

**YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

OTOMATİK HEDEF TANIMA (ATR)

Mühendis Nuri Gökhan KORKUSUZ

**FBE Elektronik ve Haberleşme Mühendisliği Anabilim Dalı
Haberleşme Programında Hazırlanan**

YÜKSEK LİSANS TEZİ

Tez Danışmanı : Prof. Dr. Vedat TAVŞANOĞLU

İSTANBUL, 2008

1.	GİRİŞ.....	1
2.	GÖRÜNTÜ İŞLEME TEKNİKLERİ LİTERATÜR TARAMASI	3
2.1	GÖRÜNTÜ İYİLEŞTİRME.....	3
2.1.1	Ortalama Filtre.....	3
2.1.2	Gausyan Filtre.....	3
2.1.3	Medyan Filtre	3
2.1.4	Görüntü Küçültme/Büyütme	4
2.1.5	Morfolojik Yöntemler.....	4
2.1.5.1	Genişletme ve Daraltma	4
2.1.5.2	Açma ve Kapama.....	5
2.2	GÖRÜNTÜ BÖLÜMLEME (SEGMENTASYON)	6
2.2.1	Histogram Temelli Bölümleme	7
2.2.2	Alan Temelli Bölümleme	7
2.2.2.1	Bölge Büyütme (Region Growing).....	7
2.2.2.2	Bölge Bölme ve Birleştirme (Region Splitting and Merging).....	8
2.2.2.3	K- Ortalama (K-means)	8
2.2.3	Kenar Çizgileri (Edge Detection) Temelli Bölümleme	8
2.2.3.1	Prewitt Kenar Belirleme Yöntemi	9
2.2.3.2	Sobel Kenar Belirleme Yöntemi.....	9
2.2.3.3	Laplasyen Kenar Belirleme Yöntemi	10
2.2.3.4	Canny Kenar Belirleme Yöntemi	10
2.3	SINIFLANDIRMA.....	13
2.3.1	Yapay Sinir Ağları.....	13
2.3.2	İstatistiksel Yöntemler	13
2.3.2.1	Temel Bileşen Analizi (PCA).....	13
2.3.2.2	Doğrusal Fark Analizi (LDA).....	14
2.3.2.3	Bağımsız Bileşen Analizi (ICA).....	15
2.3.3	Modele Dayalı Yöntemler	15
2.3.3.1	Model (Template) Eşleme	16
2.3.3.2	Kenar Piksellerini Eşleme	16
2.3.4	Parzen Penceresi (Parzen Window).....	16
2.3.5	En Yakın K Komşu (K_NN: K Nearest Neighbourhood).....	17
2.3.6	Kovaryans Analizi	18
2.3.7	Ayrık Kosinüs Dönüşümü (DCT).....	19
3.	OTOMATİK HEDEF TANIMA	20
3.1	ATR KULLANIMI.....	20
3.2	ATR SİSTEMİ.....	22
3.2.1	Gereksinimler	22
3.2.2	Görev Dağılımı	22
3.2.3	Donanım Mimarisi.....	23
3.2.3.1	Sensörler	24
3.2.3.2	Optik Düzenekler.....	26
3.2.3.3	Dijital İşaret İşleme Kartı	28
3.3	ATR ALGORİTMASI.....	29
3.3.1	Hedef Üzerindeki Piksel Sayısı	30

3.3.2	Veri Tabanı Kompleksliği	34
4.	GELİŞTİRİLEN ATR YÖNTEMİ	35
4.1	GÖRÜNTÜ İYİLEŞTİRME	37
4.1.1	Ön Filtreleme	38
4.1.2	Histogram Eşleme	39
4.1.3	Resim Toplama	40
4.1.4	Düzgün Olmayan Yayınım	41
4.1.5	Görüntü İyileştirme İşlemleri Değerlendirmesi	50
4.2	BÖLÜMLEMEDE KULLANILAN YÖNTEM	53
4.2.1	Geliştirilmiş Canny Yöntemi ile Kenar Belirleme	53
4.2.1.1	Hysteresis Yöntemi	53
4.2.1.2	Otsu Yöntemi	53
4.2.2	Kenar İyileştirme	56
4.2.3	Doldurma (Filling)	57
4.2.4	Bağlı Bileşen Etiketleme (CCL)	58
4.2.5	Bölümleme Değerlendirmesi	60
4.3	ŞEKİL SINIFLANDIRMA	61
4.4	ŞEKİL TANIMA	62
4.4.1	Şekil Tanıma Algoritmaları Değerlendirmesi	63
4.5	GELİŞTİRİLEN ALGORİTMANIN UYGULAMA ALANLARI	65
4.5.1	İNSAN TESPİTİ	65
4.6	TAŞIT TANIMA	67
4.6.1	Hava Aracı Tanıma	67
4.6.2	Deniz Aracı Tanıma	68
4.6.3	Kara Aracı Tanıma	69
4.7	MEDİKAL GÖRÜNTÜLEME	70
4.7.1	Otomatik Virüs Tanıma	70
4.7.2	Otomatik Bakteri Tanıma	71
4.7.3	Otomatik Akyuvar Hücre Sınıflandırma	72
5.	SONUÇ	74
	KAYNAKLAR	76
	EK_1 MATLAB Algoritma Analiz Kodu	78
	EK_2 MATLAB Algoritma Kodu	80
	Ek_3 Homojen Olmayan Yayınım Matlab Uygulama Kodu	84
	ÖZGEÇMİŞ	85

SİMGE LİSTESİ

I : Orijinal Resim

$\|\nabla I\|$: Gradyantın mutlak değeri

$g()$: Kenar Durdurma Fonksiyonu

σ : Varyans

S : Standart Sapma

S : Kaynak

N : Gürültü

G_x : Yatay Kenar Şiddeti

G_y : Dikey Kenar Şiddeti

G : Bileşke Kenar Şiddeti

φ : Kenar Açısı

A : Görüntü

B : Morfolojik İşlem Matrisi

$D()$: Genişletme (Dilation) Fonksiyonu

$E()$: Daraltma (Erosion) Fonksiyonu

$O()$: Açma (Opening) Fonksiyonu

$C()$: Kapatma (Closing) Fonksiyonu

X : Vektör haline getirilmiş veri matrisi

X_i : i . sıradaki X vektörü

$\bar{X}$: X vektör ortalaması

m : Genel Ortalama

m_i : i . X vektörünün Ortalaması

U : Hedef Matrisi

S_i : Kovaryans Matrislerinin toplamı

S_v : Bütün Matrislerin Toplamı

S_B : Sınıflar Arasındaki Uzaklık

V : Öz Vektörler

A : Katsayı Matrisi

S : Kaynak

x : veri Vektörleri

W : Ayrıcı Matrisi

P : Permutasyon Matrisi

S : Katsayı Matrisi

$Kurt$: Kurtosis Katsayısı

$E\{ \}$: Entropi

y : Ayrık değişken

Y : Diğer Veri Kümesi

KISALTMA LİSTESİ

ATR : Otomatik Hedef Tanıma (Automatic Target Recognition)
MNN : Modüler Yapay Sinir Ağları (Modular Neural Network)
CNN : Konvolüsyonel Yapay Sinir Ağları (Convolutional Neural Network)
PCA : Temel Bileşen Analizi (Principle Component Analysis)
LDA : Doğrusal Fark Analizi (Linear Discriminate Analysis)
ICA : Bağımsız Bileşen Analizi (Independent Component Analysis)
CCL : Bağlı Bileşen Etiketleme (Connected Component Labeling)

ŞEKİL LİSTESİ

Sayfa

Şekil 2.1 Veritabanı Tank Resmi Örneği.....	4
Şekil 2.2 Genişletme ve Daraltma Yöntemi	5
Şekil 2.3 Örnek Resim Histogramı.....	7
Şekil 2.4 K-Means Sınıflandırma Yöntemi	8
Şekil 2.5 Sobel yöntemi ile kenar tespiti	10
Şekil 2.6 Laplasyen kenar belirleme yöntemi	10
Şekil 2.7 Açılarının değerlerine göre gruplandırılması	12
Şekil 2.8 Model Eşleme Akış Modeli.....	16
Şekil 2.9 Parzen Window Uygulaması.....	17
Şekil 2.10 K_NN Metodu Uygulaması	18
Şekil 3.1 Örnek ATR Kullanımı.....	20
Şekil 3.2 Görev Planlaması	21
Şekil 3.3 ATR Sistemi Görev Dağılımı.....	23
Şekil 3.4 Örnek ATR Donanım Mimarisi	24
Şekil 3.5 CCD Sensör Çalışma Prensibi.....	24
Şekil 3.6 CCD Görüntü Sensörü Örneği (Kodak)	25
Şekil 3.7 Sensör – ATR ilişkisi (Licata, 2001).....	25
Şekil 3.8 ATR ile Hedef Tanıma (Licata, 2001)	26
Şekil 3.9 Kırıcı Optik düzeneği	27
Şekil 3.10 Yansıtıcı Optik Düzeneği	27
Şekil 3.11 (Texas Instruments) TMDXEVM6446	28
Şekil 3.12 ATR Algoritmaları (Licata, 2001)	29
Şekil 3.13 Hedef Üzerindeki Piksel Sayısı (Licata, 2001)	30
Şekil 3.14 Çözünürlüğe bağlı ATR seviyeleri.....	31
Şekil 3.15 Çözünürlük-Olasılık Eğrileri.....	33
Şekil 3.16 Model Kompleksliği (Licata,2001)	34
Şekil 4.1 ATR Algoritma Akış Diagramı.....	35
Şekil 4.2 Test Resimleri	36
Şekil 4.3 Medyan filtre uygulaması.....	38
Şekil 4.4 Resimlerin Histogram Eşlenmiş Halleri.....	39
Şekil 4.5 Resimlerin Histogramlarıyla Toplanması	40
Şekil 4.6 En yakın Komşu pikseller (Perona ve Malik (1990)).....	42
Şekil 4.7 Lorentzian, Tukey, Huber, Dikdörtgen Hata (g(x)) Fonksiyonları	45

Şekil 4.8 Resimlere Düzgün Olmayan Yayınım Uygulanması	48
Şekil 4.9 Bütünleştirilmiş Resimlerin Renklendirilmesi	49
Şekil 4.10 Canny Metodu ile Kenar Tespitinin Yapılması.....	54
Şekil 4.11 Canny Metodu ile Kenar Tespitinin Yapılması.....	55
Şekil 4.12 Closing İşlemi Uygulanmış Resimler	56
Şekil 4.13 Doldurulmuş Test Resimleri	57
Şekil 4.14 Resimlerden Küçük Şekillerin Temizlenmesi.....	59
Şekil 4.15 Anlamsız Şekillerin Elenmesi	61
Şekil 4.16 DCT ile Hedef Tanıma	62
Şekil 4.17 1. ve 2. Resimlerden Elde Edilmiş Hedef Görüntüleri.....	63
Şekil 4.18 3 ve 4. Resimlerden Elde edilmiş Hedef Görüntüleri	64
Şekil 4.19 Hedef Veri Tabanı Örneği.....	64
Şekil 4.20 İnsan Tespiti Uygulaması 1	65
Şekil 4.21 İnsan Tespiti Uygulaması 2	66
Şekil 4.22 İnsan Tespiti Uygulaması 3	66
Şekil 4.23 Veri Tabanı İnsan Resmi.....	66
Şekil 4.24 Hava Aracı Uygulaması	67
Şekil 4.25 Veri Tabanı Uçak Resmi	67
Şekil 4.26 Deniz Aracı Uygulaması 1	68
Şekil 4.27 Deniz Aracı Uygulaması 2	68
Şekil 4.28 Deniz Aracı Uygulaması 3	69
Şekil 4.29 Veri Tabanı Gemi Resmi.....	69
Şekil 4.30 Virüs Tanıma Uygulaması	70
Şekil 4.31 Veri tabanı Virüs Resmi	71
Şekil 4.32 Bakteri Tanıma uygulaması.....	71
Şekil 4.33 Veri Tabanı E.Coli Bakterisi Resimleri	72
Şekil 4.34 Akyuvar Sınıflandırma Uygulaması 1	72
Şekil 4.35 Akyuvar Sınıflandırma Uygulaması 2.....	73
Şekil 4.36 Akyuvar Sınıflandırma Uygulaması 3.....	73
Şekil 4.37 Veri Tabanı Akyuvar Resimleri	73
Şekil 5.1 FPGA ve DSP ile Donanımsal Algoritma Mimarisi (Schlosser ve Fecht,2005).....	75

ÇİZELGE LİSTESİ

Sayfa

Tablo 3.1 Johnson Kriteri Piksel Sayıları	32
Tablo 3.2 Çözünürlük Değerleri	33
Tablo 4.1 Ön işlemler Değerlendirme Çizelgesi	52
Tablo 4.2 CCL Algoritma Akışı	58
Tablo 5.1 ATR Algoritma Performans Testi	74

ÖNSÖZ

Her şeyden önce bana yol gösteren ve tezimi bitirmemde büyük emekleri olan danışmanım Prof. Dr Vedat TAVŞANOĞLU'na teşekkürlerimi sunuyorum.

Beni hiç bir zaman yalnız bırakmayıp her zaman yanımda olan aileme ve eşime bana olan desteklerinden ve güvenlerinden dolayı teşekkürlerimi sunuyorum.

Son olarak da mensubu olmaktan gurur duyduğum, ROKETSAN A.Ş.'ye Yüksek Lisans eğitimini teşviki ve sağladığı imkanlardan dolayı; başta MÜ-GE Direktörüm olmak üzere bütün amirlerime tez çalışmalarım boyunca gösterdikleri anlayıştan dolayı teşekkürlerimi ve şükranlarımı sunuyorum.

Otomatik Hedef Tanıma (ATR)

ÖZET

Otonom saldırı ve savunma sistemlerinin önem kazandığı günümüz savunma teknolojilerinde, kuşkusuz ATR (otomatik hedef tanıma) de önemli bir yere sahiptir. ATR, görüntü işleme tekniklerinden oluşan nesne tanıma algoritmalarının, askeri hedefleri otonom olarak ayırt etmesidir. ATR, 1980'lerin başlarından beri araştırılmaktadır ve aşama aşama savaş sistemlerine entegre edilmektedir (Ratches vd.,1997).

ATR algoritmalarının en önemli problemi, getirdiği işlem yükü düşünülmezsizin ve tek bir senaryoya bağlı olarak çalışmalarıdır. Ne var ki, akademik bir araştırmadan çok, sistem mühendisliği bakış açısıyla mevcut savaş sistemlerine uyum, gerçek zamanlı çalışabilme, farklı koşullara karşı dayanıklılık ve girdi olarak alınabildiği varsayılan bilgilerin savaş alanında tedarik edilebilirliği ATR algoritmalarını gerçek sistemlere dönüştürecek parametrelerdir.

ATR konusu üzerine dünya literatüründe binlerce makale ve yayın olduğu bilinmesine karşın Türkiye'den çok az sayıda makale ve tezin bu konu üzerinde yoğunlaştığı görülmüştür. İncelenen makale ve tezlerin ya literatür taraması yaptığı, ya kullanılabilecek metotları karşılaştırdığı ya da ATR algoritmasının bir kısmını anlattığı görülmüştür.

Yapılan tez çalışmasında ise literatürdeki çalışmaların bir çoğundan farklı olarak; ATR(Otomatik Hedef Tanıma) sistem olarak ele alınacak, geliştirilen ATR algoritması, ön işlemlerinden sınıflandırmasına kadar her aşamayı barındıracak ve çalışan yöntemin her aşaması detaylı olarak uygulanan resimlerle birlikte anlatılacaktır. Geliştirilen ATR algoritması, her hangi bir pozisyon veya hedef renk bilgisi kullanılmadan, gerçek zamanlı çalışabilen, farklı senaryolara karşı dayanıklı, farklı büyüklüklerde ve farklı renklerdeki hedefleri ayırt edebilen, hedef parametrelerini ve veri tabanını kullanan, bilinen görüntü işleme tekniklerinin birlikte kullanılmasıyla ve optimize edilmesiyle geliştirilmiş melez bir yöntemdir.

Anahtar Kelimeler : Otomatik hedef tanıma, Johnson kriteri, Homojen olmayan yayılım, Ayırık kosinüs dönüşümü, Virüs tanıma, Bakteri sayma, Akyuvar sınıflandırma, Görüntü iyileştirme, görüntü bölümleme

AUTOMATIC TARGET RECOGNITION (ATR)

ABSTRACT

Nowadays, autonomous attack and defense systems are becoming more popular and Automatic Target Recognition (ATR) is a very good example for those systems. ATR is an object recognition application on army targets which consists of image processing techniques. ATR has been a popular research topic since 1980s and has been integrating on army systems step by step (Ratches vd.,1997).

Most important problems for ATR algorithms are, processor needs due to the complexity of algorithms and sensitivity to different cases. In addition, ATR should be studied not only as an academic research but also as an important system which is needed to be easily adaptable, robust, and able to work in real time with easily accessible inputs.

ATR is an important worldwide research topic and there are more than thousands of articles and publications about ATR. However, researches, articles and thesis from Turkey about ATR are very limited. Moreover, those limited articles and thesis focus on only components of ATR (Bayık, 2004).

Scope of this thesis is to explain ATR System and improved ATR algorithm including all components from image enhancement to target classification with example pictures in detail. Although, improved ATR method is not aided (doesn't need any warfare information like target position, color, etc.), it is able to work robust (can classify different targets which have different sizes, colors and orientations) and real time by using target parameters and target database with optimized image processing techniques.

Keywords : Automatic Target Recognition, Johnson criteria, Anisotropic diffusion, Discrete Cosine Transform, Virus recognition, Bacteria Counting, White Cell Classification, Image Segmentation, Image Enhancement

1. GİRİŞ

Otonom saldırı ve savunma sistemlerinin önem kazandığı günümüz savunma teknolojilerinde kuşkusuz ATR (otomatik hedef tanıma) da önemli bir yere sahiptir. ATR, görüntü işleme tekniklerinden oluşan nesne tanıma algoritmalarının, askeri hedefleri otonom olarak ayırt etmesidir. ATR, 1980'lerin başlarından beri araştırılmaktadır ve aşama aşama savaş sistemlerine entegre edilmektedir (Ratches vd.,1997).

Önceleri insanların üzerinde olan düşman araçlarını tanıma ve seçme işlemleri, düşman araç ve hedef çeşitliliğinin artması ile birlikte gitgide zorlaşmıştır. Kullanıcı yükünü azaltabilmek için bu işlemlerde bilgisayar kullanılması fikri otonom çalışan algoritma ihtiyacını oluşturmuştur. Küçük ve güçlü video bilgisayarların pazarda önem kazanması ve ihtiyaç duyulan işlem yüklerini gerçek zamanlı işleyebilir hale gelmeleriyle düşman tanıma görevini bilgisayarlar üstlenmiştir.

Modern bilgisayarların daha da yüksek güçlere ulaşmaları gerektiğini, yeni savaş sistemlerinde kullanılmaya başlayacak çok daha modern bilgisayar ara yüzleriyle insan bilgisayar arasındaki geçişin hızlanacağı düşünülmektedir. Bu bilgilerin ışığında, savaş alanında kullanılan akıllı algoritmaların ve uygulamalarının ticari payının artacağı düşünülmektedir.

Tez kapsamında, 2. Bölüm ATR Algoritma alternatifleridir. Bu bölümde ATR problemine istatistiksel olarak yaklaşan PCA (Principle Component Analysis), LDA (Linear Discriminant Analysis), ICA (Independent Component Analysis) yöntemleri, ATR problemini model tabanlı çözmeye çalışan Şablon Eşleştirme (Template Matching) ve Kenar tanıma yöntemleri ile ATR problemini Yapay Sinir Ağlarıyla çözmeye çalışan yöntemlerden bahsedilecektir.

3. Bölüm ATR kavramı ve sistem olarak ATR incelemesidir. Bu bölümde ATR; algoritmalarından nasıl gerçek bir ATR sistemi oluşturulabileceği, ATR sistemi işlevselliği, mimarisi, seviyesi, alt sistemleri, ihtiyaçları, geliştirilebilirliği, görev planlaması yönleriyle incelenecektir.

4. Bölümde geliştirilen ATR yöntemi, ATR yöntemini oluşturan Görüntü iyileştirme algoritmaları, alternatifleri, neden tercih edildiği anlatılacaktır. Olası hedef bölümlleme (segmentasyon) algoritması, alternatifleri, neden tercih edildiği ve veri tabanı yardımıyla hedef tanımlama algoritması detaylı olarak örnek resimlerle birlikte yine 4. Bölümde

anlatılacaktır. Yine, geliştirilen ATR yönteminin hangi alanlarda kullanılabileceği örneklerle birlikte anlatılacaktır.

5. Bölümde yapılan ATR Algoritma geliştirme çalışması yapılan testlerin sonuçlarıyla birlikte özetlenecek, değerlendirilecek ve örnek bir donanımdan bahsedilecektir.

2. GÖRÜNTÜ İŞLEME TEKNİKLERİ LİTERATÜR TARAMASI

ATR algoritmaları incelenecek olursa; ATR algoritmalarının, ancak bir çok görüntü işleme tekniğinin birlikte kullanılması ile gerçekleştirilebileceği görülür. Görüntüleyici düzenekten gelen görüntünün önce iyileştirilmesi, sonra bölümlenmesi ve son olarak da olası nesnelerin sınıflandırılması gerekmektedir. Yapılan literatür çalışmasında da her aşama için kullanılabilecek alternatif yöntemlerden yaygın olarak kullanılan yöntemler incelenmiştir.

2.1 GÖRÜNTÜ İYİLEŞTİRME

Görüntüleyici sistemlerden elde edilen görüntülerde her zaman gürültü olduğu bilinmektedir. Görüntü üzerindeki gürültüyü azaltabilmek ve bölümleme işlemi kolaylaştırmak için çeşitli görüntü iyileştirme yöntemleri kullanılmaktadır. Bu yöntemlerden en yaygın olanları filtreler (Ortalama, Gausyan, Medyan vb) ve morfolojik yöntemlerdir.

2.1.1 Ortalama Filtre

Ortalama Filtrede ilgili piksel, komşularıyla toplanır, ortalama alınır, pikselin yeni değeri olarak ortalama değer yazılır. Tuz ve biber gürültüsünde medyan filtre kadar iyi sonuçlar vermez.

2.1.2 Gausyan Filtre

Gausyan Filtrede ise istenilen büyüklükte, gauss formülüne uygun olarak hazırlanmış matrisin resim üzerinde uygulanması ile elde edilir. Resim üzerinden gürültüyü azaltma konusunda etkin olmasına karşın istenilen objelerin kenar özelliklerine zarar verebileceği değerlendirilmiştir.

2.1.3 Medyan Filtre

Frekans domeninde uygulanacak bir filtreden olası hedef boyutları bilinmediği için vazgeçilmiştir. Ortalama ve Gausyan filtreler, düşük frekans geçiren filtreler olduğu için kenar geçişlerinde bozulmalara neden olmaktadır. Medyan filtrenin ise kenar geçişlerini bozmadan, algılayıcı gürültüsünü azalttığı Şekil 4.3 üzerinde de görülmektedir.

2.1.4 Görüntü Küçültme/Büyütme

Görüntü küçültme/büyütme işleminde resim piramidi yöntemi kullanılmıştır. Görüntü Piramidi (Image Pyramid) Yöntemi resmin bütününe veya bir bölümünün büyüklüğünün değiştirilmesinde (resizing) kullanılan metotlardan biri olarak tanımlanmaktadır. Bu yöntemde küçültme işlemi için iki veya daha fazla pikselin ortalaması alınır, bulunan ortalama tek piksel değeri olarak yazılır, bu şekilde 2 piksel için boyutlar yarı yarıya düşürülür. Aynı şekilde resmi büyütme için iki pikselin ortalaması alınır iki pikselin arasına yazılır, bir piksel yerleştirmek yetmeyecekse iki piksele olan uzaklıklarına göre ağırlıklı değerler yazılır.



Şekil 2.1 Veritabanı Tank Resmi Örneği

2.1.5 Morfolojik Yöntemler

Morfoloji kelimesi, bir nesnenin şekil ve yapısı ya da nesnenin parçaları arasındaki yerleşim ve ilişki anlamına gelmektedir. Morfoloji terimi dil ve biyoloji terimi olarak ortaya çıkmıştır; bilgisayarlar ile birlikte de sayısal görüntü işlemlerine uyarlanmıştır.

Görüntü işlemede kullanılan Morfolojinin arkasındaki matematik, basit bir küme teorisidir. Morfoloji, basit küme işlemleri yapılarak resim içerisindeki şekillerin tanınmasını ve sayılmasını kolaylaştırmaktadır.

Temel morfolojik fonksiyonlar Genişletme (Dilation), Daraltma (Erosion), Açma (Opening) ve Kapama (Closing)'dır.

2.1.5.1 Genişletme ve Daraltma

Genişletme işlemi, resmin önceden belirlenmiş bir maske ile mantıksal “veya” işlemi yapılarak maskelenmesi ile elde edilir; böylece resim üzerinde yapı elemanı gezdirilirken resimde veya yapı elemanında nesne kabul edilen değer, örneğin “1” varsa resim üzerine aktarılır. Böylece resim üzerindeki şekil yapı elemanına bağlı olarak genişlemiş ve kenar boşlukları doldurulmuş olur. Şekil 2.2’de örnek bir A resmi üzerine B yapı elemanı uygulandığında elde edilen sonuç matrisi olarak gösterilmektedir.

Daraltma işlemi, resmin önceden belirlenmiş bir maske ile mantıksal “ve” işlemi yapılarak maskelenmesi ile elde edilir; böylece resim üzerinde yapı elemanı gezdirilirken resimde ve yapı elemanında nesne kabul edilen değer, örneğin “1” varsa resim üzerindeki değer korunur, aksi durumda piksel değeri sıfırlanır. Böylece resim üzerindeki şekil, yapı elemanına bağlı olarak daralmış ve kenar çıkıntıları düzleştirilmiş olur. Şekil 2.2’de örnek bir A resmi üzerine B yapı elemanı uygulandığında elde edilen sonuç matrissel olarak gösterilmektedir. Maskeye bağlı olarak genişletme (2.1), resmin çıkıntılarını biraz daha büyütürken, daraltma (2.2) biraz daha küçültür.

$$D(A, B) = A \oplus B = \bigcup_{\beta \in B} (A + \beta) \quad (2.1)$$

$$E(A, B) = A \ominus (-B) = \bigcap_{\beta \in B} (A - \beta) \quad (2.2)$$

A: Birinci Şekil $\oplus$: Mantıksal “toplama” işlemi $\ominus$: Mantıksal “çıkarma” işlemi
B: Maske β : B maskesinin elemanı

Şekil 2.2’de basit bir uygulama görülmektedir, maske olarak belirlenmiş B cismi daraltma ve genişletme işlemleri uygulanarak resim üzerinde gezdirildiğinde birinci şekli aşağıdaki gibi değiştirmektedir.

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad B = [1 \ 1] \quad A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Örnek A kümesi ve
B yapı elemanı

Genişletme D(A,B)

Daraltma E(A,B)

Şekil 2.2 Genişletme ve Daraltma Yöntemi

2.1.5.2 Açma ve Kapama

Genişletme ve daraltma yöntemleri belirli bir sırada uygulanarak açma ve kapama yöntemleri elde edilir.

$$O(A, B) = D(E(A, B), B) \quad (2.3)$$

Açma (2.3), önce daraltma sonra genişletme işleminin A resmi üzerinde B maskesi ile yapılması olarak tanımlanabilir; resim üzerindeki gürültüyü ve kopuk bölümleri yok eder.

$$C(A, B) = E(D(A, -B), -B) \quad (2.4)$$

Kapatma (2.4) ise, önce genişletme sonra daraltma işleminin A resmi üzerinde B maskesi ile yapılması olarak tanımlanabilir; kalan büyük parçaları birleştiren, kopuklukların birleştirilmesini sağlayan morfolojik işlemidir.

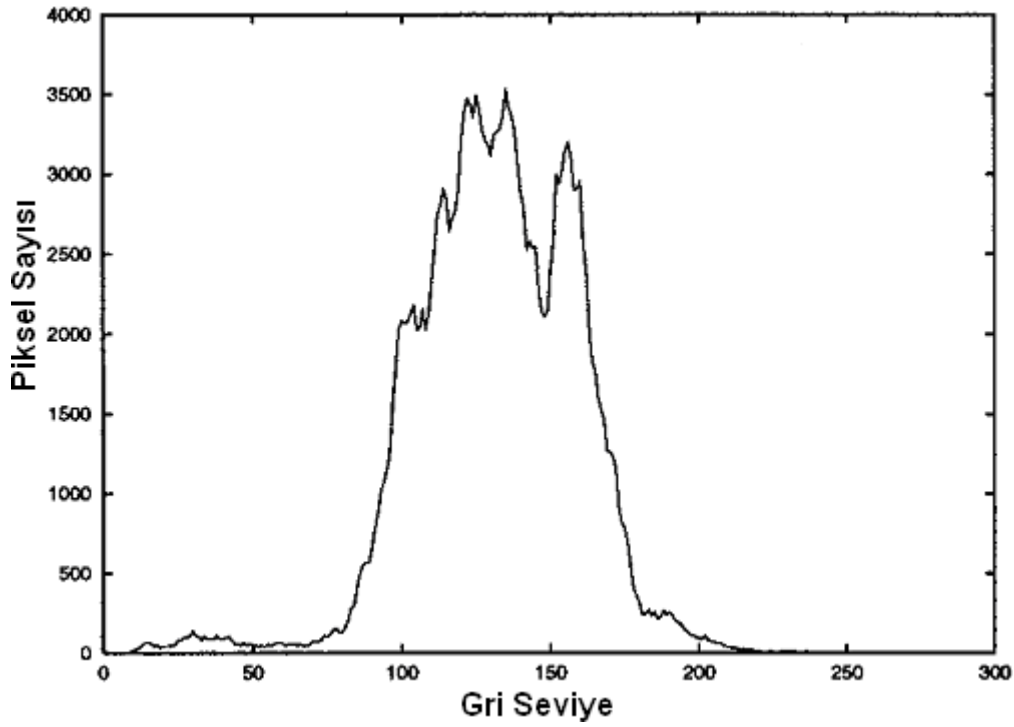
2.2 GÖRÜNTÜ BÖLÜMLEME (SEGMENTASYON)

Bölümleme resim içerisindeki farklı bölgelerin birbirinden ayırt edilmesi anlamına gelir. ATR Algoritmasında gerçekleştirilecek bölümleme işleminin amacı ise görüntü içerisindeki aday hedeflerin bulunup, arka plandan ayırt edilmesidir. Aday hedeflerin kenar özellikleri ve şekilleri ne kadar iyi ayırt edilirse veri tabanındaki hedef görüntüleriyle kıyaslamak da o kadar kolaylaşacaktır.

Bölümleme işlemini gerçekleştirebilecek bir çok yöntem türetilmiştir. Bölümleme işlemi için; Kenar temelli, Alan Temelli, ve Yapay Sinir ağları temelli bir çok yöntem bulunduğu bilinmektedir. Bölüm kapsamında ise, bölümleme yöntemlerinden histogram temelli bölümleme (Bölüm 2.2.1) alan temelli bölümleme (Bölüm 2.2.2) ve kenar temelli bölümleme (Bölüm 2.2.3) yöntemleri hakkında genel bilgi verilecektir.

2.2.1 Histogram Temelli Bölümleme

Histogram temelli yöntemler, tüm resmin histogramının alınarak histogram üzerindeki kopuklukların olduğu yerlerden ayırt edilmesi prensibine dayanır. Histogram değerlerinin üzerine bir eğri uydurmak ve eğrinin keskin noktalarını tespit etmek başarılı uygulamalardandır (Wesley vd., 2000).



Şekil 2.3 Örnek Resim Histogramı

Örnek histogramda da görüldüğü gibi geniş vadilerin olduğu noktalar, eşik değeri olarak belirlenir.

2.2.2 Alan Temelli Bölümleme

Alan temelli bölümlemede resim daha küçük alanlı parçalara ayrılarak bölümleme işlemi gerçekleştirilir.

2.2.2.1 Bölge Büyütme (Region Growing)

Bölge büyütme yaygın olarak kullanılan bir bölümleme algoritmasıdır. Bölge büyütme temel olarak önceden belirlenmiş ya da rasgele atanmış merkezlerin n komşuluğundaki piksellere bakılarak benzer pikseller yönünde genişletilmesi ile elde edilir. Benzerlik ölçütü olarak bir eşik değeri atanır. İstenilen nesnenin resim içerisinde büyük alan kapladığı bir resimde etkili olacak bir yöntemdir fakat istenilen nesne küçük boyutlara sahipse bölge büyütme ile istenilen nesne kaybedilebilir.

Bölge büyütme yönteminin önceden pozisyonu bilenen nesneleri bulmakta ise çok daha faydalı olacağı değerlendirilmektedir.

2.2.2.2 Bölge Bölme ve Birleştirme (Region Splitting and Merging)

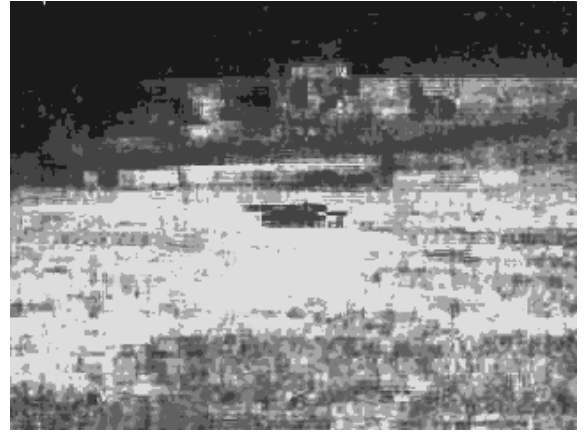
Resim önce küçük alanlara bölünür, sonra benzer özellik gösteren alanlar birleştirilerek resim bölümlenmiş olur. Yöntemin uygulanan ATR algoritması açısından çok kullanışlı değildir. Çünkü resim yeterince küçük alt parçalara bölünemezse hedef tespit edilemeyebilir; resim çok fazla alt bölüme ayrılırsa da iş yükü çok fazla artacaktır.

2.2.2.3 K- Ortalama (K-means)

Önce pikseller tonlar eşit uzaklıkta olacak şekilde gruplara ayrılır, sonra her grubun ortalaması o grubun yeni değeri olur ve bu ortalamalarına uzaklıklarına göre pikseller tekrar gruplandırılır. Bu işlem birkaç defa tekrarlanır ve pikseller gruplanmış olur. Yine Şekil 2.4' de K-Means uygulanmış bir resim gösterilmiştir.



256 renk ile Orijinal Resim



K-Means ile 8 renk' e indirgenmiş Resim

Şekil 2.4 K-Means Sınıflandırma Yöntemi

2.2.3 Kenar Çizgileri (Edge Detection) Temelli Bölümleme

Kenar tanıma, resimdeki geçişleri hedef alan ve onları tespit etmeye çalışan bir yöntemdir. En çok kullanılan yöntem maske uygulamaktır. En yaygın kenar tanıma yöntemleri Prewitt, Sobel, Laplacian, Canny, LoG 'dır. Belirtilen yöntemler içerisinde en başarılısı şüphesiz ki kenar yönlerini tespit edip inceltme ve düzeltme işlemleri de gerçekleştirdiği için Canny yöntemidir.

2.2.3.1 Prewitt Kenar Belirleme Yöntemi

Prewitt yöntemi, Tablo 2.1'deki 3x3 boyutlarındaki maskelerin resme uygulanması ve çıkan sonuçların skaler toplanmasıyla elde edilir. Matrislerden biri yatay diğeri de dikey kenarları belirginleştirir.

1	0	-1	1	1	1
1	0	-1	0	0	0
1	0	-1	-1	-1	-1
Dikey			Yatay		

Tablo 2.1 Prewitt yönteminde kullanılan maskeler

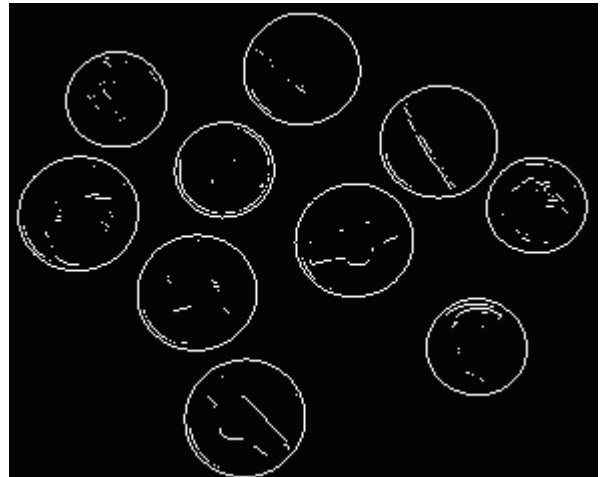
2.2.3.2 Sobel Kenar Belirleme Yöntemi

Resim üzerine istenilen boyutta (genellikle 3x3) yatay ve düşey için ayrı ayrı iki maskenin (Tablo 2.2) uygulanması ile hesaplanır, Prewitt yönteminden daha başarılı sonuçlar verir.

1	0	-1	1	2	1
2	0	-2	0	0	0
1	0	-1	-1	-2	-1
Dikey			Yatay		

Tablo 2.2 Sobel yönteminde kullanılan maskeler

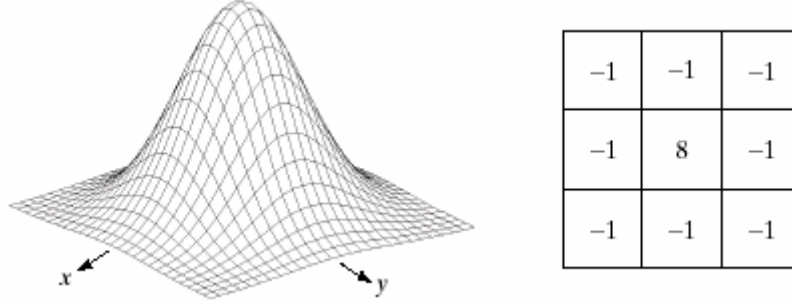
Sobel yöntemi ile filtrelenmiş örnek bir resim Şekil 2.5'de verilmiştir.



Şekil 2.5 Sobel yöntemi ile kenar tespiti

2.2.3.3 Laplasyen Kenar Belirleme Yöntemi

İkinci dereceden türevle elde edilir, kenar keskinliği birinci derece türeve göre daha iyidir. Aşağıdaki şekilde grafik olarak eğrisi ve yaygın olarak kullanılan bir maskesi verilmiştir.



Şekil 2.6 Laplasyen kenar belirleme yöntemi

2.2.3.4 Canny Kenar Belirleme Yöntemi

En başarılı kenar belirleme yöntemidir, bir kaç aşamadan oluşur; önce resme Histogram eşlemesi uygulanır, ardından Gausyan filtre uygulanır, temiz ve berraklaşan resimde Sobel yöntemi ile yatay ve dikey kenar şiddetleri bulunur daha sonra elde edilen kenar şiddetleri oranlanarak (3. aşamada kullanılan yöntemle) kenar yönleri belirlenir ve son olarak da maksimum olmayanı bastırma (Non Maxima Suppression) yöntemi ile kenarlar inceltir, (Canny, 1986).

1.Aşama

Canny'nin kenar belirleme yöntemi birkaç aşamadan oluşur. Kenarları çıkartmaya başlamadan önce yapılacak ilk iş görüntü üzerindeki gürültüyü filtrelemektir. Gausyan filtre çok fazla bir işlem yükü getirmediğinden ve tek bir maske ile resme uygulanabildiğinden filtreleme işleminde kullanılır. Bir kere istenilen parametrelere ve büyüklüğe göre maske hesaplandığında, standart konvolüsyon metoduyla bütün resme uygulanır. Konvolüsyon matrisi resme göre çok daha küçük boyutlara sahiptir; genelde 3x3 ve 5x5 en çok tercih edilen maske büyüklükleridir. Gausyan maske ne kadar büyükse gürültü de o oranda azalacaktır. Gausyan maskenin çok geniş boyutlarda alınması ise kenar tespiti sırasında konumlama hatalarına neden olacaktır. Örnek bir Gausyan maske (2.5) verilmiştir.

$$\frac{1}{115} \begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix} \quad (2.5)$$

2. Aşama

Resim bir miktar düzleştirildikten ve resim üzerindeki gürültü azaltıldıktan sonra kenar şiddetleri gradyant (kısmi türev) alınarak hesaplanır. Kısmi türev almak için 2 boyutlu Sobel operatörleri kullanılır. Sobel operatörlerinden bir tanesi yatay bir tanesi de dikey kenar şiddetlerinin hesaplanması için resim üzerine uygulanır. Resim üzerine uygulanan 3x3lük matrisler(2.6) verilmiştir.

$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad (2.6)$$

G_x G_y

Kenar şiddeti dikey ve paralel şiddetlerin bileşkesi, aşağıdaki yaklaşıklıkla hesaplanır:

$$|G| = |G_x| + |G_y|$$

Aslında gerçek kenar şiddeti, modüllerin ($|G_x|$ ve $|G_y|$) karelerinin toplanmasıyla elde edilmelidir; fakat işlem yükünü azaltabilmek için yukarıda verilen yaklaşım sıkça kullanılmaktadır.

3. Aşama

Dikey ve Yatay kenar şiddetleri ayrı ayrı hesaplandığından kenar yönleri de G_x ve G_y değerlerinden elde edilebilir. Burada dikkat edilmesi gereken nokta G_x değeri '0' olduğunda matematiksel bir tanımsızlık olduğundan yazılımın hata vermesi önlenmelidir. Eğer $G_x = 0$ ise kenar yönü G_y 'ye bağlı olarak ya 90 derecedir yada 0 derece. $G_y=0$ durumunda ise herhangi bir sorun yoktur. Açı hesaplaması ise formül(2.7) ile yapılır:

$$\varphi = \arctan(G_y / G_x) \quad (2.7)$$

4. Aşama

Bir kere kenar yönleri tespit edilince, tespit edilen kenar yönü resim içinde bulunabilen açı yönleriyle eşlenir. 5x5 bir resim parçasığı (2.8) üzerinde incelenebilir.

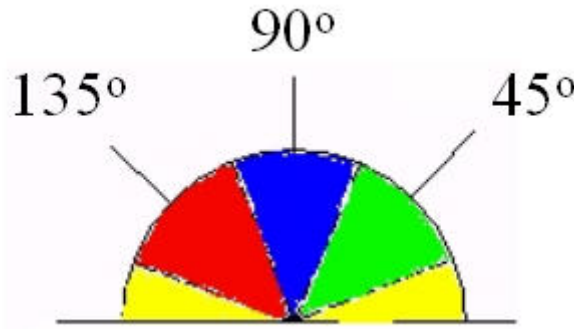
$$\begin{bmatrix} xy & nn & yy & nn & yx \\ nn & xy & yy & yx & nn \\ xx & xx & aa & xx & xx \\ nn & yx & yy & xy & nn \\ yx & nn & yy & nn & xy \end{bmatrix}$$

(2.8)

“aa” pikseline bakılarak oluşan 4 olası yönün açı değerleri:

- 0 derece (paralel yönde, xx ile ifade edilen piksellerin doğrultusunda)
- 45 derece (çapraz yönde, yx ile ifade edilen piksellerin doğrultusunda)
- 90 derece (dikey yönde, yy ile belirtilen piksellerin doğrultusunda)
- 135 derece (çapraz yönde, xy ile belirtilen piksellerin doğrultusunda)

Resimdeki piksellere ait açı yönleri yukarıda verilen açı değerleriyle karşılaştırılır.



Şekil 2.7 Açıların değerlerine göre gruplandırılması

Karşılaştırılan açı değeri; 0 ile 22.5 derece arasında ise Şekil 2.7’deki sarı bölgeye düşer ve 0 derece değerini alır, 22.5 ile 67.5 derece aralığında ise yeşil bölgeye düşer ve 45 derece değerini alır, 67.5 ile 112.5 derece aralığında ise mavi bölgeye düşer ve 90 derece değerini alır, 112.5 ve 157.5 derece arasında ise kırmızı bölgeye düşer ve 135 derece değerini alır, 157.5 ve 180 derece arasında ise tekrar sarı bölgeye düşer ve 0 derece değerini alır.

5.Aşama

Kenarları inceltmek için; tespit edilen kenar yönlerine göre aynı değere sahip pikseller için maksimum olmayanı bastırma yöntemi uygulanır. Maksimum olmayanı bastırma yöntemi (Non Maxima Suppression), açısı aynı olan geniş bir kenarın inceltilmesini ve tek sıra

piksellerden oluşan bir çizgiye dönüştürülmesini sağlar.

6.Aşama

Son olarak ise anlamsız çizgiler belirli eşik değerler (hysteresis thresholding) kullanılarak elimine edilir. Eşik değeri atanması ve kullanılabilecek yöntemlere Bölüm 4.2.1’de detaylı olarak değinilecektir.

2.3 SINIFLANDIRMA

2.3.1 Yapay Sinir Ağları

Yapay sinir ağı sistemleri insan sinir sisteminden esinlenerek geliştirilen bir yöntemdir. Yapay sinir ağları, yüksek işlem gücü ve donanımına kolay aktarılabilir yapısıyla ATR algoritmalarında kullanılabilecek yöntemlerden biridir. Güçlü öğrenme algoritmaları yapay sinir ağlarının en önemli avantajıdır. Kullanılacak özellik kümelerini oluşturabilecek yapay sinir ağları metotları içinde konvolüsyonel yapay sinir ağları ve modüler yapay sinir ağları en önemlileridir. Yapay sinir ağlarında yapılması gereken veri tabanındaki hedef verisi ile olası hedefleri karşılaştırmaktır.

2.3.2 İstatistiksel Yöntemler

İstatistiksel metotlarda, hedef görüntüden önemli istatistiksel özellikler çıkarılarak vektör haline getirilir. Özellik vektörü büyüdükçe, karşılaştırma başarısı da artacaktır; fakat vektörün boyutu da büyüdüğünden hesap yükünü artıracaktır. Bu yüzden özellik vektörünün hedefi hatasız ayırt edebilecek kadar büyük; işlemciler tarafından hesaplanabilecek kadar da küçük olması gerekir. Her hangi bir resim içerisinde hedefin olup olmadığı; olası hedeflerin özellik vektörleri ile önceden hesaplanmış özellik vektörü karşılaştırılarak bulunur. İstatistiksel metotlar içerisinde şekil tanımda kullanılan en yaygın metotlar PCA (Temel Bileşen Analizi), LDA (Doğrusal Fark Analizi) ve ICA (Bağımsız Bileşen Analizi)’dir. PCA, LDA ve ICA yöntemleri yüksek başarımların yanında; işlem miktarını da önemli ölçüde azaltır.

2.3.2.1 Temel Bileşen Analizi (PCA)

Temel bileşen analizi yöntemi, şekil tanımda sıklıkla kullanılan bir metottur. Temel bileşen analizi yönteminde temel olarak pikseller arasındaki korelasyon incelenerek bulunur. Temel bileşen analizi işleminde hedeflenen sonuç ise veri büyüklüğünü minimum hata ile azaltabilmektir (Smith, 2002). Temel bileşen analizi yöntemi, özetlenecek olursa; Örnek uzayındaki M tane, $N \times 1$ boyutlu özellik vektörünün ortalaması alınır, ortalama, özellik

vektörlerinden çıkartılır, vektörler M vektör $M \times N$ büyüklüğünde bir matris oluşturacak şekilde birleştirilir. Elde edilen matris transpozesi ile çarpılarak $N \times N$ boyutlarındaki kovaryans matrisi elde edilir, kovaryans matrisinden, matrisin özdeğerleri bulunur. Bulunan özdeğerlerden en büyük K tanesi seçilerek, K özdeğere ait özvektörler hesaplanır; en büyük özdeğerleri seçmekteki amaç, pikseller arası en büyük farklılıkları kullanabilmektir. Elde edilen öz vektörler olası hedefe ait öz vektörlerle kıyaslanır ve hedef bulunur.

2.3.2.2 Doğrusal Fark Analizi (LDA)

Doğrusal fark analizi yöntemi de temel bileşen analizi yöntemi gibi boyut indirmeyi amaçlar; temel bileşen analizi yönteminden farkı ise, sadece sınıf içi örneklerle değil, sınıflar arası farklılıkların da analizde kullanıldığı bir metottur. Sınıfları birbiriyle kıyaslama özelliği, doğrusal fark analizi yönteminin başarımının, temel bileşen analizi yönteminden daha iyi olmasını sağlar (W. Zhao vd., 1999). En yaygın ve faydalı doğrusal fark analizi yöntemi Fischer's LDA olarak bilinmektedir. Doğrusal fark analizi yöntemi ise aşağıdaki gibi özetlenebilir .

Aynı hedefe ait resimlerin ortalaması ve bütün resimlerin ortalaması hesaplanır, her sınıfa ait resimlerden kendi sınıf ortalamaları çıkarılır (2.9).

$$\forall x \in X_i, X_i \in X, \bar{X} = X - m_i \quad (2.9)$$

Sınıf ortalamalarından bütün resimlerin ortalaması çıkarılır (2.10).

$$\hat{m} = m_i - m \quad (2.10)$$

Bütün hedef resimleri (ortalamaları çıkarılmış) tek boyuta indirilip birleştirilerek büyük bir matris elde edilir, elde edilen matrisin transpozu ile hedef resim vektörleri çarpılır (2.11).

$$\tilde{x} = U^T \hat{x} \quad (2.11)$$

Elde edilen matrisin transpozu ile Sınıf ortalama vektörleri çarpılır (2.12).

$$\tilde{m}_i = U^T \hat{m}_i \quad (2.12)$$

Hesaplanan kovaryans matrisleri toplanır (2.13).

$$S_i = \sum_{x \in X_i} \tilde{x} \tilde{x}^T \quad (2.13)$$

Bütün matrisler toplanır (2.14).

$$S_w = \sum_{i=1}^c S_i \quad (2.14)$$

Sınıflar arası uzaklık hesaplaması yapılır (2.15).

$$S_B = \sum_{i=1}^c n_i \tilde{m}_i \tilde{m}_i^T \quad (2.15)$$

Özdeğer hesaplaması yapılır (2.16) ve özvektörler üzerine resimler aktarılır.

$$S_i V = \lambda S_w V \quad (2.16)$$

2.3.2.3 Bağımsız Bileşen Analizi (ICA)

Bağımsız parça analizi; amacı bir grup bağımsız değişkeni istatistiksel olarak bağımsız değişkenler halinde ifade etmektir. PCA ikinci dereceden moment kullanarak birbiri ile bağlantısı olmayan bileşenleri kullanırken; ICA olabildiğince bir birinden bağımsız değişkenler halinde ifade edilir (negentropy veya Kurtosis ile non gaussianity'e de bakıyor). PCA+ LDA ya göre PCA + ICA yönteminin daha iyi sonuç verdiği bilinmektedir (T. Bayık, 2004). ICA, lineer bir model olup, $\mathbf{x}$ veri vektörünün bilinmeyen $\mathbf{s}$ kaynaklarıyla ifade edilmesidir (2.17).

$$\mathbf{x} = \mathbf{A}\mathbf{s} \quad (2.17)$$

$\mathbf{A}$ matrisi ise bilinmeyenlerden oluşan katsayı matrisidir; yapılması gereken ise ayırıcı matris olan $\mathbf{W}$ matrisini hesaplamaktır.

$$\mathbf{S}\mathbf{P}\mathbf{s} = \mathbf{W}\mathbf{x} \quad (2.18)$$

Verilen denklemde $\mathbf{S}$ katsayı matrisi ve $\mathbf{P}$ permutasyon matrisidir. $\mathbf{s}$ matrisi ICA yönteminde bağımsız bileşenler olarak ifade edilmelidir ve bağımsız bileşenler iteratif olarak hesaplanır (2.18). Bağımsızlık değerlerinin hesaplanması ise non-gaussianity ölçütü ile yapılır. Non-gaussianity ölçütü olarak Kurtosis eşitliği (2.19) kullanılır.

$$Kurt(y) = E\{y^4\} - 3\left(E\{y^2\}\right)^2 \quad (2.19)$$

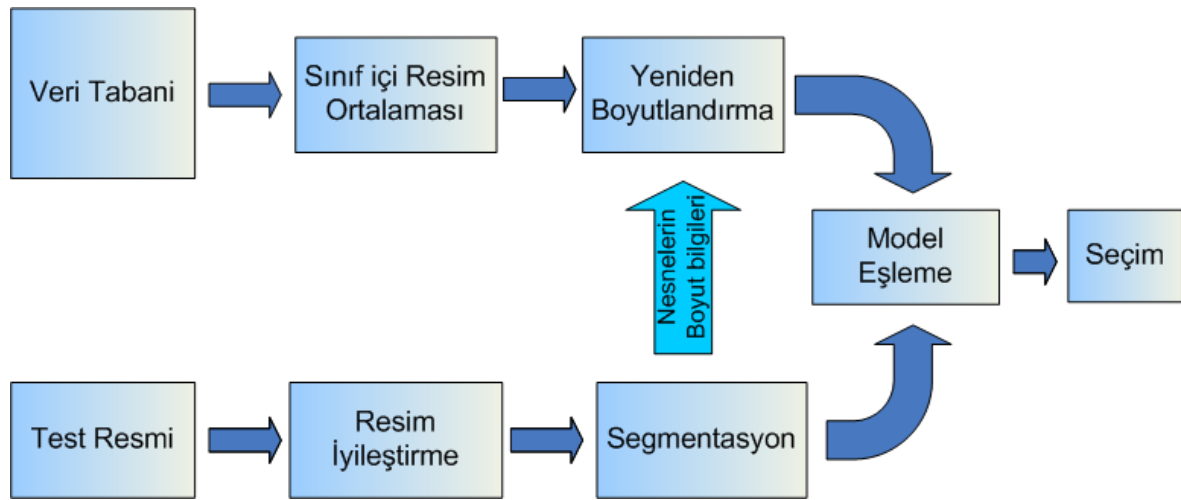
2.3.3 Modele Dayalı Yöntemler

Model tabanlı yöntemler, algılayıcının, görüntü ve hedef görüntüsünün fiziksel özelliklerini kullanır. Model tabanlı metotlarda bir veri tabanı oluşturularak, resmin belirgin özellikleri ile

veri tabanındaki hedef görüntüsü karşılaştırılır (Bhanu ve Holben, 1990). Veri tabanının çok büyük olduğu durumlarda karşılaştırılacak veriyi, belirgin özellikleri koruyarak küçültme yoluna gidilmelidir. Model tabanlı yöntemlerde de farklı yaklaşımlar mevcuttur.

2.3.3.1 Model (Template) Eşleme

Veri tabanında bulunan hedef resimleri ile görüntü üzerindeki nesneler tek tek karşılaştırılır, en iyi sonuç veren nesne hedef olarak belirlenir (Jun vd., 2005). Karşılaştırma işleminde konvolüsyon, korelasyon, vb yöntemler kullanılabilir. Şekil 2.8’de model tabanlı eşleme için örnek bir akış diyagramı verilmiştir.



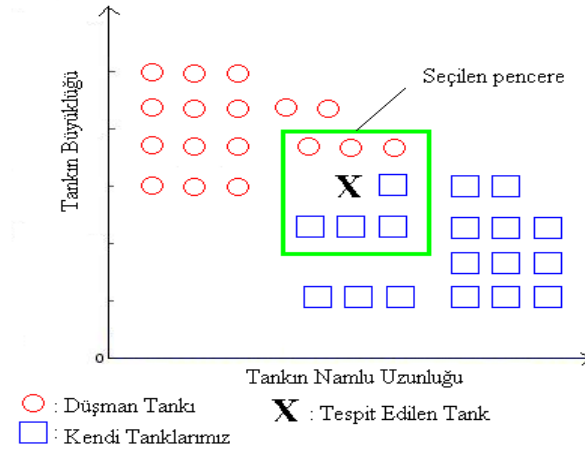
Şekil 2.8 Model Eşleme Akış Modeli

2.3.3.2 Kenar Piksellerini Eşleme

Model eşleme ile kenar pikselleri eşleme arasındaki en büyük fark; kenar eşlemede nesnelerin resminin bütün olarak kullanılması yerine, nesnelerin kenar özelliklerine göre karşılaştırılmasıdır (Olson vd., 1997). Model eşlemeye karşı en büyük avantajı işlem yükünü oldukça düşürmesidir. Geliştirilen ATR yönteminde de tanıma işlemi kenar piksellerini eşleme yöntemine dayanmaktadır; kullanılan yöntemin detayları 4. bölümde anlatılacaktır.

2.3.4 Parzen Penceresi (Parzen Window)

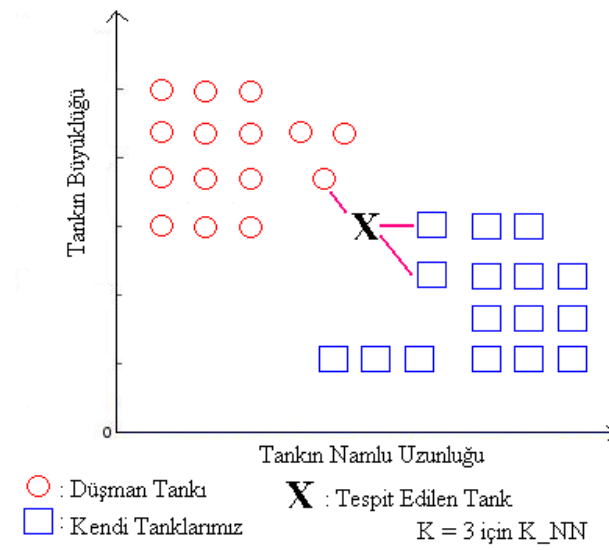
Parzen Pencereleme Yöntemi, belirli büyüklüğe sahip bir pencere bilinmeyen cismi içine alacak şekilde seçilir, hangi gruptan fazla eleman varsa cisim de o gruba aittir.



Şekil 2.9 Parzen Window Uygulaması

2.3.5 En Yakın K Komşu (K_NN: K Nearest Neighbourhood)

En yakın K komşu yönteminde, belirsiz cisim en yakın K objeye bakılır, en yakın K obje hangi gruptan fazla ise cisim de o gruba ilave edilir. K değeri seçilirken iki gruptan eşit sayıda komşu çıkmaması için tek olmasına dikkat edilmelidir. Genellikle işlem yükünü düşük tutmak amacıyla 3 veya 5 komşuya bakılır. Aşağıdaki şekilde de görsel anlatımı yapılmaktadır (Schlosser ve Fecht,2005).



Şekil 2.10 K_NN Metodu Uygulaması

2.3.6 Kovaryans Analizi

Şekillerin içinden hedefin seçilmesinde kovaryans analizi yapılmasının nedeni düşük bir işlem yükü ile gerçekleştirilebilmesidir. Kovaryans analizi, istatistiksel bir yöntem olup aşağıda hesaplanması sırasında kullanılan istatistiksel denklemler açıklanmıştır. Varyans (4.10) ve Standart Sapma (4.9) bir boyutlu hesaplamalar olmasına karşın Kovaryans iki boyutlu bir hesaplama yöntemidir ve resim de yatay ve dikey olmak üzere iki boyutlu bir veri kümesinin dağılımını hesaplamada kullanılabilir.

Varyansın formülü açık yazıldığında;

$$\text{var}(X) = \frac{\sum_{i=1}^n (X_i - \bar{X})(X_i - \bar{X})}{(n-1)} \quad (2.20)$$

Çarpım işleminde tek yönlü veri kümesi yerine iki yön de ilave edilince Kovaryans formülü elde edilmiş olur (2.21).

$$\text{cov}(X, Y) = \frac{\sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y})}{(n-1)} \quad (2.21)$$

Kovaryans yönteminde, resimden çıkarılmış her nesne için kovaryans değeri hesaplanır ve önceden elde edilmiş hedef kovaryansı ile karşılaştırılır minimum fark yaratan nesne ise istenilen hedeftir.

2.3.7 Ayırık Kosinüs Dönüşümü (DCT)

Ayrık Kosinüs dönüşümü ilk olarak, N. Ahmed, T. Natarajan, ve K. R. Rao tarafından 1974 yılında hesaplanmış ve ilgili makale yayımlanmıştır. Ayrık Kosinüs Dönüşümü, önceleri işlem yüküne bağlı olarak, spektrum analizi, ses sıkıştırma, veri iletimi gibi tek boyutlu işaret işleme gerektiren uygulamalarda kullanılmıştır.

Görüntü işleme tekniklerinde ise DCT; JPEG resim formatı ve MPEG Video formatıyla ön plana çıkmış, ilgili donanım mimarileri oluşturulmuş ve yoğun olarak kullanılmaya başlanmıştır. Spektrum analizinde DCT'nin Fourier dönüşümüne göre en büyük avantajı, şüphesiz ki karmaşık sayılar ve hesaplamalar içermeyip tam sayılar kullanılarak hesaplanabilmesidir. Kolay bir matematiksel işlem olarak ifade edilebilen DCT bu özelliğiyle Spektrum hesaplaması yapan donanım tasarlanmasını da kolaylaştırmıştır; bu sayede günümüzde fotoğraf makineleri (MJPEG, JPEG), video kameralar (MPEG2, MJPEG), uydu alıcıları (MPEG2, MPEG4) vb. sayısal görüntü aktarımı ve depolaması yapan bir çok cihazda iki boyutlu DCT(2.22) hesaplayabilen entegreler, MPEG entegresinin içinde blok olarak yer almakta, kullanılmaktadır.

$$B_{pq} = \alpha_p \alpha_q \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} A_{mn} \cos \frac{\pi(2m+1)p}{2M} \cos \frac{\pi(2n+1)q}{2N}, \quad \begin{matrix} 0 \leq p \leq M-1 \\ 0 \leq q \leq N-1 \end{matrix}$$

$$\alpha_p = \begin{cases} 1/\sqrt{M}, & p=0 \\ \sqrt{2/M}, & 1 \leq p \leq M-1 \end{cases} \quad \alpha_q = \begin{cases} 1/\sqrt{N}, & q=0 \\ \sqrt{2/N}, & 1 \leq q \leq N-1 \end{cases} \quad (2.22)$$

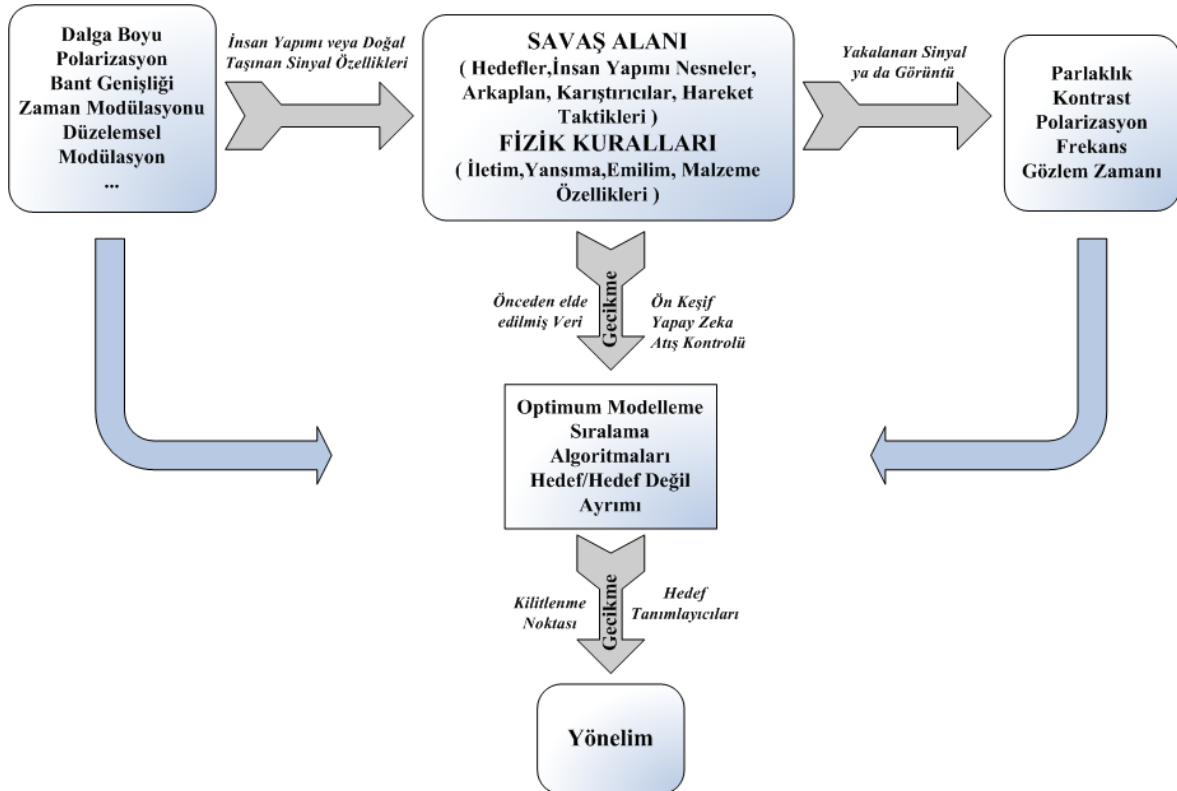
M : Birinci boyuttaki pencere büyüklüğü *n*: ikinci konum bilgisi
N : İkinci boyutaki pencere büyüklüğü *A* : *m,n* konumundaki pikselin renk değeri
m: birinci konum bilgisi *B*: Piksel için hesaplanmış DCT değeri

3. OTOMATİK HEDEF TANIMA

Geliştirilen ATR yöntemini incelemekten önce; ATR kullanımı ve görev planlaması (Bölüm3.1), gereksinimleriyle, görev dağılımı, donanım mimarisi ile ATR Sistemi (Bölüm3.2) ve ihtiyaçlarıyla ATR algoritmasından (Bölüm3.3) bahsedilecektir.

3.1 ATR KULLANIMI

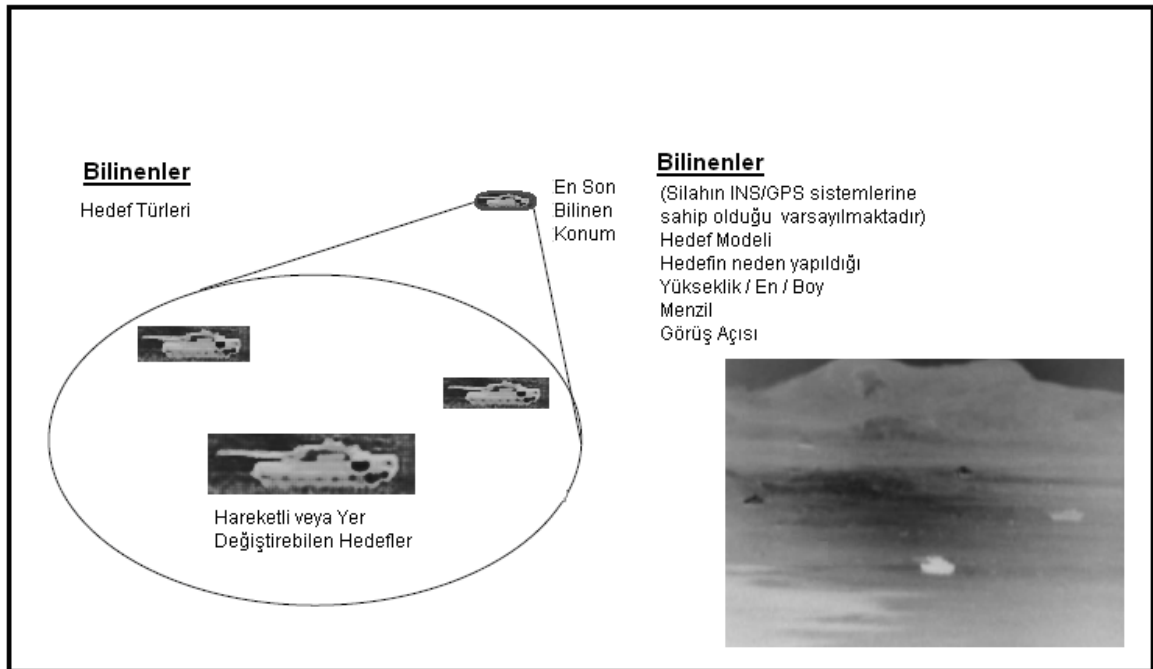
ATR bir bilgisayar,yazılım ve sensörden oluşan, savaş alanında sensörünün önünde ne olduğunu üç temel ölçümle; a) algılanan savaş alanı aydınlanma bilgisi b) savaş alanından yansıyan ve emilen enerji gözlemleri c) savaş alanına ait keşif bilgileri belirleyen alt sistemdir. İnsana benzer bir şekilde, ATR'nin savaş alanı hakkındaki durum ve bilgileri nasıl kullandığı Şekil 3.1'de anlatılmıştır. ATR topladığı bilgiyi anında işleyemediği için bir gecikme oluşmaktadır, bu gecikme miktarı füze için akıllı bir savaş ünitesine göre çok daha kritiktir (Licata, 2001).



Şekil 3.1 Örnek ATR Kullanımı

ATR Görev Planlaması

Görev planlaması ATR'nin başarımı açısından oldukça önemlidir; çünkü görev sırasında ATR'ye herhangi bir müdahale yapılamayacaktır. Geçmişte denenen metotlardan biri hedefe ait tek resim kullanmaktır. Şekil 3.2'de de görüldüğü gibi; sabit bir hedef için, saldırı yönü önceden elde edilmiş hedef görüntüsüyle aynı ise, tek hedef resmi bile yeterli olacaktır (Licata, 2001). Gemi, kamyon gibi hareketli araçlarda ise hem hedefe ait daha geniş veri tabanına sahip olmalıdır hem de hatalara karşı daha dayanıklı olmalı; görev planlaması da çok daha dikkatli yapılmalıdır. Hedefin hareketli olması, kamuflaj ve zırh bulundurması, binaların veya ağaçların arasına saklanması işlemleri çok daha zor bir hale getirmektedir.



Şekil 3.2 Görev Planlaması

ATR sistemini kullanmak ve satmak isteyen bir firmanın (Bölüm 3.2.1) görev gereksinimlerini de tanımlanması gerekir.

3.2 ATR SİSTEMİ

ATR sadece görüntüleme sensörüyle bağlantısı bulunan kendi başına bir alt sistemdir. ATR için olabilecek en iyi görüntü temel ihtiyaçtır; bu da sistem maliyetini önemli ölçüde yükseltecektir.

ATR sistemi bir alıcıya satılacağında sistemin performansını belirleyen en önemli kriterler doğru hedef algılama olasılığı ve yanlış alarm oranı (false alarm rate) veya yanlış alarm yoğunluğudur. Bir bilgi kullanıcısı olarak ATR Sistemi'nin doğru hedef algılama oranının yüksek olması, ancak ve ancak yeterince kaliteli bir sensörle, yeterince kaliteli görüntü elde edilmesine bağlıdır. Bu noktada da doğru hedef algılama olasılığının hiçbir zaman bir olmayacağı ve hiçbir sensörün her koşulda istenilen kalitede görüntü üretmesinin mümkün olmayacağı unutulmamalıdır.

3.2.1 Gereksinimler

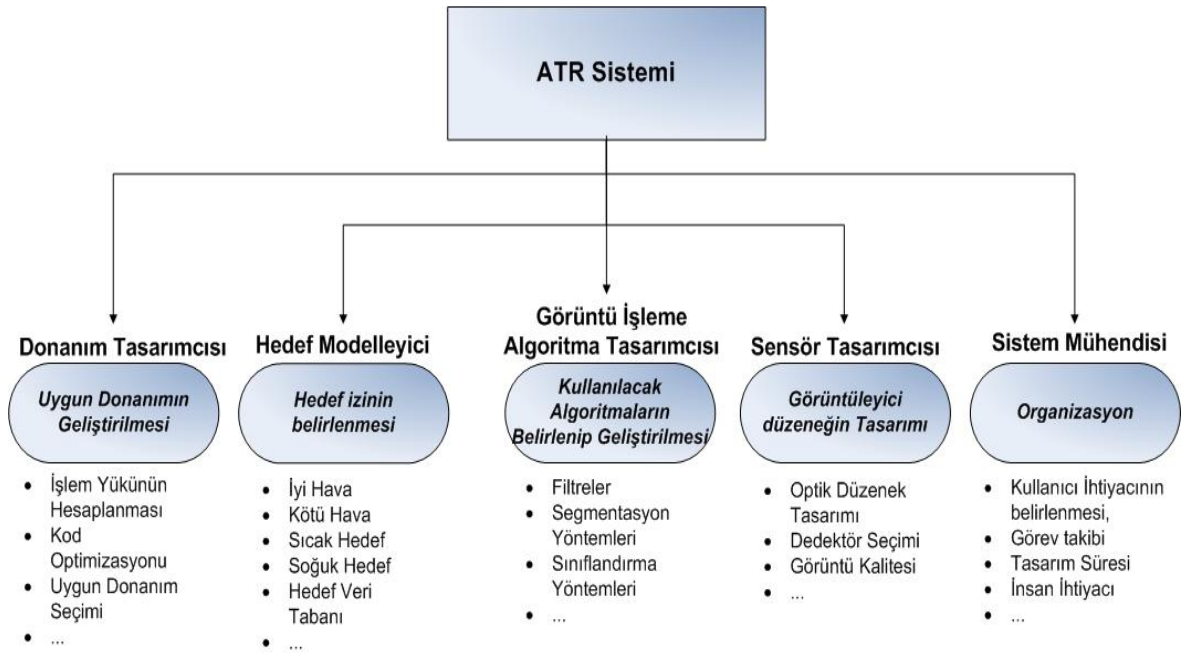
ATR Sistemi'nin ihtiyaçları, doğru hedef algılama olasılığından ve yanlış alarm oranından daha fazlasını tanımlamalıdır:

1. ATR Sistemi'nin girdileri
2. Kullanım bölgesi keşif bilgileri
3. Tanınması gereken hedeflerin sayısı ve türleri
4. Üretilen bilginin kalitesi
5. Sensör performansı
6. Ana sistem bilgisayarı (Füze güdüm bilgisayarı) çıktıları
7. Savaş alanında işlenebilecek maksimum hedef sayısı
8. Tasarım uzayındaki hedeflerin türlerinin belirlenmesi
9. Hedef ölçümlerinin doğruluğu (en,boy,alan,açı)
10. Ana yazılım için işlem yükünü kaldırabilecek bilgisayar
11. Programlama dili
12. Kabul testlerinin belirlenmesi

3.2.2 Görev Dağılımı

ATR'nin teknolojiye ürüne nasıl alt sistemlere ayrılacağı Şekil 3.3'te gösterilmektedir. ATR'yi alt sistemlere ayırarak incelemek teknolojik başarıyı beraberinde getirir de, piyasada değer elde edebilecek bir hızla ürünü piyasaya taşımayı zorlaştıracaktır. Algoritma tasarımcısı

her zaman algoritma üzerinde ayarlar yapmak, basit algoritmaları daha gelişmiş ve kompleks algoritmalarla değiştirerek başarımlarını artırmak isteyecektir. Hedef modelleyicisi, sürekli daha iyi modeller oluşturabilir, sensör tasarımcısı ise sensör performansını artırabilir. Bu alt sistemleri bütünleştirecek ve ürünü tanımlayabilecek anahtar kişi ise ATR Sistem Mühendisidir (Licata, 2001).

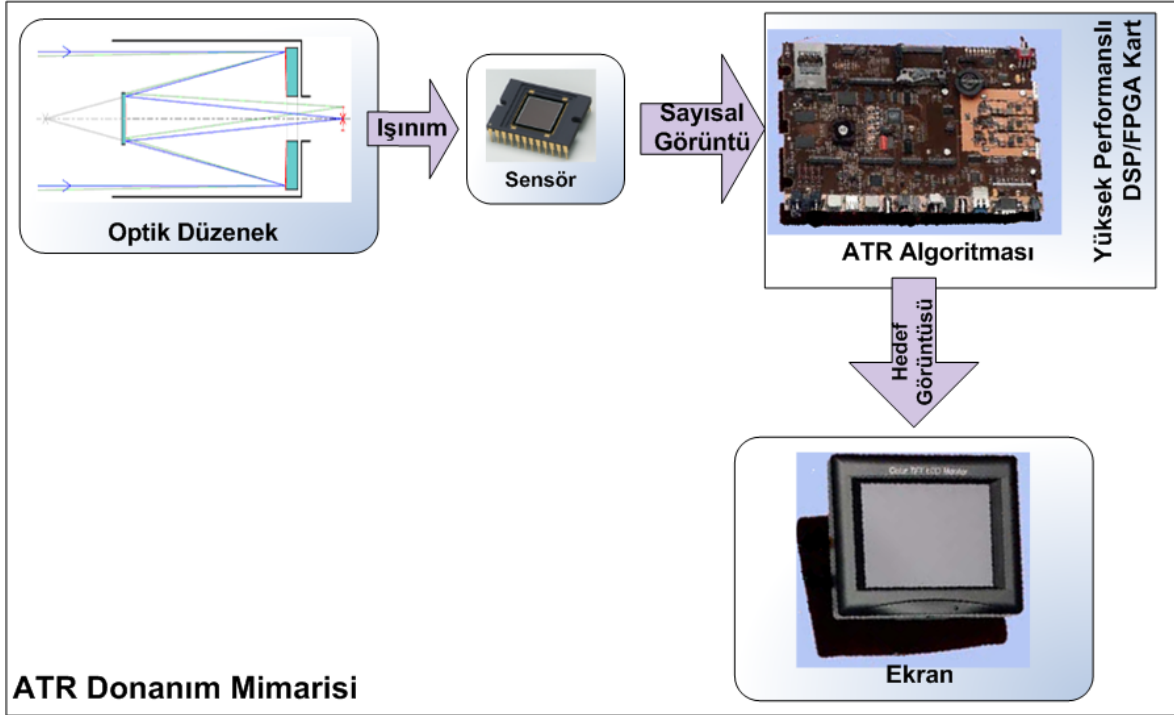


Şekil 3.3 ATR Sistemi Görev Dağılımı

ATR Sistem Mühendisi, sistem ihtiyaçlarını, bu ihtiyaçları karşılayabilecek ürün özelliklerini, kabul testlerini ve fiyat ürün endekslemesini yaparak teknolojiye ürüne geçişi sağlar. ATR'nin ürün olarak kullanılabilmesi için, istenilen hacim sınırlamasına uyan, yazılımı gerçek zamanlı koşturabilen bir işlemciye ihtiyaç vardır.

3.2.3 Donanım Mimarisi

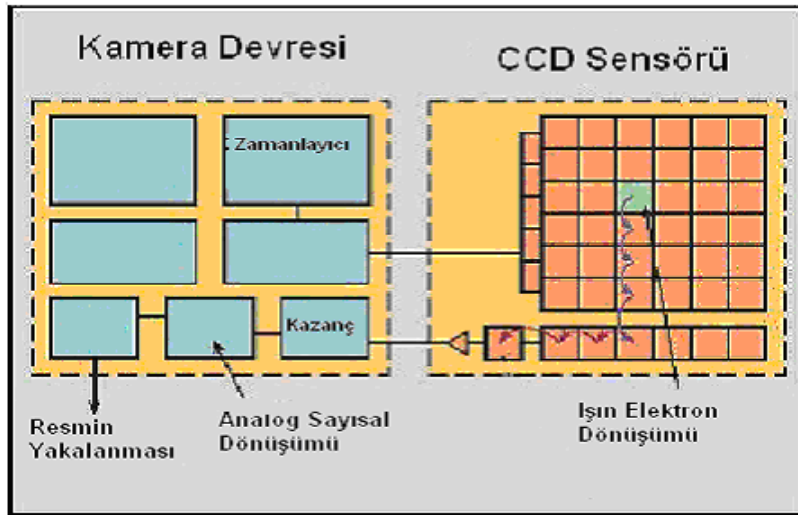
ATR algoritmasının kullanılacağı düzenek 3 ana kısımdan oluşur. Bunlar; optik düzenek, sensör ve elektronik karttır. Optik düzenek, istenilen bakış açısındaki ışığı sensörün üzerine düşürür, düzlem üzerindeki detektörler gelen ışığı algılar ve ATR algoritması da elektronik kart üzerinde koşturulur.



Şekil 3.4 Örnek ATR Donanım Mimarisi

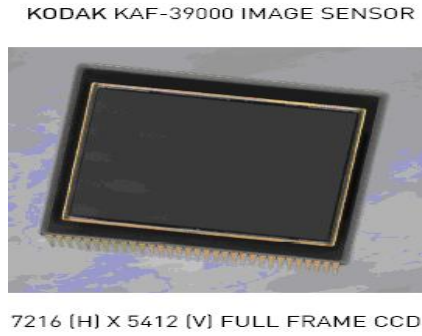
3.2.3.1 Sensörler

Kullanılabilecek sensörlere ilişkin en güzel örneklerden biri CCD sensördür. İki boyutlu bir matris şeklinde yerleştirilmiş foto diyotlardan oluşur. Işık sensör üzerine düşmesinden sonra her foto diyotun üzerine düşen yük, potansiyel farka dönüştürülür. Bu yüzden de CCD mimarisi sıralı ve düzgün dağılımlı bir görüntü verisi sağlar.



Şekil 3.5 CCD Sensör Çalışma Prensibi

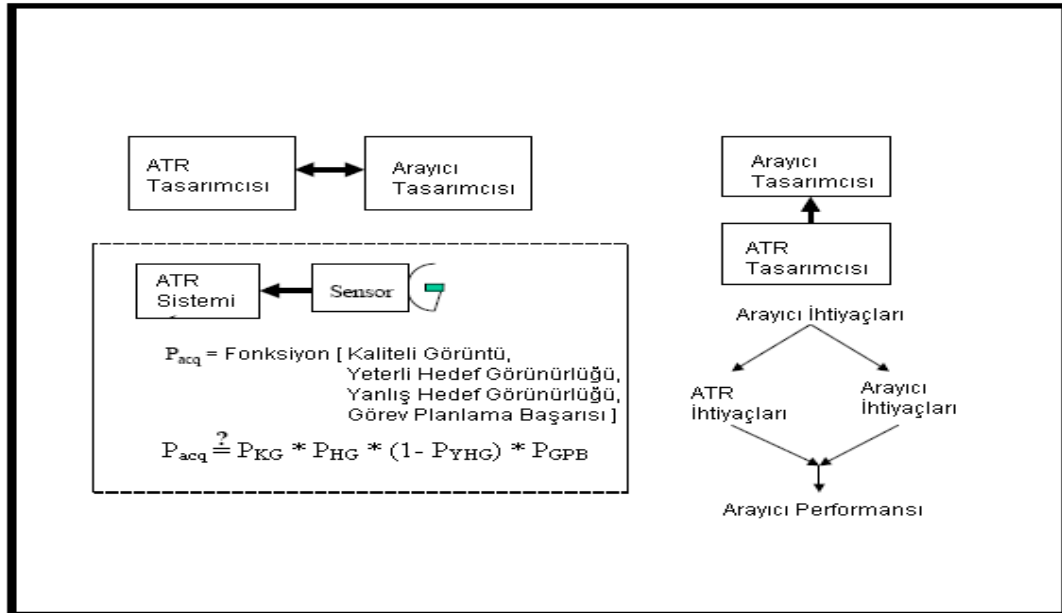
Saniyede alınan resim sayısı çok artırılırsa çözünürlük düşer, çözünürlük çok artırılırsa da saniyede alınan resim sayısı yetersiz kalabilir. Tabi bu oran değişikliği de sensör okuma devresinin limitleri ile sınırlıdır. Şekil 3.5’de de CCD Kameranın şematik anlatımı yapılmaktadır, Şekil 3.6 'da ise Şekil 3.5'de sağ tarafta yapısı anlatılan sensöre bir örnek verilmiştir. CCD kameralar için en önemli parametreler detektör büyüklüğü, piksel bazında çözünürlük ve duyarlılıktır (algılanabilen minimum ışık miktarı).



Şekil 3.6 CCD Görüntü Sensörü Örneği (Kodak)

Sensör ve ATR Sistemlerinin Ayır Geliştirilebilmesi

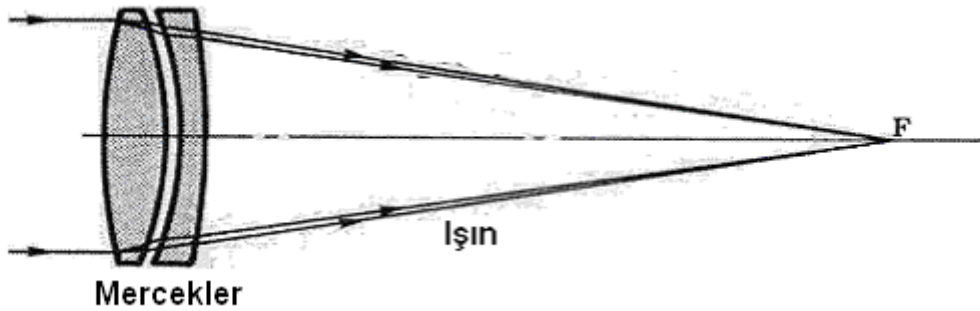
Arayıcı tasarımında sensör tasarımının ve ATR Sistemi’nin ayrı yapılabilmesi Şekil 3.7’den de anlaşılacağı gibi pek mümkün görünmemektedir. İki dizayn arasında çok fazla ortak parametre olduğundan dizaynların ayrı yapılması pek olası değildir.



Şekil 3.7 Sensör – ATR ilişkisi (Licata, 2001)

Kırıcı Düzenekler

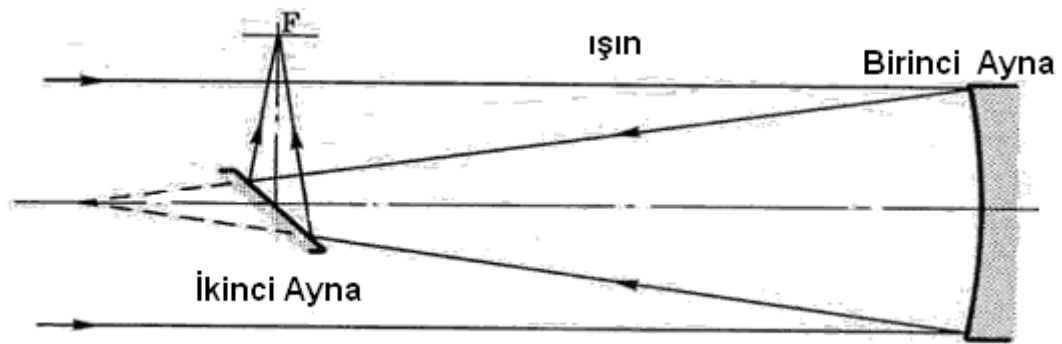
Kırıcı optik düzeneklerin mercek çeşidine ve kullanılış şekline göre çok değişik versiyonları oluşmuştur. Şekil 3.9' da görülen ince ve kalın kenarlı merceklerden oluşmuş bir kırıcı optik düzendir. Şekilden de anlaşıldığı gibi gerçekleşmesi de kolay olmasına karşın; gerek kızıl ötesinin farklı dalga boylarını farklı açılarla kırmasından, gerek odağa düşen görüntünün farklı açılardan gelmesiyle bozulmasından, gerekse de istenilen odaklanma ve görüş açısı için odak uzaklığının fazla olmasından bir arayıcı için avantajlı değildir.



Şekil 3.9 Kırıcı Optik düzeneği

Yansıtıcı Düzenekler

Yansıtıcı aynalardan oluşan optik düzenekler ise ışığı yansıttığı için farklı dalga boylarında aynı açıyla yansıtılma ve bozulmaya sebebiyet vermeme avantajlarına sahiptir. Aynalarla oluşturulan sistem örnekleri aşağıda açıklanmıştır. Şekil 3.10'da örnek bir aynalı optik düzenek gösterilmiştir.



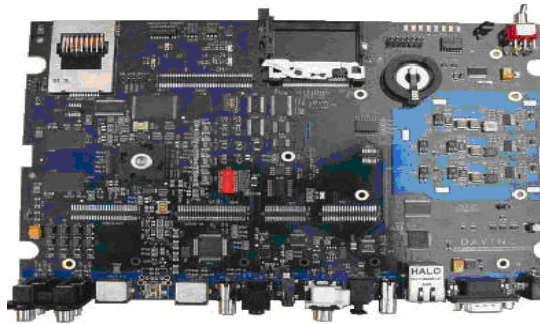
Şekil 3.10 Yansıtıcı Optik Düzeneği

3.2.3.3 Dijital İşaret İşleme Kartı

ATR donanım mimarisinde belki de en önemli bileşenlerden biri algoritmanın üzerinde koştuğu, çok yüksek işlem gücüne sahip elektronik karttır. Algoritmaların yüksek işlemci gücüne ihtiyaç duymaları; kullanılacak olan algoritmaların gerçek zamanlı bir kart üzerinde denenmesi ihtiyacını da beraberinde getirmektedir. Kartın aynı zamanda da ekrana görüntü ve veri aktarması gözlemleri ve algoritma performansı ölçümlerini oldukça kolaylaştıracaktır. Bu yüzden de hem yüksek işlem gücüne sahip, hem de sensör, kamera ve ekran ara yüzlerini aynı anda ve kolay bir şekilde destekleyebilen bir elektronik kart ihtiyacı ortaya çıkmaktadır.

Kabaca bir hesap yapmak gerekirse, 640 x 512 piksel bir görüntü sensöründe oluşan renkli bir resim için (RGB) 3x 640 x 512 'lik bir matris oluşacaktır. Görüntüleyicilerinin saniyede 50 resim ürettiğini varsayarsak kartın işlem yükünün sadece bütün resimleri okuyabilmek için bile 50 x 3 x 640 x 512 byte'ı işleyebilmesi gerekmektedir. Bu da 375 Mbps'dır ve bu resim üzerinde yapılacak en kolay görüntü işlemi (örnek: histogram elde etmek) bile yaklaşık 10 kat işlem yükü getirecektir ki 3750 Mbps eder. Görüntü işleme kartlarındaki yüksek işlemci ihtiyaçları da bu nedenle ortaya çıkmaktadır.

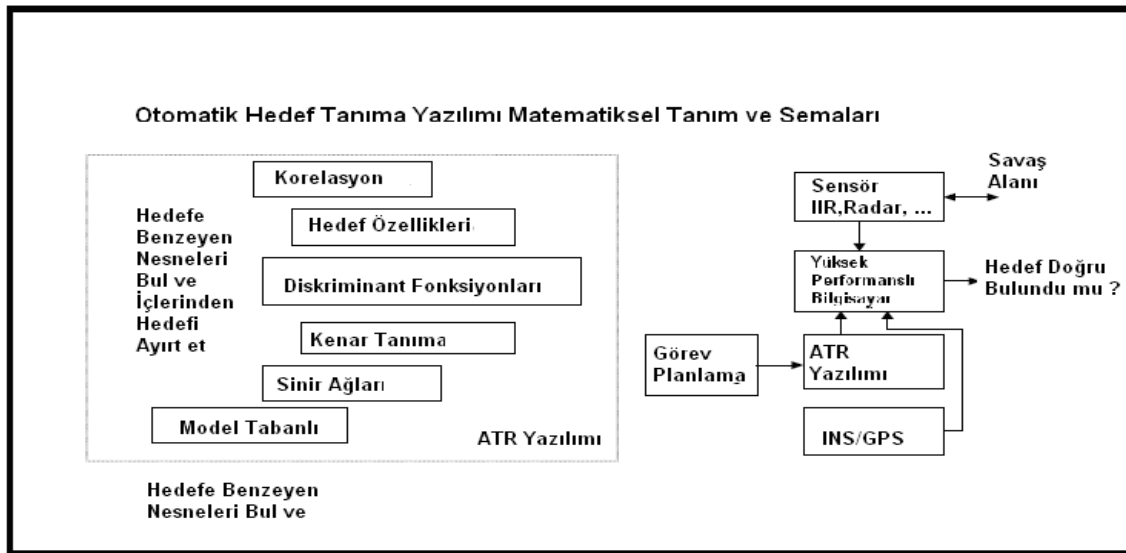
Örnek olarak, Texas Instruments Firması'nın TMS320DM6446 işlemciye sahip, Şekil 3.11' daki TMDXEVM6446 elektronik kartı istenilen çalışmaların yapılabileceği bir ortam sağlamaktadır. Kartın USB 2.0 ve PAL bağlantılarını ihtiva etmesi, karta sayısal ve analog görüntü sağlayan sensörler ile çalışabilme olanağını sağlamaktadır. Yine çift çekirdekli işlemcinin(C64X 594 Mhz ve ARM 297 MHZ) yeterli bir işlem gücü sağladığı değerlendirilmektedir. Kartın herhangi bir dönüştürücü ya da ek kart (Daughter Board) ihtiyacının olmamasının da kart ile donanımlar arasındaki uyum problemlerini minimize ettiği değerlendirilmektedir.



Şekil 3.11 (Texas Instruments) TMDXEVM6446

3.3 ATR ALGORİTMASI

ATR çalışmalarının en önemli kısmı, şüphesiz ki geliştirilen algoritmalar, kullanılan algoritma elde edilen veriyi ne kadar verimli işleyebilirse; ATR sisteminin başarımı da o oranda artacaktır. Algoritmayı yazılım dizaynının matematiği ya da yol haritası olarak tanımlayabiliriz. Şekil 3.12’de görüldüğü gibi, ATR için kullanılabilecek bir çok algoritma yaklaşımı mevcuttur, bir de bu algoritmaların karışık kullanımları düşünüldüğünde algoritma alternatifleri oldukça çoğalmaktadır. Algoritmalar özellikleriyle belirli sınıflara ayrılabilen, bu gruplar da akademik disiplinde sinir ağları, bilgisayar bilimi, matematik mühendisliği dallarının içinde kalmaktadır. Uygun görülen yöntemler ve bu yöntemlerin işleyişleri Bölüm 4’de uygulamalarıyla birlikte anlatılacaktır. ATR algoritma çalışmalarında ortaya atılan yeni programlama dilleri, işlemciler bile vardır. ATR algoritmalarını donanımsal olarak gerçekleyebilecek sistemlerden de bahsedilecek ve örnekler verilecektir.

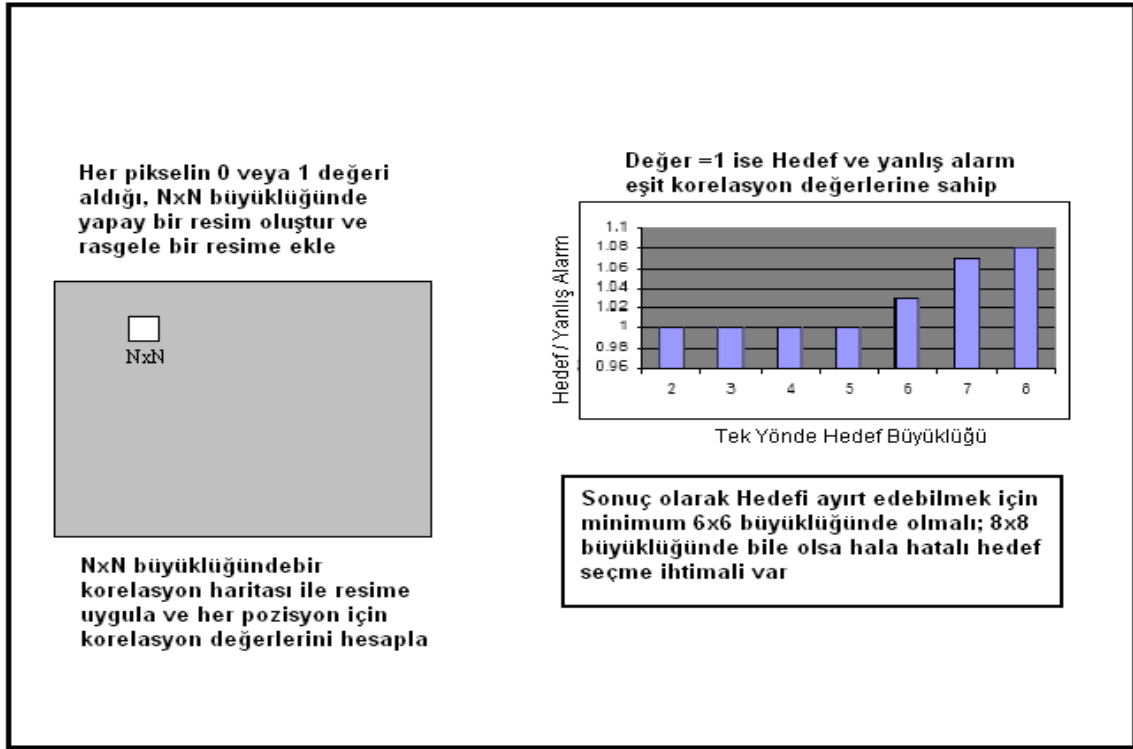


Şekil 3.12 ATR Algoritmaları (Licata, 2001)

Uygulanabilecek ATR algoritması elde edilen hedef bilgisiyle kısıtlanmaktadır; sensör türüne ve hedef çözünürlüğüne bağlı olarak, hedefe ait ne kadar fazla bilgi elde edilebilirse, tanımak ve kompleks algoritmalar uygulayabilmek de o denli mümkün olmaktadır. Eğer hedef savaş alanındaki en parlak nesne ise basit bir eşik değer atama (threshold) yeterli olacaktır, eğer hedef savaş alanında zor ayırt edilebilecek bir parlaklığa sahipse; hedefi ayırt etmek için basit bir eşik değer atama işleminden çok daha kompleks algoritmalar, hareket yönünü araç hızını araç örüntüsünü, önceden hazırlanmış veri tabanını kullanabilen, gerekebilecektir.

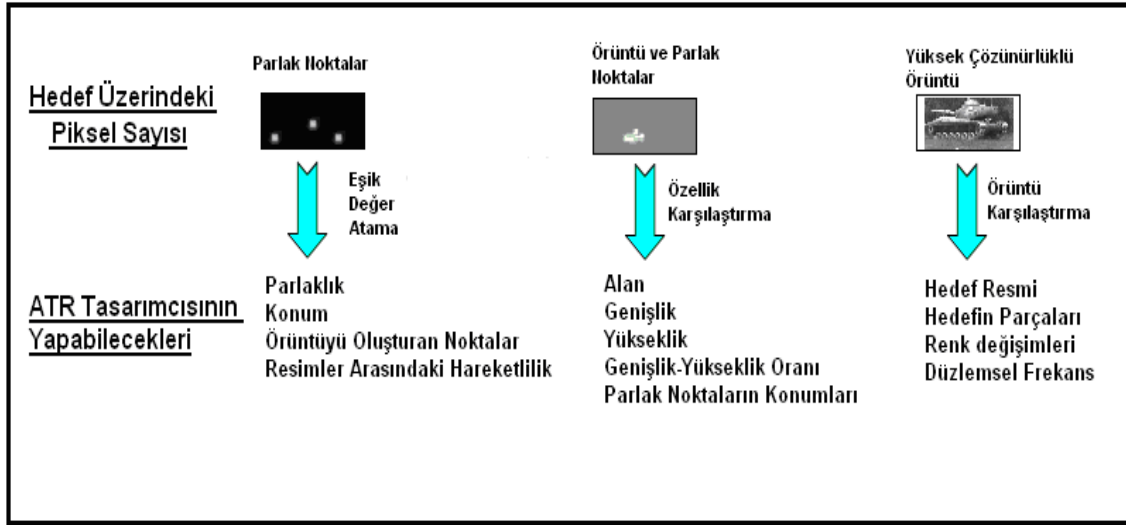
3.3.1 Hedef Üzerindeki Piksel Sayısı

ATR Sisteminin başarılı olabilmesi için en temel ihtiyacı şüphesiz ki hedefin sensör üzerinde ihtiyaç duyulan miktarda pikselle temsil edilebilmesidir. Sistem tasarlanırken kritik piksel sayısını belirleyebilmek için aşağıda anlatılacak olan “Johnson Kriteri” kullanılırken geliştirilen sistemi test etmek için Şekil 3.13’de anlatılan yöntem kullanılabilir.



Şekil 3.13 Hedef Üzerindeki Piksel Sayısı (Licata, 2001)

Hedef çözünürlüğüne ilişkin ATR tasarımcısının yapabilecekleri Şekil 3.14’de gösterilmiştir, yine Şekil 3.14’de görsel olarak verilen tanımlamaların sayısal hesaplamaları aşağıda anlatılacaktır. Hedef tek bir nokta bile olsa ayırt edilebilir; fakat tespit edilenin hedef mi, sensörden kaynaklanan gürültü mü olduğunu algılayabilmek özellikle uzak mesafeler için, değişken hava koşullarında mümkün olmayacaktır ve hata oranı çok artacaktır. Hedef noktadan büyük, örüntüsü tanınamayacak kadar küçükse, damla tanıma yöntemleri, en boy oranı, alan doluluğu gibi, kullanılarak bir sonuç elde edilebilir, ama nesneyi sınıflandırmak bu kadarıyla da mümkün olmayacaktır.



Şekil 3.14 Çözünürlüğe bağlı ATR seviyeleri

Synthetic Aperture Radar (SAR), Imaging Infra Red (IIR) gibi yüksek çözünürlüklü ve resim yakalama süresi kısa sensörlerin gelişimiyle birlikte insanların ve gelişmiş bilgisayarların rahatça tanımlayabileceği kaliteye sahip hedef görüntüleri elde edilmeye başlanmıştır.

Johnson Kriteri

İkinci Dünya Savaşı sırasında, Amerikan Gece Görüş Laboratuvarı (United States Army Night Vision Lab) görsel hedefler için bir ölçüm tekniği geliştirme çalışmalarına başladı. 1950 yılında Gece Görüş Sistemlerinin performansında da büyük gelişmeler oldu. 1957-1958 yıllarında şimdiki adı “Night Vision & Electronic Sensors Directorate” olan Gece Görüş Laboratuvarı bilim adamı, John Johnson, Hedef Tespit, Oryantasyon, Tanıma, Tanımlama kavramlarını hesaplayabilecek bir metot geliştirmeye çalışıyordu. Gönüllü gözlemcilerle yaptığı denemelerde, farklı şartlar altında görüntü yoğunluğunu kullanarak gözlemcilerin model hedefleri algılama kabiliyetlerini ölçtü. Yaptığı bu deneylerle, çizgi çiftleri cinsinden görsel eşik seviyelerini hesapladı. O zamanlarda bir koyu bir açık çizgi olan ölçütün sayısal değeri iki pikseldir.

Kasım 1958’de İlk “Night Vision Image Intensifier” Sempozyumu düzenlendi ve Bay Johnson yaptığı çalışmayı ve makaleyi (Analysis of Image Forming Systems) sundu. Yazdığı makale gözlemcilerin görsel performansını aktarabilmek için hem görüntü hem de frekans domeninde yapılan analizleri içeriyordu. Sonraları Bay Johnson’un Kriteri ve buldukları farklı dalga boylarında çalışan görüntüleme sistemlerinin performanslarını ölçmede ve yeni

sistemler tasarlamada temel olarak kabul edildi. Johnson Kriteri kullanılarak bir çok görsel model ve sensör tasarımı farklı şartlar altında çalışacak şekilde geliştirildi (Robert, 2005).

Kullanılan kavramlar :

Tespit: Nesnenin olup olmadığını algılayabilmek.

Oryantasyon: Simetrik,asimetrik, yatay veya dikey konumda olduğunu anlayabilmek

Tanıma: Nesnenin türünü belirleyebilmek; araba,insan vb..

Tanımlama: Nesnenin özelliklerini algılayabilmek; insan,erkek araba,modeli gibi.

Johnson Kriterine göre, nesne üzerine minimum Tablo 3.1'deki kadar piksel düşmesi gerekmektedir. (Burada verilen hesaplamalar %50 olasılık üzerinden hesaplanmıştır.)

Tablo 3.1 Johnson Kriteri Piksel Sayıları

Tespit %50	Oryantasyon %50	Tanıma %50	Tanımlama %50
2 +1/-0.5	2.8 +0.8/-0.4	8 +1.6/-0.4	12.8 +3.2/-2.8

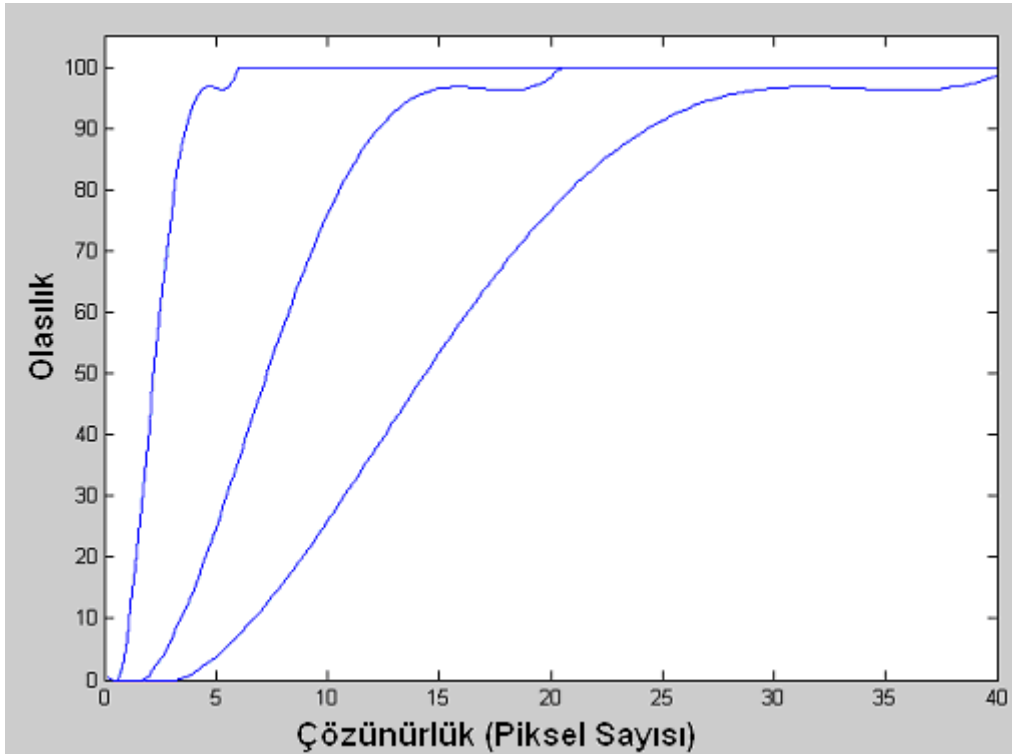
Yukarıda da verilen piksel değerleri, nesne olarak araba kabul edildiğinden ve %50 olasılığı temel aldığından genişletilme ihtiyacı gerekmiştir. Tablo 3.2 ve Şekil 3.15'de verilen tablolardan elde edilen değerlerle 4.derece bir denklem (2) hesaplanarak farklı olasılık değerlerine karşılık gelen piksel sayıları hesaplanmıştır.

Hesaplanan değerler ATR Algoritmasında minimum hedef büyüklüğü olarak kabul edildiğinden önemlidir. Yapılan çalışmada tanıma için %95 olasılık hedeflenmiş ve hesaplanan 10 piksel değerinden küçük nesneler gürültü olarak düşünülmüştür.

Tablo 3.2 Çözünürlük Değerleri

Hedef	Minimum Boyut için Çözünürlük Değeri			
	Tespit	Oryantasyon	Tanıma	Tespit
Kamyon	1.8	2.5	9.0	16.0
M-48 Tankı	1.5	2.4	7.0	14.0
Stalin Tankı	1.5	2.4	6.6	12.0
Centurion Tankı	1.5	2.4	7.0	12.0
Kamyonet	2.0	3.0	8.0	10.0
Cip	2.4	3.0	9.0	11.0
Araba	2.4	3.0	8.6	11.0
Asker	3.0	3.6	7.6	16.0
Howitzer	2.0	3.0	9.6	12.0
Ortalama	2.0	2.8	8.0	12.8

Hesaplama yöntemi incelenecek olursa; Tablo 3.2'den araç tipine göre %50 tespit için kaç piksel gerektiği öğrenilir (kamyon için 1.8 piksel), hedefin üzerine düşeceği varsayılan piksel sayısı ile oranlanarak Şekil 3.15'de verilen katsayıya ulaşılır.



Şekil 3.15 Çözünürlük-Olasılık Eğrileri

$$c = \frac{\text{Hedef Üzerindeki Piksel Sayısı}}{\%50 \text{ Tespit için gerekli Piksel Sayısı}} \quad (4.1)$$

C → Katsayı

$$olasılık = (11.2 \times C^4) - (73.3 \times C^3) + (140.5 \times C^2) - (29.7 \times C) + 0.3 \quad (4.2)$$

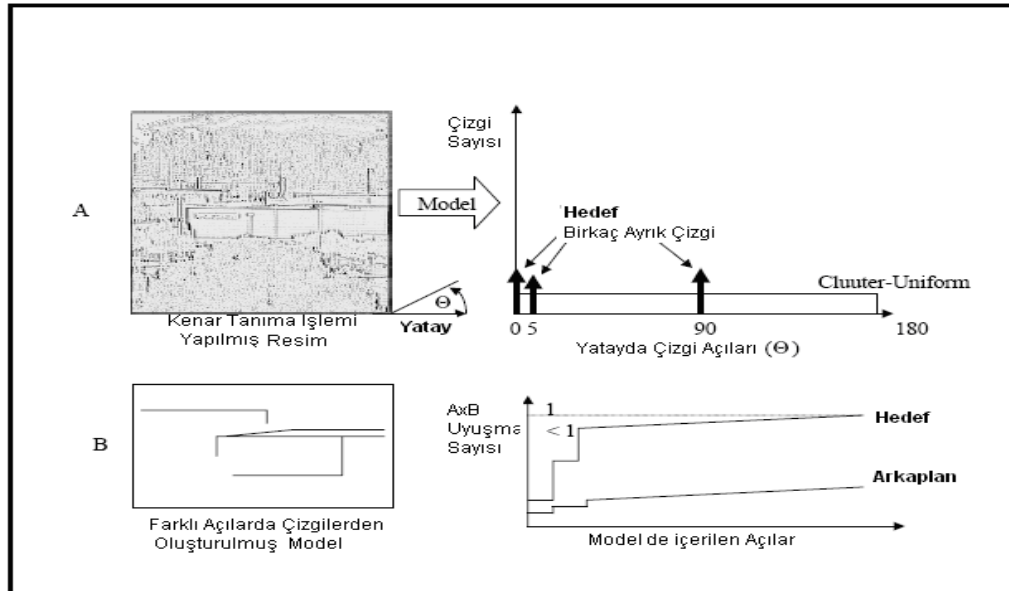
Denkleminde yerine koyulursa yüzde olarak hedef algılama olasılığı hesaplanmış olur.

Örneğin resimdeki kamyonu, üzerine düşen piksel sayısına göre yüzde kaç ihtimalle tanıyabileceğimizi merak ediyoruz; kamyonu %50 ihtimalle tanımak için Tablo 3.2’de ‘9’ piksel değeri verilmiş, resimde yer alan nesne üzerine 19 piksel düştüğü sayılmış olsun birinci işlemde: $C=19/9 \rightarrow C=2,111$ olarak katsayı hesaplanır.

İkinci denklemde yerine koyulursa; olasılık %96 olarak hesaplanır. Yani Johnson Kriterine göre, kullanılan görüntüleme sisteminde hedefin üzerine minimum 19 piksel düşerse kamyon %96 ihtimalle tanınacaktır.

3.3.2 Veri Tabanı Kompleksliği

ATR Sistemi için bir diğer önemli kriter de hedef karşılaştırması için kullanılacak örüntüler, örüntülerin çözünürlük, büyüklük gibi özellikleridir. Şekil 3.16’de açıklayıcı bir çizim verilmiştir. Kullanılacak örüntünün çözünürlüğünün artırıldığı oranda doğru hedef algılama oranının artmadığı görülmektedir. Özellikle kamuflaj tekniklerinin de kullanılması sıfır hata olasılığını ortadan kaldırmaktadır.

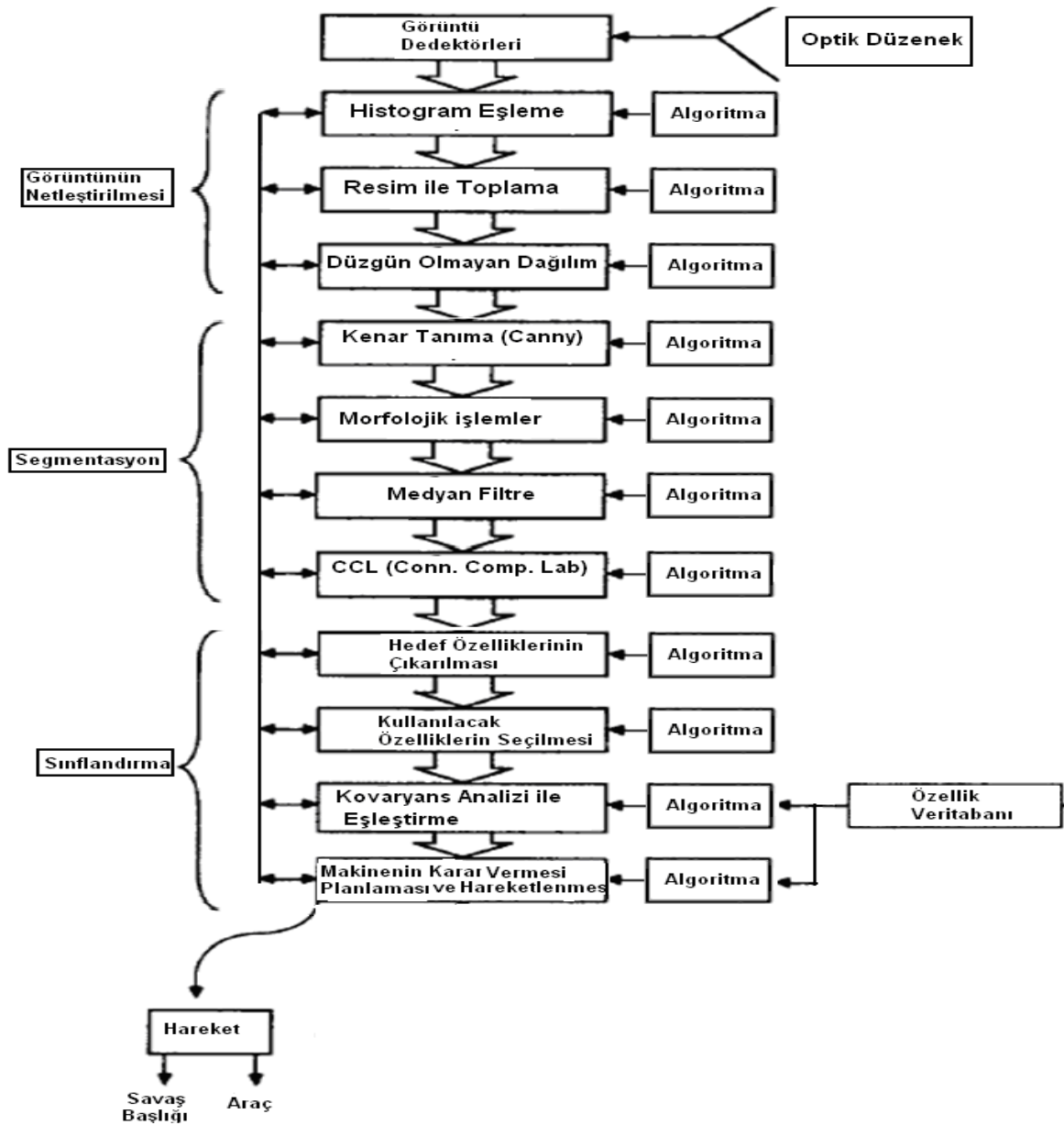


Şekil 3.16 Model Kompleksliği (Licata,2001)

4. GELİŞTİRİLEN ATR YÖNTEMİ

Uygulanan ATR Metodu, Görüntü iyileştirme (Bölüm 4.1), Görüntü Bölümleme (Bölüm 4.2) ve Sınıflandırma (Bölüm 4.3) olmak üzere üç ana bölümden oluşmaktadır.

Algoritmanın belirtilen işlemleri gerçekleştirme sırası ve işlem detayları Şekil 4.1’de detaylı olarak görülmektedir (Brown vd., 1989). 3 . bölüm boyunca, Şekil 4.1’de yer alan Akış diyagramına uygun olarak, kullanılan görüntü işleme teknikleri test resimleri üzerinden açıklanacaktır.



Şekil 4.1 ATR Algoritma Akış Diagramı



1. Test Resmi



2. Test Resmi



3. Test Resmi



4. Test Resmi

Şekil 4.2 Test Resimleri

Uygulama için kullanılacak örnek tank resimleri Şekil 4.2’de görülmektedir. Test resimleri rasgele seçilmiş resimler olup çeşitli makalelerden ve İnternet kaynaklarından tedarik edilmiştir. Resimlerin ortak özellikleri düşük kontrasta sahip, termal kamera görüntüleri olmalarıdır.

4.1 GÖRÜNTÜ İYİLEŞTİRME

Yapılan analizlerde ortalama varyans ve gerek duyulduğunda da Kurtosis algoritmasından faydalanılmıştır. Amaç geliştirilen algoritmayı gerçek zamanlı olarak kullanmak ve farklı koşullara karşı dayanıklı bir algoritma hazırlamak olduğundan Görüntü içerisinde iyileştirme yapılmadan çalıştırılan bölümlleme algoritmalarının, daha kötü sonuç verdiği hem yapılan analizlerle hem de görsel olarak anlaşılmaktadır.

Algılayıcıdan gelen görüntü üzerinde yapılacak görüntü iyileştirme işlemlerinin resimdeki bölümlmeyi kolaylaştıracağı değerlendirilmiştir. Ancak ATR algoritmasındaki görüntü iyileştirme algoritmaları basit bir eşik değer atama (Thresholding) ile yapılamayacağı; yapılsa da hedef renginin ısıya farklı değişiklik göstereceği gerçek zamanlı bir çok durumda özellikle de günün farklı saatlerinde arka plana göre koyu belli saatlerinde de arka plana göre daha açık renkli bir görünüme sahip hedeflerde işe yaramayacağı değerlendirilmektedir. ATR algoritmasının görüntü iyileştirme bölümünde kullanabilecek bir çok algoritma hem görsel olarak hem de analizlerle incelenmiştir.

K-Means, Iso DATA gibi bir çok gruplandırma algoritmalarında; temel olarak sınıf içi ortalama değerleri ve varyansları kullanılır. İteratif olarak, sınıf içi varyans azalacak ve sınıf ortalamaları arasındaki fark artacak şekilde gruplandırma devam ettirilir. ATR'de ise hedef ve arka plan birer grup olarak düşünülebileceği ve hedef varyansını küçülten; hedef renk ortalaması ile arka plan renk ortalaması arasındaki farkı artıran algoritmaların bölümlmeyi kolaylaştıracağı değerlendirilmektedir.

4.1.1 Ön Filtreleme

Algılayıcı görüntüsü üzerinde oluşabilecek “tuz ve biber ” gibi gürültülerden kurtulabilmek için ön filtrelemeye ihtiyaç vardır. İncelenen algoritmalar içerisinde, ön filtreleme için ortalama filtre, medyan filtre, gaussian filtre, bant geçiren filtre değerlendirilmiştir.



Tuz biber gürültüsü eklenmiş resim



Filtrelenmiş resim

Şekil 4.3 Medyan filtre uygulaması

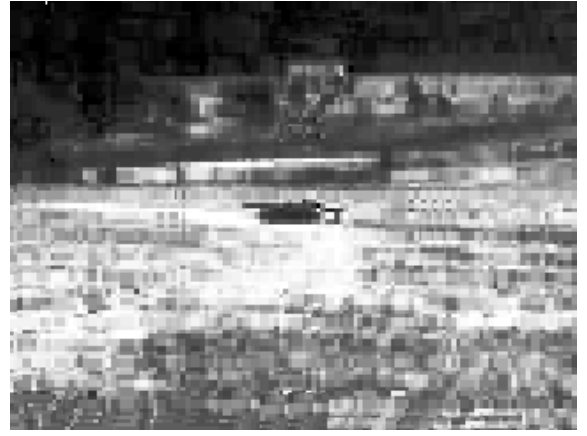
Ön filtrelemede kullanılabilecek yöntemler incelendiğinde, Ortalama ve Gaussian filtrelerin nesnelerin kenarlarına zarar verebileceği ve kenar tanıma işlemini olumsuz etkileyebileceğinden; yüksek frekanslı gürültülerden şablonlara zarar vermeden kurtulabilmek için Medyan filtre kullanılmasına karar verilmiştir (Bir Bhanu, 1986).

4.1.2 Histogram Eşleme

Histogram eşleme, koyu veya çok açık tonların olasılıklarına göre daha dengeli dağılmasını sağlayan yöntemdir (Gonzalez vd, 2004). Önce histogram değerleri piksel sayısına bölünerek gri seviyelerin resimde kullanılma yüzdeleri elde edilir, daha sonra her bir gri seviye yüzdesi öncekilerle toplanarak gri seviyeler için toplam yüzdeler bulunur, bulunan değerler düzgün (uniform) dağılım ile karşılaştırılarak eski gri seviye değerleri için yeni değerler belirlenmiş olur, bütün pikseller bu değerlere göre yeniden renklendirilir. Böylece renk bandının tamamı kullanılarak hedef şablonları ile arkaplan arasındaki farklılıklar artırılmış olur.



1. Resmin Histogram eşlenmiş hali



2. Resmin Histogram eşlenmiş hali



3. Resmin Histogram eşlenmiş hali



4. Resmin Histogram eşlenmiş hali

Şekil 4.4 Resimlerin Histogram Eşlenmiş Halleri

4.1.3 Resim Toplama

Örnek resimlerin histogramları alındığında, bölüm 3.1’de, özellikle 4. resimde görüldüğü gibi resim içindeki bazı bölgelerin tonları birbirine çok yakınlaşabilmekte ve veri kaybına yol açmaktadır. Bu tür problemleri önlemek ve resmin orijinal tonlarından çok uzaklaşmamasını sağlamak amacıyla orijinal resimle toplanıp ortalaması alınır.



Orijinal + Histogram eşlenmiş resim 1



Orijinal + Histogram eşlenmiş resim 2



Orijinal + Histogram eşlenmiş resim 3



Orijinal + Histogram eşlenmiş resim 4

Şekil 4.5 Resimlerin Histogramlarıyla Toplanması

4.1.4 Düzgün Olmayan Yayınım

Yayınım algoritmaları resim üzerindeki gürültüyü kısmi türev eşitlikleri kullanarak temizler. Düzgün yayınım eşitliği, ısı denklemi (4.1), şeklinde tanımlanmaktadır (Black vd., 1998).

$$\frac{\partial I}{\partial t} = \text{div}(\nabla I) \quad (4.1)$$

$\text{div}(\cdot)$: Divergence (x, y doğrultularındaki gradyanların mutlak değerler: ,resim 2 boyutlu olduğu için z doğrultusu yok)
 $I(x, y)$: Orijinal resim

Denklemlerde yer alan (x, y) uzaysal düzlemi ifade ederken, ∇I görüntünün kısmi türevini (gradyanını) temsil eder. Resmi bu denkleme göre işlemek, Gaussian Filtre ile aynı sonucu verecektir. Görüntü üzerindeki gri seviye yayılımı, ısı gibi homojen bir dağılıma sahip değildir. Amaç kenarları belirginleştirmek, kenar olmayan piksellerin değerlerini de birbirlerine yaklaştırmak olduğundan; Perona ve Malik ısı denklemine (4.1) hata fonksiyonu ekleyerek (4.2) resim üzerine uygulanabilir, kenarları bozmadan resmi iyileştiren bir hale getirmiştir (1990).

$$\frac{\partial I}{\partial t} = \text{div}[c(x, y)\nabla I] \quad (4.2)$$

$c(x, y)$: kenar durdurma fonksiyonu (Edge Stopping function)

İki fonksiyonun çarpımının diverjansı hesaplanırken “türev için çarpım” kuralı uygulanır (4.3).

$$\text{div}(u.v) = \nabla.(u.v) = \nabla u.v + u.\nabla v \quad (4.3)$$

Verilen denklem (4.2), çarpımın türevi kuralı ile açık olarak ifade edilirse; Hata fonksiyonu (kenar durdurma fonksiyonu) ile resmin gradyanının çarpımının diverjansı çarpımın türevi olarak hesaplanır (4.4).

$$\begin{aligned} u &= c(x, y) \\ v &= \nabla I \end{aligned} \quad \text{için;} \quad \text{div}[c(x, y)\nabla I] = c(x, y)\Delta I + \nabla c.\nabla I \quad (4.4)$$

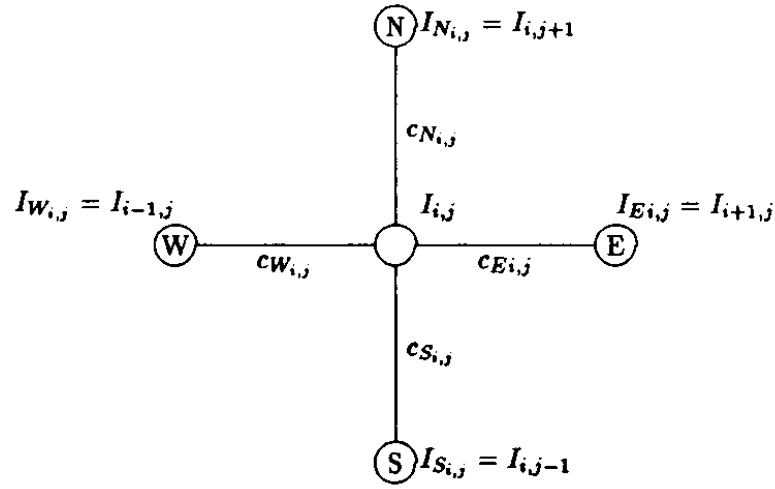
$\Delta I = \nabla^2 I$ (I 'nin Laplasyeni)

Yapılan yayılım işleminin kenarlara zarar vermeden resmi iyileştirebilmesi için kullanılabilecek en iyi hata fonksiyonu kenarlarda ‘0’, iç bölgede de ‘1’ değerini vermelidir;

böylece kenarlar hiç bozulmadan iç bölgedeki değerler birbirine yaklaştırılmış olur. Bu durumda $c(x,y,t)$ fonksiyonu sabit bir katsayı olarak kabul edilebilir ve denklem (4.4) denklem (4.5) şeklinde kısalır.

$$\begin{aligned} c &= \text{sabitkatsayı} \Rightarrow \nabla c = 0 \\ \text{div}[c(x,y)\nabla I] &= c\Delta I \end{aligned} \quad (4.5)$$

Denklemin sayısal domende, sayısal bir görüntü üzerine uygulanmasında, pikselin Şekil 4.6'de görüldüğü gibi en yakın komşularından faydalanılmıştır.



Şekil 4.6 En yakın Komşu pikseller (Perona ve Malik (1990))

Laplasyenin iki boyutlu resim için ayrık domen ifadesini bulmak için önce, resmin gradyanı hesaplanır (4.6).

$$\begin{aligned} \nabla I &= \frac{\partial I}{\partial x} + \frac{\partial I}{\partial y} \\ \frac{\partial I}{\partial x} &= I(x+1, y) - I(x, y) \quad \frac{\partial I}{\partial y} = I(x, y+1) - I(x, y) \\ \nabla I &= I(x+1, y) + I(x, y+1) - 2I(x, y) \end{aligned} \quad (4.6)$$

Gradyanı hesaplanılan görüntünün tekrar gradyanı alınarak sayısal görüntünün Laplasyeni hesaplanmış olur (4.7).

$$\begin{aligned}
\Delta I &= \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2} \\
\frac{\partial^2 I}{\partial x^2} &= I(x+1, y) - I(x, y) + I(x-1, y) - I(x, y) \\
\frac{\partial^2 I}{\partial y^2} &= I(x, y+1) - I(x, y) + I(x, y-1) - I(x, y) \\
\Delta I &= I(x+1, y) + I(x-1, y) + I(x, y+1) + I(x, y-1) - 4I(x, y)
\end{aligned} \tag{4.7}$$

Komşu pikseller ve katsayıları (Şekil 4.6), denklem (4.6)'da verilen ayrık Laplasyen operatörü üzerine uygulanırsa denklem (4.8) elde edilir.

$$\begin{aligned}
I(x, y) &= I_{i,j} \text{ şeklinde gösterilirse;} \\
c\Delta I &= \frac{\left[c_W (I_{i-1,j} - I_{i,j}) + c_N (I_{i+1,j} - I_{i,j}) + c_E (I_{i+1,j} - I_{i,j}) + c_S (I_{i,j-1} - I_{i,j}) \right]}{n_s}
\end{aligned} \tag{4.8}$$

Homojen olmayan yayılım denklemindeki kısmi türev ifadesi de ayrık domene (4.9) ifadesi ile aktarılabilir.

$$\frac{\partial I}{\partial t} \Rightarrow I_{i,j}^{t+1} - I_{i,j}^t \tag{4.9}$$

Sürekli domende denklem (4.2) ve denklem (4.5) ile ifade edilen eşitliğin iki tarafına ayrık domen ifadeleri denklem (4.8) ve denklem (4.9) yerleştirilirse ve laplasyen denklemi difüzyon katsayısı (λ) ile çarpılırsa; ayrık denklem elde edilmiş olur.

$$I_{i,j}^{t+1} = I_{i,j}^t + \lambda \frac{\left[c_W (I_{i-1,j} - I_{i,j}) + c_N (I_{i+1,j} - I_{i,j}) + c_E (I_{i+1,j} - I_{i,j}) + c_S (I_{i,j-1} - I_{i,j}) \right]}{n_s} \tag{4.10}$$

Elde edilen denklemi sadeleştirmek için komşu piksellerin merkez pikselden farkı tek sembolle ifade edilirse (buradaki sembol gradyant değil ayrık domen fark alma işlemi) Perona ve Malik'in ayrık domen denklemi (4.11) bulunur (1990).

$$\nabla I_{s,p} = I_p - I_s \Rightarrow I^{t+1}_s = I_s^t + \frac{\lambda}{n_s} \sum c \nabla I_{s,p} \tag{4.11}$$

I_s^t : Ayrık olarak örneklenmiş resim

λ : Yayılım miktarını belirleyen katsayı

s : merkez piksel için iki boyutlu düzlemsel konum

p : komşu pikseller için iki boyutlu düzlemsel konum
 t : Ayrık zaman değerleri (iterasyon sayısı)
 n_s : İşleme dahil edilen komşu piksel sayısı

Denklemden en önemli kısım şüphesiz ki kenar durdurma fonksiyonudur. Kenar durdurma fonksiyonunda katsayıyı elde etmek için farklı denklemler kullanılabilmektedir.

Amaç kenarları korumak ve belirginleştirmek olduğundan Homojen olmayan yayınıma kenarlara en az zararı vererek gerçekleştiren fonksiyonlar çok daha iyi sonuç verecektir. Çok sayıda araştırmacı bu tarz denklemlerin matematiksel özelliklerini ve rakamsal denemelerini farklı uygulamalara nasıl adapte edilebileceğini araştırmaktadır. Perona ve Malik, hata fonksiyonu olarak Lorentzian Fonksiyonu'nu (4.12) kullanmıştır (1990).

$$g(x, \sigma) = \frac{1}{1 + \left(\frac{x^2}{2\sigma^2} \right)} \quad (4.12)$$

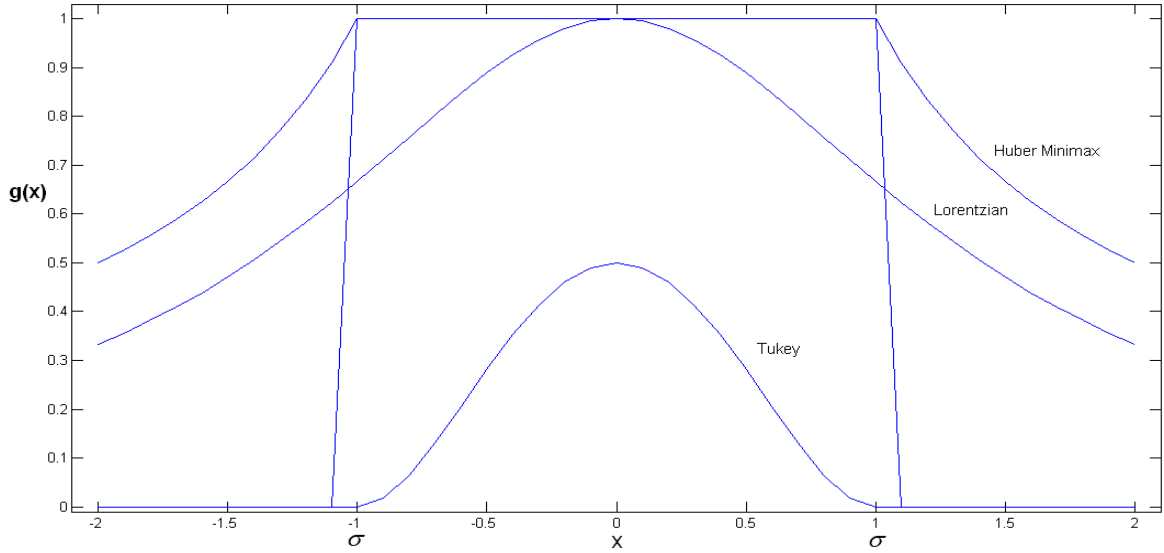
Lorentzian fonksiyonunun yerine Tukey (4.13) ve Hubert (4.14) Fonksiyonlarının kullanıldığı da bilinmektedir (Black vd., 1998).

$$g(x, \sigma) = \begin{cases} \frac{1}{2} \left[1 - \left(\frac{x}{\sigma} \right)^2 \right]^2 & |x| \leq \sigma \\ 0 & \text{diğer} \end{cases} \quad (4.13)$$

Tukey fonksiyonu varyanstan büyük değerler için yayınıma durdururken Hubert Fonksiyonu yayınıma farklı bir oranda devam etmektedir. Değişkenin varyanstan küçük olduğu durumlarda ise Tukey Fonksiyonu Hubert Fonksiyonuna göre daha düşük bir yayılım hızına müade etmektedir ki; bu da istenilen sonuca ulaşmak için gerekli olan iterasyon sayısını artırabilir (Black vd., 1998).

$$g(x, \sigma) = \begin{cases} \frac{1}{\sigma} & |x| \leq \sigma \\ \frac{\text{sign}(x)}{x} & |x| > \sigma \end{cases} \quad (4.14)$$

Lorentzian, Tukey ve Hubert Fonksiyonlarının grafikleri Şekil 4.7' de görülmektedir. Grafik incelendiğinde, belirtilen fonksiyonların ideale yakın bir bant geçiren filtreden uzak olduğu görülmektedir.



Şekil 4.7 Lorentzian, Tukey, Huber, Dikdörtgen Hata ($g(x)$) Fonksiyonları

$$g(x, \sigma) = \begin{cases} 1 & |x| \leq \sigma \\ 0 & |x| > \sigma \end{cases} \quad (4.15)$$

Dikdörtgen .hata fonksiyonu (Rectangular Error Function)

Yapılan algoritma çalışmasında ise, daha keskin bant geçiren bir fonksiyonun daha başarılı sonuç vereceği değerlendirildiğinden dikdörtgen fonksiyon (4.15) hata fonksiyonu olarak kullanılmıştır. Sayısal domen yayılım denkleminde (4.11), “c” katsayısı yerine dikdörtgen hata fonksiyonu (4.15) eklenilmiş ($c \rightarrow g(x)$), ortaya simülasyonda kullanılan denklem (4.16) ve parametreleri çıkmıştır.

$$\begin{aligned}
x &= \nabla I_{s,p} = I_p - I_s = I_{i+k,j+l} - I_{i,j} \\
g(x) &= \begin{cases} 1 & |x| < \sigma \\ 0 & |x| > \sigma \end{cases} \\
n_s &= \begin{cases} n_s + 1 & |x| < \sigma \\ n_s & |x| > \sigma \end{cases} \\
I_{i,j}^{t+1} &= I_{i,j}^t + \frac{\sum_{k=-1}^1 \sum_{l=-1}^1 (I_{i+k,j+l}^t - I_{i,j}^t) \times g(x)}{n_s}
\end{aligned} \tag{4.16}$$

$$i \in N(2, s_1 - 1)$$

$$j \in N(2, s_2 - 1)$$

$$l \in Z(-1, 1)$$

$$k \in Z(-1, 1)$$

s_1 : yatay düzlemdeki piksel sayısı

s_2 : dikey düzlemdeki piksel sayısı

σ : eşik değeri

Simülasyonunun 11x11 lik örnek bir matris üzerinde, nasıl uygulandığı, matrislerle birlikte aşağıda (4.17) gösterilmiştir, MATLAB programında yazılmış uygulama kodu da Ek_3'de verilmiştir.

$$\begin{bmatrix}
211 & 208 & 208 & 203 & 206 & 203 & 208 & 208 & 211 & 211 & 211 \\
198 & 194 & 198 & 194 & 201 & 198 & 206 & 206 & 211 & 206 & 203 \\
126 & 118 & 135 & 135 & 156 & 156 & 180 & 180 & 186 & 173 & 156 \\
88 & 85 & 90 & 90 & 96 & 96 & 99 & 97 & 99 & 96 & 96 \\
76 & 73 & 75 & 75 & 78 & 76 & 76 & 75 & 73 & 70 & 70 \\
103 & 99 & 99 & 96 & 96 & 90 & 88 & 81 & 78 & 70 & 66 \\
208 & 201 & 201 & 186 & 173 & 135 & 118 & 113 & 109 & 100 & 85 \\
219 & 217 & 217 & 214 & 211 & 206 & 203 & 203 & 203 & 191 & 146 \\
218 & 217 & 217 & 217 & 217 & 217 & 218 & 218 & 219 & 218 & 217 \\
217 & 217 & 217 & 214 & 217 & 217 & 219 & 219 & 219 & 219 & 219 \\
217 & 217 & 217 & 217 & 217 & 217 & 218 & 218 & 217 & 217 & 211
\end{bmatrix} \tag{4.17}$$

i=7,j=5 için işaretli piksele ait yeni değeri, denklem (4.18)'deki gibi hesaplanır.

$$\begin{aligned}
I_{7,5}^1 = I_{7,5}^0 &+ ((I_{6,4} - I_{7,5}) \times g(x)) + ((I_{6,5} - I_{7,5}) \times g(x)) + ((I_{6,6} - I_{7,5}) \times g(x)) \\
&+ ((I_{7,4} - I_{7,5}) \times g(x)) + ((I_{7,5} - I_{7,5}) \times g(x)) + ((I_{7,6} - I_{7,5}) \times g(x)) \\
&+ ((I_{8,4} - I_{7,5}) \times g(x)) + ((I_{8,5} - I_{7,5}) \times g(x)) + ((I_{8,6} - I_{7,5}) \times g(x))
\end{aligned} \quad (4.18)$$

İşlemde piksel değerleri yerlerine koyulduğunda pikselin yeni değeri bulunmuş olur (4.20).

$$I_{7,5}^1 = 180 \quad (4.19)$$

1 piksel için yukarıda uygulaması anlatılan yöntem bütün resim matrisi için tekrarlanır ve 1. iterasyon tamamlanmış olur. 1. iterasyon sonucunda oluşan matriste (4.21) de işaretli yeni değer görülmektedir.

$$\begin{bmatrix}
211 & 208 & 208 & 203 & 206 & 203 & 208 & 208 & 211 & 211 & 211 \\
198 & 201 & 201 & 202 & 201 & 204 & 205 & 208 & 209 & 209 & 203 \\
126 & 122 & 135 & 135 & 156 & 156 & 180 & 182 & 180 & 180 & 156 \\
88 & 81 & 83 & 84 & 94 & 97 & 97 & 98 & 97 & 97 & 96 \\
76 & 79 & 81 & 82 & 82 & 82 & 81 & 79 & 75 & 71 & 70 \\
103 & 100 & 98 & 97 & 94 & 84 & 81 & 79 & 75 & 73 & 66 \\
208 & 203 & 201 & 187 & \mathbf{180} & 135 & 116 & 113 & 107 & 98 & 85 \\
219 & 216 & 217 & 213 & 214 & 212 & 211 & 209 & 207 & 197 & 146 \\
218 & 217 & 216 & 216 & 214 & 214 & 213 & 213 & 219 & 216 & 217 \\
217 & 217 & 217 & 217 & 217 & 217 & 218 & 218 & 218 & 217 & 219 \\
217 & 217 & 217 & 217 & 217 & 217 & 218 & 218 & 217 & 217 & 211
\end{bmatrix} \quad (4.20)$$

İşlemin yapısı gereği birden fazla kez tekrarlanması gerekmektedir, yapılan uygulamada işlem 25 kez tekrarlanmıştır ve elde edilen matris de (4.22) aşağıda görülmektedir.

$$\begin{bmatrix}
211 & 208 & 208 & 203 & 206 & 203 & 208 & 208 & 211 & 211 & 211 \\
198 & 206 & 206 & 205 & 204 & 205 & 206 & 208 & 209 & 209 & 203 \\
126 & 128 & 130 & 131 & 156 & 156 & 181 & 181 & 181 & 181 & 156 \\
88 & 85 & 86 & 87 & 87 & 86 & 85 & 84 & 82 & 82 & 96 \\
76 & 85 & 87 & 87 & 87 & 86 & 85 & 84 & 82 & 78 & 70 \\
103 & 88 & 87 & 87 & 87 & 86 & 85 & 84 & 82 & 78 & 66 \\
208 & 214 & 215 & 215 & \mathbf{215} & 135 & 85 & 84 & 82 & 82 & 85 \\
219 & 215 & 215 & 215 & 215 & 215 & 216 & 216 & 216 & 216 & 146 \\
218 & 217 & 216 & 215 & 215 & 216 & 216 & 216 & 216 & 217 & 217 \\
217 & 217 & 216 & 216 & 216 & 216 & 217 & 217 & 217 & 216 & 219 \\
217 & 217 & 217 & 217 & 217 & 217 & 218 & 218 & 217 & 217 & 211
\end{bmatrix} \quad (4.21)$$

Hata fonksiyonu olarak dikdörtgen fonksiyonun (4.15) kullanılmasının daha az iterasyonla daha iyi sonuç verdiği gözlemlenmiştir. İterasyon sayısının daha az olması ve fonksiyonun diğerlerinden daha basit bir yapıda olması işlem yükünü de hafifletmiştir. Algoritmanın uygulaması ise diğer filtrelere benzemektedir. Filtre 3x3lük resim parçacıkları üzerinde gerçekleştirilmiş ve işlemin tekrarlanmasının çok daha iyi sonuç verdiği gözlemlenmiştir. Yapılan çalışmada, 4 resim üzerinde de, homojen olmayan yayılım işlemi verilen hata fonksiyonu (4.15) kullanılarak 10'ar kez uygulanmış ve Şekil 4.8'de görülen sonuçlar elde edilmiştir.



1. Resme Düzgün olmayan Yayılım Uyg.



2. Resme Düzgün olmayan Yayılım Uyg.



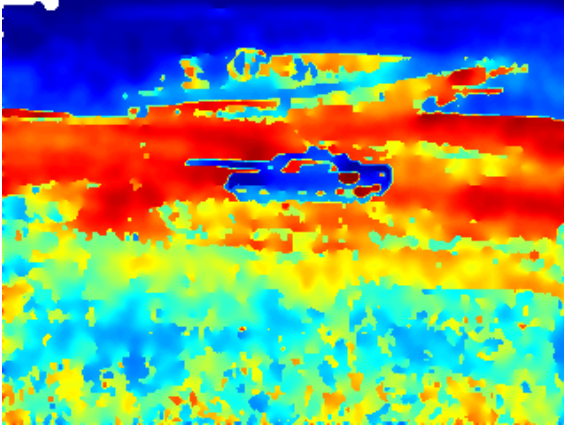
3. Resme Düzgün olmayan Yayılım Uyg.



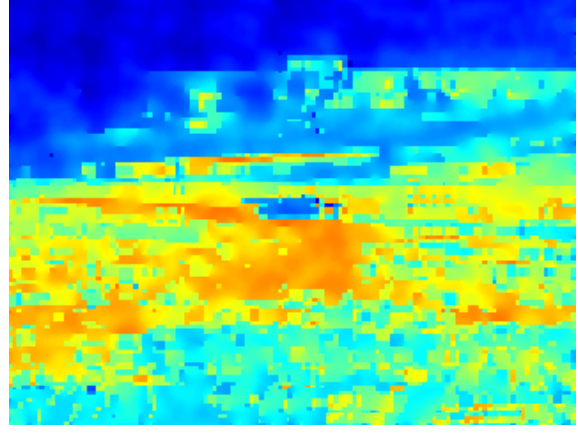
4. Resme Düzgün olmayan Yayılım Uyg.

Şekil 4.8 Resimlere Düzgün Olmayan Yayılım Uygulanması

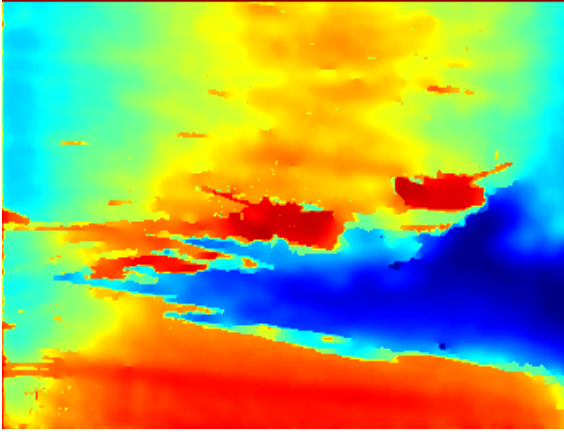
Düzgün olmayan yayılım yöntemi ile yapılan bütünleştirme işlemi, Şekil 4.9'de renklendirilmiş resimler üzerinden daha net gözlemlenebilmektedir; bütünleştirme işlemi ile oluşan kontrast farkı, hedef ve arka planın 4 resim de de farklı renkler almasını sağlayacak kadar artmıştır.



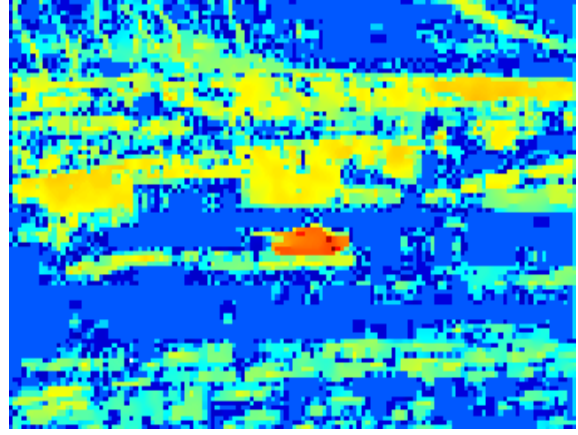
1. Resmin Renklendirilmiş Hali



2. Resmin Renklendirilmiş Hali



3. Resmin Renklendirilmiş Hali



4. Resmin Renklendirilmiş Hali

Şekil 4.9 Bütünleştirilmiş Resimlerin Renklendirilmesi

İşlem sonucunda, Şekil 4.8’de ve Şekil 4.9’de verilen resimler incelendiğinde; gürültülü arka planda ve hedef nesnenin üzerinde bütünleşme sağlandığı görülmektedir.

4.1.5 Görüntü İyileştirme İşlemleri Değerlendirmesi

Görüntü bölümlenmeden önce yapılan gürültü filtreleme (medyan), kontrast iyileştirme (Histogram eşleme ve toplama) ve yayılım (homojen olmayan yayılım) işlemlerinin gerekliliğinden ve nasıl uygulandığından 4.1.1, 4.1.2, 4.1.3, 4.1.4 bölümlerinde bahsedilmiştir. Anlatılan algoritmaların resim üzerindeki etkileri ise şablon ortalaması (4.22), standart sapma (4.23), varyans (4.24), Kurtosis (4.25) ve SNR (4.26) yöntemleri ile analiz edilmiş ve Tablo 4.1 'deki değerler elde edilmiştir.

$$\mu = \frac{\sum_{i=1}^n X_i}{n-1} \quad (4.22)$$

Resim üzerinde hedef şablonunun (Mt) ve arka planın (Mb) ortalaması ayrı ayrı hesaplanmış ve karşılaştırılmıştır.

$$s = \sqrt{\frac{\sum_{i=1}^n (X_i - \mu)^2}{n-1}} \quad (4.23)$$

Tabloda St olarak ifade edilen hedef standart sapması ve Sb olarak ifade edilen arkaplan standart sapması; Mt ve Mb değerleri kullanılarak ayrı ayrı hesaplanmıştır.

$$\sigma = \frac{\sum_{i=1}^n (X_i - \mu)^2}{n-1} \quad (4.24)$$

Standart sapmaların kareleri ise varyans değerlerini vermektedir.

$$k = \frac{E(x - \mu)^4}{\sigma^4} \quad (4.25)$$

Kurtosis değeri sıfırdan küçük ise “subgaussian” adı verilir ve incelenen veri kümesinin düzgün bir dağılımı olduğu anlamına gelir. Eğer Kurtosis değeri sıfıra çok çok yakın ise “gaussian ” adı verilir ve veri kümesinin gaussa uygun bir dağılıma sahip olduğu anlamına gelir. Kurtosis değeri sıfırdan büyük ise incelenen veri kümesi “super gaussian” olarak adlandırılır ve verinin belirli bir aralıkta yoğunlaştığı anlaşılır.

$$S = \text{div}(\nabla T), \quad N = \text{div}(\nabla I) - \text{div}(\nabla T)$$

$$SNR = \frac{S}{N} \quad (4.26)$$

Hedefin ayırt edilebilirliğini ölçebilmek için şüphesiz ki SNR da önemli bir ölçüttür. Orjinal hedef görüntüsünün gradyanı istenilen sinyal kabul edilmiş ve görüntünün kalanının gradyanı da gürültü olarak hesaplamaya dahil edilmiştir.

Ön işlemler ile hedef şablonu ve arka plan arasındaki ortalama farkları artırılmaya çalışılmış, standart sapmanın ve varyansın aynı oranda artması engellenmeye çalışılmıştır.

Nitekim, Tablo 4.1 'de ortalama farkları Histogram eşitleme ile artmış, yükselen varyans değerleri, orijinal resim ile toplama ve homojen olmayan yayılım yöntemleri ile düşürülmüştür.

Özellikle 1. test resmindeki hedef şablonunun işlemler öncesinde düzgün bir dağılımı olduğu (subgaussian), işlemler sonrasında ise gri seviyelerin belirli bir aralıkta yoğunlaştığı (supergaussian) gözlemlenmiştir. SNR değerleri ise Homojen olmayan yayılımın bölümlendirme işlemini önemli ölçüde kolaylaştırdığını göstermektedir.

Tablo 4.1 Ön işlemler Değerlendirme Çizelgesi

	Mt	Mb	Mt-Mb	St	Sb	Vt	Vb	Kt	Kb	SNR
Test Resmi 1	197,74	144,5	53,24	16,73	45,08	280	2032	-0,5	1,2	0,5498
Histogram Eşlenmesi	230,5	108,2	122,3	37,84	57,43	1432	3299	4	-0,8	0,78
Histogram + Orijinal Resim	215,36	124,3	91,06	27,6	50	762	2501	2,48	-0,22	0,75
Homojen Olmayan Yayınım	216,24	123,8	92,44	27,57	49,06	760	2407	3,83	-0,24	0,88
Test Resmi 2	74,77	103,7	28,93	17,58	11,97	309	143,3	-0,4	-0,02	0,56
Histogram Eşlenmesi	100,6	215	114,4	62,6	34,2	3919	1173	0,37	-0,73	0,76
Histogram + Orijinal Resim	88,8	164,1	75,3	41,55	23,56	1726	555,5	0,2	-0,89	0,73
Homojen Olmayan Yayınım	88,9	164,08	75,18	40,6	21,72	1648	472,15	0,366	-0,77	0,92
Test Resmi 3	85,7	136,6	50,9	39,17	17,63	1534	311	2,16	-0,1843	0,4258
Histogram Eşlenmesi	79,39	214,67	135,28	71,31	35,63	5086	1269	0,46	0,3	0,4603
Histogram + Orijinal Resim	80,5	195,1	114,6	62,72	30,8	3934	948,8	0,49	-0,04	0,4387
Homojen Olmayan Yayınım	79,9	204,64	124,74	66,82	27,97	4466	782,55	0,62	0,04	0,67
Test Resmi 4	110,55	36,5	74,05	44,46	12,24	1977	150	0,069	5,84	0,74
Histogram Eşlenmesi	235,43	114,77	120,66	49,7	72,22	2471	5215	9,99	-1,14	0,745
Histogram + Orijinal Resim	176,88	77,86	99,02	43,09	43,57	1857	1898	3,88	-0,99	0,81
Homojen Olmayan Yayınım	177,23	77,83	99,4	42,99	43,64	1848	1905	4,1	-1,04	0,87

4.2 BÖLÜMLEMEDE KULLANILAN YÖNTEM

Bölümleme işlemini gerçekleştirmek için kenar tanıma yöntemleri tek başına yeterli olmayacağından yapılan çalışmada bölümleme işlemi; kenar belirleme işlemi (Canny 2.2.3), kenar tamamlama (Morfolojik closing), kapalı şekillerin doldurulması (Morfolojik filling 4.2.3), objelerin bütünleştirilip tespit edilmesi (CCL 4.2.4) alt işlemlerinden oluşturulmuştur.

4.2.1 Geliştirilmiş Canny Yöntemi ile Kenar Belirleme

ATR algoritmasında kenar tespit işlemini gerçekleştirebilmek için geliştirilmiş bir Canny Metodu kullanılmıştır. Kenar belirlemede Canny algoritmasının nasıl kullanıldığı Bölüm 2.2.3.4’de detaylı olarak anlatılmıştı. Canny Metodunda otomatik eşik değeri belirlenirken, “Hysteresis” yönteminin kullanıldığı bilinmektedir. Yapılan araştırma ve uygulamalarda Hysteresis yöntemi yerine Otsu yöntemi kullanıldığında daha iyi sonuçlar elde edildiği görülmüştür ve Canny algoritması Otsu Yöntemi ile birlikte kullanılmıştır. Hysteresis ve Otsu Yöntemleri ile elde edilen sonuçlar aşağıda karşılaştırmalı olarak incelenecektir.

4.2.1.1 Hysteresis Yöntemi

Hysteresis yönteminde yüksek ve düşük olmak üzere iki tane eşik değeri belirlenir. İlk olarak resim üzerindeki piksellerin belirli bir yüzdesini kenar kabul edecek şekilde (MATLAB’da %30 kabul edilmiş) bir yüksek eşik değeri belirlenir. Düşük eşik değeri ise belirlenen yüksek eşik değerin belirli bir katsayı ile çarpılması sonucunda hesaplanır (MATLAB’da 0.4 ile çarpılmış; yüksek eşik değeri %40’ına karşılık geliyor).

4.2.1.2 Otsu Yöntemi

Otsu yöntemi eşik seviyesi belirlerken, istatistiksel verilerden faydalanır. Yapılan filtre ve kenar inceltme işlemlerinden elde edilen değerlerin (Bölüm 2.2.3.4) histogramı çıkartılır. Elde edilen histogram değerleri iki gruba ayrılır ve iki grup için de varyans değerleri hesaplanır, bu işlem eşik değeri değiştirilerek tekrarlanır. En düşük sınıflar arası varyans değerlerini veren değer de eşik değeri olarak kabul edilir (Otsu N.,1979). Sınıflar arasındaki varyans (2.20), iteratif olarak her eşik değeri için hesaplanır ve en yüksek sınıflar arası varyans değerini veren (en düşük sınıf içi varyans değerini veren) eşik değeri olarak seçilir.

$$\sigma_A^2(T) = n_B(T) n_O(T) [\mu_B(T) - \mu_O(T)]^2 \quad (4.27)$$

$$n_B(T) = \sum_{i=0}^{T-1} p(i) \quad (4.28)$$

$$n_O(T) = \sum_{i=T}^{N-1} p(i) \quad (4.29)$$

$\sigma_A^2(T)$ = Sınıflararası varyans



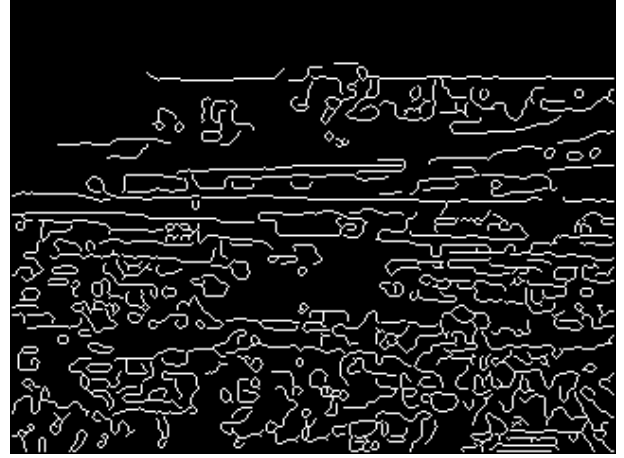
1. Test Resmi Histerisis Yöntemi ile



1. Test Resmi Otsu Yöntemi ile



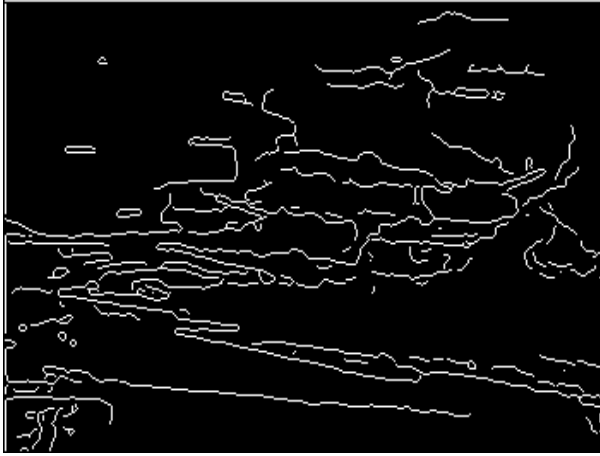
2. Test Resmi Histerisis Yöntemi ile



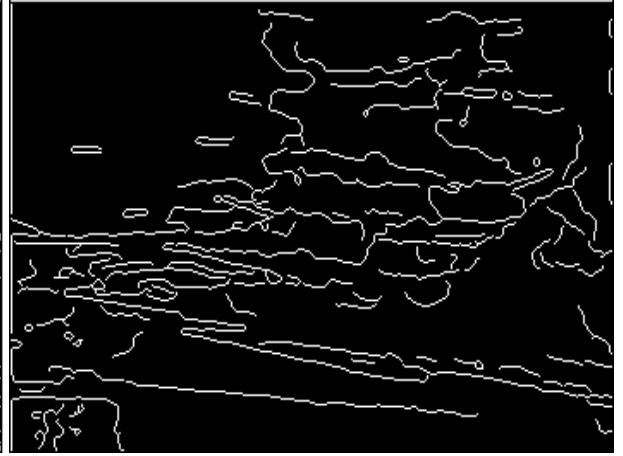
2. Test Resmi Otsu Yöntemi ile

Şekil 4.10 Canny Metodu ile Kenar Tespitinin Yapılması

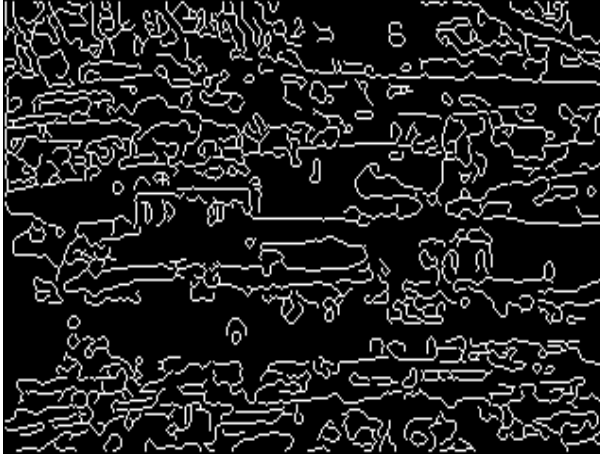
Histerisis ve Otsu yöntemleri ile tamamlanan Canny kenar tespit yöntemleri incelenirse; Şekil 4.10 ve Şekil 4.11'da görüldüğü üzere Otsu Yönteminde hedefe ait kenarlar daha belirginleşmiştir. Hedef etrafındaki istenmeyen kenarlar da yine Otsu yönteminde belirgin bir şekilde azalmıştır.



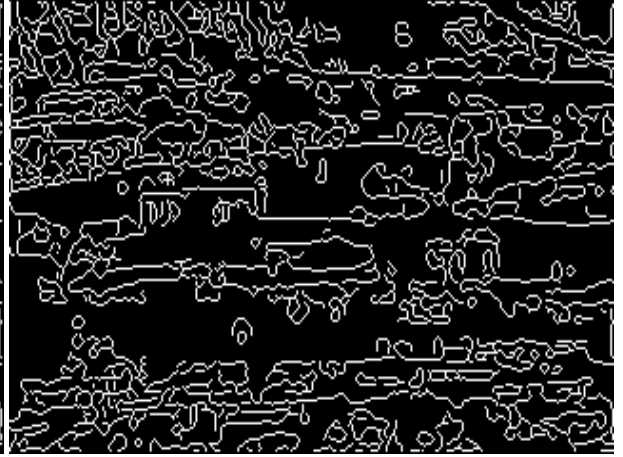
3. Test Resmi Histerisis Yöntemi ile



3. Test Resmi Otsu Yöntemi ile



4. Test Resmi Histerisis Yöntemi ile



4. Test Resmi Otsu Yöntemi ile

Şekil 4.11 Canny Metodu ile Kenar Tespitinin Yapılması

Canny ile yapılan kenar tanıma işlemi sonucunda bir çok kenar çizgisinin kaldığı Şekil 4.11’ de görülmektedir. Hedef ile arka plan arasındaki kontrast farkı fazla olmadığından istenmeyen kenar çizgilerinin sadece kenar tanıma yöntemi kullanılarak yok edilemeyeceği anlaşılmaktadır. İstenmeyen kenarların elenmesi sonraki aşamalarda gerçekleştirileceğinden, Kenar tanıma algoritması hedefin kenarlarını algılayabileceği bir eşik seviyesinde çalıştırılmalıdır.

4.2.2 Kenar İyileştirme

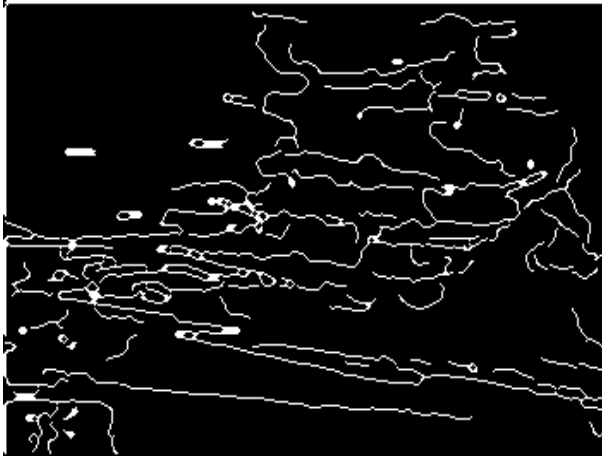
Kenar Tespit işleminin resim üzerine uygulanmasından sonra bütün kenarların birleşik olarak elde edilemediği görülmüştür. Kenarlar üzerindeki boşlukların doldurulması ve resmin içinin doldurulması içinse basit; ama hızlı olan Morfolojik yöntemler tercih edilmiştir. Kenarları tamamlamakta “Closing”, nesnenin içini doldurmak içinde “Filling” işlemi uygulanmıştır.



1. Resme “Morfolojik Closing” Uyg.



2. Resme “Morfolojik Closing” Uyg.



3. Resme “Morfolojik Closing” Uyg.

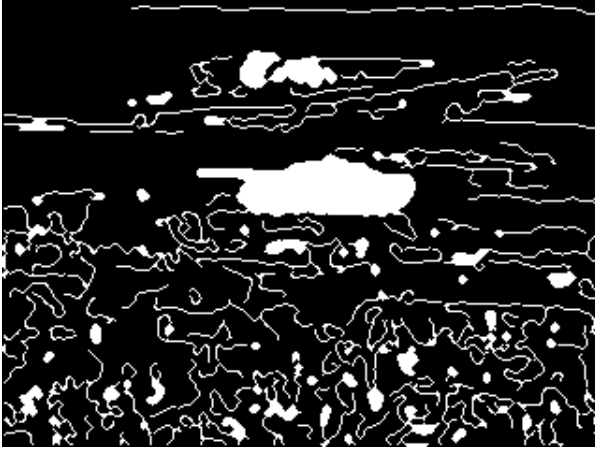


4. Resme “Morfolojik Closing” Uyg.

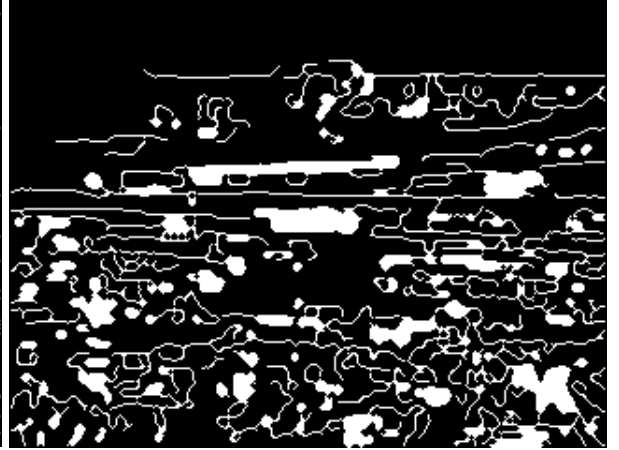
Şekil 4.12 Closing İşlemi Uygulanmış Resimler

4.2.3 Doldurma (Filling)

Filling işleminde de CCL'den faydalanılır, arka plana bağlantısı bulunan her piksel arka plandan kabul edilir diğer bölgeler ise “1” olarak işaretlenir. Filling işlemi nesnelerin çıkartılmasında önemli bir aşamadır.



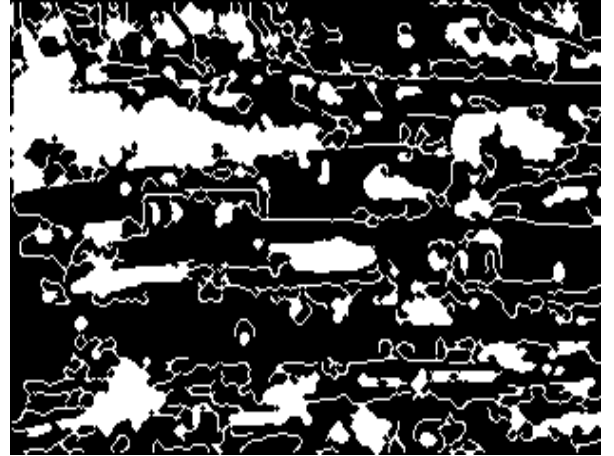
1 . Resme Doldurma İşleminin Uyg.



2. Resme Doldurma İşleminin Uyg.



3. Resme Doldurma İşleminin Uyg.



4. Resme Doldurma İşleminin Uyg.

Şekil 4.13 Doldurulmuş Test Resimleri

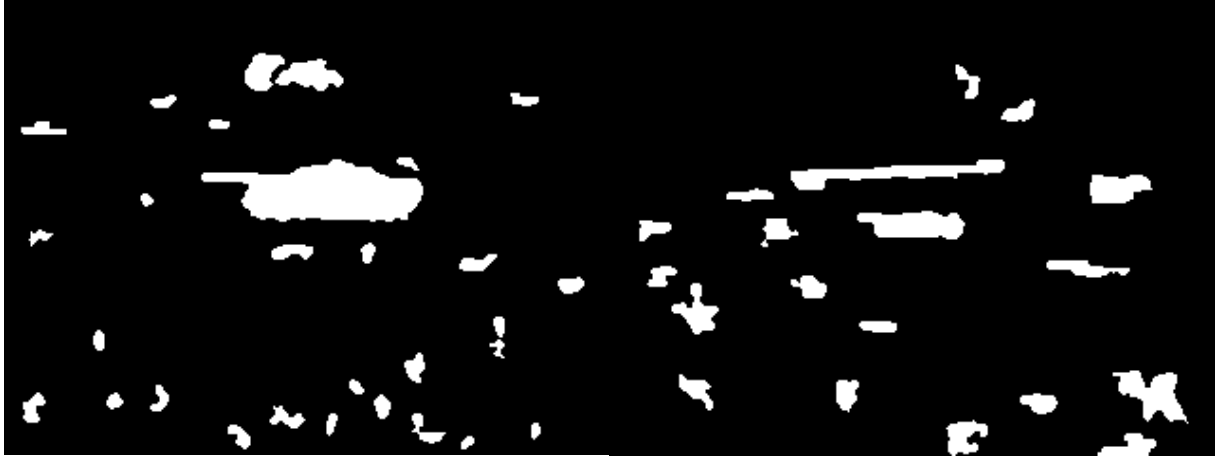
4.2.4 Bağlı Bileşen Etiketleme (CCL)

Bağlı bileşen etiketleme, aynı renkten ve birbirleriyle komşulukları olan piksellerin birlikte gruplanmasıyla gerçekleştirilen oldukça hızlı ve başarılı bir gruplandırma metodudur. Aşağıda algoritmanın çalışma mantığı anlatılmıştır:

Tablo 4.2 CCL Algoritma Akışı

- | |
|---|
| <p>1. Resimdeki bütün pikseller satır satır sırayla taranır:</p> <ul style="list-style-type: none"> • Eğer pikselin aynı değere sahip komşusu yoksa yeni bir grup oluşturulur ve gruba yeni piksel eklenir. • Eğer piksel, komşularından herhangi biriyle aynı değere sahipse ve komşusu bir gruba dahilse yeni piksel de aynı gruba eklenir. • Eğer pikselin gruplanmış birden fazla komşusu varsa hangisinin daha yakın olduğu belirlenir ve yeni piksel de o gruba eklenir. <p>2. Tarama işlemi ters yönde gerçekleştirilerek aynı veya yakın renkteki gruplar birleştirilir.</p> |
|---|

CCL işlemi sonucunda istenmeyen nesnelerin bir kısmı daha elimine edilmiş olur. Sınıflandırma öncesinde hedef olması mümkün olmayan alternatif nesnelerin ayıklanması içinse hedefin büyüklüğü, eni, boyu, alan doluluğu gibi özellikleri tespit edilir.



1. Resmin CCL ile Temizlenmesi

2. Resmin CCL ile Temizlenmesi



3. Resmin CCL ile Temizlenmesi



4. Resmin CCL ile Temizlenmesi

Şekil 4.14 Resimlerden Küçük Şekillerin Temizlenmesi

Resim içerisinde yer alan ve tanınması mümkün olmayacak büyüklükteki nesneler de gürültü kabul edilmiştir. Gürültü kabul edilen parçacıklar da CCL kullanılarak elenmiştir.

4.2.5 Bölümleme Değerlendirmesi

Kullanılan ATR yönteminde Bölüm 2.2.2’de anlatılan yöntemlerin yerine Bölüm 2.2.3’deki kenar tanıma yöntemi tercih edilmiştir.

Yapılan araştırma ve uygulamalarda eşik değer kullanan yöntemlerin daha çok; az sayıda, büyük, gri seviye farklılıkları yüksek nesneler içeren resimler üzerinde etkili olduğu gözlemlenmiştir. Yapılan çalışmada karşılaşılan nesneler ise değişken büyüklüklerde ve kontrastı düşük nesnelerdir.

Histogram temelli yöntemler incelendiğinde, gürültü duyarlılıklarının çok yüksek olduğu ve her zaman doğru eşik değerlerinin belirlenemediği görülmüştür.

Alan Temelli yöntemler incelendiğinde, ilk fark edilen işlemlerin resimdeki bütün pikseller üzerinden yapıldığından çok uzun sürmesidir. Alan temelli yöntemlerle yapılan bölümleme işleminde karşılaşılan bir diğer sorun da nesnelerin şekil özelliklerinde kayıpların oluşabilmesidir.

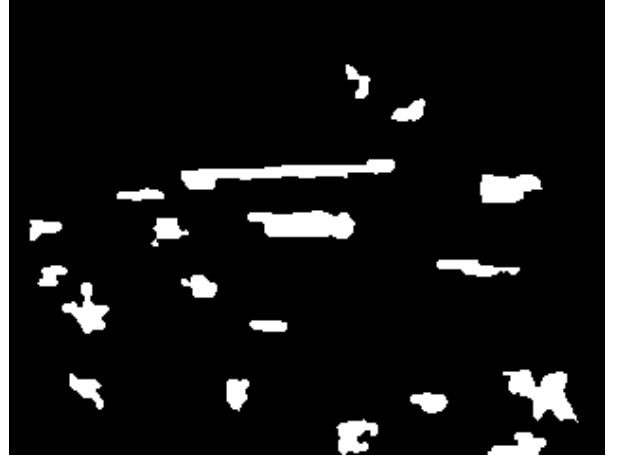
Kenar temelli yöntemlerin nesnenin bütününe daha başarılı bir şekilde ayırt ettiği görülmektedir. Yine, kenar temelli yöntemlerde resim üzerindeki piksellerin yaklaşık %10’u kullanıldığından ATR gibi üç aşamalı çok fonksiyonlu bir algorithmada daha düşük işlem süreleri sağlamaktadır (Bhanu, 1986).

4.3 ŞEKİL SINIFLANDIRMA

Hedef nesneler incelenecek olursa, anlamlı nesnelerin enleri ve boyları arasında belirli bir oran olduğu, yine içerdiği piksel sayısının da en ve boyuna bağlı olarak belirli bir miktarın altında olamayacağı, nesneyi çevreleyen piksel sayısı ile alanı arasında belirli bir oran olduğu bilindiğinden belirlenen oranların dışında kalan nesneler anlamsız kabul edilip ve elenmektedir (Kürekli, 2004). Eleme sonucunda Şekil 4.15'deki resimler elde edilmiştir.



1. Resimden Anlamsız Şekillerin Elenmesi



2. Resimden Anlamsız Şekillerin Elenmesi



3. Resimden Anlamsız Şekillerin Elenmesi



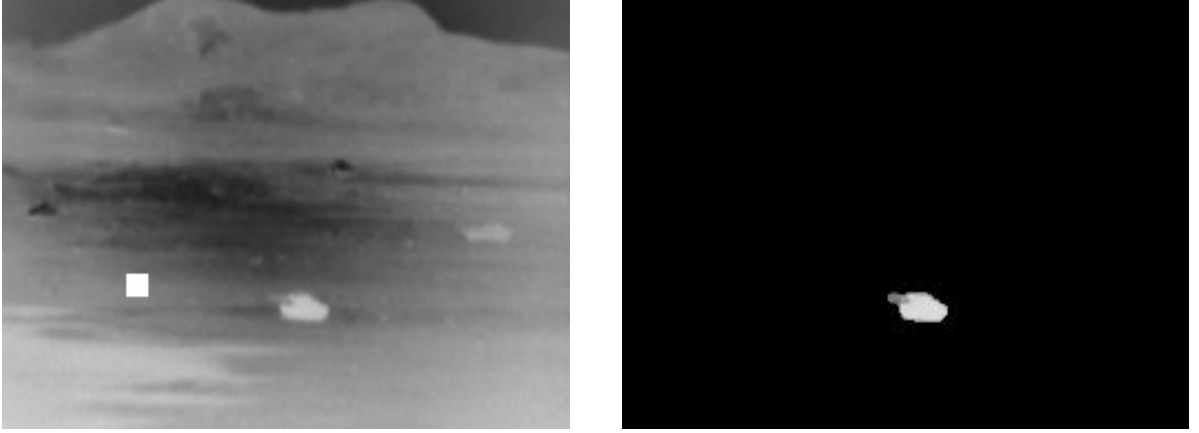
4. Resimden Anlamsız Şekillerin Elenmesi

Şekil 4.15 Anlamsız Şekillerin Elenmesi

4.4 ŞEKİL TANIMA

Şekil tanıma, veri kümesindeki nesnelerin, belirli sayıda sınıfa ayrılmasını amaçlayan yöntemlerdir (Draper, 1993). Şekil Tanıma kapsamında, Parzen Window (Bölüm 2.3.4), K_NN (Bölüm 0), Kovaryans Analizi (Bölüm 2.3.6) ve Ayrık Kosinüs Dönüşümü (Bölüm 2.3.7) yöntemleri incelenmiş ve hedefin sınıflandırılmasında hangi yöntemin daha uygun olduğu değerlendirilmiştir (Bölüm 4.2.5).

Geliştirilen Otomatik Hedef Tanıma Algoritmasında ise, resimden çıkarılmış her nesne için DCT değerleri hesaplanır ve önceden elde edilmiş hedef DCT değerleri ile karşılaştırılır; minimum fark yaratan nesne ise istenilen hedeftir.



Şekil 4.16 DCT ile Hedef Tanıma

4.4.1 Şekil Tanıma Algoritmaları Değerlendirmesi

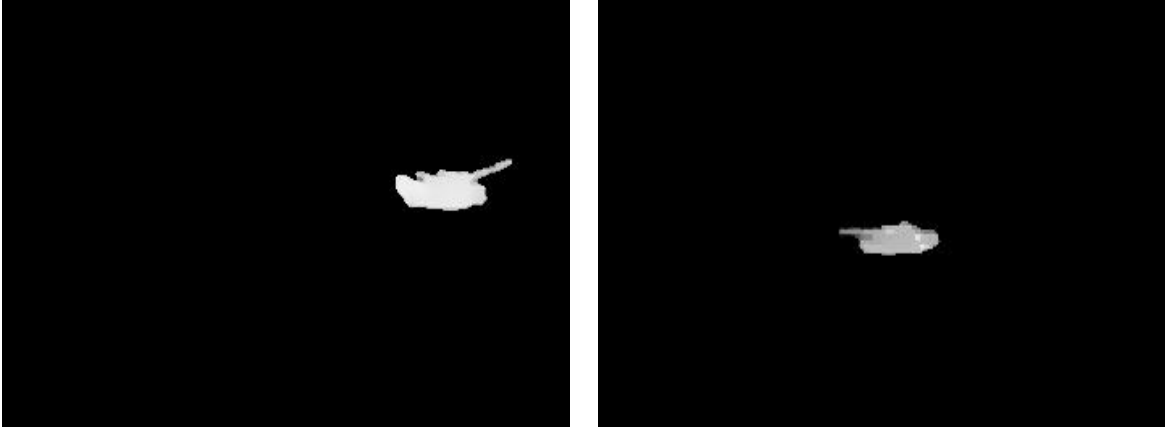
Şekil tanıma algoritmaları kapsamında bir çok algoritma incelenmiştir. Bu algoritmalar içinden Parzen window, En yakın K_Komşu, Kovaryans Analizi ve DCT algoritmalarıyla testler yapılmıştır.

Yapılan testler sonucunda; Parzen Window yönteminin başarımlarının seçilen pencere boyutları ile bire bir ilgili olduğu, en yakın K_Komşu Yönteminin arka plana ve diğer hedef tiplerine ait geniş bir veri tabanına ihtiyaç duyduğu; dolayısıyla işlem yükünü gereksiz biçimde artıracak, Kovaryans analizi yönteminin içerisinde sadece hedefe ait resim varken düzgün çalıştığı, DCT yönteminin ise resim içerisine kandırma amaçlı hedef benzeri görüntüler koyulmasına rağmen, Kovaryans işleminden daha kısa sürede, hedefi ayırt edebildiği (Şekil 4.16) görülmüştür.

Testler neticesinde kullanılan resimler üzerinde en gürbüz ve kısa sürede çalışan Şekil Tanıma Algoritmasının DCT Yöntemi olduğu ortaya çıkmıştır. DCT algoritması sonucunda elde edilen neticeler Şekil 4.16, Şekil 4.17 ve Şekil 4.18’de verilmiştir.

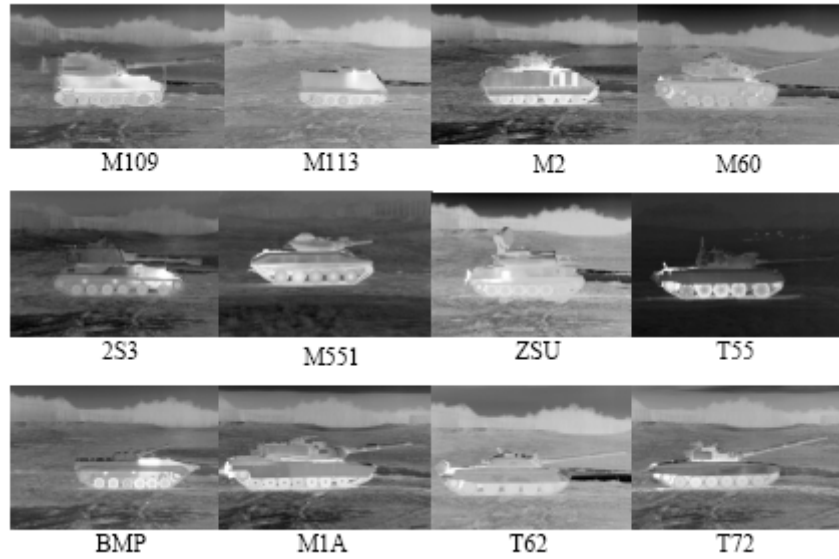


Şekil 4.17 1. ve 2. Resimlerden Elde Edilmiş Hedef Görüntüleri



Şekil 4.18 3 ve 4.Resimlerden Elde edilmiş Hedef Görüntüleri

Yine Şekil 4.19’de de örnek bir hedef veri tabanı görülmektedir. Yapılan algoritma çalışmalarının performanslarını ölçmek ve farklı koşullarda gürbüz sonuçlar elde edebilmek için geniş bir hedef veri tabanına ihtiyaç duyulduğu değerlendirilmektedir.



Şekil 4.19 Hedef Veri Tabanı Örneği

Şekil 4.19’deki veri tabanı örneğinde farklı hedeflere ait birer resim olmasına karşın, ATR algoritmasının gürbüz bir şekilde doğru sonuçlar verebilmesi için hedeflere ait farklı açılardan çekilmiş bir çok resme ihtiyaç duyulmaktadır. Hedefe ait birden fazla resim sayesinde, hedefe farklı açılardan da yaklaşılsa ilgili hedefler tanımlanabilecektir.

4.5 GELİŞTİRİLEN ALGORİTMANIN UYGULAMA ALANLARI

Geliştirilen yöntemin kara araçları tanıma (Bölüm 4.6.3) uygulamasının yanı sıra; İnsan Tespiti (Bölüm4.5.1), Hava Aracı Tanıma (Bölüm4.6.1), Deniz Aracı Tanıma (Bölüm0), Virüs Tanıma (Bölüm4.7.1), Bakteri Sayma (Bölüm4.7.1) ve Akyuvar Hücre Sınıflandırma (Bölüm4.7.3) uygulamalarında da kullanılabileceği değerlendirilmektedir ve uygulama alanları örnekleri ile bölüm içerisinde anlatılmaktadır.

4.5.1 İNSAN TESPİTİ

Geliştirilen ATR Yönteminin insan tespitinde de kullanılabileceği değerlendirilmiş, ve elde edilebilen termal görüntüler üzerinde denenmiştir.



Orijinal Resim



Hedef

Şekil 4.20 İnsan Tespiti Uygulaması 1

Birinci uygulamada(Şekil 4.20), resim içerisinde yer alan tek insan, şablona uygun olarak kolaylıkla otomatik olarak tespit edilebilmiştir.



Şekil 4.21 İnsan Tespiti Uygulaması 2

İkinci uygulamada(Şekil 4.21) ise birinci uygulamadan farklı olarak resim içerisinde üç farklı insan bulunmaktadır ve algoritma üçünü de ayırt edebilmiştir.



Orijinal Resim

Hedef

Şekil 4.22 İnsan Tespiti Uygulaması 3

Üçüncü uygulamada(Şekil 4.22) iki insan görülmektedir, insanlardan bütün olarak görülebilen yöntem tarafından ayırt edilebilirken, yarısı görünmeyen insanı algoritma ayırt edememiştir.



Şekil 4.23 Veri Tabanı İnsan Resmi

Resimlerlerle karşılaştırmak için kullanılan hedef şablonu (Şekil 4.23) başka bir resimden elde edilmiş olup rasgele seçilmiştir. Sonuçlar incelendiğinde algoritmanın insan tespitinde de kullanılabileceği değerlendirilmiştir.

4.6 TAŞIT TANIMA

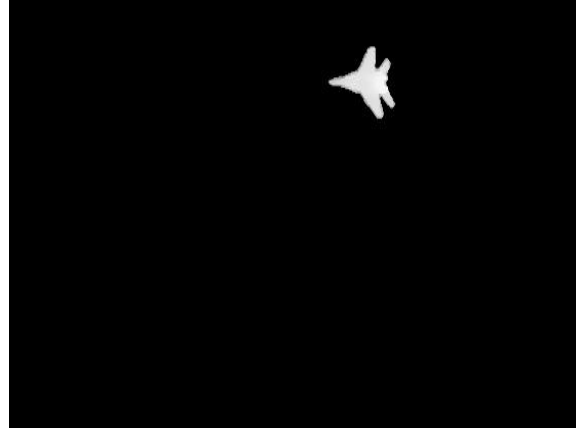
Taşıt tanıma uygulamaları kapsamında hava aracı uygulamaları için uçak , deniz aracı uygulamaları için gemi ve kara aracı uygulamaları için de tank tercih edilmiştir.

4.6.1 Hava Aracı Tanıma

Uçaklar, kara araçları ile boyut olarak çok farklı olmasalar da; yüksek hızlarda hareket ettiklerinden yüzeyleri oldukça ısınmaktadır. Özellikle karadan havaya termal olarak görüntülenen uçağın arka plan gürültüsünün de oldukça düşük olacağı ve kara araçlarına göre çok daha kolay ayırt edilebileceği değerlendirilmektedir. Geliştirilen ATR Yönteminin bahsedilen nedenlerden ötürü; hava araçlarını tanımak için de rahatlıkla kullanılabileceği değerlendirilmiştir.



Orijinal Resim



Hedef

Şekil 4.24 Hava Aracı Uygulaması

ATR yöntemi, elde edilebilen termal uçak görüntüleri üzerinde(Şekil 4.24) veri tabanı resmi(Şekil 4.25) kullanılarak test edilmiş ve sorunsuz çalıştığı görülmüştür.



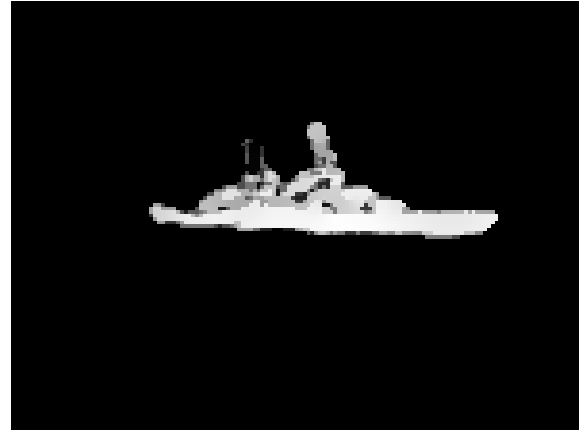
Şekil 4.25 Veri Tabanı Uçak Resmi

4.6.2 Deniz Aracı Tanıma

Gemiler, kara araçlarına göre daha büyük olduklarından ve arka plan ile hedef kontrast farklılıkları kara araçlarına göre oldukça yüksek olduğundan; geliştirilen ATR Yönteminin gemi tanımak için de rahatlıkla kullanılabileceği değerlendirilmiştir. ATR yöntemi, elde edilebilen Savaş Gemisi görüntüleri üzerinde denenmiştir.



Orijinal Resim



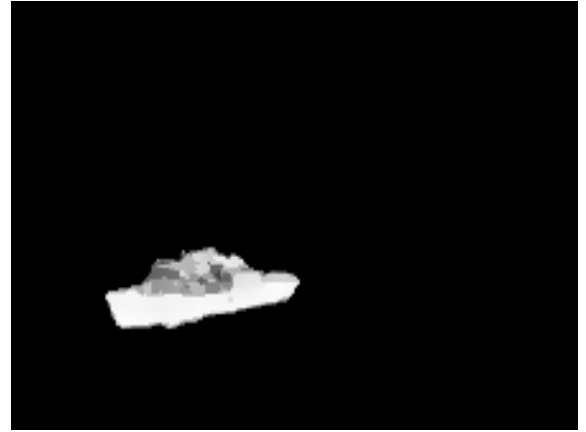
Hedef

Şekil 4.26 Deniz Aracı Uygulaması 1

Birinci uygulamada(Şekil 4.26), resim içerisinde yer alan tek gemi, şablona(Şekil 4.29) uygun olarak kolaylıkla otomatik olarak ayırt edilebilmiştir.



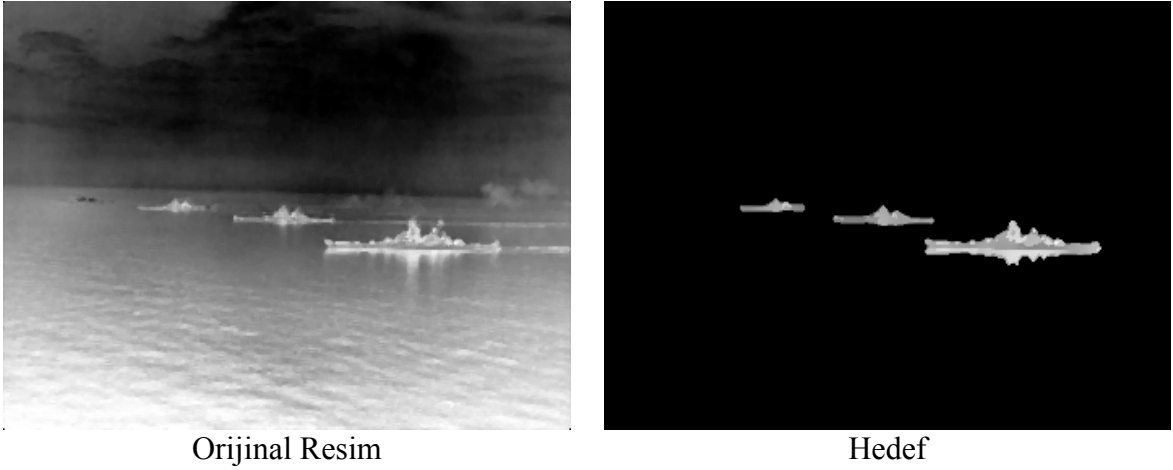
Orijinal Resim



Hedef

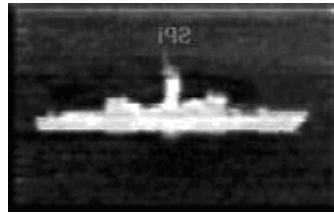
Şekil 4.27 Deniz Aracı Uygulaması 2

İkinci uygulamada(Şekil 4.27) ise, sivil bir gemi içeren resim üzerinde algoritma çalıştırılmış ve resim içerisinde yer alan tek gemi, şablona(Şekil 4.29) uygun olarak kolaylıkla otomatik olarak ayırt edilebilmiştir.



Şekil 4.28 Deniz Aracı Uygulaması 3

Üçüncü uygulamada (Şekil 4.28) ise birinci uygulamadan farklı olarak resim içerisinde dört farklı gemi bulunmaktadır. Johnson Kriterlerine(Bölüm 3.3.1) göre minimum tanıma koşullarını sağlayamayan 4. gemi ayırt edilememiştir fakat; kalan üç gemi de algoritma tarafından ayırt edilebilmiştir.



Şekil 4.29 Veri Tabanı Gemi Resmi

4.6.3 Kara Aracı Tanıma

Tez çalışması kapsamında ilgilenilen en önemli hedef şüphesiz ki kara araçları olmuştur. Kara araçları üzerinde geliştirilecek bir yöntemin gerek arka plan gürültüsünün daha fazla olması, gerekse de termal izlerinin düşük olması nedeniyle diğer uygulama alanlarına göre çok daha zor olduğu değerlendirilmektedir. Geliştirilen yöntem de özellikle kara araçlarının termal görüntüleri üzerinde çalıştırılmak üzere geliştirilmiştir. Yöntemin kara araçları üzerindeki performansı, geliştirilen ATR yönteminin detaylı olarak açıklandığı 4 . Bölümde aşama aşama, örnek resimlerle (Şekil 4.2) birlikte anlatılmıştır.

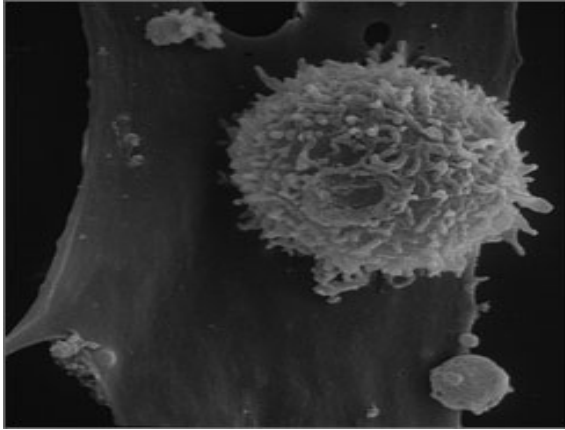
4.7 MEDİKAL GÖRÜNTÜLEME

Medikal görüntüleme ve görüntü işleme üzerine çeşitli araştırmalar yapıldığı bilinmektedir. Medikal görüntüler üzerinden görüntü işleme teknikleri ile neler yapıldığına ilişkin literatür taraması yapıldığında; alyuvar sayımı, akyuvar tanıma, akyuvar sınıflandırma, virüs tanıma, bakteri tanıma, parazit tanıma, kromozom sınıflandırma konuları üzerine bir çok tez ve makale çalışması yapıldığı görülmüştür.

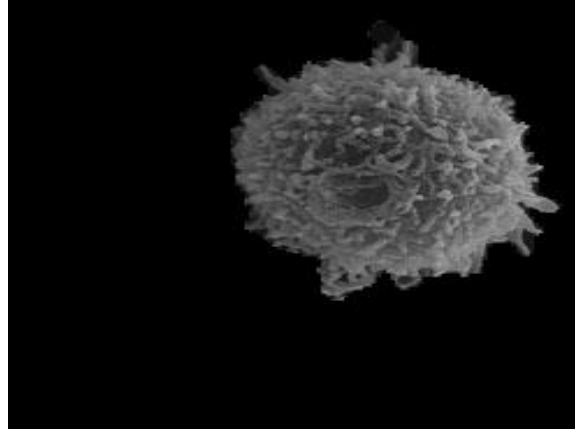
Bahsedilen konu başlıklarının her birinde geliştirilen ATR yönteminin kullanılabileceği değerlendirilmektedir ve elde edilebilen görüntüler üzerinden, otomatik virüs tanıma (Bölüm 4.7.1), otomatik bakteri tanıma (Bölüm 4.7.2) ve akyuvar hücresi sınıflandırma (Bölüm 4.7.3), için geliştirilen yöntem kullanılmış ve başarılı sonuçlar elde edilmiştir.

4.7.1 Otomatik Virüs Tanıma

Yöntemin denendiği bir diğer araştırma konusu ise virüs tanımadır, geliştirilen yöntem kullanılarak, farklı testlere gerek kalmaksızın mikroskop görüntüsü otonom olarak işlenip teşhis koyulabilir.



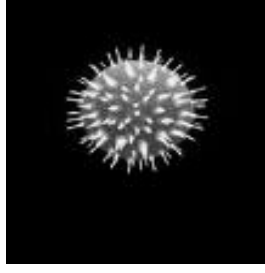
Kan içerisinde HIV Virüsü



Ayırt edilmiş HIV virüsü

Şekil 4.30 Virüs Tanıma Uygulaması

Laboratuar ortamında elektron mikroskobu ile çekilmiş HIV virüsü içeren kan resmi ve mikroskop ile çekilmiş virüs resmi görülmektedir. Virüs içeren elektron mikroskobu görüntüsü (Şekil 4.30) üzerinden veri tabanındaki virüs görüntüsü (Şekil 4.31) kullanılarak virüs ayırt edilmiştir.

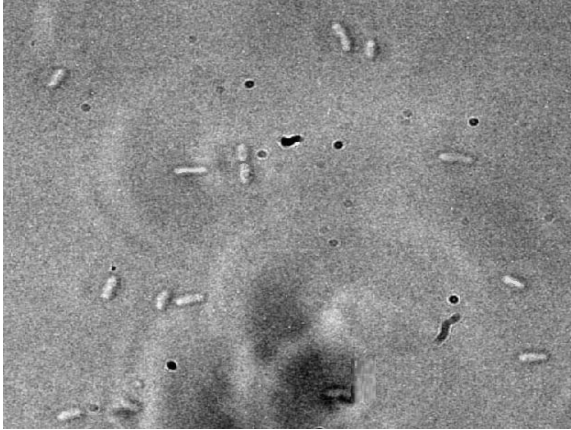


Şekil 4.31 Veri tabanı Virüs Resmi

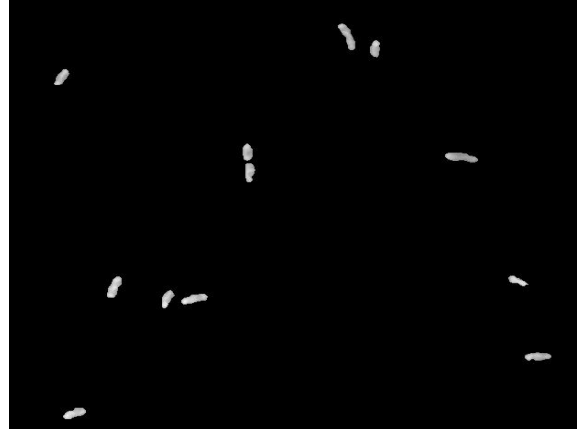
4.7.2 Otomatik Bakteri Tanıma

Sanayileşmenin bir sonucu olarak oldukça artan çevre kirliliği özellikle su kaynaklarını etkilemektedir. Su kirliliğinin yol açtığı problemleri çözebilmek için de su analizi, önem taşımaktadır. Su kirliliğini ölçebilmenin en kolay yöntemi içerisindeki zararlı bakteri ve parazit sayısını belirlemektir.

Geliştirilen yöntem, E.Coli bakterileri içeren resim (Şekil 4.32) üzerinde uygulanmış ve E. Coli Bakterileri kolayca ayırt edilebilmiştir.



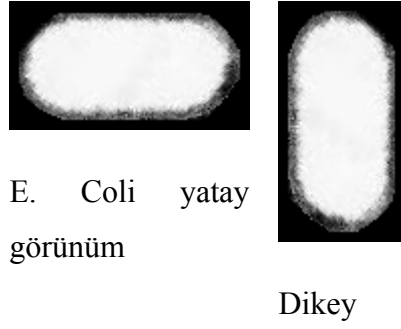
E.Coli Bakterileri içeren sıvı



Tanımlanabilen E.Coli Bakterileri

Şekil 4.32 Bakteri Tanıma uygulaması

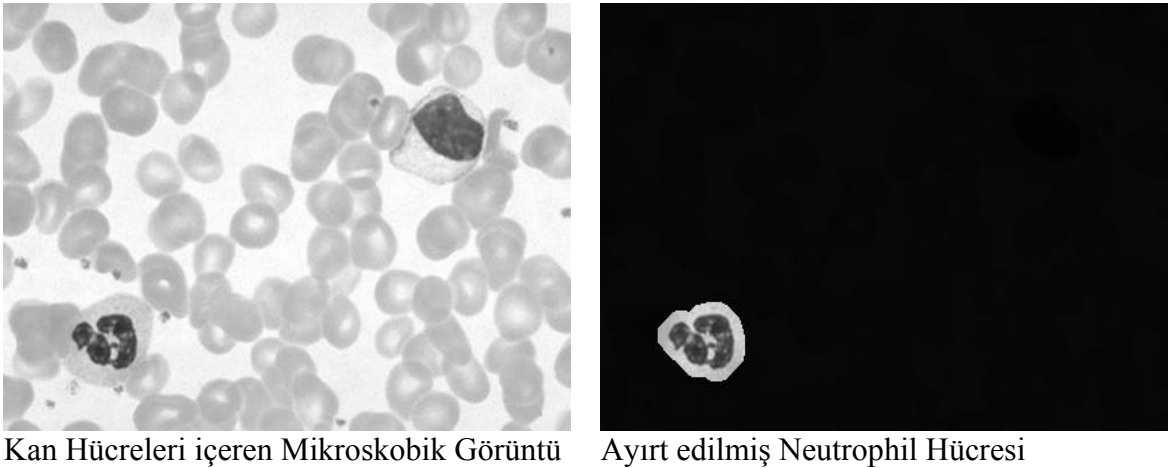
E.Coli bakterilerini(Şekil 4.32) ayırt edebilmek için veri tabanı olarak; Şekil 4.33'deki resimler yatay ve dikey konumlu bakteriler için ayrı ayrı kullanılmıştır. 14 tane bakteriden 13 tanesi tanımlanabilmiştir, tanımlanamayan bakterinin ise bölümlleme işleminde kaybedildiği görülmüştür.



Şekil 4.33 Veri Tabanı E.Coli Bakterisi Resimleri

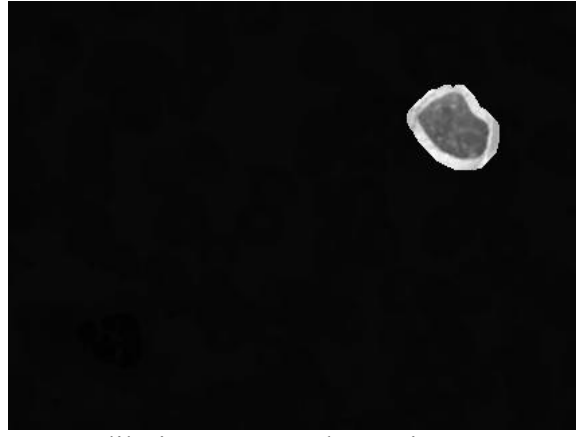
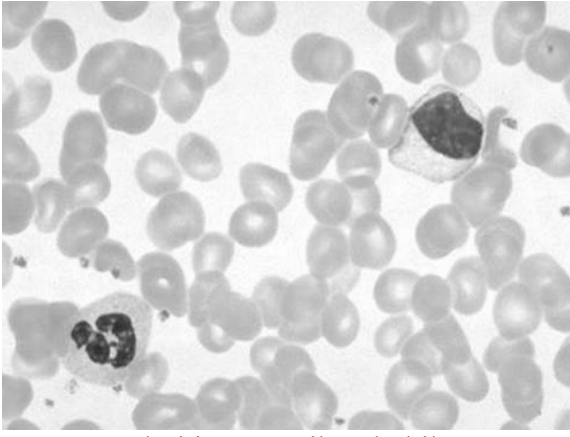
4.7.3 Otomatik Akyuvar Hücresi Sınıflandırma

Akyuvar hücreleri incelendiğinde, akyuvarların Basophil, Eosinophil, Lymphocyte, Monocyte ve Neutrophil türlerinin olduğu görülmüştür (Alfred R. J. Katz,9 Şubat 2000). Kan içerisindeki akyuvar sayılarındaki değişimlerin farklı hastalıkların habercileri olduğu bilinmektedir. Alınan bir kan örneği üzerinden akyuvar türlerinin ve sayımının otomatik olarak ve yüksek doğrulukla yapılabilmesinin de zaman kazandıracağı ve teşhisi kolaylaştıracağı değerlendirilmektedir.



Şekil 4.34 Akyuvar Sınıflandırma Uygulaması 1

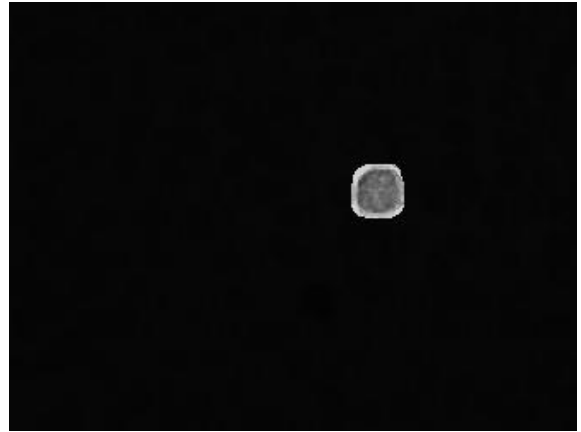
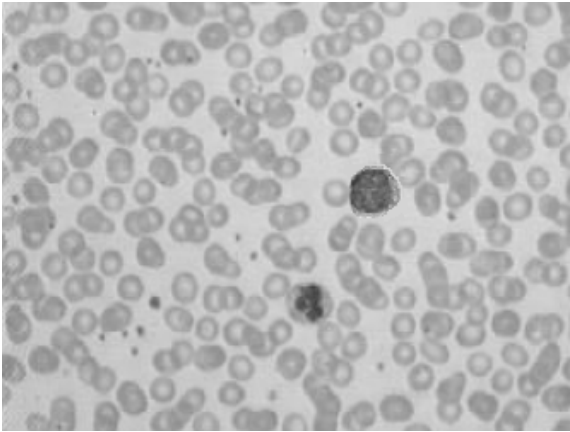
Elde edilen mikroskobik akyuvar resimleri(Şekil 4.37) kullanılarak üç farklı akyuvar hücresi içeren resim üzerinde (Şekil 4.34, Şekil 4.35 ve Şekil 4.36) geliştirilen yöntem denenmiş ve istenilen akyuvar hücreleri ayırt edilebilmiştir.



Kan Hücreleri içeren Mikroskopik Görüntü Ayırt edilmiş Monocyte hücresi

Şekil 4.35 Akyuvar Sınıflandırma Uygulaması 2

Akyuvar hücrelerinin kolay tanınmasını sağlayan en önemli etken alyuvarların yassı bir şekli varken, akyuvarların küresel bir şekli olması ve mikroskop görüntüsündeki iki boyutlu kesitin çok benzer bir görüntü oluşturmasıdır.



Kan Hücreleri içeren Mikroskopik Görüntü Ayırt edilmiş Lymphocyte hücresi

Şekil 4.36 Akyuvar Sınıflandırma Uygulaması 3

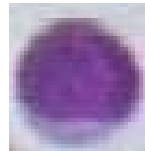
Şekil 4.37'deki veri tabanı resimlerine bakıldığında akyuvar hücrelerinin sitoplazmalarının ayırt edilmesi kolay görüntüler oluşturduğu görülmektedir.



Netrophil



Monocyte



Lymphocyte

Şekil 4.37 Veri Tabanı Akyuvar Resimleri

5. SONUÇ

ATR, bilgisayarların, yapay algılamanın ve otonom sistemlerin önem kazandığı bir dönemde ihtiyaç duyulan ve gerçekleştirilmesi gereken güncel bir konu olduğu düşünülmektedir. Tabi ki ATR algoritma düzeyinden gerçek dünyaya aktarıldığında önem kazanacaktır. ATR'nin hangi şartlarda çalışması gerektiği; hava koşulları, görüntü kalitesi, yanlış alarm oranı, görev çok iyi belirlenmesi gereken ATR limitleridir; limitler ne kadar iyi belirlenirse başarı oranı da o kadar yükselecektir.

Tablo 5.1 ATR Algoritma Performans Testi

Bilgisayar Konfigürasyonu				
İşlemci	Çift Çekirdek 2.4 Ghz			
Ram	1 Gbyte			
İşletim Sistemi	Windows XP Professional			
Progralama Dili	Matlab R2006b			
Kod Uzunluğu (Satır)	176			
Test Resimleri Çözünürlüğü	320x240			
Çalışma süresi (ms)	1. Resim	2. Resim	3. Resim	4. Resim
	4322.452	4272.710	4088.088	3769.753

Yapılan ATR çalışmasında kullanılan Görüntü İşleme Teknikleri 4. Bölümde, test resimleri üzerinden ayrıntılı olarak anlatılmıştır. Ayrıntıları verilen ATR algoritması, Matlab programlama dili üzerinde gerçekleştirilmiş, Tablo 5.1'de özellikleri verilen bir donanım üzerinde örnek resimler belirtilen sürelerde işlenmiştir.

Yapılan çalışma, algoritmanın 4 saniye civarında bir çalışma süresi olduğunu göstermektedir. Çalışmada kullanılan kodun optimize edilmemesi, yavaş çalışan bir programlama dili olan

Matlab’da hazırlanması ve gerçek zamanlı bir işletim sistemi olmayan Windows XP üzerinde çalıştırılması algoritmanın daha da hızlı çalışabileceğini göstermektedir.

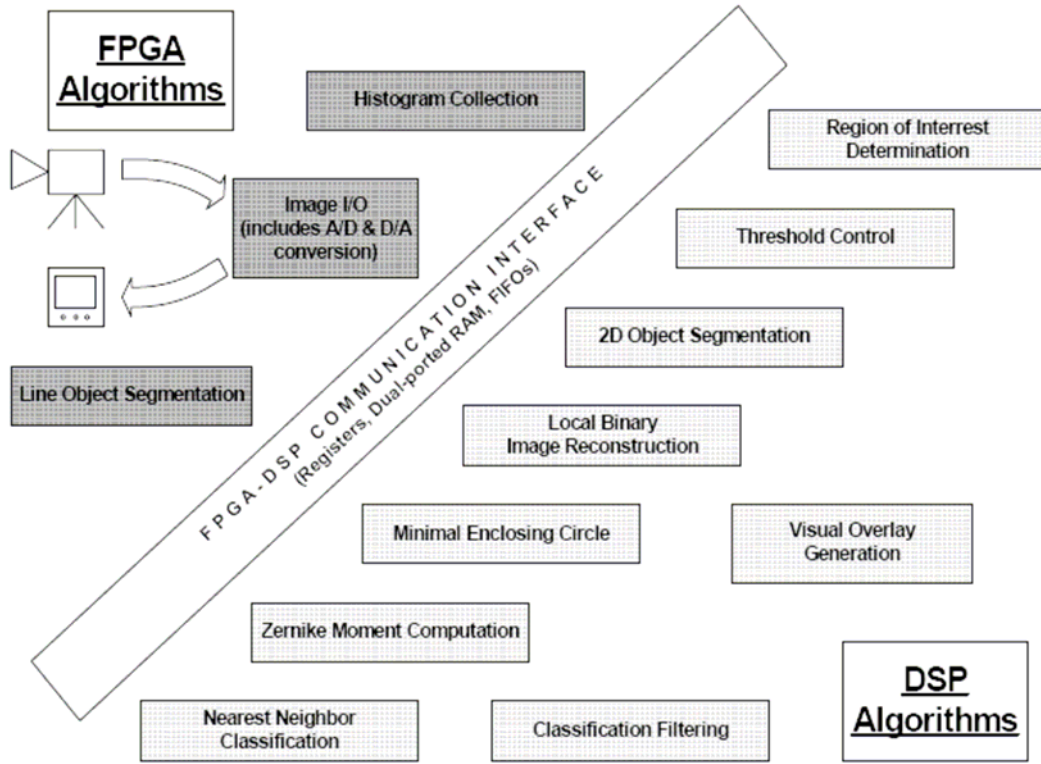


Fig. 2 Algorithm architecture – deployment on FPGA and DSP

Şekil 5.1 FPGA ve DSP ile Donanımsal Algoritma Mimarisi (Schlosser ve Fecht,2005)

Gerçek zamanlı bir işletim sistemine sahip, Şekil 5.1’ deki gibi görev dağılımı iyi yapılmış, DSP/FPGA işlemcilerle sahip bir elektronik kart üzerinde koşturulduğunda; algoritmanın çok daha kısa sürelerde çalışabileceği değerlendirilmektedir. Yakın dönem hedefleri ise, algoritmanın gürbüzleştirilip, optimize edilmesi ve belirtilen ortamda, gerçek zamanlı koşturulmasıdır.

KAYNAKLAR

Alfred R. J. Katz (2000), Image Analysis and Supervised Learning in the Automated Differentiation of White Blood Cells from Microscopic Images, Yüksek Lisans Tezi, RMIT Üniversitesi Bilgisayar Bölümü

B. Bhanu and R. Holben (1990), "Model-based segmentation of FLIR images", IEEE Trans. Aerospace Electron. Syst., 26 : 2 – 10

Bir Bhanu (1986), "Automatic Target Recognition: State of the Art Survey", IEEE Transactions on Aerospace and Electronic Systems, AES-22 : 4

Bruce A. Draper (1993), Learning Object Recognition Strategies, Colorado State Üniversitesi, Bilgisayar Bilimleri Doktora Tezi

Canny, J.F. (1986), "A computational approach to edge detection", IEEE Transactions on Pattern Analysis and Machine Intelligence, 8(6): 679-698

Clark F. Olson, Daniel P. Huttenlocher (1997), "Automatic Target Recognition by Matching Oriented Edge Pixels", IEEE Transactions on Image Processing, 6.1:103-113

Dong-San Jun, Yuseong-gu, HyunWook Park, Yuseong-gu (2005), "Automatic Target Detection Using Binary Template Matching", SPIE, Optical Engineering, 44(3): 036401

Hairong Qi, Wesley E. Snyder, David Marchette (2000), "Efficient Approach To Segment Man-Made Targets From Unmanned Aerial Vehicle Imagery", SPIE Optical Engineering, 39 :5

Harney Robert (2005), "Imaging and Image-based Perception", Combat Systems vol.1

James A. Ratches, C.P. Walters, Rudolf G. Buser, and B. D. Guenther (1997), "Aided and Automatic Target Recognition Based Upon Sensory Inputs From Image Forming Systems", IEEE Trans. On Pattern Analysis And Machine Intelligence, 19: 9

John Johnson (1958), "Analysis of image forming systems," Image Intensifier Symposium, AD 220160:244–273.

Kenan Kürekli (2004), System Parameter Adaptation Based On Image Metrics For Automatic Target Detection, Yüksek Lisans Tezi, ODTÜ Fen Bilimleri Enstitüsü

Lindsay I Smith (2002), "A tutorial on Principal Components Analysis", Otago Üniversitesi, cosc453 ders notu

Michael J. Black, Guillermo Sapiro, David H. Marimont, David Heeger (1998), "Robust Anisotropic Diffusion", IEEE Transactions On Image Processing, 7:3

Michael W. Roth (1990), "Survey of Neural Network Technology for Automatic Target Recognition", IEEE Transactions on Neural Networks, 1:1

Otsu, N. (1979), "A Threshold Selection Method from Gray-Level Histograms," IEEE Transactions on Systems, Man, and Cybernetics, 9.1: 62-66

P. Perona and J. Malik (1990), "Scale-Space And Edge Detection Using Anisotropic Diffusion," IEEE Trans. Pattern Anal. Machine Intell., 12: 629–639

Rafael c. Gonzalez, Richard E. Woods, Steven L. Eddins (2004), Digital Image Processing using MATLAB, Prentice Hall

Sandor Z. Der, Rama Chellappa (1997), "Probe-Based Automatic Target, Recognition in Infrared Imagery", IEEE Transactions On Image Processing, 6:1

Talu, F., "Model Tabanlı Yapay Sinir Ağları Kullanılarak Görüntü Kenarının Yakalanması", Fırat Üniversitesi, 2004

Tuba Makbule Bayık (2004), Automatic Target Recognition In Infrared Imagery, Yüksek Lisans Tezi, ODTÜ Fen Bilimleri Enstitüsü

William H. Licata (2001), "Automatic Target Recognition (ATR) Beyond the Year 2000", NATO Konferansı RTO-EN-018, 28-29 Haziran 2001, İsveç, Stokholm

W. M. Brown, C. W. Swonger, "A Prospectus for Automatic Target Recognition", IEEE Transactions on Aerospace and Electronic Systems, 25: 3

Wolfgang Schlosser, Arno von der Fecht (2005), "Real-time automatic target recognition using Zernike moments", Proc. of SPIE, 5988 59880I : 6

W. Zhao, R. Chellappa, R.J. Phillips (1999), "Subspace Linear Discriminant Analysis for Face Recognition", Maryland Üniversitesi, Computer Vision Laboratory, Center for Automation Research

EK_1 MATLAB Algoritma Analiz Kodu

```

b1=fix(bound1((targetind-1)*4+1));
b2=fix(bound1((targetind-1)*4+2));
b3=fix(bound1((targetind-1)*4+3));
b4=fix(bound1(targetind*4));
wins=3;
Tarmap=BW2(b2-wins:(b2+b4+wins),b1-wins:(b1+b3+wins));
Backmap=1-Tarmap;
Orgpic=Resim(b2-wins:(b2+b4+wins),b1-wins:(b1+b3+wins));
Histeqpik=double(pich(b2-wins:(b2+b4+wins),b1-wins:(b1+b3+wins)));
Histsumpic=double(pic3(b2-wins:(b2+b4+wins),b1-wins:(b1+b3+wins)));
AFpic=double(picrad(b2-wins:(b2+b4+wins),b1-wins:(b1+b3+wins)));
%figure,imshow(Backmap);
%figure,imshow(uint8(Orgpic));
figure,imshow(tarpic);
Orgtar=Orgpic.*Tarmap;
OrgBack=Orgpic.*Backmap;
[ts1 ts2]=size(Orgpic);
np=1;
np2=1;
for i=1:ts1
    for j=1:ts2
        if(Tarmap(i,j)==1)
            TO(np)=Orgpic(i,j);    %tek boyutlu gerçek hedefresmi
            TH(np)=Histeqpik(i,j);
            THS(np)=Histsumpic(i,j);
            TA(np)=AFpic(i,j);
            np=np+1;
        end
        if(Tarmap(i,j)==0)
            BO(np2)=Orgpic(i,j); %tek boyutlu gerçek arkaplan resmi
            BH(np2)=Histeqpik(i,j);
            BHS(np2)=Histsumpic(i,j);
            BA(np2)=AFpic(i,j);
            np2=np2+1;
        end
    end
end
mto=mean(TO)
mbo=mean(BO)
sto=std(TO)
sbo=std(BO)
vto=var(TO)
vbo=var(BO)
kto=kurtosis(TO)-3
kbo=kurtosis(BO)-3
eto=entropy(uint8(TO))
ebo=entropy(uint8(BO))

mth=mean(TH)
mbh=mean(BH)

```

```
sth=std(TH)
sbh=std(BH)
vth=var(TH)
vbh=var(BH)
kth=kurtosis(TH)-3
kbh=kurtosis(BH)-3
eth=entropy(uint8(TH))
ebh=entropy(uint8(BH))
```

```
mths=mean(THS)
mbhs=mean(BHS)
sths=std(THS)
sbhs=std(BHS)
vths=var(THS)
vbhs=var(BHS)
kths=kurtosis(THS)-3
kbhs=kurtosis(BHS)-3
eths=entropy(uint8(THS))
ebhS=entropy(uint8(BHS))
```

```
mta=mean(TA)
mba=mean(BA)
sta=std(TA)
sba=std(BA)
vta=var(TA)
vba=var(BA)
kta=kurtosis(TA)-3
kba=kurtosis(BA)-3
eta=entropy(uint8(TA))
eba=entropy(uint8(BA))
imwrite(Tarmap,'tarm.bmp');
```

EK_2 MATLAB Algoritma Kodu

```

clc;
clear all;
close all;
tic
%Resim=imread('11.bmp');

Resim=imread('SpTank.bmp');
picdb=imread('DataBaseT.jpg');
[dbs1 dbs2 dbs3]=size(picdb);
if (dbs3==3)
    picdb=rgb2gray(picdb);
end
BWdb=edge(picdb,'canny');
%Resim=imread('s1s.bmp');
%Resim=255-Resim;

[s1 s2 s3]=size(Resim);
if (s3==3)
    Resim=rgb2gray(Resim);
end

% Resim=255-Resim;
figure,imshow(Resim);

pich=histeq(Resim);
imwrite(pich,'P1histeq.bmp');
figure,imshow(pich);
Resim=fix((double(pich)+double(Resim))/2);
figure,imshow(uint8(Resim));
imwrite(uint8(Resim),'P1histsum.bmp');
Resim=double(Resim);

%Resim=imresize(Resim,0.5);
%[s1 s2 ]=size(Resim);
lamda=1;
looprad=25;    %loop number
picrad=Resim;
gf=12;
Ns=0;
Sn=0;
for ii=1:looprad
    for i=2:s1-1
        for j=2:s2-1

            rad=Resim(i,j);
            for k=1:3
                for l=1:3
                    ncol=Resim(i+k-2,j+l-2);
                    if abs(ncol-rad)<gf

```

```

        Sn=Sn+(ncol-rad);
        Ns=Ns+1;
    end
end
end
picrad(i,j)=round(rad+(lamda/Ns).*Sn);
end
end
ii
end

maxrad=max(picrad(:));
minrad=min(picrad(:));
ranrad=maxrad-minrad;
picrad=(picrad-minrad).*255/ranrad;
picrad=uint8(picrad);
figure,imshow(picrad);
imwrite(picrad,'P1diffusion.bmp');
%picrad=imresize(picrad,2);
[picEd2 thre]=edge(picrad,'canny');
% thre
% %thre=thre.*1.3;
% thre=thre.*0.8;
%picEd2=edge(picrad,'canny',thre);
figure,imshow(picEd2);
imwrite(picEd2,'P1Edges.bmp');
picEd2=bwareaopen(picEd2,10);
imwrite(picEd2,'P1areaopen.bmp');
figure,imshow(picEd2);
% msk=[0 0 0 0 0;
%      0 1 1 1 0;
%      0 1 1 1 0;
%      0 1 1 1 0;
%      0 0 0 0 0;];
% msk=[0 0 0 0;
%      0 1 1 0;
%      0 1 1 0;
%      0 0 0 0;];
% picEd2=conv2(double(picEd2),double(msk));
% figure,imshow(picEd2);
streli=[0 1 0;1 1 1;0 1 0];
%streli=[0 1 1 0;1 1 1 1;1 1 1 1;0 1 1 0];
picEd2=imclose(picEd2,streli);
figure,imshow(picEd2);
imwrite(picEd2,'P1close.bmp');
picEd2=imfill(picEd2,'holes');
figure,imshow(picEd2);
imwrite(picEd2,'P1fill.bmp');
picEd2=medfilt2(picEd2,[3 3]);
figure,imshow(picEd2);
imwrite(picEd2,'P1medfilt.bmp');

```

```

RGB=label2rgb(picrad);
figure,imshow(RGB);
imwrite(RGB,'P1colored1.bmp');
picEd2=bwareaopen(picEd2,30);
figure,imshow(picEd2);
imwrite(picEd2,'P1areopen2.bmp');
LabEd=bwlabel(picEd2);
RGB2=label2rgb(LabEd);
figure,imshow(RGB2);
imwrite(RGB2,'P1colored2.bmp');
% p1=regionprops(LabEd,'Perimeter');
% per1=[p1.Perimeter];
% pL1=regionprops(LabEd,'PixelList');
% pixList1=[pL1.PixelList];
s1=regionprops(LabEd,'Solidity'); %%icindeki bosluklari olcer
%solid1=[s1.Solidity];
idx = find([s1.Solidity] > 0.65);
BW2 = ismember(LabEd,idx);
figure,imshow(BW2);

```

```

LabEd2=bwlabel(BW2);
a1=regionprops(LabEd2,'Area');
%area1=[a1.Area];
idx = find([a1.Area] > 100);
BW2 = ismember(LabEd2,idx);
figure,imshow(BW2);
imwrite(BW2,'P1objectcleared.bmp');
LabEd2=bwlabel(BW2);
bo1=regionprops(LabEd2,'BoundingBox');
bound1=[bo1.BoundingBox];
sBn=size(bound1);
sBn=sBn(2)/4;
clear idx;
n=1;
% close all;
for i=1:sBn

    if(bound1((i-1)*4+3)<10)|| (bound1(i*4)<10)
        bound1((i-1)*4+1)=0;
        bound1((i-1)*4+2)=0;
        bound1((i-1)*4+3)=0;
        bound1((i-1)*4+4)=0;
    end
    if(bound1((i-1)*4+3)>9)&&(bound1(i*4)>9)
        clear Mpic;
        clear Mpic2;
        idx(n)=i;
        b1=fix(bound1((i-1)*4+1));
        b2=fix(bound1((i-1)*4+2));
        b3=fix(bound1((i-1)*4+3));
    end
end

```

```

b4=fix(bound1(i*4));
Mpic=BW2(b2:(b2+b4),b1:(b1+b3));
[ms1 ms2]=size(Mpic);
Mpic2=zeros([ms1+2,ms2+2]);
Mpic2=logical(Mpic2);
Mpic2(2:ms1+1,2:ms2+1)=Mpic;
Mpic2=imresize(Mpic2,[dbs1 dbs2]);
BWM=edge(uint8(Mpic2),'canny');
covpic=dct2(double(BWdb));
covp2=dct2(double(BWM));
MatchDif(n)=sum(sum(abs(covpic-covp2)));
%Mind(n)=i;
n=n+1;
end
end
[Mlev Mindn]=min(MatchDif);
targetind=idx(Mindn);
target=ismember(LabEd2,targetind);
% imwrite(target,'targetbw.jpg');
figure,imshow(target);
imwrite(target,'P1targetbw.bmp');
tarpic=Resim.*double(target);
tarpic=uint8(tarpic);
figure,imshow(tarpic);
imwrite(tarpic,'P1target.jpg');
%tarpicRGB=double(RGB).*double(target);
%tarpicRGB=uint8(tarpicRGB);
%figure,imshow(tarpicRGB);
% imwrite(tarpicRGB,'targetRGB.jpg');
Mlev
toc

```

Ek_3 Homojen Olmayan Yayınım Matlab Uygulama Kodu

```

clc;
clear all;
close all;
p1=imread('P1h.bmp');
pic1=p1(90:100,119:129);
looprad=25;    %loop number
picrad=pic1;
pic2=pic1;
pic1
[s1 s2]=size(pic1);
gf=16;
lamda=1;
imshow(uint8(pic1))
Ns=0;
Sn=0;
for ii=1:looprad
    for i=2:s1-1
        for j=2:s2-1

            rad=pic1(i,j);
            for k=1:3
                for l=1:3
                    ncol=pic1(i+k-2,j+l-2);
                    if abs(ncol-rad)<gf
                        Sn=Sn+(ncol-rad);
                        Ns=Ns+1;
                        %i+k-2,j+l-2
                        %ncol-rad
                        %pause
                    end
                end
            end
            picrad(i,j)=round(rad+(lamda/Ns).*Sn);
        end
    end
end
ii
end
pic2,pic1
figure,imshow(uint8(pic1))
pic1-pic2

```

ÖZGEÇMİŞ

Doğum tarihi 31.05.1982

Doğum yeri Antalya

Lise 1997-2000 Isparta Süleyman Demirel Fen Lisesi

Lisans 2000-2005 İstanbul Teknik Üniversitesi
Elektrik Elektronik Fak.
Elektronik ve Haberleşme Mühendisliği Bölümü

Yüksek Lisans 2005-2008 Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü
Elektrik Elektronik Mühendisliği Anabilim Dalı
Haberleşme Programı

Çalıştığı kurum(lar)

2006-Devam ed. ROKETSAN
Sistem Mühendisi