

ETKİLEŞİMLİ DOĞA-ESİNLİ ALGORİTMALARLA YÜZ ÜRETME

PROJİYİ YAPANLAR

GÜL NİLDEM DEMİR

ALP EMRE KANLIKILIÇER

TUĞBA AKBAL

MERVE CAN KUŞ

PROJE YÖNETİCİLERİ

YRD. DOÇ. DR. A. ŞİMA UYAR

ÖĞR. GÖR. BİNNUR KURT

1. GİRİŞ

Bir suç işlendiğinde, polis genelde görgü tanığından şüphelinin yüzünü tarif etmesini ister. Amaç, görgü tanığının ifadesine göre şüphelinin yüzünü ana hatlarıyla oluşturmaktır. Bu anlamda günümüzde kullanılan üç farklı yüz üretme yöntemi vardır.

Birinci yöntemde tanık sözle şüphelinin yüzünü tarif ederken, bir ressam yüzün resmini çizmeye çalışır.

İkinci yöntemse E-FIT [1] ve Evo-FIT [2] gibi bilgisayar destekli bir sistem kullanmaktır. Bu tip sistemlerde ağız, göz, burun, kaş gibi yüz parçalarına ait çeşitli resimler elektronik bir veritabanında bulunur. Tanık, programı kullanan bir operatör aracılığıyla söz konusu veritabanından uygun olan yüz parçalarını seçer ve parçaların yüz üzerindeki yerleriyle şekillerini belirler.

Üçüncü yöntemde EvoFIT [3] ve EigenFIT [4] gibi bilgisayarla otomatik yüz üretme araçları kullanılır. Bu yazılımlarda, evrimsel algoritmalar [5] gibi doğa-esinli sezgisel yaklaşımlarla farklı yüzler üretilir. Kullanıcı, üretilen resimleri hedef resme benzerliklerine göre puanlar ya da seçer. Sonuçta iteratif olarak gittikçe daha iyi resimler üretilir.

İlk iki yaklaşım büyük ölçüde tanığın o anki psikolojik ve duygusal durumuna dayanır. Tanığın sadece şüpheliyi hatırlaması yetmez, bunun yanında ondan şüpheliye ait yüz parçalarını ayrıntılı olarak tasvir etmesi beklenir. Bu yöntemlerde tanığın, oluşan resmi gördüğü suçluya benzetebilmesi için, resimde hangi değişikliklerin yapılması gerektiğini bilmesi gerekir. Ayrıca eğitilmiş bir ressam ya da operatör, yöntemlerin başarıya ulaşmasında büyük önem taşır. Bilgisayar destekli yüz üretme programları yeteri kadar ikna edici yüzler oluşturabilme potansiyellerine rağmen, uygulamada yetersiz kalmaktadırlar. Araştırmalar oluşturulan resimlerin en fazla %20'lik bir başarı oranı ile tanındığını ortaya koymaktadır. Bunun sebebi insanların yüz parçalarına odaklanmaktan çok, yüzleri bir bütün olarak algılamaya yatkınlıklarıdır [3].

Ancak üçüncü gruptaki bilgisayarla otomatik yüz üretme araçlarında tanığın aklındaki imgeyi sözle tasvir etmesi beklenmez. Bazen insanlar tek bir resme baktıklarında, o resmin hedef resimle arasındaki benzerliği ya da farklılığı kelimelerle ifade edemeyebilirler. Ama bir resmin kümesi içerisinde benzeyenleri ayırt etmek ya da hiç benzemeyenleri elemek çok daha kolaydır. Üçüncü gruptaki yaklaşımlara dahil olan projemiz de bu özelliği ile piyasadaki çoğu yüz üretme uygulamasından ayrılmaktadır. Projede sistem, yüz üretmek için yüz parçalarına ait resimleri birleştirmek yerine, kullanıcıya belli bir miktar yüz resmi sunacak ve ondan bu yüzleri çizdirmek istediği resme benzerliğe göre puanlamasını / seçmesini isteyecektir. Sunulan yüz resimlerinin puan / seçme değerlerine göre bir etkileşimli doğa-esinli sezgisel yaklaşım ile yeni resim kümesi oluşturulacaktır. Literatürdeki başarılı iki örnekten (EvoFIT [1] ve EigenFIT [2]) esinlenen bu yapımız sayesinde umut verici sonuçları olan bir yüz üretme uygulaması gerçekleştirdik.

Hedef resmi kullanıcı-dostu bir arayüz aracılığıyla üretmek için aktif görünüm modelinin (AGM [11]) T boyutlu parametre vektörünü kullandık. Problemi bir optimizasyon problemi olarak ele aldık ve hedef resme en çok benzeyen resmin AGM [11] parametre vektörünü oluşturmak için etkileşimli doğa-esinli sezgisel yaklaşımlar kullandık. Kullanılan etkileşimli doğa-esinli yaklaşımlar; etkileşimli kuşaksal ve kalıcı-durumlu genetik algoritmalar (EGA) [5], etkileşimli evrimsel stratejiler (EES) [18], etkileşimli diferansiyel evrim (EDE) [19] ve

etkileşimli parça sürüsü optimizasyonudur (EPSO) [17]. AGM [11] modelini kurmak için okulumuzdaki öğretim görevlisi, araştırma görevlisi, öğrenci ve yönetim kadrosu çalışanlarının resimlerinden oluşan 51 resimlik bir veritabanı oluşturduk. Tüm sezgisel yaklaşımların gerçekleşmesinden sonra, yaklaşımların başarımlarının ölçülmesi için testler yaptık. Testler algoritmaları iki farklı açıdan değerlendirmek için tasarlandı: Algoritmaların çözüm kalitesi ve kullanım kolaylıkları. Algoritmaların çözüm kalitesine başarımların testleri, algoritmaların ürettikleri resimlerle hedef resimlerin AGM'ye izdüşüm resimleriyle farklarını yorumlar. Kullanım kolaylıklarına başarımların testlerinde ise her algoritmanın ne kadar kullanıcı dostu olduğu incelenmiştir.

İkinci bölümde, gerçekleştirilen algoritmalarla ilgili kuramsal bilgilere yer verilmiş, aktif görünüm modeli açıklanmıştır.

Üçüncü bölümde, projenin modelleme ve gerçekleştirme aşamaları ile projenin nasıl yapıldığı anlatılmıştır.

Dördüncü bölümde, test sonuçları ve yorumları verilmiştir. Elde edilen sonuçlar başarımların değerlendirilmiş, sistemin yararları belirtilmiştir.

2. Kuramsal Bilgiler

2.1. Evrimsel Stratejiler

2.1.1. Temel Tanım

ES algoritması [22][23][24], n adet bireyden oluşan bir toplum üzerinde çalışır. Başlangıç toplumu rasgele olarak üretilir. Her bireyin gerçel sayıların oluşturduğu bir vektör olan kromozomu bulunur. Kromozomlar iki bölümden oluşur: Hedef parametreleri olan genler (x_1, \dots, x_n) ve strateji parametreleri olan mutasyon adım büyüklükleri olan sapmalar $(\sigma_1, \dots, \sigma_n)$. Böylece bireylerin genel formu $(x_1, \dots, x_n, \sigma_1, \dots, \sigma_n)$ şeklinde olur. Her kromozom, hedef üyeyi bulmada algoritmanın olası bir çözümünü temsil eder. μ toplumdaki bireylerin sayısını yani ebeveyn sayısını simgeler. Bu toplumdan çocuklar üretilir. λ çocukların sayısını temsil eder. λ adet çocuğun üretilmesi için, ebeveyn seçimi, rekombinasyon ve mutasyon adımları birbirini izleyecek şekilde λ kez tekrarlanır. Bu adımlar ilerleyen bölümlerde daha ayrıntılı olarak açıklanacaktır. Çocukların üretilmesinden sonra hayatta kalacakların seçimi gerçekleştirilir. Bu aşamada hayatta kalanlar seçilen stratejiye göre ya sadece çocukların oluşturduğu kümeden ya da ebeveyn ve çocukların oluşturduğu kümeden seçilirler. Hayatta kalacakların seçilmesinden sonra, seçilen bireyler yeni toplumu meydana getirirler. Aşağıdaki adımlar var olan bir toplumdaki yeni bir toplum üretilmesi sürecinin temel taslağını oluşturur. Bütün bu adımlar, hedefe yeterince benzer olan bir birey bulunana kadar gerçekleştirilir.

ES Algoritması:

```
başlangıç toplumunu rasgele üret
tekrarla
    tekrarla  $\lambda$  kere
        rasgele 2 ebeveyn seç
        rekombinasyonu gerçekleştir
        mutasyonu gerçekleştir
    son
    hayatta kalacakları seç
sonlanma kriteri sağlanana kadar
```

Ebeveyn seçiminde iki ebeveyn, toplumdan rasgele olarak seçilir.

Rekombinasyon, seçilen iki ebeveyninden bir çocuk üretir. Bu çalışmada iki tip rekombinasyon kullanılmıştır:

- Ayrık rekombinasyon: Ebeveynlere ait değerlerden biri seçilir. Yani çocuğa ait bir değişken olan z_i , rasgele olarak ya ilk ebeveynin ilgili değişkeninin değeri olan x_i ya da diğer ebeveynin ilgili değişkeninin değeri olan y_i olarak seçilir.
- Ara değer rekombinasyonu: Ebeveynlere ait değerlerin ortalaması alınır: $z_i = (x_i + y_i) / 2$.

Rekombinasyonun gerçekleştirilip gerçekleştirilmeyeceğine karar vermek için 0 ile 1 arasında rasgele bir sayı üretilir, eğer bu sayı önceden değeri belirlenmiş bir değişken olan P_c ' den küçük veya eşitse genler için ayrık rekombinasyon, sapmalar için ise ara değer

rekombinasyonu gerçekleştirilir. Eğer üretilen rasgele sayı P_c ' den büyükse, ebeveynlerden biri rasgele seçilir ve seçilen ebeveynin genleriyle sapmaları çocuğa kopyalanır. Böylece ebeveynlere ait 2 kromozomdan yeni bir kromozom üretilir.

Yeni kromozom mutasyon operatöründen geçer. Öncelikle, mutasyon adım büyüklükleri aşağıdaki formüle göre mutasyona uğrattılır ($\sigma \rightarrow \sigma'$):

$$\sigma'_i = \sigma_i \cdot \exp(\tau \cdot N_i(0,0.3))$$

Burada kendinden-uyarlanabilir mutasyon parametresi olan τ ' un değeri $1/(2n)^{1/2}$ dir. n , toplumdaki birey sayısıdır. $N(0,0.3)$ sıfır ortalamalı ve 0.3 standart sapmalı Gauss dağılıma göre rasgele belirlenmiş bir sayıyı gösterir. Sonra, yeni mutasyon adım büyüklükleri kullanılarak aşağıdaki formül ile genler mutasyona uğrattılır ($x \rightarrow x'$):

$$x'_i = x_i + \sigma'_i \cdot N_i(0,0.3)$$

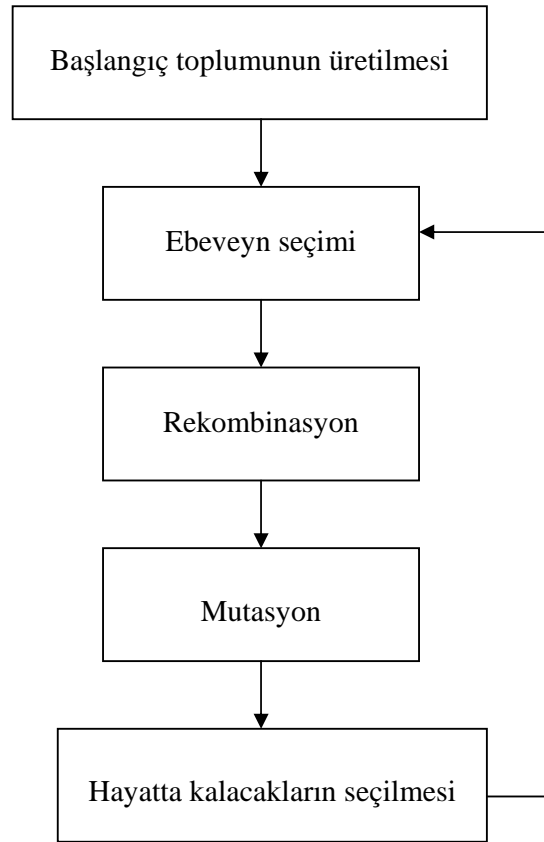
Böylece mutasyon sonucunda $(x_1, \dots, x_n, \sigma_1, \dots, \sigma_n)$ kromozomu $(x'_1, \dots, x'_n, \sigma'_1, \dots, \sigma'_n)$ halini alır.

Eğer yeni bir gen veya mutasyon adım büyüklüğü değeri önceden kararlaştırılmış sınır değerli aşarsa aynalama yapılarak değerlerin sınırlar arasına düşürülmesi sağlanır.

Son aşama hayatta kalacakların seçimidir. Bu aşamada kullanılabilen iki tip strateji vardır:

- (μ, λ) seçimi (virgül stratejisi): Seçim sadece çocukların oluşturduğu kümeden yapılır, ebeveynler atılır.
- $(\mu + \lambda)$ seçimi (artı stratejisi): Seçim ebeveynler ve çocuklar arasından yapılır. Seçim, bireylerin uygunluk değerlerinin sıralanması ve bunların ilk μ tanesinin seçilmesi yoluyla gerçekleştirilir.

2.1.2. Algoritmanın Genel Akışı



Şekil 2.1: ES blok diyagramı

2.2. Parça Sürüsü Optimizasyonu

2.2.1. Tarihsel Gelişim

Parça sürüsü optimizasyonu (PSO) nümerik ve nitel problemlere optimal sonuçlar bulmak için geliştirilmiş yeni tür bir algoritmadır (1995). PSO, sosyal-psikolog olan James Kennedy ve elektrik mühendisi Russel Eberhart [20] tarafından balık ve kuş sürülerinden ilham alınarak geliştirilmiştir. Yiyecek bulmaya çalışan bir kuş topluluğu bir senaryodur. Onlar yiyeceğin gerçek yerini bilmemelerine rağmen, yiyecekte ne kadar uzakta olduklarını öğrenmeye çalışırlar. Kuşlar tarafından kullanılan yöntem yiyeceğe en yakın olan (en iyi konum) kuşu izlemektir. PSO' da kuşlar parçalar olarak temsil edilirler ve bu parçalar sona ulaşmak için her iterasyonda en iyi değere göre güncellenirler [21].

Bir optimizasyon problemindeki en önemli nokta uygunluk fonksiyonunun belirlenmesidir. Uygunluk fonksiyonu, beklenen çözümün amaçlarını matematiksel bir fonksiyonla ifade eder. Genelde, optimizasyonun amacı bu fonksiyonun en yüksek ya da en düşük değerinin bulunmasıdır [19].

2.2.2. Temel Tanım

PSO algoritmasında başlangıç popülasyonu parça adı verilen problemin olası çözümlerinden oluşur (sürü). Parçalar rasgele olarak üretilirler. Bir parçanın konum adı verilen ve x ile temsil edilen n -boyutlu vektörü vardır; n nesne değişkenleridir. Parçanın amaç fonksiyonuna bağlı bir uygunluk fonksiyonu ($f(x)$) vardır. Ayrıca parça hız adı verilen ve v ile temsil edilen n -boyutlu bir vektöre daha sahiptir. Bu vektör parçanın olası hareketlerini gösterir. Popülasyondaki en iyi parça p ile temsil edilir ve tüm iterasyonlarda o ana kadarki en iyi parça olan global en iyi parça g ile gösterilir. İterasyonların maksimum sayısını t_{max} ve o anki iterasyonu t değişkenleri gösterir [20].

PSO Algoritması [20]

```
prosedür PSO {
  Parçanın konumunu ( $x$ ) ve hızını ( $v$ ) başlangıç durumuna getir
   $t = 1$ 'den  $t_{max}$  'a kadar {
     $j = 1$ 'den  $N$ ' e kadar {
      Uygunluk değerini  $f(x_j)$  hesapla;
      eğer  $f(x_j) > f(p_j)$  ise  $p_j = x_j$ ;
    }
    Global en iyi parçayı seç:  $g = \text{argmax}f(p_j)$ ;
     $j = 1$ 'den  $N$ ' e kadar {
      Parçanın  $x_j$  konumunu güncelle;
      Parçanın  $v_j$  hızını hesapla;
    }
  }
}
```

Problemin aday çözümleri, yani parçalar, gerekli bilgiyi bir arada tutmak için bir yapı olarak tanımlanırlar. Parçanın konumu, hızı ve uygunluk değeri tek bir yapıda tutularak, kodun gerçekleşmesini kolaylaştırılır.

Uygunluk fonksiyonun doğru tespit edilmesi çok önemlidir, çünkü uygunluk değerine algoritma parçaların hedefe göre nasıl yön değiştirmelerine karar verir. Bu fonksiyon yanlış belirlendiğinde parçalar hedeften uzaklaşarak algoritmanın düşük bir başarımlı göstermesine sebep olurlar.

Başlangıç popülasyonu genelde her parçanın konum ve hız bileşenlerine rasgele sayılar atanarak oluşturulur. Popülasyon başlangıç durumuna getirildikten sonra, algoritma belli bir miktarda tekrarlanır.

Global en iyi parçanın seçimi altındaki temel fikir, tüm popülasyonda uygunluk fonksiyonu yardımıyla hedefe en yakın parçanın bulunmasıdır.

Güncelleme ile parçaların konum ve hız bileşenleri parçanın uygunluk değerine göre hedefe yaklaşmak için yeni değerler alırlar.

Her iterasyonda parçanın konum ve hız bileşenleri aşağıdaki formüle göre güncellenir:

j . parçanın t . iterasyonda konumunun güncellenmesi:

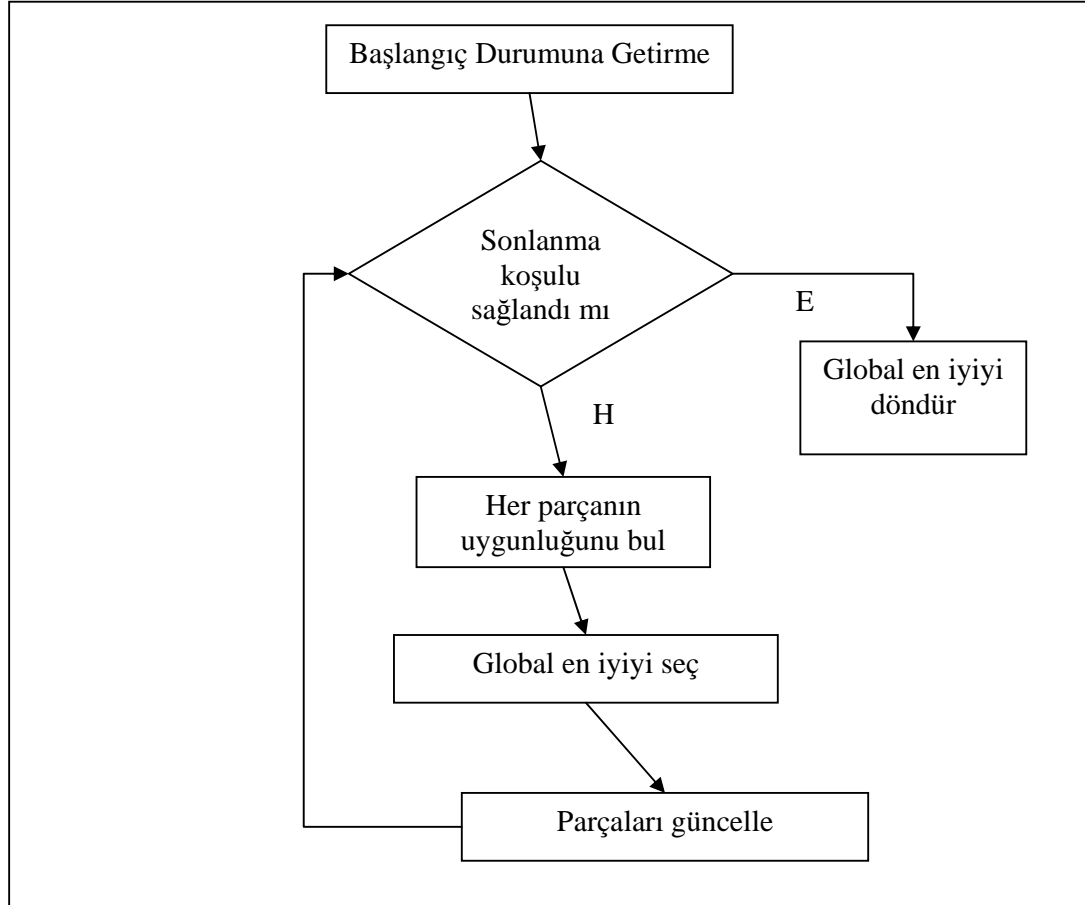
$$x_j(t+1) = x_j(t) + v_j(t)$$

j. parçanın t. iterasyonda hızının güncellenmesi:

$$v_j(t+1) = w(t)v_j(t) + c_1u_1(p_j(t) - x_j(t)) + c_2u_2(g_j(t) - x_j(t)) \text{ [hi2]}$$

Sonlanma koşulu PSO algoritmasının ne zaman sonlanacağını belirtir. Bu koşul uygulamaya göre farklılık gösterir. Belli bir iterasyon miktarı olabileceği gibi belli bir süre de olabilir.

2.2.3. Algoritmanın Genel Akışı



Şekil 2.2: PSO blok diyagramı

2.3. Genetik Algoritmalar

2.3.1. Tarihsel Gelişim

Genetik algoritmalar, dayanak noktasını doğal seleksiyon ve genetik biliminden alan adaptif, sezgisel ve çok yönlü arama algoritmalarıdır. Genetik algoritmaların temel kavramları başta Charles Darwin'in en güçlüsünün hayatta kalması prensibi olmak üzere doğal sistemdeki evrimsel süreçleri taklit etmek için tasarlanmıştır. [8]

Genetik algoritmalar ile optimizasyon fikri ilk olarak Michigan Üniversitesi profesörü John Holland tarafından 1970'lerde ortaya atılmıştır. Holland'ın 1975 yılında yayımladığı "Adaptation in Natural and Artificial System" (Doğal ve yapay sistemlerde adaptasyon) isimli kitabında karmaşık problemlerin çözümünde yapısal bir arama yapmak için matematik tabanlı yapay evrim yöntemi incelemiştir. [9]

Genetik algoritmaların diğer bir öncüsü ise Dr. David E. Goldberg'tür. 1989 tarihinde yayımlanan "Genetic Algorithms in Search, Optimization and Machine Learning" başlıklı yazısında geniş mühendislik alanlarında genetik algoritma uygulamaları önermiştir. 1996 tarihli bir Wall Street Journal yazısı genetik algoritmalar üzerinde Dr. Goldberg'ten bir alıntı yapmıştır: "Üç milyar yıllık evrim yanlış olamaz. Oradaki en güçlü algoritmadır." [8]

2.3.2. Genetik Algoritmaların Genel İşleyişi

Bir problemin çözümünde genetik algoritma kullanabilmek için ilk yapılması gereken, olası çözümlerin temsil edilmesi için bir yöntem belirlenmesidir. Aday çözümler çok karmaşık yapıları olan fenotipler olarak ele alınabilirler. Değişim operatörlerini bu yapılar üzerinde uygulamak bazen çok zor olabileceğinden fenotipler, genotiplerle temsil edilirler. Genotipler bit katarları, reel sayı vektörleri, tam sayı permutasyonları, ağaçlar gibi formlarda olabilirler. Problemin tanımına göre bu temsil çeşitlerinden biri seçilir. [5]

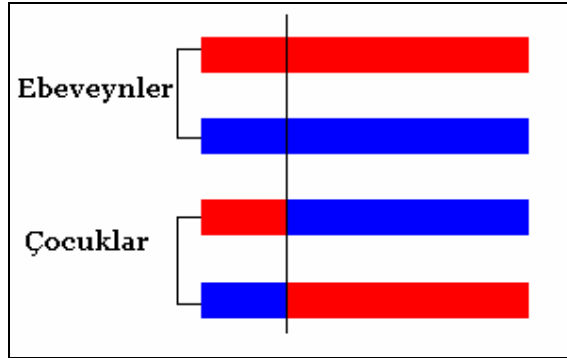
Genetik algoritmalarda her iterasyonda çözüm kümesinin iyileşmesini sağlayan etken uygunluğa dayalı seleksiyondur. Bu yüzden uygunluk fonksiyonunun doğru belirlenmesi çok önemlidir. Sonraki adım başlangıç popülasyonunun oluşturulmasıdır. [5]

Başlangıç popülasyonu genelde kromozomların genlerine belli bir dağılımdan rasgele değerler atayarak oluşturulur. Popülasyon başlangıç durumuna getirildikten sonra, evrimsel süreç belli bir miktarda tekrarlanır.

Evrimsel sürecin ilk operatörü seleksiyondur. Seleksiyon her nesilde belli bireylerin üremek üzere seçilmeleridir. Bu seçim bireylerin uygunluk değerlerine dayanarak yapılır. Daha uygun bireylerin seçilme şansları diğerlerine göre daha yüksektir. En çok kullanılan seleksiyon çeşitleri rulet tekerleği, sıralama seleksiyonu, turnuva yöntemidir. Rulet tekerleği yönteminde bireyin seçilme olasılığı o bireyin mutlak uygunluğunun popülasyonun geri kalanının mutlak uygunluğuyla karşılaştırılmasına dayanır. Sıralama seleksiyonunda bireyler uygunluk değerlerine göre sıralanırlar ve seçilme olasılıkları sıralamadaki yerlerine bağlıdır. t faktörlü turnuva seleksiyonunda, popülasyon boyu N ise; N kez rasgele t adet birey seçilir. Her t bireyli grup arasından uygunluğu en yüksek olan birey seçilir. Seleksiyon sonucunda birleşme havuzu oluşturulur. Bu havuzdaki bireyler çaprazlanarak yeni çocuk bireyler oluşturacaklardır. [5]

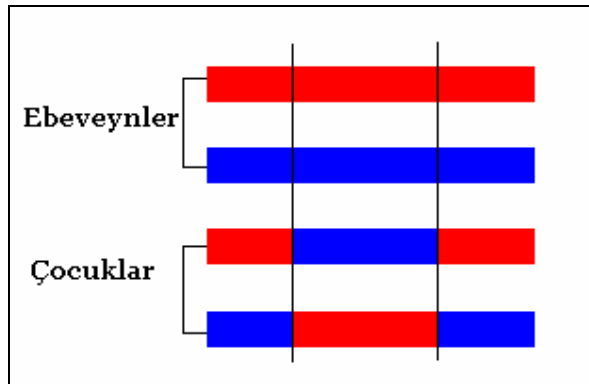
Çaprazlama operatörü iki ya da daha fazla bireyi birleşme havuzundan alır; bireyler arasındaki genetik bilgiyi değiş-tokuş ederek yeni bireyler oluşmasını sağlar. Çaprazlama operatörü çaprazlama oranı ile ilişkilidir. Belli bir dağılımdan seçilecek çaprazlama olasılığı, önceden belirlenen çaprazlama sınır değerinden küçük olduğu takdirde, seçilen bireyler çaprazlamaya tabi tutulur. Farklı çaprazlama yöntemi vardır. Bunlardan bazıları:

- Tek noktalı çaprazlama: Rasgele bir çaprazlama noktası seçilir. O noktaya kadarki tüm genetik bilgi ebeveynler arasında değiştirilir.



Şekil 2.3: Tek noktalı çaprazlama

- İki noktalı çaprazlama: Rasgele iki çaprazlama noktası seçilir. Bu iki nokta arasında kalan genetik bilgiler ebeveynler arasında değiştirilir.



Şekil 2.4: İki noktalı çaprazlama

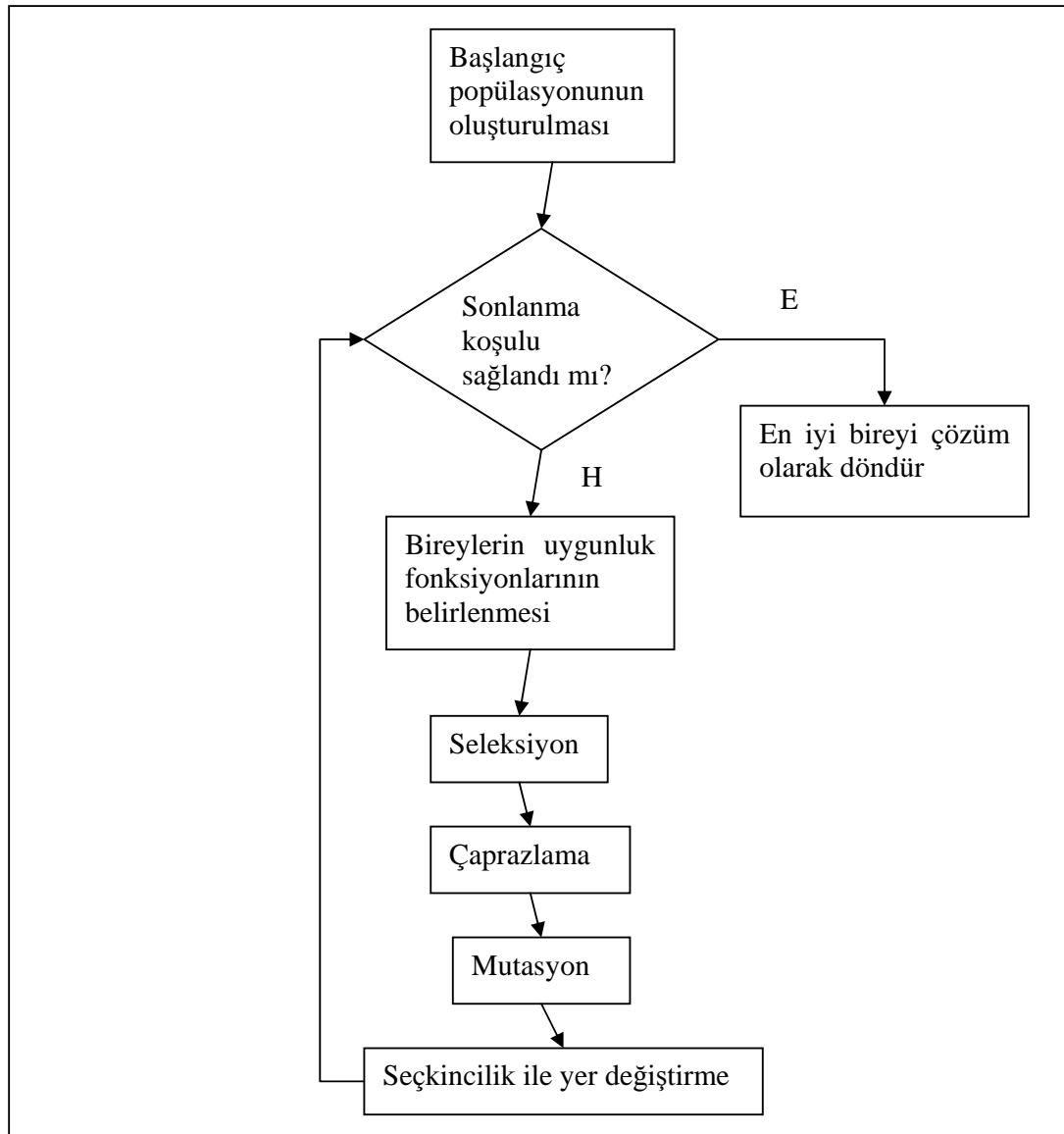
Çaprazlama sonucu oluşan çocuk popülasyondaki bireyler mutasyona tabi tutulurlar. Mutasyon operatörü bir bireyin genetik bilgisini belli bir olasılık dağılım fonksiyonuna göre değiştirir. Mutasyon operatörü mutasyon oranı ile ilişkilidir. Her bir kromozomdaki her gen için, belli bir dağılımdan seçilecek mutasyon olasılığı, önceden belirlenen mutasyon sınır değerinden küçük olduğu takdirde, ilgili gen mutasyona uğratılır. Standart bir mutasyon yöntemi yoktur. Bit katarı olarak temsil edilmiş kromozomlarda bitlerin değiştirilmesi kullanılırken, reel sayılı temsillerde belli bir dağılımdan rasgele seçilen bir değişkenin gen değeri ile toplanması tercih edilir. İkinci durumda seçilecek olasılık dağılımın fonksiyonu ve parametrelerinin doğru belirlenmesi önemlidir. [5]

Mutasyon operatörü sonucunda son aşama olan yer değiştirmeye sıra gelir. Genetik algoritmalar yer değiştirme göre ikiye ayrılırlar: Kuşaksal genetik algoritmalar ve kalıcı-

durumlu genetik algoritmalar. Kuşaksal genetik algoritmalar, genetik algoritmaların her nesilde tüm popülasyonun yer değiştirdiği bir çeşididir. Her nesilde, evrimsel süreç sonunda oluşan çocuk bireyler ebeveynlerinin yerini alırlar. Kalıcı durumlu genetik algoritmalar, genetik algoritmalara, her nesilde popülasyonun hiç yer değiştirmedeği ya da çok az yer değiştirdiği zaman verilen addır. Her nesilde, seleksiyon aşamasında tek bir çift ebeveyn seçilir. Çaprazlama sonucunda ise tek bir çocuk birey oluşur ve bu birey mutasyona uğratılır. Eğer çocuk bireyin uygunluk fonksiyonu, ebeveyn bireylerden birinden yüksekse, çocuk bireyin onun yerini alır. Aksi takdirde popülasyon aynen korunmuş olur.

Popülasyonlar üzerinde seleksiyon, çaprazlama, mutasyon ve yer değiştirmeden oluşan evrimsel sürecin ne kadar tekrarlanacağı sonlanma koşulu ile belirlenir. Bazı durumlarda hesaplama süresi olabileceği gibi, sonlanma belli bir iterasyon miktarı ile de belirlenebilir.

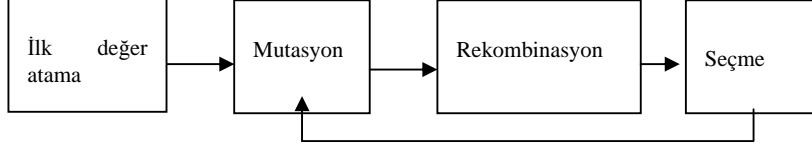
2.3.3. Genel Akış



2.5. Diferansiyel Evrim

2.5.1. Temel Tanım

Diferansiyel evrim 1996 yılında Storn ve Price tarafından ortaya atılmıştır. DE analitik olarak çözülemeyen problemleri çözmek için kullanılan bir optimizasyon algoritmasıdır ve popülasyon tabanlıdır. DE' nin blok diyagramı aşağıda gösterilmiştir.



Şekil 2.4: DE blok diyagramı

DE' nin arkasında yatan fikir test parametrelerini oluşturmak için yeni bir yöntemdir. DE 2 popülasyon elemanının ağırlıklandırılmış farkını 3. elemana ekleyerek yeni parametre vektörleri üretir. Eğer yeni üretilen birey popülasyonda bulunan önceden belirlenmiş bireyden daha iyi bir fonksiyon değerine sahipse yeni üretilen birey diğerinin yerini alır. Karşılaştırma yapılan önceden belirlenmiş olan vektör, yeni oluşan vektörün üretilmesinde kullanılabilir ancak bu zorunlu değildir.

İlk değerlerin atanmasında parametrelerin sınır değerleri belirlenir ve popülasyonda bulunan her birey için parametre vektörleri rasgele üretilir.

Popülasyon da bulunan her birey aşağıdaki x parametre vektörü ile temsil edilir. Burada D parametre sayısıdır.

$$x_{i,G} = [x_{1,i,G}, x_{2,i,G}, \dots, x_{D,i,G}] \quad i = 1, 2, \dots, N$$

Mutasyon adımında 3 tane parametre vektörü (x_{r1}, x_{r2}, x_{r3}) popülasyonda bulunan her birey için seçilir. Bu seçim de bütün vektörler birbirinden farklıdır ve seçim rasgele yapılır.

Mutasyon aşaması iki vektörün ağırlıklı farkının üçüncüsüne eklenmesi ile devam eder.

$$V_{i,G+1} = X_{r1,G} + F(X_{r2,G} - X_{r3,G})$$

Burada F mutasyon katsayısıdır ve 0 ile 2 arasında bir değere sahiptir. Sonuç vektörü V_i verici vektör olarak adlandırılır.

Rekombinasyon adımında yeni bireyler verici vektör ve popülasyonda ki orijinal bireyin vektörü kullanılarak oluşturulur. Kombinasyon aşaması aşağıda belirtilen kurala göre oluşturulur.

$$U_j = \begin{cases} V_j & j = \langle n \rangle_d, \langle n+1 \rangle_d, \dots, \langle n+L-1 \rangle_d \\ (X_{i,g})_j & \text{diğer} \end{cases}$$

Burada V_j j . parametre için verici vektördür. D sayısı bir bireyin sahip olduğu toplam parametre sayısıdır. L ise 0 ile D arasında üretilen bir rasgele sayıdır. X_i eski kombinasyonlardan gelen, popülasyonda var olan bireyin parametre vektörüdür. Burada $\langle n \rangle_d$ simgesi $n \bmod d$ yi temsil etmektedir.

Rekombinasyondan sonra seçme işlemi bir uygunluk fonksiyonu f tarafından yapılır. Popülasyona aktarılacak yeni bireyler bu fonksiyon değerlerine göre seçilir.

Mutasyon, rekombinasyon ve seçme işlemleri istenilen sonuç elde edilinceye kadar devam eder.

2.6. Aktif Görünüm Modeli

2.6.1. Matematiksel Arka Plan

Temel Bileşen Analizi

Temel bileşen analizi (TBA) olabildiğince çok bilgiyi korumaya çalışarak bir veri kümesinin boyutluluğu indirgemeye yarayan bir tekniktir. Orijinal değişkenlere tarafından tanımlana uzayda var olan eksenleri yeni konumlarına döndürmek olarak düşünülebilir. Bu döndürmede döndürme ile tanımlanmış yeni değişkenlerle bir ilişki olmayacaktır. İlk yeni değişken varyasyonun en fazla oranını içerirken ikinci yeni değişken birinciye dik ve onun tarafından anlatılmayan varyasyona sahip olacaktır. Bu diğer değişkenler için de böyle devam edecektir. [6]

Kovaryans modeline göre TBA aşağıdaki adımlardan oluşur:

- 1) Veri kümesini düzenle.
- 2) Her boyutun empirik ortalamasını hesapla.
- 3) Ortalamadan sapmaları hesapla.
- 4) Kovaryans matrisini bul.
- 5) Kovaryans matrisinin öz vektörlerini ve öz değerlerini bul.
- 6) Öz vektör ve öz değer matrislerini azalan öz değerlere göre yeniden düzenle.
- 7) Temel vektörler olarak öz vektörlerden bir alt küme belirle.
- 8) Kaynak veriyi z-skorlarına çevir.
- 9) Verinin z-skorlarını temel vektörler üzerine düşür.

Özyüzler

TBA, yüz resimlerine uygulandığında, özyüzler olarak adlandırılan bir öz vektör kümesi oluşturur. Bu özyüzler, birlikte yüz resimleri arasındaki varyasyonu karakterize eden özellikler kümesi olarak düşünülebilir. Herhangi bir insan yüzü bu özyüzlerin lineer kombinasyonu olarak ifade edilebilir. Özyüz üretmek için eşit şartlar altında çekilmiş ve gözlere hizalanmış insan yüzü resimlerinde oluşan geniş bir resim veritabanına gerek vardır. Mesela Crow ve Cameron yüz resmini n adet nokta ile işaretlemişlerdir. Bu noktalar yüzün önemli parçaları etrafında konumlanmışlardır (gözler, kaşlar, ağız, burun, çene, kulaklar). Bu noktaların ortalama konumları hesaplanır ve bu noktalar baz alınarak resim üçgenler haline getirilir. Sonra veritabanındaki her yüz ve ortalama yüze göre değiştirilir ve yüze TBA uygulanır. [3]

2.6.2. AGM Tanımı

Aktif Görünüm Modeli [11] insan yüzlerinin bir örnek veritabanından modellenmesine ve veritabanında olan ya da olmayan yüz resimlerinin üretilmesi olanak verir. Bu yüzden AGM [11] farklı nesnelere temsil etmeye yarayan güçlü bir üretimsel model olarak düşünülür. AGM [11] şu prensibe göre çalışır: Bir yüz resmi n nokta ile işaretlenir. Resmin içeriği TBA 'ya dayanarak hem şekil hem doku açılarından analiz edilir. Yüz şekli üçgenlerden oluşan bir ağ ve ağ üzerindeki kenar noktaları ile tanımlanır. Matematiksel olarak şekil modeli şu şekilde temsil edilir:

$$x = [x_1, x_2, \dots, x_n, y_1, y_2, \dots, y_n]$$

Yüz dokusu bu noktalar üzerindeki yoğunluktur (şekle göre normalize edilmiş renk beneği) ve $g = [g_1, g_2, \dots, g_n]$ formülüyle ifade edilir.

Yüz şekil ve doku TBA ile daha kısa bir forma getirilir, öyle ki

$$\begin{aligned}x &= \bar{x} + \Phi_s b_s \\g &= \bar{g} + \Phi_g b_g\end{aligned}$$

Bu formda, Φ_s en büyük öz değerlere denk düşen t adet öz vektörü içerir ve b_s t -boyutlu bir vektördür. b_s 'deki parametreleri değiştirerek şekilde değişebilir. Dokunun lineer modelinde, Φ_g varyasyonun dik modlarıdır ve b_g gri-seviye parametre kümesidir.

Şekil ve doku arasındaki ilişkiyi kaldırmak için, üçüncü bir TBA kombine model parametrelerine uygulanır, öyle ki

$$\begin{aligned}x &= \bar{x} + \Phi_s W_s^{-1} Q_s c \\g &= \bar{g} + \Phi_g Q_g c\end{aligned}$$

ve $b = \begin{bmatrix} W_s b_s \\ b_g \end{bmatrix}$ and $b = \begin{bmatrix} Q_s \\ Q_g \end{bmatrix} c$.

Bu formda, W_s her şekil parametresi için ağırlıkların diyagonal matrisidir; c modelin şekil ve gri-seviyesini kontrol eden görünüm parametreleri vektörüdür. Q_s şekil modelinin öz vektörleridir ve Q_g doku modelinin öz vektörleridir [7].

3. TASARIM VE GERÇEKLEME

3.1. Tasarım

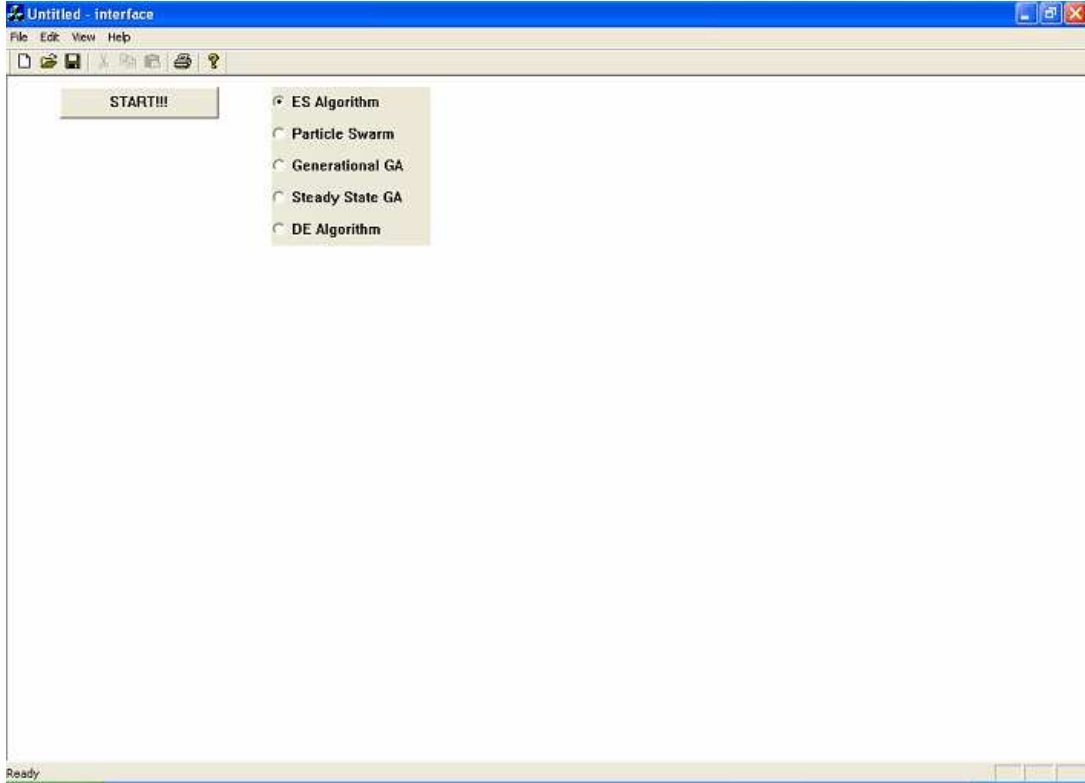
Proje tasarımı bakımından iki kısımda ele alınabilir:

1. Kullanıcı arayüzü tasarımı.
2. Algoritma tasarımı

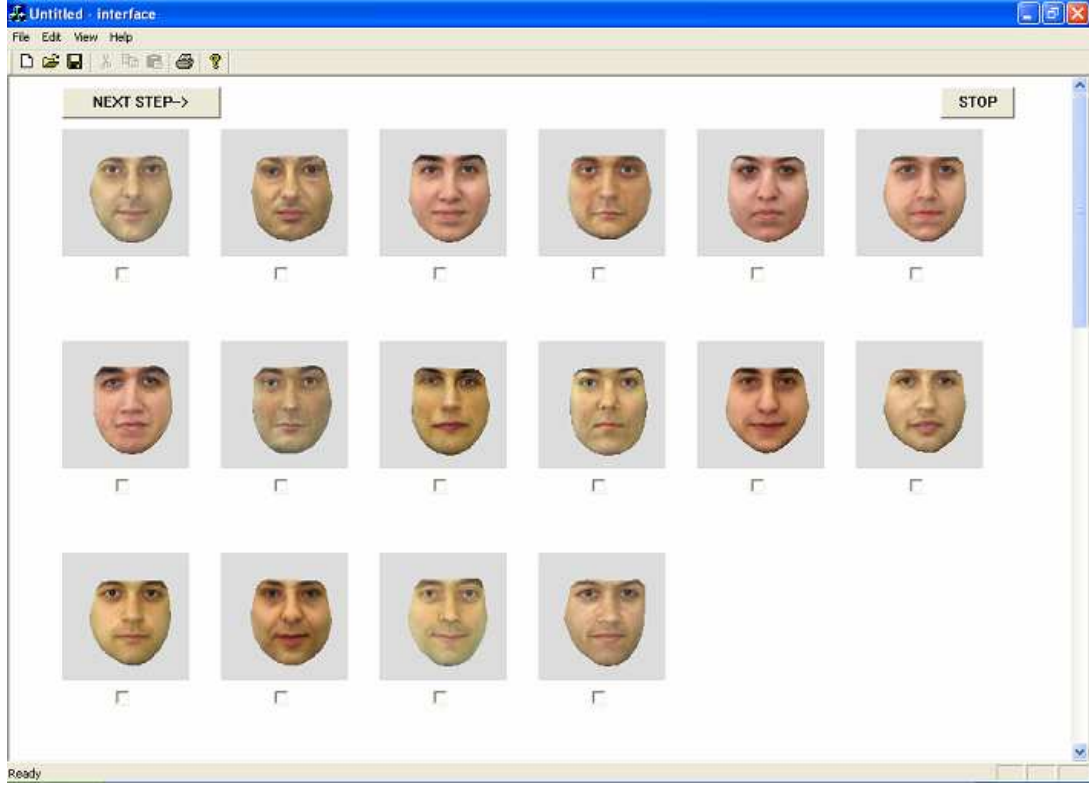
3.1.1. Kullanıcı arayüzü tasarımı

Projenin kullanıcı arayüzü 5 farklı algoritmayı çalıştırabilecek şekilde esnek olarak tasarlanmıştır.

Program ilk açıldığı zaman kullanıcıya 5 algoritmadan birini seçmesi için bir menü sunulur (Şekil 3.1). Kullanıcı koşacak algoritmayı seçerek “start” butonuna basar. “start” butonuna basıldıktan sonra başlangıç popülasyonu oluşturularak seçilen algoritma türüne göre kullanıcıya üretilen resimler ve işaretleme kutuları yada çoktan seçmeli kutular gösterilir (Şekil 3.2). Diğer iterasyonlara devam edebilmek için ekranın sol üst köşesinde yer alan “next step” butonu kullanılır. Programı sonlandırmak için ise sağ üst köşeye yerleştirilen “stop” butonu kullanılır. Algoritmanın her iterasyonunda n adet resim ekrana çıkarılır, en son adımda seçilen tek resim ekranda kullanıcıya gösterilir.



Şekil 3.1: Başlangıç Ekranı



Şekil 3.2: Başlangıç Popülasyonu

3.1.2. Algoritma Tasarımları

Proje için 5 farklı algoritma gerçekleştirilmiştir. Her algoritma kullanıcıdan aldığı seçim değerlerine göre bir popülasyon üretecek şekilde çalışır.

Algoritmaların tasarım kısmı kullanıcı arayüzüyle aralarındaki parametre aktarımı yüzünden kullanıcı arayüzüne bağlıdır. Algoritmalar arayüzden kullanıcı seçimlerini alacak, aldığı parametreleri işleyecek ve ürettiği sonucu arayüze verecek şekilde tasarlanmıştır.

Algoritmalar gerçekleştirirken her algortmada ortak olarak kullanılan bir yapı vardır:

Bireyler 17 gerçel sayıdan oluşan bir vektörler olarak temsil edilmişlerdir. 17 parametrelilik vektör belirlenmesi aktif görünüm modeliyle ilintilidir. AAM-API [12] yazılımı işaretlenmiş resimlerden oluşan resim veritabanı girdisiyle, resimleri daha az boyutlu olarak temsil etmek için söz konusu uzayın öz vektörlerini hesaplamaktadır. Daha sonra veritabanında olan veya olmayan resimler bu öz vektörlerin lineer kombinasyonu olarak ifade edilebilmektedirler. Biz projemizde 17 adet öz vektörün yeterli olacağına karar kıldık. Algoritmamızda da hedef resmin bu 17 öz vektörün katsayısını bulmak istediğimizden, bireyler gerçel sayılardan oluşan 17 elemanlık diziler olarak tasarlanmıştır.

Bu gerçel sayılar, yani genler $[-0.3, 0.3]$ aralığında tanımlı olan değerleri alabilmektedirler. Aktif görünüm modelinde, anlamlı resimler oluştuğu takdirde katsayıların da bu aralıkta yığıldıkları tespit edilmiştir. Sınırın genişletilmesi sonuç resimlerin anlamsızlaşmalarına sebep olur. Aralık daraltıldığında, resimler birbirlerine benzemektedirler.

Başlangıç popülasyonu yaratılırken her bir gene ortalaması 0, standart sapması ise 0.10 olan Gauss dağılımından rasgele değerler atanır. Eğer rasgele değer tanım aralığı içine düşmüyorsa, aynalama yapılarak değer sınırlara göre ve tanımlı aralık içinde simetriğinin alınması sağlanmıştır.

Uygunluk değerleri kullanıcı tarafında belirlenir. Uygunluk belirlenmesi algoritmaya göre ya resimler arasından belli bir sayının seçilmesidir ya da resimlerin hedef resme benzerliklerine göre puanlanmalarıdır.

Sonlanma kıstası her algoritmada kullanıcının ekranda gördüğü resimlerden hedef resme benzerliğine göre tatmin olmasıdır. Böyle bir resimle karşılaşıldığı takdirde algoritmaların koşmaları durdurulur.

3.1.2.1. Evrimsel Strateji Algoritmanın Gerçeklenmesi

Toplumdaki birey sayısı ve aynı zamanda ebeveyn sayısı olan μ 4 olarak ve her iterasyonda üretilen çocuk sayısı olan λ da 16 olarak seçilmiştir. Çaprazlama olasılığı olan p_c 0.8 olarak alınmıştır. 4 bireyden oluşan yeni toplumun sadece 16 çocuk arasından seçildiği (μ, λ) stratejisi kullanılmıştır. Kendinden-uyarlanabilir mutasyon parametresi olan τ ise $1/(2n)^{1/2}$ olarak seçilmiştir.

Uygunluk değerleri kullanıcının her iterasyonda gördüğü resimler arasından hedef resme en çok benzediğini düşündüğü 4 resmi seçmesiyle atanır. Bu durumda seçilen resimlerin uygunlukları 1, seçilmeyenlerin 0 olur.

5.1.2.2. Parça Sürüsü Optimizasyonu Algoritmasının Gerçeklenmesi

Popülasyondaki birey sayısı 4 olarak belirlenmiştir. Bu miktar bir iterasyon boyunca sabit kalmamaktadır. Öncelikle kullanıcıya 4 resim gösterilir. Kullanıcı bu 4 resimden sadece bir tane seçmek durumundadır. Daha sonra bu 4 resim sırasıyla alınarak, o resim ve o resimden üretilen ikinci bir resim çifti oluşturmak üzere, 4 adet 2'li resim kullanıcıya peş peşe sunulur. Kullanıcı her defasında bu iki resimden iyi olan bir tanesini seçer.

Kullanıcıya sunulan 4 resimden seçilen en iyi, başlangıç popülasyonu için Parça_Eniyi ve Global_Eniyi olacak şekilde atamalar yapılır. Daha sonra oluşturulan 2'li resimlerden seçilenler, Parça_Eniyi sürüsünü oluşturmaktadır. Tüm çiftler (4 defa resim çifti) kullanıcıya sunulduktan sonra, seçilenler yani Parça_Eniyi sürüsü, 4'lüleri oluşturmaktadır. Kullanıcıya sunulan bu 4'lü resimden seçilen en iyi ise, Global_Eniyi parçasını teşkil eder. Kullanıcı hedef resme yeteri kadar yakın bir resmi elde ettiğine karar verdiğinde, ekrana çıkan 4'lü resimden en iyisini seçer ve STOP düğmesine bastığında, ekranda bu resim yani Global_Eniyi resmi kullanıcıya sunulur.

Kullanıcıya sunulan 4'lü başlangıç popülasyonu sonrası, bu resimler tek tek alınır ve parametreleri kullanılarak, güncelleme fonksiyonları ile yeni bir resim türetilir. 4'lü resimden gelen eski resim ve türetilen yeni resim, 2'li resim çiftlerini oluşturur.

Hız değerinin güncellenmesi belli bir formülü uygulayan, Hız_Güncelle fonksiyonu ile gerçekleşir. Hız_Güncelle fonksiyonu için gerekli parametreler:

w: O anki iterasyon sayısı ve t_max maksimum iterasyon sayısı kullanılarak türetilen bir değer

u1, u2: [0, 1] aralığında üretilen rasgele değerlerdir

c1, c2: 2 olarak seçildi, konu hakkında yazılan kaynaklarda önerilen değer budur

t_max: 2000 olarak seçildi. Büyük seçilmesindeki amaç, otomatik testlerde iterasyon sayısını olabildiğince büyük tutularak, hedef resme daha yakın ve başarılı sonuçlar elde edebilmektir.

Hız_Güncelle fonksiyonu içinde, tüm bu parametreler, Global_Eniyi ve o anki Parça_Eniyi parçasının parametreleri kullanılarak, yeni hız parametre değerleri üretilir.

Pozisyon değerinin güncellenmesi belli bir formülü uygulayan, Pozisyon_Güncelle fonksiyonu ile gerçekleşir. Türetilen yeni parçanın pozisyon parametre değerleri, 4'lüden alınan parçanın hız ve pozisyon değerleri toplanarak, basit bir şekilde güncellenir.

Böylece yeni parçanın, 17 parametre değerinden oluşan hız ve pozisyon değerleri elde edilir.

3.1.2.3. Kuşaksal Genetik Algoritmanın Gerçeklenmesi

Popülasyondaki birey sayısı 8 olarak belirlenmiştir. Bu miktar her iterasyonda sabit kalmaktadır. Etkileşimli genetik algoritmalarda kullanıcı ekranda popülasyondaki 8 bireyi (resmi) görmektedir. Birey sayısının artırılması, etkileşimli algoritmanın kullanıcı tarafından kullanılmasını zorlaştırmıştır.

Etkileşimli kuşaksal genetik algoritmada kullanıcının her iterasyonda ekrandaki resimlere, hedef resme benzerliklerine göre verdiği sıralama puanları, ilgili bireylerin uygunluk değerleridir. Bu değerler popülasyondaki birey sayısı 8 olduğu için 1 ile 8 arasındadır. Uygunluk değeri 8 olan birey, o an için popülasyondaki en iyi bireydir.

2 faktörlü turnuva seçimi operatörü kullanılmıştır. Popülasyondaki bireyler içerisinde 8 çift rasgele olarak seçilmiştir. Her çift içerisinde uygunluk değeri daha yüksek olan birey üreme havuzuna alınmıştır.

Üreme havuzundaki bireyler, havuza geliş sıralarına göre çiftler oluşturularak yeniden birleşme aşamasına geçerler. Bu aşamada çiftler iki-noktalı çaprazlama ve mutasyon operatörlerinden geçerler.

Her çift için iki-noktalı çaprazlama uygulanmıştır. Üniform dağılımın 1 ile 17 aralığından rasgele 2 tamsayı seçilmiştir. İndisleri bu iki nokta arasında kalan katsayılar her çiftin bireyleri arasında değiştirilir. Çaprazlama oranı 0.8 olarak belirlenmiştir.

Her gen değeri için Gauss mutasyonu kullanılmıştır. Buna göre gen değeri 0 ortalamalı 0.15 standart sapmalı Gauss dağılımından rasgele seçilen gerçel sayı ile toplanmıştır. İlk yapılan testlerde, standart sapma 0.10 olarak seçilmişti. Ancak her kuşakta farklılaşmanın biraz daha artırılmasının daha uygun olacağı fark edilerek, standart sapma değeri 0.05 kadar artırıldı. 0.15'in üstündeki sapma değerleri için, resimler anlamsızlaşmaya başladılar. Yeni değer

tanımlı aralığın sınır değerlerini aşarsa aynalama yapılmıştır. Ayrıca mutasyonun oranı 1'dir; yani her gen mutasyona uğratılır.

Etkileşimli kuşaksal genetik algorithmada yeni oluşan bireyler tamamıyla eski bireylerin yerini alırlar. Böylece popülasyonun daha hızlı değişmesi sağlanır. Etkileşimsiz genetik algorithmada ise, seçkincilik ile yer değiştirme kullanılır. Yeniden birleşmenin sonunda çocuk popülasyonun en kötü bireyi yerine anne popülasyonun en iyi bireyi alınır.

3.1.2.4. Kalıcı-durumlu Genetik Algoritmanın Gerçeklenmesi

Popülasyondaki birey sayısı 8 olarak belirlenmiştir. Bu miktar her iterasyonda sabit kalmaktadır. Ancak kullanıcı etkileşimli modda, algoritma sonlanana kadar 3 adet resim değerlendirmektedir.

Etkileşimli kalıcı-durumlu genetik algorithmada kullanıcının her iterasyonda ekrandaki 3 resim içerisinden, hedef resme benzerliklerine göre seçilmeleri, ilgili bireylerin uygunluk değerleridir. Seçilen bireylerin uygunluk değerleri 1, seçilmeyeninse 0 olarak atanır. Kullanıcı her iterasyonda 3 resim arasından iki resim seçer.

Her iterasyonda sadece rasgele iki birey yeniden birleşme aşaması için seçilir. Bu durumda üreme havuzunda tek çift bulunur.

Üreme havuzundaki çift yeniden birleşme aşamasına geçer. Bu aşamada çifte iki-noktalı çaprazlama ve mutasyon operatörleri uygulanır.

Üniform dağılımın 1 ile 17 aralığından rasgele 2 tamsayı seçilir. İndisleri bu iki nokta arasında kalan katsayılar çiftin bireyleri arasında değiştirilir. Çaprazlama oranı 0.8 olarak belirlenmiştir. Çaprazlama sonunda sadece ilk birey çocuk birey olarak belirlenir ve mutasyona tabi tutulur.

Her gen değeri için Gauss mutasyonu kullanılmıştır. Buna göre gen değeri 0 ortalamalı 0.15 standart sapmalı Gauss dağılımından rasgele seçilen gerçel sayı ile toplanmıştır.

3.1.2.5. Diferansiyel Evrim Algoritmasının Gerçeklenmesi

Başlangıç Popülasyonu rasgele üretilerek ekrana 16 birey çıkarılır. Her iterasyonda toplam birey sayısı sabit kalmaktadır. Etkileşimli moda her iterasyonda kullanıcı tarafından seçilen 4 resim değerlendirilmektedir.

Mutasyon da üretilen V verici vektörü ile popülasyonda bulunan X vektörünü ile yeniden birleşirken L adet parametre verici vektörden alınmaktaydı. L rasgele sayısının üretilmesinde kullanılan sınır değeri $p_c = 0.3$ olarak belirlenmiştir.

Mutasyon aşamasında verici vektör (V) yi oluşturmak için kullanılan F mutasyon katsayısı $F = 0.9$ olarak seçilmiştir.

Etkileşimli modda kullanıcı 16 resim içerisinden en iyi olarak düşündüğü 4 resmi seçer ve seçilen bireyler diğer iterasyona katılır.

3.2. Geliştirme

Programın kullanıcı arayüzü ve bu arayüz üzerinde koşacak algoritmalar aralarındaki bağımlılık dikkate alınarak gerçekleştirilmiştir.

Programın kullanıcı arayüzü C++ dili ile Microsoft Foundation Classes (MFC) kullanılarak yazılmıştır. Algoritmalar da C++ dilinde gerçekleştirilmiştir.

Programın başlangıcında kullanıcıdan bir algoritma seçilmesi beklenir. Programda 5 farklı algoritma gerçekleştirildiğinden arayüz ile algoritmaların arasındaki bağımlılığı azaltmak için nesneye yönelik programlama tekniği olan “polymorphism” çok şekillilik kullanılmıştır.

Bütün algoritmalar “*CBaseAlgorithm*” adlı sanal bir sınıftan türetilmiştir. Bu sanal sınıfın amacı arayüzün çalışan algoritmadan bağımsız olmasını sağlamaktır ve bunun için 2 adet metod içerir (Şekil 3.3). Kullanıcı algoritmayı seçtikten sonra seçilen algoritmanın nesnesi oluşturulur ve bir *CBaseAlgorithm* işaretçisi bu nesneye işaret eder. Her algoritma *CBaseAlgorithm* sanal sınıfının içerisinde bulunan metotları tekrar yazmak zorundadır ve böylece arayüz hangi algoritmanın seçildiğinden bağımsız olarak sanal sınıfın metotlarını çağırarak çalışacaktır.

```
class CBaseAlgorithm{
public:
    virtual void Run( int * ) =0; //pure virtual function makes class abstract..
    virtual void Stop( int* ) = 0;
};
```

Şekil 3.3: CBaseAlgorithm Sınıfı

Algoritmalar ile arayüz arasındaki parametre akışı “CParameters” adlı bir sınıf tarafından sağlanmaktadır. Arayüzde kullanıcı “next step” butonuna bastıktan sonra kullanıcının seçmiş olduğu veriler işaretleme kutusu veya çoktan seçmeli kutulardan alınıp çalışan algoritmanın “Run” metoduna aktarılır. Run metodu *CBaseAlgorithm* tarafından ilgili algoritmaya sağlanır ve algoritmalar kendi içerilerinde Run metodunu tekrar yazarlar. Run metodu içerisinde resim üretmek için gerekli parametreler hesaplanır ve *CParameters* sınıfının üyeleri içerisinde bu parametreler konur. *CParameters* sınıfı statik üyelere sahiptir ve arayüz bu sınıfın üyelerini kullanarak parametrelerden resimleri üretir ve kullanıcıya gösterir. Bu sistem parametre akışını arayüz ile *CParameters* arasında kurarak arayüzü algoritmalarından bağımsız hale getirir. Algoritma arayüze değil *CParameters* sınıfına parametreleri aktarır ve arayüz *CParameters* sınıfından bu parametreleri alarak gerekli işlemleri yapar.

```
class CParameters{  
public:  
    static int picture_number; //resim sayisi  
    static double** matrix; //gen degerlerini saklayan matrix  
    static int choice; //choice : 0 seçme , 1 siralama  
};
```

Şekil 3.4 CParameters Sınıfı

4. TEST SONUÇLARI VE YORUMLAR

4.1. DENEYSEL SONUÇLAR

Deneyle, algoritmaları iki farklı açıdan değerlendirme fırsatı sunmaktadır: çözüm kalitesine bağlı olarak algoritmaların başarımlarını analiz etmek ve kullanıcı algoritma tabanlı analiz yapabilmektir. Birinci grup testler için, hedef resimleri AGM'ne göre izdüşüm resimleri oluşturulmuş, bu resimler algoritmaların ürettiği resimlerle karşılaştırılmıştır. İkinci grup testler içinse, tüm algoritmalar bir arayüz aracılığıyla kullanıcılar tarafından oluşturulmuşlardır ve kullanım kolaylıkları açısından yorumlanmışlardır.

Yapılan deneylerde

- Üretilmek üzere 4 hedef yüz seçildi
- Algoritmaları koşturmaları için 7 farklı kullanıcı belirlendi
- Her bir algoritma ve her bir hedef yüz için, 7 kullanıcının ürettiği yüzlerin AGM [11] vektör parametrelerinin ortalamaları alındı ve ortalama sonuç yüzler üretildi
- Bu üretilen ortalama sonuç yüzler, kullanıcı kümesindeki 7 kişiden farklı 10 kişiye gösterildi
- Bu kişilere üretilen yüzleri isimlendirmeleri istendi
- Algoritmaların başarımları, üretilen yüzlerin tanınma oranlarına göre yorumlandı. (Tablo 4.6)

Tüm sistem genel olarak başarılı bir şekilde derlendi ve çalıştırıldı. Parametre ayarları, konuyla ilgili araştırmalardaki öneriler doğrultusunda ve deneysel olarak belirlendi. Üretilen resimler ve hedef resimler, Tablo 4.2, 4.3, 4.4 ve 4.5'te görülebilmektedir. Tablolarda, ilk satır orijinal resmi içermektedir. İlk sütun ise, yüzleri üretmek için kullanılan algoritmaların isimlerini, bu yüzleri üretme aşamasında kullanıcıların yorumladığı ortalama resim sayısını ve tanımları için sorulan kişilerin, bu resimleri tanıma oranlarını temsil edilmektedir. İkinci sütunda, her bir yaklaşım için üretilen resimler yer almaktadır. Tablo 4.2, 4.3 ve 4.4'te yer alan hedef yüzler, deneme aşamasında veritabanında mevcut olan ve yüzlerin AGM [11] parametre vektörlerini üretmek için kullanılan yüzler arasından seçilmiştir. Tablo 4.5 teki hedef resim ise, veritabanında bulunmayan bir yüzüdür.

4.1.1. ÇÖZÜM KALİTESİNE BAĞLI BAŞARIM ÖLÇÜMLERİ

AAM-API [12] programını kullanarak hedef resimlerin elde ettiğimiz modele izdüşümü olan resimleri oluşturduk. Bu resimler hazırladığımız resim veritabanı ile elde edilebilecek en iyi resimlerdir. Tablo 4.1'te hedef resimler ve bu resimlerin model izdüşümleri görülmektedir. Elde edilen model resimler etkileşimli algoritma sonucu elde ettiğimiz ortalama resimlerle karşılaştırıldığında (Tablo 4.2, 4.3, 4.4, 4.5), sonuçların birbirlerini ne kadar yakın olduğu görülmektedir. Sonuç olarak elde edilen resim veritabanı ile elde edilecek en iyi sonuçlara yakın sonuçlar etkileşimli doğa esinlemeli algoritmalar aracılığıyla uygulamamızda da elde edilmişlerdir.

Tablo-4.1: Hedef resimlerin izdüşümleri

Hedef Resim	Model Resim
	
	
	
	

AGM ile izdüşümü oluşturulan model resimlerin ilgili hedef resimlerle aralarındaki farkın nedeni resim veritabanımızda 51 tane resim olmasıdır. Resim miktarı arttırıldığı takdirde, oluşturulacak olan yeni model ile hedef resimlerin izdüşümleri o resimlere daha çok benzeyeceklerdir.







4.1.2. ETKİLEŞİMLİ ALGORİTMALARIN BAŞARIM ANALİZİ

Bu kısımdaki deneyler, etkileşimli algoritmaların başarımların analizini, gerçekleştirebilmek amacıyla gütmektedir. Her bir yaklaşımın (algoritma) başarımlarını karşılaştırabilmek için, veri tabanında olan ya da olmayan bir hedef resim belirlenir. Daha sonra, belli sayıda kişiden, her bir hedef resim için algoritmaları koşturmaları istenir. Her bir algoritma ve hedef resim için, test yapan kişilerce üretilmiş olan yüzlerin AGM [11] parametre vektörlerinin ortalaması alınır. Daha sonra, üretilen ve ortalaması alınan bu yüzler, test yapmakta kullanılan kişiler dışında, belli sayıda kişiye gösterilir. Bu kişilere, üretilen yüzlerin kime ait olduğu sorularak, isimlendirmeleri beklenir. Algoritmaların başarımları, bu yüzlerin tanınma oranlarına bağlı olarak değerlendirilir. Etkileşimli modda, algoritmaların başarımlar ve kullanılabilirliğinin ölçümlerini yapacak ikinci bir kısımda, kullanıcıların gösterilen resimleri yorumlama yöntemi ve yorumladıkları resim miktarlarıdır. Örneğin, bir kullanıcı için, çok sayıda resim içinden, belli sayıda bir alt küme seçmek, resimleri benzerliklerine göre puanlamak ya da sıralamaktan çok daha kolaydır. Etkileşimli modda, algoritmaların kullandığı teknik ve yaklaşımlar da, elbette büyük önem taşımaktadır. Kullanıcıların yorumladığı resim sayısı







arttıkça, doğal olarak kullanıcı yorulacak, sıkılacak ve gösterilen resimleri yorumlarken gerekli ehemmiyeti göstermeyecektir. Sonuç olarak, algoritmaların başarımlarını değerlendirmesi iki kıstasa bağlı olarak gerçekleştirilmektedir: kullanıcıların ürettiği sonuç yüzlerin, hedef yüze olan bezerlik oranı ve hedef yüzü bulmak için yorumladıkları toplam resim sayısıdır.

Tablo 4.6, algoritmaların her iki değerlendirme kriterine göre kıyaslama imkanı sunmaktadır: üretilen yüzlerin tanınma oranları ve üretilene dek değerlendirilen resim sayılarının ortalamaları. Tablo 4.6’te de görüldüğü üzere, ortalama tanınma oranları göz önünde bulundurulacak olursa, testlerde EKDGA en başarılı yaklaşım olmaktadır. Kullanıcının yorumladığı ortalama resim sayısı göz önünde bulundurulacak olursa, EKDGA en başarısız yöntem olmaktadır, EDE en başarılı yaklaşım olarak gözükmektedir. Ancak, kullanıcı açısından arayüzün de önem taşıdığı varsayımıyla, EKGA için kullanıcıların tüm resimleri puanlaması gerekmektedir, EES ve EDE için, çok sayıda resim içinden belli sayıda bir alt küme seçilmesi kullanıcı açısından söz konusudur. EKDGA ve EPSO içinse, puanlama ve sıralama olmamakla birlikte, gösterilen resimler içinden, uygun olanını seçme işi, kullanıcı açısından büyük titizlik ve dikkat gerektirmektedir. Çünkü kullanıcıya sunulan resim sayısı azdır.







Tablo-4.2: Hedef Resim 1 Sonuçları

Hedef Resim	
EPSO (64.8) 100%	
EDE (54.4) 100%	
EES (65.6) 100%	
EKGA (47.2) 100%	
EKDGA (76.5) 100%	







Tablo-4.3 Hedef Resim 2 Sonuçları

Hedef Resim	
EPSO (72) 100%	
EDE (56) 70%	
EES (62.4) 90%	
EKGA (48) 80%	
EKDGA (64.2) 90%	

Tablo-4.4 Hedef Resim 3 Sonuçları

Hedef Resim	
EPSO (52.8) 90%	
EDE (52.8) 80%	
EES (62.4) 100%	
EKGA (48.8) 90%	
EKDGA (66.3) 90%	

Tablo-4.5 Hedef Resim 4 Sonuçları

Hedef Resim	
EPSO (61.2) 50%	
EDE (52.8) 100%	
EES (46.4) 100%	
EKGA (76) 100%	
EKDGA (63.6) 100%	

Tablo-4.6 Genel Sonuçlar

Algoritmalar	EPSO	EDE	EES	EKGA	EKDGA
Yukarıdaki 4 resmi bulmak için değerlendirilen ortalama resim sayısı	62.7	54	59.2	55	67.65
Yukarıdaki 4 resmin tanınma oranı	85%	87.5%	97.5%	92.5%	97.5%

4.2. SONUÇ, GELİŞTİRME OLANAKLARI VE KULLANIM ALANLARI

Bu çalışma bir ilk adım olarak değerlendirilebilir. Bu nedenle, genel bir başarımlar analizi ve her bir yaklaşımın kullanılabilirliği hakkında fikir vermektedir. Etkileşimli halinin daha iyi anlaşılabilirliği ve çözümlenebilirliği için, etkileşimli olmayan modda çok sayıda test yapılmalı ve başarımları analiz edilmelidir. Parametre ayarlamaları, doğal esinlenme yönteminde çok kritik bir rol üstlenmektedir. Bu çalışma için tüm parametreler, konuyla ilgili araştırmalarda önerilen değerler baz alınarak ayarlanmıştır. Bununla birlikte, deneysel çalışmalar başarımları yükseltecek optimum ayarlamalar üzerine kurulmuştur. Çalışmanın başarımlarını etkileyecek bir diğer faktörse, test veri tabanında AGM [11] parametre vektörlerini üretmek için kullanılan resimlerin sayısı artmalıdır. Ayrıca, sistemde başarımları arttıracak bazı yöntemlere gidilmelidir (E-FIT [1] ve Eigen-FIT [2] gibi). Bu yöntemlerle yüze, büyüklüğünün veya şeklinin değiştirilmesi, aksesuar eklenmesi, yaşlandırılması, bıyık veya sakal eklenmesi, saç modelleri eklenmesi vb. özellikler eklenerek başarımlar artırılabilir.

Bu çalışmada, başlangıç aşaması için öngörülen hedeflere başarıyla ulaşılmıştır. Elde edilen sonuçlar umut vermeye birlikte, daha önce de bahsedildiği üzere deney kapsamı genişletilmelidir. Buna rağmen, deney sonuçları sistemin kullanışlı olduğunu kanıtlamaya yeterlidir ve çalışmanın geliştirilmesi konusunda ikna edicidir. Doğal sezgisel yöntemler, etkileşimli kullanımlarıyla, günlük yaşamda da başarıyla kullanılabilir. Çalışmamız piyasadaki önemli bir boşluğu doldurabilecek ticari değere sahiptir. Bu çalışma, özellikle polis merkezlerinde suçlunun mağdur tarafından tanınmasında kolaylıkla kullanılabilir. Çünkü yüzün tüm detaylarının, bir ressam tarafından çizilmeye çalışılması mağdur taraf ve ressam için oldukça yorucu bir işlemdir. Ayrıca çizilen resim, üretilen resim kadar hedef resme yakın ve başarılı değildir. Ayrıca bilgisayar destekli programlar yüz parçalarının bulunduğu bir veritabanından parça seçme esasına dayanırlar. Ancak insanlar yüzün parçalarına odaklanmaktan çok yüzü bir bütün olarak ele alırlar. Bu da bu tip programların kullanımlarının iyi sonuçlar vermemelerine sebep olmaktadır. Biz projemizde kullanıcıya belli bir miktar yüz resmi sunarak, ondan resimleri hedefe benzerliklerine göre değerlendirmesi istiyoruz. Bu da kullanıcının programı kendi başına, bir operatörden yardım almadan kullanmasına olanak verir. Kullanıcı sözle ifade edemediklerini, resimleri birbirleriyle karşılaştırarak gösterir.

Sonuç olarak otomatik yüz üretme programlarının sayılarının azlığı hazırladığımız projeyi önemli bir hale getirmektedir. İyileştirmeler tamamladığı takdirde projemiz piyasada ticari getirisi olan ve gerçek problemlere uygulanabilir duruma gelecektir.

5. REFERANSLAR

- [1] “E-FIT: Electronic Facial Identification Software for Police Departments”, New England Press Inc., 2004, <http://www.effitforwindows.com>
- [2] “PROfit: The Professional Photo-fit System of Choice”, ABM United Kingdom Ltd., <http://www.abm-uk.com/uk/products/profit.asp>
- [3] Frowd C.D., Hancock P.J.B., “EvoFIT: Facial Composite System for Identifying Suspects to Crime”, Department of Psychology, Sterling University, <http://www.evoFIT.co.uk>
- [4] “EigenFIT: Facial Composite Technology”, VisionMetric Ltd., <http://www.eigenfit.com>
- [5] Eiben A.E., Smith J.E., Introduction to Evolutionary Computing, Springer, 2003.
- [6] “PCA - principal Component Analysis”, Datafusion, <http://www.eng.man.ac.uk/mech/merg/Research/datafusion.org.uk/pca.html>
- [7] Cootes T. F., Taylor C.J, Statistical Models of Appearance for Computer Vision, 2004
- [8] “Genetic Algorithms”
http://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/tcw2/report.html
- [9] “A Brief History of GA Optimization”, Optimatics, <http://www.optimatics.com/GAhistory.htm>
- [10] Akbal T., Demir G. N., Kanlikilicer E., Kus M. C., Ulu F. H., “Interactive Nature-Inspired Heuristics for Automatic Facial Composite Generation”, Genetic and Evolutionary Computaton Conference Undergraduate Student Workshop, 2006.
- [11] Iain Matthews, Simon Baker, “Active Appearance Models Revisited,” International Journal of Computer Vision, Vol. 60, No. 2, pp. 135-164, November 2004 .
- [12] “AAM-API”, <http://www2.imm.dtu.dk/~aam/aamapi/>
- [13] “PROfit: A Photofit System using Highly Advanced Facial Composition Tools”, ABM United Kingdom Ltd., <http://www.abm-uk.com/uk/pdfs/profit.pdf>
- [14] Frowd C.D., Hancock P.J.B., Carson D., “EvoFIT: A Holistic, Evolutionary Facial Imaging Technique for Creating Composites”, *ACM TAP*, Vol. 1 (1), pp. 1-21, 2004.
- [15] Gibson S. J., Pallares-Bejarano A., Solomon C. J., "Synthesis of Photographic Quality Facial Composites using Evolutionary Algorithms", in Proceedings of the British Machine Vision Conference, Editors R. Harvey and J.A. Bangham, pp221-230, 2003.
- [16] Solomon C. J., Gibson S. J., Pallares-Bejarano A., "EigenFit - The Generation of Photographic Quality Facial Composites", *The Journal Of Forensic Science*, 2005.
- [17] Storn R., Price K., "Differential Evolution - A Simple and Efficient Adaptive Scheme for Global Optimization over Continuous Spaces", Technical Report TR-95-012, International Computer Science Institute, Berkeley, CA, 1995.
- [18] Beyer H.G.; Schwefel H.P., “Evolution strategies *A comprehensive introduction*”, *Natural Comp.* 1: pp. 3–52, 2002.
- [19] Eberhart R. C., Kennedy J., Shi Y., *Swarm Intelligence*, 1st Ed., Morgan Kaufmann, 2001.
- [20] Pomeroy P., “An Introduction to Particle Swarm Optimization“, <http://www.adaptiveview.com/articles/ipsoprnt.html>, 2003.
- [21] Madar J., Abonyi J., Szeifert F., "Interactive Particle Swarm Optimization", 5th International Conference on Intelligent Systems Design and Applications (ISDA'05), pp. 314-319, 2005.
- [22] Breukelaar R., Emmerich M., Back T., “On Interactive Evolution Strategies”, *EvoWorkshops 2006*, LNCS 3907, pp. 530–541, 2006.
- [23] Beyer H.G.; Schwefel H.P., “Evolution strategies *A comprehensive introduction*”, *Natural Comp.* 1: pp. 3–52, 2002.

[24] Eiben A.E., Schoenauer M., “Evolutionary Computing”, Information Processing Letters, 82(1): 1–6, 2002.