

PAMUKKALE ÜNİVERSİTESİ FEN BİLİMLERİ ENSTİTÜSÜ

**GÖRÜNTÜ İŞLEME TEKNİKLERİ VE YAPAY ZEKÂ YÖNTEMLERİ
KULLANARAK GÖRÜNTÜ İÇİNDE GÖRÜNTÜ ARAMA**

**YÜKSEK LİSANS TEZİ
Mehmet KARAKOÇ**

Anabilim Dalı : Bilgisayar Mühendisliği A.B.D.

Programı : Tezli Yüksek Lisans

Tez Danışmanı: Yrd. Doç. Dr. Kadir KAVAKLIOĞLU

HAZİRAN 2011

YÜKSEK LİSANS TEZ ONAY FORMU

Pamukkale Üniversitesi Fen Bilimleri Enstitüsü 091281005 nolu öğrencisi Mehmet KARAKOÇ tarafından hazırlanan “GÖRÜNTÜ İŞLEME TEKNİKLERİ VE YAPAY ZEKÂ YÖNTEMLERİ KULLANARAK GÖRÜNTÜ İÇİNDE GÖRÜNTÜ ARAMA” başlıklı tez tarafımızdan okunmuş, kapsamı ve niteliği açısından bir Yüksek Lisans tezi olarak kabul edilmiştir.

Tez Danışmanı : 
(Jüri Başkanı) Yrd. Doç. Dr. Kadir KAVAKLIOĞLU (PAÜ)

Jüri Üyesi : 
Doç. Dr. Sezai TOKAT (PAÜ)

Jüri Üyesi : 
Doç. Dr. Abdullah Tahsin TOLA (PAÜ)

Pamukkale Üniversitesi Fen Bilimleri Enstitüsü Yönetim Kurulu'nun 03.09.2011 tarih ve 21/20.... sayılı kararıyla onaylanmıştır.


Fen Bilimleri Enstitüsü Müdürü
Prof. Dr. Nuri KOLSUZ

Bu tezin tasarımı, hazırlanması, yürütülmesi, arařtırmalarının yapılması ve bulgularının analizlerinde bilimsel etięe ve akademik kurallara özenle riayet edildiđini; bu çalıřmanın doğrudan birincil ürünü olmayan bulguların, verilerin ve materyallerin bilimsel etięe uygun olarak kaynak gösterildiđini ve alıntı yapılan çalıřmalara atfedildiđine beyan ederim.

İmza

Öğrenci Adı Soyadı : Mehmet Karakoç



ÖNSÖZ

Bu çalışmada, görüntü içinde görüntü arama problemi üzerinde durulmuştur. Bu amaçla, görüntü arama için genetik algoritmalar ve görüntü eşleme için yapay sinir ağları ile çeşitli fonksiyonlar kullanılmıştır. Görüntü aramanın optimizasyonu ve görüntü eşlemenin verimli bir şekilde yapılabilmesi için akıllı algoritmalara ve paralel programlama tekniklerine dayanan zeki bir yöntem önerilmiştir. Daha sonra, görüntüler üzerinde kaba kuvvet ve akıllı yolla gerçekleştirilen tek ve çok iş parçacıklı arama sonuçları karşılaştırılmıştır. Ayrıca uygulama süreci çeşitli yöntemlerle geliştirilmiştir. Son olarak, arama algoritmalarının bilgisayar ortamına aktarılan ve bilgisayar ortamında hazırlanan görüntüler ile vereceği sonuçlar ve çıkarımlar ortaya konarak gerekli başarımlar değerlendirilmiştir. Bu çalışmanın gerçekleştirilmesinde katkıda bulunan danışman hocam Yrd. Doç. Dr. Kadir KAVAKLIOĞLU'na, değerli hocalarıma, arkadaşlarıma ve aileme teşekkür ederim.

Haziran 2011

Mehmet Karakoç

(Bilgisayar Mühendisi)

İÇİNDEKİLER

Sayfa

ÖZET.....	xii
SUMMARY.....	xiv
1. GİRİŞ	1
1.1 Tezin Amacı.....	3
1.2 Literatür Özeti	4
1.3 Hipotez.....	11
1.4 Tezin Katkıları	15
2. GÖRÜNTÜ İŞLEME.....	18
2.1 Görüntü Nedir	20
2.2 Görüntü Yakalama	21
2.3 Sayısal Görüntü.....	21
2.4 Sayısal Görüntü Nitelikleri	22
2.5 Görüntü İşleme Adımları.....	24
2.6 Görüntü İşleme Kullanım Alanları.....	24
2.7 Görüntü İşlemede Kullanılan Temel Yöntemler.....	25
2.8 Görüntü İşleme Teknikleri.....	25
2.9 Görüntü İşleme Teknikleri Kullanılarak Yapılabilecek Çalışmalar.....	27
2.10 Görüntü Formatları.....	27
2.11 Görüntüde Alan Bulma.....	28
2.12 Görüntü İşleme Tekniklerinin Kullanımı	29
2.13 Görüntü Veri Tabanı Düzenlemesi.....	29
3. YAPAY ZEKÂ	30
3.1 Yapay Sinir Ağları.....	31
3.1.1 Yapay sinir ağı hücresi	31
3.1.2 Yapay sinir ağı türleri	32
3.1.3 Yapay sinir ağı modelleri.....	33
3.1.4 Yapay sinir ağlarında öğrenme algoritmaları.....	36
3.1.5 Geri yayımlı yapay sinir ağları.....	37
3.1.6 Yapay sinir ağlarının eğitilmesi	38
3.1.7 Bir yapay sinir ağının çalışma prosedürü	39
3.1.8 Yapay sinir ağlarının kullanım alanları	39
3.1.9 Yapay sinir ağları kullanılarak yapılabilecek çalışmalar	40
3.1.10 Yapay sinir ağlarının kullanımı.....	40
3.2 Genetik Algoritmalar	40
3.2.1 Genetik algoritmaların diğer yöntemlerden farkları.....	44
3.2.2 Genetik algoritmaların temel teoremi.....	44
3.2.3 Yeniden üretim.....	46
3.2.4 Genetik algoritma işlemcileri.....	47
3.2.5 Uygunluk fonksiyonu	49
3.2.6 Seçim	50
3.2.7 Genetik algoritmaların kullanım alanları.....	50

3.2.8 Genetik algoritmaların kullanımı	52
3.2.9 Genetik algoritmaların uygulandığı bir problem.....	53
4. GÖRÜNTÜ İÇİNDE GÖRÜNTÜ ARAMA.....	54
4.1 Büyük Bir Görüntüde Küçük Bir Görüntü Aramak	55
4.2 Kaba Kuvvet Arama	57
4.3 Akıllı Arama	58
4.4 Paralel Arama.....	61
4.5 Uygunluk Fonksiyonları ve Benzerlik Ölçümleri	62
4.5.1 Yapay sinir ağları	63
4.5.2 Ortalama renk değeri	68
4.5.3 Renk değerlerinin standart sapması.....	68
4.5.4 Korelasyon	69
4.5.5 Kenar tespiti	69
4.5.6 Görüntü işleme kütüphanesi fonksiyonları	70
4.6 Görüntü İşleme Filtreleme Seçenekleri	70
4.7 Tez Kapsamında Hazırlanan Yazılımın Diyagramları	71
5. YAZILIM VE ARAYÜZ	76
5.1 Yazılım Kapsamında C# Programlama Dili Kullanılması	76
5.2 Görüntü Arama Yazılımının Başlangıç Arayüzü	77
5.3 Görüntü Arama Yazılımının YSA Arayüzleri	81
5.4 Referans Görüntüden Alan Seçme Arayüzü	86
5.5 Uygulamadaki İlerlemeleri Gösteren Arayüzler	87
5.5.1 YSA eğitiminin ilerlemesini gösteren arayüz	87
5.5.2 Görüntü aramanın ilerlemesini gösteren arayüz.....	87
5.6 Görüntüleri Farklı Boyutlarda Karşılaştırmak	88
5.7 Görüntüleri Aramada Paralel Programlama Teknikleri Kullanmak.....	89
5.8 Kodlama Sürecindeki İyileştirmeler	90
5.9 YSA Kütüphanesinin Sınıf Hiyerarşisi.....	93
6. SONUÇ VE ÖNERİLER	97
6.1 Kaba Kuvvet Arama	99
6.1.1 Tek iş parçacığı ile arama	99
6.1.2 Çok iş parçacığı ile paralel olarak arama.....	99
6.2 Akıllı Arama	100
6.2.1 Tek iş parçacığı ile arama	100
6.2.2 Çok iş parçacığı ile paralel olarak arama.....	100
6.3 Görüntü Arama Sonuçları.....	100
6.4 Görüntü Arama Önerileri.....	101
6.5 Görüntü Eşleme Sonuçları	102
6.6 Görüntü Eşleme Önerileri	103
6.7 GİGA'nın Kullanımı ve Uygulanabilirliği	104
6.8 Uygulama Platformu Önerisi	104
6.9 Uygulama Sürecinde Karşılaşılan Zorluklar.....	105
6.9.1 Ön görüntü işleme teknikleri.....	105
6.9.2 GA için kromozomların kodlanması ve uygunluk	106
6.9.3 Görüntü aramalarının paralelleştirilmesi	108
6.9.3.1 Çok iş parçacığı paralellliği	108
6.9.3.2 Akıllı arama paralellliği	108
6.9.4 GİGA kapsamında geliştirilen algoritmalar	109
6.9.5 Görüntü eşleme için YSA kullanılması	110
6.10 Tez Çalışması Süreci ve Sonuçların Genel Değerlendirilmesi	111

KAYNAKLAR.....	116
EKLER.....	120

KISALTMALAR

CPU	: Central Processing Unit (Merkezi İşlem Birimi)
CT	: X-ray Computed Tomography (Bilgisayarlı Tomografi)
ÇOV	: Çoklu Ortam Verileri (Multimedia Datas)
GA	: Genetik Algoritmalar (Genetic Algorithms)
GB	: Gigabyte (10^9 Bayt (Uluslararası Birim Sistemi)) (2^{30} Bayt (ikili kullanım))
GHZ	: Gigahertz (10^9 hertz, GHz)
GİGA	: Görüntü İçinde Görüntü Arama (Image Searching Inside Image)
GİT	: Görüntü İşleme Teknikleri (Image Processing Techniques)
GY	: Geri Yayılımlı (Back Propagation)
İTGE	: İçerik Tabanlı Görüntü Erişim (Content Based Image Retrieval)
JPEG	: Joint Photographic Experts Group (Birleşik Fotoğraf Uzmanları Grubu)
MATLAB	: Matrix Laboratory (Matris Laboratuvarı)
MRI	: Magnetic Resonance Imaging (Manyetik Rezonans Görüntüleme)
NASA	: National Aeronautics and Space Administration (Ulusal Havacılık ve Uzay Dairesi)
NLM	: The United States National Library of Medicine (Amerika Birleşik Devletleri Ulusal Tıbbi Kütüphanesi)
NP	: Nondeterministic Polynomial (Üstel Zaman Belirsizliği)
ORD	: Ortalama Renk Değeri (Average Color Value)
RAM	: Random Access Memory (Rasgele Erişimli Bellek)
RDSS	: Renk Değerlerinin Standart Sapması (Standart Deviation of Color Values)
RGB	: Red-Green-Blue (Kırmızı-Yeşil-Mavi, K-Y-M)
UML	: Unified Modelling Language (Birleşik Modelleme Dili)
USGS	: United States Geological Survey (Amerikan Yerbilimsel Araştırma Kurumu)
X	: X-ray (X-ışınları, Röntgen Işınları)
YSA	: Yapay Sinir Ağları (Artificial Neural Networks)
YZ	: Yapay Zekâ (Artificial Intelligence)

TABLO LİSTESİ

Tablolar

4.1 : GİGA işlemleri ve yaklaşımları.....	71
4.2 : Görüntü eşleme için uygunluk fonksiyonları ve yaklaşımları.....	72
5.1 : Görüntü arama sonuçları.	78
A.1 : Görüntü arama için GA'nın parametre değerleri.	124
A.2 : Görüntü arama için GA'nın birey seçim yöntemleri.....	124
A.3 : GİGA için parametre değerleri.	125
B.1 : YSA'nın eğitimi için GY'nin parametre değerleri.....	125
B.2 : YSA'nın eğitimi için GA'nın parametre değerleri.....	125
C.1 : GİGA'nın örnek uygulamaları.....	141
D.1 : YSA ile dört farklı şekilde görüntü arama.	142
D.2 : ORD ile dört farklı şekilde görüntü arama.	142
D.3 : RDSS ile dört farklı şekilde görüntü arama.....	142
D.4 : Korelasyon ile dört farklı şekilde görüntü arama.	143
D.5 : Kenar tespiti ile dört farklı şekilde görüntü arama.	143

ŞEKİL LİSTESİ

Şekiller

1.1 : Tüm İTGE sistemlerinin temel bileşenleri (Müller ve diğ., 2004).	6
2.1 : Görüntü işleme sistem yapısı.	18
2.2 : Sayısal ortama geçirilen gürültülü ve gürültüleri giderilmiş görüntüler.	19
2.3 : Görüntüyü gürültüden arındırma örneği.	19
2.4 : (a)Dağınık piksel dağılımlı ve (b)histogram eşitlemeli görüntüler.	20
2.5 : Görüntü işlemede histogram yardımıyla resmin belirginleştirilmesi.	20
2.6 : Bir görüntüden bu görüntüdeki kan sayısı gibi veriler elde edilebilir.	20
2.7 : Görüntünün gözde oluşumu.	21
2.8 : Görüntü yakalama ve sayısallaştırma aşamaları.	21
2.9 : (a)Analog görüntü ve (b)sayısal karşılığı.	22
2.10 : (a)A harfi, (b)gösterimi ve (c)sayısallaştırılması.	22
2.11 : H harfi için sayısallaştırma ve S harfi için üç ayrı gösterim.	22
2.12 : Bir görüntü için piksellerin ifadesi.	23
2.13 : RGB renk uzayına göre renklerin oluşturulması.	23
2.14 : Aynı görüntünün piksellerinin farklı bit değerlerindeki durumları.	23
2.15 : Farklı çözünürlükteki görüntüler.	24
2.16 : Görüntü işleme adımlarının gösterimi.	24
2.17 : Görüntü işleme tekniklerinin dallandırılması.	26
2.18 : Görüntü eşleme yöntemlerinden görüntüde alan bulma örneği.	28
2.19 : Küçük a harfinin aranması örneği.	28
2.20 : Orijinal görüntü ve bulunan hedef nesnelere.	28
3.1 : Yapay sinir ağı birbirine bağlı düğümlerden oluşan bir gruptur.	31
3.2 : Genel bir yapay sinir ağı hücresi modeli ve ağırlıklı giriş toplamı.	32
3.3 : N boyutlu uzayın bir doğru ya da düzlem ile iki bölgeye ayrılması.	33
3.4 : Çok katmanlı sinir ağları örneği.	33
3.5 : Bir yapay sinir ağının genel yapısı.	35
3.6 : GY bir yapay sinir ağının genel yapısı.	38
3.7 : Arama yöntemleri içerisinde GA'nın yeri.	42
3.8 : Genel evrimsel bilgi işleme döngüsü.	43
3.9 : Çaprazlama noktası dört olan bir çaprazlama örneği.	47
3.10 : Mutasyon örneği.	48
4.1 : YSA ile görüntü eşleme sürecinin genel gösterimi.	64
4.2 : Şablon görüntü ve bu görüntünün YSA'ya verilecek hâli.	65
4.3 : GİGA yazılımının blok diyagramı.	71
4.4 : GİGA sürecini gösteren akış diyagramı.	72
4.5 : Görüntü eşleme sürecini gösteren akış diyagramı.	73
4.6 : GA ile akıllı arama sürecini gösteren akış diyagramı.	74
4.7 : Paralel arama ve çoklu boyutlandırma sürecini gösteren akış diyagramı. ..	75
5.1 : Görüntü arama ana formu birinci sekmesi.	77
5.2 : Görüntü arama ana formu ikinci sekmesi.	79
5.3 : Görüntü arama ana formu üçüncü sekmesi.	80

5.4 : YSA ana formu.	81
5.5 : YSA'nın GA ile eğitilmesi için kullanılan sekme.	82
5.6 : Yeni şablon görüntü taslağı belirleme formu.	82
5.7 : Örnek şablon taslakları.	83
5.8 : YSA'nın katmanlarını gösteren form.	84
5.9 : YSA'ya yeni katman ekleme formu.	84
5.10 : Seçili YSA katmanını gösteren form.	85
5.11 : Aktivasyon fonksiyonu seçim formu.	86
5.12 : Görüntüden alan seçme formu.	86
5.13 : YSA eğitimi sürecini gösteren form.	87
5.14 : Görüntü arama sürecini gösteren form.	87
5.15 : YSA kütüphanesindeki sınıflar.	94
5.16 : YSA kütüphanesinin sinir hücresi sınıfı.	94
5.17 : YSA kütüphanesinin katman sınıfı.	95
5.18 : YSA kütüphanesinin sinir ağı sınıfı.	95
5.19 : YSA kütüphanesinin GY öğrenme algoritması sınıfı.	95
6.1 : Referans görüntüde şablon görüntünün aranması örneğı.	98
6.2 : Şablon görüntünün kaba kuvvet aranması örneğı.	99
A.1 : Örnek görüntüler: (a)512x512. (b)400x400. (c)200x219. (d)350x256. (e)349x446. (f)512x512.	121
A.2 : Örnek görüntüler: (a)185x209. (b)412x270. (c)500x333. (d)179x187. (e)171x180. (f)163x180.	122
A.3 : Örnek görüntüler: (a)324x223. (b)512x384. (c)512x512. (d)180x180. (e)225x225. (f)640x480.	123
A.4 : Tek bir bireyin uygunluk orantılı seçim örneğı.	124
A.5 : Aktivasyon fonksiyonları: (a)Sigmoid. (b)Linear. (c)Heaviside. (d)Gaussian.	137
A.6 : Aktivasyon fonksiyonlarının tanımlamaları: (a)Sigmoid. (b)Linear. (c)Heaviside. (d)Gaussian.	138
A.7 : Referans görüntü parçaları: (a)512x512. (b)512x512. (c)200x219. (d)512x512.	139
A.8 : GİGA örnek uygulamaları.	140

SEMBOL LİSTESİ

$i; j$	İndis değerleri
$P; Q$	(p_1, p_2, \dots, p_n) ve (q_1, q_2, \dots, q_n) noktaları
X	Görüntünün yataydaki değeri
Y	Görüntünün düşeydeki değeri
x	Sinir hücresi için giriş değeri
w	İlgili giriş değerinin ağırlık değeri
f	YSA aktivasyon fonksiyonu
y	YSA çıkış değeri
e	YSA çıkışı için hata değeri
op	Genetik işlemci olasılığı (Çaprazlama veya mutasyon)
r	Rasgele üretilen değer
M	Referans görüntünün yatay boyutu
N	Referans görüntünün dikey boyutu
m	Şablon görüntünün yatay boyutu
n	Şablon görüntünün dikey boyutu
S	Referans görüntüdeki $m \times n$ boyutlarındaki toplam alt görüntü sayısı
mx	$(M - m)$
ky	$(N - n)$
g	Kaba kuvvet arama için iki pencere (alt görüntüler) arası geçiş değeri
mns	En küçük boyutlandırma değeri
mxs	En büyük boyutlandırma değeri
ns	Boyutlandırma sayısı
inc	Yeni boyutlandırma için artış miktarı
sc	Boyutlandırma değeri
th	Görüntü eşlemenin kabul edilebilirliği için eşik değeri
d	Şablon görüntü için görüntü eşleme yöntemi çıkış değeri
s	Alt görüntü için görüntü eşleme yöntemi çıkış değeri
$c; l$	İki ayrı görüntü için piksellerinin renk değerleri (K, Y veya M)
$\bar{c}; \bar{l}$	İki ayrı görüntü için ortalama renk değerleri (K, Y veya M)
σ	Bir görüntünün renk değerlerinin standart sapması (K, Y veya M)
cc	İki görüntünün renk değerleri arasında korelasyon katsayısı (K, Y veya M)
F	Bireyin uygunluk değeri
I	Nüfustaki toplam birey sayısı
p	Bireyin uygunluk değerine göre seçilme olasılığı
BFS	Kaba kuvvet arama
ST	Tek iş parçacıklı (paralel olmayan)
MT	Çok iş parçacıklı (paralel)
SS	Akıllı arama

ÖZET

GÖRÜNTÜ İŞLEME TEKNİKLERİ VE YAPAY ZEKÂ YÖNTEMLERİ KULLANARAK GÖRÜNTÜ İÇİNDE GÖRÜNTÜ ARAMA

Bu çalışmada esas alınan problem, görüntü içinde görüntü aramayı hızlı ve verimli bir biçimde gerçekleştirebilmektir. Bu problem, bir görüntünün bu görüntüyü kapsayan bir başka görüntüde tespit edilmesini içerir.

Görüntü içinde görüntü arama, arama işlemlerinin ve görüntü işleme kapsamında yer alan görüntü eşleme uygulamalarının birlikte kullanımını gerektirir. Problemin istenen şekilde çözülebilmesi amacıyla yapay zekâ yöntemleri ile çalışılmıştır. Görüntü eşleme için yapay sinir ağları ve görüntünün ortalama renk değeri, görüntüdeki renk değerlerinin standart sapması, korelasyon ve görüntülerden kenar tespiti gibi görüntü işleme kütüphanelerine dayalı çeşitli fonksiyonlar kullanılmıştır. Görüntü arama için genetik algoritmalar kullanılmıştır.

Bu çalışmada, akıllı arama algoritmaları, hızlı görüntü eşleme yöntemleri ve paralel programlama tekniklerine dayanan bütünleşik bir yöntem önerilmiştir. Görüntü içinde görüntü arama, tek iş parçacığı ve çok iş parçacığı ile kaba kuvvet arama, tek iş parçacığı ve çok iş parçacığı ile akıllı arama olarak dört ayrı arama içermektedir.

Görüntü aramada paralel programlama tekniklerinin uygulanması ve arama uzayının daraltılması; görüntü eşlemede eşlemenin hızlandırılması amacıyla, zeki yöntemler önerilerek işlemler gerçekleştirilmiştir.

Arama süresini en aza indirebilecek ve görüntü eşlemeyi en verimli şekilde yapabilecek algoritmalar tasarlanarak kaba kuvvet aramalara göre çok daha verimli olacak bir sistem oluşturulmaya çalışılmıştır. Genetik algoritmalar, bir görüntüyü bir başka görüntü içinde aramada en uygun sonuçları verebilecek şekilde; arama ve eşleme süreçleri ise sonuçları doğrulama ile geliştirilmiştir. Paralel programlama için ise algoritmalar parçala ve fethet mantığına dayalı olarak tasarlanmıştır.

Son olarak, hazırlanan yazılımın uygulanabilir olması için çalışılmıştır. Ayrıca, çalışmadaki en uygun parametre değerleri, arama verileri, görüntü eşleme sonuçları, uygunluk fonksiyonları, akıllı arama algoritmaları, paralel programlama teknikleri, uygulama iyileştirmeleri ve başlangıç koşullarına dayalı çıkarımlar verilmiştir.

Bu çalışmanın esas özelliği, akıllı algoritmalar ve paralel programlama fikrine dayanan hesaplamasal olarak akıllı bir yol içeriyor olmasıdır. Önerilen tekniklerin başarısı, çok sayıda düşük ve yüksek çözünürlüklü referans ve şablon görüntü üzerinde test edilmiştir. Sonuçlar, önerilen yöntemin eşleşen görüntüleri elde etmede başarılı bir şekilde toplam arama süresini düşürebildiğini göstermiştir.

Anahtar Kelimeler: Görüntü İçinde Görüntü Arama, Görüntü Eşleme, Yapay Zekâ Yöntemleri, Yapay Sinir Ağları, Ortalama Renk Değeri, Renk Değerlerinin Standart Sapması, Korelasyon, Kenar Tespiti, Genetik Algoritmalar, Akıllı Arama Algoritmaları, Paralel Programlama Teknikleri, İş Parçacığı, Kaba Kuvvet Arama, Parçala ve Fethet, Uygunluk Fonksiyonları, Uygulama İyileştirmeleri, Referans ve Şablon Görüntü

SUMMARY

IMAGE SEARCHING INSIDE ANOTHER IMAGE USING IMAGE PROCESSING TECHNIQUES AND ARTIFICIAL INTELLIGENCE METHODS

The main focus of this work is to realize image searching inside another image in a quick and efficient way. The problem consists of detecting an image inside another image that contains the original.

Image searching inside image requires the integrated use of searching operations and image matching applications inside image processing. Artificial intelligence methods were used to solve the problem in requested way. Artificial neural networks and image processing libraries' various functions for image matching, genetic algorithms for image searching were used. These functions are average color value, color standard deviation, correlation and edge detection.

In this work, an integrated method based smart searching algorithms, quick image matching methods and parallel programming techniques were proposed. Image searching inside image has four different search types: Brute-force search with one thread and many threads, smart search with one thread and many threads.

Smart algorithms were proposed to apply parallel programming techniques and narrowing search space in image searching and getting matches faster.

A system which is more efficient against brute-force searching were created designing algorithms to minimize search time and make the matching more effective. Genetic algorithms were implemented in a way that gives optimal results in image searching. Operations were also improved in image matching with verified integrated working methods. Parallel programming algorithms were designed based on divide and conquer philosophy.

Finally, a software application has been developed that encapsulates the image searching techniques. Furthermore, inferences based on optimal parameter values, searching datas, image matching results, fitness functions, smart searching algorithms, parallel programming techniques and application optimizations, initial conditions were given.

A computationally intelligent technique relying on the idea of smart algorithms and parallel programming is the main feature of this work. The success of the proposed techniques has been experimented on several low and high-dimensional reference and template images. Simulation results have revealed that the proposed techniques can significantly reduce the total searching time for reaching to the matched images.

Key Words: Image Searching Inside Another Image, Image Matching, Artificial Intelligence Methods, Artificial Neural Networks, Average Color Value, Color Standard Deviation, Correlation, Edge Detection, Genetic Algorithms, Smart Searching Algorithms, Parallel Programming Techniques, Thread, Brute-Force Search, Divide and Conquer, Fitness Functions, Application Optimizations, Reference and Template Images

1. GİRİŞ

Görüntü işleme, askerî endüstriden sualtı görüntülemeye, robotikten astronomiye (gök bilimi), fizik, sanat, biyomedikalden uzaktan algılama uygulamalarına; coğrafi bilgi sistemlerinden, gözlem ve tahmin uygulamalarına, hayvancılık ve petrol aramaya, gazete ve fotoğraf endüstrisinden trafik, radar, tıp, güvenlik, suç (kriminal) laboratuvarlarına kadar pek çok alanda kullanılmaktadır.

Görüntü işleme, Yapay Sinir Ağları (YSA), dalgacık dönüşümü, yönlendirme süzgeçleri, Genetik Algoritmalar (GA), bulanık mantık, markov¹ rasgele alan süzgeçleri, iteratif (tekrarlamalı) hücresel resim işleme algoritması, görüntü iletimi gibi pek çok alanda uygulama alanı bulmaktadır.

Ürün pazarlama, reklam, yazıları süsleme, çeşitli resim koleksiyonları oluşturma gibi uygulamalarda; tarih, sanat, eğlence, bilim, endüstri, hayvanlar alemi gibi pek çok alanda görüntülere gerek duyulmaktadır. Teknoloji dünyasındaki hızlı gelişmelere paralel olarak sayısal (dijital) görüntüler çok büyük bir artış göstermekte ve kapsamlarının gittikçe genişlemesi ile birlikte görüntü içeren veri tabanlarının önemi artmaktadır.

Görüntü veri tabanları, sanat koleksiyonlarından, uydu resimlerinden, tıbbi resimlerden ve genel amaçlı fotoğraf koleksiyonlarından oluşur. Görüntü veri tabanlarının uygulamaya göre kullanımları değişir.

İnternet ağı metin ve görüntü içeren çok büyük boyutlu bir veri tabanıdır. İnternet ağı üzerinden görüntü arayan arama motorlarının kapsamı kelime bazlı olduklarından sınırlıdır. Fotoğraf ve çizimlere ek olarak bilimsel ve tıbbi resim koleksiyonları da mevcuttur. NLM² X ışınlarıyla taranmış resimleri, CT taramalarını, MRI resimlerini ve kadavralardan kısa aralıklarla alınmış resimleri saklar. Bunları tıbbi araştırmalarda kullanmak isteyenler için 14000'den fazla resim içerir. NASA, uydularından elde

¹ Markov özelliği, mevcut durumun açıklamasının sürecin gelecekteki evrimini etkileyebilecek tüm bilgiyi kapsamı demektir ve gelecek durumlara olasılıksal bir süreçle ulaşılmasını içerir.

² NLM, Amerika Birleşik Devletleri federal hükümeti tarafından işletilen, dünyanın en büyük tıbbi kütüphanesidir.

ettiği çok büyük miktarlardaki resimleri kamu yararına ücretsiz paylaşmaktadır. USGS Amerikan Coğrafya Dairesi kullanıcılarına, bünyesinde bulundurduğu sayısal uydu resimleri ve havadan çekilmiş resimler için internet ağını (uluslararası bilgi iletişim ağı) arama imkânı sunmaktadır.

Belli bir metinden bir görüntüye ulaşmak gibi bir görüntüden de bu görüntü ile ilgili birtakım verilere ulaşmak istenebilir. Sanat koleksiyonları içerisinde bir görüntü aranmak, bir çalışma hakkında bilgi edinmek, tıbbi veri tabanları üzerinde sorgulamalar yapılmak istenebilir. Bu görecelilikten ötürü benzer uygulamaların kapsamı oldukça geniştir. Görüntü veri tabanları milyonlarca resim içerebilir ve bu resimler içeriklerine göre sorgulanırlar. İçerikleri renk dağılımlarına, doku yapılarına, bölgesel şekillere, mekânsal¹ ve sınırsal uyuma, taslak uyumuna ve nesnel sınıflandırmalara dayanabilir. Oysa metin tabanlı bir erişim sistemi, görüntülere erişimin çok kısıtlı olmasına neden olabilir. Görüntüler çoğu durumda sadece kelimelerle dizinlenmiş olabilir ve bu dizinleme şeklinin bu işlemi yapan kişiye özgü olması bu görüntülere daha sonra erişilmesine engel olabilir.

Görüntüleri kodlamada kullanılan anahtar kelimeler görüntülere erişimleri zorlaştırabilir, çünkü resimleri ifade eden ilgili bazı terimler hariçte bırakılabilir. Ayrıca, anahtar kelime yaklaşımında bir sınırlama söz konusudur ve anahtar kelimelerin otomatik aranmasından dönen sonuçlarda kullanıcı aradığı resimden çok farklı bir tanesiyle karşılaşabilir.

Pek çok sayıda resim içeren görüntü veri tabanlarında, kelime tabanlı arama yaparak ilgili görüntülere ulaşmaya çalışma, tüm görüntüleri birer birer karşılaştırma ya da görüntüleri rasgele gözden geçirme gibi uygulamalar oldukça maliyetli işlemlerdir. Çoğu zaman bu büyük külfetin getirisi pek çok resmin atlanmasına neden olabilmektedir.

Çok büyük boyutlardaki görüntü veri tabanları üzerinde sorgulamalar yapmak büyük önem taşımaktadır. Görüntüler üzerinde arama ve karşılaştırma, İçerik Tabanlı Görüntü Erişim (İTGE) işlemlerinin gerçekleştirilebilmesi amacıyla sorgulamalara gerek duyulmaktadır. Öte yandan görüntüler metin tabanlı aramalardan çok daha fazlasına gerek duymaktadırlar.

¹ Mekânsal (spatial), uzaysal anlamına da gelir ve üç boyutluluğu içerir. Mekânsal uyum, uzaysal şekil ve durumların biçimsel olarak değişim ve dönüşümleri arasındaki uyumunu içerir.

Görüntü İşleme Teknikleri (GİT) kapsamında yer alan görüntü eşleme, bir görüntü parçasının diğer görüntüdeki yerinin araştırılması işlemidir. Sayısal görüntü işleme kapsamında yer alan ve görüntü eşleme yöntemlerinden biri olan şablon eşleme yöntemi ile bir görüntüde şablon bir görüntü aranabilir. Şablon görüntü büyük görüntüde yer alıyor ise eşleşen küçük görüntü parçalarının konumu tespit edilebilir.

Görüntü eşleme, uzaktan görüntü algılama uygulamalarında, görüntüyü daha büyük görüntülerin içerisinde aramada, İTGE sistemlerinde, kalite kontrolün bir parçası olarak üretimde, mobil bir robota erişim yolu olarak ya da bir görüntüdeki kenarları tespit etme yolu olarak kullanılabilir. Ayrıca histoloji, patoloji, tomografi görüntülerini sınıflandırma (tıbbi görüntü işleme), yüz tanıma, görüntüler arasındaki birtakım ilişkilerin belirlenmesi ve görüntülerin düzenlenmesi, nesnelere sınıflandırma veya kaynak görüntüde benzer nesnelere tespit etme gibi pek çok alanda da farklı uygulama alanları bulabilir. Görüntüler daha küçük görüntüler ile sorgulanmak istenebilir.

Şablon eşlemenin çok farklı uygulamaları vardır ve küçük bir görüntüyü büyük bir görüntü içerisinde arayacak eşleme algoritmalarının geliştirilmesine gerek duyar. Bu noktada mühim olan, küçük bir görüntünün daha büyük olanının bir parçası olacağı şekilde eşleşen konumların doğru ve hızlı bir şekilde bulunmasıdır.

1.1 Tezin Amacı

Günümüzde metin tabanlı aramalar sayısal ortamdaki verilerin aranmasında çok büyük bir yer tutmaktadır. Aramaların Çoklu Ortam Verileri (ÇOV) üzerinde yapılması için metin tabanlı arama algoritmalarından çok daha fazlasına gerek duyulur. Çünkü ÇOV metinsel nitelik taşıyan üst veri ve görsel nesneden birtakım hesaplama ve GİT sonucunda elde edilen görsel nitelikler ya da ses içermektedir. Dolayısıyla, ÇOV üzerinde yapılan aramalarda metin tabanlı aramalarda olduğu gibi sadece üst veri üzerinde değil, içerik ve diğer nitelikler üzerinde de sorgulamalar söz konusudur. ÇOV sadece metinlerden ibaret olmadıkları için bu veriler üzerinde yapılan sorgulamalar içerik tabanlıdır. Bu sorgulamalar, bir görüntünün içeriğinde ne olduğuyla, ne gibi şekiller içerdiğiyle ya da bir ses dosyasının veya video kaydının içeriğiyle ilgili yapılacak aramalar olabilirler.

İTGE'de arşivlerdeki görüntüler sadece veri tabanında saklanan pasif nesnelere olmaktan kurtularak sorgulama ve erişim sürecine doğrudan katılan aktif nesnelere hâline gelmelidirler. Örneğin kullanıcı, herhangi bir görüntüde yer alan bir tümörün dokusuna veya şekline benzerlerinin yer aldığı diğer görüntülerin arşivden getirilmesi şeklinde bir sorgulama yapmak isteyebilir.

Bu tezde, bir görüntünün başka bir görüntüde aranarak yer alıp almadığının tespit edilmesi ve büyük görüntüde yer alan küçük görüntü konumları ile eşleşme bilgilerini benzerlik derecelerine göre listelemek amaçlanmaktadır.

Bu çalışmada, son kullanıcının dizinleme, etiketleme ve benzeri birtakım ek işlemler yapmak zorunda kalmadan sorgulamalarını yapabileceği etkili bir sistem tasarımı ve aramaların olabildiğince hızlı bir biçimde gerçekleştirilmesi amaçlanmaktadır. Paralel programlama teknikleri ve kodlama kapsamındaki iyileştirmelerle görüntü aramalarının verimli ve hızlı bir şekilde yapılması mümkündür.

Görüntü İçinde Görüntü Arama (GİGA) probleminde esas üzerinde durulan konu, en hızlı aramayı en verimli şekilde gerçekleştirebilmektir. Bu çalışmadaki esas amaç, bu aramayı akıllı arama algoritmaları tasarlayarak icra edebilmektir. Çeşitli uygunluk fonksiyonları (farklı görüntü eşleme yöntemleri), bütünleşik kullanılan yaklaşımlar ve paralel (çoklu) programlama teknikleri de bu amaç kapsamında yer almaktadır.

1.2 Literatür Özeti

Literatürde pek çok görüntü eşleme çalışması yer almaktadır. Bu çalışmaların bir kısmı görüntüleri birbiri ile eşleme, bir kısmı ise görüntüyü bir başka görüntüde alt görüntü olacak şekilde eşleme şeklindedir.

Literatürdeki görüntü eşleme çalışmalarının çoğu özel olarak gerçekleştirilen çalışmalardır. İhtiyaca yönelik olarak yüz tanıma, plaka tanıma, kanser hücresi tespit etme, belli bir dokunun başka dokularla eşlenmesi, görüntülerden şekillerin elde edilmesi gibi oldukça fazla uygulama ile karşılaşılabilir. GİGA çalışmasında ise bu sözü geçen uygulamaları genelleştirerek ilgili bir görüntüyü bir başkasında aramak amaçlanmıştır.

Bu çalışma özel bir görüntü eşlemeyi değil, görüntü aramanın optimizasyonunu (en uygun duruma getirme, en iyileme) gerektirir. Bu çalışmada görüntü eşleme teknikleri kullanılarak tek ve çok sayıda iş parçacığı ile icra edilen aramalar

karşılaştırılmıştır. Ayrıca arama, akıllı algoritmalarından GA ile tek ve çok sayıda iş parçacığı ile gerçekleştirilmiştir.

Literatür çalışması kapsamında incelenen çalışmalardan bazıları öznitelik çıkarımı uygulamalarını, bazıları şablon eşleme yöntemlerini, bazıları bu uygulamaların birlikte kullanımı ile verimli görüntü eşleme işlemlerini içermektedir.

Görüntü içerikli sorgulamaların renk ortalama tabanlı tekniklerle yapılması mümkündür (Kekre ve diğ., 2010). Görüntü özniteliklerinin boyutları, her görüntü için özellik vektör boyutu, satır, sütun ve köşegen ortalamaları kullanarak çokça düşürülebilir. Görüntü erişimi için özellik vektörü olarak görüntünün tüm piksellerini kullanmak yerine özellik vektörlerinin kullanımıyla daha iyi başarımlar (performans) ve daha düşük hesaplama sonuçları ortaya çıkabilir. Renk ortalaması tabanlı teknikler yardımı ile İTGE'nin zaman karmaşıklığını azaltarak başarımları artıran geliştirme işlemleri elde edilebilir.

Renk histogramı, renk momenti ve doku gibi olağan özelliklerden ayrı olarak, kenar yoğunluğu için kenar histogramı olarak adlandırılan yeni bir öznitelik çıkarım algoritması kullanılabilir (Dubey ve diğ., 2010). Kenarlar resimdeki gerekli bilgileri taşırlar ve bu nedenle de görüntü erişimine uygulanabilirler. Kenar histogram tanımlayıcısı, kenarların mekânsal dağılımını yakalar. Dikey, yatay, 45 ve 135 derece köşegensel ve eş yönlü beş grup içeren kenar nitelikleri kullanılarak ve tekniklerin ortalamaları elde edilerek görüntülere erişim sağlanabilir. Kullanılan teknikler ile piksel piksel mesafeler ayrı ayrı hesaplanır ve sonuç görüntüler için ortalamalar bulunarak n adet görüntü ve öklid uzaklığı¹ matrisleri elde edilir. Bir görüntüye erişim birden fazla özelliğin dâhil edilmesi ile geliştirilebilir.

Görüntülerin donuk yüzeyleri iyileştirilerek bir İTGE sistemi tasarlanabilir (Singh ve diğ., 2010). Görüntülere ışıklandırma ve gölgeleme dikkate alınarak birtakım ön işlemler uygulanabilir. Böylece bir görüntünün görsel kalitesi iyileştirilerek öznitelikleri daha doğru bir şekilde çıkarılabilir. Görüntülerden daha iyi renk deseni ve kenar yoğunluk değerleri üretilebilir. Böylelikle ulaşılmak istenen ile daha ilgili görüntülere erişim gerçekleştirilebilir.

¹ Öklid uzaklığı, iki nokta arasındaki doğrusal uzaklıktır ve n boyutlu Öklid uzayı için P ve Q

noktaları arasındaki Öklid uzaklığı şu şekilde tanımlanır: $\sqrt{\sum_{i=1}^n (p_i - q_i)^2}$.

Görüntü tanımda temel problemlerden biri görüntüler arasındaki benzerliği tespit etmektir. Bu problemin NP¹ bir problem olması, çözümü için birtakım yakınsama algoritmalarını ve basitleştirmeleri gerektirir (Keysers ve Unger, 2003).

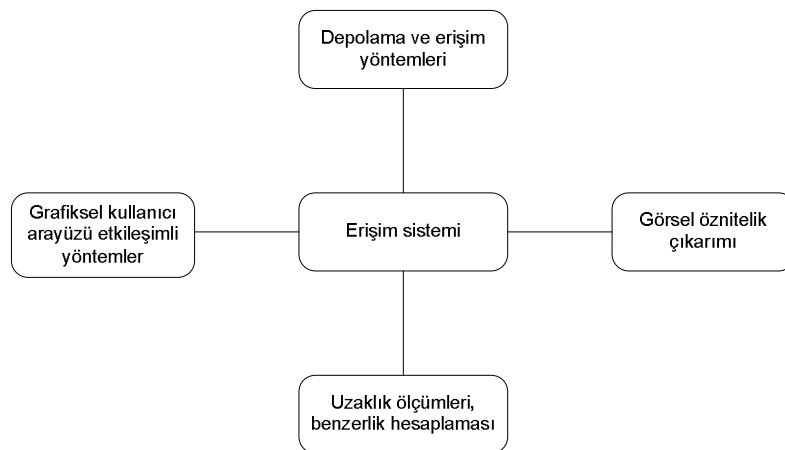
İki boyutlu şekil eşlemeleri için şeklin sınır noktalarının dağılım histogramı kullanılabilir (Shu ve Wu, 2011). Bu tanımlayıcı ile şekil tabanlı eşleme ve erişim işlemleri gerçekleştirilebilir.

GA, görüntü eşlemeleri öncesinde görüntülerden elde edilen öznitelik uzayındaki karmaşıklık azaltmak için kullanılabilir (ElAlami, 2011). Öznitelikler bu yolla ayırt edilerek görüntülere erişimde en ilgili olacak şekilde çıkarılabilirler. GA sayesinde görüntü erişim süreci hesaplamaları basitleşebilir ve görüntülerin tespit edilme oranları artabilir.

İTGE sistemlerinin etkisi, çoklu öznitelik vektörlerinde hesaplanan görüntü benzerliklerinin ağırlıklandırılması ve doğrusal olmayan birleşimleri (kombinasyon) ile geliştirilebilir (Torres ve diğ., 2009). Öznitelik birleşimi her zaman başarılı olmayabilir ve bütünleşik benzerlik fonksiyonu karmaşıklaşabilir. Kullanıcı beklentilerini karşılamak için bu problem genetik programlama yolu ile çözülebilir.

Kullanıcının ilgili görüntüleri belirtmesi ile gerçekleştirilen İTGE, ilgililerle beraber ilgisiz görüntülerin de belirtilmesi ile geliştirilebilir (Ferreira ve diğ., 2011). Bu işlemler için genetik programlama kullanılabilir.

İTGE sistemlerinin temel yapısı Şekil 1.1’de görülmektedir.



Şekil 1.1 : Tüm İTGE sistemlerinin temel bileşenleri (Müller ve diğ., 2004).

¹ Problemin çözümü için gereken zamanın, problemin boyutunun büyümesi ile çok hızlı bir şekilde arttığı problemler olan NP problemlerin çözümü, yakınsama algoritmaları kullanımını gerektirebilir.

İTGE sistemlerinin erişim doğruluğunu geliştirmek amacıyla, düşük seviyeli öznitelik çıkarımları yerine görsel öznitelikler ile kullanıcının ilgili olduğu görüntüler arasındaki anlamsal boşluğu azaltmaya odaklanılmalıdır (Liu ve diğ., 2007). Bu sebeple yüksek seviyeli anlamsal tabanlı görüntü erişimleri sağlanmalıdır. Anlamsal boşluğun düşürülmesinde Liu ve diğ. (2007), resmin çeşitli özelliklerine dayalı tekniklerin içerdiği beş temel yöntem ifade etmişlerdir:

- Nesneleri gruplayarak yüksek seviyeli içerikleri tanımlamak
- Makine öğrenmesi yöntemlerini kullanarak düşük seviyeli öznitelikleri sorgu içerikleri ile ilişkilendirmek
- Uygunluk geri beslemesi (geri bildirim) ile kullanıcıların ilgili olduğu görüntüleri öğrenmek
- Anlamsal şablon üreterek yüksek seviyeli görüntü erişimini desteklemek ve doğrulamak
- İnternet ortamında görüntü erişimi için görüntülerin görsel içeriklerini ve metinlerden elde edilen verileri beraber kullanmak

Daha verimli görüntü erişimleri için renk, doku ve şekil bilgisi birleştirilebilir (Wang ve diğ., 2011). Görüntülerden ilk olarak renk, ardından doku ve sonra şekil tabanlı öznitelikler çıkarılarak görüntü erişimi için gürültüye karşı gürbüz olan öznitelik kümesi elde edilebilir. Bu sayede kullanıcının ilgili olduğu görüntülere erişim daha doğru ve verimli bir şekilde gerçekleştirilebilir.

Görüntüleri eşlemek için iki görüntüde benzeşen noktalar çeşitli uygulamalarla tespit edilebilir. Benzeşen nokta çiftleri tam doğru eşlemeler olabileceği gibi olma ihtimali olan nokta çiftleri de olabilirler. Her aday (tam doğru olma ihtimali değişebilir) nokta çifti aynı zamanda diğer aday nokta çiftleri için oylayıcı olarak değerlendirilebilir (Yuan ve diğ., 2011). Eşleşen çiftler diğer çiftlerin bu çift için belirlediği değerler üzerinde toplama ve sıralama işlemleri ile en uygun şekilde hesaplanabilirler. Eşleşen çiftlerin tutulduğu matris yapısındaki bilgi tam olarak kullanılarak ve bu basit işlemler ile ilgili çiftler bire bir eşlemeden birden çoğa eşlemeye genişletilebilir. Böylelikle daha etkili ve yeterli görüntü eşlemeleri icra edilebilir.

Yüz tanıma uygulamalarında, eğitim ve test süreçleri bu süreçlerin karmaşıklığı daha az aşama ve daha basit ifadeler ile azaltılarak çok daha düşük hatada daha az faktör

ile yapılabilir (Yan ve diğ., 2011). Görüntülerdeki insan, ışıklandırma, bakış açısı ve pikseller gibi faktörler, insan faktörü ve diğer faktörler olarak iki aşamada değerlendirilebilir. İnsan faktörü dışındaki faktörler doğrudan tanıma ile ilgili değildir, ama bu dış faktörler ayrı bir aşama olacak şekilde bütün olarak ele alınabilir. Bu iki aşamalı model, yüz tanıma için kullanılabilir, ayrıca benzer şekil ve biçimi paylaşan görüntüleri birleştirme gibi çeşitli uygulamalara da genişletilebilir.

Şablon eşleme ile tanıma uygulamalarında en düşük maliyetli (zaman açısından verimli) fonksiyonları elde etmek amacıyla paralel programlama kullanılabilir (Lin ve Chang, 2011). Referans ve test kümeleri arasındaki benzerlik paralel programlama teknikleri ile hesaplanabilir. Veri kümeleri üzerindeki eşlemeler için değişebilen (dinamik) eşik değerleri kullanılabilir

Gri tondaki görüntülere nazaran çok kanallı görüntüler için şablon eşlemenin verimli bir şekilde yapılabilmesi amacıyla, sıfır ortalamalı korelasyon tabanlı hızlı ve ayrıntılı bir teknik kullanılabilir (Mattocchia ve diğ., 2011). Böylelikle görüntülerde eşleşmeyen aday noktalar tespit edilebilir.

Her parmak izi görüntüsünün gereksiz ayrıntı noktaları dağılımını içeren benzersiz birer içeriği vardır ve bu içerik ile parmak izi eşlemeleri yapılabilir (Kumar ve Vikram, 2010). Görüntülerin işlemler öncesi iyileştirilmesi ve ardından inceltilmiş ikili parmak izi görüntülerinin elde edilmesinin ardından çok boyutlu YSA kullanarak görüntü eşlemeleri gerçekleştirilebilir.

Görüntüleri tanımlamak için renk dışında pek çok özellik daha kullanılabilir, fakat cilt hastalıklarının (dermatolojik) analizi için İTGE'de renk güçlü bir araçtır (Bunte ve diğ., 2010). En kaliteli renk tabanlı görüntü öznelik kümesi elde edilerek ve görüntü özneliklerinin daha düşük boyutlu uzaya dönüştürülmesi ile renk tabanlı erişimlerde iyileştirmeler sağlanabilir. Bu uygulama, benzersiz renk tanımlayıcıları ile cildiye alanında İTGE uygulamaları için farklı renk uzayları üzerinde icra edilebilir ve her renk uzayı için sonuçlar karşılaştırılabilir.

Görüntü eşleme uygulamalarında farklı ölçekler için görüntü sınırları parçala ve fethet (divide and conquer) algoritmaları ile belirlenebilir (Moon ve diğ., 2010). Bu yöntem özniyelemeli algoritmalar kullanılarak icra edilebilir. Farklı ölçeklerde

görüntü eşleme problemi için görüntü uzayından zaman serileri¹ uzayına geçişle karmaşıklık azaltılabilir. Verilen bir sorgu dizisine benzer veri dizilerini veri tabanında arayarak zaman serileri eşlemesi ile görüntü eşleme icra edilebilir.

Görüntü eşleme uygulamalarında, eşleşen aday görüntü noktaları elde edilerek sadece bu noktalar üzerinde görüntü eşleme uygulaması icra edilebilir (Choi ve Kim, 2002). Böylelikle daha düşük boyutlu öznitelik vektörleri ile daha hızlı görüntü eşleme uygulamaları iki aşamalı olarak gerçekleştirilebilir.

Nokta deseni eşleme uygulamaları görüntü eşleme kapsamında yer alan uygulamalardan biridir. Bu uygulama ölçeklendirme ve açı farkları gibi durumlarda ya da gürültü içeren görüntüler üzerinde başarılı olabilmelidir (Chang ve diğ., 1997). Görüntülerden elde edilen parametre uzayının boyutu düşürülerek işlemlerin daha hızlı yapılması sağlanabilir. Böylece gürültü etkilerine karşı daha gürbüz bir sistem elde edilebilir ve bu sistem çizgi ve sınır desenleri eşleme uygulamasına genişletilebilir.

Görüntü eşlemeye görüntülerdeki benzer alanlardan elde edilen bilgileri kullanmak için gerek duyulabilir. İki görüntüde birbirlerine uyan kontrol noktaları ile açısız ve yönsel farklar içeren görüntüler eşleştirilebilir ve böylece bir şablon görüntü ile görünüşsel olarak aynı görüntü bir başka görüntüde konumlandırılabilir (Ding ve diğ., 2001).

Görüntü veri tabanlarında etkili görüntü erişimi için içerik tabanlı nesne düzenlemesi yapılabilir. Görüntüleri sınıflandırma, görüntüleri anlam düzeni tabanlı gözden geçirme ve çok boyutlu dizinleme gibi uygulamalarla işlemler daha verimli yapılabilir. Görüntülerden öznitelikler açısından nispeten homojen olan görüntü kısımları çıkarılarak elde edilen yapılar kullanılarak görüntü verilerini düzenlemek ve benzerlik tabanlı görüntü sorgulamalarını çok boyutlu dizinleme ile daha hızlı gerçekleştirebilmek mümkündür (Kwok ve Zhao, 2006). Görüntüleri tanımlayan yapılar ve indeksleme yöntemleri kullanılarak daha sonra görüntülere erişimde daha düşük boyutlu arama uzayı elde edilebilir ve görüntülerden elde edilen bu yapıların

¹ Zaman serisi, tahmini önceden bilinen olayları temel alarak gelecek olayları tahmin etmenin bir kavramsal modelidir. Zaman serisi analizlerinde veri istatistiklerini almak için çeşitli yöntemler mevcuttur.

hassas bir şekilde sınıflandırılması görüntülerin otomatik olarak sınıflandırılmasını hızlandırabilir.

Gri tondaki görüntülerde ikili yapıdaki hedef nesnelere için şablon eşleme yöntemi olarak korelasyon yöntemi yerine parametrik olmayan istatistiksel bir yapı ile çok daha hızlı sonuçlar elde edilebilir (Fernandez, 1997). Görüntü histogramlarının kümülatif dağılımı arasındaki mutlak farklılıklar dikkate alınarak şablon eşleme icra edilebilir.

Görüntü eşleme için problem piksel değerlerinin satır satır okunması ile tek boyutlu dizilerde alt dizinin aranması problemine çevrilebilir. Görüntü eşleme uygulamaları için kombinasyonel yöntemler ve işlem sürelerinin kısaltması için sezgisel yöntemler kullanılabilir (Fredriksson ve Ukkonen, 1999).

Görüntüde piksel boyutlarının artırılması ile eşleme hataları artar (Debella-Gilo ve Kaab, 2011). Alt piksel hassaslığı ile düşük çözünürlüklü görüntüleri eşleme, piksel çözünürlüğü tabanlı görüntü eşlemelerine göre farklı sonuçlar verebilir. Temel piksel boyutunu düşürmek sureti ile eşleşen görüntülerin uzaysal çözünürlüğü artırmak aynı çözünürlüklü orijinal referans görüntüye göre hata değerindeki düşüşü artırabilir. Uzaysal çözünürlüğün düşürülerek piksel boyutlarının artırılması ile ortalama hata, yanlış veya tespit edilemeyen eşlemeler için artış görülebilir. Görüntülerin farklı çözünürlükleri ve piksel hassaslıklarına göre eşlemelerdeki değişimler gözlemlenebilir.

GİGA çalışması, GA'nın problemin arama sürecinde kullanılması ile gerçekleştirilir. Görüntü eşleme amacıyla gereken öznitelik çıkarımı için hazır kütüphane fonksiyonları kullanılır. Bu yüzden öznitelik çıkarımı sürecinde, öznitelik uzayının boyutunun azaltılması, öznitelik vektörlerinin farklı değerler ile ağırlıklandırılması gibi uygulamalar icra edilmez.

Tez çalışması kapsamında incelenen çalışmalar, görüntü eşleme uygulamalarında en hızlı ve kaliteli görüntü öznitelikleri üzerine yoğunlaşmışlardır. Literatürdeki çalışmalardan farklı olarak yapılan tez çalışmasında ise asıl amaç bir şablon görüntünün referans görüntü ya da görüntülerde aranmasıdır. Görüntü aramalarının hızlı bir şekilde gerçekleştirilmesinin hedeflendiği bu çalışmanın üstünlüğünü, akıllı arama algoritmalarının paralel programlama teknikleri ve kodlama iyileştirmeleri ile birleştirilerek şablon görüntüyü araması oluşturmaktadır.

1.3 Hipotez

Görüntü içerikli aramalarda anlamsal boşluklar bulunmaktadır. Görüntülerden otomatik olarak ayıklanan görsel özniteliklere ek olarak, görüntüyü tanımlayan betimlemelerin de kullanılması anlamsal boşluğu kapatmanın olası bir yoludur. Ancak, erişim sonuçlarının optimizasyonu için hangi görsel ve metinsel öznitelik eşlerinin en iyi olacağıın da tespit edilmesi gerekmektedir. Dolayısıyla, giriş görüntüsü en iyi şekilde analiz edilmeli, analiz sonuçlarına göre taranacak alandaki görüntüler üzerinde uygun işlemler yapılmalı ve sorgulamalar buna göre gerçekleştirilmelidir. En iyi sonuçlar için kullanıcıya parametreleri düzenleyebilme imkânı ve birtakım ek seçenekler sunulabilir. Görüntü içerikli arama ve görüntü eşlemede bu gibi uygulamalardan faydalanılması, genel ve yerel özelliklerden beraberce yararlanmak suretiyle daha iyi bir erişim sağlanması bu tez çalışması dâhilindedir. Fakat, metinsel özniteliklerden ve görüntüyü tanımlayan farklı betimlemelerden yararlanılması ile görüntü eşleme için en iyi görüntü özniteliklerinin elde edilmesine yoğunlaşılması ve karşılaştırılacak görüntülerin üzerinde gerekli işlemler yapılmadan önce bu görüntüleri farklı görüntü parçalarına ayırarak her görüntü için farklı yapılar elde etmek bu çalışmada yer almamaktadır.

Bu çalışmada gürbüzlük, arama algoritmalarının ve arama süresince sabit değerdeki statik değişkenlerin¹ değerlerinin hassasiyetini artırarak; ayrıştırıcılık ise eşleşen ve eşleşmeyen görüntülerin sayısını ve kalitesini artırarak sağlanmaktadır.

Görüntü eşleme için YSA ile bütünleşik olarak ek yöntemler, arama süresini en aza indirebilmek için etkili ve akıllı algoritmalarından GA kullanılmaktadır. Görüntü arama sürecinde benzerlik hesaplaması için alternatif yöntemler önerilmekte ve arama optimizasyonu için paralel programlama tekniklerinden yararlanılmaktadır.

Tez çalışmasında, görüntüler arasında ilişki ölçümü (metrik), piksel benzerlik karşılaştırmaları, görüntülerden elde edilen niteliklerin karşılaştırılması gibi işlemleri gerçekleştiren algoritmalar ile renk dağılımı ve şekilsel benzerlikler gibi birtakım özellikler üzerinde eşlemeler gerçekleştirilmektedir. Böylece görüntü arama sürecinde İTGE, sadece renk tabanlı özellikler üzerinden değil, doğrudan girilen

¹ Statik değişkenler, sınıf örneği yaratılmadan erişilen alanlardır. Birden fazla sınıf örneğinin içerdiği statik olmayan her alan her sınıf örneği için farklı değerler içerebilir, fakat ilgili sınıfta yer alan statik alanlar ortaktır ve her biri tek bir alanı ifade eder.

görüntüyü sorgulayarak daha net sonuçlar elde edebilmek için çeşitli nitelikler üzerinden sağlanmaktadır.

Gerçekleştirilen uygulama tıpta bilgisayar destekli tanı amacıyla, karakollarda kanıta dayalı çıkarımlarda ve çeşitli güvenlik uygulamalarında kullanılabilir.

Bir görüntünün bir şablon görüntü ile eşleşen küçük parçalarını bulmak için tüm söz konusu küçük görüntülerin sırasıyla şablon görüntü ile karşılaştırılması sureti ile yapılan aramalara kaba kuvvet aramalar denir. Bu uygulama zaman, maliyet ve işlemler açısından çok pahalı olmaktadır.

Sorgulanan görüntüde her bir olası görüntü parçası ile şablon görüntünün karşılaştırılması ve uyum algoritmasının korelasyon¹ tabanlı olması fazlaca işlem yükü yaratmaktadır.

Şablon bir görüntü ile eşleşen küçük görüntüleri daha büyük bir görüntüde aramak için öznitelik seçiminde amaç, bir başarımlı ölçütünü en iyi şekilde ifade eden altküme bulmaktır. Bu başarımlı ölçütü, aranan görüntü ile karşılaştırılan görüntüler arasındaki uzaklığı en aza düşürmek ve sınıflandırıcının başarımlı yükseltmektir. Bu çalışmada çözüm uzayındaki aramanın optimizasyonu için GA kullanılmaktadır. Bu tekniğin üstünlüğü, GA'nın altküme seçim problemlerine çözüm getirebilmeleridir.

Büyük görüntü içinde alt görüntüleri aramada, büyük görüntü küçük görüntüyü kapsayacak küçük görüntülere ayrılabilir. Alt parçalara (büyük görüntüden elde edilen görüntü parçaları) ayırmada kaybolabilecek ara kısımlara mutlaka dikkat etmek gereklidir. Alt görüntü parçaları üzerinde paralel programlama uygulanmak suretiyle yapılacak işlemler ile toplam sürenin optimizasyonu sağlanabilir.

Ayrıca, arama öncesi bilgi kaybı en az olacak şekilde aranacak görüntüyü ve şablon görüntüyü yeniden boyutlandırmak veya şablon görüntünün bazı kısımlarını dikkate almamak suretiyle arama uzayı daraltılarak da işlemler icra edilebilir.

Bu çalışmada icra edilen görüntü eşleme işlemi, sadece yüz eşleme veya herhangi bir nesne eşleme gibi belli bir özel eşleme işleminden ibaret değildir. YSA sürecinde, eşleşen kısma aranmak istenen ve aranılana yakın görüntüler, eşleşmeyen kısma ise

¹ Korelasyon, nicel x ve y değişkenlerinin içerdikleri değerler dizisi arasında bir uygunluk hâlinin olup olmadığını araştıran yöntemdir.

eşlenen görüntülere uzak ve aranmak istenen görüntülerin haricindeki görüntüler vererek bir görüntüde ilgili olunan alt bir görüntüyü aramak için ağ eğitilebilir. Paralel programlama teknikleri sürecinde ise iş parçacıklarının her biri diğer iş parçacıklarından bağımsız ama bu iş parçacıkları ile paralel çalışacak şekilde tasarlanmıştır. Bu süreçlerde çeşitli çıkarımlar yapılarak birtakım öneriler sunulmuştur.

GİT ile birlikte Yapay Zekâ (YZ) yaratabilmek için YSA ve GA birlikte kullanılmıştır. Görüntü eşleme ve görüntü arama algoritmalarının verimli çalışabilmesi için akıllı algoritmalar, paralel programlama teknikleri, parçala ve fethet mantığına dayalı algoritmalar bütünleşik olarak kullanılmıştır.

Ayrıca, şablon görüntüden ve aranacak görüntüdeki eşleme işlemine tabi tutulacak alt görüntülerden elde edilen çeşitli özellikler (ilgili olunan görüntüyü en iyi şekilde ifade eden en az özellik) ve görüntülerin belli kısımları eşleştirilerek arama uzayı daraltılmış ve arama süresi düşürülmüştür. Böylece şablon görüntüdeki tüm piksellerin, referans görüntüdeki şablon görüntü boyutlarında her alt görüntüde karşılık gelen tüm piksellerle karşılaştırılmasının önüne geçilmiştir.

Görüntülerde bir başka görüntüyü aramak için çeşitli öneriler sunulularak birtakım uygunluk fonksiyonları oluşturulmuş ve bu fonksiyonların bir arada çalışmaları sağlanarak hem doğruluk hem de hız yönünden verim ve zaman tasarruf yolları aranmıştır.

Paralel programlama teknikleri GA ile birlikte kullanılarak paralelliğin işlevselliği ön plana çıkarılmıştır. Özellikle GA kısmında en uygun nesil, birey sayıları, çaprazlama ve mutasyon işlemcileri olasılıkları ile şekilleri, yeni nesli oluşturma uygulamaları, birey seçim yöntemleri ve etkili uygunluk fonksiyonları, yeterince testin yapıp değerlendirilmesi ile en iyi sonuçları verecek şekilde belirlenmiştir.

Elde edilen sonuçların doğruluğu birtakım analizlerle kontrol edilmiş, amaçlananların gerçekleştirilebilmesi için çok sayıda görüntü ile çalışılmıştır. Görüntü eşleme kısmında YSA kullanılırken uygun şekilde sinir ağının eğitimi tamamlanmış ve arama işlemlerine geçilmiştir.

Görüntü karşılaştırmada kullanılan renk, doku, şekil gibi özellikler mesafe ölçümleridir ve görüntü aramalarının doğruluğu için bütünleşik olarak kullanılmaları gerekmektedir. Aranmak istenen görüntüyü tanımlayan kaliteli ve yeterli bilgi ile son

kullanıcı, insan yüzü ya da bir ifade gibi belirli nesnelere veya bir metindeki karakterleri arayabilir. Bu bağlamda nesne tanıma arařtırmaları büyük önem taşımaktadır.

Bu çalışmada, kodlama ve arayüz tasarımı için programlama platformu olarak nesneye yönelik programlama dillerinden biri olan C# (C Sharp) programlama dili kullanılmaktadır. Microsoft Visual Studio¹ 9.0 ile GİT'yi uygulayabilmek için C# ile yazılmış .NET² kütüphanelerinden biri olan Emgu CV³ ve Aforge.NET⁴ kullanılmaktadır.

Çalışma genel olarak GİT ve YZ yöntemlerinden oluşmaktadır. GİT ile görüntülerde ön işlemler, öznitelik çıkarımı, bölümlenme (segmentasyon) ile analiz işlemleri icra edilmekte ve görüntü eşleme gerçekleştirilmekte; YZ yöntemleri ile ise GİT'yle çıkarılan verilerin benzerliklerine göre sınıflandırılmaları ve şablon görüntülerin hızlı şekilde aranması sağlanmaktadır. Arama algoritmaları GA kullanılarak icra edilmektedir.

Kullanıcıdan aradığı görüntü ile ilgili herhangi bir kelime girmesi veya dizinleme, etiketleme gibi işlemler yapması beklenmeden kullanıcıya sorgulamak istediği görüntüyü içeren görüntülere erişebilme olanağı sağlayacak bu çalışmada, doku ve şekilsel benzerliklerle beraber yine renk benzerliği ve birtakım nesnel sınıflandırmalar üzerinde de durulmaktadır. Bu sayede kullanıcıya daha kesin eşleşen görüntüler sunabilecek bir sistem tasarımı bu tezin bir diğer önemidir.

Çalışmada aranmak istenen görüntü bütünüyle taranacak alandaki görüntünün herhangi bir kısmında geçebilir. Sorgulamalar, "Bir kedi resmi görüntü kümesindeki resimlerin içeriğinde herhangi bir yerde var mı?" ya da "Bir kişinin giriş olarak verdiği bir nesnenin parçasını taranacak resimlerden hangileri içeriyor?" şeklinde olabilir.

¹ Microsoft Visual Studio, Microsoft tarafından üretilmiş bütünleşik bir geliştirme ortamıdır. .NET platformunu destekleyen Visual Studio ile masaüstü ve internet ortamı tabanlı projeler geliştirilebilir. Bu projeler grafik kullanıcı arayüzleri içerebilir.

² .NET, Microsoft tarafından geliştirilen bir yazılım geliştirme platformudur.

³ Emgu CV, Intel OpenCV görüntü işleme kütüphanesinin C# paketidir. <http://sourceforge.net/projects/emguvcv/> adresinden ulaşılabilir.

⁴ Aforge.NET, bilgisayar bilimi ve YZ alanlarındaki uygulama geliştiren kişiler ve arařtırmacılar için tasarlanan bir sistemdir. <http://www.aforgenet.com/framework/downloads.html> adresinden ulaşılabilir.

Bu iki görüntü işleme kütüphanesi ile uygulamada görüntüyü işleyebilme imkânı elde edilir.

Sorgulamalarda eşleme esnasında ilişkisel benzerlik ölçümü olarak daha önceden belirlenmiş mekânsal ilişkilere ve elde edilen nesne kümelerine göre görüntüler arasındaki ilişkilerin belirlenmesi yolu ile mekânsal uyum, karşılaştırılacak görüntülerin sınırlarının gösterimi ve çıkarımın sınır uyum algoritmalarıyla elde edilmesi ile sınır uyumu kullanılabilir. Sınır uyumu, sınır piksellerin bir dizisi olarak ifade edilebilir veya bir çokgenle (poligon) yaklaşık olarak ifade edilir.

Tez kapsamında yapılan çalışma ile GİGA uygulamalarının hızlı bir şekilde gerçekleştirilebilmesi sağlanmıştır. Görüntü eşleme için farklı yöntemler ve görüntü aramaları için farklı yaklaşımlar kullanılmıştır. Geliştirilen yöntemler, kısa sürede başarılı sonuçlar elde edilmesini sağlamıştır. Bu yöntemlerin çeşitli görüntülere uygulanması ile elde edilen sonuçlar karşılaştırılarak değerlendirilmiştir.

1.4 Tezin Katkıları

GİGA için ele alınan uygulamalar şunlardır:

- Arama problemi
- Paralel programlama teknikleri
- Görüntü eşleme yöntemleri
- Kodlama iyileştirmeleri ve yazılım optimizasyonu
- Farklı ölçeklerde görüntü arama

Bu çalışma ile görüntü işleme teknikleri ve YZ yöntemleri tabanlı görüntü eşleme; paralel programlama teknikleri ve YZ yöntemleri tabanlı görüntü arama uygulamaları gerçekleştirilmiştir.

Görüntü aramalarının yapılabilmesi için problem alt problemlere ayrılabilir. Bu durum, büyük bir görüntünün aranacak görüntüyü kapsayacağı küçük görüntülere ayrılarak bu küçük görüntülerde aranmak istenen görüntünün aranması şeklinde icra edilir.

Bu çalışma ile kaba kuvvet ve kaba kuvvet olmayan arama yaklaşımları, paralel ve paralel olmayan arama teknikleri gözlemlenebilir. Bu alternatifler kullanılarak bir görüntüde bir başka görüntü aranabilir. Uygulama, görüntülerin farklı ölçeklendirme değerlerinde farklı görüntü eşleme yöntemleri ile gerçekleştirilebilir.

Kaba kuvvet ve akıllı arama sonuçları, paralel ve paralel olmayan arama yaklaşımları karşılaştırılabilir. Farklı görüntü eşleme yöntemleri ile aranmak istenen görüntüyü büyük görüntüde aramak için farklı fonksiyonlar kullanılabilir.

Görüntü aramaları öncesinde büyük ve küçük görüntü aynı oranlarla boyutlandırılarak GİGA'ya tabi tutulabilir. Böylelikle farklı ölçeklendirme değerlerinde ne gibi sonuçlarla karşılaşılacağı takip edilebilir.

GİGA çalışması ile uygulamanın paralel programlama teknikleri, farklı görüntü eşleme yöntemleri ve arama yaklaşımları ile gerçekleştirilmesi amaçlanmıştır. Elde edilen sonuçlar, görüntü aramaları için kullanılan birden fazla görüntü eşleme yöntemi, iki farklı arama yaklaşımı ve uygulama süreci için paralel ve paralel olmayan teknikler ile gerçekleştirilen uygulamaları içermektedir.

Literatürde etkili görüntü eşleme uygulamaları için kullanılan öznitelik tabanlı yaklaşımlar, bu çalışmadaki paralel programlama yaklaşımı ile birleştirilerek iyileştirilebilir. Şablon eşleme yöntemleri için genetik algoritma tabanlı yaklaşımlarda, bu çalışmanın genetik algoritma sürecinde kullanılan yöntemlerinden faydalanılabilir.

Öte yandan bu çalışmadaki arama tabanlı problem, literatürdeki diğer çalışmalarda yer alan birden fazla öznitelik vektörünün bir arada kullanılması, bu vektörlerin farklı ağırlık değerleri ile ağırlıklandırılması, elde edilen vektörlerin boyutlarının daraltılması, nokta eşleme ya da çeşitli ayrıntılar tabanlı görüntü eşleme gibi çeşitli uygulamalar ile daha genel bir yapı hâline getirilebilir.

Bu tez ile görüntüde görüntü aramak için GA'nın nasıl bir yaklaşım olabileceği görülmüştür. Çoklu iş parçacıklarının arama sürecine getirdiği iyileştirmeler göz önüne alınarak iş parçacığı sayısının değişimine göre arama süresindeki değişimlere dayalı parametrik olarak çıkarımlarda bulunulabilir.

Örüntü tanıma uygulamalarında yaygın olarak kullanılan YSA, özel örüntü tanıma uygulamaları yerine genel olarak görüntü tanıma uygulamalarında da başarılıdır. Bu tez ile YSA ve diğer görüntü eşleme yöntemleri arasında karşılaştırmalar yapılabilir.

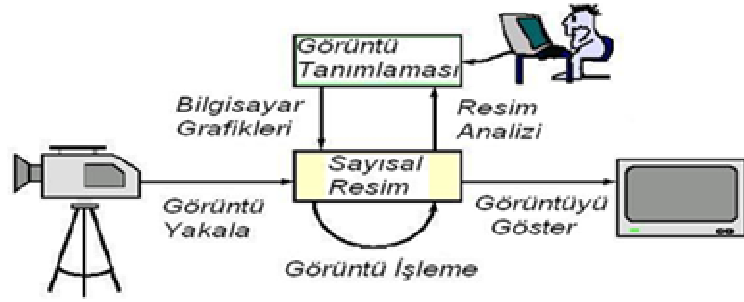
Sözü geçen yöntem ve yaklaşımların çeşitli birleşimleri ile bir görüntüde bir başka görüntü aranabilir.

Bu çalışma ile elde edilen sonuçlar ve sonuçlara dayalı yapılan çıkarımlar, yapılacak görüntü arama ve görüntü eşleme çalışmaları için yol gösterici nitelikte olabilir.

Bu çalışmanın takip eden kısımları şu şekilde düzenlenmiştir. İkinci bölümde görüntü işleme, görüntü işleme ile ilgili yöntem ve yaklaşımlar, görüntü işleme teknikleri ve görüntü işleme teknikleri ile gerçekleştirilebilecek çalışma ve uygulamalar; üçüncü bölümde YZ yöntemlerinden YSA ve GA ile bu yöntemlerle gerçekleştirilebilecek uygulama ve çalışmalar anlatılmaktadır. Dördüncü bölümde bu tezde ele alınan problem, GİGA çalışmasının içeriği ve ilgili diyagramlar, ilgili görüntü arama yaklaşımları ve görüntü eşleme yöntemlerine; beşinci bölümde tez çalışması için gerçekleştirilen yazılım, arayüz ve kodlama sürecindeki iyileştirmelere ve son bölümde ise elde edilen sonuçlara ve bu sonuçlara dayalı çıkarımlara yer verilmektedir.

2. GÖRÜNTÜ İŞLEME

Görüntü işleme, sayısal bir resim hâline getirilmiş olan gerçek yaşamdaki görüntülerin bir giriş resmi olarak işlenerek o resmin özelliklerinin ve görüntüsünün değiştirilmesi sonucunda yeni bir resim oluşturulmasıdır. Şekil 2.1’de bir görüntü işleme sistem yapısının blok şeması verilmiştir.



Şekil 2.1 : Görüntü işleme sistem yapısı.

Görüntü işleme, verilerin, yakalanıp ölçme ve değerlendirme işleminden sonra başka bir aygıtta okunabilir bir biçime dönüştürülmesi ya da bir elektronik ortamdan başka bir elektronik ortama aktarılmasına yönelik bir çalışma olan işaret işlemeden farklı bir işlemdir. Daha çok, kaydedilmiş olan mevcut görüntüleri işlemek yani mevcut resim ve grafikleri değiştirmek, yabancılaştırmak ya da iyileştirmek için genellikle de fotoğrafçılık ve grafik tasarım alanlarında kullanılır.

Görüntü işleme, yaşam boyunca önemini koruyacak olan bir çalışma alanıdır. İnsanlar ve hayvanlar gözleri ile analog temele dayanan görüntü işleme yapmaktadırlar. Bu olay beyin yardımı ile çevrim içi (online), paralel ve çok spektrumlu¹ olarak meydana gelir.

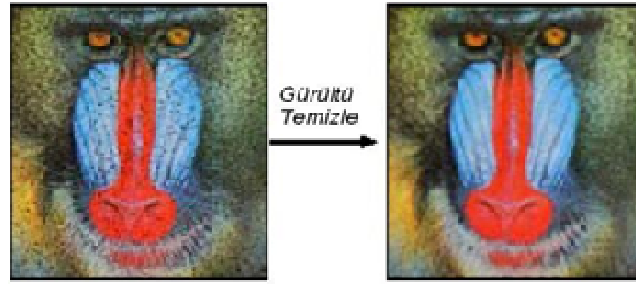
Resimlerin bilgisayar ortamında değerlendirilebilmeleri için veri biçimlerinin bilgisayar ortamına uygun hâle getirilmesi gerekmektedir. Bu dönüşüme sayısallaştırma adı verilir.

¹ İnsan gözünün görebileceği elektro manyetik dalga boyu aralığına spektrum denir.

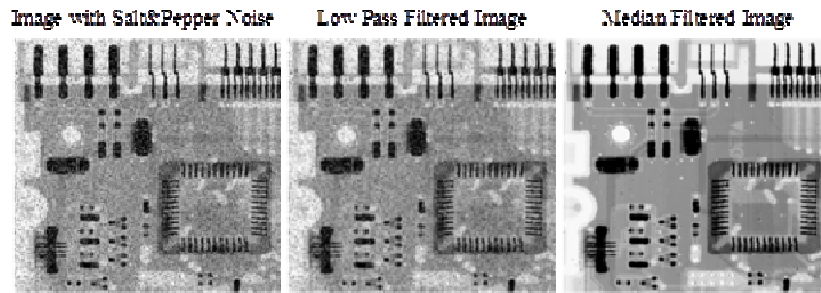
Bir resmin fotografik sunumu daha doğrusu sayısal biçime dönüştürülmesi çeşitli şekillerde mümkündür. Bunlara farklı teknikler kullanılarak resmin sayısallaştırıldığı tarayıcılar, analog/sayısal dönüşümün kullanılarak resmin sayısal hâle dönüştürüldüğü sistemler (Frame-Grapper), uzaktan algılamada uçak ya da uydulara yerleştirilen çok kanallı tarayıcılar örnek olarak verilebilir.

Sayısal bir resim deyince akla analog bir işaretin sayısal bir işarete dönüştürülmesi gelmelidir. Bu da nesne tarafından yayılan enerjinin (analog işaret) bir algılayıcı tarafından öngörülen elektromanyetik aralıkta algılanarak sayısal işaret hâline dönüştürülmesi ile mümkündür.

Resimlerin analog ortamlardan sayısal ortamlara geçirilmesi süreci sonrasında yeni oluşan görüntü, gürültü (noise) içerir. Görüntü işleme bu bozuklukları gidermek için kullanılabilir. Görüntü işleme ile gürültülerin giderilmesi Şekil 2.2’de ve Şekil 2.3’te görülmektedir.



Şekil 2.2 : Sayısal ortama geçirilen gürültülü ve gürültüleri giderilmiş görüntüler.



Şekil 2.3 : Görüntüyü gürültüden arındırma örneği.

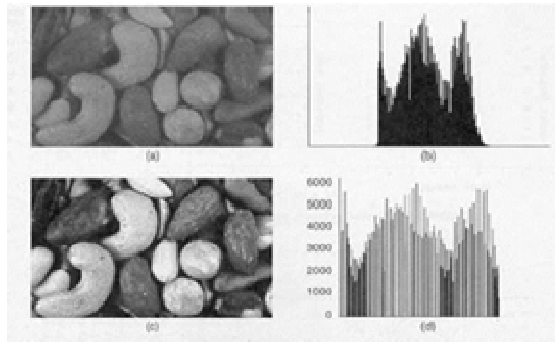
Şekil 2.3’te soldan sağa (metinde yan yana sıralı verilen her bileşen grubu için geçerli) sırayla, tuz ve biber gürültülü, alçak geçiren filtreli (süzgeç) ve medyan (orta değer) filtreli görüntüler görülmektedir.

Histogram kavramı, görüntünün içerdiği piksel değerlerinin ağırlığını belirten grafiksel bir gösterimdir. Görüntü işlemede histogramlar kullanılarak görüntüler Şekil 2.4’te ve Şekil 2.5’te görüldüğü gibi belirgin hâle getirilebilirler.

Daha net bir görüntü için histogramın daha ayırık ve düzgün bir yapıya gelmesi sağlanmalıdır. Bu işleme histogram eşitlemesi (Histogram Equalisation) denir.

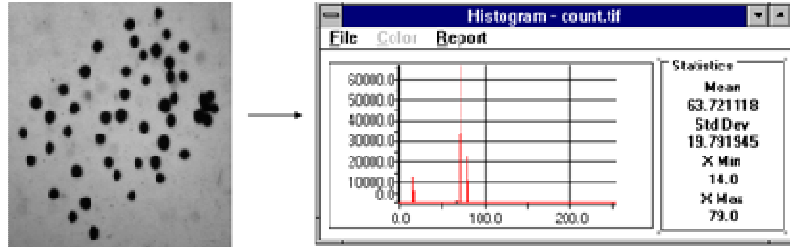


Şekil 2.4 : (a)Dağınık piksel dağılımlı ve (b)histogram eşitlemeli görüntüler.



Şekil 2.5 : Görüntü işlemede histogram yardımıyla resmin belirginleştirilmesi.

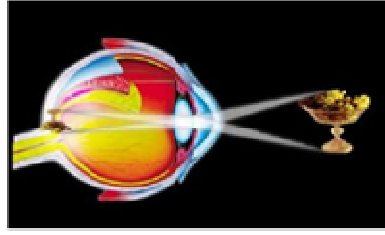
Resimlerin faydalı işler için işlenmesi örneklerinden biri Şekil 2.6'da verilmiştir.



Şekil 2.6 : Bir görüntüden bu görüntüdeki kan sayısı gibi veriler elde edilebilir.

2.1 Görüntü Nedir

Görüntü, gerçek yaşamdaki üç boyutlu nesnelere oluşan bir sahnenin basit iki değişkenli bir fonksiyon olarak tanımlanmasıdır. Başka bir deyişle görüntü, üç boyutlu görünümün iki boyut üzerindeki haritası olarak tarif edilebilir. Şekil 2.7'de görüntünün oluşumu resmedilmiştir.



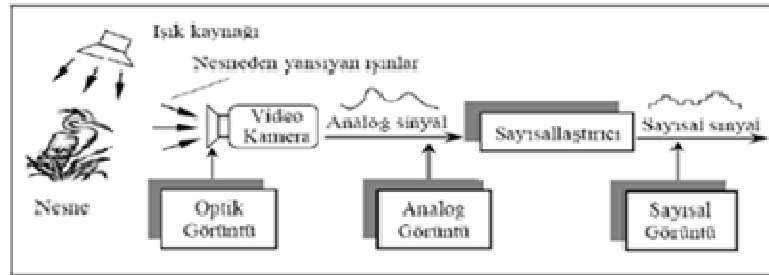
Şekil 2.7 : Görüntünün gözde oluşumu.

2.2 Görüntü Yakalama

Görüntü yakalama, gerçek yaşamdaki bir nesne için herhangi bir sahnenin fotoğraf makinesi veya kamera gibi donanımlarla fotoğrafının çekilmesidir. Görüntü yakalama, film ve benzeri hareketli görüntülerden programlar aracılığıyla da gerçekleştirilebilir (Klavyedeki *Print Screen* tuşuna basmak suretiyle ekran görüntüsünü hafızaya almak gibi).

2.3 Sayısal Görüntü

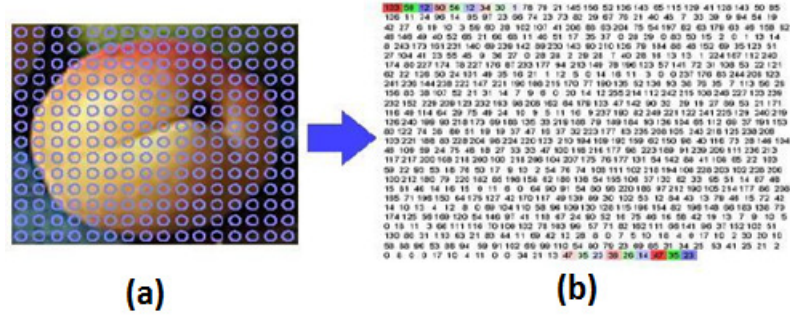
Sayısal görüntü, sayısal değerlerden oluşan ve bilgisayar ortamında görüntülenebilen görüntü olup oluşturulma aşamaları Şekil 2.8’de verilmiştir.



Şekil 2.8 : Görüntü yakalama ve sayısallaştırma aşamaları.

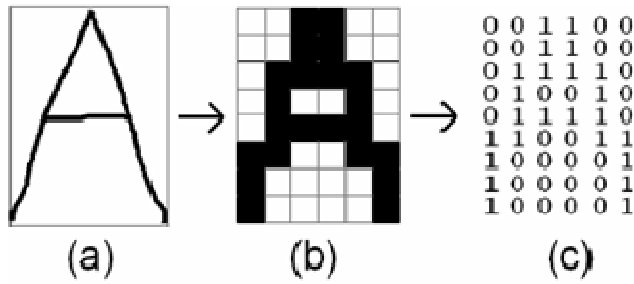
$I(x,y)$ gibi bir fonksiyonla temsil edilen analog bir görüntü veya resimde, I bir şiddet birimi (örneğin parlaklık), x ve y değişkenleri ise görüntünün sırayla yatay ve dikey eksenindeki koordinatlarına karşı düşen değerlerdir. Sayısal görüntü ise bu analog görüntünün M sütun ve N satırdan oluşacak şekilde örneklenmesi (yeniden boyutlandırma) sonucu elde edilir. Satır ve sütunun kesiştiği her bölgeye piksel adı verilir ve sayısal görüntüye çevrilen resimde $M \times N$ adet piksel bulunur.

Analog bir görüntü ve bu görüntünün örneklenmesi ile elde edilen sayısal görüntünün içeriği (matris olarak) Şekil 2.9’da verilmiştir.

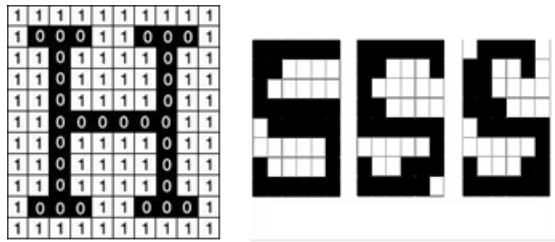


Şekil 2.9 : (a)Analog görüntü ve (b)sayısal karşılığı.

Görsel karakterlerin sayısallaştırılması ise Şekil 2.10’da ve Şekil 2.11’de görülmektedir.



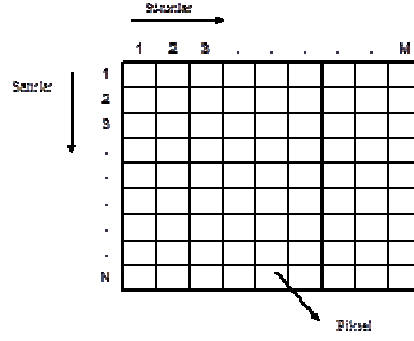
Şekil 2.10 : (a)A harfi, (b)gösterimi ve (c)sayısallaştırılması.



Şekil 2.11 : H harfi için sayısallaştırma ve S harfi için üç ayrı gösterim.

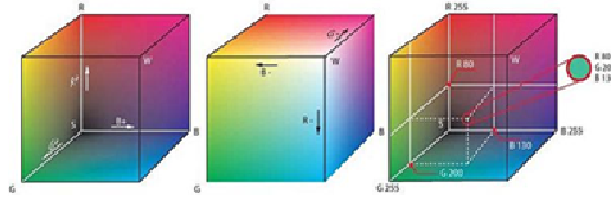
2.4 Sayısal Görüntü Nitelikleri

1 ve 0 değerleri sırasıyla, aydınlık ve karanlık bölgeleri veya nesne ve nesnenin önünde veya üzerinde bulunduğu zemini temsil ederler. Sayısal görüntü dosyaları, renkli olarak genellikle 24 ya da 8 bit; gri seviye görüntüler olarak 1, 2, 4 ya da 8 bit olabilirler. Bir görüntü için piksellerinin gösterimi Şekil 2.12’de verilmiştir.



Şekil 2.12 : Bir görüntü için piksellerin ifadesi.

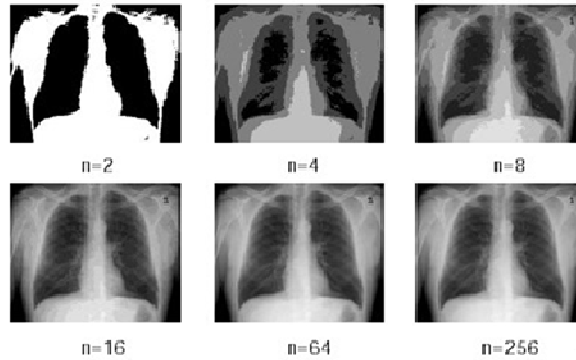
Sayısal bir görüntünün en temel parçası olan her bir piksele ilişkin bir renk söz konusudur. En sık kullanılan renk uzaylarından biri RGB'dir. RGB renk uzayı, Kırmızı (K), yeşil ve mavi ana renklerinin belirli oranlarda karışımı ile elde edilen yaklaşık 17 milyon rengi içerir. Şekil 2.13'te bu renklerin oluşturulma biçimi verilmiştir.



Şekil 2.13 : RGB renk uzayına göre renklerin oluşturulması.

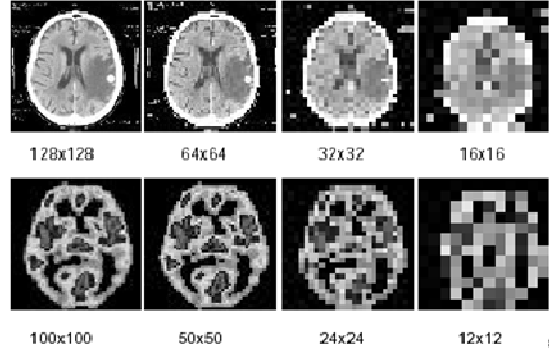
İnternette kullanılan renk sistemi RGB renk sistemidir. Bunun sebebi, RGB'nin 1953'te ilk fotoğraf makinesi Polaroid'de ve ondan sonra da televizyonlarda standart kabul edilmiş olmasıdır. RGB günümüzde de tüplü ekranlarda, tarayıcılarda, televizyon ve otomatik olmayan fotoğraf makinelerinde standart olarak kullanılır.

$n = 2^b$ olmak üzere, b değeri görüntünün bir pikselini ifade etmek için gereken bit sayısı iken n değeri bu görüntünün her pikseli için söz konusu gri tonu sayısıdır. Şekil 2.14'te aynı görüntünün farklı bit değerlerindeki görünüşleri verilmiştir.



Şekil 2.14 : Aynı görüntünün piksellerinin farklı bit değerlerindeki durumları.

Uzaysal çözünürlük (Spatial Resolution), bir resmin bir pikselinin fiziksel büyüklüğüne eşittir. Kısaca çözünürlük, bir resmin detaylanabilir en küçük parçasıdır. İki görüntünün farklı çözünürlükteki görüntüşleri Şekil 2.15'te verilmiştir.



Şekil 2.15 : Farklı çözünürlükteki görüntüler.

2.5 Görüntü İşleme Adımları

Şekil 2.16'da verilen görüntü işleme adımları, ilk olarak sayısallaştırılan resmi griye çevirme ile gürültü temizleme uygulanması ve diğer filtrelemeler de uygulanarak gerekli algoritmalarla yeni görüntüler elde edilmesi şeklindedir. Görüntü işleme kullanılarak yapılan tüm işlemler, görüntü üzerindeki en küçük görüntü elemanı olan pikseller üzerinde gerçekleştirilmektedir.



Şekil 2.16 : Görüntü işleme adımlarının gösterimi.

Görüntü işleme adımlarından bazıları şunlardır:

- Gürültü temizlemek
- Parlaklık, koyuluk ve doğru renk ayarlamak
- Görüntü keskinleştirmek ve bulanıklaştırmak

2.6 Görüntü İşleme Kullanım Alanları

- Askerî endüstri (denizaltı ses dalga taramaları), sualtı görüntüleme
- Güvenlik, suç laboratuvarları

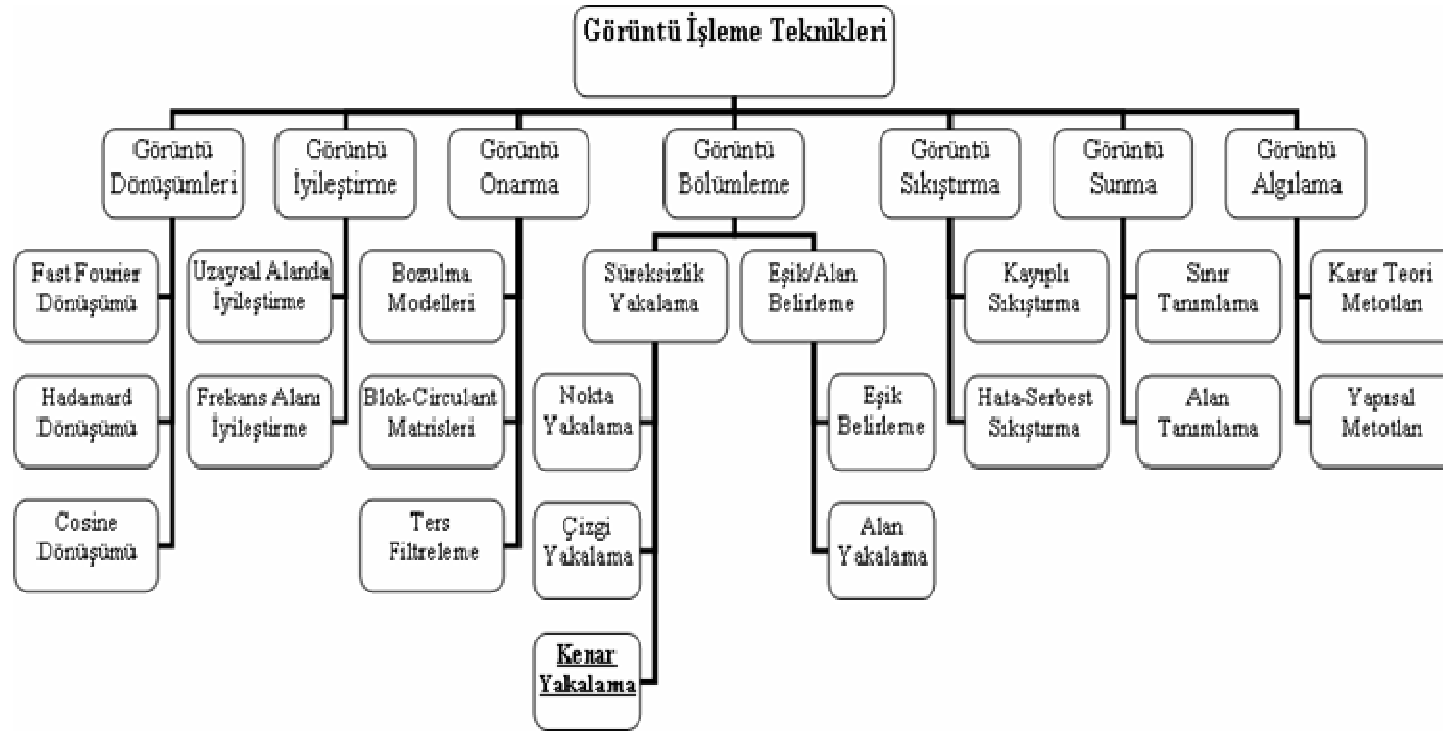
- Tıp (tümör, damar gibi yapıların belirginleştirilmesi, tomografi, ultrason)
- Robotik, trafik, astronomi, radar
- Gazete ve fotoğraf endüstrisi uygulamaları
- Hayvancılık (sığır eti kalite tayini), petrol arama
- Fizik, sanat, biyomedikal alanları
- Uzaktan algılama uygulamaları: Uydu görüntüleri üzerinde nüfus yoğunluğu, yerleşim yerleri, çevre kirliliği ve benzeri çevresel şartların tespiti, hava gözlem ve tahmin uygulamaları

2.7 Görüntü İşlemede Kullanılan Temel Yöntemler

- Geometrik dönüştürme: Boyutlandırma, büyütme, küçültme ve döndürme gibi işlemler
- Renk düzeltme: Parlaklaştırma, keskinlik ayarı, renk dönüştürme gibi işlemler
- Sayısal karıştırma veya optik karıştırma: İki veya daha fazla görüntüyü birleştirme
- Görüntü düzenleme: Görüntünün kalitesini yükseltme ve görüntü netleştirme
- Sabit cisimleri tespit ederek iki boyutlu nesne tanıma
- Parazit oluşturma
- İstatiksel bilgiler çıkarma
- İşlem ve filtre seçenekleri (Gaussian Filters)
- Görüntü üzerinde birtakım bilgileri gizleme (Steganografi)
- Kayıpsız görüntü sıkıştırma
- Diğer matematiksel ve biçimsel (morfolojik) işlemler

2.8 Görüntü İşleme Teknikleri

Şekil 2.17’de görülen GİT kullanılarak görüntüler işlenebilir ve yeni görüntüler elde edilebilir.



Şekil 2.17 : Görüntü işleme tekniklerinin dallandırılması.

2.9 Görüntü İşleme Teknikleri Kullanılarak Yapılabilecek Çalışmalar

- Kameralar ile uzaktan izleme, plaka tanıma sistemi
- Yüz, parmak izi, karakter, kâğıt para (banknot) veya görüntü tanıma
- Bir fabrikada üretilen ürünün görüntü kalitesini test etmek: Bir portakal işleme bandında belli boyutlardaki portakalları ayırmak
- Bozuk para sayma makinesi yapıp paraları ayırmak
- Ürün üretim hatalarını kontrol etmek ve hatalı olan ürünleri ayırmak
- Farklı şekillerin kenarları arasındaki açılar ölçmek
- Bir kamera ve basit birkaç motor ile program kontrollü herhangi bir şeyi takip eden robot kolu tasarımı
- Çeşitli kalite kontrol testleri

2.10 Görüntü Formatları

GİT'den biri olan görüntü sıkıştırma uygulaması için ilk akla gelen, kayıplı görüntü sıkıştırmada bir standart hâline gelmiş olan JPEG yöntemidir. Bu yöntem fotoğraf gibi görüntülerin sıkıştırılması için çok elverişlidir. Fakat şekiller gibi düşük renkli görüntülerde, kayıplı sıkıştırmanın sonucunda ortaya çıkan görüntüdeki bozulma daha fazla gözle görünür hâle gelmektedir. Bu yüzden bu tür görüntülerin sıkıştırılmasında genellikle kayıpsız yaklaşımlar tercih edilir.

Renkli görüntüler bilgisayarlarda gri düzeyli görüntülere göre daha fazla yer kaplarlar. Örneğin 8 bitlik¹ 1024×1024 pikselden oluşan bir görüntü, bilgisayarda 1024x1024 byte = 1048576 byte = 1024 kilobyte (KB) = 1 megabyte (MB)² yer kaplar. Renkli olarak ise (K-Y-M, üç band) 3x1 = 3 MB yer kaplar. Görüntüdeki piksel sayısı ve görüntünün gri renk düzeyi, kaliteyi ve boyutu etkiler.

¹ 8 bit 1 byte'a (Bayt, B) karşılık gelir. Sekiz bitlik bir yapıda her bit 1 veya 0 değeri ile ifade edilir. Bu sebeple $2^8 = 256$ gri ton değer aralığı ile görüntüler tek kanallı; K, Y, M ve alfa (Alpha, saydam (şeffaf) olmama) olarak dört renk kanalı ile görüntüler 32 bitlik ifade edilebilirler.

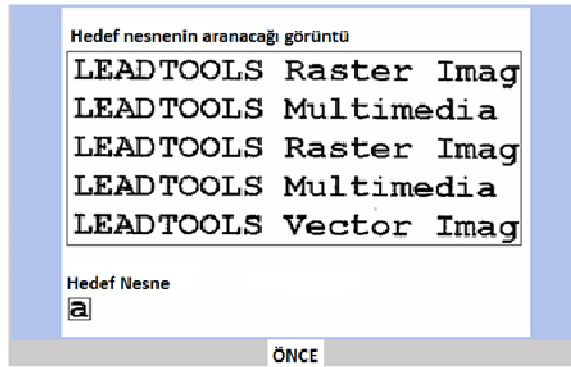
² 1024 byte 1 KB'a, 1024 KB 1 MB'a ve 1024 MB ise 1 GB'a karşılık gelmektedir.

2.11 Görüntüde Alan Bulma

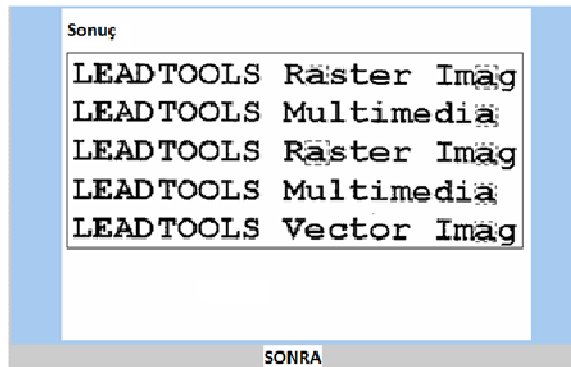
Bir görüntü parçasının başka bir görüntüdeki yerinin araştırılması işlemine görüntü eşleme adı verilir. Şekil 2.18’de görüntü eşleme örneği görülmektedir. Korelasyon yöntemi ile bir görüntüde bir alan bulunması Şekil 2.19’da ve sonucu Şekil 2.20’de verilmiştir. Görüntü eşleme ile bir görüntüde başka bir görüntü aranabilir. Görüntüden nokta detaylar, kenarlar ya da alanlar çıkarılarak çeşitli detaylara dayalı eşleme icra edilebilir.



Şekil 2.18 : Görüntü eşleme yöntemlerinden görüntüde alan bulma örneği.



Şekil 2.19 : Küçük a harfinin aranması örneği.



Şekil 2.20 : Orijinal görüntü ve bulunan hedef nesnelere.

2.12 Görüntü İşleme Tekniklerinin Kullanımı

Görüntü işlemek için kullanılan yöntemler, ilgilenilen probleme uygun olarak geliştirilebilirler. Uygulamalar, çeşitli yöntemlerin birlikte kullanılmasını, birtakım ölçütlerin dâhil edilmesini ve işlem hızlarının artırılmasını gerektirebilirler. Görüntü işlemek için farklı alanların farklı yöntemleri kullanılabilir.

Görüntü işlemek için kullanılan teknikler, problemlere göre başka yöntemlerle çalışılmasını gerektirebilirler. Görüntülerin modellenmesi, görüntülerden çeşitli verilerin çıkarılması, görüntü karşılaştırılması, benzerlik ölçümleri, görüntü aramaları, görüntü eşlemeleri gibi uygulamalar ya da İTGE sistemleri çok farklı yöntemlere gereksinim duyabilirler.

Yöntemlerin seçimi, üzerinde çalışılan görüntülere, beklenen çıkışlara ve uygulama sürecine bağlıdır. Üzerinde çalışılan görüntüler çok farklı çözünürlükte ya da bit derinliğinde olabilirler ve gürültü içerebilirler. Görüntü işleme uygulamaları, uygulamanın bilgisayar ortamında en uygun şekilde tasarlanmasını ve uygulama süreci için etkili iyileştirme yöntemlerini gerektirebilir.

Öte yandan, gerçekleştirilen uygulamaların gerçek hayatta uygulanabilir, kullanıcı etkileşimli ve çeşitli koşullara karşı sonuç verebilen bir yapıda olması gerekebilir. Tüm bu gereksinimler göz önüne alınarak görüntü işleme için gerekli altyapı oluşturulmalı ve işlemler bu şekilde icra edilmelidir.

2.13 Görüntü Veri Tabanı Düzenlemesi

Geniş resim veri tabanlarındaki bütün resimleri aramak zordur. Bunun yerine resimler düzenlenmeli ve dizinlenmelidir. Böylece verilen sorgu için veri tabanının sadece ilgili kısmına erişilir. Görüntü kümesindeki resimleri gruplara ayırarak aranan resmin ilgili grupta aranması suretiyle arama uzayı daraltılabilir. Çoğu ilişkisel veri tabanında¹ sayısal ve metin tabanlı veriyi dizinlemek için standart yöntemler mevcuttur. Mekânsal verileri dizinlemek için de mevcut yöntemler kullanılmaktadır. Mevcut arama sistemlerinde, içerik tabanlı resim araması için resimleri dizinleme yöntemleri geliştirilmektedir. Özel olarak geliştirilen yöntemler kullanarak görüntü veri tabanları ile ilgili işlemler yapılabilir.

¹ Bir ilişkisel veri tabanı veriyi veri kümesi içinde bulunan ortak özellikleri kullanarak eşler.

3. YAPAY ZEKÂ

YZ, insan zekâsına özgü olan algılama, öğrenme, çoğul kavramları bağlama, düşünme, fikir yürütme, sorun çözme, iletişim kurma, çıkarım yapma ve karar verme gibi fonksiyonları analiz ederek bunların benzeri yönergeleri geliştirmeye çalışmaktır. Bulanık mantık, YSA, GA, uzman sistemler, örüntü tanıma gibi dallar YZ'nin alt dallarıdır.

YZ ile pek çok alanda karşılaşılabilir. Bu alanlara şu örnekler verilebilir:

- Şekil tanıma, ses tanıma, optik harf veya nesne tanıma
- Veri madenciliği
- Bilgi sistemleri
- Doğal dil işleme
- Teorem ispatlama
- Yapay yaşam
- Satranç, dama ya da strateji oyunları
- Makine öğrenmesi
- Robotik

YZ yöntemleri, mühendislik, satış ve ulaşım, üretim, dağıtım, finans gibi alanlarda; tasarım, çizelgeleme ve sıralama gibi uygulamalarda kullanılmaktadır. Diğer teknikler kullanarak problem çözmede yetersiz kalınan durumlar ve karmaşık problemlerin çözümü için yeni teknikler geliştiren YZ yöntemleri, insanın ve çevrenin davranışını modelleyerek problem çözmeyi amaçlamıştır.

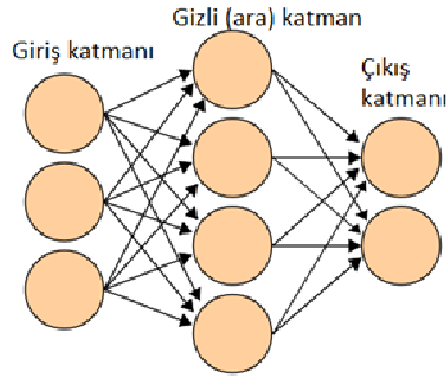
Problem çözümünde dilsel ifadeler kullanarak bulanık mantık, beyin sinir hücrelerinin (nöronlar) çalışma şekline benzer şekilde yararlanarak YSA, doğal olaylarının optimizasyonu ile GA, YZ'nin birer alt dalı olarak kullanılmaktadırlar.

İnsana özgü öğrenme, genelleme, yorumlama gibi eylemlerin benzetimi (simülasyon) bilgisayar ortamında problemlere uygulanabilir. Problemlerin çözümünde belli bir ifade olmadığı durumlarda bilgisayarlarla problemleri çözebilmek gerekir. Bu işlemler YZ sistemleri ile gerçekleştirilebilir. Bu sistemlerle ilgili araştırmaların yapılması ve yeni yöntemlerin geliştirilmesi mümkündür.

YZ'nin temel amacı problemlerin bilgisayarlar yardımıyla modellenebilmesidir. Öğrenme yöntemlerinin bilgisayar sistemlerine aktarılması, uzman sistemlerin oluşturulması, insanlar ve bilgisayarlar arasındaki iletişimi kolaylaştırabilecek birimlerin geliştirilmesi, bilgisayarların buluşlar ve bilimsel araştırmalarda kullanılabilmesi YZ amaçları kapsamındaki uygulamalardır.

3.1 Yapay Sinir Ağları

Bilgisayarlarda YZ yaratma yöntemlerinden biri olan YSA, biyolojik sinir ağlarından modellenmiş, insan beyninin bilgi işleme sürecinden esinlenerek geliştirilmiş bir yöntemdir. Geleneksel algoritmalar ve programlama teknikleri kullanarak çözümlerin çok zor ya da karmaşık olduğu problemleri çözümlerin bir yoludur. Bir giriş, bir çıkış ve bir ara katmandan oluşan tipik bir yapay sinir ağı Şekil 3.1'de görülmektedir.



Şekil 3.1 : Yapay sinir ağı birbirine bağlı düğümlerden oluşan bir gruptur.

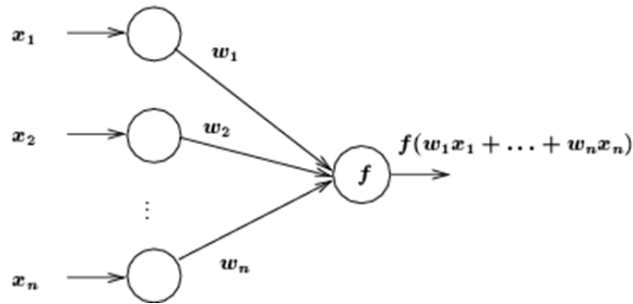
3.1.1 Yapay sinir ağı hücresi

Yapay sinir hücreleri gerçek sinir hücrelerinin benzetimi ile gerçekleştirilir. Biyolojik sinir hücresine göre çok daha basit bir yapıya sahip temel bir yapay sinir ağı hücresi Şekil 3.2'de görülmektedir. Yapay sinir ağı hücresinde temel olarak, dışarıdan girilen ya da diğer sinir hücrelerinden alınan giriş verileri, ağırlık değerleri, toplam fonksiyonu, ağırlık fonksiyonu ve çıkışlar bulunmaktadır.

Dış ortamdan girilen veriler ağırlıklar ile sinir hücresine bağlanır ve bu ağırlıklar ilgili girişin etkisini belirler. Ağırlıklar hücreler arasındaki bağlantıların sayısal değeridirler ve hücre üzerine gelen bilginin değeri ile hücre üzerindeki etkisini gösterirler. Net etkinin en yaygın kullanım şekli toplama fonksiyonudur. Şekil 3.2’de verilen toplama fonksiyonu, girişlerin ilgili ağırlık değerleriyle çarpımlarının toplamıdır ve hücreye gelen net girişin hesaplanmasını sağlar. Aktivasyon fonksiyonu¹ işlem süresince ağ çıkışını hesaplar ve probleme bağlı olarak kullanımı belirlenir. Bu fonksiyon hücrenin girişine karşılık üretilecek çıkışı belirlemeyi sağlar ve en yaygın olarak sigmoid fonksiyonu² kullanılır.

Basit bir sinir hücresinde giriş sinyallerinin ağırlıklandırılmış toplamı alınır ve seçilen eşik değeri ile karşılaştırılır. Toplam eşik değerinden küçük ise çıkış -1, büyük ise +1 olur. Çıkışın +1 olması, söz konusu hücrenin aktifleşmiş olduğu anlamına gelir. Aktivasyon fonksiyonunun verdiği çıkış değeri 3.1’de verilen denklemde görüldüğü gibidir (n bu denklemde YSA için giriş sayısıdır).

$$y = f(w_1 \cdot x_1 + \dots + w_n \cdot x_n) \quad (3.1)$$



Şekil 3.2 : Genel bir yapay sinir ağı hücresi modeli ve ağırlıklı giriş toplamı.

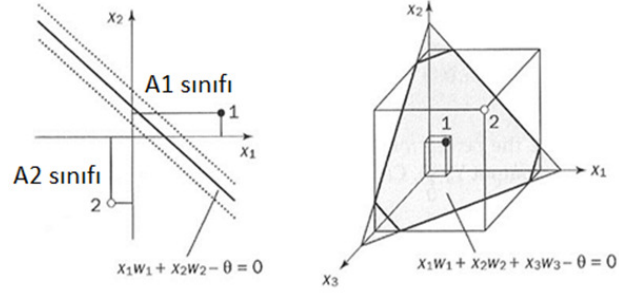
3.1.2 Yapay sinir ağı türleri

Sadece giriş ve çıkış katmanlarından oluşan sinir ağları, tek katmanlı sinir ağları adını alırlar. En basit tek katmanlı sinir ağı modeli perseptrondur³. Perseptron modeli, eğitilebilen tek bir yapay sinir hücresinden oluşur. Bu modelde ağırlıklandırılmış girişler aktivasyon fonksiyonuna uygulanır ve +1 ya da -1 şeklinde bir çıkış alınır. Şekil 3.3’te verilen perseptronun amacı girişleri sınıflandırmaktır.

¹ Aktivasyon fonksiyonu, YSA çıkışını hesaplamak için yapay sinir hücresine gelen net girişin verildiği fonksiyondur.

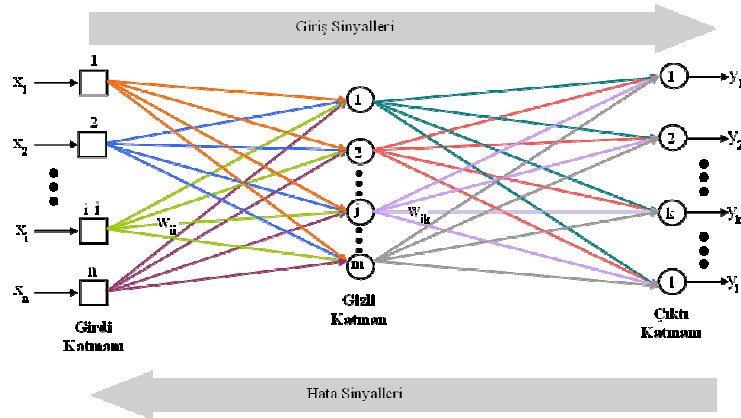
² S harfine benzeyen bu fonksiyon EK E.1’de, bu fonksiyonun tanımı EK E.2’de ilk sırada verilmiştir.

³ Perseptron (perceptron) algılayıcı olarak ifade edilir ve en basit ileri beslemeli sinir ağı türü (doğrusal bir sınıflandırıcı) olarak görülebilir.



Şekil 3.3 : N boyutlu uzayın bir doğru ya da düzlem ile iki bölgeye ayrılması.

Şekil 3.4’te verilen çok katmanlı sinir ağları, bir ya da daha fazla gizli katmanı olan ağlardır. Gizli katmanlara, giriş katmanından gelen genellikle işlenmemiş sinyallerin özelliklerini belirlemek ve bu sinyalleri ağırlıklandırarak sonuçları çıkış katmanına yönlendirmek amaçlı gerek duyulur. Gizli katmanlar istenilen çıkış değerini saklarlar ve bu katmanlardaki sinir hücreleri, ağın giriş ve çıkış davranışlarının incelenmesi yoluyla gözlemlenemez.



Şekil 3.4 : Çok katmanlı sinir ağları örneği.

Öğrenebilme, yeni bilgiler üretebilme gibi yeteneklerin otomatik olarak gerçekleştirilebilmesi amacı ile geliştirilen bilgisayar sistemleri olan YSA, doğrusal olmayan özelliklere sahiptirler ve makine öğrenmesi gerçekleştirirler. Eksik bilgi ile çalışabilmeleri, belirsiz olan ve tam olmayan verileri işleyebilmeleri en dikkat çeken özellikleri arasındadır. Ani bozulma göstermeyen YSA’nın hata toleransları (işlemlerde izin verilen hata sınırı) olması, dereceli bozulma göstermelerini sağlar.

3.1.3 Yapay sinir ağı modelleri

İşlemlerin girişlerden çıkışlara doğru ilerlediği ve gecikmelerin olmadığı ileri yayımlı (ileri beslemeli) YSA ve çıkışların girişlere bağlanarak ileri yayımlı ağıdan

elde edilen ve gecikmelerin olduđu Geri Yayılımlı (GY, geri beslemeli) YSA iki temel yapay sinir ađı modelidir.

YSA, problemleri çözmek için geleneksel hesaplama yöntemlerinden farklı bir yol takip eden bir yaklaşımdır. Geleneksel yöntemler nasıl çözüleceđi bilinen problemlerin çözümünde kullanılırken bu algoritmalarda takip edilmesi gereken belirli adımların bilinmemesi problemin çözümüne engel olur. Ancak YSA'nın sözü geçen güçlü yetenekleri ile bazı durumlarda nasıl çözüleceđi tam olarak bilinemiyen problemleri çözebilmesi mümkündür.

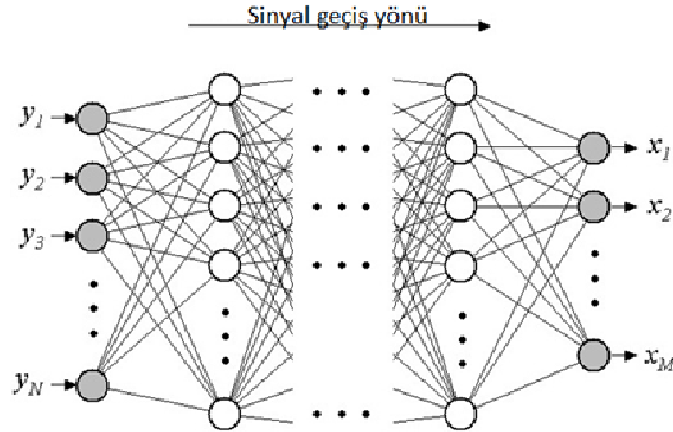
Geleneksel sınıflandırıcılardan olmayan YSA, çok büyük boyutlarda veri içeren çeşitli uygulamalar için güçlü bir YZ yöntemidir. Akıllı sınıflandırma yapılarından olan YSA yönteminin olumlu yönleri şunlardır:

- YSA'nın öğrenebilir olması, hiç karşılaşmadığı sorunları çözebilmesi ve insan zekâsının çalışma şekline uygun modellenmesi oldukça önemli üstünlüklerindedir.
- YSA, yüksek başarımlı orana sahiptir ve bu başarımlı orana bađlı olarak çok kullanılan bir yöntemdir.
- YSA, istatistiksel ve yapısal yaklaşıma alternatif sađlayan, örüntü tanımaya öğrenme boyutu katarak akıllı tanıma niteliđi kazandıran bir tekniktir.
- YSA'nın birbirine bađlı paralel işlem elemanlarından oluşması veriyi hızlı işleyebilmesini sađlar. Bu durum YSA'ya özellikle endüstriyel hayatta büyük öneme sahip gerçek zamanlı çalışma kabiliyeti kazandırır.
- YSA'nın ortak noktası, bir müdahale yapılmaksızın elinde bulunan bilgilere göre sonuç üretebilmesidir. YSA, eğitildikten sonra kullanılacağı sistem için bir veri tabanıdır.
- YSA, öğrenme işlemi esnasında verilen bilgiler ile kendini düzenleyerek daha sonraki girişler için dođru kararlar verebilme yeteneđine sahiptir.
- YSA, gürültülü, eksik, zarar görmüş, nesnenin tanımının belirsiz olduđu veri kümeleri üzerinde uygulanabilir.
- YSA'nın veriyi öğrenebilmesi gibi veriye uyarlanma, veriyi genelleme, kümeleme ve düzenleme gibi yetenekleri de vardır.

Öte yandan, YSA'nın olumsuz bazı yönleri ise şunlardır:

- Sinir ağları kullanıldığında iyi sonuçlar elde edilmesine rağmen işlem yükünün çok fazla arttığı görülmektedir.
- Farklı sistemlere uygulanması zor olabilir.
- YSA uygulamaları ile elde edilen sonucun nereden, nasıl çıkarıldığı bilinmemesi, algoritmaların eğitimlerinin zaman alıcı ve zor olması, YSA'nın olumsuz yönleri arasındadır.
- Donanım bağımlıdır.
- Uygun ağ yapısının ve ağın parametrelerinin belirlenmesinde belli bir kural yoktur.
- Öğrenilecek problemin ağa gösterilmesi önemli bir problemdir.
- Ağın eğitiminin ne zaman bitirilmesi gerektiğine ilişkin belli bir yöntem yoktur.
- Ağın davranışları açıklanamamaktadır.

YSA, sınıflandırma, örüntü tanıma, tahmin, kümeleme, bellek benzetimi, robot kontrolü, saldırı tespit sistemleri, virüs tespiti, kümeleme ve diğer pek çok farklı iş için uygulanabilmektedir. YSA'nın genel yapısı Şekil 3.5'te görülmektedir.



Şekil 3.5 : Bir yapay sinir ağının genel yapısı.

Çözümleri için uygun algoritmaları bulmanın zor olduğu birçok farklı problem vardır. Bazı problemler geleneksel yöntemlerle kolayca çözülemeyebilirler. Hatta henüz çözümleri bile olmayan problemler de vardır. Pek çok benzer problem için büyük bir değer aralığında daha iyi sonuçlar gösteren sinir ağları kullanılabilir.

YSA ile basit biyolojik sinir sisteminin çalışma şekline benzetimi yapılan sinir hücreleri çeşitli şekilde birbirine bağlanarak ağı oluştururlar.

Biyolojik sistemlerde öğrenme, sinir hücreleri arasındaki sinaptik¹ bağlantıların ayarlanması ile olur. İnsanlar doğumlarından itibaren bir yaşayarak öğrenme sürecine girerler, tecrübelerle bu bağlantılar ayarlanır ve hatta yeni bağlantılar oluşur. Bu sayede öğrenme gerçekleşir. YSA için de geçerli olan bu durum, sinir ağını örnekler kullanarak eğitime yoluyla gerçekleştirilir. Eğitim sonucunda öğrenme, eğitim algoritmasının giriş ve çıkış verilerini kullanarak bağlantı ağırlıklarını bir yakınsama sağlanana kadar tekrar tekrar ayarlaması ile olur.

YSA, ağırlıklandırılmış şekilde birbirine bağlanmış ve sinir hücresi olarak adlandırılan birçok işlem biriminden oluşan matematiksel sistemlerdir. YSA, her biri büyük bir problemin bir parçası ile ilgilenen çok sayıda basit işlem biriminden oluşmaktadır. En basit şekilde bir işlem birimi, bir girişi bir ağırlık kümesi ile ağırlıklandırır, doğrusal olmayan bir şekilde dönüşümünü sağlar ve bir çıkış değeri oluşturur. İşlem birimlerinin çalışma şeklinin basit olmasına rağmen toplam işlem yükünü paylaşan işlem birimlerinin birbirleri arasındaki yoğun bağlantı yapısı, sinirsel hesaplamanın gücünü ortaya koymaktadır. Bu sistemlerde, takip eden ikinci madde başlığında anlatılan GY yaklaşım ile daha sağlıklı öğrenme sağlanmaktadır.

YSA'nın öğrenme esnasında karşılaşmamış olduğu örüntülere genelleştirme gösterebilmesi bir başarı ölçütüdür. Bununla birlikte ağ karmaşıklığı ve öğrenme zamanı da göz önüne alınması gereken durumlardandır. İşlemcilerin karmaşıklığı, giriş çıkış bağlantı sayıları, doğrusal olmayan fonksiyonlar, işlem ve bellek duyarlılığı ağ karmaşıklığına neden olan durumlardır. Ayrıca genelleştirme yeteneği kadar öğrenme zamanı da yine YSA için dikkate alınması gereken durumlar arasındadır.

3.1.4 Yapay sinir ağlarında öğrenme algoritmaları

Öğrenme algoritmaları danışmanlı, danışmansız ve destekleyici olarak sınıflandırılır. Danışmanlı (öğreticili) öğrenme için YSA ilk olarak eğitilir. Girişler ve eğitim

¹ Sinaptik sinapsa ait olan demektir. Sinaps, sinir sinyallerinin bir sinir hücresinden diğer sinir hücresine iletiildiği, iki sinir hücresinin birbiri ile bağlantı kurduğu temas yerleridir.

esnasında bu girişlere karşılık oluşturulması gereken çıkışlar ilgili sisteme verilir. Bu sayede girişler ve çıkışlar arasındaki ilişkiler öğrenilebilir.

Destekleyici (pekiştirmeli) öğrenmede her giriş kümesi için üretilmesi gereken çıkış kümeleri sisteme verilmez. Sistemin girişlere karşılık çıkış üretmesi beklenir ve çıkışların doğru ya da yanlış olduğunu gösteren sinyaller üretilir. Bu sinyaller dikkate alınarak eğitim süreci devam ettirilir.

Danışmansız (öğreticisiz) öğrenmede ise sisteme sadece giriş değerleri verilir ve parametreler arasındaki ilişkiyi sistemin kendi kendine öğrenmesi beklenir.

Öğrenme kuralları çevrim içi ya da çevrim dışı (off-line) olabilir. Sistemin öğrenmeye devam ederken kullanılması çevrim içi öğrenme, sistemin eğitildikten sonra kullanılması ise çevrim dışı öğrenme olarak ifade edilir.

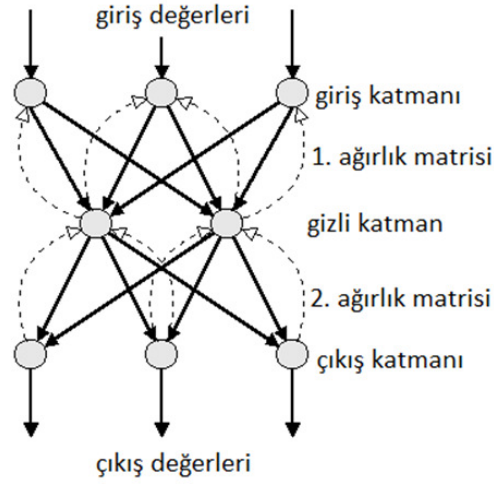
3.1.5 Geri yayımlı yapay sinir ağları

Yapay sinir ağı öğrenme algoritmalarından biri olan geri yayılım algoritması, en yaygın sinir ağı öğrenme algoritmasıdır. GY YSA, her biri bir sonraki katmana bağlı bir veya daha fazla katman içerebilir. İlk katman başlangıç girişlerini alan giriş katmanı olarak adlandırılır ve bu girişler bir görüntüdeki piksel değerleri olabilir. Son katman genellikle girişin tanımlayıcısını tutan çıkış katmanıdır. Giriş ve çıkış katmanları arasındaki katmanlara gizli katman denir. Gizli katmanlar önceki katmanın çıkışlarını bir sonraki katmana yayar ve takip eden katmanın hatasını bir önceki katmana geçirir. Bu işlemler bir yapay sinir ağını eğitmenin birkaç adımdan oluşan temel işlemleridir.

Tipik bir GY yapay sinir ağının eğitimi iki aşama gerektirmektedir. İlk aşamada girişler ağa uygulanarak ileri yayılır ve her çıkış düğümü için çıkışlar hesaplanır. Sonra, verilen girişler için bu çıkışların her biri istenilen çıkışından çıkarılır. Bu değerler her çıkış düğümü için bir hata değeridir. İkinci aşamada bu çıkış hatalarının her biri geriye geçirilir ve ağırlıklar güncellenir. Bu iki aşama, çıkış hatalarının kareköklerinin toplamı kabul edilir bir değere (örneğin $e \leq \%1$) ulaşana kadar devam eder. Bu durum, sinir ağının başarılı bir şekilde eğitildiği ya da tekrarlama (iterasyon) sayısına ulaşıldıysa sinir ağının eğitiminin başarılı olmadığı anlamına gelmektedir.

Sinir ağı mimarisi olarak eğitim algoritmaları ve bu algoritmaların uygulanabileceği mevcut çeşitli örnekler vardır. Çok katmanlı GY YSA, çok geniş kapsamda çeşitli problemler (tahmin problemi gibi) için uygulanabilir.

GY ağlarda aktivasyon fonksiyonunun türevi de kullanıldığından hesaplamaların yavaşlamaması için türevi kolay hesaplanır bir fonksiyon seçilir. GY bir sinir ağı örneği Şekil 3.6'da görülmektedir.



Şekil 3.6 : GY bir yapay sinir ağının genel yapısı.

Geri yayılımı olmayan ileri yayımlı ağlar için çok kullanışlı olan ve danışmanlı öğrenme yöntemlerinden biri olan GY sinir ağlarının, eğitim kümesindeki herhangi bir giriş için istenilen çıkış değerini hesaplayabilmesi gerekir.

GY öğrenme ile elde edilen yakınsama çok yavaştır ve garantili değildir. GY sinir ağları genellikle bir giriş katmanı, bir gizli katman ve bir çıkış katmanından oluşan çok katmanlı perseptronları gerektirir. Çok katmanlı ağlar için doğrusal olmayan aktivasyon fonksiyonları gerekir. Sadece doğrusal aktivasyon fonksiyonları kullanan çok katmanlı bir sinir ağı, bazı tek katmanlı doğrusal sinir ağları ile örtüşmektedir.

3.1.6 Yapay sinir ağlarının eğitilmesi

Çeşitli öğrenme algoritmalarıyla hatanın azaltılarak istenen çıkışlara yaklaşılmaya çalışılması, bu süreçte yapay sinir ağının ağırlıklarının güncellenmesi ile olur. Giriş çıkış çiftleri ile amaca ulaşıncaya kadar ağırlık değerleri saklanır. Ağırlıkların sürekli güncellenerek istenilen sonuca ulaşılması süresine öğrenme zamanı denir.

Ağın öğrenme süreci tamamlandıktan sonra, ağa daha önce verilmeyen girişler verilerek sinir ağı çıkışlarının gerçek çıkış değerlerine yaklaşımları incelenir. İstenen, ağın yeni verilen örneklere de yaklaşabilmesidir.

3.1.7 Bir yapay sinir ağının çalışma prosedürü

Bir yapay sinir ağı için örnek çalışma prosedürü şu adımlarla ifade edilebilir:

- Örneklerin belirlenmesi
- Giriş ve çıkış sayılarının belirlenmesi (ağın topolojisi)
- Öğrenme katsayısı ve sabitlerin belirlenmesi (ağın öğrenme parametreleri)
- Ağın başlangıç değerlerinin atanması
- Öğrenme kümesinden bir örneğin ağa gösterilmesi
- Hatanın hesaplanması
- Hata değerine göre ağırlıkların güncellenmesi
- Veri kümesindeki tüm girişlerin belirlenen süreç tekrarı kadar ağa öğretilmesi
- Süreç tekrarları bitince sistemin toplam hatasının hesaplanması

3.1.8 Yapay sinir ağlarının kullanım alanları

YSA'nın pek çok kullanım alanından bazıları ve bu alanlardaki birtakım uygulamaları şunlardır:

- Eğitim
- Finans (bütçe kestirimi, hedef belirleme)
- Endüstri (ürünlerin kalite kontrolü, müşteri tahmin analizleri)
- Eğlence (animasyonlar, özel efektler)
- Güvenlik (yüz eşleme)
- Sağlık (göğüs kanseri erken teşhisi)
- Uzay (uçuş benzetimleri, otomatik pilot uygulamaları)
- Otomotiv (yol koşullarına göre sürüş analizi, otomatik yol izleme ve rehberlik)

- Bankacılık (kredi uygulamaları geliştirilmesi, müşteri analizi, kredi müracaat değerlendirilmesi, bütçe yatırım tahminleri)
- Savunma (hedef seçme, radar sistemleri)
- Askerî (mayın arama)
- Sinyal işleme
- Görüntü işleme (hastalıkların görüntülerden teşhisi)

3.1.9 Yapay sinir ağları kullanılarak yapılabilecek çalışmalar

- Sınıflandırma, yorumlama
- Örüntü, karakter, el yazısı tanıma
- Veri filtreleme, ilişkilendirme ve yorumlama
- Robotik, kontrol
- Görüntü işleme
- Teşhis, tahmin
- Optimizasyon uygulamaları, öngörü modellemesi

3.1.10 Yapay sinir ağlarının kullanımı

Herhangi bir problemin çözümü için yeterli, uygun ve verimli çözüm yöntemi ya da yöntemlerinin olması veya ilgili olay hakkında örneklerin bulunmaması, bu problemin çözümünde YSA'nın kullanımını gerektirmez.

Sadece YSA ile çözüm üretilebilecek bir problemin çözümü, problemlerin mevcut çözüm yollarına rağmen YSA'nın daha etkin ve kolay çözümler üretebilecek olması gibi durumlar YSA'nın kullanımını gerektirirler.

3.2 Genetik Algoritmalar

Şekil 3.7'de arama yöntemleri içindeki yeri görülen GA, doğada gözlemlenen evrimsel sürece benzer bir şekilde çalışan arama ve optimizasyon yöntemidir. Karmaşık ve çok boyutlu arama uzayında en iyinin hayatta kalması ilkesine göre bütünsel en iyi çözümü arar. Diğer optimizasyon yöntemlerinde olduğu gibi problemlere tek bir çözüm üretmek yerine farklı çözümlerden oluşan bir çözüm

kümesi üretir. Böylelikle arama uzayında aynı anda birçok nokta değerlendirilmekte ve sonuçta bütünsel en iyi çözüme ulaşma olasılığı artmaktadır. Çözüm kümesindeki çözümler birbirinden tamamen bağımsızdırlar. Çözümlerin her biri çok boyutlu uzay üzerinde bir vektördür. Çözüm kümesi problem için bir noktada birleşir ve GA'nın kümedeki en iyi elemanı arayıp bulması istenir.

GA, diğer yöntemler kullanılırken büyük zorluklarla karşılaşılan oldukça büyük arama uzayına sahip problemlerin çözümünde başarı göstermektedirler. Bir problemin bütünsel en iyi çözümünü bulmak için garanti vermezler, ancak problemlere makul bir süre içinde kabul edilebilir iyi çözümler bulurlar. GA'nın kullanımı, uzun matematiksel işlemler ve yoğun bellek kullanımı gerektiren zor problemlerin çözümü için oldukça uygun olabilir.

GA, arama uzayının büyük ve karmaşık olduğu, mevcut bilgiyle sınırlı arama uzayında çözümün zor olduğu, problemin belirli bir matematiksel modelle ifade edilemediği, geleneksel optimizasyon yöntemlerinden istenen sonucun alınmadığı durumlarda etkili ve kullanışlıdır.

GA, parametre ve sistem tanımlama, kontrol sistemleri, robot uygulamaları, görüntü ve ses tanıma, mühendislik tasarımları, YZ uygulamaları, uzman sistemler; fonksiyon ve birleşimsel optimizasyon problemleri, ağ tasarım problemleri, yol bulma problemleri, sosyal ve ekonomik planlama problemleri için diğer optimizasyon yöntemlerinin yanında başarılı sonuçlar vermektedir.

GA problem için olası pek çok çözümü temsil eden bir küme oluştururlar. Bu kümeye popülasyon ya da nüfus denir. Nüfustaki her bir eleman kromozom veya birey olarak adlandırılan sayı dizilerinden oluşur. Her kromozom olası bir çözümü ifade etmektedir. Kromozomları oluşturan her bir eleman çözümün bir özelliğidir ve gen ya da bit olarak adlandırılır. Kümedeki bireyler evrimsel süreç içinde GA'nın işlemcileri tarafından belirlenirler.

Problemin bireyler içindeki gösterimi problemden probleme değişiklik gösterir. GA'nın problemin çözümündeki başarısına karar vermedeki en önemli faktör, problemin çözümünü temsil eden bireylerin gösterimidir.

Nüfus içindeki her bireyin problem için çözüm olup olmayacağına karar veren bir uygunluk fonksiyonu vardır. Yeni bireyler elde edilirken daha iyi uygunluk değerine sahip bireylerin oluşturulması beklenir. Probleme göre uygunluk değerinin artması

ya da azalması daha iyi bir uygunluk değerine gidış olarak değeriendirilebilir. Yeni bireyler işlemlerle elde edilir ve bu amaçla seçilecek bireyler önemlidir.

Yeni nüfus, bir önceki nüfusta yer alan uygunluğu yüksek bireylerin bir araya gelip çoğalmalarıyla oluşur. Aynı zamanda bu nüfus, önceki nüfusun uygunluğu yüksek bireylerinin sahip olduğu özelliklerin büyük bir kısmını içerir. Böylelikle pek çok nesil aracılığıyla iyi özellikler nüfus içerisinde yayılırlar ve genetik işlemler aracılığıyla da diğer iyi özelliklerle birleşirler. Uygunluk değeri daha iyi olan ne kadar çok birey bir araya gelip yeni bireyler oluşturursa, arama uzayı içerisinde o kadar iyi bir çalışma alanı elde edilir. Probleme ait en iyi çözümün bulunabilmesi için üç önemli duruma dikkat edilmelidir:

1. Bireylerin gösterimi doğru bir şekilde yapılmalı,
2. Uygunluk fonksiyonu etkin bir şekilde oluşturulmalı,
3. Doğru genetik işlemler seçilmelidir.



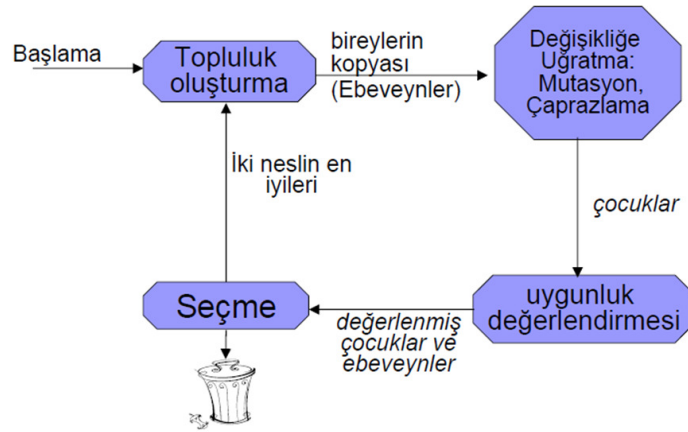
Şekil 3.7 : Arama yöntemleri içerisinde GA'nın yeri.

Evrimsel süreçte olduğu gibi GA için de seçim söz konusudur. İyi örneklerden daha iyi örnekler oluşturulması ve kısa sürede iyi sonuçlar elde edilebilmesi GA'nın temel özellikleri arasındadır. GA, amaç fonksiyonunun türevlerini ve ek bilgileri matematiksel bir taban üzerinde kullanmak yerine doğrudan amaç fonksiyonunu kullanırlar ve rasgele geçiş kuralları içerirler. GA ile çözümün evrimleştiği söylenebilir.

Bilgisayar biliminde bilgi işleme (berim), problemlerin modellenmesi ve çözülebilmesi için gerekli uygulamaların bir bütünüdür. Pratikte sayısal bilgi işleme

çoğu zaman doğal süreçlerin benzetimi için kullanılır. Bu doğal süreçlere örnek olarak Şekil 3.8’de görülen evrimsel bilgi işleme (evrimsel berim) verilebilir.

Bilgisayar biliminde bir problemin çözümünü ya da çözümlerini çok miktarda olası çözüm adayının arasından bir değerlendirme yaparak bulma işlemine arama denir. Arama, bilginin işlenerek yeni bilgilerin elde edilmesinde kullanılacak yöntemlerden biridir. Arama için evrimsel bilgi işleme alanındaki GA’dan yararlanılabilir. Bu yöntemlere düzenleme, saklama, sorgulama ve işleme gibi uygulamalar da örnek verilebilir.



Şekil 3.8 : Genel evrimsel bilgi işleme döngüsü.

Evrimsel bilgi işleme, sayısal optimizasyon, otomatik programlama, makine öğrenmesi, pazarlık ve açık artırma stratejilerinin geliştirilmesi, yaratıcı girişimin modellenmesi gibi uygulamalarda ve ekonomi alanında kullanılmaktadır. Evrimsel bilgi işleme kapsamındaki GA, biyolojik evrim sürecinden faydalanmaktadır ve GA'nın temel özellikleri şunlardır:

- Kalıtım, GA'nın çözüm uzayındaki bireylerinin ve özelliklerinin sonraki nesillere aktarılması ve temel genetik özelliklerin ortaya çıkmasıdır.
- Seçim, seçilen bireylerden çaprazlama (çeşitlilik) ve mutasyon (değişim) ile farklı bireyler ortaya çıkmasıdır.
- Seçim işlemcileri genellikle tüm nüfusu dikkate alır. Ayrıca uygulamaya göre değişiklik göstererek söz konusu nesil içinden en iyi bireyleri seçmeye çalışabilir.
- Nesildeki çeşitlilik farklı çözümlerin varlığının bir ölçüsüdür.
- Birkaç gen canlının karakteristiğini etkiler.

Evrimsel bilgi işlemede ilk nesil rasgele yaratılır ve ilk neslin tüm bireyleri değerlendirilir. Ardından, seçilen birey çiftlerinin çaprazlanması, ortaya çıkan çocuklara mutasyon uygulanması ve yeni neslin değerlendirilmesi gerçekleştirilir. Ebeveynler (anne ve baba) ile çocuklardan yeni neslin oluşturulması ile kabul edilir bir sonuca varılana ya da önceden belli tekrarlar sayısına (nesil sayısı) ulaşılan kadar bu sürece devam edilir. Üzerinde işlemler yapılmayan bireyler ebeveyn bireyler, ebeveyn bireylere bu işlemlerin uygulanması ile elde edilen bireyler ise çocuk bireylerdir.

3.2.1 Genetik algoritmaların diğer yöntemlerden farkları

- GA, problemlerin çözümünü parametrelerin değerleriyle değil kodlarıyla arar. Parametreler kodlanabildiği sürece çözüm üretilebilir. Bu sebeple GA ne yaptığı konusunda bilgi içermez, ama nasıl yaptığını bilir.
- GA aramaya tek bir noktadan değil, noktalar kümesinden başlar. GA'nın aday çözümleri içeren küme yaklaşımı ile çoğunlukla yerel en iyi çözümle (en iyi çözüm kümesinin bir kısmı) sınırlı kalınmaz.
- GA matematiksel fonksiyonlar yerine uygunluk fonksiyonlarının değerlerini kullanır. Bu değerlerin kullanılması ayrıca yardımcı bir bilginin kullanılmasını gerektirmez ve süresizlik noktalarında da çözüm üretilebilir.
- GA gerekirci kuralları değil, olasılıksal kuralları kullanır.
- GA paralel arama yapan bir tekniktir ve çeşitli işlemlerle elde edilen verilerden uygunluk değeri daha iyi bireyler elde edilmeye çalışılır.
- GA'nın yeni bireylerin yaratılmasında rol oynayan işlemcileri rasgele yapıdadırlar. Bu yapıları, uygulamalarda yerel en iyi çözümle sınırlı kalınmasına engel olurlar.

3.2.2 Genetik algoritmaların temel teoremi

GA'nın nasıl arama yaptığını öğrenmek için çözüm adaylarını kodlamak amacıyla kullanılan dizileri ve bu dizilerin içerdiği alt dizileri öğrenmek gerekir. Örneğin H dizisi beş boyutlu bir kromozom olmak üzere 01010 şeklinde genlerden oluşuyor olabilir. Genlerin ikilik sistemde ifade edildiği bu gösterimde, bu kümenin bir altkümesi (örneğin 101) H dizisi için bir alt dizidir. Onluk sistemdeki istenen

çözüm değerlerinin ikili sistemdeki gösterimi GA için kullanılıyorsa, işlemler bu dizideki elemanlar üzerinde yapılır. Bu yapılar olası çözümlerdir.

GA, farklı sonuçları eş zamanlı değerlendirerek ve arama eğilimini daha iyi çözüm alanlarına yönlendirerek bir döngü (çevrim) içerisinde durdurma ölçütü sağlanana kadar devam eder. İşlemciler sayesinde kötü bireyler elenir.

GA için gösterim sayısal dizinler şeklindedir (Örneğin ikili karakterler dizisi şeklinde: ...100110...). Probleme uygun gösterimin ve bu gösterime uygun değişim işlemcilerinin (çaprazlama ve mutasyon) seçilmesi önemlidir. Seçim işlemcileri sadece uygunluğu değerlendireceğinden gösterimden bağımsızdır.

Bireyler arasında aday çözümler bulunmaktadır ve en iyi ya da en iyiye yakın çözüm veya çözümleri bulabilmek için her türlü olası çözümün probleme uygun gösterimlerde temsil edilmesi gerekir.

Hayatta kalanların seçilerek (diğer bireyler elenir) yeni neslin oluşturulması farklı şekillerde olabilir:

- Ebeveynlerin ve çocukların uygunluğa göre sıralanıp en iyilerin nesil yapılması ve kalanların elenmesi uygunluk tabanlı bir yöntemdir.
- Çocukların ebeveyn sayısı kadar yaratılması ve tüm ebeveynlerin elenerek sadece çocuklardan nesil oluşturulması yaş tabanlı bir yöntemdir.
- Ebeveynlerin ve çocukların en iyilerinin karışımı olarak yeni neslin oluşturulması en iyilerin seçilmesi yöntemidir.

Genetik algoritma uygulamalarında başlangıç ve bitiş durumları önemlidir:

- Başlangıç genellikle rasgele yapılır.
 - Başlangıç çözüm kümesi problemin en iyi çözümünü aramak için rasgele dağıtılmış genetik özellikler içeren bireylerden oluşmaktadır. İşlemciler kullanılmadan oluşturulan başlangıç nüfusu, belirlenen birey sayısı kadar kromozom içerir.
 - Problemin yapısına ve içeriğine göre başlangıç çözümündeki birey sayısı belirlenebilir.

- GA'nın başlangıç nüfusunun tamamen rasgele olması, kimi zaman yerel çözümlerde kalınmasına kimi zaman çözümden uzaklaşılmasına neden olabilir.
- Çözümden uzak çok sayıda kromozom içeren başlangıç kümesine mevcut veriden hareket edilerek bilinen çözümler ya da özellikler dâhil edilebilir. Böylece, başlangıcın tamamen rasgele yapısının önüne geçilerek çözüme ulaşmanın zorluğu ve çok zaman alması ortadan kaldırılabilir.
- Bitiş koşulu her nesilde kontrol edilir.
 - Bilinen ya da istenen (umulan) uygunluk değerine erişilmesi
 - Kararlaştırılan nesil sayısına ulaşılması
 - Çeşitliliğin belirli değer altına düşmesi
 - Belli nesil geçmesine rağmen uygunluk değerinin değişmemesi

Genetik algoritma uygulamalarında probleme uygun olarak nesil sayısının belirlenmesi önemlidir. Son uygunluk değerlerinin ne kadar önemli olduğu ile nesil sayısı doğru orantılıdır. İyi çözümler başlangıcın rasgele ya da akıllıca yapılmasına ve koşullara, başlangıç ise probleme bağlıdır. Dikkat gerektiren bu durum, uygulama sürecinde farklı sonuçlar ortaya çıkarabilir.

3.2.3 Yeniden üretim

Yeniden üretim için yaygın olarak kullanılan yöntemlerden biri rus ruleti yöntemidir. Bu yöntemde en iyi bireylerin bir sonraki nesle seçilme olasılıkları daha yüksektir. Yeniden üretim için kötü çözümler de seçilebilir, çünkü kötü çözümler ilerde daha iyi çözümlerin oluşmasını sağlayabilirler.

Yeniden üretim için Şekil A.4'te (şekli takip eden kısımda içeriği) verilen rulet tekeri yöntemi kullanılabilir. Bu seçimde kromozomlar uygunluk değerlerine göre bir rulet etrafına gruplanırlar. Uygunluk değerleri herhangi bir ölçüte uyan bireylerin seçilmesi için kullanılan bu yöntem ile daha büyük alana sahip (uygunluk değeri daha iyi ya da kapladığı alan daha çok olan) bireyin seçilme şansı daha büyük olur. Bu yöntemde bireylerin seçilme olasılığı, her birey için o bireyin uygunluk

fonksiyonu deęerinin tüm bireylerin uygunluk fonksiyonu deęerlerinin toplamına oranı kadardır. Birey seçimi rulet üzerinden rasgele yapılır.

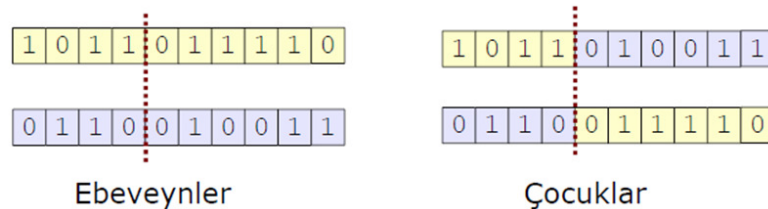
Uygulanması daha kolay olan turnuva seçim yönteminde ise iki birey nesilden seçilir. Uygunluk deęeri daha büyük olan birey işlemlere tabi tutulur. Bu yöntemin üstünlüğü, herhangi bir bireyin süreç esnasında kaybedilme olasılığının rulet tekerleęi yöntemine göre daha az olmasıdır.

Genetik algoritma işlemcilerinin uygulanacağı bireylerin seçimi tamamen rasgele olabileceęi gibi bu madde başlığında bahsedilen yöntemlerle de olabilir. Bu yöntemler seçim işlemcileridir ve gelecek nesle aktarılacak yeni bireyleri oluşturacak ebeveyn bireyleri seçmede kullanılırlar. Kromozomların uygunluk deęerlerine göre yapılan bu yöntemlerle seçilen bireylerin iyi olma olasılığı artar. Uygunluk deęerleri daha iyi olan bireylerden ortaya çıkacak yeni bireyler daha iyi olabilir.

3.2.4 Genetik algoritma işlemcileri

Genetik işlemcileri, rasgele sayı üretimi, algoritmaların uygulanması için kullanılan veri yapıları üzerinde kopyalama, kısmi dizi alışverişleri ve gen deęişimlerini gerektirir. GA'nın gerçek üstünlüğü işlemcilerinin ortak etkisinden kaynaklanmaktadır. Başlangıçta rasgele olarak atanan çözümler daha sonra GA'nın çalışma yöntemlerine göre iyileştirilmektedirler.

İki ebeveyn bireyin rasgele belirlenen bir çaprazlama noktasından kromozomlarının bölünerek yarı parçalarının deęiştirildięi ve çocukların elde edildięi çaprazlama işlemi Şekil 3.9'da görülmektedir. Çaprazlama, çözüm kümesindeki kromozomların ikişer ikişer birleştirilerek yeni çözümlerin elde edilmesi işlemidir. Çaprazlama işlemi yapılacak iki kromozomun bölünme noktalarından itibaren genleri karşılıklı olarak deęiştirilir.



Şekil 3.9 : Çaprazlama noktası dört olan bir çaprazlama örneęi.

Kromozomlar birden fazla çaprazlama noktasından bölünebilirler. Daha sonra, parçalar bir deęiştirilip bir deęiştirilmeden çaprazlama gerçekleştirilebilir.

Çaprazlama olasılığının sağlanması ($r \leq op$) ile nüfustan rasgele seçilen kromozomlar arasındaki gen alışverişleri ile daha iyi bireylerin elde edilmesi amaçlanmaktadır.

Her elemanı bağımsız olarak belli bir olasılıkla değiştirilen bir birey için mutasyon örneği Şekil 3.10'da görülmektedir. Örneğin 1001 gen birimine sahip bir kromozom için mutasyon olasılığı %0,1 olabilir. Mutasyon, çeşitlendirmede faydalıdır ve en iyi çözüme ulaşmada gereklidir. Şekil 3.10'daki örnekte olasılık sağlandıkça 1 olan genler 0, 0 olan genler 1 yapılır.

ebeveyn	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Çocuk	0	1	0	0	1	0	1	1	0	0	0	1	0	1	1	0	0	1

Şekil 3.10 : Mutasyon örneği.

Bu işlemcilerin Şekil 3.9'da ve Şekil 3.10'da görüldüğü gibi farklı rolleri vardır. Çaprazlama ve mutasyon probleme bağlı olarak gerçekleştirilir.

Bireylerin ikilik sistemdeki değerlerden farklı olarak tamsayılar (integer) ya da gerçek (reel: float, double) sayılar şeklinde de gösterimi kullanılmaktadır. Ayrıca uygulamalara her bireyin bir nesil yaşaması, her nesilde bir çocuk yaratılması ve nüfusun bir elemanının değiştirilmesi gibi koşullar getirilebilir.

GA'nın seçim uygulamaları iki farklı seçimi içermektedir. Biri çaprazlama ve mutasyon işlemlerinin uygulanacağı bireylerin seçimi, diğeri ise bir sonraki nesle aktarılan bireylerin seçimidir.

Mutasyon ve çaprazlama işlemleri, bir sonraki neslin seçimi gibi uygulamalar tamamen üzerinde çalışılan probleme uygun olarak icra edilirler ve özel durumları da içerebilirler. İşlemlerin sonlandırılması belli bir uygunluk değerine ulaşmayı veya belirli sayıda döngü gerçekleşmesini gerektirir. Sadece seçim ve çaprazlama ile yerel en iyi noktalara takılmamak için mutasyon işlemcisi gereklidir. Mutasyon, canlılarda radyasyon veya başka sebeplerle değişiklik meydana gelmesinden esinlenerek geliştirilmiş bir işlemcidir. Çaprazlama neticesinde farklı çözümlere ulaşmak bazen zor olmaktadır. Mutasyon, yeni çözümleri aramanın kolaylaştırılması ve aramanın yönünü değiştirmek amacı ile bir kromozomun bir geninin değiştirilmesi işlemidir. Bir problemin çözüm kümesi içinden kaç kromozomun mutasyona uğratılacağına

mutasyon olasılığına göre karar verilmektedir. Kromozomlarda tesadüfi olarak belirlenen bir noktada yapılan, birey üzerinde çeşitliliği koruyan ve çeşitlilik yaratan mutasyon, kromozom genlerinin sırasının değişmesi, kayması, ters dönmesi ya da bazı genlerde değişim olması şeklinde gerçekleşebilir.

Çaprazlama ile eski bireylerden daha iyi bireylerin elde edilmesi beklenir. Seçilen bireyler çaprazlama işlemine tabi tutulmayacaksa gelecek nesle doğrudan aktarılırlar. Ebeveynlerin genlerinin birbirine aktarılması sonucunda oluşan bireyler ebeveynlerden özellik taşırlar. Bu özelliklerin birleşerek oluşturduğu çocuklar çoğu zaman ebeveynlere göre çok daha kötü uygunluk değerlerine sahip olabilirler. Oluşan bireylerin yaşayıp yaşamayacağına GA'nın başarımını olumlu yönde etkileyecek şekilde karar verilir. Kromozomların hangi özelliklerinin iyi özellikler sağlayacağı bilinmediğinden işlemler rasgele yapılır.

GA için temel işlemcilerden olan çaprazlama ve mutasyon ile nüfustaki birey sayısı kadar birey elde edilmelidir. Bu işlemciler kullanılarak elde edilen yeni çözüm adayları ile GA'nın yapısının kötüye gidişi engellenerek hızlı bir şekilde daha iyiye gitmesi sağlanmalıdır.

Yeni bireyleri oluşturmak için seçilecek bireyler tamamen rasgele seçilebildiği gibi belli uygunluk değerlerindeki aralıktan da seçilebilir. Genetik algoritma uygulamaları tasarımında çaprazlama ve mutasyon işlemlerinin kaç bireye ne şekilde ne olasılıklarla uygulanacağı önceden belirlenmelidir.

3.2.5 Uygunluk fonksiyonu

GA'nın olası çözüm adaylarının problem çözümüne uygunluk değerlerini ölçmede kullanılan fonksiyona uygunluk fonksiyonu denir. Olasılığa dayalı olan GA, kullandıkları problemlerde uygunluk (amaç) fonksiyonunun belirlenmesine gereksinim duyarlar. Bireylerin uygunluk değerlerinin hesaplanması, GA'nın en temel yapısı olan uygunluk fonksiyonu ile olur ve uygunluk fonksiyonu problemin yapısına göre değişir. Ayrıca, çeşitli kısıtların olduğu problemler için uygunluk fonksiyonlarında bu kısıtların da ifade edilmesi ve kısıtların sağlanmadığı durumlarda ilgili işlemlerin yapılması çözüme ulaşımı hızlandırabilir.

Probleme özgü olarak uygunluk fonksiyonunun en düşük ya da en yüksek değerleri ya da bu değerlere yakın değerleri vermesi beklenebilir. Belli bir noktadan sonra

uygunluk deęerlerinin iyileřtirilememesi (düşürülememesi ya da yükseltilememesi) GA'nın sonlandırılmasını gerektirir.

GA'nın modellenmesi esnasında uygunluk fonksiyonunun seçimi önemli bir aşamayı oluşturmaktadır. Nesildeki bireylerin istenen özelliklere uyumunu temsil eden uygunluk fonksiyonu, hedef fonksiyon olarak da tanımlanabilir.

Uygunluk deęerlendirmesi, GA'nın uygulanması sürecinde gelişmenin ne olduğunu ifade eder. Bu süreçte çeşitli fonksiyonlar kullanılabilir ve uygunluk deęerinin gittikçe daha iyiye gitmesi arzu edilir.

3.2.6 Seçim

Seçim, bir nesildeki uygunluk deęerleri düşük olan bireylerin elenerek yerlerine uygunluk deęerleri yüksek bireylerin kopyasının konmasıdır. Hangi bireylerin bir sonraki nesle aktarılacağı, seçim ile probleme göre belirlenir.

Seçim aşamasında nesildeki birey sayısı önem taşımaktadır. Nesilde çeşitlilik sağlanabilmesinde yeterli olmayan yöntemler kullanılmamalıdır. Seçim aşamasında topluluk içinde verilerden olabildiğince farklı bireylere ulaşabilmek önemlidir. Bu çeşitliliği sınırlandıracak yöntemlerin uygulanması iyi sonuçlar vermeyebilir.

3.2.7 Genetik algoritmaların kullanım alanları

GA'nın yaygın kullanım alanları ve bu alanlarda uygulandıkları çeşitli problemler şunlardır:

- Farklı bilim dallarındaki optimizasyon problemleri
 - Süreksiz, zor ve gürültülü fonksiyonların optimizasyonu
 - İstenen amaçlara ulaşmada iş gücü, tedarik, zaman gibi sınırlı kaynakların en verimli şekilde tahsis edilmesi
- Endüstri
 - Montaj hattı dengeleme problemi, üretilen ürünlerin toplam birim üretim maliyeti, işçilik maliyetleri ve toplam üretim zamanı gibi ölçütleri dengeleyebilmesini amaçlayan bir uygulamadır.

- Tesis ya da üretim hattı yerleşimi problemleri, kaynakların ya da araç ve gereçlerin belirli kısıtlar çerçevesinde en yüksek başarıyı sağlayacak şekilde yerleştirilmesi kararını içermektedir.
- Üretim stok kontrolü
- YSA
- Görüntü işleme
- Görüntü ve ses tanıma
- Kontrol sistemleri ve uzman sistemler
- Sosyal ve ekonomik planlama problemleri
 - Sosyal sistemlerin evrimsel yönlerini anlama (iş birliği ve iletişimin evrimi, karıncalardaki iz takibinin evrimi)
 - Kazanç getiren pazarların ortaya çıkış süreçlerini modelleme
- Yol bulma ve araç rotalama problemleri
- Yöneylem araştırması
- Ekonomi, muhasebe, ulaştırma, dağıtım
- Atama problemi, n elemanın, n farklı göreve atanması işinin yapılma maliyetini en aza düşürebilmektir. Amaç, bu koşul altında hangi elemanın hangi göreve atandığını gösteren kümenin bulunmasıdır.
- Taşıma problemi, n tüketicinin ürün taleplerinin m tedarikçi tarafından maliyetin en düşük olması şartı ile karşılanabilmesidir. Tek tedarikçiden her bir tüketiciye bir birim ürün ulaştırma maliyeti bilinmektedir. İstenen, bu şartlar altında ürünlerin arz yerinden talep yerine en uygun şekilde tahsisini sağlamaktır.
- İşletmeler
 - Finansal modelleme uygulamaları için kaynak tahsisi, tahmin uygulamaları
 - Pazarlama uygulamaları için tüketici verilerinin analizleri, pazarlama stratejileri uygulanması

- Verimlilik, kârlılık, müşteri tatmini, rekabet edebilme yeteneği, karar alma kalitesi gibi işletmeler için yaşamsal öneme sahip konularda veri madenciliği kullanılmaktadır. Veri madenciliğinde kullanılan bir teknik olan GA ile veri yığınlarından modeller elde edilebilir.
- Graf teorisi
 - En düşük yayılan ağaç problemi, coğrafi bir alanda dağılmış tüm şehirleri fiber optik kablo ağına bağlayan yerleşimi en düşük maliyetle bulabilme problemidir. Yerleşim herhangi iki şehir arasında veri paketinin gidip gelebilmesine olanak tanınmalıdır.
- Mekanik öğrenme
 - Sınıflandırma sistemlerinde kural bulma mekanizması geliştirilebilir. Bu mekanizma bir durum sağladığında ilgili faaliyetin uygulanmasını içermektedir.
 - Robot uygulamaları
- Bilgi sistemleri ve otomatik programlama
 - GA çeşitli görevler için gerekli olan bilgisayar programlarını geliştirmede kullanılabilir. Teslim tarihleri ve işlem süreleri belli işlerin çizelgelenmesi amacıyla GA kullanılabilir.
 - Veri tabanı yönetimi

3.2.8 Genetik algoritmaların kullanımı

GA'nın oldukça genel yapıları herhangi bir probleme uygulanabilir olmalarını sağlar. Problemlerin çözülebilmesi için gösterimlerin, tanımlamaların ve işlemlerin probleme uygun olarak yapılması gerekir.

GA, bulanık mantık, YSA gibi yöntemlerle birlikte kullanılabilirler. İşletmelerdeki uygulamalar için çok büyük veri tabanlarını amaç doğrultusunda hızlı ve etkin kullanabilmek gerekmektedir. GA bu tip durumlar için veri madenciliği gibi teknikler gerektirebilirler. Kısıtlar ve bu kısıtlara yönelik ceza fonksiyonları GA'nın uygulamalarında birtakım sorunlar ile karşılaşılmasına neden olabilmektedir.

3.2.9 Genetik algoritmaların uygulandıđı bir problem

GA'nın en sık uygulandıđı alıřmalardan biri gezgin satıcı problemidir. Gezgin satıcı probleminde amaç, katedilen toplam mesafeyi en aza indiren bir yolculuk planı oluřturmaktır. Birok problem tipi gezgin satıcı problemi gibi modellenebilmektedir. Bunlara rnek olarak devre tasarımı ve posta tařıyıcılarının, havayolu uaklarının, okul otobuslerinin rotalarının bulunması verilebilir.

Gezgin satıcı probleminin bir zelliđi de deđiřken sayısı arttıa stel artıř gsteren zaman srecinde zme ulařtırılabilmesidir. Bu durum bir rnekle řyle aıklanabilir: Bir satıř grevlisinin ziyaret etmek durumunda olduđu n tane řehir olsun. Burada tm řehirler arasındaki en fazla izlenecek rota sayısı $(n - 1)!$ dir. Tm mmkn rotaları basite inceleyen ve en kısa olan rotayı bulan bir algoritma kullanılır. Fakat řehir sayısı arttıa, algoritmanın hesaplama iin gereksinim duyduđu zaman daha da byk bir oranda artmaktadır. Ziyaret edilmesi gereken 25 řehir varsa, algoritmanın inceleyeceđi rota sayısı $24!$ dir. Bu da yaklařık $6,2 \times 10^{23}$ sayısına karřılık gelmektedir. Saniyede bir milyon rota inceleme kapasitesine sahip bir bilgisayar bu problemi, $6,2 \times 10^{17}$ saniyede yani $1,96 \times 10^{10}$ yılda zezebilmektedir.

Herhangi bir problemin zm iin kullanılan algoritmanın en yaygın bařarım lt, algoritmanın zme ulařma sresidir. Gezgin satıcı gibi deđiřken sayısı arttıa zm zamanı stel olarak artan problemlerde bu durum daha da nemlidir. GA, optimizasyon gerektiren byle problemleri klasik yntemlere gre ok daha kısa srede zmektedirler. Sonuta en iyi zme yakın ve kabul edilebilir bir zm bulunmaktadır (bu durum GA'nın bu alıřmada kullanılma sebeplerinden biridir).

stel artıř gsteren bu problem iin deđiřken sayısının artması ile gz nne alınması gereken toplam rota sayısının deđiřimi, 3.2'deki denklem grubunda ok kk bir aralık iin rnek olarak verilmiřtir:

$$\begin{aligned} 1.(3! &= 6) \\ 2.(4! &= 24) \\ 3.(5! &= 120) \end{aligned} \tag{3.2}$$

3.2'de verilen denklem grubunda ilk olarak  řehir sz konusudur. řehir sayısının bir artması ile incelenecek rota sayısı drt katına ve tekrar bir artması ile en son deđerinin beř katına ıkar. Bu sre zm sresini ok byk miktarlarda artırır.

4. GÖRÜNTÜ İÇİNDE GÖRÜNTÜ ARAMA

GİGA, görüntü eşleme ve görüntü arama algoritmalarının bir arada kullanılmasıyla icra edilmektedir. Arama süreci GA kullanılarak gerçekleştirilmektedir. Büyük bir görüntüde küçük bir görüntünün aranması oldukça zaman alıcı (en fazla görüntü eşleme sayısı: S) olabilen bir problemdir ve optimizasyon işlemleri gerektirmektedir. GİGA problemi, en iyi ya da makul seviyedeki çözümlerin elde edilebilmesi için akıllı algoritmalara gerek duyar.

Görüntü arama sürecinde görüntü eşleme için seçilen uygunluk fonksiyonları da büyük önem taşımaktadır. Daha net ve ayrıntılı bir eşleme sağlayabilecek bir yöntem aynı zamanda büyük zaman kaybına neden olabileceği gibi zamandan tasarruf etmek adına yetersiz veri eşlemesiyle de doğru olmayan sonuçlar ortaya çıkabilir.

Görüntü aramalarının doğru bir şekilde yapılabilmesi, eşleme ile aramaların bütünlük içinde icra edilmesini gerektirir ve çeşitli tekniklerle iyileştirmeler sağlanabilir. Arama sürecindeki iyileştirmeler, kod optimizasyonu, paralel programlama teknikleri, akıllı arama algoritmaları ve etkili uygunluk fonksiyonları ile sağlanabilir. İlgili durumlarda birden fazla fonksiyon bütünleşik olarak kullanılabilir.

GİGA çalışmasında özellikle üzerinde durulması gereken hususlar şunlardır:

- Arama işlemlerini kısa sürede gerçekleştirebilmek için şablon görüntü ile istenen oranda eşleşmeyen, ama belli bir değerde ve bu değer üzerinde (belli değerden daha fazla eşleşme) eşleşen (eşleşme derecesi) alt görüntülerden eşleşen görüntülere ulaşmak gerekir. Bu amaçla çeşitli uygunluk fonksiyonlarından yararlanmak gerekir.
- Paralel programlama teknikleri ile arama işlemlerinde yüksek başarımlar elde edebilmek için iş parçacıklarının birbiri ile aynı yere tekabül edeceği durumların tam olarak giderilmesi gerekmektedir.
- GİGA'da GA ve paralel programlama teknikleri kullanımı her ne kadar iyileştirmeler sağlasa da uygunluk fonksiyonlarının seçimi çok büyük önem

taşımaktadır. Uygunluk fonksiyonlarının gerekli durumlarda bütünlük içinde kullanılması gerekebilir.

Bu çalışmada YZ yöntemlerinin kullanılma sebebi, akıllı sınıflandırma algoritmalarının istatistiksel bir yapı üzerine kurulu olmamaları, gürültüye karşı gürbüz olmaları ve hata ölçütlerinden ötürüdür.

Bu çalışmada esas beklenen, erişim sonuçlarının doğru ve yeterli, erişim sürelerinin kısa olmasıdır. Bu da ancak eşleme ve arama algoritmalarının doğru bir şekilde ve bütünlük içinde çalışıyor olması ile gerçekleşebilir.

4.1 Büyük Bir Görüntüde Küçük Bir Görüntü Aramak

Daha büyük bir görüntüde aranacak görüntü şablon görüntü olarak adlandırılır. Şablon görüntü aranmak istenen küçük görüntüdür. Alt görüntü olarak da tanımlanabilen şablon görüntünün yer alıp almadığının ve yer alıyor ise hangi konumlarında yer aldığının tespit edilmesi için taranacak büyük görüntü ise referans görüntü ya da sorgu görüntü olarak adlandırılır. Referans görüntü şablon görüntüyü içeriyor olabilir ya da içermeyebilir, fakat referans görüntünün şablon görüntüyü boyutsal olarak kapsaması gerekir ($M \geq m$ ve $N \geq n$).

Bu çalışma, bir referans görüntünün ya da çok sayıda görüntünün bir şablon görüntüyü içerip içermemesi sorgusuna cevap aranması için gerçekleştirilir. Şablon görüntünün referans görüntü veya görüntülerde tespit edildiği durumlarda, ilgili konumların yatay ve düşey düzlemde karşılık geldiği piksel noktalarının ve eşleme bilgilerinin (eşleşme derecesi gibi) sunulabilmesi gerekmektedir.

Referans görüntü ya da görüntüler tek bir görüntü, bir görüntü klasörü ya da görüntü veri tabanı olabilir. Şablon görüntü, referans görüntülerin kapsadığı herhangi bir görüntü, çizim, taslak ya da referans görüntülerden elde edilen görüntü parçası olabilir. Bu görüntüler bilgisayar ortamına aktarılan ya da bilgisayar ortamında hazırlanan görüntüler olabilirler.

Referans görüntü ile şablon görüntünün aynı boyutlarda olmaması, karşılaştırılacak görüntü sayısının birden fazla olmasına neden olmaktadır. Aynı boyutlarda olan bu görüntüler için tek bir eşleme gerekmektedir ve bu durum tek bir görüntü eşleme olarak ifade edilir. Bu tez çalışmasında gerçekleştirilen uygulamada ise bir görüntünün bir başka görüntüde aranması, çok sayıda görüntünün eşleştirilmesini

gerektirmektedir ve bu durum arama problemi olarak nitelendirilir. Bu problem görüntü eşleme değil, alt görüntü eşleme olarak ifade edilir ve bir görüntüde bir görüntünün kaba kuvvet yolla hiçbir olası durumun göz ardı edilmeden aranması çok uzun sürebilir. Bu durumda şablon görüntü ile karşılaştırılacak toplam alt görüntü sayısı 4.1'deki denklemde görüldüğü gibi hesaplanır.

$$S = (M - m + 1) \times (N - n + 1) \quad (4.1)$$

Kaba kuvvet aramalar birer sırayla yapılmadan da gerçekleştirilebilir. Pencere arası geçiş değeri birden fazla olabilir. Bu durumda, hesaba katılan toplam alt görüntü sayısı 4.2'de verilen denklemde görüldüğü gibi azalır.

$$S = ((mx - mx \% g) \div g + 1) \times ((ky - ky \% g) \div g + 1) \quad (4.2)$$

Bu problemde ilgili görüntülere göre eşleşecek görüntü sayısının artması, görüntü arama süresini görüntü boyutlarının artış oranına göre daha büyük bir oranda uzatabilir. Bu sebeple görüntü aramak için GA'nın kullanılması uygundur. Çünkü arama uzayının çok büyük olması, değerlendirilen toplam alt görüntü sayısı kadar görüntü eşlemenin gerçekleştirilmesine ve hâliyle işlemlerin oldukça uzun sürmesine sebep olabilir. Bu sebeple bu çalışmada, GA ile arama kaba kuvvet aramalara bir alternatif olacak şekilde kullanılmıştır.

Görüntü aramaları, referans ve şablon görüntü tekrar boyutlandırılarak yapılabilir. Çoklu boyutlandırma aktif (seçim kutusu seçili) iken en küçük ve en büyük boyutlandırma değerleri aynı ise o değerde, aktif değil iken orijinal boyutta tek bir boyutlandırma ile arama gerçekleştirilir. Diğer durumda ise en küçük boyutlandırma değerinden en büyük boyutlandırma değerine kadar belirlenen boyutlandırma sayısı kadar boyutlandırma ile aramalar gerçekleştirilir. Her yeni boyutlandırma değeri bir önceki boyutlandırma değerine 4.3'teki denklemde verilen değer eklenerek hesaplanır.

$$inc = (mxs - mns) \div (ns - 1) \quad (4.3)$$

Referans ve şablon görüntüyü yeniden boyutlandırma işlemi, 4.4'teki denklem grubunda verilmiştir. Bu işlemler görüntülerin orijinal boyutları üzerinde yapılır.

$$\begin{aligned}
M &= M \times sc : N = N \times sc \\
m &= m \times sc : n = n \times sc
\end{aligned}
\tag{4.4}$$

Arama sürecinde görüntü eşleme işlemlerinin değerlendirilmesi 4.5'teki grupta verilen iki farklı koşulla gerçekleştirilir.

$$\begin{aligned}
1. &(s \geq th) \\
2. &(s = d)
\end{aligned}
\tag{4.5}$$

YSA çıkışı ve korelasyon ile görüntü eşleme için eşik değeri kullanılırken (4.5'te verilen birinci koşul); Ortalama Renk Değeri (ORD), Renk Değerlerinin Standart Sapması (RDSS) ve kenar tespiti ile görüntü eşleme için şablon görüntüye uygulanan fonksiyon ile elde edilen değer (4.5'te verilen ikinci koşul) karşılaştırılır.

GİGA, paralel olmayan kaba kuvvet arama, paralel kaba kuvvet arama, paralel olmayan akıllı arama ve paralel akıllı arama olmak üzere dört arama şekli içermektedir. Takip eden kısımlarda bu uygulamalar anlatılmaktadır.

4.2 Kaba Kuvvet Arama

Kaba kuvvet arama, sorgu görüntüdeki şablon görüntü boyutundaki her görüntünün şablon görüntü ile potansiyel uyum sağlayacak (örtüşecek) şekilde tek tek eşleştirilmesi ile yapılan aramadır. Eşleme için renk mesafe ölçümü kullanılabilir. Sonuçlar bir tek görüntü mesafesinde toplanır (şablon görüntünün K renk bandı için ORD ile ifade edilmesi).

Sırayla yapılan bu arama çok zaman almaktadır. Referans görüntüde arama yapılırken şablon görüntü ile eşleşecek büyük görüntü alt parçaları için iki pencere arasındaki geçiş değeri birden fazla olabilir. Geçiş değeri artırılarak belirli aralıklarla aramaların yapılması durumunda ise atlanan piksellerin alt görüntülerin sol üst köşelerine tekabül etmesi ve tespit edilmelerine engel olması durumu ortaya çıkabilir.

Referans görüntüde arama yapılırken yatayda ya da düşeyde g kadar sonra gelen (düşeyde ilerlerken X aynı $(X, Y + g)$, yatayda ilerlerken Y aynı $(X + g, Y)$ olacak şekilde) bir sonraki alt görüntüye bakılır. Bu değer bir olması art arda bakılan alt görüntüler arasında başka bir alt görüntünün yer almaması demektir. Görüntüler

kaçar sırayla karşılaştırılırsa, her görüntü eşlemede g değerinin bir eksiği kadar alt görüntü göz ardı edilmiş olur.

4.3 Akıllı Arama

Bu tez çalışmasında akıllı arama GA kullanılarak yapılmaktadır. GA'nın paralel arama gerçekleştirmesi, tek bir çözüm yerine çözüm kümesi olarak sonuç üretmesi ve arama süreci optimizasyonunu sağlayabilecek bir yöntem olması, GİGA probleminde kullanılma gerekçeleri arasındadır.

Zor ve karmaşık problemlere hızlı ve kolayca uygulanan GA, görüntü aramada alternatif bir yöntem olarak kullanılabilir.

GİGA'da en iyi çözümü ve bu çözüme yakın çözüm kümesini bulabilmek için GA'nın arama parametrelerini oluşturan değerler dikkate alınmıştır. Bu sebeple, en uygun değerlerin neler olabileceğini tespit edebilmek için görüntü arama sonuçları değerlendirilmiştir. Bu problem için en uygun nesil ve birey sayıları, çaprazlama ve mutasyon işlemcileri olasılıkları ile şekilleri, yeni nesli oluşturma uygulamaları ve birey seçim yöntemleri üzerinde durulmuştur.

Görüntü aramada GA kapsamındaki durumlar şunlardır:

- Şablon görüntü ile eşleşen görüntülerin referans görüntüdeki sol üst piksel konumlarının X ve Y değerleri ikilik sistemdeki değerler şeklinde ifade edilmiştir. Tamsayıların kullanılması ile yeterince çeşitlilik sağlanamadığı görülmüş ve ikilik sistemdeki değerler ile çalışılmıştır.
- Küçük değişiklikler yapacak şekilde bir mutasyon işlemcisi kullanılmıştır. Bu değişiklik genler üzerinde yapılan değişiklikleri içermektedir.
- GA'nın uygulanma sürecinde sıralama işlemleri kodlamadan çıkarılmıştır. Bireylerin bir arada bulunduğu veri yapısı olarak listelerin kullanılması ve uygunluk değerlerinin listeye sıralı olarak yerleştirilmesi suretiyle sıralama için ayrıca zaman harcanmamıştır. Bu uygulama, listeye eklenecek bireyin uygunluk değeri ile nüfustaki bireylerin uygunluk değerlerinde ikili arama (binary search) yaparak (bu arama için liste sıralı olmalıdır) liste küçükten büyüğe sıralı olacak şekilde listenin en uygun yerine yerleştirilmesi suretiyle

gerçekleştirilir. Böylelikle, liste üzerinde arama ve liste elemanlarına erişim gibi özel işlemler daha hızlı bir şekilde yapılabilir.

- GA'nın uygulanma sürecinde kopyalama işlemleri de kodlamadan çıkarılmıştır. Kopyalama işlemi yapılmadan söz konusu nesildeki bireyler üzerinde ilgili işlemler yapılarak ortaya çıkan bireyler bir sonraki nesle eklenmiştir. Bireylerin tamamının yeni nesle aktarılıp o nesil üzerinde değişiklik yapılması uygulaması, bireylerin değişikliklerle beraber yeni nesle aktarılması şeklinde gerçekleştirilmiş ve kopyalama işlemi için ayrıca zaman harcanmamıştır.
- Yeni nesil oluşturulurken yeni yaratılan bireylerle beraber en iyi uygunluk değerlerine sahip bireylerin bir kısmı da doğrudan yeni nesle aktarılmıştır. Ayrıca kalanlar arasında bir kısım da rasgele seçilerek yeni nesle dâhil edilmiştir.

Görüntü aramada GA'nın başarılı sonuçlar verebilmesi ilk olarak kodlamanın doğru şekilde yapılmasını gerektirir. Birden fazla kodlama bir arada kullanılabilir. Bu problemde arama sonucu olarak elde edilmek istenen, şablon görüntü ya da görüntülerin sol üst piksel köşelerinin referans görüntüde eşleştiği X ve Y koordinatlarıdır. Bu değerler onluk sistemdeki sayılardır. GA için bu değerler ikilik sistemde kodlanmışlardır.

Problemin karar değişkenlerinin (parametreler ve bireyler) uygun bir yöntemle kodlanmış olması kadar GA için hangi değerlerin ya da değerlerin iyi olduğunu en iyi şekilde ifade eden uygunluk fonksiyonlarının da tasarlanması önemlidir.

GİGA'da kaba kuvvet arama yöntemi, hesaplama zamanını görüntülerin boyutunun artması ile büyük miktarda artırmaktadır. Bu probleme kısa sürede kabul edilebilir çözümlerin getirilebilmesi, GA ile başarılabilir. Bu çalışmada kabul edilebilir çözümlere ulaşılması oldukça yeterli bir sonuç olabilir. Çünkü bu sonuçlardan en iyi sonuçlara oldukça kolay bir şekilde kısa süreli kaba kuvvet aramalar yapılarak ulaşılabilir. Böylece, daraltılan çözüm uzayından elde edilen olası çözüm adayları daha iyi sonuçlara ulaşabilmede yardımcı olabilirler. Fakat bu noktada, uygulama sürecinin çok iyi bir şekilde oluşturulması büyük öneme sahiptir.

Bu tez çalışmasında kaba kuvvet arama tek bir iş parçacığı ile ve paralel olarak yapılmaktadır. Akıllı aramalar için kullanılan GA için de yine tek iş parçacığı ile ve

paralel olarak aramaların yapılabilmesi sağlanmıştır. Bu dört farklı seçenek arama sonuçlarının karşılaştırılması amacıyla kullanılmaktadır. Akıllı aramalara bu çalışmada ek olarak getirilen paralel akıllı arama ile tek iş parçacıklı akıllı arama ve paralel akıllı arama işlemleri de kendi aralarında karşılaştırılabilir.

GA'nın bu tez çalışmasında kullanılmasının en önemli sebeplerinden biri, çözüm kümesinin tamamını taramadan verdiği sonuçlarla daha iyi sonuçlara daha kısa sürede ulaşılabilmesidir. Bu noktada akıllı aramalarla elde edilebilecek belli derece iyi sonuçlar yeterli olabilir. Bu sonuçlar her zaman en iyi çözüm ya da çözümleri içermeyebilir, ama çoğu zaman en iyiye yakın çözümlere ulaşılmasını sağlayabilirler. İyi sonuçların elde edilebilmesi, tüm algoritma tasarımının etkili olarak icra edilebilmesini gerektirir. Çözüm uzayında çok sayıda olası çözüm içinde yapılan aramalarla kısa sürede yeterli sonuçlara ulaşmak mümkündür.

Ayrıca, GA'nın birden fazla iş parçacığı ile referans görüntüden elde edilen görüntü parçaları üzerinde paralel olarak çalıştırılması arama zamanını kısaltabilir. Bu süreçte paralellik ve kodlamanın optimizasyonu, sisteme fazla ve gereksiz yük bindirmeyecek şekilde oluşturulabilmelidir. Eş zamanlı aramalarla çözüm uzayının bir kısmı taranarak çözüme daha kısa sürede ulaşabilmek mümkündür.

Daha iyi sonuçlar verme ihtimali düşük olan alanlarda arama yapılmayabilir. Fakat bu durumda, iyi sonuçlar vermeme ihtimali olan alanların uygunluk değerlerine bakılması ve uygulamaya çeşitli koşulların getirilmesi arama süresini oldukça uzatabilir. Örneğin her nesil tekrarında ve döngü yapılarındaki pek çok çevrimde ele alınan ek koşullar bu duruma neden olabilir. Bu yüzden, bahsedilen bu ve benzeri koşulların en aza indirilerek çeşitliliğin ortaya çıkarılması sağlanmalıdır. Çok sayıda koşul ve matematiksel işlem yerine genetik işlemcilerle yeni ve daha iyi bireylerin elde edilmesi ile arama süreci daha verimli bir şekilde getirilebilir. Sözü geçen yeni bireyler, nüfustaki bireylerden daha iyi ya da daha kötü ve hatta nüfustaki bireyler ile aynı da olabilirler.

Bu çalışmada, nesildeki en iyi bireylerin bir kısmı çaprazlama ve mutasyon işlemcilerine tabi tutulmadan bir sonraki nesle aktarılır. Elitizm denen bu uygulama ile uygunluk değerleri en iyi olan bazı bireyler yitirilmeden bir sonraki nesle aktarılmış olur. Böylece yeni nesildeki en iyi bireyin bir önceki nesildeki en iyi

bireyden daha kötü olması ihtimalinin önüne geçilmiş olur. Uygunluk değeri daha iyi olan bireylerle beraber diğer bireylere de şans tanınmış olur.

En iyi çözüme ulaşmada çeşitlilik gerekebilir ve bu yüzden eldeki verilerle birçok yere ulaşabilmek arama için çok önemlidir. Bu çalışmada GA ile yeni nesil oluşturulurken, nesildeki en iyi bireylerden bir kısmı ve kalan bireylerden rasgele bir kısım kullanılır. Kalan bireyler ise çaprazlama ve mutasyon işlemleri ile yeni bireyler olarak yeni nesle katılırlar ve toplam birey sayısı kadar bireyle yeni nesil elde edilir. Böylece yeni neslin yaratılması nesildeki bireylerin en iyilerinden, kalanların belli bir kısmından, kalanların bir kısmı ile çaprazlama yaparak, geriye kalanları da mutasyona uğratarak dört farklı şekilde olur.

Önceki nesilden yeni nesle aktarılan bireyler ile çaprazlama ve mutasyon işlemcileri ile oluşan tüm bireyler nesillerdeki toplam birey sayısı kadardır ve toplam birey sayısını aşmamaktadır. Böylece, fazlalık bireylerin sonradan elenmesi için harcanacak zamanın önüne geçilmiştir. Bireylerin elenmesi, bireylerin kodlamada kullanılan veri yapılarından çıkarılması demektir.

Probleme yönelik olarak farklı işlemciler çeşitli şekillerde kullanılabilir. Genetik işlemciler için farklı yollar izleyerek yeni bireyleri elde etmede çeşitlilik sağlanabilir.

Ebeveyn bireylerden elde edilen çocuk bireylerin uygunluk değerleri en iyi olan belli miktarının, ebeveyn bireylerin uygunluk değerleri en kötü olan aynı miktar ile değiştirilmesi uygulaması bu çalışmada kullanılmamıştır.

Yeni neslin oluşturulması için gerekli bu işlemler nesil sayısı kadar tekrarlanarak her yeni topluluk bir sonraki nesil olarak belirlenir ve en son nesildeki uygunluk değeri en iyi birey ya da bireyler çözüm kümesi olarak elde edilir.

4.4 Paralel Arama

Paralel aramada referans görüntü bütün olarak ele alınmak yerine ön işlemlerle küçük bloklara ayrılabilir. Referans görüntü ilk olarak dikey şekilde bölünür ve ardından aynı yükseklik değerindeki bu görüntüler kendi içlerinde yatay olarak bölünürler. Burada bölmek, görüntüyü alt parçalara ayırmak demektir. Referans görüntünün örneğin dikey olarak bölünmesi, dikey boyutları referans görüntünün dikey boyutu ile aynı yatay boyutları ise referans görüntünün yatay boyutundan daha küçük referans görüntü parçaları elde edilmesidir. Bu süreç sayesinde küçük bloklar

elde edilebilir. Öte yandan aranan görüntü ile birlikte şablon görüntü de alt parçalara ayrılabilir ve paralel programlama teknikleri kullanılarak aramalar gerçekleştirilebilir. Bu çalışmada şablon görüntü alt parçalara ayrılmamıştır.

Bu uygulamada referans görüntüdeki olası hiçbir alt görüntü parçası atlanmayacak şekilde bloklar elde edilmelidir. Uygulamanın çalıştırıldığı makine de göz önüne alınarak bu yolla alt iş parçacıkları ile yapılan aramalarda iyileştirmeler olabilir. Her bir blok şablon görüntüyü kapsayacak şekilde ve kullanılan bilgisayarın da özelliklerine bağlı olarak paralel şekilde sorgulanabilir.

Dikkat edilmesi gereken husus, alt iş parçacıklarının kendi görüntü blokları açısından bağımsız olarak ve referans görüntü açısından bütünlük içerisinde çalışabilmesidir. Ayrıca referans görüntüleri alt bloklara ayırma, kullanılan bilgisayarın özelliklerine ve sorgulanan referans görüntüye göre dikkatle yapılmalıdır. Öte yandan bu uygulamada, örneğin birinci iş parçacığına verilecek alt görüntü parçasının diğer görüntü parçalarına (diğer görüntü parçaları aynı boyuttalar) göre çok daha dar bir alan kaplaması bu iş parçacığının erken sürede bitmesine neden olur. Diğer iş parçacıkları çoğu zaman birbirlerine göre yakın zamanlarda bitebilirler. Fakat paralelliğin verimliliği, görüntü parçalarının boyut olarak birbirlerine daha yakın olması ile genel olarak artar.

4.5 Uygunluk Fonksiyonları ve Benzerlik Ölçümleri

Referans görüntünün hangi konum ya da konumlarında şablon görüntünün yer aldığı tespit edilmesi için görüntülerden birtakım veriler elde edilmeli ve ardından bu veriler karşılaştırılmalıdır. Görüntü sorgulama renk, doku, şekil gibi görüntü öznitelikleri gerektirir. Görüntülere içerik tabanlı erişimlerde veya görüntüleri eşlemede bu özellikleri karşılaştırmak için çeşitli ölçümler kullanılmaktadır.

Renk ve doku bir resmin genel özellikleridir. Bu özelliklere istinaden bir resmin belirli bir renk veya dokusu olup olmadığı ise mesafe ölçümleri ile belirlenir.

Renk benzerliği görüntünün her bir pikseli için söz konusu olan bir özelliktir. Doku benzerliği ise bir aradaki piksellerin oluşturduğu yapı için geçerlidir. Doku renk benzerliğinden daha karmaşıktır ve içerik tanımlamada önemli bir özniteliktir. Belli bir dokuya benzer doku içeren görüntülerde benzer bir renk dağılımı veya gri

tonlama söz konusu olabilir, fakat bu görüntüler tam olarak aynı renkler ve gri tonlar içermemektedir.

Doku tabanlı görüntü aramalarında görüntüdeki renklerden tamamen bağımsız çeşitli yöntemlere dayalı bir benzerlik ölçümü kullanılabilir. Bu ölçüt bir uzaklık birimidir ve benzerlik tanımı gösterime göre değişiklik gösterebilir.

Öte yandan doku ve renkten ötede şekil benzerliği de içerik tabanlı görüntü sorgulamalarında kullanılabilir. Şekil, bir görüntü için resim özelliği değil, görüntü içeriğindeki özel bir bölgeyi ifade eden yapıdır. Şekil benzerliği ölçümlerinde bölge tanıma işlemleri gerekir.

İlgili olunan şekil ya da şekilleri görüntüden ayırmak zor bir uygulamadır. Çoğu zaman otomatik olarak yapılamayabilir ve bu sebeple uygulama gereksinimine göre özel olarak yapılması gerekebilir. Ayrıca şekil tabanlı aramaların geniş çerçevede yapılabilmesi için bölümlene ciddi bir problem olabilir.

Mesafe ölçümleri ile benzerlik ölçümleri ters orantılıdır. Mesafenin artması benzerliği azaltırken azalması benzerliği artırır. Görüntü eşleme öncesi görüntülerden mümkün olduğu kadar detaylı ve benzersiz niteliklerin çıkarılmasıyla en iyi eşleme konumları bulunabilir. Ayrıca, farklı tanımlayıcılardan elde edilen birden fazla öznitelik için farklı uzaklık ölçü birimleri ve ağırlık değerleri kullanılabilir. Öznitelik vektörleri buna göre oluşturulabilir. Bu çalışmada, birden fazla öznitelik çıkarımı ve öznitelik vektörlerinin ağırlıklandırılması gibi uygulamalar yer almamaktadır.

Görüntü eşleme için benzerlik uyumu olarak renk uyumu, doku uyumu, şekil uyumu ele alınabilir. Probleme yönelik olarak görüntü içeriklerinin ilgili kısımları ya da tamamı taslak olarak ana hatlarıyla elde edilip eşleştirilebilir. Sorgular problemlere yönelik olarak tasarlanabilir. Arama sonuçları da yine probleme yönelik olarak düzenlenebilir.

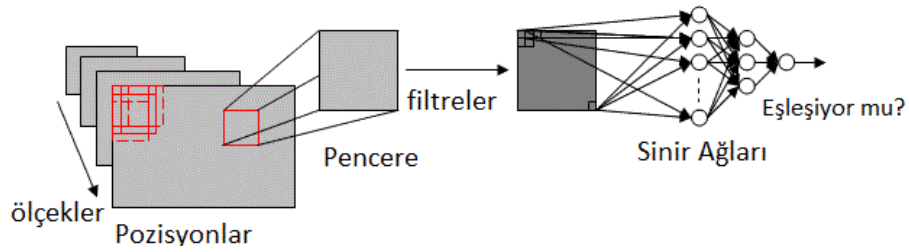
Bu tez çalışmasında görüntü eşleme için çeşitli uygunluk fonksiyonlarından yararlanılmıştır. İlgili yöntemler takip eden alt madde başlıklarında anlatılmıştır.

4.5.1 Yapay sinir ağları

Bu çalışmada görüntü eşleme amacıyla temel olarak kullanılan yöntem YSA'dır. YSA'nın öğrenme yöntemleri ise GY öğrenme ve GA'dır. Çok katmanlı sinir ağlarının öğrenmesi için en çok bilinen algoritmalarından biri olan GY öğrenme

algoritmaları pek çok probleme uygulanabilmektedir. Yakınsama, tahmin, nesne tanıma gibi uygulamalarda yaygın olarak kullanılan GY öğrenme ve arama problemleri, optimizasyon, planlama problemleri, mühendislik tasarımları gibi uygulamalarda kullanılan GA, bu çalışmada YSA'nın öğrenme yöntemleridir.

Görüntü eşleme amacıyla YSA kullanılarak görüntülerin arandığı sistem için temel süreç genel olarak Şekil 4.1'de verilmiştir. Bu yapı YSA kullanılarak farklı ölçeklendirmelerde kaba kuvvet yollarla yapılan aramaları içermektedir. GİGA çalışması için gerçekleştirilen uygulama öncesinde incelenen YSA ile yüz tespiti çalışmasında, YSA'ya görüntüler giriş olarak verilmeden önce görüntülere birtakım filtreler uygulanır. Bu filtreler sırayla, görüntülerin gri tonlara çevrilmesi, parlaklığın normalize edilmesi (normalleştirme, standartlaştırma) ve ardından kontrastın (zıtlık) histogram eşitleme ile normalize edilmesi şeklindedir.

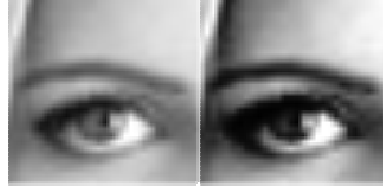


Şekil 4.1 : YSA ile görüntü eşleme sürecinin genel gösterimi.

Referans görüntünün her farklı ölçeği (farklı boyutlandırma değeri) için referans görüntüden elde edilen her pencere (şablon görüntü boyutundaki alt görüntü), filtreli olarak sinir ağlarına giriş olarak verilir ve eğitimi tamamlanmış sinir ağı kullanılarak şablon görüntü ile eşleşip eşleşmediği tespit edilir.

Şekil 4.1'de referans görüntüden elde edilen her pencere filtrelerden geçirilerek YSA'ya verilmektedir. Bu çalışmada bu uygulama, her pencere için filtreleme işlemlerinin tekrar tekrar yapılmaması amacıyla referans görüntünün filtreli hâlinde elde edilen pencerelerin doğrudan YSA'ya verilmesi şeklinde icra edilmiştir. Görüntü aramaları böylelikle YSA için çok daha hızlı olmuştur. Bu noktada her pencerinin ayrı ayrı filtrelerden geçirilmesi gerekmemektedir. Ağ eğitimi filtreli referans görüntüden elde edilen pencerelerle yapılır ve görüntü aramak için de yine filtreli referans görüntüden elde edilen pencereler YSA'ya verilir. Böylelikle örtüşmeyen herhangi bir durum ile karşılaşılmaz.

Renk farklarından kurtulmak için renkli görüntüler gri tonlara çevrilir. Şekil 4.2’de görüldüğü gibi görüntüler histogram eşitlemeyle hem aydınlatma için normalize edilirler hem de görüntülerin gri ton aralıkları genişletilir. Daha sonra, şablon görüntü ile ilgili olan ve ilgili olmayan görüntüler kullanılarak sinir ağı eğitilir ve eğitilen ağ görüntü eşleme işlemi için kaydedilir.



Şekil 4.2 : Şablon görüntü ve bu görüntünün YSA’ya verilecek hâli.

Şekil 4.2’de solda referans görüntüden kırılan şablon görüntü, sağda ise şablon görüntünün gri tonlara çevrildikten sonra histogram eşitlemesi ile elde edilmiş hâli görülmektedir. Bir görüntüden bir görüntünün kırılması, bir görüntünün herhangi bir kısmının seçilerek bu görüntüden ayrılması ve başka bir görüntü olarak elde edilmesidir.

Bu çalışmada, doğruluğundan emin olmak için ilk olarak küçük bir görüntü büyük görüntüden kırılır. Ardından sadece bu görüntü ile veya birkaç görüntü ile sinir ağı eğitilerek arama gerçekleştirilir. Kaba kuvvet arama yüksek eşik değeri ile ve pek çok piksel atlanarak yapılır. Görüntü aramada bu noktada YSA ile görüntü eşlemesi oldukça fazla yanlış görüntüyü sonuç olarak verebilir. Bu sebeple, sonuç görüntülerin en aza indirilmesi için eşik değerinin yüksek tutulması ve pencereler arasındaki geçiş değerinin oldukça artırılması gerekir.

YSA’nın arama sonucunda verdiği çoğu yanlış görüntünün belli bir kısmı eşleşmeyen görüntülere dâhil edilerek sinir ağı güncellenir ve ağ tekrar eğitilerek kararlı bir duruma gelene kadar bu işlemlere devam edilir. Bu sürecin sonunda sinir ağı bekleneni karşılayacak şekilde ilgili olunan alt görüntü ya da görüntüleri bulur.

Sinir ağı daha kararlı bir hâle geldikçe ve sonuç olarak verdiği görüntüler azaldıkça eşik değeri düşürülebilir ve kaba kuvvet aramalar için pencereler arasındaki geçiş değeri azaltılabilir. Böylelikle, doğru ve yakın görüntü eşleme sonuçlarının elde edilmesi mümkün olur.

YSA ile görüntü eşleyerek hızlı arama için YSA’nın en iyi ya da en iyiye yakın sonuçlar verdiği diğer alt konumlar üzerinden esas konuma ulaşılmaya çalışılmıştır.

Fakat yapılan denemelerde YSA, eğitim durumuna, giriş görüntülerine ve diğer özelliklerine göre şablon görüntünün sol üst köşesinin referans görüntüde yer aldığı piksel konumunun bir piksel komşuluklarında çok farklı değerler vermiştir. Bu durum arama sürecini zorlaştırmıştır ve daha farklı uygunluk fonksiyonları da kullanılarak akıllı algoritmalarından faydalanılmasını gerektirmiştir. Bu noktada, kaba kuvvet aramalara akıllı ve paralel aramalar, YSA'ya ise çeşitli görüntü eşleme yöntemleri dâhil edilerek arama süreçleri karşılaştırılmıştır.

En uygun eşleşen ve eşleşmeyen görüntülerle başarılı bir şekilde eğitilmiş ve aranmak istenen görüntü için kararlı bir yapıya gelmiş ağı bu yakın komşuluklarda oldukça düşük çoğu zaman sıfır ya da sıfıra yakın değerler verdiği görülmüştür. Bu sebeple komşulukların birbirine göre durumlarından, daha hızlı bir şekilde YSA kullanılarak en fazla eşleşen görüntü ya da görüntülere ulaşılması çok zor olmuştur. Kaba kuvvet haricindeki aramalardan iyi sonuçlar elde edilememektedir. Dolayısıyla YSA yönteminde bire bir eşleşen görüntülerin geçtiği konumların haricindeki konumlarda eşleme yapılarak en fazla eşleşen görüntü ya da görüntülere ulaşmak genelleştirilememiştir. Özellikle bu sebepten görüntü aramalarının akıllı algoritmalar kullanılarak yapılmasına karar verilmiştir.

YSA kullanılarak görüntü eşleme için ağa eşleşmeyen çok sayıda görüntü ve eşleşen tek bir görüntü verilebilir. Mühim olan, küçük görüntüler için aranan görüntüye göre ilgililerin ne kadar ilgili olduğu iken ilgisizlerin de ne kadar ilgisiz olduğudur. Ağ eğitiminin başarılı bir şekilde gerçekleştirilebilmesi ve aramaların doğru şekilde yapılabilmesi, eşleşen ve eşleşmeyen görüntülerin bir bütünlük içinde sinir ağına giriş olarak verilmesine bağlıdır.

YSA ile GİGA'da, ağın eğitim durumu ve içeriği çok büyük rol oynamaktadır. YSA çıkışlarının doğru bir şekilde değerlendirilmesi, arama algoritmalarının ilgili duruma göre oluşturulmasını gerektirir. YSA'nın eğitim sürecinde eşleşen ve eşleşmeyen alt giriş görüntülerinin uygun olarak seçimi ve aranmak istenen alt görüntüye göre gerçekleştirilen eğitim, aramada belirleyici bir rol oynamaktadır. Küçük bir görüntünün daha büyük bir görüntü içerisinde yer alıyor ise doğru bir şekilde konumunun, yer almıyor ise olmadığının tespit edilmesi bu şekilde gerçekleştirilir. Bu durum başarımda büyük öneme sahiptir.

Bu çalışmada YSA için söz konusu durumlar şunlardır:

- Eğitimde, eşleşen veya eşleşmeyen görüntü ya da görüntüler için aynı görüntünün birden fazla kullanımı gibi işlemler uygulanmamalıdır. Fakat şablon görüntü ile bire bir aynı olmayan ama eşleşebilen benzer görüntülerin verilmesinde büyük fayda vardır. Bu durum eşleşmeyen görüntüler için de geçerlidir.
- YSA en düşük hata değeri ile eşleşen yerlerde en yüksek çıkışları vermektedir. Birer piksel komşuluktaki şablon görüntüler için ise zaman zaman en yükseğe ya da sıfıra yakın değerler verebilmektedir.
- Eğitim sürecinde giriş görüntülerinin sayısının fazlaca olması test sürecinde görüntü aramada daha sağlıklı sonuçlar vermesine rağmen eğitim sürecinde işlem yükünü de oldukça artırmaktadır.
- YSA ile görüntü eşleyerek arama yapmada kaba kuvvet aramalar yerine akıllı algoritmalarından yararlanılmalıdır.

YSA ile GİGA sürecinde daha iyi erişim sonuçları elde edebilmede ilgili olmayan görüntüler büyük rol oynamaktadırlar. YSA kullanılarak yapılan görüntü aramalarında, sorgu sonuçları arasından ilgili olanlar eşleşen, ilgili olmayanlar eşleşmeyen filtreli gri görüntülere dâhil edilecek şekilde geri bildirimlerle sinir ağı kararlı bir hâle gelene kadar eğitilebilir. Böylelikle, kullanıcının gittikçe arzu ettiği görüntüleri ve bu görüntülere daha yakın görüntüleri elde edebilmesi mümkün olur.

Özellikle örüntü tanıma uygulamalarında yaygın olarak kullanılan YSA, bu tez çalışmasında GİGA amacıyla kullanılmıştır. Bu problemi çözebilmek için YSA, giriş olarak verilen belli bir görüntüyü tanımaya genelleştirilmiştir.

Algoritması veya kuralları tam olarak bilinmeyen durumlar için geliştirilmiş bir bilgi işleme sistemi olan YSA, bu çalışmada bir görüntüyü başka bir görüntüde aramak için kullanılmıştır. Görüntü aramada kurallar tam olarak belli değildir. İlgili görüntüyü tespit edebilmek için ilgili görüntüyü ya da görüntüleri ilgili olarak, ilgili olmayan görüntü ya da görüntüleri ise ilgili olmayan görüntüler olarak sinir ağına verip ağı eğitimi gerçekleştirme gerekir. Bu noktada, ilgili olmayan görüntülerin aynı zamanda ilgili görüntüler ile örtüşmemesi (girişlerin en uygun şekilde ayarlanması) YSA ile görüntü eşlemeleri için önemlidir. İlgili ve ilgili olmayan

görüntüler kendi içlerinde bütünlük içinde, birbirlerine göre ayrık olabilmelidirler. Bu görüntüler aynı görüntüden elde edilen alt görüntüler de olabilirler ve bu yüzden birbirlerinden tamamen bağımsız olmak zorunda değildirler.

Bu çalışmada görüntü eşleme için ilk olarak YSA'nın kullanılmasının en önemli sebeplerinden biri, YSA'nın belirsiz, eksik ve gürültülü verilerle de çalışabilmesidir. Fakat problem arama problemi olduğundan YSA sürecinde sistemin öğrenmesini zorlaştıracak tüm durumlar ortadan kaldırılmalı ve öğrenme süreci optimizasyonu gerçekleştirilmelidir. Bu işlemler aynı zamanda YSA'nın hata değerlerinin artmasına, ağırlık eğitimi süresinin uzamasına ya da doğru olmayan bir eğitime sebep olmayacak şekilde yapılabilir.

Herhangi bir uygulamada girişler ve çıkışlar arasındaki ilişkiyi mevcut örneklerden öğrenip çözümler üretebilme kabiliyeti, bu tezde YSA'nın bir YZ yöntemi olarak görüntü eşleme amacıyla kullanılması sebeplerindedir.

4.5.2 Ortalama renk değeri

İlgili görüntü alanlarını ortalama renk değerleri (görüntünün tüm piksellerinin) ile ifade etmek YSA'ya alternatif bir yöntem olarak kullanılabilir. Şablon görüntünün ORD ile referans görüntüden çıkarılan şablon görüntü boyutundaki tüm alt görüntülerin ortalama renk değerleri karşılaştırılır. ORD 4.6'daki denklemde görüldüğü gibi hesaplanır (n bu denklemde görüntüdeki toplam piksel sayısıdır ($X \times Y$)).

$$\bar{c} = \frac{1}{n} \sum_{i=1}^n c_i \quad (4.6)$$

4.5.3 Renk değerlerinin standart sapması

Standart sapma verilerin ortalamaya göre yayılmasını gösteren istatistiksel bir ölçüdür. İlgili görüntü alanları için görüntülerin ortalama renk değerlerinden elde edilen standart sapma değerleri bir başka uygunluk fonksiyonu olarak kullanılabilir. Şablon görüntünün RDSS ile referans görüntüden çıkarılan şablon görüntü boyutundaki tüm alt görüntülerin renk değerlerinin standart sapmaları karşılaştırılır. RDSS 4.7'deki denklemde görüldüğü gibi hesaplanır (n bu denklemde görüntüdeki toplam piksel sayısıdır ($X \times Y$)).

$$\sigma = \sqrt{\frac{1}{n} \sum_{i=1}^n (c_i - \bar{c})^2} \quad (4.7)$$

4.5.4 Korelasyon

İki bağımsız değişken arasındaki doğrusal ilişkinin yönünü ve kuvvetini belirten korelasyon yöntemi ilinti tabanlı bir yöntemdir. İki veri grubu arasındaki ilişki değerinin ölçülmesi amacıyla kullanılmaktadır. Genel istatistiksel kullanımda korelasyon, bağımsızlık durumundan ne kadar uzaklaşıldığını gösterir. Şablon görüntünün renk değerlerinden elde edilen tek boyutlu dizi ile referans görüntüden çıkarılan şablon görüntü boyutundaki tüm alt görüntülerin renk değerlerinden elde edilen tek boyutlu diziler arasında korelasyon hesabı yapılır ve bu veriler arasındaki ilişki değerleri ölçülür. Tek boyutlu diziler, görüntülerin herhangi bir renk kanalı için piksel değerlerini içerirler. Görüntülerin iki boyutlu olması ve birden fazla renk kanalı içermesi sebebiyle Emgu CV’de matris olarak piksel değerlerinin tutulduğu yapı (görüntü özelliklerinden *Data*)dan tek boyutlu diziler elde edilmelidir. İki boyutlu diziler kullanarak dizilerin bir boyutunda görüntülerin yatay ekseninde ve diğer boyutunda da görüntünün dikey ekseninde karşılık gelen piksel noktası için renk değerleri atanır.

Korelasyon hesabı için Aforge.NET kütüphanesi kullanılır.

Görüntülerin renk değerlerinden elde edilen iki boyutlu iki dizi arasındaki korelasyon katsayısının hesabı, 4.8’deki denklemde görüldüğü gibi hesaplanır (n bu denklemde görüntüdeki toplam piksel sayısıdır ($X \times Y$)).

$$cc = \frac{\sum_{i=1}^n (c_i - \bar{c})(l_i - \bar{l})}{\sqrt{\sum_{i=1}^n (c_i - \bar{c})^2 \sum_{i=1}^n (l_i - \bar{l})^2}} \quad (4.8)$$

4.5.5 Kenar tespiti

Görüntü işlemede bölümlenme önemli ve zor bir tekniktir. Bu teknik görüntüde çeşitli bölge tanıma işlemleri gerektirir ve görüntüde ilgili olunan kısımları görüntünün diğer kısımlarından ayırmak için kullanılabilir. Görüntü sorgulamalarında görüntü boyutunu düşürmek ya da sadece ilgili olunan görüntü kısmı ile çalışmak büyük

öneme sahiptir. Görüntüdeki kenarlar bulunarak gri tonda veya siyah beyaz bir görüntü elde edilebilir ve aramalar dar bir alanda yapılabilir.

Bu çalışmada, şablon görüntünün kenarları çıkarılarak elde edilen gri tonlu görüntü ile referans görüntüden çıkarılan şablon görüntü boyutundaki tüm alt görüntülerin kenarları çıkarılarak elde edilen gri tonlu görüntüler bire bir piksel karşılaştırması ile eşleştirilir. Böylece görüntü içeriklerinin ana hatları taslak olarak elde edilir ve eşleştirilir. Bu çalışmada bu yöntem için ilk olarak söz konusu görüntüler siyah beyaz olarak elde edilmiş ve bu görüntülerdeki kenarları ifade eden siyah piksel sayıları karşılaştırılmıştır.

4.5.6 Görüntü işleme kütüphanesi fonksiyonları

Görüntüler değişik renk çeşitleri içerebilir ve görüntü içeriğindeki nesnelere düz (düşey (dikey) eksene paralel olacak şekilde) ya da değişik açılarda duruyor olabilir. Görüntüler, içeriklerindeki nesnelere yönsel ve açısal değişiklikleri göz ardı edilmeksizin bu görüntülerden birtakım öznitelikler çıkarılarak eşleştirilmek istenebilir. Bu öznitelikler, görüntüleri tanımlayan ve görüntülerin birbirleri ile karşılaştırılmasında kullanılacak nitelikteki özelliklerdir.

Eğitilmiş yapay sinir ağı kullanılarak ilgili olunan görüntüler aranabilir. YSA eksik veri ile çalışabilir, aranmak istenen görüntü ve bu görüntüye belli oranda yakın görüntüler bulunabilir. ORD, RDSS, korelasyon, kenar tespiti gibi fonksiyonlarla oldukça hızlı bir şekilde eşleme gerçekleştirilebilse de aranacak görüntülerin farklı şekil ve açılarda olması bu fonksiyonların yetersiz kalmalarına ve eşlemelerin başarısız olmasına neden olur. Çünkü söz konusu şekillerin duruşu düz olmayabilir. Sorgulamalarda göz önünde bulundurulması gereken en olası durumlardan biri olan bu durum, görüntüden çeşitli özniteliklerin çıkarılarak görüntü eşlemenin gerçekleştirilmesini gerektirir. Bu işlem görüntü işleme kütüphanesinin hazır fonksiyonları görüntülere uygulanarak gerçekleştirilebilir, fakat bu çalışmada YSA dışında kullanılan görüntü işleme kütüphanesi fonksiyonları bu gürbüzlukte olan yöntemler değillerdir.

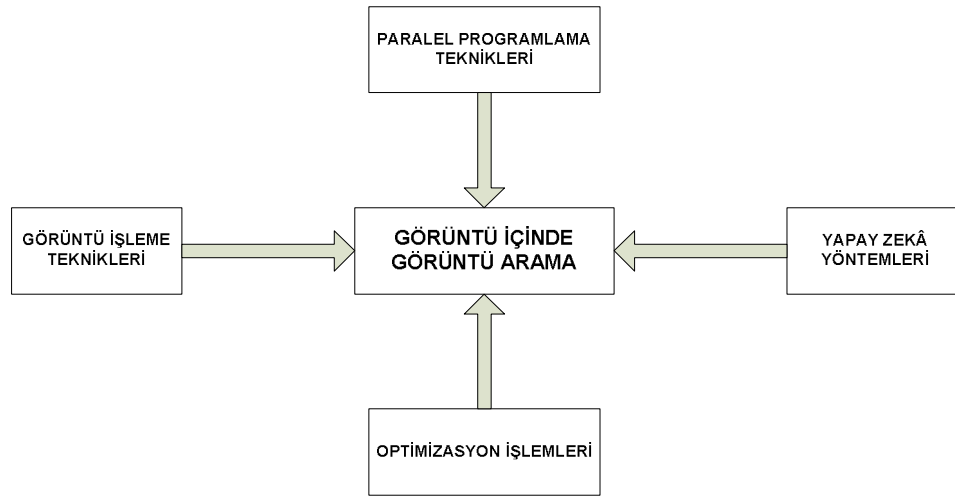
4.6 Görüntü İşleme Filtreleme Seçenekleri

Görüntü arama çalışmasında, arama öncesinde ilgili görüntü ya da görüntü parçalarına görüntü kalitesi artırma, görüntü iyileştirme, görüntüyü daha algılanabilir

kılma gibi ön işlemler uygulanabilir. Bu işlemler görüntü işleme kütüphanesinin sunduğu filtreleme seçenekleri ile mümkündür. İlgili olunan filtre ya da filtrelerden geçirilmiş görüntüler ile aramalarda ne gibi sonuçlarla karşılaşılacağı böylelikle takip edilebilir. Bu çalışma, görüntü eşlemeleri için farklı filtreleme işlemlerini gerektirmez (YSA süreci için de alt görüntüler filtersiz olabilirdi).

4.7 Tez Kapsamında Hazırlanan Yazılımın Diyagramları

Yazılımın genel işleyişini gösteren blok diyagramı Şekil 4.3'te verilmiştir.



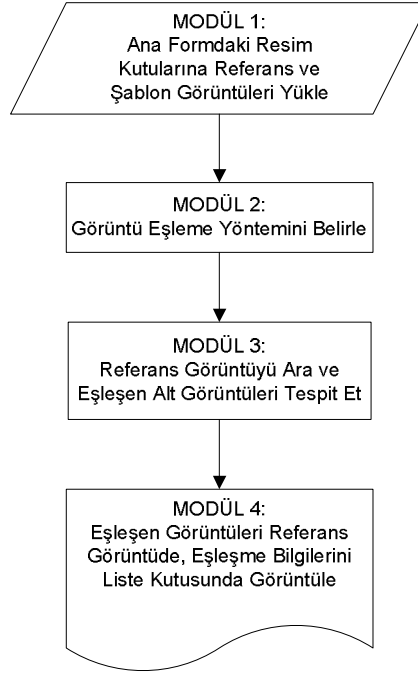
Şekil 4.3 : GİGA yazılımının blok diyagramı.

Şekil 4.3'te, tez çalışması kapsamında gerçekleştirilen yazılım genel olarak görülmektedir. Yazılımın içerdiği genel işlemler ve bu işlemlerin yaklaşımları Tablo 4.1'de verilmiştir.

Tablo 4.1: GİGA işlemleri ve yaklaşımları.

Yazılım Kapsamındaki İşlemler	Yaklaşım
GİT	Görüntü işleme kütüphanesi fonksiyonları
YZ yöntemleri	YSA ve GA
Paralel programlama teknikleri	Sınıf örnekleri ile çoklu iş parçacıkları
Optimizasyon işlemleri	Kodlama ve arayüz iyileştirmeleri

GİGA, Tablo 4.1'de görülen dört temel yapıyı içermektedir. GİGA işleminin temel süreci ise Şekil 4.4'teki akış diyagramında görülmektedir. Bu süreç, şablon görüntüyü kapsayan bir görüntüde bu şablon görüntünün aranması işlemlerini içerir.



Şekil 4.4 : GİGA sürecini gösteren akış diyagramı.

Şekil 4.4'te verilen akış diyagramı dört temel modül içermektedir. Birinci modülde referans ve şablon görüntü formdaki (arayüzler, formlar kullanılarak ve bu formlara çeşitli bileşenler eklenerek oluşturulur) resim kutularına yüklenir.

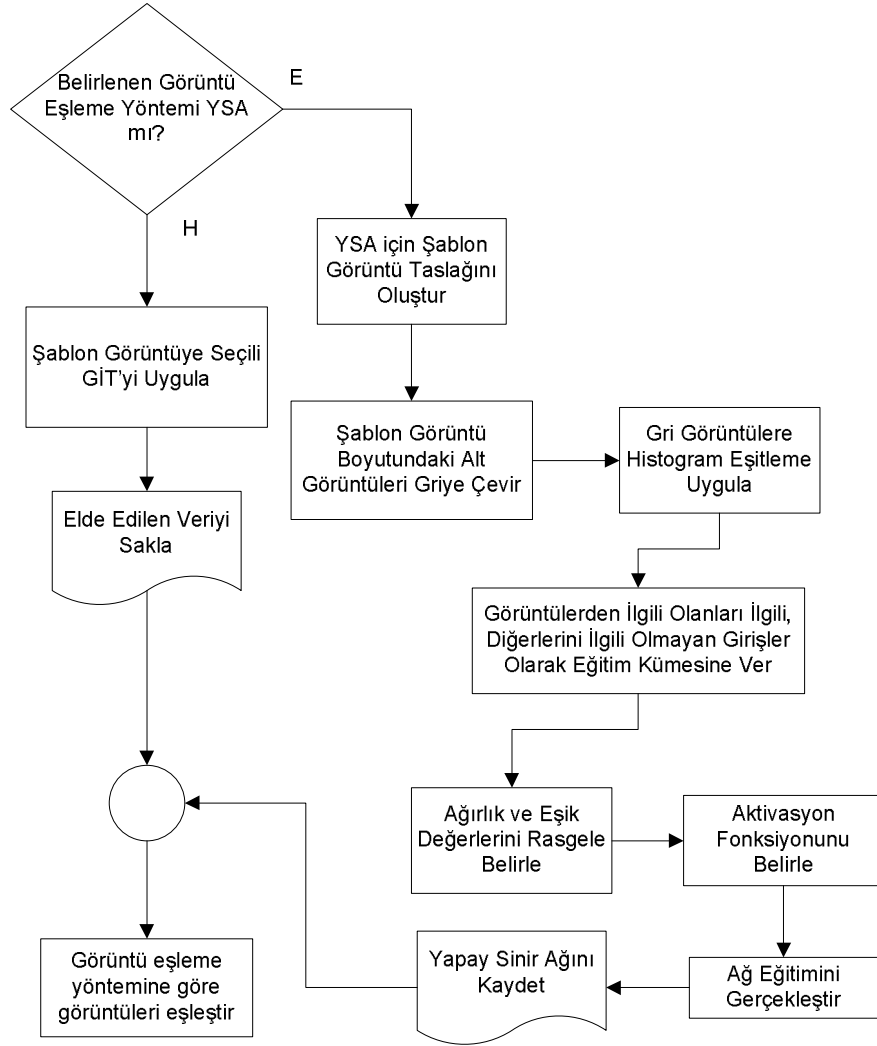
Görüntü eşlemeyi YSA ile gerçekleştirmek için sinir ağının çeşitli görüntülerle başarılı bir şekilde eğitilmesi gerekir. Daha sonra, şablon görüntüyü aramak için YSA'ya giriş olarak alt görüntüler verilir. Bu görüntüler referans görüntüden elde edilen şablon görüntü boyutundaki görüntülerdir.

Görüntü eşleme için YSA dışındaki yöntemlerin kullanılması durumunda ise şablon görüntüye ilgili görüntü işleme kütüphane fonksiyonları uygulanarak elde edilen veriler saklanır. Daha sonra, şablon görüntüyü aramak için referans görüntüden elde edilen şablon görüntü boyutundaki alt görüntüler üzerinde aynı fonksiyonlar uygulanır ve elde edilen veriler saklanan veriler ile karşılaştırılır. Görüntü eşleme için kullanılan yöntemler Tablo 4.2'de verilmiştir.

Tablo 4.2: Görüntü eşleme için uygunluk fonksiyonları ve yaklaşımları.

Uygunluk Fonksiyonları	Yaklaşım
YSA	Ağ eğitimi
ORD	Tüm pikseller
RDSS	ORD (tüm pikseller)
Korelasyon	İlişki tabanlı
Kenar tespiti	Bölümleme

Şekil 4.4'ün ikinci modülündeki görüntü eşleme süreci, YSA ve diğer GİT kullanılarak yapılan görüntü eşlemelerini içeren Şekil 4.5'teki akış diyagramında verilmiştir.



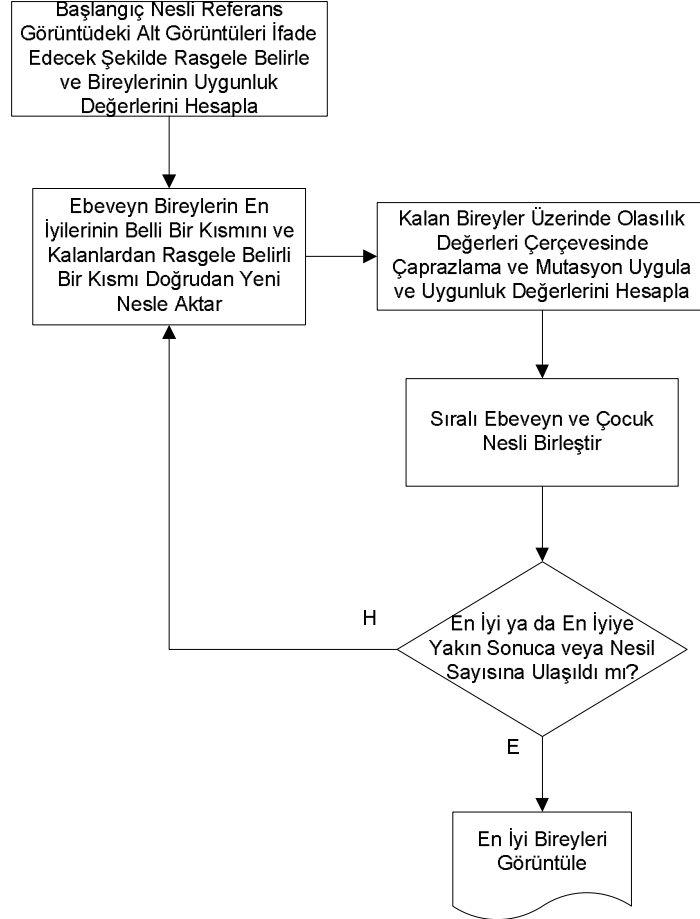
Şekil 4.5 : Görüntü eşleme sürecini gösteren akış diyagramı.

GA ile akıllı görüntü arama için başlangıç nesil rasgele oluşturulur. Nesildeki bireyler, referans görüntüden şablon görüntü boyutunda elde edilen alt görüntülerin sol üst köşelerinin referans görüntüde karşılık geldiği X ve Y değerleridir. Kromozomlar bu iki değerlerin ikilik sistemdeki karşılıkları ile gösterilirler. Her kromozom X ve Y değeri olmak üzere iki gösterim içerir (0011 ve 0100 gibi).

Bu çalışmada, çaprazlama ve mutasyon işlemleri aynı bireylere uygulanmaz. Bireylerin bir kısmına çaprazlama uygulanarak bir kısmına mutasyon uygulanarak yeni bireyler elde edilir. Elde edilen bireyler uygunluk değerleri hesaplanarak sıralı bir şekilde kodlama sürecinde kullanılan veri yapılarına aktarılırlar. Böylelikle,

ebeveyn bireylerin kopyalanarak işlemlerin yapılmasının ve bireylerin sıralanması için ayrıca zaman harcanmasının önüne geçilmiş olur.

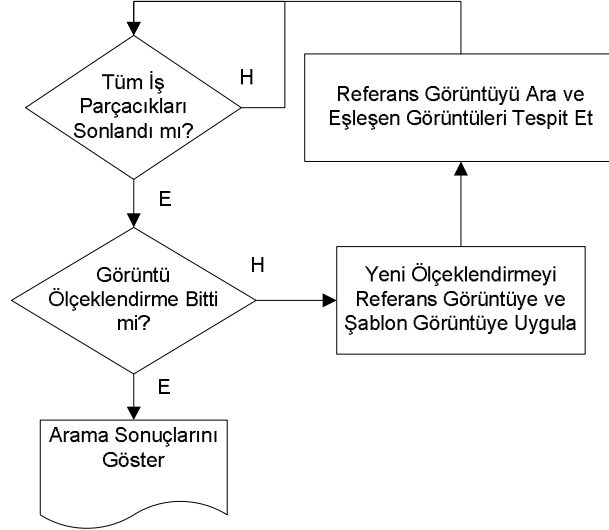
Üçüncü modülde verilen görüntü aramalarının GA ile yapılması süreci Şekil 4.6'daki akış diyagramında verilmiştir.



Şekil 4.6 : GA ile akıllı arama sürecini gösteren akış diyagramı.

Görüntü aramaları paralel olarak ya da farklı boyutlandırmalarda yapılabilir. Paralel olarak yapılmayan aramalarda tek bir iş parçacığı kullanılır. Birden fazla boyutlandırma ile yapılan aramalarda, her boyutlandırma için işlemler tekrar edilir. Bu süreci gösteren akış diyagramı Şekil 4.7'de verilmiştir.

Bu çalışmada, şablon görüntü aynı boyutta olmak koşulu ile referans görüntünün farklı çözünürlüklerinde aranmamaktadır. Uygulama, şablon görüntünün referans görüntüde aranmasıdır ve referans görüntünün farklı çözünürlükleri için şablon görüntü de aynı ölçüde boyutlandırılarak referans görüntüde aranır. Bu işlem, şablon ve referans görüntüyü aynı ölçüde boyutlandırma ile arama sürecindeki değişimin gözlemlenmesi için yapılmaktadır.



Şekil 4.7 : Paralel arama ve çoklu boyutlandırma sürecini gösteren akış diyagramı.

Bireylerin seçimi üç farklı şekilde yapılabilir:

1. Tüm bireyler arasından rasgele seçim
2. Tüm bireylerin en iyilerinin belli bir kısmı arasından rasgele seçim
3. Uygunluk değeri tabanlı rulet tekerleği seçimi

Seçilen bireylerin işlemcilerle uğratılması, çaprazlama ve mutasyon olasılıklarının üretilen rasgele değere eşit ya da bu değerden daha büyük olmasına bağlıdır ($op \geq r$).

Çaprazlama işlemi ile değerlerin tekrar etmesi gibi bir durum ortaya çıkmamaktadır. Çünkü kodlama, sadece 1 ve 0 değerleri kullanılarak yapılmıştır. Listelerdeki bu ikilik sistemdeki gösterimlerin onluk sistemdeki karşılıkları ise referans görüntüde yataydaki ve düşeydeki konumlara karşılık gelmektedir.

Yatay değerinin mx değerinden büyük olması bu değer mx değerine göre modunun alınmasını, dikey değerinin ky değerinden büyük olması ise bu değer ky değerine göre modunun alınmasını gerektirir. Bu durum alt görüntünün referans görüntüde yer alması gereğinden ötürüdür. Böylelikle, bu işlemler için ayrı koşulların getirilmesine gerek kalmamış ve arama süresinin uzamasına sebep olacak durumlar en aza indirilmiştir.

GİGA için parametre değerleri Tablo A.3'te verilmiştir.

5. YAZILIM VE ARAYÜZ

Bu bölümde tez çalışmasının uygulama kısmını oluşturan kodlama ve arayüz anlatılmaktadır.

Tez çalışması kapsamında yazılım için C# programlama dili kullanılmıştır. Uygulamaya görüntü işleme filtreleme seçenekleri dâhil edilmemiştir. Arayüzde görüntüleri görüntülemek için kullanılan resim kutuları, bu çalışmada kullanılan açık kaynak kodlu görüntü işleme kütüphanesi bileşenleridir. Bu resim kutuları kullanılarak kütüphane dâhilindeki tüm fonksiyonlardan yararlanılabilmektedir.

Gerçekleştirilen yazılımın çalışma aşamaları, takip eden kısımlarda ayrıntılı olarak anlatılmaktadır.

5.1 Yazılım Kapsamında C# Programlama Dili Kullanılması

Nesneye yönelik bir programlama dili olan C# ile gerçek dünyadaki nesnelere yazılım nesnelere¹ bire bir eşlenebilmektedir. Bu durum yazılım tasarımını kolaylaştırmakta ve anlaşılabilirliği artırmaktadır.

Modern programlama platformlarından biri olan .NET ortamı dillerinden C# ile çok esnek programlar yazılabilmektedir. Bu tez için geliştirilen yazılımda, YSA, GA ve diğer yöntemlerin gerektirdiği tüm altyapı kolaylıkla hazırlanmıştır. Uygulamanın kodlama sürecinde hazır veri yapıları (collection classes²) doğrudan kullanılabilir. YSA'yı uygulayabilmek için hazır olarak kullanılan sınıflar ve GİT'yi kullanabilmek için kullanılan görüntü işleme kütüphanesi rahatlıkla yazılıma dâhil edilmiştir.

Uygulamanın en verimli şekilde gerçekleştirilmesi, kullanılan programlama dilinin yardım olanaklarına, sorunların çözümünün hızlandırılmasına, yazılım geliştirme

¹ C# gibi nesne tabanlı programlama dillerinde en temel yapı nesnedir. Yazılım nesnelere örnekleri yaratılarak kullanılabilirler.

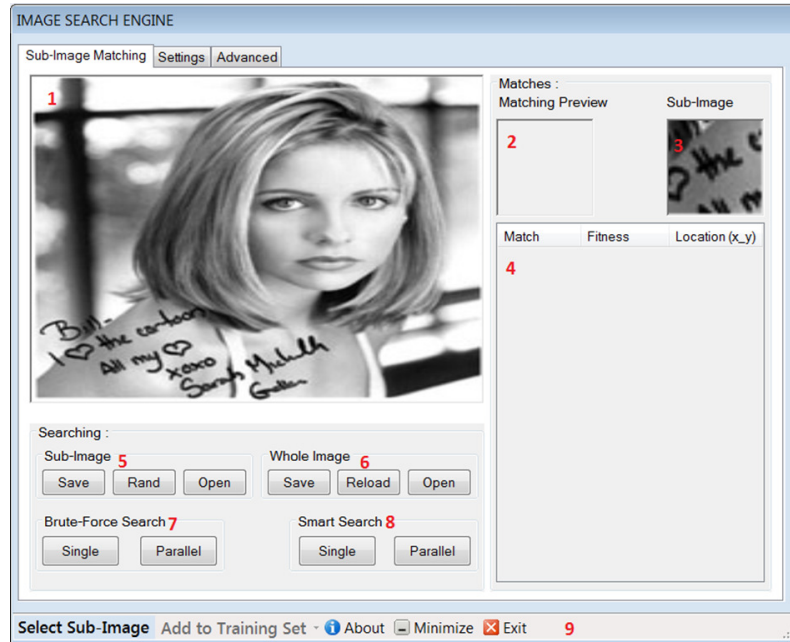
² System.Collections isim uzayı ile *collection* sınıfları ve hazır veri yapıları kullanılabilir. İlgili sınıfa "using System.Collections;" kodu eklenmelidir.

maliyetlerinin düşürülerek kalitenin artırılmasına ve programlama dilinin sürekli güncellenmesi gibi uygulamalara bağlıdır. C#'ın güçlü yapısı ile arayüzlerin oluşturulmasına imkân vermesi ve bahsedilen özellikleri taşıması, bu tez çalışmasında gerçekleştirilen uygulama için programlama dili olarak kullanılması sebeplerindedir.

Yazılım sürecinde gerçekleştirilen temel uygulamalar, GİT'nin uygulanması, YSA'nın ve GA'nın kullanılması, paralel programlama teknikleri ve optimizasyon işlemlerinden yararlanılması şeklindedir. Yazılım bu beş temel uygulamayı bütünlük içinde içerecek şekilde gerçekleştirilmiştir.

5.2 Görüntü Arama Yazılımının Başlangıç Arayüzü

C# programlama dili ile gerçekleştirilen yazılımın ana (başlangıç) formunda arama sekmesi, ayarlar sekmesi ve GA parametreleri için gelişmiş sekmesi olmak üzere üç sekme vardır. Ana formun birinci sekmesi Şekil 5.1'de görülmektedir.



Şekil 5.1 : Görüntü arama ana formu birinci sekmesi.

Ana formun birinci sekmesinde 1 numaralı resim kutusu referans görüntünün yüklendiği, 2 numaralı resim kutusu arama işlemi sonucu eşleşen görüntülerin görüntülediği, 3 numaralı resim kutusu ise şablon görüntünün yüklendiği resim kutusudur.

Formun eşleme kısmında 4 numaralı liste kutusu, arama sonuçlarının içerdiği eşleşen görüntü, bu görüntünün eşleme sırası, eşleme uygunluk değeri ve eşleme konumlarının X ve Y koordinatları ile referans görüntünün ölçek değerini görüntülemektedir. Formun 4 numaralı liste kutusunun sütunları ve bu sütunların içerdiği veriler Tablo 5.1’de verilmiştir.

Tablo 5.1: Görüntü arama sonuçları.

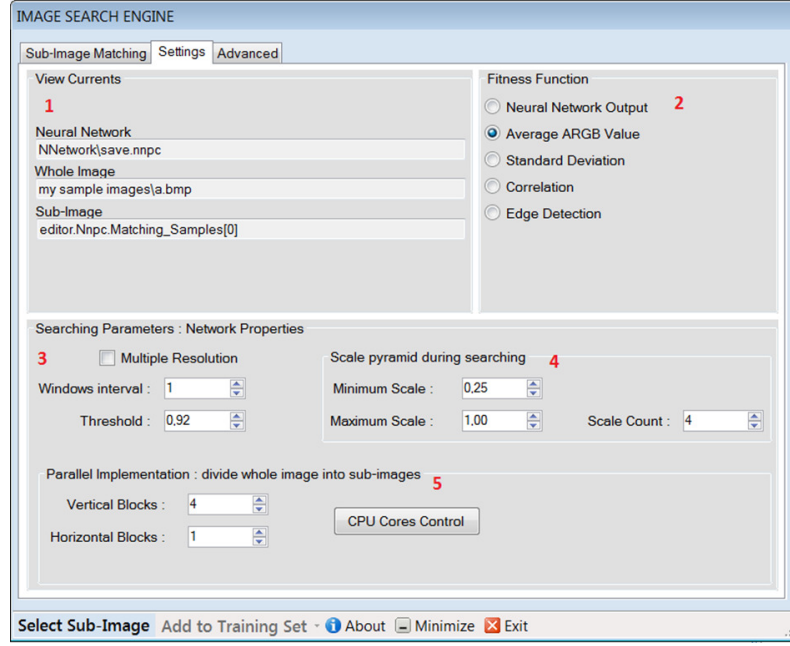
Sütun	İçerdiği Veri
Eşleşme	Eşleşen görüntü ve bu görüntünün sıra numarası
Uygunluk	Eşleşen görüntünün eşleşme derecesi
X	Eşleşen görüntünün sol üst köşesinin referans görüntüde yatay değeri
Y	Eşleşen görüntünün sol üst köşesinin referans görüntüde düşey değeri
Ölçek	Eşleşen görüntü için referans görüntünün boyutlandırma değeri

Formun arama kısmında 5 numaradaki butonlar sırayla, şablon görüntüyü kaydetmek, referans görüntüden şablon görüntü boyutlarında rasgele bir alt görüntü kırmak ve şablon görüntü yüklemek için kullanılan bileşenlerdir.

6 numaralı gruptaki butonlar sırasıyla referans görüntüyü eşleşen kısımlarla beraber kaydetmek, arama işlemi sonucunda referans görüntüde eşleşen kısımların bulunup görüntülendikten sonra referans görüntüyü yeniden yüklemek ve referans görüntü yüklemek amaçlı kullanılan bileşenlerdir.

7 numaralı grubun içerdiği ilk buton kaba kuvvet aramayı tek bir iş parçacığı ile gerçekleştirmekte, ikinci buton paralel olarak gerçekleştirmektedir. 8 numaralı grupta ise ilk buton GA ile akıllı aramayı tek bir iş parçacığı kullanarak, ikinci buton paralel olarak gerçekleştirmektedir.

Son olarak 9 numaralı grup sırayla, referans görüntüden ilgili olunan kısmın kırılması amaçlı kullanılan, arama işlemi sonucunda bulunan görüntülerden eşleşen görüntüleri ve eşleşmeyen görüntüleri ilgili YSA eğitim kümesine ekleyen (görüntü eşleme yöntemi olarak YSA seçili ise aktiftir), tez yazarı hakkında bilgi veren, formu görev çubuğuna indiren ve uygulamadan çıkılmasını sağlayan butonlardan oluşmaktadır. Ana formun ikinci sekmesi Şekil 5.2’de görülmektedir.



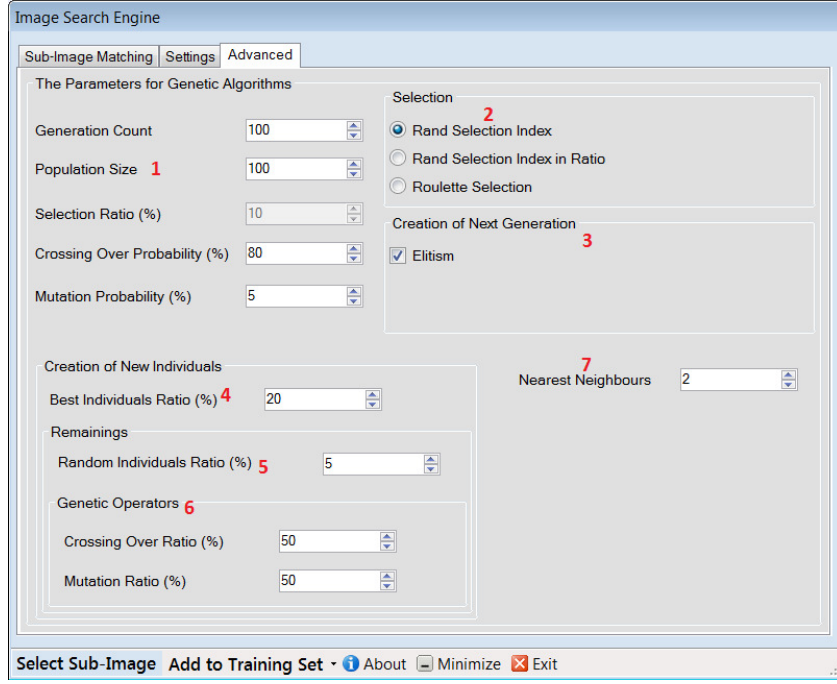
Şekil 5.2 : Görüntü arama ana formu ikinci sekmesi.

Ana formun ikinci sekmesinde 1 numaralı grup yukarıdan aşağıya (metinde alt alta sıralı verilen her bileşen grubu için geçerli) sırayla, açık olan yapay sinir ağının, referans görüntünün ve şablon görüntünün konum bilgilerini göstermektedir. 2 numaralı grup YSA, ORD, standart sapma, korelasyon ve kenar tespiti sırasıyla uygunluk fonksiyonlarını içermektedir.

Bu sekmede 3, 4 ve 5 numaralı kısımlar arama parametreleri grubunu oluşturmaktadırlar.

Arama parametreleri grubunun 3 numaralı kısmı görüntü arama parametrelerini içeren kısımdır. İlk olarak çoklu çözünürlük seçim kutusu görülmektedir ve hemen altında sırayla, pencere geçiş değeri ve eşik değeri vardır. Pencere geçiş değeri, kaba kuvvet aramalarda aranacak alt görüntülere kaçır sırayla bakılacağını; eşik değeri ise uygunluk fonksiyonlarının eşleme sonuçlarına göre toleransını ifade etmektedir. Çoklu çözünürlük seçim kutusu seçili ise 4 numaralı gruptaki parametreler aktif olur. Bu parametreler sırayla, en küçük ölçek, en büyük ölçek ve toplam ölçeklendirme sayısıdır.

5 numaralı grup referans görüntüyü dikey olarak alt bloklara ve dikey blokları da kendi içlerinde yatay olarak bloklara ayırarak paralel aramanın gerçekleştirilmesini sağlayan parametreleri içermektedir. Ayrıca kullanıcının çoklu iş parçacıklarının çalışmasından önce sistemin kaç işlemcisi olduğunu görebileceği bir buton bulunmaktadır. Ana formun üçüncü sekmesi Şekil 5.3'te görülmektedir.



Şekil 5.3 : Görüntü arama ana formu üçüncü sekmesi.

Ana formun üçüncü sekmesinde 1 numaralı kısım GA'nın temel parametrelerini içermektedir. Bu parametreler sırayla, nesil sayısı, birey sayısı, seçim oranı, çaprazlama olasılığı ve mutasyon olasılığı değerleridir.

2 numaralı grupta birey seçim yöntemleri bulunmaktadır. Bu yöntemler sırayla, rasgele bir birey seçme, tüm bireylerin en iyilerinin seçim oranı kadarlık kısmından bir birey seçme ve uygunluk değerleriyle orantılı olarak (rulet tekerleği yöntemi) birey seçme yöntemleridir.

Bir sonraki nesil yaratılırken, nesildeki ebeveynlerden en iyilerin bir kısmı ve kalanlardan rasgele belli bir kısım doğrudan yeni nesle aktarılır. 3 numaralı grupta elitizm kutusu seçilidir.

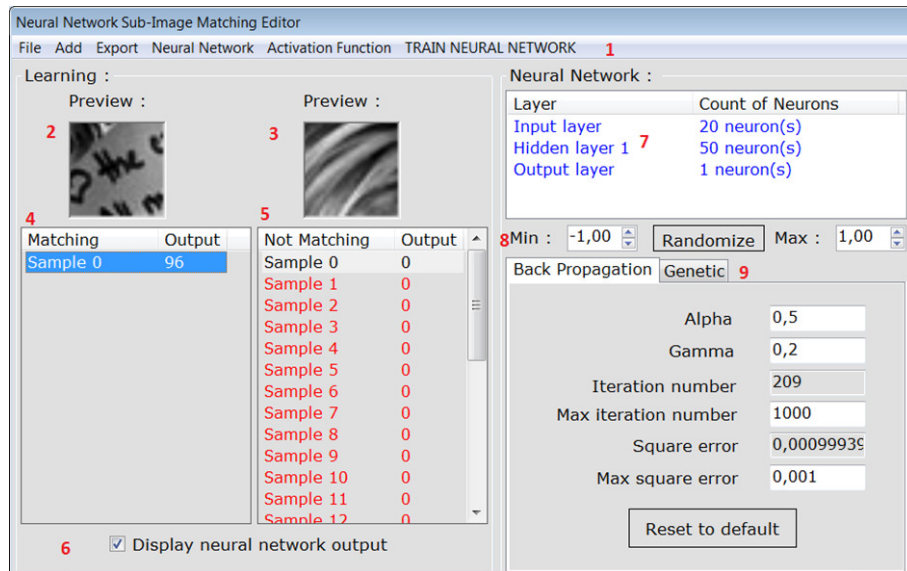
4 numarada yeni bireylerin yaratılmasında en iyi bireylerin hiç değiştirilmeden bir sonraki nesle belli oranının aktarılması ve 5 numarada kalan bireylerin belli bir oranından da rasgele bireylerin aktarılması için kullanılan parametreler vardır. Yeni bireylerin yaratılması grubunun diğer kısmını oluşturan 6 numaralı grupta genetik işlemciler için çaprazlama ve mutasyon olasılıkları görülmektedir. Bu olasılıklar, genetik işlemcilere uğratılacak bireyler arasından ne kadarının çaprazlama ne kadarının mutasyon ile yeni bireyleri oluşturacağını ifade etmektedir. Çaprazlama ve mutasyon işlemlerine uğratılacak birey oranlarının belirli olması çaprazlama olasılığı kullanımını gerektirmez, fakat çaprazlama ve mutasyon işlemleri için çaprazlama

olasılığının sağlanması durumunda çaprazlama, diğer durumda da mutasyon işlemlerin uygulanabilmesi için çaprazlama olasılığı uygulamadan çıkarılmamıştır. Mutasyon olasılığı, mutasyon işlemi sürecinde genlerin değiştirilip değiştirilmeme kararı için geçerlidir.

7 numara ise GA ile arama süreçleri ardından elde edilen alt görüntülere en yakın kaç piksel komşuluktaki alt görüntülere bakılacağını ifade eden parametreyi içerir.

5.3 Görüntü Arama Yazılımının YSA Arayüzleri

Bu çalışmada görüntü eşleme amacıyla temel olarak kullanılan YSA için ana form Şekil 5.4'te görülmektedir.



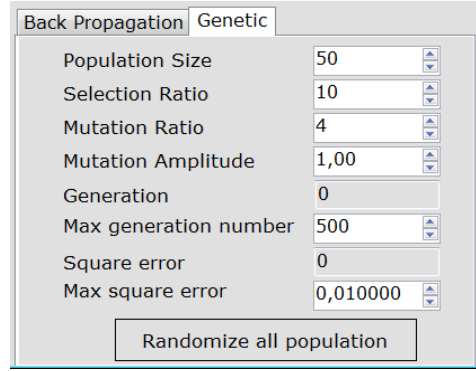
Şekil 5.4 : YSA ana formu.

YSA için kullanılan ana formda 1 numaralı kısım menü grubudur. Sırayla, YSA ile görüntü eşleme için yeni bir form açmak, kayıtlı dosyalardan birini yüklemek, kaydetmek ve formdan çıkmak için seçenekler içeren dosya menüsü bulunmaktadır. YSA eğitim kümesine eşleşen ve eşleşmeyen görüntüleri yüklemek için ekle menüsü, arama maskesi ve tüm görüntüleri kaydetmek için dışa aktar menüsü takip eden menülerdir.

Yeni sinir ağı oluşturmak, kayıtlı bir sinir ağı açmak ve bir sinir ağını kaydetmek için gereken sinir ağı menüsü; aktivasyon fonksiyonunun kullanımı için gereken ve yapay sinir ağını eğitmek için kullanılan menüler de takip eden menüleri oluşturmaktadırlar.

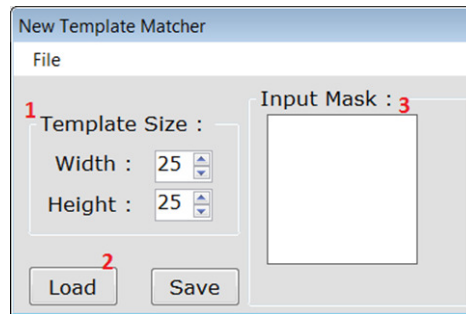
2 numara eğitim kümesindeki eşleşen, 3 numara eğitim kümesindeki eşleşmeyen görüntülerin görüntülediği resim kutularıdır. 4 ve 5 numara sırası ile eşleşen ve eşleşmeyen giriş görüntülerini içeren liste kutularıdır. 6 numara ise bu görüntülerin yapay sinir ağı çıkış değerlerini göstermek için kullanılan seçim kutusudur.

Sinir ağı grubunun 7 numaralı liste kutusu sinir ağındaki katmanların görüntülediği (ilgili katmanı çift tıklayarak), 8 numaralı kısmı ağırlık ve eşik değerlerinin en düşük (8 numarada soldaki açılan karma kutucuk) ve en yüksek (8 numarada sağdaki açılan karma kutucuk) değerler arasında rasgele belirlendiği ve 9 numaralı sekme grubu YSA'nın Şekil 5.4'te verilen geri yayılım ve Şekil 5.5'te verilen GA ile eğitilebildiği kısımlardır. 9 numaralı sekme grubu geri yayılım ve GA için parametreleri içermektedir. YSA ile görüntü eşleme için yeni bir şablon tayin etmede Şekil 5.6'daki form kullanılmaktadır.



Şekil 5.5 : YSA'nın GA ile eğitilmesi için kullanılan sekme.

YSA'yı GY öğrenme ile eğitmek için kullanılan parametreler Tablo B.1'de, GA ile eğitmek için kullanılan parametreler Tablo B.2'de verilmiştir.



Şekil 5.6 : Yeni şablon görüntü taslağı belirleme formu.

Şekil 5.6'daki formda 1 numaralı grup şablon görüntü için tayin edilecek genişlik ve yükseklik değerlerini belirtmektedir. 2 numaralı kısımda solda yer alan buton şablon taslağı (giriş maskesi) yüklemek, bu butonun sağında yer alan buton ise şablon

taslađını kaydetmek için kullanılmaktadır. 3 numaralı panel ise Şekil 5.7’de iki örneđi verilen şablon taslađının görüntülenmesi içindir ve içerdđi giriř maskesindeki siyah pikseller sinir ađına giriř vektörü olarak verilmez. Giriř maskesi için boyutlar belirlenir ve YSA’nın giriř sayısı bu maskedeki toplam beyaz piksel sayısı kadardır.



Şekil 5.7 : Örnek şablon taslakları.

Şekil 5.7’deki taslak görüntüler orijinal olarak 25×25 boyutlarındadır ve 100×100 ’e yeniden boyutlandırılmıştır. Soldaki taslak görüntü YSA ile yüz tanıma çalışmasında yüz görüntülerini tespit edebilmek için kullanılan arama maskesidir ve kullanımı ile YSA’ya giriř olarak verilecek deđerler kümesi azalır. Bu taslak ile yüz görüntülerinin arka planı dikkate alınmaz. Bu taslađın kullanılması ile şablon ve referans görüntüden elde edilen alt görüntüler maskelenerek arama yapılacak ilgili desene uyarlanabilirler. Böylelikle arama uzayı daraltılır, aramalarda sadece bu desenin ifade ettiđi kısımlar (beyaz pikseller) eşlenir ve arama süreleri kısalmır (her alt görüntü için $25 \times 25 = 625$ piksel yerine 521 piksel kontrol edilir ve alt görüntülerde karşılık gelen şablon taslak görüntüdeki $625 - 521 = 104$ siyah piksel dikkate alınmaz). Örneđin bu taslakta hiç siyah piksel olmaması (taslađın tüm pikselleri beyaz ise), referans görüntüden elde edilen şablon görüntü boyutundaki tüm alt görüntülerde hiçbir pikselin göz ardı edilmemesini gerektirir.

Sađdaki taslakta ise taslađın yatay ve düşey boyutlarında tam ortadaki pikseli beyaz diđer tüm pikselleri siyah renktedir. Bu taslak, referans görüntüde ele alınan 25×25 boyutlarındaki alt görüntülerin sadece tam ortadaki piksellerinin şablon görüntüde tam ortadaki piksel (12,12) ile eşleştirilmesini gerektirir ve alt görüntülerde karşılık gelen bu taslak görüntüdeki $625 - 1 = 624$ siyah piksel dikkate alınmaz.

YSA için şablon görüntü taslađının belirlenmesinden sonra sinir ađlarını oluşturmak amacıyla YSA’ya katman eklemek gerekir. YSA katmanlarını düzenlemek için kullanılan form Şekil 5.8’de verilmiştir.

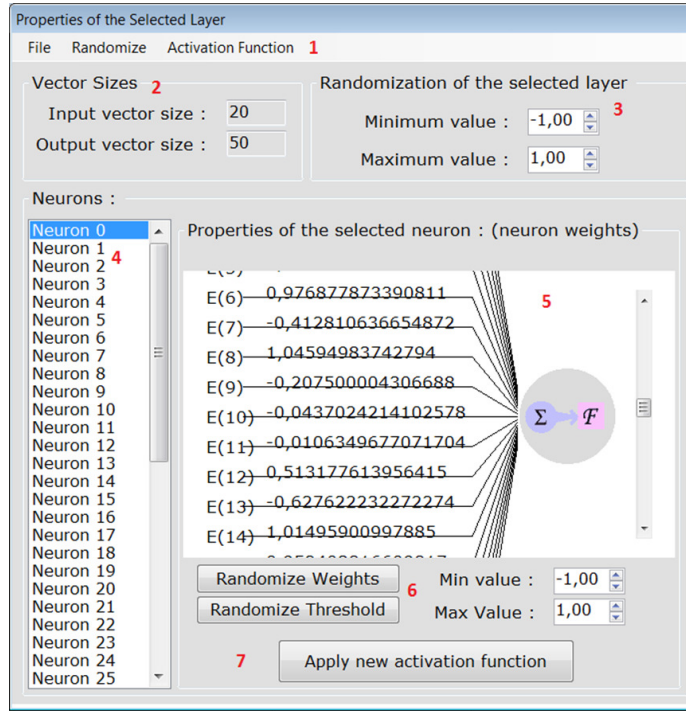
Layer	The count of neurons
Input layer	20 neuron(s)
Hidden layer 1	50 neuron(s)
Output layer	1 neuron(s)

Şekil 5.8 : YSA'nın katmanlarını gösteren form.

Şekil 5.8'de görülen formda 1 numaralı kısımdaki metin kutuları, YSA'nın giriş ve çıkış değerlerini belirtmektedir. Bu formda 30×30 boyutlarındaki bir şablon görüntü için YSA'nın giriş sayısı $30 \times 30 = 900$ 'dür. Çıkış sayısı bir olan ağ için 2 numaralı liste kutusunda görüldüğü gibi 20 sinir hücresinden oluşan giriş katmanı, 50 sinir hücresinden oluşan gizli katman ve bir sinir hücresinden oluşan çıkış katmanı vardır. 2 numaralı kısımda çıkış katmanı (en alttaki katman) dışında bir katmana sağ tıklayarak o katman güncellenebilir ya da sinir ağından çıkarılabilir. Ayrıca bu kısımda sağ tıklama ile sinir ağına yeni bir katman daha eklenebilir. Yeni bir katman eklemek (çıkış katmanına ilk olarak giriş katmanı eklenir, ardından katman eklendikçe eklenen katman giriş katmanı bu katman ise gizli (ara) katman olur) ya da seçili katmanı güncellemek için kullanılan form Şekil 5.9'da, seçili katmanı gösteren form ise Şekil 5.10'da verilmiştir.

Şekil 5.9 : YSA'ya yeni katman ekleme formu.

Şekil 5.9'daki formda YSA'ya eklenecek yeni katman için sinir hücresi sayısı belirlenmektedir.



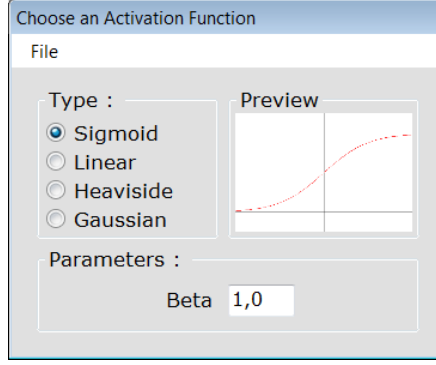
Şekil 5.10 : Seçili YSA katmanını gösteren form.

YSA'nın seçili katmanının özelliklerini gösteren Şekil 5.10'daki formda, 1 numaralı menü grubu sırasıyla formu kapatma, seçili katman için 3 numaralı gruptaki en düşük ve en yüksek değerler arasında ağırlık ve eşik değerlerini belirleme ve seçili katman için aktivasyon fonksiyonunu tayin etme menülerini içerir. 2 numaralı grup seçili katman için giriş ve çıkış vektörlerinin boyutunu gösterir.

4 ve 5 numaralı kısımlar sinir hücreleri grubunda yer almaktadır. 4 numaralı kısım seçili katmandaki sinir hücrelerini gösteren bir liste kutusundan oluşmaktadır. 5 numaralı panelde seçili sinir hücresinin ağırlık değerleri görüntülenmektedir.

6 numaralı kısım, seçili sinir hücresi için en düşük ve en yüksek değerler arasında ağırlık ve eşik değerlerinin belirlenmesi için kullanılır. 7 numaradaki buton ile seçili sinir hücresi için yeni bir aktivasyon fonksiyonu tayin edilir.

YSA eğitiminde kullanılan aktivasyon fonksiyonunun seçimi için kullanılan form ise Şekil 5.11'de görülmektedir.

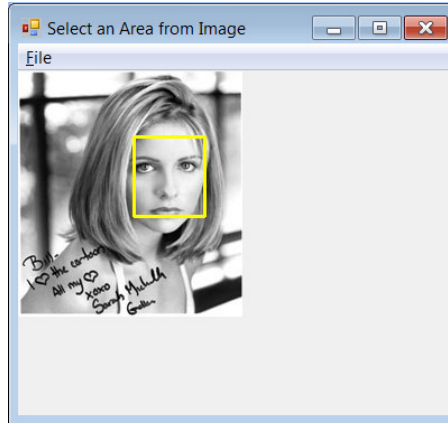


Şekil 5.11 : Aktivasyon fonksiyonu seçim formu.

Şekil 5.11’de görülen form ile dört farklı aktivasyon fonksiyonu seçilebilir. Formda bu fonksiyonların parametre değerleri görülmektedir. Formda görülen dört aktivasyon fonksiyonu EK E.1’de yer alan Şekil A.5’te ve bu fonksiyonların tanımlamaları EK E.2’de yer alan Şekil A.6’da verilmiştir.

5.4 Referans Görüntüden Alan Seçme Arayüzü

Şekil 5.12’deki form ile resim kutusuna yüklenen referans görüntüden küçük bir parça kırılarak şablon görüntü olarak seçilebilir.



Şekil 5.12 : Görüntüden alan seçme formu.

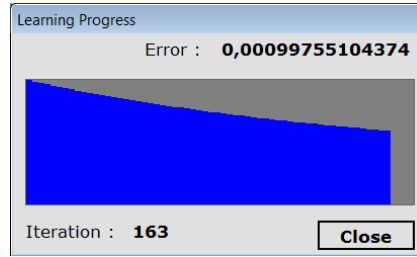
Şekil 5.12’deki form ile görüntüden bir alan elde edilerek aramak için kullanılabilir. Ayrıca bu form ile YSA’ya giriş olarak görüntü verilebilir. Bu formun dosya menüsü, yeni bir görüntü yükleme, yüklü görüntüden bir alan seçili iken yüklü görüntüyü tekrar yükleme ve yüklü görüntüyü seçili alan ile birlikte kaydetmeyi içerir.

5.5 Uygulamadaki İlerlemeleri Gösteren Arayüzler

Kullanıcıya eşleme ve arama süreçlerini arayüz olarak sunan formlar ile işlemlerin ilerlemesi takip edilebilir.

5.5.1 YSA eğitiminin ilerlemesini gösteren arayüz

YSA'nın eğitim sürecini arayüz olarak sunan form Şekil 5.13'te görülmektedir. Görüntü eşlemenin YSA ile yapılabilmesi bu sürecin verimli bir şekilde gerçekleştirilmesi ile mümkün olur.



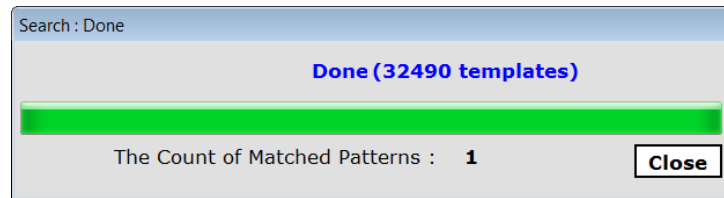
Şekil 5.13 : YSA eğitimi sürecini gösteren form.

Şekil 5.13'teki formda, ağ eğitiminin ilerlemesini gösteren bir panel, hata değerini ve tekrarlama sayısını gösteren etiketler ile eğitimi durdurarak formu kapatan bir buton bulunmaktadır. Bu form ile ağın öğrenme durumu hakkında fikir edinilebilir.

Paneldeki mavi renkli kısım ile eğitim süreci takip edilebilir. Mavi renkli kısmın olabildiğince paneli doldurması ve üst kısmının yatay düzleme paralel olması, yapay sinir ağı eğitiminin o derece az hata ile başarılı bir şekilde gerçekleştirildiğini gösterir.

5.5.2 Görüntü aramanın ilerlemesini gösteren arayüz

Görüntü arama sürecini arayüz olarak sunan form Şekil 5.14'te görülmektedir.



Şekil 5.14 : Görüntü arama sürecini gösteren form.

Şekil 5.14'te verilen formda, arama sürecinin ilerlemesini gösteren bir ilerleme çubuğu, eşleme yapılan şablon görüntü sayısını ve eşleşen görüntü sayısını gösteren etiketler ile aramayı durdurarak formu kapatan bir buton bulunmaktadır.

5.6 Görüntüleri Farklı Boyutlarda Karşılaştırmak

Görüntüler orijinal hâllerinden farklı çözünürlükte karşılaştırılabilir. Referans ve şablon görüntü ilk hâllerine göre ciddi bilgi kaybı olmayacak şekilde düşük boyutlara örneklenebilirler. Sözü geçen bilgi kaybı, görüntü içeriğinin belirsizleşmesi demektir. Görüntüler daha düşük boyutlara örneklendikçe belirsizlik artar ve görüntüde ilgili olunan kısım anlamsızlaşır. Örneğin görüntüler boyutlarının onda birine kadar küçültülebilirler. Ardından bu boyutlardan daha yüksek boyutlara kadar çeşitli aralıklarla karşılaştırılabilirler. Görüntüler ilk boyutlarından daha büyük boyutlarda da karşılaştırılabilirler. Orijinal görüntüleri belli bir boyuta kadar indirmek (örneklemek) amacıyla eşik değerleri kullanılabilir. Bu değer, görüntüde ilgili olunan kısım ya da kısımların belirginlik ve anlaşılabilirlik durumunu belirler.

Şablon ve referans görüntü olduklarından daha küçük olacak şekilde boyutlandırıldıkları takdirde aramalar daha dar kapsamda gerçekleştirilebilir. Bu noktada ifade edilen değerlendirilecek görüntü eşleme sayısının daha az olmasıdır. Uygulamanın bu kısmında, referans ve şablon görüntüyü en küçük ve en büyük boyutlandırma değerleri aralığında boyutlandırma sayısı kadar boyutlandırma imkânı sunulmaktadır.

Örneğin, referans ve şablon görüntünün yatay ve dikey boyutlarının sırayla iki katına çıkarılması ve yarıya düşürülmesi ile referans görüntüde ele alınması gereken şablon görüntü boyutundaki toplam alt görüntü sayısı 5.1’de verilen denklem grubundaki gibi değişiklik göstermektedir.

$$\begin{aligned} S &= (M - m + 1) \times (N - n + 1) \\ S &= (2M - 2m + 1) \times (2N - 2n + 1) \\ S &= ((M - m + 2) \times (N - n + 2)) \div 4 \end{aligned} \quad (5.1)$$

Görüntüler orijinal boyutlarında ya da farklı tek bir boyutlandırma ile de karşılaştırılabilirler. Farklı çözünürlükte karşılaştırma boyut problemine çözüm olarak düşünülebilir. Burada boyut problemi, referans görüntünün çok büyük ya da şablon görüntünün çok küçük olması veya arama uzayının oldukça büyük olması durumlarını içerir. Şablon ve referans görüntü daha düşük boyutlarda karşılaştırılarak arama uzayı daraltılabilir.

Görüntüler 1x1, 2x2, 4x4, 8x8, 16x16, ... 800x800 gibi boyutlarda kaydedilerek karşılaştırılabilirler. En düşük çözünürlükte eşleme söz konusu değil ise sorgulama sonlandırılabilir ve eşleme olduğu sürece bir üst boyuttaki daha yüksek çözünürlükte sorgulamalara devam edilerek işlemler sürdürülebilir. Böylece en büyük hangi boyutta görüntü eşlemenin olabileceği görülebilir ve görüntüler farklı boyutlarda karşılaştırılabilir. Ayrıca, en düşük hangi boyutta eşlemenin olabileceği de tespit edilebilir.

Referans ve şablon görüntüyü resim kutuları ile hazır olarak gelen fonksiyonları kullanarak boyutlandırmak mümkündür. Ayrıca, uygulamada en küçük ve en büyük boyutlandırma değerleri belirlenerek görüntülerin istenen boyutlandırma sayısı kadar doğrusal olarak boyutlandırılması ve işlemlerin tekrarlanması da mümkündür. Doğrusal olarak boyutlandırma, görüntülerin yatay ve dikey boyutlarını aynı değerle eşit oranda boyutlandırmaktır.

5.7 Görüntüleri Aramada Paralel Programlama Teknikleri Kullanmak

Kullanılan programlama dili paralel programlamayı icra edebilmek için gerekli altyapıyı sunmaktadır. Ayrıca paralel programlama için pek çok sınıf (class) ve program da mevcuttur. Fakat bu uygulamaların çoğunda bu kodları probleme özelleştirerek istendiği şekilde (paralel aramaların en verimli şekilde yapılabilmesi) esneklikte dâhil etmek sorun olabilir. Ayrıca birtakım bağımlılık problemleri de ortaya çıkabilmektedir.

Bu sebeplerle GİGA için geliştirilen yazılımda paralel aramaların yapılabilmesi için çoklu iş parçacıklarının çalışabilmesini sağlayacak kodlar yazılmıştır. Özellikle, çalışan iş parçacıklarını kontrol edebilme, bu iş parçacıklarının önceliklerini ayarlayabilme ve problemin belli kısımlarına uygulanarak tüm probleme çözüm olarak getirilmeleri sağlanmıştır.

Tüm iş parçacıkları her biri referans görüntüden elde edilen bir parçayı arayacak şekilde bütün referans görüntüyü ele alabilmeli ve her referans görüntü parçasında yer alan tüm şablon görüntü boyutundaki alt görüntü sayısı referans görüntüde yer alan tüm şablon görüntü boyutundaki alt görüntü sayısına eşit olmalıdır.

Paralel aramalarda referans görüntü şablon görüntüyü kapsayacak büyüklükte küçük görüntülere ayrılmıştır. Şablon görüntünün yer alması muhtemel hiçbir söz konusu

büyükükte alt görüntü ihmal edilmemiştir. Şablon görüntüyü içeren görüntü parçaları, her biri çalıştırılacak iş parçacığına referans görüntü olarak verilecek şekilde düzenlenmiştir.

Paralel programlama teknikleri kullanılırken tüm iş parçacıklarının çalıştırılması esnasında yine tümünün bitip bitmediğinin kontrol edilme sürecinde kullanılan döngüler sisteme oldukça yük bindirmiştir. Bu yük, bu esnada işlemlerin yapılması süreci ile eş zamanlı olarak sürekli tüm iş parçacıklarının bitip bitmediğinin kontrol edilmesidir. İlk başta yer alan bu uygulama görüntü arama süresini oldukça uzatmış ve verimli görüntü aramalarına engel olmuştur.

5.8 Kodlama Sürecindeki İyileştirmeler

Uygulamanın gerçekleşmesi sürecinde kodlamanın etkili ve verimli bir şekilde yapılabilmesi için şu noktalara dikkat edilmiştir:

- Uygulamanın gerçekleşmesi sürecinde nesnelere içeren *ArrayList* veri yapısı ve dizi değişkenler kullanılmamıştır. *ArrayList* veri yapısının kullanılmama sebebi, tüm uygulama sürecinde kullanılan tüm dizi gruplarının aynı türde değişkenler içermesidir. Veri yapıları üzerinde ikili arama (binary search) ve veri yapılarına verileri sıralı ekleme amaçlı özel işlemleri uygulayabilmek için listeler (*List* veri yapısı) kullanılmıştır. *ArrayList* veri yapısına nesne olarak aktarılan veriler kullanılmak için daha sonra ilgili veri ya da veri yapısı türüne tekrar çevrilmelidirler (*casting*) ve bu durum süreci uzatır.
- Kodlamada kullanılan tüm döngü yapıları gözden geçirilerek döngülerdeki her çevrimde ele alınması gerekmeyen durumlar bu yapılardan çıkarılmıştır. Döngü süresince aynı değere sahip değerlerin her seferinde hesaplanmasının ya da tanımlanmasının önüne geçilmiştir.
- Uygulamanın çalıştırılması sürecinde yapılacak görüntü aramaları için gerekli veri ve veri yapıları en aza indirilerek bu ifadeler için bellekte ayrılacak yer de en aza indirilmiştir. Kullanılmayan değişkenler için bellekte alan tahsis edilmesine ve yeni tanımlamalara gerek yoktur. Bu sebeple, tanımlı olan ve söz konusu işlemlerin yapılması esnasında kullanılmayan değişkenlerden faydalanılmıştır. Bu yolla, tüm işlemler için tanımlı tüm değişkenler ve veri yapıları en verimli şekilde kullanılmıştır.

- Sıralı listelerde indeks ya da anahtar kullanarak nesnelere erişmek hızlıdır. Bu çalışmada GA ile akıllı aramalar yapabilmek için kullanılan liste yapılarına elemanlar sıralı bir şekilde eklenmektedir. Böylece, her nesilde tüm bireyler oluşturulduktan sonra ayrıca ebeveyn ve çocuk nesil için listelerin sıralanmasına gerek kalmamaktadır.
- Nesli içeren listenin bir sonraki nesil için tahsis edilen listeye kopyalanıp kopya nesli içeren bu liste üzerinde işlemler yapılması, nesli içeren listedeki veriler (bireyler) üzerinde gerekli işlemlerin yapıp bir sonraki nesil olarak bu verilerin yeni listeye aktarılması şeklinde uygulanmıştır. Böylelikle kopyalama işlemine gerek kalmamıştır.
- Nesne tabanlı programlama dili kullanılarak gerçekleştirilen uygulamada, nesne örneklerinin gerekmedikçe yaratılmamasına özenle dikkat edilmiştir. Nesne örnekleri, nesnenin başka bir örneği yok iken gerekli ise ya da mevcut örneklerinden farklı bir örnek elde edilmeli ise yaratılmalıdır.
- Kodlama sürecinde örneği oluşturulması gerekmeyen durumlar için statik alanlar kullanılmıştır. Örneğin kaba kuvvet ya da akıllı yolla paralel arama yapmak için ilgili sınıfın birden fazla örneğinin oluşturulması gerekir. İş parçacıklarının bütünlük içinde, ama bağımsız olarak çalışabilmesi kullandıkları değişken ve veri yapılarının ayrı birer örnek olmasını gerektirir. Her iş parçacığı için ilgili sınıfın bir örneğini yaratma yoluyla tüm veriler ve veri yapıları da her ayrı sınıfın kendine ait ve bellekte ayrı yerlere sahip olurlar. Sözü geçen bu verilere ve veri yapılarına her sınıftan ayrı ayrı erişebilmek için bu alanları statik olarak tanımlamamak gereklidir. Öte yandan her sınıf örneği için aynı değere sahip (örneğin nesil sayısı her sınıf için aynı değere sahiptir ve değeri değiştirilirse yine aynı durum geçerli olur) alanlar statik olarak tanımlanırlar.
- Etkili ve verimli bir uygulama için kodun çalışma zamanında optimizasyonu önemlidir. Uygulamanın çıkış dosyasını daha küçük, daha hızlı ve daha verimli yapmak için derleyici tarafından gerçekleştirilen optimizasyonlar etkinleştirilmiştir.

- Çalışmada referans tipindeki değişkenler arasındaki dönüşümler için (int) x yapısı yerine x as^1 int yapısı kullanılmıştır.
- GA için kodlamanın ikilik sistemde yapılması, genetik işlemcileri ile elde edilen yeni bireylerin içerdiği değerlerin kontrol edilmesini gerektirir. Bu durumun sürekli kontrol edilmesi ve işlemlerin tekrarlanması, arama süresini gereksiz yere çok uzatabilir. Bu sebeple mod alma (%) işlemi kullanılarak değerlerin istenen sınırlarda kalması sağlanmıştır. Böylece, elde edilen her birey için ilgili alt görüntü referans görüntüde yer almış olur.

Görüntü eşlemenin ve aramalarının, YSA sürecinin, paralel arama ile akıllı arama işlemlerinin gerçekleştirilmesi için tasarlanan fonksiyonlar, çeşitli iyileştirmelere gereksinim duyarlar. Bu süreçlerden en verimli şekilde faydalanılması, sistemi olası her durumda en uygun şekilde çalışabilecek bir hâle getirmekle mümkün olur. Uygulama sonuçları;

- problemin olası giriş görüntülerine,
- üzerinde çalışılan sistemin özelliklerine,
- seçilen eşleme fonksiyonuna ve
- arama şekline bağlıdır.

Bu sebeplerden ötürü kodlama süreci, farklı görüntülerde ve ölçeklendirmelerde, aramaların paralel yapılıp yapılmaması durumlarında ve farklı görüntü eşleme yöntemleri ile farklı sistemlerde gürbüz bir şekilde sonuçların elde edilebileceği şekilde tasarlanmalıdır.

¹ as işlemcisi referans türündeki (veriyi saklamak yerine verinin hafızada saklanacağı adresi saklayan değişken türü) uygun (uyumlu) değişkenler arasında dönüşüm işlemlerinin yapılmasını sağlar. Dönüşüm işlemi mümkün değil ise bu durumda boş değer (*null*) döndürülür. Bu işlemci *null* olabilen türler için de kullanılabilir.

Bu çalışmada, YSA için dört farklı aktivasyon fonksiyonu olması ve YSA'nın eğitiminin GY öğrenme ve GA ile iki farklı yolla yapılabilmesi, bu işlemcinin kullanımını gerektirir. Örneğin aktivasyon fonksiyonları için tanımlı veri yapısına bu işlem uygulanarak dört aktivasyon fonksiyonundan hangisi söz konusu ise o aktivasyon fonksiyonu ve öğrenme algoritmaları için tanımlı veri yapısına bu işlemci uygulanarak iki öğrenme yönteminden hangisi söz konusu ise o yöntem belirlenerek kullanılabilir.

5.9 YSA Kütüphanesinin Sınıf Hiyerarşisi

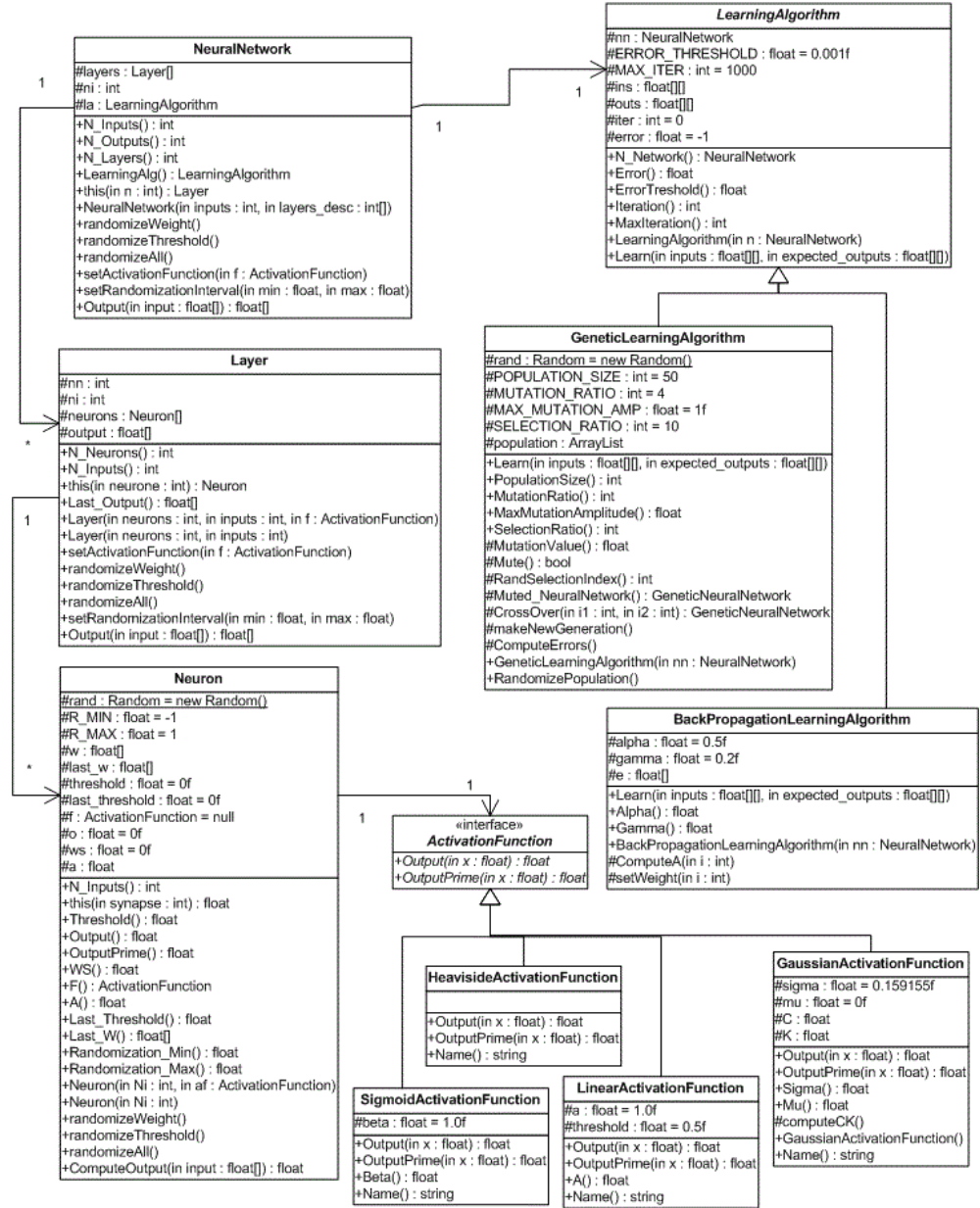
Uygulamada kullanılan ve beş ana sınıf içeren YSA kütüphanesinin¹ hiyerarşik yapısı Şekil 5.15'te verilmiştir. Bu yapıya (*uml diyagram*), kütüphaneye ulaşılan internet adresinin sınıf hiyerarşisi bağlantısından ulaşılmıştır.

Bir .NET sınıf kütüphanesi olarak C#'ta geliştirilmiş bu kütüphane, GİGA için görüntü eşleme yöntemi olarak YSA yönteminin kullanılması için bu çalışmada kullanılır. Sinir ağları tasarımı için Franck Fleurey tarafından geliştirilen grafiksel arayüzden faydalanılır. Söz konusu kütüphane, .NET sinir ağı kütüphanesinin ilk versiyonudur (sürüm). İşlemlerin daha verimli yapılabilmesi için çeşitli iyileştirmelerle ve grafiksel arayüzlerden faydalanılması ile uygulamalar icra edilir.

¹ Bu çalışmada kullanılan ve geliştirilerek GİGA için daha hızlı bir hâle getirilen YSA kütüphanesine, bu çalışma için YSA tasarımında arayüz olarak faydalanılan ve değiştirilerek çalışmaya uyarlanan kullanıcı arayüzüne <http://franck.fleurey.free.fr/NeuralNetwork/> adresinden ulaşılabilir. Aynı adresten örnek sinir ağı uygulamalarına da erişilebilir (Franck Fleurey tarafından geliştirilen sinir ağı projesi internet sayfası).

Ayrıca, bu çalışma için incelenen yüz bulma çalışması ve ilgili uygulamaya da <http://franck.fleurey.free.fr/FaceDetection/> adresinden ulaşmak mümkündür. Verilen adreslerden genel kamu lisansına erişilebilir.

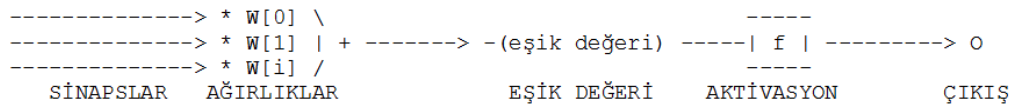
Ayrıca http://www.ri.cmu.edu/publication_view.html?pub_id=926 adresinde, Franck Fleurey tarafından gerçekleştirilen ve bu çalışma için tez yazarı tarafından incelenen çalışmanın süreci anlatılmıştır.



Şekil 5.15 : YSA kütüphanesindeki sınıflar.

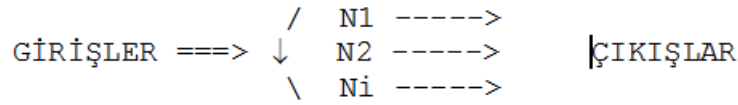
Şekil 5.15'te verilen şekilde sinir hücresi, katman, yapay sinir ağı, öğrenme algoritmaları ve aktivasyon fonksiyonlarından oluşan YSA yapısı görülmektedir. Takip eden kısımlarda her bir sınıf verilmiştir.

Yapay bir sinir hücresini sunan gösterim Şekil 5.16'da verilmiştir.



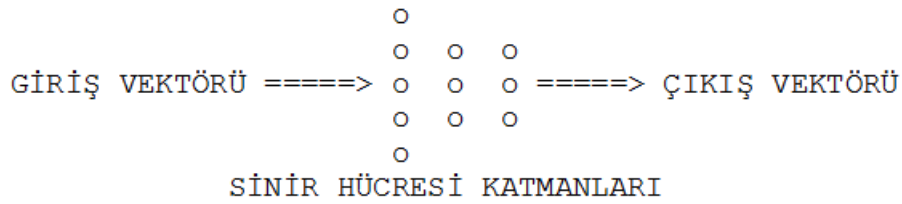
Şekil 5.16 : YSA kütüphanesinin sinir hücresi sınıfı.

Bir sinir ağındaki sinir hücresinin bir katmanı Şekil 5.17'de verilmiştir. Katmanın her sinir hücresi için bir çıkış vardır.



Şekil 5.17 : YSA kütüphanesinin katman sınıfı.

Katmanın her sinir hücresinin girişlerinin sayısı aynıdır. Bu katmanın kendi girişlerinin sayısıdır. Yapay sinir ağının uygulaması Şekil 5.18'de verilmiştir.

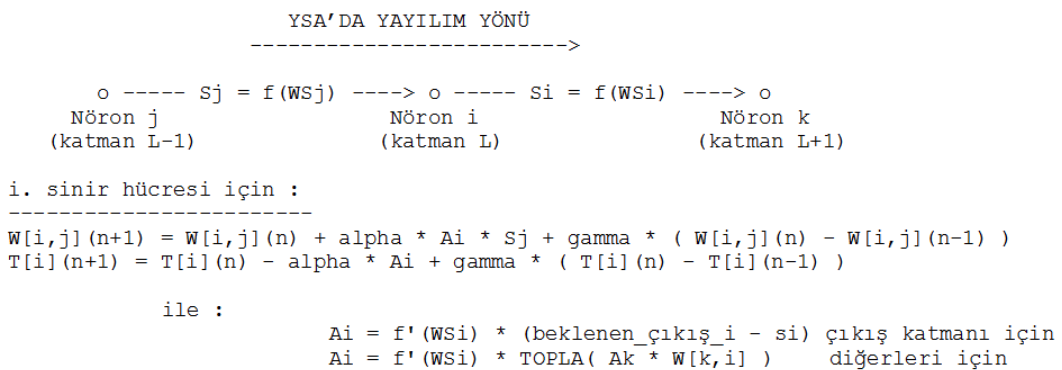


Şekil 5.18 : YSA kütüphanesinin sinir ağı sınıfı.

Çıkış katmanı dışındaki her katmanın her sinir hücresi bir sonraki katmanın her sinir hücresine bağlıdır. Başlangıçta, sinir ağı giriş ve beklenen çıkış çifti ile bir öğrenme algoritmasını kullanarak eğitilmelidir.

Bir sinir ağı için bir öğrenme algoritması tanımlamak amacıyla bu kütüphanede soyut (*abstract*) bir sınıf vardır.

Stokastik¹ tabanlı GY öğrenme algoritmasının uygulaması Şekil 5.18'de verilmiştir.



Şekil 5.19 : YSA kütüphanesinin GY öğrenme algoritması sınıfı.

¹ GY öğrenme stokastik bir süreçtir. Bu süreç, elde edilen veriler üzerinde değişimi ve davranışı inceleyerek daha sonra neler olabileceğini tahmin etmeyi, elde edilen verileri kullanarak işlemlere devam etmeyi içerir.

Bu algoritmanın stokastik tabanlı olmasının sebebi, GY öğrenme esnasında hatanın her öğrenme süreci ardından geriye yayılmasıdır.

Sinir ağlarını eğitmek için genetik algoritma uygulaması bu çalışmada kullanılan YSA kütüphanesinde yer alan bir başka YSA öğrenme yöntemidir. Nüfus, sinir ağını temel olarak içeren genetik sinir ağı örnekleri ile oluşturulur. Bir genetik sinir ağı, belirli bir sinir ağı için ağırlık ve eşik değerleri kümesini içerir ve ve genetik sinir ağı sınıfı genetik öğrenme algoritması sınıfında yer almaktadır. Genetik öğrenme algoritmasının temel yapısı şu adımları içermektedir:

- Birey sayısı kadar rasgele sinir ağı üreterek başlangıç nüfusu oluştur
 - Uygunluk fonksiyonu değerlendirmesi yap (öğrenme verileri üzerinde karesel hata (square error))
 - En iyi sinir ağlarını seç
 - Seçili sinir ağlarını yeni nesli elde etmek için çaprazla
- Yeni nesil üzerinde mutasyon uygula
- Bu süreci, elde edilen hata değeri hata için eğitim öncesi belirlenen eşik değerinden büyük olduğu sürece tekrarla

Bir sinir hücresi için aktivasyon fonksiyonu seçmek amacıyla bu kütüphanede bir de arayüz (*interface*) yer almaktadır. Bu çalışmada aktivasyon fonksiyonu olarak dört alternatif mevcuttur ve bu fonksiyonların tanımlamaları EK E.2'de yer almaktadır.

6. SONUÇ VE ÖNERİLER

Son kullanıcılar kesin nitelikler içeren nesnelere arama gereği duyabilirler. Bu nesnelere içeren ya da içermeyen görüntüler, bu nesne görüntüleri ile sorgulanmak istenebilir. Bu sebeple, GİGA amacıyla tasarlanacak sistemler belli bir nesneyi tanıma odaklı değil, şablon görüntüyü tanıma yönünde daha fazla araştırma gerektirirler.

İTGE sistemlerinin optimizasyonu ile kullanıcının aramak istediği görüntüye en yakın, en uygun görüntüleri elde edebilmesi görüntü eşleme alanında önemli bir yer tutmaktadır.

Teknolojinin ilerlemesi, görüntü veri tabanlarının hızla artması ve kapsamlarının gittikçe genişlemesi ile birlikte görsel sorgulamaların önemi artmaktadır. Öte yandan bu durum erişim hızlarının önemli bir faktör olması sonucunu doğurmuştur.

Son kullanıcının pratik çözüm gereksinimi, mevcut sistemlerin optimizasyonu ve içerik tabanlı görüntü aramalarındaki anlamsal boşlukların en aza indirilmesi gibi uygulamaları gerektirmektedir. Görsel olarak benzer görüntülerin elde edilme gereksinimi, özellikle teşhislerde görüntü eşlemeden yararlanılması ve İTGE sonuçlarının bu alandaki uzman kişiler tarafından kabul edilebilir, son kullanıcıyı tatmin edecek şekilde hızlı ve söz konusu işlemlere dâhil edilecek düzeyde olması büyük önem taşımaktadır.

Görüntü eşleme sistemlerinin mimarileri en iyi sonuçları en hızlı şekilde verebilecek yapıya getirilebilmelidir. Uzman kişilere yararlı sonuçlar verebilecek, kullanıcıların erişim sürecine dâhil olabileceği hassas ve tam olarak kullanılabilen yeterli sayıda sistem olmayabilir.

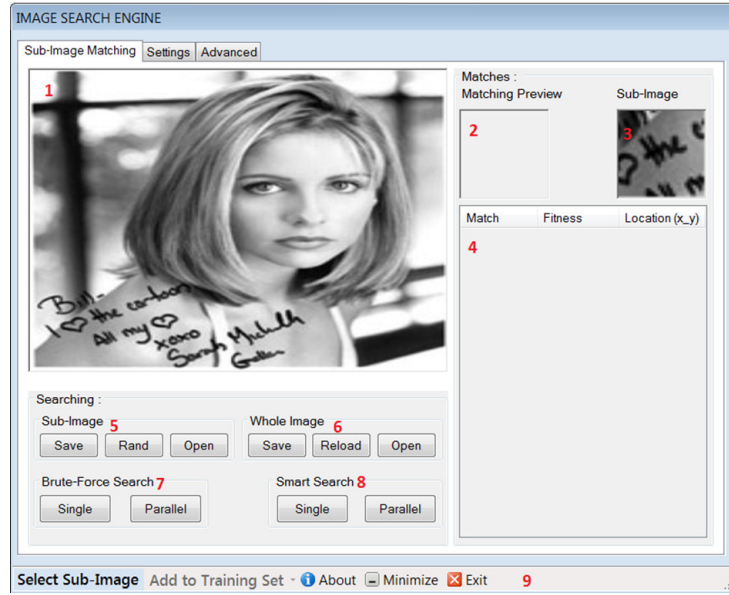
Görüntü eşlemede amaca uygun olarak kullanılacak görsel nitelikler iyi belirlenmeli, erişim kontrol modelleri ve karar destekli sistemler bir arada kullanılabilenmelidir. Probleme uygun olarak düşük seviyeli niteliklerle beraber anlamsal boşluğu en aza indirecek yüksek seviyeli nitelikler de kullanılabilir. Gruplara ayırma, dizinleme ve gerekli ise görüntüleri yeniden düzenleme gibi işlemlerden de faydalanılabilir.

GİGA'da ilgili görüntülerle birlikte ilgili olmayan görüntülerin de ele alınması gerekir. Görsel nesne gruplarını öğrenecek pek çok yaklaşım yüzlerce eğitim görüntüsüne gereksinim duyabilir. YZ yöntemleri ile akıllı arama ve arama uzayı daraltma işlemleri gerçekleştirilebilir.

Bir görüntüde önemli bir bilgi aranmak istenebilir. Bir görüntü içindeki bir alan ile ilgili olunabilir ya da görüntüler arasında çeşitli ilgiler kurulmak istenebilir.

Görsel sorgulamalar, görüntüdeki hastalıkların teşhisi, benzer hastalıkları olan hastalarla beraber görsel olarak benzer farklı hastalıkları olan hastalara erişim gibi olanaklar sağlar. Sorgulamalarda karşılaştırma ölçümleri, test verisi olarak kullanılacak görüntüler, kararlı bir başarımla gelmede tekrarlama sayısı gibi parametre değerleri önemli yer tutmaktadırlar.

Şekil 6.1'de görülen formun 1 numaralı resim kutusunda yüklü görüntüde (200×219), 3 numaralı resim kutusundaki görüntünün (30×30) YSA kullanılarak aranması dört farklı şekilde gerçekleştirilmiştir.

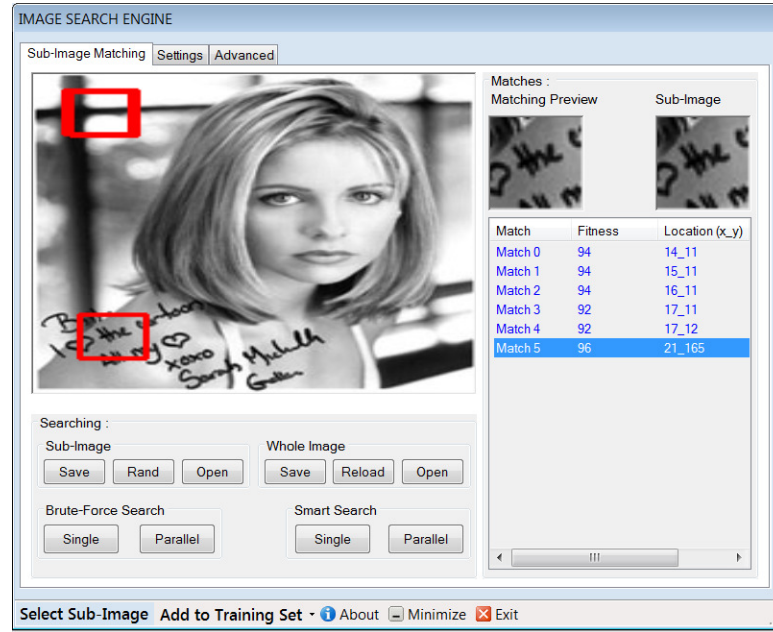


Şekil 6.1 : Referans görüntüde şablon görüntünün aranması örneği.

Şekil 6.1'deki formda görüldüğü gibi referans görüntünün bir parçası olan şablon görüntü referans görüntüde dört farklı şekilde aranabilir. Kaba kuvvet yolla tek iş parçacıklı ve paralel aramalar, GA ile tek iş parçacıklı ve paralel aramalar takip eden kısımlarda ilgili alt madde başlıklarında Şekil 6.1'de verilen aramanın sonuçları ile beraber verilmiştir.

6.1 Kaba Kuvvet Arama

Aramaların, akıllı algoritmalar kullanılmadan birer birer tüm olası durumların göz önüne alınarak yapılması kaba kuvvet aramadır. Bu çalışmada bu arama işlemi, tek bir iş parçacığı ile ve paralel olarak yapılmaktadır. Şekil 6.2’de kaba kuvvet arama örneği ve sonuçları görülmektedir.



Şekil 6.2 : Şablon görüntünün kaba kuvvet aranması örneği.

6.1.1 Tek iş parçacığı ile arama

Tek iş parçacığı çalıştırılarak tüm görüntü üzerinde yapılan arama, tek iş parçacığı ile kaba kuvvet aramadır (geçen süre (çalışma süresi): 11060 milisaniye).

Şekil 6.1’de görülen görüntü arama örneği, $32490 ((200 - 30 + 1) \times (219 - 30 + 1))$ alt görüntüyü şablon görüntü ile eşleştiren bir iş parçacığı ile gerçekleştirilmiştir.

6.1.2 Çok iş parçacığı ile paralel olarak arama

En az iki iş parçacığı çalıştırılarak referans görüntüden elde edilen alt görüntü parçaları üzerinde gerçekleştirilen arama, çok iş parçacığı ile kaba kuvvet aramadır (geçen süre: 7621 milisaniye). Her iş parçacığı ile bir referans görüntü parçasında şablon görüntü aranır.

Şekil 6.1’de görülen görüntü arama örneği, 32490 alt görüntüyü şablon görüntü ile eşleştiren dört iş parçacığı ile gerçekleştirilmiştir. Her iş parçacığının referans görüntüde aradığı görüntü kısımları EK F.1’de üçüncü sırada (c) verilmiştir. EK

F.1’de yer alan Şekil A.7’de verilen diğer referans görüntü parçaları ise bu çalışmada kullanılan referans görüntülerden ilgili şablon görüntüye, yatay ve dikey blok sayılarına bağlı olarak elde edilen alt görüntülerdir.

6.2 Akıllı Arama

Aramaların, akıllı algoritmalar kullanılarak tüm olası durumların göz önüne alınmadan yapılması akıllı aramadır. Bu çalışmada bu arama işlemi, tek bir iş parçacığı ile ve paralel olarak yapılmaktadır.

GA ile yapılan aramalarda en iyi çözüme yakın çözümler bulunduğu durumda, oldukça kısa süreli kaba kuvvet taramalarla daha iyi çözümler bulunabilmektedir.

6.2.1 Tek iş parçacığı ile arama

Tek iş parçacığı çalıştırılarak tüm görüntü üzerinde yapılan akıllı arama, tek iş parçacığı ile akıllı aramadır (geçen süre: 3017 milisaniye).

Şekil 6.1’de görülen görüntü arama örneği, 100 nesil süren bir iş parçacığı ile gerçekleştirilmiştir.

6.2.2 Çok iş parçacığı ile paralel olarak arama

En az iki iş parçacığı çalıştırılarak referans görüntüden elde edilen alt görüntü parçaları üzerinde gerçekleştirilen akıllı arama, çok iş parçacığı ile akıllı aramadır (geçen süre: 1781 milisaniye). Her iş parçacığı ile bir referans görüntü parçasında şablon görüntü aranır.

Şekil 6.1’de görülen görüntü arama örneği, her biri 25 nesil süren dört iş parçacığı ile gerçekleştirilmiştir.

6.3 Görüntü Arama Sonuçları

Bu çalışmada genel olarak paralel aramalar ile tek iş parçacıklı aramalara göre, akıllı aramalar ile kaba kuvvet aramalara göre daha başarılı sonuçlar elde edilmiştir. Akıllı aramaların paralel olarak yapılıp yapılmaması değişiklik gösterebilir. GA’nın yaklaşımının paralel olması ve aranacak görüntü için çok büyük sayıda alt görüntünün olması akıllı aramaların da paralel olarak yapılmasını gerektirebilir. Öte

yandan, arama uzayının dar olması durumunda paralel olarak aramaların yapılması başarılı sonuçlar vermeyebilir.

Şablon görüntü ile belli değerlerde eşleşen alt görüntüler, akıllı aramalar ile oldukça başarılı bir şekilde bulunabilir. Bu aramaların paralel olarak yapılması ile yine oldukça başarılı sonuçlar elde edilebilir. GA ile her zaman en iyi çözüm ya da çözümler bulunamayabilir, ama her iş parçacığı ile daha dar bir arama uzayında akıllı yollarla yapılan paralel aramalar sayesinde daha iyi sonuçlar elde etmek mümkündür.

Bu noktada başarı, sonuç görüntüler arasında arzu edilen görüntülerin bulunmasını, ama diğer görüntülerin olabildiğince az olmasını ve işlemlerin paralel bir şekilde olabildiğince hızlı yapılabilmesini içerir.

Kaba kuvvet ve akıllı aramalar için paralel yaklaşım, kullanılan bilgisayara ve sistemin arama sürecindeki durumuna bağlıdır. Görüntü aramalarının paralelleştirilmesi arama süresini genel olarak kısaltır, fakat bilgisayarın işlemcisine ve çalışan programlara bağlı olarak iş parçacığı sayısının arttırılmaya devam edilmesi arama sürecini kötüleştirir. Bu sebeple paralel aramalar sistemin özelliklerine ve görüntü arama uygulamalarına bağlıdır. Paralellik, bu durumlar göz önüne alınarak süreci iyileştirecek, ama arama süresini uzatmayak şekilde oluşturulmalıdır.

6.4 Görüntü Arama Önerileri

Bu çalışmada GİGA problemi, aramanın akıllı bir şekilde yapılmasını ve optimizasyonunu gerektirir. Paralel aramalar için referans görüntünün şablon görüntüyü kapsayacak boyutlarda görüntü parçalarına ayrılması gibi şablon görüntü de alt parçalara ayrılabilir. Fakat bu durum, çok çeşitli koşulların göz önüne alınmasını gerektirir. Aranılan görüntünün bütün olarak değerlendirilmesi en doğru yaklaşımdır. Ayrıca şablon görüntünün de alt parçalara ayrılması, uygunluk değerlendirmelerinin doğru bir şekilde yapılmasına engel olur.

Görüntü aramalarının kaba kuvvet bir şekilde yapılmaması için tüm olası durumları değerlendirmeden aramaları yapabilecek akıllı arama algoritmaları gerekir. Bu yöntemler, arama sürecinde gittikçe daha iyi bir sonuç verecek yapıda olmalıdırlar. Birden fazla olası çözüm içeren çözüm kümesi yaklaşımı ve eldeki veriler ile gittikçe daha iyi verileri elde etmeyi amaçlayan GA, arama işlemi için oldukça uygun bir yöntemdir. Bu çalışmada ters çevirme, bir yöne öteleme, araya sokma, yer değiştirme

ve kırpma gibi mutasyon işlemleri kullanılabilirdi. Gen üzerinde küçük deęişiklikler yapan mutasyon işlemine, gen grupları üzerinde büyük deęişiklikler yapan mutasyon işlemleri de dâhil edilerek çeşitlilik sağlanabilirdi.

Genetik işlemciler için farklı yollar izlenerek yeni bireylerin elde edilmesinde çeşitlilik sağlanabilir. Ek yöntemler ve koşullar, arama sürecinde çeşitlilięi sağlarken aynı zamanda arama süresini de uzatabilir. Bu yüzden, işlemlerin olabildiğince dikkatli ve arama süresini gereksiz yere uzatmayacak şekilde yapılması gerekir.

Örneğın arama süresini kısaltmak için bölme işlemleri farklı yaklaşımlarla gerçekleştirilebilir, çünkü C# programlama dili bölme uygulamaları için zaman alıcı olabilmektedir. Fakat, bölme işlemlerinin yoğun olarak tamsayı deęerler üzerinde icra edildięi uygulama kısımlarının bu çalışmada yer almaması, YSA eğitim ve test aşamalarında doğrusal olmayan fonksiyonların kullanımı ve arama işlemlerinde bölme işlemleri için hassas sayıların kullanımı bölme işlemleri için farklı bir yaklaşımın kullanılmasını gerektirmemektedir.

6.5 Görüntü Eşleme Sonuçları

Görüntü eşleme için kullanılan YSA oldukça başarılı sonuçlar vermektedir. YSA'nın eğitimi zaman alır ve doğru görüntü eşlemeleri için güncellenmesi gerekebilir. YSA'nın pek çok üstün yönü görüntü tanıma amacıyla başarılı bir şekilde kullanılmasını sağlar.

Bu çalışmada YSA dışında görüntü eşleme için kullanılan yöntemler oldukça hızlı eşleme yöntemleridir. Bu yöntemler genel olarak hızlı görüntü aramaları ve ilgili yöntemlere göre fikir edinmek için kullanılabilirler. Görüntü işleme kütüphaneleri kapsamında olan bu yöntemlere göre görüntü tanımda YSA oldukça başarılı bir yaklaşımdır.

YSA ile görüntü eşleme amacıyla sinir ağı eğitimi, diğer yöntemler ile görüntü eşleme amacıyla görüntü işleme kütüphane fonksiyonları gerekir. Görüntü eşlemelerinin gürbüz bir şekilde yapılabilmesi için YSA diğer görüntü eşleme yöntemlerine göre üstündür. Görüntü eşlemeleri öncesinde eğitim işlemleri ve ek başka uygulamalarının gerekmemesi ise YSA dışındaki yöntemlerin YSA'ya göre uygun taraflarıdır.

6.6 Görüntü Eşleme Önerileri

Bu çalışmada GİGA sürecinin verimli ve hızlı bir şekilde gerçekleşmesi, arama kadar eşlemenin de hızlı ve etkili bir şekilde yapılması ile sağlanabilir. Görüntü eşlemeleri için çok farklı yaklaşımlar izlenebilir. Bu yaklaşımlar tek bir tekniği ya da birden fazla tekniğin bir arada kullanımını içerebilir.

Örüntü tanımada oldukça başarılı bir yöntem olan YSA kullanılarak görüntü aramak için paralel aramaların kullanılması ve ağ eğitiminin aranmak istenen görüntüye göre verimli bir şekilde yapılması gerekir.

ORD ve RDSS kullanılarak yapılan görüntü aramalarında ise istenenden çok daha farklı görüntüler de elde edilebilir. Bu görüntülerin en aza indirilmesi için şablon görüntü ve sonuç görüntüler birbiri ile örtüşecek şekilde küçük parçalara ayrılarak bu görüntü parçalarından elde edilen yeni ORD ve RDSS karşılaştırılabilir. Bu süreç, doğru olmayan sonuç görüntüler olduğu sürece, şablon görüntü parçaları ve sonuç görüntü parçalarına aynı işlemler uygulanarak sürdürülebilir.

GİGA için YSA kullanılırken örneğin iki sinir ağından faydalanılabilir. Arama sürecinin bir aşaması şablon görüntünün olabildiğince iyi bir şekilde öğretildiği, diğer aşama ise bu seviyeye yakın seviyede öğretildiği birer sinir ağı içerebilir. İkinci aşamada sözü geçen YSA kullanılarak aranan görüntüler üzerinde birinci aşamada sözü geçen YSA icra edilebilir ve arama sonuçlarındaki değişim gözlemlenebilir. Ayrıca, şablon görüntü için başarılı bir şekilde eğitilen iki sinir ağı kullanılarak bir sinir ağı ile arama işleminin tamamlanmasının ardından diğer sinir ağı ile doğrulama işlemleri icra edilebilir. Böylelikle, eğitim ve test süreçlerine doğrulama süreci de dâhil edilerek arama süreci iyileştirilebilir.

Görüntü eşleme amacıyla örneğin YSA yöntemi dışında bir yöntem kullanılabilir. Arama süreci hızlı bir şekilde gerçekleştirildikten sonra YSA yöntemi sonuç görüntülere uygulanarak şablon görüntü tespitinde daha verimli ve etkili olabilecek bir sonuç elde edilebilir.

6.7 GİGA'nın Kullanımı ve Uygulanabilirliği

Bu çalışmadaki uygulama, farklı alanlar için özelleştirilebilir ve pek çok alanda büyük görüntü veri tabanlarındaki her bir görüntüde bir şablon görüntünün aranmasını sağlayabilir. Bu alanlar ve uygulanabilirlik şu şekilde ifade edilebilir:

- Kişisel kullanımlar için ilgili olunan bir görüntünün başka bir görüntüde aranması
- Görüntü veri tabanları üzerinde İTGE'nin sağlanması
- Emniyet, hastane, eğitim kurumları, işletmeler gibi pek çok kurum ya da kuruluşta görüntü içerikli aramalar
- Çeşitli gereksinimler doğrultusunda görüntüleri başka görüntüler ile sorgulama
- Görüntü veri tabanları yönetim sistemleri
- Taslak görüntüleri ya da çizimleri başka görüntülerde sorgulama ve bilgi çıkarımı
- Tıpta bilgisayar destekli tanı
- Karakollarda kanıta dayalı çıkarımlar
- Çeşitli teşhis, saptama ve güvenlik uygulamaları

Bu çalışma, görüntü içerikli erişimlerin görüntüdeki herhangi bir alan üzerinde gerçekleştirilmesini sağlamaktadır. Söz konusu alan, yüz ya da parmak izi görüntüsü ile sınırlı değildir. Bu çalışmada, görüntü eşleme genelleştirilmiştir ve esas olarak görüntü arama süreci üzerinde durulmuştur. Görüntü arama süreci etkili görüntü eşleme işlemlerini de gerektirir. Bu sebeple, görüntü aramanın etkili olabilmesi için tüm görüntü eşleme süreci en verimli şekilde getirilmiştir.

6.8 Uygulama Platformu Önerisi

Bu çalışmadaki uygulama, teknik hesaplamalar ve matematiksel problemlerin çözümü ve analizi için tasarlanmış, özellikle mühendislik alanındaki sistemlerin analizlerinde kullanılan bir yazılım geliştirme aracı olan MATLAB ile gerçekleştirilebilirdi. MATLAB'in görsel yazılımlar geliştirmek için sunduğu

grafiksel kullanıcı arayüzü geliştirme aracı, görüntü işleme araç kutusu ve yüksek seviyeli programlama imkânı ile yüksek kalitede bir hızlı uygulama geliştirme ortamından yararlanılabildi. Bu uygulama için nesne tabanlı programlama desteği ve şu araçları ile MATLAB çok iyi bir ortam sunabilir:

- Görüntü işleme
- YSA
- Optimizasyon
- Algoritma tasarlama, uygulama geliştirme ve kullanıcı arayüzleri yaratma
- Modelleme ve benzetim
- Bilgi toplama, veri analizi ve veri inceleme

6.9 Uygulama Sürecinde Karşılaşılan Zorluklar

Tez çalışması sürecinde birtakım sorunlarla karşılaşmış ve sorunların çözülmesine ilişkin bazı işlemler yapılmıştır. Bu sorunlardan takip eden alt madde başlıklarında bahsedilmektedir.

6.9.1 Ön görüntü işleme teknikleri

Bu çalışmada, referans görüntüde bir şablon görüntünün aranması sürecinde çeşitli ön GİT kullanılmaktadır. Referans görüntüde herhangi bir problemin çözümüne yönelik işlemler değil, şablon görüntü ile kesişen birtakım işlemlerin yapılması gerekmektedir. Referans ve şablon görüntülerin aynı süzgeçlerden geçirilerek aramaların icra edilmesi gerekmektedir.

Şablon görüntü üzerinde yapılan işlemler sonucu elde edilen veriler, referans görüntüden şablon görüntü boyutunda elde edilen görüntüler üzerinde aynı işlemlerin yapılması ile elde edilen verilerle karşılaştırılmalıdır.

Görüntü eşlemeleri, bu işlemlerin yapılmasının ardından gerçekleştirilebilir. Renk tabanlı, şekil tabanlı ya da bütünleşik bir eşleme yöntemi için görüntülere uygulanan GİT farklılıklar göstermektedir.

YSA ile görüntü eşlemede, şablon görüntü için tahsis edilen desene göre eşlemelerin yapılması, eşleme süresini kısaltabilmektedir. Desenin bütün olarak ele alınması,

şablon görüntünün tümüyle referans görüntüde aranmasını gerektirir. Bu noktada, kullanılan eşleme yöntemleri farklılıklar göstermektedir.

Referans görüntüden belli bir iş için birtakım verilerin elde edilmesi, bu görüntünün yeterince belirgin ve kaliteli olmaması durumunda, gürültü giderme ve birtakım süzgeçlerden geçirilmesini gerektirebilir. Fakat bu çalışmada, GİGA problemi, referans ve şablon görüntü üzerinde ortak işlemlerin yapılmasını gerektirir. Referans görüntünün seçilen bir kısmının bu görüntüde aranması için ön filtreler gerekmez iken; herhangi bir şablon görüntü aramada ön işlemler gerekebilir. Bu durum, görüntülerin kamera, fotoğraf makinesi ya da bilgisayar ortamı kullanılarak elde edilmesine bağlıdır. Kullanıcı, arama gereksinimine göre görüntü işleme kütüphanesinin çeşitli ön fonksiyonlarını kullanabilir.

Resim üzerindeki gürültülerin, belirsizliklerin ve gereksiz ayrıntıların giderilmesi, renk değerlerinin ayarlanması ve istenen ayrıntıların ortaya çıkarılması gerekebilir.

Görüntülerin kenarlarının geniş çerçevede tespit edilmesi için *Canny*¹ kenar bulma algoritması kullanılmıştır. Elde edilen siyah beyaz görüntüler görüntü işleme kütüphanesi yardımı ile karşılaştırılmıştır.

6.9.2 GA için kromozomların kodlanması ve uygunluk

Bu çalışmada akıllı aramaların GA ile yapılması, ilk olarak referans görüntüdeki şablon görüntülerin sol üst köşe piksellerinin yatay ve dikey konum değerleri, ikinci olarak bu değerlerin ikili sistemde gösterimleri ile kodlanarak gerçekleştirilmiştir.

Birinci durumda, çaprazlama işlemi iki şablon görüntü için sadece X ve Y değerlerinin karşılıklı değiştirilmesi ile mutasyon işlemi bu değerlerin kendi içlerinde değiştirilmesi ile yapılmıştır. Çaprazlama işlemi ile elde edilen, bir alt görüntünün X değeri ile diğer bir alt görüntünün Y değerinin referans görüntüde karşılık geldiği alt görüntünün sol üst köşesi olmuştur. Mutasyon işlemi ile elde edilen, bir alt görüntünün X ya da Y değerinin değiştirilmesi ile referans görüntüde karşılık gelen alt görüntü olmuştur.

Birinci kodlama şekli, bir alt görüntü için sadece iki değerden ibaret bir gösterimdir ($X : 15, Y : 35$ gibi).

¹ Canny, 1986'da John F. Canny tarafından geliştirilen ve kenarları bulan çok aşamalı bir kenar bulma algoritmasıdır.

İkinci durumda, onluk değerler ikili olarak kodlandığından, çaprazlama ve mutasyon genler üzerinde icra edilerek yeni bireylerle her yere ulaşılabilmesi sağlanmıştır. İkinci kodlama şekli ile bir alt görüntünün X ve Y değerleri, çok sayıda 0 ve 1 değerleri ile ifade edilen bir gösterime çevrilmiştir ($X : 001111$, $Y : 100011$ gibi). Bu aşamada, ilk olarak farklı (X,Y) noktaları üretilerek yatay ve düşey değerleri ikilik sisteme çevrilmiştir. Daha sonra, başlangıç nesilde çeşitliliğin sağlanması için 0 ya da 1 olarak her kromozom için bit bit rasgele değerler üretilmiştir.

İkinci kodlama şekli ile çaprazlama işlemi kromozomların genleri arasında, mutasyon işlemi kromozomların genleri üzerinde gerçekleştirilmiştir.

Akıllı aramaların yapılması sürecinde ilk olarak bireylerin uygunluk değerlerine göre yeni bireylerin seçimi gerçekleştirilmiştir. Bu işlem, yeni bir bireyin iki bireyden uygunluk değeri daha iyi olana daha yakın olacak şekilde belirlenmesi ile yapılır. Fakat her yeni birey için eldeki bireylerin tekrar tekrar uygunluk değerlerinin hesaplanması ve daha sonra yeni bireylerin uygunluk değerlerinin de hesaplanması işlemleri oldukça yoğunlaştırmıştır. Bu yüzden EK B.1'de C# kodları verilen bu işlemler süreçten çıkarılarak genetik işlemcilerine ve sürecin iyileştirilmesine odaklanılmıştır.

Nüfustaki bireylerin uygunluk değerlerine göre yeni birey eldesi, işlemlerin tamamen rasgele olmasının önüne geçebilir, fakat rasgelelik ve daha iyi sonuçlar elde etme üzerine kurulu olan GA yöntemi için işlemlerin oldukça fazlalaşarak arama süresinin uzamasına neden olur. Bu yüzden şu gibi uygulamalar yapılabilir:

- Süreç tekrarının artırılarak evrimleşmenin devamı ve daha iyi birey elde etme olasılığının artması
- Nüfustaki birey sayısını artırarak daha fazla çeşitlilik sağlamak ve paralel arama için çözüm kümesini genişletmek
- Aramalar için parametre değerlerini değiştirerek daha uygun hâllere getirmek
- Genetik işlemcileri ve yapıyı genişleterek farklı birey eldesi

Bu işlemlere görüntü arama sonuçları ve gözlemlenen süreçlere göre devam edilerek daha iyi sonuçlar elde edilebilir.

6.9.3 Görüntü aramalarının paralelleştirilmesi

Bu çalışmada paralellik, aramaların çok iş parçacığı ile yapılması ve akıllı aramalar için GA kullanılmasını içermektedir.

6.9.3.1 Çok iş parçacığı paralelliği

Görüntü aramalarının paralel olarak yapılabilmesi, iş parçacıklarının aynı yere müdahale edebilme ve karşılık gelme durumlarının ortadan kaldırılması ile tümünün sonlanıp sonlanmadığının kontrol edilmesini gerektirir.

İş parçacıklarının tümünün bitip bitmediğinin kontrol edilmesi için yazılan C# kodları EK B.2’de verilmiştir. Bu kodların, arama sürecinde işletim sistemine çok yük bindirdiği ve görüntü aramalarının verimli bir şekilde yapılmasına engel olduğu görülerek kaldırılmasına karar verilmiştir.

6.9.3.2 Akıllı arama paralelliği

Akıllı görüntü aramalarının GA ile yapılabilmesi, mimarinin probleme en uygun şekilde oluşturulması ve işlemlerin olabildiğince verimli bir şekilde yapılabilmesine bağlıdır. Bu yüzden, bireylerin ifade edildiği aynı uzunluktaki veri yapıları üzerinde işlemlerin ve çözüm kümesinde değerlendirmelerin çok dikkatli yapılması gerekir. Bireyler için X değeri mx değerini aşmamalı, Y değeri ise ky değerini aşmamalıdır ($X = X \% mx$, $Y = Y \% ky$).

Başlangıç nesilde kromozomların bit bit yaratılması ve bireylerin işlemlere tabi tutulması ile yeni bireylerin elde edilmesi, GA için kullanılan veri ve veri yapıları üzerinde işlemler açısından büyük dikkat gerektirir. Listedeki bireyler için ikili gösterim uzunlukları değişiklik göstermemeli ve bireylerin çeşitli işlemlere tabi tutulması için uzunluk değerleri birbiri ile örtüşmelidir. mx değerinin ikilik sistemdeki gösterimi kaç bit içeriyorsa her birey için X değeri o uzunlukta ve ky değerinin ikilik sistemdeki gösterimi kaç bit içeriyorsa her birey için Y değeri o uzunlukta olmalıdır. Bu uygulama için söz konusu uzunlukta olmayan ikilik sistemdeki gösterimlerin en başlarına (en sola) 0 değerleri eklenerek tüm bireylerin X ve Y değerleri için gösterimler ayrı ayrı kendi içlerinde eşit uzunlukta olurlar.

GA’nın uygulanması esnasında temel parametrelerinin aynı değerde olması gibi gösterimlerin de aynı uzunluklarda olması ile işlemler rahatça yapılır ve çözüm

kümesinde daha iyi bireyler elde etmeye çalışılarak görüntüler aranır. Böylelikle, ek koşullar kullanılarak görüntü arama süresini uzatacak işlemlere yer verilmeden arama süreci icra edilir.

Paralel arama yaklaşımli GA ile akıllı aramaların, bu çalışmanın bir parçası olan çok iş parçacıklı yapılabilmesi için ilk olarak nesil sayısı toplam iş parçacığı sayısına oranlanarak her bir iş parçacığı için yeni nesil sayısı olarak belirlenmiştir. Bu durum evrimleşme sürecini oldukça daraltmış ve daha iyi bireylerin elde edilme ihtimalini düşürmüştür. Daha sonra, nesil sayısının değiştirilmemesi gerektiğine karar verilerek nüfustaki toplam birey sayısı toplam iş parçacığı sayısına oranlanarak yeni birey sayısı olarak belirlenmiştir. Çünkü her iş parçacığının referans görüntüden elde edilen şablon görüntüyü kapsayan referans görüntü parçası için arama uzayı daralır ve bu sebeple nesil sayısının değil birey sayısının düşürülmesi gerekir. Arama uzayının daralması ile çözüm kümesindeki birey sayısının azalması birbiri ile örtüşen durumlardır. Nesil sayısı ise arama uzayını değil evrimleşme sürecini etkilemektedir.

6.9.4 GİGA kapsamında geliştirilen algoritmalar

Yazılım kapsamında kaba kuvvet ve akıllı aramalar için tek iş parçacıklı ve paralel olarak arama seçenekleri olması, görüntülerin yeniden boyutlandırılabilmesi ve farklı görüntü eşleme seçeneklerinin olması, yazılan kodların her durum için tekrarlanmasını gerektirebilir. Bu sebeple, ortak olan kısımlar için farklı yöntemler kullanılmalıdır. Ayrı kodlamaların yapılmaması için *delegate*¹ sınıfı kullanılmıştır. Böylelikle, kaba kuvvet ve akıllı aramalar, görüntü boyutlandırmaları ve uygunluk fonksiyonları için belirlenen yapılarla daha sade, okunabilir ve anlaşılır, başka eşleme yöntemleri ve çeşitli seçenekler için genelleştirilebilecek bir kodlama sistemi ortaya çıkmıştır.

Diğer durumda ayrı kodlamaların yapılması, program kodlarında ortak olan pek çok kısmın tekrar tekrar yazılmasını gerektirebilir. Bu uygulama hem kullanışlı değildir hem de hataların ortaya çıkmasına neden olabilir. Bu durumda tüm alternatifler için tekrar eden pek çok kod, hem okunabilirliği azaltır hem de uygulamaya gereksiz yere yük bindirir. Örneğin aynı görüntü arama seçeneği için farklı bir görüntü eşleme

¹ Delegate (temsilci), bir fonksiyona referans olan bir yapıdır. İlk olarak fonksiyon bu yapıya atanır ve delegate tam olarak o fonksiyon gibi davranır. Delegate yöntemi parametrelili ve değer döndüren fonksiyonlarla da kullanılabilir.

yönteminin seçilmesi, görüntü aramalarında bir değişiklik yaratmaz iken sadece görüntülerin karşılaştırılmasında farklı işlemlerin yapılmasını gerektirir. Aynı görüntü eşleme yöntemi kullanarak tek iş parçacıklı ya da çok iş parçacıklı aramalarda ise sadece ele alınan görüntü ve iş parçacığı sayısı değişir.

Aramaların GA ile yapılıp yapılmaması iki farklı süreçtir ve ayrı olarak ele alınmasını gerektirir. Öte yandan görüntüleri boyutlandırarak aramaların yapılması ise sadece görüntülerin ilgili boyut ile örneklenip süreçlerin tekrarlanması ile gerçekleştirilir.

Çalışma kapsamında geliştirilen algoritmalar, sonrasında temel olarak bir görüntüde bir başka görüntüyü problemsiz arayacak şekilde düzenlenmiştir. Tıpkı YSA'nın eğitimi ve kararlı bir yapıya gelmesi gibi algoritmalar da görüntü arama ve görüntü eşleme süreçleri ile ortaya çıkan sıkıntılara göre düzenlenerek ve güncellenerek en etkili hâllerine getirilmiştir. Bu noktada sözü geçen sıkıntılar şu şekilde ifade edilebilir:

- Farklı ve büyük boyutlardaki görüntülerde arama sürecinde doğabilecek problemler
- İş parçacığı sayısının artması ile programın yanıt vermemesi
- GİGA'da arama ve eşleme alternatifleri için yeterli esnekliğin olmaması
- Görüntü arama sürecinin verimli olmaması
- Parametreleri düzenleyebilme imkânı olmaması
- İlgili olunan görüntüyü seçebilme esnekliğinin olmaması
- GİGA öncesinde olduğu gibi arama işlemleri sonunda sonuçların iyi bir şekilde kullanıcıya yansıtılmaması

Bu durumların giderilebilmesi için tasarlanan algoritmaların en uygun hâllere getirilerek GİGA çalışmasını bütün olarak oluşturmalarını sağlamak gerekir.

6.9.5 Görüntü eşleme için YSA kullanılması

Görüntü tanımada oldukça başarılı bir yaklaşım olmasına rağmen bu çalışmada kullanılan YSA ile görüntü eşleme kaba kuvvet aramalarda oldukça zaman almıştır. Hazır olarak kullanılan sınıfların mimarileri bozulmadan yapılan çeşitli iyileştirmelerle YSA ile görüntü eşlemeleri oldukça hızlandırılmıştır. Bölüm 5.8'de

verilen tekniklerle YSA süreci iyileştirilmiştir. Böylelikle YSA ile görüntü eşleme süreci en iyi durumuna getirilerek YSA'nın görüntü aramadaki üstünlüklerinden bu çalışmada en iyi şekilde yararlanılmıştır.

Bu çalışmada, YSA ile görüntü eşleme gri tonda filtreli referans görüntüden elde edilen pencerelerin YSA'ya, diğer yöntemlerle görüntü eşleme ise renkli orijinal referans görüntüden elde edilen pencerelerin ilgili görüntü eşleme yöntemine giriş olarak verilmesi ile yapılır. Diğer yöntemleri kullanmadan önce referans ve şablon görüntü aynı filtre işlemlerine tabi tutulabilir. Tüm görüntü eşleme yöntemleri ile arama sürecinde, referans görüntüden elde edilen pencereler herhangi bir filtreleme işlemine uğratılmadan şablon görüntü ile eşleştirilir. Böylelikle filtreleme işlemleri göz ardı edilerek ilgili görüntüler eşlenir ve arama süresi oldukça kısalmıştır.

Görüntü aramak için öncesinde eğitilmesi gereken sonrasında gerekirse doğrulama işlemlerini gerektiren YSA, ilgili olunan görüntüyü öğrenme ve daha sonra arama uygulamasında etkili bir yöntemdir. Diğer görüntü eşleme yöntemleri görüntü işleme kütüphanesi kapsamındaki fonksiyonlara dayalıdır ve bir YZ yöntemi olan YSA ayrı bir görüntü eşleme yaklaşımı olarak ele alınır. Bu sebeple, görüntü tespitinde etkili ve elverişli taraflarından en iyi şekilde istifade edebilmek için dikkatli bir şekilde tasarlanmalı ve kararlı bir yapıya gelene kadar güncellenebilmelidir.

6.10 Tez Çalışması Süreci ve Sonuçların Genel Değerlendirilmesi

Görüntü aramalarının bu çalışmada YZ yöntemleri ile gerçekleştirilmesi ve tez kapsamında hazırlanan yazılımın kullanıcı ile etkileşim içinde olan bir arayüz içermesi, yazılımın işlevselliğini ön plana çıkararak programı daha kullanılabilir kılmıştır. Bu çalışma, herhangi bir görüntüde bir başka görüntüyü aramanın farklı görüntü eşleme yöntemleri kullanılarak kaba kuvvet, akıllı ve paralel yollarla ne şekilde gerçekleştirilebileceğini göstermiştir.

Görüntü aramalarının gerçekleştirildiği örnek referans görüntüler Şekil A.1'de, Şekil A.2'de ve Şekil A.3'te verilmiştir.

Görüntü aramalarının olabildiğince hızlı bir şekilde yapılabilmesi için GİGA sürecinin bütünüyle iyileştirilmesi gerekmektedir. Bu süreç, görüntü eşleme, görüntü arama, kodlama, bellek kullanımı, matematiksel işlemler, görüntüleme ve diğer programlama işlemlerinin tümünü içermektedir.

Sürecin iyileştirilmesi, bellek kullanımına, etkili ve verimli bir kodlama sisteminin tasarımına bağlıdır. Bu yüzden, arama sürecini olumsuz yönde etkileyecek hiçbir durum göz ardı edilmeden arama süresini uzatabilecek olası tüm durumlar ortadan kaldırılmalıdır. Arama süresini kısaltabilecek yöntemler aynı zamanda en doğru tasarımı gerektirirler. İşlemlerin verimli bir şekilde gerçekleştirilebilmesi için arama süresinin kısa olması ve aynı zamanda görüntü eşlemelerinin de etkili ve doğru bir şekilde yapılabilmesi gerekir.

Akıllı aramalar için GA'nın parametreleri en uygun sonuçları elde etmek için çeşitli şekillerde belirlenebilir. Probleme ve ilgili görüntülere göre değiştirilebilir.

Bu tezde GA kullanılarak akıllı aramaların yapılmasının sebebi, olası tüm durumlara bakılmaksızın kısa sürede yeterli sonuca ulaşılabilmesidir. Bu noktada söz konusu sonuç, şablon görüntü ile önceden belirlenen eşik değeri ya da bu değer üzerinde eşleşen görüntü ya da görüntülerdir. GA sürecinde elde edilen bireyler ve uygunluk değerleri her iş parçacığı için oluşturulan dosyalarda görülebilmektedir. GA'nın varsayılan (başlangıç (parametre içeren tüm tablolar)) parametre değerleri Tablo A.1'de verilmiştir.

Bu çalışmada, bellek kullanımını olabildiğince etkili ve verimli bir biçimde gerçekleştirerek işlemlerin hızlı bir şekilde yapılması sağlanmıştır. İşlemlerin uzun zaman almasına neden olacak ve yazılımı hantallaştıracak tüm olası durumlar en aza indirilmiştir.

Tez kapsamında hazırlanan çalışmada genel olarak, paralel aramaların kaba kuvvet aramalara göre ciddi iyileştirmeler sağlayabildiği görülmüştür. Akıllı aramalar ile daha kısa sürede şablon görüntü aranabilmektedir.

YSA ile görüntü eşlemelerinin doğruluğu, eğitim kümesindeki görüntülere ve eğitim sürecine büyük ölçüde bağlıdır. Diğer yöntemlerle eşlemeler ise belirlenen fonksiyona bağlıdır. Tasarlanan algoritmalar, her arama şeklinde başarılı olabilecek yeterlilikteki şekillere getirilmiştir. Çeşitli şekillerdeki görüntü aramaları EK G.1'de yer alan Şekil A.8'de ve Tablo C.1'de verilmiştir.

Tablo D.1'de, Tablo D.2'de, Tablo D.3'te, Tablo D.4'te ve Tablo D.5'te 20 (5x4) farklı durum için görüntü arama sonuçları ve ortalama süreler parametrik olarak verilmiştir.

Bu çalışma ile ulařılan sonuçlar genel olarak řu řekilde ifade edilebilir:

- YSA eđitiminde aynı grntnn birden fazla kullanılması yerine eřitlilik oluřturulmalıdır. Bu noktada eřitlilik kavramı ile ifade edilmek istenen giriřlerin bire bir aynı olmaması geređidir. YSA test srecinde referans grntden elde edilen grntler řablon grnt ile eřleřtirilir. Bu alt grntlerin her biri farklı grntlerdir ve bu iřlem grnt arama iin yapılır. Dolayısıyla, eđitim sreci de test sreci ile paralel olacak řekilde icra edilmelidir.
- YSA iin eđitim ve test sreleri, ilgili olan ve ilgili olmayan grntlere, seilen aktivasyon fonksiyonlarına bađlıdır.
- YSA dıřındaki grnt eřleme yntemleri olduka hızlı yaklařımlardır.
- YSA yntemi ile grnt eřlemelerde, arama sonularında YSA'nın hatalı sonuları iin ilgili grntler eřleřmeyen grntler olarak deđerlendirilmeli ve eđitim kmesine verilerek sinir ađı gncelleřtirilmelidir.
- Yeni oluřturulan sinir ađı yapısı iin az giriř ile ilk đrenme sreci olduka hızlı gerekleřtirilir, ama sre tekrarları ile sinir ađı kararlı ve grnt eřlemeleri iin en uygun hle getirilmelidir. Bu noktada YSA'nın daha kararlı bir yapıya gelmesi iin gerekleřtirilecek eđitimler, giriř sayısına gre gittike daha zaman alıcı olabilir. Fakat test srecinde YSA ile verimli sonular elde edebilmek iin gerekli olan bu iřlemler, YSA'nın bařarımını olumlu ynde etkileyen uygulamalardır.
- YSA ile grnt eřlemede giriř olarak verilecek grntlerin seimi nemlidir. Bu grntler referans grntden kırpılan (kesilen) grntler olabilirler, fakat sinir ađı giriřlerinin deđiřken olmaması ele alınan giriř grntlerin yatay ve dřey boyutlarının standart olmasını gerektirir.
- Sinir ađı iin en uygun yapı eđitim ve test sreleri ile belirlenebilir. Bu noktada en uygun yapı YSA'ya dhil edilecek katmanlardır. Giriř grntlerin đrenilerek eđitimin bařarılı bir řekilde gerekleřtirilmesi mimarinin iyi bir řekilde oluřturulmasına bađlıdır. Test sreci iin ise grnt aramaları ve dođrulama iřlemleri gerekir.

- GY öğrenme ya da GA ile YSA'nın eğitilmesi farklılıklar göstermektedir. Örneğin GA'nın doğası gereği süreç boyunca elde edilen en iyi çözüm bir önceki elde edilen en iyi çözümden daha kötü olamaz. Fakat, GY ile öğrenmede süreç boyunca her zaman sürekli bir iyiye gidiş daha doğrusu kötüye gitmeyişi gözlemlenemeyebilir.
- YSA'nın çıkış değerinin göz önüne alınan tüm test girişi için eşik değerinden daha büyük olması, arama sonuçlarında hiç görüntünün olmaması anlamına gelmektedir. Bu sebeple bu süreç eşik değerinin azaltılması ile sürdürülebilir. İşlemlerin devam ettirilmesi ile elde yanlış sonuçlar ise eşleşmeyen görüntüler olarak YSA'ya verilerek YSA eğitilebilir. YSA'ya yeni giriş görüntüler dâhil edilerek eğitim öncesinde ağırlık değerlerinin rasgele belirlenmesi eğitim süresini daha da uzun tutar. Diğer durumda ise YSA durumunu koruyarak girdi görüntüleri öğrenir.
- Kaba kuvvet aramalarda sonuçların kaliteliliği için geçiş değeri artırılabilir, ama bu durum aynı zamanda arama süresinin artmasına neden olur.
- YSA'nın hangi görüntüyü öğrenmek için eğitildiği, referans görüntülerden elde edilen şablon görüntü boyutlarında alt görüntüleri giriş olarak değerlendirirken arzu edilen sonuçları ne olması istendiği bu çalışma için önemlidir.
- YSA ile görüntü arama uygulamaları, kaliteli test sürecine ve doğal olarak başarılı bir eğitim sürecine gerek duyar. Bu sebeple YSA'nın tasarımı önemlidir.
- YSA ile yüz tanıma, parmak izi tanıma ya da benzeri tanıma ve tespit uygulamalarında YSA'nın başarılı olabilmesi için söz konusu alanlar için en etkili ve çeşitli görüntüler ile eğitilebilmesi gerekir. Bu süreç sonunda bu çalışmada ise arzu edilen şablon görüntünün öğrenilerek referans görüntüde alt görüntüler ile eşlenerek tespit edilmesidir.
- Görüntüleri yeniden boyutlandırmada görüntülerde gerekli bilginin kaybedilmemesi gerekir.
- YSA eğitimi için girdi görüntülerin filtreli gri görüntüler olması eğitim süreci için ve test sürecinde başarılı olunması için gereklidir.

GİGA kapsamında gerçekleştirilen arama algoritmalarının zaman karmaşıklığı (time complexity) açısından değerlendirmeleri önem taşımaktadır. Paralel olmayan kaba kuvvet bir arama için tt şablon görüntü ile alt görüntüyü eşleme süresi olmak üzere toplam arama süresini içeren zaman karmaşıklığı şu şekilde ifade edilebilir: $O(mx \times ky \times tt)$. Paralel aramalarda iş parçacığı sayısının artırılması ile ya da GA ile arama uzayının daraltılması şeklinde uygulama ile zaman karmaşıklığı düşürülebilir. Fakat bu durum özellikle GA ile aramalar için her zaman en iyi sonuçların elde edilebileceği garantisini vermemektedir, ama GA ile GİGA uygulamaları sayesinde başarılı sonuçlar elde edilebilir. Bu yaklaşım için de paralel arama seçeneği sayesinde sonuçlar değerlendirilebilir ve karşılaştırılabilir.

Sürecin iyileştirilmesi paralellik, optimizasyon, görüntü eşleme ve akıllı arama gibi uygulamalara bağlıdır. Bu çalışma kapsamındaki alternatifler sayesinde sözü geçen bu çeşitli uygulamalar icra edilebilir ve süreçler gözlemlenebilir.

GİGA çalışması, arama işlemlerinin görüntü veri tabanı üzerinde çok sayıda görüntü ile gerçekleştirilebileceği bir şekilde getirilebilir. Daha farklı görüntü eşleme yöntemleri kullanımı, bu yöntemlerin birleşimleri, ağırlıklandırılması, kullanıcı geri bildirimleri, arama algoritmalarında daha farklı yaklaşımların izlenmesi gibi uygulamalarla tez kapsamında hazırlanan bu çalışma geliştirilebilir.

KAYNAKLAR

- Bunte, K., Biehl, M., Jonkman, M. F., and Petkov, N.,** 2010: Learning effective color features for content based image retrieval in dermatology. *Pattern Recognition*. Vol. **44**, pp. 1892-1902.
- Chang, S. H., Cheng, F. H., Hsu, W. H., and Wu, G. Z.,** 1997: Fast algorithm for point pattern matching: Invariant to translations, rotations and scale changes. *Pattern Recognition*. Vol. **30**, no. 2, pp. 311-320.
- Choi, M. S., and Kim, W. Y.,** 2002: A novel two stage template matching method for rotation and illumination invariance. *Pattern Recognition* 35. pp. 119-129.
- Debella-Gilo, M., and Kaab, A.,** 2011: Sub-pixel precision image matching for measuring surface displacements on mass movements using normalized cross-correlation. *Remote Sensing of Environment* 115. pp. 130-142.
- Ding, L., Goshtasby, A., and Satter, M.,** 2001: Volume image registration by template matching. *Image and Vision Computing* 19. pp. 821-832.
- Dubey, R. S., Choubey, R., and Bhattacharjee, J.,** 2010: Multi Feature Content Based Image Retrieval. (*IJCSE*) *International Journal on Computer Science and Engineering*. Vol. **02**, no. 06, pp. 2145-2149.
- ElAlami, M. E.,** 2011: A novel image retrieval model based on the most relevant features. *Knowledge-Based Systems* 24. pp. 23-32.
- Fernandez, X.,** 1997: Template Matching of Binary Targets in Grey-Scale Images: A Nonparametric Approach. *Pattern Recognition*. Vol. **30**, no. 7, pp. 1175-1182.
- Ferreira, C. D., Santos, J. A., Torres, R. S., Gonçalves, M. A., Rezende, R. C., and Fan, W.,** 2011: Relevance feedback based on genetic programming for image retrieval. *Pattern Recognition Letters* 32. pp. 27-37.
- Fredriksson, K., and Ukkonen, E.,** 1999: Combinatorial methods for approximate image matching under translations and rotations. *Pattern Recognition Letters* 20. pp. 1249-1258.
- Kekre, H. B., Thepade, S. D., and Maloo, A.,** 2010: Query by Image Content Using Colour Averaging Techniques. *International Journal of Engineering Science and Technology*. Vol. **2(6)**, pp. 1612-1622.
- Keysers, D., and Unger, W.,** 2003: Elastic image matching is NP-complete. *Pattern Recognition Letters* 24. pp. 445-453.
- Kumar, R., and Vikram, B. R. D.,** 2010: Fingerprint matching using multi-dimensional ANN. *Engineering Applications of Artificial Intelligence* 23. pp. 222-228.

- Kwok, S. H., and Zhao, J. L.,** 2006: Content-based object organization for efficient image retrieval in image databases. *Decision Support Systems* 42. pp. 1901-1916.
- Lin, C., and Chang, H. J.,** 2011: Identification of pressurized water reactor transient using template matching. *Annals of Nuclear Energy*.
- Liu, Y., Zhang, D., Lu, G., and Ma, W. Y.,** 2007: A survey of content-based image retrieval with high-level semantics. *Pattern Recognition* 40. pp. 262-282.
- Mattocchia, S., Tombari, F., and Stefano, L. D.,** 2011: Efficient template matching for multi-channel images. *Pattern Recognition Letters* 32. pp. 694-700.
- Moon, Y. S., Kim, B. M., Kim, M. S., and Whang, K. Y.,** 2010: Scaling-invariant boundary image matching using time-series matching techniques. *Data & Knowledge Engineering* 69. pp. 1022-1042.
- Müller, H., Michoux, N., Bandon, D., and Geissbuhler, A.,** 2004: A review of content-based image retrieval systems in medical applications—clinical benefits and future directions. *International Journal of Medical Informatics* 73. pp. 1-23.
- Shu, X., and Wu, X. J.,** 2011: A novel contour descriptor for 2D shape matching and its application to image retrieval. *Image and Vision Computing* 29. pp. 286-294.
- Singh, U. P., Jain, S., and Ahmed, G. F.,** 2010: Content Base Image Retrieval Using Phong Shading. (*IJCSIS*) *International Journal of Computer Science and Information Security*. Vol. 8, no. 1, pp. 301-306.
- Torres, R. S., Falcao, A. X., Gonçalves, M. A., Papa, J. P., Zhang, B., Fan, W., and Fox, E. A.,** 2009: A genetic programming framework for content-based image retrieval. *Pattern Recognition* 42. pp. 283-292.
- Url-1** <<http://knol.google.com/k/abdullah/content-based-image-retrieval-içerik/>>, alındığı tarih 25.03.2011.
- Url-2** <http://en.wikipedia.org/wiki/Digital_image_processing>, alındığı tarih 26.03.2011.
- Url-3** <http://en.wikipedia.org/wiki/Template_matching>, alındığı tarih 26.03.2011.
- Url-4** <http://en.wikipedia.org/wiki/Stochastic_optimization>, alındığı tarih 27.03.2011.
- Url-5** <<http://www.goruntuisleme.org/2011/>>, alındığı tarih 29.03.2011.
- Url-6** <http://en.wikipedia.org/wiki/Artificial_neural_network>, alındığı tarih 22.05.2011.
- Url-7** <<http://en.wikipedia.org/wiki/Backpropagation>>, alındığı tarih 24.05.2011.
- Url-8** <http://tr.wikipedia.org/wiki/Genetik_algoritma>, alındığı tarih 24.05.2011.
- Url-9** <http://tr.wikipedia.org/wiki/Yapay_sinir_ağları>, alındığı tarih 24.05.2011.
- Url-10** <http://tr.wikipedia.org/wiki/Yapay_Zeka>, alındığı tarih 24.05.2011.

- Url-11** <<http://www.sayisalyontemler.com/?q=node/78>>, alındığı tarih 30.05.2011.
- Url-12** <http://www.vcskicks.com/optimize_csharp_code.php>, alındığı tarih 06.06.2011.
- Url-13** <<http://www.dotnetperls.com/optimization>>, alındığı tarih 07.06.2011.
- Url-14** <<http://www.techgalaxy.net/Docs/Dev/5ways.htm>>, alındığı tarih 08.06.2011.
- Url-15** <<http://en.wikipedia.org/wiki/MATLAB>>, alındığı tarih 12.06.2011.
- Url-16** <http://en.wikipedia.org/wiki/Canny_edge_detector>, alındığı tarih 13.06.2011.
- Url-17** <<http://en.wikipedia.org/wiki/Perceptron>>, alındığı tarih 14.06.2011.
- Url-18** <http://tr.wikipedia.org/wiki/Öklid_uzaklığı>, alındığı tarih 16.06.2011.
- Url-19** <http://tr.wikipedia.org/wiki/RGB_renk_uzayı>, alındığı tarih 19.06.2011.
- Url-20** <<http://en.wikipedia.org/wiki/MRI>>, alındığı tarih 26.06.2011.
- Url-21** <<http://en.wikipedia.org/wiki/NASA>>, alındığı tarih 26.06.2011.
- Url-22** <http://en.wikipedia.org/wiki/X-ray_computed_tomography>, alındığı tarih 26.06.2011.
- Url-23** <http://en.wikipedia.org/wiki/National_Library_of_Medicine>, alındığı tarih 27.06.2011.
- Url-24** <http://en.wikipedia.org/wiki/Relational_database>, alındığı tarih 28.06.2011.
- Url-25** <<http://msdn.microsoft.com/en-us/library/ms973852.aspx>>, alındığı tarih 28.06.2011.
- Url-26** <<http://tr.wikipedia.org/wiki/USGS>>, alındığı tarih 28.06.2011.
- Url-27** <<http://tr.wikipedia.org/wiki/X-ray>>, alındığı tarih 28.06.2011.
- Url-28** <http://en.wikipedia.org/wiki/.NET_Framework>, alındığı tarih 29.06.2011.
- Url-29** <http://en.wikipedia.org/wiki/Gaussian_function>, alındığı tarih 29.06.2011.
- Url-30** <http://en.wikipedia.org/wiki/Heaviside_function>, alındığı tarih 29.06.2011.
- Url-31** <http://en.wikipedia.org/wiki/Linear_function>, alındığı tarih 29.06.2011.
- Url-32** <http://en.wikipedia.org/wiki/Roulette-wheel_selection>, alındığı tarih 29.06.2011.
- Url-33** <http://en.wikipedia.org/wiki/Sigmoid_function>, alındığı tarih 29.06.2011.
- Url-34** <http://en.wikipedia.org/wiki/Visual_Studio>, alındığı tarih 29.06.2011.
- Url-35** <<http://msdn.microsoft.com/en-us/library/ff647790.aspx>>, alındığı tarih 29.06.2011.
- Url-36** <[http://msdn.microsoft.com/en-us/library/ms173171\(v=vs.80\).aspx](http://msdn.microsoft.com/en-us/library/ms173171(v=vs.80).aspx)>, alındığı tarih 29.06.2011.

- Url-37** <<http://en.wikipedia.org/wiki/NP-complete>>, alındığı tarih 06.07.2011.
- Url-38** <<http://msdn.microsoft.com/en-us/library/cscsdfbt.aspx>>, alındığı tarih 12.07.2011.
- Url-39** <http://tr.wikipedia.org/wiki/Markov_zinciri>, alındığı tarih 12.07.2011.
- Url-40** <<http://franck.fleurey.free.fr/NeuralNetwork/javadoc/NeuralNetwork/>>, alındığı tarih 13.07.2011.
- Url-41** <<http://tr.wikipedia.org/wiki/Stokastik>>, alındığı tarih 13.07.2011.
- Url-42** <http://en.wikipedia.org/wiki/Correlation_and_dependence>, alındığı tarih 14.07.2011.
- Url-43** <<http://tr.wikipedia.org/wiki/UML>>, alındığı tarih 14.07.2011.
- Url-44** <http://tr.wikipedia.org/wiki/Kolmogorov-Smirnov_sunamasi>, alındığı tarih 17.07.2011.
- Url-45** <http://tr.wikipedia.org/wiki/Zaman_serisi>, alındığı tarih 17.07.2011.
- Wang, X. Y., Yu, Y. J., and Yang, H. Y.,** 2011: An effective image retrieval scheme using color, texture and shape features. *Computer Standards & Interfaces* 33. pp. 59-68.
- Yan, H., Yang, J., and Yang, J.,** 2011: Bimode model for face recognition and face representation. *Neurocomputing* 74. pp. 741-748.
- Yuan, Y., Pang, Y., Wang, K., and Shang, M.,** 2011: Efficient image matching using weighted voting. *Pattern Recognition Letters*.

EKLER

EK A.1 : Referans görüntüler.

EK A.2 : GİGA süreçlerini doğrulamak için kullanılan referans görüntüler.

EK A.3 : GİGA süreçlerini doğrulamak için kullanılan diğer görüntüler.

EK B.1 : Yeni bireyler yaratmada eldeki bireylerin uygunluk değerlerini kullanarak uygunluk değeri orantılı bireyler elde etmek için yazılan C# kodu.

EK B.2 : Paralel aramalarda iş parçacıklarının bitme durumunu kontrol için yazılan iki C# kodu.

EK B.3 : GA için eldeki bireylerden farklı bireyler elde etme kontrolü amaçlı yazılan C# kodu.

EK C.1 : Görüntü arama öncesinde uygulanan işlemleri içeren C# fonksiyonu (metot, yöntem).

EK C.2 : Görüntü arama sonrasında uygulanan işlemleri içeren C# fonksiyonu.

EK C.3 : Referans görüntüyü alt görüntülere ayırma C# fonksiyonu.

EK C.4 : Görüntü arama sonuçlarını görüntüleme C# fonksiyonu.

EK C.5 : YSA ile görüntü arama süreci C# fonksiyonu.

EK D.1 : Eşleşen görüntüleri ve eşleşme bilgilerini görüntülemek için örneği oluşturulan C# sınıfı.

EK D.2 : Alt görüntülerin referans görüntüdeki sol üst köşelerinin X ve Y konumlarını ve bu değerlerin ikili sistemdeki karşılıklarını içeren C# sınıfı.

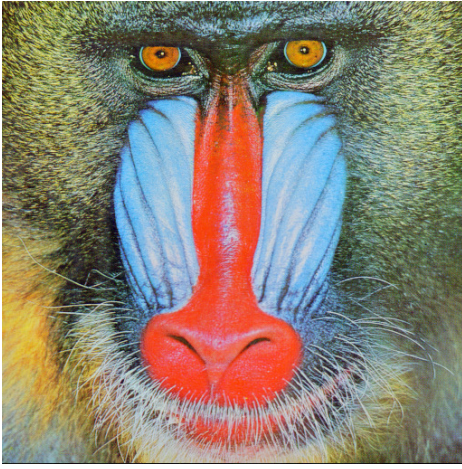
EK E.1 : YSA için kullanılan aktivasyon fonksiyonları.

EK E.2 : YSA için kullanılan aktivasyon fonksiyonlarının tanımlamaları.

EK F.1 : Referans görüntülerden paralel aramalar için elde edilen görüntüler.

EK G.1 : Örnek uygulamalar.

EK A.1



(a)



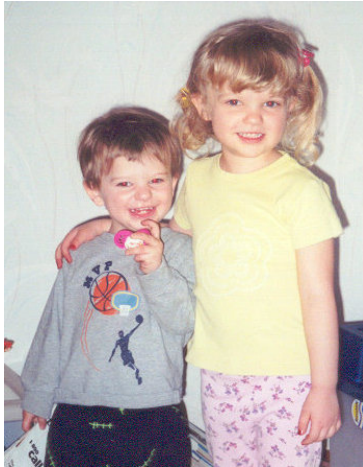
(b)



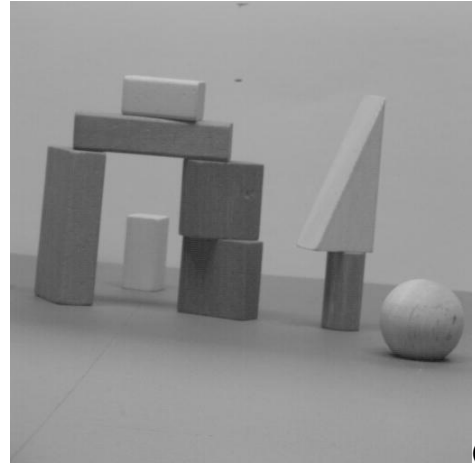
(c)



(d)



(e)



(f)

Şekil A.1 : Örnek görüntüler: (a)512x512. (b)400x400. (c)200x219. (d)350x256. (e)349x446. (f)512x512.

EK A.2



(a)



(b)



(c)



(d)



(e)



(f)

Şekil A.2 : Örnek görüntüler: (a)185x209. (b)412x270. (c)500x333. (d)179x187. (e)171x180. (f)163x180.

EK A.3



(a)



(b)



(c)



(d)



(e)



(f)

Şekil A.3 : Örnek görüntüler: (a)324x223. (b)512x384. (c)512x512. (d)180x180. (e)225x225. (f)640x480.

Tablo A.1: Görüntü arama için GA'nın parametre değerleri.

Parametre	Değer
Nesil sayısı	100
Birey sayısı	100
Birey seçim oranı (%)	10
Çaprazlama olasılığı (%)	80
Çaprazlama işlemcisi	Normal çaprazlama
Mutasyon olasılığı (%)	5
Mutasyon işlemcisi	Gen değiştirme
Aynı birey üzerinde çaprazlama ve mutasyon	Yok
Birey seçim yöntemi	Rasgele seçim
Yeni neslin oluşturulması	Elitizm
Yeni nesil için ebeveyn bireylerden en iyiler (%)	20
Kalan bireylerden rasgele (%)	5
Kalan bireylerin çaprazlama kısmı (%)	50
Kalan bireylerin mutasyon kısmı (%)	50
En yakın komşuluk değeri	2

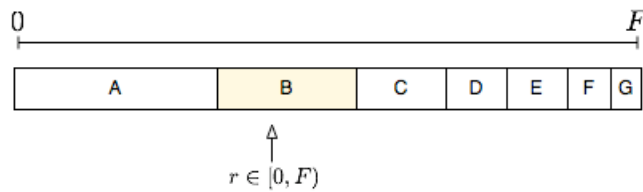
Tablo A.2: Görüntü arama için GA'nın birey seçim yöntemleri.

Birey Seçim Yöntemi	Yaklaşım
1	Rasgele seçim
2	Belli bir oranda rasgele seçim
3	Rulet tekerleği seçimi

Tablo A.1'in en alt satırındaki en yakın komşuluk değeri, elde edilen en iyi sonuçların belli miktara kadar piksel komşuluklarındaki eşleşme sonuçlarını da elde etmek için kullanılır. Bu konumlar bu değerlerin bir olması durumunda (X, Y) için şu şekilde ifade edilirler: $(X-1, Y-1)$, $(X, Y-1)$, $(X+1, Y-1)$, $(X-1, Y)$, $(X+1, Y)$, $(X-1, Y+1)$, $(X, Y+1)$, $(X+1, Y+1)$. Bu konumların referans görüntüde sırayla yatay ve düşey düzlemde karşılık geldiği noktaları olan X ve Y için A.1'deki grupta verilen şu iki koşul sağlanmalıdır.

$$\begin{aligned}
 1. & (0 \leq X \leq mx) \\
 2. & (0 \leq Y \leq ky)
 \end{aligned}
 \tag{A.1}$$

Tablo A.2'de verilen rasgele birey seçimi, nüfustaki bireylerden herhangi birini rasgele seçme; belli bir oranda rasgele birey seçimi, nüfustaki en iyi bireylerin belli bir kısmından rasgele seçme; rulet tekerleği seçimi ise uygunluk tabanlı seçimdir. Tablo A.2'deki 2 numaralı yöntemin seçimi durumunda, Tablo A.1'deki birey seçim oranı (%) geçerli olur. Rulet tekerleği seçimi olarak da bilinen uygunluk orantılı seçim Şekil A.4'te verilmiştir.



Şekil A.4 : Tek bir bireyin uygunluk orantılı seçim örneği.

Şekil A.4'te F değeri tüm bireylerin uygunluk değerlerinin toplamıdır. Uygunluk orantılı seçimle yeni bireyler elde etmek için uygunluk değeri daha iyi olan bireyler kullanılır. Bir bireyin uygunluk değerinin tüm bireylerin uygunluk değerlerinin toplamı içindeki payı A.2'deki denklemde verilmiştir. Bu pay tüm bireyler arasından bu bireyin seçilme olasılığıdır.

$$p_i = \frac{F_i}{\sum_{j=1}^I F_j} \quad (\text{A.2})$$

Tablo A.3: GİGA için parametre değerleri.

Parametre	Değer
Uygunluk fonksiyonu	ORD
Kaba kuvvet aramalar için pencereler arası geçiş	1
Eşik değeri	0,92
Çoklu çözünürlük	Seçili değil
En küçük ölçek	0,25
En büyük ölçek	1,00
Ölçeklendirme sayısı	4
Yatay blok sayısı	4
Dikey blok sayısı	1

Tablo A.3'teki çoklu çözünürlük seçili ise bu tablodaki en küçük ölçek, en büyük ölçek ve ölçeklendirme sayısı geçerli olur.

Tablo B.1: YSA'nın eğitimi için GY'nin parametre değerleri.

Parametre	Değer
Alfa (Alpha)	0,5
Gama (Gamma)	0,2
En fazla tekrarlama sayısı	1000
En büyük karesel hata	0,001

Tablo B.2: YSA'nın eğitimi için GA'nın parametre değerleri.

Parametre	Değer
Birey sayısı	50
Birey seçim oranı (%)	10
Çaprazlama işlemcisi	Normal çaprazlama
Mutasyon olasılığı (%)	4
Mutasyon işlemcisi	Gen değiştirme
Mutasyon genişliği (genlik)	1
Aynı birey üzerinde çaprazlama ve mutasyon	Var
En fazla tekrarlama sayısı	1000
En büyük karesel hata	0,001
Yeni nesle aktarılacak çocuk birey sayısı	1

Tablo B.2'deki yeni nesle aktarılacak çocuk birey sayısı, yeni nesle önceki nesildeki uygunluk değeri en iyi kaç bireyin aktarılacağını ifade eder. Ayrıca, YSA'nın GA ile eğitilmesi sürecinde çaprazlama olasılığı söz konusu değildir.

EK B.1

thisFit[r1] : (x1, y1) için uygunluk değeri
thisFit[r2] : (x2, y2) için uygunluk değeri

```
tot = thisFit[r1] + thisFit[r2];
```

```
tx = Math.Min(x1, x2);  
ty = Math.Min(y1, y2);
```

```
fx = (tx == x1) ? thisFit[r1] : thisFit[r2];  
fy = (ty == y1) ? thisFit[r1] : thisFit[r2];
```

```
ax = (int)(tx + (Math.Abs(x2 - x1) / tot * fx));  
ay = (int)(ty + (Math.Abs(y2 - y1) / tot * fy));
```

Bu kod ile eldeki iki bireyin uygunluklarına göre bu iki bireyin X ve Y koordinatlarına orantılı uzaklıktaki konumları içeren yeni birey (ax, ay) elde edilerek yeni bireyin uygunluk değeri hesaplanmıştır. Bu uygulama, uygunluk değeri daha iyi olan bireye daha yakın alanlarda aramaların yapılması için denemiştir, fakat işlemler oldukça zaman almıştır. Öte yandan, en iyi ve en iyiye yakın çözümler bu problemde arama uzayında belli bir alanda yoğunlaşmış olmayabilirler. Referans görüntüye göre arama uzayında her yerde şablon görüntü yer alabilir.

Sözü geçen işlemlerin arama süresini uzatması sebebiyle, görüntü aramalarının GA ile daha verimli bir şekilde yapılabilmesi amacıyla bu problem için en uygun genetik algoritma parametrelerine yoğunlaşmıştır.

EK B.2

Birinci kod parçası:

```
stt = true;
while (stt)
{
    stt = false;
    for (int i = 0; i < l; i++)
    {
        if (multis[i].t.ThreadState == ThreadState.Running)
        {
            stt = true;
            break;
        }
    }
}
```

Bu kod ile tüm iş parçacıklarının bitip bitmediği kontrol edilmektedir.

İkinci kod parçası:

```
while (true)
{
    if (MultiSearch.done == tot)
        break;
}
```

İkinci kod ile ilerleme çubuğunun en büyük değerine ulaşarak tüm iş parçacıklarının sonlanması durumu kontrol edilmektedir.

Fakat bu iki kod, paralel aramalarda çeşitli problemler yaratmıştır ve bu sebeple kaldırılmıştır. Sözü geçen problemler, arama sürecinde paralel arama işlemlerinin tamamen bitip bitmeme kontrolünün sistemi çok yorması ve işlemleri zorlaştırmasıdır. Paralel programlama ile işlemlerin daha hızlı icra edilebilmesi, algoritma tasarımının en uygun şekilde yapılması ile birlikte iş parçacıklarının çalışma ve bitiş kontrollerini de gerektirir.

EK B.3

Başlangıç nesildeki tüm bireylerin birbirinden farklı olması için iki boyutlu dizinin elemanlarına ilk olarak sıfır değeri atanması:

```
protected int[,] control;  
control = new int[mx, ky];  
for (i = 0; i < mx; i++)  
    for (j = 0; j < ky; j++)  
        control[i, j] = 0;
```

Yeni bir birey yaratmadan önce nesilde olup olmadığının kontrolü:

```
if (control[x, y] == 0)  
{  
    control[x, y] = 1;  
    child1 = new KeyPoint(genX, genY);  
}
```

GA için ilk olarak X ve Y değerleri onluk sistemde kodlanmıştır. Bu yüzden ilk olarak başlangıç nesil yaratılırken her bireyin birbirinden farklı olmasına dikkat edilmiştir. Ayrıca yeni bireylerin elde edilmesi sürecinde, eldeki bireylerden daha farklı bireylerin elde edilmesi için çeşitli koşullar getirilmiştir.

Daha sonra kodlamanın ikilik sistemdeki değerlerle yapılması ve bireylerin genlerinin birer birer belirlenmesi, bu durumların kaldırılmasını gerektirmiştir. Böylelikle başlangıç nesil tamamen rasgele elde edilmiş ve arama süreleri oldukça iyileştirilmiştir.

EK C.1

```
private void BeforeSearching()
{
    if (wholeImg == null || subImg == null)
        return;
    else if (SIW > WIW || SIH > WIH)
        throw new Exception("Sub-Image dimensionals are over Whole Image
dimensionals!");

    subImgE.Clear();
    wholeImgE.Clear();
    wholeImgEG.Clear();
    scales.Clear();
    remainings.Clear();

    inc = (MaxScale == MinScale) ? 1 : ((MaxScale - MinScale) / (NbScale - 1));
    i = 0;

    for (scale = MinScale; scale <= MaxScale; scale += inc)
    {
        Wi = (int)(WIW * scale);
        He = (int)(WIH * scale);

        wi = (int)(SIW * scale);
        he = (int)(SIH * scale);

        x = Wi - wi;
        x -= (x % WinGap);
        y = He - he;
        y -= (y % WinGap);

        subImgE.Add(subImg.Resize(wi, he, lin, true));
        wholeImgE.Add(wholeImg.Resize(Wi, He, lin, true));
        wholeImgEG.Add(wholeImgE[i].Clone().Convert<Gray, byte>());
        wholeImgEG[i++]._EqualizeHist();
        scales.Add(scale);
        remainings.Add((x / WinGap + 1) * (y / WinGap + 1));
    }

    r = remainings.Count;
}
```

EK C.2

```
public void AfterSearching()
{
    sw.Stop();
    MessageBox.Show(sw.ElapsedMilliseconds.ToString() + " Milliseconds",
"Elapsed Time");

    if (multis != null)
        for (i = 0; i < l; i++)
            matchs.AddRange(multis[i].result);
    else
    {
        for (i = 0; i < l; i++)
            matchs.AddRange(genetics[i].result);

        pop = GeneticSearch.population = (pop * l);
    }

    if (++c < r)
        GoOn();
    else
    {
        GetImageMatches();
        WH();
    }
}
```

EK C.3

```
private void SubImages()
{
    wi = WIW / partX;
    he = WIH;

    if (wi < SIW || (he / partY) < SIH)
    {
        l = -1;
        return;
    }

    y = wholeImgE[c].Width;
    x = y % wi;
    y -= x;

    recL.Clear();
    recL.Add(new Rectangle(0, 0, wi, he));
    for (i = wi; i < y; i += wi)
        recL.Add(new Rectangle(i - SIW + 1, 0, wi + SIW - 1, he));
    if (x != 0)
        recL.Add(new Rectangle(i - SIW + 1, 0, x + SIW - 1, he));

    L = recL.Count;
    liS.Clear();
    liSG.Clear();
    for (i = 0; i < L; i++)
    {
        liS.Add(wholeImgE[c].GetSubRect(recL[i]));
        liSG.Add(wholeImgEG[c].GetSubRect(recL[i]));
    }

    he /= partY;
    kp.Clear();
    for (j = 0; j < L; j++)
    {
        bmp = liS[j];
        bmpG = liSG[j];
        wi = bmp.Width;

        y = bmp.Height;
        x = y % he;
        y -= x;

        recSL.Clear();
        recSL.Add(new Rectangle(0, 0, wi, he));
        for (i = he; i < y; i += he)
            recSL.Add(new Rectangle(0, i - SIH + 1, wi, he + SIH - 1));
        if (x != 0)
            recSL.Add(new Rectangle(0, i - SIH + 1, wi, x + SIH - 1));
    }
}
```

```

l = recSL.Count;
x = liS.Count;
for (i = 0; i < l; i++)
{
    liS.Add bmp.GetSubRect(recSL[i]);
    liSG.Add bmpG.GetSubRect(recSL[i]);
    kp.Add(new KeyPoint(recL[j].X, recSL[i].Y));

    liS[x++].Save("ImageParts\\" + (i + "_" + j) + " - " + recL[j].X + "_" +
recSL[i].Y + ".jpg");
}
}

liS.RemoveRange(0, L);
liSG.RemoveRange(0, L);

l = liS.Count;
}

```

Bu fonksiyon ile görüntüler ilk olarak dikey olarak ardından yatay olarak YSA için yeni bir iş parçacığı başlatılır:

EK C.4

```
public void GetImageMatches()
{
    list_res.Items.Clear();
    i = 0;

    result = new Image<Bgr, byte>(openImage.FileName);

    /*fitness -> + 0.005f*/
    foreach (SubImageMatch sim in matches)
    {
        items[0] = "M " + (i++);
        items[1] = string.Empty + (int)(sim.Fitness * 100);
        items[2] = string.Empty + sim.X;
        items[3] = string.Empty + sim.Y;
        items[4] = string.Empty + sim.Scale;

        list_res.Items.Add(new ListViewItem(items));

        x = (int)(sim.X / sim.Scale);
        y = (int)(sim.Y / sim.Scale);

        lx = (int)(SIW / sim.Scale);
        ly = (int)(SIH / sim.Scale);

        result.Draw(new Rectangle(x, y, lx, ly), bgr, 2);
    }
    orgIB.Image = result;

    //img.Save(@"my sample sub-images\Founds\found" + found.ToString() +
    ".bmp");

    multis = null;
    genetics = null;
}
```

EK C.5

Şu fonksiyon ile YSA için yeni bir iş parçacığı başlatılır:

```
public void Start_NeuralNetworkOutput()
{
    t = new Thread(new ThreadStart(Apply_NeuralNetworkOutput));
    t.Start();
}
```

Şu fonksiyon ise YSA ile görüntü arama sürecini içermektedir:

```
public void Apply_NeuralNetworkOutput()
{
    for (i = 0; i < mx; i += WinGap)
    {
        for (j = 0; j < ky; j += WinGap)
        {
            done++;

            bmpG = imgG.GetSubRect(new Rectangle(i, j, SIW, SIH));
            s = nn.Output(nnpc.checkOutput(bmpG.Clone().Data))[0];

            if (s > threshold)
            {
                result.Add(new SubImageMatch(bmpG.Clone(), s, (i + kp.X), (j + kp.Y),
scale));
                found++;
            }
        }
        Progress(done, found);
        if (!Continue()) break;
    }
    listener.Texts(--state);
}
```

EK D.1

```
public class SubImageMatch
{
    protected Image<Bgra, Byte> img;
    protected Image<Gray, Byte> imgG;
    protected double f;
    protected int x, y;
    protected float s;

    public Image<Bgra, Byte> Window { get { return img; } }
    public Image<Gray, Byte> WindowG { get { return imgG; } }
    public double Fitness { get { return f; } }
    public int X { get { return x; } }
    public int Y { get { return y; } }
    public float Scale { get { return s; } }

    public SubImageMatch(Image<Bgra, Byte> image, double fit, int dx, int dy, float
scale)
    {
        img = image;
        f = fit;
        x = dx;
        y = dy;
        s = scale;
    }

    public SubImageMatch(Image<Gray, Byte> imageG, double fit, int dx, int dy,
float scale)
    {
        imgG = imageG;
        f = fit;
        x = dx;
        y = dy;
        s = scale;
    }
}
```

Bu sınıfa iki isim uzayı dâhil edilmiştir:

- using Emgu.CV;
- using Emgu.CV.Structure;

Bu sınıf, eşleşen görüntü, bu görüntünün uygunluk değeri, sol üst köşesinin referans görüntüdeki *X* ve *Y* konumları ve referans görüntü için ölçeklendirme değeri bilgilerini içermektedir.

Bu sınıf için iki kurucu (constructor) mevcuttur. YSA ile görüntü eşlemede görüntü arama sonuçlarında gri tonlu görüntülerin ve diğer bilgilerin, diğer görüntü eşleme yöntemleri ile görüntü arama sonuçlarında ise renkli görüntülerin ve diğer bilgilerin görüntülenebilmesi (verilere başka sınıflardan erişebilmek) için bu kuruculara gerek duyulur.

EK D.2

```
public class KeyPoint
{
    protected int x, y;
    protected List<int> xBits, yBits;

    public int X
    {
        get { return x; }
    }

    public int Y
    {
        get { return y; }
    }

    public KeyPoint(int px, int py)
    {
        x = px;
        y = py;
    }

    public List<int> XBits
    {
        get { return xBits; }
    }

    public List<int> YBits
    {
        get { return yBits; }
    }

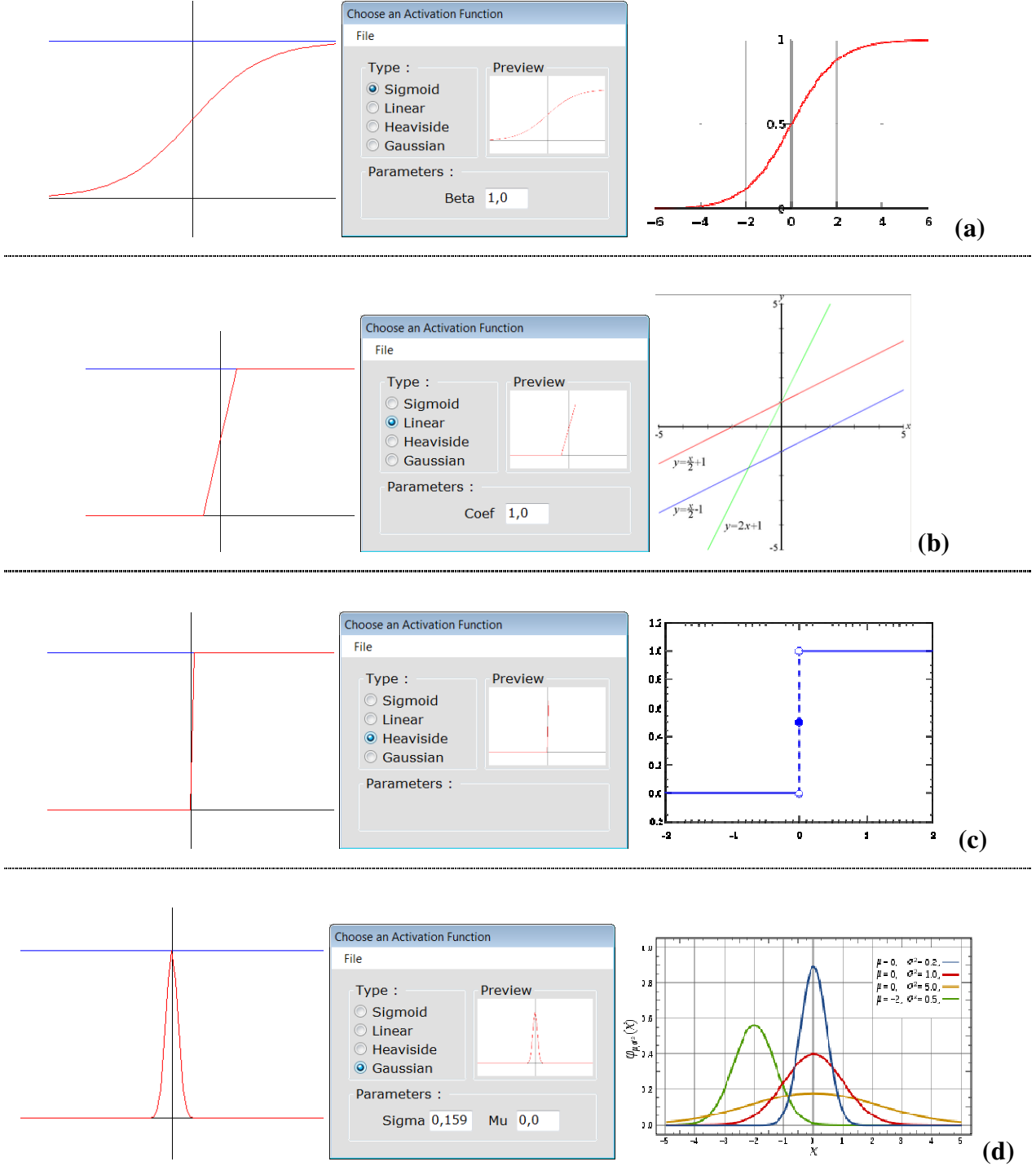
    public KeyPoint(List<int> pxBits, List<int> pyBits)
    {
        xBits = pxBits;
        yBits = pyBits;
    }
}
```

Bu sınıfa bir isim uzayı dâhil edilmiştir:

- using System.Collections.Generic;

Bu sınıf, *X* ve *Y* değerleri ile bu değerlerin ikilik sistemdeki karşılıklarını içermektedir.

EK E.1



Şekil A.5 : Aktivasyon fonksiyonları: (a)Sigmoid. (b)Linear. (c)Heaviside. (d)Gaussian.

YSA'nın eğitim ve test süreci için aktivasyon fonksiyonu olarak yukarıdaki şekilde verilen dört farklı fonksiyon kullanılabilir. Her fonksiyon için sırayla, yazılımda aktivasyon fonksiyonunun seçimine göre YSA çıkış değerlerine dayalı grafiğinin kaydedilmiş hâli, bu grafiğin formdaki görünümü ve bu fonksiyonun örnek grafiksel görüntüsü olacak şekilde üçer görüntü verilmiştir.

EK E.2

$$f(x) = \frac{1}{1 + e^{(-\beta x)}} \quad \beta > 0$$
$$f'(x) = \beta * f(x) * (1 - f(x)) \quad \text{(a)}$$

$$f(x) = \begin{cases} 1 & \text{if } x > 0.5/A \\ |A * x + 0.5 & \text{if } 0.5/A > x > -0.5/A \\ 0 & \text{if } -0.5/A > x \end{cases}$$
$$A > 0 \quad \text{(b)}$$

$$f(x) = 0 \text{ if } 0 > x$$
$$f(x) = 1 \text{ if } x > 0$$
$$\text{(c)}$$

Sqrt karekök alma işlemidir.

$$f(x) = \frac{1}{\sqrt{2 * \pi * \sigma}} * e^{-\frac{(x-\mu)^2}{2 * \sigma^2}}$$
$$f'(x) = y(x) * -2 * K * (x - \mu)$$

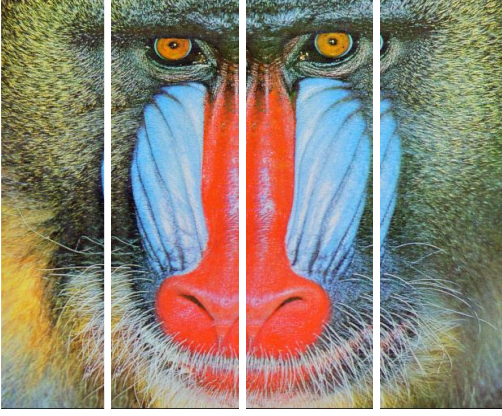
Daha verimli bir hesaplama için tanımlama şu şekilde yapılabilir:

$$C = 1/\sqrt{2 * \pi * \sigma}$$
$$K = 1/(2 * \sigma^2) \quad \text{(d)}$$

Şekil A.6 : Aktivasyon fonksiyonlarının tanımlamaları: (a)Sigmoid. (b)Linear. (c)Heaviside. (d)Gaussian.

Yukarıdaki dört aktivasyon fonksiyonu için verilen içerikler, bu fonksiyonların bu çalışma için kullanılan YSA kütüphanesi kapsamındaki tanımlamalarıdır.

EK F.1



Dikey dört iş parçacığı
(şablon görüntü 30x30)

(0_0, 128x512 ; 99_0, 157x512 ; 227_0,
157x512 ; 355_0, 157x512)

(Her alt görüntü parçasının referans görüntüdeki
sol üst pikselleri: X_Y , $M \times N$ (yatay ve düşey
boyutları))

(a)



Yatay dört iş parçacığı
(şablon görüntü 30x30)

(0_0, 512x128 ; 0_99, 512x157 ;
0_227, 512x157 ; 0_355, 512x157)

(b)



Dikey dört iş parçacığı
(şablon görüntü 30x30)

(0_0, 50x219 ; 21_0, 79x219 ; 71_0, 79x219 ;
121_0, 79x219)

(c)



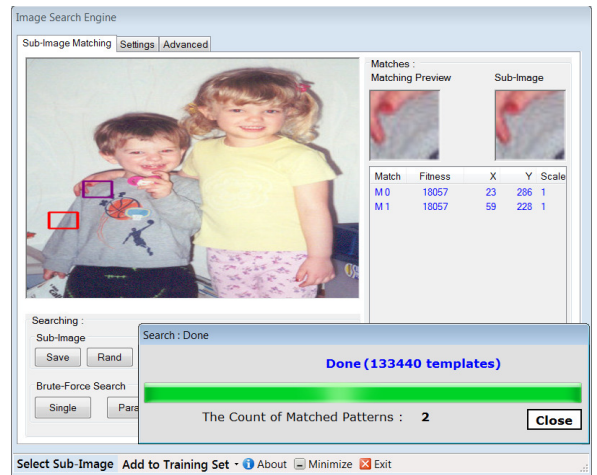
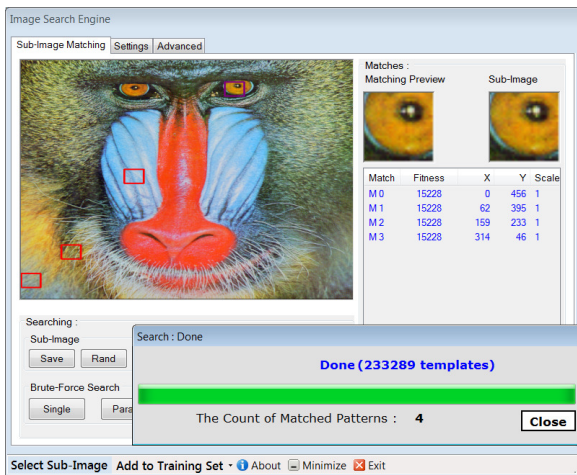
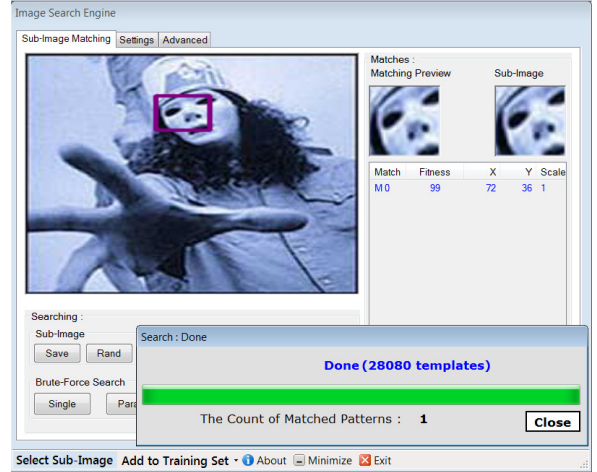
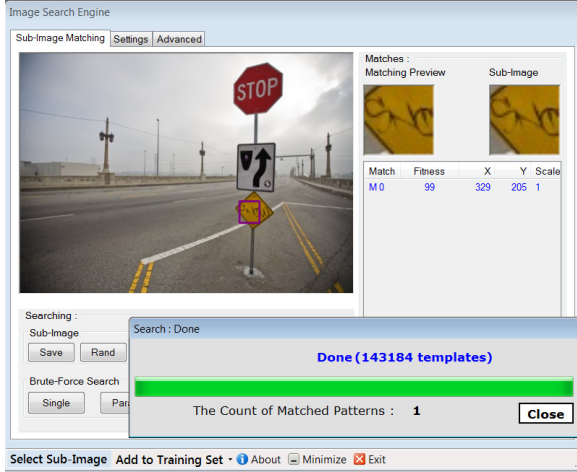
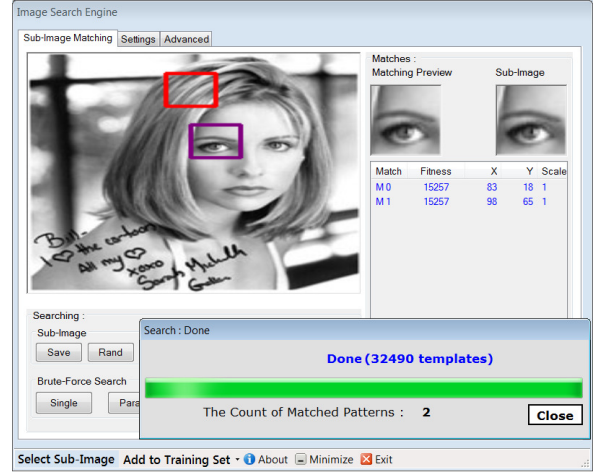
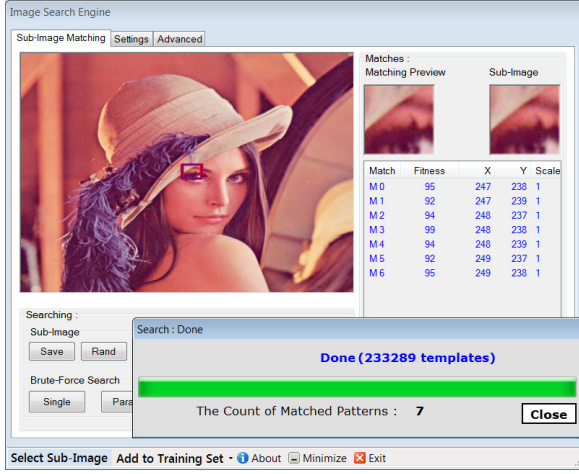
Dikey iki ve yatay iki iş parçacığı
(şablon görüntü 30x24)

(0_0, 256x256 ; 227_0, 285x256 ;
0_223, 256x279 ; 227_233, 285x279)

(d)

Şekil A.7 : Referans görüntü parçaları: (a)512x512. (b)512x512. (c)200x219.
(d)512x512.

EK G.1



Şekil A.8 : GİGA örnek uygulamaları.

Tablo C.1: GİGA'nın örnek uygulamaları.

Arama	Toplam Alt Görüntü	Referans Görüntü	Şablon Görüntü	Görüntü Eşleme Yöntemi	<i>BFS</i>		<i>SS</i>	
					<i>ST</i>	<i>MT</i>	<i>ST</i>	<i>MT</i>
1	32490	200x219	30x30	YSA	6825	3432	1651	3247
2	233289	512x512	30x30	ORD	2040	1069	-	-
3	32490	200x219	30x30	YSA	11523	7069	-	-
4	78075	512x512	166x288	ORD	15917	11867 (4)	-	-
5	233289	512x512	30x30	ORD	2915	2117	-	-
6	236187	512x512	30x24	Kenar tespiti	26133	16569 (4)	-	-
7	32490	200x219	30x30	YSA	6390	3930	-	-
8	133440	349x446	30x30	RDSS	2732	1277	-	-

Yukarıdaki tabloda ve takip eden tüm tablolarda arama sütunlarındaki değerler milisaniye cinsinden arama süreleridir. Ayrıca (4) şeklindeki ifadelerde parantezler arasındaki değerler iş parçacığı sayısını ifade etmektedir. Örnek uygulamaların gerçekleştirildiği bilgisayar şu özelliklere sahiptir:

- İşlemci (CPU) : 2.20 GHZ (çift çekirdek).
- Bellek (RAM) : 3,00 GB.

Tablo D.1: YSA ile dört farklı şekilde görüntü arama.

Arama	Arama Süresi (ms)	Doğruluk
<i>BFS , ST</i>	3474	(1D, 0Y)
<i>BFS , MT</i>	3116	(1D, 0Y)
<i>SS , ST</i>	1565	1D, 0Y
<i>SS , MT</i>	-	-

Tablo D.2: ORD ile dört farklı şekilde görüntü arama.

Arama	Arama Süresi (ms)	Doğruluk
<i>BFS , ST</i>	204	(1D, 1Y)
<i>BFS , MT</i>	178	(1D, 1Y)
<i>SS , ST</i>	151	1D, 0Y)
<i>SS , MT</i>	-	-

Tablo D.3: RDSS ile dört farklı şekilde görüntü arama.

Arama	Arama Süresi (ms)	Doğruluk
<i>BFS , ST</i>	612	(1D, 0Y)
<i>BFS , MT</i>	369	(1D, 0Y)
<i>SS , ST</i>	-	-
<i>SS , MT</i>	-	-

Tablo D.4: Korelasyon ile dört farklı şekilde görüntü arama.

Arama	Arama Süresi (ms)	Doğruluk
<i>BFS , ST</i>	1514	(2D, 0Y)
<i>BFS , MT</i>	-	-
<i>SS , ST</i>	-	-
<i>SS , MT</i>	-	-

Tablo D.5: Kenar tespiti ile dört farklı şekilde görüntü arama.

Arama	Arama Süresi (ms)	Doğruluk
<i>BFS , ST</i>	4095	(1D, 0Y)
<i>BFS , MT</i>	2487	(1D, 0Y)
<i>SS , ST</i>	-	-
<i>SS , MT</i>	-	-

ÖZGEÇMİŞ



Mehmet KARAKOÇ, 26 Haziran 1987 tarihinde doğmuştur. Mehmet Kemal Dedeman ve Hüseyin Avni Çöllü İlköğretim Okulları'nda ilköğrenimini tamamlamıştır. 2005 yılında Antalya Metin Nuran Çakallıklı Anadolu Lisesi'nden mezun olmuştur. 2009 yılında Pamukkale Üniversitesi Bilgisayar Mühendisliği Bölümü'nü bitirmesinin ardından, aynı üniversitede Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı'nda yüksek lisansa başlamıştır. Tez aşamasında hâlen öğrenimine devam etmektedir.

Ad Soyad: Mehmet Karakoç
Doğum Yeri ve Tarihi: Antalya - 26.06.1987
Adres: Yeşiltepe Mahallesi, 2672 Sokak, Gül Apartman,
No:15, K:2, D:5 / 07020 - Dokuma / Kepez - Antalya
Lisans Üniversitesi: Pamukkale Üniversitesi
Yayın Listesi: