



T.C.
SELÇUK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

YAPAY ARI KOLONİSİ OPTİMİZASYONU İLE KENAR BULMA

Elif Deniz YİĞİTBAŞI

YÜKSEK LİSANS TEZİ

Bilgisayar Mühendisliği Anabilim Dalı

Haziran – 2014
KONYA
Her Hakkı Saklıdır

ÖZET

YÜKSEK LİSANS TEZİ

YAPAY ARI KOLONİSİ OPTİMİZASYONU İLE KENAR BULMA

Elif Deniz YİĞİTBAŞI

Selçuk Üniversitesi Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Yrd.Doç.Dr. Nurdan BAYKAN

(2014, 85 Sayfa)

Jüri
Doç. Dr. Harun UĞUZ
Yrd. Doç. Dr. M. Taner ESKİL
Yrd. Doç. Dr. Nurdan BAYKAN

Kenar bulma metotları, görüntü işleme alanında önemli bir uygulama alanına sahiptir. Günümüzde birçok alanda görüntü işlemeden yararlanıldığı da bir gerçektir. Bu nedenle kullanılan yöntemler her geçen gün daha da gelişiyor ve bilgisayarlı görme sistemlerindeki hataların en aza indirilmesi için çalışmalar yapılıyor.

Görüntü işlemede daha iyi sonuçlar için önceki pek çok çalışmada optimizasyon metotlarından yararlanılmıştır. Bu çalışmada ise" Yapay Arı Kolonisi Optimizasyon" metodunun kenar bulma işlemlerinde kullanımı gerçekleştirilmiştir.

Çalışmada öncelikle Yapay Arı Kolonisi (YAK) optimizasyonu anlatılmıştır. Daha sonra optimizasyon ve optimizasyon metotlarından bahsedilmiştir. Ardından Yapay arı kolonisi optimizasyonu ve geliştirilen metot açıklanmıştır. Son olarak, geliştirilen metodun RADIUS/DARPA-IU Fort Hood veritabanından alınan farklı gri seviye nesne görüntüleri, siyah-beyaz görüntüler ve gri seviye hava fotoğrafları üzerinde uygulama sonuçları verilmiş ve sonuçların literatürde daha önce yapılan uygulamalarla karşılaştırılması yapılmıştır. Elde edilen sonuçlara göre, geliştirilen metodun, görüntü işlemede kenar bulma uygulamalarında kullanılabilirliği Hamming Uzaklığı, Hata Tespiti ve Benzerlik oranları verilerek ortaya konmuştur.

Anahtar kelimeler: Görüntü İşleme, Kenar Bulma Yöntemleri, Metasezgisel Yöntemler, Yapay Arı Kolonisi (YAK) Optimizasyonu

ABSTRACT

MS THESIS

EDGE DETECTION WITH ARTIFICIAL BEE COLONY OPTIMIZATION

Elif Deniz YİĞİTBAŞI

**THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCE
OF SELÇUK UNIVERSITY
THE DEGREE OF MASTER OF SCIENCE
IN COMPUTER ENGINEERING**

Advisor: Assist.Prof.Dr. Nurdan BAYKAN

(2014, 85 Pages)

Jury
Assoc. Prof. Dr. Harun UĞUZ
Assist. Prof. Dr. M. Taner ESKİL
Assist. Prof. Dr. Nurdan BAYKAN

Edge detection methods have an important application area in image processing. Today, in many areas of image processing are used for so different fields. Therefore methods which are used are improving day by day and works on decreasing of errors which are about computer vision systems are being done.

For better results on image processing, optimization methods were used in many studies. In this study, edge detection process is carried out using "Artificial Bee Colony Optimization".

In the first image processing and edge detection algorithms are mentioned. Subsequently optimization and kinds of optimization algorithm methods are described. Then Artificial Bee Colony (ABC) optimization and developed method are explained. Finally, as a result of developed method was implemented on different gray levels of object images, thresholded (black and white) images and gray level aeriel images which were retrieved from RADIUS/DARPA-IU Fort Hood database. Improved method's results are given and these results compared with the results of the methods in the previous literatüre. According to results obtained, Hamming Distance, Detection Error and Similarity rates have been revealed that the improved method can be use for edge detection in image processing applications.

Keywords: Artificial Bee Colony Optimization, Edge Detection Algorithms, Image Processing, Meta-Heuristic Algorithms

ÖNSÖZ

Çalışmamın başlangıcından bitimine kadar, bana her zaman destek olan aileme, benden yardımlarını hiçbir zaman esirgemeyen çok değerli danışman Hocam Yrd. Doç. Dr. Nurdan AKHAN BAYKAN'a, kendisinden aldığım dersler ışığında bana yol gösteren, desteğini ve bilgisini esirgemeyen Sayın Hocam Yrd. Doç. Dr. Ömer Kaan BAYKAN'a, Konya Selçuk Üniversitesi Bilgisayar Mühendisliği Bölümü öğretim elemanlarına, Işık Üniversitesi Bilgisayar Mühendisliği Bölümü öğretim elemanlarına da teşekkürü bir borç bilirim.

Elif Deniz YİĞİTBAŞI
KONYA-2014

ŞEKİLLER ve TABLOLAR

Şekil 3.1	Siyah – beyaz bir görüntüde kenarların gösterilmesi
Şekil 3.2	Görüntülerdeki kenar geçişlerinin grafiksel gösterimi
Şekil 3.3	Türev operatörleri ile kenar belirleme
Şekil 3.4	Sobel operatör maskeleri
Şekil 3.5	Canny yönteminde kullanılan Gaussian maske örneği (sigma değeri=1.4)
Şekil 3.6	Prewitt operatör maskeleri
Şekil 3.7	Roberts operatör maskeleri
Şekil 3.8	3x3'lük bir Laplace filtresi
Şekil 3.9	Yüksek geçiren filtreleme işlemi
Şekil 3.10	Alçak geçiren filtreleme işlemi
Şekil 3.11	Sezgisel Yöntemler
Şekil 3.12	Arı dansı ve gözcü arılara bilgi aktarımı
Şekil 3.13	Bal arılarının yiyecek arama döngüsü
Şekil 3.14	YAK Akış Diyagramı
Şekil 4.1	Bowyer'dan alınan nesne görüntülerine örnek (207.pgm)
Şekil 4.2	Bowyer'dan alınan hava görüntülerine örnek (airfield.pgm)
Şekil 4.3	Gonzales'ten alınan nesne görüntülerine örnek (Fig0914(a)(licoln from penny).tif)
Şekil 4.4	Arıların resim üzerinde rasgele konumlanmasının temsili gösterimi
Şekil 4.5	Kaynaklar Matrisinden örnek
Şekil 4.6	Komşular Matrisi
Şekil 4.7	Kaynağın komşu pikselinin olasılık değerinin kontrolü
Şekil 4.8	Gidilen kaynağın hem kendisinin hem de komşusunun olasılık değerinin belirlenen değerden düşük olması durumu
Şekil 4.9	Gidilen kaynağın olasılığı belirlenen değerden yüksek, fakat komşusunun olasılık değerinin belirlenen olasılık değerinden küçük olması durumu
Şekil 4.10	Gidilen kaynağın olasılığı belirlenen değerden yüksek olduğu halde komşusunun olasılık değerinin daha iyi olması durumunda yeni kaynağın komşu kaynak olarak belirlenmesi durumu
Şekil 4.11	Gidilen kaynağın kenar piksel olup olmamasının komşularının gri seviye değerlerine bakılarak karar verilmesi aşaması
Şekil 5.1	36.pgm için farklı maksimum çevrim sayılarında (MCS) YAK sonuç görüntüleri
Şekil 5.2	Gri seviye resimleri üzerinde YAK kenar bulma uygulamasının sonuçları (nesne resimleri)
Şekil 5.3	Fig0905(a)(wirebond-mask) için farklı MCS için YAK sonuç görüntüleri
Şekil 5.4	YAK kenar bulma uygulamasının sonuçları
Şekil 5.5	Buildings.pgm için farklı MCS için YAK sonuç görüntüleri
Şekil 5.6	YAK kenar bulma uygulamasının sonuçları (hava resimleri)
Şekil 5.7	RADIUS/DARPA-IU Fort Hood nesne resimlerinde Canny, Sobel, Roberts, YAK kenar tespit sonuçları
Şekil 5.8	Gonzales'in siyah-beyaz resimlerinde Canny, Sobel, YAK kenar tespit sonuçları
Şekil 5.9	RADIUS/DARPA-IU Fort Hood hava resimlerinde YAK kenar tespit sonuçları ve orijinal kenar bilgisi haritaları
Şekil 6.1	RADIUS/DARPA-IU Fort Hood nesne resimlerinde Canny, Sobel, Roberts, YAK kenar tespit sonuçları

Şekil 6.2	Çevresel elemanlar

Tablo 3.1	Kenar bulma için Duyarlılık (Sensitivity)- Özgüllük (Specificity) analiz değerleri
Tablo 5.1	RADIUS/DARPA-IU Fort Hood nesne görüntüleri için farklı MCS için YAK çalışma süreleri
Tablo 5.2	Gonzales'ten alınan test resimleri için farklı MCS için YAK çalışma süreleri
Tablo 5.3	RADIUS/DARPA-IU Fort Hood hava görüntüleri için farklı MCS için YAK çalışma süreleri
Tablo 5.4	RADIUS/DARPA-IU Fort Hood nesne görüntüleri için HU sonuçları (MCS=5000)
Tablo 5.5	Gonzales'e ait görüntüler için HU, HT ve Benzerlik sonuçları (MCS= 50000)
Tablo 5.6	RADIUS/DARPA-IU Fort Hood hava görüntüleri için HU, HT ve Benzerlik sonuçları (MCS= 50000)
Tablo 5.7	Hata Tespiti Tablosu
Tablo 5.8	Benzerlik Tablosu
Tablo 6.1	RADIUS/DARPA-IU Fort Hood nesne görüntüleri için farklı MCS için YAK çalışma süreleri
Tablo 6.2	Gonzales'ten alınan test resimleri için farklı MCS için YAK çalışma süreleri
Tablo 6.3	RADIUS/DARPA-IU Fort Hood hava görüntüleri için farklı MCS için YAK çalışma süreleri
Tablo 6.4	Hata Tespiti Tablosu
Tablo 6.5	Benzerlik Tablosu

İÇİNDEKİLER

TEZ BİLDİRİMİ ……...……. iii

ÖZET iv

ABSTRACT..... v

ÖNSÖZ vi

ŞEKİLLER ve TABLOLAR vii

İÇİNDEKİLER ix

SİMGELER VE KISALTMALAR x

1. GİRİŞ **1**

2. KAYNAK ARAŞTIRMASI **3**

3. MATERYAL VE METOT **15**

3.1.Resimde Kenar Bulma 15
3.2. Yapay Arı Kolonisi Optimizasyon Algoritması 24
3.3. Performans Analizi 35
3.3.1. Hamming Uzaklığı..... 35
3.3.2. Duyarlılık (Sensitivity)- Özgüllük (Specificity) Analizi, Doğruluk Oranı Ve Hata Tespiti, Benzerlik 36
3.3.3. Hata Tespiti ve Benzerlik Hesapları 37

4. YAK ile KENAR BULMA..... **39**

4.1. Kullanılan Görüntüler 39
4.2. Yapılan Uygulama 40
4.3. Geliştirilen YAK Algoritmasının Özellikleri 48

5. **ARAŞTIRMA SONUÇLARI**..... **49**

6. SONUÇLAR VE ÖNERİLER..... **70**

KAYNAKLAR **77**

ÖZGEÇMİŞ **85**

SİMGELER VE KISALTMALAR

Simgeler

D:	Optimizasyon Probleminin Boyutu
fit:	Kaynağa Ait Uygunluk Değeri
İ:	i.kaynak
J:	j.komşu
P:	Kaynağın Olasılık Değeri
SN:	Toplam Kaynak Sayısı
X:	Mevcut Kaynak
V:	Komşu Kaynak
φ:	(0-1) Arası Rasgele Bir Değer

Kısaltmalar

ARP	Araç Rotalama Problemi
B	Benzerlik (Similarity - S)
CT	Bilgisayarlı Tomografi
HT	Hata Tespiti (Detection Error- DE)
DPB	Doğru Parça Bulucusu
EPF	Edge Potential Function (Kenar Bilgisine Dayanan Fonksiyon)
GA	Genetik Algoritma
GFP	Gauss Filtrelenmiş Laplace
GN	Gerçek Negatif
GP	Gerçek Pozitif
HU	Hamming Uzaklığı (Hamming Distance- HD)
KKA	Karınca Kolonisi Algoritmsı
KKS	Karınca Koloni Sistemi
KSO	Kedi Sürüsü Optimizasyonu
KT	Kesişimlerin Tutarlılığı
MAED	Çoklu Alanda Ekonomik Dağıtım(Multi Area Economic Dispatch)
MCS	Maksimum Çevrim Sayısı
RGB	Red-Green-Blue (Kırmızı-Yeşil-Mavi) Değeri
SAR	Synthetic Aperture Radar (Yapay Açıklıklı Radar)
SQP	Sequential Quadratic Programing (Sıralı Karesel Programlama)
PSO	Parçacık Sürü Optimizasyonu

YAK	Yapay Arı Kolonisi
YBS	Yapay Bağışıklık Sistemleri
YN	Yanlış Negatif
YP	Yanlış Pozitif
YSA	Yapay Sinir Ağları

1. GİRİŞ

Görüntü işleme, bir tür sinyal işleme işlemidir. Sinyal girişi olarak bir görüntü alır ve işlem sonucundaki çıktı ise yine bir görüntüdür. Çıktı bilgisi ilgili görüntü hakkında karakteristikler, ayırt edici veya tanımlayıcı bilgiler içerir.

Günümüzde teknolojinin gelişmesi ile birlikte görüntü işleme tekniklerinden de oldukça fazla yararlanılmaktadır. Hayat ile iç içe bir yapıya bürünen teknolojik cihazların neredeyse tümü bir görüntüyü işleme üzerine yapılandırılmıştır. Bu alanda kullanılan algoritmalar ise teknolojinin hızı ile doğru orantılı bir şekilde çok hızlı bir şekilde gelişmektedir. Güvenlik sistemleri, trafik uyarı sistemleri, mikroskobik ve medikal görüntüleme sistemleri, uzaktan algılama sistemleri gibi pek çok alanda görüntü işleme tekniklerinden yararlanılarak çalışmalar yapılmaktadır.

Görüntü işlemedeki en önemli alanlardan biri kenar bulma işlemidir. Kenar bulma, en önemli görüntü bulma görevlerinden biridir. Kenarlar, görüntü içindeki en önemli bilgidir. İnsan görme sistemi, doğrudan kenarların algılanmasına dayanır (Marr ve Hildreth, 1980). Görüntülerdeki nesnelerin ayırt edilmesi, çıplak gözle fark edilemeyen ayrıntıların gösterilmesi, iki farklı desenin karşılaştırılması gibi durumlarda kenar tespiti algoritmaları oldukça yaygın bir şekilde kullanılmaktadır. Haralick (1984), pikseller üzerinde uyguladığı interpolasyon denklemi üzerinden bulduğu türev değerini kullanarak bulduğu gradyan değeriyle, ikinci türev üzerinde Gradyan yönündeki sıfıra geçişleri arayan bir yöntem sunmuştur. Canny (1986), Gaussian maskenin türevinden elde edilen işlecin uygulanmasına dayanan bir yöntem önermiştir. Görüntü işleme alanında kullanılan ilk yöntemler olan Roberts, Prewitt ve Sobel işleçleri de en uygun işleçlere örnektir (Bovik, 2010).

Mevcut teknolojik cihazlar, her zaman daha iyi bir performansı amaçladığından, geliştiriciler açısından optimizasyon yöntemleri de oldukça önemlidir. Optimizasyon matematiksel olarak bir fonksiyonun minimize ya da maksimize edilmesi olarak tanımlanabilir. Genel bir kavram olarak ise kısıtlı kaynakların en optimum biçimde kullanılmasıdır. Değişen teknoloji, artan rekabet, sınırlı kaynaklar ve sistemlerin değişimi problemlerinin tümü optimizasyon kavramını güncelleştirmektedir. Bu nedenle optimizasyon algoritmalarının kullanılmadığı bilim dalı hemen hemen yok gibidir.

2005 yılında, Derviş Karaboğa tarafından geliştirilen Yapay Arı Kolonisi (YAK) optimizasyon algoritması, son zamanlarda oldukça sık kullanılmaktadır. Kullanılan alanların çoğunda da başarılı sonuçlar göstermiştir. Doğadaki arıların yiyecek arama davranışlarından

esinlenilerek geliştirilmiş olan bu yöntem, optimum çözüme hızlı bir şekilde yaklaşması ile dikkat çekmektedir.

Bu çalışmada, görüntü işlemenin önemli alanlarından biri olan görüntüde kenar bulma işleminin, optimizasyon algoritmalarından Yapay Arı Kolonisi algoritması kullanılarak gerçekleştirilmesi amaçlanmıştır ve geliştirilen program tanıtılmaktadır. Kenar bulmada türeve dayalı olmayan ve herhangi bir maskeye ihtiyaç duymayan, optimizasyona dayalı bir yöntem geliştirilmiştir.

Bilinen kenar bulma yöntemlerinde önceden tanımlanan bir maske resim üzerinde gezdirilmektedir. Bu yöntemlerin uygulanması sırasında giriş resminin kenar bölgelerinde bulunan pikseller yok sayılmakta veya sıfır olarak kabul edilmektedir. Geliştirilen yöntemle bu bilgi kaybı önlenmiştir. Ayrıca önceden tanımlanan bir maske tanımlanmasına olan bağlılık da ortadan kaldırılmıştır.

Doğadaki arıların yiyecek arama davranışlarından esinlenilerek geliştirilen bu optimizasyon algoritması farklı resimler üzerinde uygulanmış ve çalışma süreleri tespit edilmiştir. Elde edilen görüntüler, bilinen diğer kenar bulma yöntemlerinin sonuç görüntüleri ile karşılaştırılmıştır. Bilinen yöntemlerde olan maske bağımlılığını ortadan kaldıran bu yöntem ile başarılı kenar bilgileri elde edilmiştir. Bu çalışmada öncelikle üzerinde çalışılan konu hakkında önceden yapılmış çalışmalar hakkında kısaca bilgi verilmiştir. Daha sonra yapay arı kolonisi optimizasyon algoritmasının adımları, resimlerde kenar bulma ve geliştirilen yöntem detaylı bir şekilde anlatılmıştır. Araştırma sonuçları bölümünde, yapılan uygulamanın verdiği sonuç görüntüleri verilmiş ve bilinen yöntemlerin sonuç görüntüleri ile sonuç bölümünde karşılaştırmalar yapılmıştır.

2. KAYNAK ARAŞTIRMASI

Görüntü işleme teknolojileri, günlük yaşam içinde oldukça fazla kullanılmaktadır. Bilgisayarlı görme sistemlerinin yaygın kullanımının sonucu olarak, görüntü işleme teknolojileri oldukça önemli hale gelmiştir. Görüntü işleme teknolojileri ile geliştirilen sistemler, günlük yaşamda neredeyse her alanda karşımıza çıkabilecek kadar yaygınlaşmıştır. Bu durum, pek çok araştırmacıyı bu alanda çalışmaya yöneltmiştir. Örnek olarak nesnelerin farklı özelliklerinin görüntü işleme yöntemleri ile çıkarılabilmesi mümkündür. Bu amaçla, ANFIS tabanlı görüntü işleme uygulaması ile portakal tadının karakterize edilmesi ile ilgili olarak yapılan çalışmada 300 farklı portakal örneği kamera görüntüleri ile belirlenmiş ve ilgili özellikleri çıkarılmıştır. On panelist tarafından yapılan bir duyusal testin görüntüleri, görüntü analizi için referans veri olarak kullanılmıştır. Özellikler ANFIS'e giriş bilgisi olarak gönderilmiş ve sınıflandırmada %90 üzerinde başarı elde edilmiştir (Adelkhani, 2013).

Robotik alanında da görüntü işleme tekniklerinden yararlanılmaktadır. Görüntü işleme desteği ile paralel robot kontrolünün yapıldığı tez çalışmasında, kameradan alınan görüntüler işlenerek; paralel robotun, pozisyonu değişen bir parçayı farklı konumlarda tutması ve istiflemesi amaçlanmıştır (Kulaksız, 2010).

Robotik alanındaki diğer bir çalışmada ise, serbest programlama robotik uygulamalarında yol planlama ve makine görmesi birleştirilmiştir. Yapılan çalışmada; otomasyon sürecinde, robotik alanında, çevre algılama ve karar verme için makine görme gelişiminin önemi vurgulanmıştır. Üç aşamadan oluşan çalışmada ilk aşama, temsil hatlarının belirlenmesi için web kamerası tarafından bir yüz görüntüsünün alınması işlemidir. İkincisi, bulunan hatların robotun takip etmesi için yollara dönüştürülmesidir. Sonuncu adım da, kontrol kodları ve otomatik kalibrasyonla, çizim için uyarlamalı olarak robotun kontrol edilmesidir. Prototip sistem farklı yüz görüntüleri için test edilmiştir (Mohammad, 2013).

Sanayi alanında da görüntü işleme teknikleri kullanılarak, çeşitli çalışmalar yapılmıştır. Horozoğlu'nun çalışmasında tornalama işleminde, görüntü işleme teknolojisinden yararlanılmıştır. Tornalama yapılan ürünlerin yüzeyinde oluşan pürüzlülük, ürünün çalışmasını etkileyen en önemli faktörlerden biridir. Görüntü alınarak yapılan yüzey pürüzlülüğü ölçümü, temassız olduğundan ölçme işlemini kolaylaştırmakta ve yüzeyde herhangi bir tahribata sebep olmamaktadır. Ayrıca yeterli şartlar sağlanabilirse, tornalama işlemi sırasında işlenen parça yerinden çıkartılmadan da görüntü alınabilirse yüzey pürüzlülüğü analizi yapmak mümkün olabilmektedir. Yapılan tez çalışmasında alınan

görüntüler, yüzey pürüzlülüğü açısından analiz edilerek, yüzey pürüzlülüğün ölçülmesinde sıklıkla kullanılan izleyici uçlu yöntemle karşılaştırılmıştır (Horozoğlu, 2013).

2006 yılında yapılan bir çalışmada ise, görüntü işleme teknikleri ile değişik büyüklükteki domino taşlarına ait görüntülerden, malzeme tane büyüklüğü analizi gerçekleştirilmiştir (Bal, 2006).

İnşaat sektöründeki çalışmada ise, dijital görüntü işleme yöntemleriyle lifli beton numunelerindeki çatlakların tespit edilmesi için, betonun kırılmadan önceki potansiyel davranışının görüntü işleme teknikleri kullanılarak incelenmesi amaçlanmıştır. Görüntüler Matlab programında kodlanan yazılımla işlenmiş ve ortaya çıkan deplasman miktarları aynı anda elde edilen komparatör değerleri ile karşılaştırılmıştır (Onat, 2008).

Görüntü işleme yöntemleri sayesinde insan görme sistemi ile algılanması zor ayrıntılar bulunabilmektedir.

Lazzaro, tarihi önem taşıyan arkeolojik nesnelerin incelenmesinde görüntü işlemeden yararlanmıştır. Düşük kontrastlı görüntülerin, yazılım teknikleri ile detaylandırılmasının yanıltıcı etkisinin araştırıldığı çalışmada, tarihin en çok çalışılan arkeolojik nesnelerinden Turin kefenindeki lekeler örnek olarak sunulmuştur. Hem eski hem yeni fotoğraflar üzerinde yapılan araştırmalarda, gerçekte var olmayan desenlerin algılanması, görüntü geliştirme ve manipülasyonu arasında dar bir sınır olduğunu göstermektedir. Bu bilgiler ışığında desen tanıma ve görüntü işlemeye dair detaylı bir çalışma yapılmıştır (Lazzaro, 2013).

Biyometrik tanıma sistemlerinde de görüntü işleme kullanılmaktadır. Yüz tanıma, parmak izi tanıma, iris tanıma, plaka tanıma gibi tanımaya dayalı güvenlik sistemleri pek çok alanda kullanılabilen görüntü işleme teknolojisinin ürünleridir. Tanıma sistemleri üzerine pek çok çalışma yapılmıştır.

İris tanıma için tüm bir resmin değil, sadece irise ait kesit alanlarının kullanıldığı tez çalışmasında, hem işlem süresi kısaltılmış, hem de tanıma işlemi esnasında test edilen piksel sayısı için yapılacak işlem miktarı azaltılmıştır. Böylece tanımadaki performans arttırılmıştır (Akhan, 2004).

Kişilerden alınan göz resimleri üzerinde iris desenine göre tanıma işlemi gerçekleştiren bir diğer tez çalışmasında ise yapay zekâ yöntemleri kullanılmıştır (Koçer, 2007).

Yapay zekâ ile teknikleri ile yapılan bir başka çalışmada ise, görüntü işleme teknikleri ile araç plaka tanıma sistemleri geliştirilmiştir (Çevik, 2010).

Destek sağlayan sınıflandırıcılar kullanılarak gerçek zamanlı 3 boyutlu yüz tanımanın gerçekleştirilmesinin amaçlandığı 2013 yılındaki çalışmada yüz bilgisinin, kameradan farklı uzaklıklarda çekilerek elde edilmiş olan dikey (ortogonal) izdüşüm görüntüleri, verimliliği ve algılama sürecinin etkinliğini geliştirmek için kullanılmıştır. Önerilen detektör sadece ön kamera açısından değil, aynı zamanda sahnenin döndürülerek elde edilen görünümlerde, yüksek pozlama ile algılama yapmaktadır. Farklı yöntemler ile karşılaştırma yapılmış ve 13000 test görüntüsünden %99 oranında başarı elde edilmiştir (Segundo, 2013).

Görüntü işleme, tanıma sistemlerinde yaygın olduğu kadar, uzay teknolojisinde de uydu görüntülerinden nesne tanıma ve çıkarımı işlemlerinde de kullanılmaktadır. .

Yüksek çözünürlüklü uydu görüntülerinden yol bilgisi çıkarımı süreci oldukça karmaşıktır. 2011 yılında yapılan çalışmada, yol bilgisi çıkarımı için, farklı spektral kontrast ve lokal doğrusal yörünge bilgileri olmak üzere, yolların iki belirgin özelliğinden yararlanılmıştır. Şehirlerarası bölgeler ve kentsel alanlardan dört farklı kategoride eğitim örnekleri çıkarılmıştır. Destek vektör makineleri ile gerçekleştirilen bu yöntemin üstünlüğünü doğrulamak için birkaç farklı metotla karşılaştırma yapılmıştır (Das, 2011).

Tıp alanında bilgisayarlı görme sistemleri ile sıkça karşılaşılmaktadır. Uzmanlar, hastalık teşhisi ve erken tanı gibi konularda, görüntü işleme teknolojisinden elde edilen bilgiler ışığında hastalık hakkında fikir sahibi olabilmektedirler. Özellikle röntgen, ultrason, MR, tomografi gibi pek çok görüntü üzerinde çalışılmaktadır. Bu amaçla 1995 yılındaki çalışmada, fetal kafa görüntüleri için görüntü işleme sistemi tasarım ve gerçekleştirimi çalışması yapılmıştır (Odçıkın, 1995).

Bilgisayarlı tomografi (CT) ölçümleri sırasında doğruluk kaybını etkileyen pek çok faktör vardır. Çünkü metroloji uygulamaları için CT işlemleri oldukça karmaşıktır. En kritik olanı ise görüntü segmentasyonu için yüzey çıkarma denilen kenar algılama işlemidir. Makine ve bilgisayar görme sistemlerinde yaygın olarak kullanılan farklı kenar tespit algoritmaları ve alternatif bir metot olarak analiz edilen CT metroloji uygulamasının sunulduğu çalışmada, karşılaştırmalar sonucunda sistemin avantaj ve dezavantajları üzerinde durulmuştur (Ontiveros, 2013).

Tıbbi görüntüleme alanında grafik işleme birimlerini (GPU) konu alan çalışmada, GPU algoritmalarının pratik kullanım için önemli katkılarının olduğu anlatılmaktadır. GPU ile medikal görüntü işleme alanında geçmiş ve mevcut çalışmalar sunulmuştur (Eklund, 2013).

"Click and Grow" algoritmasını temel alan tek tıklama grup segmentasyonu (SCES) yaklaşımının anlatıldığı çalışmada, 129CT akciğer tümörü görüntüsü, benzerlik indeksi (SI) kullanılarak değerlendirilmiştir. Daha sonra analizler SCES ile gerçekleştirilmiş ve sonuçlar karşılaştırılmıştır (Gu, 2013).

Tıp alanında görüntü işleme tekniklerinin kullanıldığı bir diğer çalışmada, klinik bulgular kullanılarak mesane geometrisinin idrar akım hızına etkisi incelenmiştir. Yapılan çalışmada, mesane şeklinin küreye benzerliği arttıkça mesanenin daha düşük akım ve basınç unsuruna sahip olduğu gözlenmiştir (Çınar, 2011).

Tıp alanında olduğu gibi diş hekimliği alanında da görüntü işleme teknikleri kullanılabilmektedir. Bu alandaki çalışmalara örnek olarak, diş hekimlerinin hastalık teşhis ve tedavi planlamasında değişik çekim teknikleriyle alınmış diş radyografları sıklıkla kullanılır. Eğer elde edilen diş görüntülerinde uygun çekim teknikleri kullanılmazsa röntgen sonuçları bulanık olur. Bu istenmeyen etkilerden kurtulmak, iyi bir seçiciliğin sağlanabilmesi için bilgisayar tabanlı görüntü işlemeye ihtiyaç vardır. Bilgisayar tabanlı görüntü işleme teknikleriyle diş radyograflarının incelenmesi ve YSA kullanılarak sınıflandırılması ile dişin ayırt edici özellikleri çıkartılmış ve sonra elde edilen bu özellikler yapay sinir ağına verilerek sınıflandırma başarıları çıkartılmıştır (Özbek, 2006).

Maddelerin tanınması ve kalitelerinin belirlenmesi problemlerinde de görüntü işlemeden yararlanılabilmektedir. Bu amaçla, mineral tanıma için de dijital görüntülerden yararlanılmıştır. Toplam 600 adet piksel değeri kullanılarak, geliştirilen bir bulanık mantık tabanlı önişlem algoritması ile RGB renk uzayında, mineral sınıflandırma çalışmaları yapılmıştır. Hem önişlem uygulanmış, hem de uygulanmamış verilerle yapılan sınıflandırmada %98'in üzerinde başarı elde edilmiştir (Baykan, 2010).

Yapay sinir ağları, son yıllarda oldukça popüler olan bir sınıflandırma tekniğidir. İnsan sinir sisteminin modellendiği bu sistemde yüksek oranda başarılar da elde edilmektedir. Görüntü işleme teknolojisi ve yapay sinir ağları kullanarak üç farklı gıda maddesinin bazı kalite özelliklerinin belirlenmeye çalışıldığı çalışmada; 3 faktörlü merkezi kompozit deney

tasarımı ile hamburger köftelerinde pişme sonrası yağ, su ve boyut kaybına ürün bileşiminde bulunan yağ (% 15-30), su (% 10-20) ve tekstüre soya proteini (% 3-9) oranlarının etkisi incelenmiştir (Velioğlu, 2010).

Kenar bulma, görüntü işlemenin en önemli bilgi çıkarımı yöntemlerinden birisidir. Çünkü kenarlar, görüntü içindeki en önemli bilgidir. İnsan görme sistemi, doğrudan kenarların algılanmasına dayanır (Marr ve Hildreth, 1980). Farklı yöntemlerle birlikte kenar bilgisinin elde edilmesi fikri pek çok araştırmacının çalışma konusu olmuştur. Bulanık topolojiye dayalı, kenar bulma algoritması çalışmasında; kenar bulma işleçlerinin daha geniş bir pencere içinde hesaplanmasını sağlayan, bulanık topoloji ve bağlantı derecesi haritalarını kullanan bir yöntem geliştirilmiştir (Hazer, 2007).

Karınca kolonisi optimizasyon algoritmasının yardımı ile yapılan dijital görüntülerde kenar özelliği bilgisinin çıkarımı çalışmasında, deney sonuçlarına göre karınca kolonisi sistemi etkili bir özellik ayırt edici yapı olarak nitelendirilmiştir. Önerilen metot, katmanlı bir makine görme sistemine uygulanmış ve sonuçları gösterilmiştir (Zhuang, 2004).

Karınca kolonisi optimizasyon algoritması ile yapılan bir diğer dijital resimlerde kenar bulma işleminde, bir graf üzerinde noktalar arasındaki ilişkilerin gösterilmesi amaçlanmış ve yapılan deneyler etkili sonuçlar vermiştir (Zhu,2007).

Yapay sinir ağları kullanan özniteliğe dayalı kenar bulma yöntemi çalışmasında ise, metodun ışık temelli kenar algılama (standart bir optik görüntü uygulaması), Canny yöntemi ile karşılaştırma ve sismik ölçümlerden kaya-tipi bulunması (optik olmayan örnek olarak yer bilimleri / ölçüm bilimleri uygulaması) alanlarına uygulanmasından elde edilen başarılı sonuçlar verilmiştir (Arpat, 2001).

Kenar bulma amacıyla kontrast farkına göre konum bulma konusunda 1998'de yapılan bir çalışmada, eldeki görüntü bilgilerinin yapısı üzerinde durulmuş, filtreleme tiplerinden ve kenar bulma algoritmalarından bahsedilmiştir. Görüntü bilgileri olarak kameradan çekilmiş olan gerçek, durağan görüntüler bilgisayar ortamına imge dosyası olarak aktarılmıştır. Medyan filtreleme ve komşuluk ortalaması filtrelemesi üzerinde durulup, bu filtreler imgeler üzerinde uygulanmış ve etkileri gözlenmiştir (Kefeli, 1998).

Görüntüdeki nesnelerin kesişim noktalarının tutarlılığı kullanılarak kenar bulma iyileştirme çalışmasında, daha önce kesişimlerin tespitinin iyileştirilmesinde kullanılan

görüntü yönelim tabanlı yerel tutarlılık ölçümünün (Kesişimlerin Tutarlılığı- KT) kenar bulma operatörleri ile hibrit kullanımları incelenmiştir. Bu amaçla sırasıyla Canny, Roberts, Prewitt, Sobel, Gauss Filtrelenmiş Laplace (GFP), Esas Boyutluluk, Doğru Parça Bulucusu (DPB) olmak üzere yedi farklı kenar tespit edicinin kalitesinin arttırılıp arttırılmayacağı test edilmiştir. Araştırmalar sonucunda, KT'nin Canny, GFP ve DPB gibi bazı kenar bulucularında kenar bulma kalitesini iyileştirebileceği sonucuna varılmıştır (Çiftçi, 2011).

Hücresel sinir ağları (CNN) ve dağıtık genetik algoritma (DGA) ile kenar tespiti işleminin, kızıl ötesi görüntüler üzerinde gerçekleştirildiği bir çalışmada DGA, CNN ağının eğitilmesi için kullanılmıştır. Deney sonuçlarına göre, DGA-CNN algoritmalarının kenar bulmada kullanılmasıyla kenar çıkarımı doğru olarak sonuçlanmıştır. Dahası, kızıl ötesi görüntülerdeki gürültü de oldukça azaltılmıştır. CNN tabanlı diğer kenar bulma algoritmaları ile karşılaştırmalar yapılmış ve yakınsama hızı oldukça geliştirilmiştir (Wang, 2013).

Renkli görüntülerde özyinelemeli bir kenar algılama yönteminin anlatıldığı Zareizadeh'in çalışmasında, Green fonksiyonu yaklaşımı kullanılmıştır. Önerilen yöntemde; kırmızı, yeşil, mavi bileşenler arasındaki ilişki ile renkli görüntülerde kenar saptanması için bir diferansiyel operatör tasarlanmıştır. Bu operatörün kullanımıyla, kenar algılama sürecinde renkli görüntünün tüm bileşenlerinin kısmi türevleri aynı anda etkilenmektedir. Böylece görüntü işleme aşamaları sırasında renkli görüntülerin vektör dalgası korunmaktadır. Amaçlanan bu metot diğer renkli kenar bulma operatörleri ile karşılaştırılmıştır (Zareizadeh, 2013).

İki boyutlu görüntülerde kenar bulma yöntemlerinden yararlanılabildiği gibi boyut dönüştürme işlemlerinde de kullanılabilmektedir.

İki boyutlu videoyu üç boyuta dönüştürme çalışmasında, kamera parametreleri bilinmeyen 2 boyutlu bir videoya 2D/3D dönüştürme algoritması uygulanarak 3 boyutlu görüntü efektinin oluşturulması hedeflenmiştir. İlk yöntemde derinlik haritası; hareket kestirimi yöntemi sonucu elde edilen hareket vektörleri ile oluşturulmuştur. Ve ikinci yöntemde ise derinlik haritası; kenar bilgisi kullanılarak oluşturulmuştur. Derinlik bilgisi sayesinde, iki metot da aynı kaydırma algoritmasını kullanarak yapay stereoskopik çift (sol ve sağ resim) oluşturmuştur. Bilinen yöntemlerle çekilmiş 2 boyutlu videoları kamera parametre bilgileri olmadan 3 boyut efekti verebilen otomatik 2D/3D dönüştürücü algoritmaları böylelikle geliştirilebilmiştir. (Aydoğmuş, 2011) .

Görüntü işleme, farklı yöntemlerle birlikte kullanılarak; görüntü iyileştirme, kenar bulma, gürültü içeren görüntülerde düzeltme ve onarma amaçlı çalışmalar için de kullanılabilmektedir. Toplamsal beyaz Gauss gürültüsü ile bozulmuş görüntülerin yapay sinir ağlarının yardımı ile onarımını amaçlayan çalışmada, görüntü verisi olarak, geliştirilen algoritmaların test edilmesi için siyah-beyaz görüntüler kullanılmıştır. Uygulamalarda, geliştirilmiş algoritmaların klasik algoritmalara göre gürültü gidermede daha başarılı olduğu gösterilmiştir (Tsoy, 2011).

Optimizasyon, kısıtlı kaynaklarla en iyi sonuca ulaşma çabasıdır. Bu amaçla farklı yöntemler geliştirilmiş ve faklı problemler üzerinde araştırmalar yapılmıştır.

Uzaktan algılanan iki görüntü arasındaki değişimin algılanması ile bina çıkarımı fonksiyonunun gerçekleştirildiği 2012 yılında yapılan çalışmada, olasılıksal yöntemlerden yararlanılmış ve bir global optimizasyon işlemi ile binaların optimal konfigürasyonunun ve bina parçaları arasında gözlenen verilerin etkileşimleri dikkate alınarak işlem yapılmıştır. Global optimizasyon tekniği olarak ise çoklu doğum ve ölüm optimizasyon tekniği kullanılmıştır (Benedek, 2012).

Sezgisel optimizasyon algoritmaları, sürü zekasına dayalı olan algoritmalardır. Büyük boyutlu optimizasyon problemlerinde kullanılmaktadır ve optimuma yakın çözüm sunabilmektedirler. Sürülerin ortak bir amaçla, birbirleri ile etkileşim içinde en iyi çözüme sezgisel olarak yaklaşma davranışını modelleyen bu yöntemlerle pek çok araştırma yapılmıştır.

Belirli sınırlamaları sağlayacak şekilde, bilinmeyen parametre değerlerinin bulunmasını içeren herhangi bir problem, optimizasyon problemi olarak adlandırılabilmektedir (Karaboğa, 2011). Sezgisel algoritmalar ise, çözüm uzayında optimum çözüme yakınsaması ispat edilemeyen algoritmalardır (Karaboğa, 2011).

Metasezgisel algoritmalar modelleme ve optimizasyon için güçlü araçlardır. Özellikle son yıllarda doğadaki canlıların yiyecek arama yöntemlerini modelleyen birçok algoritma ortaya konulmuştur(Gandomi, 2013). Bu algoritmalar, birçok alanda olduğu gibi görüntü işleme çalışmalarında da kullanılmaktadır.

Görüntü işlemede segmentasyon işlemi, basit anlamda bölgelerin ayrıştırılmasıdır. Segmentasyon problemlerinde iki seviyeli segmentasyon, bir eşik değerli olan

segmentasyonlara göre daha iyi sonuçlar vermektedir. Bu yöntemlerin gelişmiş bir alternatifi olarak segmentasyonda çok seviyeli eşikleme (multi-level thresholding) için Parçacık Sürü Optimizasyon (PSO) algoritması ile birlikte Yapay Arı Kolonisi (YAK) algoritması incelenmiştir (Akay, 2013).

Genişletilmiş ayna curvelet (ME-Curvelet) dönüşümü ve geliştirilmiş PSO algoritmasını temel alan çalışmada, SAR (Sentetik açıklık radar) görüntülerindeki kontrast ve kenar bilgisindeki benek gürültüsünü azaltma işlemi gerçekleştirilmiştir. PSO algoritması en iyi benek giderici ve gelişmiş görüntüyü veren global arama stratejisi olarak kullanılmaktadır. Yakınsamada erken yakınsamayı önlemek amacıyla klasik PSO algoritması için iki farklı geliştirme yapılmıştır. Yani yeni bir öğrenme düzeni ve bir mutasyon operatörü tanıtılmıştır. Bu yöntem, deneysel sonuçlara göre, Wavelet dönüşümü ve Curvelet tabanlı yöntemleri geride bırakmıştır (Li, 2011).

Sezgisel algoritmalar, depolama sistemleri ve araç rotalama problemlerinde de kullanılarak daha doğru ve düzenli bir işleyiş sağlanması amaçlanmıştır. . Bu amaçla depolama sistemlerinde sipariş toplama işlemlerinin optimizasyonunda, sipariş toplayıcı seyahat mesafesinin azaltılması amacıyla, sipariş toplama probleminin çözümünde kullanılan tekniklerin irdelendiği çalışmada Clarke-Wright (CW) sezgiseli, Rastsal Tasarruf (RT) sezgiseli ve Genetik Algoritma (GA) tabanlı metasezgisel yöntem olmak üzere üç farklı sipariş toplama tekniği değerlendirilmiştir (Görenler, 2012).

Araç rotalama problemi (ARP), toplam kat edilen mesafeyi, dolayısıyla da toplam maliyeti optimize etmek amacıyla depoda başlayıp depoda biten ve her müşterinin bir defa ziyaret edildiği rotaların oluşturulması olarak adlandırılır. ARP, çözümü zor ve zaman alan optimizasyon problemlerinden biridir. Bu yüzden araştırmalarda ARP çözümleri için sezgisel ve metasezgisel yöntemler ağırlıklı olarak kullanılmaktadır.

Araç rotalama problemleri için popülasyon ve komşuluk tabanlı meta sezgisel bir algoritmanın tasarımı ve uygulaması çalışmasında, firmalarda özellikle lojistik planlarının oluşturulması sırasında epeyce sık karşılaşılan kapasite ve mesafe kısıtlı Araç Rotalama Problemi için, popülasyon ve lokal arama tabanlı bir meta sezgisel algoritma önerilmiştir (Erol, 2006).

Optimizasyon problemlerinin çözümünde melez metasezgisel bir algoritmanın tasarımı çalışmasında, çok depolu araç rotalama probleminin çözümü için Karınca Kolonisi

Optimizasyonu (KKO) ve GA'nın bir arada kullanılmasıyla oluşturulan melez metasezgisel bir yapı önerilmiştir (Yücenur, 2011). Metasezgisel yöntemlerden biri olan KKO ile çözüm önerisinin sunulduğu bir diğer çalışmada, yöntem literatürde bulunan test problemlerine uygulanmış ve elde edilen çözümler, problemlerin bilinen en iyi çözümleri ile karşılaştırılmıştır (Ekizler, 2011).

Kuruluş yeri seçimi, işletmeler için tedarik zinciri yönetimi açısından önemli bir hale gelmiştir. Hizmet veren tesisler ve talep noktaları arasındaki taşımalardan dolayı ortaya çıkan maliyetlerin minimize edilmesi, önemli bir problemdir. Bir şantiyede bulunan tesislerin en uygun yerlere yerleştirilmesi için yapılan çalışma kapsamında metasezgisel bir yöntem olan KKO tabanlı bir algoritma geliştirilmiştir. Algoritmanın etkinliği, literatürde ulaşılan veri setleri üzerinde test edilerek doğrulanmış ve sonuç olarak taşıma maliyetleri açısından daha ekonomik şantiye yerleşimi sağlanmıştır. Geliştirilen algoritma, devam eden Beydağ Barajı inşaatı projesine uygulanmıştır (Çalış, 2009).

Tesis ve taşıma noktaları arasındaki taşıma maliyetlerini minimize etmek amacıyla yapılan kuruluş yeri seçimi problemlerinde, meta sezgisel olanPSO, diferansiyel gelişim algoritması ve YAK algoritması kullanılmıştır. Çalışmanın sonunda sözü edilen üç ayrı meta sezgisel algoritma kullanılarak geliştirilen modeller, problemin çözümü için uygulanmış ve elde edilen sonuçlar karşılaştırılmıştır (Bastı, 2012).

Metasezgisel algoritmalar içinde, son zamanlarda, özellikle YAK algoritması daha popüler hale gelmiştir. Bunun sebebi olarak; yöntemin birçok alana uygulanabilmesi, sürekli bir algoritma olmasına karşın kesikli problemlere de uygulanabilmesi ve diğer birçok yöntemden daha iyi sonuçlar üretebilmesi verilebilir Genel olarak YAK optimizasyon algoritması, sürü zekâsına dayalı bir algoritmadır, esnek ve basittir, oldukça az kontrol parametresine sahiptir, gerçek yiyecek arayıcı arıların davranışlarını simüle eder, nümerik problemler için geliştirilmiştir ama ayrık problemler içinde kullanılabilir, Kâşif arılar tarafından gerçekleştirilen küresel; görevli ve gözcü arılar tarafından gerçekleştirilen bölgesel araştırma kabiliyetine sahiptir. Bu lokal ve global araştırma süreçleri paralel yürütülmektedir (Karaboğa, 2011).

YAK optimizasyon algoritması farklı türde problemlere uygulanmıştır. Örneğin, ilaç dozlarının ayarlanmasında YAK optimizasyonundan yararlanılabilmektedir. Bir hekimin doz planlamasına karar vermesinde etkili olan klinik parametrelerinin (hasta yaşı, tümör boyutu

vs…) tanımlaması yapılmış, I-131 (radyoaktif iyot) iyot dozu için gerçeğe yakın sonuçlar elde edilmiştir (Teodorovic´ D. ve ark., 2013).

Dağıtım sistemleri, gezgin satıcı problemleri, çizelgeleme, veri sınıflandırma, depolama sistemleri ve bulmacalar gibi pek çok alanda YAK optimizasyonu kullanılabilmektedir.

Isı ve gücün ekonomik dağıtımında arı kolonisi optimizasyonunun kullanıldığı çalışmada, YAK sonuçları hibrit-PSO, SQP, hibrit evrimsel programlama sonuçları ile karşılaştırılmış ve minimum üretim maliyeti ve en az işlem süresini SQP ile birlikte PSO sağlamıştır (Basu M., 2011).

Güç sistemlerinin planlanmasında, enerjinin üretiminde, iletiminde, dağıtımında enerji verimliliğini sağlamak ve maliyetleri en aza indirmek amacıyla çeşitli optimizasyon metotlarının irdelendiği çalışmada, sezgisel Optimizasyon Metotlarından olan GA, YAK, PSO ve Diferansiyel Gelişim Algoritması (DGA), IEEE 30 baralı test sistemine ayrı ayrı uygulanmış ve sonuçlar birbiriyle karşılaştırılmıştır (Doğan, 2011).

MAED (çoklu alanda ekonomik dağıtım-multi area economic dispatch) için YAK kullanılan çalışmada 3 farklı test sistemi ile algoritmanın etkisi doğrulanmış ve Diferansiyel Evrim (DE), Evrimsel Programlama (EP), gerçek kodlu GA ile karşılaştırıldığında, MAED sistemlerinde YAK algoritmasının alternatif olarak kullanılabileceği görülmüştür (Basu, 2013).

YAK algoritması kullanarak gezgin satıcı probleminin Türkiye'deki il ve ilçe merkezlerine uygulanması çalışmasından alınan sonuçlara göre, gezgin satıcı probleminin çözümünde YAK, PSO ve GA'ya oranla daha yüksek performans göstermiştir (Akça, 2011).

YAK algoritması ile ders çizelgeleme probleminin çözümü sonucunda kullanılan algoritma ile uygun bir ders çizelgesi oluşturulmuş ve bu çizelgelerin raporları alınmıştır (Tokmak, 2011).

Permütasyon akış tipi çizelgeleme problemleri için de YAK algoritması uygulanmıştır. Geliştirilen algoritmanın performansı literatürde sıklıkla kullanılan Carlier, Reeves ve Taillard test problemlerine uygulanmış ve sonuçları farklı meta sezgisel yöntemlerle karşılaştırılmıştır. Geliştirilen algoritma Carlier ve Reeves problem gruplarında literatürde yer alan pek çok çalışmadan daha başarılı sonuçlar üretmiştir (Tosun, 2012).

Veri sınıflandırma problemlerini çözmek için GS-LSSVM adlı karma bir çekirdek oluşturulmuştur. Destek vektör makinesine (DVM), bir ızgara düzeni (GS) entegre edilerek bir makine öğrenmesi kavramı önerilmiştir. GS-LSSVM özellik çıkarımı için kullanılmıştır. Örnekleme ve değerlendirme amaçlı California Irvine üniversitesi veri tabanından alınan on veri seti kullanılmıştır. Sonuçlara göre, daha küçük ölçekte özellikler kullanılarak bir deneyin daha kolay sınıflandırılmasını sağlayan bir model oluşturulduğu görülmektedir. YAK algoritması ise LSSVM'nin performansını arttırmak için kullanılmıştır (Hsieh,2011).

DE, PSO ve Evrimsel Algoritmanın (EA), çok boyutlu sayısal problemlerin çözümü alanında YAK ile karşılaştırıldığı çalışmada, simülasyon sonuçları, YAK algoritmasının, yüksek boyutlu olan mühendislik sorunlarını çözmede kullanılabileceğini göstermektedir. YAK, farklı popülasyon boyutları için analiz edilmiş ve en iyi sonucu 1000 çevrim sayısında vermiştir. Koloni büyüklüğü ise sırasıyla 10, 50, 100 iken denemeler gerçekleştirilmiştir. Belirlenen kriterler sonucunda 5 farklı fonksiyonda deney yapılmış ve en iyi sonuçlar maksimum çevrim sayısı 1000 iken elde edilmiştir (Karaboga,2008).

DE-ABC adlı yeni bir algoritmanın önerildiği çalışmada, DE operatörlerinin, YAK algoritmasına dâhil edilmesi gerçekleştirilmiştir. Geliştirilen bu metot, on adet Benchmark problemine uygulanmıştır. Sonuçlar, temel DE ve temel YAK ile elde edilen sonuçlarla karşılaştırılmıştır. Sayısal sonuçlara göre, DE operatörlerinin varlığı YAK algoritmasının performansını önemli bir ölçüde geliştirmiştir (Sharma, 2011).

2011 yılında yapılan çalışmada, bir kitap perakendecisinin raf alanı tahsis kararı için yeni bir doğrusal olmayan karışık tam sayılı programlama modeli geliştirilmiştir. Geliştirilen modelin yapısından dolayı, PSO ve YAK sezgisel yaklaşımlar önerilmiştir (Özcan, 2011).

Bir ızgara (grid) üzerindeki zamanlama deneyleri için, bir multi-objective sürü optimizasyon algoritması sunulmaktadır. MOABC olarak tanıtılan algoritma multi-objective yapay arı kolonisi algoritmasıdır. Bu algoritma zaman ve maliyet açısından minimumlaştırmaya bağımlı işler hakkındaki bir deneyde zamansal optimizasyon problemine uygulanmıştır (Arsuaga, 2011).

Sudoku bulmacaları NP-complete problemler olarak bilinen zor problemler seti grubuna aittir. Bir sudoku bulmacası nispeten basit kuralları olan, mantık esaslı, kombinasyonel bir bulmacadır. Sudoku bulmacasının çözümünde YAK algoritmasının bir

türevi kullanılmıştır. Sonuçlar, çözümde bu algoritmanın etkili ve verimli olduğunu göstermektedir (Pacurib, 2009).

Görüntü işleme alanındaki problemlere ve karmaşık mühendislik problemlerine de YAK optimizasyonu ile çözüm üretilebilmektedir.

SAR (Synthetic Aperture Radar-Yapay açıklıklı radar) görüntülerinde bölünme (segmentasyon) işlemi için YAK algoritmasında yararlanılmıştır. Çalışma sonuçları geleneksel GA'ya göre daha iyi sonuç vermiştir (Ma M., 2011).

YAK, alçak uçuş uçakları için yapılan bir çalışmada EPF (edge potential function) yaklaşımının optimize edilmesi uygulamasında da kullanılmıştır (Xu ve ark., 2010).

Bilgisayar yazılımı ve donanımı alanlarında ise YAK ile iyileştirme çalışmaları yapılmıştır. YAK algoritması, Alanda Programlanabilir Kapı Dizileri (APKD) (FPGA-Field Programmable Gate Array) üzerinde donanımsal olarak gerçeklenmiş ve YAK'ın APKD üzerinde etkin bir şekilde gerçeklenmesini sağlayan donanım mimarisi önerilmiştir. Önerilen mimari sayesinde algoritma APKD üzerinde daha az yer kaplamakta ve PC tabanlı yazılım versiyonlarına göre daha hızlı çalışmaktadır (Avcı, 2011).

Yapar sinir ağlarının eğitimi konusunda da yapay arı kolonisi algoritması kullanılmıştır (Öztürk, 2011).

3.MATERYAL ve METOT

3.1. Resimde Kenar Bulma

Her nesnenin sahip olduğu belli özellikler vardır. Bu özelliklerden en önemlisi de nesnelerin şekil bilgisidir. Çünkü canlılar göz ya da dokunma duyuları ile o cismin ne olduğunu sadece şekillerini algılayarak anlayabilirler. Nesnelerin algılanması, günlük yaşam için önemli olduğu kadar bilimsel hayatta da oldukça büyük önem taşımaktadır. İnsan görme sistemi, araştırmalarda oldukça fazla yer almakta ve bu sisteme benzer bilgisayarlı görme sistemleri geliştirilmeye çalışılmaktadır.

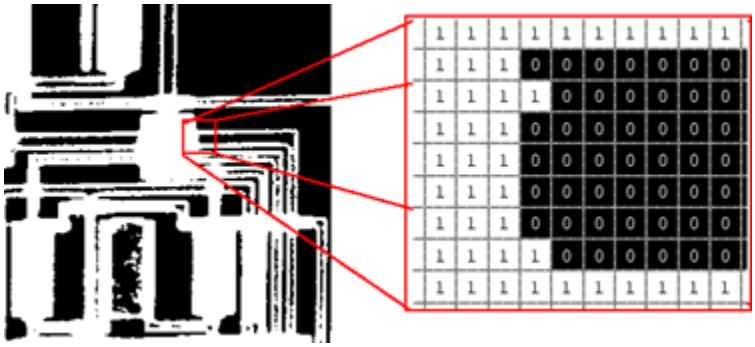
Görüntü işleme, gerçek görme sisteminin araştırılması ve günümüz teknolojisinden yararlanılarak uygulanması alanlarında gelişmekte olan bir konudur. Görme ve tanıma işlemleri ne kadar önemli ise bilim dünyası için de görüntü işleme o kadar önemli bir yer tutmaktadır.

Görüntü işleme teknolojileri; askeri endüstri, tıp, güvenlik, kriminal laboratuvarlar, uydu görüntüleri ile hava gözlem ve tahmin uygulamaları, uzaktan algılama uygulamaları, gıda sektörü (etin kalite tayini gibi), petrol arama, robotik, radar, astronomi, biyomedikal, fizik ve daha birçok alanda kullanılmaktadır.

Kenar bulma, görüntü işlemenin en önemli görevlerinden biridir. Çünkü kenarlar, görüntü içindeki en önemli bilgidir. İnsan görme sistemi, doğrudan kenarların algılanmasına dayanır (Marr ve Hildreth, 1980).

Nesnelerin şekillerine ilişkin bilgi çıkarımını, görüntü işlemenin çok önemli bir dalı olan kenar bulma metotları gerçekleştirmektedir. Gerçek görme sisteminde de kenar bilgisine bağlı olarak cisimlerin ne olduğu kararına varılabilmektedir. Bu nedenle de kenar bulma yöntemleri, bilgisayarlı görme sistemlerinde oldukça büyük önem taşımaktadır.

Kenar bilgisi, görüntüdeki ani renk ya da parlaklık değişimlerinin olduğu noktalardan çıkarılabilmektedir. Görüntülerin kenar bilgisi sınır eğrisi olarak da tanımlanabilir ve bu eğriler lokal süreklilik gösteren değişim noktalarından oluşmaktadır (Şekil 3.1) (Arı, 2008).



Şekil 3.1. Siyah – beyaz bir görüntüde kenarların gösterilmesi

Gürültünün genel anlamı istenmeyen etkidir (Tolluoğlu ve ark., 2005). Görüntüler üzerinde ise karşımıza piksellerdeki istenmeyen değişiklikler olarak çıkmaktadır.

Kenar bulma işlemleri ve sonuçları; resmin türüne, resimdeki gürültüye ve resim boyutuna göre farklı sürelerde sonuç verebilmektedir.

Görüntü içerisinde gürültünün varlığı; kenar belirleme ve konum bilgi çıkarımı ve kenarların çok ölçekli yapısı açısından karşılaşılabilecek önemli sorunlardan biridir.

Yaygın olarak kullanılan pek çok kenar bulma algoritması maskeleme yöntemini esas almaktadır. Her yöntemin kendine has kullandığı bir maske bulunmaktadır. Bu maskeler daha önceden tanımlanmıştır ve algoritmalar maske bilgisini kullanarak kenar bulma işlemi gerçekleştirirler. Bu kenar bulma algoritmalarına aşağıdaki metotlar örnek olarak verilebilir:

1) Log Filtresi

2) Canny

3) Roberts

4) Prewitt

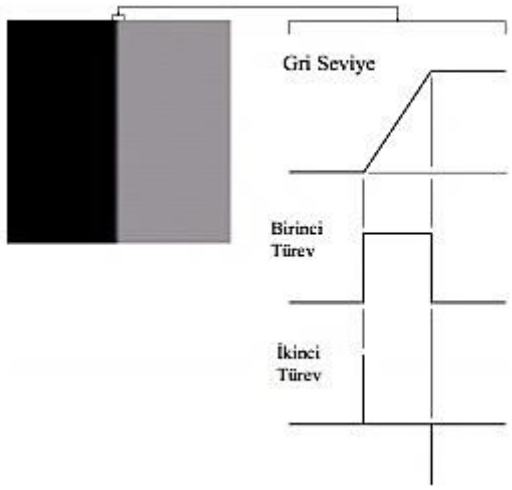
5) Sobel

Kenar bulmak için kullanılabilecek en temel yöntem, türevin kullanılmasıdır. Bu metotlar iki gruba ayrılır ve bu metotlar da türev almaya dayanan yöntemlerdir (Şekil 3.2, Şekil 3.3) (Gonzalez ve Woods, 2007):

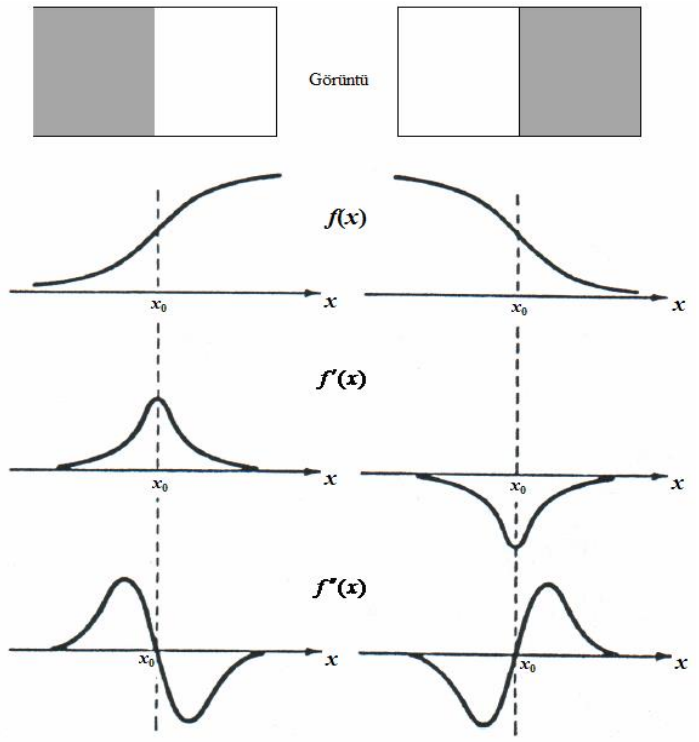
1) Gradyan (Eğim) Temelli Yöntemler: Görüntünün birinci türevindeki maksimum ve minimum değerleri inceler ve böylece kenarları tespit eder. (Gradyan değerini eşik değeri ile

karşılaştırabiliriz ve eşik değeri aşıldığında kenarı tespit edebiliriz.). Sobel, Roberts, Prewitt, Canny bu tür yöntemlere örnektir.

2) Laplasyan Temelli Yöntemler: Kenarları bulmak için ikinci türevdeki sıfır geçişlerini arar. Birinci türev görüntüde genellikle kalın kenarlar üretir, ikinci türev ise görüntüdeki ince kenar, gürültü noktaları gibi ince detaylara daha iyi cevap verir. Log Filtresi de bu yöntemlere örnektir.



Şekil 3.2. Görüntülerdeki kenar geçişlerinin grafiksel gösterimi



Şekil 3.3. Türev operatörleri ile kenar belirleme: **(a)** Koyu arka plan üzerindeki beyaz bant görüntüsü; **(b)** Açık arka plan üzerindeki siyah bant görüntüsü. Bu görüntülere

ilişkin fonksiyon grafikleri ve bu fonksiyonların 1. ve 2.türevleri

X_0 Şekil 3.3'e göre muhtemel bir kenar noktasıdır. f'(x)'in sonucu hem kenarın yönünü hem de büyüklüğünü belirlemektedir. Eğer bu değerin mutlak değeri çok büyük ise f(x) çok hızlı değişir ve buradan parlaklıkta hızlı bir değişim olduğu anlaşılır.

| f'(x) | önceden belirlenmiş eşik değerinden büyükse bu görüntü pikseli bir kenar adayıdır sonucuna ulaşabiliriz. Kenar görüntüsünün performansı ise seçilen eşik değer ile çok yakından ilgilidir. Eşik değer, görüntüdeki piksel değerlerine göre üretilen sayısal bir değerdir. Görüntülerin siyah beyaza dönüştürülmesi gibi işlemler, görüntüdeki her bir piksel değerinin eşik değerden büyük veya küçük olmasına bağlı olarak gerçekleştirilir. Bu değer, çoğu zaman görüntülerin histogramından elde edilebilmektedir. f'(x) pozitif bir değer ise f(x) de artan bir fonksiyondur. Bir f(x,y) fonksiyonu için f'(x,y), f(x,y)'nin gradienti olarak tanımlanır ve denklem 3.1' e göre hesaplanır:

$$f'(x,y)=\left(\frac{df(x,y)}{dx}\right)\vec{i}_x+\left(\frac{df(x,y)}{dy}\right)\vec{i}_y \qquad \textbf{(3.1)}$$

Yukarıdaki denklemde $\vec{i}_x$ x yönündeki, $\vec{i}_y$ y yönündeki birim vektörleri temsil eder. Sadece düşey ve yatay durumlardaki lokal maksimumu kontrol etmek kenar bilgisi için yeterli olacaktır.

Maskeler sadece ilgili doğrultudaki kenarları tanımakta ancak diğer doğrultulardaki kenar bilgisini tespit edememektedir. Genellikle bu yöntemlerde, gradient görüntüsünden kenar görüntüsüne ulaşabilmek için, en büyük gradient değerinin belli bir yüzdesi alınarak eşik değerleri belirlenmektedir.

Görüntülerde türev alma işlemi gürültüden çok fazla etkilenebilmektedir. Bu yüzden gürültülü resimlerde önce gürültü azaltma yani yumuşatma işlemi yapılmalı ardından kenar belirleme işlemine geçilmelidir.

Sobel, Canny, Roberts ve Prewitt maskeleme ile kenar bilgisi çıkarımı yapabilen metotlara örnektir. Bu metotlarda belli kare matris yapısına sahip maskeler, kenar bilgisi çıkarılacak görüntü üzerinde x ve y koordinatları boyunca gezdirilir.

Prewitt ve Sobel en iyi bilinen kenar bulma operatörleridir ve 3x3'lük düşey ve yatay yönlü maskeye sahiptirler. Bu operatörlerin en büyük sorunu yüksek gürültülü ortamlarda kenarları doğru olarak bulamamalarıdır. Bu maskelerin boyutları da önemlidir. Konvolüsyon maskelerinin boyutu ne kadar büyürse, alçak geçiren filtre etkisi de o kadar güçlü olur ve operatörlerin neden olduğu frekans etkisine karşı hassasiyeti azalır (Sharcanski, 1997).

Sobel kenar bulma metodu ile 3x3'lük komşulukta merkez noktanın Gradyan değeri, dik vektör çiftlerinin toplamları olarak bulunmaktadır (Şekil 3.4). Bu işlemin sonucunda ise düşey ve yatay yöndeki ağırlık fonksiyonlarından yararlanılarak Şekil 3.5'teki matrisler elde edilir ve bu matrisler sayesinde de kenar bulma işlemi gerçekleştirilir (Chao, 2010).

Sobel kenar bulma algoritmasında, önceden belirlenmiş bir eşik değer vardır ve hesaplanan gradyan değerinin karesinin bu eşik değerden büyük olduğu noktalar kenar noktası olarak işaretlenmektedir (Aybar, 2008). Gradyan değeri denklem 3.2'deki gibi hesaplanmaktadır:

$$\|Vf\|=\sqrt{\left(\frac{df}{dx}\right)^2+\left(\frac{df}{dy}\right)^2} \rightarrow |G|=|G_x|+|G_y| \qquad \textbf{(3.2)}$$

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

(a) Sobel Metodunda Düşey Yönde Uygulanan Matris

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

(b) Sobel Metodunda Yatay Yönde Uygulanan Matris

Şekil 3.4. Sobel operatör maskeleri

Canny kenar bulma algoritması, kenar bulmada oldukça etkin kullanılmakta olan algoritmalardan biridir. Önce görüntüdeki gürültü, bir sigma değerine göre üretilen Gaussian çekirdekle görüntüye konvolüsyon uygulanması ile azaltılır. Yani görüntünün türevi alınmadan önce yumuşatma filtresi uygulanır. Daha sonra bir Gradyan operatörü kullanılarak, kenar Gradyan büyüklüğü ve yönü hesaplanır.

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

1/115

Şekil 3.5. Canny yönteminde kullanılan Gaussian maske örneği (sigma değeri=1.4)

Prewitt kenar bulma metodunda görüntüye uygulanan düşey ve yatay matrisler Şekil 3.6'daki gibidir (Chao, 2010).

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

(a) Prewitt Metodunda Düşey Yönde Uygulanan Matris

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

(b) Prewitt Metodunda Yatay Yönde Uygulanan Matris

Şekil 3.6. Prewitt operatör maskeleri

Roberts kenar bulma metodu, 2x2'lik bir matris kullanarak kenar bulma işlemini gerçekleştirir. Ancak 3x3'lük bir matris olarak kullandığı maskelerin gösterimi Şekil 3.7'deki gibidir (Chao, 2010).

0	0	0
0	0	1
0	-1	0

(a) Roberts Metodunda Düşey Yönde Uygulanan Matris

0	0	0
0	1	0
0	0	-1

(b) Roberts Metodunda Yatay Yönde Uygulanan Matris

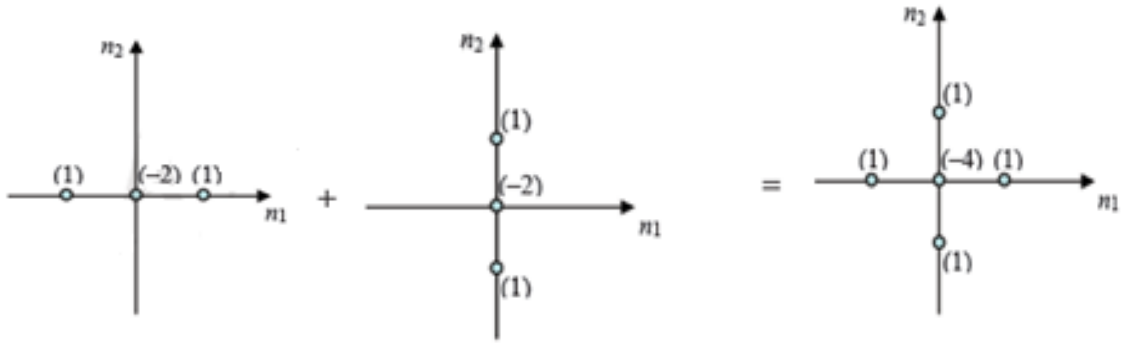
Şekil 3.7. Roberts operatör maskeleri

LoG filtresi başlıca kenar bulma yöntemlerinden biridir. Laplace ve Gaussian filtrelerinin birbiri ardına uygulanmasına dayanır. İlk olarak 1980 yılında Marr ve Hildreth tarafından önerilmiştir. İlk olarak Marr-Hildreth tarafından önerildiğinden bu yönteme aynı

zamanda “Marr-Hildreth kenar belirleme yöntemi” adı da verilmektedir. Laplasyan-Gauss yöntemindeki temel mantık, iki boyutlu bir Gaussian fonksiyonu ile görüntünün konvolüsyonuna dayanır (Marr ve Hildreth, 1980).

Görüntü işlemede, Laplasyan filtresi genellikle ön işlemlerde kullanılmaktadır. Resimdeki gürültüyü azaltmak, daha işlenebilir hale getirme gibi amaçlarla kullanılır. Birinci türevin maksimum veya minimum olduğu noktada, ikinci türev sıfıra eşittir.

Bir görüntü fonksiyonunda, ikinci türev alınıp sıfır geçiş noktalarının tespit edilmesi ile görüntüye ait kenar görüntüsüne ulaşılabilmektedir (Şekil 3.8).



(a) Satır boyunca 2.türev **(b)**Sütun boyunca 2.türev **(c)**Laplasyan maskesi

0	1	0
1	-4	1
0	1	0

Şekil 3.8. 3x3’lük bir Laplace filtresi

Sonuç olarak, Laplasyan kenar tanıma yönteminde, Laplasyan kenar maskeleri ile konvolüsyona tabi tuttuktan sonra elde edilen Laplasyan görüntüsünün mutlak değerinin bir eşik değeri ile karşılaştırılması sonucunda kenar görüntüsüne ulaşılır. (Jane, 2010)

$$f''(x,y) = \left(\frac{df}{dx}\right)^2 + \left(\frac{df}{dy}\right)^2 \qquad (3.3)$$

Gradient tabanlı yöntemler ile aynı işlemler yapılmasına rağmen Laplasyan tabanlı yöntemlerden elde edilen kenar görüntüleri arasında bazı farklılıklar vardır. Bu farklar aşağıdaki gibi sıralanabilir:

1) Laplasyan kenar tespiti yönteminde ikinci dereceden türevler olduğundan dolayı, Gradientine göre görüntünün Laplasyanı gürültüye karşı fazla derecede hassas yapıda olmaktadır.

2) Görüntünün Laplasyanının mutlak değeri, çift kenarların oluşmasına neden olmaktadır ve bu durumda kenar yönünün belirlenmesi mümkün olmamaktadır. Bu durum, bir pikselin, kenarın koyu bölgesinde mi yoksa açık bölgesinde mi olduğunu belirlenmesini zorlaştırmaktadır.

Sıfır geçişler, görüntü içerisindeki cisimlere ilişkin sınırları temsil ettiği için kenarlar sürekli çizgiler halinde olacaktır. Bu sayede, kenar inceltme işlemi Gradient tabanlı yöntemlerde gerekli iken, Laplasyan tabanlı yöntemlerde bu tip bir işleme ihtiyaç duyulmamaktadır.

Görüntü içerisindeki en küçük ani değişimler bile bir sıfır geçişi oluşturacakları için, Laplasyan-tabanlı yöntem ile elde edilen kenar görüntülerinde çok fazla sayıda yanlış kenar noktalarının oluşması ihtimali oldukça yüksektir.

Görüntünün türevini almak demek, görüntüdeki farklılıkların daha çok vurgulanması demektir. Yani kenarların bulunması için türev kavramından yararlanılmaktadır.

Laplasyan filtresi aynı zamanda bir yüksek geçiren filtredir. Bir bakıma yüksek farklılıkların geçmesine izin veren bir filtredir. Farklılık azaldıkça bu filtrenin geçirgenliği de az olmaktadır. Yüksek geçiren filtreler görüntünün keskinleştirilmesini sağlarlar. Bu tip filtrelerde filtre merkezindeki pikseller daha çok önem taşımaktadır (Şekil 3.9).



(a) Orijinal görüntü

(b) Orijinal görüntüye yüksek geçiren filtre uygulanması sonucundaki görüntü

Şekil 3.9. Yüksek geçiren filtreleme işlemi

Alçak geçiren filtreler ise genel olarak görüntüyü yumuşatmak için kullanılır. Böylece bazı tip gürültüler görüntüden çıkarılabilir. Gaussian ve medyan filtreler alçak geçiren filtrelere örnek olarak verilebilir (Şekil 3.10).



(a) Orijinal görüntü



(b) Orijinal görüntüye alçak geçiren filtre uygulanması sonucundaki görüntü

Şekil 3.10. Alçak geçiren filtreleme işlemi

Filtreler genel olarak, bir çekirdek matrisin tüm resim boyunca gezdirilip, her piksel için çarpımlarının toplamlarının alınmasıdır. Böylece her piksel değeri için yeni bir değer hesaplanır.

Filtre çekirdek matrisleri görüntü üzerinde dolaştırılır ancak köşelerde ya da kenarlarda bulunan pikseller söz konusu olduğunda, bu pikseller üzerinde işlem yapılamadığından ya yok sayılmaktadır ya da sabit bir değer verilmektedir. Dışarıda kalan bu pikseller için yeni piksel değeri tayin edilir. Bu işleme interpolasyon ya da ekstrapolasyon denir. Birden çok farklı yöntemle bu işlem gerçekleştirilebilir. Örneğin, komşu pikselin değeri alınabilir. Komşu piksellerin ortalaması alınabilir ya da sıfır değeri ile interpolasyon yapılabilir.

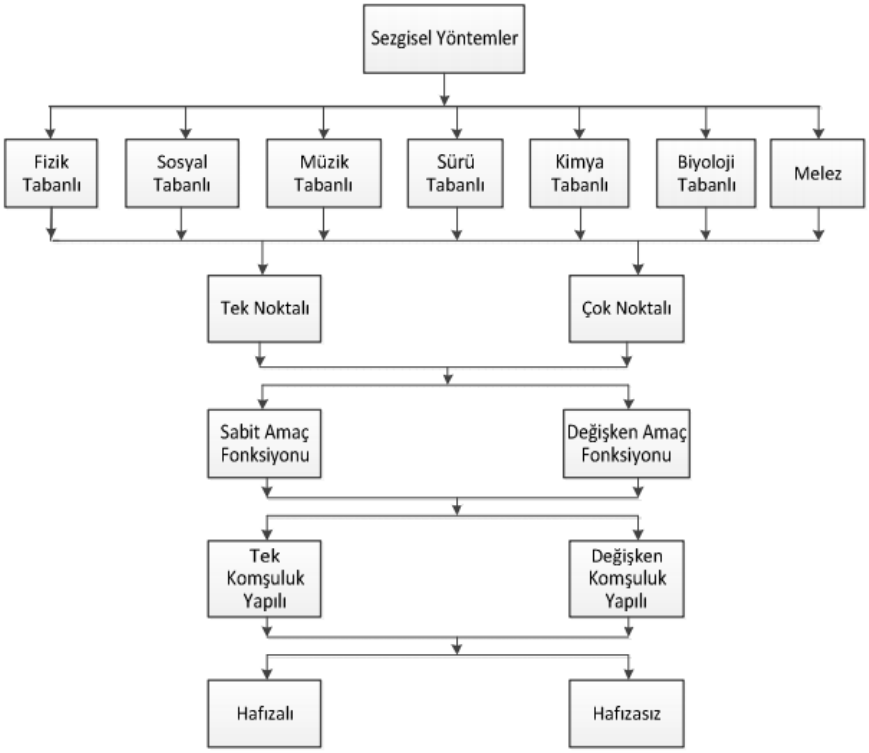
Matrisin büyüklüğü ve içerdiği değerlere göre yapılabilen işlemler de çeşitlilik gösterebilmektedir. Resmi yumuşatma, keskinleştirme, resimde kenar tespiti, alçak geçiren ya da yüksek geçiren filtreler gibi pek çok işlemde çeşitli filtreler kullanılabilmektedir. Ancak genel olarak filtreler ön işleme sırasında kullanılır ve asıl yapılacak işlemlerin daha rahat bir biçimde yapılmasına olanak sağlar. Örneğin, çok parlak bir görüntüde ya da gürültünün çok olduğu bir görüntüde istenen nesneyi tespit etmek çok zordur ve bu tip görüntülerin bir ön işlemden geçirilmesi ise kaçınılmazdır.

3.2.Yapay Arı Kolonisi Optimizasyon Algoritması

Optimizasyon en iyileştirme, kısıtlı kaynakların en iyi şekilde kullanılmasıyla en iyi sonuca ulaşabilme çabasıdır. Matematiksel olarak ise gerçel bir fonksiyonu, maksimize ya da minimize etmek amacıyla, problemi incelemek ya da çözmek kavramını ifade etmektedir.

Sezgisel algoritmalar, büyük boyutlu optimizasyon problemleri için, optimuma yakın çözümler sunan algoritmalardır. Sezgisel optimizasyon algoritmaları sürü tabanlı, biyoloji tabanlı ve kimya tabanlı gibi alt alanlarda da incelenebilmektedir. Sürü zekâsı tabanlı optimizasyon algoritmaları kuş, balık, karınca, bakteri ve arı gibi canlı sürülerinin hareketlerinin incelenmesiyle geliştirilmiştir. Canlıların doğadaki zekice davranışlar sergilemesi ve bu davranışların gözlemlenmesi sürü zekâsı kavramını ortaya çıkarmıştır. Sürüdeki bireyler, davranışlarını sürünün en iyi bireyinin davranışlarından esinlenerek güncellemektedir. Örneğin, bir karınca kolonisinde en yakın yiyecek kaynağını bulan karıncanın feromon salgısı miktarının diğer karıncalarca takip edilmesi davranışı bu zekice davranışa örnek olarak gösterilebilir.

Genetik algoritma (GA), diferansiyel gelişim algoritması, karınca koloni algoritmaları(KKA), yapay sinir ağları (YSA), arı koloni algoritmaları (YAK) ve yapay bağışıklık sistemleri(YBS) biyolojik tabanlı; emperyalist yarışmacı algoritma, parlamenter optimizasyon algoritması ve tabu arama sosyal tabanlı; yapay kimyasal reaksiyon algoritması kimya tabanlı; armoni arama algoritması müzik tabanlı; ısıl işlem, büyük patlama büyük sıçrama, yerçekimsel arama algoritması, merkez kuvvet optimizasyonu, zeki su damlacıkları algoritması ve elektromanyetizma algoritması fizik tabanlı ve Parçacık Sürü Optimizasyonu (PSO), Kedi Sürüsü Optimizasyonu (KSO), Ateş böceği sürü optimizasyonu, yapay balık sürüsü algoritması sürü tabanlı algoritma ve modellerdir. Kültürel algoritma da hem biyoloji hem de sosyal tabanlı algoritma olarak sınıflandırılabilir (Şekil 3.11) (Alataş, 2007).



Şekil 3.11. Sezgisel Yöntemler (Akyol ve Alataş, 2012).

Son yıllarda, araştırmacılar doğrusal veya doğrusal olmayan problemlerin çözümünde, klasik optimizasyon metotlarından yararlanmak yerine sezgisel optimizasyon metotlarını tercih etmeye başlamışlardır. Belirli sınırlamaların sağlanması şartıyla, bilinmeyen parametre değerlerinin bulunmasını içeren herhangi bir problem, optimizasyon problemi olarak adlandırılabilmektedir (Murty, 2003). Sezgisel algoritmalar, çözüm uzayı içerisinde optimum çözüme yakınsaması ispat edilemeyen algoritmalardır (Karaboğa, 2011). Bu tür algoritmalar kesin çözümü garanti etmezler. Modern sezgisel optimizasyon algoritmalarının avantajları matematiksel ön hesaplama gerektirmemeleri, probleme kolaylıkla uygulanabilmeleri, esnek ve doğru çözümler üretebilme yetenekleridir.

Sezgisel algoritmalar, genel olarak doğadaki canlıların yiyecek arama yöntemlerini modelleyen bir yaklaşımlardır. Bu arama süreci ise bir sürüye bağlı olarak gerçekleştirilmektedir. Sürüden edinilen bilgiler ışığında en iyi ve kaliteli yiyeceğin bulunması amaçlanmaktadır.

Sürü, birbirleriyle etkileşim içerisinde olan bireyler yığını anlamında kullanılmaktadır. Bireyler insan, karınca, arı, balık veya bakteri olarak ifade edilebilir. Sürü içerisinde N adet temsilci birey, bir amacı gerçekleştirmek ve hedefe ulaşmak için birlikte çalışmaktadır. Temsilci bireyler, faaliyetlerini gerçekleştirmek için basit bireysel kurallar kullanmakta ve

grubun kalan kısmıyla etkileşim halinde kalarak, başarıya ulaşmaya çalışmaktadır. Bu davranış biçimi ise karşımıza sürü zekâsı kavramı olarak çıkmaktadır (Alataş, 2012).

Millonas bir sürüde sürü zekâsı kavramından bahsedebilmek için sürünün sağlaması gereken beş temel özelliğin sağlanması gerektiğini vurgulamaktadır (Millonas, 1994):

1.Sürü, temel uzay ve zaman hesaplamalarını yapabilmelidir (yakınlık prensibi).

2.Yiyeceklerin kalitesi veya yerin güvenliği gibi çevresel etkenleri değerlendirebilmeli ve tepki verebilmelidir (kalite prensibi).

3.Tüm kaynaklarını dar boğazlarda kullanmamalı ve kaynakları birden fazla noktaya dağıtabilmelidir (dağılım prensibi).

4.Çevrede oluşan her bir değişimde veya dalgalanmada çalışma şeklini değiştirmemeli ve kararlılığını koruyabilmelidir (kararlılık prensibi).

5.Enerji tüketimine değecek maliyette ise çalışma şeklini değiştirebilmelidir (uyarlanabilirlik prensibi).

Son zamanlarda araştırmacılar sürü modellerinden yararlanarak; trafik ve network yönlendirme, endüstriyel problemler, ekonomi problemleri, robotik gibi gerçek hayattaki problemlere bu modelleri uyarlamaktadır.

Doğadaki en ilginç sürülerden biri topluluk zekâlarıyla işlerini dinamik olarak dağıtabilen bal arısı sürüsüdür. Hatta öyle ki, bal arıları yeni yuva seçerken grup olarak karar verme özelliğine sahiptir. Koloni halinde yaşayan arılar; navigasyon sistemine, görsel hafızaya ve kavrama yeteneklerine sahiptirler.

Doğada bulunan bir arı kolonisinde üç çeşit arı bulunmaktadır: Kraliçe arı, erkek arı ve dişi arı (Akay, 2009).

Kraliçe arı sadece birkaç yıl yaşayabilmektedir. Yumurtlama özelliğine sahip olan tek arı olmakla beraber kolonideki tüm arıların annesidir. Yaşamı boyunca genellikle bir kez çiftleşir ve iki yıl boyunca da yumurtlayabilir. Yeni kraliçe arının belirlenmesi ise, kolonideki dişi arılardan birinin yumurtlamanın devamı için kraliçe olarak seçilmesi ile gerçekleşir. (Akay, 2009)

Erkek arıların ömrü en fazla altı ay sürmektedir. Yaz aylarında sayıları birkaç yüz civarı olabilmektedir. Temel görevleri kraliçe arının döllenmesidir. Kraliçe arı ile çiftleşmenin gerçekleşmesi sonucunda ise ölürler (Akay, 2009).

İşçi arılar altı hafta ile dokuz ay arası bir süre yaşayabilmektedirler. Yiyecek toplama, toplanan yiyeceklerin saklanması, kovanın temizlenmesi ve havalandırılması gibi görevlerden sorumludurlar. İşçi arıların görevleri koloni ihtiyaçlarına göre değişmektedir. Yaşamının ilk

başlarında kovan yerini ve çevresini tanıyabilmek için kısa uçuşlar yapar. Geri kalan yaşamında ise yiyecek bulucu olarak çalışmaktadır (Akay, 2009).

Yiyecek bulucu arıların hepsi kendi kaynaklarını kendileri bulmaz. Bu arıların çoğu diğer arılardan aldığı bilgiye göre kaynaklara konumlanır. Arılar sağırdır. Bu nedenle bilgi aktarımını birbirlerine dans aracılığıyla gerçekleştirirler. Bilgi getiren arı dans ederken gözlem yapan arılar da antenleri ile dans eden arıya dokunarak nektarın tat ve koku bilgisini almaya çalışır. Ne kadar fazla arı bu şekilde dans ederse o kadar çok kaynağa yönelme söz konusudur (Akay, 2009).

Arıların yiyecek aramalarından esinlenerek, Yapay Arı Kolonisi Algoritması (YAK) 2005 yılında Derviş Karaboğa tarafından geliştirilmiştir (Karaboğa, 2011).

Arıların yiyecek arama davranışı, bireyler arası bilgi paylaşımı ve ezberleme nitelikleri, son zamanlarda oldukça ilgi çeken araştırma alanlarından biri olmuştur. Bu davranışlardan esinlenilerek oluşturulan Yapay Arı Kolonisi optimizasyon algoritması, yapılan pek çok çalışmada diğer optimizasyon problemlerine göre daha iyi bir performans göstermektedir.

Doğal bir arı kolonisinde arılar arasında bir görev paylaşımı vardır. Arılar bu iş paylaşımını kendi kendilerine gerçekleştirebilmektedirler. Görev paylaşımı ve kendi kendine organize olabilme yeteneği sürü zekâsının iki önemli özelliğidir. Tereshko'ya göre bu zekânın ortaya çıkmasını sağlayan minimal yiyecek arama modelinde üç temel bileşen vardır. Bunlar:

1) Yiyecek kaynakları,

2) Görevli işçi arılar ve

3) Görevsiz işçi arılardır.

Ayrıca bu minimal model:

1) Bir yiyecek kaynağına yönelme ve

2) Yiyecek kaynağını bırakma olmak üzere iki şekilde çalışmaktadır (Tereshko, 2000).

Arılar bal, polen veya nektar olarak tanımlanabilen yiyecek kaynaklarına gitmektedirler. Yiyecek arama davranışlarında; kaynağının miktarı, kovana olan uzaklığı, kaynağın kalitesi, nektarın çıkarılmasının kolaylığı kavramları kaynak kriterleridir. Basit olması açısından sadece yiyecek kaynağının kalitesi gibi tek bir özellik de ele alınabilir.

YAK algoritmasında yiyecek kaynakları, optimize edilmeye çalışılan problemin olası çözümlerine karşılık gelmektedir. Bir kaynağa ait nektar miktarı, o kaynakla ifade edilen çözümün kalite değerini ifade etmektedir. Diğer bir deyişle, kovandaki kullanılan arıların sayısı nektar kaynaklarının sayısına eşittir (Baştürk ve ark., 2006).

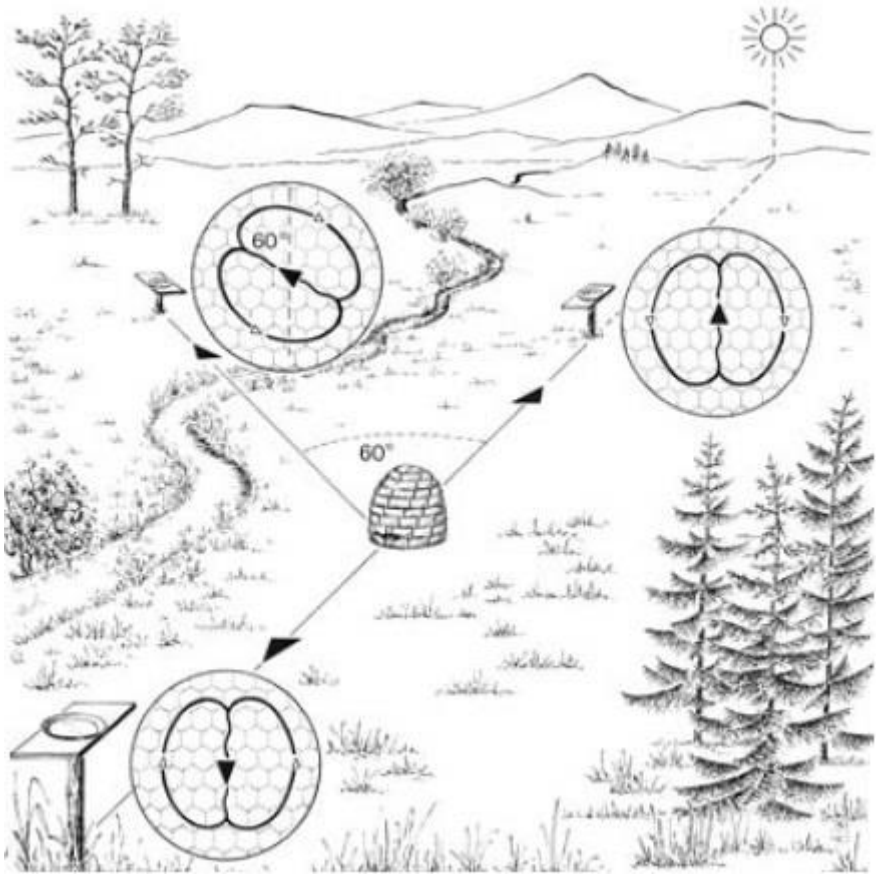
Yapay arı kolonisi algoritmasında, koloni üç gruptan oluşur. Bunlar:

1.Çalışan (işçi-görevli) arılar,

2. Gözcü arılar,

3. Kâşif arılar.

Görevli işçi arılar, önceden keşfedilmiş olan belli kaynaklardan kovana nektar getirilmesinden sorumludurlar. Bu arılar gittikleri yiyecek kaynağının kalitesi ve yeriyle ilgili bilgileri diğer arılarla paylaşmaktadırlar. Bu bilgi paylaşımını "arı dansı" adı verilen dansları ile göstermektedirler. Bu dansa göre yiyecek kaynağının kovana olan mesafesi ve güneşe göre konumu bilgileri gözcü arılara iletilmektedir (Şekil 3.12) (Barth, 1982)



Şekil 3.12. Arı dansı ve gözcü arılara bilgi aktarımı

Gözcü arılar danslardan öğrendikleri bilgilerle bir kaynak seçmektedirler. Burada nektar miktarı uygunluk değerine karşılık gelmektedir. Yiyecek kaynağının seçilme olasılığı rulet tekerleği yöntemi kullanılarak hesaplanmaktadır. Rulet tekerleği yöntemine göre daha iyi uygunluk değerine sahip bireylerin daha fazla seçilme olasılığı vardır. Tüm bireylerin bulunduğu bir rulet tekeri üzerindeki yerleri, bireylerin uygunluk değerleri ile orantılı olarak düşünülmektedir. Daha uygun olan birey daha geniş bir alana sahip olmaktadır. Böylece seçilme olasılıkları da artmaktadır.

A ve B kaynakları iki ziyaret edilen kaynak olarak düşünüldüğünde, kaynak araştırmasının başlangıcında bir işçi arı için kaynakların yerleri bilinmemektedir. Bu arı için iki durum vardır (Şekil 3.13) (Akay, 2009):

1. Kaşif arı olmak. Şekil 3.13'de S olarak gösterilmektedir.

2. Gözcü arı olmak, kuyruk dansı izlemek ve izlediği kaynağa yönelmek. Şekil 3.13'de R ile gösterilmektedir.

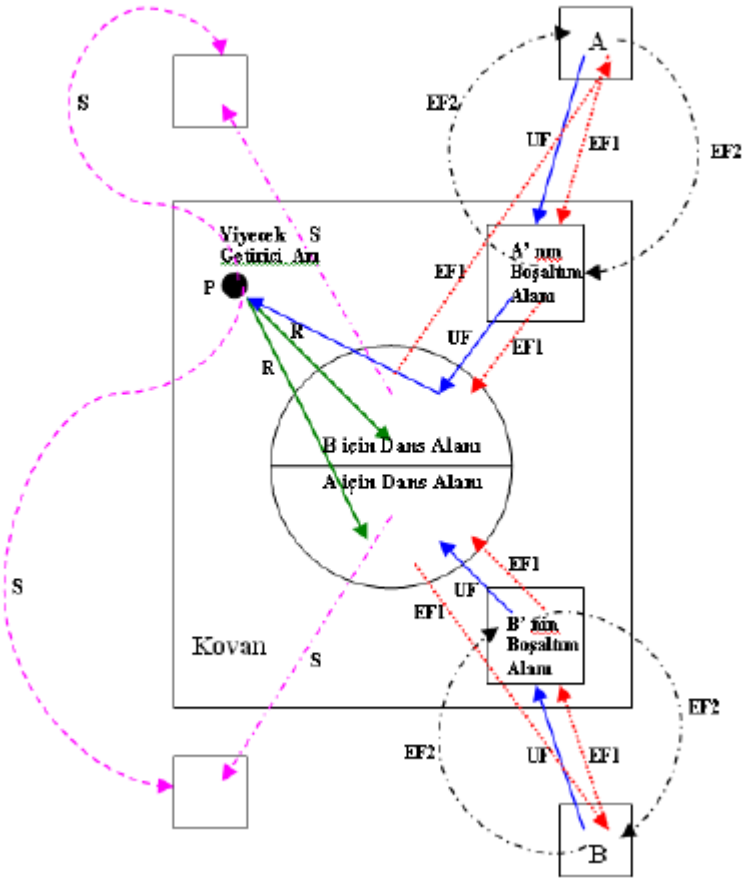
İşçi arı kaynaktan döndükten sonra üç seçenek vardır (Şekil 3.13) (Akay, 2009):

1. Gittiği kaynağı terk eder ve bağımsız işçi arı olur. Şekil 3.13'de UF ile gösterilmektedir.

2. Gittiği kaynağa dönmeden önce dans eder ve diğer arılara bilgi aktarımında bulunur. Şekil 3.13'de EF1 ile gösterilmektedir.

3. Diğer arılara herhangi bir bilgi aktarımında bulunmaz ve aynı kaynağa yönelir. Şekil 3.13'de EF2 olarak gösterilmektedir.

Kolonideki her bir bireyin, uygun bir çözüm olup olmayacağına karar veren bir fonksiyona göre sahip olduğu değer uygunluk değeridir. Bir kaynağın uygunluk değeri ne kadar fazla ise bu kaynağı seçecek olan gözcü arı sayısı da o kadar fazla olmaktadır. Gözcü arılar daha iyi kaynağı seçebilmek için açgözlü seçim işlemi (greedy search) yapmaktadırlar. Bunun için, eski kaynak ile yeni kaynak arasında karşılaştırma yaparlar ve eski kaynak daha iyi ise mevcut kaynak terkedilmez ancak daha iyi bir kaynak bulunamadığından, geliştirilememe durumunu arttırırlar. Ancak yeni çözüm daha iyi ise eski kaynak terkedilmektedir. Mevcut kaynaktan daha iyi bir kaynak bulunduğundan o kaynak geliştirilmiş olur ve geliştirilememe sayacı da sıfırlanmaktadır. Bütün gözcü arılar yiyecek kaynaklarına dağılıncaya kadar bu işlemler devam etmektedir (Karaboğa, 2011).



Şekil 3.13. Bal arılarının yiyecek arama döngüsü

YAK için temel adımlar:

Adım 1.Başlangıç yiyecek kaynağı bölgelerinin üretilmesi.

TEKRAR ET

Adım 2. Görevli arıların yiyecek kaynağı bölgelerine gönderilmesi.

Adım 3. Olasılık değerlerine göre yapılan seçimde kullanılacak olasılık değerlerinin görevli arılardan gelen bilgiye göre hesaplanması.

Adım 4. Gözcü arıların olasılık değerine göre yiyecek kaynağı bölgesi seçmeleri.

Adım 5. Bırakılacak kaynakların bırakılışı ve kâşif arı üretimi.

KADAR (çevrim sayısı=Maksimum çevrim sayısı) (Karaboğa,2011)

Temel Yapay Arı Kolonisi algoritmasının ayrıntılı adımları aşağıdaki gibidir:

1. Denklem 3.4 ile X_{ij}, i=1, …, N, j=1, …, M, çözümlerine başlangıç değeri atanır, geliştirilememe sayaçları ($hata_i$) başlangıçta sıfır olarak ayarlanır. Ve uygunluk değeri hesaplanır. N, yiyecek sayısı ve M ise optimize edilecek parametre sayısıdır.

$$X_{ij} = X_j^{min} + rand(0,1)(X_j^{maks} - X_j^{min}) \quad \textbf{(3.4)}$$

X_{ij} arının konumlanmış olduğu ilk kaynaktır. Konum ise gerçek hayatta arının yiyecek kaynaklarından birisi iken YAK algoritmasında uygunluk değeri hesaplanacak olan, çözüm uzayındaki çözüm adaylarından birini nitelemektedir.

2. i=1 den N'ye kadar Denklem 3.5 kullanılarak X_i çözümünün görevli arısı için yeni bir kaynak üretilir.

$$V_{ij} = X_{ij} + \varphi_{ij}(X_{ij} - X_{kj}) \quad \textbf{(3.5)}$$

V_{ij} X_{ij} 'nin komşuluğundaki yeni bir kaynağı temsil etmektedir. Φ [-1,1] aralığında rastgele bir değerdir ve komşuluktaki yer değiştirmenin rastgele yapılmasını sağlamaktadır.

Üretilen V_{ij} komşu değerinin daha önceden belirtilmiş olan alt ve üst sınırları aşmaması gerekmektedir. Aşması durumunda Denklem 3.6 kullanılarak j. parametreye ait alt veya üst sınır değerleri belirlenir. Böylece belirlenen komşu çözümün, mevut kaynağın komşuluk sınırları içerisinde kalması sağlanmaktadır.

$$V_{ij} = \begin{cases} X_j^{min}, & V_{ij} < X_j^{min} \\ V_{ij}, & X_j^{min} \leq V_{ij} \leq X_j^{maks} \\ X_j^{maks}, & V_{ij} > X_j^{maks} \end{cases} \quad \textbf{(3.6)}$$

Bu kaynağın maliyet değerini f(V_i), Denklem 3.7'de yerine koyarak, bu çözüm için uygunluk değeri hesaplanır. V_i ve X_i arasında seçim işlemi uygulanır ve daha iyi olan seçilir. Eğer X_i geliştirilememiş ise geliştirememe sayacı yani $hata_i$ bir arttırılır

$$\text{Uygunluk}=\begin{cases} \frac{1}{1+f_i}, & f_i \geq 0 \\ 1+abs(f_i), & f_i < 0 \end{cases} \quad \textbf{(3.7)}$$

f_i, kaynağın maliyet değeridir. Maliyet fonksiyonu, probleme ait çözüm adayının çözüm değerini üretmek için kullanılan bir fonksiyondur. Elde edilen maliyet değerine göre çözümün uygun olup olmadığı hesaplanmaktadır.

3. Gözcü arıların seçim işlemi yapılırken, kullanacakları uygunluk değerine bağlı olasılık değerleri Denklem 3.8 kullanılarak hesaplanır.

$$P_i = \frac{uygunluk}{\sum_{j=1}^{N} uygunluk_j} \quad \textbf{(3.8)}$$

4. Rulet tekerleğine göre seçim işleminde her bir kaynak için, [0,1] aralığında üretilen P_i değeri gelişigüzel üretilen bir değerden büyükse, gözcü arı için Denklem 3.6 kullanılarak yeni bir kaynak üretilir ve üretilen V_i ile X_i arasında seçim işlemi uygulanır, daha iyi olan seçilir. X_i çözümü gelişmemişse $hata_i = hata_i$ +1, gelişmişse $hata_i = 0$ yapılır. Bu adım, tüm gözcü arılar yiyecek kaynağı bölgelerine dağılıncaya kadar tekrar edilir.

5. Kaynağın nektarının tükenip tükenmediği kontrol edilir. Eğer tükenmişse Denklem 3.5 kullanılarak yeni üretilen rastgele bir çözümle değiştirilir.

6. En iyi çözüm hafızada tutulur.

7. Sonlandırma koşulları kontrol edilir. Eğer koşullar sağlanmıyorsa, Adım 2'den Adım 6'ya kadar devam edilir.

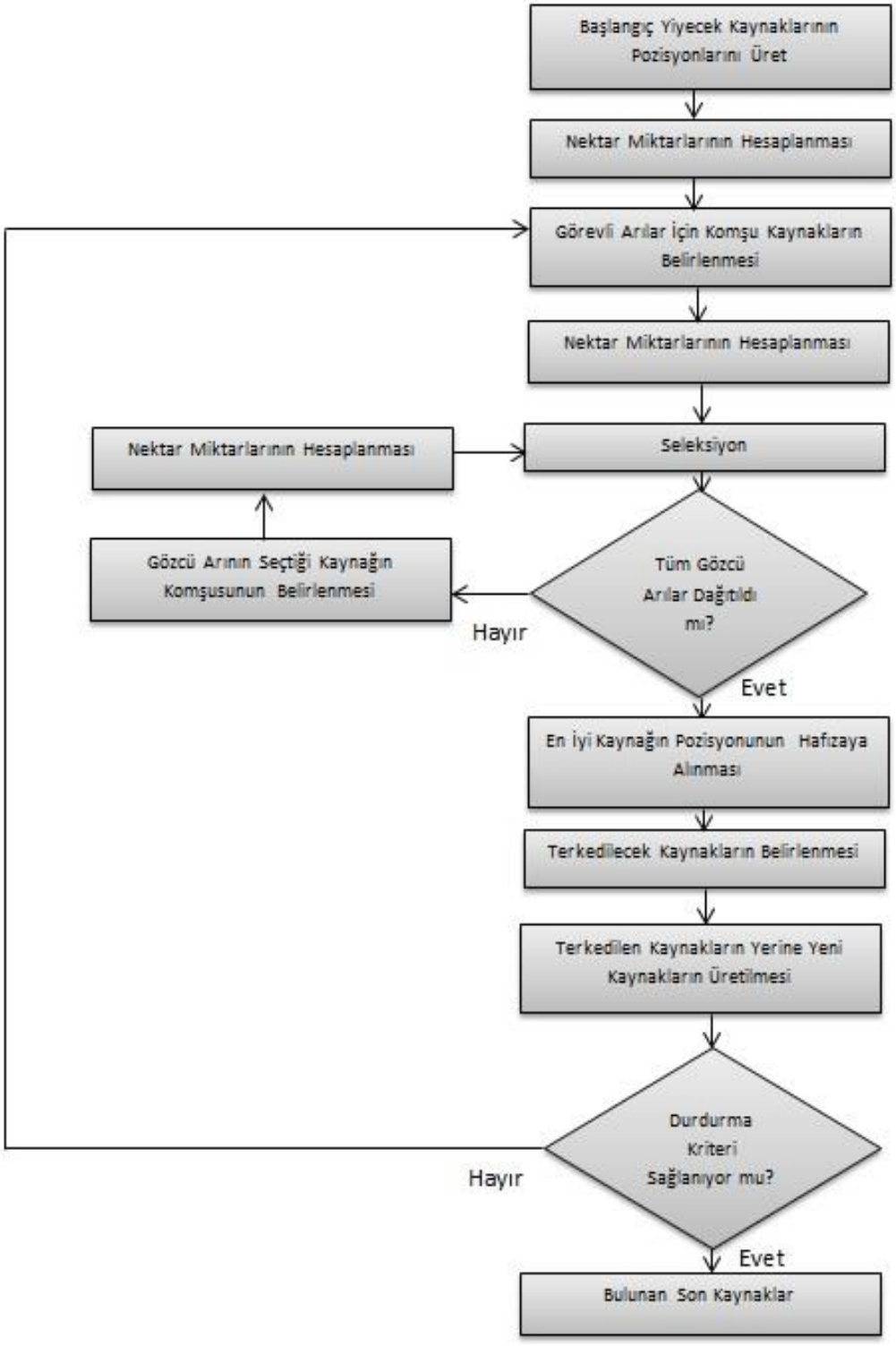
YAK algoritmasında işlem adımlarının tamamlanması yani durdurma kriteri olarak maksimum çevrim sayısı (YAK algoritmasının tüm adımlarının tekrarlanacağı adım sayısı) kullanılabileceği gibi, maksimum kâşif arı sayısı da kullanılabilmektedir (Karaboğa, 2011).

Genel olarak YAK optimizasyon algoritması özellikleri:

- Sürü zekâsına dayalı bir algoritmadır.

- Esnek ve basittir.
- Oldukça az kontrol parametresine sahiptir
- Gerçek yiyecek arayıcı arıların davranışlarından esinlenir.
- Nümerik problemler için geliştirilmiştir ama ayrık problemler içinde kullanılabilir.
- Kâşif Arılar tarafından gerçekleştirilen küresel; görevli ve gözcü arılar tarafından gerçekleştirilen bölgesel araştırma kabiliyetine sahiptir. Bu lokal ve global araştırma süreçleri paralel yürütülmektedir (Karaboğa, 2011).

YAK optimizasyon algoritması için temel adımların akış diyagramı Şekil 3.14'da verilmiştir.



Şekil 3.14. YAK Akış Diyagramı

Temel yapay arı kolonisi algoritması üç adet kontrol parametresine sahiptir:

1. Maksimum çevrim sayısı,
2. Koloni büyüklüğü,
3. Limit

Popülasyon büyüklüğünün belirlenmesi için fazla bir çaba harcanmasına gerek duyulmamaktadır. Çünkü daha düşük popülasyon büyüklüğüne sahip durumlarda da iyi sonuçlar üretilebilmektedir. Ancak popülasyon büyüklüğünün gereksiz arttırılması, belli bir aşamadan sonra, işlem karmaşıklığının artmasına ve tesadüfi sonuçların ortaya çıkmasına neden olabilmektedir. Bu da algoritmanın araştırma yeteneğinden yararlanılmıyor olduğunu göstermektedir.

Limit değerinin, olması gerekenden küçük bir değer seçilmesi, yakınsama olmadan popülasyondaki bilgilerin yok olmasına ve performansın düşmesine sebep olabilmektedir. Bu durum ihtiyaç duyulandan fazla kâşif arı oluşmasına neden olmaktadır. Eğer popülasyon büyüklüğü fazla ise limit değerinin büyük olması, problemin çözümü açısından bir sorun değildir. Bunun nedeni nüfus büyüklüğünden kaynaklanan yeterli derecede farklılığın mevcut olmasıdır. Yani algoritmada, yeterince farklılık olduğundan yeni kaynakların keşfedilmesi için kâşif arı üretimine gerek duyulmaz. Ancak tersi bir durumda büyük limit değeri gereğinden az kâşif arı çıkmasına ve küresel araştırma yeteneğinin zayıflamasına neden olabilmektedir.

Yapay arı kolonisi algoritmasının her çevrimde N adet arı ile işlem yaptığından dolayı maksimum çevrim sayısının artması hesaplama maliyetini attırabilmekte ve performansı zaman açısından negatif etkileyebilmektedir. Bu nedenle dikkatli bir şekilde belirlenmelidir.

Yapılan araştırmalarda YAK algoritmasının, çoğu sürü zekâsına dayalı algoritmadan daha başarılı sonuçlar ürettiği ve basit yapısına rağmen karmaşık yapıdaki problemler üzerinde de başarılı performans sergilediği görülmüştür.

3.3. Performans Analizi

3.3.1. Hamming Uzaklığı

Hamming Uzaklığı (HU) (Hamming Distance-HD), ismini Richard Hamming'den almaktadır (Polat, 2010). Bilgisayar bilimlerinde, aynı uzunluktaki iki dizgi (string) arasındaki farklılığın hesaplanması alanında kullanılmaktadır.

Hamming mesafesini hesaplamak için iki dizgi arasında XOR uygulanır (Denklem 3.9) ve 1'lerin sayısı hesaplanır (Corner, 2008).

$$\mathrm{HU} = \frac{1}{N}\sum_{j=1}^{N} X_j \,(XOR) Y_j \qquad \textbf{(3.9)}$$

Elde edilen sonuç 0 ile 1 arasında değişiklik göstermektedir. Eğer iki dizgi de aynı ise sonuç 0 olmaktadır. Denklem 3.9' da N dizgilerdeki eleman sayısını, X_j ve Y_j de karşılaştırılan dizgi elemanlarını temsil etmektedir.

Görüntü işlemede ise HU, iki görüntünün piksel değerlerinin karşılaştırılması ile gerçekleştirilir. Doğru sonuç olarak kabul edilmiş bir görüntüdeki her bir piksel, test görüntüsünün aynı koordinatta bulunan pikselleri ile XOR işlemine sokulur. Bu işlemler için kullanılan görüntüler siyah beyaz resimlerdir. Görüntüdeki tüm pikseller kullanılarak Denklem 3.9 hesaplanır ve sonuç HU değeri elde edilir.

3.3.2. Duyarlılık (Sensitivity)- Özgüllük (Specificity) Analizi, Doğruluk Oranı, Hata Tespiti ve Benzerlik

Problemlerin analiz sürecinde, elde edilen sonuçların gerçeği ne kadar yansıttığı ispatlanmaya çalışılmaktadır. Bu analiz yöntemlerinden biri de Duyarlılık (Sensitivity)- Özgüllük (Specificity) analizidir. Tablo 3.1'de kenar tespiti için oluşturulmuş analiz tablosu değerleri görülmektedir. Bu tablo farklı problemler için problemin ismine göre uyarlanabilmektedir (Karlıkaya, 2003).

Tablo 3.1. Kenar bulma için Duyarlılık (Sensitivity)- Özgüllük (Specificity) analiz değerleri

	Kenar (+)	**Kenar (-)**
Test (+)	Gerçek Pozitif (GP)	Yanlış Pozitif (YP)
Test (-)	Yanlış Negatif (YN)	Gerçek Negatif (GN)

Duyarlılık, analizin doğru olarak gösterdiği sonuçların, gerçekte olan tüm doğrulara olan oranıdır. Aranan kriterin bulunması durumunda test sonucunun pozitif olma olasılığıdır (Denklem 3.10). (Altınışık, 2004)

$$\text{Duyarlılık (\%)} = [\mathrm{GP}/(\mathrm{GP}+\mathrm{YN})] \times 100 \qquad \textbf{(3.10)}$$

Özgüllük, analizin yanlış olarak gösterdiği sonuçların, gerçekte olan tüm yanlışlara olan oranıdır. Aranan kriterin bulunmaması durumunda test sonucunun negatif olma olasılığıdır (Denklem 3.11) (Altınışık, 2004).

Özgüllük (%) = [GN/(GN+YP)] x 100 **(3.11)**

Doğruluk, doğru sonuçların genel popülasyona oranıdır (Denklem 3.12) (Altınışık, 2004).

Doğruluk (%) = [GP+GN/(GN+GP+YN+YP)] x 100 **(3.12)**

3.3.3. Hata Tespiti ve Benzerlik Hesapları

Bir problemin analizinde ölçüm kavramı olan Hata Tespiti (HT) (Detection Error-DE), duyarlılık-özgüllük analizinden elde edilen GP ve YP değerlerine bağlı olarak hesaplanmaktadır (Denklem 3.13) (Giannarou ve Stathaki, 2011).

$$\text{Hata Tespiti} = \sqrt{(1 - GP)^2 + (YP)^2} \quad \textbf{(3.13)}$$

Benzerlik (B) (Similarity-S), kenar bilgisi içeren iki görüntü arasında, hata tespiti analizinde olduğu gibi GP ve YP oranlarına göre yapılan analizdir (Giannarou ve Stathaki, 2011).

Test sonuçlarının karşılaştırılması aşamasında benzerlik oranını hesaplamak için uzaklık kavramından yararlanılmaktadır. Bir popülasyondaki tüm sonuçların, doğru kabul edilen değerlere olan uzaklıklarının, denklem 3.14'deki gibi kullanılmasıyla benzerlik değeri elde edilmektedir. Bu denklemdeki N, mesafe bilgisi hesaplanması gereken toplam birim sayısıdır (Giannarou ve Stathaki, 2011).

$$Benzerlik = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{1 + d_i^2} \quad \textbf{(3.14)}$$

Denklem 3.14'de d mesafesi test aşamasında öklit (euclidean) mesafesi kullanılarak belirlenmiştir (Denklem 3.15).

$$\text{d} = \sqrt{(X_1 - X_2)^2 + (Y_1 - Y_2)^2} \quad \textbf{(3.15)}$$

Denklem 3.15' de X_1, Y_1 ground truth görüntülerden kenar bilgisi çıkarılmış resimlerin koordinat bilgilerini gösterirken, X_2, Y_2 karşılaştırma yapılacak olan ve yine kenar bilgisi

içeren görüntünün koordinatlarını göstermektedir. Tüm pikseller için d değeri hesaplanmakta ve denklem 3.14'de kullanılarak benzerlik analizi gerçekleştirilmektedir.

4. YAK İLE KENAR BULMA UYGULAMASI

4.1. Kullanılan Görüntüler

YAK ile kenar bulma yöntemi hem gri seviye görüntüler, hem de siyah beyaz görüntüler üzerinde test edilmiştir.

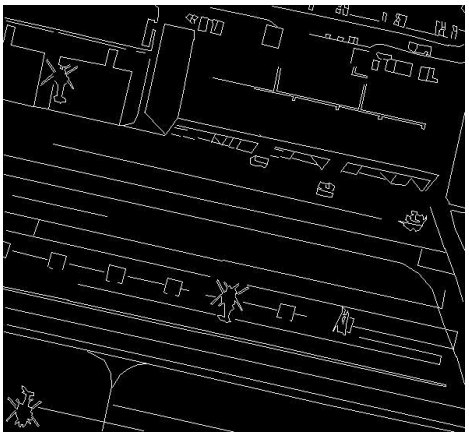
Yapılan çalışmada, kenar bulma algoritması için RADIUS/DARPA-IU Fort Hood hava görüntüleri veri tabanından (Bowyer K. W., 1997) farklı türde 2 grup halinde test görüntüsü alınmıştır. İlk grup veri setinde 5 adet gri seviye nesne görüntüsü bulunmaktadır. İkinci grupta ise 10 adet gri seviye hava fotoğrafı bulunmaktadır. Hava fotoğraflarının, "ground truth" diye isimlendirilen siyah-beyaz kesin referans haritası sonuç görüntüleri de bulunmaktadır (Şekil 4.2.b). İlk grup veri setindeki görüntüler farklı boyutlarda ama tümü .tif formatında görüntülerdir. Hava fotoğrafları ise .pgm dosya formatında ve farklı piksel boyutlarında görüntülerdir. Bowyer'dan alınan nesne görüntülerine örnek Şekil 4.1'de, hava fotoğraflarına örnek ise Şekil 4.2'de verilmiştir (Heath ve ark., 1997).



Şekil 4.1. RADIUS/DARPA-IU Fort Hood nesne görüntülerine örnek (207.pgm)



(a) Orijinal görüntü



(b) Kenar bilgisi çıkarılmış hava görüntüsü (Referans haritası)

Şekil 4.2. RADIUS/DARPA-IU Fort Hood hava görüntülerine örnek (airfield.pgm)

Tez çalışmasında kullanılan diğer altı test görüntüsü ise Gonzales ve Woods'dan alınmıştır. Bu görüntülerden biri gri seviyeli, diğerleri ise siyah beyaz görüntülerdir ve tiff dosyası tipinde tanımlanmışlardır. Her bir görüntünün boyutları birbirinden farklıdır (Gonzalez ve Woods, 2007). Gonzales'ten alınan görüntülere örnek Şekil 4.3'de verilmiştir.



Şekil 4.3. Gonzales'ten alınan nesne görüntülerine örnek (Fig0914(a)(licoln from penny).tif)

4.2. Yapılan Uygulama

Bu çalışmada MATLAB 7.9.0,2009 kullanılarak görüntüler üzerinde kenar bilgisi çıkarımı gerçekleştirilmiştir. Gri seviyeli ve siyah beyaz görüntüler üzerinde, geliştirilen algoritma ile uygulamalar denenmiştir. Farklı iterasyon sayılarında uygulama test edilmiş ve sonuç görüntülerinde başarılı kenar bilgisi elde edilmiştir. Elde edilen bu sonuç görüntüleri

literatürdeki diğer kenar bulma yöntemleri ile karşılaştırılmış ve benzerlik analizleri hesaplanmıştır.

Yapılan tez çalışmasında, geleneksel olan (türeve dayalı) ya da önceden eğitim gerektiren (önceden bir desen tanıtma) kenar bulma yaklaşımlarından daha farklı bir yöntem geliştirilmesi amaçlanmıştır. Bilinen kenar bulma yöntemleri bir maske aracılığı ile kenar haritalarını verebilmekte ve bu bağımlılık Yapay Arı Kolonisi optimizasyonu ile bu çalışmada yok edilebilmeye çalışılmıştır.

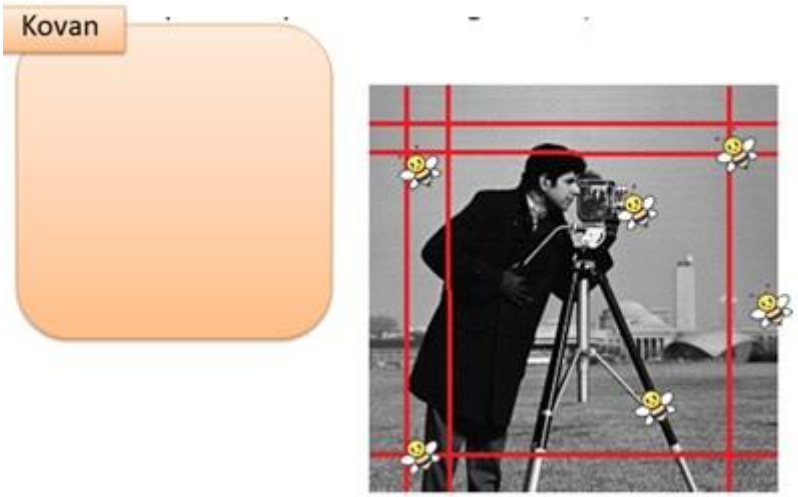
YAK ile görüntüden kenar bulma işlemi gerçekleştirmek için, başlangıçta kenar bilgisi çıkarımı yapılacak olan görüntü(initial image), çözüm uzayı olarak tanımlanmaktadır. Koloni büyüklüğü, limit ve maksimum çevrim sayılarının başlangıç değerleri ayarlanmaktadır. Koloni büyüklüğü, bir popülasyondaki birey sayısıdır. Limit, YAK algoritmasında bir kaynağın geliştirilememe sayacının, bu değere eşitlenmesi sonucunda, terkedilmesi aşamasına geçilmesini gerektiren sınır değeridir. Maksimum çevrim sayısı, problemi çözmek için yapılacak işlemin, tekrar sayısıdır. Koloni büyüklüğü, görüntünün satır ve sütun sayılarının çarpımının kareköküne eşit olarak belirlenmektedir. Koloni büyüklüğü, her görüntü için sabit bir değer olmaması gerektiğinden formüle edilmiştir ve Denklem 4.1'e göre hesaplanmaktadır. Burada K toplam kaynak sayısı, N görüntüdeki satır, M'de sütun sayısıdır (Nezamabadi, 2006).

$$K = \sqrt{N x M} \qquad \textbf{(4.1)}$$

İlk adımda gidilen kaynak sayısı koloni büyüklüğünün yarısı olarak ayarlanmaktadır. Gidilen kaynak sayısı aynı zamanda görevli arı sayısı olmaktadır ve Denklem 4.2'deki formüle göre hesaplanmaktadır (Nezamabadi, 2006).

$$Gidilen\ Kaynak\ Sayısı = \frac{K}{2} \qquad \textbf{(4.2)}$$

Diğer bir kontrol parametresi olan limit ise bu tez çalışmasında 5 olarak belirlenmiştir. Bu ayarlamalardan sonra görevli arılar belirlenen görüntü üzerinde rasgele konumlanmaktadırlar. Şekil 4.4 arıların görüntü üzerinde rastgele konumlanmasını göstermektedir.



Şekil 4.4: Arıların resim üzerinde rasgele konumlanmasının temsili gösterimi

Görevli arıların gitmiş olduğu her piksel, algoritma yapısındaki kaynak bilgisine karşılık geldiğinden, her kaynağa ait uygunluk değeri hesaplanır. Görevli arıların konumlandığı kaynaklara ait koordinat, gri seviye değerleri, uygunluk değeri (*fitness*), geliştirilememe sayacı (*failure)*, olasılık, yasaklı kaynak olup olmama bilgileri "Kaynaklar" adlı matriste tutulmaktadır. Uygunluk fonksiyonu, popülasyondaki her bir bireyin uygunluk değerini elde etmesini sağlayan fonksiyondur. Burada gidilen kaynak (piksel) için uygunluk değeri; o kaynağın gri seviye değerinin uygunluk fonksiyonundan çıkan değeridir. Geliştirilememe sayacı, gidilmiş olan kaynağın komşuluğunda daha iyi bir kaynak bulunamamasında arttırılan bir sayaçtır. Olasılık, mevcut kaynağın uygunluk değerinin bütün kaynakların toplam uygunluk değerine oranını ifade ederken, yasaklı kaynak ise geliştirilememe sayacı limit değerine (yani 5'e) eşitlenmiş olan kaynaktır. Şekil 4.5, Kaynaklar matrisinden örnek göstermektedir.

Gidilen Kaynak Sayısı	Gri Seviye Değeri	X Koordinatı	Y Koordinatı	Geliştirilememe Sayacı	Olasılık	Uygunluk Değeri	Yasaklı Kayna Mı
1	12	199	51	1	0.0265	0.0769	0
2	80	138	100	5	0.0079	0.0250	1
3	14	192	118	2	0.0230	0.0667	0
...	...	...	...	...	...	...	...
128	169	19	69	0	0.0020	0.0059	0

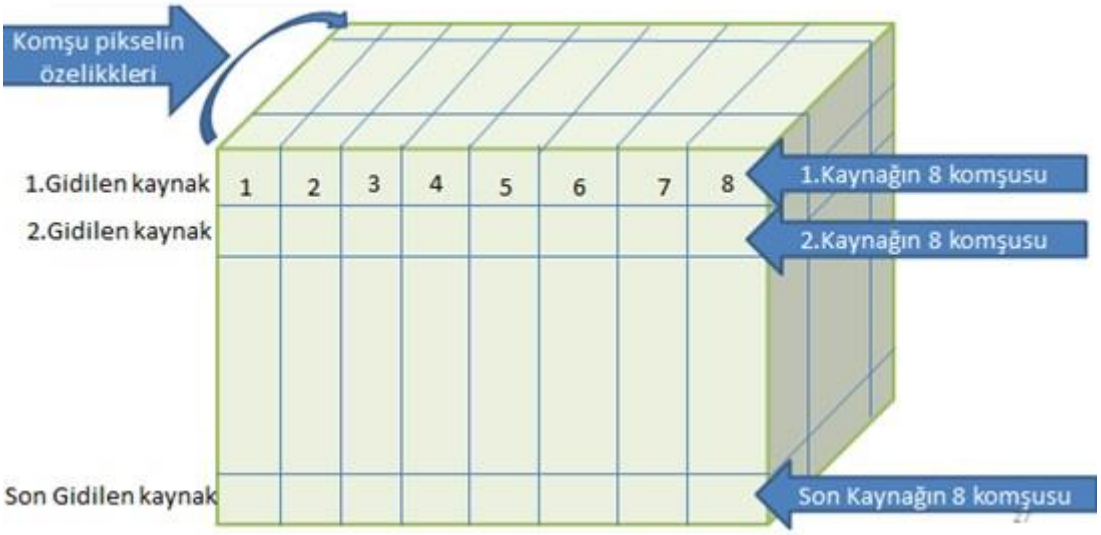
Şekil 4.5. Kaynaklar matrisinden örnek

Görevli arılar, gitmiş oldukları kaynakların geliştirilip geliştirilmemesini de, YAK algoritmasının adımlarına göre kontrol etmektedir. YAK algoritmasında, görevli arıların gitmiş olduğu kaynakların komşuları arasında uygunluk değeri daha iyi olan bir kaynağın olup olmadığı araştırılmaktadır. Bundan dolayı kaynak piksellerin komşuluk bilgileri de tutulmaktadır.

Gidilen kaynakların bilgileri haricinde, yasaklı kaynaklara ait bilgiler de tutulmaktadır. Yasaklı kaynak geliştirilemeyen ve geliştirilememe sayacı limit değerine eşitlenmiş olan kaynak yani yiyecek kaynağıdır.

Gidilen kaynağın konum bilgisi olan koordinat bilgileri, yasaklı yani geliştirilememiş (terkedilmiş) kaynak koordinatlarına eşit mi diye kontrol edilmekte ve eşit ise arının konumlandığı o kaynak üzerinde bir daha işlem yapılmamak üzere terk edilmektedir. Keşfetme ve sömürme işlemleri, bir kaynağın bulunması ve onun tüketilmesi anlamına gelmektedir ve geliştirilen yöntemde bu adımlar yasaklı olmayan kaynaklar üzerinden devam etmektedir.

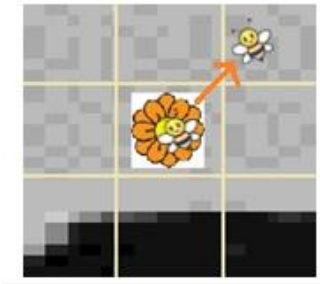
Gidilen kaynak, yasaklı değil ise konumlanılan pikselin komşularının bilgileri de (koordinat, gri seviye değerleri, geliştirilememe sayacı ”failure”, olasılık, yasaklı kaynak olup olmama bilgileri) “Komsular” adlı bir matriste tutulmaktadır. Komsular matrisi Şekil 4.6’teki gibidir.



Şekil 4.6. Komşular matrisi

YAK algoritmasında uygunluk değerinin hesaplanmasında, kaynak piksellerin gri seviye değerleri ile görüntünün eşik değeri arasındaki fark kullanılmaktadır. Gidilen kaynak piksellerinin uygunluk değerleri hesaplanmakta ve ardından da her bir kaynağa ait olasılık değeri yani gözcü arılar tarafından seçilebilme değerleri hesaplanmaktadır.

Eğer gidilen kaynağın olasılık değeri, belirlenen olasılık sınırından küçük ise, gidilen kaynağın olasılığının, rastgele belirlenen komşusunun olasılığından az olup-olmadığına bakılmaktadır. Böylece rastgele gidilen kaynak geliştirilmeye çalışılmıştır. Şekil 4.7 gidilen kaynağın geliştirilmesini göstermektedir.

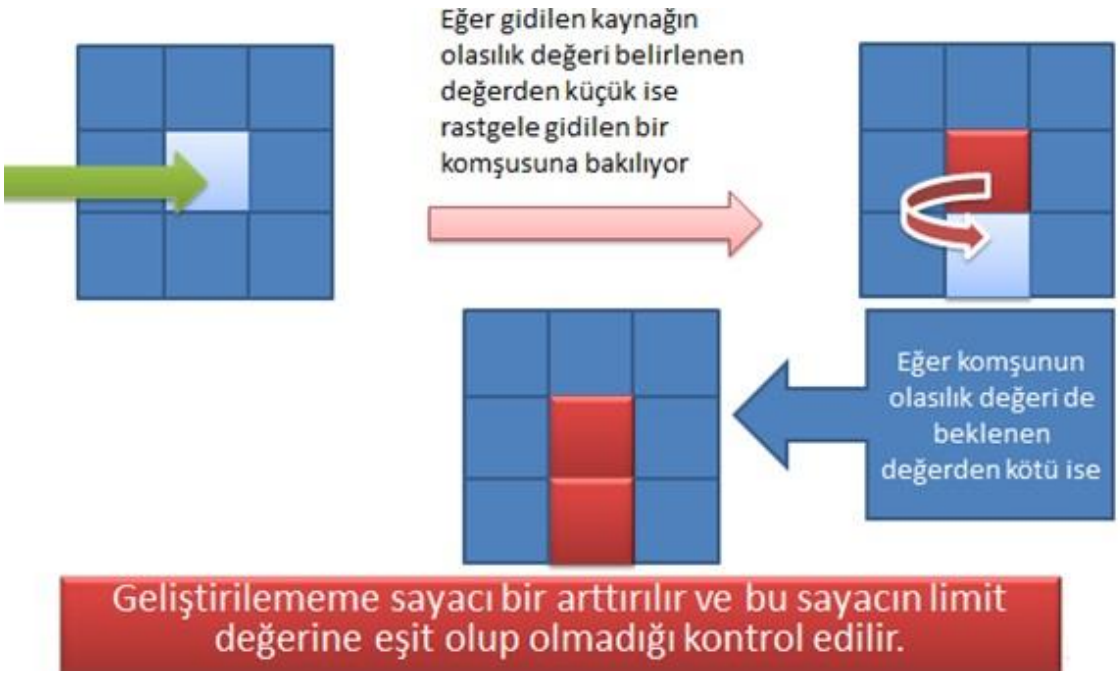


Şekil 4.7. Kaynağın komşu pikselinin olasılık değerinin kontrolü

Eğer komşu pikselin olasılık değeri de belirlenen değerden küçükse o kaynağın geliştirilememe sayacı "failure" bir arttırılmaktadır ve limit değerine ulaşılmışsa o kaynak terk edilmektedir. Şekil 4.8 geliştirilememe sayacının artışını göstermektedir.

Yapılan çalışmada limit değeri 5 olarak belirlenmiştir. Gidilen kaynak için geliştirilememe sayacı limit değerine ulaştıysa kâşif arı (scout_bee) sayısı bir arttırılmaktadır. Çünkü gidilen kaynak terk edilmektedir ve onun yerini alacak olan yeni bir kaynak arayışı

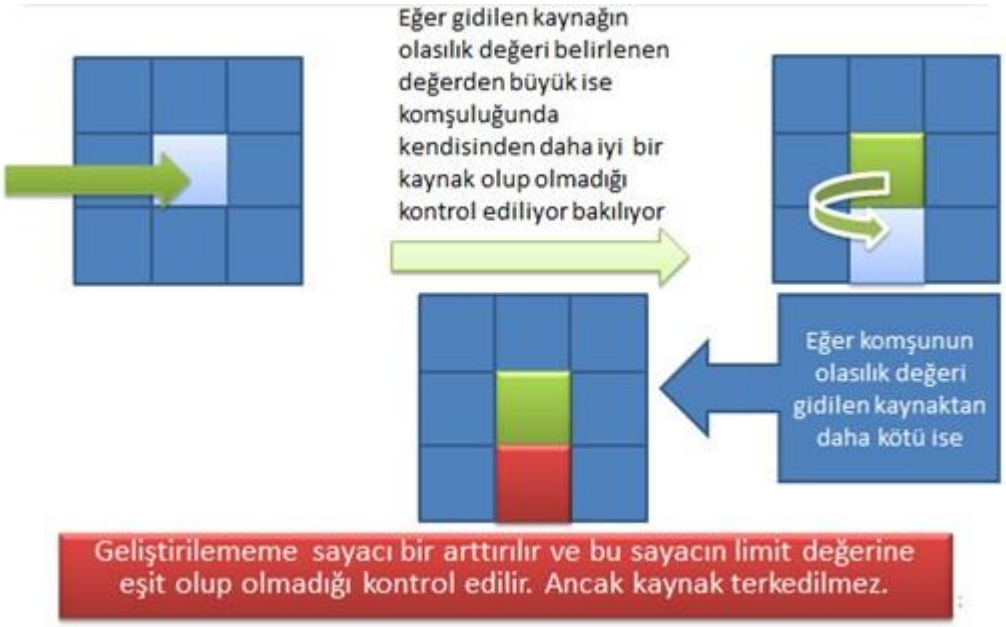
başlamaktadır. Terkedilen kaynağın koordinat bilgileri de yasaklı kaynaklar matrisinde tutulmakta ve böylece o piksellere tekrar arıların konumlanması engellenmektedir.



Şekil 4.8. Gidilen kaynağın hem kendisinin hem de komşusunun olasılık değerinin belirlenen değerden düşük olması durumu

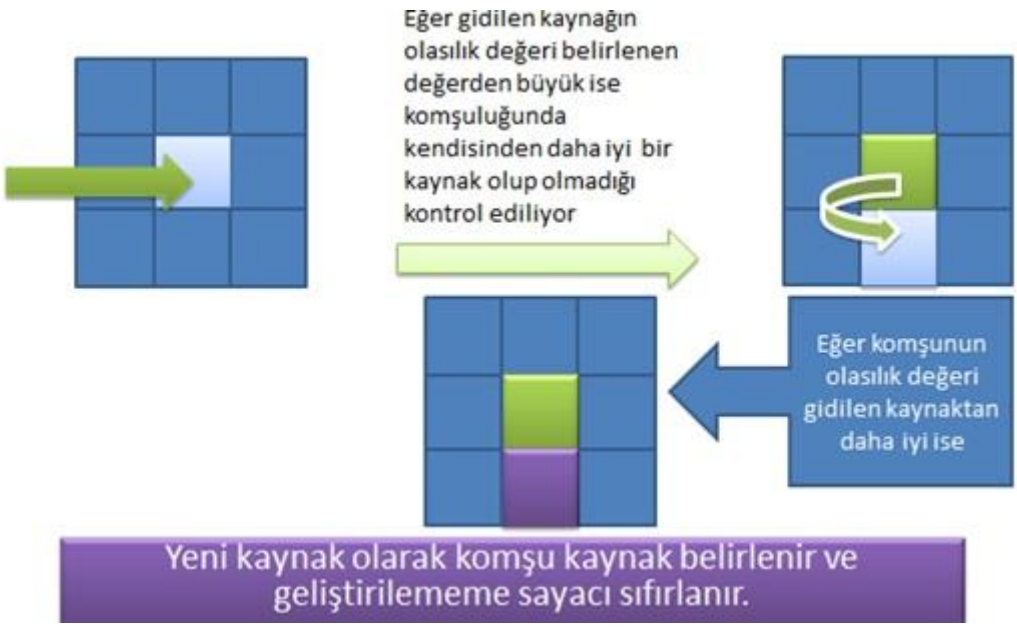
Eğer gidilen kaynağın olasılık değeri, belirlenen olasılık değerinden büyük ise, komşuluğunda kendisinden daha iyi bir kaynak olup – olmadığına bakılmaktadır. Şekil 4.9'de, komşunun olasılık değeri, gidilen kaynaktan daha kötü ise geliştirilememe sayacının bir arttırılıp, limit değerine ulaşıp ulaşmadığı kontrol edildiği gösterilmektedir

Limit değerine ulaşılmamışsa, gidilen kaynak, başka bir komşunun olasılık değerinin kontrolü yapılabilmesi için terk edilmemektedir. Ancak limit değerine ulaşılmışsa kaynak terk edilmekte ve yasaklı kaynak matrisine kayıt edilmektedir. Böylece bu kaynağa daha sonraki adımlarda konumlanma engellenmiş olmaktadır.



Şekil 4.9. Gidilen kaynağın olasılığı belirlenen değerden yüksek, fakat komşusunun olasılık değerinin belirlenen olasılık değerinden küçük olması durumu

Eğer gidilen kaynağın olasılık değeri, belirlenen olasılık değerinden büyük ve komşunun olasılık değeri de gidilen kaynağın olasılık değerinden büyük ise, komşu piksel yeni kaynak olarak belirlenmekte ve gidilen kaynak terk edilmektedir. Komşu piksel yeni kaynak olarak belirlendiğinden dolayı, geliştirilememe sayacı tekrar sıfırlanmaktadır. Şekil 4.10 komşu pikselin yeni kaynak olarak atanmasını göstermektedir.



Şekil 4.10. Gidilen kaynağın olasılığı belirlenen değerden yüksek olduğu halde komşusunun olasılık değerinin daha iyi olması durumunda yeni kaynağın komşu kaynak olarak belirlenmesi durumu

Gidilen kaynakların olasılıklarının, belirlenen olasılık değerinden büyük olması durumunda, gözcü arıların kaynaklara yönlendirilmesi aşamasına geçilmektedir.

Eğer gidilen kaynağın olasılık değeri, belirlenen olasılık değerinden büyük ise o kaynağın komşuları arasındaki yönsel açıdan *(*sağ-sol, alt-üst, çapraz olmak üzere*)* gri seviye farkları alınmaktadır. Şekil 4.11 yönsel açıdan komşulukların alınmasını göstermektedir.



Şekil 4.11. Gidilen kaynağın kenar piksel olup olmamasının komşularının gri seviye değerlerine bakılarak karar verilmesi aşaması

Eğer hesaplanan gri seviye farkı, belirlenen değerin üçte birinden yani belirlenen "eşik değerden" daha fazla ise o pikselin kenar olduğu belirlenmekte ve sonuç resimde o pikselin değeri 1 yapılmaktadır. Belli bir oran seçilmesinin nedeni ise kenar bilgisinin kaybolmasını engellemektir. Eşik değer bilgisi her görüntü için o görüntüdeki gri seviyelerin standart

sapmasına bağlıdır ve Denklem 4.3'e göre hesaplanmaktadır. Burada N piksel sayısı, $\bar{x}$ ortalama gri seviye değeri ve x_i ilgili pikselin gri seviye değeridir.

$$esik\ deger = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2} \qquad \textbf{(4.3)}$$

Bu işlemler 500'den başlayarak 50000'e kadar farklı çevrim sayıları için denenmiştir.

Test görüntüleri için uygunluk değeri belirlemesinde kullanılmak üzere eşik değerler belirlenmiştir**.** Gri seviye olan tüm resimler için siyah-beyaz resme çevrilmede kullanılan eşik değeri (threshold), her bir resim için değişiklik göstereceğinden, eşik değeri resimdeki piksellerin gri seviyelerinin standart sapmasına göre belirlenen bir formülle hesaplanmıştır. Böylece tek bir resim için değil tüm resimler için ayarlanmış bir optimizasyon yöntemi elde edilmiştir. Eşik değeri için hesaplanan formül Denklem 4.3'de verilmiştir.

Olasılık değeri sınırı, gidilen kaynakların olasılık değerlerinin ortalaması alınarak hesaplanmaktadır. (Denklem 4.4)

$$\text{olasılık_deger} = \frac{Tüm\ Kaynakların\ Olasılık\ Değerleri\ Toplamı}{Toplam\ Kaynak\ Sayısı} \qquad \textbf{(4.4)}$$

4.3 Geliştirilen YAK algoritmasının özellikleri

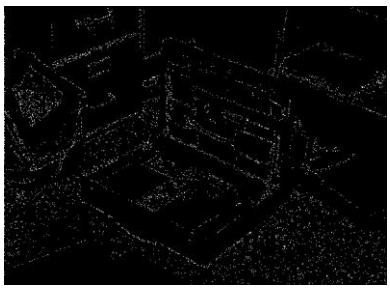
Genel YAK algoritmasına göre, başlangıç yiyecek kaynaklarının üretilmesi formüle bağlı yapılmamaktadır. Genel YAK algoritmasındaki bu formül çözüm uzayındaki maksimum ve minimum sınırlar arasında rasgele bir kaynak üretilmesi için bulunmaktadır. Yapılan çalışmada, görüntü ve üzerindeki pikseller çözüm uzayını oluşturmaktadır ve her bir piksel bir kaynak olarak düşünülmektedir. Görüntünün satır ve sütun boyutları ise çözüm uzayının sınırlarını belirtmektedir. Görevli arılar, bu sınırlar içerisindeki piksellere gönderilmektedir.

Durdurma kriteri olarak maksimum çevrim sayısı kullanılmıştır. Limit değeri ise 5 olarak belirlenmiştir. Popülasyon büyüklüğü de görüntü boyutlarına göre ayarlanmıştır.

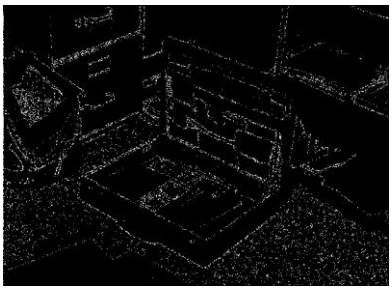
Genel YAK algoritmasında kaynağın tükenip-tükenmemesi durumuna göre yeni bir kaynak arayışı işlemi başlarken, yapılan çalışmada terk etme kriteri nektarın kalitesi olarak da değerlendirilebilecek olan, daha iyi bir kaynağın bulunup bulunmamasıdır.

5. ARAŞTIRMA SONUÇLARI

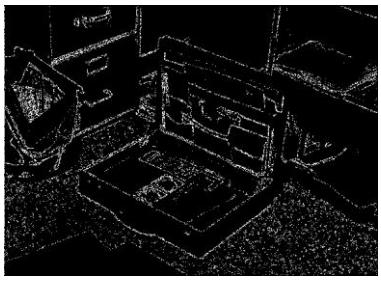
YAK algoritması ile kenar bulma algoritması uygulamasında ilk önce RADIUS/DARPA-IU Fort Hood veri tabanından alınan 5 adet gri seviye nesne görüntüsü üzerinde işlem yapılmıştır (Bowyer, 1997). Bu resimler üzerinde 500'den 5000'e kadar farklı çevrim sayılarında, YAK algoritması ile kenar bulma işlemleri yapılmıştır. En iyi sonuç görüntüleri Maksimum Çevrim Sayısı (MCS) 5000 çevrim sayısında elde edilmiştir. Farklı çevrim sayılarında elde edilen sonuç görüntülerine örnek Şekil 5.1'de verilmiştir.



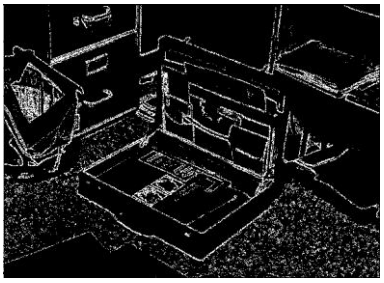
(5.a) MCS = 500



(5.b) MCS = 1000



(5.c) MCS = 2000



(5.d) MCS = 5000

Şekil 5.1 36.pgm için farklı maksimum çevrim sayılarında (MCS) YAK sonuç görüntüleri

En iyi sonuç görüntüsü 5000 çevrim sayısında elde edildiği için, diğer gri seviye resimlerde de 5000 çevrim sayısında kenar bulma işlemi yapılmıştır. RADIUS/DARPA-IU Fort Hood gri seviyeli test resimleri ve bu resimler için YAK ile kenar bulma sonuç resimleri Şekil 5.2'de verilmiştir (Bowyer, 1997).



(a1) Orijinal resim (46.pgm)



(b1) YAK ile kenar tespiti sonucu



(a2) Orijinal resim (36.pgm)



(b2) YAK ile kenar tespiti sonucu



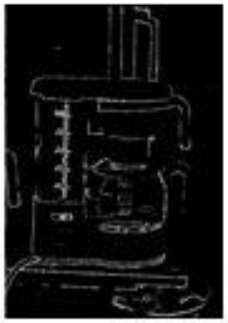
(a3) Orijinal resim (207.pgm)



(b3) YAK ile kenar tespiti sonucu



(a4) Orijinal resim (105.pgm)



(b4) YAK ile kenar tespiti sonucu



(a5) Orijinal resim (106.pgm)



(b5) YAK ile kenar tespiti sonucu

Şekil 5.2 Gri seviye resimleri üzerinde YAK kenar bulma uygulamasının sonuçları (nesne resimleri)

RADIUS/DARPA-IU Fort Hood veritabanından alınan 5 adet gri seviye nesne görüntüsü için farklı çevrim sayılarında YAK ile kenar bulma işleminde elde edilen işlem süreleri Tablo 5.1'de verilmiştir.

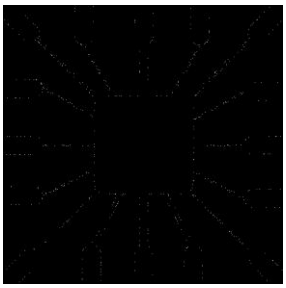
Tablo 5.1: RADIUS/DARPA-IU Fort Hood nesne görüntüleri için farklı MCS için YAK çalışma süreleri

	36.pgm	115.pgm	207.pgm	46.pgm	106.pgm
MCS =500	1,8931 sn	2,2447 sn	2,0685 sn	2,0101 sn	1,9159 sn
MCS =1000	3,6782 sn	4,4005 sn	4,1403 sn	3,8991 sn	3,8907 sn
MCS =2000	7,56 sn	9,0679 sn	8,1809 sn	7,794 sn	7,7016 sn
MCS =5000	19,4457 sn	23,4549 sn	21,6806 sn	19,9766 sn	20,5069 sn

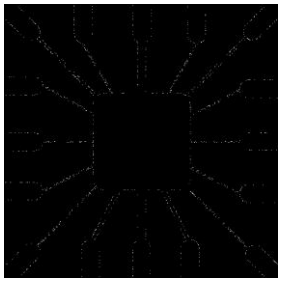
(MCS: Maksimum Çevrim Sayısı)

RADIUS/DARPA-IU Fort Hood veri tabanından alınan nesne görüntülerinin farklı çevrim sayılarında test edilmesi sonucunda elde edilen çalışma süreleri Tablo 5.1'de verilmiştir. Çevrim sayısı arttıkça çalışma süreleri de buna bağlı olarak artmaktadır. Ayrıca test görüntülerinin boyutları da birbirinden farklı olduğundan YAK kolonisinin büyüklüğü ve işlem yapan birey sayısının da değişken olabilmesinin sonucu olarak çalışma süreleri farklılık gösterebilmektedir. Yapılan çalışma, bir optimizasyon olduğundan optimum sonucun, en kısa sürede elde edilmesi de çalışma açısından önem teşkil etmektedir. Gri seviye nesne görüntüleri için çevrim sayısı 500, 1000, 2000 ve 5000 iken, 36.pgm görüntüsünde en kısa çalışma süresi elde edilmiştir.

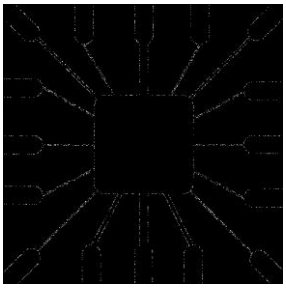
Gonzales'ten (Gonzales ve Woods, 2007) alınan 6 adet test görüntüsü üzerinde yapılan kenar bulma işleminde de yine 500'den 5000'e kadar farklı çevrim sayılarında, YAK algoritması ile kenar bulma işlemleri yapılmıştır. En iyi sonuç görüntüleri Maksimum Çevrim Sayısı (MCS) 5000 çevrim sayısında elde edilmiştir. Farklı çevrim sayılarında elde edilen sonuç görüntülerine örnek Şekil 5.3'de verilmiştir.



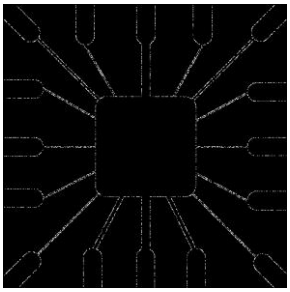
(a) MCS = 500



(b) MCS = 1000



(c) MCS = 2000



(d) MCS = 5000

Şekil 5.3. Fig0905(a)(wirebond-mask) için farklı MCS için YAK sonuç görüntüleri

En iyi sonuç görüntüsü 5000 çevrim sayısında elde edildiği için, Gonzales'ten alınan diğer resimlerde de 5000 çevrim sayısında kenar bulma işlemi yapılmıştır. Bu resimler ve bu resimler için YAK ile kenar bulma sonuç resimleri Şekil 5.4'de verilmiştir. Gonzales'ten alınan resimler üzerinde, farklı çevrim sayılarında yapılan kenar bulma işleminde elde edilen işlem süreleri Tablo 5.2'de verilmiştir.



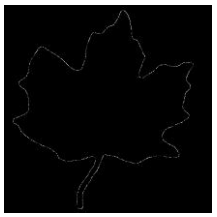
(a1) Orjinal Penny



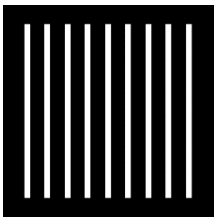
(b1) YAK sonucu



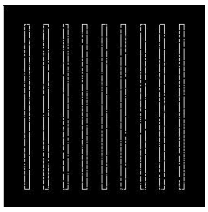
(a2) Orjinal Leaf



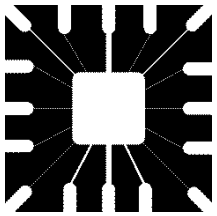
(b2) YAK sonucu



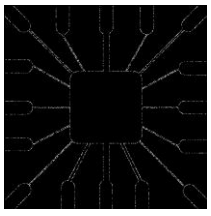
(a3) Orjinal Filter



(b3) YAK sonucu



(a4) Orjinal Wire



(b4) YAK sonucu



(a5) Orjinal Region (b5) YAK sonucu (a6) Orjinal Threshold (b6) YAK sonucu

(a1), (a2), (a3), (a4), (a5), (a6) : Orjinal görüntüler
(b1), (b2), (b3), (b4), (b5), (b6) geliştirilen metodun sonuç görüntüleri (YAK)
Şekil 5.4. YAK kenar bulma uygulamasının sonuçları

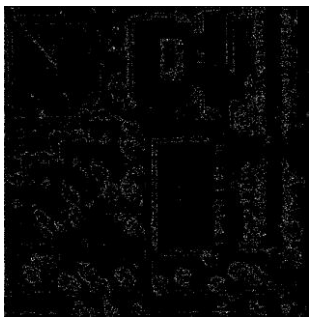
Tablo 5.2. Gonzales'ten alınan test resimleri için farklı MCS için YAK çalışma süreleri

	Penny	Leaf	Filter	Wire	Region	Threshold
MCS =500	0.6628 sn	0.7034 sn	1.2331 sn	1.5084 sn	1.4922 sn	1.5464 sn
MCS =1000	1.2523 sn	1.2796 sec	2.4397 sn	2.9910 sn	3.0246 sn	3.0755 sn
MCS =2000	2.2587 sn	2.4795 sec	4.8883 sn	5.9005 sn	6.0248 sn	6.1897 sn
MCS =5000	5.5673 sn	6.6609 sn	12.1797 sn	14.6908 sn	14.9603 sn	15.3021 sn

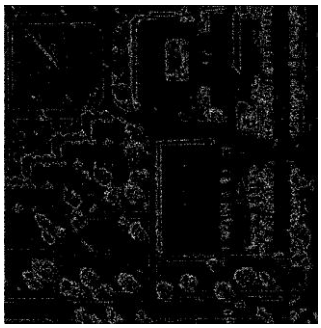
(MCS: Maksimum Çevrim Sayısı)

Gonzales'ten alınan siyah-beyaz görüntülerin farklı çevrim sayılarında test edilmesi sonucunda elde edilen çalışma süreleri Tablo 5.2'de verilmiştir. Çevrim sayısı arttıkça çalışma süreleri de buna bağlı olarak artmaktadır. Ayrıca test görüntülerinin boyutları da birbirinden farklı olduğundan YAK kolonisinin büyüklüğü ve işlem yapan birey sayısının da değişken olabilmesinin sonucu olarak çalışma süreleri farklılık gösterebilmektedir. Yapılan çalışma, bir optimizasyon olduğundan optimum sonucun, en kısa sürede elde edilmesi de çalışma açısından önem teşkil etmektedir. Çevrim sayısı sayısı 500, 1000, 2000 ve 5000 iken Penny görüntüsünde en kısa çalışma süresi elde edilmiştir.

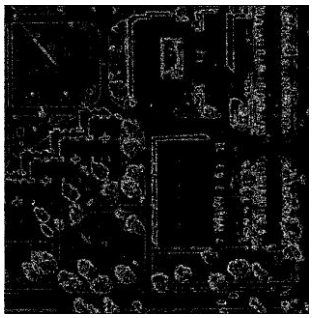
Yapılan çalışmada son olarak RADIUS/DARPA-IU Fort Hood veri tabanından alınan 10 adet gri seviye hava fotoğrafına ait görüntüler üzerinde de, YAK ile kenar bulma işlemleri yapılmıştır. Kenar bulma işleminde de yine 500'den 5000'e kadar farklı çevrim sayılarında, YAK algoritması ile kenar bulma işlemleri yapılmıştır. En iyi sonuç görüntüleri Maksimum Çevrim Sayısı (MCS) 5000 çevrim sayısında elde edilmiştir. Farklı çevrim sayılarında elde edilen sonuç görüntülerine örnek Şekil 5.5'de verilmiştir.



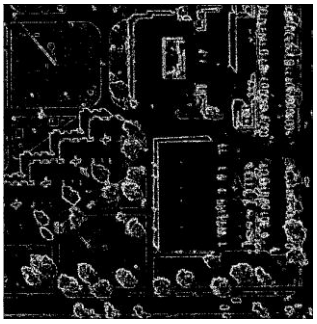
(a) MCS = 500



(b) MCS = 1000



(c) MCS = 2000



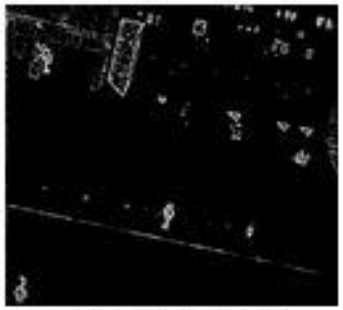
(d) MCS = 5000

Şekil 5.5. Buildings.pgm için farklı MCS için YAK sonuç görüntüleri

En iyi sonuç görüntüsü 50000 çevrim sayısında elde edildiği için, RADIUS/DARPA-IU Fort Hood gri seviye hava fotoğraflarında da 50000 çevrim sayısında kenar bulma işlemi yapılmıştır. Bu resimler ve bu resimler için YAK ile kenar bulma sonuç resimleri Şekil 5.6'de verilmiştir. Gri seviye hava fotoğrafları üzerinde, farklı çevrim sayılarında yapılan kenar bulma işleminde elde edilen işlem süreleri Tablo 5.3'de verilmiştir.



(a1) Orjinal Airfield



(b1) YAK sonucu



(a2) Orjinal Baseball



(b2) YAK sonucu



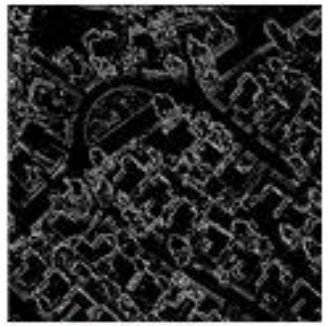
(a3) Orjinal Buildings



(b3) YAK sonucu



(a4) Orjinal Homes



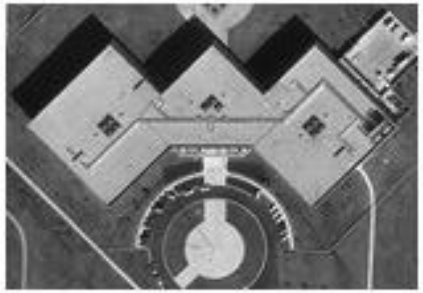
(b4) YAK sonucu



(a5) Orjinal Largebuilding



(b5) YAK sonucu



(a6) Orjinal Mainbuilding



(b6) YAK sonucu



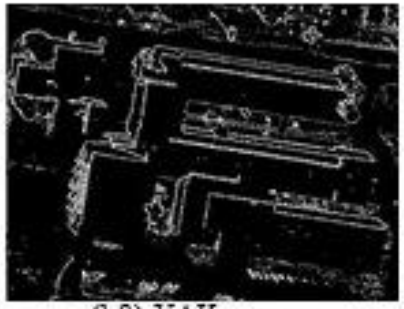
(a7) Orjinal Pool_tennis



(b7) YAK sonucu



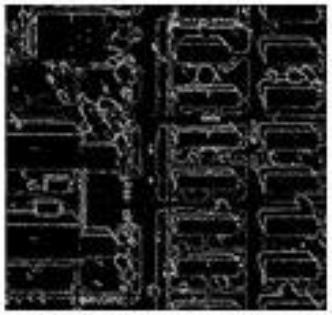
(a8) Orjinal School



(b8) YAK sonucu



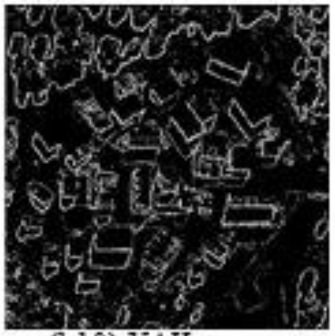
(a9) Orjinal Series



(b9) YAK sonucu



(a10) Orjinal Woods



(b10) YAK sonucu

(a1), (a2), (a3), (a4), (a5), (a6), (a7), (a8), (a9), (a10): Orjinal görüntüler
(b1), (b2), (b3), (b4), (b5), (b6) , (b7), (b8), (b9), (b10) geliştirilen metodun sonuç görüntüleri (YAK)

Şekil 5.6. YAK kenar bulma uygulamasının sonuçları (hava resimleri)

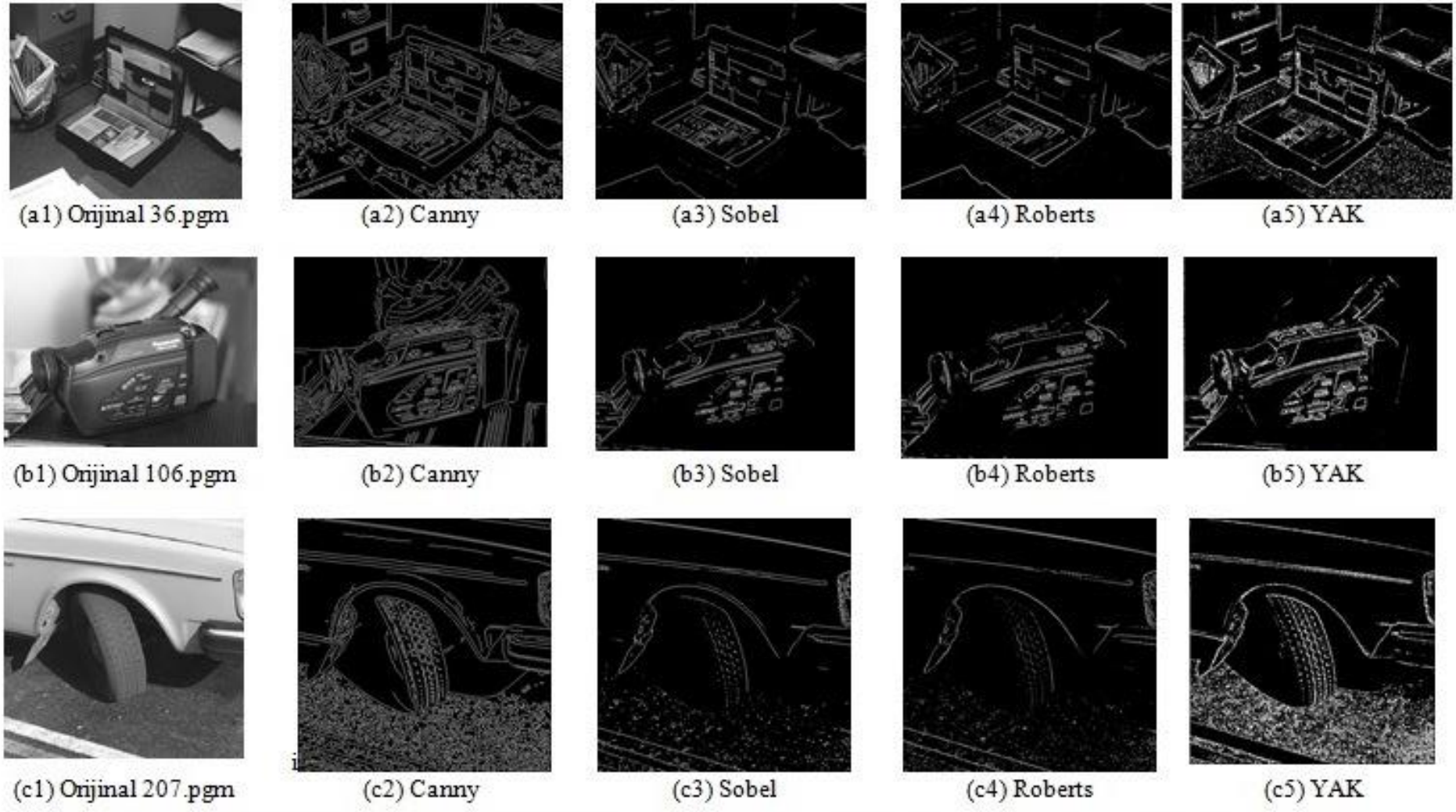
Tablo 5.3. RADIUS/DARPA-IU Fort Hood hava görüntüleri için farklı MCS için YAK çalışma süreleri

	MCS = 500	MCS = 1000	MCS = 2000	MCS = 5000
Airfield.pgm	7.7980 sn	13.4088 sn	15.5987 sn	40.7852 sn
Baseball.pgm	5.2223 sn	10.6755 sn	26.1252 sn	40.1774 sn
Buildings.pgm	5.0082 sn	12.8795 sn	20.7892 sn	36.4024 sn
Homes.pgm	6.7658 sn	14.3119 sn	24.0111 sn	54.1682 sn
Largebuilding.pgm	8.5662 sn	14.6113 sn	24.0915 sn	33.2375 sn
Mainbuilding.pgm	6.0354 sn	12.9617 sn	19.2325 sn	52.5555 sn
Pool_tennis.pgm	7.4204 sn	12.8962 sn	19.1993 sn	58.2725 sn
School.pgm	6.7865 sn	9.8385 sn	22.1863 sn	47.9266 sn
Series.pgm	5.0448 sn	10.8320 sn	17.6782 sn	41.8780 sn
Woods.pgm	5.8150 sn	11.6933 sn	21.4146 sn	42.0459 sn

(MCS: Maksimum Çevrim Sayısı)

RADIUS/DARPA-IU Fort Hood veri tabanından alınan hava görüntülerinin farklı çevrim sayılarında test edilmesi sonucunda elde edilen çalışma süreleri Tablo 5.3'te verilmiştir. Çevrim sayısı arttıkça çalışma süreleri de buna bağlı olarak artmaktadır. Test görüntülerinin boyutları da birbirinden farklıdır. Bu nedenle YAK kolonisinin büyüklüğü ve işlem yapan birey sayısının da değişken olabilmesinin sonucu olarak çalışma süreleri farklılık gösterebilmektedir. Yapılan çalışma, bir optimizasyon olduğundan optimum sonucun, en kısa sürede elde edilmesi de çalışma açısından önemlidir. Gri seviye hava görüntüleri için çevrim sayısı 500 iken baseball.pgm, 1000 iken school.pgm, 2000 iken airfield.pgm ve 5000 iken largebuilding.pgm görüntülerinde en kısa çalışma süresi elde edilmiştir.

Kenar bulma işlemlerinden elde edilen sonuç görüntüleri, literatürde bulunan farklı kenar bulma metotları ile karşılaştırılmıştır. Sonuç görüntülerinin analizi için sonuçların aralarındaki benzerlik oranını gösteren "Hamming Uzaklığı" yöntemi kullanılmıştır. RADIUS/DARPA-IU Fort Hood veri tabanından (Bowyer, 1997) alınan 5 adet gri seviye test görüntülerinin YAK algoritmasıyla kenar tespiti yönteminden çıkan sonuç görüntüleri ile Canny, Sobel ve Roberts kenar bulma algoritmalarından çıkan sonuç görüntüleri Şekil 5.7'de ve bu resimlerin sonuçlarına ait Hamming Uzaklıkları (HU) Tablo 5.4'de gösterilmiştir.



Şekil 5.7–a RADIUS/DARPA-IU Fort Hood nesne resimlerinde Canny, Sobel, Roberts, YAK kenar tespit sonuçları

Şekil 5.7 (devam)



(d1) Orijinal 105.pgm



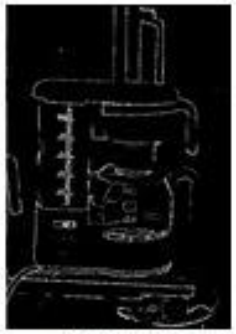
(d2) Canny



(d3) Sobel



(d4) Roberts



(d5) YAK



(e1) Orijinal 46.pgm



(e2) Canny



(e3) Sobel



(e4) Roberts



(e5) YAK

Şekil 5.7 –b RADIUS/DARPA-IU Fort Hood gri seviye nesne resimlerinde Canny, Sobel, Roberts, YAK kenar tespit sonuçları

Tablo 5.4. RADIUS/DARPA-IU Fort Hood nesne görüntüleri için HU sonuçları (MCS=5000)

		Canny	Sobel	Roberts	YAK
46.pgm	Canny	0	0.2318	0.2591	0.2775
	Sobel	0.2318	0	0.1554	0.2191
	Roberts	0.2591	0.1554	0	0.2256
	YAK	0.2775	0.2191	0.2256	0
207.pgm	Canny	0	0.3079	0.3170	0.3040
	Sobel	0.3079	0	0.1680	0.2767
	Roberts	0.3170	0.1680	0	0.2788
	YAK	0.3040	0.2767	0.2788	0
36.pgm	Canny	0	0.3070	0.3237	0.3385
	Sobel	0.3070	0	0.1393	0.3065
	Roberts	0.3237	0.1393	0	0.3101
	YAK	0.3385	0.3065	0.3101	0
106.pgm	Canny	0	0.2636	0.2743	0.2810
	Sobel	0.2636	0	0.1058	0.1480
	Roberts	0.2743	0.1058	0	0.1493
	YAK	0.2810	0.1480	0.1493	0
105.pgm	Canny	0	0.2145	0.2301	0.2467
	Sobel	0.2145	0	0.0905	0.1356
	Roberts	0.2301	0.0905	0	0.1399
	YAK	0.2467	0.1356	0.1399	0

Tablo 5.4'te RADIUS/DARPA-IU Fort Hood veri tabanından alınan görüntüler üzerinde geliştirilen YAK yöntemi ile bilinen diğer kenar bulma yöntemleri (Sobel, Canny, Roberts) arasındaki Hamming Uzaklıkları sonuçları verilmiştir. Elde edilen sonuçlara göre YAK kenar bulma algoritması ile en fazla benzerliğin Sobel yöntemi ile olduğu görülmektedir. Hamming Uzaklığı sonuçlarından da görüleceği gibi, gri seviye görüntüler için geliştirilen yöntem,

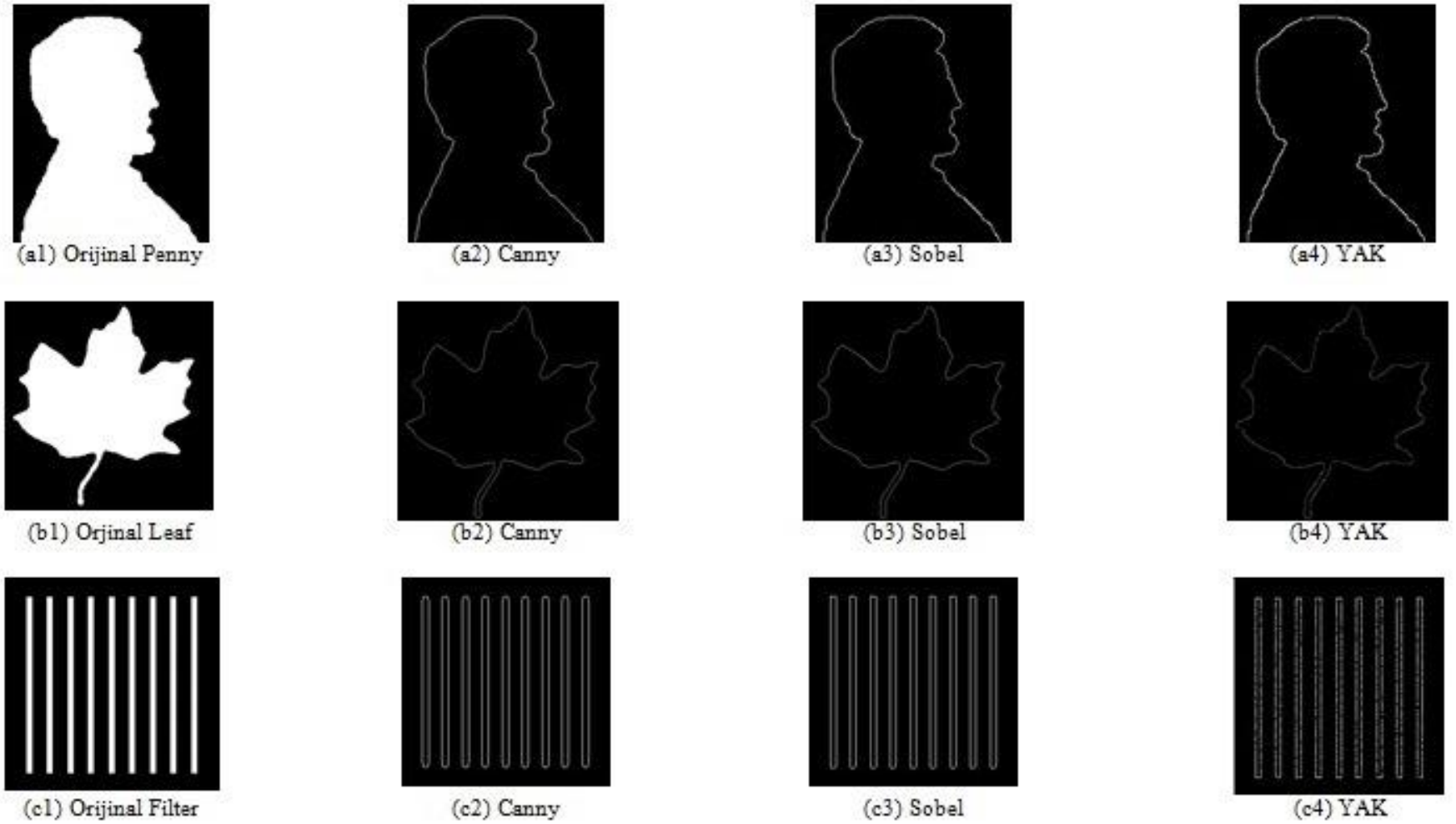
Sobel kenar bulma operatörünün sonuç görüntülerine benzer sonuçlar üretmektedir. Gri seviye görüntüler için, geliştirilen yöntem litaratürde bulunan Sobel operatörüne alternatif bir yaklaşım olarak verilebilmektedir.

Gonzales'ten (Gonzales ve Woods, 2007) alınan görüntülerin YAK algoritmasıyla kenar tespiti yönteminden çıkan sonuç görüntüleri ile Canny ve Sobel kenar bulma algoritmalarından çıkan sonuç görüntüleri Şekil 5.8'de ve bu resimlerin sonuçlarına ait Hamming Uzaklıkları (HU), Hata Tespiti (HT) ve Benzerlik (B) sonuçları da Tablo 5.5'de verilmiştir.

Tablo 5.5. Gonzales'e ait görüntüler için HU, HT ve Benzerlik sonuçları (MCS= 50000)

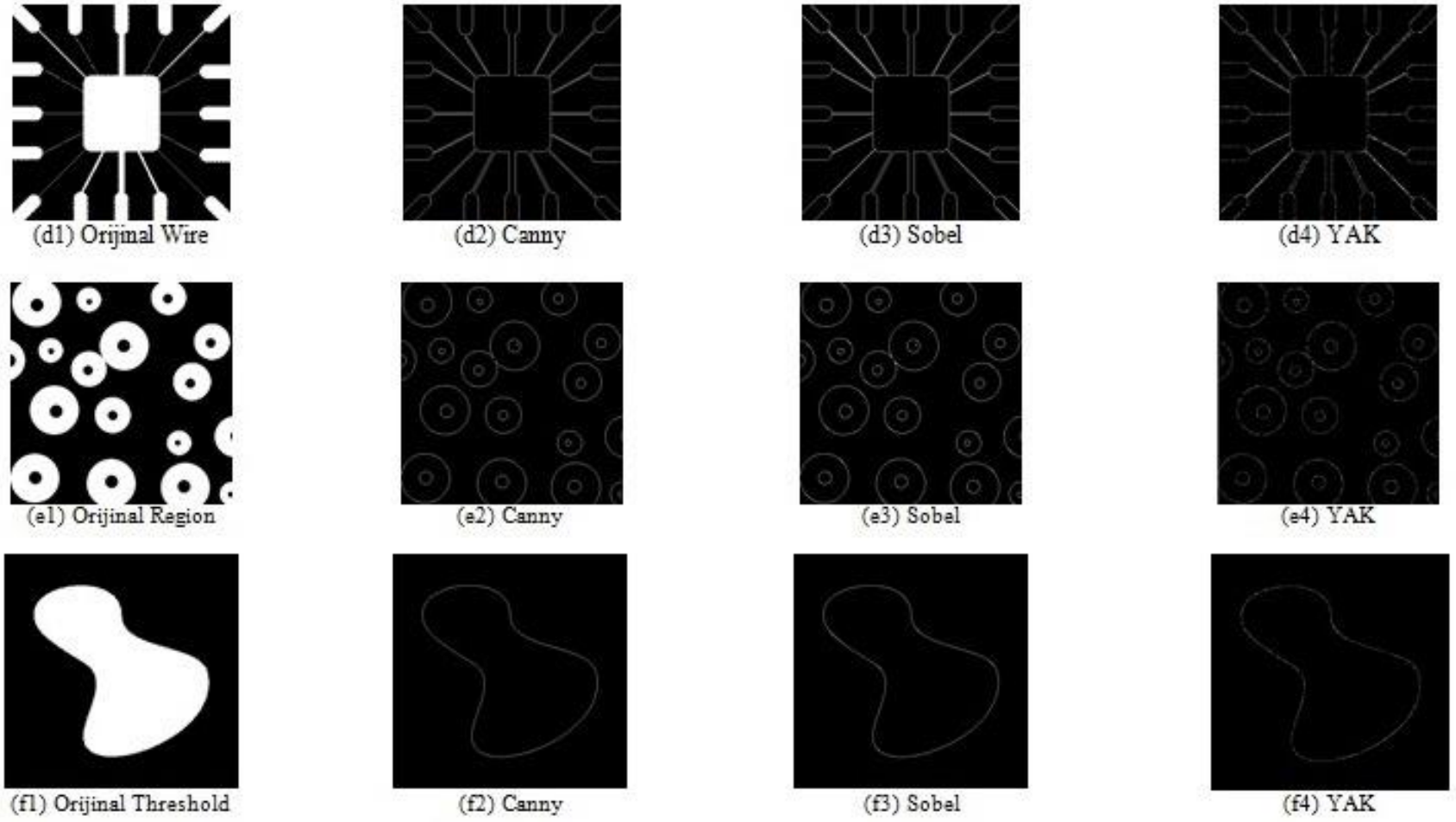
		Hamming Uzaklığı		Hata Tespiti		Benzerlik	
		Canny	Sobel	Canny	Sobel	Canny	Sobel
Penny	YAK	0.0587	0.0566	0.64	0.45	0.63	0.75
Filter	YAK	0.1893	0.1235	0.52	0.50	0.59	0.74
Wire	YAK	0.0964	0.086	0.73	0.52	0.58	0.72
Region	YAK	0.1070	0.1018	0.69	0.57	0.58	0.68
Threshold	YAK	0.0220	0.0217	0.58	0.51	0.64	0.71
Leaf	YAK	0.0352	0.0334	0.75	0.60	0.55	0.66

Siyah-beyaz görüntüler üzerinde YAK algoritması, Canny ve Sobel kenar bulma yöntemleri uygulanmıştır. Bu yöntemlerin sonuç görüntüleri arasındaki Hamming Uzaklığı, Hata Tespiti ve Benzerlik sonuçları Tablo 5.5'te gösterilmiştir. Tüm siyah-beyaz resimlerde, geliştirilen yöntem ile Sobel kenar bulma yöntemi arasındaki Hamming Uzaklığı ve Hata Tespiti sonuçları, Canny sonuçlarına göre daha küçük bulunmuştur. Ayrıca Benzerlik sonuçlarında da Sobel ile benzerliğin fazla olduğu görülmektedir. Sonuçlardan da görüleceği gibi, geliştirilen yöntem, siyah-beyaz görüntülerden kenar bulma uygulamalarında, Sobel kenar bulma yöntemine bir alternatif olarak verilebilmektedir.



Şekil 5.8 –a Gonzales'in siyah-beyaz resimlerinde Canny, Sobel, YAK kenar tespit sonuçları

Şekil 5.8 (devam)



Şekil 5.8 –b Gonzales'in siyah-beyaz resimlerinde Canny, Sobel, YAK kenar tespit sonuçları

RADIUS/DARPA-IU Fort Hood veri tabanından alınan 10 adet gri seviye hava fotoğrafına ait görüntülerin YAK algoritmasıyla kenar tespiti yönteminden çıkan sonuç görüntüleri ile kenarları verilmiş siyah-beyaz orijinal sonuç görüntüleri Şekil 5.9'da ve bu resimlerin sonuçlarına ait Hamming Uzaklığı, Hata Tespiti ve Benzerlik sonuçları da Tablo 5.6'da verilmiştir.

Tablo 5.6. RADIUS/DARPA-IU Fort Hood hava görüntüleri için HU, HT ve Benzerlik sonuçları (MCS= 50000)

		Hamming Uzaklığı	Hata Tespiti	Benzerlik
Airfield	YAK	0.0656	0.86	0.85
Baseball	YAK	0.1406	0.76	0.63
Buildings	YAK	0.1506	0.59	0.69
Homes	YAK	0.1988	0.64	0.68
Large building	YAK	0.1547	0.59	0.73
Mainbuilding	YAK	0.1086	0.54	0.73
Pool tennis	YAK	0.1365	0.63	0.66
School	YAK	0.1335	0.66	0.74
Series	YAK	0.1605	0.63	0.70
Woods	YAK	0.1561	0.68	0.64

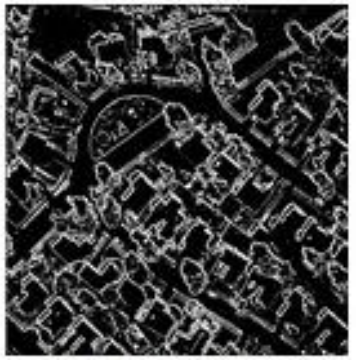
Tablo 5.6, sonuçlara göre en iyi Hamming Uzaklığı değeri Airfield görüntüsünde, en az Hata Tespiti değeri Mainbuilding görüntüsünde, en fazla benzerlik oranı ise yine Airfield görüntüsünde elde edilmiştir.



(d1) Homes



(d2) Homes (Orijinal kenarlar)



(d3) Homes (YAK)



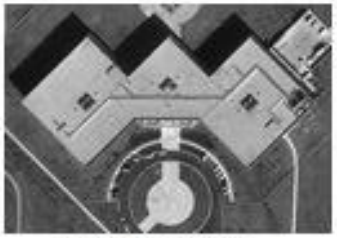
(e1) Largebuilding



(e2) Largebuilding (Orijinal kenarlar)



(e3) Largebuilding (YAK)



(f1) Mainbuilding



(f2) Mainbuilding (Orijinal kenarlar)



(f3) Mainbuilding (YAK)



(g1) Pool_tennis



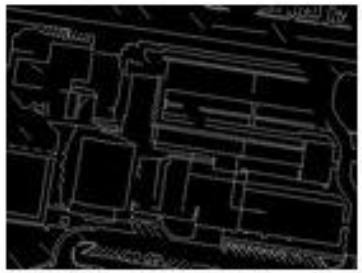
(g2) Pool_tennis (Orijinal kenarlar)



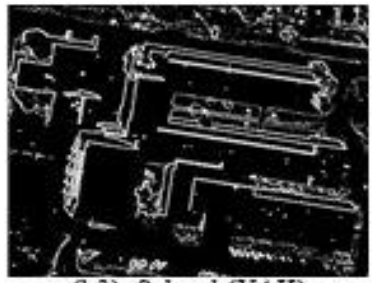
(g3) Pool_tennis (YAK)



(h1) School



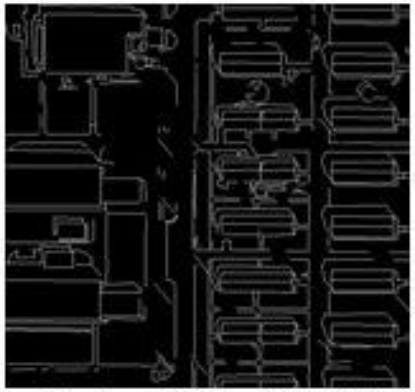
(h2) School (Orijinal kenarlar)



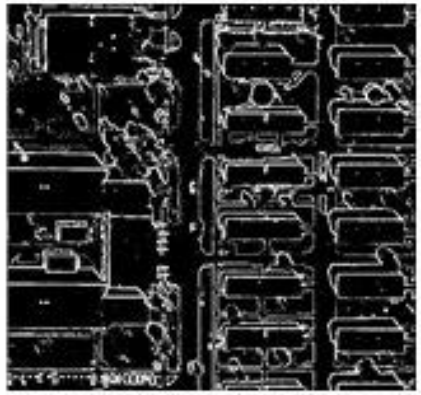
(h3) School (YAK)



(i1) Series



(i2) Series (Orijinal kenarlar)



(i3) Series (YAK)



(j1) Woods



(j2) Woods (Orijinal kenarlar)



(j3) Woods (YAK)

Şekil 5.9. RADIUS/DARPA-IU Fort Hood hava resimlerinde YAK kenar tespit sonuçları ve orijinal kenar bilgisi haritaları

RADIUS/DARPA-IU Fort Hood orijinal gri seviye hava fotoğraflarının YAK algoritmasıyla kenar tespiti yönteminden çıkan sonuç görüntüleri ile kenarları bulunmuş orijinal siyah-beyaz görüntüleri ile karşılaştırma yapılarak, geliştirilen algoritmanın literatürde daha önceden yapılan çalışmalarla da karşılaştırılması yapılmıştır. Karşılaştırma sonuçları için Hata Tespiti sonuçları Tablo 5.7'de ve Benzerlik sonuçları ise Tablo 5.8'da verilmiştir (Giannarou ve Stathaki, 2011).

Tablo 5.7. Hata Tespiti Tablosu

	Rothwell	Bergholm	Canny	Schunck	Lacroix	Deriche	ROC	Kappa r = 0.7	YAK
Airfield	0.4707	0.4643	0.5057	0.5147	0.5340	0.5276	0.4634	0.4126	0.86
Baseball	0.5710	0.5968	0.5668	0.5689	0.5865	0.5726	0.5505	0.5183	0.76
Buildings	0.5506	0.6047	0.5593	0.5640	0.5736	0.5500	0.5425	0.4885	0.59
Homes	0.5991	0.5703	0.5712	0.5743	0.4851	0.6302	0.5177	0.4402	0.64
Large building	0.5292	0.5970	0.5328	0.5459	0.5180	0.5194	0.5084	0.4440	0.59
Main building	0.5608	0.5651	0.6180	0.6130	0.5193	0.6677	0.5439	0.4561	0.54
Pool tennis	0.5273	0.5639	0.5086	0.5367	0.5396	0.4852	0.5029	0.4343	0.63
School	0.5638	0.5844	0.5571	0.5647	0.5701	0.5928	0.5444	0.5001	0.66
Series	0.4321	0.5297	0.4143	0.4276	0.4522	0.5303	0.4139	0.3619	0.63
Woods	0.5634	0.5846	0.5526	0.5695	0.5274	0.6856	0.5241	0.4602	0.68

Tablo 5.7'de en kötü Hata Tespiti sonuçları kırmızı ile gösterilmiş, en iyi olan sonuçlar yeşil ile gösterilmiş ve kabul edilebilir sonuçlar da mavi ile gösterilmiştir. Test görüntülerinin 6'sı en kötü sonucu verirken, 4'ünde ortalama sonuçlar elde edilmiştir. Buradaki oranların yüksek çıkmasının nedeni ise geliştirilen algoritmanın, orijinal kenarların gösterildiği görüntülerde belirtilmemiş olan çevresel faktörlere ait kenarlara duyarlılığının fazla olmasıdır.

Tablo 5.8. Benzerlik Tablosu

	Rothwell	Bergholm	Canny	Schunck	Lacroix	Deriche	ROC	Kappa r = 0.7	YAK
Airfield	0.7692	0.7651	0.7593	0.7382	0.7162	0.7516	0.7760	0.8492	0.85
Baseball	0.6815	0.6359	0.6961	0.6848	0.6579	0.6934	0.6995	0.7249	0.63
Buildings	0.6886	0.5895	0.6892	0.6804	0.6570	0.6871	0.6946	0.7355	0.69
Homes	0.6332	0.5978	0.6851	0.6762	0.7143	0.6169	0.7064	0.7599	0.68
Large building	0.6844	0.5401	0.6976	0.6730	0.6823	0.7082	0.7014	0.7506	0.73
Main building	0.6967	0.6191	0.6612	0.6499	0.6988	0.6088	0.6998	0.8357	0.73
Pool tennis	0.6722	0.5923	0.7208	0.6596	0.6527	0.7100	0.7046	0.7639	0.66
School	0.6495	0.5846	0.6743	0.6474	0.6419	0.6458	0.6669	0.7835	0.74
Series	0.7577	0.6084	0.7682	0.7656	0.7259	0.6810	0.7767	0.8090	0.70
Woods	0.6581	0.5912	0.6849	0.6748	0.6760	0.5667	0.6922	0.7448	0.64

Tablo 5.8’de en kötü benzerlik sonuçları kırmızı ile gösterilmiş, en iyi olan sonuçlar yeşil ile gösterilmiş ve kabul edilebilir sonuçlar da mavi ile gösterilmiştir. Airfield görüntüsünde en fazla benzerlik oranı, diğer görüntülerde ise %60’ın üzerinde kabul edilebilir değerler elde edilmiştir.

6. SONUÇ ve ÖNERİLER

Yapılan uygulamada, görüntüde kenar bulma işlemi, YAK optimizasyon algoritması kullanılarak gerçekleştirilmiştir. Kenar bulmada türeve dayalı olmayan ve herhangi bir maskeye ihtiyaç duymayan, optimizasyona dayalı bir yöntem geliştirilmiştir. Test görüntüleri olarak 5'i gri seviyeli nesne, 10'u gri seviyeli hava resimleri ve 6'sı siyah beyaz olan görüntü olmak üzere toplam 21 resim kullanılmıştır. Her bir görüntü üzerinde farklı çevrim sayılarında uygulama test edilmiştir ve çalışma süreleri de hesaplanmıştır.

Tablo 6.1: RADIUS/DARPA-IU Fort Hood nesne görüntüleri için farklı MCS için YAK çalışma süreleri

	36.pgm	115.pgm	207.pgm	46.pgm	106.pgm
MCS =500	1,8931 sn	2,2447 sn	2,0685 sn	2,0101 sn	1,9159 sn
MCS =1000	3,6782 sn	4,4005 sn	4,1403 sn	3,8991 sn	3,8907 sn
MCS =2000	7,56 sn	9,0679 sn	8,1809 sn	7,794 sn	7,7016 sn
MCS =5000	19,4457 sn	23,4549 sn	21,6806 sn	19,9766 sn	20,5069 sn

(MCS: Maksimum Çevrim Sayısı)

Tablo 6.2. Gonzales'ten alınan test resimleri için farklı MCS için YAK çalışma süreleri

	Penny	Leaf	Filter	Wire	Region	Threshold
MCS =500	0.6628 sn	0.7034 sn	1.2331 sn	1.5084 sn	1.4922 sn	1.5464 sn
MCS =1000	1.2523 sn	1.2796 sec	2.4397 sn	2.9910 sn	3.0246 sn	3.0755 sn
MCS =2000	2.2587 sn	2.4795 sec	4.8883 sn	5.9005 sn	6.0248 sn	6.1897 sn
MCS =5000	5.5673 sn	6.6609 sn	12.1797 sn	14.6908 sn	14.9603 sn	15.3021 sn

(MCS: Maksimum Çevrim Sayısı)

Tablo 6.3. RADIUS/DARPA-IU Fort Hood hava görüntüleri için farklı MCS için YAK çalışma süreleri

	MCS = 500	MCS = 1000	MCS = 2000	MCS = 5000
Airfield.pgm	7.7980 sn	13.4088 sn	15.5987 sn	40.7852 sn
Baseball.pgm	5.2223 sn	10.6755 sn	26.1252 sn	40.1774 sn
Buildings.pgm	5.0082 sn	12.8795 sn	20.7892 sn	36.4024 sn
Homes.pgm	6.7658 sn	14.3119 sn	24.0111 sn	54.1682 sn
Largebuilding.pgm	8.5662 sn	14.6113 sn	24.0915 sn	33.2375 sn
Mainbuilding.pgm	6.0354 sn	12.9617 sn	19.2325 sn	52.5555 sn
Pool_tennis.pgm	7.4204 sn	12.8962 sn	19.1993 sn	58.2725 sn
School.pgm	6.7865 sn	9.8385 sn	22.1863 sn	47.9266 sn
Series.pgm	5.0448 sn	10.8320 sn	17.6782 sn	41.8780 sn
Woods.pgm	5.8150 sn	11.6933 sn	21.4146 sn	42.0459 sn

(MCS: Maksimum Çevrim Sayısı)

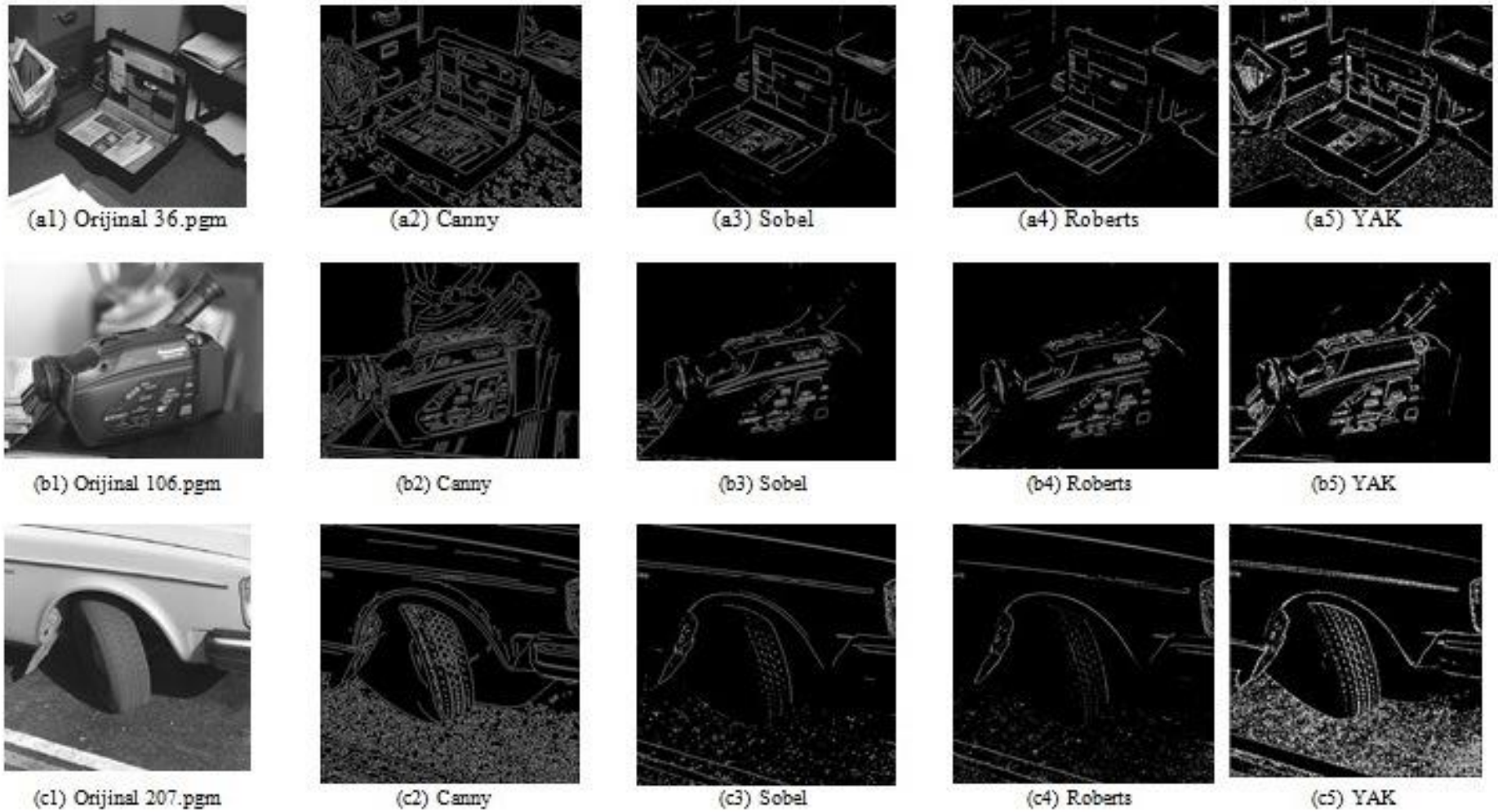
Tablo 6.1, RADIUS/DARPA-IU Fort Hood gri seviye nesne görüntülerinde, Tablo 6.2, Gonzales'ten alınan siyah beyaz görüntülerde ve Tablo 6.3, RADIUS/DARPA-IU Fort Hood hava görüntülerinde farklı çevrim saylarındaki çalışma sürelerini göstermektedir. Bu tablolara göre maksimum çevrim sayısı arttıkça çalışma süreleri de artmaktadır. Siyah beyaz görüntülerde çalışma sürelerinin, gri seviyeli görüntülerde yapılan test sonuçlarına göre daha kısa olduğu görülmüştür.

Yapılan tez çalışmasında, genel YAK optimizasyon algoritmasındaki kaynak konumlandırması rasgele bir şekilde giriş resmi pikselleri üzerinde gerçekleştirilmiştir. Temel YAK algoritmasında yeni kaynak bir formül ile belirlenirken, yapılan çalışmada ilk gidilen kaynağın 8'li komşuluğundaki pikseller arasından yeni kaynak belirlenmiştir.

Nektar kalitesi olarak piksellerin gri seviye değerlerinden yararlanılmıştır. Koloni büyüklüğü resim boyutlarına bağlı olarak değişiklik gösterirken, maksimum çevrim sayısı 500, 1000, 2000, 5000…,50000 şeklinde farklı sayılar test edilmiştir. Limit değeri ise 5 olarak belirlenmiştir. Eşik değer ise her resmin standart sapmasına bağlı olarak belirlenmiştir (Denklem 6.1). Böylelikle görüntülerin eşik değerleri, her görüntünün gri seviye dağılımına göre farklılık göstermektedir. Bu nedenle tüm test görüntüleri için kenar bulmada aynı eşik değerin kullanılmaması her resim için daha doğru sonuç elde edilmesini sağlamıştır. Eşik değerin standart sapmaya bağlı olarak belirlenmesi ile her görüntü için kullanılan sabit bir eşik değeri belirlemesinin getireceği dezavantajlar yok edilmiştir.

$$esik\ deger = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2} \qquad (6.1)$$

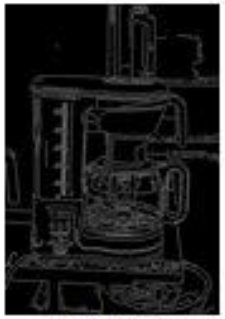
Yapılan çalışmanın sonuç resimleri, literatürde bilinen ve kullanılan yöntemlerin sonuçları ile karşılaştırılmıştır. Kenar bulma işleminde önceden tanımlanan bir maskeye ihtiyaç duyan bu yöntemlerde maskeleme işlemleri tüm pikseller üzerinde yapılamamaktadır. Bu maskenin resim üzerindeki tüm pikseller üzerinde dolaşması gerekmektedir ancak kenar ve köşe pikseller bu metotlarda göz ardı edilmektedir. Önemsiz sayılan bu pikseller bazı durumlarda da sıfır olarak alınmaktadır. Bu tür bilgi kayıpları olmadan, hiçbir pikseli yok saymadan ve tüm pikselleri dolaşmadan resimden kenar bilgisi çıkarımı YAK ile gerçekleştirilmiştir ve maskeye olan bağımlılık kaldırılmıştır. Ayrıca diğer yöntemlere göre YAK kenar bulma yöntemi daha fazla ayrıntıyı gösterebilme yeteneğine de sahiptir (Şekil 6.1).



Şekil 6.1–a RADIUS/DARPA-IU Fort Hood nesne resimlerinde Canny, Sobel, Roberts, YAK kenar tespit sonuçları

Şekil 6.1 (devam)


(d1) Orijinal 105.pgm


(d2) Canny


(d3) Sobel


(d4) Roberts


(d5) YAK


(e1) Orijinal 46.pgm


(e2) Canny


(e3) Sobel


(e4) Roberts


(e5) YAK

Şekil 6.1–b RADIUS/DARPA-IU Fort Hood nesne resimlerinde Canny, Sobel, Roberts, YAK kenar tespit sonuçları

RADIUS/DARPA-IU Fort Hood (Bowyer K. W., 1997) hava görüntülerindeki karşılaştırmalarda, Tablo 6.4'de görüldüğü gibi, analiz sonuçlarına göre, çevresel elemanların da YAK algoritması tarafından bulunuyor olması nedeniyle, Hata Tespiti istenilen düzeyde bulunamamıştır. Şekil 6.2'de kırmızıyla gösterilen alanlar, çevresel elemanlara örnek olarak verilebilir. Ancak ilgili alanın çıkarılması ya da hibrit yöntemlerle birleştirilerek, YAK'ın kullanılması ileriki çalışmalarda daha başarılı sonuçlar verebilecektir. Bunun yanında Tablo 6.5'deki sonuçlardan da görüleceği gibi benzerlik oranı referans çalışmalarla aynı, hatta daha iyi olduğu sonuçlar da elde edilmiştir.

Tablo 6.4. Hata Tespiti Tablosu

	Rothwell	Bergholm	Canny	Schunck	Lacroix	Deriche	ROC	Kappa r = 0.7	YAK
Airfield	0.4707	0.4643	0.5057	0.5147	0.5340	0.5276	0.4634	0.4126	0.86
Baseball	0.5710	0.5968	0.5668	0.5689	0.5865	0.5726	0.5505	0.5183	0.76
Buildings	0.5506	0.6047	0.5593	0.5640	0.5736	0.5500	0.5425	0.4885	0.59
Homes	0.5991	0.5703	0.5712	0.5743	0.4851	0.6302	0.5177	0.4402	0.64
Large building	0.5292	0.5970	0.5328	0.5459	0.5180	0.5194	0.5084	0.4440	0.59
Main building	0.5608	0.5651	0.6180	0.6130	0.5193	0.6677	0.5439	0.4561	0.54
Pool tennis	0.5273	0.5639	0.5086	0.5367	0.5396	0.4852	0.5029	0.4343	0.63
School	0.5638	0.5844	0.5571	0.5647	0.5701	0.5928	0.5444	0.5001	0.66
Series	0.4321	0.5297	0.4143	0.4276	0.4522	0.5303	0.4139	0.3619	0.63
Woods	0.5634	0.5846	0.5526	0.5695	0.5274	0.6856	0.5241	0.4602	0.68

Tablo 6.5. Benzerlik Tablosu

	Rothwell	Bergholm	Canny	Schunck	Lacroix	Deriche	ROC	Kappa r = 0.7	YAK
Airfield	0.7692	0.7651	0.7593	0.7382	0.7162	0.7516	0.7760	0.8492	0.85
Baseball	0.6815	0.6359	0.6961	0.6848	0.6579	0.6934	0.6995	0.7249	0.63
Buildings	0.6886	0.5895	0.6892	0.6804	0.6570	0.6871	0.6946	0.7355	0.69
Homes	0.6332	0.5978	0.6851	0.6762	0.7143	0.6169	0.7064	0.7599	0.68
Large building	0.6844	0.5401	0.6976	0.6730	0.6823	0.7082	0.7014	0.7506	0.73
Main building	0.6967	0.6191	0.6612	0.6499	0.6988	0.6088	0.6998	0.8357	0.73
Pool tennis	0.6722	0.5923	0.7208	0.6596	0.6527	0.7100	0.7046	0.7639	0.66
School	0.6495	0.5846	0.6743	0.6474	0.6419	0.6458	0.6669	0.7835	0.74
Series	0.7577	0.6084	0.7682	0.7656	0.7259	0.6810	0.7767	0.8090	0.70
Woods	0.6581	0.5912	0.6849	0.6748	0.6760	0.5667	0.6922	0.7448	0.64



Şekil 6.2. Çevresel elemanlar

Yapılan karşılaştırmalar, YAK algoritması ile kenar bulmanın bir alternatif olarak kullanılabileceğini göstermektedir. Bu konuda daha ileri bir çalışma olarak, YAK optimizasyonu ile kenar bulma yönteminin farklı yöntemlerle hibrit yapılar oluşturması ve parametrelerinin geliştirilerek, süre kavramının daha da kısaltılması amaçlanabilir. Ayrıca hiçbir önişlem adımı olmadan geliştirilen bu yöntemle, farklı gürültü giderici filtrelerin birleştirilmesi konusunda da çalışmalar yapılabilir.

KAYNAKLAR

A. Adelkhani, B. Beheshti, S.Minaei, P. Javadikia, M.Ghasemi-Varnamkhasti, 2013, "Taste characterization of orange using image processing combined with ANFIS", Measurement, Volume 46, Issue 9, syf. 3573-3580.

Akay B., 2009 (a) "Nümerik Optimizasyon Problemlerinde Yapay Arı Kolonisi Algoritmasının Performans Analizi", Fen Bilimleri Enstitüsü, Erciyes Üniversitesi,325 syf.

Akay B., Karaboğa D., 2009, (b) "Parameter Tuning for the Artificial Bee Colony Algorithm" Computational Collective Intelligence. Semantic Web, Social Networks and Multiagent Systems Lecture Notes in Computer Science , syf. 608-619.

Akay B. ,2013, "A study on particle swarm optimization and artificial bee colony algorithms for multilevel thresholding", Applied Soft Computing, syf. 3066–3091.

Akça M.R., 2011, "Yapay arı kolonisi algoritması kullanarak gezgin satıcı probleminin Türkiyedeki il ve ilçe merkezlerine uygulanması", Selçuk Üniversitesi , Fen Bilimleri Enstitüsü, Elektronik ve Bilgisayar Sistemleri Eğitimi Anabilim Dalı, 98 syf.

Akhan N., 2004, "Bilgisayar destekli iris tanıma", Selçuk Üniversitesi, Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği Anabilim Dalı, 85 syf.

Akyol S., Alataş B., 2012, "Güncel Sürü Zekâsı Optimizasyon Algoritmaları", Nevşehir Üniversitesi Fen Bilimleri Enstitü Dergisi 1, syf. 36-50.

Alataş B., 2007, "Kaotik Haritalı Parçacık Sürü Optimizasyon Algoritmaları Geliştirme", Doktora Tezi, Fırat Üniversitesi, Fen Bilimleri Enstitüsü.

Altınışık M., 2004, *İnternet kaynaklarına dayanarak* [online], http://www.mustafaaltinisik.org.uk/34-adsem-01.ppt [Ziyaret Tarihi : 10 Haziran 2014] .

Arı İ., 2008, *İnternet kaynaklarına dayanarak* [online], http://ismailari.com/blog/matlab-ile-imge-islemeye-giris/ [Ziyaret Tarihi : 14 Haziran 2014] .

Arpat G.B., 2001, "Developing a feature-based edge detector using artificial neural networks", Orta Doğu Teknik Üniversitesi , Enformatik Enstitüsü, 175 syf.

Arsuaga R. M., 2011, "Multi-Objective Artificial Bee Colony for scheduling in Grid environments ", Swarm Intelligence (SIS), 2011 IEEE Symposium, syf. 1-7.

Avcı G., 2011, "Gerçek zamanlı uygulamalar için ABC algoritmasının FPGA üzerinde gerçeklenmesi", Niğde Üniversitesi , Fen Bilimleri Enstitüsü, Elektrik-Elektronik Mühendisliği Anabilim Dalı, 73 syf.

Aybar E., 2008, " Sobel İşleci Kullanılarak Renkli Görüntülerde Kenar Bulma", Afyon Kocatepe Üniversitesi, Fen Bilimleri Dergisi, Cilt VIII, Sayı 1, syf. 205-217.

Aydoğmuş Ö., 2011, "2d to 3d video conversion", İstanbul Teknik Üniversitesi , Fen Bilimleri Enstitüsü , Elektronik ve Haberleşme Mühendisliği Anabilim Dalı , Telekomünikasyon Mühendisliği Bilim Dalı, 77 syf.

Bal H., 2006, "Kamera ile görüntü işleme teknikleriyle malzeme tane büyüklüğü analizi", Gazi Üniversitesi, Fen Bilimleri Enstitüsü, Makine Mühendisliği Anabilim Dalı, 253 syf.

Barth F. G. 1982, "Insects and Flowers: The Biology of a Partnership", Princeton, N.J.: Princeton University Press.

Bastı M., 2012, "Kuruluş yeri seçimi probleminin çözümünde meta sezgisel algoritmalar", İstanbul Üniversitesi , Sosyal Bilimler Enstitüsü , İşletme Anabilim Dalı, Üretim Bilim Dalı, 226 syf.

Basturk, B., Karaboga, D., 2006 ,"An Artificial Bee Colony (ABC) Algorithm for Numeric function Optimization", IEEE Swarm Intelligence Symposium Indianapolis, Indiana, USA.

Basu M., 2013,"Artificial bee colony optimization for multi-area economic dispatch", Electrical Power and Energy Systems, syf. 181–187.

Basu M., 2011,"Bee colony optimization for combined heat and power economic dispatch", Expert Systems with Applications, syf. 13527–13531.

Baykan N.,2010, "Robotik bir mikroskop sisteminden elde edilen görüntülerin görüntü işleme ve yapay zeka yöntemleri ile analizi", Selçuk Üniversitesi, Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği Anabilim Dalı, 151 syf.

Benedek C., Descombes X. ve Zerubia J., 2012, "Building Development Monitoring in Multitemporal Remotely Sensed Image Pairs with Stochastic Birth-Death Dynamics", IEEE Transactions On Pattern Analysis And Machine Intelligence, Vol. 34, No. 1, syf. 33-49.

Bovik A., 2010, “Handbook of Image anf Video Processing”,Academic Press, 1384 syf.

Bowyer K.W., 1997, *İnternet kaynaklarına dayanarak* [online], http://marathon.csee.usf.edu/edge/edge_detection.html [Ziyaret Tarihi : 1 Haziran 2013] .

Canny, J F., 1986, “A Computational Approach to Edge Detection”, IEEE Transactions on Pattern Analysis and Machine Intelligence, syf. 679-698.

Chao L., 2010, “An adaptive morphological edge detection algorithm based on image fusion”, Image and Signal Processing (CISP), 2010 3rd International Congress, syf. 1072- 1076.

Changsheng Z., Dantong O., Jiaxu N., 2010, “An artificial bee colony approach for clustering”, Expert Systems with Applications, syf. 4761–4767.

Comer D. E., 2008, “Computer Networks and Internets (5th Edition)”, 768 syf.

Çalış G., 2009, “Şantiye yerleşim problemi çözümünde karınca koloni optimizasyonunun kullanımı”, Ege Üniversitesi, Fen Bilimleri Enstitüsü, İnşaat Mühendisliği Anabilim Dalı, 113 syf.

Çevik K. K., 2010, “Yapay zeka yöntemleri ile araç plaka tanıma sistemi”, Selçuk Üniversitesi, Fen Bilimleri Enstitüsü, Elektronik ve Bilgisayar Sistemleri Eğitimi Anabilim Dalı, 75 syf.

Çınar M., 2011, “Görüntü işleme teknikleri ve klinik bulgular kullanılarak mesane geometrisinin idrar akım hızına etkisinin belirlenmesi”, Ege Üniversitesi, Fen Bilimleri Enstitüsü, Nükleer Bilimler Anabilim Dalı, 99 syf.

Çiftçi S., 2011, “Improving Edge Detection Using İntersection Consistency”, Orta Doğu Teknik Üniversitesi , Fen Bilimleri Enstitüsü , Bilgisayar Mühendisliği Bölümü, 57 syf.

Das S., Mirnalinee T. T. Ve Varghese K., 2011, “Use of Salient Features for the Design of a Multistage Framework to Extract Roads From High-Resolution Multispectral Satellite Images”, IEEE Transactions On Geoscience And Remote Sensing, Vol. 49, NO. 10, syf. 3906-3931.

Doğan A., 2011, “Enerji sistemlerinde en uygun yük akışının sağlanabilmesi için optimizasyon algoritmalarının kullanılması”, Erciyes Üniversitesi, Fen Bilimleri Enstitüsü , Elektrik-Elektronik Mühendisliği Anabilim Dalı, 119 syf.

Ekizler H., 2011, “Araç rotalama probleminin çözümünde karınca kolonisi optimizasyonu algoritmasının kullanılması”, İstanbul Üniversitesi , Fen Bilimleri Enstitüsü, Endüstri Mühendisliği Anabilim Dalı, 162 syf.

Eklund A., Dufort P., Forsberg D., LaConte S. M., 2013, "Medical image processing on the GPU – Past, present and future", Medical Image Analysis 17 (2013), syf. 1073-1094.

Erol V., 2006, "Araç rotalama problemleri için populasyon ve komşuluk tabanlı metasezgisel bir algoritmanın tasarımı ve uygulaması", Yıldız Teknik Üniversitesi, Fen Bilimleri Enstitüsü, 172 syf.

Giannarou S. ve Stathaki T., 2011, "Optimal edge detection using multiple operators for image understanding", EURASIP Journal on Advances in Signal Processing, 19 syf.

Gonzalez R. C., Woods R. E., 2007, "Digital Image Processing Using MATLAB (3rd Edition)", 976 syf.

Gonzales R. C. ve Woods R. E., 2007, *İnternet kaynaklarına dayanarak* [online], http://www.imageprocessingplace.com/root_files_V3/image_databases.htm [Ziyaret Tarihi : 11 Mart 2014] .

Görenler A., 2012, "Depolama sistemlerinde sipariş toplama işlemlerinin genetik algoritmalarla optimizasyonu", İşletme İktisadi Enstitüsü Yönetim Dergisi, Sayı 71, syf. 118-144.

Gu Y., 2013, **"** Automated delineation of lung tumors from CT images using a single click ensemble segmentation approach", Pattern Recognition , Volume 46 Issue 3, syf. 692-702.

Haralick, R.M., 1984," Digital step edges from zero crossing of second directional derivatives", IEEE Trans. Pattern Anal. Mach. Intell syf. 58–68.

Hazer M., 2007, "Fuzzy topology based edge detection algorithm", Anadolu Üniversitesi , Fen Bilimleri Enstitüsü , Elektrik-Elektronik Mühendisliği Anabilim Dalı, 61 syf.

Heath M., Sarkar S., Sanocki T., and Bowyer K.W., 1997, "A Robust Visual Method for Assessing the Relative Performance of Edge-Detection Algorithms" IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 19, No. 12, syf. 1338-1359.

Horozoğlu E., 2013, "Görüntü işleme ile yüzey pürüzlülüğü ölçümü ve analizi", Selçuk Üniversitesi, Fen Bilimleri Enstitüsü, Elektronik ve Bilgisayar Sistemleri Eğitimi Anabilim Dalı, 72 syf.

Hsieh T. J., Yeh W.C., 2011, "Knowledge Discovery Employing Grid Scheme Least Squares Support Vector Machines Based on Orthogonal Design Bee Colony Algorithm", IEEE Transactions On Systems, Man, And Cybernetics—Part B: Cybernetics, VOL. 41, NO. 5, syf 1198-1212.

Jane O., 2010, "Görüntü Pekiştirmede Kullanılan Uyarlanır Bulanık Maskeleme Tekniği İçin En Uygun Süzgeç Ve Parametre Seçimi Analizi", Elektronik Mühendisliği Anabilim Dalı, Ankara Üniversitesi Fen Bilimleri Enstitüsü, 109 syf..

Karaboga D., Basturk B., 2008, "On The Performance Of Artificial Bee Colony (ABC) Algorithm, Applied Soft Computing", Volume 8, Issue 1, syf. 687-697.

Karaboğa D., 2011 "Yapay Zeka Optimizasyon Algoritmaları", Nobel Akademik Yayıncılık, 231syf.

Karaboga D., Ozturk C. , 2011, "A novel clustering approach: Artificial Bee Colony (ABC) algorithm", Applied Soft Computing, syf. 652–657.

Karlıkaya C., Erdoğan S., Akkoçlu A., Oktay G., Güner G., Uçan E. S., Çımrın A. H., 2003, "Akciğer kanserinde çoklu tümör belirleyicisi analizi", Türk Toraks Dergisi, Cilt 4, Sayı 3, syf. 248-259.

Kefeli T., 1998, "Kontrast farkına göre konum bulma", Kocaeli Üniversitesi, Fen Bilimleri Enstitüsü, 66 syf.

Koçer H.E., 2007, "İris deseninin yapay zeka yöntemleri ile tanınması", Selçuk Üniversitesi, Fen Bilimleri Enstitüsü, Elektrik-Elektronik Mühendisliği Anabilim Dalı, 149 syf.

Lazzaro P. D., Murra D., Schwortz B., 2013, "Pattern recognition after image processing of low-contrast images, the case of the Shroud of Turin", Pattern Recognition 46 (2013), syf. 1964-1970.

Li Y., Gong H., Feng D. ve Zhang Y., 2011, "An Adaptive Method of Speckle Reduction and Feature Enhancement for SAR Images Based on Curvelet Transform and Particle Swarm Optimization", IEEE Transactions On Geoscience And Remote Sensing, Vol. 49, No. 8, syf. 3105-3116.

Kulaksız A., 2010, "Görüntü işleme destekli paralel robot kontrolü",: Marmara Üniversitesi, Fen Bilimleri Enstitüsü, Mekatronik Mühendisliği Anabilim Dalı,123 syf.

Ma M.,, Liang J., Guo M., Fan Y., Yin Y., 2011, "SAR image segmentation based on Artificial Bee Colony algorithm", Applied Soft Computing syf . 5205–5214.

Marr, D., Hildreth E., 1980" Theory of edge detection", Proc. R. Soc. Lond. A, Math. Phys. Sci., B 207, syf. 187–217.

Millonas,M., 1994, "Artificial Life III, Chapter Swarms, Phase Transitions, And Collective Intelligence", Reading, MA:Addison-Wesley.

Mohammad A., Wang L., Gao R. X., 2013, "Integrated image processing and path planning for robotic sketching", 8th CIRP Conference on Intelligent Computation in Manufacturing Engineering, syf. 199-204.

Murty, K. G., 2003, "Optimization Models For Decision Making, vol. 1, Internet Edition, Chapter 1: Models for Decision Making", 1-18 syf.

Nezamabadi-pour H., Saryazdi S., Rashedi E., 2006, "Edge detection using ant algorithms ", Soft Comput 10: 623–628 syf.

Odçıkın Z.,1995, "Fetal kafa görüntüleri için görüntü işleme sistemi tasarım ve gerçekleştirimi", Hacettepe Üniversitesi , Fen Bilimleri Enstitüsü, 115 syf.

Ontiveros S., Jimenez R., Chitchian S., Tosello G., Carmignato S., 2013, "A 3D edge detection technique for surface extraction in computed tomography for dimensional metrology applications**",** C I R P Annals, Vol. 62, No. 1, syf. 531-534.

Onat M., 2008, "Dijital görüntü işleme yöntemleriyle lifli beton numunelerindeki çatlakların tespit edilmesi", Fırat Üniversitesi, Fen Bilimleri Enstitüsü, İnşaat Mühendisliği Bölümü , İnşaat Mühendisliği Anabilim Dalı, 54 syf.

Özcan T., 2011, "Perakende endüstrisinde raf alanı yönetimine veri madenciliği esaslı analitik bir yaklaşım", İstanbul Üniversitesi, Fen Bilimleri Enstitüsü, Endüstri Mühendisliği Anabilim Dalı ,Endüstri Mühendisliği Bilim Dalı,177 syf.

Özbek U., 2006, "Diş radyoraflarının bilgisayar tabanlı görüntü işleme teknikleriyle incelenmesi ve YSA kullanılarak sınıflandırılması", Selçuk Üniversitesi , Fen Bilimleri Enstitüsü , Elektrik-Elektronik Mühendisliği Anabilim Dalı, 90 syf.

Öztürk C., 2011, "Yapay sinir ağlarının yapay arı kolonisi algoritması ile eğitilmesi", Erciyes Üniversitesi, Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği Anabilim Dalı, 205 syf.

Pacurib J. A., Seno G. M. M., 2009, "Solving Sudoku Puzzles using Improved Artificial Bee Colony Algorithm", Fourth International Conference on Innovative Computing, Information and Control, syf . 885-888.

Polat D., 2010, "Turbo Kod Analizi ve FPGA Uygulaması", Başkent Üniversitesi, Fen Bilimleri Enstitüsü, Elektrik Elektronik Mühendisliği Anabilim Dalı, 205 syf.

Segundo M. P., Silva L., Bellon O. R. P., Sarkar S., 2013, "Orthogonal projection images for 3D face detection", Pattern Recognition Letters.

Sharcanski, J., Venetsanopoulos A. N., 1997 "Edge Detection of Color Images Using Directional Operators", IEEE Trans. Or Circuit and Systems for Video Technology, Vol.7, No.2, syf. 397 – 401.

Sharma T. K., Pant M., 2011, "Enhancing the Food Locations in an Artificial Bee Colony Algorithm", Soft Computing, Volume 17, Issue 10, syf. 1939-1965.

Shin M. C., Goldgof D. B., Bowyer K. W., Nikiforou S., 2001, "Comparison of Edge Detection Algorithms Using a Structure from Motion Task", Computer vision and image understanding, syf. 160-178.

Teodorovic´ D., Šelmic´ M., Mijatovic´-Teodorovic´ L., 2013 ,"Combining case-based reasoning with Bee Colony Optimization for dose planning in well differentiated thyroid cancer treatment", Expert Systems with Applications syf. 2147–2155.

Tereshko V., 2000 , "Reaction-Diffusion Model of a Honey Bee Colony's Foraging Behaviour, 6th International Conferenceon Parallel Problem Solving from Nature", 3-540-41056-2, London, UK, syf. 807-816

Tokmak M., 2011, "Yapay arı kolonisi algoritması ile ders çizelgeleme probleminin çözümü", Süleyman Demirel Üniversitesi, Fen Bilimleri Enstitüsü , Elektronik-Bilgisayar Eğitimi Anabilim Dalı, 62 syf.

Tosun Ö.,2012, "Yapay arı kolonisi algoritması ve permütasyon akış tipi çizelgeleme problemine uygulanması", Akdeniz Üniversitesi, Sosyal Bilimler Enstitüsü ,İşletme Anabilim Dalı, 111 syf.

Tolluoğlu A. O., Kargın S., Uçan O. N., Kent S., 2005, "Hücresel Yapay Sinir Ağları İle Kızılötesi Görüntülerde Gürültünün TemizlenmesiHavacılık Ve Uzay Teknolojileri Dergisi, Cilt 2 Sayı 1, syf. 77-86.

Tsoy S., 2011, "Gürültülü görüntülerin akıllı bir yöntem ile onarımı", Kocaeli Üniversitesi , Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği Anabilim Dalı, 61 syf.

Velioğlu H.M., 2010, "Görüntü işleme teknolojisi ve yapay sinir ağlarının gıda ürünlerinde bazı kalite özelliklerinin belirlenmesinde kullanılması", Namık Kemal Üniversitesi , Fen Bilimleri Enstitüsü, Gıda Mühendisliği Anabilim Dalı,90 syf.

Wang W., Yang L. J., Xie Y. T., An Y. W., 2013, "Edge detection of infrared image with CNN_DGA algorithm", Optik - International Journal for Light and Electron Optics, Volume 125, Issue 1, syf 464-467.

Xu, C., Duan H.,2010, "Artificial bee colony (ABC) optimized edge potential function (EPF) approach to target recognition for low-altitude aircraft", Pattern Recognition Letters, syf. 1759–1772.

Yücenur G.N., 2011, "Optimizasyon problemlerinin çözümünde melez metasezgisel bir algoritmanın tasarımı", Yıldız Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Endüstri Mühendisliği Anabilim Dalı, Endüstri Mühendisliği Bilim Dalı, 214 syf.

Zareizadeh Z., Hasanzadeh R. P. R., Baghersalimi G., 2013, "A recursive color image edge detection method using Green's function approach", Optik - International Journal for Light and Electron Optics, Volume 124, Issue 21, syf. 4847-4854.

Zhu H., Zhuang X., Meng X. Z., 2007, "Emergent Behavior of Ant Colony System in Digital Images for Acquiring Edge Information", Information Reuse and Integration, syf. 116-121.

Zhuang X., 2004 "Edge Feature Extraction in Digital Images with the Ant Colony System", IEEE lntemational Conference an Computational lntclligence for Measurement Systems and Applicatians Boston, MA, USA, syf. 133-136.

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı : ELİF DENİZ YİĞİTBAŞI
Uyruğu : T.C.
Doğum Yeri ve Tarihi : İZMİR / 11.06.1988
Telefon : 0507 – 722 11 51
Faks : -
deniz.yigitbasi@isikun.edu.tr
e-mail : elifdeniz.yigitbasi@hotmail.com

EĞİTİM

Derece	Adı, İlçe, İl	Bitirme Yılı
Lise :	Muğla Anadolu Lisesi	**2006**
Üniversite :	Selçuk Üniversitesi Bilgisayar Mühendisliği	**2011**
Yüksek Lisans :	Selçuk Üniversitesi Bilgisayar Mühendisliği	**Devam Ediyor.**
Doktora :	-	

İŞ DENEYİMLERİ

Yıl	Kurum	Görevi
2011-2012	FAndFps Private Solutions	Bilgisayar Mühendisi
2012-(devam ediyor)	IŞIK ÜNİVERSİTESİ	Araştırma Görevlisi

UZMANLIK ALANI

Görüntü İşleme ve Meta sezgisel Görüntü İşleme

YABANCI DİLLER

İngilizce: İleri Seviye

Almanca: Temel Seviye