

KARADENİZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

İSTATİSTİK VE BİLGİSAYAR BİLİMLERİ ANABİLİM DALI

AĞIRLIKLİ YÖNSEL KÜMELEME ALGORİTMALARI KULLANILARAK
RENKLİ GÖRÜNTÜLERİN RENK ÖZÜ TABANLI BÖLÜTLENMESİ

YÜKSEK LİSANS TEZİ

Soheila VEFAEINHR

MART 2015
TRABZON

KARADENİZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

İSTATİSTİK VE BİLGİSAYAR BİLİMLERİ ANABİLİM DALI

AĞIRLIKLİ YÖNSEL KÜMELEME ALGORİTMALARI KULLANILARAK
RENKLİ GÖRÜNTÜLERİN RENK ÖZÜ TABANLI BÖLÜTLENMESİ

Soheila VEFAEINHR

Karadeniz Teknik Üniversitesi Fen Bilimleri Enstitüsünde
"YÜKSEK LİSANS (İSTATİSTİK)"
Unvanı Verilmesi İçin Kabul Edilen Tezdir.

Tezin Enstitüye Verildiği Tarih : 30.01.2015
Tezin Savunma Tarihi : 05.03.2015

Tez Danışmanı : Yrd. Doç. Dr. Orhan KESEMEN

Trabzon 2015

Karadeniz Teknik Üniversitesi Fen Bilimleri Enstitüsü
İstatistik ve Bilgisayar Bilimleri Anabilim Dalında
Soheila VEFAEINAHİR tarafından hazırlanan

**AĞIRLIKLI YÖNSEL KÜMELEME ALGORİTMALARI KULLANILARAK
RENKLİ GÖRÜNTÜLERİN RENK ÖZÜ TABANLI BÖLÜTLENMESİ**

**başlıklı bu çalışma, Enstitü Yönetim Kurulunun 10 / 02 / 2015 gün ve 1589 sayılı
kararıyla oluşturulan jüri tarafından yapılan sınavda
YÜKSEK LİSANS TEZİ
olarak kabul edilmiştir.**

Jüri Üyeleri

Başkan : Yrd. Doç. Dr. Emine TANIR KAYIKÇI

Üye : Yrd. Doç. Dr. Orhan KESEMEN

Üye : Yrd. Doç. Dr. Halil İbrahim ŞAHİN

Prof. Dr. Sadettin KORKMAZ
Enstitü Müdürü

ÖNSÖZ

Bu tez, Karadeniz Teknik Üniversitesi Fen Bilimleri Enstitüsü İstatistik ve Bilgisayar Bilimleri Anabilim Dalı, Yüksek Lisans Programı'nda hazırlanmıştır. Bu tez, “Ağırlıklı Yönel Kümeleme Algoritmaları Kullanılarak Renkli Görüntülerin Renk Özü Tabanlı Bölütlenmesi” için yapılmıştır. Tez çalışma süresinde değerli yardım ve katkılarıyla beni yönlendiren danışman hocam Yrd. Doç. Dr. Orhan KESEMEN 'ne, teşekkürlü bir borç bilirim.

Çalışmalarım boyunca maddi ve manevi desteklerini esirgemeyen anne ve babama teşekkür ederim. Bu tezin, bundan sonraki çalışmalara katkı sağlamasını temenni ederim.

Soheila VEFAEINAHİR

Trabzon 2015

TEZ BEYANNAMESİ

Yüksek Lisans Tezi olarak sunduğum “Ağırlıklı Yönel Kümeleme Algoritmaları Kullanılarak Renkli Görüntülerin Renk Özü Tabanlı Bölütlenmesi” başlıklı bu çalışmayı baştan sona kadar danışmanım Yrd. Doç. Dr. Orhan KESEMEN’in sorumluluğunda tamamladığımı, verileri/örnekleri kendim topladığımı, deneyleri/analizleri ilgili laboratuvarlarda yaptığımı/yaptırdığımı, başka kaynaklardan aldığım bilgileri metinde ve kaynakçada eksiksiz olarak gösterdiğimi, çalışma sürecinde bilimsel araştırma ve etik kurallara uygun olarak davrandığımı ve aksinin ortaya çıkması durumunda her türlü yasal sonucu kabul ettiğimi beyan ederim. 30/01/2015

Soheila VEFAEINHR

İÇİNDEKİLER

	<u>Sayfa No</u>
ÖNSÖZ.....	III
TEZ BEYANNAMESİ.....	IV
İÇİNDEKİLER.....	V
ÖZET	VII
SUMMARY	VIII
ŞEKİLLER DİZİNİ	IX
TABLolar DİZİNİ.....	XII
SEMBOLLER DİZİNİ	XIII
1. GENEL BİLGİLER	1
1.1. Sayısal Görüntülerde Renkler.....	2
1.2. Renk Sayısına Göre Görüntü Çeşitleri	3
1.3. Renk Uzayları	4
1.3.1. RGB Renk Uzayı.....	5
1.3.2. CMY/CMYK Renk Uzayı	6
1.1.1. HSI Renk Uzayı.....	7
1.3.3. RGB Renk Uzayından HSI Renk Uzayına Geçiş	9
1.3.1. HSI Renk Uzayından RGB Renk Uzayına Geçiş.....	10
1.1.2. HSV Renk Uzayı	12
1.3.2. RGB'den HSV Geçiş.....	13
1.3.1. HSV'den RGB'ye Geçiş.....	15
1.4. Sayısal Görüntüde Histogram.....	16
1.5. Görüntü Eşikleme	18
1.5.1. Eşikleme İşlemi	18
1.5.2. Optimal Eşik Değerinin Belirlenmesi.....	20
1.6. Bulanık C-Ortalamalar Kümeleme Algoritması.....	23
1.7. Yönel Veriler İçin Bulanık C-Ortalamalar Kümelemesi	26
1.8. Bulanık Kümeleme Algoritmalarında Optimal Küme Sayısı.....	30
1.8.1. Bölünme Katsayısı (PC)	30
1.8.2. Bölünme Entropisi (CE)	31

2.	YAPILAN ÇALIŞMALAR.....	32
2.1.	Gri Ölçekli Görüntülerde Bölütleme	32
2.1.1.	Gözet Tabanlı Kümeleme ile Bölütleme	32
2.1.2.	Histogram Tabanlı Kümeleme ile Bölütleme	36
2.2.	Renkli Görüntülerde Bölütleme.....	41
2.2.1.	Gözet Tabanlı Kümeleme ile Bölütleme	42
2.2.2.	Histogram Tabanlı Kümeleme ile Bölütleme	43
3.	BULGULAR VE SONUÇLAR.....	52
4.	ÖNERİLER.....	54
5.	KAYNAKLAR	55

ÖZGEÇMİŞ

Yüksek Lisans

ÖZET

AĞIRLIKLI YÖNSEL KÜMELEME ALGORİTMALARI KULLANILARAK RENKLİ
GÖRÜNTÜLERİN RENK ÖZÜ TABANLI BÖLÜTLENMESİ

Soheila VEFAEINHR

Karadeniz Teknik Üniversitesi
Fen Bilimleri Enstitüsü
İstatistik ve Bilgisayar Bilimleri Anabilim Dalı
Danışman: Yrd. Doç. Dr. Orhan KESEMEN
2015, 56 Sayfa

Bu çalışmada, renkli görüntüleri çözümlmek için ön işlem adımlarından birisi olan görüntü bölütleme işlemi üzerinde durulmuştur. Bu işlem için RGB uzayında gerçekleştirilecek bir kümeleme işleminin küçük bir görüntüde bile inanılmaz hesaplama zamanlarına ulaştığı görülmüştür. Diğer bir yaklaşım ise histogram tabanlı bir yaklaşım olmuştur. Ancak histogram tabanlı yaklaşımlarda tek kanallı veya gri-ölçekteki görüntülere uygulanabilmektedir. Dolayısıyla renkli bir görüntüde bölütleme yapmak için renk özü tabanlı bir yaklaşım düşünülmüştür. Ancak renk özü açısıl tabanlı bir değişim gösterdiğinden sayı doğrusu tabanlı işlemlerin kullanımı mümkün olmamıştır. Problemi çözmek için yönsel tabanlı kümeleme algoritmalarına başvurulmuştur. Yönsel tabanlı algoritmaların görüntü bölütleme işlemi üzerindeki başarımı ölçülerek karşılaştırmaları yapılmıştır.

Anahtar Kelimeler: Görüntü Bölütleme, Renk Özü Tabanlı Bölütleme, Yönsel Veriler, Kümeleme Algoritmaları.

Master Thesis

SUMMARY

HUE-BASED SEGMENTATION OF COLOR IMAGES USING WEIGHTED
DIRECTIONAL CLUSTERING ALGORITHMS

Soheila VEFAEINHR

Karadeniz Technical University
The Graduate School of Natural and Applied Sciences
Statistical and Computer Sciences Graduate Program
Supervisor: Assist. Prof.Dr. Orhan KESEMEN
2015, 56 Pages

In this study, to analyze the color images, one of the pre-processing steps are focused on the image segmentation process. For this procedure, a clustering process to be performed in RGB space, a small image is seen that even reached the incredible computing time. Another approach has been a histogram based approach. However, in histogram approach can be applied in single-channel or gray-scale images. Thus, in a color image, color-based segmentation is considered an approach to extract. However, since the hue angle based showed a change, it was not possible to use the number line based operations. To solve the problem, directional-based clustering algorithms have been applied. Directional based algorithms, measuring performance on the image segmentation process are compared.

Key Words: Image Segmentation, Hue-Based Segmentation, Directional Data, Clustering Algorithms.

ŞEKİLLER DİZİNİ

Sayfa No

Şekil 1.	Kübik (RGB) renk uzayı.....	6
Şekil 2.	Konik (HSI) renk uzayı	8
Şekil 3.	Renk özü (H) değerinin değiştirilmesiyle görüntüdeki değişim; (a) H; (b) H + 60°; (c) H + 120°; (d) H + 180°; (e) H + 240°; (f) H + 300°	9
Şekil 4.	Konik (HSV) renk uzayı.....	13
Şekil 5.	Histogram görüntüleme; (a) Örnek Lena görüntüsü; (b) Lena görüntüsünün histogramı	18
Şekil 6.	İki seviyeli görüntü eşikleme; (a) Özgün görüntü; (b) Eşiklenmiş görüntü.....	19
Şekil 7.	Şekil 6’teki özgün görüntüye göre eşik değerinin doğru seçilmesi; (a) Eşik değerinin (T = 0.3) düşük seçilmesi; (b) Eşik değerinin yüksek seçilmesi (T = 0.7).....	20
Şekil 8.	İki seviyeli görüntü eşikleme; (a) Özgün görüntü; (b) Eşiklenmiş görüntü; (c) Optimal eşik değerinin belirlenmesi.....	22
Şekil 9.	FCM kümeleme algoritması için seçilen iki boyutlu örnek veriler	25
Şekil 10	Şekil 9’da verilen örnek verilerin FCM kümeleme algoritması ile 3 kümeye ayrılması.....	26
Şekil 11.	FCM4DD kümeleme algoritması için seçilen dairesel örnek veriler	29
Şekil 12	Şekil 11’de verilen örnek verilerin FCM4DD kümeleme algoritması ile 3 kümeye ayrılması.....	29
Şekil 13.	Örnek olarak kullanılan özgün görüntü.	33
Şekil 14.	Şekil 13’teki görüntünün iki kümeye ayrılarak bölütlenmesi ile elde edilen görüntü.	34
Şekil 15	Şekil 14’de verilen bölütlenmiş (renk indirgenmiş) görüntünün önceden seçilen renkler yardımıyla etiketlenmiş olarak gösterimi.	34

Şekil 16.	Şekil 13'teki görüntünün dört kümeye ayrılarak bölütlenmesi ile elde edilen görüntü.	35
Şekil 17.	Şekil 16'da verilen bölütlenmiş (renk indirgenmiş) görüntünün önceden seçilen renkler yardımıyla etiketlenmiş olarak gösterimi.	35
Şekil 18.	Şekil 8'deki görüntünün histogramı ve $C = 2$ için kümeleme sonucunda elde edilen üyelik değerleri.....	37
Şekil 19.	Şekil 13'deki görüntünün iki kümeye ayrılarak bölütlenmesi ile elde edilen görüntü	38
Şekil 20	Şekil 19'da verilen bölütlenmiş (renk indirgenmiş) görüntünün önceden seçilen renkler yardımıyla etiketlenmiş olarak gösterimi.	38
Şekil 21.	Şekil 8'deki görüntünün histogramı ve kümeleme sonucunda elde edilen üyelik değerleri	39
Şekil 22.	Şekil 8'deki görüntünün iki kümeye ayrılarak bölütlenmesi ile elde edilen görüntü	40
Şekil 23.	Şekil 11'de verilen bölütlenmiş (renk indirgenmiş) görüntünün önceden seçilen renkler yardımıyla etiketlenmiş olarak gösterimi.	40
Şekil 24.	Plastik toplar (URL-2, 2015)	41
Şekil 25.	Tığ işlemeli çiçek örgüleri (URL-3, 2015).....	42
Şekil 26.	Plastik toplar görüntüsünün ağırlıklandırılmış FCM algoritmasıyla 5 bölüte ayrılması	42
Şekil 27.	Tığ işlemeli çiçek örgüleri görüntüsünün ağırlıklandırılmış FCM algoritmasıyla 8 bölüte ayrılması.....	43
Şekil 28.	Şekil 15'te verilen örnek görüntünün S ve V değerlerinin 1'e eşiklenmesiyle elde edilen görüntü.....	45
Şekil 29	Şekil 19'daki görüntünün renk özü (H) değerlerinin polar histogramı	46
Şekil 30.	Optimal küme sayısının belirlenmesi	46
Şekil 31.	Şekil 29'daki histograma ağırlıklı FCM4DD kümeleme algoritması uygulanmasıyla elde edilen üyelik değerleri	47
Şekil 32.	Şekil 28'deki görüntüdeki renk özü (H) değerlerine göre 5 parçalı bölütleme yapıldıktan sonra elde edilen bölütlenmiş görüntü.....	47
Şekil 33.	Şekil 25'de verilen örnek görüntünün S ve V değerlerinin 1'e eşiklenmesiyle elde edilen görüntü.....	48

Şekil 34.	Şekil 23'teki görüntünün renk özü (H) değerlerinin polar histogramı	49
Şekil 35.	Optimal küme sayısının belirlenmesi	49
Şekil 36.	Şekil 34'teki histograma ağırlıklı FCM4DD kümeleme algoritması uygulanmasıyla elde edilen üyelik değerleri.	50
Şekil 37.	Şekil 33'teki görüntüdeki renk özü (H) değerlerine göre 2 parçalı bölütleme yapıldıktan sonra elde edilen bölütlenmiş görüntü	50
Şekil 38.	Şekil 34'teki histograma göre gözlem sonucu küme sayısı 8 olarak belirlenip ağırlıklı FCM4DD algoritmasıyla elde edilen üyelik değerleri.....	51
Şekil 39.	Şekil 33'teki görüntüdeki renk özü (H) değerlerine göre 8 parçalı bölütleme yapıldıktan sonra elde edilen bölütlenmiş görüntü	51

TABLULAR DİZİNİ

	<u>Sayfa No</u>
Tablo 1. Işık dalga boylarına ve frekanslarına göre renkleri	2
Tablo 2. Gözek derinliğine göre görüntü çeşitleri	4
Tablo 3. Gri ölçekli görüntü bölütleme sonuçlarının karşılaştırılması	52
Tablo 4. Renkli görüntü bölütleme sonuçlarının karşılaştırılması	53

SEMBOLLER DİZİNİ

α_{rad}	:Radyan cinsinden tanımlanan açı
α_{deg}	:Derece cinsinden tanımlanan açı
$\bar{\theta}$:Ortalama Yön
μ	:Bulanık üyelik değerleri
FCM	:Bulanık c-Ortalamlar (Fuzzy C-Means) Algoritması
FCM4DD	:Yönsel Veriler (Directional Data) için Bulanık c-Ortalamlar (Fuzzy C-Means) Algoritması
RGB	:Kırmızı (Red, R), Yeşil (Green, G) ve Mavi (Blue, B) renk uzayı
CMY-K	:Turkuaz (Cyan, C), Güvez (Magenta, M), Sarı (Yellow, Y) ve Siyah (Black, K) renk uzayı
HSI	:Renk Özü (Hue, H), Doygunluk (Saturation, S), Yoğunluk (Intensity, I) uzaylarını göstermektedir.
HSV	:Renk Özü (Hue, H), Doygunluk (Saturation, S), Renk Değerli (Value, V) uzaylarını göstermektedir.
YIQ	:YIQ renk uzayı
XYZ	:XYZ renk uzayı
Lab	:Lab renk uzayı
YUV	:YUV renk uzayı
UVW	:UVW renk uzayı

1. GENEL BİLGİLER

Günümüzde teknolojilerin gelişmesiyle birlikte insan odaklı birçok çalışma makineler tarafından gerçekleştirilmeye başlamıştır. Akıllı makinelerin gerçek hayatı algılamasındaki en yaygın aracı olarak kameralar kullanılmaktadır. Kameralar tarafından algılanan görüntüler sayısal biçime dönüştürülerek bilgisayarlara aktarılmaktadır. Aktarılan bu sayısal görüntülerdeki bilgilerden yararlanıp makinelere gerekli komutların verilebilmesi için bu görüntülerin çözümlenmesi gerekmektedir. Görüntü çözümlenmedeki ilk işlem adımlarından birisi görüntü bölütleme işlemidir.

Görüntü bölütleme, bir görüntünün anlamlı parçalara bölünmesine denir (Gonzalez ve Woods 1992). Bölütleme işlemi görüntü çözümlenmede, örüntü tanımada, nesne algılamada, gösterimlerde ve birçok görüntü işleme uygulamalarında sıklıkla tercih edilen bir ön işlem adımıdır (Pal ve Pal, 1993; Zhang, 1996). Görüntü bölütleme işlemi günümüzde kullanıldıkları yere ve görüntünün içeriğine göre değişik tekniklerle gerçekleştirilmektedir. Bu teknikler,

- Eşikleme tabanlı bölütleme
- Kenar tabanlı bölütleme
- Bölge tabanlı bölütleme
- Kümeleme tekniği ile bölütleme
- Çakıştırma tekniği ile bölütleme

biçiminde kategorilere ayrılabilir. Gerçekte iyi bir görüntü bölütlemesi yapmak için görüntü içerisindeki nesnelere bir birlerinden ayırt edebilmek gerekir. Bu her zaman mümkün olmayabilir. Özellikle nesnelere birbirleriyle örtüşükleri zaman aynı nesne gibi algılanırlar ve nesnelere sınırlarındaki renk geçişlerinde bir uyum olması sınırın algılanmasını güçleştirmektedir.

Görüntü bölütleme bir gözetim sınıflandırma işlemidir. Bu işlem sayısal bir görüntüdeki bir nesneyi veya bölgeyi arka plandan ayırmak için kullanılır. Bu işlem görüntü tanıma işlemlerinde (Pachowicz, 1994), görüntü sıkıştırma (Belloulata ve Konrad, 2002), görüntü gösteriminde (Hartmann ve Galloway, 2000) hassas bir süreç olarak kabul edilmektedir. Görüntü bölütlenmede şimdiye kadar birçok değişik yöntem geliştirilmiştir

(Pal ve Pal, 1993). Buna rağmen şimdiye kadar tam olarak standart bir yöntem önerilememiştir (Shih 2010).

Bu çalışmada, eşikleme tabanlı bölütlemeler ile kümeleme tabanlı bölütlemeler ele alınacaktır. Eşikleme tabanlı bölütleme işleminde optimal eşik noktasının belirlenmesinde kümeleme yöntemleri tercih edilmektedir. Görüntü bölütlemeye geçmeden önce görüntü işleme ile ilgili bazı kavramlara sonraki bölümlerde ele alınmıştır.

1.1. Sayısal Görüntülerde Renkler

Renk, ışığın değişik dalga boylarının gözün retinasına ulaşması ile ortaya çıkan bir algılamadır. Bu algılama, ışığın nesnelere üzerine çarpması ve kısmen soğurulup kısmen yansması nedeniyle çeşitlilik gösterir. Bu çeşitlilik renk özütü veya renk olarak adlandırılır. Tüm dalga boyları birden aynı anda gözümüze ulaşırsa bunu beyaz, hiç ışık ulaşmazsa siyah olarak algılarız. İnsan gözü 380nm ile 780nm arasındaki dalga boylarını algılayabilir. Dolayısıyla elektromanyetik spektrumun bu bölümüne görünen ışık denir.

Renkler için genelde kulağımızla duyduğumuz ince ve kalın ses benzetmesi yapılsa da, ses algısının aksine aynı anda gelen ışık frekansları değişik kanallardan algılanamaz (başka bir deyişle göz frekans analizi yapamaz). Dolayısıyla aynı anda ince ve kalın sesleri birbirine karıştırmadan duymamıza karşın gözümüz için bu “çok seslilik” söz konusu olmadığından değişik ışık frekanslarının sadece birleşimlerini algılayabiliriz. Bu ilkeyi açıklamak veya uygulamalarda kullanmak için çeşitli renk modelleri (uzayları) geliştirilmiştir (URL-1, 2014).

Tablo 1. Işık dalga boylarına ve frekanslarına göre renkleri

Renk	Dalgaboyu	Frekans
Kırmızı	~ 625-740 nm	~ 480-405 THz
Turuncu	~ 590-625 nm	~ 510-480 THz
Sarı	~ 565-590 nm	~ 530-510 THz
Yeşil	~ 500-565 nm	~ 600-530 THz
Mavi	~ 450-485 nm	~ 680-620 THz
Mor	~ 380-440 nm	~ 790-680 THz

1.2. Renk Sayısına Göre Görüntü Çeşitleri

Gözet tabanlı bir görüntüde olabilecek renk sayısı (renk paleti), görüntünün renk derinlik bilgisi ile ilgilidir. İkili derinlikli bir görüntüde bir gözet ya 1 ya da 0 değerini alabilir. Böyle bir gözetin alabileceği renklerin sayısı 2 dir. 8 ikili derinliğinde olan bir görüntü $2^8 = 256$ farklı renk alabilir. Bu durumda 1 gözet için ayrılan bellek alanı 1 Byte olacaktır. 2×2 'lik ve 8 ikili derinliğindeki bir görüntü 4 Byte, 3×2 gözet boyutunda ve 8 ikili derinliğindeki bir görüntü 6 Byte belleğe gereksinim duyar.

Bir gözet için ayrılan 8 ikili farklı biçimlerde kullanılabilir. Örneğin siyahtan beyaza kadar bir ölçeği gösteriyorsa 8 ikili 256 çeşit gri ölçeğe bir görüntü elde edilmiş olur. Eğer 256 değer her biri için görüntü içerisinde sık kullanılan gerçeğe yakın renklerden biri örnek değer olarak seçilmişse, bu durumda indekslenmiş renk ölçeğinden söz edebiliriz. Gerçeğe yakın görünmesine rağmen bazı renkler ölçeğe bulunan en yakın renk ile değiştirilmiştir. Böyle bir kayıp, ayrılan ikili sayısının azlığından doğmuş olup, her gözeğe daha fazla bellek ayrılarak aşılabilmektedir. Renklerin gerçek yaşamda olduğu gibi görünmesi için kırmızı, yeşil ve mavi renk kanallarına birer Byte (8 ikili) ayırmak gerekir. Böyle bir görüntünün derinliği (3 Byte) 24 ikilidir. 24 ikilik görüntünün bulundurabileceği renk sayısı 2^{24} yani 16'777'216 (16 Milyon) renktir. Böyle bir görüntüde 2×2 gözet boyundaki bir görüntü 12 Byte bellek alanına gereksinim duyar. 2×3 boyutundaki bir görüntü 18 Byte bellek alanı gerektirir. Matbaalar için gereken CMYK görüntülerde RGB sistemi kullanılmadığından bulundurulabilen renk sayısı daha az olmasına rağmen gözet başına gereken bellek 4 Byte'dır. Alfa kanalı adı verilen ve genelde maskeleyme amaçlı kullanılan her ilave kanal için bellekte gözet başına 1 Byte eklemek gerekir. Katman adı verilen ve RGB görüntülerin tamamını kapsayan katmanların her biri için gözet başına 3 Byte bellek kullanmak gerekir. Yukarıda sözü edilen tüm renklere göre görüntü çeşitleri Tablo 1'de özetlenmiştir.

Tablo 2. Gözet derinliğine göre görüntü çeşitleri

Gözet Derinliği	Gözet başına Byte sayısı	Renk Sayısı	
1 İkil	1/8 Byte	2^1	İki Renk; Siyah-Beyaz
4 İkil	1/2 Byte	2^4	16 Renk; Dizinli Renk
8 İkil	1 Byte	2^8	256 Renk; Dizinli veya Gri Ölçekli
16 İkil	2 Byte	2^{16}	65'536 Renk, Çoklu Renk
24 İkil	3 Byte	2^{24}	16'777'216 Renk, RGB Gerçek Renk
32 İkil	4 Byte	2^{32}	16'777'216 Renk, ARGB Gerçek Renk A bileşeni saydamlığı göstermektedir.

1.3. Renk Uzayları

Renk uzayları toplamsal ve çıkarımsal renk düzenekleri olarak iki ayrı ilkeye dayanır. Toplamsal ile kastedilen değişik dalga boylarındaki ışıkların birleşerek gözümüze ulaşmasıdır. Doğada ışığı yalnızca nesnelere yansıdığı şekliyle gördüğümüzden pratik olarak gözlenmesi güçtür, ancak değişik renkteki ışık kaynaklarını bir duvar üzerine yansıtarak veya bilgisayar ekranlarında olduğu gibi aynı noktadan değişik frekansta ayrı ışıklar yayarak gözlemlenebilir. Çıkarımsal düzenekler ise ışık frekanslarının aynı anda toplanarak değil, birbirlerini engellemesi, süzmesi sonucunda gözlemlenir. Boya veya mürekkepleri karıştırılırken kullanılır.

Her donanım üreticisi, görüntüleme için kullanılmak üzere ürettikleri donanım birimleri için kendisine özgü bir renk-uzayı tasarlamıştır. Birçok nedenden dolayı farklı görüntü işleme düzenekleri farklı renk uzayları kullanırlar. Baskı işlerinde renkli bir görüntü basılırken CMY renk uzayı kullanılırken, görüntüleme için renkli CRT ekranları ve birçok bilgisayar grafik düzenekleri RGB renk uzayı kullanırlar. Renk özü, doygunluk ve parlaklık seçeneklerini ayrı ayrı kullanabilen düzeneklerde HSI, HSL (HSB) ve HSV renk uzayları olarak sıralanabilir.

İnsanlar renkleri, üç ayrı koninin işlevi şeklinde algırlar. Çünkü renkler üç ana rengin üzerine kurulmuş bir düzenektir. Bu renkler değerlerine üçüz değerler de denir. Bu çalışmada, RGB, CMY ve HSI renk uzayları incelenecektir.

Yukarıda sözü edilen bu üçüz değerlerin dışında birçok renk uzayı bulunmaktadır. Örneğin YIQ renk uzayı televizyon yayınlarında kullanılır, XYZ uzayı fiziksel bir

büyüklik değil, renk ölçünü olarak kullanılır. XYZ uzayı basit düzey işlemleri ile diğer renk uzaylarına kolayca çevrilebilir. Diğer renk uzayları Lab, YUV ve UVW'dur [].

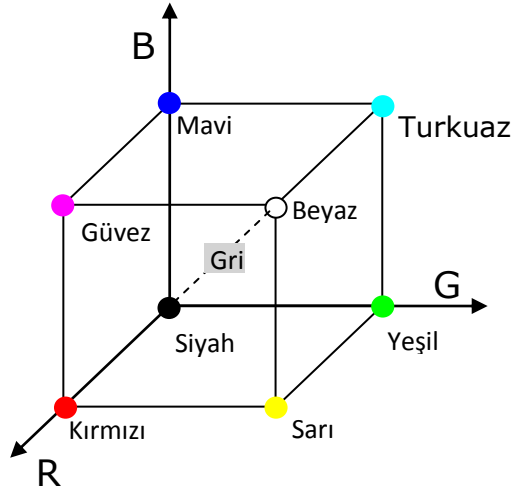
Bu çalışmada tüm renk uzaylarının 0 ile 1 arasında görelî durumu üzerinden işleme sokulacağı varsayılmıştır. Bu işlemi gerçekleştirmek için tüm değerler en büyük değere bölünerek yapılabilir. Örneğin 8-ikillik bir renk 255'e bölünerek birim değere indirgenebilir.

1.3.1. RGB Renk Uzayı

RGB (Red, Green, Blue) sırasıyla kırmızı, yeşil ve mavinin değişik oranlarda birleştirilmesiyle elde edilmektedir. Bu renk uzayı en çok bilinen ve en çok kullanılan renk uzayı olarak karşımıza çıkmakta ve giderek kullanımı artmaktadır. Bu üç renk ışık yayılım ortamında ana renkler olarak bilinmektedir. Farklı dalga boylarındaki bu ışık renk birleşmelerinin değişik oranlarda birleşerek ışığın olmadığı bir ortama (siyah renge) eklenmesiyle diğer renklerin oluşması sağlanır.

Bu renk uzayında her ana renk 0 ile 255 arasında tam değerler ile temsil edilmektedir. RGB renk düzeneği 3-boyutlu renk uzayında kırmızı, yeşil ve mavi ile her eksenin birim yöneyini oluşturmaktadır. Bu şekilde oluşan birim küpün başlangıç noktası siyah ve bunun tam karşıtı ise beyaz olmaktadır (Şekil 1). Siyahtan beyaza doğru köşegen boyunca renk değişimi gri renk ölçeğini göstermektedir. 24-ikillik renkli görüntülerde her renk birleşeni 8-ikille gösterilir. Örneğin kırmızı rengi (255,0,0) şeklinde bir renk yöneyi olarak gösterilir. Bu ise birim renk küpünde (1,0,0)'a karşılık gelir.

RGB renk uzayı bilgisayar grafikleri için kullanılan temel renk düzeni olmasına rağmen her uygulamada uygulanabilirliği pek uygun olmayabilir. Kırmızı, yeşil ve mavi renkler birbiriyle çok uyumlu olmalarına rağmen bazı görüntü işleme algoritmalarında zorluklar çıkarmaktadır. Birçok görüntü işleme teknikleri içerisinde, örnek olarak sıklık eşitlemesi işlemi yalnızca görüntünün parlaklık değerleri ile yapılabilir. Bunun için farklı renk uzayında çalışmak kolaylık sağlamaktadır.



Şekil 1. Kübik (RGB) renk uzayı

1.3.2. CMY/CMYK Renk Uzayı

CMY (Cyan, Magenta, Yellow) sırasıyla turkuaz, güvez ve sarının değişik oranlarda birleştirilmesiyle elde edilir. RGB sistemine göre ara renk olarak kabul edilen bu renkler basım işlemlerinde genellikle kullanılan bir renk ayrımıdır. CMY renkleri RGB renklerinin tamlayıcıları olarak adlandırılmaktadır. Öyle ki turkuaz, güvez ve sarı renkleri sırasıyla kırmızı, yeşil ve mavinin bütünleyenleridir. Turkuaz, güvez ve sarı renkler beyaz renkten ana renklerin tek tek çıkarılmasıyla elde edilmektedir. Onun için bu renklere ara renkler de denilmektedir. Öyle ki, turkuaz kırmızıyı soğururken, güvez yeşili ve sarı ise maviyi soğurmaktadır. Örneğin yeşil renginin değerinin artırılması için turkuaz ve sarı renk değerlerinin artmasıyla sağlanabildiği gibi güvez renginin azaltılmasıyla da sağlanabilir.

RGB ve CMY birbirini bütünleyen düzenekler olduğu için, bu iki renk uzayı arasında dönüşüm yapmak oldukça kolaydır. Öyle ki, RGB den CMY 'a geçmek için beyazdan bütünleyeni çıkarmak yeterlidir,

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = L - \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (1)$$

burada L değeri en büyük değerli renk bileşenidir. Aynı şekilde CMY'den RGB'ye geçmek için ise,

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = L - \begin{bmatrix} C \\ M \\ Y \end{bmatrix} \quad (2)$$

eşitliği kullanılmaktadır. (2) eşitliğine göre 24 ikillik bir renk düzeninde turkuaz rengi 255-kırmızı rengine eşit olmaktadır.

Baskı alanında CMY renk uzayına dördüncü bir renk daha eklenmektedir. Bu üç renk: turkuaz, mor ve sarı +siyah renk baskı renkleri olarak bilinmektedir ve CMYK olarak gösterilmektedir. K burada siyah rengini göstermektedir. Siyah rengi CMY renk düzenine eklenmesinin nedeni tam duru bir siyah renk elde etmekte karşılaşılan güçlükleri yok etmektir. Bu şekilde daha duru bir görüntü elde edilmektedir. Aynı zamanda siyah renkli boyalar renkli boyalara göre daha ucuz olmaktadır.

CMY renk düzeninden CMYK renk düzenine geçiş için aşağıdaki eşitlikte verildiği gibi siyah renk değeri, bulunur,

$$K = \min(C, M, Y) \quad (3)$$

biçiminde bulunur. Burada *min* renkler değerlerin en küçüğünü bulan fonksiyondur. (1) ve (3) eşitliğinden

$$\begin{bmatrix} C \\ M \\ Y \\ 0 \end{bmatrix} = \begin{bmatrix} C' \\ M' \\ Y' \\ K \end{bmatrix} - K \quad (4)$$

eşitliği elde edilebilir. Bu eşitlikte C' , M' ve Y' bileşenleri CMYK uzayındaki renk değerlerdir. Bu eşitlik yardımıyla CMYK düzeninden CMY geçmek için her bileşene siyah (K) değeri eklenerek gerçekleştirilir.

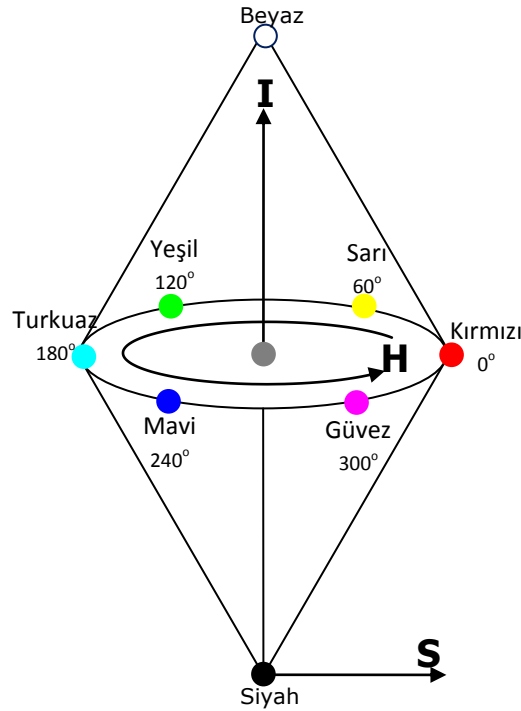
1.3.3. HSI Renk Uzayı

HSI (Hue, Saturation, Intensity) renk uzayı daha çok Amerikan görüntüleme ölçünü olarak bilenen NTSC düzenin renk bileşenleridir. Daha çok televizyon düzeneklerinde kullanılan bu renk bileşenleri sırasıyla renk özü (H), doygunluk (S) ve parlaklık (I) değerlerini göstermektedir. Bu üç özellik kullanılarak renk tanımlanabilir. HSI renk uzayı

kullanıldığında, kırmızı, yeşil ve mavinin renk oranlarının bilinmesine gerek yoktur. Kısaca renk özü ile istenilen renk değeri ayarlanabilir. Kırmızı rengini pembe rengine çevirmek için doygunluk bileşenini ayarlamak yeterlidir. Karartma ve aydınlatma için ise parlaklık bileşeni kullanılır.

Görüntü ile ilgili birçok uygulama, HSI renk uzayı kullanılarak gerçekleştirilir. Örneğin bilgisayarlı görmede farklı nesnelere renkleri tanımlamak için HSI renk uzayı kullanılır. Görüntü işleme uygulamalarında, örneğin sıklık işlemlerinde, yoğunluk dönüşümlerinde ve evrişim gibi işlemlerde yalnız görüntünün parlaklığı kullanılmaktadır.

HSI renk uzayı silindirik koordinata benzetilmektedir. Birbiri üzerine çevrilmiş iki koni şeklinde düşünülebilir (Şekil 2) (Crane 1996).

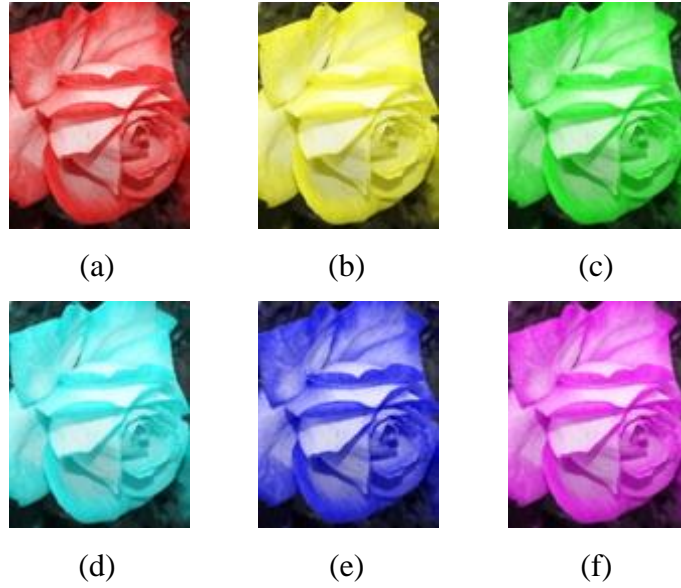


Şekil 2. Konik (HSI) renk uzayı

Renk özü koninin çevresini açısal bir değişimle gösterir ve 0° ile 360° arasında değişimle ifade edilir. Çoğu uygulamada bu değer 0 ile 1 arasına indirildiği de görülmektedir. Doyma koninin taban alanının yarıçapını gösterir ve 0 dan 1'e kadar değişen değerler alabilir. Parlaklık ise düşey eksenini gösterirken 0 siyah, 1 ise beyaz olacak şekilde düzenlenmiştir. $S = 0$ olursa, renk gri yoğunluğuna göre değişim gösterir. $S = 1$ ise renkler dış sınırlarda olur ve renkleri oldukça canlı gösterir.

Renk özü farklı renklere göre ayarlanmasıyla, 0° 'de kırmızı, 120° 'de yeşil, 240° 'de mavi ve 360° de tekrar kırmızıya geri dönmektedir. $I = 0$ olduğunda renk siyah olduğu için H tanımsızdır. $S = 0$ da ise renk gri yoğunluğuna göre değişmektedir. Bu durumda da H tanımsızdır.

Bu çalışmada daha çok renk özü üzerinde durulacağından H değişiminin bir görüntüdeki etkisi Şekil 3'te verilmiştir. Şekil 3(a) özgün bir görüntü verilmiştir. Özgün görüntüye 60° , 120° , 180° , 240° ve 300° lik bir eklemeye nasıl farklı renkler aldığı Şekil 3(b-f) görülmektedir.



Şekil 3. Renk özü (H) değerinin değiştirilmesiyle görüntüdeki değişim; (a) H ; (b) $H + 60^\circ$; (c) $H + 120^\circ$; (d) $H + 180^\circ$; (e) $H + 240^\circ$; (f) $H + 300^\circ$

1.3.4. RGB Renk Uzayından HSI Renk Uzayına Geçiş

(5-7) eşitlikleriyle RGB uzayından bir gözeğin HSI uzayına nasıl çevrildiği gösterilmektedir. HSI renk uzayındaki parlaklık (yoğunluk, I) değeri,

$$I = \frac{R + G + B}{3L} \quad (5)$$

biçiminde hesaplanır. Burada L , RGB renk uzayındaki en büyük renk değerini göstermektedir. 24 ikilik 3 kanallı renkli bir görüntüde L değeri 255'dir. HSI uzayının diğer bir değişkeni olan doygunluk değeri (S),

$$S = 1 - \left(\frac{3 \min(R, G, B)}{R + G + B} \right) \quad (6)$$

biçiminde hesaplanır. Buradaki \min fonksiyonu RGB uzayındaki renk değerlerinin en küçüğünü verir. HSI renk uzayında en önemli değişkeni ise renk özü (H) değişkenidir. Bu değişken gözeğin hangi renkte olacağını belirler. Bu değişkenin en önemli özelliği dairesel bir değişken olmasıdır.

$$H = \text{Cos}^{-1} \left[\frac{\frac{1}{2} [(R - G)(R - B)]}{\sqrt{(R - G)^2 + (R - G)(G - B)}} \right] \quad (7)$$

biçiminde hesaplanır. Burada H değeri açısal bir değişken olduğundan Cos^{-1} fonksiyonu $[0,180)$ aralığında bir değer verir. Bu değişken $[0,360)$ değer aralığına çekilmesi gerekir.

1.3.5. HSI Renk Uzayından RGB Renk Uzayına Geçiş

HSI renk uzayından RGB renk uzayına dönüşüm yapılırken öncelikle RGB renk değerlerinin birim uzaydaki karşılıkları bulunur. Daha sonra RGB uzayına çevrilir. Eğer B , G den büyükse, $H = 360^\circ - H$ dönüşümü gerçekleştirilir.

RG bölgesi ($0^\circ \leq H \leq 120^\circ$) için normalize edilmiş mavi renk (B') bileşeni,

$$B' = \frac{1}{3} (1 - S) \quad (8)$$

biçiminde hesaplanır. Normalize edilmiş kırmızı renk (R') bileşeni,

$$R' = \frac{1}{3} \left[1 + \frac{S \cos(H)}{\cos(60^\circ - H)} \right] \quad (9)$$

biçiminde hesaplanır. Normalize edilmiş yeşil renk (G') bileşeni,

$$G' = 1 - (R' + B') \quad (10)$$

biçiminde hesaplanır.

GB bölgesi ($120^\circ \leq H \leq 240^\circ$) için, renk özü (H)

$$H = H - 120^\circ \quad (11)$$

biçiminde hesaplanır. Normalize edilmiş yeşil renk (G') bileşeni,

$$G' = \frac{1}{3} \left[1 + \frac{Scos(H)}{\cos(60^\circ - H)} \right] \quad (12)$$

biçiminde hesaplanır. Normalize edilmiş kırmızı renk (R') bileşeni,

$$R' = \frac{1}{3} (1 - S) \quad (13)$$

biçiminde hesaplanır. normalize edilmiş mavi renk (B') bileşeni,

$$B' = 1 - (R' + G') \quad (14)$$

biçiminde hesaplanır.

BR bölgesi ($240^\circ \leq H \leq 360^\circ$) için, renk özü (H)

$$H = H - 240^\circ \quad (15)$$

biçiminde hesaplanır. Normalize edilmiş mavi renk (B') bileşeni,

$$B' = \frac{1}{3} \left[1 + \frac{Scos(H)}{\cos(60^\circ - H)} \right] \quad (16)$$

biçiminde hesaplanır. Normalize edilmiş yeşil renk (G') bileşeni,

$$G' = \frac{1}{3}(1 - S) \quad (17)$$

biçiminde hesaplanır. Normalize edilmiş kırmızı renk (R') bileşeni,

$$R' = 1 - (G' + B') \quad (18)$$

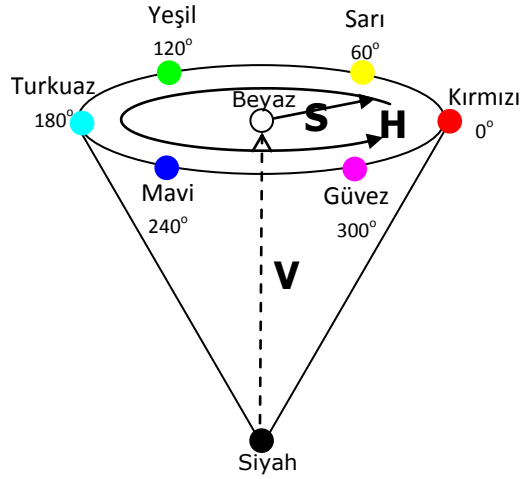
biçiminde hesaplanır. Bu eşitliklerde R' , G' ve B' değerleri R, G ve B değerlerinin birim değerleridir. Bu değerleri R, G ve B değerlerine çevirmek için:

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = L \begin{bmatrix} R' \\ G' \\ B' \end{bmatrix} \quad (19)$$

eşitliği verilmektedir (Crane 1996).

1.3.6. HSV Renk Uzayı

HSB veya HSV olarak da anılan bu renk uzayı tek koniden oluşan bir konum düzeneği bulunmaktadır. Şekil 2-8'de bu uzay için koni düzeneği görüntüsü verilmektedir. Konide H renk ayrınç değerini, S doygunluk katsayısını, V ise parlaklık değerini vermektedir. H değerinin artması renklerin değişmesini sağlar. Öyle ki, bir renk kıızıdan sarıya, sarıdan yeşile, yeşilde gökçeye, gökçeden göke, gökten güveze dönüşebilmektedir.



Şekil 4. Konik (HSV) renk uzayı

1.3.7. RGB'den HSV Geçiş

RGB renk uzayındaki bir gözeği HSV renk uzayına dönüştürmek için öncelikle RGB renk bileşenlerinin doygunluk değerinin belirlenmesi gerekir. Bu renk bileşenleri $R, G, B \in [0, C_{enbüyük}]$ aralığında tanımlıdır. 24 ikilik bir görüntüde $C_{enbüyük}$ genelde 255 olarak verilmektedir. Doygunluk değeri,

$$S = \begin{cases} \frac{C_{aralık}}{C_{yüksek}}, & C_{yüksek} > 0 \\ 0, & \text{diğer durumlarda} \end{cases} \quad (20)$$

biçiminde verilirken, parlaklık değeri,

$$V = \frac{C_{yüksek}}{C_{enbüyük}} \quad (21)$$

biçiminde hesaplanır. Bu eşitliklerde $C_{yüksek}$ değeri,

$$C_{yüksek} = \max(R, G, B) \quad (22)$$

$C_{düşük}$ değeri,

$$C_{düşük} = \min(R, G, B) \quad (23)$$

ve $C_{aralık}$ değeri ise,

$$C_{aralık} = C_{yüksek} - C_{düşük} \quad (24)$$

biçiminde verilmektedir.

Son olarak Renk özü değeri hesaplanırsa bazı özel durumların oluşması söz konusu olur. Bu özel durum tüm renk bileşenlerinin bir birine eşit ($R = G = B$) olması durumudur bu durumda görüntü gri ölçekli bir görüntüdür ve $C_{aralık}$ değeri sıfır olmaktadır. Bu ise $S = 0$ değerine neden olduğundan H değeri bu durumda tanımsız olur. Dolayısıyla H değerinin hesaplanabilmesi için $C_{aralık} > 0$ olmalıdır. Öncelikle RGB renk uzayındaki tüm renk bileşenlerinin normalize edilmesi gerekir. Bu eşitlikler, normalize edilmiş kırmızı renk (R') bileşeni,

$$R' = \frac{(C_{yüksek} - R)}{C_{aralık}} \quad (25)$$

biçiminde hesaplanır. Normalize edilmiş yeşil renk (G') bileşeni,

$$G' = \frac{C_{yüksek} - G}{C_{aralık}} \quad (26)$$

biçiminde hesaplanır. Normalize edilmiş mavi renk (B') bileşeni,

$$B' = \frac{C_{yüksek} - B}{C_{aralık}} \quad (27)$$

biçiminde verilmektedir. Bu bileşenlere göre, Renk özü (H) değeri her renk bileşenin büyük olması durumuna göre tanımlanırsa,

$$H' = \begin{cases} B' - G', & R = C_{yüksek} \\ R' - B' + 2, & G = C_{yüksek} \\ G' - R' + 4, & B = C_{yüksek} \end{cases} \quad (28)$$

biçiminde verilir. Buradaki H' değeri $[-1,5]$ aralığında bir değer alır. Bu değeri $[0,1]$ aralığına indirgemek için,

$$H = \frac{1}{6} \begin{cases} (H' + 6), & H' < 0 \\ H', & \text{diğer durumlarda} \end{cases} \quad (29)$$

eşitliği kullanılır.

Tüm hesaplamalar sonucunda H , S ve V değerleri $[0,1]$ aralığında bir değer alır. Bu değerlerden Renk özü (H) açısıl bir değişken olduğundan derece cinsine çevirmek için

$$H^\circ = H \cdot 360 \quad (30)$$

eşitliği kullanılır.

1.3.8. HSV'den RGB'ye Geçiş

HSV renk uzayındaki (H, S, V) bileşenleri $[0,1]$ aralığında tanımlandığı kabul edilir. Bu bileşenler (R, G, B) bileşenlerine çevirmek için

$$H' = (6 \cdot H) \bmod 6$$

eşitliği kullanılır. Burada $H' \in [0,6)$ aralığında bulunmaktadır. Bu değer yardımıyla aşağıdaki ara değişkenler tanımlanır. Bu ara değişkenler,

$$c_1 = [H'] \quad (31)$$

$$v = V \quad (32)$$

$$x = (1 - S) \cdot v \quad (33)$$

$$c_2 = H' - c_1 \quad (34)$$

$$y = (1 - (S - c_2)) \cdot v \quad (35)$$

$$z = (1 - (S \cdot (1 - c_2))) \cdot v \quad (36)$$

biçiminde hesaplanmaktadır. Buradaki c_1 değişkenin değerine göre normalize edilmiş (R', G', B') parameterleri hesaplanır. Bu hesaplama,

$$(R', G', B') = \begin{cases} (v, z, x) & c_1 = 0 \\ (y, v, x) & c_1 = 1 \\ (x, v, z) & c_1 = 2 \\ (x, y, v) & c_1 = 3 \\ (z, x, v) & c_1 = 4 \\ (v, x, y) & c_1 = 5 \end{cases} \quad (37)$$

biçiminde yapılmaktadır. Bu bileşenleri $[0,255]$ aralığına çekmek için,

$$R = 255 \cdot R' \quad (38)$$

$$G = 255 \cdot G' \quad (39)$$

$$B = 255 \cdot B' \quad (40)$$

eşitlikleri kullanılır (Burger ve Burge 2009).

1.4. Sayısal Görüntüde Histogram

Sayısal görüntüler genelde tek bileşenli gri seviyeli veya 3 bileşenli renkli görüntüler biçiminde olmaktadır. Tek bileşenli gri seviyeli görüntülerin tek bir histogramı olurken, üç bileşenli görüntülerin birden çok histogramı olmaktadır.

Görüntü sıklığı bir görüntünün renk yoğunluğunu göstermek için kullanılan değerli bir araçtır. Sıklık görüntüde bir araya gelen renk karışımlarını verir. Bu ise bir görüntüde yoğunluk dağılımını ve ayrımlılık bilgisini elde etmemizi sağlar. Görüntüdeki her bir renk değerinin sıklığını bulmak için (41) eşitliği kullanılmaktadır.

$$h(r) = \sum_{x=0}^{W-1} \sum_{y=0}^{H-1} \delta(g(x,y) - r), \quad (r = 0,1,2, \dots, L) \quad (41)$$

burada, $\delta(*)$ fonksiyonu,

$$\delta(t) = \begin{cases} 1, & t = 0 \\ 0, & t \neq 0 \end{cases}$$

biçiminde gösterilen dirac fonksiyonudur. Diğer parameterler,

L	en büyük renk değerini (255),
r	renk indisini,
W	incelenen görüntünün gözek cinsinden genişliğini,
H	incelenen görüntünün gözek cinsinden yüksekliğini,
$g(x,y)$	incelenen görüntü bileşenini,
$h(r)$	incelenen görüntü bileşeninin sıklıklarını

vermektedir.

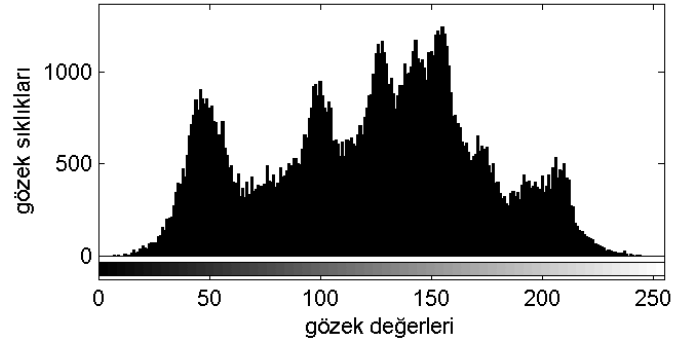
Histogram basitçe gözek yoğunluğunun çubuk grafik olarak gösterilmesidir. Gözek yoğunluğu x-ekseni boyunca çizilirken o gözeğe ait yoğunluk sayısı ise y-ekseni olarak verilmektedir.

Karanlık görüntüler gözek dağılımı sola dayalı histograma sahipken, aydınlık görüntüler gözek dağılımı sağa dayalı histograma sahiptir. Ayrımlılık olarak iyi bir görüntüde, histogramın düzgün dağılıma sahip olması beklenir.

Şekil 5'te Lena'nın görüntüsü ve bu görüntüden elde edilen sıklıkların histogramı görülmektedir. Bu histogramda 0'a yakın renk değerleri siyahı gösterirken 255'e yakın değerleri beyaz rengini göstermektedir.



(a)



(b)

Şekil 5. Histogram görüntüleme; (a) Örnek Lena görüntüsü; (b) Lena görüntüsünün histogramı

1.5. Görüntü Eşikleme

Eşikleme görüntü çözümleme ilk adımlardan birisidir. 24 ikilik renkli bir görüntüde 16 milyondan fazla renk bulunmaktadır. bu renklerin organize edilmesi ve bunlardan sonuç çıkarmak oldukça güç olmaktadır. Dolayısıyla daha düşük renk bileşenlerine sahip bir görüntüyü çözümlemek oldukça kolay olacağından bu yola başvurulmaktadır. Eşikleme işlemi kullanıcı destekli yapılabildiği gibi otomatik olarak da yapılabilmektedir. Eşikleme işlemi sadece görüntü çözümlemenin yanı sıra optimal renk indirgemesi yaparak görüntü sıkıştırma veya indeksli görüntü elde etmek için de kullanılmaktadır.

1.5.1. Eşikleme İşlemi

Eşikleme işlemi gri ölçekteki bir görüntü üzerinde uygulanmaktadır. Ancak renkli görüntülerde uygulandığı zaman her renk bileşeni kendi arasında eşikleme işlemine tabi tutulur. Eşikleme işlem sonucunda görüntü verilen renk indekslerine indirgenmektedir. Eğer iki seviyeli bir eşikleme yapılacaksa bu görüntünün eşikleme işlemi sonucunda iki renkten oluşmasına neden olur. Bu renkler ise genelde 8 ikilik bir görüntüde 0 ve 255'tir. Bu şekilde eşiklenmiş görüntülerde sadece siyah (0) ve beyaz (255) renkleri bulunmaktadır. Çoğu zaman bu renkler 0 ve 1 değerlerine indirgenerek mantıksal birer değere dönüştürülebilmektedir. Bu ise hesaplama işlemlerinde hem bellek kullanımı hem de hesaplama zamanında kolaylıklar sağlamaktadır.

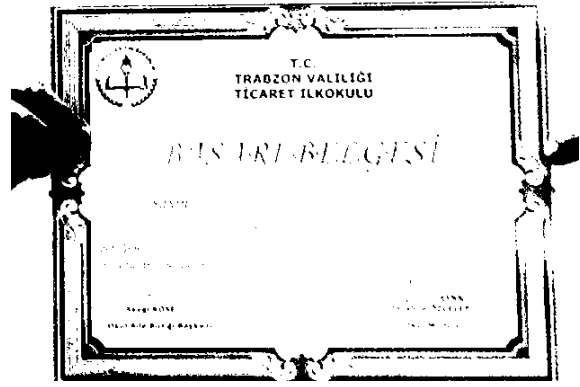
Kullanıcı destekli eşikleme işleminde bir eşik değeri (T) seçilerek bu değerden küçük olan tüm renk değerleri 0'a büyük olan renk değerleri ise 1'e eşiklenmesi,

$$g_{out}(x, y) = \begin{cases} 0 & g_{in}(x, y) < T \\ 1 & g_{in}(x, y) \geq T \end{cases} \quad (42)$$

biçiminde gösterilmektedir. Burada $g_{in}(x, y)$ değeri (x, y) görüntü koordinatlarında bulunan ve işleme sokulacak gözelek değeridir. $g_{out}(x, y)$ ise işlem sonucunda (x, y) noktasındaki eşiklenmiş değeri göstermektedir. Elde edilen yeni görüntü siyah-beyaz renklerden oluşan bir ikil görüntüdür (Şekil 6(b)). Bu ikil görüntüde genelde siyah renkler arka planı gösterirken, beyaz renkler nesnelere göstermektedir. Bunun tersi olduğu durumlarda söz konusu olmaktadır. Eğer nesne tanıma işlemleri yapılacaksa görüntü renksel olarak ters çevrilmektedir.



(a)



(b)

Şekil 6. İki seviyeli görüntü eşikleme; (a) Özgün görüntü; (b) Eşiklenmiş görüntü.

Eşikleme işlemi gerçekleştirilirken en önemli kısım eşik değerinin iyi seçilmesidir. Eğer eşik değeri gereğinden düşük seçilirse görüntüde beyaz gözeklerin sayısı artar (Şekil 7(a)), eğer gereğinden yüksek seçilirse de siyah gözeklerin sayısı artar (Şekil 7(b)).



(a)

(b)

Şekil 7. Şekil 6’teki özgün görüntüye göre eşik değerinin doğru seçilmesi; (a) Eşik değerinin ($T = 0.3$) düşük seçilmesi; (b) Eşik değerinin yüksek seçilmesi ($T = 0.7$).

Eşikleme işleminde kullanıcı destekli eşikleme işlemi yapılırken görsel sonuçlar karşılaştırılarak en iyi eşik değeri belirlenebilir. Ancak görüntü sayısının fazla olması durumunda veya otomatik sistemlerde bu işlem yapmak imkansız olmaktadır. Bir kaç görüntü için uygun bir T değeri bulunabilse de, görüntü sayısı fazla olduğunda, bu söz konusu olamaz. Dolayısıyla optimal bir eşik değerinin bulunması için bir algoritmaya gerek duyulur.

1.5.2. Optimal Eşik Değerinin Belirlenmesi

Gri ölçekli görüntüde nesne ile arka planı en etkili biçimde ayıran optimal eşik değerinin belirlenmesi oldukça önemlidir (Huang ve Wang, 2009). Bu konuda geliştirilen en yaygın yöntem Otsu tarafından (1979) ortaya atılan algoritmadır. Bu yöntemin temel yaklaşımı gri ölçekli bir görüntünün histogramını iki sınıflara bölmektir. Bu iki sınıf bir sınırlar değeri ile belirlenir. Bu sınırlar değeri öyle seçilmelidir ki, sınıf içindeki varyans en düşük değerini verirken sınıflar arasındaki varyans en büyük değerini versin (Demirkaya ve Asyali, 2004).

Otsu algoritmasının görüntü histogramı üzerindeki uygulamak için gri ölçekteki bir görüntünün L tane seviyeden oluştuğu kabul edilir. Görüntüdeki i . seviyeki gözetlerin sayısı f_i gösterilirse tüm gözetlerin sayısı $N = f_1 + f_2 + \dots + f_L$ biçiminde bulunur. Dolayısıyla gri ölçekteki bir görüntüde i . seviyedeki bir gözetin olasılığı,

$$p_i = \frac{f_i}{N}, \quad \left(p_i \geq 0, \sum_{i=1}^L p_i = 1 \right) \quad (43)$$

biçiminde verilir. Eğer bir görüntü C_1 ve C_2 gibi iki kümeye t eşik değeri yardımıyla ayrılırsa, C_1 kümesi 0'dan t 'ye kadar tüm gözetleri içerirken C_2 kümesi $t + 1$ 'den L 'ye kadar gözetleri içermektedir. Her iki kümenin birikimli olasılık değerleri sırasıyla,

$$w_1 = \sum_{i=1}^t p_i \quad (44)$$

ve

$$w_2 = \sum_{i=t+1}^L p_i \quad (45)$$

biçiminde verilmektedir. Bu iki kümenin ortalama seviyeleri ise,

$$m_1 = \sum_{i=1}^t i \cdot \frac{p_i}{w_1} \quad (46)$$

ve

$$m_2 = \sum_{i=t+1}^L i \cdot \frac{p_i}{w_2} \quad (47)$$

biçiminde verilmektedir.

Otsu (1979) optimal eşik değerini T^* sınıflar arasındaki varyans (σ_B^2) değerini en büyük yapan eşik değeri olarak seçmiştir. Bu eşik değeri,

$$T^* = \arg \max_{1 \leq T < L} \{\sigma_B^2(t) | \sigma_B^2 = w_1(m_1 - m^T)^2 + w_2(m_2 - m^T)^2\} \quad (48)$$

biçiminde bulunabilir. Burada, $m_T = \sum_{i=1}^L i \cdot p_i$ biçiminde verilen tüm gözleklerin ortalama deęeridir.

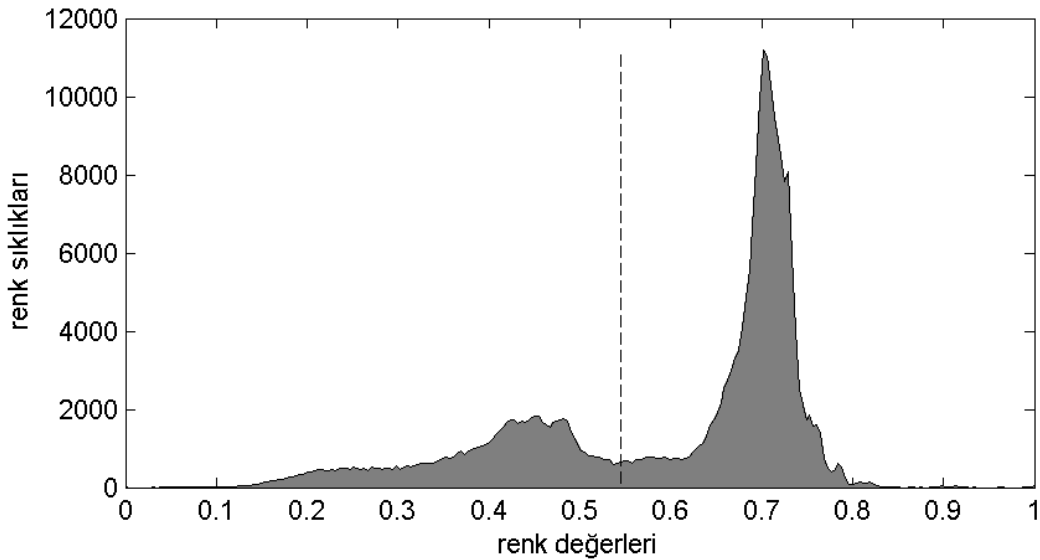
(48) eřitlięi kullanılarak Őekil 8(a)'daki grntnn her eřit deęeri iin sınıflar arası varyansının hesaplanması saęlanır. Her bir histogram elemanı iin retilen bu deęerin sonuları karřılařtırılır ve en yksek varyans deęerine karřılık gelen histogram elemanı eřit olarak belirlenir (Őekil 8(c)).

Grnt zerinde uygulanan bu iřlemin sonucu Őekil 8(b)'de grlmektedir. Eřit deęeri $T = 0.54$ seilmiř olan bu grntde, gri lekli dnřmnn sonucu olan ikili grnt, en dřk kayıpla gsterilmiřtir.



(a)

(b)



(c)

Őekil 8. İki seviyeli grnt eřitleme; (a) zgn grnt; (b) Eřitlenmiř grnt; (c) Optimal eřit deęerinin belirlenmesi

İki kümeli eşikleme işlemindeki eşitlikler çok kümeli eşikleme işlemi için rahatlıkla genişletilebilir.

1.6. Bulanık C-Ortalama Kümeleme Algoritması

Bulanık c-ortalama kümeleme (FCM) algoritması 1973 yılında Dunn tarafından ortaya atılmış ve 1981’de Bezdek tarafından geliştirilmiştir (Höppner vd., 2000). Amaç fonksiyonunu temel alan bulanık c-ortalama kümeleme algoritması bir veri kümesindeki bir elemanın birden fazla kümeye farklı üyelik dereceleriyle ait olabilmesi ilkesine dayanır. Bulanık mantığa göre bu üyelik dereceleri $[0,1]$ arasında değişen değerler almaktadır ve bir verinin tüm kümelere ait üyelik derecelerinin toplamı 1 olmalıdır. Herhangi bir küme merkezindeki bir elemanın o kümeye ait olma üyelik değeri 1’e eşitken diğer kümelere ait olma üyelik değeri sıfır olmalıdır. Nesne hangi küme merkezine yakın ise o kümeye ait olma üyeliği diğer kümelere ait olma üyeliğinden daha büyük olmalıdır.

D-boyutlu bir Öklid uzayında N örnekten oluşan bir $X = \{x_1, x_2, \dots, x_N\}$ veri kümesinin verildiğini varsayalım ($x_i \in \mathbb{R}^D$). Öbekleme, bu veri kümesinin, küme merkezleri $\{v_1, v_2, \dots, v_j, \dots, v_C\}$ olan C tane alt kümeye ayrılması işlemidir. Veri kümesini alt kümelere ayırırken istenen optimal kriter amaç fonksiyonunu minimize etmektir. Algoritma, (49) eşitliğinde verilen, en küçük kareler yönteminin genellemesi olan amaç fonksiyonunu öteleyerek minimize etmeye çalışır (Höppner vd., 2000).

$$J_m = \sum_{i=1}^N \sum_{j=1}^C \mu_{ij}^m \|x_i - v_j\|^2, \quad 1 < m < \infty \quad (49)$$

Burada m değeri bulanıklaştırma parametresidir ve genelde değeri 2 olarak seçilmektedir. u_{ij} ise i . elemanın j . kümedeki üyelik değeridir ve (50-52) eşitliklerindeki koşulları sağlamak zorundadır (Bezdek 1981).

1. Üyelik değerleri (50) eşitliğinde gösterildiği gibi $[0,1]$ aralığında olmalıdır.

$$\mu_{ij} \in [0,1], \quad \forall i, j \quad (50)$$

2. Bir veri setindeki bir elemanın tüm kümelere ait üyelik değerlerinin toplamı (51) eşitliğinde gösterildiği gibi 1'e eşit olmalıdır.

$$\sum_{j=1}^c \mu_{ij} = 1, \quad \forall i \quad (51)$$

3. Bir kümedeki tüm üyelik değerlerinin toplamı (52) eşitliğinde gösterildiği gibi eleman sayısından küçük olmalıdır.

$$0 < \sum_{i=1}^N \mu_{ij} < N, \quad \forall N \quad (52)$$

Bulanık c-ortalama kümeleme (FCM) (Bezdek vd., 1984) algoritması basit bir yöntem olmasının yanı sıra, tüm bulanık kümeleme yöntemlerinin içerisinde hala en yaygın kullanıma sahip olan bir yöntemdir.

Algoritma 1. FCM Algoritması

Adım 1. Verileri (x_i) , önceden belirlenen öbek sayısını $C \in [2, N)$, bulanıklaştırma parametresini $(m > 0)$ ve durdurma kriterini $(\varepsilon > 0)$ algoritmaya giriş parametreleri olarak ver.

Adım 2. Başlangıç üyelik değerlerini $\mu^{(0)} = [\mu_{ij}]$, $[0,1)$ aralığında düzgün dağılımdan rastgele olarak belirle, Çevrim sayacını bire $(t = 1)$ eşitle.

Adım 3. (t) . çevrimini başlat,

Adım 4. Yeni üyelik değerlerini kullanarak öbek merkezlerini (53) eşitliği yardımıyla hesapla,

$$v_j = \frac{\sum_{i=1}^N \mu_{ij}^m x_i}{\sum_{i=1}^N \mu_{ij}^m}, \quad (j = 1, 2, \dots, C) \quad (53)$$

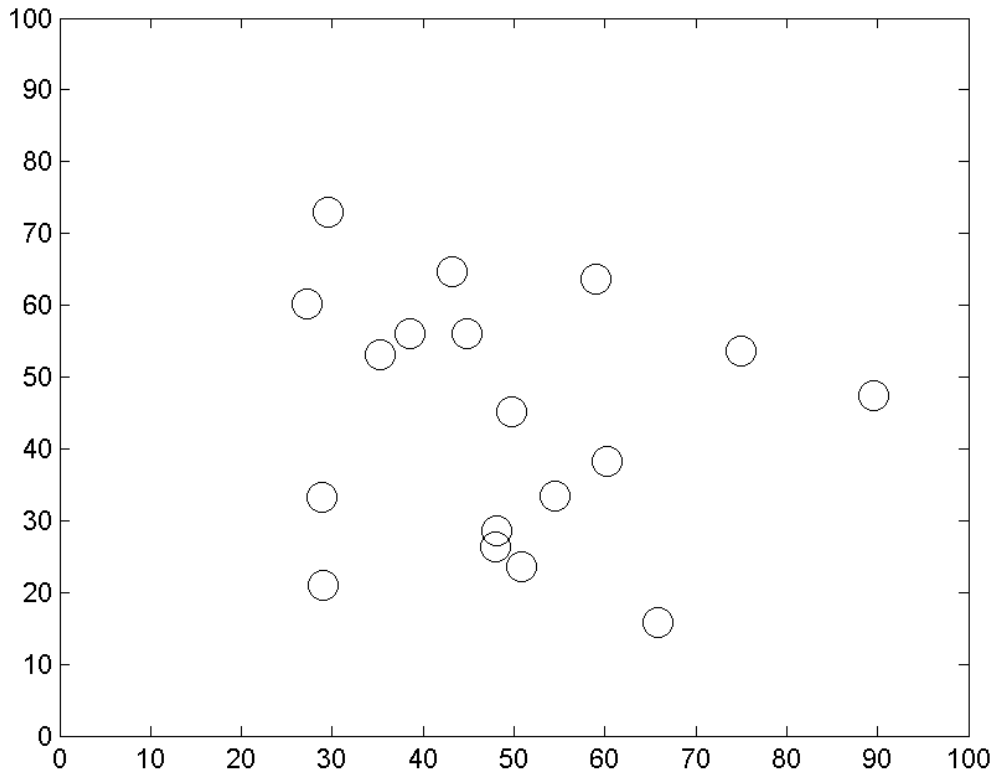
Adım 5. Üyelik değerlerini (54) eşitliği yardımıyla güncelleştir,

$$\mu_{ij} = \left(\sum_{k=1}^c \left(\frac{\|x_i - v_j\|}{\|x_i - v_k\|} \right)^{\frac{2}{m-1}} \right)^{-1} \quad (54)$$

Adım 6. (55) eşitsizliği sağlanıyorsa dur, yoksa t değerini bir artır ve Adım 3.'e giderek yeni çevrimi başlat,

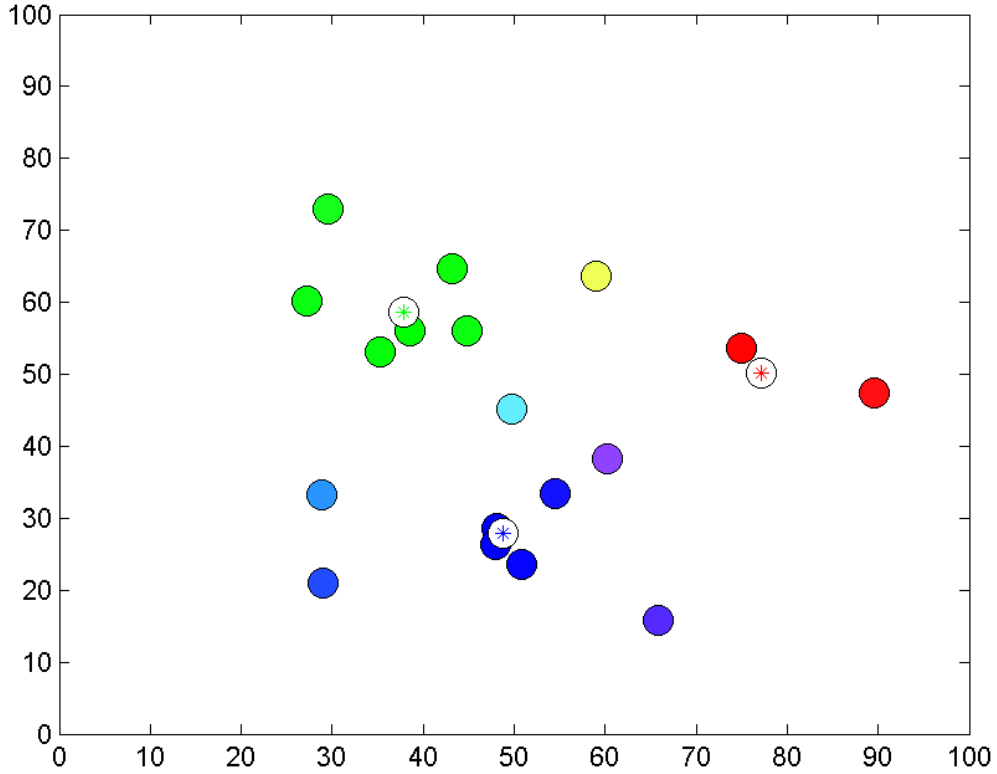
$$\|\mu^{(t)} - \mu^{(t-1)}\| < \varepsilon \quad (55)$$

Bulanık c-ortalamar algoritmasının Şekil 9'da verilen iki boyutlu 18 tane veriden oluşan veri kümesinin 3 ayrı kümeye ayrılması Şekil 10'da verilmektedir.



Şekil 9. FCM kümeleme algoritması için seçilen iki boyutlu örnek veriler

Şekil 10'da kümeler kırmızı, yeşil ve mavi renklere göre kümelendirilmiştir. Küme merkezleri siyah çerçeveli ve beyaz arka alanlı biçimde gösterilmiştir. Kırmızı küme ile yeşil küme arasında kalan nokta sarı renkte gösterilirken, mavi ile yeşil arasında kalan nokta ise turkuaz renkte gösterilmiştir.



Şekil 10 Şekil 9'da verilen örnek verilerin FCM kümeleme algoritması ile 3 kümeye ayrılması

1.7. Yönsel Veriler İçin Bulanık C-Ortalamlar Kümelemesi

Bu çalışmada kümeleme analizinde kullanılan bulanık c-ortalama kümeleme algoritması yönsel veriler (FCM4DD) için uyarlanmıştır. Bu yöntem, yönsel uzaklık ölçüsü iki açı arasındaki farkı temel alan bir yöntemdir. Bu yöntem kullandığı uzaklık ölçüsü bakımından FCM algoritması ile benzerlik göstermektedir.

Yönsel veriler $\Theta = \{\theta_1, \theta_2, \dots, \theta_N\}$ biçiminde verildiği gibi periyodik $X = \{x_1, x_2, \dots, x_i, \dots, x_N\}$ biçiminde de verilebilir. Eğer veriler periyodik değerler biçiminde verilmişse (56) eşitliği yardımıyla yönsel verilere dönüştürülebilir.

$$\theta_i = \frac{2\pi x_i}{T} \quad (56)$$

burada T , x_i değişkeninin periyodunu vermektedir.

Bu çalışmada yönsel veriler, $\theta_i \in [-\pi, \pi)$ aralığında tanımlanabileceği gibi $\theta_i \in [0, 2\pi)$ aralığında da tanımlanabilmektedir.

Önerilen kümeleme algoritması, en küçük kareler yönteminin genellemesi olan aşağıdaki amaç fonksiyonunu minimize etmek için çalışır.

$$J_m = \sum_{i=1}^N \sum_{j=1}^c \mu_{ij}^m \|\theta_i - \Phi_j\|^2, \quad 1 < m < \infty \quad (57)$$

Burada m değeri bulanıklaştırma parametresidir ve genellikle değeri iki olarak seçilir. Φ_j j . küme merkezini u_{ij} ise i . verinin j . kümeye olan üyelik derecesini göstermektedir ve μ_{ij} (49-51) eşitliklerindeki koşulları sağlamalıdır.

Yönsel veriler için önerilen yöntemin algoritması aşağıda verilmiştir (Tezel, 2014).

Algoritma 2. FCM4DD Algoritması

Adım 1. Yönsel veriler (θ_i), önceden belirlenen öbek sayısı (C), bulanıklaştırma (m) parametresi ve durdurma kriteri (ϵ) algoritmaya giriş verileri olarak ver.

Adım 2. Başlangıçtaki öbek merkezlerini $\Phi_j^{(0)}$ $[-\pi, \pi)$ aralığında düzgün dağılımdan rastgele olarak belirle ve (t) çevrim indisini 1'e eşitle.

Adım 3. (t) . çevrimi başlat,

Adım 4. Her bir verinin geçici öbek merkezine göre iki açı arasındaki farkı hesapla, bu farkı,

$$\psi_{ij} = \left(\left((\theta_i - \Phi_j) + \pi \right) \bmod 2\pi \right) - \pi \quad (58)$$

biçiminde $[-\pi, \pi)$ aralığına indirge. Bu fark yardımıyla üyelik değerlerini,

$$\mu_{ij} = \left(\sum_{k=1}^C \left(\frac{\|\Psi_{ij}\|}{\|\Psi_{ik}\|} \right)^{\frac{2}{m-1}} \right)^{-1} \quad \begin{array}{l} i = 1, 2, \dots, N \\ j = 1, 2, \dots, C \end{array} \quad (59)$$

eşitliğiyle hesapla. İki açı arasındaki fark ve yeni üyelik değerleri kullanılarak küme merkezlerini

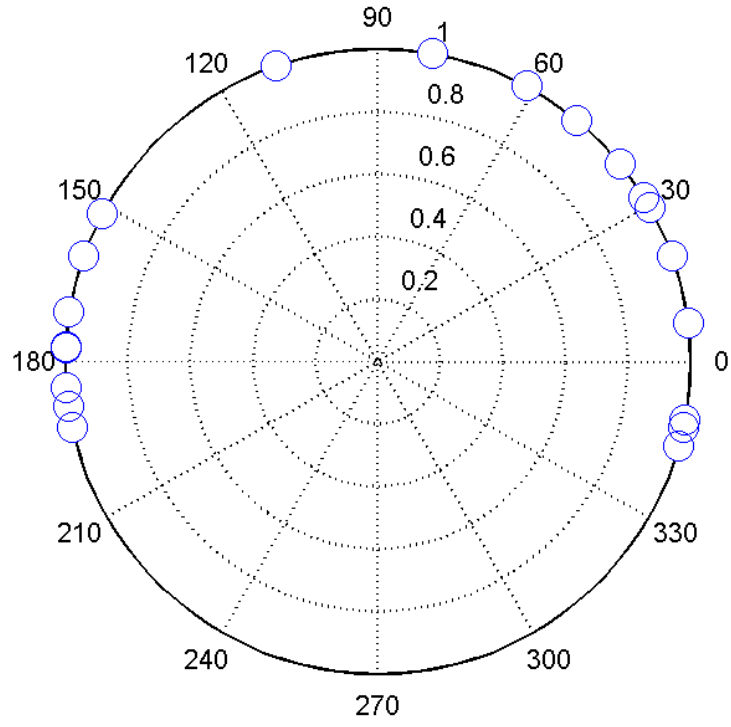
$$\phi_j^{(t+1)} = \left(\phi_j^{(t)} + \frac{\sum_{i=1}^N \mu_{ij}^m \Psi_{ij}}{\sum_{i=1}^N \mu_{ij}^m} \right) \text{mod } 2\pi, \quad (j = 1, 2, \dots, C) \quad (60)$$

biçiminde güncelle.

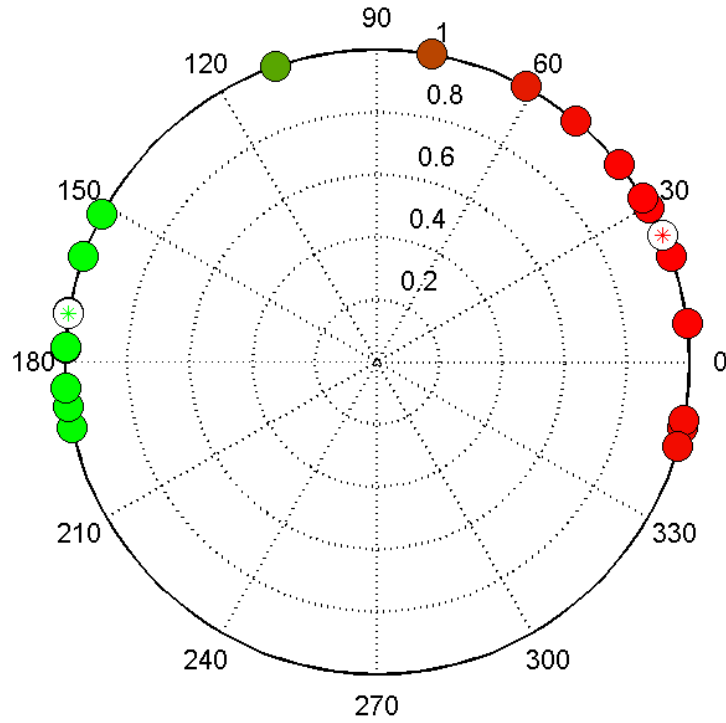
Adım 5. (61) eşitsizliği sağlanıyorsa dur, yoksa Adım 3.'e git ve (t+1). çevrimini başlat.

$$\|\mu^{(t)} - \mu^{(t-1)}\| < \varepsilon \quad (61)$$

Yönsel veriler için bulanık c-ortalamlar algoritmasının Şekil 11'de verilen dairesel 20 tane veriden oluşan veri kümesinin 2 ayrı kümeye ayrılması Şekil 12'de verilmektedir. Şekil 12'de kümeler kırmızı ve yeşil renklere göre kümelendiği görülmüştür. Küme merkezleri siyah çerçeveli ve beyaz arka alanlı biçimde gösterilmiştir. Kırmızı küme ile yeşil küme arasında kalan iki noktadan kırmızıya yakın olan bordo rengindeyken yeşile yakın olan koyu bir yeşil renkte gösterilmiştir.



Şekil 11. FCM4DD kümeleme algoritması için seçilen dairesel örnek veriler



Şekil 12 Şekil 11'de verilen örnek verilerin FCM4DD kümeleme algoritması ile 3 kümeye ayrılması

1.8. Bulanık Kümeleme Algoritmalarında Optimal Küme Sayısı

Genelde kümeleme analizinde küme sayı araştırmacı tarafından önceden belirlenmektedir. Tek ve iki boyutlu veri kümelerinde bu durum bir sorun oluşturmasa da yüksek boyutlu verilerde kullanıcının küme sayısını kestirmede karşılaşacağı güçlükleri ortadan kaldırmak için optimal küme sayısının belirlenmesi gerekir. Eğer bulunan küme sayısı gerçek küme sayısından az çıkarsa, mevcut kümelerden bir veya birkaçı birleşmek durumunda olacaktır, çok çıkarsa mevcut kümelerden bir veya birkaçı bölünmelere uğrayacaktır. Farklı başlangıç küme sayısı seçimleri, farklı kümeleneceklerin ortaya çıkmasına sebep olur. Onun için, kümeleneceklerin analizinden sonra her bir bulanık kümenin doğrulamasının yapılması gereklidir (Murat ve Şekerler, 2009). Optimal küme sayısının belirlenme işlemlerine genel olarak Küme Geçerliliği (Cluster Validity) adı verilmektedir. Bazı karmaşık yapılar içeren verilerde, küme üyeliklerindeki karasızlıklar nedeniyle, küme geçerlilik indeksleri en uygun küme sayısını belirlemede birbirleri ile çelişen sonuçlar verebilmektedir. Ayrıca hangi indeksin en uygun küme sayısını belirlediğini ortaya koyan bir ölçüt de bulunmamaktadır (Alpaslan, Erilli, Yolcu, Eğrioğlu, & Aladağ, 2011).

Genel olarak, en uygun küme sayısının belirlenebilmesi için iki kriterden bahsedilir. Bu kriterlerden ilki küme elemanlarının birbiriyle olan benzerliklerini gösteren yoğunluk ölçütü diğeri de iki kümenin birbirinden ne kadar farklılaştığını gösteren yani iki küme arasındaki uzaklığı ölçen farklılık ölçütüdür.

Küme geçerlilik indekslerinden literatürdeki en çok kullanılan Bölünme Katsayısı ve Bölünme Entropisi izleyen bölümlerde ayrıntılı biçimde verilmiştir.

1.8.1. Bölünme Katsayısı (PC)

Bölünme katsayısı indeksi Bezdek tarafından önerilmiştir (Bezdek J. C., 1974). Bölünme katsayısı indeksi (PC) iki bulanık kümenin üst üste gelme miktarını ölçer ve (61) eşitliği yardımıyla hesaplanır. Optimum küme sayısı, bu indeksin maksimum değerine karşılık gelen küme sayısıdır.

$$v_{PC} = \frac{\sum_{j=1}^N \sum_{i=1}^c \mu_{ij}^2}{n} \quad (62)$$

1.8.2. Bölünme Entropisi (CE)

Bölünme entropisi indeksi küme ayrımlarının bulanıklığını ölçer (Murat ve Şekerler, 2009) ve (62) eşitliği yardımıyla hesaplanır. Optimum küme sayısı, bu indeksin minimum değerine karşılık gelen küme sayısıdır.

$$v_{PE} = -\frac{1}{n} \sum_{j=1}^N \sum_{i=1}^c [\mu_{ij} \log_e(\mu_{ij})] \quad (63)$$

2. YAPILAN ÇALIŞMALAR

Görüntü bölütleme işlemleri genelde gri-ölçekteki görüntülere uygulanır. Bu uygulama doğrudan iki boyutlu bir matris olan gri-ölçekteki görüntü üzerinde olduğunda bölütleme görüntünün boyutuna bağlı olarak hesaplama zamanı artmaktadır. Bu sorunu ortadan kaldırmak için görüntünün öncelikle histogramı alınır. Tek kanallı histogram üzerinde değişik teknikler ile daha kısa sürede bölütleme yapıldığı görülmektedir. Ancak renkli görüntülerde görüntü bölütleme işlemi o kadar da kolay olmamaktadır. Dolayısıyla renkli görüntülerde bölütleme yapmak için renk bileşenleri üzerinde bir bölütleme yapabilmek için RGB renk uzayında işlenmesi gerekir. Ancak tüm gözetlerde için 3 boyutlu (R-G-B) bir veri kümesi ile yapılacak bir kümeleme oldukça fazla işlem hacmine sahip olacaktır. Oysa renkli görüntülerin renk özü bileşeni elde edilerek bu bileşenin histogramı yarımıyla bölütleme yapılması amaçlanmıştır.

2.1. Gri Ölçekli Görüntülerde Bölütleme

Gri ölçekli görüntüler tek kanaldan oluşan siyahla beyaz arasında gri seviyeleri içeren görüntü çeşididir. Gri ölçekli görüntüler 2 boyutlu bir matristen oluşmaktadır. matrisin her elemanı bir gözet değeri karşılık gelmektedir. Bu matris içindeki gözet değerlerine göre sınıflandırma yapılmasıyla görüntü gözet seviyesinde bir bölütleme yapılır.

2.1.1. Gözet Tabanlı Kümeleme ile Bölütleme

Gri ölçekli görüntülerdeki her gözet bir bileşenden oluştuğundan gözetler 1 boyutlu bir uzay oluşturmaktadır. Klasik FCM algoritması kullanılarak bir görüntü istenen kadar sınıfa (kümeye ayrılabilir). Her gözet için elde edilen üyelik değerlerinden en büyük üyelik değerine sahip olan küme o gözeği içerdiği kabul edilerek üyelik değerleri keskinleştirilmektedir. Normal eşikleme işleminde belli aralıktaki renkler tek bir renge ötelenirler. Kümeleme işlemiyle bölütlemeye ise her kümeyle ait renkler o kümenin ortalama renginin değerine ötelenmektedir. Örnek olarak Şekil 13'teki görüntü seçilmiştir. Bu görüntü klasik FCM algoritması yardımıyla $C = 2$ için kümeleme işlemine sokulur ve

her renk ait olduđu k¼me ortalamasına ¼telenirse Őekil 14'teki g¼r¼nt¼ elde edilir. B¼l¼tlenmiŐ g¼r¼nt¼ renkli olarak etiketlenirse Őekil 15'te g¼sterilmiŐtir. Bu g¼r¼nt¼ Aynı Őekilde $C = 4$ i¼in aynı iŐlemler uygulanırsa Őekil 16'daki g¼r¼nt¼ elde edilir. 4 b¼l¼te ayrılmıŐ g¼r¼nt¼de aynı Őekilde renkli etiketlenirse Őekil 17'deki g¼r¼nt¼ elde edilir. Őekil 14 ve Őekil 16'da renk indirgemesi uygulanan g¼r¼nt¼ler aynı zamanda b¼l¼tlenmiŐ olmaktadır. Ancak Őekil 15 ve Őekil 17'deki g¼r¼nt¼ler yalnızca b¼l¼tlenmiŐtir.



Őekil 13. ¼rnek olarak kullanılan ¼zg¼n g¼r¼nt¼.



Şekil 14. Şekil 13'teki görüntünün iki kümeye ayrılarak bölütlenmesi ile elde edilen görüntü.



Şekil 15 Şekil 14'de verilen bölütlenmiş (renk indirgenmiş) görüntünün önceden seçilen renkler yardımıyla etiketlenmiş olarak gösterimi.



Şekil 16. Şekil 13'teki görüntünün dört kümeye ayrılarak bölütlenmesi ile elde edilen görüntü.



Şekil 17. Şekil 16'da verilen bölütlenmiş (renk indirgenmiş) görüntünün önceden seçilen renkler yardımıyla etiketlenmiş olarak gösterimi.

2.1.2. Histogram Tabanlı Kümeleme ile Bölütleme

Histogram tabanlı bölütlemeye en yaygın yöntem doğrudan eşikleme yöntemidir. İki seviyeli eşikleme işlemi bölütleme işleminde iyi sonuçlar vermesine rağmen çok seviyeli bölütlemelerde sorunlara neden olmaktadır. Bu sorunu çözmek ağırlıklı kümeleme algoritması kullanılarak işlemin hem doğruluğu hem de hızı artırılmaktadır. Ağırlıklı kümeleme algoritması FCM algoritmasının üyelik değerlerinin ağırlıklandırılması ile gerçekleştirilmektedir. Bu algoritma ağırlıklandırma her renk seviyesinin frekansı ile yapılmaktadır.

Algoritma 3. Ağırlıklı FCM ile Bölütleme Algoritması

Adım 1. Gri ölçekli görüntü seviyeleri ($x_i = 1, 2, \dots, L$) ve bu seviyelere ait frekanslar (f_i), önceden belirlenen öbek sayısını $C \in [2, L]$, bulanıklaştırma parametresini ($m = 2$) ve durdurma kriterini ($\varepsilon = 0.01$) algoritmaya giriş parametreleri olarak ver.

Adım 2. Başlangıç üyelik değerlerini $\mu^{(0)} = [\mu_{ij}]$, $[0, 1)$ aralığında düzgün dağılımdan rastgele olarak belirle, Çevrim sayacını bire ($t = 1$) eşitle.

Adım 3. (t). çevrimini başlat,

Adım 4. Yeni üyelik değerlerini kullanarak öbek merkezlerini (64) eşitliği yardımıyla hesapla,

$$v_j = \frac{\sum_{i=1}^L f_i \mu_{ij}^m x_i}{\sum_{i=1}^L f_i \mu_{ij}^m}, \quad (j = 1, 2, \dots, C) \quad (64)$$

Adım 5. Üyelik değerlerini (65) eşitliği yardımıyla güncelleştir,

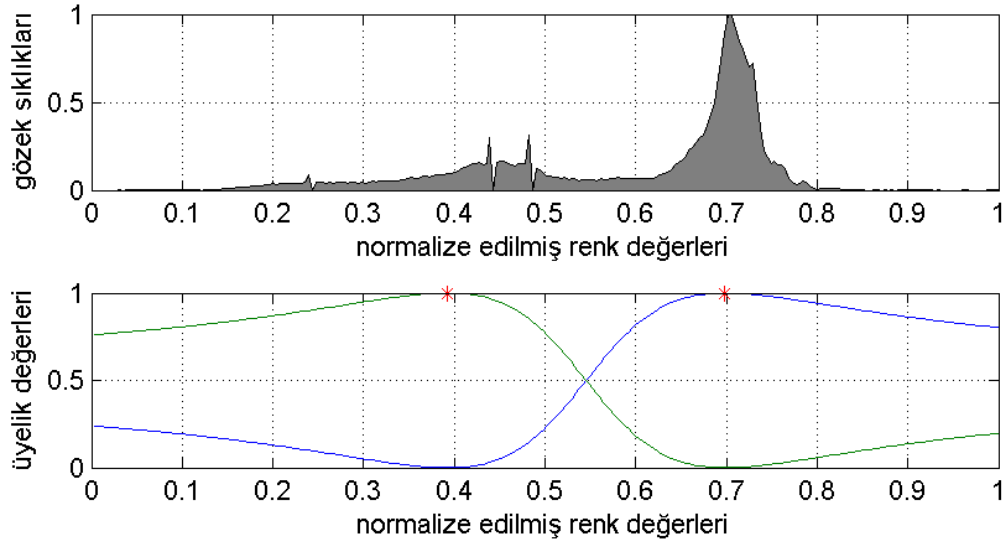
$$\mu_{ij} = \left(\sum_{k=1}^C \left(\frac{\|x_i - v_j\|}{\|x_i - v_k\|} \right)^{\frac{2}{m-1}} \right)^{-1} \quad (65)$$

Adım 6. (66) eşitsizliği sağlanıyorsa dur, yoksa t değerini bir artır ve Adım 3.'e giderek yeni çevrimi başlat,

$$\|\mu^{(t)} - \mu^{(t-1)}\| < \varepsilon \quad (66)$$

biçiminde verilen algoritmada x_i değerleri renk seviyelerini göstermektedir.(61) eşitliğindeki f_i değeri ise x_i renk seviyesindeki frekans değeridir.

Bu bölümde Şekil 9’da verilen özgün görüntü örnek olarak seçilmiş ve ağırlandırılmış FCM algoritması uygulanarak bölütleme yapılmıştır. Öncelikle iki seviyeli bir kümeleme ile üyelik değerleri elde edilmiştir (Şekil 18). Kümeleme sonucunda her renk değeri iki üyelik değerine sahip olmuştur. Her renk değerinin sahip olduğu iki üyelik değerinden büyük olanı seçilerek o renk değerinin hangi kümeye ait olduğu kesinleştirilmiştir. Keskinleştirilen aitliklere göre her renk değeri ait olduğu kümenin ortalama renk değerine ötelenerek bölütleme yapılmıştır (Şekil 19). Bölütlenmiş görüntünün önceden seçilen renkler yardımıyla etiketlenmesi sonucu elde edilen görüntü Şekil 20’de verilmiştir.



Şekil 18. Şekil 8’deki görüntünün histogramı ve $C = 2$ için kümeleme sonucunda elde edilen üyelik değerleri



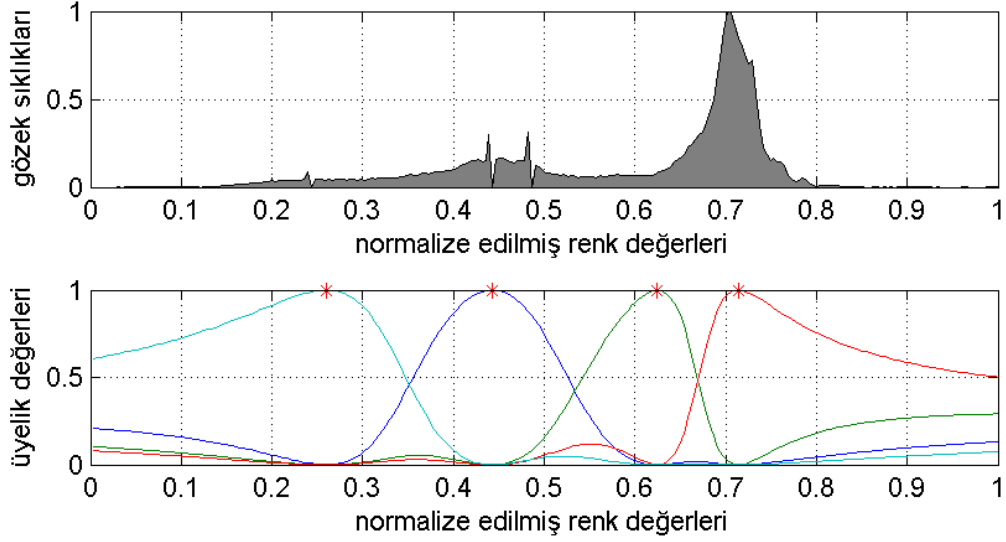
Şekil 19. Şekil 13'deki görüntünün iki kümeye ayrılarak bölütlenmesi ile elde edilen görüntü



Şekil 20 Şekil 19'da verilen bölütlenmiş (renk indirgenmiş) görüntünün önceden seçilen renkler yardımıyla etiketlenmiş olarak gösterimi.

Şekil 13'teki özgün görüntüye küme sayısı $C = 4$ için bir kümeleme işlemi uygulanırsa Şekil 21'deki üyelik değerleri elde edilir. En büyük üyelik değerlerine göre

aitlik kesinleştirilir. Kesinleşmiş üyelik değerlerine her renk ait olduğu kümenin ortalama rengine ötelenerek bölütleme yapılırsa Şekil 22'deki görüntü elde edilir. Elde edilen bu görüntü renk indirgenmiş görüntü olduğu gibi aynı zaman renk bölütlemesi yapılmış görüntüdür. Bu görüntünün etiketlenmesi sonucu elde edilen görüntü Şekil 23'te gösterilmiştir.



Şekil 21. Şekil 8'deki görüntünün histogramı ve kümeleme sonucunda elde edilen üyelik değerleri



Şekil 22. Şekil 8'deki görüntünün iki kümeye ayrılarak bölütlenmesi ile elde edilen görüntü



Şekil 23. Şekil 11'de verilen bölütlenmiş (renk indirgenmiş) görüntünün önceden seçilen renkler yardımıyla etiketlenmiş olarak gösterimi.

2.2. Renkli Görüntülerde Bölütleme

24 ikillik renkli görüntülerdeki her gözek üç bileşenden oluştuğundan gözlekler 3 boyutlu bir uzay oluşturmaktadır. Klasik FCM algoritması kullanılarak bir görüntü istenen kadar kümeye ayrılabilir. Kümeleme sonucunda her gözek için elde edilen üyelik değerlerinden en büyük üyelik değerine sahip olan küme o gözeği içerdığı kabul edilerek üyelik değerleri keskinleştirilmektedir. Her kümeye ait gözeler o kümenin tüm renk bileşenlerine göre ortalama renk değerlerine ötelenerek bölütleme yapılmaktadır. Renkli görüntüler için iki ayrı görüntü kullanılmıştır. Bunlar Şekil 24'te verilen plastik toplar ile Şekil 25'te verilen tığ işlemeli örgü görüntüleridir.



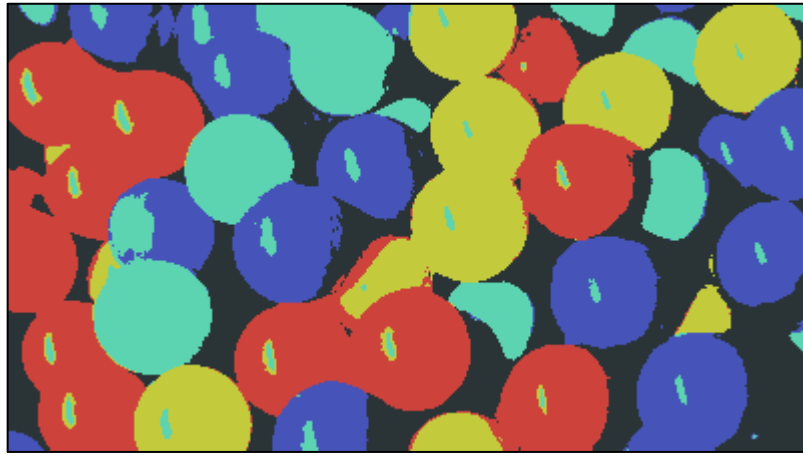
Şekil 24. Plastik toplar (URL-2, 2015)



Şekil 25. Tığ işlemeli çiçek örgüleri (URL-3, 2015)

2.2.1. Gözet Tabanlı Kümeleme ile Bölütleme

Şekil 24’te verilen plastik toplardan oluşan örnek görüntü klasik FCM algoritması yardımıyla 5 bölüte ayrılmıştır. Bölütleme sonucunda elde edilen görüntü Şekil 26’da verilmektedir. bölütleme sonucunda topların parlak kısımları genelde sarı ve turkuaz bileşenlerine dönüşüme eğiliminde olduğu görülmektedir.



Şekil 26. Plastik toplar görüntüsünün ağırlıklandırılmış FCM algoritmasıyla 5 bölüte ayrılması

Şekil 25'te verilen tığ işlemeli çiçek örgülerinden oluşan örnek görüntü klasik FCM kümeleme algoritması yardımıyla 8 bölüte ayrılmış görüntüsü Şekil 27'de gösterilmiştir. Bu görüntüde aynı iplik rengine sahip kısımlar birbirinden bağımsız olmalarına rağmen aynı rengi göstermektedirler.



Şekil 27. Tığ işlemeli çiçek örgülerinin görüntüsünün ağırlıklandırılmış FCM algoritmasıyla 8 bölüte ayrılması

2.2.2. Histogram Tabanlı Kümeleme ile Bölütleme

Gri ölçekli görüntülerde histogram tabanlı kümeleme yapmak için ağırlıklandırılmış FCM kümelemesi kullanılmaktadır. Gri ölçekli görüntülerde histogram tabanlı kümelemeyle bölütleme yapmak sorun oluşturmamasına rağmen renkli görüntülerde birden çok histogram olduğundan kümelemenin uygulanmasını imkansız kılmaktadır. Tüm renk bileşenlerinin kullanılması için her gözetim için renk özü (H) bileşeni hesaplanmaktadır. Bu bileşenin histogramı hesaplanarak kümeleme bu histogram üzerinde yapılmıştır. Burada dikkat edilmesi gereken bir nokta renk özü verilerinin açısal bir değişken olmasıdır. Açısal değişkenler normalde 360° ile 0° arasında sürekli olmalarına rağmen doğrusal hesaplamalarda süreksiz diye algılanmaktadır. Bu ise kümeleme algoritmasında sorunların çıkmasına neden olur. Bu sorunu çözmek için yönsel verilerde kümeleme tekniklerinden olan FCM4DD algoritması üzerinde durulmuştur.

Algoritma 4. Ağırlıklandırılmış FCM4DD Algoritması

Adım 1. Renkli görüntüde renk özü ($H = \theta$) seviyeleri ($\theta_i = 1, 2, \dots, L$) ve bu seviyelere ait frekanslar (f_i), önceden belirlenen öbek sayısı ($C \in [2, 360]$), bulanıklaştırma ($m = 2$) parametresi ve durdurma kriteri ($\varepsilon = 0.01$) algoritmaya giriş verileri olarak ver.

Adım 2. Başlangıçtaki öbek merkezlerini $\phi_j^{(0)}$ $[-\pi, \pi)$ aralığında düzgün dağılımdan rastgele olarak belirle ve (t) çevrim indisini 1'e eşitle.

Adım 3. (t). çevrimi başlat,

Adım 4. Her bir verinin geçici öbek merkezine göre iki açı arasındaki farkı hesapla, bu farkı,

$$\psi_{ij} = \left(\left((\theta_i - \phi_j) + \pi \right) \bmod 2\pi \right) - \pi \quad (67)$$

biçiminde $[-\pi, \pi)$ aralığına indirge. Bu fark yardımıyla üyelik değerlerini,

$$\mu_{ij} = \left(\sum_{k=1}^C \left(\frac{\|\psi_{ij}\|}{\|\psi_{ik}\|} \right)^{\frac{2}{m-1}} \right)^{-1} \quad i = 1, 2, \dots, 360 \quad j = 1, 2, \dots, C \quad (68)$$

eşitliğiyle hesapla. İki açı arasındaki fark ve yeni üyelik değerleri kullanılarak küme merkezlerini

$$\phi_j^{(t+1)} = \left(\phi_j^{(t)} + \frac{\sum_{i=1}^N f_i \mu_{ij}^m \psi_{ij}}{\sum_{i=1}^N f_i \mu_{ij}^m} \right) \bmod 2\pi, \quad (j = 1, 2, \dots, C) \quad (69)$$

biçiminde güncelle.

Adım 5. (70) eşitsizliği sağlanıyorsa dur, yoksa Adım 3.'e git ve (t+1). çevrimini başlat.

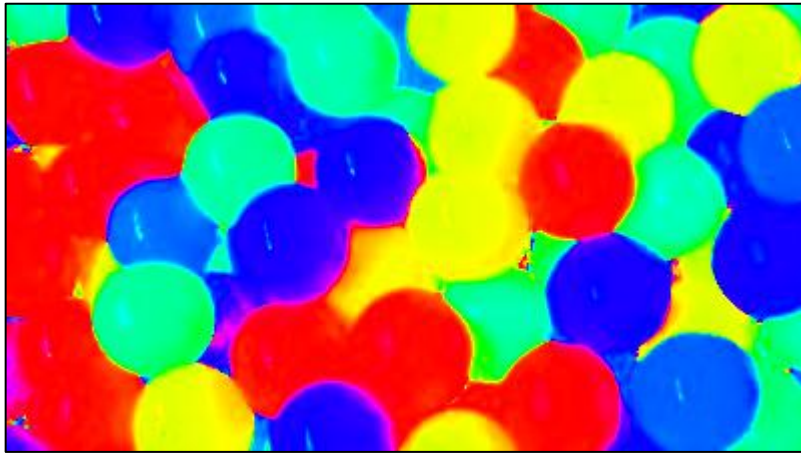
$$\|\mu^{(t)} - \mu^{(t-1)}\| < \varepsilon \quad (70)$$

biçiminde verilen algoritmada θ_i değerleri renk seviyelerini göstermektedir.(69) eşitliğindeki f_i değeri ise θ_i renk seviyesindeki frekans değeridir.

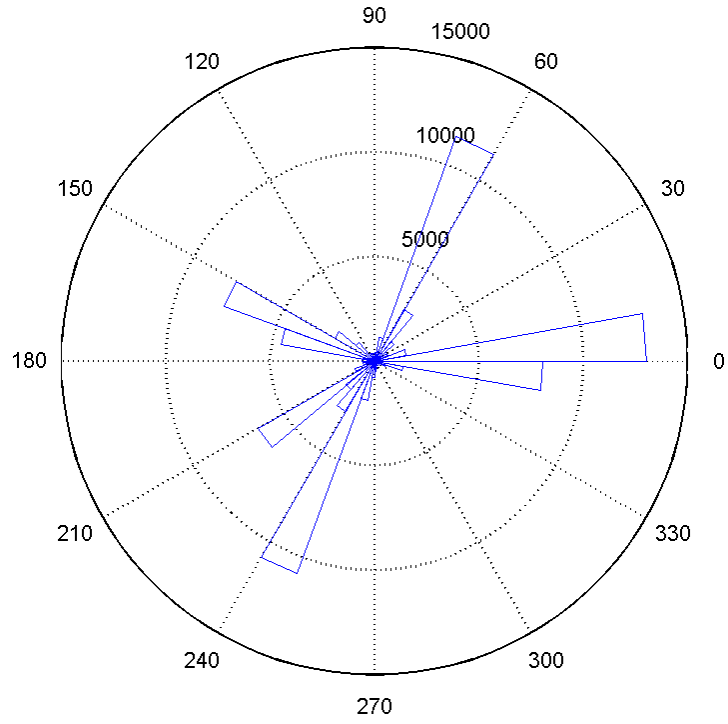
Ağırlıklandırılmış FCM4DD algoritması Şekil 24 ve Şekil 25'te verilen örnek görüntülere uygulanması için görüntülerin renk özü değerlerinin elde edilmesi gerekir. Renk özünü elde etmek için HSV renk uzayındaki renk özü bileşeni hesaplanmıştır. Örnek görüntülere bölütleme uygulanmadan önce görüntülerin daha iyi karşılaştırılması için görüntünün S ve V bileşenleri en büyük değerlerine eşiklenmiştir (Şekil 28, Şekil 33).

Eşiklenen görüntülerdeki renk özü (H) bileşenin histogramları Şekil 29 ve Şekil 34'te verilmiştir. Şekil 29'daki histogram verilen değerlere kullanılarak ağırlıklı FCM4DD algoritması yardımıyla optimal küme sayısı 4 olarak belirlenmiştir (Şekil 30). Bu küme sayısına göre ağırlıklı FCM4DD küme algoritması ile yapılan kümeleme işlemi sonucu Şekil 31'de üyelik fonksiyonları elde edilmiştir. Bu üyelik fonksiyonlarının en büyük olduğu bölgelere o kümenin ortalama değerinin atanmasıyla elde edilen bölütlenmiş görüntü Şekil 32'de verilmiştir.

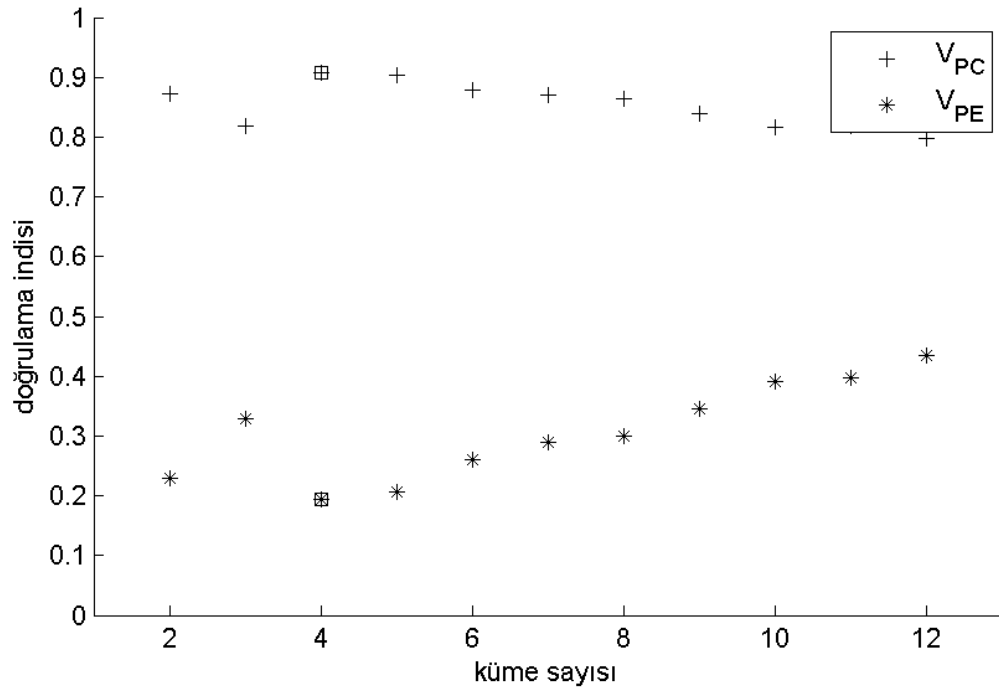
Örnek görüntülerin bulanık olan bölgelerinin bölütleme sonucunda keskinleştiği görülmektedir.



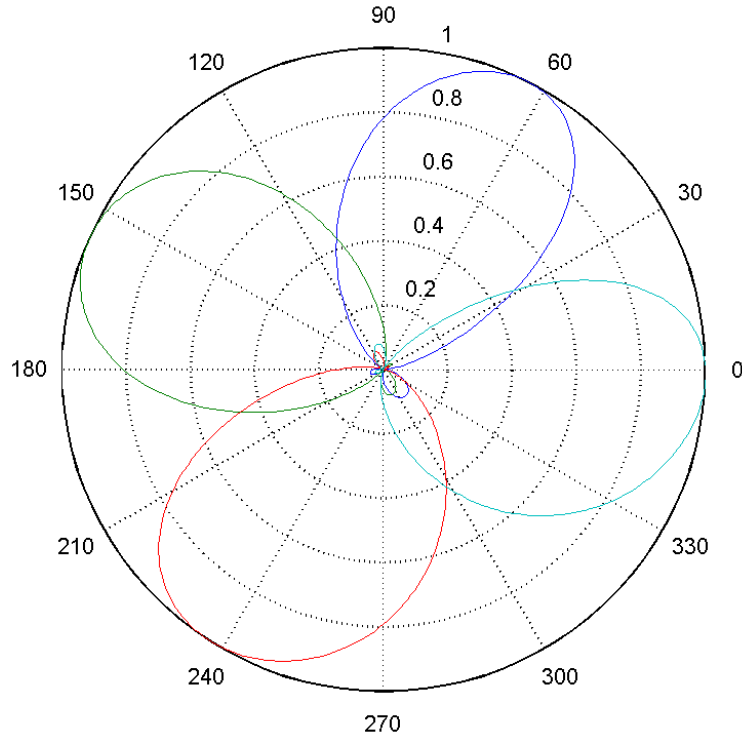
Şekil 28. Şekil 15'te verilen örnek görüntünün S ve V değerlerinin 1'e eşiklenmesiyle elde edilen görüntü



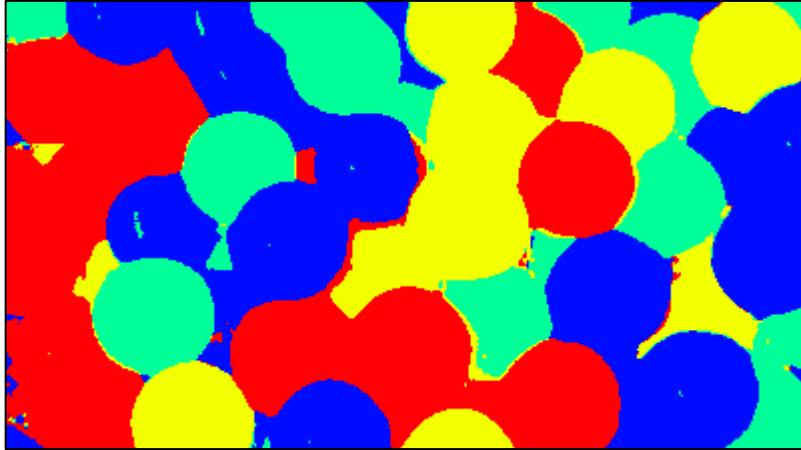
Şekil 29 Şekil 19'daki görüntünün renk özü (H) değerlerinin polar histogramı



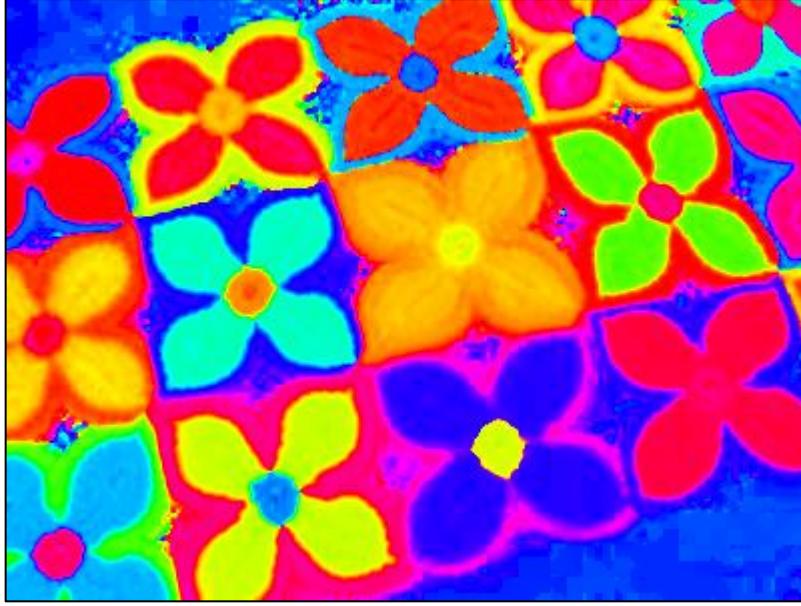
Şekil 30. Optimal küme sayısının belirlenmesi



Şekil 31. Şekil 29'daki histograma ağırlıklı FCM4DD kümeleme algoritması uygulanmasıyla elde edilen üyelik değerleri

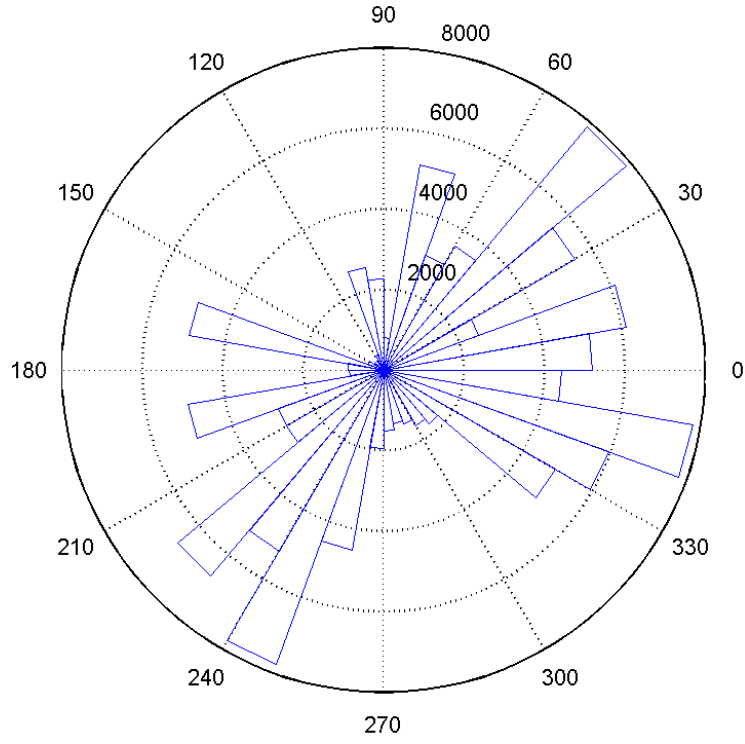


Şekil 32. Şekil 28'deki görüntüdeki renk özü (H) değerlerine göre 5 parçalı bölütleme yapıldıktan sonra elde edilen bölütlenmiş görüntü

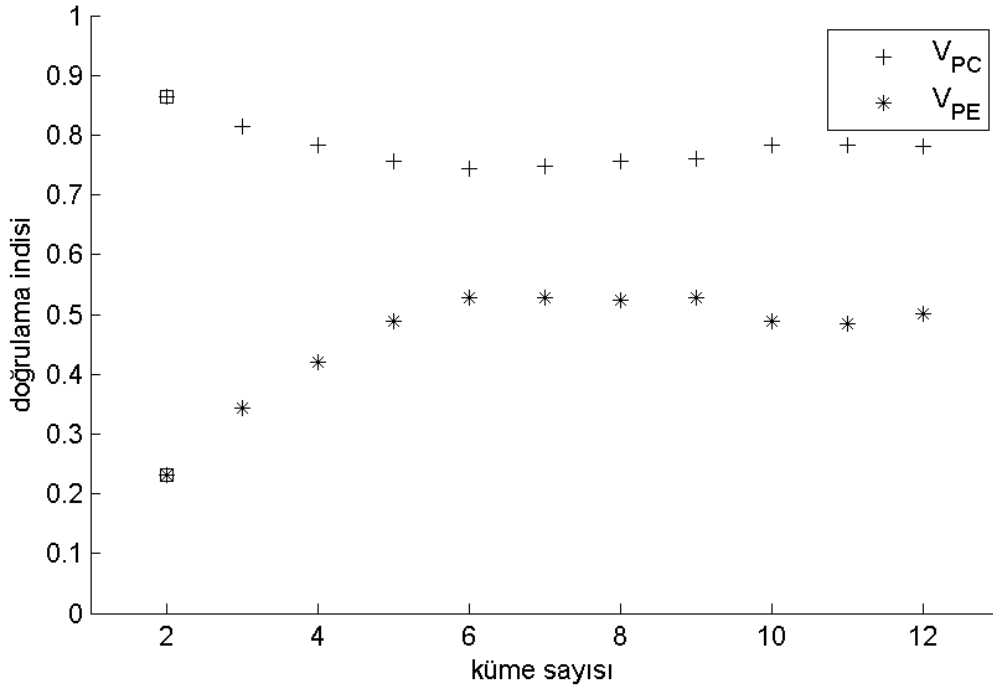


Şekil 33. Şekil 25’de verilen örnek görüntünün S ve V değerlerinin 1’e eşiklenmesiyle elde edilen görüntü

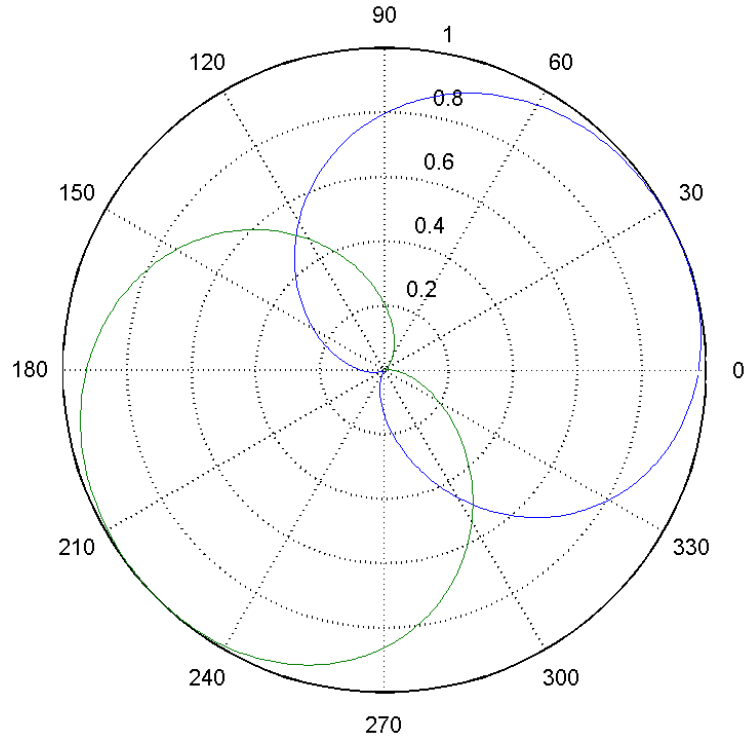
Şekil 34’deki histogram verilen değerlere kullanılarak ağırlıklı FCM4DD algoritması yardımıyla optimal küme sayısı 2 olarak belirlenmiştir (Şekil 35). Bu küme sayısına göre ağırlıklı FCM4DD küme algoritması ile yapılan kümeleme işlemi sonucu Şekil 36’da üyelik fonksiyonları elde edilmiştir. Bu üyelik fonksiyonlarının en büyük olduğu bölgelere o kümenin ortalama değerinin atanmasıyla elde edilen bölütlenmiş görüntü Şekil 37’de verilmiştir. Şayet küme sayısı 8 seçilirse Şekil 38’de elde edilen üyelik fonksiyonları verilmiştir. Bu üyelik fonksiyonlarının en büyük olduğu bölgelere o kümenin ortalama değerinin atanmasıyla elde edilen bölütlenmiş görüntü Şekil 39’da verilmiştir. Şekil 33’teki bulanık bölgelerin keskinleştiği görülmüştür.



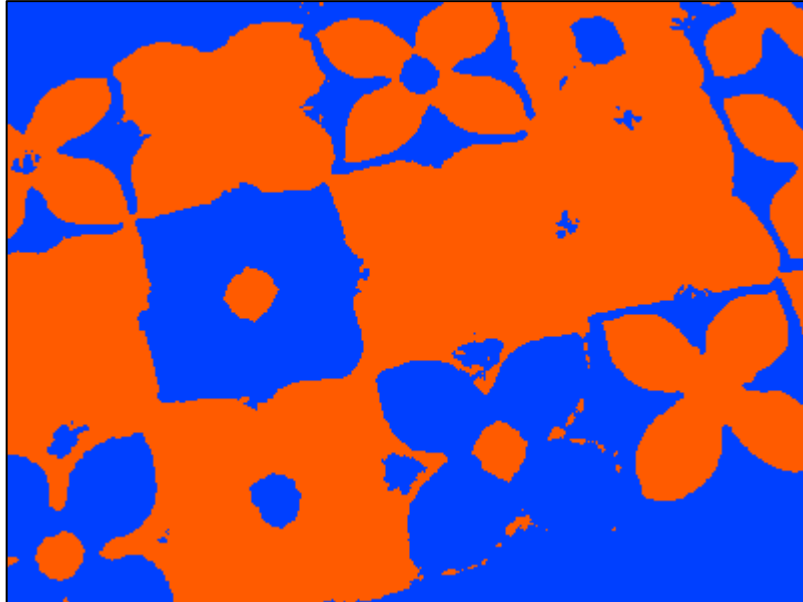
Şekil 34. Şekil 23'teki görüntünün renk özü (H) değerlerinin polar histogramı



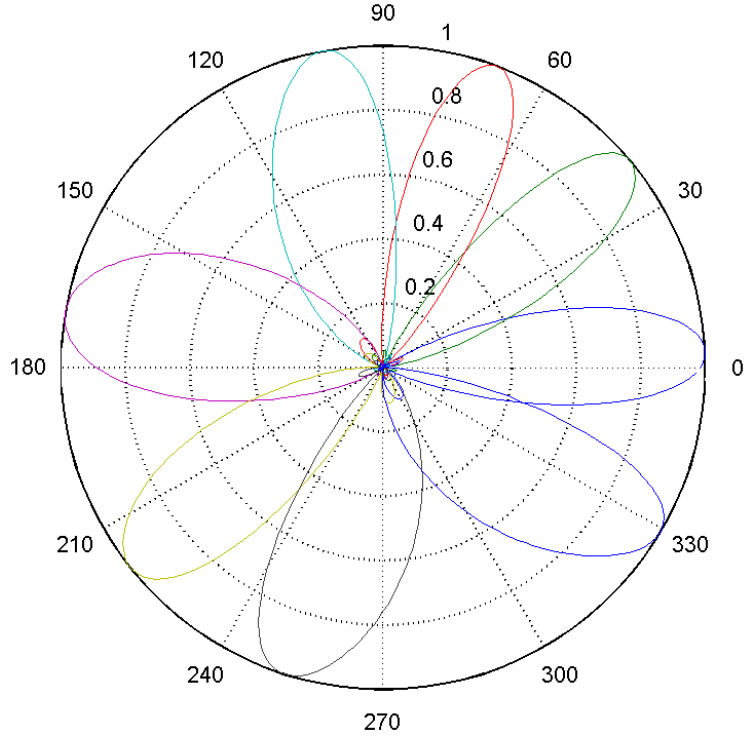
Şekil 35. Optimal küme sayısının belirlenmesi



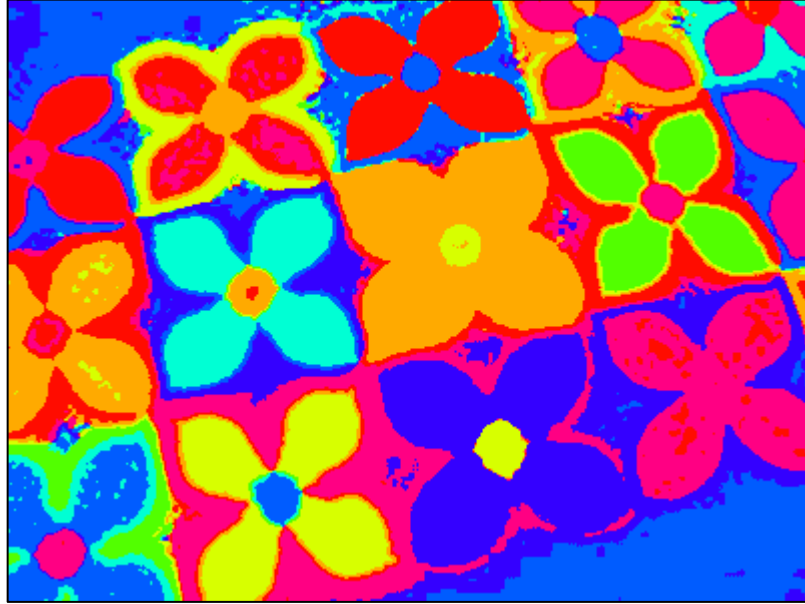
Şekil 36. Şekil 34'teki histograma ağırlıklı FCM4DD kümeleme algoritması uygulanmasıyla elde edilen üyelik değerleri.



Şekil 37. Şekil 33'teki görüntüdeki renk özü (H) değerlerine göre 2 parçalı bölütleme yapıldıktan sonra elde edilen bölütlenmiş görüntü



Şekil 38. Şekil 34'teki histograma göre gözlem sonucu küme sayısı 8 olarak belirlenip ağırlıklı FCM4DD algoritmasıyla elde edilen üyelik değerleri.



Şekil 39. Şekil 33'teki görüntüdeki renk özü (H) değerlerine göre 8 parçalı bölütleme yapıldıktan sonra elde edilen bölütlenmiş görüntü

3. BULGULAR VE SONUÇLAR

Bu çalışmada, gri ve renkli görüntülerin bölütlemesini bulanık kümeleme yardımıyla yapılması konusunda algoritmalar geliştirilmiştir. Gri görüntülerde gözek tabanlı bölütleme işlemi literatürde kullanılmasına rağmen hesaplama zamanındaki kazanımlarından dolayı histogram tabanlı eşiklemeyle bölütleme yapılması yaygın olarak tercih edilmektedir. Bölüm 2.1.1’de ve bölüm 2.1.2’de iki farklı yöntemle kümeleme sonuçları verilmiştir. Görüntüsel olarak aynı sonucu vermiştir ve Tablo 3’te sayısal sonuçları verilmiştir. Tüm denemelerde bulanık parametresi $m = 2$ ve tolerans değeri $\epsilon = 0.01$ alınmıştır.

Tablo 3. Gri ölçekli görüntü bölütleme sonuçlarının karşılaştırılması

	Gözek Tabanlı Kümeleme ile Bölütleme		Histogram Tabanlı Kümeleme İle Bölütleme	
	$C = 2$	$C = 4$	$C = 2$	$C = 4$
Küme sayısı				
Eşik değerleri	0.54	0.35 0.53 0.67	0.54	0.35 0.53 0.67
Ortalama Hesaplama Zamanı (sn) (Deneme sayısı:10)	78.5969	1242.8766	0.4908	1.2390

Tablo 3’te iki yönteminde normalize edilmiş aynı eşik değeri vermesine rağmen hesaplama zamanı açısından büyük bir fark gözükmektedir. Dolayısıyla histogram üzerinde ağırlıklandırılmış kümeleme algoritması ile kümelemenin zaman açısından önemi ortaya çıkmaktadır.

Renkli görüntülerde ise 3 boyutlu renk bileşen uzayından dolayı klasik kümeleme algoritmaları hesaplama zamanı açısından oldukça verimsiz sonuçlar üretmektedir. Bu durum özellikle gerçek zamanlı ve otomatik sistemlerde istemeyen sorunların çıkmasına neden olmaktadır. Renkli görüntülerde histogram tabanlı bir bölütleme yapılması gerekir. Ancak renkli görüntülerde her renk bileşenini kendi histogramı olduğundan bunlar arasında bir eşikleme yapmanın güçlüğü ortadadır. Bu sorunu gidermek için üç renk bileşenini içeren renk özüne göre bölütleme yoluna gidilmiştir. Oysa renk özü bileşeni açısıl bir

değişim gösterdiği için klasik eşikleme veya kümeleme yöntemleri yetersiz kalmaktadır. Bunun için renk özü histogramı üzerinde yönsel veriler için ağırlıklı bulanık kümeleme algoritması kullanılmıştır. Bu işlem için Şekil 24 ve Şekil 25'deki iki örnek görüntü seçilmiştir. Bu görüntülere uygulanan bölütleme işlemi sonucunda elde hesaplama zamanları Tablo 4'de verilmiştir. Tüm denemelerde bulanık parametresi $m = 2$ ve tolerans değeri $\epsilon = 0.01$ alınmıştır.

Tablo 4. Renkli görüntü bölütleme sonuçlarının karşılaştırılması

	Gözet Tabanlı Kümeleme ile Bölütleme		Histogram Tabanlı Kümeleme ile Bölütleme	
	$C = 5$	$C = 8$	$C = 5$	$C = 8$
Küme sayısı				
Ortalama Hesaplama Zamanı (sn) (Deneme sayısı:10)	568.4827	2220.4970	0.708214	5.6307

Tablo 4'te göre histogram tabanlı bölütlemenin, gözet tabanlı bölütlemeye göre oldukça hızlı çalıştığı görülmüştür.

4. ÖNERİLER

Gerçek zamanlı çalışan otomatik sistemlerde bölütleme işlemi önemli bir yer tutmaktadır. Özellikle video işlemi uygulamaları günümüzde yaygın bir kullanım alanına sahip olmuştur. Ancak otomatik sistemlerde görüntü bölütlemesi için kullanılan zamanın hızlı olması her şeyden önemlidir. Önerilen bu yöntem ile otomatik sistemlerde daha hızlı sonuçlar alınabileceği düşünülebilir.

Sadece gri görüntülerde değil renkli görüntülerde de bölütleme yapmak renk ayrımlılığı içeren bir çok uygulamada tutarlı sonuçların alınmasının sağlayacaktır.

Bu çalışmada renkli görüntüler için renk özü tabanlı değişimlere göre bölütleme yapılmıştır. Ancak farklı renk uzayları (örneğin $La*b^*$) için de önerilen yöntem uyarlanarak kullanılabilir.

Ağırlıklandırılmış FCM4DD algoritması renkli görüntülerin bölütlenmesinin yanı sıra görüntülerin renk indirgenmesi işlemlerinde de rahatlıkla kullanılabilceği görülmektedir.

5. KAYNAKLAR

- Alpaslan, F., Erilli, N. A., Yolcu, U., Eğrioglu, E. ve Aladağ, Ç. H., 2011. Bulanık Kümelemede En Uygun Küme Sayısının Yapay Sinir Ağları ve Diskriminant Analizi İle Belirlenmesi, Atatürk Üniversitesi İktisadi ve İdari Bilimler Dergisi, 475-488.
- Belloulata, K. ve Konrad, J., 2002. Fractal image compression with region-based functionality, IEEE Trans. Image Process., 11, 4, 351-362.
- Bezdek, J. C., 1974. Numerical Taxonomy with Fuzzy Sets, Journal of Mathematical Biology, 57-71.
- Bezdek, J. C., 1981. Pattern Recognition with Fuzzy Objective Function Algorithms, Plenum Press, New York.
- Bezdek, J. C., Ehrlich, R. ve Full, W., 1984. FCM: The Fuzzy c-Means Clustering Algorithm, Computer & Geosciences, 10, 2-3, 191-203.
- Burger, W. ve Burge, M. J., 2009. Principles of Digital Image Processing - Fundamental Techniques, Springer, London.
- Crane, R., 1996. Simplified approach to image processing: classical and modern techniques in C, Prentice Hall PTR.
- Demirkaya, O. ve Asyali, M. H., 2004. Determination of image bimodality thresholds for different intensity distribution, Signal Proc.-Image Commun., 19, 6, 507-516.
- Gonzalez, R. C. ve Woods, R. E., 1992. Digital Image Processing, Addison-Wesley Publishing Company, Reading.
- Hartmann, S. L. ve Galloway, R. L., 2000. Depth-buffer targeting for spatially accurate 3-D visualization of medical images, IEEE Trans. Med. Imaging, 19, 10, 1024-1031.
- Höppner, F., Klawonn, F., Kruse, R. ve Runkler, T., 2000. Fuzzy Cluster Analysis, John Wiley & Sons, Chichester.
- Huang, D. Y. ve Wang, C. H., 2009. Optimal multi-level thresholding using a two-stage Otsu optimization approach., Pattern Recognition Letters, 30, 3, 275-284.
- Murat, Y. Ş. ve Şekerler, A., 2009. Trafik Kaza Verilerinin Kümeleme Analizi Yöntemi ile Modellenmesi, İMO Teknik Dergi, 4759-4777.
- Otsu, N. A., 1979. A Threshold Selection Method from Gray-Level Histogram, IEEE Transactions on Systems, Man and Cybernetics, 9, 1, 62-66.

- Pachowicz, P. W., 1994. Semi-autonomous evolution of object models for adaptive object recognition, IEEE Trans. Syst. Man Cybernet., 24, 8, 1191–1207.
- Pal, N. R. ve Pal, S. K., 1993. A review on Image Segmentation Techniques, Pattern Recognition, 26, 9, 1277-1294.
- Shih, F. Y., 2010. Image processing and pattern recognition: fundamentals and techniques., John Wiley & Sons..
- Tezel, Ö., Yönsel Verilerin Kümelenmesinde Bulanık C-Ortalamlar Algoritması, Yüksek Lisans Tezi, KTU, Fen Bilimleri Enstitüsü, Trabzon 2014.
- Zhang, Y. J., 1996. A survey on evaluation methods for image segmentation, Pattern Recog., 29, 8, 1335-1346.
- URL-1,[http://tr.wikipedia.org/wiki/Renk Renk](http://tr.wikipedia.org/wiki/Renk_Renk). 12 Kasım 2014.
- URL-2,<http://www.alibaba.com.cn> Plastik toplar. 12 Kasım 2014.
- URL-3,<http://www.zamanvadisi.com> Tığ işi örgü çiçekler. 12 Kasım 2014.

ÖZGEÇMİŞ

Soheila VEFAEINHR, 09 Mart 1986 tarihinde Tebriz (İran)'da doğdu. Lisans öğrenimini Şebister Bilişim Üniversitesinde, Mühendislik Fakültesi, Yazılım Teknolojisi – Bilgisayar Mühendisliği Bölümünü 2011 yılında tamamladı. 2011 yılında Karadeniz Teknik Üniversitesi, Fen Bilimler Enstitüsü, İstatistik ve Bilgisayar Anabilim dalında tezli yüksek lisans programına başladı.