanılan görüntülerde kenar tespiti için Sobel, Prewitt, Roberts ve Canny algoritmaları denenmiş; yüksek hassasiyet ve detaylı sınır çıkarımı sağlamasından dolayı ideal olarak Canny algoritması tespit edildikten sonra *IMAQ Canny Edge Detector* nesnesi aracılığıyla kenar tespiti gerçekleştirilmiştir. Ancak bunların yanında Canny algoritmasının işlemsel yükü de fazladır. *Canny Edge Detector* nesnesinde, kenar tespiti başarısında etken olan 3 önemli parametre bulunmaktadır: Sigma, H threshold ve L threshold. Sentetik yol görüntülerinde Gauss düzgünleştirme filtresi değeri 2 iken; gerçek yol görüntülerinde bu değer 3 olarak alınmıştır. Diğer taraftan tüm veri setlerinde yer alan görüntüler için, piksel değerlerinin üst yüzdesi (H

threshold) 0,90 ve piksel değerlerinin alt yüzdesi (L threshold) 0,20 olarak belirlenmiştir. Şekil 2.12’ de Canny filtresinin uygulanmasından önce ve sonraki görüntülere yer verilmiştir.

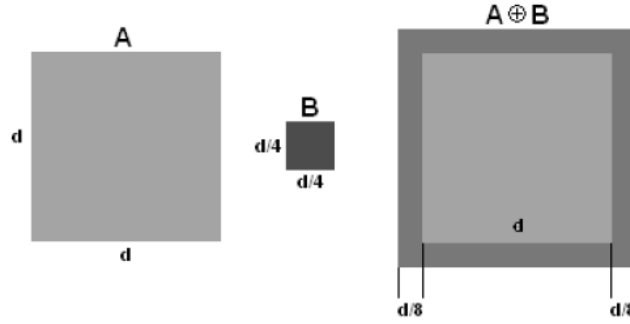


Şekil 2.12 Kenar belirleme işlemi öncesi ve sonrası gerçek yol görüntüsü ve bu aşamaya ait blok diyagram

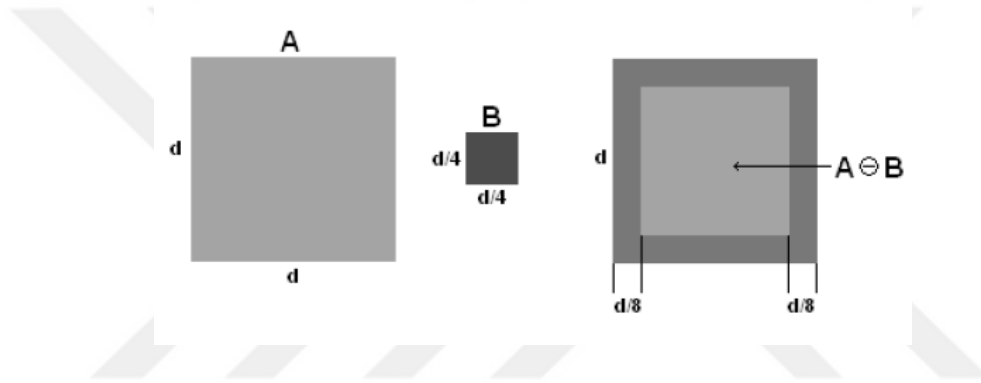
#### 2.2.2.4. Morfolojik işlemler

Görüntü bölgelerinin tanımlanmasında matematiksel morfoloji kullanılmaktadır. Temel küme işlemlerine dayanan bu süreçte görüntü sınırı, görüntünün iskeleti gibi yapılar ortaya çıkarılmaktadır. Gri skalalı ve ikili görüntüler üzerine uygulanabilmektedir. Genişletme ve yayma operasyonları sıkça kullanılan türlerdir. Şekil 2.13 ve Şekil 2.14’ te  $A \oplus B$  ile temsil edilen genişletme (*dilate*) ve  $A \ominus B$  ile temsil edilen aşındırma (*erode*) operatörlerinin görüntüler üzerindeki etkileri gösterilmektedir. Burada A işlem yapılacak görüntüyü; B ise işleminin şeklini belirleyen yapıtaşı belirtmektedir. Yapıtashi 2B bir eleman olup tek sayılı büyüklük değerlerinden oluşmalıdır. Yapıtashi

büyükliklerinin çift sayıda verilmesi halinde kullanılan nesne otomatik olarak değerlerini bir alt sınırdan yer alan tek sayı olarak değiştirerek işlem yapmaktadır.

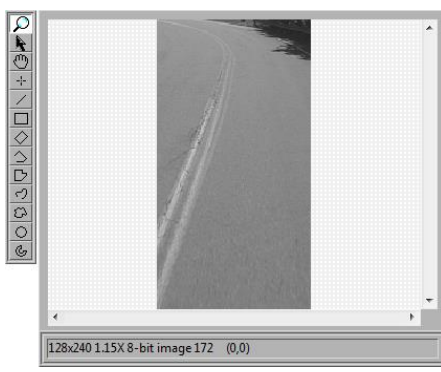


**Şekil 2.13** Genişletme işleminin görüntü üzerindeki etkisi

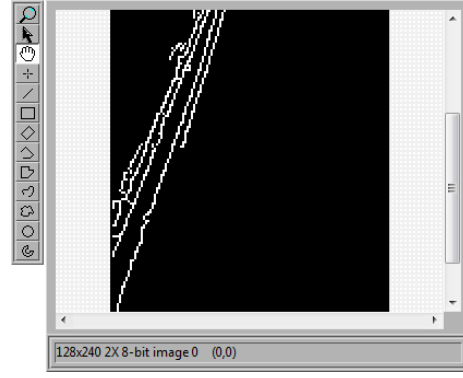


**Şekil 2.14** Aşındırma işleminin görüntü üzerindeki etkisi

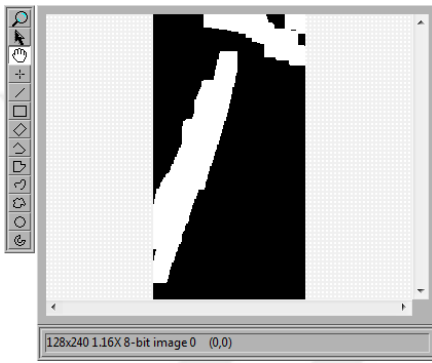
Bu çalışmada kullanılan veri setinde yer alan görüntülerde silinmiş veya eskimiş şerit çizgisi gibi olumsuz koşulların bulunması nedeniyle şerit çizgilerinde kopukluklar bulunmaktadır. Görüntüdeki kopuklukların giderilmesi ve iyileştirmelerin yapılmasında, U8, U16, I16 ve SGL formatındaki görüntülerde işlem yapılmasına imkân sunan *IMAQ Gray Morphology* nesnesi kullanılmıştır. İşlem hızı, yapıtaşının büyüklüğüne bağlıdır. Bu nesne ile gerçekleştirilebilecek işlemler listesi Tablo 2.2’ de belirtilmiştir. Şekil 2.15’ te ise silinmiş bir yol çizgisinin bulunduğu görüntü üzerinde Canny kenar bulma algoritması uygulandıktan sonra sırasıyla genişletme ve aşındırma operatörleri ile iyileştirilmesine ait örneğe yer verilmiştir.



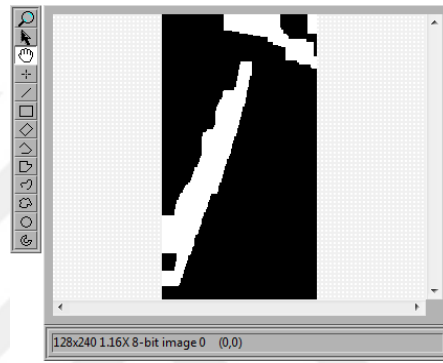
a)



b)



c)



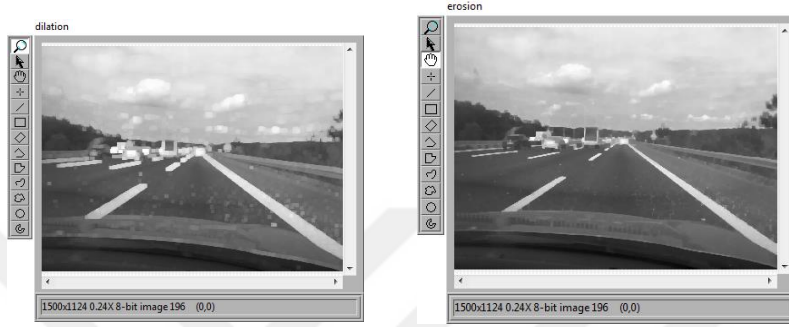
d)

**Şekil 2.15** Silinmiş şerit çizgisi iyileştirme aşamaları **a)** Giriş görüntüsü **b)** Canny filtresi uygulanmış görüntü **c)** Genişletme işlemi etkisi **d)** Aşındırma işlemi etkisi

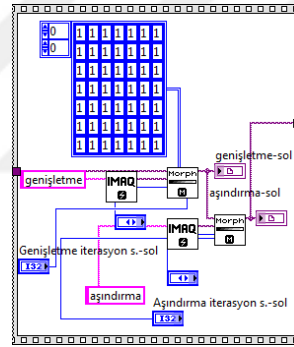
**Tablo 2.2** Morfolojik İşlemlerde Kullanılabilecek Operatör Türleri ve İşlevleri

İşlem Türü	İşlem Tanımları
<b>AutoM</b>	Otomatik Medyan
<b>Close</b>	Genişletme ve Aşındırma
<b>Dilate</b>	Genişletme (Aşındırmanın tersi)
<b>Erode</b>	Aşındırma (Genişletmenin tersi)
<b>Open</b>	Aşındırma ve Genişletme
<b>PClose</b>	7 ardışık Genişletme ve Aşındırma
<b>POpen</b>	7 ardışık Aşındırma ve Genişletme

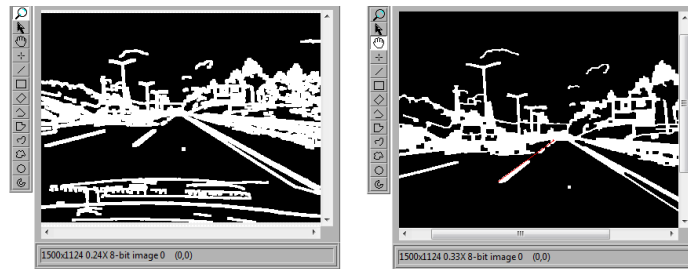
Bu işlem adımının başlangıcında Şekil 2.16’ da gösterilen genişletme (*dilate*) operatörü uygulanmıştır. Kopuklukların giderilmesinin ardından iyileştirilen görüntüde, sonraki aşamada aşındırma (*erode*) operatörü eklenerek genişleyen şerit çizgilerinin inceltilmesi yapılmıştır. Ayrıca, bu sistem için gerçek yol görüntüleri veri setlerinin kullanıldığı deneylerde 7x7 lik yapıtaşı elemanı, diğer görüntülerin yer aldığı deneylerde ise 3x3 lük bir yapıtaşı elemanı kullanılmıştır.



a)



b)



c)

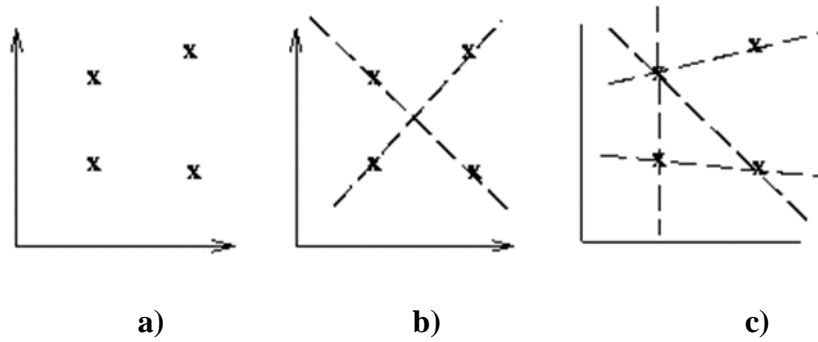
**Şekil 2.16** Morfolojik işlemler a) Genişletme ve aşındırma operatörleri uygulanmış örnek görüntüler b) Blok diyagram c) Genişletme ve aşındırma operatörlerinin bu sistemdeki çıkış görüntüleri

### 2.2.2.5. Şerit çizgisi algılama algoritmaları

Literatürde, çizgi tespiti için hangi noktaların çizgiye ait olduğu bilindiği uygulamalarda en küçük kareler [31], aykırı noktaların bulunduğu görüntülerde gürbüz eşleme ve RANSAC [16-20], ve çok fazla çizginin yer aldığı görüntülerde RANSAC ve Hough dönüşümü [20-32] algoritmalarının uygulandığı görülmüştür. Bu çalışmada, şerit çizgisi tespitinde düz çizgi algılamadaki başarısından dolayı Hough dönüşümü tercih edilmiştir.

#### Hough Dönüşümü (*Hough Transform*)

Görüntü içerisindeki geometrik şekillerin ortaya çıkarılmasında kullanılır. Özellikle çizgi, daire ve elips gibi şekillerde oldukça başarılı çalışmaktadır. Klasik ve geliştirilmiş Hough dönüşüm yöntemleri mevcuttur. Genelleşmiş Hough dönüşümü, klasik yöntemle göre daha karmaşık hesaplamalar barındırmaktadır. İşlem basamaklarında öncelikle Canny, Roberts gibi kenar algılama operatörlerinden uygun olanı tercih edilir. Fakat buradan elde edilen görüntüler gürültü barındırılabilir. Ardından doğrunun analitik ifade edilmesinde kullanılan denklemlerden yararlanılır. Şekil 2.17’ de çizgi parçasına ait kenar noktaları ve bu noktalardan oluşabilecek muhtemel doğrular gösterilmektedir.

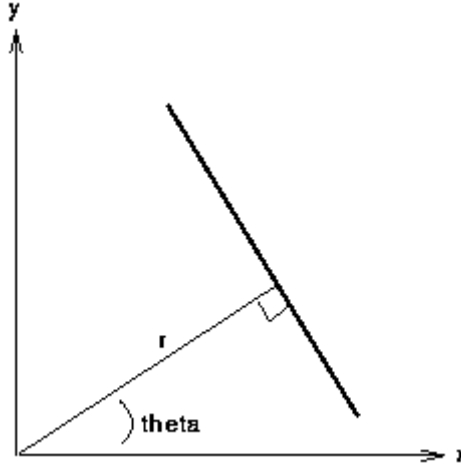


**Şekil 2.17** Hough Dönüşümü a) Elde edilen koordinat noktaları b) ve c) Muhtemel doğrusal bağlantı parçaları

Doğru analitik olarak farklı şekillerde tanımlanabilmektedir. Şekil 2.17’ deki gibi düzgün bir eğrinin tanımlanmasında aşağıdaki eşitlik kullanabilir. Eşitlik 2.6’ da “ $r$ ” çizgiye ait orijinden uzaklık mesafesini, “ $\theta$ ” ise bu çizginin eğim açısını belirtmektedir.

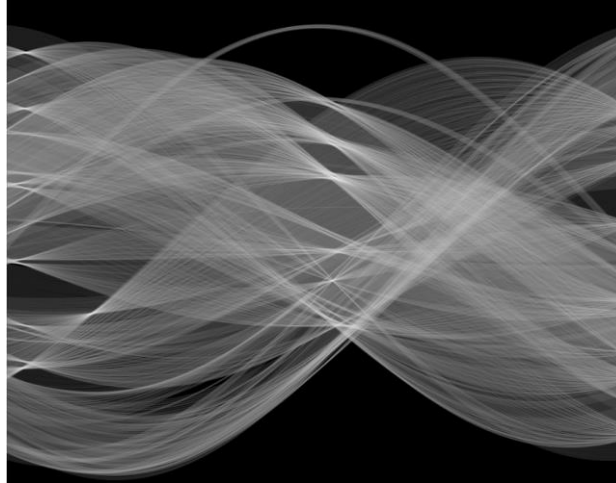


$$x\cos\theta + y\sin\theta = r \quad (2.6)$$



**Şekil 2.18** Düz bir eğrinin parametrik tanımlanması

Burada  $r$  ve  $\theta$  aranan değerlerdir. Denkleme uygun olarak olası  $r$  ve  $\theta$  değerleri ile bir grafik çizdirildiğinde sinüzoidal eğrilerden oluşan Hough yüzeyi (*Hough Space*) elde edilir. Hough yüzeyinde kesişen noktaların varlığı, görüntüde bir çizginin varlığını işaret etmektedir. Bu noktalardan çizgiye dönüşüm işlemi, Hough dönüşümü olarak ifade edilmektedir. Dönüşüm, Şekil 2.19’ da örneği verilen Hough yüzeyi incelenerek gerçekleştirilir. İnceleme sonrası, bir akumulator dizisi (*accumulator array*) elde edilir. Bu dizide oylama (*voting*) yapılır. En fazla oy, çizgi koordinatlarını göstermektedir [33].



**Şekil 2.19** Hough yüzeyi

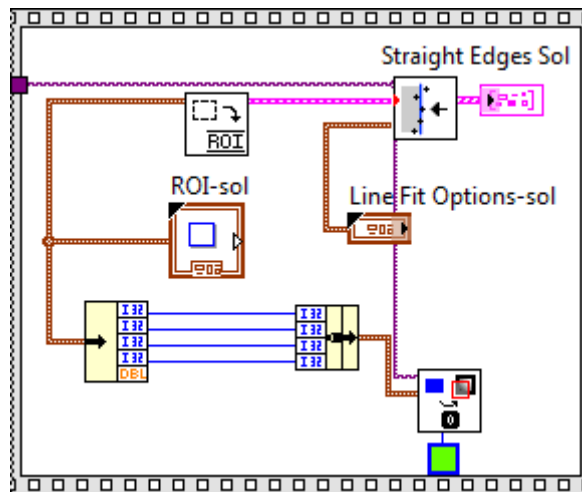
Diğer bir doğru denklemi olarak ise Eşitlik 2.7 kullanılır.

$$y = mx + b \tag{2.7}$$

Fakat burada  $m$  sonsuz değerler gerektirdiğinden polar forma geçilmesi tercih edilir. İlave olarak, dairesel şekil tespitinde Eşitlik 2.8’ de yer alan denklem kullanılır.

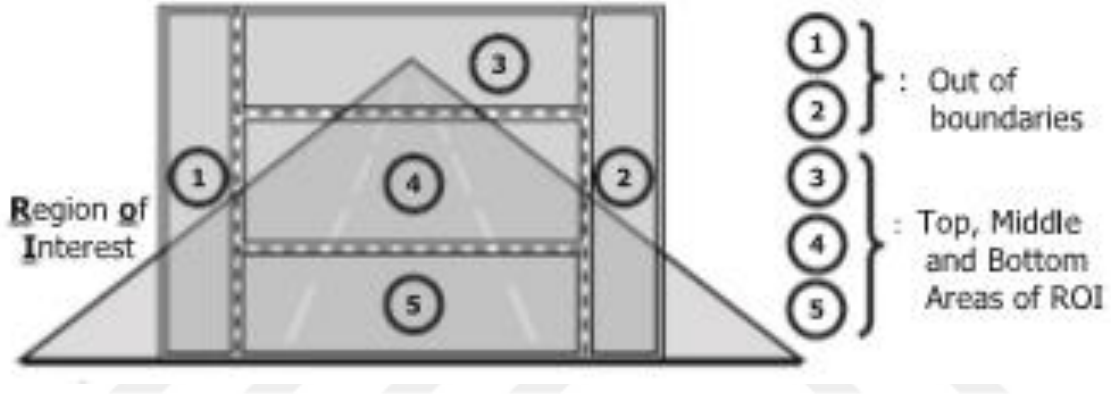
$$r^2 = (x - a)^2 + (y - b)^2 \quad (2.8)$$

Bu tez çalışmasında Hough dönüşümü uygulanması için, Şekil 2.20’ de blok diyagramı görülen, görüntüdeki düz çizgilerin bulunmasını sağlayan *IMAQ Find Edge* nesnesi kullanılmıştır. Bu nesne, U8, U16, RGB U32, I16, RGB U64, SGL ve HSL U32 görüntü türlerinde işlem yapabilmektedir. Kenarları arama yönünü (sağdan sola, soldan sağa, yukarıdan aşağıya veya aşağıdan yukarıya), gradyanlarını (yükselen kenar, düşen kenar ya da her iki yönde) hesaplama ve *First Edge*, *Best Edge*, *Hough Edge Rake* (ilk bulunan, eşleşme skoru en yüksek ya da Hough dönüşümünde bulunan kenar) metotlarından istenileni seçebilme olanağı sunmaktadır. Belirlenen seçimlerin sonuçlarına göre, ön panelde algılanan kenarlara ait, başlangıç- bitiş koordinat bilgileri ve eğim açısı değerleri, kullanıcı bilgisine sunulmaktadır.



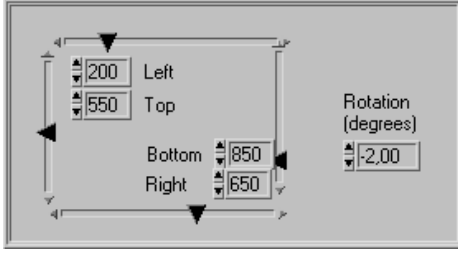
**Şekil 2.20** Şerit çizgisi tespiti blok diyagramı

Hough dönüşümünde çok sayıda yapılan hesaplamalar işlem süresini arttırmaktadır. Hesaplama yükünün azaltılması ve işlem süresinin kısaltılması amacıyla, bir ROI belirlenmiş; böylelikle işlemler tüm görüntüne matrisine değil de sadece ilgi alanına uygulanmıştır. Literatürde, ufuk noktası tabanlı (*vanishing point based*) ve alan tabanlı (*region based*) olmak üzere 2 temel seçim bulunmaktadır [34]. Ayrıca alan tabanlı yöntem, algılama ve izleme modları olmak üzere kendi içinde 2 alt tip daha içermektedir. Bu çalışmada ROI seçimi, Şekil 2.21’de de görüldüğü gibi alan tabanlı-algılama modu yöntemine göre belirlenmiştir. Görselliğin güçlendirilmesi amacıyla belirlenen alan üzerinde vurgulama yapılmıştır.

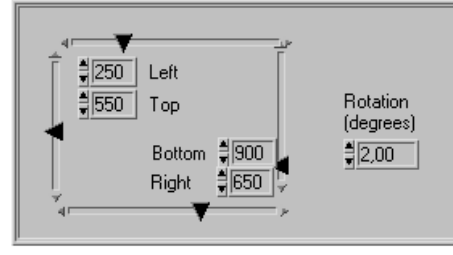


**Şekil 2.21** Alan-tabanlı yaklaşım ile ROI seçimi [33]

Sistemde, ön panel üzerinden “*Hough Edge Rake*” metodunun seçimi yapıldıktan sonra, bu seçeneğin parametreleri veri setine göre düzenlenmektedir. Gerçek yol görüntüleri için açı derecesi aralığı parametresi  $360^\circ$  iken; yapay yol görüntülerinde  $90^\circ$  dir. Bu farklılığın nedeni gerçek yol görüntülerinin geniş açılı kamera ile elde edilmiş olmasıdır. Ardından veri seti türüne göre farklılık gösteren dikdörtgen ROI seçimi uygulanarak, kapsadığı alanlarda sol ve sağ bölge için tarama işlemi yapılmakta; şerit çizgisi aday bölgeleri tespit edilmektedir. Şekil 2.22’ de mobil cihaz 1 kamerasından elde edilen gerçek yol görüntülerinde kullanılan değerler gösterilmiştir. Bu değerler, Şekil 2.23’te görülen araç görüş alanı temel alınarak, belirlenmiştir.

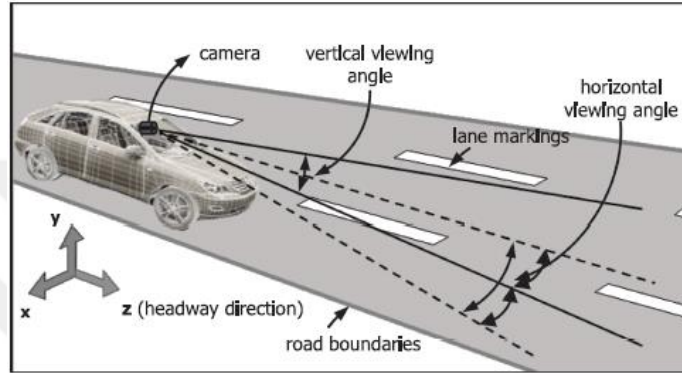


a)



b)

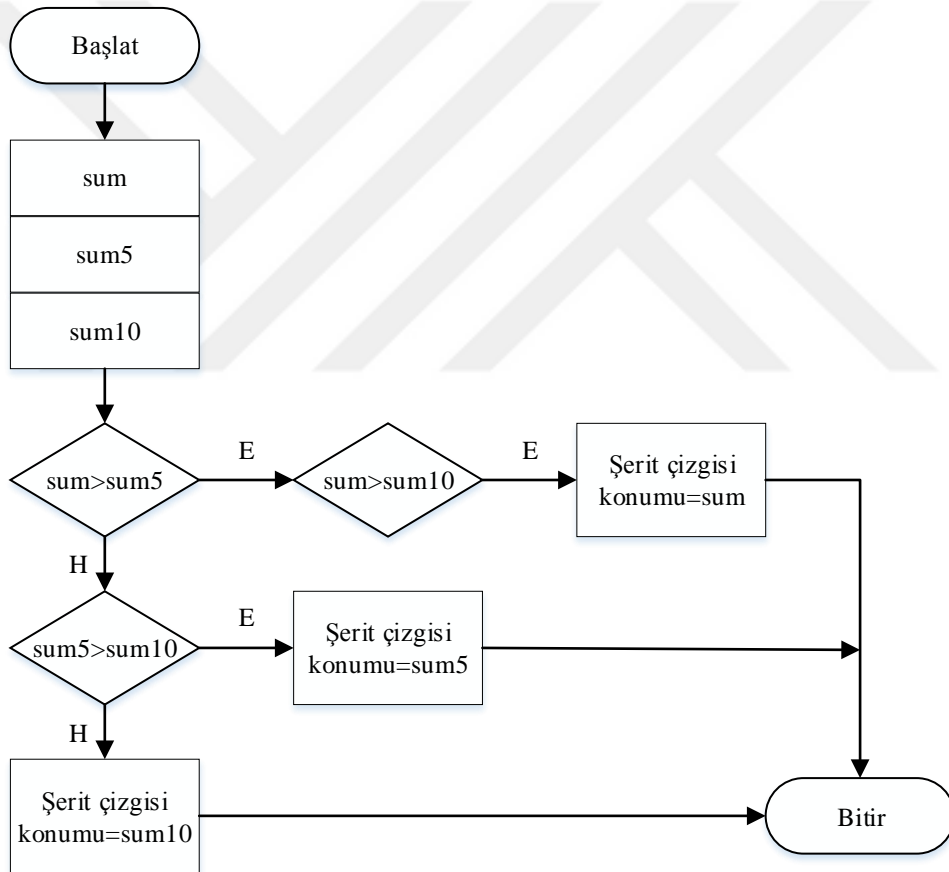
**Şekil 2.22** ROI bölgesi sınırlarının belirlenmesi a) Sağ bölüm b) Sol bölüm



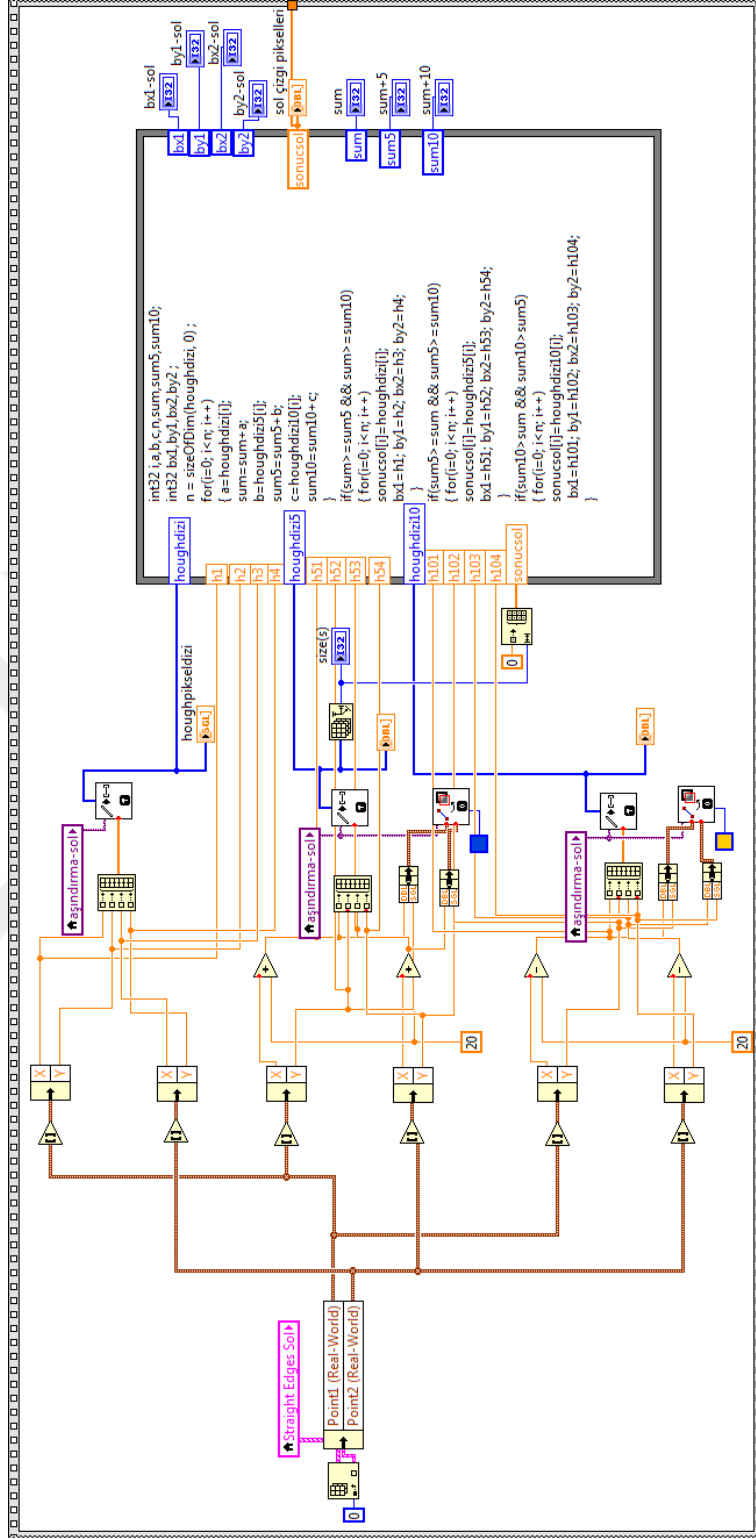
**Şekil 2.23** Aracın görüş alanı [33]

Hough dönüşümünün ardından elde edilen çizgiler genellikle doğru sonuç vermesine karşın hatalı sonuçlar da üretebilmektedirler. Yapılan deneylerde Hough dönüşümü sonucunda tespit edilen çizgilerin konumu, kenar belirleme algoritmasının sınırlara duyarlılığının yüksek olması sebebi ile çoğunlukla sınır bölgelerinde olmaktadır. Sınır bölgesinde algılama, örneğin şeritten ayrılma uyarı sistemi gibi sadece çizgi tespiti gerektiren uygulamalar için bir problem üretmemekte; ancak çizgi sınıflandırma konusunda yanlış sonuçlara sebep olabilmektedir. Bunu engellemek ve şerit çizgisi genişliğinin orta noktasına doğru ilerleyebilmek için, algılanan konumun her iki yönünde belli bir piksel adım aralığı (istenildiği durumda ön panelden değer girilebilir) ile tarama gerçekleştirilmiştir. Tarama sonucunda, geçerli algılanan çizgi konumu piksel yoğunluğu ile geçerli algılanan çizgi konumuna paralel ve her iki yönden (sağ ve sol yöndeki) elde edilen çizgi konumlarına ait piksel yoğunlukları karşılaştırılarak en büyük değere sahip olan konum, geçerli konum olarak belirlenir. Bu işleme ait akış şeması Şekil 2.24' te gösterilmiştir. Burada sum, Hough dönüşümünden tespit edilen şerit çizgisi konumuna ait 1B dizi içindeki tüm piksel değerlerinin toplamıdır. Sum5 ise,

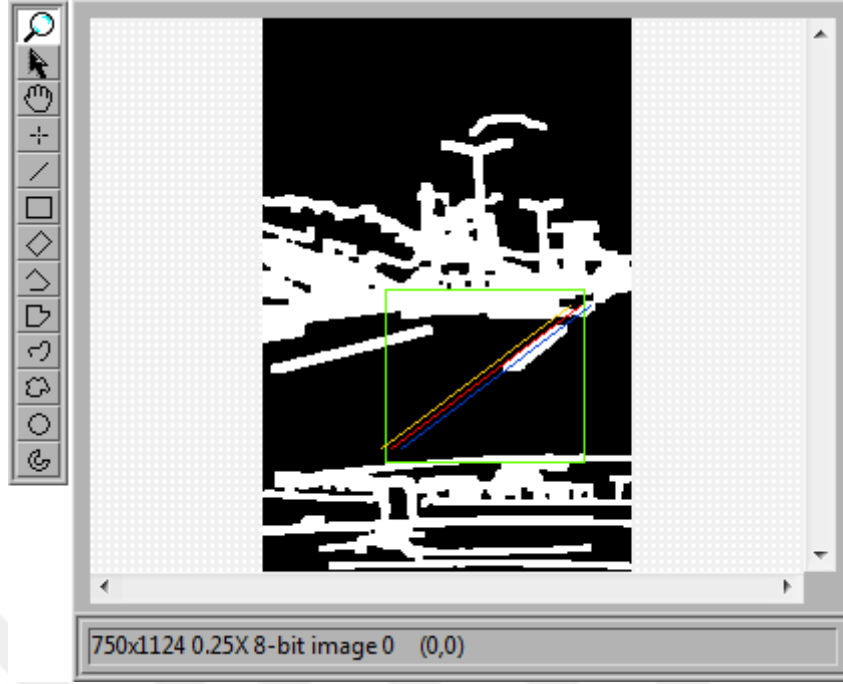
Hough dönüşümünden tespit edilen şerit çizgisinin belirlenen başlangıç noktasına göre x düzlemine göre 20 adım sağ tarafa ilerleyerek elde edilen ikinci konuma ait piksel değerlerinin toplamını göstermektedir. Sum10 ise, 20 adım sol yönde ilerleyerek elde edilen üçüncü konuma ait piksel değerlerinin toplamını göstermektedir. Daha sonra bu değerler Şekil 2.24' teki gibi karşılaştırılarak, en büyük piksel değerine sahip konum, geçerli konum olarak tespit edilmektedir. Şekil 2.25' te Hough dönüşümünde elde edilen ilk konum ve diğer konumların belirlenmesi ve değerlendirme işlemlerini gösteren blok diyagram görüntüsü verilmiştir. Şekil 2.26' da ise Hough dönüşümü sonucunda elde edilen aday çizgi konumu ile iyileştirme amaçlı oluşturulan ikinci ve üçüncü aday konumları gösteren çıkış görüntülerine yer verilmiştir.



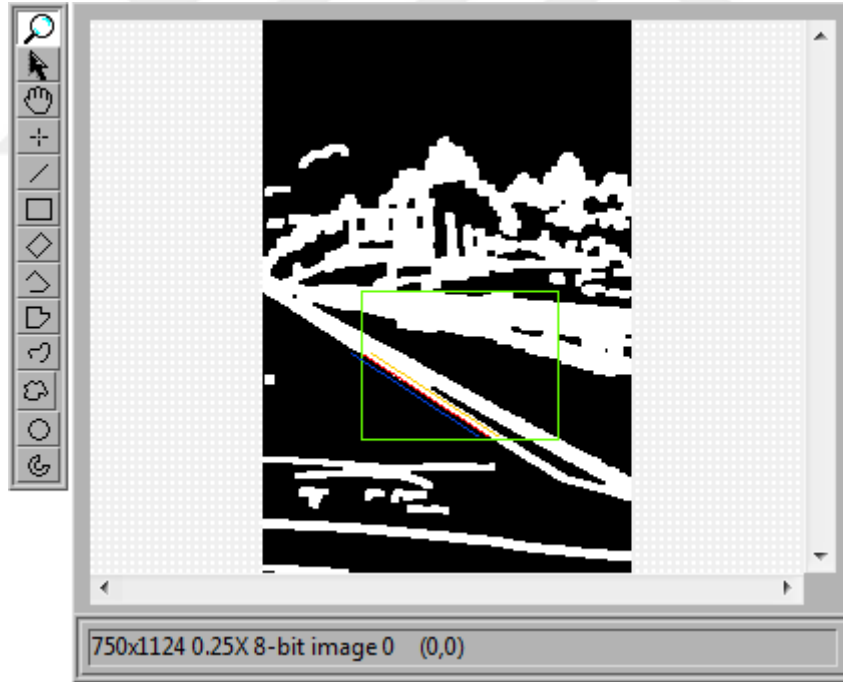
**Şekil 2.24** Şerit çizgisi konumu tespitinin iyileştirilmesine ait akış şeması



Şekil 2.25 Algılanan şerit çizgisinin doğru konumunun elde edilmesi için eklenen iyileştirmelere yönelik blok diyagram



a)



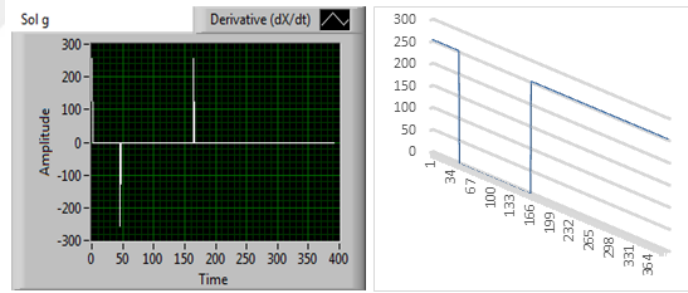
b)

**Şekil 2.26** Hough dönüşümü çıkış görüntüleri a) Sol şerit çizgisinin tespiti b) Sağ şerit çizgisinin tespiti

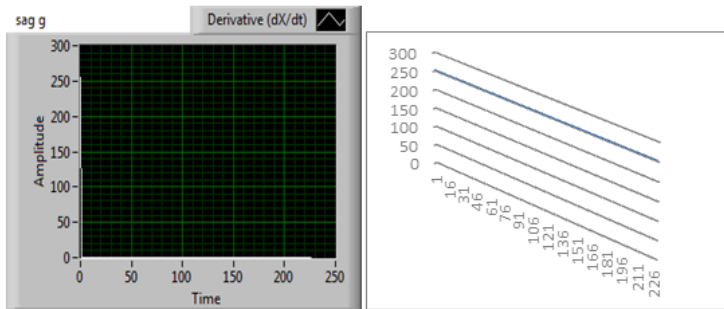
### 2.2.2.6.Sınıflandırma

Sınıflandırma, görüntü işleme uygulamalarında ilgi alanına giren piksel değerlerine göre bir durum hakkında amaca göre karar verebilme ve mevcut görüntünün o duruma ait önceden tanımlanmış kategorilerden hangisine ait olduğunu belirleme sürecidir [35]. Literatürde karar ağacı sınıflandırıcı, kural tabanlı sınıflandırıcı, Bayesian sınıflandırıcı, Destek Vektör Makineleri (DVM), Yapay Sinir Ağlarıyla (YSA) sınıflandırma gibi çok sayıda teknik bulunmaktadır. Kural tabanlı sınıflandırıcı, “if...then...” yapısına göre oluşturulan sınıflandırma türüdür [36].

Sınıflandırma işlemi bu çalışmada kural tabanlı olarak gerçekleştirilmiştir. Oluşturulan kuralın blok diyagrama uygulanmasında, şerit çizgisinin algılanması adımının ardından elde edilen koordinat bilgileri ile bu koordinatlardaki ışık şiddeti (f) seviyelerinin birinci dereceden türevleri alındığında Şekil 2.27’ de grafiksel gösterimlerine yer verilen çizgi bölgelere ayrılarak bölge sayısı tespit edilmiştir. Ardından bu değer ile eşik seviyesinin karşılaştırma sonuçlarına göre, “*Elde edilen bölge sayısı 0’a EŞİT ise DÜZ; değil ise KESİK’ tir*” kuralı oluşturulmuştur.



a)



b)

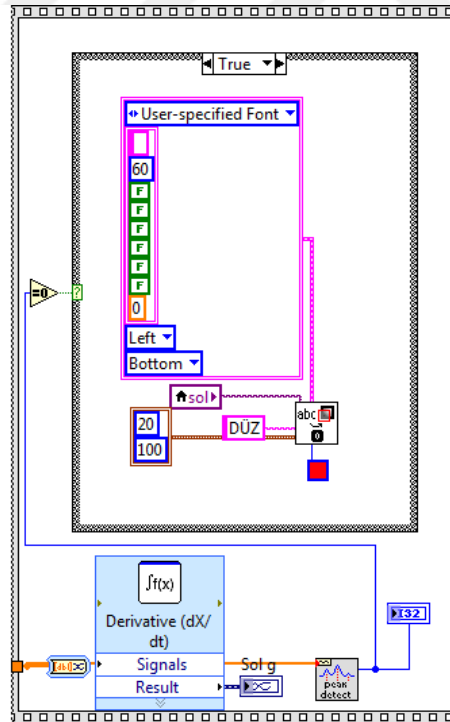
Şekil 2.26 Işık şiddeti türev grafikleri a) Kesik şerit çizgisi b) Düz şerit çizgisi



Değerlendirme sonucunda verilen karar, Şekil 2.28’ de gösterildiği gibi DÜZ veya KESİK olarak yazılı şekilde kullanıcı bilgisine sunulmaktadır. Sonucun ekrana bildirilmesinde *IMAQ Overlay Text* nesnesinden faydalanılmıştır. Bu basamağa ait blok diyagram Şekil 2.29’ da gösterilmiştir.



Şekil 2.27 Şerit çizgisi sınıflandırılması çıkış görüntüleri a) Sol şerit b) Sağ şerit

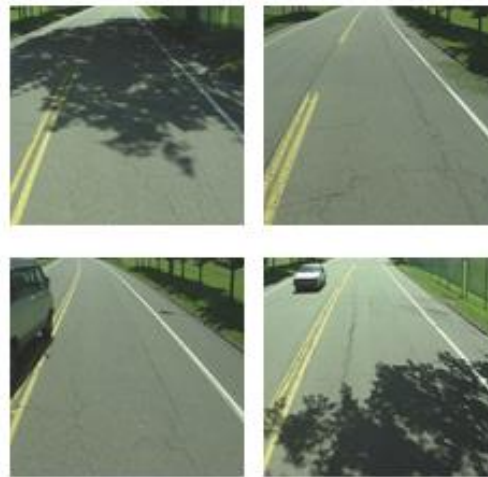


Şekil 2.28 Şerit çizgisi sınıflandırılması blok diyagramı

### 3. DENEYSEL SONUÇLAR

Bu bölümde, geliştirilen yapay görü teknikleri ile şerit çizgisi konum algılama ve sınıflandırma sisteminin, sentetik ve gerçek yol görüntüleri üzerinde yapılan deney sonuçları değerlendirilmiştir. Deneyler esnasında, Şekil 3.1’ de örnekleri sunulan farklı görüş açılarına ve parametrelere sahip kameralardan elde edilmiş, olumsuz yol koşullarını da barındıran görüntüler kullanılmıştır. Sırasıyla sentetik ve gerçek yol görüntülerine ait örnek deney sonuçları Şekil 3.2, Şekil 3.3, Şekil 3.4 ve Şekil 3.5’ te verilmiştir.

Sistemin geliştirilmesi aşamasında, veri setlerinde yer alan görüntülerin çok fazla sayıda ve farklı niteliklerde olumsuzluklar barındırması sebebiyle başlangıç aşamasında kenarların tespit edilmesinde zorluklar yaşanmıştır. Canny filtresinin, Sobel ve Prewitt filtrelerinden daha başarılı çalıştığı gözlemlenmiştir. Görüntüler farklı gürültüler barındırdığından, çıkış görüntülerinde temiz görüntünün elde edilmesi problemi, Canny filtresi yumuşatma katsayısı değiştirilerek çözümlenmiştir. Sonraki aşamada deneylerde kullanılan görüntülerde silinmiş şerit çizgilerinin bulunması sebebiyle şerit çizgisinin algılanmasında zorluk yaşanmıştır. Bu basamak da genişletme işleminin uygulanmasıyla çözümlenmiştir. Genişletme operatörü ile çizgilerdeki kopukluklar/boşluklar giderildikten sonra aşındırma işlemiyle kenarlar tekrar inceltilerek daha net görüntüler elde edilmiştir.

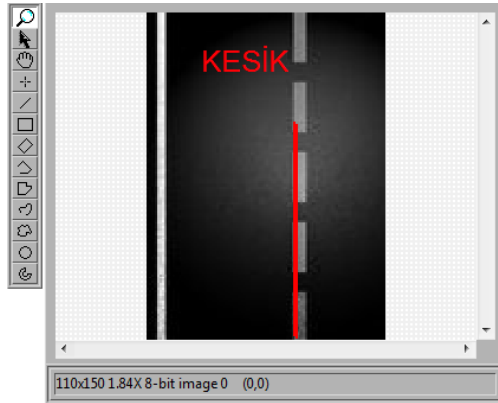


**Şekil 3.1** Olumsuz koşullara sahip yol görüntüsü örnekleri

Sol ve sađ řerit çizgilerinin her ikisinin de algılanması için bölütleme yapılarak; her iki bölge için paralel işlem süreci başlatılmıştır. Araç içinde sabit bir konuma takılmış bir kameradan alınan görüntüler yerine, bu çalışmada farklı görüş açılarından yakalanmış ve çözünürlükleri de farklı olan görüntüler üzerinde çalışılmıştır. Bu durum da, bazı zorluklar doğurmuştur. Bu yüzden, üzerinde çalışılacak ilgi alanı seçimi için (ROI) değer aralıkları değiştirilmiş; aynı zamanda Hough yüzeyinin tarama açı aralığı da farklılaştırılmıştır. Sınıflandırmada farklı görüş açılarından elde edilen görüntülerin bulunması, řerit çizgisi uzunluklarının ölçeklendirilememesi de kural oluşturmada yaşanan sıkıntılar arasında yer almıştır. Bu durum ise türev işlemi sonunda çizginin bölgelere ayrılmasıyla aşılmıştır.



a)



b)

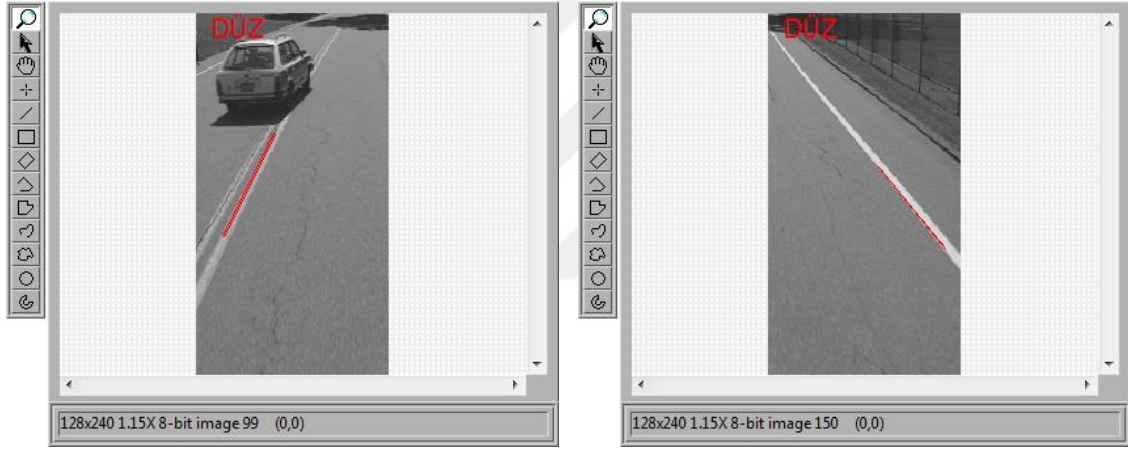


c)

**Şekil 3.2** Clipartfest veri seti ile bulunan deney sonuçları a) Sentetik yol görüntüsü b) Sol řeridin algılanmasına ait sonuç görüntüsü c) Sağ řeridin algılanmasına ait sonuç görüntüsü



a)



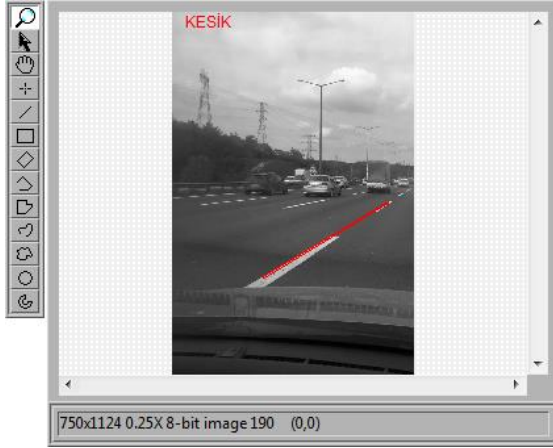
b)

c)

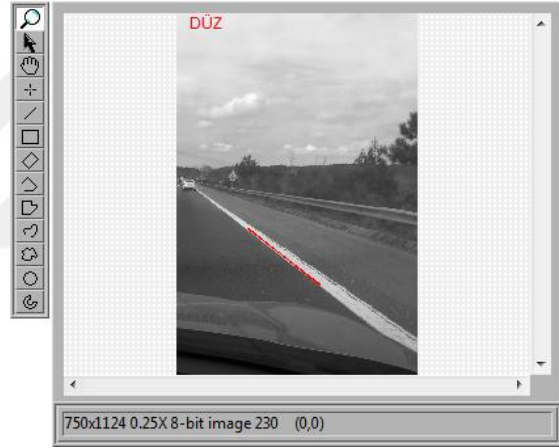
**Şekil 3.3** Carnegie Mellon Üniversitesi veri seti ile bulunan deney sonuçları **a)** Gerçek yol görüntüsü **b)** Sol şeridin algılanmasına ait sonuç görüntüsü **c)** Sağ şeridin algılanmasına ait sonuç görüntüsü



a)

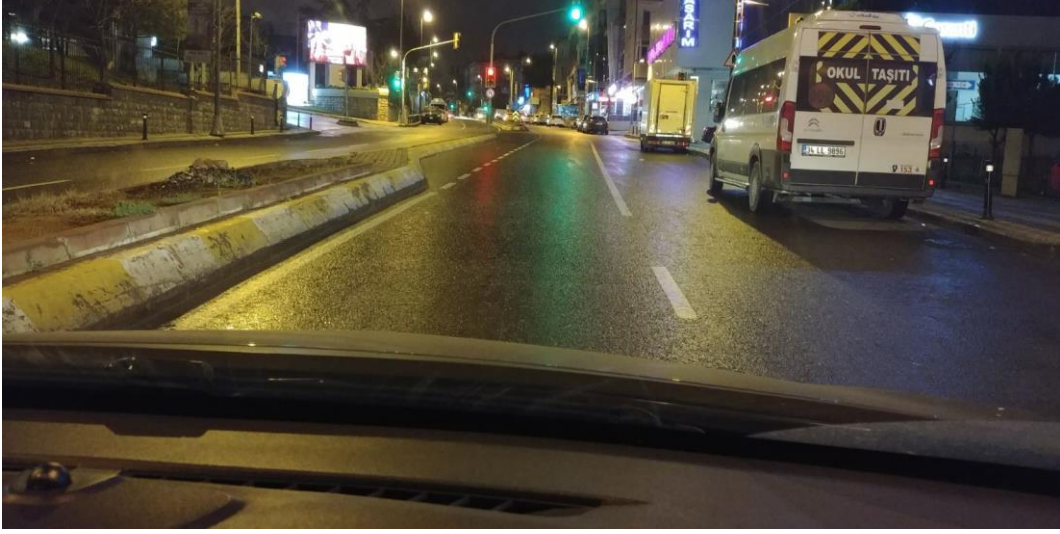


b)

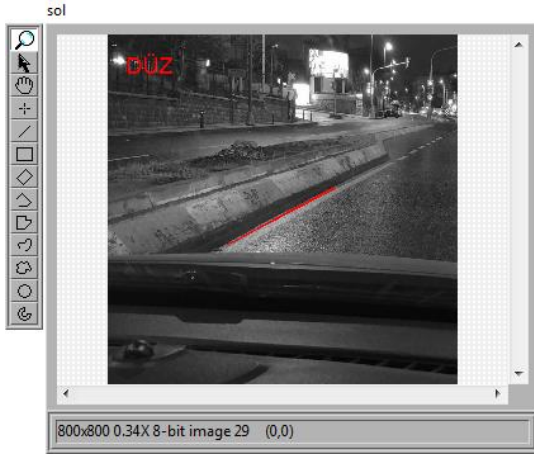


c)

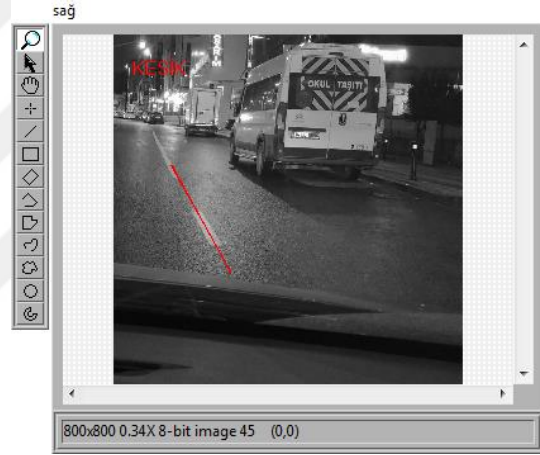
**Şekil 3.4** Mobil cihaz 1 veri seti ile bulunan deney sonuçları **a)** Gerçek yol görüntüsü **b)** Sol şeridin algılanmasına ait sonuç görüntüsü **c)** Sağ şeridin algılanmasına ait sonuç görüntüsü



a)



b)



c)

**Şekil 3.5** Mobil cihaz 2 veri seti ile bulunan deney sonuçları **a)** Gerçek yol görüntüsü **b)** Sol şeridin algılanmasına ait sonuç görüntüsü **c)** Sağ şeridin algılanmasına ait sonuç görüntüsü

#### 4. DEĞERLENDİRME VE ÖNERİLER

Bu çalışmada, olumsuz koşullardan bazılarını içeren yollarda şerit çizgisinin tespit edilmesi ve türünün kesik/düz olarak sınıflandırılması amaçlanmıştır. Bu amaç doğrultusunda 6 temel basamaktan oluşan bir sistem tasarımı gerçekleştirilmiştir. Şerit çizgisi algılama ve sınıflandırma uygulamalarına yönelik yapılan literatür taramasında (benzer çalışmalar bölümünde de belirtildiği gibi) sıkça MATLAB programlama dilinin kullanıldığı görülmüştür. Bu çalışma ise LabVIEW platformu ile tamamlanmıştır.

Geliştirilen sistemde, Clipartfest, Carnegie Mellon Üniversitesi Görüntü veri seti ve farklı mobil cihaz kameralarından elde edilen görüntüler üzerinde algılama ve sınıflandırma deneyleri yapılmıştır. Deneyler, bahsedilen veri setlerinden farklı olumsuz koşullarının bulunduğu rastgele seçilmiş toplam 30 görüntü üzerinde gerçekleştirilmiştir. Algılama ve sınıflandırma işlemlerinin başarı oranları, Tablo 4.1 ve Tablo 4.2’ de ayrı ayrı verilmiştir.

**Tablo 4.1** Şerit çizgisi konumu algılama başarı sonuçları

Veri Seti Türü	Örnek Sayısı	Doğru Algılama Sayısı	Yanlış Algılama Sayısı	Yanlış Algılama Sebebi
Clipartfest	5	5	---	---
Carnegie Mellon Üniversitesi	15	9	6	Yoğun gölge ve silinmiş çizgi durumları
Mobil Cihaz 1 Kamera	5	5	---	---
Mobil Cihaz 2 Kamera	5	4	1	Yağışlı yol durumları
Toplam	30	23	7	---
Yüzde (%)	---	77	23	---



**Tablo 4.2** Şerit çizgisi sınıflandırma başarı sonuçları

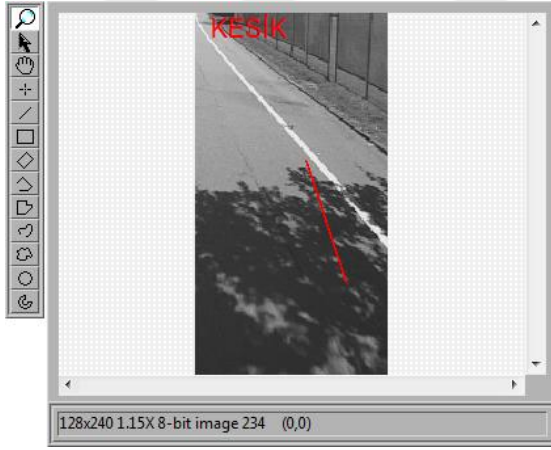
Veri Seti Türü	Örne k Sayıs ı	Doğru Sınıflandır ma Sayısı	Yanlış Sınıflandır ma Sayısı	Olması gereken Sınıflandır ma Sonucu	Yanlış Sınıflandır ma Sonucu
Clipartfest	5	5	---		---
Carnegie Mellon Üniversitesi	15	9	6	Düz	Kesik
Mobil Cihaz 1 Kamera	5	5	---		---
Mobil Cihaz 2 Kamera	5	4	1	Düz	Kesik
Toplam	30	23	7		---
Yüzde (%)	---	77	23		---

Bu tezde önerilen çözümün şerit çizgisi algılama ve sınıflandırma genel başarısı % 77 (3 farklı veri seti ele alındığında) olarak tespit edilmiştir. Ancak bu başarı yüzdesi, yalnızca Carnegie Mellon Üniversitesi veri seti ile gerçekleştirilen deneylerde % 65'e kadar düşmektedir. Benzer veri setinin kullanıldığı [22] nolu çalışmada başarı % 97,2 olarak belirtilmiştir. Bu çalışmada sadece farklı gün ışığı koşulları değerlendirmeye alınmış; gölgeli görüntüler ise dışarda tutulmuştur. Bu tezde önerilen çözümün başarısı % 65 gibi düşük bir oran olarak görülse de; çalışılan görüntüler içinde az ve yoğun gölgeli görüntülerin varlığı dikkate alındığında bu oranın çok da düşük sayılmayacağı söylenebilir. Şekil 4.1' deki örnekte de görüldüğü gibi, bu tez çalışmasında yapılan deneylerde tercih edilen görüntülerde yoğun gölge ve silinmiş şerit çizgileri yer almaktadır. Önerilen çözümde yanlış sınıflandırmalar zaten sadece az ve yoğun gölgeli görüntülerde oluşmuştur. Aradaki söz konusu başarı yüzdesi farklılığının nedeni, bu şekilde açıklanabilir. Gölgeli görüntüler için, seçilen filtre parametreleri yetersiz kalmaktadır. Çizgi konumunun yanlış tespit edilmesi sonucunda da doğal olarak sınıflandırma türü de, yanlış belirlenmektedir.

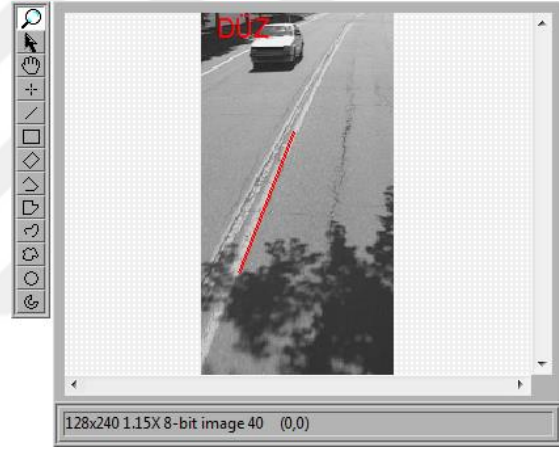




a)



b)

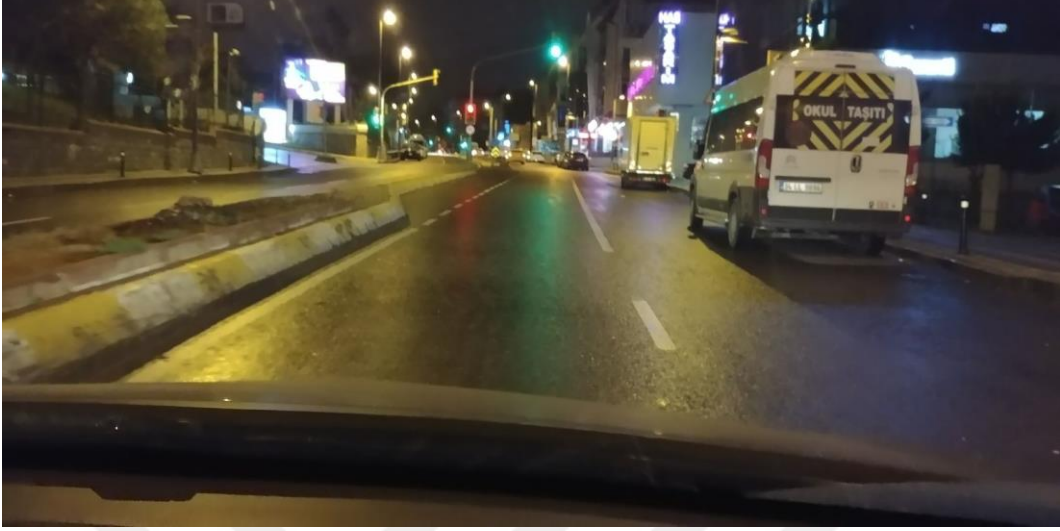


c)

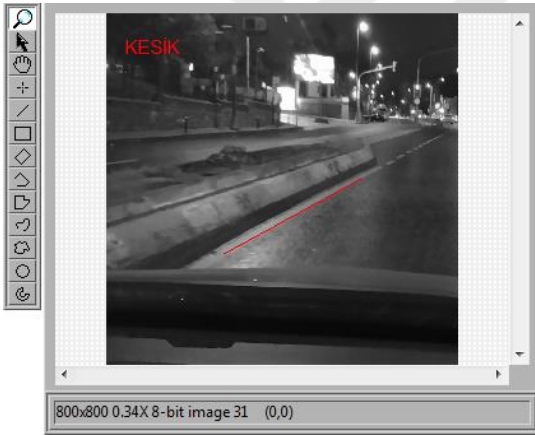
**Şekil 4.1** Carnegie Mellon Üniversitesi veri seti ile yanlış bulunan deney sonuçları **a)** Gerçek yol görüntüsü **b)** Sol şeridin algılanmasına ait sonuç görüntüsü **c)** Sağ şeridin algılanmasına ait sonuç görüntüsü

Ayrıca, başka bir yanlış sınıflandırma sonucu da Şekil 4.2’ de de görüldüğü gibi mobil cihaz 2 veri setinde yapılan deneylerde oluşmuştur. İşlenen görüntü, gece görüş ve yağışlı hava olumsuz koşullarında elde edilmiş olmasından dolayı sınır (uç noktalar) bölgelerinde ilerlenen adım aralığı yetersiz kalmıştır. Ayrıca sınıflandırma adımı

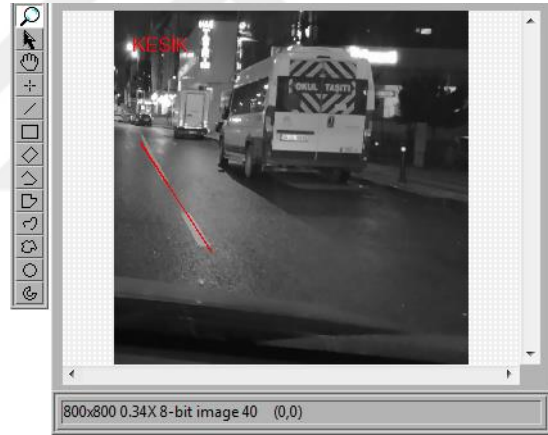
hazırlanan algoritmanın basit yapıda tasarlanmış olması da sınıflandırma sonucunun hatalı olarak belirlenmesine yol açmaktadır.



a)



b)



c)

**Şekil 4.2** Mobil cihaz 2 veri seti ile yanlış bulunan deney sonuçları a) Gerçek yol görüntüsü b) Sol şeridin algılanmasına ait sonuç görüntüsü c) Sağ şeridin algılanmasına ait sonuç görüntüsü

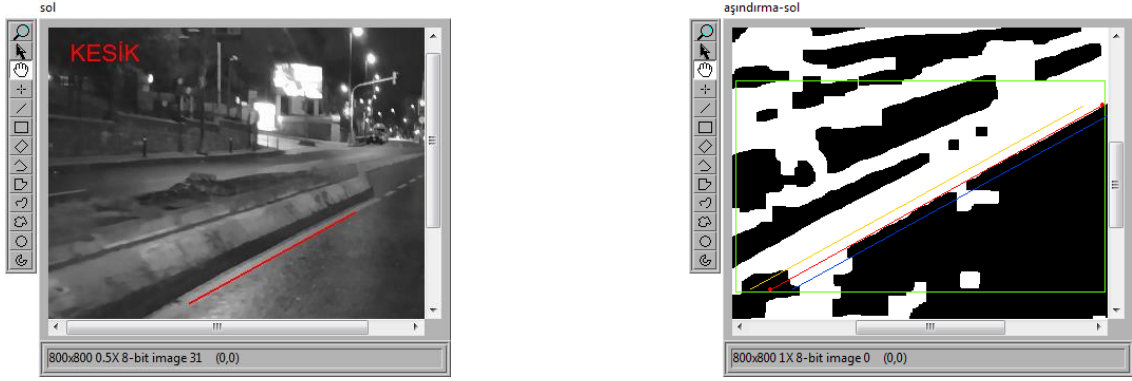
Şerit çizgisi algılama başarısının artırılması, test edilen görüntülerde şerit çizgisi konumunun ne kadar doğru algılandığına bağlıdır. Konumun doğru olarak algılanması ise, Hough parametrelerinin ve ilgi alanı seçiminin doğru yapılmasına bağlıdır. İlgi alanı seçimi, doğrudan görüntülerin elde edildiği kameranın araçtaki yerleşimine ve görüş açısına bağlıdır. Bu çalışmada farklı konumlara yerleştirilmiş ve farklı görüş açılarına sahip kameralardan elde edilmiş görüntüler üzerinde çalışılmıştır. Dolayısı ile ROI seçiminde sabit bir seçim yapılamamış; görüntü grupları arasında bir miktar değişiklik yapılarak sisteme uyarlanmıştır.

Ayrıca Hough dönüşüm algoritmasının başarısı da, aday noktaların koordinatlarının ne kadar doğrulukla elde edilmiş olduğuna bağlıdır. Bu da uygulamada tercih edilen kenar belirleme yöntemine ve görüntü üzerinde var olan bozucu etmenlere bağlıdır.

Hough dönüşüm çıktıları ise, 3 parametre ile belirlenmektedir: Kenarları arama yönü (sağdan sola, soldan sağa, yukarıdan aşağıya veya aşağıdan yukarıya), gradyan hesaplama (yükselen kenar, düşen kenar ya da her iki yönde) ve First Edge, Best Edge, Hough Edge Rake (ilk bulunan, eşleşme skoru en yüksek ya da Hough dönüşümünde bulunan kenar).

Şekil 4.3' te mobil cihaz 2 veri setinde yanlış sınıflandırma durumuna ait detaylı görüntüye yer verilmiştir. Bu duruma çözüm olarak, hazırlanan sınıflandırma algoritmasında, eğer algılanan bölgeler uç noktalarda ise bu durumda bölge mesafelerinin belirli bir eşik değerini geçmesi ile değerlendirmeye katılması; değil ise dikkate alınmaması önerilebilir.

Diğer taraftan yine silinmiş/eskimiş çizgiler, belli bir oranda araçlar tarafından örtülmüş şerit çizgileri veya gölgeli yol görüntüleri üzerinde çalışılması gereken noktalardır.



**Şekil 4.3** Mobil cihaz 2 veri seti ile yanlış tespit edilen duruma ait detaylı görüntü

Gelecek çalışma olarak, bu sistemde yoğun gölge koşullarının, görüntü daha ilk aşamada ikili görüntüye dönüştürüldükten sonra morfolojik işlem yapıtaşlarının saptanan çizgi eğimleri doğrultusunda değiştirilmesiyle yok edilmesi ya da en az düzeye indirilmesi hedeflenmektedir.

Ayrıca, basit kural tabanı ile oluşturulmuş bu çözüm önerisine ek olarak Bayes sınıflandırıcı veya YSA' da eklenerek başarı yüzdesinin artırılması planlanmaktadır. Bu yöntemler arasında karşılaştırmalar yapılarak; tespit ve sınıflandırmada hangi yöntemin daha yüksek başarı vereceği belirlenebilir.

Gerçekleştirilen sistem gerçek zamanlı olarak tasarlanmamıştır. Gömülü sistem kullanarak sistemin gerçek zamanlı olarak kullanılması sağlanabilir. Bunun için, hazırlanan sistemde kullanılan nesnelere ilave olarak görüntü yardımcılar paletinin “Image Transfer” nesnelerinin kullanılmasının yeterli olacağı düşünülmektedir.

## KAYNAKLAR

- [1] [http://www.ti.com/solution/automotive\\_vision\\_control](http://www.ti.com/solution/automotive_vision_control) (Eriřim Tarihi 25.11.2017)
- [2] Fasih Alireza, Schwarzlmueller Christopher, Kyamakya Kyandoghere, Machot Fadi Al “Video Enhancement for ADAS Systems based on FPGA and CNN Platform”, International Journal of Signal & Image Processing;2010, Vol. 1 Issue 3, p169 – (Mayıs 2010)
- [3] <http://toshiba.semicon-storage.com/ap-en/application/automotive/safetyassist/image-recognition.html> (Eriřim Tarihi 02.10.2015)
- [4] <http://binekarac.vw.com.tr/vwSozluk/ShowComponent.aspx?ComponentID=15172> (Eriřim Tarihi 22.08.2015)
- [5] <http://www.tuik.gov.tr/PreHaberBultenleri.do?id=18760> (Eriřim Tarihi 07.09.2015)
- [6] İslim Sungur, Recep Akdur, Birgöl Piyal “Türkiye’deki Trafik Kazalarının Analizi”, Ankara Üniversitesi Halk Sağlığı AD – (2014)
- [7] <http://binekarac.vw.com.tr/volkswagen-sozluk.aspx?ComponentID=17752> (Eriřim Tarihi 22.08.2015)
- [8] Ge, P. S., Guo, L., Xu, G. K., Zhang, R. H., & Zhang, T. (2012). A Real-Time Lane Detection Algorithm Based on Intelligent CCD Parameters Regulation. Discrete Dynamics in Nature and Society, 2012.
- [9] <http://www.ni.com/pdf/manuals/322916b.pdf> [Eriřim Tarihi 18.09.2015]
- [10] Tran, T. T., Cho, H. M., & Cho, S. B. (2011). A robust method for detecting lane boundary in challenging scenes. Information Technology Journal, 10(12), 2300-2307.
- [11] Li, Q., Chen, L., Li, M., Shaw, S. L., & Nuchter, A. (2014). A sensor-fusion drivable-region and lane-detection system for autonomous vehicle navigation in challenging road scenarios. IEEE Transactions on Vehicular Technology, 63(2), 540-555.

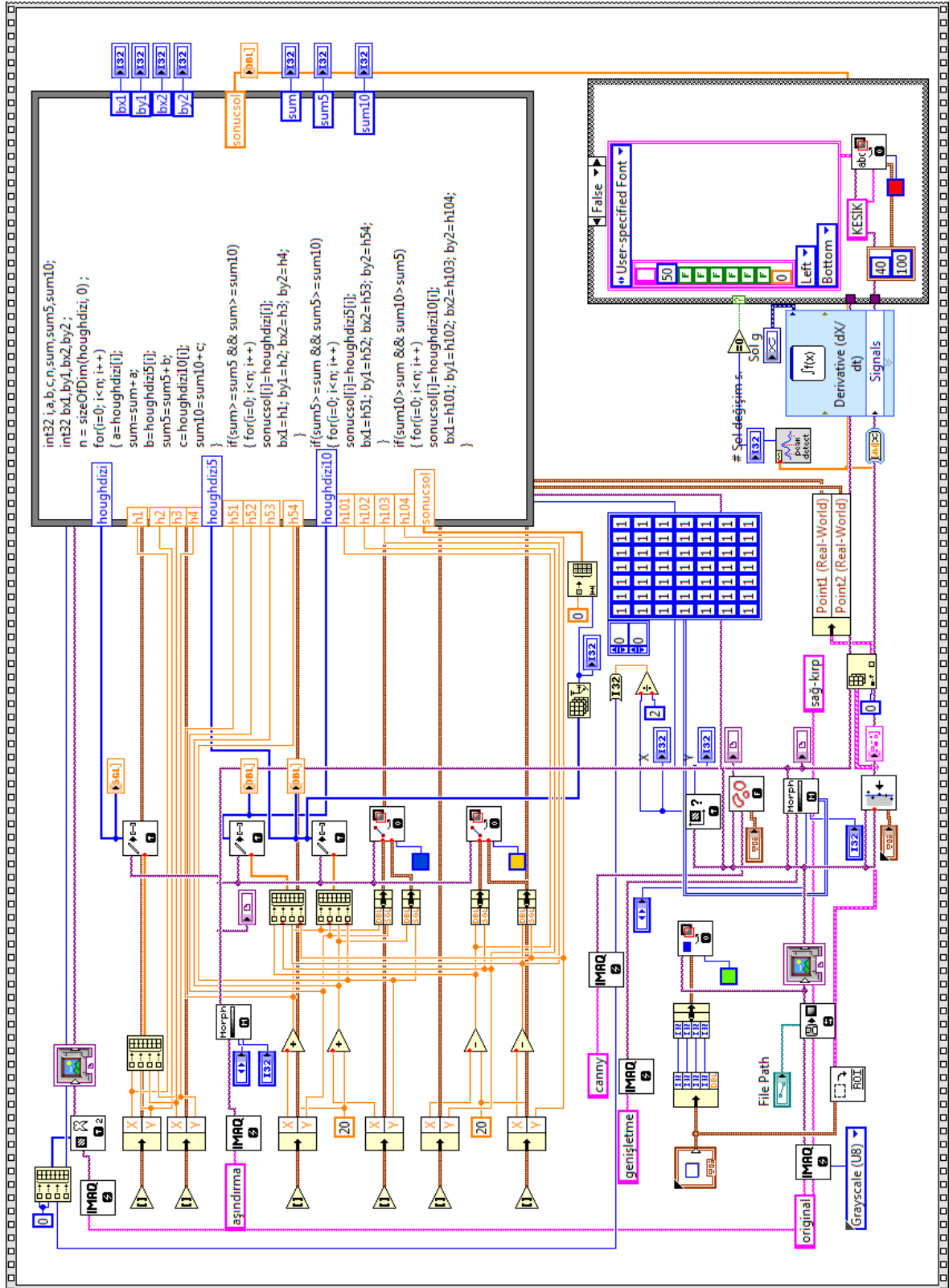
- [12] de Paula, M. B., & Jung, C. R. (2015). Automatic detection and classification of road lane markings using onboard vehicular cameras. *IEEE Transactions on Intelligent Transportation Systems*, 16(6), 3160-3169.
- [13] Daigavane, P. M., & Bajaj, P. R. (2010, November). Road lane detection with improved canny edges using ant colony optimization. In *Emerging Trends in Engineering and Technology (ICETET)*, 2010 3rd International Conference on (pp. 76-80). IEEE.
- [14] Parajuli, A., Celenk, M., & Riley, H. B. (2013). Robust lane detection in shadows and low illumination conditions using local gradient features. *Open Journal of Applied Sciences*, 3(01), 68.
- [15] Yoo, H., Yang, U., & Sohn, K. (2013). Gradient-enhancing conversion for illumination-robust lane detection. *IEEE Transactions on Intelligent Transportation Systems*, 14(3), 1083-1094.
- [16] Gopalan, R., Hong, T., Shneier, M., & Chellappa, R. (2012). A learning approach towards detection and tracking of lane markings. *Intelligent Transportation Systems, IEEE Transactions on*, 13(3), 1088-1098.
- [17] Gulve, R., & Diggikar, A. (2014). LDWS For Smart Vehicles Using Image Processing Based On Hough Transform. *International Journal of Advanced Research in Computer and Communication Engineering*, 3(10).
- [18] Vacek, S., Schimmel, C., & Dillmann, R. (2007). Road-marking Analysis for Autonomous Vehicle Guidance. In *EMCR*.
- [19] Collado, J. M., Hilario, C., De La Escalera, A., & Armingol, J. M. (2006, January). Adaptive road lanes detection and classification. In *Advanced Concepts for Intelligent Vision Systems* (pp. 1151-1162). Springer Berlin Heidelberg.
- [20] Aly, M. (2008, June). Real time detection of lane markers in urban streets. In *Intelligent Vehicles Symposium, 2008 IEEE* (pp. 7-12). IEEE.

- [21] Shin, W. S., Song, D. H., & Lee, C. H. (2006). Vehicle Classification by Road Lane Detection and Model Fitting Using a Surveillance Camera. JIPS, 2(1), 52-57.
- [22] Javadi, M. S., Hannan, M. A., Samad, S. A., & Hussain, A. (2012). A robust vision-based lane boundaries detection approach for intelligent vehicles. Information Technology Journal, 11(9), 1184-1192.
- [23] Zhu, W., Liu, F., Li, Z., Wang, X., & Zhang, S. (2008, December). A vision based lane detection and tracking algorithm in automatic drive. In Computational Intelligence and Industrial Application, 2008. PACIA'08. Pacific-Asia Workshop on (Vol. 1, pp. 799-803). IEEE.
- [24] Saha, A., Roy, D. D., Alam, T., & Deb, K. (2012). Automated Road Lane Detection for Intelligent Vehicles. Global Journal of Computer Science and Technology, 12(6).
- [25] <https://www.youtube.com/watch?v=wMGTyJjk2oM> (Erişim Tarihi 15.12.2017)
- [26] <https://clipartfest.com/download/56ab3a930a0842eff946bc37d23b9811f9c219b3.html> (Erişim Tarihi 24.12.2016)
- [27] [http://vasc.ri.cmu.edu/idb/html/road/may30\\_90/](http://vasc.ri.cmu.edu/idb/html/road/may30_90/) (Erişim Tarihi 23.09.2015)
- [28] Jeffrey Travis ve Jim Kring, “LabVIEW for Everyone: Graphical Programming Made Easy and Fun”, Prentice Hall, 2006.
- [29] [https://www.wikizero.com/en/Canny\\_edge\\_detector](https://www.wikizero.com/en/Canny_edge_detector) (Erişim Tarihi 01.10.2017)
- [30] Rong, W., Li, Z., Zhang, W., & Sun, L. (2014, August). An improved CANNY edge detection algorithm. In Mechatronics and Automation (ICMA), 2014 IEEE International Conference on (pp. 577-582). IEEE.
- [31] Sun, P., & Chen, H. (2014, November). Lane detection and tracking based on improved Hough transform and least-squares method. In International Symposium on Optoelectronic Technology and Application 2014 (pp. 93011U-93011U). International Society for Optics and Photonics.

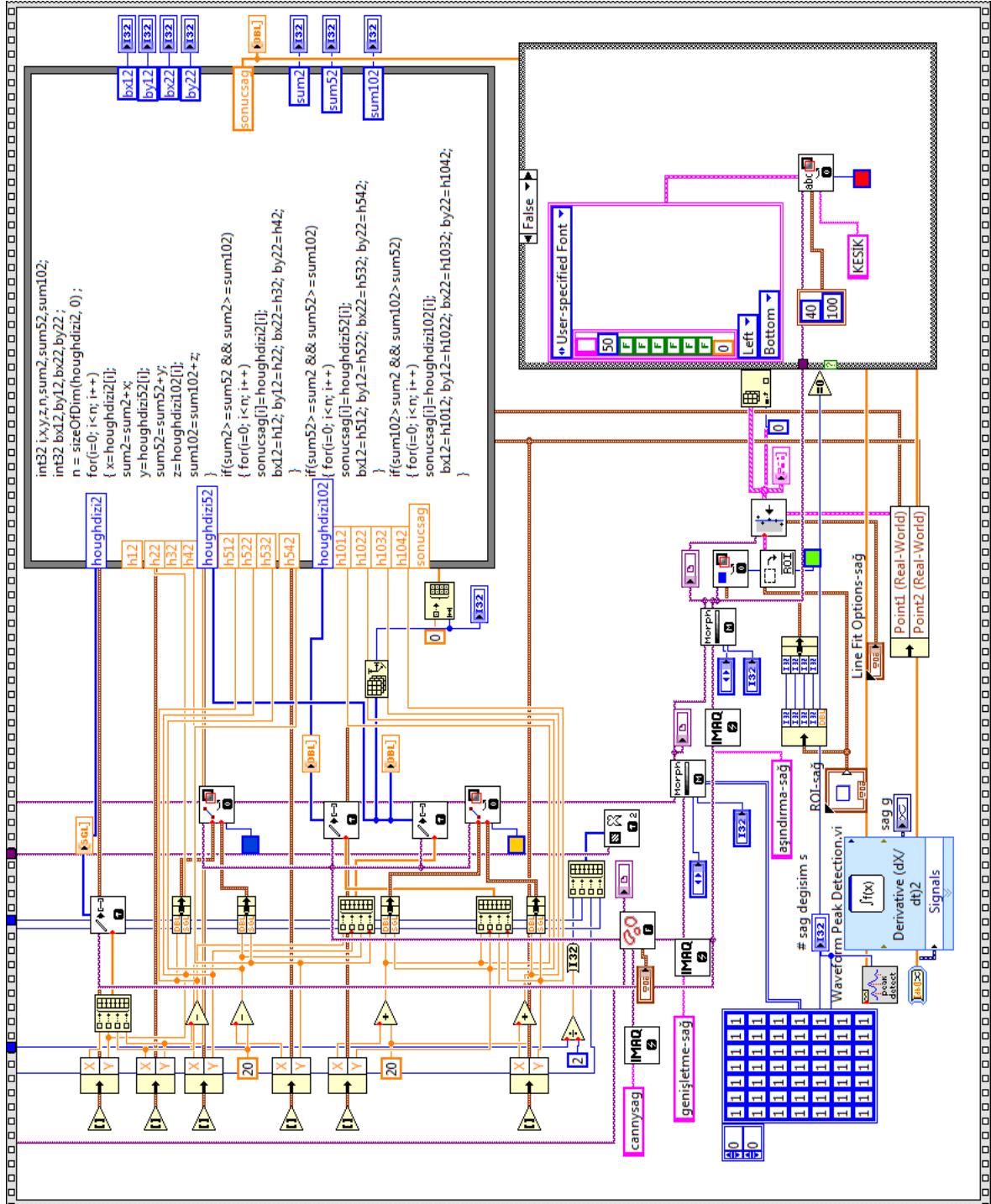
- [32] Assidiq, A. A., Khalifa, O. O., Islam, M. R., & Khan, S. (2008, May). Real time lane detection for autonomous vehicles. In Computer and Communication Engineering, 2008. ICCCE 2008. International Conference on (pp. 82-88). IEEE.
- [33] Mukhopadhyay, P., & Chaudhuri, B. B. (2015). A survey of Hough Transform. Pattern Recognition, 48(3), 993-1010.
- [34] Yenikaya, S., Yenikaya, G., & Düven, E. (2013). Keeping the vehicle on the road: A survey on on-road lane detection systems. ACM Computing Surveys (CSUR), 46(1), 2.
- [35] Schowengerdt, R. A. (2012). Techniques for image processing and classifications in remote sensing. Academic Press.
- [36] Qin, B., Xia, Y., Prabhakar, S., & Tu, Y. (2009, March). A rule-based classification algorithm for uncertain data. In Data Engineering, 2009. ICDE'09. IEEE 25th International Conference on (pp. 1633-1640). IEEE.



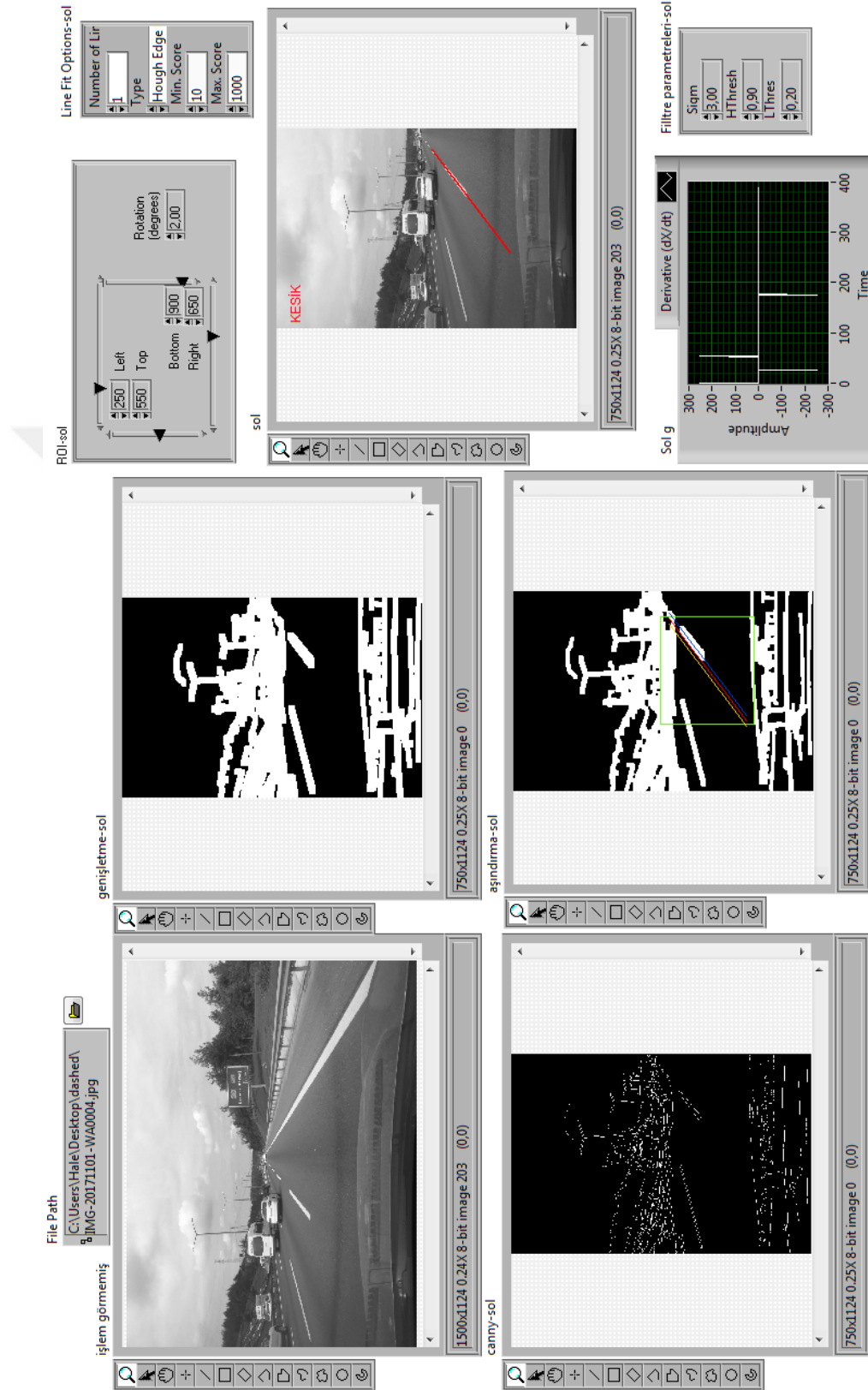
## EK A: Sistem Blok Diyagramı - Sol Bölüm



## EK B: Sistem Blok Diyagramı - Sağ Bölüm



## EK C: Sistem Ön Paneli



## ÖZGEÇMİŞ

1989 yılında Kocaeli’ de doğdu. 2013 yılında T.C. Marmara Üniversitesi, Elektronik-Bilgisayar Eğitimi Bölümü, Elektronik ve Haberleşme Öğretmenliği (İngilizce) lisans programından mezun oldu. 2013 yılında T.C. Marmara Üniversitesi, Elektrik-Elektronik Mühendisliği Anabilim Dalı, Elektrik-Elektronik Mühendisliği (Türkçe) Programında, yüksek lisans eğitime başladı. 2013 yılında T.C. İstanbul Aydın Üniversitesinde öğretim görevlisi olarak çalışmaya başladı.

