



T.C.
SELÇUK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

MİNİMUM YAYILAN AĞAÇ TABANLI SIRALI GÖRÜNTÜ BÖLÜTLEME

Ali SAĞLAM

YÜKSEK LİSANS TEZİ

Bilgisayar Mühendisliği Anabilim Dalı

Şubat-2016
KONYA
Her Hakkı Saklıdır

ÖZET

YÜKSEK LİSANS TEZİ

MİNİMUM YAYILAN AĞAÇ TABANLI SIRALI GÖRÜNTÜ BÖLÜTLEME

Ali SAĞLAM

Selçuk Üniversitesi Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Yrd. Doç. Dr. Nurdan BAYKAN

2016, 83 Sayfa

Jüri
Doç. Dr. Halis ALTUN
Doç. Dr. Halife KODAZ
Yrd. Doç. Dr. Nurdan BAYKAN

Bir görüntüyü belirli açılardan benzer piksellerden oluşan bölümlere (bölütlere) ayıran görüntü bölütleme (segmentasyon) işlemi görüntü işleme uygulamaları için çok önemli bir aşamadır. Görüntü bölütleme işlemi için birçok yöntem önerilmiştir. Graf tabanlı görüntü bölütleme de bu yöntemlerden birisidir. Temsil gücü ve kullanım kolaylı gibi avantajlarından dolayı graflar birçok görüntü işleme çalışmasında önemli araçlar olarak sıklıkla kullanılmaktadır. Görüntü bölütleme ile veri kümeleme konuları birbirleriyle daima bağlantılı olmuşlardır. Birçok graf tabanlı görüntü bölütleme ve graf tabanlı veri kümeleme yöntemi içerisinde minimum yayılan ağaç (MYA) tabanlı yaklaşımlar, işlem kolaylığı ve düşük hesaplama karmaşıklığından dolayı kritik bir rol oynamaktadırlar.

Bu tez çalışmasında, ilk olarak, daha önceden literatürde sunulmuş bir algoritma olan, Prim'ın sıralı MYA temsili ile kümeleme algoritması görüntü bölütleme amacıyla gerçek görüntüler üzerinde uygulanmıştır. Bu algoritma ile bir veri kümesinin Prim algoritması ile çıkarılmış bütün bir MYA yapısının sıralı temsili taranmaktadır. Tarama sırasında MYA yapısı üzerindeki, hat kesme kriterini karşılayan uyumsuz hatlar belirlenerek ağaçtan çıkarılmakta (kesilmekte) ve bölütleme işlemi gerçekleşmektedir. Tez kapsamında ayrıca bu sıralı kümeleme algoritmasından geliştirilen bir metot önerilmiştir. Önerilen metoda göre, belirlenen uyumsuz hat kesildiği zaman oluşacak alt ağaçlar gürültü özelliğine sahip olacaksa bu hat kesilmemektedir. Böylece, algoritmanın sonunda ortaya çıkan bölütleme çıktısı, işlem sonrasında ortaya çıkabilecek ve gürültü olarak tanımlanabilecek istenmeyen küçük parçacıkları yok etmek için herhangi bir ek işleme gerek duymamaktadır. Uyumsuz hatların belirlenmesi için, uyarlanan sıralı kümeleme algoritmasında kullanılan kesme kriteri yerine, literatürden de faydalanılarak daha iyi sonuç veren ve farklı görüntü özelliklerine göre uyarlanabilen yeni bir karşılaştırma kriteri de tez kapsamında sunulmuştur. Son olarak önerilen metot gerçek görüntülerden oluşan veri setleri üzerinde test edilmiştir. Elde edilen sonuçlara göre, önerilen metot düşük işlem süresi ile en popüler görüntü bölütleme metotları ile doğruluk oranı açısından rekabet edebilecek düzeyde sonuçlar vermiştir. Test sonuçları görsel, grafiksel ve sayısal olarak tezin son bölümünde sunulmuştur.

Anahtar Kelimeler: Görüntü bölütleme, graf, kümeleme, minimum yayılan ağaç

ABSTRACT

MS THESIS

MINIMUM SPANNING TREE-BASED SEQUENTIAL IMAGE SEGMENTATION

Ali SAĞLAM

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

Advisor: Asst. Prof. Dr. Nurdan BAYKAN

2016, 83 Pages

Jury
Assoc. Prof. Halis ALTUN
Assoc. Prof. Halife KODAZ
Asst. Prof. Nurdan BAYKAN

The image segmentation process that separates an image to segments formed similar pixels in specific aspects is a very important stage for various image-processing applications. Many methods are proposed for the process of image segmentation. Graph-based image segmentation is also one of them. Due to their advantages such as representation convenience and ease of use, graphs are used as important tools in many image-processing studies. Image segmentation and data clustering issues are always been connected to each other. In many graph-based data clustering methods and many graph-based image segmentation methods, minimum spanning tree (MST)-based approaches play a crucial role due to ease of operation and low computational complexity.

In this thesis work, firstly, the algorithm of clustering with Prim's sequential representation of MST, which is proposed in the literature in advance, is applied on real images for the purpose of image segmentation. The sequential representation of the complete MST structure of a data set is scanned by the algorithm. During the scanning, the inconsistent edges on the MST structure, which satisfy the cut criterion, are removed from the tree (cut) and the segmentation is accomplished. In the scope of the thesis, additionally, a method improved from the sequential clustering algorithm is proposed. According to the proposed method, if the sub trees, which would occur when the determined inconsistent edge is cut, have the noise feature, this edge would not be cut. In this way, the output of the segmentation does not need to be done any post-processing for eliminating the undesired small particles, which might be emerged after the processing and can be defined as noise. For defining the inconsistent edges, instead of the cut criterion used in the data-clustering algorithm implemented, a novel comparison criterion, which performs better and can be adapted for images that have different features, is developed by benefiting from the literature. Finally, the proposed algorithm is tested on real image data sets. According to the results obtained, the proposed algorithm yields results at a level that it can challenge with the most popular image segmentation algorithms in terms of accuracy within low execution time. The results are showed in this thesis visually, graphically, and numerically.

Keywords: Image segmentation, graph, clustering, minimum spanning tree

ÖNSÖZ

Akademik çalışmamda desteğini hem bilgisiyle hem de özverisiyle bana hep hissettiren, yol göstericim sayın danışman hocam Nurdan AKHAN BAYKAN'a, her aşamamızda bilgisini ve yardımını hiç esirgemeyen sayın hocam Ömer Kaan BAYKAN'a, her türlü desteklerinden ötürü Konya Selçuk Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği Bölüm başkanımız sayın Halife KODAZ hocama, bölümdeki bütün öğretim üyesi hocalarıma ve öğretim elemanı arkadaşlarıma teşekkürlerimi sunarım.

Her çalışmamda olduğu gibi bu tez çalışmasında da gönülden desteğiyle hep yanımda olan sevgili eşim Duygu SAĞLAM'a, bugünlere gelmemi sağlayan annem Ayşe SAĞLAM'a ve babam Mehmet SAĞLAM'a, motivasyonumun en düşük olduğu zamanlarda bile sevgileriyle beni hep ayakta tutmayı başaran ve yüzümü güldüren kızlarım Emine Beyza SAĞLAM ve Zeynep Berra SAĞLAM'a teşekkürü bir borç bilirim.

Ali SAĞLAM
KONYA-2016

İÇİNDEKİLER

ÖZET ..... **iv**

ABSTRACT ..... **v**

ÖNSÖZ ..... **vi**

İÇİNDEKİLER ..... **vii**

SİMGELER VE KISALTMALAR ..... **viii**

1. GİRİŞ **1**

2. KAYNAK ARAŞTIRMASI **5**

3. MATERYAL VE YÖNTEM **11**

3.1. Görüntü Bölütleme Yöntemleri 11
3.2. Graf Teorisi ve Minimum Yayılan Ağaç 12
3.2.1. Minimum yayılan ağaç algoritmaları 17
3.2.1.1. Boruvka algoritması 17
3.2.1.2. Prim algoritması 19
3.2.1.3. Kruskal algoritması 20
3.2.1.4. Ters çevir - sil algoritması 22
3.3. Graf Tabanlı Görüntü Bölütleme Yöntemleri 23
3.4. Görüntünün Bölütleme İşleminin Graf Tabanlı Kümeleme ile Temsili 26
3.4.1. Graf tabanlı kümeleme problemlerinin tanımları 29
3.5. Prim Algoritması ile Bir Görüntünün MYA Yapısının Sıralı Temsili 31
3.6. Önerilen Yayılan Ağaç Tabanlı Sıralı Görüntü Bölütleme Metodu 36

4. ARAŞTIRMA SONUÇLARI VE TARTIŞMA **44**

5. SONUÇLAR VE ÖNERİLER **51**

5.1 Sonuçlar 51
5.2 Öneriler 52

KAYNAKLAR **53**

EK 1 ..... **59**

ÖZGEÇMİŞ **83**

SİMGELER VE KISALTMALAR

Simgeler

c : Sabit bir sayı
C : Küme
$|C|$: Kümedeki eleman sayısı
C_i : i numaralı alt küme
$d_e(p, q)$: p ve q pikselleri arasındaki Öklit uzaklığı
e_j : j numaralı hat
e_t^Q : Hat kuyruğunun t. indeksindeki hat
E : Hat kümesi
E_C : Aday hat kümesi
E_i : i numaralı hat alt kümesi
E_T : T ağacına ait hatların kümesi
$|E_T|$: T ağacındaki hat sayısı
F : Orman yapısı
F_i : i. aşamadaki orman yapısı
G : Graf
G_i : i. aşamadaki graf yapısı
$Int(T)$: T ağacı içerisindeki en uzun hat
k : Sabit bir sayı
$Komşu[v]$: v düğümüne komşu olan düğüm listesi
l : Karşılaştırılacak hat dizilerinin uzunlukları
m : Hat sayısı
n : Düğüm sayısı
M : Görüntünün piksel olarak uzunluk değeri
N : Görüntünün piksel olarak genişlik değeri
O : Zaman karmaşıklığı
Q_E : Hat kuyruğu
Q_W : Hat ağırlık değerleri kuyruğu
(R_p, G_p, B_p) : p pikselinin RGB değer vektörü
σ : Gauss filtresi parametresi
T : Ağaç
T_i : i numaralı ağaç
T_{MYA} : Minimum yayılan ağaç yapısı
τ : Eşik değeri
(u, v) : u ve v düğümleri arasındaki bağlantıyı sağlayan hat
$w(e_j)$: e_j hattının ağırlık değeri
w_t^Q : Hat ağırlık değeri kuyruğunun t. indeksindeki sayı
$w(u, v)$: u ve v düğümleri arasındaki hattın ağırlık değeri
V : Düğüm kümesi
v_i : i numaralı düğüm
$|V|$: V düğüm kümesinin eleman sayısı
V_i : i numaralı düğüm alt kümesi

Kısaltmalar

2D	: İki boyutu
3D	: Üç boyutlu
FP	: Fonksiyon problemleri
MYA	: Minimum yayılan ağaç
max	: En büyük değer
min	: En küçük değer
MYA-PST	: Prim algoritmasının sıralı minimum yayılan ağaç temsili
ODS	: En iyi veri seti skoru (optimal dataset score)
OIS	: En iyi görüntü skoru (optimal image score)
OS	: Ortalama işlem süresi
P	: Polinomsal zaman
PST	: Prim algoritmasının sıralı temsili
RGB	: Kırmızı, yeşil, mavi (red, green, blue) renk uzayı
sn	: Saniye

1. GİRİŞ

Bir görüntünün bilgisayar ortamına veya uyumlu sayısal cihazlara çeşitli yollarla ve farklı biçimlerde aktarılmış hali "sayısal görüntü" olarak adlandırılmaktadır. Bir sayısal görüntü kendine ait değerleri olan piksellerin birleşiminden oluşmaktadır. Sayısal görüntü işleme ise çok çeşitli amaçlar için sayısal bir görüntü üzerinde belirli analizlerin yapılması ve sonuçların değerlendirilmesi işlemidir (Gonzalez ve Woods, 2001).

Görüntü bölütleme (image segmentation), sayısal bir görüntüyü benzer özellikli piksellere sahip olan bölgelere ayırma işlemidir ki; görüntü analizi ve örüntü tanımada çok önemli bir rol oynamaktadır. Görüntü bölütleme, görüntü işleme uygulamalarındaki en zor aşamalardan birisidir ve analiz sonuçlarının kalitesini belirlemede çok önemli bir etkendir (Cheng ve ark, 2001). Görüntü bölütleme işlemi sonrasında ayrılan bölgeler, özellikleri sayesinde bir nesneyi tespit ve takip etme, nesne sayma veya değişiklik tespit etme gibi görüntü analizi işlerini kolaylaştırmak amacıyla kullanılmaktadır (Suresh, 2012).

Görüntü bölütleme bir görüntünün analizini ve anlamlandırılmasını daha da kolaylaştırmaktadır. Tipik olarak görüntü içindeki nesnelere ve sınırlara (alt bölgeler arasındaki ayrımlara) odaklanan bir işlem adımıdır. Bölütleme, kenar bulma veya örüntü tanıma işlemleri ile yapılabilmektedir. Bazı teknikler renk farklılıklarını, bazıları ise örüntü (doku) farklılıklarını, benzer görsel karakteristikleri veya yoğunluk farklılıklarını kullanarak birbiri ile ilişkili pikselleri aynı bölgeler içinde kümelemeye çalışmaktadırlar. Görüntü bölütleme işlemi görüntü çıkarma, görsel sadeleştirme, görüntü tabanlı modelleme gibi amaçlar için birçok uygulamada kullanılmaktadır. Bu işlemlerin çok zor olduğu düşünülmüştür. Çünkü görüntü bölütleme işlemi genellikle görüntüye ve amaca göre özel ve hesaplama maliyeti çok yüksek olan bir işlemdir (Dudgeon ve Mersereau, 1984; Wang, 2007; Çamalan, 2013).

Görüntü bölütleme işlemi özellikle nesne tanıma sistemlerinde önemli bir ön işleme adımı olarak görülür. Bir görüntüyü birden fazla bölgeye ayırmak için renk, doku ve yoğunluk gibi düşük seviye özellikler kullanılır. Piksel kümelerinden oluşan ilgili alt bölgeler nesne tanımada ayrı olarak değerlendirilir (Pantofaru, 2008). Görüntü bölütleme birçok biyomedikal görüntü analizinde de çok önemli bir aşama olmuştur (Pham ve ark, 2000; Cardenes ve ark, 2009; Njeh ve ark, 2012). Biyomedikal bölütleme için hangi çözümün uygulanacağı genelde görüntünün özelliklerine bağlıdır. Ancak

bütün durumlarda, teşhis ve tedavi koyabilmek için görüntüdeki nesneleri bölütlemeye ihtiyaç duyulmuştur. Görüntü bölütleme günümüzde bilgisayar destekli cerrahide de artan bir şekilde kullanılmaktadır (Glisson ve ark, 2011). Görüntü bölütlemenin popüler olduğu diğer bazı alanlar ise yüz tanıma (Al Haj ve ark, 2009), iris tanıma (He ve Shi, 2007) ve astronomidir (Nunez ve Llacer, 2003). Görüntü bölütlemenin bugünlerde yoğun olarak kullanıldığı bir başka uygulama alanı ise içerik tabanlı görüntü çıkarmadır (Rugna ve ark, 2011). Bu tez çalışmasında, görüntü bölütlemede görüntüye ve amaca bağlı olmadan bir bölütleme işlemi amaçlanmıştır.

Görüntü bölütleme ile ilgili çalışmaların başlarından itibaren, görüntü bölütleme veri kümeleme ile yakından ilişkili olmuştur. Böyle bir ilişki Wertheimer'ın Gestalt teorisiyle açık bir şekilde gösterilmiştir. Bu teoride; benzerlik, yakınlık ve düzgün süreklilik gibi gruplama kuralları dizisi insan algı sisteminin gruplama şekli gibi bazı yolları açıklamak için tanımlanmıştır (Wertheimer, 1938). Bu teoriden yola çıkarak, bu tezde veri kümeleme ve görüntü bölütleme yaklaşımları birlikte ele alınmıştır.

Graf teorisi bilgisayarlı görme problemlerinde önemli bir uygulama alanına sahiptir. Güçlü temsil yeteneği ve işlem kolaylığı sayesinde graf teorisi, bilgisayarlı görme problemlerinde her geçen gün daha fazla popüler olmaktadır. Graf teorisi görüntü işleme problemlerinin saf ve soyut bir yapıda temsil edilmesini sağlamaktadır (Bayramoglu ve Bazlamacci, 2005). Görüntü bölütleme teknikleri arasından birçok başarılı uygulama, görüntü elemanlarını bir grafla eşleştirmekten faydalanmıştır. Böylece görüntü bölütleme problemi, graf teorisindeki etkili araçlar sayesinde uzaysal bir düzlemde ayrık yapıda çözülmüştür. Bölütlemeyi bir graf üzerine formüle etmenin faydalarından birisi de tamamen bütünleşmiş işleçler sayesinde bölütleme sonrasında verilerin kesikli yapıya dönüştürülmesine gerek duyulmamasıdır. Bu yüzden ayrıklaştırma hatalarına maruz kalmamaktadır (Xu ve Uberbacher, 1997; Felzenszwalb ve Huttenlocher, 2004; Wang, 2007; Yi ve Moon, 2012; Peng ve ark, 2013).

Görüntü bölütleme işlemi için graf tabanlı yaklaşımlarda, bir görüntü graf teorisi araçları kullanılarak bir düzlem üzerine haritalandırılır (Bondy ve Murty, 1976). Graf teorisi, görüntü özellikleri hakkında bilgi elde etme işlemini kolaylaştırmaktadır. Graf tabanlı görüntü bölütleme ile özellik ve örüntü bilgileri gruplanır ve organize edilir (Tao ve ark, 2007). Özellik benzerliklerinin yanında, graf tabanlı bölütleme yaklaşımları, görüntü içindeki pikselleri temsil eden düğümlerin yapısal benzerliklerini ve bağlantılarını da göz önüne almaktadır (Zahn, 1971; Yoshida ve ark, 2006; Zhou ve ark, 2009). Görüntüyü bir graf üzerine haritalandırdıktan sonra, bölütleme işlemi ayrıksal bir

uzayda gerçekleştirilmektedir. Bu avantajlarına ek olarak, etkili ve güçlü veri temsilinden dolayı, graf tabanlı yaklaşımlar birçok popüler görüntü bölütleme ve veri kümeleme metotları tarafından kullanılmaktadır (Zahn, 1971; Xu ve Uberbacher, 1997; Karypis ve ark, 1999; Xu ve ark, 2002; Felzenszwalb ve Huttenlocher, 2004; Haxhimusa ve Kropatsch, 2004; Bayramoglu ve Bazlamacci, 2005; Franti ve ark, 2006).

Ağırlık değerleri ile ilişkilendirilmiş ve yönsüz hatlar ile bir graftaki bütün düğümleri döngü oluşturmadan en az toplam maliyet ile bağlayan alt graf yapısına o grafın minimum yayılan ağaç (MYA) yapısı denilir (Cormen ve ark, 2001). MYA yapısının ve algoritmalarının yüksek performansı, veriyi ve kümeleri hafızada tutma kolaylığı ve üzerinde işlem yapma kolaylığından dolayı birçok graf tabanlı kümeleme ve graf tabanlı görüntü bölütleme metodu tarafından yaygın olarak kullanılmıştır (Zahn, 1971; Morris ve ark, 1986; Kwok ve Constantinides, 1997; Xu ve Uberbacher, 1997; Felzenszwalb ve Huttenlocher, 2004; Haxhimusa ve Kropatsch, 2004; Grygorash ve ark, 2006; Wang ve ark, 2014; Zhong ve ark, 2015).

Bu tezde önerilen metot, son yıllarda ortaya konulmuş olan, MYA tabanlı ve doğruluk ve işlem süresi performansı açısından başarı göstermiş bir veri kümeleme metodundan geliştirilmiştir (Wang ve ark, 2014). Bu metoda göre ilk olarak Prim algoritması (Prim, 1957) ile bütün veriyi kapsayan MYA yapısı elde edilmektedir. Prim algoritması ile her çevrimde oluşturulmakta olan MYA yapısına bir hat ve düğüm eklenmektedir. Bu metotta MYA yapısına eklenecek olan hat aynı zamanda tek boyutlu olan bir hat kuyruğuna eklenmektedir. MYA elde edildiğinde, kümeleme algoritması eklenen hatlardan oluşan ve eklenme sırasına göre dizilmiş olan bir hat kuyruğu oluşturmuş olmaktadır. Algoritma, MYA temsili olan bu hat kuyruğu içerisinden kuyruktaki sırayı bozmadan ve kuyruktaki sıraya göre sıralı olarak tek tek alınan ve kontrol edilen hatların uyumsuz hat olup olmadığını tespit etmektedir. Hatlar, kuyruktaki sıraya göre kendisinin her iki tarafında bulunan ve sayısı kullanıcı tarafından önceden belirlenmiş olan hatlar ile karşılaştırılarak kontrol edilmektedir. Karşılaştırma sonucunda o hattın ağırlığı karşılaştırıldığı hatların hepsinden daha fazla ağırlık değerine sahipse o hat MYA yapısından çıkarılmaktadır (kesilmektedir). Bütün bir graf üzerinde değil de sadece MYA üzerinde dolaştığı için ve bütün yapıyı ele almadan sadece boyutu kısıtlanan dizgi üzerinde karşılaştırma işlemi yaptığı için algoritma işlem süresi açısından iyi kalitede sonuçlar vermiştir. Bu algoritma, veri kümeleme üzerinde iyi sonuçlar vermiş olmasıyla birlikte görüntü bölütleme için kullanıldığında yetersiz

kalmaktadır. Sadece, dizgi boyutunu temsil eden tek bir parametre alıyor olması ise görüntü çeşitliliğine göre adaptasyonu zorlaştırmaktadır (Wang ve ark, 2014).

Tez çalışması kapsamında önerilen algoritmada, karşılaştırma kriteri olarak, bütün bir dizgi yerine, dizginin merkezinin her iki tarafı ayrı ayrı ele alınmaktadır ve adaptasyon için bir parametre daha kullanılmaktadır. Buna ilaveten, dizgi uzunluğu için kullanılan parametre değeri kullanılarak, görüntü bölütleme sonrasında ortaya çıkabilecek ve gürültü olarak tanımlanabilecek olan küçük bölütler önlenmektedir. Algoritma, gerçek görüntülerden oluşan veri tabanları üzerinde test edilmiş ve sonuçları en popüler olan görüntü bölütleme metotları ile karşılaştırılarak tez içerisinde gösterilmiştir.

2. KAYNAK ARAŞTIRMASI

Var olan bir veri topluluğunu bazı benzerlik veya farklılık kriterlerine göre gruplara ayrılmasına kümeleme denilmektedir. Johnson (1967) yaptığı kümeleme çalışmasında, ilk önce bütün elamanları bir küme olarak düşünmüş ve aralarındaki benzerliklerine göre sıralayarak en benzer olan kümeleri istenilen sayıda küme kalana kadar birleştirmiştir. Johnson kümeleme metotları üzerine olan bir makalesinde iyi bir kümeleme algoritmasında olması gerek üç adet özellik olması gerektiğini iddia etmiştir. Bu özellikleri de, giriş kümesi bir nokta kümesi ve benzerlik matrisinden oluşmalı, metodun tanımı açık, belirgin ve sezgisel olmalı ve metot benzerlik ölçümü dönüşümünün altında değişken olmamalı şeklinde ifade etmiştir (Johnson, 1967). Zahn'ın çalışmasındaki çoğu örnekte MYA metotları bu prensipleri karşılamıştır.

Graf teorisi, çizgeleri inceleyen matematik dalıdır. Graf tabanlı birçok çalışma yapılmıştır. Wang ve Siskind (2001) Ortalama Kesim maliyet fonksiyonunun global çözümünü minimize etmek için hat ağırlıklarının doğrusal dönüşümü yöntemini önermişlerdir. Wang ve Siskind (2003) graf tabanlı metotları kullanarak, Oranlı Kesim olarak adlandırdığı yeni bir maliyet fonksiyonu önermişlerdir. Pavan, M. Pelillo (2003) bir grafı sezgisel optimizasyon yöntemleri ile graftaki baskın düğüm setini kullanarak yerele takılmadan çözme yöntemi üzerinde çalışmışlardır (Wang ve Siskind, 2001; Pavan ve Pelillo, 2003; Wang ve Siskind, 2003).

Bir graftan çıkarılabilecek bütün düğümleri kapsayan ağaç yapılarından en düşük toplam hat maliyeti olan ağaca minimum yayılan ağaç (MYA) denilmektedir. MYA kullanılarak veriler arasında farklılıklarına göre bağlantılar oluşturulmaktadır. MYA üzerindeki bu farklılıkları kullanarak yapılan kümeleme işlemi MYA tabanlı kümeleme olarak adlandırılmaktadır. Gower ve Rose (1969), MYA tabanlı tek bağlantılı kümeleme olarak adlandırılan iki adet iteratif algoritma önermişlerdir. Bu metotlar klasik kümeleme metotları ile karşılaştırıldığında daha pratik oldukları görülmüştür. Çünkü bu algoritmalar MYA üzerinde işlem yapmakta ve böylece sonuçlar kısa bir işlem süresi içerisinde elde edilebilmektedir. Her iki metot da kümeleme yaparken bir eşik değeri kullanmıştır (Gower ve Ross, 1969).

Zahn (1971) grafın MYA'sına dayalı bir kümeleme sunmuştur. Verilen küme sayısı kadar küme elde etmek için görüntünün bütününü kapsayan MYA'yı parçalama yöntemini kullanmıştır. Bu yöntem Zhan'ın çalışmasından bu yana hem nokta kümelemede, hem de gri seviye 2D görüntü bölütlemede uygulanmaktadır. Nokta

kümesinin yapısını tespit etmek ve tanımlamak için grafın MYA'sının kullanılması fikrini Zahn'a, Clark ve Miller tarafından rapor edilen kıvılcım odası fotoğrafları vermiştir (Clark ve Miller, 1966). Zhan, kümeleme üzerinde sabit eşik değerleri ve lokal ölçümleri kullanmıştır. Lokale takılmamak için ise silinecek hattın her iki yanındaki standart sapmalar ile silinecek hattın mesafesini karşılaştıran bir silme kriteri önermiştir (Zahn, 1971). İlerleyen yıllarda Lopresti ve Zhou (2000), Zahn'ın kümelemede kullandığı MYA tabanlı algoritmayı Euclidean mesafesini kullanarak renkli görüntüler üzerinde kullanmışlardır. Amaçları web üzerindeki görüntülerde karakterlerin yerlerini bulmak ve karakterleri graf yardımıyla tanımak olmuştur (Lopresti ve Zhou, 2000).

Xu ve ark. (2002), Öklit mesafesini kullanan üç adet kümeleme algoritması ortaya koymuştur. Birinci algoritma basitçe k sayısı kadar küme elde etmek için verinin MYA'sı üzerindeki $k-1$ tane en uzun hattı silerek kümeleme yapmaktadır. İkinci algoritma k kadar küme için düğümlerin, düğümlerin ait oldukları kümelerinin merkezine (ortalama konum/renk değerine) olan toplam uzaklığı ile birbirleri arasındaki toplam uzaklığı temel alan özyinelemeli bir algoritmadır. Bunun için de her iterasyonda bir kümeyi iki parçaya bölerken hangi hatlardan bölüneceğini bulan bir optimizasyon işlemi uygulamışlardır. Üçüncü algoritma ise kümeleme problemi için verinin genelinde en iyi çözümü bulmaya çalışmaktadır. Bu algoritma k adet temsili düğümü, düğümlerin kendilerine en yakın temsili düğüme olan uzaklıkları en küçük olacak şekilde seçmektedir. Bu algoritma ikinci algoritmaya benzemektedir. İkinci algoritmadan biraz farklı bir amaç fonksiyonu kullanmaktadır (Xu ve ark, 2002).

Olman ve ark. (2004) tek boyutlu bir dizi üzerinde gerçekleşen bir kümeleme algoritması önermiştir. Bu yaklaşıma göre kümeleme problemi MYA yapısından elde edilen ve tek boyutlu olan bir dizgi parçalama yöntemine çevrilmiştir (Olman ve ark, 2004).

Theoharatos ve ark. (2004) her bir piksel için etrafındaki komşu pikselleri de içine alacak 3x3'lük bir pencere içindeki düğümlerin MYA'sını almışlardır (Theoharatos ve ark, 2005). Bu MYA'ları hat uzunlukları toplamlarına göre küçükten büyüğe doğru sıralamışlardır. Kümelerin ortalamalarını da hesaba katarak bir eşik değerine göre kenar bulma yöntemi geliştirmişlerdir. Paivinen ve ark. (2005) MYA'daki hat uzunluklarına bakmaksızın en çok bağlantısı olan düğümleri birleştirerek kümeleme yapmışlardır (Paivinen, 2005).

Chrobak ve ark. (2006) K-means (Jain ve ark, 1999) problemlerini açgözlü algoritmaların tersine çevrilmesiyle çözülmesi yöntemini önermişlerdir. Zhong ve ark.

(2014) K-means algoritmasını ön işlem olarak kullanarak MYA'nın hızını artırma çalışması yapmışlardır. Belirli bir düğüm sayısından oluşan graf önce k-ortanca algoritması ile kümelere ayrılmakta, daha sonra her kümenin MYA'sı çıkarılıp tekrar kümeler MYA algoritması ile birleştirilmektedir (Chrobak ve ark, 2006).

Wassenberg ve ark. (2009) birbirinden bağımsız MYA'ları hesaplayan ve onları birleştiren bir algoritma geliştirmişlerdir. Bu algoritma nesneleri sınırlardan ayırmadan paralel programlamaya izin vermektedir Eşik değerini uyarlamak yerine, eşik değeri olarak daha küçük ağırlıkta bir sınırı tahmin eden bir başlangıç bağlantılı bileşen etiketleme yöntemi kullanmışlardır (Wassenberg ve ark, 2009).

Zhong ve ark. (2010) MYA bulma algoritmasını graf üzerinde iki defa çalıştırmışlardır. İlk MYA yapısında bulunan hatları graf üzerinden kaldırmışlar ve tekrardan graf üzerinde kalan hatlar için bir MYA çıkarmışlardır. Bu metoda göre kümeleme problemi küme ayırma ve küme bitiştirme problemi olarak iki gruba ayrılmaktadır (Zhong ve ark, 2010). Bu iki problemi çözmek için de aynı graftan çıkarılan iki farklı MYA üzerinde sırayla çalışan iki farklı algoritma önermişlerdir. Daha sonraki çalışmalarında da kümeleme için böl ve parçala algoritmasını önermişlerdir (Zhong ve ark, 2011). Algoritmaya göre, başlangıç prototipi olarak graftan çıkarılmış MYA yapısı, kullanım kolaylığından dolayı klasik bir kümeleme algoritması olan K-means algoritması (Jain ve ark, 1999) ile bölünmektedir. Bu işlemden sonra da oluşturulan alt graflar filtrelenmekte ve komşu hatlarla birleştirilmektedirler.

MYA tabanlı kümeleme çalışmalarından yola çıkarak Morris (1986), MYA tabanlı özyinelemeli bir görüntü bölütleme yöntemi sunmuştur. Metot Kruskal algoritmasını kullanmaktadır. Metoda göre başlangıçta her düğüm bir bölge olarak görülmektedir. Her bölgenin komşu olduğu bir bölge ile arasında yalnızca bir hat bulunmaktadır. En az ağırlıklı olan hattın her iki ucundaki komşu bölgeler birleştirilmektedir. Birleştirilen iki bölge artık tek bir bölge olarak görülmektedir ve yeni oluşturulan bölgedeki düğümlerin her birinin renk değeri o bölgenin ortalama renk değeri ile değiştirilmektedir. Yeni oluşturulan bölgenin renk değeri değiştiği için komşu bölgeler arasındaki hat ağırlıkları yeniden hesaplanmaktadır. Sonrasında yeniden en az ağırlıklı hattı bulma aşamasına dönülmekte ve bu işlemler aynı sıra ile istenilen küme sayısı kadar bölge kalana kadar devam etmektedir (Morris ve ark, 1986). Kwok ve Constantinides (1997), Morris'in algoritmasının hızını artırmak için hızlı yinelemeli bir MYA algoritması önermişlerdir. Burada işlem zamanını hızlandırmak için hat

ağırlıklarını kesikli yapıya dönüştürmüş ve eşit ağılıktaki hatları gruplayarak gruplar içinde yığın yerine kuyruk yapısı kullanmışlardır. Daha sonraları, Bayramoğlu (2005) tezinde bu yöntemleri dağıtık sistemler üzerinde uygulayıp karşılaştırma sonuçlarını göstermiştir (Kwok ve Constantinides, 1997; Bayramoglu ve Bazlamacci, 2005).

İçerisinde döngü olmayan yani bir düğümden yola çıkarak aynı hattan ikinci kez geçmeden tekrar aynı düğümle sonlanan bir yol içermeyen ve düğümler arasındaki hatlarında belirlenmiş bir yön bulunmayan graf yapılarına ağaç denilmektedir. Ağaç yapısını kullanarak, Wu ve Leahy (1990) MR görüntüler üzerinde doku sınıflaması yapmışlardır. Görüntünün ağaç yapısını çıkararak kesme ve birleştirme yöntemleri ile görüntüdeki en homojen bölgeleri bulmaya çalışmışlardır (Wu ve Leahy, 1990). Bunun için de önce sınıflaması kolay piksellere yer vererek, sınıflaması zor pikselleri sona saklayan hiyerarşik bir yöntem izlemişlerdir. Etiketleme ve bölge genişletme yöntemlerini kullanmışlardır. En son olarak da bölgelerin sayısını istenildiği kadar düşürmeye çalışmışlardır. Daha sonraları Wu ve Leahy (1993) görüntü üzerinde komşuluk yapısına göre benzer pikselleri ağaç yapısı ile birleştirmiş ve komşu ağaçlar (kümeler) arasındaki maksimum akışı hesaplayarak istenilen küme sayısının bir eksiği kadar hattı kümeler arasında en optimal olacak şekilde eklemişlerdir. Bunun için de hiyerarşik bir yöntem kullanmışlardır (Wu ve Leahy, 1993).

Xu ve Uberbacher (1997) gri seviye görüntü üzerinde MYA tabanlı bir görüntü bölütleme çalışması yapmışlardır. Her bir ağacın standart sapmasını minimize etmeye çalışarak ağaç bölme algoritması kullanmışlardır. Büyük boyutlu görüntülerde ortalama almak maliyetli olacağından sezgisel algoritmalar kullanılması yoluna gidilmiştir (Xu ve Uberbacher, 1997). Grygorash ve ark. (2006) o döneme kadar olan MYA tabanlı görüntü bölütleme algoritmalarını incelemiş ve iki yeni algoritma önermişlerdir. Bunlardan birisi hiyerarşik bir yöntemle kümeleri parçalama ve birleştirme yöntemi ile istenilen kadar küme elde edilmesini sağlayan bir algoritmadır. Diğer yöntem ise oluşacak kümelerin standart sapmalarını da hesaplayarak belirlenen bir eşik değerine göre otomatik bölütleme sağlamaya çalışan bir algoritmadır (Grygorash ve ark, 2006).

Görüntü bölütlemesi için yapılan bir diğer çalışmada, Cox ve ark. (1996) graf parçalama yöntemi kullanarak sınır değerleri ve bölge içindeki değerler arasındaki oranı minimize etmeye çalışarak görüntü bölütlemesi yapmaya çalışmışlardır (Cox ve ark, 1996).

Graf Kesim algoritması bir görüntüden elde edilen graf üzerinde bazı optimizasyon algoritmaları ile graf üzerindeki gerekli hatları silerek istenilen sayıda

bölüt elde eden bir algoritmadır. Boykov ve ark. (1999) Graf Kesim algoritması üzerinde enerji fonksiyonunu minimize etmeye çalışmışlardır. Bunun için iki algoritma önermişlerdir. İlk algoritma ile rastgele seçilen iki piksel üzerinden etiketleme yaparak piksellerin yer aldığı bölütlerin standart sapmalarını minimize etmeye çalışmışlardır. İkinci algoritma ile de rastgele seçilen bir piksel üzerinden etiketleme yaparak yerele takılmadan global çözüme ulaşmaya çalışmışlardır. Shi ve Malik (2000) Graf Kesim algoritması üzerindeki kesme kriterini optimize etmek için genelleştirilmiş öz-değer problemine dayalı etkili bir hesaplama tekniği önermişlerdir. Görüntü bölütleme için Graf Kesim algoritması ile de farklı çalışmalar yapılmıştır. Wang (2007) tezinde spektral görüntü parçalama ile Graf Kesim algoritmasının bir kombinasyonunu kullanarak görüntü üzerindeki nesnelerin graflarla temsil edilen 3D modellemesini çıkarmıştır. Yi ve Moon (2012) Graf Kesim tabanlı algoritmalar üzerine genel bir tarama çalışması yapmışlardır. Hızlanan Graf Kesim Tabanlı, Etkileşim Graf Kesim Tabanlı ve Şekil Öncelikli Graf Kesim Tabanlı olarak üç kategoride inceleme yapmışlardır. Çamalan (2013) tezinde görüntü segmentasyonu için bazı yöntemleri açıklamış ve önişlem yöntemlerinin görüntü segmentasyonu üzerindeki etkilerini farklı açılardan incelemiştir. Çalışmada, doğal ve sentetik görüntüler deneysel olarak test edilmiş ve doğru örneklemenin önemi üzerinde durulmuştur. Durusoy (2014) tezinde Graf Kesim tekniği ve bölge genişletme algoritması kullanarak medikal görüntü bölütleme çalışması yapmıştır. Test amaçlı olarak da 3D X-ray CT görüntülerde kalbi bölütleme işlemlerini gerçekleştirmiştir (Shi ve Malik, 2000; Boykov ve ark, 2001; Wang, 2007; Faliu ve Inkyu, 2012; Çamalan, 2013; Durusoy, 2014).

Felzenszwalb ve Huttenlocher (2004) minimum yayılma ağacının yapısını kullanarak birleştirme yöntemi ile etkili bir görüntü bölütleme algoritması ortaya koymuşlardır. En başta bütün pikseller birer küme olarak görülmüş ve küme birleştirme işlemelerini Kruskal algoritmasına uyarlayıp kümelerin yapısını da hesaba katan yeni bir karşılaştırma kriteri ileri sürmüşlerdir. Mohanram ve Sudhakar (2011) verilen bir ağdaki güç akışı probleminde en uygun yolu bulmak için ters çevir-sil algoritmasını kullanmışlardır. Suresh (2012) tezinde Mahalanobis mesafesini kullanarak, Felzenszwalb ve Huttenlocher'in yöntemine benzer bir yöntem uygulamıştır. Birebir uyarlamada algoritmanın başarılı olamadığını ifade etmiş ve mevcut algoritmada yeni bir karşılaştırma kriteri denemiştir. Haxhimusa ve Kropatsch (2004), Boruvka algoritmasını kullanarak ve Felzenszwalb ve Huttenlocher'in homojenlik kriterinden de faydalanarak hiyerarşik bir görüntü bölütleme yöntemi sunmuştur (Felzenszwalb ve

Huttenlocher, 2004; Haxhimusa ve Kropatsch, 2004; Mohanram ve Sudhakar, 2011; Suresh, 2012).

Peng ve ark. (2013) görüntü bölütlemesine graf teorisi yaklaşımları ile ilgili genel çaplı bir araştırma yapmışlardır. Graf teorisi tabanlı görüntü bölütleme metotlarından bazılarını uygulamaya döküp sonuçları test etmişlerdir (Peng ve ark, 2013).

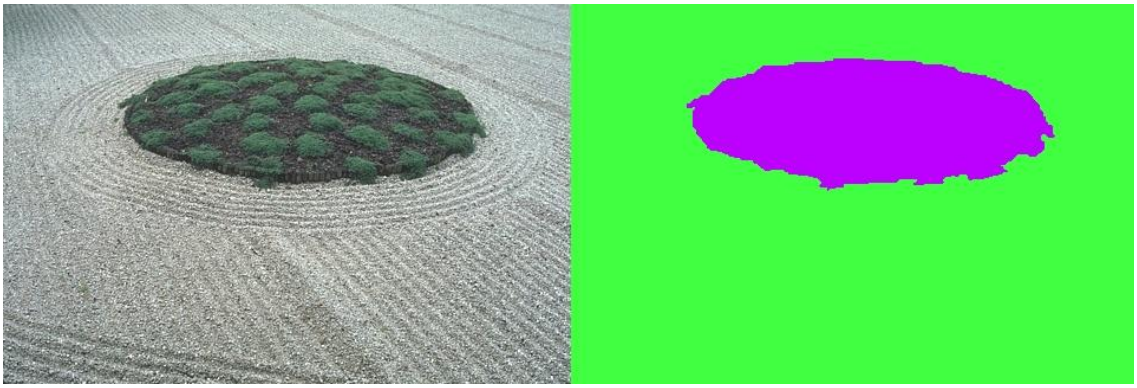
Wang ve ark. (2014), Prim'ın algoritmasını kullanarak Olman ve ark. (2002) önerdiği algoritmadan faydalanarak Prim algoritmasının MYA temsili ile sıralı kümeleme algoritmasını tanıtmışlardır. Buna ilaveten, önerilen algoritmayı kullanarak istenilen küme sayısı kadar parçalama yöntemini de tanıtmışlardır (Wang ve ark, 2014).

Yapılan tez çalışmasında da Wang ve ark. Önerdiği Prim tabanlı MYA temsili ile sıralı kümeleme algoritması temel alınarak, renkli görüntülerde bölütleme işlemi uygulanmıştır.

3. MATERYAL VE YÖNTEM

3.1. Görüntü Bölütleme Yöntemleri

Görüntü bölütleme konusu bilgisayarlı görmede eskilere dayanan temel bir konudur. Literatürde 3D ve 2D görüntülerin bölütlenmesi hakkında çok sayıda çalışma bulunmaktadır (Lezoray, 2003; Petrov ve ark, 2008; Wang ve ark, 2011). 3D görüntü bölütlemede farklı açılardan alınarak yapılandırılmış 3D görüntü içindeki piksellerin derinlikleri de bulunmaktadır. Bu görüntülerde nesnelerin 3D özelliğe sahip geometrik konumları da dikkate alınarak bölütlemesi yapılır. 2D görüntü bölütlemesi ise gri seviyeli veya renkli 2D görüntüler üzerinde sadece piksel özellikleri kullanılarak gerçekleştirilmektedir (Şekil 3.1) (Wang, 2007).



Şekil 3.1. Orijinal görüntü ve bölütlenmiş hali

Yi ve Moon, görüntü bölütleme metotlarının beş kategori altında gruplanabileceğini öne sürmüşlerdir (Yi ve Moon, 2012). Birincisi, eşik değeri tabanlı bölütleme metotlarıdır (Raut ve ark, 2009; Naz ve ark, 2010). Bu metotlar genellikle bir görüntüyü ön plan ve arka plan olarak iki bölüme ayırmak için kullanılmaktadırlar. Eşik değeri tabanlı bölütleme, görüntü bölütleme metotları arasında uygulaması en basit ve en hızlı olan metotlardır. Bu yöntemin zor kısmı ise uygun eşik değerini belirleme kısmıdır. İkinci kategori, kenar tabanlı görüntü bölütleme metotlarıdır (Gonzalez ve Woods, 2001; Raut ve ark, 2009; Naz ve ark, 2010). Bu metotlar ön planı ve arka planı bağlayan piksellerin değerlerinin arasındaki farklılıklardan faydalanmaktadırlar. Bunun için de kenar bulma operatörlerine gereksinim duyarlar. Üçüncü kategoride, bölge tabanlı görüntü bölütleme metotları yer almaktadır (Raut ve ark, 2009; Karoui ve ark, 2010; Naz ve ark, 2010). Genelde bölge genişletme ve bölme birleştirme işlemlerinden

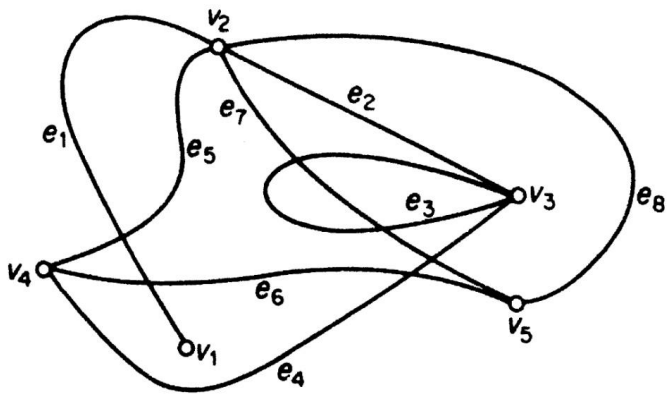
faydalanırlar. Bölge genişletme işleminde başlangıç olarak görüntü üzerinde bazı noktalar belirlenmekte ve komşu pikseller bu noktalar etrafında önceden tanımlanmış benzerlik kriterini göz önünde bulundurarak yoğunluklarına, renklerine veya dokularına göre kümelenmektedirler. Bölge bölme-birleştirme işleminde ise bir görüntü önce küçük bölgelere ayrılmakta ve daha sonra bu küçük bölgeler belirlenen bir kriteri sağlayıp sağlamadıklarına göre birleştirilmektedirler. Dördüncü kategori, su-seddi (watershed) tabanlı algoritmalardır (Gonzalez ve Woods, 2001; Rambabu ve ark, 2004; Raut ve ark, 2009; Naz ve ark, 2010). Bu kategorideki metotlarda, bir bölgeyi görüntüden çıkarmak için su toplama alanı olarak kabul edilen nesnenin bulunduğu bölgedeki gradyan görüntüye Su-Seddi dönüşümü algoritması uygulanmaktadır. Beşinci kategori ise enerji tabanlı metotlar olarak adlandırılmaktadır (Falcao ve ark, 2000; Boykov ve Jolly, 2001; Yu ve ark, 2001; Boykov ve Funka-Lea, 2006; Willett ve Nowak, 2007; Sundaramoorthi ve ark, 2008). Bu metotlar, görüntü bölütlendiği zaman istenilen sonuçlarla karşılaştırma yapabilmek için bir amaç (enerji) fonksiyonuna ihtiyaç duyarlar. Bu karşılaştırma sonuçlarına göre bölütleme tekrarlı olarak yeniden optimize edilmeye çalışılır.

Peng ve arkadaşları görüntü bölütlemeyi otomatik metotlar ve etkileşimli metotlar olarak iki gruba ayırmışlardır. Ancak, genelde ikisinin birlikte kullanıldığını da belirtmişlerdir (Peng ve ark, 2013).

Şimdiye kadar birçok görüntü bölütleme metodu geliştirilmiştir. Bu metotlardan birçoğu görüntü bileşenlerini bir graf üzerinde haritalama yönteminden istifade etmişlerdir. Böylece bölütleme problemini graf teorisi araçlarından faydalanarak kesikli bir yapı içerisinde çözebilmektedirler. Bu tür metotlar "graf tabanlı metotlar" olarak adlandırılmaktadırlar (Peng ve ark, 2013).

3.2. Graf Teorisi ve Minimum Yayılan Ağaç

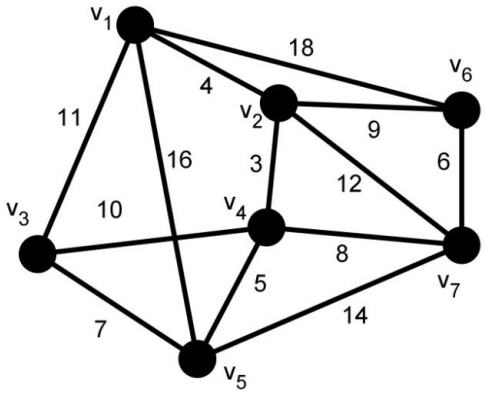
Graf teorisi (Çizge kuramı) (Graph theory), çizgeleri inceleyen matematik dalıdır. Graf, düğümler (tepe) ve bu düğümleri birbirine bağlayan hatlardan (kavis, kenar) oluşan bir tür ağ yapısıdır. Matematik ve bilgisayar biliminde kullanılan bu kuram, bir toplulukta bulunan nesneler arasındaki ilişkileri modelleyen matematiksel yapıları incelemektedir (Bondy ve Murty, 1976; Biggs, 1993; Gross ve Yellen, 2005; Diestel, 2010).



Şekil 3.2. Örnek graf yapısı (Bondy ve Murty, 1976)

Şekil 3.2'de görülmekte olan graf yapısı $G = (V, E)$ şeklinde gösterilmektedir. 5 adet düğümden ve 8 adet hattan oluşan G grafı için, $V = \{v_1, v_2, \ldots . v_5\}$ graftaki düğümleri saklayan bir düğüm kümesi ve $E = \{e_1, e_2, \ldots . e_8\}$ ise bu düğümler arasındaki hatları gösteren kümedir. Örnekteki v_1 ve v_2 düğümleri arasındaki bağlantıyı sağlayan hat (v_1, v_2) veya e_1 şeklinde gösterilmektedir.

Graflar, temsil gücü ve esneklikleri sayesinde bilgisayarlı görme çalışmalarında önemli bir araç olarak kullanılmaktadırlar. Eğer bir grafın her bir hattı ağırlık veya maliyet olarak adlandırılan bir sayı ile ilişkilendirilmişse bu graf ağırlıklandırılmış graftır (Şekil 3.3).

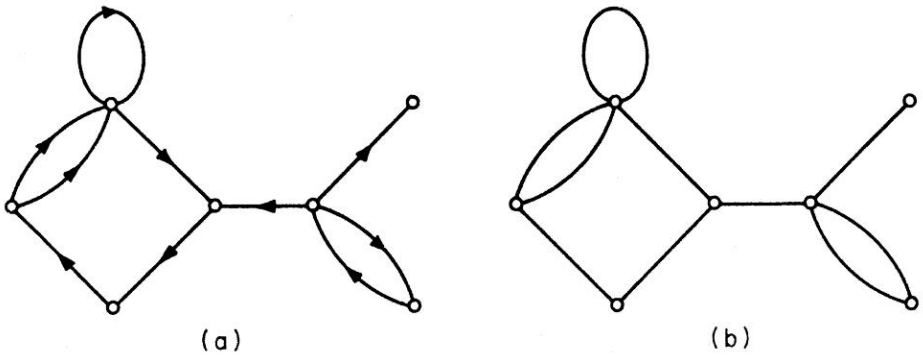


Şekil 3.3. Ağırlıklandırılmış graf örneği

Graflar kullanarak yapılan görüntü bölütleme işlemi, ağırlıklandırılmış bir grafı belirli bir kritere göre keserek çözen bir parçalama problemidir. Görüntü üzerinden elde edilen graflarda, hat ağırlıkları (mesafe uzaklığı) piksel özeliklerinden ve

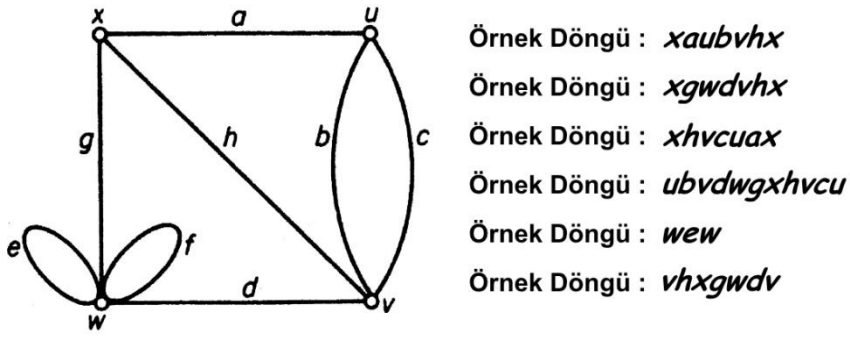
dağılımlarından elde edilen verilerden faydalanılarak ölçülmektedir (Janakiraman ve Mouli, 2008).

Eğer bir graftaki her bir hattın belirli bir yönü varsa o graf yönlü bir graf, yoksa yönsüz bir graftır. Şekil 3.4 (a)'da görülen yönlü grafta hatlar sadece okların gösterdiği yönde ilerlemektedir. Yani bir düğümden yola çıkarak ilerlendiğinde sadece okun gösterdiği yönde ilerlenebilmektedir. Şekil 3.4 (b)'de görülen yönsüz grafta hatların belirli bir yönü yoktur. Bu durumda graftaki bütün hatlar üzerinde her iki yöne doğru ilerlenebilmektedir.



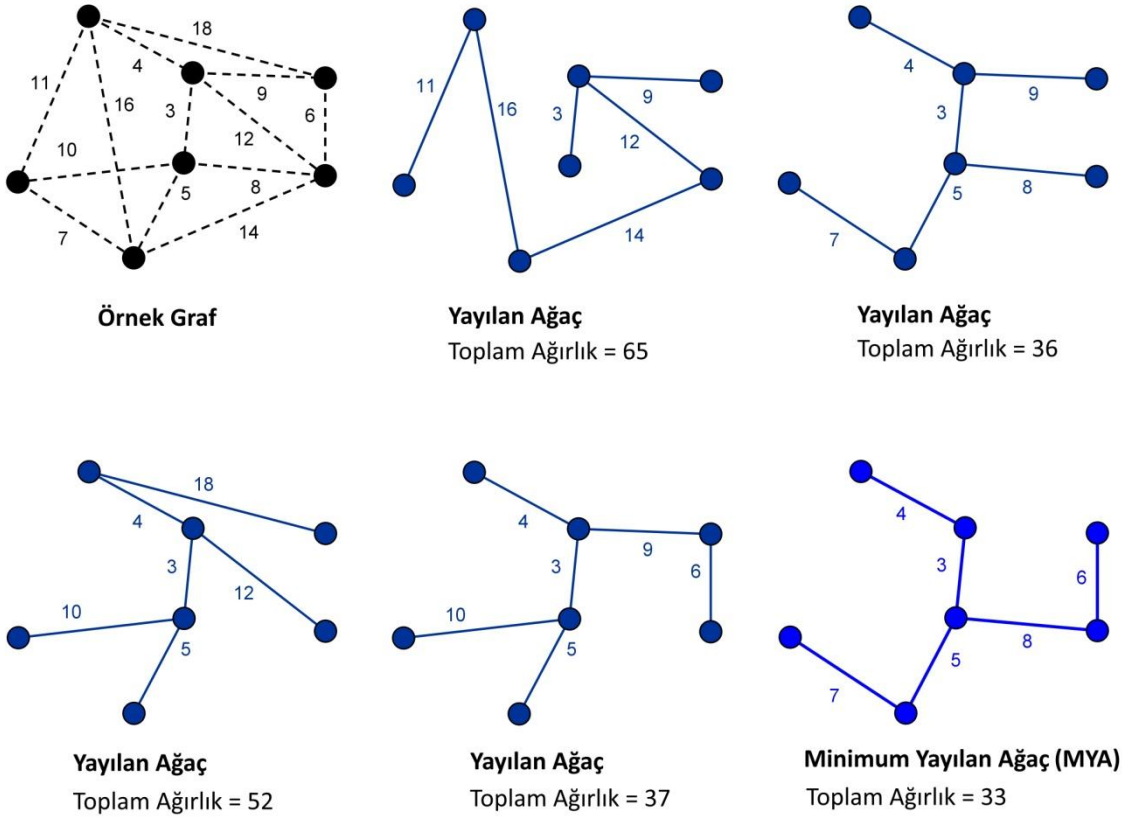
Şekil 3.4. (a) Yönlü graf; (b) yönsüz graf (Bondy ve Murty, 1976)

Bir grafın içinde, başlangıç noktası olarak seçilen herhangi bir düğümden 0 değerinden büyük olacak bir mesafe kadar, tek yönlü olarak ilerlendiğinde aynı düğümle sonlanan bir alt graf varsa döngü oluşmuştur. Şekil 3.5'de döngü içeren bir graf ve içerdiği döngülerden bazılarının, gidiş yolu sırasına göre düğüm ve hatlardan oluşan güzergahları görülmektedir.



Şekil 3.5. Döngü içeren bir graf (Bondy ve Murty, 1976)

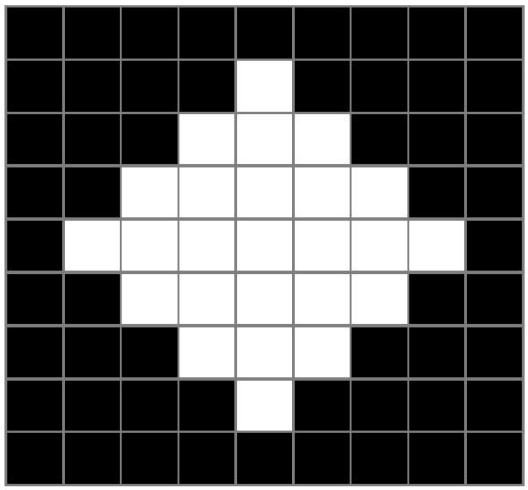
Bir graf üzerinde işlem yaparken genelde işlem kolaylığı ve güçlü bir veri temsili sağlamak için ağaç yapısına çevrilmektedir. Ağaç yapısı (Tree), sahip olduğu düğümler arasındaki hatlar vasıtasıyla herhangi bir döngü oluşturmayan, yönsüz ve ağırlıklandırılmış graflara denilmektedir (Bondy ve Murty, 1976). Diğer bir ifade ile herhangi bir grafın "yayılan ağaç" (spanning tree) yapısındaki gösteriminde, çizilen ağaç hiçbir döngü oluşturmadan bütün düğümleri kapsamaktadır. Bir graftan birden fazla şekilde yayılan ağaç yapısı çıkarılabilir. Bu çıkarılabilecek ağaç yapılarından, hat ağırlıklarının toplamı en düşük olan ağaç ise "Minimum Yayılan Ağaç" (MYA)(minimum spanning tree – MST) olarak adlandırılmaktadır (Boruvka, 1926; Pettie ve Ramachandran, 2002; Srivastava ve Ghosh, 2003; Özsoyeller, 2008; Zhong ve ark, 2015). Şekil 3.6'da örnek bir graf ve bu graftan çıkarılmış yayılan ağaç örnekleri görülmektedir. Şekilde görüldüğü gibi MYA, bir graftan çıkarılabilecek yayılan ağaçlar arasındaki en düşük toplam ağırlığa sahip olan ağaçtır. Bazı graflardan toplam hat ağırlıkları birbirine eşit olan birden fazla MYA yapısı da çıkarılabilmektedir.



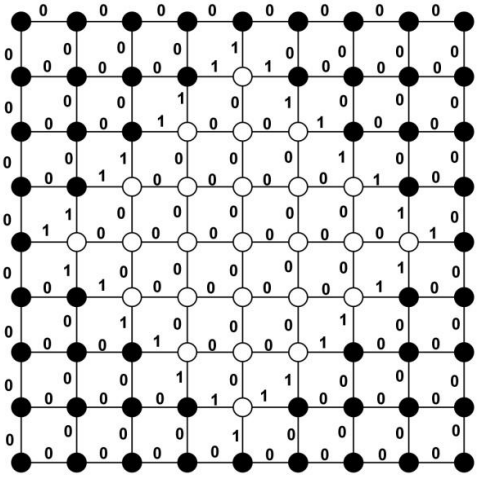
Şekil 3.6. Bir graftan çıkarılabilen örnek ağaçlar ve MYA

$T = (V, E_T)$ ağaç temsili $G = (V, E)$ grafındaki bütün V düğüm kümesini içermektedir. T_{MYA} ağaç yapısı G grafından çıkarılan MYA yapısıdır. Eğer T ağacı içerisindeki düğüm sayısını gösteren $|V|$ değeri n ise, T ağacı içerisindeki hat sayısını gösteren $|E_T|$ değeri ise $n - 1$ kadardır (Kruskal, 1956)

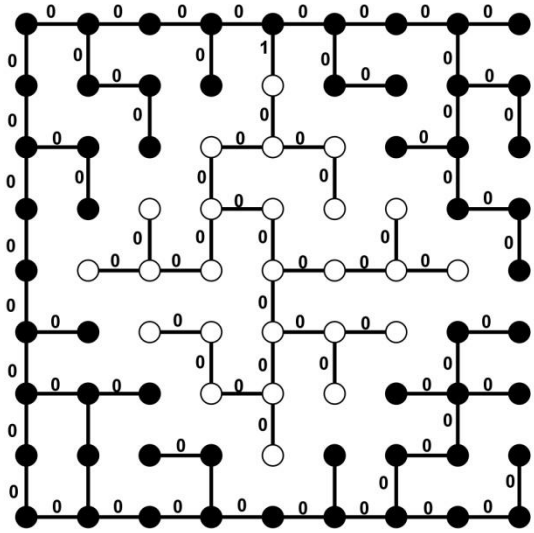
Düğümlerden oluşan bir topluluk, içerisinde birbiri ile bağlantısı olmayan birden fazla minimum yayılan ağaç barındırabilmektedir. Bu yapı da en küçük yayılan ağaç ormanı (minimum spanning forest) olarak adlandırılmaktadır (Boruvka, 1926).



(a) İki renkten oluşan örnek görüntü pikselleri



(b) Örnek görüntüden oluşturulmuş 4 bağlantılı komşuluktan oluşan graf yapısı



(c) Örnek görüntü grafından çıkarılmış bir MYA yapısı

Şekil 3.7. Bir görüntünün MYA yapısı

MYA ile görüntü bölütleme ise, görüntünün piksellerinin kümelere ayrılarak anlamlı alt bölgeler elde edilmesidir. İlk olarak, 2D bir görüntüde, düğüm olarak kabul edilen görüntünün pikselleri arasına farklılıklarına göre ağırlıklandırılmış hatlar çizilerek düzlemsel bir graf oluşturulur. Her düğüm bağlantılı olduğu benzer özelliğe sahip komşu düğümlerle döngü oluşturmayacak şekilde birleşerek minimum yayılan ağaç veya ağaçlar (alt bölgeler) oluşturmaktadır (Şekil 3.7) (Morris ve ark, 1986; Xu ve Uberbacher, 1997; Felzenszwalb ve Huttenlocher, 2004; Ning ve ark, 2010; Peng ve ark, 2013; Wang ve ark, 2014).

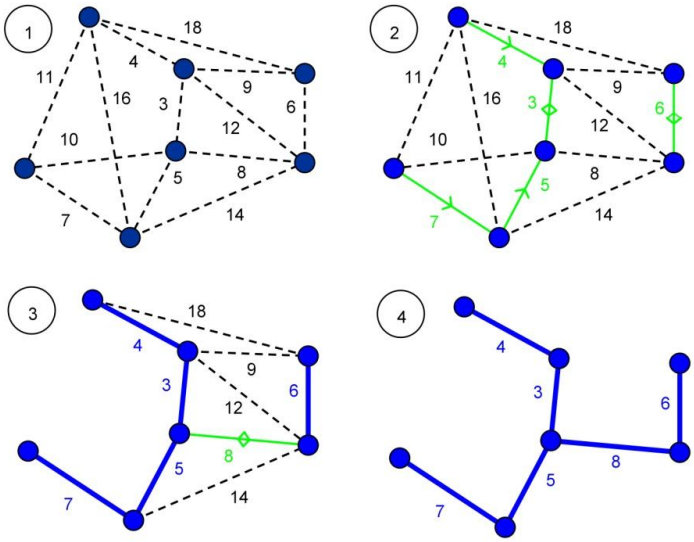
3.2.1. Minimum yayılan ağaç algoritmaları

Bir grafın MYA yapısını çıkarmak için literatürde farklı algoritmalar bulunmaktadır. Bu algoritmalardan en çok kullanılanları Boruvka, Prim, Kruskal ve Ters Çevir-Sil algoritmalarıdır. Bu algoritmalar "Açgözlü (Greedy) Algoritmalar" olarak adlandırılmaktadırlar (Black, 2005). Yani en uygun sonuca ulaşmak için her adımda uygulanacak belirli bir stratejiyi izlerler. Açgözlü denmesinin sebebi algoritmanın her adımında, sonraki adımları düşünmeden sadece bulunduğu durumdaki en iyi seçeneğe yönelmesindendir. Bu algoritmalar zaman karmaşıklığı açısından problemin sonlanma şekline ve MYA'daki hatların sınırlandırılıp sınırlandırılmayacağına bağlı olarak FP (Function Problems) veya P (Polinomial Time) problemlerdir (Kruskal, 1956).

3.2.1.1. Boruvka algoritması

İlk algoritma 1926 yılında Çek bilim adamı Otakar Boruvka tarafından geliştirilen Boruvka algoritmasıdır. Moravia şehri için etkili bir elektrik dağıtım ağını amaçlamıştır. Bu algoritma bir aşamalar dizisinden oluşur. Boruvka adımı olarak adlandırılan her bir aşamada, G grafındaki her bir düğüm birbirinden bağımsız alt ağaçlar ve G grafı da en küçük ağırlıklı ağaçlardan oluşan bir F_1 ormanını oluşturur. Böylece, algoritma $G_1 = G/F_1$ grafını bir sonraki adım için giriş verisi olarak oluşturur. Burada G/F_1, F_1'de ağaçlar arasında hatlar oluşturarak G'den türeyen grafı temsil etmektedir. Her Boruvka adımı lineer düzeyde zaman alır. Düğüm sayısı her adımda en az yarıya düşer ve algoritmanın zaman karmaşıklığı $O(m \cdot \log n)$ kadardır. Burada m graftaki toplam hat sayısını ve n düğüm sayısını temsil eder. Bu algoritmada aynı adım

içerisinde, bütün alt ağaçlar birbirinden bağımsız bir şekilde hareket ettiği için bu algoritma paralel işlem için en uygun MYA bulma algoritması olarak görülmüştür (Boruvka, 1926; Karger ve ark, 1995; Nesetril ve ark, 2001; Bader ve Cong, 2006).



Şekil 3.8. Boruvka algoritması ile bir grafın MYA yapısının çıkarılması

Bu algoritmaya göre, ilk başta G grafındaki her bir düğüm birer ağaç olarak varsayılır ve F ormanına eklenir. İkinci adım olarak, her bir ağaç G grafındaki kendisi ile bağlantılı olan hatlardan en az ağırlıklı olanını seçer. Seçilen hatlar bir döngü oluşturmayacaksa F ormanına eklenir. Eğer F ormanında tek bir ağaç kaldıysa MYA oluşmuştur ve işlem sonlanır. Birden fazla ağaç varsa tekrar ikinci adıma gidilir (Şekil 3.8). Boruvka algoritmasına ait işlem adımları aşağıda açıklanmaktadır.

Adım 1 - Her bir düğüm bir ağaç olarak görülmekte ve bütün graf bir ağaç ormanı olarak görülmektedir.

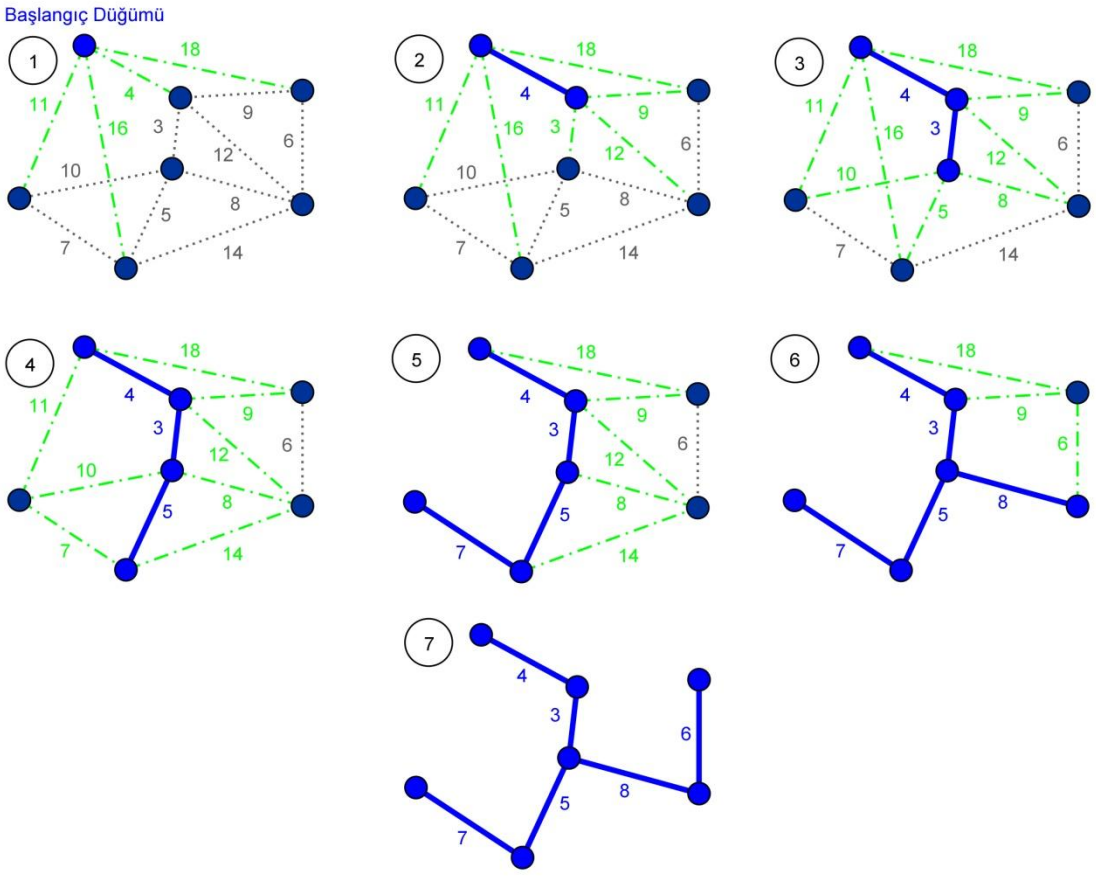
Adım 2 – Her ağaç kendi içerisinde olmayan ama kendisi ile bağlantılı olan hatlardan en az ağırlıklı olanını seçmektedir.

Adım 3 - Her ağaç Adım 2'de seçtiği hattın öbür ucundaki ağaç ile döngü - oluşturmamak şartıyla birleştirilmektedir. (Birleşme işlemi herhangi bir sırada yapıldığı için birleşme esnasına kadar birleştirilecek ağaçlar başka ağaçlarla birleştirilmiş olabilmekte ve bu da döngü oluşmasına sebep olabilmektedir.)

Adım 4 - Ağaç ormanında tek bir ağaç kaldıysa işlem sonlandırılmakta, birden fazla ağaç varsa Adım 2'ye gidilmektedir.

3.2.1.2. Prim algoritması

Bu algoritma 1930 yılında matematikçi bilim adamı Vojtěch Jarník tarafından ortaya atılmıştır (Jarník, 1930). Daha sonra bilgisayar bilimcileri olan Robert C. Prim tarafından (Prim, 1957) ve Edsger W. Dijkstra tarafından (Dijkstra, 1959) tekrar keşfedilmiştir. Basitçe, ilk başta sadece belirlenen bir düğümü içeren MYA (T_{MYA}) kendisine en yakın düğümü seçerek ve kendisine dâhil ederek her adımda bir hat ve bir düğüm daha genişlemektedir. Diğer bir deyişle T_{MYA} kendisine bağlantısı olan hatlardan ağırlık değeri en az olan (v, u) hattı kadar artırılarak işlem adımları devam etmektedir. Öyle ki, bu genişleme işlemi v düğümü T_{MYA} ağacının içerisinde ve u düğümü henüz T_{MYA} ağacının dışında iken gerçekleştirilmektedir. İşlem G grafındaki bütün düğümler ağaç yapısına eklendikten sonra sonlanmaktadır (Şekil 3.9).



Şekil 3.9. Prim algoritması ile bir grafın MYA yapısının çıkarılması

Prim algoritmasına ait işlem adımları aşağıda açıklanmaktadır.

Adım 1 – Herhangi bir düğüm başlangıç düğümü olarak belirlenmektedir.

Adım 2 – Başlangıç düğümü, oluşturulmakta olan ve o aşamada düğüm kümesi boş olan MYA yapısına eklenmektedir.

Adım 3 – (v,u) bağlantısı, oluşturulmakta olan MYA ile bağlantılı herhangi bir hat ve v düğümü MYA yapısı içerisinde iken u düğümü MYA yapısı dışında olsun. Oluşturulmakta olan MYA yapısı ağırlık derecesi en düşük olan (v,u) hattını ve bu hattın diğer ucundaki u düğümünü MYA yapısı içerisine eklemektedir.

Adım 4 – Oluşturulmakta olan MYA yapısı graftaki bütün düğümleri içeriyorsa işlem sonlandırılmakta, değilse Adım 3'e gidilmektedir.

Bu algoritma, kullanılan veri yapısına bağlı olarak $O(m \cdot \log n)$ veya $O(m + n.\log n)$ zamanda çalışmaktadır (Prim, 1957; Cheriton ve Tarjan, 1976; Wang ve ark, 2014).

3.2.1.3. Kruskal algoritması

Bu algoritma da $O(m + n.\log n)$ zamanda çalışmaktadır. Graftaki bütün hatlar ağırlıklarına göre küçükten büyüğe doğru sıralanmakta ve en küçük hattan başlanarak sırasıyla başlangıçta hiç bir elemanı olmayan ve bütün düğümlerin bir ağacı temsil ettiği F ormanına eklenmektedir. Eğer eklenecek hat bulunduğu ağaç üzerinde döngü oluşturarak ağaç yapısını bozacaksa eklenmez ve sıradaki hat işleme alınır. Bu işlem F ormanı üzerinde tek bir ağaç kalana kadar devam etmektedir (Şekil 3.10) (Kruskal, 1956; Zahn, 1971; Morris ve ark, 1986; Felzenszwalb ve Huttenlocher, 2004).

Bir grafın bütün düğümlerini kapsayan MYA yapısı graftaki n adet düğüme sahipse, o halde MYA yapısında $n-1$ kadar da hat var demektir. (Bondy ve Murty, 1976). Bu hesapla F ormanına eklenen hat sayısından ormanda tek bir ağaç kalıp kalmadığı anlaşılabilir. En son kalan tek ağaç ise grafın MYA yapısını temsil etmektedir.

Kruskal algoritmasına ait işlem adımları aşağıda açıklanmaktadır.

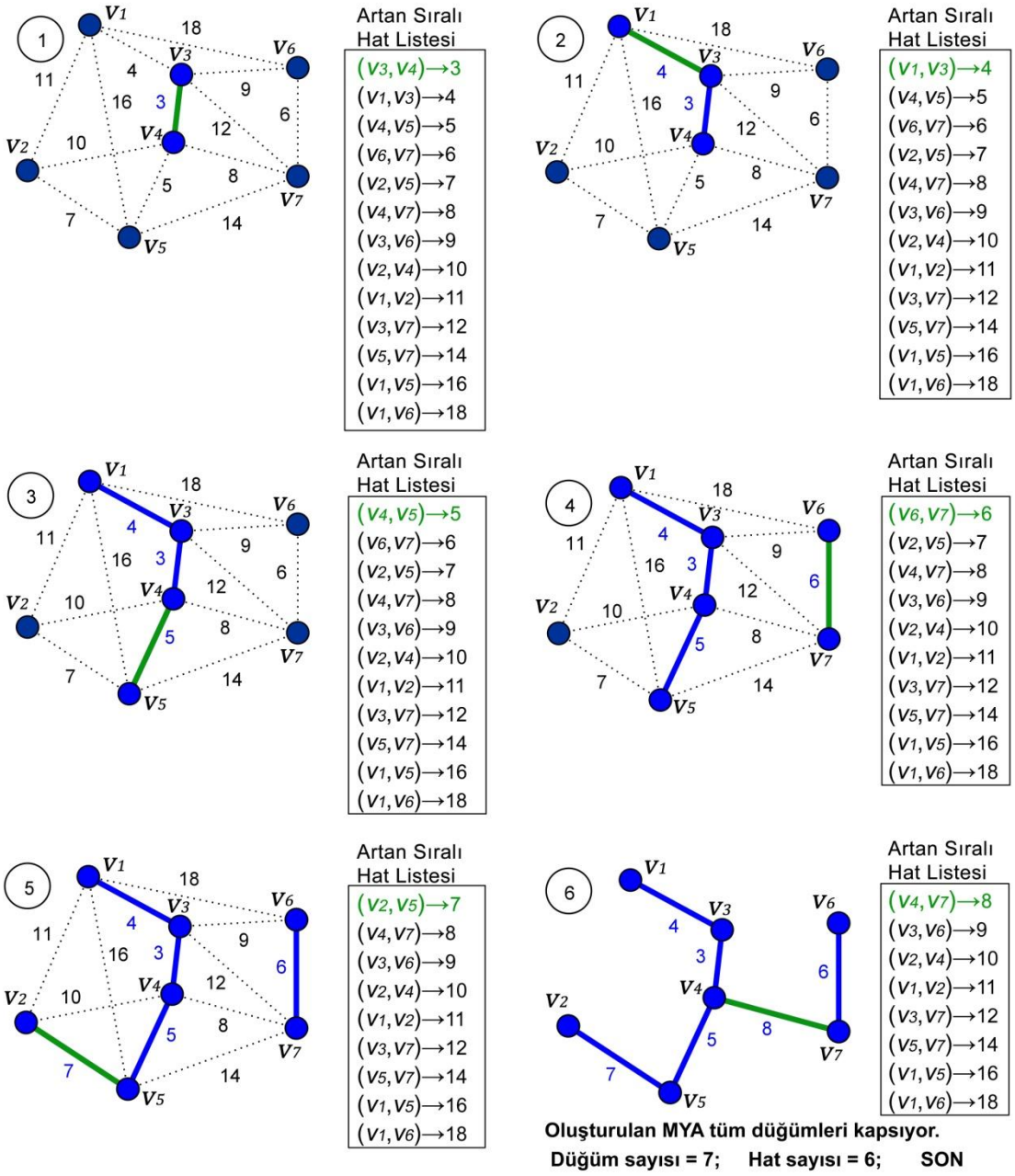
Adım 1 – Oluşturulacak MYA yapısı başlangıçta boştur ve graf üzerindeki bütün hatlar en düşük ağırlığa sahip olan hattan en yüksek ağırlığa sahip olan hatta doğru sıralı olacak şekilde boş bir hat listesine eklenmektedir.

Adım 2 – Sıralı hat listesinin en başındaki hat seçilmektedir.

Adım 3 – Seçilen hat, oluşturulmakta olan MYA yapısı içerisinde bir döngü oluşturulmayacaksa MYA yapısına eklenmektedir.

Adım 4 – Seçilen (listenin başındaki) hat listeden çıkarılmaktadır.

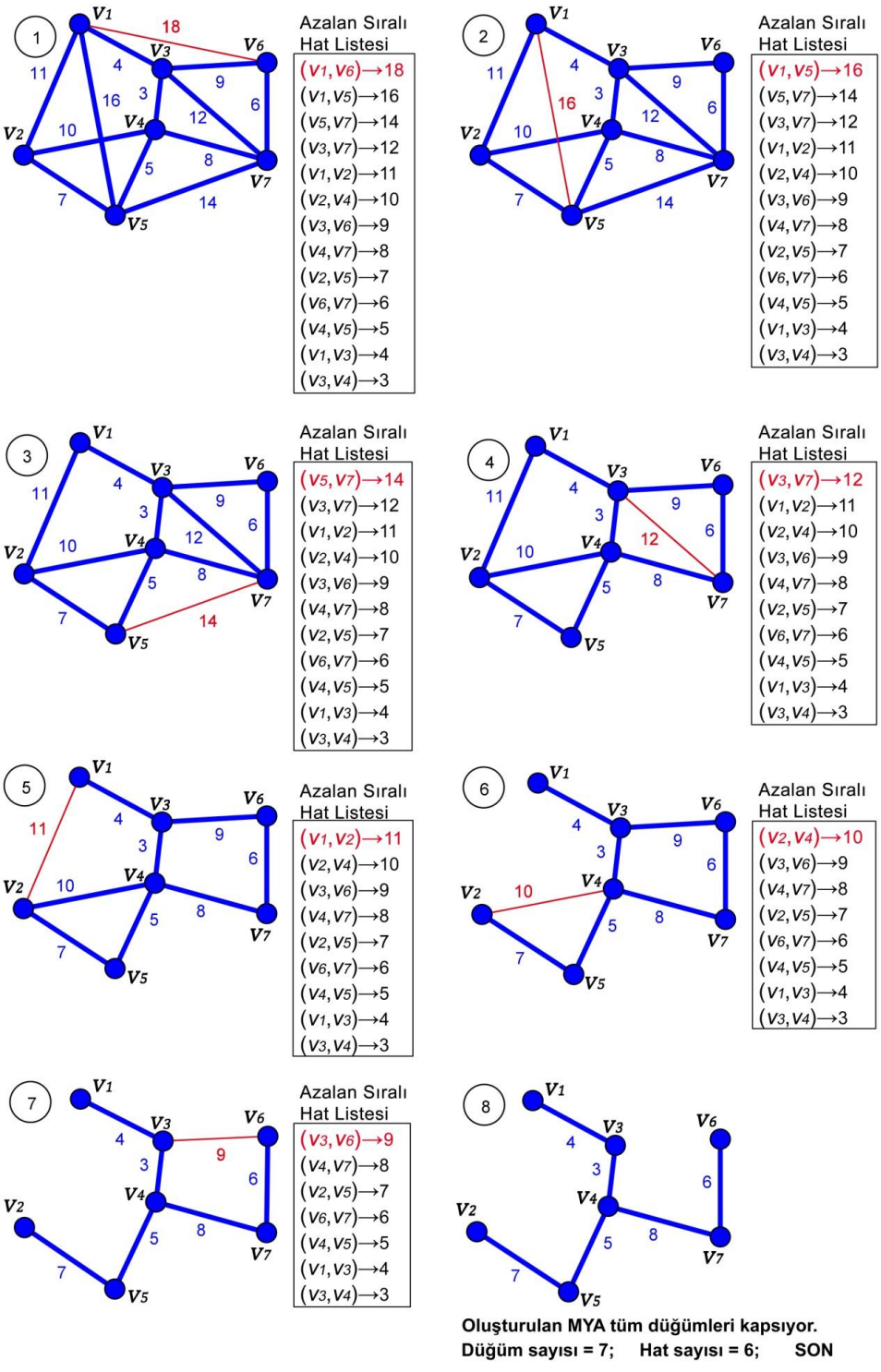
Adım 5 – Sıralı hat listesi boşsa veya graftaki n toplam düğüm sayısına karşılık, oluşturulmakta olan MYA yapısına $n-1$ sayıda hat eklenmişse işlem sonlandırılmakta, değilse Adım 2'ye gidilmektedir (Şekil 3.10).



Şekil 3.10. Kruskal algoritması ile bir grafın MYA yapısının çıkarılması

3.2.1.4. Ters çevir - sil algoritması

Bu algoritma Kruskal tarafından ortaya konulmuştur ama Kruskal algoritması ile karıştırılmamalıdır. Bu algoritma Kruskal algoritmasının tersi yönünde çalışmaktadır.



Şekil 3.11. Ters Çevir - Sil algoritması ile bir grafın MYA yapısının çıkarılması

Kruskal algoritmasında, başlangıçta düğümler üzerinde hatlar bulunmamakta ve hatlar küçükten büyüğe doğru döngü oluşturulmayacak şekilde F ormanındaki ağaçlar arasına eklenmektedir. Bu algoritmada ise başlangıçta, düğümler arasında verilen G grafındaki ile aynı şekilde bütün hatlar mevcuttur ve büyükten küçüğe doğru sırasıyla hatlar silinmeye başlanmaktadır. Eğer silinecek hat, graf bütünlüğünü bozacaksa yani bütün düğümleri kapsayacak olan MYA yapısını parçalayacak ise bu hat silinmez ve sıralamaya göre bir sonraki hat işleme alınmaktadır. Bu işlem Kruskal algoritmasında olduğu gibi n düğümlü bir ormanda $n-1$ sayıda hat kalana kadar devam etmektedir (Şekil 3.11) (Kruskal, 1956).

Ters çevir-sil algoritmasına ait işlem adımları aşağıda açıklanmaktadır.

Adım 1 – Graf üzerindeki bütün hatlar en yüksek ağırlığa sahip olan hattan en düşük ağırlığa sahip olan hatta doğru sıralı olacak şekilde, boş bir hat listesine eklenmektedir.

Adım 2 – Sıralı hat listesinin en başındaki hat seçilmektedir.

Adım 3 – Seçilen hat, graftan çıkarıldığı zaman birbiri ile bağlantısı olmayan alt graflar ortaya çıkarmayacaksa graftan çıkarılmaktadır.

Adım 4 – Seçilen (listenin başındaki) hat listeden çıkarılmaktadır.

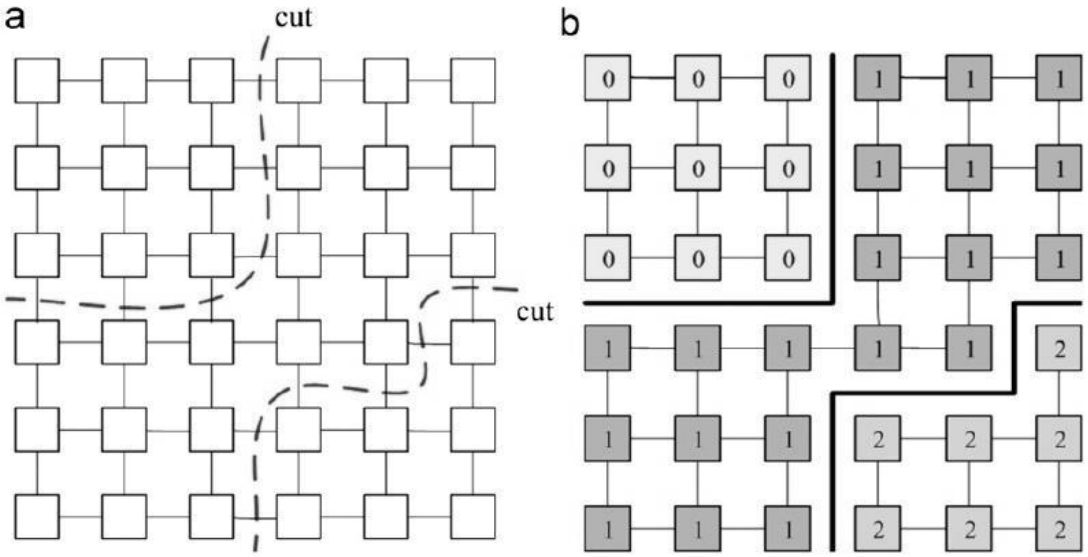
Adım 5 – Sıralı hat listesi boşsa veya graf yapısı üzerinde n toplam düğüm sayısına karşılık, graf üzerinde $n-1$ sayıda hat kalmışsa işlem sonlandırılmakta, değilse Adım 2'ye gidilmektedir (Şekil 3.11).

Bu algoritma ise $O(m \log n\,(\log\log n)^3)$ zamanda çalışmaktadır (Kleinberg ve Tardos, 2005; Chrobak ve ark, 2006; Mohanram ve Sudhakar, 2011).

3.3. Graf Tabanlı Görüntü Bölütleme Yöntemleri

1960'lara dönüldüğünde, ilk graf teorisi metotları, yakalanan algısal kümelerde benzerlik veya yakınlığın Gestalt (biçimsel) esaslarının önemini vurgulamıştır. Daha sonra graf, bu kriterlere göre kısımlara ayrılmıştır. Öyle ki, görüntüdeki her parça bir nesne olarak düşünülmüştür. Graf tabanlı yaklaşımların bu aşamasında, görüntü bölütleme işlemi kümeleme ve MYA'yı temel alan üstü kapalı bir işlem olmuştur (Durusoy, 2014). Bu metotlar, genellikle bölütleme sonuçları üzerinde yerel hesaplamalar için kullanılmıştır ve bütün görüntüler üzerinde genel bir bölütlemeyi garanti etmemişlerdir. Bu sorunun üstesinden gelebilmek için daha sonra özyinelemeli ve alt kümelerin özelliklerini de hesaba katan metotlar MYA tabanlı metotlara entegre

edilmiştir (Kwok ve Constantinides, 1997; Xu ve ark, 2002; Felzenszwalb ve Huttenlocher, 2004). Genel maliyet fonksiyonlu bölütlemeye genel bir yaklaşım olarak grafın takdim edilmesi 1990'larda Wu ile gelmiştir (Wu ve Leahy, 1993). Ondan sonra birçok araştırma odağı, grafta optimizasyon teknikleri çalışmasına kaymıştır. Görüntü bölütlemedeki sıkıntılardan birisi de görüntülerin doğasında var olan düzensiz bir yapıya sahip olmasıdır. Bir görüntüden birçok yorumlama çıkarılabileceği için, elde edilen görüntü bölütlemesinin doğruluğunun ölçülmesi zordur. Bu durum, ilgili nesnenin doğru bir şekilde çıkarılması için, görüntü bölütlemede orta ve yüksek seviye bilginin bölütleme işlemi ile birlikte kullanılması gerekliliğini ortaya koymuştur. 1990'ların sonlarında, modele özel ipuçları ile bağlamsal bilginin birlikte kullanılmasında etkili bir graf tekniği olan Graf Kesim (Graph Cut) algoritması ortaya çıkmıştır. Bu yöntem komşuluklarına göre birbirine hatlarla bağlı düğüm olarak düşünülen pikseller arasındaki hatları, piksel benzerliğine göre bir maliyet fonksiyonu uygulayarak kesme işlemidir. Küme içindeki pikseller benzer ve ayrı kümelerdeki pikseller ise farklı özelliklerde olacak şekilde bir optimizasyon uygulanır. Aynı küme içindeki düğümler aynı numara ile etiketlenir (Şekil 3.12) (Cox ve ark, 1996; Shi ve Malik, 2000; Boykov ve ark, 2001; Wang ve Siskind, 2001; Peng ve ark, 2013; Durusoy, 2014).



Şekil 3.12. a) Pikselleri temsil eden düğümlerin dört hatlı komşuluğu b) Graf kesme sonrası ve düğümlerin 0,1,2 olarak etiketlenmesi (Peng ve ark, 2013)

Graf teorisi teknikleri kabaca 5 kategoriye ayrılmaktadır (Peng ve ark, 2013; Durusoy, 2014):

1) Minimum yayılan ağaç tabanlı metotlar: Görüntü piksellerini minimum yayılan ağaç üzerinde gruplamak veya kümelemek amacıyla kullanılmaktadır. Graf düğümleri bağlantıları, belirlenmiş hat ağırlıklarının en küçük toplamını karşılar ve hatlar kaldırılarak farklı alt graflarla görüntü parçasına ulaşılır (Zahn, 1971; Morris ve ark, 1986; Kwok ve Constantinides, 1997; Xu ve Uberbacher, 1997; Xu ve ark, 2002; Felzenszwalb ve Huttenlocher, 2004; Haxhimusa ve Kropatsch, 2004).

2) Ağırlık fonksiyonlarıyla graf kesim: Graf kesim görüntü bölütlemenin doğal tanımıdır. Farklı kesme kriterleri kullanılarak, graf parçalama için genele hitap eden fonksiyonlar farklı farklı olacaktır. Bu fonksiyonları Ncut maliyet fonksiyonu (Shi ve Malik, 2000), bölgesel graf kesim fonksiyonu (Cox ve ark, 1996), ortalama kesim fonksiyonu (Wang ve Siskind, 2001), oran kesim (Rcut) fonksiyonu (Wang ve Siskind, 2003) ve minimal kesim fonksiyonu (Wu ve Leahy, 1993) olarak sayabiliriz. Genel özellikler ile bu fonksiyonlar optimize edilerek istenilen bölütleme elde edilmektedir.

Çizelge 3.1. Bazı Graf kesim maliyet fonksiyonları (Peng ve ark, 2013)

Algoritma	**Fonksiyon**	**Optimizasyon Metodu**	**Hesaplama Karmaşıklığı**	**Ön İşlem**
Minimal Kesim	$Mincut(A,B) = \sum_{u\in A, v\in B} w(u,v)$	Gomory-Hu'nun K-way maxflow algoritması	Polinomsal zaman	Kısa sınır
Ncut	$\mathrm{Ncut(A,B)} = \frac{cut(A,B)}{vol(A)} + \frac{cut(A,B)}{vol(B)}$	Genelleştirilmiş özsistemin çözümü	$O(m,n)$	Benzer ağırlıkları ayırma
Bölge Oranları	$\mathrm{Regioncut(A,B)} = \frac{cost(P)}{weight(P)}$	Minimum çözüm için yerel arama	$O(n \log n)$	Sınır yumuşatma
Ortalama Kesim	$Meancut(A,B) = \frac{cut(A,B\|w(u,v))}{cut(A,B\|1)}$	Minimum ağırlıkla en iyi eşleştirme	Polinomsal zaman	Yok
Oranlı Kesim	$Rcut(A,B) = \frac{cut1(A,B)}{cut1(A,B)}$	Baseline metod	$O(n^{\frac{7}{4}})$	Yok

3) Markov rassal alan (MRA) üzerinde graf kesim: Hedef, graf kesim fonksiyonlarında yumuşatma düzenlemesiyle yüksek düzeyde etkileşimli bilgiyi birleştirmektir. MRA bir Markov model özelliğine sahip rastgelen değişkenler dizisi içeren yönsüz bir graftır. Bayes ağlarından farkı yönsüz olması ve döngü içerebilmesidir. Bu yüzden MRA, Bayes ağları ile gösterilemeyen bazı bağlantıları temsil edebilir. Görüntüleri gürültüden arındırmak için MRA modeli oluşturulur ve

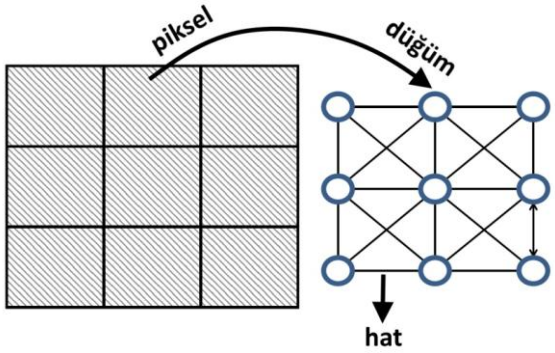
algoritmada kullanılan enerji fonksiyonuna en iyileme yapılarak sonuç görüntüler elde edilir (Kindermann ve Snell, 1980; Sağıroğlu ve Uçan, 2002; Li, 2009).

4) En kısa yol tabanlı metotlar: Nesne sınırları graf düğüm çiftleri arasındaki en kısa yol dizisinde belirlenmektedir. Bu metotlar, bölütlemeye rehberlik etmesi için kullanıcı etkileşimini gerektirir. Bu yüzden, bu işlem daha esnek ve dostça geri dönüş sağlayabilmektedir (Dijkstra, 1959; Agarwal ve ark, 2009; Chow ve ark, 2013).

5) Diğer metotlar: Yukarıdaki kategorilere konulmayan, baskın dizi (Pavan ve Pelillo, 2003) ve rastgele ilerleme (Grady, 2005) gibi metotlardır.

3.4. Görüntünün Bölütleme İşleminin Graf Tabanlı Kümeleme ile Temsili

Sayısal bir görüntü graf gibi düşünülebilir. Graf tabanlı görüntü işlemede pikseller bir graftaki düğümler olarak kabul edilmekte ve M piksel uzunluğunda ve N piksel genişliğinde bir 2D görüntü MxN boyutlarında iki boyutlu bir düğüm dizi olarak ele alınmaktadır. Her bir piksel, kullanılan renk uzayındaki renk değerlerinden oluşan bir değer vektörüne sahiptir. Yapılan tez çalışmasında RGB renk uzayı kullanılmıştır. RGB renk uzayına göre her bir piksel kırmızı (R), yeşil (G) ve mavi (B) renk değerlerinden oluşan bir renk değer vektörüne sahiptir ve her bir renk 0-255 değer aralığında bir tamsayı değeri ile ifade edilmektedir (Cheng ve ark, 2001). Bir görüntü grafında, komşu piksel çiftleri arasında yönsüz ve ağırlıklandırılmış bir hat olduğu varsayılmaktadır. Her bir piksel için komşuluk sayısı, çalışmada kullanılan komşuluk şekline göre değişebilir. Örneğin sadece yatay ve dikey olarak bitişiğinde olan pikseller hesaba katılırsa 4 komşuluklu bir graf çizilmiş olur. Çapraz olarak bitişik pikseller de hesaba katılırsa 8 komşuluklu bir graf çizilmiş olur (Şekil 3.13) (Morris ve ark, 1986; Lopresti ve Zhou, 2000).

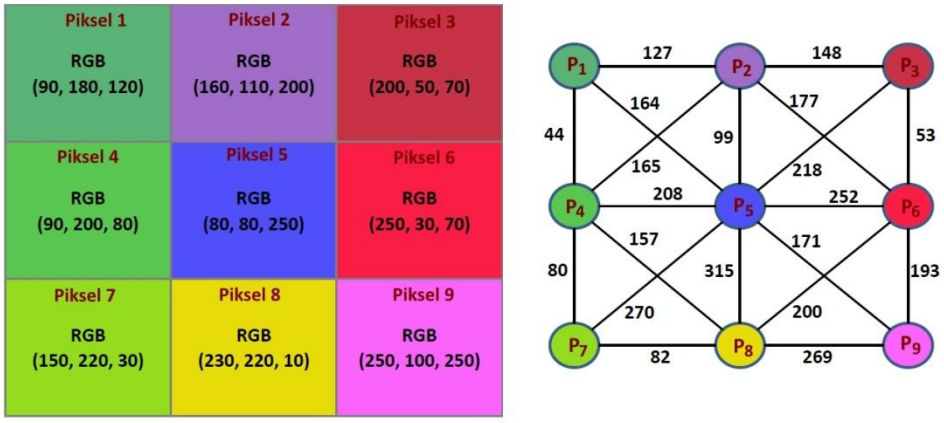


Şekil 3.13. Piksellerin 8 komşuluklu graf temsili

Bir görüntünün graf temsilindeki hatların ağırlık değerleri ise iki komşu (bitişik) piksel çifti arasındaki renk değerleri farklılığından elde edilebilir. RGB renk uzayında renk değerlerine sahip olan bir görüntünün, herhangi komşu iki pikselinin RGB uzayındaki değer vektörlerinin farkı için Öklit mesafesi (Euclidean distance) ölçümü kullanılarak, aralarındaki bağlantıyı kuracak hattın ağırlık değeri Denklem 3.1'e göre elde edilebilmektedir.

$$d_e(p,q) = \sqrt{\left(R_p - R_q\right)^2 + \left(G_p - G_q\right)^2 + (B_p - B_q)^2} \tag{3.1}$$

Burada p ve q piksellerinin RGB renk değerini tutan (R_p, G_p, B_p) ve (R_q, G_q, B_q) değer vektörleri arasındaki Öklit mesafesi $d_e(p,q)$ hesaplanmaktadır (Cheng ve ark, 2001). Bu hesaplamayla bütün komşu pikseller arasındaki renk vektörlerinin farkları hesaplanarak hat ağırlık değerleri bulunmuş olur (Şekil 3.14).

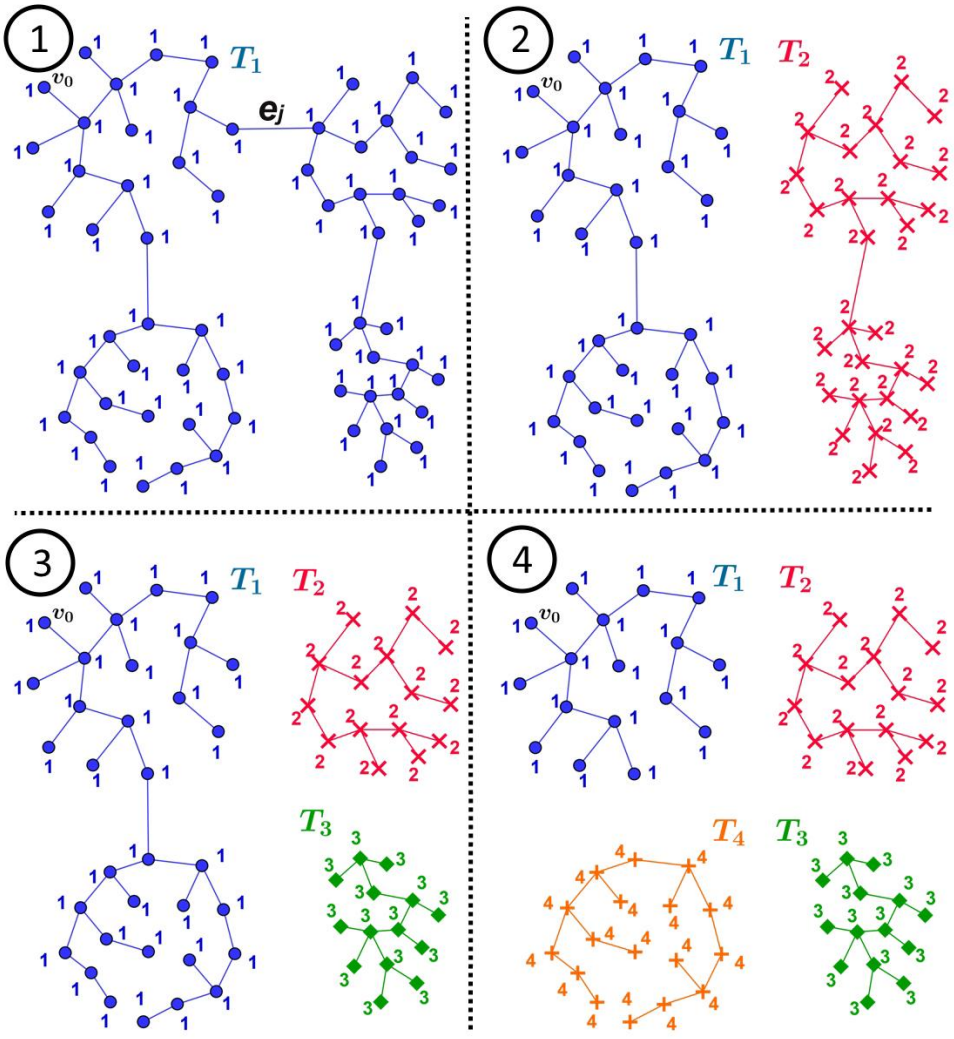


Şekil 3.14. Renkli bir görüntünün RGB renk uzayına göre piksel değerleri ve 8 komşuluklu graf temsilindeki Öklit mesafesine göre hesaplanmış hat ağırlık değerleri

Sayısal bir görüntüye göre yapılandırılmış olan $G = (V, E)$ grafı yönsüz ve ağırlıklandırılmış bir graf olsun. $V = \{v_1, v_2, \ldots, v_n\}$ düğüm kümesi görüntüdeki pikselleri ve $E = \{e_1, e_2, \ldots, e_m\}$ hat kümesi de komşu pikseller arasındaki hatları içermektedir. $i = 1{:}n$ ve $j = 1{:}m$ olarak düşünüldüğünde, her bir v_i düğümü verilen görüntü verisindeki bir pikseli ve değerlerini, ve her bir $e_j = (u, v)$ hattı ise u ve v düğümleri arasındaki bağlantıyı ve bu düğümler arasındaki farklılıkları veya benzerlikleri temsil etmektedir. Her bir düğüm kendisine komşu olan düğümlerin bir listesini tutar. Örneğin $Komşu[u]$ listesi u düğümüne komşu olan düğümlerin listesini

tutar ve (u, v) hat komşuluğuna göre v düğümü $Komşu[u]$ listesi içerisinde yer almaktadır. Örneğin; Şekil 3.14'e göre $Komşu[\mathrm{P}_1] = \{\mathrm{P}_2, \mathrm{P}_4, \mathrm{P}_5\}$ veya $Komşu[\mathrm{P}_5] = \{\mathrm{P}_1, \mathrm{P}_2, \mathrm{P}_3, \mathrm{P}_4, \mathrm{P}_6, \mathrm{P}_7, \mathrm{P}_8, \mathrm{P}_9\}$ olmaktadır.

Graf tabanlı görüntü bölütlemede, görüntü bölütleme işlemi görüntüden elde edilen grafı parçalama problemine indirgenmektedir (Morris ve ark, 1986; Xu ve Uberbacher, 1997; Felzenszwalb ve Huttenlocher, 2004). Bu fikre dayanarak, bir görüntünün her bir C alt kümesi, ilgili görüntünün bir alt bölütünü temsil etmektedir. Benzer bir şekilde, MYA tabanlı kümeleme ve görüntü bölütleme metotlarında, graf parçalama problemi de ağaç parçalama problemine indirgenmektedir. Bu yönteme dayanan metotlarda da her bir alt ağaç, bir alt grafı temsil etmektedir (Şekil 3.15) (Morris ve ark, 1986; Xu ve Uberbacher, 1997; Felzenszwalb ve Huttenlocher, 2004; Zhong ve ark, 2011; Wang ve ark, 2014).



Şekil 3.15. Adım adım bir ağacın alt ağaçlara bölünmesi ve düğümlerin etiketlenmesi

$T = (V, E_T)$ MYA yapısı için eğer $E_T \neq \emptyset$ ve $e_j \in E_T$ iken e_j hattı kesilirse (MYA'dan kaldırılırsa), T ağacı $T_1 = (V_1, E_1)$ ve $T_2 = (V_2, E_2)$ şeklinde iki ayrı alt ağaç yapısına ayrılmış olur (Şekil 3.15). Bu ağaç parçalama işlemi ise $V = V_1 \cup V_2$, $V_1 \cap V_2 = \emptyset$, $E_1 \cap E_2 = \emptyset$, $e_j \notin E_1$, $e_j \notin E_2$ ve $E_T = E_1 \cup E_2 \cup e_j$ şeklinde sonuçlanır. Böylece her iki T_1 ve T_2 alt ağaçları sırasıyla C_1 ve C_2 alt kümelerinin MYA'ları olmuş olur ve her bir alt ağaç görüntünün bir bölütünü temsil eder (Xu ve Uberbacher, 1997).

3.4.1. Graf tabanlı kümeleme problemlerinin tanımları

Kümeleme, eldeki verileri "küme" olarak adlandırılan gruplara ayırma işlemidir. Öyle ki, her bir küme kendi içinde birbirine benzer olan ama diğer kümelerdekilerden farklı olan elemanlardan oluşmaktadır (Wang ve ark, 2014). Kümeleme problemi görüntü bölütleme (Wu ve Leahy, 1993), örüntü tanıma (Zhong ve ark, 2011), veri madenciliği (Devi ve Devi, 2014), bilgisayarlı görme (Robles-Kelly ve Hancock, 2004) ve biyoinformatik (Xu ve Wunsch, 2005; Yu ve ark, 2007) gibi birçok çalışma alanında ortaya çıkmaktadır ve kümeleme araçlarından yararlanılmaktadır. Yapılan tez çalışmasında üzerinde durulan görüntü bölütleme işlemi, görüntüyü kendi içinde gruplara ayırdığı ve kümelemedeki gibi görüntü içindeki piksellerin benzerlikleri ve farklılıkları göz önünde bulundurulduğu için aslında bir kümeleme işlemi olarak kabul edilebilir.

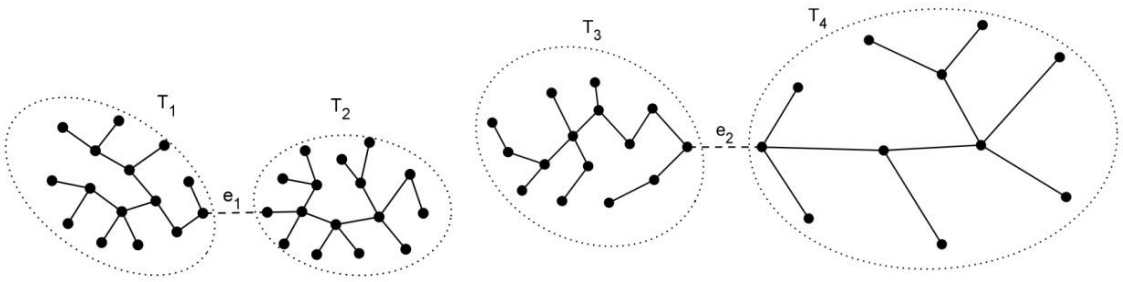
Bir T ağacının içerisindeki en fazla ağırlıklı hattın ağırlık değeri "iç farklılık" (internal difference) olarak isimlendirilmektedir ve $Int(T)$ şeklinde ifade edilmektedir. (Felzenszwalb ve Huttenlocher, 2004).

MYA üzerinden kaldırılacak hatlar için uygulanan hat kesme kriterini karşılayan hatlar "uyumsuz hat" (inconsistent edge) olarak isimlendirilmektedir. Örneğin; Şekil 3.16'daki e_1, e_2, e_3 ve e_4 hatları uyumsuz hatlardır (Zahn, 1971; Grygorash ve ark, 2006).

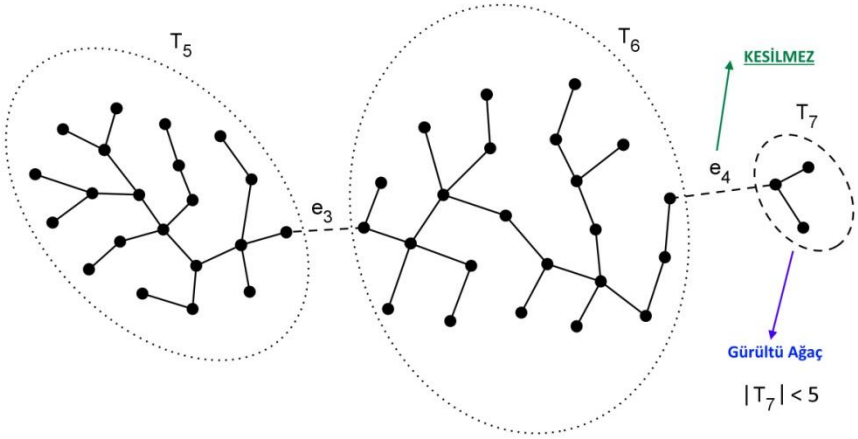
Bir ağacı iç farklılık açısından benzer iki alt ağaca bölen ve her iki alt ağacın da iç farklılıklarından daha fazla ağırlık değerine sahip olan uyumsuz hatlar "belirgin hat" (explicit edge) olarak isimlendirilmektedir. Örneğin; Şekil 3.16'daki T_1 ve T_2 ağaçları iç farklılık açısından benzerdirler yani ikisinin de en yüksek ağırlıklı hattı $w(e_1)$ değerinden daha küçüktür. Diğer bir ifadeyle $w(e_1) > Int(T_1)$ ve $w(e_1) > Int(T_2)$

olduğu düşünülmektedir. Bu şartlar altında e_1 hattı "belirgin hat" olarak isimlendirilebilir.

Bir uyumsuz hat, ağacı iç farklılık açısından birbirinden farklı iki alt ağaca bölüyorsa ve ağırlık değeri bu alt ağaçlardan birisinin iç farklılığından daha fazla iken diğerininkinden daha az ise bu hat "geçiş hattı" (transition edge) olarak isimlendirilmektedir. Örneğin; Şekil 3.16'daki T_3 ve T_4 ağaçları iç farklılık açısından birbirinden farklıdırlar ve $w(e_2) > Int(T_3)$ ve $w(e_2) < Int(T_4)$ olduğu düşünülmektedir. Bu şartlar altında e_2 hattı "geçiş hattı" olarak isimlendirilebilir.



(a) Belirgin hat e_1 **(b)** Geçiş hattı e_2



(c) Gürültü alt ağaç T_7 (Olabilecek en küçük küme büyüklüğü limiti = 5 için)

Şekil 3.16. Uyumsuz hatlar (e_1, e_2, e_3 ve e_4) ve alt ağaçlar ($T_1, T_2, \ldots, T_7$)

Eğer bir alt ağacın düğüm veya hat sayısı kullanıcı tarafından belirlenen en küçük küme büyüklüğünü gösteren parametre değerinden daha küçük ise bu alt ağaç "gürültü ağaç" (noisy tree) olarak adlandırılmaktadır. Daha önce de bahsedildiği gibi, her bir alt ağaç görüntünün bir alt bölütünü temsil etmektedir. Kullanılan görüntü bölütleme yöntemi bazı çalışmalarda (Morris ve ark, 1986; Xu ve Uberbacher, 1997)

gürültü diye adlandırılan bazı istenmeyen küçük alt bölümler oluşturabilmektedir. Şekil 3.16'da T_7 alt ağacının belirlenen en küçük küme büyüklüğü limitinden daha küçük olduğu varsayılmaktadır. Bu yüzden bu alt ağaç "gürültü ağaç" olarak görülebilir. Burada istenilen bölütlemeye göre T_7 alt ağacı, T_6 alt ağacından ayrılmamalıdır ve bunun için de e_4 hattı kesilmemelidir.

Birçok kümeleme ve görüntü bölütleme uygulamasında, gürültü bölütleri problem oluşturmuştur. Bu problemin üstesinden gelebilmek için bazı metotlar görüntü üzerindeki genel bilgileri kullanma yöntemlerinden faydalanmış (Xu ve ark, 2002), bazıları ise görüntü bölütleme işlemi tamamlandıktan sonra gürültüleri gidermek için gürültü bölütleri uygun görülen komşu bölütler ile birleştirme yöntemlerinden faydalanmışlardır (Felzenszwalb ve Huttenlocher, 2004). Yapılan tez çalışmasında, önerilen algoritma bölütleme işlemi sırasında gürültü bölüt oluşmasını engellediği için gürültü giderme ile ilgili ek bir işlem yapılmasına gerek duyulmamaktadır.

3.5. Prim Algoritması ile Bir Görüntünün MYA Yapısının Sıralı Temsili

Bir $G = (V, E)$ grafı için, MYA'nın Prim algoritmasının sıralı temsili (MYA-PST veya kısaca PST) grafın MYA'sı olan $T = (V, E_T)$ ağacı için $T_{PST}(v_s, V_T, Q_E, Q_W)$ olarak ifade edilmektedir. Burada v_s gelişi güzel seçilen bir başlangıç düğümünü göstermektedir. Düğüm kümesi V_T, hat kuyruğu Q_E ve ağırlık değeri kuyruğu olan Q_W başlangıç olarak boştur. Bu notasyon Wang ve arkadaşlarının veri kümeleme ile ilgili olan çalışmasındaki (Wang ve ark, 2014) notasyona benzemektedir.

Bir $e_c = (v_{c1}, v_{c2})$ hattı için, eğer $v_{c1}, v_{c2} \in V$, $v_{c1} \in V_T$, $v_{c2} \notin V_T$ ve $v_{c2} \in Komşu[v_{c1}]$ ise bu hat v_{c1} düğümü için bir "aday hat" (candidate edge) olarak adlandırılmaktadır. Hattın diğer her iki ucundaki düğümler V_T düğüm kümesine aitse oluşturulmakta olan MYA üzerinde döngü oluşturacağı için bu hat aday hat değildir. E_C hat kümesi ise algoritmanın mevcut durumundaki V_T düğüm kümesine ait olan düğümlerin hepsinin aday hatlarını tutan bir aday hat kümesidir (Şekil 3.17). min (E_C) ise E_C aday hat kümesi içerisindeki en az ağırlıklı olan hattı ifade etmektedir.

Prim algoritmasına göre (Prim, 1957) ilk başta v_s düğümü başlangıçta boş olan V_T düğüm kümesine eklenir. Daha sonra v_s düğümünün komşu düğümleri olan $Komşu[v_s]$ listesindeki düğümler ile arasındaki bağlantıları sağlayan aday hatlar E_C aday hat kümesine eklenir. Bundan sonraki her bir çevrimde $(u, v) =$ min (E_C) hattı E_C

kümesinden silinir ve v düğümü V_T kümesine, $e_t^Q = (u, v)$ hattı Q_E hat kuyruğunun sonuna, $w_t^Q = w(u, v)$ ağırlık değeri Q_W kuyruğunun sonuna ve v düğümünün aday hatları da E_C kümesine eklenir. Eklenen herhangi bir (u, v) aday hattı için, eğer E_C içinde $v = v_{c2}$ olacak şekilde herhangi bir (v_{c1}, v_{c2}) hattı yoksa hat kümeye eklenir, eğer varsa ama $w(u, v) < w(v_{c1}, v_{c2})$ ise E_C içindeki (v_{c1}, v_{c2}) hattı (u, v) olarak güncellenir, eğer bu da değilse E_C üzerinde hiçbir değişiklik yapılmaz. Örneğin; Şekil 3.17'de ikinci aşamada (v_1, v_5) ve (v_4, v_5) hatlarının her ikisi de oluşturulmakta olan MYA ile v_5 düğümü arasında bir bağlantı sağlayabilmektedir. Ancak (v_4, v_5) hattının ağırlığı (v_1, v_5) hattının ağırlığından daha yüksek olduğu için Fibonacci Heap veri yapısına göre (v_4, v_5) hattı aday hatlar listesine eklenmemiştir. Artık (v_4, v_5) hattı ile herhangi bir işlem yapılmayacağı için şekil üzerinde de bu aşamadan itibaren bu hat gösterilmemektedir. Şekil 3.17'de ikinci aşamada E_C hat listesinde olan (v_4, v_8) hattının ağırlığı (v_7, v_8) hattının ağırlığından daha yüksek olduğu için üçüncü aşamada (v_4, v_8) hattı listeden çıkarılmış, yerine (v_7, v_8) hattı eklenmiştir. Kısaca belirmek gerekirse; E_C hat listesinde, oluşturulmakta olan MYA ile MYA dışındaki bir düğüm arasındaki bağlantıyı kuran hatlardan yalnızca en düşük ağırlıklı olan hat yer alabilmektedir.

Graf içinde toplam düğüm sayısı n ise, oluşturulacak MYA üzerindeki hat sayısı da $n - 1$ olacaktır. Bu durumda e_t hattının ve w_t ağırlık değerinin indeks değeri olan t değeri 1'den başlayarak ve her çevrimde 1 artarak $n - 1$ değerine kadar olan tam sayı değerlerini almaktadır. e_t^Q hattı Q_E kuyruğunun $t.$ indeksindeki hattı ve bu hattın ağırlık değeri olan w_t^Q değeri de Q_W kuyruğunun $t.$ indeksindeki değeri ifade etmektedir.

Bütün bu işlemler, bir algoritma akışı olarak adım adım aşağıdaki şekilde tanımlanabilmektedir.

Girdi : Bir görüntüden alınmış $G = (V, E)$ grafı

Çıktı : G grafından çıkarılacak olan $T_{PST}(v_s, V_T, Q_E, Q_W)$ sıralı MYA temsili

Adım 1 – Başlangıç değerleri: $t = 1$, $V_T \leftarrow \emptyset$, $Q_E \leftarrow \emptyset$, $Q_W \leftarrow \emptyset$ ve $E_C \leftarrow \emptyset$

Adım 2 – V düğüm kümesi içerisinden gelişigüzel bir v_s belirle

Adım 3 – $v_s$düğümünü V_T düğüm kümesine ekle

Adım 4 – v_s düğümünün aday hatlarını E_C hat kümesine ekle

Adım 5 – $(u, v) = \min(E_C)$ hattını bul ve E_C kümesinden sil

Adım 6 – v düğümünü V_T kümesine ekle

Adım 7 – (u, v) hattını Q_E kuyruğunun sonuna ekle

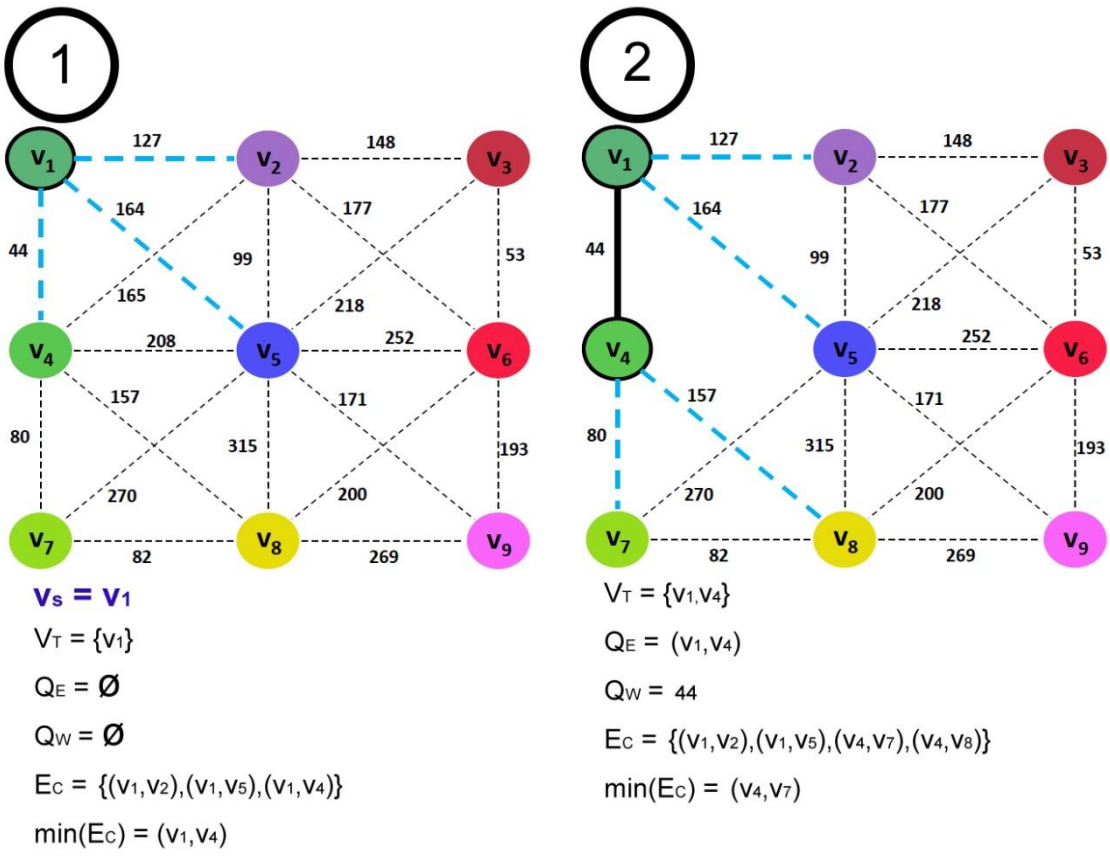
Adım 8 – $w(u, v)$ değerini Q_W kuyruğunun sonuna ekle

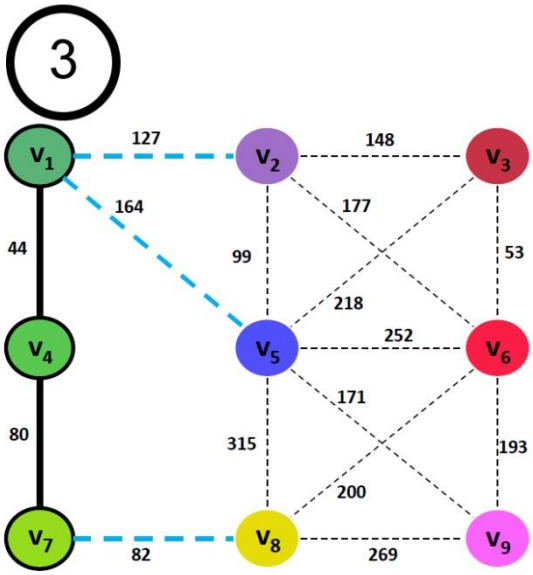
Adım 9 – v düğümünün aday hatlarını E_C hat kümesine ekle. Eklenen herhangi bir (u, v) aday hattı için, eğer E_C içinde $v = v_{c2}$ olacak şekilde herhangi bir (v_{c1}, v_{c2}) hattı yoksa hat kümeye eklenir, eğer varsa ama $w(u, v) < w(v_{c1}, v_{c2})$ ise E_C içindeki (v_{c1}, v_{c2}) hattı (u, v) olarak güncellenir, eğer bu da değilse E_C üzerinde hiçbir değişiklik yapılmaz.

Adım 10 – $|V_T| < |V|$ veya $E_C \neq \emptyset$ ise 5. adıma git, yoksa işlemi sonlandır.

Bu algoritma ile basit bir 3x3 boyutunda görüntüden çıkarılmış bir 8 komşuluklu graf yapısının PST-MYA yapısının çıkarılması adım adım Şekil 3.17'de gösterilmektedir. Şekilde örnek olarak Şekil 3.14'deki görüntü grafı ele alınmış ve ilk başta v_s başlangıç düğümü olarak v_1 düğümü seçilmiştir.

Prim algoritmasının zaman karmaşıklığı $O(E \cdot \log V)$ kadardır (Prim, 1957). Oluşturulmakta olan MYA için en az ağırlıklı olan aday hattı bulma işlemi önemli miktarda işlem zamanı tüketmektedir. Eğer PST-MYA algoritmasında Fibonacci Heap veri yapısı ve metotları kullanılırsa zaman karmaşıklığı $O(E + N \cdot \log N)$ zamana düşmektedir (Fredman ve Tarjan, 1987; Cormen ve ark, 2001).





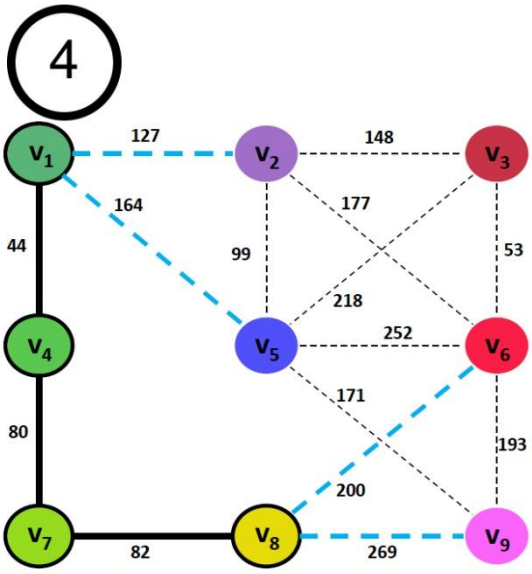
$V_T = \{v_1, v_4, v_7\}$

$Q_E = (v_1, v_4) \leftarrow (v_4, v_7)$

$Q_W = 44 \leftarrow 80$

$E_C = \{(v_1, v_2), (v_1, v_5), (v_7, v_8)\}$

$\min(E_C) = (v_7, v_8)$



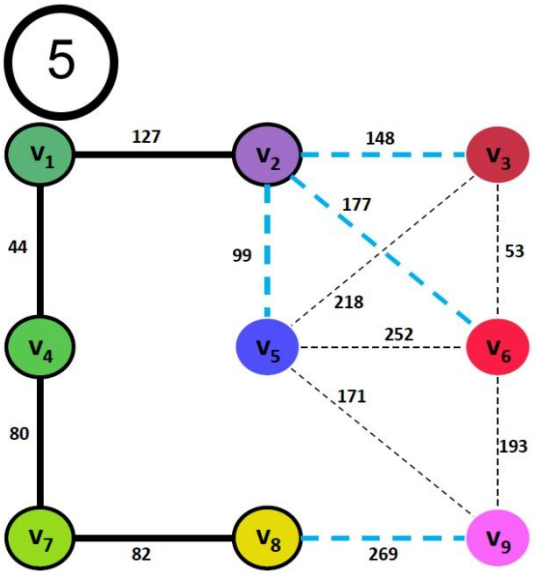
$V_T = \{v_1, v_4, v_7, v_8\}$

$Q_E = (v_1, v_4) \leftarrow (v_4, v_7) \leftarrow (v_7, v_8)$

$Q_W = 44 \leftarrow 80 \leftarrow 82$

$E_C = \{(v_1, v_2), (v_1, v_5), (v_8, v_6), (v_8, v_9)\}$

$\min(E_C) = (v_1, v_2)$



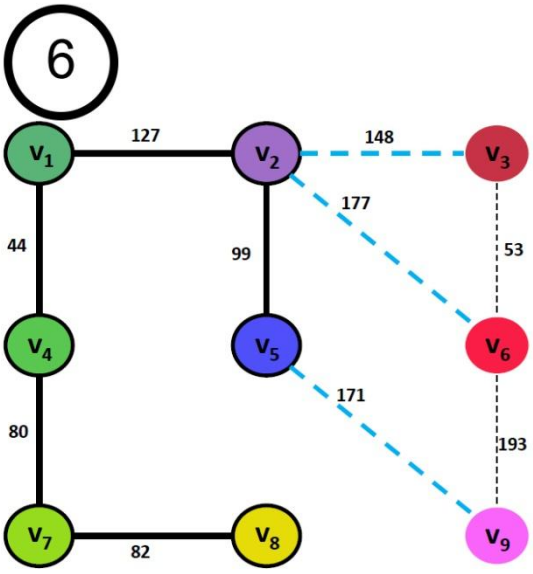
$V_T = \{v_1, v_4, v_7, v_8, v_2\}$

$Q_E = (v_1, v_4) \leftarrow (v_4, v_7) \leftarrow (v_7, v_8) \leftarrow (v_1, v_2)$

$Q_W = 44 \leftarrow 80 \leftarrow 82 \leftarrow 127$

$E_C = \{(v_2, v_3), (v_2, v_5), (v_2, v_6), (v_8, v_9)\}$

$\min(E_C) = (v_2, v_5)$



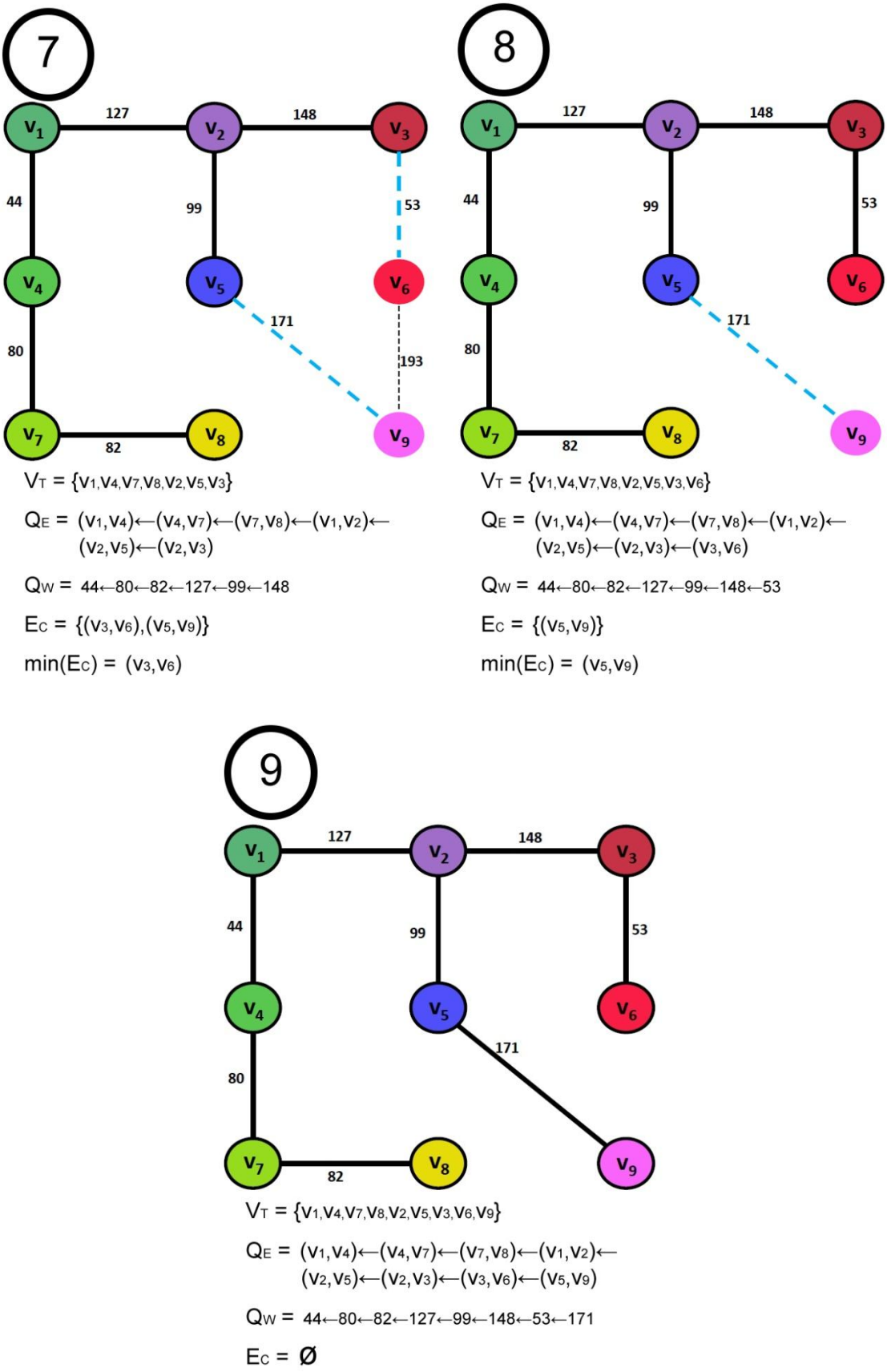
$V_T = \{v_1, v_4, v_7, v_8, v_2, v_5\}$

$Q_E = (v_1, v_4) \leftarrow (v_4, v_7) \leftarrow (v_7, v_8) \leftarrow (v_1, v_2) \leftarrow (v_2, v_5)$

$Q_W = 44 \leftarrow 80 \leftarrow 82 \leftarrow 127 \leftarrow 99$

$E_C = \{(v_2, v_3), (v_2, v_6), (v_5, v_9)\}$

$\min(E_C) = (v_2, v_3)$



Şekil 3.17. 3x3 görüntü grafından Prim algoritması ile PST-MYA yapısı çıkarma adımları

Fibonacci Heap veri yapısının kendine göre ekleme, silme ve güncelleme metotları vardır. Fibonacci Heap metotları aday hatları sıralı olarak ve döngü oluşmasına izin vermeyecek şekilde E_C hat kümesine eklemektedir (Zhong ve ark, 2015). Bununlar birlikte E_C kümesinden hat silme ve E_C kümesini güncellemek için kendine has metotları kullanmaktadır. Fibonacci Heap veri yapısını ve metotlarını daha iyi anlayabilmek için bununla ilgili kaynaklar incelenebilir (Fredman ve Tarjan, 1987; Cormen ve ark, 2001). Belirtilen kaynaklar incelendiğinde, E_C hat kümesi içindeki hatların Fibonacci Heap için birer düğüm niteliğinde olduğu görülecektir. Bu tezde de Fibonacci Heap veri yapısı kullanılmıştır. Çizelge 3.2'de de görüldüğü gibi doğrusal arama yöntemi ile karşılaştırıldığında işlem süresinde önemli derecede azalma olmuştur.

Çizelge 3.2. Prim algoritması ile MYA çıkarılmasında, Fibonacci Heap veri yapısı ve metotlarının kullanılması ile Doğrusal Arama algoritmasının kullanılmasının zaman maliyetleri açısından karşılaştırılması

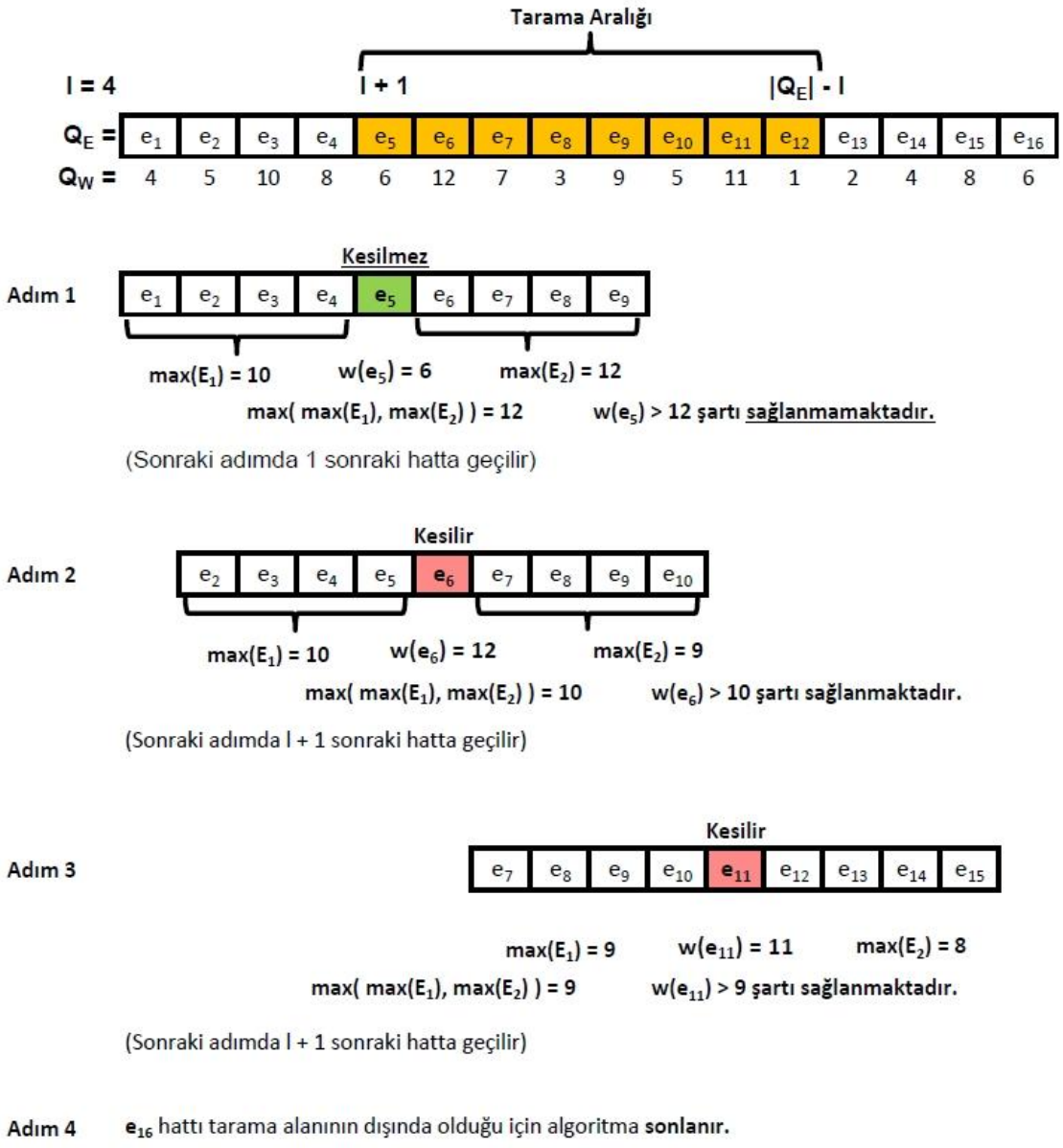
Görüntü Boyutu	Doğrusal Arama (sn)	Fibonacci Heap (sn)
61x41	0.16	0.001
121x81	0.125	0.006
241x161	1.856	0.028
481x321	36.880	0.115

3.6. Önerilen Yayılan Ağaç Tabanlı Sıralı Görüntü Bölütleme Metodu

Wang ve arkadaşları (Wang ve ark., 2014) Prim'in sıralı MYA yapısını kullanan bir kümeleme algoritması önermişlerdir. Bu algoritmaya göre, bir hat kuyruğu olan $\boldsymbol{Q_E}$ aynı zamanda bir $\boldsymbol{T}$ ağacının hatlarından oluşan $\boldsymbol{E_T}$ hat kümesi olarak düşünülebilir. $\boldsymbol{E_1}$ ve $\boldsymbol{E_2}$ alt hat kümeleri de $\boldsymbol{Q_E}$ kuyruğundaki bir indeksin her iki tarafından, kullanıcı tarafından belirlenen miktarda hat elemanlarından $\boldsymbol{Q_E}$ kuyruğundaki sıralamayı bozmadan oluşturulan alt hat kümeleridir (Wang ve ark, 2014). Önceden kullanıcı tarafından $\boldsymbol{l} > 0$ olacak şekilde belirlenen $\boldsymbol{l}$ parametresinin değeri $\boldsymbol{E_1}$ ve $\boldsymbol{E_2}$ alt hat kümelerinin uzunluklarını belirler ($|\boldsymbol{E_1}| = \boldsymbol{l}$ ve $|\boldsymbol{E_2}| = \boldsymbol{l}$) (Şekil 3.18).

Wang ve arkadaşlarının algoritmasına göre, bir görüntünün bölütlenmesi, görüntünün MYA'sı üzerindeki bir hattı keserek gerçekleşmektedir (Denklem 3.2). Yaklaşımın asıl hedefi ise kesilecek olan hattın uyumsuz hat olup olmadığının nasıl belirleneceği sorusuna cevap verebilmektir. Bu metotta bu soruya cevap verebilmek için, Q_E kuyruğundaki bir e_i hattı, her iki tarafından bulunan E_1 ve E_2 alt hat kümeleri

ile karşılaştırılmaktadır. Karşılaştırılacak olan e_i hattı $i = l+1, l+2, \ldots, |Q_E| - l$ olacak şekilde sıralı olarak Q_E kuyruğundan seçilmektedir (Şekil 3.18).



Şekil 3.18. Sıralı MYA temsilindeki hat kuyruğunun taranması örneği

Şekil 3.18'de l değeri 4 olarak belirlenmiş ve algoritma karşılaştırmaya ilk olarak $l+1$ indeksindeki hat olan e_5 hattından başlamıştır. e_5 hattı sol tarafındaki 4 hattan oluşan $E_1 = \{e_1, e_2, e_3, e_4\}$ alt hat kümesi ve sağ tarafındaki 4 hattan oluşan $E_2 = \{e_6, e_7, e_8, e_9\}$ alt hat kümesi ile karşılaştırılmaktadır. Daha sonraki adımda da şekilde görüldüğü gibi bir sonraki hat olan e_6 hattı sağındaki ve solundaki 4'er adet hattan oluşan E_1 ve E_2 alt hat kümeleri ile karşılaştırılmıştır.

Şekil 3.18'de de görüldüğü gibi E_1 ve E_2 alt hat kümeleri içindeki hatlar her adımda sıralı olarak değişmektedir. Bu tez kapsamında, yapılan bu sıralı kaydırma işlemi "tarama işlemi" olarak adlandırılmaktadır (Şekil 3.18).

MYA üzerindeki bir hattın ağırlık derecesini diğer hatlar veya kümeler ile karşılaştırmada bazı kriterler kullanılmıştır. Wang ve ark. (2014) hat kesme kriteri olarak Denklem 3.2'deki kriteri kullanmışlardır.

$$w(u,v) > \max(\max(E_1), \max(E_2)) \quad (3.2)$$

Burada $\max(E_1)$ ve $\max(E_2)$ sırası ile E_1 ve E_2 alt hat kümeleri içindeki en yüksek ağırlıklı olan hatların ağırlık değerlerini göstermektedir. Denkleme göre, iki küme ortasındaki hat ağırlığı bu iki dizgi içindeki hatlardan en yüksek olanından daha yüksek ise bu hat kesilmektedir.

Adım 1'de $\max(E_1)$ e_1, e_2, e_3 ve e_4 hatlarının arasındaki ağırlık değeri en yüksek olan hattın ağırlık değerini ve $\max(E_2)$ de e_6, e_7, e_8 ve e_9 hatlarının arasındaki ağırlık değeri en yüksek olan hattın ağırlık değerini ifade etmektedir. Örneğin; Şekil 3.18'de $e_1 = 4$, $e_2 = 5$, $e_3 = 10$ ve $e_4 = 8$ olsun. $E_1 = \{e_1, e_2, e_3, e_4\}$ olduğundan $\max(E_1) = max\{e_1, e_2, e_3, e_4\} = 10$ olur. $e_6 = 12$, $e_7 = 7$, $e_8 = 3$ ve $e_9 = 9$ olsun. $E_2 = \{e_6, e_7, e_8, e_9\}$ olduğundan $\max(E_2) = max\{e_6, e_7, e_8, e_9\} = 12$ olur. Eğer $w(u,v) = w(e_5) = 6$ ise, Denklem 3.2'ye göre $6 > \max(10,12)$ ifadesi sağlanmadığından, e_5 hattı kesilmemektedir. Hat kesilmediği takdirde tarama işlemi bir sonraki hattan devam etmektedir. Eğer şart sağlanır ve hat kesilirse, tarama işlemi $l + 1$ sonraki hattan devam eder. Tarama işlemi $|Q_E| - l$ numaralı indeksi aştığında algoritma sonlanır (Şekil 3.18).

Denklem 3.2'deki kriter "uyumsuz hat" olarak "belirgin hatları" belirlemektedir ama "geçiş hatlarını" belirleyememektedir. Bu probleme çözüm olarak Felzenszwalb ve Huttenlocher (2004) algoritmalarında hat birleştirme kriteri olarak Denklem 3.3'deki kriteri kullanmışlardır. Bu algoritma Kruskal algoritmasını kullanarak MYA oluşturma aşamasında bölütleme yapmaktadır. Elde edilmiş bir MYA yapısı üzerinde değil de MYA oluşturma aşamasında bölütleme yaptığı için kesme değil birleştirme kriteri vardır. Kruskal algoritmasında eklenecek hat birleştirme kriterini karşılamazsa, oluşturulmakta olan MYA üzerine eklenmemekte ve bu şekilde bölütleme sağlanmaktadır.

$$w(u,v) < \min\left(Int(C_1) + \tau(C_1), Int(C_2) + \tau(C_2) \right) \quad (3.3)$$

Denklem 3.3'e göre $Int(C_1)$ ve $Int(C_2)$ sırasıyla C_1 ve C_2 kümelerindeki en yüksek ağırlıklı olan hatları ifade etmektedir. $\tau(C_1)$ ve $\tau(C_2)$ ise sırasıyla C_1 ve C_2 kümelerinin büyüklüklerini hesaba katarak ayarlanan eşik değerleridir ve Denklem 3.4'deki gibi hesaplanmaktadır (Felzenszwalb ve Huttenlocher, 2004). C_1 ve C_2 kümeleri Kruskal algoritmasında eklenecek hattın iki ucundaki var olan (o aşamaya kadar eklenmiş olan) ağaçlardaki düğüm kümeleridir.

$$\tau(C) = k/|C| \quad (3.4)$$

Denklem 3.4'de k sabiti iki küme arasındaki farklılığın kümelerin iç farklılıklarından ne kadar büyük olması veya olmaması gerektiğini ayarlayan bir parametre değeridir. Genellikle k değeri 481x321 piksel bir görüntü için 500 gibi değerler almaktadır. $|C|$ ise C kümesinin eleman sayısını göstermektedir.

Yapılan tez çalışmasında, önerilen metoda göre, Felzenszwalb ve Huttenlocher'in kullandığı Denklem 3.3'deki C_1 ve C_2 kümelerin yerine Wang ve arkadaşlarının algoritmasındaki E_1 ve E_2 alt hat kümeleri dikkate alınmaktadır. Denklem 3.3 iki kümenin birleştirilip birleştirilmeyeceğini belirlerken, Denklem 3.2 ve tez kapsamında önerilen metotta kullanılan Denklem 3.5, bir hattın kesilip kesilmemesini belirlemektedir. Bu sayede ağaç yapısı alt ağaçlara ayrılmakta ve benzerlik açısından daha homojen ağaçlar (kümeler) elde edilmektedir.

$$w(u,v) > \min\left(\max(E_1), max(E_2) \right) + c \quad (3.5)$$

Tez kapsamında önerilen metot için girilen iki parametreden birisi olan c sabiti Q_E üzerindeki bir hattın uyumsuz hat olabilmesi için, iki tarafındaki E_1 ve E_2 alt hat kümelerinin en az birisinin iç farklılığından ne kadar büyük olması gerektiğini ayarlamaktadır. Bu sayede geçiş hatları da uyumsuz hat olarak değerlendirilebilmekte ve karşılaştırılacak hat sayısı (l değeri) değiştirilmeden farklı görüntü türleri için adaptasyon sağlanabilmektedir. Bu denklem ile hem belirgin hatlar ile birlikte geçiş hatlarının da tespit edilmesi sağlandığından hem de karşılaştırılan hat değerleri arasında belirli bir mesafe olması gerektiği şartı koyulduğundan genel olarak daha iyi sonuçlar vermektedir. Bu denklem, kesme koşulunda geçiş hatlarının kesilmesine izin

verdiğinden daha esnek bir kural oluştururken, karşılaştırıldığı hat ile arasına belirli bir ağırlık değeri mesafesi koyduğu için de bir birine yakın ağırlık değerlerine sahip olan hatlar için ise daha katı bir kural oluşturmuş olmaktadır.

Diğer parametre olan l değeri ise Şekil 3.18'de görüldüğü gibi ağırlıkları, e_i hattının ağırlık değeri ile karşılaştırılacak olan hatların sayısını ve bölütleme işlemi sonrasında oluşacak alt bölütlerin büyüklüğünün alt limitini ayarlamakta yani gürültü bölütleri için bir kriter koymaktadır. Bunu yapmak için de kesilecek hattın her iki ucundaki alt ağaçlardan en az birisi l değerinden daha az sayıda hatta sahip ise o hat kesilmemektedir (Şekil 3.16 (c)).

Herhangi bir v_i düğümü ve e_j hattı için, $T(v_i)$ ve $T(e_j)$ ağaçları sırası ile v_i düğümünün ve e_j hattının ait oldukları ağaç yapılarını göstermektedirler. $T(v_i)$ ağacı T_t olarak etiketlendiği zaman bu $T(v_i)$ ağacı içerisindeki düğümlerin t sayısı ile numaralandırıldığı anlamına gelmektedir. Bir ağaç içerisindeki bir düğüm o ağaç içerisindeki bütün düğümler ile aynı, diğer ağaçlardaki düğümler ile farklı bir sayı ile numaralandırılır. Şekil 3.15'de ilk başta bütün MYA yapısı 1 ile numaralandırılmıştır. Ağaçtan bir hat kesildikten sonra, kesilen hattın bir ucunda yeni bir alt ağaç oluşmuş ve düğümleri öncekinden farklı bir numara ile numaralandırılmıştır. Bundan sonraki her hat kesilmesinde oluşan alt ağaçlara ait olan düğümler diğer ağaçlardaki düğümlerden farklı olacak bir numara ile numaralandırılmaktadır (Şekil 3.15).

Girdi : $T_{PST}(v_s, V_T, Q_E, Q_W)$ ve c ve l parametreleri
Çıktı : $T = T_1, T_2, , T_t$ alt ağaç kümesi

```
1  t ← 1
2  T(v_s)'yi T_t olarak etiketle
3  for i ← l + 1 to |Q_E| − l do
4      w_max1 ← max(Q_{w,i−l,i−1})
5      w_max2 ← max(Q_{w,i+1,i+l})
6      (u, v) ← e_i
7      if w_i^Q > min(w_max1, w_max2) + c then
8          if |T(u)| > l ve |T(v)| > l then
9              kes (u, v)
10             t ← t + 1
11             T(v)'yi T_t olarak etiketle
12             i ← i + l
```

Şekil 3.19. Önerilen algoritmanın sözde kodu

Her ağaçta, ağacın düğüm sayısı gibi ağacın geneli ile ilgili bazı bilgileri olan bir kök düğüm bulunmaktadır ve ağaç içerisindeki bütün düğümlerden o kök düğüme bir bağlantı bulunmaktadır. t sayısı ile etiketlenmiş bir ağaçtaki düğüm sayısı $|T_t|$ ile ifade edilmektedir. Ağaç yapısından bir hat kesildiği zaman yeni bir alt ağaç oluşmakta ve oluşan yeni ağaçtaki bütün düğümler diğer ağaçlardaki düğümlerden farklı bir numara ile etiketlenmektedirler. Algoritmanın sözde kodu Şekil 3.19'da gösterilmektedir.

Şekil 3.19'a göre PST-MYA yapısının daha önceden çıkarılmış olması ve c ve l parametrelerinin belirlenmiş olması gerekmektedir. İşlem adımları aşağıda açıklanmaktadır.

Adım 1 – t değişken değeri 1 olarak atanmaktadır.

Adım 2 – v_s başlangıç düğümünün ait olduğu ağaç (bütün MYA) T_t yani T_1 olarak etiketlenmektedir. Bu da ağaç içerisindeki bütün düğümlerin 1 olarak numaralandırılması ile gerçekleşmektedir.

Adım 3 – Bu adım ilk çalıştırıldığında i değeri $l + 1$ olarak atanmaktadır. Daha sonraki her bu adıma gelişte i değeri 1 artırılmakta ve eğer i değişkeninin değeri $|Q_E| - l$ değerinden yani MYA'daki hat sayısının l eksiğinden daha fazla olduysa işlem sonlandırılmaktadır.

Adım 4 – Q_W ağırlık kuyruğunun i değerinin l eksiği ile 1 eksiği arasındaki indekslerdeki ($|i - l|$ - $|i - 1|$) en yüksek ağırlıklı hat w_{max1} değişkenine atanmaktadır.

Adım 5 – Q_W kuyruğunun i değerinin 1 fazlası ile l fazlası arasındaki indekslerdeki ($|i + 1|$ - $|i + l|$) en yüksek ağırlıklı hat w_{max2} değişkenine atanmaktadır.

Adım 6 – Q_E hat kuyruğunun $i.$ indeksindeki hat olan e_i hattının iki ucundaki düğümler u ve v düğüm değişkenlerine atanmaktadır. Burada u düğümü ilk düğümü yani MYA üzerinde, başlangıç düğümünden ilerlerken e_i hattına ilk ulaşan düğümü, v düğümü de sonraki düğümü temsil etmektedir.

Adım 7 – w_{max1} ve w_{max2} değerlerinden küçük olanına c parametre değeri eklenmekte ve eğer Q_W ağırlık kuyruğunun $i.$ indeksindeki değer olan w_i^Q değeri bu değerden daha fazla ise sonraki adıma geçilmekte, değilse Adım 3'e gidilmektedir.

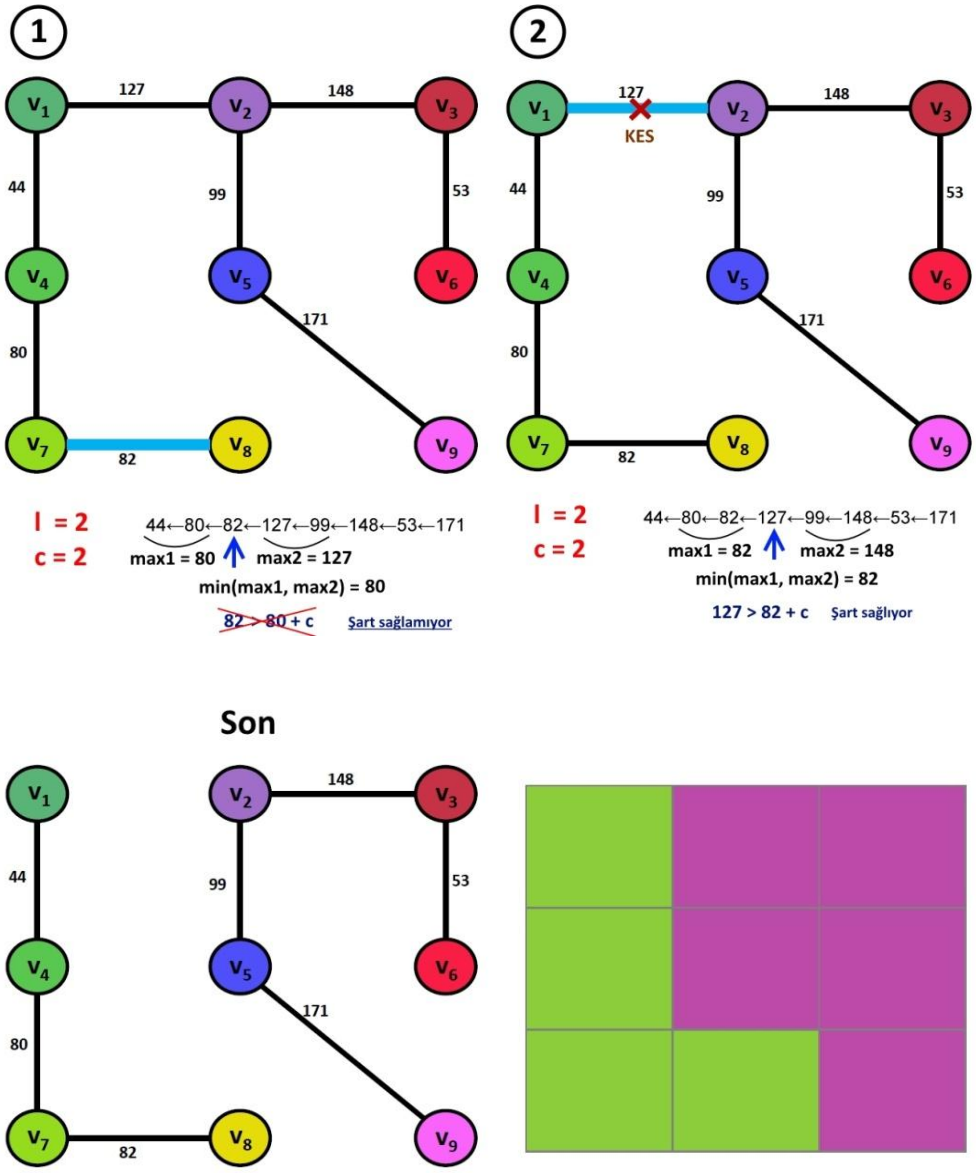
Adım 8 – u ve v düğümlerinin ait olduğu ağaçların her ikisinin de düğüm sayısı l parametre değerinden daha büyük ise sonraki adıma geçilmekte, değilse Adım 3'e gidilmektedir.

Adım 9 – (u, v) yani e_i hattı kesilmektedir (MYA'dan kaldırılmaktadır).

Adım 10 – t değişkeninin değeri 1 artırılmaktadır.

Adım 11 – v düğümünün ait olduğu ağaçtaki bütün düğümler t olarak numaralandırılmaktadır.

Adım 12 – i değeri l kadar artırılmakta ve Adım 3'e gidilmektedir. Algoritmaya göre i değeri aslında $l + 1$ kadar artırılmaktadır. Ama Adım 3'te i değeri 1 kadar daha artırılacağı için bu adımda l kadar artırılmıştır.



Şekil 3.20. Hat kesme kriterini karşılama ve görüntü bölütleme örneği

Şekil 3.20'de önerilen algoritmaya göre hat kesme kriterini karşılamayan ve karşılayan iki hat için örnek görülmektedir. Örnek olarak Şekil 3.17'de oluşturulan PST-MYA yapısı ele alınmıştır. Parametre olarak $l = 2$ ve $c = 2$ değerleri kullanılmıştır. Karşılaştırma ilk olarak $l + 1$ numaralı indeksin karşılığı olan (v_7, v_8) hattından

başlamıştır. Hattın ağırlık değeri olan 82 değeri Denklem 3.5'te sunulan kriteri sağlamadığı için ikinci adımda bir sonraki hat olan (v_1, v_2) hattına geçilmiştir. Bu hattın ağırlık değeri olan 127 değeri şekilde görüldüğü gibi kriteri karşılamıştır. Hattın her iki ucunda oluşacak her iki ağaçtaki düğüm sayısı da l değeri olan 2'den büyük olduğu için algoritmaya göre bu hat kesilmiştir. Algoritmaya göre, hat kesildikten sonra hat sırası 1 değil $l + 1$ kadar atlanmaktadır. Ancak algoritma $|Q_E| - l$ 'inci sıradaki hatta kadar ilerleyebildiği için $indeks(v_1, v_2) + l + 1$ ile $|Q_E| - l$ arasında bir hat bulunmadığı için ilk hat kesme işleminden sonra algoritma sonlanmıştır. Kullanılan graf yapısı Şekil 3.14'deki boyutları 3x3 olan örnek görüntünün grafı olduğu için bu görüntünün bölütlemeden sonraki hali de Şekil 3.20'de görülmektedir.

4. ARAŞTIRMA SONUÇLARI VE TARTIŞMA

Tez kapsamında hem Wang ve arkadaşlarının Prim'ın MYA temsilini kullanan kümeleme algoritması, hem de bu algoritmayı temel alarak tez kapmasında önerilen algoritma The Berkeley Segmentation Dataset üzerindeki BSDS300 veri seti kullanılarak test edilmiştir (Martin ve ark, 2001; BSDS300, 2016). Veri seti 200 adet deneme görüntüsü ve 100 adet de test görüntüsü olmak üzere toplam 300 adet gerçek görüntüden oluşmaktadır. Bu görüntüler 481x321 ya da 321x481 piksel büyüklüklerindedirler. Yapılan çalışmada C++ programlama dili ile OpenCV destekli nesne yönelimli programlama tekniklerinden yararlanılmıştır. (OpenCV, 2016). Renk değerleri için RGB renk uzayı ve pikseller arasındaki renk farklılığı için Öklit mesafesi kullanılmıştır (Cheng ve ark, 2001). Program Intel(R) Core(TM) i5 1.70 GHz işlemci ve 6 GB RAM özelliklerine sahip bir bilgisayar üzerinde çalıştırılmıştır.

Gerçek görüntüler düzensiz yapılara sahiptirler ve bu da bölütleme işlemini yanlış yönlendirebilmekte ve gerçek sınırlar ihlal edilebilmektedir. Bu problem için bir ön işlem olarak 5x5 çekirdek büyüklüğü ile Gauss filtresi kullanılmıştır (Nixon ve Aguado, 2012). Gauss filtresi, görüntü üzerinde gözle görülür bir değişiklik yapmamaktadır; ama görüntü üzerindeki istenmeyen dokuları ve dalgalanmaların ortadan kaldırılmasında yardımcı olmaktadır (Felzenszwalb ve Huttenlocher, 2004). Şekil 4.1 ve Şekil 4.2'deki örneklerde Gauss filtresi $\sigma = 0{,}5$ değeri ile kullanılmıştır. İlk olarak Wang ve arkadaşlarının sıralı kümeleme algoritması (Wang ve ark, 2014) görüntü bölütleme işlemi için kullanılan data set üzerinde uygulanmış ve görsel sonuçları Şekil 4.1'de gösterilmiştir. Şekil 4.1'de görüldüğü gibi görüntü bölütleme işlemini sadece l parametresi ile kontrol etmek oldukça zor olmaktadır. Çünkü veri setinde bulunan yüz binlerce pikselden bir resim için l parametresi çok geniş değer aralığında değişebilmektedir.

Şekil 4.1'de veri setinden seçilen bazı resimlerin bölütleme sonuçları görsel olarak gösterilmiştir. Her sütunun üzerinde o sütundaki görüntüler için ortak olarak kullanılan l parametresinin değeri ve o sütundaki görüntülerin bölütlemesi için kullanılan ortalama işlem süresi (OS) değeri gösterilmektedir. Ön işlem olarak Gauss ($\sigma = 0.5$) filtresi ve işlem süresi için saniye (sn) ölçü birimi kullanılmıştır.



Şekil 4.1. Wang ve ark. (2014) sıralı kümeleme metodu kullanılarak BSDS300 veri setindeki görüntülerden bazılarının bölütleme sonuçları (σ=0.5)

Önerilen metodun başarısının ölçülebilmesi için ilk olarak BSDS300 veri seti üzerindeki 200 adet deneme görüntüsü üzerinde, veri setindeki görüntü türleri ve boyutları için uygun parametre değerleri ayarlanmıştır. Veri seti üzerinde Benchmark sonuçları alabilmek için Boundary-Benchmarking programı kullanılmıştır (Arbelaez ve ark, 2011; BSDS300, 2016). Bu program optimal dataset scale (ODS) ve optimal image scale (OIS) için veri seti üzerinde bölütlenmiş görüntüdeki sınırlar üzerinde F-score ortalaması olarak özet bir skor tablosu çıkarmaktadır.

F-score (F_1 score veya F-measure) görüntü bölütlemesi doğruluğunun ölçülmesi için kullanılan ölçümlerden birisidir (Sokolova ve ark, 2006; Arbelaez ve ark, 2011). Klasik F-measure ölçümü precision ve recall değerlerinin harmonik ortalamasıdır (Denklem 4.1). Precision elde edilen değerlerin yüzde kaçının gerçek değer olduğu, recall ise gerçek değerlerin yüzde kaçının elde edildiğidir. Bu yüzdelik oran genelde 0

ile 1 değerleri arasında gösterilmektedir. Bir F-score değeri ne kadar yüksek ise bölütleme başarısı da o kadar yüksektir.

$$F - score = 2.\frac{Precision.Recall}{Precision + Recall} \tag{4.1}$$

ODS yani "optimal veri seti skoru", farklı değerleri için test edilen bir parametrenin değerleri içerisinde bütün bir veri seti içerisinde en iyi ortalama F-score sonucunu veren parametre değerinin ortalama F-score sonucudur. OIS yani "optimal görüntü skoru" ise yine farklı değerleri için test edilen bir parametrenin her görüntü için ayrı ayrı en iyi F-score sonucunu veren parametre değeri için kullanılmaktadır. Her bir görüntü için elde edilen en iyi F-score sonuçları kullanılarak, veri seti içerisindeki bütün görüntülerin en iyi F-score sonuçlarının ortalaması alındığında OIS elde edilmektedir (Arbelaez ve ark, 2011). ODS ve OIS değerleri de aslında bir F-score değeri olduğu için, F-score değerinde olduğu gibi 0 ile 1 değerleri arasındadır ve bu değer ne kadar büyük ise bölütleme de o kadar başarılı olmaktadır.

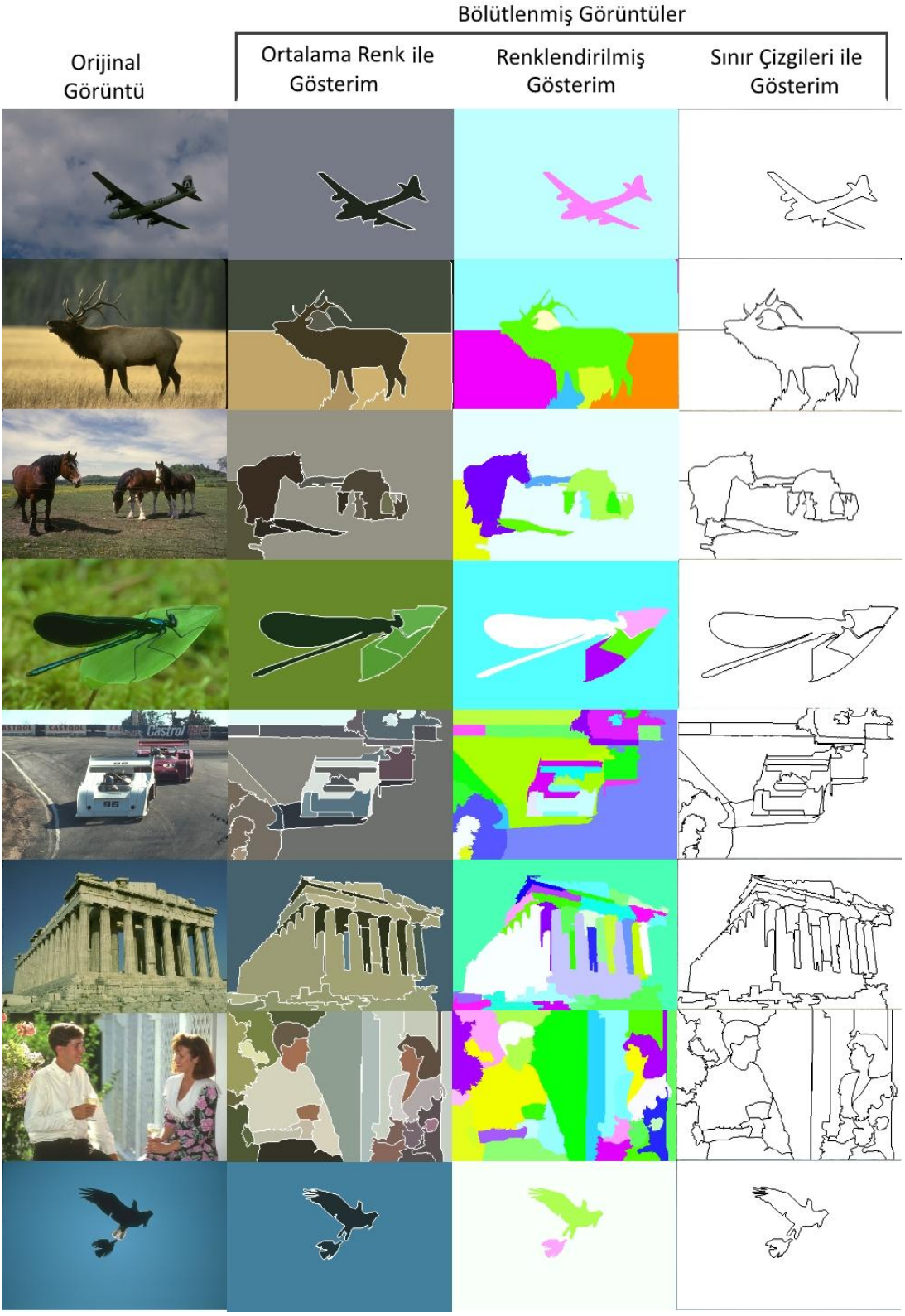
RGB renk uzayında her üç renk kanalı için 0-255 renk aralığı göz önünde bulundurularak c değeri belirlenmiştir. Bu yüzden c değeri en fazla kullanılan değer aralığındaki en büyük Öklit mesafesi olan $\sqrt{255^2 + 255^2 + 255^2} \cong 441{,}67$ değeri kadar olabilmektedir.

Tez kapsamında kullanılacak c değerleri için ilk olarak ortalama hat ağırlığı değişimi kullanılmıştır. Daha önce elde edilen sıralı MYA temsilindeki Q_W hat ağırlıkları kuyruğundaki hat ağırlıklarının, kuyruktaki bir önce gelen hattın ağırlığına göre değişimlerinin toplamının, bütün bir kuyruğa göre ortalaması alınmıştır (Denklem 4.2).

$$Ortalama\ hat\ ağırlık\ değişimi = c = \frac{\sum_{j=1}^{|Q_W|-1} |w_{j+1} - w_j|}{|Q_W| - 1} \tag{4.2}$$

İlk kullanılacak c değerlerini hesaplamak için veri setindeki 200 adet deneme görüntüsü kullanılmıştır ve Denklem 4.2 kullanılarak bütün görüntülerin ortalama hat ağırlığı değişimlerinin ortalaması alınmıştır. Bu değer, PST-MYA yapısı elde edildikten sonra elde edildiği için kullanılan önişlemden etkilenmektedir. Çizelge 4.1'de 200

deneme görüntüsü için tez kapsamında kullanılan değişik önişlemler için ortalama hat ağırlığı değişim sonuçları görülmektedir.



Şekil 4.2. Önerilen metot kullanılarak BSDS300 veri setindeki görüntülerden bazılarının bölütleme sonuçları ($\sigma = 0.5, l = 500, c = 1.96$ ve $ortalama\ işlem\ süresi = 0.40\ sn$)

Çizelge 4.1. BSDS300 veri setindeki 200 deneme görüntüsünün farklı önişlemlerden sonraki ortalama hat ağırlığı değişim değerlerinin ortalaması

Önişlem	**Ortalama Hat Ağırlığı Değişimi = c**
Yok	2.58
Gauss ($\sigma = 0.5$)	1.96
Gauss ($\sigma = 1.2$)	0.96

Şekil 4.1'deki orijinal görüntüleri bölütleme işlemi, tez kapsamında önerilen algoritma kullanılarak da yapılmıştır. Önerilen algoritmada, aynı görüntüler için Gauss ($\sigma = 0.5$) önişlemi, $l = 500$ ve $c = 1.96$ değerleri kullanılmıştır.

Veri setindeki bütün test görüntüleri üzerinde test edilecek c değeri için Çizelge 4.1'de elde edilen değerler kullanılmıştır (Çizelge 4.2).

Çizelge 4.2. Tez kapsamında önerilen algoritmanın The Berkeley Segmentation Dataset (BSDS300) üzerindeki 100 adet test görüntüsü için Boundary Benchmark sonuçları

Önişlem	**l**	**c**	**En iyi l değeri**	**ODS**	**OIS**
Yok	100-2000*	2.58	200	0.53	0.55
Gauss ($\sigma = 0.5$)	100-2000*	1.96	200	0.54	0.56
Gauss ($\sigma = 1.2$)	100-2000*	0.96	200	0.57	0.59

* 100'er aralıklı olarak

Çizelge 4.3'de iki metodun bazı parametre değerleri için OIS ve ODS sonuçları görülmektedir. Ek.1'de ise, BSDS300 üzerindeki 100 adet test görüntüsü için alınan tüm Boundary Benchmark sonuçları verilmiştir.

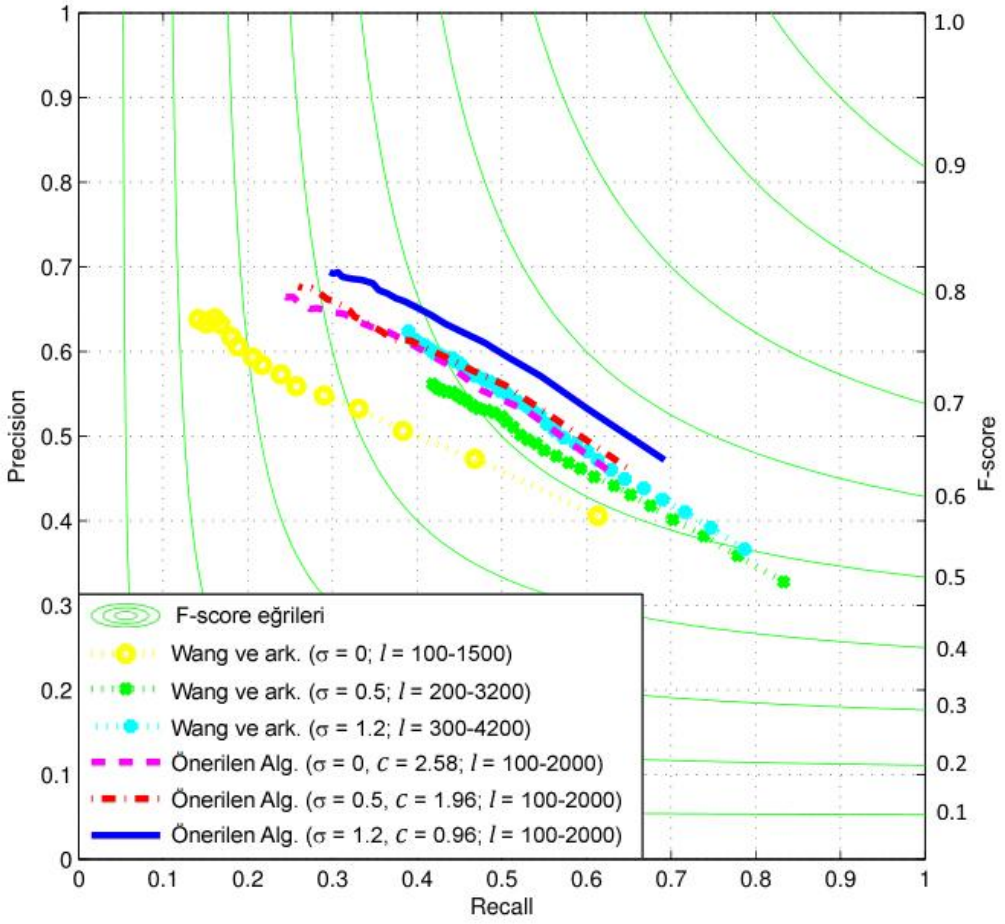
Çizelge 4.3. Önerilen algoritmanın ve Wang ve arkadaşlarının algoritmasının The Berkeley Segmentation Dataset (BSDS300) üzerindeki 100 adet test görüntüsü için Boundary Benchmark sonuçları

Algoritma	**Önişlem**	**l**	**c**	**En iyi l değeri**	**ODS**	**OIS**	**En iyi l için ortalama işlem süresi**
Wang ve ark.	Yok	100-1500*	-	100	0.49	0.50	0.30 sn
Önerilen	Yok	100-2000*	2.58	200	0.53	0.55	0.36 sn
Wang ve ark.	Gauss ($\sigma = 0.5$)	200 - 3100*	-	800	0.52	0.56	0.58 sn
Önerilen	Gauss ($\sigma = 0.5$)	100-2000*	1.96	200	0.54	0.56	0.41 sn
Wang ve ark.	Gauss ($\sigma = 1.2$)	300 - 4300*	-	1100	0.54	0.58	0.68 sn
Önerilen	Gauss ($\sigma = 1.2$)	100-2000*	0.96	200	0.57	0.59	0.41 sn

* 100'er aralıklı olarak

Benchmark sonucu alabilmek için bir parametrenin değişik değerleri ve eğer varsa diğer parametrelerin sabit bir değeri kullanılmaktadır. Tez kapsamında önerilen metotta iki parametre kullanılmaktadır (l ve c). Bu yüzden her birinde bir parametre

sabit tutularak iki farklı değer aralığı için test edilmiştir. Şekil 4.3'de bu iki metot ve tabloda girilen değerleri için elde edilen precision ve recall eğrilerinin bulunduğu grafik görülmektedir.



Şekil 4.3. BSDS300 veri seti üzerindeki Boundary Benchmark sonuçlarının precision ve recall eğrileri

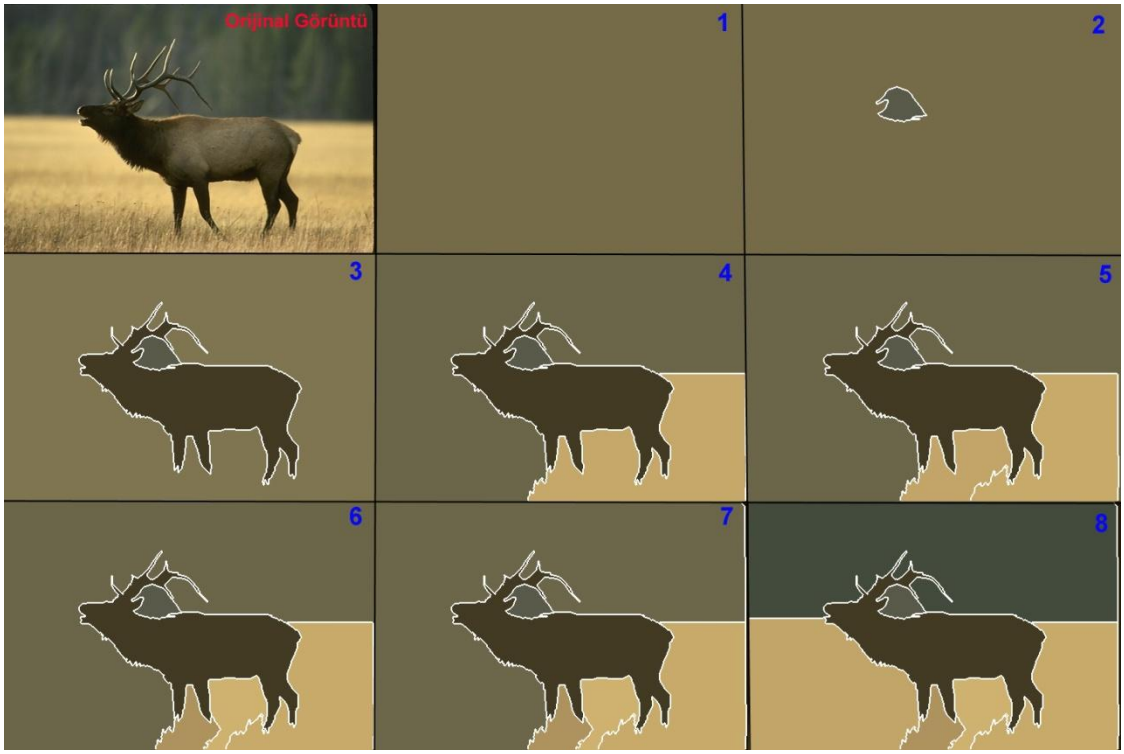
Yapılan tez çalışmasında ayrıca, önerilen metodun sonuçları, görüntü bölütleme alanında şu ana kadarki en çok bilinen metotlardan bazılarının aynı veri seti üzerindeki sonuçları ile karşılaştırılmıştır (Çizelge 4.4).

Çizelge 4.4. Önerilen metot ile en popüler görüntü bölütleme metotların BSDS300 veri seti üzerindeki 100 adet test görüntüsü için Boundary Benchmark sonuçları

Metot	**ODS**	**OIS**
Mean Shift (Comaniciu ve Meer, 2002)	0.63	0.66
NCuts (Cour ve ark, 2005)	0.62	0.66
Efficient Graph-Based (Felzenszwalb ve Huttenlocher, 2004)	0.58	0.62
Önerilen Metot ($\sigma = 1.2, c = 0.96$)	0.57	0.59
SWA (Sharon ve ark, 2006)	0.56	0.59
Wang ve ark. ($\sigma = 1.2$)	0.54	0.58

Çizelge 4.4'e bakıldığında, önerilen metodun doğruluk başarısı olarak en popüler olan görüntü bölütleme metotlarının arasına girebildiği görülmektedir. Önerilen metot öz-vektör ve doku özellikleri çıkarma ve görüntü bölütleme işleminden sonra gürültü olarak tanımlanabilecek küçük bölütleri birleştirme gibi çok fazla işlem süresi gerektiren işlemleri yapmamaktadır.

Şekil 4.4'de bölütlemenin sıralı olarak yapıldığını gösteren bölütleme adımları örnek bir görüntü üzerinde her hat kesimi için aşama aşama görülmektedir.



Şekil 4.4. Önerilen algoritma ile sıralı görüntü bölütlemede her hat kesimdeki bölütlemenin durumu

5. SONUÇLAR VE ÖNERİLER

5.1 Sonuçlar

Görüntü segmentasyonu orijinal görüntüden daha anlamlı bilgi sağladığı için birçok yüksek seviye görüntü işleme uygulaması için çok önemli bir aşamadır. Aynı zamanda gerçek zamanlı görüntü işleme uygulamaları için de kullanılmaktadır. Bu yüzden, görüntü bölütleme işlemi için kullanılacak metotta hem bölütleme doğruluğu hem de işlem süresi açısından performans çok önemlidir. Graf teorisi araçları, görüntü işleme uygulamaları için hem veri temsili açısından hem de işlem kolaylığı açısından sıklıkla kullanılmaktadır. MYA algoritmaları graf tabanlı metotlar için işlem kolaylığını bir adım daha öteye taşımakta ve görüntü içinde daha hızlı operasyon yapılmasını sağlamaktadırlar.

Bu tezde önce graf teorisindeki bazı araçlardan ve MYA yapısı ve algoritmalarından bahsedilmiş, daha sonra da bazı MYA tabanlı görüntü bölütleme ve veri kümeleme metotları üzerinde durulmuştur. Görüntü bölütleme için Prim'ın sıralı MYA temsili yapısını kullanan, alanında başarılı bir kümeleme algoritması açıklanmış (Wang, 2014) ve tez kapsamında önerilen algoritma için bu kümeleme algoritması temel alınmıştır. Tez kapsamında, hem temel alınan MYA tabanlı kümeleme algoritması hem de önerilen algoritma, görüntü bölütleme işlemi için görüntüler üzerine ayrı ayrı uygulanmış ve sonuçlar elde edilmiştir.

Tez kapsamında önerilen algoritma için temel olarak faydalanılan Wang ve arkadaşlarının kümeleme algoritmasının tek parametresi olan l sabitinin değeri arttıkça, elde edilen bölüt sayısı azalmakta ve karşılaştırılacak hat sayısı arttığı için işlem süresi artmaktadır. Bununla birlikte farklı görüntü türleri için istenilen bölütleme sonucuna ulaşmakta yetersiz kalmaktadır. Ayrıca l parametresi çok geniş aralıklarda değerler aldığı için bölütleme işlemini kontrol etmek oldukça zor olmaktadır.

Tez kapsamında önerilen algoritmada, l parametresinin yanında kullanılan c parametresi bölütleme işlemi üzerindeki kontrolü artırmaktadır. Diğer bir ifade ile farklı türde görüntüler üzerinde istenilen şekilde bölütleme yapabilmek için adaptasyonu artırmaktadır. c parametre değeri arttıkça, l parametresinde olduğu gibi elde edilen bölüt sayısı azalmaktadır. Önerilen algoritmadaki kesme kriteri graftaki belirgin hatların yanı sıra geçiş hatlarını da dikkate almaktadır. Bu özelliğinden dolayı bu kriter kesme şartını biraz daha esnetmektedir. Ama kullanılan c eşik değerinden dolayı ağırlık farkı

karşılaştırıldığı diğer hatlara çok yakın olan hatlarda, o hat en yüksek ağırlıklı olsa bile hat kesilmeyebilmektedir. Bu yönüyle, böyle yapılarda ise daha sert bir kesme kriteri olmaktadır. Buradan yola çıkıldığında, önerilen algoritmanın Wang ve arkadaşlarının algoritmasına göre daha iyi sonuçlar vermesinin nedeni olarak, görüntüdeki hafif renk geçişli bölgeleri ayırmayıp, ani değişimli bölgeleri ayırması görülmektedir.

Gauss filtresi için farklı σ değerleri uygulandığında ise Çizelge 5.1'de görüldüğü gibi her iki algoritma da benzer oranlarda etkilenmektedir.

Çizelge 5.1. Farklı σ değerleri için 100 adet test görüntüsünün ortalama F-score sonuçları

Algoritma	$\boldsymbol{\sigma = 0}$	$\boldsymbol{\sigma = 0.5}$	$\boldsymbol{\sigma = 1}$	$\boldsymbol{\sigma = 1.2}$	$\boldsymbol{\sigma = 1.5}$
Wang ve ark. algoritması ($l = 300$)	0.44	0.49	0.50	0.50	0.50
Önerilen algoritma ($c = 0.6, l = 300$)	0.54	0.55	0.58	0.57	0.58

Önerilen algoritma düşük işlem süresi ile bölütleme işleminde performansı arttırmıştır. Ayrıca, en popüler görüntü bölütleme algoritmaları ile de yapılan karşılaştırmalar sonucunda, başarılı sonuçlar elde edilmiştir.

5.2 Öneriler

Önerilen algoritma, bir görüntünün herhangi bir yerinden başlayarak sıralı bir şekilde, belirli büyüklükteki yerel bölgeler üzerinden, görüntünün bölütlenecek kısımları için çıkarım yapan bir yaklaşım kullanmıştır. Bu yaklaşım daha önceden, Wang'ın algoritmasında olduğu gibi, verileri konumlarına göre kümelemek amacıyla bazı algoritmalar tarafından kullanılmıştır. Bu tez kapsamında, bu yaklaşım görüntü bölütleme amacıyla kullanılmıştır. Bir görüntüyü bölütlemeden önce MYA yapısının çıkarılmış olması, algoritma öncesinde görüntü hakkında genel bilgiler de vermektedir. Bu sayede gürültülü bölütlerin oluşmasını engelleme gibi bazı avantajlar elde edilebilmektedir. Sıralı MYA temsilini tarama esnasında daha iyi bir karşılaştırma yöntemi geliştirilebilir. Görüntü üzerinde daha farklı ön işlemler veya son işlemler gerçekleştirilerek bölütleme işleminin başarısı artırılabilir. Graf tabanlı yaklaşımlar, diğer görüntü işleme metotları ile hibrit bir şekilde kullanılarak başarı arttırılabilir ya da işlem süresi daha da azaltılabilir.

KAYNAKLAR

Agarwal P. K., Sharathkumar R., Yu H. Approximate Euclidean shortest paths amid convex obstacles, Proceedings of the twentieth Annual ACM-SIAM Symposium on Discrete Algorithms, 283-92, New York, New York.

Al Haj M., Bagdanov A. D., Gonzalez J., Roca X. F., 2009, Robust and Efficient Multipose Face Detection Using Skin Color Segmentation, Pattern Recognition and Image Analysis, Proceedings, 5524, 152-9.

Arbelaez P., Maire M., Fowlkes C., Malik J., 2011, Contour Detection and Hierarchical Image Segmentation, Ieee T Pattern Anal, 33, 5, 898-916.

Bader D. A., Cong G. J., 2006, Fast shared-memory algorithms for computing the minimum spanning forest of sparse graphs, J Parallel Distr Com, 66, 11, 1366-78.

Bayramoglu N. Y., Bazlamacci C. N. F., 2005, Recursive shortest spanning tree algorithms for image segmentation, Eurocon 2005: The International Conference on Computer as a Tool, Vol 1 and 2 , Proceedings, 979-82.

Biggs N., 1993, Algebraic Graph Theory, Cambridge Mathematical Library, Cambridge University Press, p.

Black P. E., 2005, greedy algorithm, In: Dictionary of Algorithms and Data Structures. Eds: U.S. National Institute of Standards and Technology (NIST), p.

Bondy J. A., Murty U. S. R., 1976, Graph Theory with Applications, New York, USA, Elsevier Science Publishing Co.., Inc., p.

Boruvka O., 1926, O jistem problému minimalnim (About a certain minimal problem), Prace mor. prirodoved. spol. v Brne III (in Czech, German summary), 3, 37–58.

Boykov Y., Funka-Lea G., 2006, Graph cuts and efficient N-D image segmentation, Int J Comput Vision, 70, 2, 109-31.

Boykov Y., Veksler O., Zabih R., 2001, Fast approximate energy minimization via graph cuts, Ieee T Pattern Anal, 23, 11, 1222-39.

Boykov Y. Y., Jolly M. P., 2001, Interactive graph cuts for optimal boundary & region segmentation of objects in N-D images, Eighth Ieee International Conference on Computer Vision, Vol I, Proceedings, 105-12.

The Berkeley Segmentation Dataset and Benchmark, 2016, Erişim tarihi 2016, Güncelleme tarihi 2007, Erişim adresi, https://www.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/.

Cardenes R., de Luis-Garcia R., Bach-Cuadra M., 2009, A multidimensional segmentation evaluation for medical image data, Comput Meth Prog Bio, 96, 2, 108-24.

Cheng H. D., Jiang X. H., Sun Y., Wang J. L., 2001, Color image segmentation: advances and prospects, Pattern Recogn, 34, 12, 2259-81.

Cheriton D., Tarjan R., 1976, Finding Minimum Spanning Trees, Siam J Comput, 5, 4, 724-42.

Chow S. N., Lu J., Zhou H. M., 2013, Finding the shortest path by evolving junctions on obstacle boundaries (E-JOB): An initial value ODE's approach, Appl Comput Harmon A, 35, 1, 165-76.

Chrobak M., Kenyon C., Young N., 2006, The reverse greedy algorithm for the metric k-median problem, Inform Process Lett, 97, 2, 68-72.

Clark R., Miller W. F., (1966), Computer-based data analysis systems at Argonne Methods, Computational Physics, New York, **5:** 47-98.

Comaniciu D., Meer P., 2002, Mean shift: A robust approach toward feature space analysis, Ieee T Pattern Anal, 24, 5, 603-19.

Cormen T. H., Leiserson C. E., Rivest R. L., Stein C., 2001, Introduction to Algorithms, Second Edition, Cambridge, Massachusetts London, England, McGraw-Hill Book Company, p.

Cour T., Benezit F., Shi J., 2005, Spectral segmentation with multiscale graph decomposition, 2005 Ieee Computer Society Conference on Computer Vision and Pattern Recognition, Vol 2, Proceedings, 1124-31.

Cox I. J., Rao S. B., Yu Z. Ratio regions: a technique for image segmentation, Pattern Recognition, 1996., Proceedings of the 13th International Conference on, 557-64 vol.2, 25-29 Aug 1996.

Çamalan S., 2013, Analysis of filtering and quantization preprocessing steps in image segmentation, Master Thesis, Atılım Üniversitesi, Ankara.

Devi R. D. H., Devi M. I., 2014, Survey of Clustering and Outlier Detection Techniques in Data Mining: A Research Perspective, Appl Mech Mater, 573, 511-8.

Diestel R., 2010, Graph Theory, Heidelberg Graduate Texts in Mathematics, Springer-Verlag, p.

Dijkstra E. W., 1959, A note on two problems in connexion with graphs, Numer. Math., 1, 1, 269-71.

Dudgeon D. E., Mersereau R. M., 1984, Multidimensional digital signal processing, Prentice-Hall, p.

Durusoy G., 2014, Graph Cuts And Region Growing Algorithms in Medical Image Segmentation, Master thesis, New York University, New York.

Falcao A. X., Udupa J. K., Miyazawa F. K., 2000, An ultra-fast user-steered image segmentation paradigm: Live wire on the fly, Ieee T Med Imaging, 19, 1, 55-62.

Faliu Y., Inkyu M. Image segmentation: A survey of graph-cut methods, Systems and Informatics (ICSAI), 2012 International Conference on, 1936-41, 19-20 May 2012.

Felzenszwalb P. F., Huttenlocher D. P., 2004, Efficient graph-based image segmentation, Int J Comput Vision, 59, 2, 167-81.

Franti P., Virmajoki O., Hautamaki V., 2006, Fast agglomerative clustering using a k-nearest neighbor graph, Ieee T Pattern Anal, 28, 11, 1875-81.

Fredman M. L., Tarjan R. E., 1987, Fibonacci Heaps and Their Uses in Improved Network Optimization Algorithms, J Acm, 34, 3, 596-615.

Glisson C. L., Altamar H. O., Herrell S. D., Clark P., Galloway R. L., 2011, Comparison and assessment of semi-automatic image segmentation in computed tomography scans for image-guided kidney surgery, Med Phys, 38, 11, 6265-74.

Gonzalez R. C., Woods R. E., 2001, Digital Imaging Processing, New Jersey, USA, Prentice Hall, p.

Gower J. C., Ross G. J. S., 1969, Minimum Spanning Trees and Single Linkage Cluster Analysis, Journal of the Royal Statistical Society. Series C (Applied Statistics), 18, 1, 54-64.

Grady L., 2005, Multilabel random walker image segmentation using prior models, 2005 Ieee Computer Society Conference on Computer Vision and Pattern Recognition, Vol 1, Proceedings, 763-70.

Gross J. L., Yellen J., 2005, Graph Theory and Its Applications, CRC Press, Taylor & Francis Group, p.

Grygorash O., Zhou Y., Jorgensen Z., 2006, Minimum spanning tree based clustering algorithms, Proc Int C Tools Art, 73-81.

Haxhimusa Y., Kropatsch W., 2004, Segmentation graph hierarchies, Lect Notes Comput Sc, 3138, 343-51.

He X., Shi P., 2007, A new segmentation approach for iris recognition based on hand-held capture device, Pattern Recogn, 40, 4, 1326-33.

Jain A. K., Murty M. N., Flynn P. J., 1999, Data clustering: A review, Acm Comput Surv, 31, 3, 264-323.

Janakiraman T. N., Mouli P. V. S. S. R. C. Dominant Pixel Identification from Gray-level Image using various Matrix Rotation Operations – An Application to Edge Detection, Proceedings of IEEE sponsored Int. Conference ICSCN-08, 510-3, Chennai.

Jarník V., 1930, O jistém problému minimálním (On a certain problem of minimization), Práce moravské přírodovědecké společnosti, 6, 4, 57-63.

Johnson S. C., 1967, Hierarchical Clustering Schemes, Psychometrika, 32, 241–54.

Karger D. R., Klein P. N., Tarjan R. E., 1995, A Randomized Linear-Time Algorithm to Find Minimum Spanning-Trees, J Assoc Comput Mach, 42, 2, 321-8.

Karoui I., Fablet R., Boucher J. M., Augustin J. M., 2010, Variational Region-Based Segmentation Using Multiple Texture Statistics, Ieee T Image Process, 19, 12, 3146-56.

Karypis G., Han E. H., Kumar V., 1999, Chameleon: Hierarchical clustering using dynamic modeling, Computer, 32, 8, 68-+.

Kindermann R., Snell J. L., 1980, Markov Random Fields and Their Applications, Providence, Rhode Island, p.

Kleinberg J., Tardos É., 2005, Algorithm Design, Pearson; 1 edition (March 26, 2005), p.

Kruskal J. B., 1956, On the Shortest Spanning Subtree of a Graph and the Traveling Salesman Problem, Proceedings of the American Mathematical Society, 7, 1, 48-50.

Kwok S. H., Constantinides A. G., 1997, A fast recursive shortest spanning tree for image segmentation and edge detection, Ieee T Image Process, 6, 2, 328-32.

Lezoray O., 2003, Unsupervised 2D multiband histogram clustering and region merging for color image segmentation, Proceedings of the 3rd Ieee International Symposium on Signal Processing and Information Technology, 267-70.

Li S. Z., 2009, Markov Random Field Modeling in Image Analysis, Springer, p.

Lopresti D., Zhou J., 2000, Locating and Recognizing Text in WWW Images, Information Retrieval, 2, 2-3, 177-206.

Martin D., Fowlkes C., Tal D., Malik J., 2001, A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics, Eighth Ieee International Conference on Computer Vision, Vol Ii, Proceedings, 416-23.

Mohanram S., Sudhakar T. D. Power System Restoration using Reverse Delete Algorithm Implemented in FPGA, Chennai and Dr.MGR University Second International Conference on Sustainable Energy and Intelligent System, India.

Morris O. J., Lee M. D. J., Constantinides A. G., 1986, Graph-Theory for Image-Analysis - an Approach Based on the Shortest Spanning Tree, Iee Proc-F, 133, 2, 146-52.

Naz S., Majeed H., Irshad H. Image segmentation using fuzzy clustering: A survey, Emerging Technologies (ICET), 2010 6th International Conference on, 181-6, 18-19 Oct. 2010.

Nesetril J., Milkova E., Nesetrilova H., 2001, Otakar Boruvka on minimum spanning tree problem - Translation of both the 1926 papers, comments, history, Discrete Math, 233, 1-3, 3-36.

Ning J. F., Zhang L., Zhang D., Wu C. K., 2010, Interactive image segmentation by maximal similarity based region merging, Pattern Recogn, 43, 2, 445-56.

Nixon M. S., Aguado A. S., 2012, Gaussian averaging operator, In: Feature Extraction & Image Processing for Computer Vision. Eds, Third edition, London, UK: Elsevier Ltd., p. 86-8.

Njeh I., Ben Ayed I., Ben Hamida A., 2012, A Distribution-Matching Approach to Mri Brain Tumor Segmentation, 2012 9th Ieee International Symposium on Biomedical Imaging (Isbi), 1707-10.

Nunez J., Llacer J., 2003, Astronomical image segmentation by self-organizing neural networks and wavelets, Neural Networks, 16, 3-4, 411-7.

Olman V., Xu D., Xu Y., 2004, Solving data clustering problem as a string search problem, Statistical Data Mining and Knowledge Discovery, 415-32.

OpenCV Library, 2016, Erişim tarihi 2016, Güncelleme tarihi 2016, Erişim adresi, http://opencv.org/.

Özsoyeller D., 2008, Telsiz duyarga ağlarında eşuyumlu çalışma için dağıtık kapsayan ağaç ve kümeleme algoritmaları, Yüksek Lisans Tezi, Ege University, İzmir.

Paivinen N., 2005, Clustering with a minimum spanning tree of scale-free-like structure, Pattern Recogn Lett, 26, 7, 921-30.

Pantofaru C., 2008, Studies in using image segmentation to improve object recognition, Phd Thesis, Carnegie Mellon University, Pittsburgh, PA.

Pavan M., Pelillo M., 2003, A new graph-theoretic approach to clustering and segmentation, 2003 Ieee Computer Society Conference on Computer Vision and Pattern Recognition, Vol 1, Proceedings, 145-52.

Peng B., Zhang L., Zhang D., 2013, A survey of graph theoretical approaches to image segmentation, Pattern Recogn, 46, 3, 1020-38.

Petrov G., Iliev P., Tzvetkov P., 2008, Comparison of global histogram methods for 2D and 3D entropy based image segmentation, Artif Int Ser Wseas, 57-62.

Pettie S., Ramachandran V., 2002, A randomized time-work optimal parallel algorithm for finding a minimum spanning forest, Siam J Comput, 31, 6, 1879-95.

Pham D. L., Xu C. Y., Prince J. L., 2000, Current methods in medical image segmentation, Annu Rev Biomed Eng, 2, 315-+.

Prim R. C., 1957, Shortest connection networks and some generalizations, Bell System Technical Journal, The, 36, 6, 1389-401.

Rambabu C., Chakrabarti I., Mahanta A., 2004, Flooding-based watershed algorithm and its prototype hardware architecture, Iee P-Vis Image Sign, 151, 3, 224-34.

Raut S., Raghuvanshi M., Dharaskar R., Raut A., 2009, Image Segmentation - A State-Of-Art Survey for Prediction, International Conference on Advanced Computer Control : Icacc 2009 - Proceedings, 420-4.

Robles-Kelly A., Hancock E. R., 2004, A probabilistic spectral framework for grouping and segmentation, Pattern Recogn, 37, 7, 1387-405.

Rugna J. D., Chareyron G., Konik H. About segmentation step in content-based image retrieval systems, In World Congress on Engineering and Computer Science, 550-4, San Francisco, USA.

Sağıroğlu M. Ş., Uçan O. N., 2002, İlintili Görüntülerin Birleşik Markov Rastgele Alan Modeli ve Gerçek - Zamanlı Uygulanabilir En - İyileme Yaklaşımı, Doğuş Üniversitesi Dergisi, 3, 1, 141-9.

Sharon E., Galun M., Sharon D., Basri R., Brandt A., 2006, Hierarchy and adaptivity in segmenting visual scenes, Nature, 442, 7104, 810-3.

Shi J. B., Malik J., 2000, Normalized cuts and image segmentation, Ieee T Pattern Anal, 22, 8, 888-905.

Sokolova M., Japkowicz N., Szpakowicz S., 2006, Beyond accuracy, F-Score and ROC: A family of discriminant measures for performance evaluation, AI 2006: Advances in Artificial Intelligence, Proceedings, 4304, 1015-21.

Srivastava S., Ghosh R. K., 2003, Distributed algorithms for finding and maintaining a k-tree core in a dynamic network, Inform Process Lett, 88, 4, 187-94.

Sundaramoorthi G., Yezzi A., Mennucci A. C., 2008, Coarse-to-fine segmentation and tracking using Sobolev active contours, Ieee T Pattern Anal, 30, 5, 851-64.

Suresh R., 2012, An efficient image segmentation algorithm using bidirectional Mahalanobis distance, Master thesis, Clemson University, South Carolina, USA.

Tao W. B., Jin H., Zhang Y. M., 2007, Image segmentation based on mean shift and normalized cuts, Ieee T Syst Man Cy B, 37, 5, 1382-9.

Theoharatos C., Economou G., Fotopoulos S., 2005, Color edge detection using the minimal spanning tree, Pattern Recogn, 38, 4, 603-6.

Wang G. W., Zhang C. X., Zhuang J., 2014, Clustering with Prim's sequential representation of minimum spanning tree, Appl Math Comput, 247, 521-34.

Wang J., 2007, Graph based image segmentation, Doctor of Philosophy, The Hong Kong University of Science and Technology, Hong Kong.

Wang S., Siskind J. M., 2001, Image segmentation with minimum mean cut, Eighth Ieee International Conference on Computer Vision, Vol I, Proceedings, 517-24.

Wang S., Siskind J. M., 2003, Image segmentation with ratio cut, Ieee T Pattern Anal, 25, 6, 675-90.

Wang W., Kravchenko A. N., Smucker A. J. M., Rivers M. L., 2011, Comparison of image segmentation methods in simulated 2D and 3D microtomographic images of soil aggregates, Geoderma, 162, 3-4, 231-41.

Wassenberg J., Middelmann W., Sanders P., 2009, An Efficient Parallel Algorithm for Graph-Based Image Segmentation, Computer Analysis of Images and Patterns, Proceedings, 5702, 1003-10.

Wertheimer M., 1938, Laws of Organization in Perceptual Forms (partial translation), In: A Sourcebook of Gestalt Psychology. Eds: Ellis WB: Harcourt, Brace, p. 71-88.

Willett R. M., Nowak R. D., 2007, Minimax optimal level-set estimation, Ieee T Image Process, 16, 12, 2965-79.

Wu Z., Leahy R., 1990, Tissue Classification in Mr Images Using Hierarchical Segmentation, 1990 Ieee Nuclear Science Symposium Conference Record, Vols 1 and 2, 1410-4.

Wu Z., Leahy R., 1993, An Optimal Graph-Theoretic Approach to Data Clustering - Theory and Its Application to Image Segmentation, Ieee T Pattern Anal, 15, 11, 1101-13.

Xu R., Wunsch D., 2005, Survey of clustering algorithms, Ieee T Neural Networ, 16, 3, 645-78.

Xu Y., Olman V., Xu D., 2002, Clustering gene expression data using a graph-theoretic approach: an application of minimum spanning trees, Bioinformatics, 18, 4, 536-45.

Xu Y., Uberbacher E. C., 1997, 2D image segmentation using minimum spanning trees, Image Vision Comput, 15, 1, 47-57.

Yi F., Moon I. Image segmentation: A survey of graph-cut methods, International Conference on Systems and Informatics (ICSAI), 1936 – 41, Yantai.

Yoshida T., Shoda R., Motoda H., 2006, Graph clustering based on structural similarity of fragments, Federation over the Web, 3847, 97-114.

Yu H. C., Wang D. J., Tang Z. S., 2001, Level set methods and image segmentation, International Workshop on Medical Imaging and Augmented Reality, Proceedings, 204-8.

Yu Z. W., Wong H. S., Wang H. Q., 2007, Graph-based consensus clustering for class discovery from gene expression data, Bioinformatics, 23, 21, 2888-96.

Zahn C. T., 1971, Graph-Theoretical Methods for Detecting and Describing Gestalt Clusters, IEEE Trans. Comput., 20, 1, 68-86.

Zhong C. M., Malinen M., Miao D. Q., Franti P., 2015, A fast minimum spanning tree algorithm based on K-means, Inform Sciences, 295, 1-17.

Zhong C. M., Miao D. Q., Franti P., 2011, Minimum spanning tree based split-and-merge: A hierarchical clustering method, Inform Sciences, 181, 16, 3397-410.

Zhong C. M., Miao D. Q., Wang R. Z., 2010, A graph-theoretical clustering method based on two rounds of minimum spanning trees, Pattern Recogn, 43, 3, 752-66.

Zhou Y., Cheng H., Yu J. X., 2009, Graph clustering based on structural/attribute similarities, Proc. VLDB Endow., 2, 718-29.

EK 1

Wang ve arkadaşlarının algoritmasının BSDS300 tüm "test" veri seti için (100 görüntü) değer bazında ortalama sonuçları ($\sigma = 0$; $l = 100 - 1500$)

l	Precision	Recall	F-score
100	0.61354	0.405504	0.488287
200	0.468058	0.473215	0.470622
300	0.382998	0.506183	0.436058
400	0.330561	0.532508	0.407908
500	0.289971	0.547872	0.379228
600	0.257214	0.558942	0.352305
700	0.238884	0.573057	0.337202
800	0.216884	0.583996	0.316301
900	0.205845	0.59286	0.305588
1000	0.188228	0.605857	0.287222
1100	0.180179	0.617151	0.278925
1200	0.167531	0.63218	0.26487
1300	0.160707	0.639277	0.256846
1400	0.150848	0.633174	0.243648
1500	0.141591	0.638068	0.231755

Wang ve arkadaşlarının algoritmasının BSDS300 tüm "test" veri seti için (100 görüntü) değer bazında ortalama sonuçları ($\sigma = 0.5$; $l = 200 - 3100$)

l	Precision	Recall	F-score
200	0.83289	0.327757	0.470403
300	0.77856	0.358655	0.491085
400	0.738552	0.382061	0.503603
500	0.702285	0.401258	0.510714
600	0.675793	0.417714	0.516299
700	0.65195	0.430537	0.518599
800	0.632773	0.441374	0.520022
900	0.609092	0.451704	0.518722
1000	0.592704	0.461659	0.519038
1100	0.578752	0.468399	0.517761
1200	0.564401	0.476345	0.516648
1300	0.54997	0.483302	0.514485
1400	0.540688	0.49112	0.514713
1500	0.530415	0.495698	0.512469
1600	0.522055	0.501386	0.511512
1700	0.513757	0.510645	0.512196
1800	0.505913	0.51696	0.511377
1900	0.50026	0.523058	0.511405
2000	0.490955	0.527681	0.508656
2100	0.482623	0.530626	0.505487
2200	0.472493	0.532427	0.500673
2300	0.466056	0.536027	0.498599
2400	0.459067	0.541608	0.496933
2500	0.453177	0.544334	0.49459
2600	0.447323	0.548141	0.492627
2700	0.441637	0.551987	0.490684
2800	0.434147	0.552239	0.486124
2900	0.428641	0.554618	0.483559
3000	0.423791	0.556544	0.481179
3100	0.418295	0.561463	0.479419

Wang ve arkadaşlarının algoritmasının BSDS300 tüm "test" veri seti için (100 görüntü) değer bazında ortalama sonuçları ($\sigma = 1.2$; $l = 300 - 4200$)

l	Precision	Recall	F-score
300	0.786886	0.366199	0.499802
400	0.74721	0.391602	0.513885
500	0.7165	0.410233	0.521742
600	0.690148	0.424916	0.525988
700	0.667922	0.438257	0.529248
800	0.64535	0.449339	0.529796
900	0.628941	0.460072	0.531414
1000	0.61341	0.471894	0.533426
1100	0.601044	0.482009	0.534985
1200	0.587404	0.491024	0.534907
1300	0.572901	0.498197	0.532944
1400	0.562304	0.506743	0.53308
1500	0.552933	0.514489	0.533019
1600	0.544737	0.52428	0.534313
1700	0.53555	0.530912	0.533221
1800	0.528383	0.536219	0.532272
1900	0.518964	0.540214	0.529376
2000	0.511228	0.544004	0.527107
2100	0.505522	0.550233	0.52693
2200	0.497894	0.553388	0.524176
2300	0.492891	0.56124	0.52485
2400	0.485592	0.563892	0.521821
2500	0.478584	0.567086	0.51909
2600	0.469895	0.571286	0.515654
2700	0.462733	0.572735	0.511891
2800	0.456416	0.57959	0.510681
2900	0.450457	0.585931	0.509339
3000	0.444941	0.588332	0.506687
3100	0.442171	0.591952	0.506214
3200	0.434973	0.591998	0.501481
3300	0.429798	0.593868	0.498685
3400	0.426656	0.596051	0.497325
3500	0.420038	0.598004	0.493466
3600	0.416027	0.601922	0.492
3700	0.409654	0.606978	0.489166
3800	0.405489	0.607896	0.486479
3900	0.40184	0.6093	0.484287
4000	0.398329	0.613076	0.482904
4100	0.393526	0.615114	0.479979
4200	0.389746	0.623676	0.479712

Tez kapsamında önerilen algoritmanın BSDS300 tüm "test" veri seti için (100 görüntü) değer bazında ortalama sonuçları (*σ = 0; l = 100 - 2000; c = 2.58*)

l	Precision	Recall	F-score
100	0.624117	0.460679	0.530086
200	0.53054	0.529277	0.529908
300	0.474995	0.554641	0.511738
400	0.439946	0.582211	0.501178
500	0.408917	0.599765	0.486287
600	0.385622	0.613019	0.473431
700	0.366433	0.622783	0.461392
800	0.346852	0.628722	0.447067
900	0.328489	0.636385	0.433311
1000	0.313187	0.644543	0.421544
1100	0.302052	0.645922	0.411619
1200	0.293149	0.648696	0.403813
1300	0.281495	0.651234	0.393081
1400	0.273315	0.650324	0.384876
1500	0.269566	0.654269	0.381818
1600	0.258979	0.658586	0.371766
1700	0.253544	0.664505	0.367042
1800	0.244765	0.664187	0.357708
1900	0.239528	0.669189	0.352782
2000	0.234091	0.67144	0.347152

Tez kapsamında önerilen algoritmanın BSDS300 tüm "test" veri seti için (100 görüntü) değer bazında ortalama sonuçları ($\sigma = 0.5$; $l = 100 - 2000$; $c = 1.96$)

l	Precision	Recall	F-score
100	0.64718	0.463371	0.540064
200	0.556421	0.52479	0.540143
300	0.504374	0.560009	0.530738
400	0.464695	0.577777	0.515103
500	0.434952	0.592918	0.501797
600	0.406966	0.605509	0.486771
700	0.392433	0.61286	0.47848
800	0.371334	0.613996	0.462784
900	0.358134	0.623742	0.455013
1000	0.340939	0.633104	0.443204
1100	0.325955	0.641281	0.432219
1200	0.320848	0.647763	0.429137
1300	0.313433	0.653424	0.42365
1400	0.30289	0.658497	0.414926
1500	0.293485	0.661866	0.406652
1600	0.287456	0.666758	0.401721
1700	0.281385	0.671258	0.396543
1800	0.274927	0.675114	0.390734
1900	0.265636	0.674757	0.381202
2000	0.259302	0.676751	0.374942

Tez kapsamında önerilen algoritmanın BSDS300 tüm "test" veri seti için (100 görüntü) değer bazında ortalama sonuçları ($\sigma = 1.2$; $l = 100 - 2000$; $c = 0.96$)

l	Precision	Recall	F-score
100	0.692241	0.471919	0.561232
200	0.599279	0.533662	0.564571
300	0.546017	0.570868	0.558166
400	0.507635	0.593005	0.547009
500	0.479234	0.610535	0.536974
600	0.455142	0.622115	0.525689
700	0.432808	0.632951	0.514087
800	0.417229	0.64268	0.505977
900	0.402919	0.650276	0.49755
1000	0.384608	0.659432	0.485848
1100	0.375363	0.662692	0.479262
1200	0.364648	0.668822	0.471972
1300	0.354401	0.672524	0.464188
1400	0.347569	0.68072	0.460176
1500	0.335033	0.684495	0.449872
1600	0.322067	0.686059	0.438352
1700	0.311693	0.6881	0.429041
1800	0.306291	0.693746	0.424961
1900	0.301136	0.692006	0.419654
2000	0.296589	0.693893	0.415558

Wang ve arkadaşlarının algoritmasının BSDS300 görüntü bazında en iyi sonuçları (*σ = 0; l = 100 - 1500*)

Görüntü dosyası adı	En iyi sonuç verdiği *l* değeri	Precision	Recall	F-score
3096	300	0.557447	0.569692	0.563503
8023	100	0.527521	0.573607	0.5496
12084	500	0.472121	0.512315	0.491398
14037	100	0.433511	0.439382	0.436426
16077	200	0.533307	0.299383	0.383487
19021	100	0.654654	0.368154	0.471278
21077	100	0.547191	0.325887	0.408491
24077	100	0.494824	0.205228	0.290127
33039	100	0.41529	0.175256	0.246491
37073	100	0.433667	0.33145	0.375731
38082	500	0.612113	0.705345	0.65543
38092	100	0.399931	0.29436	0.339119
41033	200	0.416274	0.244134	0.307769
41069	200	0.403585	0.505405	0.448792
42012	1500	0.371817	0.648855	0.472738
42049	100	0.45139	0.529618	0.487385
43074	300	0.487338	0.619456	0.545512
45096	100	0.642288	0.591928	0.61608
54082	400	0.454842	0.899274	0.604125
55073	200	0.517638	0.562456	0.539117
58060	1000	0.463892	0.545505	0.501399
62096	200	0.610452	0.435599	0.508412
65033	100	0.659036	0.319276	0.430158
66053	100	0.669548	0.599767	0.632739
69015	100	0.656118	0.457115	0.538829
69020	100	0.354819	0.213648	0.266704
69040	100	0.626357	0.367553	0.46326
76053	100	0.402173	0.315214	0.353423
78004	800	0.238088	0.616817	0.343563
85048	500	0.468695	0.644715	0.542792
86000	100	0.443325	0.431433	0.437298
86016	100	0.579912	0.360721	0.444778
86068	100	0.649451	0.636655	0.642989
87046	100	0.519472	0.580828	0.548439
89072	400	0.566056	0.757202	0.647823
97033	300	0.516036	0.606541	0.55764
101085	100	0.621146	0.15185	0.24404
101087	1.200	0.442678	0.762971	0.56028
102061	100	0.491351	0.505128	0.498145
103070	100	0.492345	0.392114	0.436551
105025	200	0.714217	0.669529	0.691151

106024	100	0.685645	0.629834	0.656556
108005	100	0.587116	0.423128	0.491812
108070	100	0.630906	0.472248	0.540168
108082	100	0.643496	0.585892	0.613344
109053	100	0.688367	0.722628	0.705082
119082	300	0.66077	0.452256	0.536981
123074	100	0.587737	0.37522	0.458028
126007	100	0.654622	0.579986	0.615048
130026	100	0.618509	0.57858	0.597879
134035	100	0.679297	0.487344	0.567529
143090	100	0.594307	0.623639	0.60862
145086	100	0.475385	0.416271	0.443868
147091	100	0.275268	0.168127	0.208753
148026	100	0.688535	0.501557	0.580358
148089	100	0.659213	0.585082	0.619939
156065	300	0.523475	0.286194	0.370066
157055	100	0.714316	0.456372	0.556927
159008	100	0.505367	0.495658	0.500466
160068	200	0.658661	0.618902	0.638163
163085	100	0.560273	0.592758	0.576058
167062	100	0.649617	0.332912	0.440221
167083	100	0.562242	0.383851	0.456228
170057	300	0.398022	0.193444	0.260353
175032	200	0.646411	0.390494	0.486872
175043	100	0.490566	0.383907	0.430732
182053	200	0.634708	0.617391	0.62593
189080	100	0.648488	0.333304	0.440304
196073	200	0.615329	0.505724	0.555168
197017	100	0.586072	0.378774	0.460154
208001	100	0.684643	0.739511	0.563503
210088	700	0.500478	0.470621	0.48509
216081	100	0.529724	0.401167	0.456568
219090	100	0.647682	0.434459	0.520064
220075	100	0.621361	0.435654	0.512194
223061	300	0.367512	0.712241	0.484847
227092	200	0.577042	0.267213	0.365276
229036	100	0.537506	0.617247	0.574623
236037	200	0.770507	0.711101	0.739613
241004	100	0.415319	0.297847	0.346908
241048	100	0.504259	0.538767	0.520942
253027	100	0.398798	0.601529	0.47962
253055	100	0.599931	0.404209	0.482995
260058	100	0.337327	0.346776	0.341986
271035	100	0.652703	0.485958	0.557121
285079	100	0.594814	0.420353	0.492592
291000	100	0.540441	0.425728	0.476274

295087	100	0.510314	0.645667	0.570066
296007	100	0.721155	0.282802	0.40628
296059	100	0.468883	0.283913	0.353674
299086	100	0.541444	0.475151	0.506136
300091	100	0.725101	0.445304	0.551758
302008	100	0.579259	0.180243	0.274936
304034	100	0.654017	0.455184	0.536779
304074	200	0.596916	0.482246	0.533489
306005	100	0.510127	0.377233	0.433729
351093	300	0.466584	0.363136	0.408411
361010	100	0.588019	0.398754	0.475236
376043	100	0.60495	0.377786	0.465113
385039	100	0.692128	0.451245	0.546312

Wang ve arkadaşlarının algoritmasının BSDS300 görüntü bazında en iyi sonuçları (*σ = 0.5; l = 200 - 3100*)

Görüntü dosyası adı	En iyi sonuç verdiği *l* değeri	Precision	Recall	F-score
3096	800	0.712742	0.577929	0.638294
8023	2400	0.531111	0.72054	0.611491
12084	2200	0.670439	0.409549	0.508483
14037	800	0.612628	0.497781	0.549266
16077	1500	0.570379	0.417043	0.481805
19021	2400	0.667132	0.496646	0.569401
21077	900	0.528019	0.377044	0.439939
24077	200	0.626866	0.218292	0.323821
33039	400	0.554274	0.17978	0.271498
37073	500	0.709187	0.345452	0.464595
38082	800	0.774476	0.6578	0.711386
38092	1700	0.364779	0.462109	0.407716
41033	500	0.774714	0.24764	0.375311
41069	1400	0.633356	0.553807	0.590916
42012	1400	0.598086	0.322626	0.41915
42049	300	0.747133	0.579448	0.652692
43074	1300	0.801818	0.494763	0.611932
45096	3100	0.663973	0.561896	0.608684
54082	1900	0.751576	0.642793	0.692941
55073	3100	0.549051	0.478648	0.511438
58060	1700	0.620753	0.538574	0.57675
62096	600	0.632906	0.493058	0.554297
65033	2000	0.580879	0.563494	0.572055
66053	1000	0.704314	0.803993	0.75086
69015	500	0.717598	0.502761	0.591269
69020	1100	0.560632	0.245846	0.341805
69040	500	0.748169	0.357362	0.48369
76053	700	0.578483	0.393631	0.468482
78004	3000	0.669549	0.275079	0.38995
85048	1600	0.614199	0.485073	0.542052
86000	300	0.704964	0.408865	0.517557
86016	200	0.738373	0.385621	0.506643
86068	200	0.793921	0.63445	0.705283
87046	300	0.778041	0.434684	0.557755
89072	3100	0.654613	0.650882	0.652742
97033	1700	0.614189	0.591251	0.602502
101085	2900	0.336378	0.193114	0.245365
101087	2600	0.530419	0.473032	0.500085
102061	400	0.636887	0.561955	0.597079
103070	400	0.86944	0.385478	0.534138
105025	2000	0.574456	0.61862	0.595721

106024	300	0.759981	0.616823	0.68096
108005	3000	0.500968	0.51797	0.509328
108070	1700	0.574752	0.650505	0.610287
108082	3100	0.508389	0.823129	0.62856
109053	700	0.724177	0.550355	0.625413
119082	1200	0.638939	0.4893	0.554196
123074	200	0.719635	0.364461	0.483867
126007	600	0.741969	0.590146	0.657406
130026	3100	0.727667	0.913336	0.809998
134035	1100	0.656862	0.659099	0.657979
143090	300	0.702441	0.464497	0.55921
145086	2400	0.460311	0.373198	0.412202
147091	2100	0.586118	0.570047	0.577971
148026	400	0.816657	0.522717	0.637433
148089	400	0.734943	0.58358	0.650573
156065	1200	0.477611	0.329273	0.389807
157055	2100	0.539472	0.530652	0.535025
159008	1000	0.654328	0.506907	0.57126
160068	1400	0.712565	0.563171	0.62912
163085	2100	0.557769	0.720508	0.628779
167062	1900	0.563785	0.285214	0.378797
167083	700	0.622418	0.435217	0.51225
170057	1800	0.516268	0.277824	0.361246
175032	3000	0.550496	0.541713	0.546069
175043	900	0.629969	0.488514	0.550296
182053	3100	0.688193	0.368935	0.480355
189080	400	0.748286	0.396533	0.51837
196073	1900	0.674621	0.597015	0.63345
197017	2700	0.469118	0.601878	0.52727
208001	1000	0.698537	0.779556	0.736826
210088	800	0.66739	0.448669	0.536598
216081	400	0.656385	0.432633	0.521523
219090	1800	0.652783	0.738879	0.693168
220075	900	0.636561	0.556316	0.59374
223061	2700	0.564861	0.495672	0.52801
227092	2600	0.582968	0.414474	0.484489
229036	500	0.6806	0.546469	0.606203
236037	3100	0.770474	0.750788	0.760503
241004	1900	0.435386	0.385877	0.409139
241048	900	0.699959	0.361552	0.476814
253027	300	0.683868	0.488159	0.569673
253055	300	0.702395	0.440902	0.541744
260058	200	0.651712	0.407162	0.501197
271035	2200	0.549887	0.608504	0.577712
285079	800	0.575169	0.542354	0.558279
291000	800	0.727639	0.582846	0.647244

295087	400	0.719665	0.638934	0.676901
296007	1300	0.629815	0.527213	0.573965
296059	200	0.747016	0.315284	0.44342
299086	500	0.681348	0.536034	0.600018
300091	2500	0.589576	0.732231	0.653205
302008	3000	0.361917	0.265799	0.306499
304034	700	0.77929	0.563361	0.653962
304074	1100	0.61269	0.53085	0.568842
306005	200	0.728827	0.351767	0.474512
351093	1300	0.666118	0.366716	0.473021
361010	500	0.654468	0.450955	0.533978
376043	3000	0.564304	0.47484	0.515721
385039	1600	0.718551	0.613873	0.6621

Wang ve arkadaşlarının algoritmasının BSDS300 görüntü bazında en iyi sonuçları (*σ = 1.2; l = 300 - 4200*)

Görüntü dosyası adı	En iyi sonuç verdiği *l* değeri	Precision	Recall	F-score
3096	1800	0.664021	0.762379	0.709809
8023	600	0.670545	0.612273	0.640086
12084	1500	0.624442	0.363909	0.459837
14037	800	0.668956	0.533237	0.593436
16077	1300	0.657476	0.499622	0.567782
19021	4200	0.644811	0.668732	0.656554
21077	1500	0.48164	0.518102	0.499206
24077	300	0.551156	0.223506	0.31804
33039	600	0.617525	0.27251	0.378146
37073	1600	0.472031	0.425527	0.447574
38082	1200	0.760563	0.687046	0.721938
38092	3700	0.444399	0.670628	0.534564
41033	3500	0.511132	0.432938	0.468796
41069	2600	0.623613	0.652708	0.637829
42012	3800	0.440346	0.478019	0.45841
42049	300	0.716267	0.584025	0.643422
43074	4200	0.543898	0.835847	0.658985
45096	3900	0.634476	0.608647	0.621293
54082	3100	0.578381	0.832363	0.682509
55073	4200	0.502856	0.563281	0.531356
58060	2700	0.585209	0.661817	0.62116
62096	500	0.709138	0.481876	0.573825
65033	3400	0.529347	0.668824	0.590967
66053	800	0.706839	0.802925	0.751824
69015	1100	0.604683	0.602173	0.603425
69020	1000	0.593779	0.26387	0.365372
69040	1400	0.540789	0.492403	0.515463
76053	3000	0.488041	0.585139	0.532198
78004	1700	0.724057	0.275977	0.399633
85048	3700	0.514238	0.653008	0.575374
86000	300	0.672167	0.423073	0.519294
86016	300	0.716384	0.448402	0.551565
86068	300	0.673257	0.658912	0.666007
87046	300	0.818304	0.448803	0.579678
89072	4200	0.669399	0.612653	0.63977
97033	1400	0.681216	0.648011	0.664199
101085	3800	0.4533	0.26705	0.336097
101087	2200	0.602772	0.423169	0.49725
102061	400	0.658777	0.576408	0.614846
103070	400	0.858174	0.386165	0.532647
105025	2700	0.565029	0.642398	0.601235

106024	400	0.701801	0.65145	0.675689
108005	3900	0.408168	0.560664	0.472415
108070	2300	0.592561	0.631981	0.611637
108082	400	0.735428	0.514717	0.605589
109053	4000	0.594394	0.7101	0.647115
119082	1700	0.584863	0.561185	0.57278
123074	300	0.654526	0.378458	0.479602
126007	600	0.722969	0.583721	0.645926
130026	2600	0.765551	0.884206	0.820612
134035	1100	0.662652	0.67132	0.666958
143090	400	0.59066	0.441233	0.505127
145086	700	0.662518	0.283665	0.397245
147091	1100	0.681724	0.452052	0.543625
148026	700	0.714677	0.60132	0.653116
148089	500	0.742038	0.622024	0.676751
156065	3600	0.402339	0.574227	0.473156
157055	4200	0.548137	0.650467	0.594934
159008	4000	0.602612	0.696847	0.646313
160068	2500	0.645725	0.747319	0.692817
163085	1000	0.681168	0.545154	0.605618
167062	2700	0.577544	0.364788	0.447148
167083	600	0.60789	0.457783	0.522265
170057	4200	0.5349	0.400088	0.457775
175032	3700	0.52144	0.604112	0.55974
175043	800	0.691501	0.508539	0.586072
182053	3200	0.643954	0.448541	0.528771
189080	300	0.793703	0.409395	0.540168
196073	1100	0.721492	0.471947	0.57063
197017	4200	0.463537	0.586957	0.517997
208001	800	0.71209	0.686099	0.698853
210088	700	0.720365	0.400209	0.514551
216081	300	0.688135	0.432209	0.530941
219090	2700	0.592186	0.846128	0.696739
220075	900	0.675924	0.573246	0.620365
223061	3800	0.526135	0.599232	0.560309
227092	2600	0.630636	0.46085	0.532538
229036	500	0.744061	0.556335	0.636648
236037	3900	0.662811	0.907879	0.766226
241004	3100	0.486068	0.495314	0.490647
241048	3700	0.451024	0.673273	0.540182
253027	300	0.675238	0.497325	0.572784
253055	300	0.733565	0.434364	0.54564
260058	300	0.579913	0.444458	0.50323
271035	3500	0.539009	0.688879	0.604798
285079	400	0.715331	0.492643	0.583461
291000	1900	0.649651	0.640067	0.644823

295087	300	0.792275	0.621796	0.696759
296007	2000	0.646576	0.646123	0.646349
296059	300	0.64316	0.331763	0.43773
299086	500	0.742285	0.522472	0.613277
300091	1900	0.628472	0.802661	0.704966
302008	2300	0.283613	0.258236	0.27033
304034	700	0.76811	0.555569	0.644776
304074	400	0.733338	0.432426	0.544046
306005	300	0.728652	0.403188	0.519126
351093	2800	0.48813	0.461791	0.474595
361010	800	0.634099	0.517945	0.570167
376043	2300	0.553599	0.480649	0.514551
385039	1300	0.778691	0.622364	0.691806

Tez kapsamında önerilen algoritmanın BSDS300 görüntü bazında en iyi sonuçları (*σ = 0; l = 100 - 2000; c = 2.58*)

Görüntü Dosyası Adı	En İyi Sonuç Verdiği *l* Değeri	Precision	Recall	F-score
3096	300	0.736983	0.626486	0.677257
8023	200	0.608092	0.794841	0.689037
12084	800	0.627142	0.561459	0.592486
14037	100	0.371057	0.523519	0.434296
16077	1300	0.445638	0.481384	0.462822
19021	200	0.397659	0.51697	0.449533
21077	200	0.487539	0.375744	0.424403
24077	100	0.421138	0.253372	0.316391
33039	100	0.379503	0.178452	0.242754
37073	100	0.434864	0.439354	0.437097
38082	500	0.731911	0.718624	0.725207
38092	100	0.369276	0.325556	0.34604
41033	200	0.351482	0.292757	0.319443
41069	400	0.509352	0.750091	0.606713
42012	1900	0.333539	0.395363	0.361829
42049	100	0.708641	0.652612	0.679473
43074	100	0.62596	0.68672	0.654933
45096	200	0.516486	0.860233	0.645445
54082	600	0.679677	0.954761	0.79407
55073	200	0.702671	0.710859	0.706741
58060	900	0.561919	0.593821	0.57743
62096	300	0.619164	0.47876	0.539985
65033	200	0.596673	0.439787	0.506356
66053	600	0.558747	0.784277	0.652576
69015	700	0.442221	0.629038	0.51934
69020	100	0.512596	0.242242	0.329003
69040	200	0.556295	0.47659	0.513367
76053	100	0.186224	0.217841	0.200795
78004	100	0.353249	0.783505	0.486953
85048	700	0.732397	0.517362	0.60638
86000	100	0.370967	0.527414	0.435568
86016	100	0.685452	0.358501	0.470778
86068	100	0.689962	0.632948	0.660227
87046	100	0.709319	0.602135	0.651347
89072	100	0.284957	0.656076	0.397336
97033	400	0.57889	0.619883	0.598686
101085	100	0.546912	0.345899	0.423777
101087	200	0.676617	0.638865	0.657199
102061	100	0.536746	0.661227	0.592519
103070	100	0.399769	0.471206	0.432558
105025	200	0.851707	0.663057	0.745634

106024	100	0.713793	0.717181	0.715483
108005	100	0.647579	0.43474	0.520232
108070	100	0.521262	0.498334	0.509541
108082	200	0.689478	0.612586	0.648762
109053	100	0.628451	0.887306	0.735776
119082	300	0.753751	0.490299	0.594129
123074	100	0.53854	0.376808	0.443386
126007	100	0.687064	0.628896	0.656695
130026	200	0.501806	0.891641	0.642193
134035	100	0.696669	0.542649	0.610088
143090	100	0.697478	0.753917	0.7246
145086	100	0.548126	0.865507	0.671188
147091	100	0.199051	0.648611	0.304618
148026	100	0.826283	0.559751	0.66739
148089	100	0.710536	0.637374	0.671969
156065	200	0.644027	0.288572	0.39856
157055	200	0.684115	0.569634	0.621648
159008	100	0.39908	0.666972	0.499366
160068	100	0.6547	0.743729	0.69638
163085	100	0.577537	0.791841	0.66792
167062	100	0.757262	0.473794	0.582892
167083	100	0.400785	0.400312	0.400548
170057	300	0.405132	0.269422	0.323626
175032	600	0.720456	0.686296	0.702961
175043	300	0.518339	0.682723	0.589282
182053	100	0.0820768	0.97619	0.151422
189080	100	0.804284	0.325516	0.463458
196073	500	0.639897	0.584035	0.610691
197017	100	0.634565	0.55289	0.590919
208001	100	0.754622	0.910354	0.825205
210088	200	0.553018	0.419781	0.477275
216081	100	0.503831	0.472128	0.487465
219090	200	0.686642	0.644463	0.664884
220075	200	0.660364	0.549151	0.599645
223061	100	0.320584	0.348662	0.334034
227092	500	0.594692	0.556433	0.574926
229036	100	0.395589	0.53773	0.455836
236037	100	0.864182	0.867137	0.865657
241004	700	0.299278	0.87659	0.446213
241048	100	0.0907524	0.840807	0.163823
253027	100	0.377589	0.590664	0.460681
253055	100	0.6874	0.426253	0.526207
260058	100	0.443198	0.33346	0.380576
271035	100	0.713435	0.507839	0.593331
285079	100	0.700856	0.559224	0.62208
291000	100	0.41556	0.36803	0.390354

295087	100	0.559978	0.833789	0.669988
296007	100	0.548205	0.421873	0.476813
296059	100	0.402111	0.27539	0.326899
299086	100	0.420711	0.574493	0.485721
300091	500	0.645004	0.643365	0.644184
302008	100	0.240833	0.265328	0.252488
304034	300	0.701943	0.599124	0.646471
304074	200	0.62619	0.455887	0.527637
306005	100	0.53575	0.326611	0.40582
351093	900	0.625635	0.64788	0.636563
361010	100	0.594096	0.370341	0.456263
376043	100	0.688562	0.407037	0.511629
385039	900	0.773173	0.716593	0.743809

Tez kapsamında önerilen algoritmanın BSDS300 görüntü bazında en iyi sonuçları (*σ = 0.5; l = 100 - 2000; c = 1.96*)

Görüntü Dosyası Adı	En İyi Sonuç Verdiği *l* Değeri	Precision	Recall	F-score
3096	500	0.700598	0.700517	0.700557
8023	200	0.61006	0.738699	0.668245
12084	800	0.529021	0.445409	0.483627
14037	100	0.252319	0.469444	0.328223
16077	1200	0.417844	0.510074	0.459376
19021	900	0.4051	0.544586	0.464599
21077	100	0.54461	0.333245	0.413482
24077	300	0.319608	0.306555	0.312945
33039	200	0.39484	0.236142	0.295534
37073	100	0.418935	0.400638	0.409582
38082	600	0.742758	0.710991	0.726528
38092	100	0.444706	0.324437	0.375168
41033	300	0.276387	0.316138	0.294929
41069	300	0.619445	0.65988	0.639023
42012	1200	0.363173	0.410422	0.385355
42049	100	0.691827	0.626535	0.657565
43074	200	0.598429	0.735028	0.659732
45096	200	0.604736	0.789175	0.684753
54082	1600	0.714812	0.908238	0.799999
55073	100	0.84764	0.658578	0.741243
58060	900	0.621622	0.598675	0.609932
62096	200	0.687267	0.427297	0.526963
65033	100	0.768152	0.367227	0.496903
66053	400	0.608506	0.798726	0.69076
69015	800	0.446224	0.659179	0.532188
69020	100	0.537481	0.23845	0.330345
69040	300	0.544737	0.472266	0.505919
76053	100	0.360395	0.281108	0.315851
78004	100	0.410639	0.775769	0.537018
85048	900	0.647357	0.605458	0.625707
86000	100	0.41419	0.623079	0.497601
86016	100	0.698362	0.368627	0.482545
86068	100	0.686948	0.644535	0.665066
87046	200	0.759392	0.657979	0.705058
89072	1200	0.643041	0.890213	0.746704
97033	200	0.624323	0.612351	0.618279
101085	100	0.431725	0.445929	0.438712
101087	200	0.750746	0.655509	0.699903
102061	100	0.517504	0.654969	0.578178
103070	100	0.458839	0.452698	0.455748
105025	200	0.853909	0.645089	0.734954

106024	100	0.728916	0.723891	0.726395
108005	100	0.610442	0.437221	0.509511
108070	100	0.512115	0.511198	0.511656
108082	200	0.706029	0.672814	0.689021
109053	100	0.504975	0.854045	0.63468
119082	400	0.732255	0.513917	0.603959
123074	100	0.573723	0.373928	0.452764
126007	100	0.7319	0.592683	0.654976
130026	100	0.315515	0.667153	0.428419
134035	100	0.67853	0.637599	0.657428
143090	100	0.647133	0.788636	0.710912
145086	100	0.613654	0.885673	0.724988
147091	100	0.0908695	0.402545	0.148269
148026	100	0.755655	0.5391	0.629267
148089	200	0.689465	0.64994	0.669119
156065	800	0.432832	0.4	0.415769
157055	500	0.54197	0.684695	0.60503
159008	100	0.34197	0.613544	0.439164
160068	100	0.659424	0.758973	0.705705
163085	100	0.599626	0.823633	0.694001
167062	300	0.711908	0.611111	0.65767
167083	100	0.407119	0.40165	0.404366
170057	1.000	0.537767	0.269042	0.358652
175032	300	0.799656	0.58075	0.672846
175043	300	0.559161	0.741581	0.637579
182053	300	0.61522	0.994577	0.7602
189080	200	0.735181	0.362653	0.485713
196073	500	0.742212	0.61365	0.671836
197017	100	0.719583	0.589796	0.648257
208001	100	0.779233	0.93109	0.84842
210088	700	0.523404	0.516104	0.519729
216081	100	0.492162	0.44792	0.469
219090	1200	0.538763	0.869048	0.665162
220075	400	0.622303	0.618704	0.620498
223061	100	0.310193	0.560196	0.39929
227092	900	0.575883	0.547705	0.56144
229036	100	0.480277	0.545281	0.510719
236037	100	0.875444	0.857854	0.86656
241004	100	0.301915	0.364115	0.33011
241048	100	0.299128	0.817851	0.438042
253027	100	0.401581	0.700787	0.510579
253055	100	0.669019	0.452566	0.539906
260058	100	0.464016	0.328841	0.384905
271035	100	0.735384	0.463261	0.568433
285079	100	0.677303	0.573061	0.620836
291000	100	0.481998	0.420178	0.44897

295087	100	0.607773	0.771894	0.680072
296007	200	0.574847	0.595076	0.584787
296059	100	0.367295	0.25881	0.303653
299086	100	0.512722	0.592702	0.549819
300091	1200	0.534267	0.861824	0.659619
302008	100	0.50974	0.284349	0.365058
304034	200	0.741198	0.572361	0.645929
304074	300	0.619014	0.483693	0.54305
306005	100	0.61116	0.358153	0.451637
351093	200	0.608481	0.393976	0.478279
361010	100	0.616669	0.379183	0.469608
376043	100	0.68794	0.388027	0.496185
385039	800	0.60857	0.65289	0.629952

Tez kapsamında önerilen algoritmanın BSDS300 görüntü bazında en iyi sonuçları (*σ = 1.2; l = 100 - 2000; c = 0.96*)

Görüntü Dosyası Adı	En İyi Sonuç Verdiği *l* Değeri	Precision	Recall	F-score
3096	900	0.685059	0.693055	0.689033
8023	100	0.643591	0.732249	0.685063
12084	600	0.574291	0.40575	0.475528
14037	100	0.606446	0.546701	0.575026
16077	800	0.594592	0.547093	0.569854
19021	100	0.528006	0.438893	0.479343
21077	200	0.485832	0.413575	0.446801
24077	100	0.37276	0.26415	0.309194
33039	100	0.615028	0.226576	0.331155
37073	100	0.43636	0.361698	0.395536
38082	600	0.788861	0.684494	0.732981
38092	400	0.503097	0.478515	0.490498
41033	400	0.482639	0.313502	0.380105
41069	200	0.700307	0.602034	0.647463
42012	400	0.483099	0.395973	0.435219
42049	100	0.713685	0.608929	0.657159
43074	400	0.449043	0.86008	0.590033
45096	100	0.594634	0.691628	0.639474
54082	300	0.776581	0.818412	0.796948
55073	100	0.813119	0.633231	0.711989
58060	400	0.687581	0.497345	0.577193
62096	400	0.63902	0.544658	0.588078
65033	1.500	0.454215	0.735697	0.561663
66053	400	0.701943	0.81427	0.753946
69015	400	0.630136	0.58469	0.606563
69020	200	0.595877	0.288491	0.388764
69040	900	0.512471	0.686111	0.586714
76053	100	0.368326	0.294008	0.326998
78004	100	0.300314	0.667513	0.414255
85048	1700	0.557051	0.711722	0.624959
86000	100	0.525873	0.549508	0.537431
86016	100	0.748018	0.410812	0.530353
86068	100	0.751162	0.640679	0.691536
87046	100	0.830809	0.617765	0.708621
89072	1400	0.648586	0.754698	0.69763
97033	500	0.658355	0.72396	0.689601
101085	100	0.604704	0.760741	0.673807
101087	300	0.797228	0.759701	0.778012
102061	100	0.524952	0.605148	0.562205
103070	200	0.543328	0.490789	0.515723
105025	200	0.785095	0.629682	0.698852

106024	100	0.755925	0.723384	0.739296
108005	100	0.612126	0.465373	0.528756
108070	600	0.499758	0.679149	0.575805
108082	100	0.717555	0.573399	0.637429
109053	100	0.476454	0.851033	0.610896
119082	700	0.64644	0.573124	0.607578
123074	100	0.67065	0.376131	0.481958
126007	300	0.622409	0.620469	0.621437
130026	300	0.3782	0.709043	0.493285
134035	100	0.717395	0.564236	0.631664
143090	100	0.604184	0.795605	0.686806
145086	100	0.572664	0.785517	0.662411
147091	500	0.253919	0.988636	0.40406
148026	200	0.714727	0.633353	0.671584
148089	100	0.74671	0.645451	0.692398
156065	400	0.547118	0.400893	0.462728
157055	300	0.682314	0.652314	0.666977
159008	100	0.633494	0.695409	0.663009
160068	200	0.754192	0.834874	0.792485
163085	100	0.586522	0.852006	0.694766
167062	200	0.666723	0.614797	0.639708
167083	100	0.504338	0.431295	0.464965
170057	1.500	0.539057	0.344195	0.420131
175032	800	0.657346	0.551336	0.599692
175043	300	0.535708	0.728859	0.617532
182053	200	0.608535	0.939772	0.738722
189080	300	0.728427	0.426618	0.538092
196073	700	0.668334	0.685699	0.676905
197017	100	0.76448	0.61399	0.68102
208001	100	0.66179	0.906645	0.765105
210088	200	0.603561	0.36689	0.456366
216081	100	0.589222	0.468218	0.521797
219090	200	0.660407	0.606763	0.632449
220075	200	0.625419	0.573823	0.598511
223061	100	0.527947	0.491097	0.508856
227092	500	0.611698	0.580302	0.595587
229036	100	0.649784	0.535342	0.587037
236037	200	0.842368	0.857543	0.849888
241004	100	0.36372	0.671548	0.471869
241048	100	0.442507	0.793266	0.568107
253027	100	0.412881	0.642265	0.50264
253055	200	0.600959	0.480018	0.533723
260058	100	0.478545	0.382581	0.425216
271035	1300	0.413338	0.821322	0.549922
285079	100	0.774556	0.512735	0.61702
291000	100	0.664694	0.486932	0.562094

295087	100	0.752751	0.765083	0.758867
296007	300	0.616947	0.694593	0.653472
296059	100	0.494723	0.287063	0.363313
299086	100	0.516784	0.598893	0.554817
300091	500	0.706264	0.701921	0.704086
302008	100	0.379679	0.140687	0.205302
304034	500	0.681708	0.664844	0.67317
304074	300	0.646298	0.550295	0.594445
306005	100	0.497286	0.400661	0.443774
351093	300	0.649376	0.46974	0.54514
361010	200	0.66168	0.500478	0.569899
376043	100	0.708592	0.393113	0.505683
385039	300	0.694703	0.571038	0.62683

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı	**:**	Ali Sağlam
Uyruğu	**:**	T.C.
Doğum Yeri ve Tarihi	**:**	ANKARA / 09.10.1983
Telefon	**:**	0-505-393 91 30
Faks	**:**	--------
e-mail	**:**	alisaglam@selcuk.edu.tr

EĞİTİM

Derece		**Adı, İlçe, İl**	**Bitirme Yılı**
Lise	:	Ankara Aydınlıkevler Lisesi	2000
Üniversite	:	Çukurova Üniversitesi	2004
Üniversite	:	Ahmet Yesevi Üniversitesi	2012
Yüksek Lisans	:	Selçuk Üniversitesi	Devam

İŞ DENEYİMLERİ

Yıl	**Kurum**	**Görevi**
2004-2012	MEB	Bilgisayar Öğretmeni
2012-2014	Karabük Üniversitesi	Araştırma Görevlisi
2014-Devam	Selçuk Üniversitesi	Araştırma Görevlisi

UZMANLIK ALANI

Görüntü İşleme, Graf Teorisi, Paralel Programlama

YABANCI DİLLER

İngilizce

YAYINLAR

1) Saglam A. ve Baykan N. A., 2015, "An Efficient Object Extraction with Graph-Based Image Segmentation", ACSAT'2015 4th International Conference on Advanced Computer Science Applications and Technologies, Kuala Lumpur, Malaysia (Yüksek Lisans tezinden yapılmıştır)

2) Saglam A. ve Baykan N. A., "Sequential image segmentation based on minimum spanning tree representation", Pattern Recognition Letters (Hakem İncelemesinde)