

T.C.
SAKARYA ÜNİVERSİTESİ
BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ

BSM 498 BİTİRME ÇALIŞMASI

2B FOTOĞRAF ÜZERİNDE İNSAN YÜZÜNÜN TESPİTİ

B1412.10301 – İsmet SANDIKÇI
B1412.10305 – Barış KAYA
B1412.10307 – M. Olcay TERZİOĞLU

Fakülte Anabilim Dalı : BİLGİSAYAR MÜHENDİSLİĞİ
Tez Danışmanı : Arş. Gör. Dr. Gülizar ÇİT

2017-2018 Bahar Dönemi

T.C.
SAKARYA ÜNİVERSİTESİ
BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ

2B FOTOĞRAF ÜZERİNDE İNSAN YÜZÜNÜN TESPİTİ

BSM 498 - BİTİRME ÇALIŞMASI

Adı SOYADI
İsmet SANDIKÇI
Barış KAYA
M. Olcay TERZİOĞLU

Fakülte Anabilim Dalı : BİLGİSAYAR MÜHENDİSLİĞİ

Bu tez .. / .. / ... tarihinde aşağıdaki jüri tarafından oybirliği / oyçokluğu ile kabul edilmiştir.

.....
Jüri Başkanı

.....
Üye

.....
Üye

ÖNSÖZ

Eğitim hayatımızda her zaman destek olan ailemize, yakın arkadaşlarımıza, bitirme çalışmamızda desteğini esirgemeyen danışman hocamız Arş. Gör. Dr. Gülizar ÇİT ‘e teşekkürlerimizi borç biliriz.

İÇİNDEKİLER

ÖNSÖZ.....	iii
İÇİNDEKİLER.....	iv
ŞEKİLLER LİSTESİ.....	vi
TABLolar LİSTESİ.....	vii
ÖZET.....	viii
BÖLÜM 1.	
GİRİŞ.....	1
BÖLÜM 2.	
ÖN İŞLEME.....	2
BÖLÜM 3.	
KARINCA KOLONİ ALGORİTMASI.....	4
3.1. Karınca Koloni Algoritması Nedir?.....	4
3.2. Feromon.....	4
3.3. Karınca Koloni Algoritması Adımları.....	5
3.4. Karıncaların Sezgisel ve Feromon Davranışları.....	7
3.5. Karıncaların Sezgisel ve Feromon Davranışlarının Algoritmadaki Yeri.....	8
3.6. Karınca Tur Kuralları.....	9
3.7. Feromon Güncellemesi.....	10
3.7.1. Local feromon güncellemesi.....	10
3.7.2. Global feromon güncellemesi.....	11

BÖLÜM 4.

KENAR BELİRLEME.....	12
4.1. Kenar Belirleme Nedir?	12
4.2. Kenar Belirleme Yöntemlerinde En Çok Kullanılan Algoritmalara Ait Filtreler.....	12
4.2.1. Kullandığımız kenar belirleme tekniği.....	12
4.2.2. Sobel filtresi.....	14
4.2.3. Prewitt filtresi.....	14
4.2.4. Canny filtresi.....	15
4.3. Karınca Koloni Algoritması İle Kenar Belirleme Örnekleri.....	16

BÖLÜM 5.

UYGULAMA.....	17
---------------	----

BÖLÜM 6.

SONUÇLAR VE ÖNERİLER.....	22
---------------------------	----

KAYNAKLAR.....	23
----------------	----

ÖZGEÇMİŞ.....	24
---------------	----

BSM 498 BİTİRME ÇALIŞMASI DEĞERLENDİRME VE SÖZLÜ SINAV

TUTANAĞI.....	25
---------------	----

ŞEKİLLER LİSTESİ

Şekil 2.1.	Piksel değeri hesaplanırken kullanılan 16 adet komşu.....	2
Şekil 2.2.	Orijinal resim ve ön işleme uygulanmış resim.....	3
Şekil 2.3.	Orijinal resim ve ön işleme uygulanmış resim.....	3
Şekil 3.1.	Karınca koloni algoritması sözde kodu.....	5
Şekil 3.2.	Karıncaların feromon sıvı miktarı ile en kısa yol üzerinden yiyeceklere ulaşabilmesi.....	6
Şekil 3.3.	Karıncaların en kısa yol ile hedefe ulaşmaları.....	7
Şekil 4.1.	Karınca koloni algoritması ile elde edilmiş resim kenarları.	16

TABLÖLAR LİSTESİ

Tablo 5.1.	Örnek karınca uygulaması 1.....	19
Tablo 5.2.	Örnek karınca uygulaması 2.....	20
Tablo 5.3.	Örnek karınca uygulaması 3.....	21

ÖZET

Anahtar kelimeler: Karınca Kolonisi, Kenar Belirleme, Resim Eşikleme

Bu çalışmada, iki boyutlu fotoğraf üzerindeki insan yüzünün görüntü işleme algoritmaları ile tanınarak yüz hatlarının otomatik olarak tespit edilmesini sağlayan bir sistem geliştirilmiştir.

Geliştirilen çalışmada, fotoğraf ilk önce siyah beyaz forma dönüştürülüp piksel değerleri alınmaktadır. Bu piksel değerleri üzerinden fotoğrafın deseni oluşturulmaktadır. Oluşturulan bu desen ile insan yüzünün hatlarının kenarları belirlenmektedir.

BÖLÜM 1. GİRİŞ

Görüntü işleme, ölçülmüş veya kaydedilmiş olan elektronik görüntü verilerini, elektronik ortamda (bilgisayar ve yazılımlar yardımı ile) amaca uygun şekilde değiştirmeye yönelik yapılan bilgisayar çalışmasıdır. Görüntü işleme, kaydedilmiş olan mevcut görüntüleri işlemek yani mevcut resim ve grafikleri değiştirmek, yabancılaştırmak ya da iyileştirmek için kullanılır.

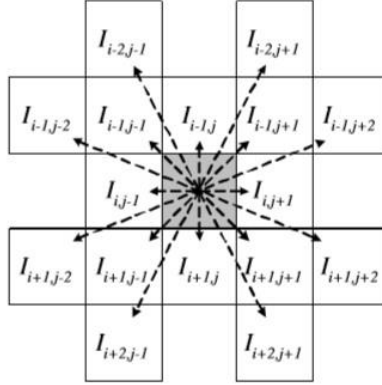
Geliştirilen çalışmada görüntü işlemenin kullanım alanlarından birisi olan insan yüzü ele alınmıştır. Arka planları beyaz olarak çekilmiş olan 300x200 piksel boyutlarındaki fotoğraflar kullanılarak insan yüzünde kenar belirleme uygulanmıştır.

Bu çalışmada, fotoğraf ilk önce siyah beyaz forma dönüştürülüp piksel değerleri alınmaktadır. Bu piksel değerleri üzerinden Ön İşleme yöntemiyle fotoğrafın deseni oluşturulmaktadır (Bölüm 2). Oluşturulan bu desene karınca kolonisi optimizasyonu uygulanarak kenar değerleri bulunmaktadır(Bölüm 3). Kenar değerleri üzerinde ‘Eşikleme’ işlemi yapılarak insan yüzünün kenarları belirlenmektedir. (Bölüm 4)

BÖLÜM 2. ÖN İŞLEME

Ön işlemede 300x200 boyutlarındaki resimlerin her bir pikselinin yeni değeri komşusu olan 16 adet piksel değeri kullanılarak hesaplanıp yeni değerleri atanmaktadır.

Komşu pikseller aşağıdaki resme göre belirlenmektedir.



Şekil 2.1. Piksel değeri hesaplanırken kullanılan 16 adet komşu

Yukarıdaki şekilde komşuları belirlenmiş olan piksellerin yeni değerleri hesaplanırken birbirlerine simetrik olan piksellerin mutlak değerce farkları alınarak yeni değerleri oluşturulur. Bu farklar toplanarak merkez pikselin yeni değeri hesaplanmaktadır. Bu işlem denklem 2.1’de gösterilmektedir. [1]

$$\begin{aligned}
 V_c(I_{i,j}) = & f(|I_{i-2,j-1} - I_{i+2,j+1}| + |I_{i-2,j+1} - I_{i+2,j-1}| + |I_{i-1,j-2} - I_{i+1,j+2}| + \\
 & |I_{i-1,j-1} - I_{i+1,j+1}| + |I_{i-1,j} - I_{i+1,j}| + |I_{i-1,j+1} - I_{i+1,j-1}| + \\
 & |I_{i-1,j+2} - I_{i+1,j-2}| + |I_{i,j-1} - I_{i,j+1}|)
 \end{aligned} \quad (2.1)$$

Fakat bu formül kendi başına yeterli olmamaktadır. Bir sonraki adımda uygulayacağımız Karınca Kolonisi Algoritması’nda sezgisel matris değerleri hesaplanırken bir lambda(λ) değeri kullanılması gerekmektedir.

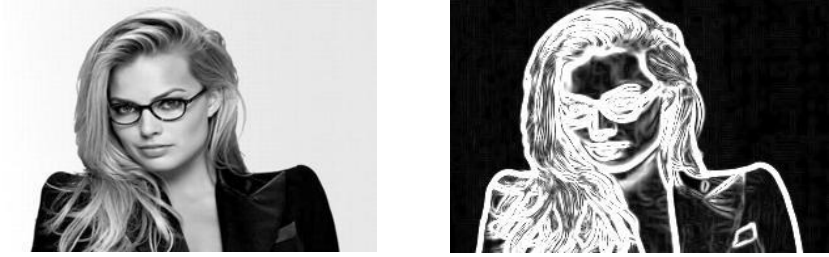
Lambda değerleri denklem 2.1’de aşağıdaki gibi kullanılmaktadır; [2]

$$f(x) = \lambda x, \quad \text{for } x \geq 0, \quad (2.2)$$

$$f(x) = \lambda x^2, \quad \text{for } x \geq 0, \quad (2.3)$$

$$f(x) = \begin{cases} \sin\left(\frac{\pi x}{2\lambda}\right), & 0 \leq x \leq \lambda; \\ 0, & \text{else.} \end{cases} \quad (2.4)$$

$$f(x) = \begin{cases} \frac{\pi x \sin\left(\frac{\pi x}{\lambda}\right)}{\lambda}, & 0 \leq x \leq \lambda; \\ 0, & \text{else.} \end{cases} \quad (2.5)$$



Şekil 2.2. Orijinal resim ve ön işleme uygulanmış resim



Şekil 2.3. Orijinal resim ve ön işleme uygulanmış resim

BÖLÜM 3. KARINCA KOLONİ ALGORİTMASI

3.1. Karınca Koloni Algoritması Nedir?

Karınca Kolonisi Algoritması (Ant Colony Algorithm - ACA), 1991 yılında Marco Dorigo tarafından tasarlanmış bir algoritmadır.

Algoritmanın genel bakış açısı; gerçek karınca kolonilerinde yasamlarını sürdüren karıncaların kendi sistemlerini sürdürebilmeleri için yiyeceklerini araştırıp bulmaları gerekmektedir. Bunun için yiyecekleri ile yuvaları arasındaki mesafeyi her zaman için en kısa sürede katetmeleri gerekmektedir. Bunun içinde yiyecekleri ve yuvaları arasındaki en kısa yolu seçmek zorundadırlar ve zamanla bu sezgisel davranışlarını bir kabiliyet haline getirmişlerdir. Bu kabiliyet sayesinde zaman içerisinde sürekli kullandıkları en kısa yolun çevresinde fiziksel, kimyasal veya çevresel herhangi bir değişim olduğunda, en kısa yol artık onlar için iyi bir tercih olmayabilir. Bunun içinde yeni en kısa yollar bulmaya başlarlar.

Tüm bu sezgisel davranışlarının yanı sıra, karıncaların önemli olan diğer bir özellikleri ise çok iyi bir görme kabiliyetine sahip olmamalarıdır. Bu en kısa yolu seçmek için tabikide etrafı tam olarak görebilecekleri anlamına gelmiyor. Tam olarak kör de değillerdir. Bu sebepten, karıncalar birbirleri ile haberleşebilmeleri için kendileri tarafından ürettikleri bir kimyasal maddeyi kullanmaktadırlar.[3]

3.2. Feromon

Karıncalar kendi aralarındaki iletişimi sağlayabilmeleri için Feromon isimli bir kimyasal madde kullanmaktadırlar. Karıncalar yiyecek buldukları hedeflerine yada yuvalarına ulaşabilecekleri yolları katederken, yolları üzerine kendilerinin salgılamış olduğu bir miktar Feromon kimyasal sıvısını veya kokusunu bırakırlar. Karıncalar yönlerini tespit ederken feromon sıvısının miktarına önem vermektedirler. Eğer ki

ulaşılabacak hedefe ait tüm yönlerin feromon sıvı miktarları birbirine eşit ise, tüm karıncaların bu yönleri seçebilme ihtimalleri yani olasılıkları aynı olacaktır.

Karıncaların hepsinin aynı hızlara veya bırakmış oldukları sıvı miktarlarının aynı olması, daha kısa yollar birim zamanda daha çok feromon maddesi alacağını belirler. Bu sebepten, karıncalar hızlı bir şekilde en kısa yolu bulurlar. Karıncaların tüm bu uğraşları tamamen doğal bir optimizasyon işlemine örnek olabilecek seviyede bir davranıştır. Tüm bu optimizasyon işlemleri içinde bir karınca algoritması oluşturulmuştur.

Karınca algoritmaları genetik algoritma gibi bir popülasyon sistemini yaklaşım olarak benimsemiştir. Karınca popülasyonu içerisinde bulunan tüm karıncalar bir çözüm yolunu temsil etmektedir. Tüm bu çözüm yolları sayesinde popülasyon içerisindeki tüm karıncaların hareketlerini belirlemelerinde referans olabilmektedirler.

Algoritma, karınca kolonilerinden esinlenerek tasarlandığından oluşan sisteme; karınca sistemi (KS), algoritmaya ise karınca kolonileri algoritması (KKA) ismi verilmektedir.

```

BEGIN
  REPEAT
    Bütün yapay karıncalar için yolların üretilmesi
    Bütün yapay yolların uzunluğunun hesaplanması
    Yapay yollar üzerinde bulunan feromon maddesi miktarının güncellenmesi
    Şu ana kadar bulunan en kısa yapay yolun hafızada tutulması
  UNTIL ( iterasyon = maksimum iterasyon yada yeterlilik kriteri )
END.

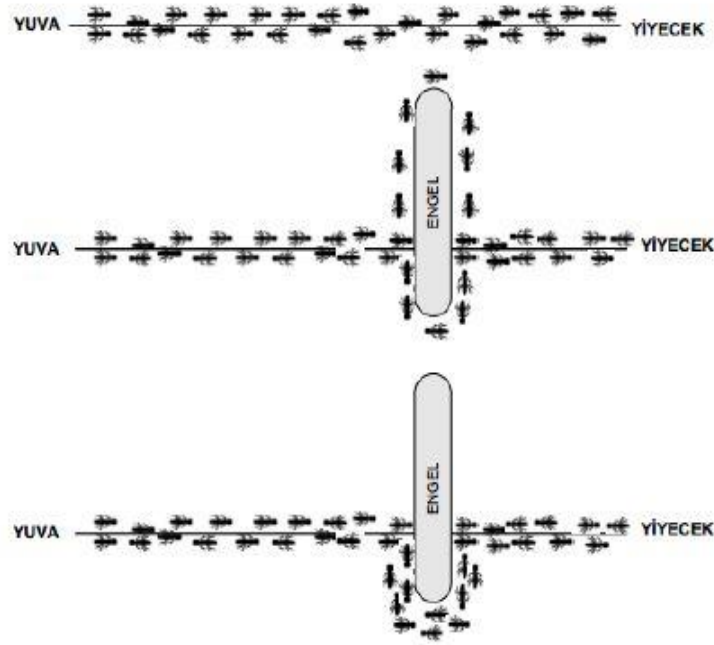
```

Şekil 3.1. Karınca koloni algoritması sözde kodu

3.3. Karınca Koloni Algoritması Adımları

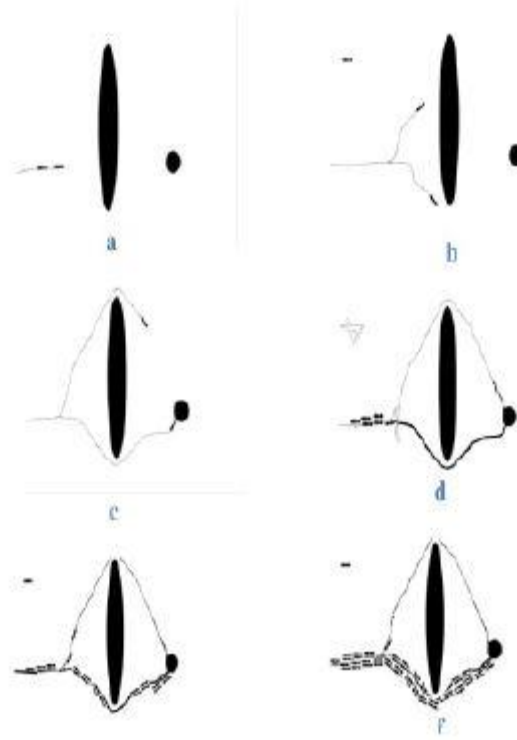
- I. Karıncaların yönlerini bulabilmesi için kullandıkları feromon sıvılarına ait başlangıç feromon sıvı değerleri belirlenir.
- II. Karıncalar her farklı noktaya rastgele yerleştirilir.

- III. Karıncalar, sonraki hedeflerine olasılık denklemlerine bağlı olacak şekilde turlarını tamamlarlar.
- IV. Karıncaların katettikleri yollar ve buna ait olan feromon sıvı miktarları hesaplanır, sonrasında yeni lokal feromon sıvı miktarları oluşturularak ilgili bilgi güncellenir.
- V. En iyi çözüme ait yol hesaplanır ve global feromon güncellenmesinde kullanılır.
- VI. İterasyon sayıları tamamlandıktan sonra II. Adım'a gidilir.



Şekil 3.2. Karıncaların feromon sıvı miktarı ile en kısa yol üzerinden yiyeceklere ulaşabilmesi

3.4. Karıncaların Sezgisel ve Feromon Davranışları



Şekil 3.3. Karıncaların en kısa yol ile hedefe ulaşmaları

Karıncalar yaşamlarını sürdürebilmeleri için yiyeceklerini çevreden araştırmaları gerekir. Bunun için öncelikle, karınca kolonisinden öncü karıncalar araştırma yapmak için yuvalarından çıkarlar.

Öncü karıncalar çevrede sezgisel olarak bulmuş oldukları herhangi bir yiyecek kaynağının konumunu tespit edip hafızalarında bunu hedef olarak belirlerler. Sonrasında öncü karıncalar hafızalarındaki bu bilgi ile yuvalarına geri dönerler.

Geri dönüş esnasında vücutlarından salgıladıkları kimyasal sıvı olan feromon yardımıyla, geçtikleri yollara koku veya sıvı olacak şekilde izler bırakırlar. Öncü karıncalar yuvalarına vardıklarında karınca kolonisindeki diğer karıncalara hedef hakkında hafızasındaki bilgiyi aktarır ve popülasyondaki belirli sayıdaki karıncalar bilgiyi alarak hedefe varmak için yola çıkarlar. Karıncalar hedefe ulaşana kadar öncü karıncaların yollara önceden bırakmış oldukları izleri referans alarak rastsal bir şekilde ilerlerler. Böylelikle iz bulunan tüm yollarda farklı sayılarda karıncalar

bulunmaktadır. Bu yollardan kısa yolları tercih eden karıncalar hedefle yuva arasında daha sık bir şekilde gidip gelecekleri için artık o yoldaki feromon kokusu yada sıvısının miktarı daha fazla olacaktır. Az kullanılmaya başlanan yollardaki feromon sıvı yada kokusu zamanla buharlaşmaya başlayacaktır.

Daha önceden kısa yolu seçmeyen ve diğer yollara rastsal bir şekilde dağılan karıncalar yuvalarından ayrılırken yol üzerindeki daha yoğun olan feromon kokusuna yada sıvısına yönelme olasılıkları çok daha fazla artacaktır. Zaman içerisinde bu kısa yolu tercih eden diğer tüm karıncaların sayıları artacaktır. Böylelikle karıncalar yuvaları ve hedefleri arasında düzenli bir köprü kurmuş olacaklardır.

3.5. Karıncaların Sezgisel ve Feromon Davranışlarının Algoritmadaki Yeri

Karıncaların bir problem üzerindeki çözümlerinin yaklaşımında kullanacakları yöntemleri öncesinde sezgisel sonrasında ise feromon kimyasal maddesi olarak sınıflandırabiliriz. Bunun içinde algoritma içerisinde adım adım ilerleyebilmemiz için belirlenen karıncaların sayıları ile birlikte matrisler oluşturuyoruz. Bu matrisleri sezgisel yaklaşımları ifade edebilmesi sezgisel matris, feromon kimyasalini ifade edebilmesi içinde feromon matrisi olarak isimlendiriyoruz. Bu matrisler olasılık formüllerinde kullanılarak iyi sonuçlara ulaşmamızı sağlayacaktır.

Feromon matrislerinin olasılık formüllerindeki önemi; karıncaların hedefle yuva arasındaki seçmiş oldukları yollarda bulunan feromon sıvı miktarlarının düzeyi yer almaktadır.

Sezgisel matrislerinin olasılık formüllerindeki önemi; karıncaların vardıkları bir düğümden varacakları başka bir düğüme kadar olan mesafenin uzaklığıyla ters orantılı bir başlangıç değeridir.

Algoritma çözümünde kullanılacak karınca sayısının artırılması çözüm için iyi sonuçlar vereceği gibi bu matrislere değerlerin atanması ve olasılığın çok fazla artması ile çözüme ulaşmak için tahmin edilen sürenin çok fazla artmasına sebebiyet

vereceğinden zaman gittikçe artacaktır. Karıncaların sayısı, çözülecek problemlerin büyüklüğüne ve algoritmanın kullanılacağı alana bakılarak veriler değiştirilebilir.

3.6. Karınca Tur Kuralları

Karınca Koloni Algoritmasında bir tur esnasında, i noktasında yer alan k karıncası, j noktasına yönelirken iki alternatif yol seçmek zorundadır. Bu seçimde Feromon değerinin en yüksek olmasına dikkat edecektir. Genellikle bu seçim ilk alternatifte olasılık değeri $q_0 = \%90$ seviyesinde belirlenmektedir. İkinci alternatifte ise, olasılık dağılımına bağlı olarak yollar seçilecektir. Bu tur esnasında i noktasındaki k karıncasının u adet alternatif yoluna ait çözüm yapılacak formül şöyledir; (Denklem 3.1)

$$j = \max_{u \in J_k(i)} \{[\tau(i, u)]^\alpha \times [\eta(i, u)]^\beta\} \quad \text{eğer } q \leq q_0 \quad (3.1)$$

Burada ilk parantez içerisindeki değerler feromon izleridir. İkinci parantez arasındaki değerler ise i noktasından u noktasına ait uzaklığın tersi bulunmaktadır. $J_k(i)$ ise, i noktasında bulunan k karıncasının henüz gitmediği noktaları temsil eder. β ($\beta > 0$) feromon güncellemesinde, uzaklığın göreceli önemliliğini belirleyen bir değerdir. q_0 ($0 < q_0 < 1$) çözüm uzayını belirleyen bir değerdir.

$q \leq q_0$ durumu gerçekleştiğinde, gidilmesi gereken diğer hedef seçim değerlerine bağlı olarak rastsal bir şekilde seçilmektedir. Böylelikle, feromon miktarı yoğun olan yolların seçilme olasılığı daha fazla olacaktır.

Gidilecek olan yolların seçilme olasılığına ait formül ise şöyledir; (Denklem 3.2)

$$P_k(i, j) = \begin{cases} \frac{[\tau(i, j)]^\alpha \times [\eta(i, j)]^\beta}{\sum_{u \in J_k(i)} [\tau(i, u)]^\alpha \times [\eta(i, u)]^\beta} & \text{eğer } j \in J_k(i) \\ 0 & \text{Diğer durumlarda} \end{cases} \quad (3.2)$$

$P_k(i, j)$: k karıncasının i noktasından j noktasına geçme olasılığı

$r(i, j)$: i ve j noktaları arasındaki feromon matris değeri

$n(i, j)$: i ve j noktaları arasındaki sezgisel matris değeri

α : feromon katsayısı

β : sezgisel katsayısı

J_k : yollara ait noktaların tamamıdır.

3.7. Feromon Güncellemesi

Popülasyondaki karıncaların tamamı yuva - hedef arasındaki turlarını tamamladıktan sonra feromon sıvı miktarları zaman içerisinde yenilenmekte yani güncellenmektedir. İlk işlem, noktaların etrafındaki tüm yollarda feromonlar belirli oranlarda sırasıyla buharlaşmaktadır. Buharlaştırma işlemi az tercih edilen tüm yollardaki feromonlara uygulanmaktadır. Buharlaştırma için 0 ve 1 arasında sabit bir değere sahiptir. Karıncaların turlamış oldukları yollardaki feromon miktarları, o yolu önceden kullanan karıncanın toplam yol uzunluğuyla ters orantılı olarak artış miktarında değişim gösterecektir. Böylelikle, kısa yollardan geçmiş olan karıncaların feromon miktarları daha fazla artışa sebep olacaktır.

Feromon güncellemeleri iki şekilde yapılmaktadır. Bunlar; Local ve Global feromon güncellemeleridir.

3.7.1. Local feromon güncellemesi

Feromon değerleri bir matris olarak düşünülecek olursa; matris değeri t iterasyonuna kadar ilerleyen feromon değerlerine ait bir vektörü oluşacaktır. t iterasyonundaki feromon düzeyi ve $0 < \rho < 1$ aralığındaki buharlaştırma sabiti üzerine ilgili local feromon güncelleme formülü şöyle olacaktır; (Denklem 3.3 – 3.4)

$$\tau_{ij}(t + 1) = (1 - \rho) \tau_{ij}(t) + \sum_{k=1}^m \Delta \tau_{ij}^k(t + 1) \quad (3.3)$$

$$\Delta \tau_{ij}^k(t+1) = \begin{cases} 1/L^k(t+1) & k \text{ karıncası } (i,j) \\ & \text{yolunu kullanmışsa,} \\ 0 & \text{diğer durumlarda} \end{cases} \quad (3.4)$$

$L^k(t+1)$, k karıncasının $t+1$ iterasyonundaki toplam tur miktarıdır. Karıncalar değişen feromon miktarları ile birlikte her iterasyonda turlarını da değiştirmektedirler. Amaç kısa yola ait turları tespit edebilmektir.

3.7.2. Global feromon güncellemesi

Karınca koloni algoritmasında geçerli yollara ait en iyi sonuca sahip olan k karıncasının izlediği yolun feromon düzeyinin artırılması sağlanır. Bununla birlikte iterasyonlarda bulunan en iyi sonuçların belli bir oranda ileriki iterasyonlara aktarılması gerçekleştirilir. $L_{best}(t+1)$, geçerli iterasyonda bulunan en iyi yola ait olan turun uzunluk miktarıdır.

Local feromon güncellemesine benzeyen bu olayın formülü şöyle olacaktır; (Denklem 3.5 – 3.6)

$$\tau_{ij}(t+1) = (1 - \rho) \tau_{ij}(t) + \Delta \tau_{ij}^k(t+1) \quad (3.5)$$

$$\Delta \tau_{ij}(t+1) = \begin{cases} \frac{1}{L_{best}(t+1)} & (i,j) \text{ en iyi tura ait ise} \\ 0 & \text{diğer durumlarda} \end{cases} \quad (3.6)$$

BÖLÜM 4. KENAR BELİRLEME

4.1. Kenar Belirleme Nedir?

Kenar belirleme (Edge Detection) olarak bilinen en önemli görüntü işleme tekniğine ait bir çok algoritma bulunmaktadır. Bu algoritmaların aracılığıyla var olan görüntümüze ait kenarları tespit edebiliriz.

Kenar olarak bildiğimiz tüm tanımlar image'lar yani görüntüler üzerinde farklı bir boyut almaktadır. Image'larda kenarın tek bir anlamı vardır o da, görüntüler arasındaki renk değişimleridir. Fakat her renk değişiminde kenar olarak tespit edilemez.

Kenar tespit algoritmalarında çalışırken öncelikle üzerinde çalıştığımız görüntünün siyah - beyaza dönüştürülmesi gerekmektedir. Kenar tespit edilmeden önce belli bir threshold yani eşik değeri belirlenmelidir. Eşik değerinden sonra görüntü üzerindeki satır ve sütunlarda bulunan tüm pixel'ler arasındaki renk tonlarının değişimi bu eşik değerini referans alarak kenarlar tespit edilebilir. Eğer görüntü üzerindeki renk değişimi eşik değerin altında bir sonuç ise, kenar olarak isimlendirilemez. [4]

4.2. Kenar Belirleme Yöntemlerinde En Çok Kullanılan Algoritmalara Ait Filtreler

4.2.1. Kullandığımız kenar belirleme tekniği

Bu yöntemde, son feromon matrisi $T^{(n)}$ üzerinde bir eşik değeri uygulayarak her bir pikselin kenar olup olmadığı konusunda bir karar verilmektedir. Başlangıç eşiği $T^{(0)}$ feromon matrisinin ortalama değeri olarak seçilmiştir.

Daha sonra, feromon matrisinin değerleri $T^{(0)}$ eşik değerinden küçük veya büyük olduğu kriterlere göre iki kategoriye ayrılmaktadır.

Daha sonra, yeni eşik değeri hesaplanan iki kategorinin ortalaması olarak bulunmaktadır. Bu süreç eşik değeri artık değişmeyene kadar tekrarlanmaktadır (Kullanıcı tarafından belirlenen hata payına göre).

Bu tekrarlanan prosedür aşağıdaki gibi özetlenebilir;

Adım 1: Başlangıç eşik değeri $T^{(0)}$ hesapla.

$$T^{(0)} = \frac{\sum_{i=1:M_1} \sum_{j=1:M_2} T_{i,j}^{(N)}}{M_1 M_2} \quad (4.1)$$

Ve iterasyon girdisi olan l 'yi sıfıra eşitle. ($l = 0$)

Adım 2: Eşik değerini $T^{(l)}$ kullanarak Feromon Matrisi $T^{(n)}$ 'i iki ayrı kategoriye ayrılmaktadır. İlk kategori eşik değerinden $T^{(l)}$ küçük olan değerleri tutmaktadır. Diğer kategori ise geriye kalan değerleri tutmaktadır. Daha sonra bu kategoriler denklem 4.2 ile hesaplanmaktadır. [5]

$$m_L^{(l)} = \frac{\sum_{i=1:M_1} \sum_{j=1:M_2} g_{T^{(l)}}^L(T_{i,j}^{(N)})}{\sum_{i=1:M_1} \sum_{j=1:M_2} h_{T^{(l)}}^L(T_{i,j}^{(N)})} \quad (4.2)$$

$$m_U^{(l)} = \frac{\sum_{i=1:M_1} \sum_{j=1:M_2} g_{T^{(l)}}^U(T_{i,j}^{(N)})}{\sum_{i=1:M_1} \sum_{j=1:M_2} h_{T^{(l)}}^U(T_{i,j}^{(N)})} \quad (4.3)$$

$$g_{T^{(l)}}^L(x) = \begin{cases} x & \text{eğer } x \leq T^{(l)}; \\ 0 & \text{diğer durumlarda} \end{cases} \quad (4.4)$$

$$h_{T^{(l)}}^L(x) = \begin{cases} 1 & \text{eğer } x \leq T^{(l)}; \\ 0 & \text{diğer durumlarda} \end{cases} \quad (4.5)$$

$$g_{T^{(l)}}^U(x) = \begin{cases} x & \text{eğer } x \geq T^{(l)}; \\ 0 & \text{diğer durumlarda} \end{cases} \quad (4.6)$$

$$h_{T^{(l)}}^U(x) = \begin{cases} 1 & \text{eğer } x \geq T^{(l)}; \\ 0 & \text{diğer durumlarda} \end{cases} \quad (4.7)$$

Adım 3: İterasyon girdisini bir arttırarak ($l = l + 1$) yeni eşik değeri aşağıdaki formül ile güncellenmektedir.

$$T^{(l)} = \frac{m_L^{(l)} + m_U^{(l)}}{2} \quad (4.8)$$

Adım 4: Eğer $|T^{(l)} - T^{(l-1)}|$ işleminin sonucu kullanıcı tarafından belirlenen hata payından büyük ise tekrar Adım 2'ye gidilmektedir. Eğer değilse karar aşamasına geçilerek her bir pikselin kenar olup olmadığı kararı verilmektedir.

$$E_{i,j} = \begin{cases} 1 & \text{eğer } T_{i,j}^{(N)} \geq T^{(l)}; \\ 0 & \text{diğer durumlarda} \end{cases} \quad (4.9)$$

4.2.2. Sobel filtresi

Sobel filtresine ait algoritmada iki tane konvolasyon kerneli bulunmaktadır. Birisi yatay kenarları tespit etmek için kullanılırken diğeri ise dikey kenarları tespit etmek için kullanır. Eksenler üzerindeki piksellere daha çok ağırlık verir.

4.2.3. Prewitt filtresi

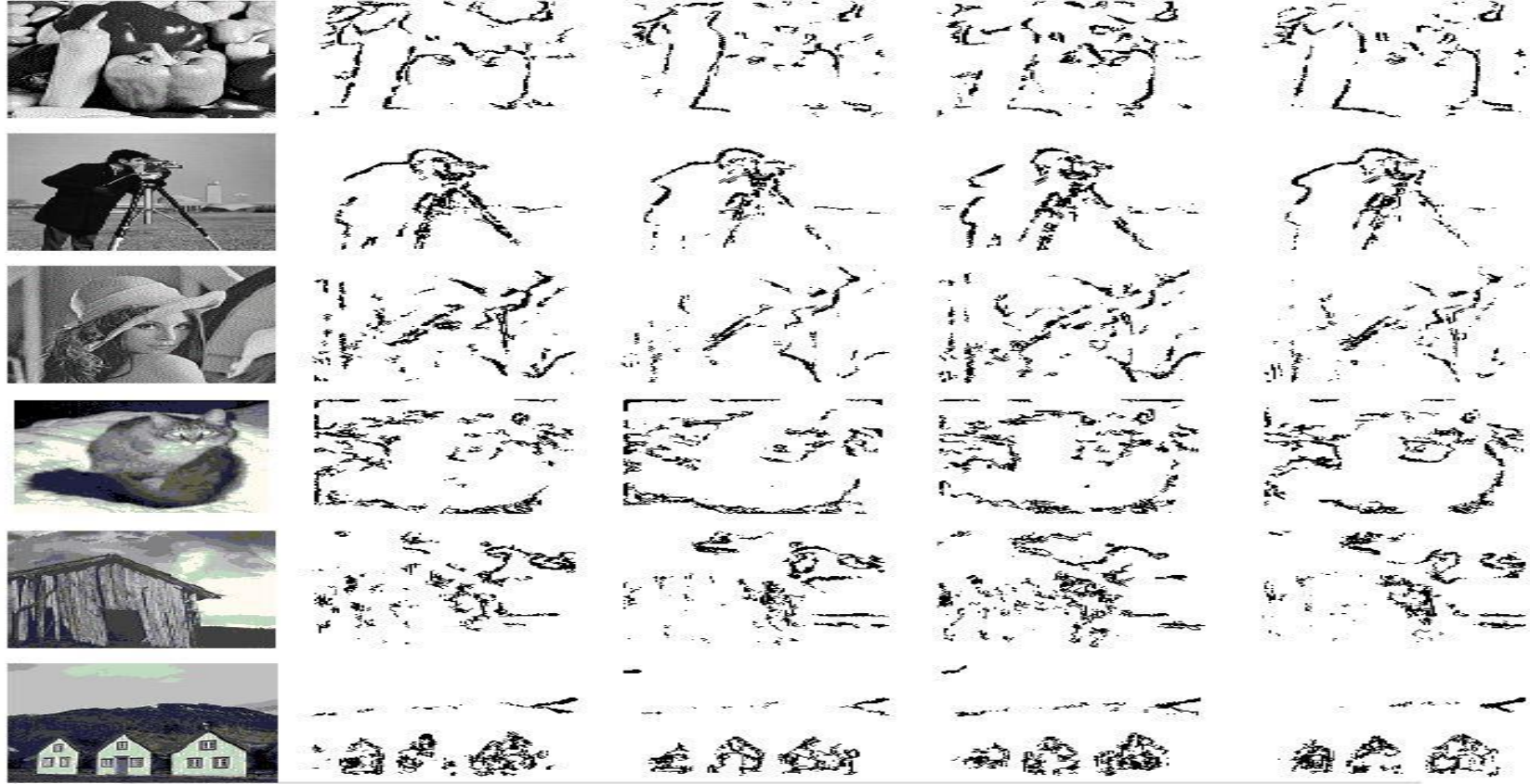
Görüntüler üzerinde yatay ve dikey yönlere ait olan kernellerle birlikte eğimlere odaklanarak sonuçlar vermektedir.

4.2.4. Canny filtresi

Kenar tespit yöntemlerinde en başarılı sonucu veren bir kernel yapısına sahiptir. Görüntü türevi alınmadan önce yumuşatma filtresi uygulanmaktadır. Tek piksel kalınlığında kenarlar üretir ve kırık çizgilerle birlikte pikselleri birleştirir.

Kenar bulma işlemlerinde genel amaç; görüntüdeki gürültülere karşı düşük duyarlılığı bulabilmektedir. Sınırların iyi belirlenebilmesi ve geri kalan tüm kenarlardaki karışıklıkları elemektir. Bu sebepten, kerneller sayesinde görüntü içerisindeki ışık yoğunluklarına ait değişikliklerin ani olduğu yerleri yakalayabiliriz.

4.3. Karınca Koloni Algoritması İle Kenar Belirleme Örnekleri



Şekil 4.1. Karınca koloni algoritması ile elde edilmiş resim kenarları

BÖLÜM 5. UYGULAMA

İlk aşamada iki boyutlu fotoğrafın MATLAB programında piksel değerleri alınarak bir matris oluşturulmaktadır. C++ programlama dili kullanılarak bu matris üzerinde her piksel için ön işleme algoritması uygulanmıştır.

Elde edilen bu matris üzerinde karınca kolonisi algoritması aşağıdaki feromon ve olasılık hesaplama formülleri ile uygulanmıştır.

$$T_{i,j}^{(n-1)} = \begin{cases} (1-p)T_{i,j}^{(n-1)} + p\Delta_{i,j}^{(k)}, & (i,j) \text{ karınca tarafından;} \\ & \text{ziyaret edilmişse;} \\ T_{i,j}^{(n-1)}, & \text{diğer durumlarda} \end{cases} \quad (5.1)$$

$$p_{(i,j)}^{(n)} = (T_{i,j}^{(n-1)})^\alpha \cdot (n_{i,j})^\beta \quad (5.2)$$

Bu işlemden sonra eşik değeri hesaplanmaktadır.

$$T^{(0)} = \frac{\sum_{i=1:M_1} \sum_{j=1:M_2} T_{i,j}^{(N)}}{M_1 M_2} \quad (5.3)$$

$$m_L^{(l)} = \frac{\sum_{i=1:M_1} \sum_{j=1:M_2} g_{T^{(l)}}^L(T_{i,j}^{(N)})}{\sum_{i=1:M_1} \sum_{j=1:M_2} h_{T^{(l)}}^L(T_{i,j}^{(N)})} \quad (5.4)$$

$$m_U^{(l)} = \frac{\sum_{i=1:M_1} \sum_{j=1:M_2} g_{T^{(l)}}^U(T_{i,j}^{(N)})}{\sum_{i=1:M_1} \sum_{j=1:M_2} h_{T^{(l)}}^U(T_{i,j}^{(N)})} \quad (5.5)$$

$$g_{T^{(l)}}^L(x) = \begin{cases} x & \text{eğer } x \leq T^{(l)}; \\ 0 & \text{diğer durumlarda} \end{cases} \quad (5.6)$$

$$h_{T^{(l)}}^L(x) = \begin{cases} 1 & \text{eğer } x \leq T^{(l)}; \\ 0 & \text{diğer durumlarda} \end{cases} \quad (5.7)$$

$$g_{T^{(l)}}^U(x) = \begin{cases} x & \text{eğer } x \geq T^{(l)}; \\ 0 & \text{diğer durumlarda} \end{cases} \quad (5.8)$$

$$h_{T^{(l)}}^U(x) = \begin{cases} 1 & \text{eğer } x \geq T^{(l)}; \\ 0 & \text{diğer durumlarda} \end{cases} \quad (5.9)$$

$$T^{(l)} = \frac{m_L^{(l)} + m_U^{(l)}}{2} \quad (5.10)$$

Bulunan bu eşik değeri ile feromon matrisi karşılaştırılmaktadır. Bu karşılaştırma sonucunda feromon değerleri aşağıdaki ifade ile kenar olup olmadıkları belirlenmektedir.

$$E_{i,j} = \begin{cases} 1 & \text{eğer } T_{i,j}^{(N)} \geq T^{(l)}; \\ 0 & \text{diğer durumlarda} \end{cases} \quad (5.11)$$

Tablo 5.1. Örnek karınca uygulaması 1



İterasyon	Karınca Sayısı	Adım 25	Adım 50	Adım 100	Adım 150	Adım 200
100	244					
	500					
200	244					
	500					

Tablo 5.2. Örnek karınca uygulaması 2



İterasyon	Karınca Sayısı	Adım 25	Adım 50	Adım 100	Adım 150	Adım 200
100	244					
	500					
200	244					
	500					



İterasyon	Karınca Sayısı	Adım 25	Adım 50	Adım 100	Adım 150	Adım 200
100	244					
	500					
200	244					
	500					

BÖLÜM 6. SONUÇLAR VE ÖNERİLER

Biz, bu çalışmada bir fotoğraftaki insan yüzünün bilgisayar ortamında görüntü işleme yapılarak elde edilebileceğini gösterdik. Çalışmada kullandığımız yaklaşım üzerinden ilerlenerek bir nesnenin tanımı, insan yüzünün algılanabilmesi ve yüz tanıma sistemleri gibi çalışmalar geliştirilebilir.

KAYNAKLAR

- [1,2,5] Jing Tian, Weiyu Yu and Shengli Xie, An Ant Colony Optimization Algorithm For Image Edge Detection
- [3,4] Yalova Üniversitesi, Bilgisayar Muhendisliği - Bilgisayarlı Görme Dersi
- [6] M. Dorigo and S. Thomas, Ant Colony Optimization. Cambridge: MIT Press, 2004.
- [7] H.-B. Duan, Ant Colony Algorithms: Theory and Applications. Beijing: Science Press, 2005.
- [8] M. Dorigo, V. Maniezzo, and A. Coloni, "Ant system: Optimization by a colony of cooperating agents," IEEE Trans. on Systems, Man and Cybernetics, Part B, vol. 26, pp. 29–41, Feb. 1996.
- [9] M. Dorigo, M. Birattari, and T. Stutzle, "Ant colony optimization," IEEE Computational Intelligence Magazine, vol. 1, pp. 28–39, Nov. 2006.
- [10] <https://www.eneskamis.com/c-programlama-dilini-ogrenmeliyim/>
- [11] T. Stutzle and H. Holger H, "Max-Min ant system," Future Generation Computer Systems, vol. 16, pp. 889–914, Jun. 2000.
- [12] M. Dorigo and L. M. Gambardella, "Ant colony system: A cooperative learning approach to the traveling salesman problem," IEEE Trans. on Evolutionary Computation, vol. 1, pp. 53–66, Apr. 1997.
- [13] M. Dorigo, G. D. Caro, and T. Stutzle, Special Issue on Ant Algorithms, Future Generation Computer Systems, vol. 16, Jun. 2000.
- [14] O. Cordon, F. Herrera, and T. Stutzle, Special Issue on Ant Colony Optimization: Models and Applications, Mathware and Soft Computing, vol. 9, Dec. 2002

ÖZGEÇMİŞ

İsmet SANDIKÇI

8 Aralık 1994 tarihinde Bağcılar/İstanbul'da doğdu. İlk ve orta eğitimini Bahçelievler'de, lise eğitimini Bakırköy'de, ön lisans eğitimini ise Avcılar'da tamamladı. 2012 Yılında Bakırköy Anadolu Ticaret Meslek Lisesi, Web Programlama bölümünden, 2014 yılında İstanbul Gelişim Üniversitesi Meslek Yüksek Okulu, Bilgisayar Programcılığı bölümünden mezun oldu. 2015 yılında Sakarya Üniversitesi Bilgisayar Mühendisliği Bölümü'nü kazandı. 2016 yılında Gantek Teknoloji de donanım stajını yaptı. Yazılım stajını da 2017 yılında Mobus Bilişim Çözümleri ve Danışmanlık San. Tic. Ltd. Şti. de yaptı.

Barış KAYA

19 Mayıs 1995 tarihinde Eldivan/Çankırı'da doğdu. İlk ve orta eğitimini Sultangazi/İstanbul'da, lise eğitimini Bayrampaşa/İstanbul'da, ön lisans eğitimini ise Avcılar/İstanbul'da tamamladı. 2013 yılında İnönü Mesleki ve Teknik Anadolu Lisesi, Web Programcılığı bölümünden, 2015 yılında İstanbul Üniversitesi Meslek Yüksek Okulu, Bilgisayar Programcılığı bölümünden mezun oldu. 2015 yılında Sakarya Üniversitesi Bilgisayar Mühendisliği Bölümü'nü kazandı. 2017 yılında Erkay Teknoloji Geliştirme Hizmetleri San. Tic. LTD. ŞTİ. de yazılım stajını yaptı ve Ağustos 2017 tarihinden itibaren yarı zamanlı yazılım geliştiriciliği yapmaktadır.

Mustafa Olcay TERZİOĞLU

26 Temmuz 1993 tarihinde Fatih/İstanbul'da doğdu. İlk, orta, lise ve ön lisans eğitimini İstanbul'da tamamladı. 2011 Yılında Atatürk Anadolu Teknik Lisesi, Bilişim Teknolojileri bölümünden, 2015 yılında Yıldız Teknik Üniversitesi Meslek Yüksek Okulu, Bilgisayar Programcılığı bölümünden mezun oldu. 2015 yılında Sakarya Üniversitesi Bilgisayar Mühendisliği Bölümü'nü kazandı. 2017 yılında MBIS Bilgi Teknolojisi ve Hizmetleri'nde yazılım stajını yaptı.

BSM 498 BİTİRME ÇALIŞMASI DEĞERLENDİRME VE SÖZLÜ SINAV TUTANAĞI

KONU : 2B FOTOĞRAF ÜZERİNDE İNSAN YÜZÜNÜN TESPİTİ

ÖĞRENCİLER (Öğrenci No/AD/SOYAD):

B1412.10301 – İsmet SANDIKÇI

B1412.10305 – Barış KAYA

B1412.10307 – M. Olcay TERZİOĞLU

Değerlendirme Konusu	İstenenler	Not Aralığı	Not
Yazılı Çalışma			
Çalışma klavuza uygun olarak hazırlanmış mı?	x	0-5	
Teknik Yönden			
Problemin tanımı yapılmış mı?	x	0-5	
Geliştirilecek yazılımın/donanımın mimarisini içeren blok şeması (yazılımlar için veri akış şeması (dfd) da olabilir) çizilerek açıklanmış mı?			
Blok şemadaki birimler arasındaki bilgi akışına ait model/gösterim var mı?			
Yazılımın gereksinim listesi oluşturulmuş mu?			
Kullanılan/kullanılması düşünülen araçlar/teknolojiler anlatılmış mı?			
Donanımların programlanması/konfigürasyonu için yazılım gereksinimleri belirtilmiş mi?			
UML ile modelleme yapılmış mı?			
Veritabanları kullanılmış ise kavramsal model çıkarılmış mı? (Varlık ilişkisi modeli, noSQL kavramsal modelleri v.b.)			
Projeye yönelik iş-zaman çizelgesi çıkarılarak maliyet analizi yapılmış mı?			
Donanım bileşenlerinin maliyet analizi (prototip-adetli seri üretim vb.) çıkarılmış mı?			
Donanım için gerekli enerji analizi (minimum-uyku-aktif-maksimum) yapılmış mı?			
Grup çalışmalarında grup üyelerinin görev tanımları verilmiş mi (iş-zaman çizelgesinde belirtilebilir)?			
Sürüm denetim sistemi (Version Control System; Git, Subversion v.s.) kullanılmış mı?			
Sistemin genel testi için uygulanan metotlar ve iyileştirme süreçlerinin dökümü verilmiş mi?			
Yazılımın sızma testi yapılmış mı?			
Performans testi yapılmış mı?			
Tasarımın uygulamasında ortaya çıkan uyumsuzluklar ve aksaklıklar belirtilerek çözüm yöntemleri tartışılmış mı?			
Yapılan işlerin zorluk derecesi?	x	0-25	
Sözlü Sınav			
Yapılan sunum başarılı mı?	x	0-5	
Soruları yanıtlama yetkinliği?	x	0-20	
Devam Durumu			
Öğrenci dönem içerisindeki raporlarını düzenli olarak hazırladı mı?	x	0-5	
Diğer Maddeler			
Toplam			

DANIŞMAN (JÜRİ ADINA):

DANIŞMAN İMZASI: