

**GÖMÜLÜ TEKNİKLER KULLANILARAK ÇİZGE MEDYANININ  
HESAPLANMASI**

**AHMET SORAN**

**YÜKSEK LİSANS TEZİ  
BİLGİSAYAR MÜHENDİSLİĞİ**

**TOBB EKONOMİ VE TEKNOLOJİ ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ**

**HAZİRAN 2012**

**ANKARA**

Fen Bilimleri Enstitü onayı

---

Prof. Dr. Ünver KAYNAK

Müdür

Bu tezin Yüksek Lisans derecesinin tüm gereksinimlerini sağladığını onaylarım.

---

Doç. Dr. Erdoğan DOĞDU

Anabilim Dalı Başkanı

Ahmet SORAN tarafından hazırlanan GÖMÜLÜ TEKNİKLER KULLANILARAK ÇİZGE MEDYANININ HESAPLANMASI adlı bu tezin Yüksek Lisans tezi olarak uygun olduğunu onaylarım.

---

Yrd. Doç. Dr. Muhammed Fatih DEMİRCİ

Tez Danışmanı

Tez Jüri Üyeleri

Başkan : Yrd. Doç. Dr. A.Murat ÖZBAYOĞLU \_\_\_\_\_

Üye : Yrd. Doç. Dr. M.Fatih DEMİRCİ \_\_\_\_\_

Üye : Yrd. Doç. Dr. Zülfükar SAYGI \_\_\_\_\_

## TEZ BİLDİRİMİ

Tez içindeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edilerek sunulduğunu, ayrıca tez yazım kurallarına uygun olarak hazırlanan bu çalışmada orijinal olmayan her türlü kaynağa eksiksiz atıf yapıldığını bildiririm.

AHMET SORAN

Aileme, Hocalarıma, Sevdiklerime ve Beni Sevenlere

Her şeyini kaybettiğin anda artık özgürsündür..

ELFİDA

**Üniversitesi** : TOBB Ekonomi ve Teknoloji Üniversitesi  
**Enstitüsü** : Fen Bilimleri  
**Anabilim Dalı** : Bilgisayar Mühendisliği  
**Tez Danışmanı** : Yrd. Doç. Dr. Muhammed Fatih DEMİRCİ  
**Tez Türü ve Tarihi** : Yüksek Lisans – Haziran 2012

**Ahmet SORAN**

## **GÖMÜLÜ TEKNİKLER KULLANILARAK ÇİZGE MEDYANININ HESAPLANMASI**

### **ÖZET**

Graph (Çizge) Teorisinin birçok uygulaması görüntü işleme, bilgisayarla görü, biyoinformatik, veri madenciliği ve yapay zeka çalışmalarında sıklıkla kullanılmaktadır. Objelerin çizge gösterimiyle gösterilmesi ve resim eşleştirilmesi ise nesne tanıma çalışmalarında önemli bir yer kaplamaktadır. Birçok örnek resim verilerinde gürültü miktarı doğal koşullar neticesinde fazlaca bulunmaktadır ve elde edilen sonuçlarda da hata miktarı fazla çıkmaktadır. Her bir resmin çizge gösterimiyle gösterilmesi sonucu eşleştirme problemi özünde çizge karşılaştırma problemiyle eşdeğer bir hal almaktadır. Veri miktarının fazla olması ve çizge karşılaştırma işleminin maliyetli olması sebebiyle yakınsama algoritmaları (approximation algorithms) kullanılarak, yapılan işlemin maliyetini azaltma çalışmaları yapılmaktadır. Verilen çizgeleri karşılaştırmak için her grup çizge için, o grubu temsil eden yeni bir çizge hesaplanarak, aranan çizgeyi temsilci çizge ile karşılaştırmanın daha hızlı ve tolere edilebilecek seviyede gerçeğe yakın sonuçlar vereceği öngörülmüştür. Daha önce yapılan çalışmalarda, temel olarak, iki çizgenin karşılaştırılması için çizge uzayındaki her bir çizgeyi, temsil eden vektör uzayına dönüştürme ve daha az maliyetle vektör uzayında işlemler yapma üzerinde durulmuştur. Yapılan çalışmada, her veri sınıfı için veri kümemiz içindeki tüm çizgeler, veri kaybını minimum düzeyde tutacak şekilde, geometrik uzaya izomorfik olarak  $\ell_1$  norm (Manhattan Uzaklık Metriği) altında dönüştürülmüştür. Bu sayede temsilci çizge bulunması esnasında yapılacak işlemlerin masrafları azaltılmış ve kullanmış olduğumuz Tırtıl Ayrışması (Caterpillar Decomposition) tekniğiyle minimum seviyede veri kaybı olması sağlanmıştır. Daha önce yapılan çalışmalardan farklı olarak, elde edilen geometrik uzayda her bir nokta, çizge üzerindeki bir düğüme (node) denk gelmektedir ve bu sayede veri kaybı azalmaktadır. Temsilci çizgenin belirlenmesi için, oluşturulan vektör uzayında, K-Means Algoritması kullanılmaktadır. Elde edilen temsilci nokta kümesi ile çizgeleri karşılaştırmak için

Hausdorff Mesafesi algoritmasının 25 farklı hesaplama metriđi kullanılmaktadır. Bu iřleyiř yapısı ierisinde minimum seviyede hatayla daha hızlı sonular elde etmek mmkn olmaktadır.

**Anahtar Kelimeler:** izge Teorisi, izge Gmme, izge Medyanı, Tırtıl Ayrıřması, K-Means Kmeleme, Hausdorff Mesafesi

**University** : **TOBB University of Economics and Technology**  
**Institute** : **Institute of Natural and Applied Sciences**  
**Science Programme** : **Computer Engineering**  
**Supervisor** : **Assistant Professor Muhammed Fatih DEMİRCİ**  
**Degree Awarded and Date** : **M.Sc. – June 2012**

**Ahmet SORAN**

## **FINDING GRAPH MEDIANS USING GRAPH EMBEDDING TECHNIQUES**

### **ABSTRACT**

Graph theory applications are frequently used in image processing, computer vision, bioinformatics, data mining and artificial intelligence related works. On the other hand, representing objects with graphs and matching the images are important part of object recognitions applications. In most of sample image data, images have a lot of noise caused by natural conditions and the error levels in the final results are high. When each image is represented by a graph, matching problem inherently turns to be the same as graph comparison. Since the amount of data is huge and the graph comparison is costly, approximation algorithms are used to decrease the total cost. To compare the given graphs, for each group of graphs, a representative is selected. It is anticipated that, comparing the graph only with the representative graphs gives faster results and the results are tolerably close to the exact truth. In the previous works, to compare the graphs, mapping of each graph in the graph space into a vector in the vector space and then working on the vector space with minimum cost is used. In this work, for each data class we have, each graph in the set is mapped to geometric space by using isomorphic  $\ell_1$  norm (Manhattan Distance metric), so that, the data lose is minimum. Consequently, the cost of finding the representative graph is decreased and the data lose is kept at the minimum by using Caterpillar Decomposition technique. Differently from the previous works, each point in the geometric space represents a node in the graph. Thus, the data lose is lowered. In the vector space, K-Means Algorithm is used to find the representative graph. To compare the representative points set with the graphs, 25 different distance metric calculations of Hausdorff Distance Algorithm is used. This working mechanism makes it possible to have faster results with minimum error level.

**Keywords:** Graph Teory, Graph Embedding, Graph Median, Caterpillar Decomposition, K-Means Clustering, Hausdorff Distance



# İçindekiler

<b>ÖZET</b>	<b>6</b>
<b>ABSTRACT</b>	<b>8</b>
<b>1. GİRİŞ</b>	<b>12</b>
1.1. <i>Temel Bilgiler</i>	16
1.1.1. Çizge Teorisi	16
1.1.2. NP-Problemler	19
1.1.3. Yakınsama Algoritmaları	19
1.1.4. Kümeleme Algoritmaları	20
<b>2. LİTERATÜR TARAMASI</b>	<b>21</b>
2.1. <i>Geçmiş Çalışmalar</i>	21
2.2. <i>Tırtıl Ayrışması</i>	23
2.3. <i>K-Means Kümeleme Algoritması</i>	24
2.4. <i>Nokta Kümesi Eşleme Algoritmaları</i>	24
<b>3. YAPILAN İŞLEMLER</b>	<b>27</b>
3.1. <i>Giriş</i>	27
3.2. <i>İşlem Adımları</i>	27
3.2.1. Medyanların Çıkarılması	27
3.2.2. Karşılaştırma İşlemi	30
3.3. <i>İşlemlerin Detayları</i>	31
3.3.1. Tırtıl Ayrışması	31
3.3.2. Ekleme(Padding) İşlemleri	36
3.3.3. K-Means Kümeleme	36
3.3.4. Hausdorff Mesafesi	38
<b>4. DENEYSEL SONUÇLAR</b>	<b>40</b>
4.1. <i>Amaç</i>	40
4.2. <i>Veri Kümesi</i>	41
4.3. <i>Elde Edilen Sonuçlar</i>	43
4.3.1. Medyan hesaplanması analizi	43
4.3.2. Sonuçların Karşılaştırmaları	47
<b>5. SONUÇ</b>	<b>65</b>
5.1. <i>Değerlendirme</i>	65
5.2. <i>Gelecek Çalışmalar</i>	70
<b>KAYNAKLAR</b>	<b>72</b>

## ÇİZELGELERİN LİSTESİ

Çizelge 3.1 Tırtıl Ayrışması Yol Çizelgesi.....	35
Çizelge 4.1 K-Means Analiz Çizelgesi.....	44
Çizelge 4.2 Medyan ile Karşılaştırma Sonuçları.....	49
Çizelge 4.3 Medyan Kullanılarak Gerçekleştirilen Hibrit Örneği .....	51
Çizelge 4.4 Medyan Kullanılarak Hibrit ile Ortalama Karşılaştırması .....	52
Çizelge 4.5 Medyan Kullanılmadan Karşılaştırma Sonuçları .....	54
Çizelge 4.6 Medyan Kullanılmadan Gerçekleştirilen Hibrit Örneği .....	56
Çizelge 4.7 Medyan Kullanılmadan Hibrit ile Ortalama Karşılaştırması .....	57
Çizelge 4.8 Medyanlı ve Medyansız Ortalama Başarımın Karşılaştırılması .....	58
Çizelge 4.9 Medyanlı ve Medyansız Başarımın Karşılaştırılması .....	60
Çizelge 4.10 Medyan Kullanılmadan Karşılaştırma Süresi .....	62
Çizelge 4.11 Medyan Kullanılarak Karşılaştırma Süresi.....	63

## ŞEKİLLERİN LİSTESİ

Şekil 1.1 Çizge Gömme Adımları .....	14
Şekil 1.2 Çizge Medyanı Çıkarma .....	14
Şekil 1.3 Karşılaştırma Adımları.....	15
Şekil 1.4 Minimum Kapsama Ağacı .....	18
Şekil 3.1Tırtıl Ayrışması Örneği .....	32
Şekil 3.2 Tırtıl Ayrışması Örneği - Yollar .....	34
Şekil 4.1 Veri Sınıfları.....	41
Şekil 4.2 Sandalye Sınıfına(cha) Ait Objeler .....	42
Şekil 4.3 Bardak Sınıfına(brd) Ait Objeler.....	42
Şekil 4.4 K-Means Süre Analizi .....	44
Şekil 4.5 K-Means Performansına Etki Eden Faktörler.....	45
Şekil 4.6 K-Means İşlem Adımları Analizi .....	46
Şekil 4.7 Medyanlı Hibrit ve Ortalama Sonuçların Karşılaştırılması .....	52
Şekil 4.8 Medyanlı/Medyansız Sonuçların Karşılaştırılması .....	59
Şekil 4.9 Hibrit Yöntemi İçin Medyanlı/Medyansız Karşılaştırması.....	60
Şekil 4.10 Ortalama Yöntemi İçin Medyanlı/Medyansız Karşılaştırması.....	61
Şekil 4.11 Süre Bazlı Medyanlı/Medyansız Karşılaştırması .....	64

# BÖLÜM 1

## 1. GİRİŞ

Bilgisayarla görü ve görüntü işleme uygulamaları, elde edilen görüntülerin bilgisayar ortamında zaman zaman akıllı yöntemler de kullanılarak işlenmesini ve gerçek hayatta yaptığımız işlemleri bilgisayar ortamında gerçekleştirebilmemizi sağlamaktadır[4]. Bu uygulamaların en önemlilerinden biri, objelerin birbiriyle karşılaştırılarak birbirlerine benzerliklerinin tespit edilmesidir. Bu sayede insanın farkında olmadan otomatik olarak yaptığı görüntülerin tanınması ve eşleştirilmesi işlemlerinin bilgisayar ortamında da yapılabilmesine çalışılmaktadır. Bu yöntemle çok büyük bir veri kümesi içerisinde de aranan resmin, logonun veya herhangi bir objenin hızlı bir şekilde bulunması amaçlanmaktadır.

Obje eşleştirme uygulamaları içerisinde öznitelik eşleştirme (feature matching) sıklıkla kullanılan ve son zamanlarda daha hızlı ve daha az hatayla sonuçlar alabilmek için üzerinde çalışılan temel bilgisayarla görü konularından biridir. Karşılaştırılacak iki obje için karakteristik özniteliklerin belirli yöntemlerle çıkarılması ve daha sonra bu iki objenin özniteliklerinin karşılaştırılması yordamıyla iki objenin birbirine olan benzerliklerinin tespit edilmesi amaçlanmaktadır [5]. Özniteliklerin bire bir eşleştirilmesi veya çoklu eşleştirilmesi uygulamaları son dönemlerde üzerinde çalışılan konulardır [1].

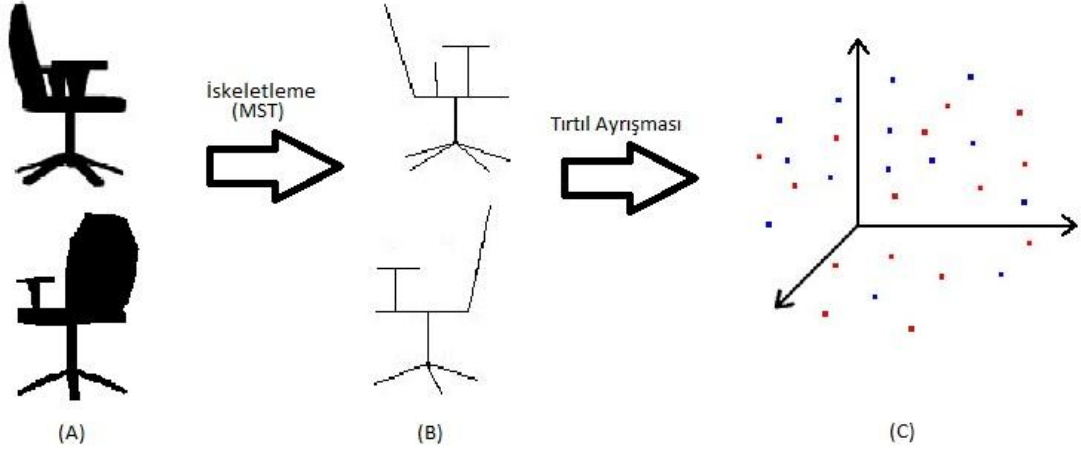
Her bir objenin karakteristik özniteliklerinin en popüler ifade biçimlerinden biri çizgelerdir. Objelerin çizgelerle ifade edilmesi, problemi obje eşleştirme probleminden iki çizgenin benzerliği problemine indirgemektedir. Burada dikkat edilmesi gereken husus olabildiğince minimum veri kayıplarıyla bu dönüşümün yapılmasıdır. İlişkisel olarak iki çizgenin birbirine olan uzaklıklarının hesaplanmasının ardından, ana objeler için aynı sonucun doğruluğunu ifade edebilmemiz için bu kayıpların en az seviyede tutulması ve çizgenin izomorfik olarak üretilmesi gerekmektedir.

Elde edilen çizgelerin benzerliklerinin hesaplanması ise başka bir problemi beraberinde getirecektir. İki çizgenin birbirlerine olan benzerliklerinin tespiti NP-Tam bir problemdir. Bu tarz problemlerin kesin sonuçlarının elde edilmesi oldukça zordur. Yapılması gereken işlemler çok fazla zaman ve işlem gerektirmektedir. Bu

problemin önüne geçebilmek için çeşitli sezgisel fonksiyonları ve bir takım yakınsama algoritmaları kullanılmaktadır [6]. Yapılan çalışmalar göz önünde bulundurulursa, oldukça sık kullanılan yöntemlerden biri çizge uzayından, yeni oluşturulan vektör uzayına minimum veri kaybıyla geçilip, gerçekleştirilecek işlemlerin daha kolay uygulanabileceği bir ortamda yapılmasıdır.

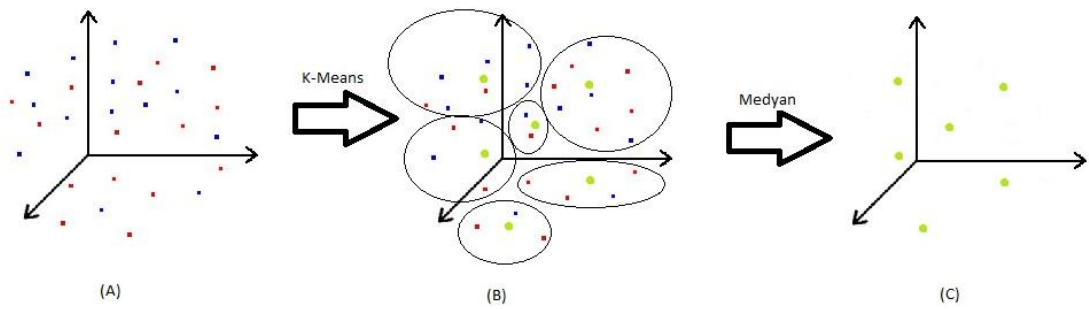
Literatür bilgisi ve yukarıda belirtilen temel bilgiler göz önünde bulundurulursa, sunulan tezin amacı, minimum veri kaybı, minimum işlem ve en güvenilir sonuçla obje karşılaştırma işlemlerini yerine getirebilecek yeni bir çözüm oluşturmaktır. Senaryo olarak elimizde bulunan veri kümesi ile sistemimizi oluşturup, daha sonra verilen bir objenin hangi veri sınıfına ait olabileceğinin bulunması belirlenmiştir. Çizgelerin güçlü temsil yeteneklerinden dolayı elimizdeki verilerin ilk olarak iskeletleme yöntemiyle çizge formuna çevrilmesi ve işlemlerin çizgeler üzerinde devam edilmesi sağlanmıştır. Elimizdeki çizgeler içerisinde temsilci bir çizge belirlenerek, mevcut sınıf için bir temsilci oluşturulması planlanmıştır. Bu sayede karşılaştırılacak olan veriyi, veri kümesindeki tüm objeler ile karşılaştırmak yerine temsilcilerimizle karşılaştırıp çok daha hızlı bir şekilde verinin hangi veri sınıfına dahil olabileceğinin belirlenmesi sağlanmıştır. Seçtiğimiz temsilcinin mevcut sınıf içerisindeki tüm verilere uzaklıklarının toplamının olabildiğince az olması gerekmektedir. Bu şekilde seçilen temsilci tüm çizgelerin medyanı olarak da ifade edilmektedir. Temsilci seçme işleminin çizgeler üzerinde yapılması oldukça zorlu bir süreç olacağı için verilen çizgeler Tırtıl Ayrışması(Caterpillar Decomposition) yöntemiyle vektör uzayına geçirilmiştir. Vektör uzayına aktarılan veriler içinden, K-Means yöntemiyle temsilci içerisinde bulunması gereken temel noktalar belirlenmiş ve temsilci nokta kümemiz oluşturulmuştur. Bu aşamada elde edilen temsilci ile diğer verilerin karşılaştırılması için yapılan işlemler de vektör uzayında yapıldığı için temsilci nokta kümemizin tekrar çizge uzayına dönüştürülmesi gerekmemektedir. Temsilci ile sorgulanan verinin birbirlerine olan uzaklıklarının belirlenmesi, nokta kümelerinin birbirlerine olan uzaklıklarının belirlenmesi problemdir. Tezin bu aşamasında ise Hausdorff Mesafesi algoritmasınının 25 farklı versiyonu kullanılmıştır. Elde edilen sonuçlar ve algoritmanın başarısı ve değerlendirme Bölüm 4 ve Bölüm 5 içinde tartışılmıştır.

Tez çalışmasında her bir veri sınıfındaki objelerin vektör uzayına gömülmesi, her veri sınıfı için medyanın hesaplanması ve karşılaştırma işlemi yapılmaktadır. Yapılan çalışmanın işlem adımları sırasıyla Şekil 1.1, Şekil 1.2 ve Şekil 1.3’de verilmiştir.



**Şekil 1.1 Çizge Gömme Adımları**

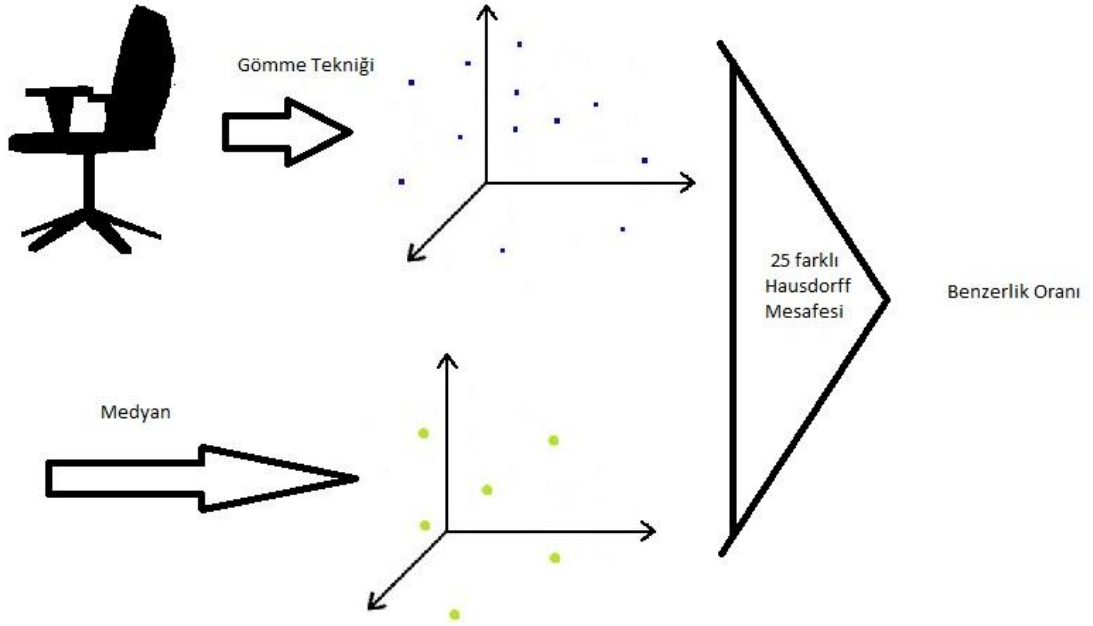
Veri sınıfı içinde bulunan her bir obje Şekil 1.1’de gösterildiği gibi vektör uzayında temsil edilmektedir. (A) kısmında gösterilen objelerin iskeletleri çıkarılarak (B) kısmındaki yapılara çevrilmektedir. Bu geçiş esnasında minimum kapsama ağacı algoritması kullanılmaktadır. (B)’de gösterilen minimum kapsama ağaçları Tırtıl Ayrışması yönteminin kullanılmasıyla ağaçtaki her bir düğüm vektör uzayında bir nokta ifade edecek şekilde vektör uzayında gösterilmektedir(C kısmı).



**Şekil 1.2 Çizge Medyanı Çıkarma**

Şekil 1.1’de elde edilen vektör uzayı Şekil 1.2 (A) ile belirtilmiştir. Mevcut nokta kümesi üzerinde K-Means uygulamasıyla tüm noktaların temsilcilerinden oluşan yeni nokta kümesi oluşturulmaktadır. (B)’de oluşturulan yeni nokta kümesinin

temsilcileri veri sınıfının medyanı olmaktadır. Bu noktalar (C) kısmında gösterilmiştir.



Şekil 1.3 Karşılaştırma Adımları

Veri sınıfı için elde edilen medyan ile sorgulanacak ve sınıflandırılacak olan obje ilk olarak çizge gömme tekniğiyle vektör uzayına geçirilmektedir(Şekil 1.3). Vektör uzayında temsil edilen medyan ile karşılaştırılması ise 25 farklı Hausdorff Mesafesi yardımıyla tespit edilmektedir. Sorgulanacak olan obje tüm veri sınıflarına ait medyanlar ile karşılaştırılarak en yakın sonucun bulunduğu sınıfa ait olduğu sonucuna varılmaktadır.

Tezin ilk bölümünde temel bilgilerden bahsedilmiştir. İkinci bölümde daha önce yapılan çalışmalara atıfta bulunulmuştur. İlk olarak Tırtıl Ayrışması ve çizge gömme teknikleri daha sonra K-Means ve nokta kümeleri arasındaki uzaklığın bulunmasından bahsedilmiştir. Üçüncü bölümde yaptığımız çalışmalar, dördüncü bölümde testlerimiz ve elde ettiğimiz sonuçlar, beşinci bölümde sonuçların değerlendirilmesi ve gelecek çalışmalar anlatılmıştır.

## 1.1. Temel Bilgiler

Tez çalışmamız temel çizge teorisi bilgisi, algoritmik olarak problemlere yaklaşım, problemlere genel matematiksel yaklaşım, kümeleme algoritmaları hakkında temel bilgilerin üzerine inşa edilmiştir.

### 1.1.1. Çizge Teorisi

Çizge teorisi daha çok matematik ve bilgisayar bilimleri alanında kullanılmaktadır. Bir takım nesnenin kendi aralarındaki ilişki modelini matematiksel yöntemlerle açıklamak için kullanılır. Çizgeler iki temel kavram üzerine inşa edilmiştir. Bunlardan ilki olan düğümler (node), sabit noktalardır. İkinci temel kavram ise kenarlardır (edge). Kenarlar iki düğüm arasındaki ilişkiyi temsil etmektedirler. Bir çizge matematiksel olarak  $G = (E, V)$  ikilisiyle gösterilir. Bu yapıda E ve V değerleri birer küme olup, V kümesi düğümleri E kümesi de düğümler arasındaki ilişkileri gösteren kenarları temsil etmektedir. Örnek olarak bir çizge  $V = \{1, 2, 3, 4, 5\}$  düğümleri ve  $E = \{(1,2), (1,4), (1,5), (2,3), (2,5), (3,5), (4,5)\}$  kenarlarından oluşabilir. Çizgeler yönsüz (undirected) veya yönlü (directed) olarak tasarlanabilir. Yönsüz çizgelerde E kümesi içerisindeki her kenarın tersi de mevcut demektir. Yönlü çizgelerde ise E kümesindeki her bir kenar yönlü bir şekilde ifade edilmiş olur.

### Temel Çizge Kavramları

- **Düğüm:** Çizgenin ana yapısını oluşturan sabit noktalardır.
- **Kenar:** İki düğüm arasındaki ilişkiyi belirlemek için kullanılan kavramdır. İki çıkışı vardır ve her ikisi de bir düğümle ifade edilir.
- **Ağırlık:** Herhangi bir düğüm için veya kenar için belirtilen değerdir.
- **Derece:** Bir düğümden çıkan kenar sayısına verilen isimdir. Başka bir deyişle verilen bir çizge içerisindeki tüm düğümlerin derecelerinin toplamı kenar sayısının iki katını verecektir.
- **Yol:** Bir düğümden başka bir düğüme bağlantılı yollardan gitmek için kullanılan sıralı düğümler veya kenarlar kümesidir.
- **Döngü:** Bir yol üzerinde başladığımız noktaya tekrar dönebiliyorsak bu yol üzerinde döngü var demektir.
- **Root:** Ağaçlarda kullanılan bir terimdir, başlangıç düğümünü ifade eder.



- Ebeveyn Dügüm: Gidilen bir düğüme hangi düğümden gelindiğinin belirlenmesidir.
- Yaprak Dügüm: Ağaç yapısı içinde derecesi bir olan düğümdür.

### Çizge Türleri

- Yönsüz çizge: Kenarlar simetrikse ve tek yönlü kenar yoksa verilen çizge yönsüz çizgedir.
- Yönlü çizge: Kenarların simetrik olmadığı ve sadece gösterilen yönde bir kenar olduğunu belirtir çizgelerdir.
- Ağırlıklı çizge: Verilen kenarlarda veya düğümlerde ağırlık bilgisi bulunan çizgelerin tanımlanma biçimidir.
- Tam çizge: Verilen çizgede tüm düğümlerden tüm düğümlere kenar bulunuyorsa verilen çizge tam çizgedir.
- Ağaç: Herhangi iki düğümler arasında sadece bir yol varsa ve çizge bağlı ise verilen çizge ağaç olarak ifade edilmektedir. Ağaçların başlangıç düğümünü belirten düğümler ise “root”(kök) düğümleri olarak ifade edilmektedir.
- Alt Çizge: Verilen bir çizgeden çıkarılabilecek yeni çizgelerin her biri alt çizgedir.

### Uygulamalar

Çizgeler günümüzde birçok bilgisayar bilimi alanında kullanılmaktadır[7]. Verilen öznitelikler arasındaki bağları belirtmek için çizgeler sıklıkla tercih edilmektedir. Daha çok ilişkisel gösterimlerde kullanılır. Bu nedenle çok karmaşık sistemlerde en çok başvurulan yöntemlerdendir. Network, veri organizasyonu, akış problemleri, sinir ağları, kimya ve biyoinformatik projeleri, sosyoloji ve sosyal ağlarda, matematiksel problemlerin çözümünde ve topolojilerde ilişkileri belirlemek için kullanılır[8,9].

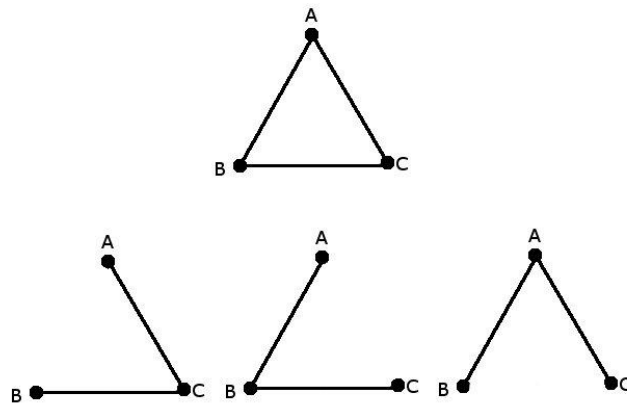
### Minimum Kapsama Ağacı:

En çok bilinen uygulamalardan olan Gezgin Satıcı Problemi (Travelling Salesman Problemi) [10] gibi bazı problemler için verilen çok düğümlü ve çok kenarlı bir çizgenin daha küçük alt çizgesini kullanmamız gerekmektedir. Özellikle seçilmiş bazı düğümlerin kullanılması ve alt çizgelerden oluşturulacak ağaç yapılarının

minimum kapsayan şekilde olması, bazı problemler için olması gereken bir uygulamadır. Görüntü işleme tekniklerinde, iskeletleme yöntemi sıklıkla kullanılmaktadır ve bu yaklaşım için minimum kapsama ağacı uygulaması gerçekleştirilmektedir [11]. Bu durumlarda verilen bir çizgeden minimum kapsama ağacının çıkarılması uzunca bir süredir çözümü olan bir problemdir. Minimum kapsama ağacından beklentimiz, bir düğümden başka bir düğüme sadece bir yoldan gitmek istemek ve bu yolların toplamının olabilecek en kısa şekilde belirlenmesini sağlamaktır [12]. Daha açık olarak, bir çizgenin kapsama ağacı o çizgenin tüm düğümlerini birbirine bağlayan bir ağaçtır. Minimum kapsama ağacı ise o çizgenin tüm kapsama ağaçları içinde toplam ağırlığı minimum olan ağaçtır.

Bu işlemin gerçekleştirilmesi için iki temel felsefe uzun yıllardır kullanılmaktadır. Bu yaklaşımlardan birincisi temel düğümün seçilmesi ve döngü olmayacak şekilde bu düğüm kümesine yeni düğümlerin eklenmesidir [13]. İkinci temel yöntem ise tüm kenarlar içerisinden döngü olmayacak şekilde en küçük kenarların düğümlere eklenmesi yöntemidir [14].

Bu algoritmaların tezimize etkisini göz önünde bulunduracak olursak, her bir çizge için tekil bir ağaç çıkarılamaması, yapmış olduğumuz işlemlerin her defasında aynı sonucu vermesini engellemektedir. Bu nedenle de sonuçlarımız her zaman aynı olmamakta ve yanlış sonuç verme ihtimalini artırmaktadır. Algoritmaların doğası gereği tüm kenar uzunluklarının birbirinden farklı olması durumunda ortaya çıkarılan ağaç, tekil ve en kısa ağaç olacaktır [15]. Ancak aynı değere sahip birden fazla kenar olması durumunda ortaya çıkabilecek minimum kapsama ağacı tekil olmayacaktır. Tüm kenarların ağırlıkları birbirine eşit olduğu için oluşturulabilecek olan üç farklı ağaç yapısı Şekil 1.4’de gösterilmiştir.



Şekil 1.4 Minimum Kapsama Ağacı

Minimum Kapsama Ağacının sonuçları tekil olmamaktadır.

Objelerin alınıp, çizge haline dönüştürülmesinde kullanılan minimum kapsama ağacı algoritmalarının tezimize olan etkisi tekil çözümler üretmiyor olmasıdır. Bu nedenle farklı zaman veya farklı açılarda aynı objeye ait çok farklı minimum kapsama ağacı gösterimleri elde edilebilmektedir. Bunun neticesinde ortaya çıkan sonuçlar da beklenen doğruluk düzeyinde çıkamayabilmektedir. Deney kümesi içerisinde, minimum kapsama ağaçlarının çıkarılmış olduğu, iskeletleme yönteminin gerçekleştirildiği, veriler bulunmaktadır.

### 1.1.2. NP-Problemler

Bilgisayar bilimlerinde problem çözümlerinin performansları matematiksel olarak karmaşıklık tespitiyle yapılmaktadır. Bir problemin çözümü eğer polinom( $n^c$ ; n girdinin eleman sayısı, c sabit) zamanda yapılabilir ise bu problem basit problem olarak adlandırılmaktadır (P Sınıfı). Verilen bazı problemlerin çözümü şu ana kadar c ne kadar büyük olursa olsun polinom zamanda bulunamamıştır. NP-Tam sınıfı şu ana kadar çözümü polinom zamanda bulunamayan fakat verilen potansiyel bir çözümün (sertifika) doğrulamasını polinom zamanda yapabilen problemlerden oluşmaktadır.

Veri olarak verilen “n” adet çizgenin medyanının bulunması problemi NP-Tam problemlere bir örnektir. Verilen çizgeler içerisinde öyle bir çizge çıkarılmalıdır ki çıkarılan çizgenin tüm çizgelere olan mesafeleri toplamı minimum olmalıdır. [16]

$$S = \{g_1, g_2, \dots, g_n\} \subset U$$

$$g' = \operatorname{argmin}_{g \in U} \sum_{g_i \in S} d(g, g_i)$$

### 1.1.3. Yakınsama Algoritmaları

Bilgisayar bilimlerinde birçok problemin çözümü polinom zamanda yapılabilir. Fakat bazı durumlarda problemin karmaşıklığının üssel olması pratikte tam bir çözüm yerine çözüme belli bir oranda yaklaşan yakın bir çözümü bulmayı gerektirmektedir. Çoğunlukla NP-Tam problemlerinin kesin çözümlerini elde etmek yerine yakınsama algoritmaları kullanılmaktadır.

Çizge medyanı bulunması NP-Tam bir problemlerden biridir. Bu sebeple tam çözüm bulunmasına çalışmak yerine yakınsama algoritmalarının kullanılmasıyla çözüme belirli oranda yakın bir çözüm bulunması hedeflenmektedir. Yakınsama algoritması olarak çizgelerin vektör uzayına gömülmesi ve bu uzay üzerinde medyan çıkarma işlemlerinin yapılması amaçlanmıştır.

#### **1.1.4. Kümeleme Algoritmaları**

Kümeleme, birbirine benzer noktaların belirli metrikler kullanılarak bir araya getirilmesi işlemleridir. Genelde istatistiksel modeller kullanılarak noktalar kümesi içerisinde kategorilerin ortaya çıkarılmasını amaçlamaktadır [17]. Elimizde bulunan veriler üzerinden bir takım mesafe metriklerinin [18] de yardımıyla birbirine işlevsel olarak, şekil olarak vb. benzeyen verilerin ortak bir küme bütünlüğünde toplanmasını sağlamaktadır. Bu sayede verilerin birbirleriyle ortaklık gösteren yönleri belirlenmektedir. Yoğun veri kümeleri içerisinde birbirine fonksiyonel olarak benzeyen verilerin belirlenmesi, çeşitli çalışmaların daha hızlı ve daha doğru sonuçlarla yürütülmesi için gerekli olmaktadır. Bu teknik ve değişik türleri, günümüzde sosyoloji, psikoloji, makine öğrenmesi, biyoinformatik, istatistik, veri madenciliği gibi alanlarda sıklıkla kullanılmaktadır.

Son dönemlerde bulanık mantık kavramının [19] gelişmesiyle beraber temelde iki farklı şekilde kümeleme yapıldığı söylenebilir. Bulanık kümeleme işlemi sonucunda bir veri birden fazla kümenin elemanı olabileceken [20], birebir örtüşen kümeleme işlemi sonucunda bir veri sadece bir kümenin elemanı olabilmektedir. Birebir kümelemede oldukça sık kullanılan iki farklı yöntem ise hiyerarşik kümeleme ve merkez (centroid) tabanlı kümeleme algoritmalarıdır. Hiyerarşik kümelemede kümeler arasında bir hiyerarşi oluşturulabilmekte ve bu sayede ağaç yapısı gibi bir yapı elde edilebilmektedir [21]. Merkez tabanlı kümelemede ise en çok kullanılan yöntem, sunulan tez çalışmasında da kullanılan K-Means kümeleme algoritmasıdır. K-Means kümeleme algoritmasına ait detaylı bilgi Bölüm 2 ve Bölüm 3'de verilmiştir.

## BÖLÜM 2

### 2. LİTERATÜR TARAMASI

#### 2.1. Geçmiş Çalışmalar

Verilen iki obje arasındaki benzerliklerin tespit edilmesi, bilgisayarla görüş uygulamaları içerisinde oldukça önemli bir yere sahiptir. Obje eşleştirme yöntemleri için de, objelerin özniteliklerinin çıkarılması ve özniteliklere göre bire bir veya çoklu eşleştirme yapma tekniği uygulanmaktadır [1]. Karakteristik özniteliklerinin çizge formunda ifade edilip, çizgeler üzerinde işlemlerin gerçekleştirilmesi temel yaklaşımlardan biridir. Fakat verilen bir nesnenin neye benzediğinin tespit edilmesi oldukça yoğun veri kümeleri içerisinde çok ciddi problemlere ve zaman kaybına yol açmaktadır.

[22] ile geliştirilen yöntem alt çizge izomorfizm için kullanılan ilk yöntemlerden biridir. Bu yöntemle çizge eşleştirme işlemi gerçekleştirilmeye çalışılmıştır. Bir başka çalışmada, bu algoritmadan yola çıkarak ağaç geriye dönük arama (tree backtracking) yöntemiyle daha iyi sonuçlar elde edilmesi amaçlanmıştır [23]. Bu çalışmaları sezgisel tabanlı çalışmalar [24], çizge benzetimi teknikleri [25,26], istatistiksel çalışmalar [27], lineer programlama çözümleri [28] ve optimizasyon teknikleri [29] izlemiştir.

Yapılan çalışmalardan biri de verilen çizgeler arasında çoklu eşleştirme yapabilmek için, çizgeyi vektör uzayına geçirerek geometrik taşıma yöntemleriyle verilen iki küme arasındaki benzerliğin bulunmasını amaçlayan çalışmadır [30]. Bu çalışmada, metrik olarak  $\ell_2$  norm (Euclidian Distance) kullanılmış ve düğümlerin Öklit uzayında temsil edilmesi sonucunda bozulma meydana gelmiştir.

Tezimizin de temel kaynağını oluşturan [1]'de ise çizge gömme yöntemi için  $\ell_1$  norm [31] kullanılarak çizgelerin vektör uzayına geçirilmesi amaçlanmıştır. Vektör uzayındaki noktaların benzerliklerinin bulunması için ise Earth Movers Distance (EMD) [32] kullanılmıştır. Bir önceki çalışmadan farklı olarak bu çalışmada girdi ağaçları vektör uzayında izomorfik olarak temsil edilmektedir. Kullanılan Tırtıl Ayırışması [33] kapsamı tez çalışmamızda da kullanılmış olup, detaylı olarak Bölüm 3'de açıklanmıştır.

Daha önce de açıklandığı gibi verilen çizgelerin medyanlarının hesaplanması ise çizge uzayında yapılması oldukça masraflı ve zor bir problemdir. Medyan çizge, verilen bir çizge kümesinden üretilen temsilcidir. Üretilen bu temsilcinin o küme içerisindeki diğer tüm çizgelere olan uzaklıklarının toplamının minimum olması gerekmektedir [16]. Matematiksel işlemlerin kullanılmasıyla verilen bir çizge kümesinin medyanının tespit edilmesi, birçok defa üzerinde çalışılan bir konudur [34,35,36].

Yapılan işlemlerin masraflı olmasından dolayı verilen bir çizgenin vektör uzayına geçirilerek yakınsama işlemi uygulanması son dönemde sıklıkla yapılmaktadır [2]. Graph Edit Distance [1] çalışmasında kullanılarak, çizgenin vektör uzayına geçirilmesi [37] tarafından gerçekleştirilmiştir. Yapılan çalışmalarda temsilci çizgenin çıktı olarak verilmesi amaçlandığı için üçgenleme (Triangulation) yöntemleri kullanılarak bir medyan çizge oluşturma üzerinde durulmuştur [38]. Yapılan bir başka çalışmada [2], verilen çizgelerin tamamının bir bütün olarak değerlendirilip vektör uzayına yeni bir yaklaşımla geçirilmesi amaçlanmıştır. Verilen “n” adet çizgenin tamamının kullanılmasıyla vektör uzayındaki her bir noktanın üzerinde verilen her bir çizgenin etkisinin olması amaçlanmaktadır. Çizge uzayındaki tüm çizgeler için, diğer tüm çizgelere uzaklık değerleri Graph Edit Distance [1] ile belirlenmiş ve bu sayede  $n \times n$  tane veriden oluşan bir matris elde edilmiş olmaktadır. Elde edilen matrisin her bir satırı o çizgenin vektör uzayındaki noktasına karşılık gelecek şekilde yeni yaklaşım planlanmıştır. Bu yöntem [2] ile her bir çizgenin vektör uzayındaki değerinin belirlenmesi hedeflenmiştir. Ancak sunulan çalışmada, düğümlerin öznitelikleri kaybedilerek vektör uzayına geçiş işlemi gerçekleştirilmiş olduğundan, bu yöntemin yeterince iyi sonuç vermeyeceği öngörülmüştür. Sunulan çalışmanın özgün değeri, olabildiğince az bozulma (distortion) ile çizgelerin vektör uzayında ifade edilmesidir. Olası ufak değişiklikler iki objenin birbirine benzerliğinde oldukça büyük sorunlar çıkarabilmektedir. Öznitelik kaybetmemiz sistemimizin vermiş olduğu sonuçların doğruluk oranını oldukça bozmaktadır. Fakat bu araştırma, başka gömme tekniklerinin kullanılmasıyla çizge uzayından vektör uzayına geçtikten sonra çizge medyanı oluşturulabileceğini ve literatüre katkı sağlanabileceğini göstermiştir. Tez çalışmasında, çizge olarak aldığımız verileri gömme teknikleri kullanarak işlem yapmaya daha uygun bir şekle çevirip, yapılması

gereken işlemlerin yapılması ve yeni bir bakış açısıyla beraber vektör uzayında kendi yöntemlerimizi geliştirerek bir temsilci oluşturulabilmek amaçlanmıştır.

Yukarıda belirtilen yöntemlerle elde edilen temsilci ile sorgulanan objenin karşılaştırılması ise nokta kümelerinin benzerliklerinin hesaplanması problemidir. Bu işlem için daha önce yapılan çalışmalarda EMD[32] kullanılmıştır [1]. Sunulan tezde ilk olarak [39] tarafından geliştirilen Hausdorff Mesafesi yöntemi kullanılmıştır. Ancak bu yöntemin dezavantajlarından dolayı üzerinde bir takım değişiklikler yapılmış ve 24 farklı uzaklık hesaplama metotları [3] tarafından sunulmuştur.

## 2.2. Tırtıl Ayrışması

Tırtıl Ayrışması, verilen bir çizgenin yaprakları ile root düğüm arasındaki yollarla çizgenin ifade edilebilmesini sağlayan bir yaklaşımdır [33]. Çizge gömme tekniklerinde daha önce yapılan çalışmalarda Tırtıl Ayrışması kapsamının kullanılmasıyla çizgeler Öklit uzayında temsil edilmiştir. Fakat Öklit uzayının kullanıldığı durumlarda düğümler arasındaki uzaklıklar tam olarak o düğümleri temsil eden vektörler arasındaki uzaklığa eşit olmamaktadır. [40] tarafından yapılan çalışmada çizge uzayından geometrik uzaya geçiş esnasında  $\ell_2$  norm kullanıldığında veri bozulma oranı şöyle olmaktadır:

*T bir ağaç ve  $\ell(T)$  T ağacına ait yaprak sayısı olsun*

$$\phi: V(T) \rightarrow \ell_2^d$$

$$D(\phi) \leq O(\ell^{1/(d-1)}(\min\{\log \ell(T), d\})^{1/2})$$

*d, T ağacı için boyut sayısıdır.*

Yapılan deneysel çalışmalarda da 100 boyutlu bir uzayda  $\ell_2$  norm ile yapılan gömme tekniğinde ortalama bozulma oranı %17 olmaktadır [30].

$\ell_2$  normdaki bozulmaların bu oranda oluşmasından dolayı gömme tekniklerinin izomorfik olarak çalışması için farklı normlar aranmıştır.  $\ell_1$  normu ağaç yapıları için gerçekleştirilecek bu tür bir izomorfik gömme oluşmasını sağlamaktadır.  $\ell_1$  norm matematiksel olarak şu şekilde ifade edilmektedir:

$$U_1(X, Y) = \sum_{k=1}^d |X_k - Y_k|$$

*U: iki nokta kümesi arasındaki mesafe*

*X: nokta kümesi  $[x_1, x_2, x_3..x_d]$*

*Y: nokta kümesi  $[y_1, y_2, y_3..y_d]$*

*d: noktanın vektörel boyutu*

Çizge gömme tekniği olarak  $\ell_1$  norm kullanılması ve vektör uzayına geçiş aşamasında Tırtıl Ayırışması kullanılması [1] çalışmasında yapılmıştır. Tırtıl Ayırışmasının detayları Bölüm 3.3.1’de açıklanmıştır.

### **2.3. K-Means Kümeleme Algoritması**

K-Means Kümeleme Algoritması, diğer kümeleme algoritmalarında olduğu gibi, verilen küme içerisindeki noktaların birbirlerine benzerliklerine göre sınıflandırılmasını sağlamak amacıyla kullanılır. Kümeleme işlemi sonucunda elde edilen kümelerde merkez noktanın, tüm küme elemanlarına mesafeleri toplamının minimum olması amaçlanmaktadır. Bu şekilde bir küme merkezi oluşturulması, bir nevi kümenin ağırlık merkezini vermektedir. Biz de tezimizde K-Means algoritmasını vektör uzayımızda bulunan noktaların ağırlık merkezlerini bularak olası temsilci nokta kümemizde bulunması gerektiğini düşündüğümüz noktaları tespit etmek için kullanılmıştır. Bu algoritmanın detayları Bölüm 3.3.3’de verilmiştir.

### **2.4. Nokta Kümesi Eşleme Algoritmaları**

Obje eşleştirme algoritmalarında alınan objelerin işlemlerden geçirilerek nokta kümesi haline getirilmesi sıklıkla kullanılmaktadır. Bu durumda elde edilen iki farklı nokta kümesinin birbirlerine olan benzerliklerinin veya başka bir deyişle birbirlerine olan uzaklıklarının nasıl hesaplanacağı, üzerinde çalışılan ve çalışılması gereken konulardan biridir.

Daha önce yapılan çalışmalarda [1] EMD [32] metriği kullanılmıştır. Bu metriğin temelinde iki nokta kümesinin birbirine nasıl döndürülebileceği hesaplanmaktadır. Bunun için de Ağ Akış (Network Flow) algoritmalarından biri kullanılmaktadır. Bu algoritma, oldukça uzun sürede sonuç vermesi ve çok fazla işlem yapması nedeniyle



masraflı bir yöntem olarak görülmektedir. Bir başka çalışmada ise daha matematiksel ve kolay sonuca ulaşılabilecek bir algoritma olan Hausdorff Mesafesi [39] kullanılmıştır. Fakat çalışmada da karşılaşılan problemlerden biri yanlış sonuçların fazlalığı olmuştur. Olası gürültülerden dolayı bazı durumlarda beklenen sonuçtan çok uzak çözümler elde edilmiştir. Yapılan çalışmalarla beraber bu problemin giderilmesine çalışılmış ve Hausdorff Mesafesi algoritması [3] tarafından yeniden düzenlenmiştir. Bu sayede hassasiyetin biraz daha azaltılması sağlanmıştır.

Hausdorff Mesafesi iki nokta kümesini alarak bu iki küme arasındaki mesafenin nasıl belirlenebileceğini belirtmiştir.

$$A = \{a_1, a_2, a_3, \dots, a_n\}$$

$$B = \{b_1, b_2, b_3, \dots, b_m\}$$

$$d(a, b) = |a - b|$$

$$d(a, B) = \forall b \in B \mid \min |a - b|$$

Verilen bu ana denklemleri kullanarak Hausdorff Mesafesi toplam 24 farklı şekilde sonuç verebilecek durumda [3]de yeniden düzenlenmiştir.

$$d_1(A, B) = \min_{a \in A} d(a, B)$$

$$d_2(A, B) = {}^{50}K_{a \in A}^{th} d(a, B)$$

$$d_3(A, B) = {}^{75}K_{a \in A}^{th} d(a, B)$$

$$d_4(A, B) = {}^{90}K_{a \in A}^{th} d(a, B)$$

$$d_5(A, B) = \max_{a \in A} d(a, B)$$

$$d_6(A, B) = \frac{1}{N_a} \sum_{a \in A} d(a, B)$$

Hausdorff Mesafesi temel yapısı nedeniyle  $d(A, B) \neq d(B, A)$  dır. Bu nedenle iki nokta kümesi arasındaki mesafenin tespit edilmesinde yukarıda verilen uzaklık metriklerinin yanı sıra aşağıda belirtilen fonksiyonel farklılıklar da hesaplanabilmektedir.

$$f_1(d(A, B), d(B, A)) = \min(d(A, B), d(B, A))$$

$$f_2(d(A, B), d(B, A)) = \max(d(A, B), d(B, A))$$

$$f_3(d(A, B), d(B, A)) = \frac{d(A, B) + d(B, A)}{2}$$

$$f_4(d(A, B), d(B, A)) = \frac{N_a d(A, B) + N_b d(B, A)}{N_a + N_b}$$

Bu sayede 6 farklı uzaklık metriğine ait 4 farklı fonksiyonla toplam 24 farklı Hausdorff Mesafesi uzaklığı kullanılabilir.

Çalışmamız kapsamında bu uzaklık hesaplamalarına ek olarak  $f_5(d(A, B), d(B, A)) = d(A, B)$  fonksiyonu da uzaklık metriklerine dahil edilmiştir. Ancak deneysel çalışmalar sonucunda  $d_1$  uzaklığının kullanılmamasına karar verilmiştir. Bu nedenle toplam 25 farklı uzaklık hesaplanmasıyla sonuçlar üretilmiştir.

## BÖLÜM 3

### 3. YAPILAN İŞLEMLER

#### 3.1. Giriş

Tez çalışmasında temel motivasyon, obje eşleştirme algoritmalarına yeni bir teknik geliştirerek, geliştirilen tekniğin uygulanmasıyla daha hızlı ve en az seviyede hatayla sonuç üretebilmektir. Daha önce yapılan çalışmaların da incelenip analiz edilmesiyle, birkaç çalışmanın harmanlanması ve yeni bir bakış açısıyla probleme yaklaşım elde edilmesi muhtemel sonuçların incelenmesi amaçlanmıştır.

Üzerinde çalışılan problem, verilen bir objenin, veri kümesi içerisinde bulunan veri sınıflarından hangisine dahil olduğunun bulunmasıdır. Bu işlemi yaparken en temel sistem, verilen objenin veri kümesi içerisinde bulunan tüm verilerle karşılaştırılmasıdır. Tahmin edileceği üzere bu yöntem, veri kümesinin büyüklüğünden dolayı oldukça yavaş ve masraflı bir yaklaşımdır. Bu masrafın önüne geçebilmek için veri sınıflarının her birinin temsilcisinin belirlenip, sadece bu temsilciler üzerinden karşılaştırma işlemlerinin yapılması öngörülmüştür. Objelerin işlem yapılabilir hale getirilmesi, temsilcilerin belirlenmesi, uzaklık metriklerinin seçilmesi gibi problemlere çeşitli çözümler üreterek, zengin ve yeni bir yaklaşım geliştirilmesi amaçlanmıştır. Elde edilen sonuçlar, karşılaştırma işlemi için tüm veri sınıflarının kullanılmasıyla karşılaştırılmıştır.

#### 3.2. İşlem Adımları

Sistemin iki temel parçası bulunmaktadır. Birincisi offline olarak çalıştırılabilen ve mevcut veri sınıflarına bağlı olarak değişebilecek temsilci seçilmesi işlemi ve ikincisi de karşılaştırılacak verinin verildiği ve bu objenin hangi sınıfa ait olduğu bilgisinin sonuç olarak üretildiği eş zamanlı çalışacak kısımdır.

##### 3.2.1. Medyanların Çıkarılması

Medyan hesaplanması aşağıdaki adımlardan oluşur:

1. Üzerinde işlem yapılacak olan sınıfın belirlenmesi

2. Sınıftaki tüm objeler için
  - a. Objenin iskelet yöntemiyle iskeletinin çıkarılması
  - b. Tırtıl Ayrışması yöntemiyle objenin vektör uzayına geçirilmesi
3. Sınıftaki tüm objeler için en büyük boyut sayısının hesaplanması
4. Sıfır Ekleme ile tüm noktaları aynı boyuta getirmek
5. Düğüm Ekleme ile tüm çizgeleri dengeli bir şekilde vektör uzayına eşit nokta ile taşımak
6. K-Means ile vektör uzayında medyan hesaplanması
7. Verilen veri sınıfı için medyan nokta kümesinin saklanması

Deney kümesinde bulunan toplam 9 veri sınıfı için medyan hesaplama işlemi gerçekleştirilmiştir. Her bir veri sınıfı içerisindeki çizgeler sırayla vektör uzayında temsil edilecek olan nokta kümesi haline getirilmektedir. Böylelikle vektör uzayında bulunan her bir nokta aslında çizge uzayındaki bir düğüme karşılık gelmiş olmaktadır ve sonuçta elde edilen nokta kümesi de çizge medyanını temsil etmiş olmaktadır.

Tez kapsamında iskeletleme algoritması üzerine herhangi bir çalışma yapılmamıştır. Objelerin orijinal hallerinden iskeletleme algoritması uygulanmış hale geçişi daha önce yapılan çalışmalarda kullanıldığı şekliyle yapılmaktadır ve üzerinde çalışılan deney kümesi, çizgelerden oluşan bir küme olarak alınmaktadır. İskeletleme algoritmasının en önemli noktalarından biri objeye göre çıkartılan her düğümün aynı ağırlık değerine sahip olmamasıdır. Bazı düğümler daha çok karakteristik öznelik bulundururken bazı düğümler göz ardı edilebilecek kadar öneme sahip olabilmektedir. İskeletleme algoritması her düğüm için bir ağırlık verisi de oluşturarak resmin ana hatlarında olması gereken düğümlere daha büyük ağırlıklar vermektedir.

İskeletleme işlemi neticesinde veri kümesi düğümlerin ağırlığa sahip olduğu çizgelerden oluşmaktadır. Veri kümesi içerisindeki tüm çizgelere benzer bir temsilci oluşturabilmek, çizge medyanı oluşturmak hedeflenmektedir. Çizge medyanının, veri sınıfı içerisinde bulunan tüm çizgelere olabildiğince eşit uzaklıkta ve tamamını temsil edebilecek bir çizge olması gerekmektedir. Bir başka ifadeyle temsilci çizgenin tüm çizgelere uzaklıkları toplamı en küçük olmalıdır. Bu işlemlerin gerçekleştirilmesi çizge uzayında oldukça masraflı olmaktadır. Bu nedenle çizgeler öncelikle vektör uzayına çizge gömme teknikleriyle gömülecek ve daha sonra vektör uzayında işlemler yapılacaktır.

Çizgelerin vektör uzayına gömülmesi işleminden sonra artık üzerinde işlem yapılan veri kümesi “a” boyutlu uzayda yer alan noktalar kümesi olacaktır. Tüm noktalar kümesinden seçilen temsilci kümesinin ise tüm noktaların az da olsa bir etkisinin görüldüğü küme olmasına dikkat edilecektir. Bu sayede tüm noktaları temsil edebilecek bir nokta kümesi oluşturulacaktır.

Bu problemin çözümü için kümeleme algoritmalarından K-Means algoritması kullanılmıştır. Bu algoritmanın seçilmesindeki temel düşünce, özniteliklerin tamamını içerebilen temsilci noktaların seçilmesidir. Aslında her bir düğüm için bir temsilci düğüm seçilmiş olacaktır. Birbirine özellik olarak yakın olan “m” adet düğüm bu işlem sonucunda bir düğüm olarak ifade edilmiş olacaktır. Bu düğümlerin tamamının oluşturacağı küme de çizgelerin tamamının temsilcisi bir çizge vermiş olacaktır.

Bu aşamada dikkat edilmesi gereken nokta her bir düğümün belirli bir ağırlığının olmasıdır. Her bir düğümün ağırlığı o düğümün çizge medyanında hangi oranda bulunması gerektiğinin bir göstergesi olmaktadır. Vektör uzayına geçiş esnasında düğümlerin ağırlıklarını kaybetmemek adına bir işlem uygulanmamaktadır. Düğüm ağırlıklarının kaybedilmesi halinde oluşturulacak temsilci çizge gürültülerden etkilenmiş olacaktır. Bunun engellenmesi için çizge uzayından vektör uzayına geçerken her bir düğümün ağırlığı ölçüsünde vektör uzayına nokta eklenmesi sağlanmıştır. Başka bir ifadeyle her bir düğüm vektör uzayında bir nokta ile ifade edilmekte fakat vektör uzayında belirtilen noktada o düğümün ağırlığı kadar nokta bulunmaktadır. Bu sayede K-Means algoritması uygulanırken oluşturulacak merkezlerin, çizge medyanında bulunması gerektiğini düşündüğümüz noktalara daha yakın olması sağlanacaktır.

Benzer bir sebepten dolayı tüm çizgelerdeki düğümlerin aynı sayıda olması sağlanmaya çalışılmaktadır. Aksi takdirde fazla düğüme sahip olan çizgeler, çizge medyanını manipüle ederek ortaya çıkacak olan sonucun kendisine daha yakın bir sonuca dönüşmesini sağlayabilecektir. Bu problemin önüne geçebilmek için tüm çizgelerin aynı düğüm sayısında olması ve bu sayede medyan hesaplanırken tüm çizgelerin eşit şartlarda temsil edilmesi sağlanmaya çalışılmaktadır. Tüm çizgeleri aynı düğüm sayısına çekebilmek için bazı çizgelere boş düğümler eklenmesi gerekmektedir. Bu işlemin gerçekleştirilmesi aşamasında da dikkat edilmesi gereken nokta eklenen yeni düğümlerin çizgenin ağırlık dengesini bozması, hatta çizgenin

temel özelliklerine olabildiğince az şekilde etki ediyor olmasıdır. Bu nedenle eklenen yeni düğümler mevcut düğümlerin üstüne eklenmiş gibi düşünülmektedir. Mevcut düğümün tüm ağacın ağırlığına oranı kadar eksik düğüm, belirtilen düğümün üzerine eklenmektedir. Başka ve daha doğru bir ifadeyle, yeni düğüm eklemek yerine düğümlerin ağırlıkları belirli oranda yukarı çıkarılarak sanki yeni düğüm eklenmiş gibi davranılmaktadır.

K-Means algoritmasının üretmiş olacağı çıktı verilen veri sınıfı için temsilci çizgenin vektör uzayındaki görüntüsü olacaktır. Hangi veri kümesine ait olduğu bulunmak istenen objenin, vektör uzayına geçişinin yapılması ve nokta kümesi karşılaştırma işlemleri yordamıyla çizge medyanını temsil eden nokta kümesi ile karşılaştırılması verilen objenin veri kümesine ne kadar benzer olduğunu gösterecektir.

### 3.2.2. Karşılaştırma İşlemi

Karşılaştırma işlemi aşağıdaki adımlardan oluşmaktadır

1. Tahmin edilmesi istenen verinin alınması
2. Verinin Tırtıl Ayrışması ile vektör uzayına geçirilmesi
3. Veri kümesinde bulunan tüm medyanlar için
  - a. Veri ile medyan arasındaki boyut farkının Sıfır Ekleme ile giderilmesi
  - b. Verinin Medyan ile karşılaştırılmasının Hausdorff Mesafesi yöntemiyle yapılması
    - i. 25 farklı Hausdorff Mesafesi metriği ile karşılaştırmanın yapılması
    - ii. Tüm verilerin kaydedilmesi
  - c. İki farklı yöntemle algoritmanın başarısının tespit edilmesi
    - i. Yöntem 1: Her medyan için o sınıftaki tüm doğruların ortalamasının alınması
    - ii. Yöntem 2: Her medyan için o sınıfta tüm uzaklık metriklerinden en fazla doğru sonuç verenin alınması

Senaryo gereği sorgu için verilen objenin veri sınıflarında hangisine ait olabileceğinin tahmininin yapılması gerekmektedir. Karşılaştırma işlemi iki objenin de vektör uzayında noktalar kümesi olarak temsil edilmesi ve bu iki nokta kümesinin birbiriyle karşılaştırılarak aradaki mesafenin bulunması yöntemiyle yapılmaktadır.

Dolayısıyla verilen yeni objenin ilk olarak çizge uzayına gömülmesi gerekmektedir. Karşılaştırma işleminde simetri olmadığı için ve objenin medyana ne kadar benzediği bulunmak istendiğinden karşılaştırma işlemi yeni objenin, medyanlarla karşılaştırılması şeklinde olmaktadır. Vektör uzayında nokta kümesi olarak tespit edilen yeni obje sırasıyla tüm medyanlar ile karşılaştırılmakta ve en yakın olduğu medyan sınıfına ait olduğu tahmin edilmektedir.

İki nokta kümesi arasındaki mesafenin tespit edilebilmesi için Hausdorff Mesafesi kullanılmaktadır. Hausdorff Mesafesinin yenilenmiş toplam 25 farklı versiyonu daha önce yapılan çalışmalarda geliştirilmiştir. Mevcut versiyonların tamamı için karşılaştırma işlemi tez çalışmasında gerçekleştirilmiştir. Hausdorff Mesafesinin hesaplanması sırasında iki nokta arasındaki mesafenin hesaplanması ise Manhattan Mesafesi yöntemi kullanılarak yapılmaktadır.

Deney senaryosu için veri kümesinde bulunan ve gerçekte hangi sınıfa ait olduğunu bildiğimiz objeler kullanılmıştır. Dolayısıyla her obje için elimizde toplam 25 farklı sonuç bulunmaktadır. Doğruluk oranı hesaplanırken bu 25 farklı sonucun kullanıldığı iki farklı yöntem bulunmaktadır. Birinci yöntem, her mesafe ölçüm yöntemi için başarı oranı her bir yöntemde bulunan doğru oranı ve bunların tamamının ortalamasıdır. İkinci yöntem ise hibrit yöntemdir. Tüm yöntemler içerisinde en fazla tespit edilen sınıf verilen objenin sınıfı olarak belirlenir ve doğruluk oranı hesaplanır. Elde edilen sonuçlar ve sonuçların değerlendirilmesi Bölüm 4 ve Bölüm 5’de aktarılmıştır

### **3.3. İşlemlerin Detayları**

Bu bölümde yapılan işlemlerin detayları hakkında bilgi verilecektir. Hangi tekniğin neden seçildiği, nasıl kodlandığı, eksiklikleri ve fazlalıkları konularında aktarım yapılacaktır.

#### **3.3.1. Tırtıl Ayrışması**

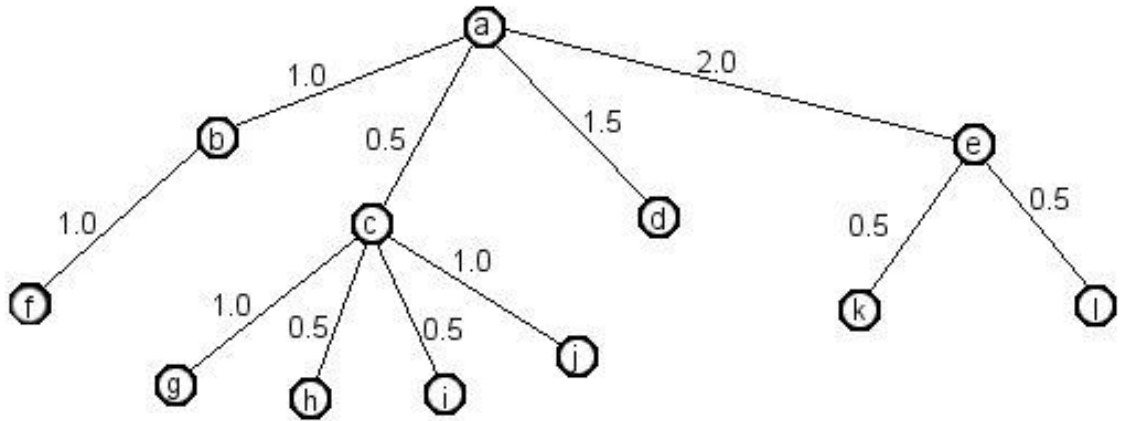
Daha önce yapılan çalışmalarda [2] elde edilen vektörlerin hiç biri düğümlerin hiçbir özelliğini göz önünde bulundurmamaktadır. Çünkü temel motivasyon tüm çizgelerin kendi aralarındaki ilişkileri üzerine kurulmuştur. Ancak vektör uzayına geçiş esnasında çok ciddi bir kayıp söz konusudur. Tırtıl Ayrışması yöntemiyle çizge

üzerinde her bir düğümün vektör uzayında temsil edilmesi ve bu esnada kenarların ağırlıklarının da kaybedilmediği bir vektör elde edilmesi sağlanmıştır. Bu dönüşümün en büyük avantajı minimum veri kaybıdır.

### İşlem adımları

1. Her bir düğüm için “root” düğüme olan uzaklıklar tespit edilir
2. Ağaç içerisindeki tüm “yapraklar“ tespit edilir
3. Yapraklar kendi aralarında roota en uzak olandan en yakın olana doğru sıralanır
4. En uzak yapraktan başlayarak
  - a. “root” düğüme kadar geri dönüş yapılır ve yollar (path) belirlenir
  - b. Kesişen yol varsa sadece kesiştiği yere kadar yeni yol numarası verilir
5. En uzak yapraktan başlayarak özyinelemeli olarak
  - a. “root” düğüme kadar geri dönüş yapılır
  - b. Ebeveyn düğüm ile ilişkili olarak yol numarasına göre boyutta değişiklik yapılır
6. Elde edilen vektörler her bir düğüm için geometrik uzayda ifade biçimidir

Tırtıl Ayrışması örneği Şekil 3.1’de gösterilmiştir.



Şekil 3.1Tırtıl Ayrışması Örneği

Verilen ağaç yapısında “a” düğümü root düğüm, “f,g,h,i,j,d,k,l” düğümleri yaprak düğümlerdir. Her bir kenar için bir ağırlık değeri verilmiştir. Bu yapı, obje üzerinden iskeletleme yöntemi kullanılarak minimum kapsama ağacının tespit edilmesiyle ortaya çıkarılmıştır.



Tırtıl Ayrışması yönteminde yapraklardan root düğüme olan uzaklıklar göz önünde bulundurularak işlemler gerçekleştirilmektedir. İşlem sonucunda toplam yaprak düğüm sayısı kadar yol tespit edilmiş olacaktır. Yollar için ise derece kavramı mevcuttur. Örneğin Şekil 3.1 için “f, g, d, k” düğümleri direk olarak root düğüme ulaşabilmektedir. Bu yollar “1.düzyer” yollar olarak tabir edilmektedir. Bunun haricinde “h, i, j” düğümlerinden root düğüme ulaşabilmek için “g” düğümünden root düğüme giden yol üzerinden geçilmesi gerekmektedir. Bu nedenle bu yollara “2.düzyer” yollar denir. Aynı durum “l” düğümü için de geçerlidir. “l” düğümünden root düğüme ulaşabilmek için “k” düğümünden root düğüme ulaşan yolun kullanılması gerekmektedir. Aslında bu problem ayrık kümelerin birleştirilmesi olarak tarif edilebilmektedir.

Tırtıl Ayrışması yöntemiyle vektör uzayına geçiş esnasındaki temel motivasyonumuz, boyut kavramının yol sayısı ile ifade edilebiliyor olmasıdır. Bu sayede, toplam yol sayısı üzerinde işlem yapılan ağacın geometrik uzayda kaç boyutlu olacağı bilgisi elde edilmektedir. Çizge üzerindeki her bir düğüm vektör uzayında bir noktaya karşılık gelecek ve hangi noktaya karşılık geldiğini hangi yolun üzerinde bulunduğu belirtecektir. Verilen bir “v” düğümünün vektör uzayına geçişi esnasında gerçekleştirilen işlemler şu şekildedir:

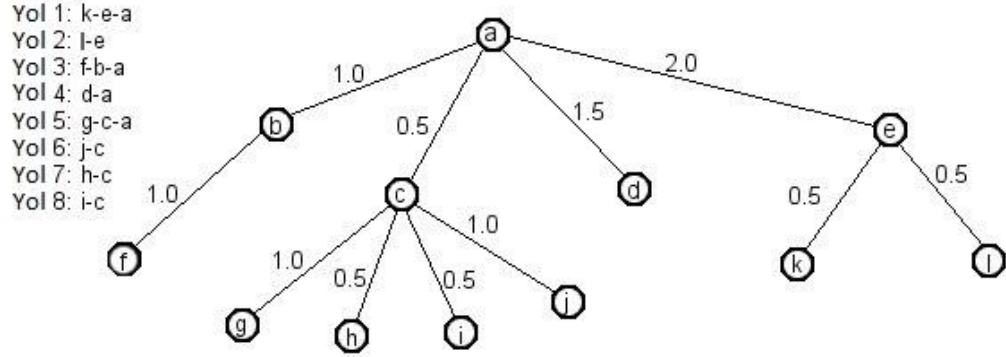
1. “v” ile root düğüm arasında tekil olacak  $P(v)$  yolu belirlenir
2.  $P(v)$  yolu kendi içerisinde başka yolların birleşimi olarak belirtilebilir.

Örneğin

$P(v) = \{v, t_1, t_2, \dots, t_n, r\}$  olsun.  $P(v)$  yolunun bir kısmı  $p^1$  yolu üzerinde, bir kısmı  $p^2$  ve bir kısmı da  $p^a$  yolu üzerinde bulunabilir. Örneğin Şekil 3.1 de bulunan “j” düğümü “2.düzyer” bir düğümdür ve “1.düzyer” yol ile kesişmektedir. Bu durumda  $P(j)$  içerisinde birden fazla yol ile ilgili bilgi içermektedir. Dolayısıyla  $P(v)$  yolu aslında birçok yolun birleşimi olarak belirtilmelidir.

3. Her bir yolun bir  $l$  ağırlığı olsun. Bu durum  $P(v)$ ,  $\{p^1, p^2, \dots, p^a\}$  decomposition kümesi ve bu kümedeki her bir yolun ağırlığına denk gelen  $\{l^1, l^2, \dots, l^a\}$  ağırlık kümesinin birleşimi olarak ifade edilebilir olacaktır.

4. Düğümün her bir yol üzerinden geçiş ağırlığı bize  $a$  boyutlu bir vektör verecektir. Bu sayede izometrik olarak bir düğümün vektör uzayına geçişi sağlanmış olacaktır.



Şekil 3.2 Tırtıl Ayrışması Örneği - Yollar

Şekil 3.2, Şekil 3.1’de verilen çizgenin yollarının çıkarılmış halini gösterilmektedir. yol 1, yol 3, yol 4 ve yol 5 yolları “1.düzyer” yollardır. Örneğin “h” düğümü için  $P(h) = Yol(h, c) + Yol(c, a)$  dır ve hem Yol 7 hem de Yol 5 üzerinde yer aldığı görülmektedir. Yol 7 üzerindeki ağırlığı sadece (h,c) yolundan gelen 0.5 ve Yol 5 üzerindeki ağırlığı da (c,a) yolundan gelen 0.5 ağırlığıdır. Diğer yollar üzerinde ise herhangi bir ağırlığı bulunmamaktadır. Bu durumda “h” düğümü için üretilecek olan noktanın vektörel gösterimi  $[0,0,0,0,0.5,0,0.5,0]$  olacaktır. Tüm düğümler için vektör uzayında temsil edilecek nokta kümesi Çizelge 3.1 de verilmiştir.

Çizelge 3.1 Tırtıl Ayrışması Yol Çizelgesi

	Yol 1	Yol 2	Yol 3	Yol 4	Yol 5	Yol 6	Yol 7	Yol 8
<b>A</b>	0	0	0	0	0	0	0	0
<b>B</b>	0	0	<b>1.0</b>	0	0	0	0	0
<b>C</b>	0	0	0	0	<b>0.5</b>	0	0	0
<b>D</b>	0	0	0	<b>1.0</b>	0	0	0	0
<b>e</b>	<b>2.0</b>	0	0	0	0	0	0	0
<b>f</b>	0	0	<b>2.0</b>	0	0	0	0	0
<b>g</b>	0	0	0	0	<b>1.5</b>	0	0	0
<b>h</b>	0	0	0	0	<b>0.5</b>	0	<b>0.5</b>	0
<b>i</b>	0	0	0	0	<b>0.5</b>	0	0	<b>0.5</b>
<b>j</b>	0	0	0	0	<b>0.5</b>	<b>0.5</b>	0	0
<b>k</b>	<b>2.5</b>	0	0	0	0	0	0	0
<b>l</b>	<b>2.0</b>	<b>0.5</b>	0	0	0	0	0	0

Çizelge 3.1’den görüldüğü üzere Tırtıl Ayrışması sonucu tekil bir sonuç değildir. Hangi yaprak sırasıyla ayrışmanın yapılması sonuçları da değiştirmiş olacaktır. Bunun önüne geçebilmek için yapraklar arasındaki yol uzunluğunun en büyükten en küçüğe doğru sıralanması ve yol numaralandırmasının bu sırayla yapılması düşünülmüştür. Dikkat edilmesi gereken bir diğer konu ise bu yöntem sayesinde  $\ell_1$  norm için tamamen izometrik ve veri kaybı olmadan çizge gömme tekniği uygulanmış olmaktadır.

Görüldüğü üzere her bir küme içinden çıkarılacak her bir çizgenin vektör uzayındaki boyutları farklı olabilecektir. Vektör uzayındaki boyut tamamen objeden gelen minimum kapsama ağacının yaprak sayısı ile bağıntılıdır. Bu nedenle elde edilen tüm noktaların aynı düzleme taşınması gerekecektir.

### 3.3.2. Ekleme(Padding) İşlemleri

Tez çalışmamız içerisinde iki farklı yerde ekleme işlemi gerçekleştirmekteyiz (Sıfır Ekleme ve Düğüm Ekleme). Tırtıl Ayırışması neticesinde vektör uzayımızda elde ettiğimiz noktalardan farklı ağaçta olanlardan her biri farklı boyutta olacaktır. Çünkü ağaç yapısına göre root düğümden en uzakta olan yaprak düğüm arasındaki mesafe her zaman aynı olmayabilecektir. Bu durumda tüm noktalarla beraber işlem yapmamız gerektiği için tüm noktalar en büyük boyuta çekilmektedir. Bir diğer yöntem ise büyük boyutlu noktaların küçültülmesidir. Fakat bu yöntem kullanıldığı takdirde veriler üzerinde bozulma olacaktır. Bu nedenle daha küçük bozulmaya sebep veren, ancak kabul gören bir teknik olan boş boyutlara sıfır eklenmesi yöntemini kullanılmıştır.

Bir diğer durum ise tüm çizgelerin aynı sayıda düğüme sahip olmamasıdır. Bu ilk bakışta önemli bir problem değil gibi görülse de, tüm çizgelerimizi aynı vektör uzayı içerisinde ele aldığımız için bazı çizgelerin yüksek düğüm sayıları nedeniyle medyanı kendilerine benzetebileceği öngörülmüştür. Bu nedenle tüm düğümlerin sayıları eşitlenmiştir. Vektör uzayında nokta ifade edilirken her bir düğümün ağırlığı oranınca nokta olarak temsil edilmesinden bahsedilmişti. Başka bir deyişle Düğüm Ekleme işlemi tüm ağırlıkların ağacın ağırlık merkezi değişmeyecek şekilde yukarı tamamlanması manasına gelmektedir. Yeni düğüm eklenirken dikkat edilmesi gereken esas nokta ağacın dengesinin bozulmamasıdır.

### 3.3.3. K-Means Kümeleme

Daha önce yapılan çalışmalardan farklı olarak çizgelerin medyanlarının hesaplanması aşamasında kümeleme algoritması olan K-Means Algoritması kullanıldı. Vektör uzayına eklediğimiz her bir nokta bir düğümü ifade etmektedir. Ayrıca vektör uzayında elde ettiğimiz noktaların her biri kendi üzerinde kenar ağırlıklarını taşımaktadır. Bunun yanı sıra düğüm ağırlıklarını da kaybetmemek için düğüm ağırlığı kadar nokta vektör uzayına eklenmiştir.

#### K-Means adımları

- Adım 1: Merkez sayısı belirlenir
- Adım 2: Başlangıç merkezleri belirlenir

- Adım 3: Tüm noktalar için en yakın olduğu merkez hesaplanır ve nokta o merkezin noktası olarak belirlenir
- Adım 4: Tüm merkezlere ait noktaların orta noktası hesaplanır ve o kümenin yeni merkezi hesaplanan yeni nokta yapılır
- Adım 5: Tüm merkezlerin ne kadar değiştiği hesaplanır
- Adım 6: Toplam değişim mesafesi belirli bir eşik noktasının altında değilse Adım 3'e dönlür. Altındaysa Adım 7'ye geçilir.
- Adım 7: Merkezler yazdırılır

K-Means uygulamaları esnasında dikkat edilmesi gereken ve sonucu etkileyen bazı noktalar şunlardır:

- Merkez Sayısı: Problem türüne göre dinamik olarak belirlenebilir veya belirli sabit bir sayı verilebilir.[41]
- Başlangıç Merkezleri: Başlangıç merkezleri bazı durumlarda sonucu değiştirecek kadar etkiyebilir. Rastgele veya belirli bir sıraya göre başlangıç merkezi seçilebilir.
- İki Nokta Arası Mesafe: Herhangi iki nokta arasındaki mesafenin belirlenmesi(merkez-nokta, nokta-nokta, merkez-merkez) için birden fazla uzaklık metriği bulunmaktadır. Hangi metriğin kullanıldığına göre algoritma sonucunda değişiklikler olabilmektedir.
- İterasyon İçin Bitirme Noktası: Sistemin adımları ne zamana kadar devam ettirmesi gerektiğini belirleyen koşuldur. Problemin türüne ve amacına göre değişiklik gösterebilir. Merkezlerin değişmesi, noktaların küme değiştirmesi gibi durumlar göz önünde bulundurulabileceği gibi sabit bir sayı da verilebilir. Eğer değişken yapılacaksa seçilecek olan eşik değeri önemli bir noktadır.

Tez çalışmasında, vektör uzayında bulunan sınıfa ait tüm elemanların en benzer temsilcisinin, tüm çizgelerin medyanını vereceği öngörülmüştür. Çünkü K-Means algoritması sonucunda ortaya çıkacak olan merkezler, kendisine yakın olan noktaların temel karakteristik özelliklerini sağlıyor olacaktır. Burada merkez sayısı, başlangıç noktaları olmak üzere iki önemli nokta vardır. Merkez sayısı, tüm çizgelere düğüm eklenmesiyle aynı seviyeye çıkarıldığı için düğüm sayısı olarak

belirlenmiştir. Başlangıç noktası olarak hangi noktaların seçileceği ise rastgele bir şekilde ilk obje olarak seçilmiştir.

K-Means algoritmasında bir diğer önemli parametre ise hangi uzaklık metriğinin kullanılacağıdır. Noktaların hangi merkeze dahil edileceği ve merkezlerin ortalama noktalarının bulunması esnasında iki nokta arasındaki mesafe hesaplanmalıdır. Daha önce de değinildiği gibi  $\ell_2$  norm kullanıldığı durumlarda distortion fazlaca olduğu için tüm bu metriklerimizde Manhattan Mesafesi olarak da bilinen  $\ell_1$  norm kullanılmıştır.

K-Means esnasında kaç tekrarlar (iterasyon) yapılacağı hakkında kesin bir bilgi vermek zaman zaman yanlış sonuçlar elde edilmesine neden olmaktadır. Tezimizde her adımda değişen merkezlerin kaç birim değiştiğinin hesaplanması ve bu değişim miktarına göre devam edilip edilmeyeceğine karar verilmesi öngörülmüştür. Bu nedenle belirlenen değişim eşik değeri “200” birim değerinden daha düşük değişim olduğu zaman sistem yeni merkez hesaplama işlemini sonlandırmaktadır.

1. Vektör uzayındaki objelerin her bir düğümünün ağırlıkları değerince nokta olarak K-Means sistemine eklenmesi
2. Merkez sayısının verilmesi (En büyük düğüm sayısı)
3. Başlangıç noktalarının verilmesi (İlk dosyanın noktaları)
4. Tüm noktaların hangi merkeze bağlı olduğunun belirlenmesi
  - a. İki nokta arasındaki mesafenin belirlenmesi  $\ell_1$  norm
5. Yeni merkez noktalarının hesaplanması
  - a. İki nokta arasındaki mesafenin belirlenmesi  $\ell_1$  norm
6. Merkezlerin değişimleri toplamı “200” eşik değerinden düşükse Madde 7, değilse Madde 1
7. Merkezlerin belirlenmesi

### 3.3.4. Hausdorff Mesafesi

Hausdorff Mesafesi iki nokta kümesinin birbirlerine olan benzerliklerinin tespiti için kullanılan bir metriktir. [3] de de verildiği gibi toplamda 24 farklı yeniden düzenlenmiş versiyonu bulunmaktadır.

Sistem girdi olarak iki nokta kümesini (S1, S2) almaktadır. Bölüm 2.4’de belirtilen yöntemin uygulanmasıyla iki nokta kümesi arasındaki benzerlik oranı

belirlenmektedir. Karşılaştırma sürecinde yani testlerin yapılmasında son aşama olarak bu metrik kullanılmıştır. Tüm yeni düzenlenmiş versiyonların sonuca etkisi belirlenmiştir. Ancak bu tezdeki ana motivasyonumuz hangi düzenlenmiş versiyonun daha faydalı olabileceğinin belirlenmesi değildir. Bu nedenle Hausdorff Mesafesi tez kapsamında uzaklık hesaplama metriği olarak kullanılmıştır. Hangi versiyonun neden daha faydalı olmuş olabileceği, objelerin hangi karakteristik yapılarının kullanılması gereken versiyonu nasıl belirleyebileceği gibi konular üzerinde çalışılmamıştır. Karşılaştırma işleminin yapılması sırasında sorguyla verilen çizgenin tüm medyanlarla karşılaştırılıp, sorgu çizgesinin hangi veri sınıfına ait olması gerektiğinin tespitinde, hibrit (Hybrid) yapı kullanılarak olası hassas sonuçların ortadan kaldırılması amaçlanmıştır.

## BÖLÜM 4

### 4. DENEYSEL SONUÇLAR

#### 4.1. Amaç

Sistemin altyapısı kurulduktan sonra en temel problemlerden biri yeni geliştirilen yaklaşımın başarısının nasıl ölçülebileceğidir. Tez çalışmasının amacı doğrultusunda, verilen obje kümelerinin K-Means algoritması kullanılarak medyanlarının hesaplanmasına, sorgu çizgelerinin sadece bu medyanlarla karşılaştırılmasının hangi kazançları ve kayıpları beraberinde getirdiğinin tespit edilmesine çalışılmıştır.

Yeni geliştirilen tekniğin temelde iki ana bölümü mevcuttur:

1. Medyanların çizge gömme teknikleri ve K-Means algoritmasıyla hesaplanması
2. Hesaplanan medyanların başarı oranının tespit edilmesi

Yapılan testler ve sonuçlar bu iki ana başlık altında sunulacaktır. Medyanların hesaplanma zorlukları ve geçilen aşamalar bu bölümde anlatılmıştır. Hesaplanan medyanların başarı oranının belirlenmesi için Hausdorff Mesafesi yönteminin yenilenen 25 versiyonu kullanılmıştır.

Başarı oranının belirlenmesi için iki farklı senaryo ile karşılaştırma işlemi yapılmıştır. Her senaryo için sonuçlar iki farklı yöntemle hazırlanmıştır. Temelde oluşturulan iki farklı senaryo şunlardır:

1. Sorgu çizgesinin sadece veri sınıflarının medyanları ile karşılaştırılması
2. Sorgu çizgesinin veri sınıflarının tamamında bulunan tüm çizgelerle karşılaştırılması

Bu iki işlem tüm 25 farklı Hausdorff Mesafesi metrikleri için gerçekleştirilmiştir. Ortaya çıkan uzaklık sonuçlarına göre sorgulanan çizgenin hangi veri kümesine ait olduğuna karar verilmesi aşamasında ise iki temel yapı tercih edilmiştir:

1. Çıkan sonuçlardan en yakın olanın doğru kabul edilmesi ve ortalama kaç tane doğru cevap verildiğinin tespit edilmesi



2. Sorgu çizgesi için tüm mesafe metrikleri içinden en çok sonuca sahip olan sınıfın içerisinde olduğunun tespit edilmesi

Sorgulanan verinin sadece medyanlarla karşılaştırılıp bir sonuca varılması işleminde %100 doğru cevap beklenmemektedir. Oransal olarak gerçeğe ne kadar yakın sonuçlar verdiği araştırılmış olup, bu işlemlerin gerçekleştirilmesi esnasında da zamandan ne oranda kazanıldığının belirlenmesi için çalışılmıştır.

#### 4.2. Veri Kümesi

Yapılan çalışmalarda silüet veri kümesinde faydalanılmıştır. Her bir cismin 10'ar derecelik açı farklılıklarıyla resmedildiği veri kümesi kullanılmıştır. Veri tabanı içerisinde toplam 9 farklı veri sınıfı (dürbün, kamera, araba, sandalye, bardak, saat, ütü, telefon, çaydanlık) bulunmaktadır. Her bir sınıf içerisinde toplam 18 adet farklı veri bulunmaktadır.



Şekil 4.1 Veri Sınıfları

Şekil 4.1'de tüm veri sınıflarından birer tane örnek gösterilmiştir. Alınan bu verilerin iskeletleme tekniği uygulandıktan sonra elde edilen ağaç yapıları, veri kümesi olarak kullanılmıştır. Özellikle karşılaştırma aşamasında yaşanan temel sorunların başında iskeletleme yönteminde kullanılan minimum kapsama ağacının tekil bir sonuç üretmemesi bulunmaktadır.

Veri kümesinde bulunan “sandalye” sınıfına ait objelerden birkaçı ve “bardak” sınıfına ait objelerden bazıları sırasıyla Şekil 4.2 ve Şekil 4.3’de gösterilmiştir.



Şekil 4.2 Sandalye Sınıfına(snd) Ait Objeler



Şekil 4.3 Bardak Sınıfına(brd) Ait Objeler

#### Çalışmanın adımları

1. Medyan hesaplanmasının analizi
2. Karşılaştırmaların ne ölçüde katkı sağladığının tespiti

### **4.3. Elde Edilen Sonular**

Geliřtirilen yeni yntemin bařarısı, karřılařtırmalardaki bařarısına gre llebilecektir. Bu nedenle karřılařtırmalı sonular medyanların oluřturulması ile alakalı olmayacaktır. Dolayısıyla sonu kısmında medyanların hesaplanmasının analizi zerinde alıřılacak daha sonra hesaplanan medyanların karřılařtırmalı sonuları verilecektir.

#### **4.3.1. Medyan hesaplanması analizi**

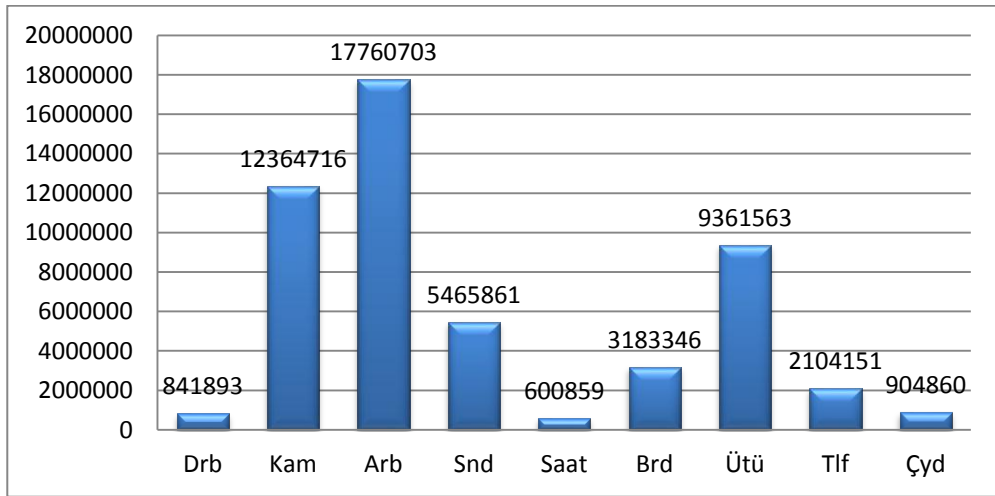
izge medyanının hesaplanması NP-Tam bir problemdir. Deęiřik yntemlerle bu hesaplama iřlemi yapılmaya alıřılmıřtır. Bu teknikle, birok uygulamada bařarılı sonular elde edilse de yapılan incelemeler sonucunda izge uzayından vektr uzayına geiř esnasında verilerin bir kısmının kaybolduęu gzlenmiřtir. Sunulan tezde, bu soruna ynelik zm geliřtirilmeye alıřılmıřtır. Yeni bir yaklařımla izge medyanının hesaplanması saęlanmış ve kurgulanan senaryoya gre yeni yaklařım bařarısını tespit edilmiřtir.

Kurgulanmıř olan yeni yaklařıma gre izge zerinde, Tırtıl Ayrıřması iřlemi ile yapraklar ile root dęm arasındaki mesafeler tespit edilmiř ve bu mesafe metriklerinin izge gmme teknięinde kullanılması saęlanmıştir. Geiř iřlemi neticesinde edilen vektr uzayındaki noktaların her biri, izge uzayındaki bir dęm temsil etmiř ve izge uzayındaki dęmn aęırlıęı lsnde vektr uzayına nokta eklenmiřtir. Tm noktalar kmesi ierisinde K-Means uygulaması alıřtırılarak olası izge medyanının hesaplanması saęlanmıştir. Bu iřlem veri tabanına her yeni eleman eklendięinde tekrar hesaplanması gereken fakat sadece bir kere hesaplanması yeterli olan bir uygulamadır. Veri sınıfının olası yeni elemanlarla gclendirilmesi halinde mevcut medyan K-Means uygulamasına bařlangı merkez noktaları olarak verilecek ve bu sayede daha nce yapılan iřlemler kaybedilmeden hızlı bir Őekilde yeni medyan hesaplanabilir olacaktır. Bu nedenle olası zaman kayıpları offline olarak deęerlendirilmelidir. rnek veri sınıfı iinde K-Means uygulamasıyla izge medyanının seilmesi sırasında geen sre, adım sayısı, sistemin toplam ka nokta iin ve ka merkez noktası iin alıřtırıldıęına dair bilgi izelge 4.1 iinde verilmiřtir.

Çizelge 4.1 K-Means Analiz Çizelgesi

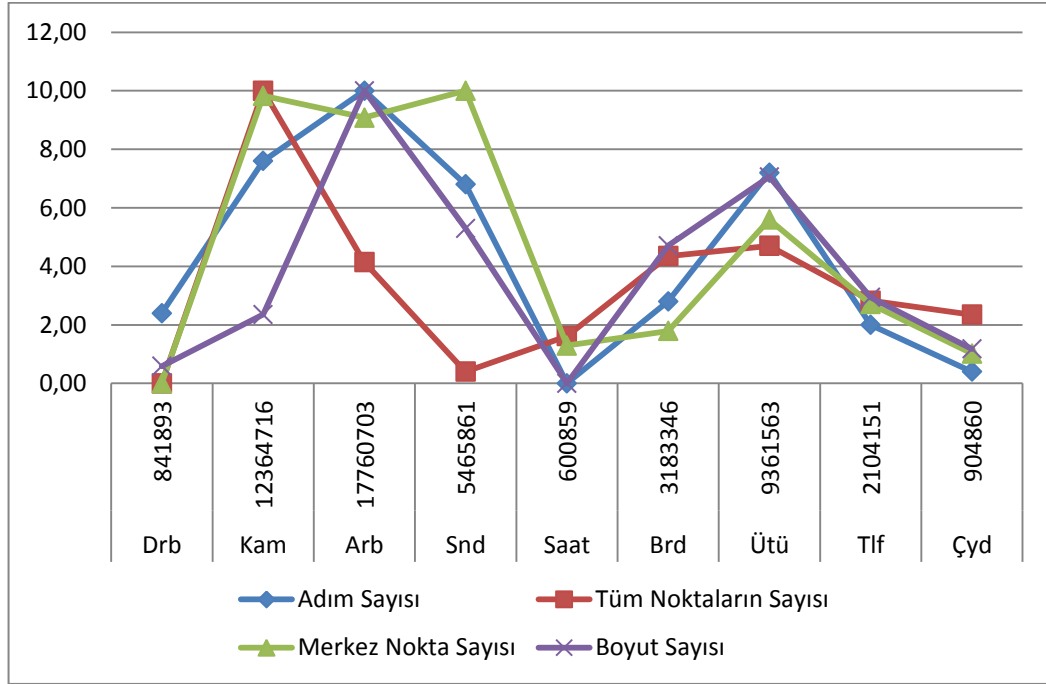
Veri Sınıfı	Geçen Süre	Adım Sayısı	Tüm Noktaların Sayısı	Merkez Nokta Sayısı	Boyut Sayısı
<b>Drb</b>	0 sa 14 dk 15 s 393 ms	14	215.212	398	8
<b>Kam</b>	3 sa 26 dk 43 s 416 ms	27	658.864	793	11
<b>Arb</b>	4 sa 56 dk 05 s 203 ms	33	398.901	763	24
<b>Snd</b>	1 sa 31 dk 50 s 861 ms	25	232.882	800	16
<b>Saat</b>	0 sa 10 dk 07 s 159 ms	8	286.893	450	7
<b>Brd</b>	0 sa 53 dk 26 s 746 ms	15	408.093	470	15
<b>Ütü</b>	2 sa 36 dk 07 s 863 ms	26	423.704	623	19
<b>Tlf</b>	0 sa 35 dk 32 s 951 ms	13	340.530	507	12
<b>Çyd</b>	0 sa 15 dk 47 s 160 ms	9	319.284	439	9

Çizge medyanı hesaplanırken birçok metrik yapılan işlem sayısını ve dolayısıyla geçen süre miktarını etkilemektedir. Toplam yapılan işlem sayısı ve algoritmanın karmaşıklık hesabı Bölüm 2’de verilmiştir. Elde edilen Çizelge 4.1 bir veri sınıfının hangi karakteristik özellikler içeriyor olmasına göre ne kadar süreyle medyan hesabının yapılabilir olduğunu göstermektedir.



**Şekil 4.4 K-Means Süre Analizi**  
x ekseninde veri sınıflarını, y ekseninde K-Means esnasında geçen sürenin ms cinsinden karşılığıdır

Veri kümesi içinde bulunan tüm veri sınıfları için medyan hesaplanması esnasında geçen süreye yönelik zaman grafiği Şekil 4.4’de gösterilmiştir. Her bir veri sınıfının medyan hesabının gerçekleştirilmesi için geçen süre “milisaniye” cinsinden verilmiştir.



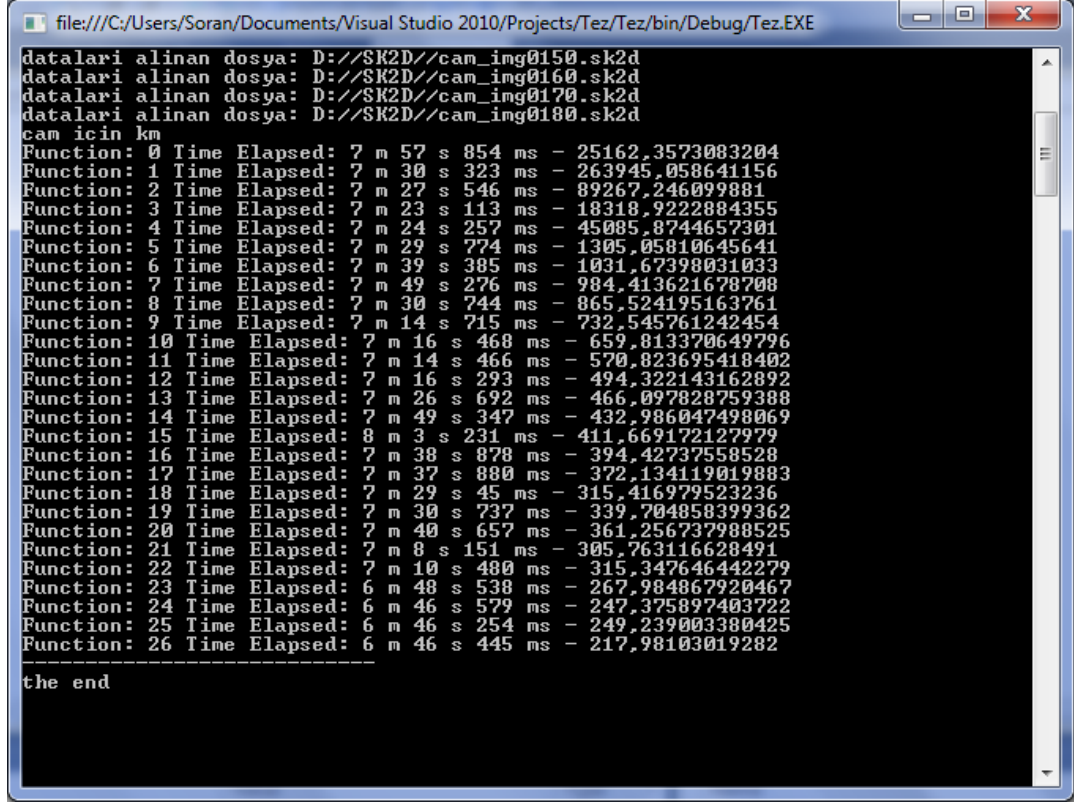
**Şekil 4.5 K-Means Performansına Etki Eden Faktörler**  
x eksenini veri sınıflarını, y eksenini K-Means performansına etki eden faktörlerin [0,10] aralığına normalize edilmiş halini göstermektedir.

Tüm veri sınıfları için “adım sayısı”, “tüm noktaların sayısı”, “merkez nokta sayısı” ve “boyut sayısı” değişkenlerinin geçen süreye nasıl etki gösterdiği tek grafik halinde gösterilmiştir (Şekil 4.5). Bölüm 2’de bahsedildiği gibi yapılan işlem sayısı ve geçecek olan süre tek bir değişkene bağlı değildir ve Şekil 4.5’de gösterilen tüm değişkenler medyan hesaplanması süresine etki etmektedir. Tüm değişkenler kendi değerleri içinde Eşitlik 4.1 ile [0,10] aralığında normalleştirilmiştir.

$$v \in A \text{ ve } v' = \frac{v - \min(A)}{\max(A) - \min(A)} * 10 \quad \text{Eşitlik 4.1}$$

Medyan hesaplanması sırasında en çok zaman alan “arb” sınıfına ait tüm veriler kendi içlerinde en yüksek değerlere yakınken, en az zaman alan “saat” sınıfına ait tüm verilerin kendi içlerinde en düşük değerlere yakın olduğu görülmektedir.

Örnek bir veri sınıfı için her bir adımın ne kadar süreyle gerçekleştirildiği ve her bir adımda merkezlerin yerlerinin kaç birim değiştiğinin analizi, K-Means algoritmasının hangi aşamalardan geçtiği Şekil 4.6’da sunulmuştur. Şekil 4.6 örnek ekran görüntüsüdür ve veri kümesi içinde bulunan “kam” veri sınıfı için uygulama çıktısıdır.



```
file:///C:/Users/Soran/Documents/Visual Studio 2010/Projects/Tez/Tez/bin/Debug/Tez.EXE
datalari alinan dosya: D://SK2D//cam_img0150.sk2d
datalari alinan dosya: D://SK2D//cam_img0160.sk2d
datalari alinan dosya: D://SK2D//cam_img0170.sk2d
datalari alinan dosya: D://SK2D//cam_img0180.sk2d
cam icin km
Function: 0 Time Elapsed: 7 m 57 s 854 ms - 25162,3573083204
Function: 1 Time Elapsed: 7 m 30 s 323 ms - 263945,058641156
Function: 2 Time Elapsed: 7 m 27 s 546 ms - 89267,246099881
Function: 3 Time Elapsed: 7 m 23 s 113 ms - 18318,9222884355
Function: 4 Time Elapsed: 7 m 24 s 257 ms - 45085,8744657301
Function: 5 Time Elapsed: 7 m 29 s 774 ms - 1305,05810645641
Function: 6 Time Elapsed: 7 m 39 s 385 ms - 1031,67398031033
Function: 7 Time Elapsed: 7 m 49 s 276 ms - 984,413621678708
Function: 8 Time Elapsed: 7 m 30 s 744 ms - 865,524195163761
Function: 9 Time Elapsed: 7 m 14 s 715 ms - 732,545761242454
Function: 10 Time Elapsed: 7 m 16 s 468 ms - 659,813370649796
Function: 11 Time Elapsed: 7 m 14 s 466 ms - 570,823695418402
Function: 12 Time Elapsed: 7 m 16 s 293 ms - 494,322143162892
Function: 13 Time Elapsed: 7 m 26 s 692 ms - 466,097828759388
Function: 14 Time Elapsed: 7 m 49 s 347 ms - 432,986047498069
Function: 15 Time Elapsed: 8 m 3 s 231 ms - 411,669172127979
Function: 16 Time Elapsed: 7 m 38 s 878 ms - 394,42737558528
Function: 17 Time Elapsed: 7 m 37 s 880 ms - 372,134119019883
Function: 18 Time Elapsed: 7 m 29 s 45 ms - 315,416979523236
Function: 19 Time Elapsed: 7 m 30 s 737 ms - 339,704858399362
Function: 20 Time Elapsed: 7 m 40 s 657 ms - 361,256737980525
Function: 21 Time Elapsed: 7 m 8 s 151 ms - 305,763116628491
Function: 22 Time Elapsed: 7 m 10 s 480 ms - 315,347646442279
Function: 23 Time Elapsed: 6 m 48 s 538 ms - 267,984867920467
Function: 24 Time Elapsed: 6 m 46 s 579 ms - 247,375897403722
Function: 25 Time Elapsed: 6 m 46 s 254 ms - 249,239003380425
Function: 26 Time Elapsed: 6 m 46 s 445 ms - 217,98103019282
the end
```

Şekil 4.6 K-Means İşlem Adımları Analizi

“kam” veri sınıfı için toplam 27 adım sonucunda çizge medyanı hesaplanmış ve her bir adımın ne kadar sürede gerçekleştirildiği verilmiştir. Her bir adım sonucunda merkez noktaların ne kadar yer değiştirdikleri hesaplanmış ve bu değer “200” birimin altına düştüğü anda sistem yeni adıma geçmeyi otomatik olarak durdurmuştur.

Gerçekleştirilen işlemler neticesinde veri sınıflarının her birine ait bir medyan nokta kümesi hesaplanmıştır. Medyan hesaplanması sonucunda veri tabanımıza ek olarak elimizde her bir veri sınıfı için bir medyan olmak üzere toplam 9 adet medyan bulunmaktadır.

Artık ele alınan temel problem, elde edilen bu medyanların gerçekten de obje eşleştirme problemlerinde kullanılabilir kullanılamayacağının tespit edilmesidir.

Kazanç ve kayıpların iyi analiz edilmesinin ve sistemin bu şekilde çalışmasının neden tercih edilmesi gerektiğinin tespit edilmesi önemlidir.

Medyanların oluşturulması aşamasında yapılan işlemler ve medyanların hesaplanmasının performans ölçümleri üzerinde çalışıldı. Yapılan işlemlerin performansa etkisi ve performans değerlendirmesi Bölüm 5 kısmında ele alınacaktır.

#### **4.3.2. Sonuçların Karşılaştırmaları**

Veri kümesinde 9 farklı veri sınıfı için her birinde 18 tane olmak üzere toplam 162 adet obje bulunmaktadır. Mevcut veri sınıfı üzerinden sistemin offline olarak eğitilmesiyle beraber her bir veri sınıfı için medyan hesabı yapılmıştır. Sorgu kümesi ile sistemin eğitim kümesi aynı küme olarak belirlenmiştir. Performans ölçümü için veri kümesinde yer alan her bir objenin hangi veri sınıfı içerisinde olabileceği tahmin edilmeye çalışılmış ve tahminin doğruluğu kontrol edilmiştir. Aynı obje ek olarak her bir veri sınıfına ait medyanlar ile karşılaştırılmış ve sadece medyanların kullanılmasıyla objenin hangi sınıfa ait olduğunun tespit edilmesi çalışılmıştır. Medyan kullanıldığı durumlar ve kullanılmadığı durumlarda doğruluk oranının nasıl değiştiği ve geçen sürenin nasıl etkilendiği incelenmiştir. Tüm bu karşılaştırma işlemleri için Hausdorff Mesafesinin 25 farklı hesaplama tekniği kullanılmıştır. Dolayısıyla her bir karşılaştırma işlemi için 25 farklı sonuç bulunmaktadır. Bu sonuçlardan hangilerinin doğru olduğunu karşılaştırmak için de iki farklı yöntem kullanılmıştır. İlk yaklaşımda (ortalama), her bir hesaplama tekniği kendi içerisinde özel olarak değerlendirilmiş ve doğru sonuçların ortalama değerleri alınmıştır. Böylelikle ortalama ne kadarlık bir doğruluk oranı sağlandığı tespit edilmiştir. Örneğin “drb” sınıfına ait bir obje için tüm medyanlarla karşılaştırma sonuçları tespit edilmiştir. Aynı işlem tüm hesaplama teknikleri için gerçekleştirilmiştir. “drb” sınıfının ortalama kaç kere doğru sonucu verdiği tüm hesaplama tekniklerinin doğruluk oranlarının ortalamasının alınmasıyla tespit edilmektedir. İkinci yaklaşımda (hibrit) ise sorgulan objenin, her bir veri sınıfı için kullanılan hesaplama yöntemlerinin sonuçlarına göre en fazla doğruluk oranına sahip olan sınıfın elemanı olabileceği tahmin edilmektedir. Örneğin yine “drb” sınıfı için, elimizde “drb” sınıfına ait her bir nesne için 25 farklı sonuç bulunmaktadır. “drb010” objesi için tüm hesaplama teknikleri içinde en fazla doğru sonucun verildiği sınıf, sorgulanan objenin o sınıfa ait olduğunun bir göstergesidir. Bu iki teknik arasındaki fark, ilk

yaklaşımında her bir hesaplama tekniğinin özel olarak ele alınması ve ikinci yaklaşımla karşılaştırma yapılabilmesi için ortalama alınması, ikinci yaklaşımda ise tek bir sonucun verilmesidir.

### **Ortalama - Medyanlı Sonuçlar**

Toplam 25 farklı metrik için her bir dosya medyanlarla karşılaştırılmıştır. En yüksek ortalama doğru tahmin oranı “d4-f5” metriği için çıkmıştır. Birbirine görüntü olarak yakın olan cisimlerin doğru tahmin oranları daha düşük çıkmaktadır. Bunun temel sebebi iskeletleme yönteminde kullanılan minimum kapsama ağacı algoritmasının tek bir çözüme sahip olmaması ve bu nedenle de objeden çıkartılan ağacın başka karakteristik öznitelikler gösterebiliyor olmasıdır. Tüm veriler için doğru sonuç verme oranı %44,91 olarak gözlenmiştir.



Çizelge 4.2 Medyan ile Karşılaştırma Sonuçları

	drb	kam	arb	snd	saat	brd	ütü	tlf	çyd	Ortalama
d2-f1	11,11	61,11	22,22	61,11	27,78	55,56	61,11	38,89	55,56	43,83
d2-f2	11,11	11,11	5,56	38,89	22,22	0,00	11,11	38,89	72,22	23,46
d2-f3	11,11	11,11	16,67	44,44	22,22	0,00	16,67	44,44	72,22	26,54
d2-f4	11,11	11,11	16,67	44,44	22,22	0,00	11,11	38,89	72,22	25,31
d2-f5	11,11	66,67	22,22	88,89	27,78	61,11	77,78	38,89	50,00	49,38
d3-f1	16,67	72,22	50,00	<b>100,0</b>	33,33	<b>100,0</b>	83,33	50,00	50,00	61,73
d3-f2	5,56	11,11	5,56	38,89	11,11	0,00	11,11	38,89	83,33	22,84
d3-f3	5,56	38,89	16,67	61,11	22,22	5,56	27,78	50,00	<b>88,89</b>	35,19
d3-f4	5,56	27,78	16,67	66,67	22,22	33,33	44,44	55,56	<b>88,89</b>	40,12
d3-f5	16,67	83,33	50,00	<b>100,0</b>	33,33	<b>100,0</b>	83,33	50,00	50,00	62,96
d4-f1	33,33	72,22	<b>72,22</b>	<b>100,0</b>	38,89	<b>100,0</b>	<b>88,89</b>	50,00	61,11	<b>68,52</b>
d4-f2	16,67	22,22	5,56	38,89	11,11	0,00	11,11	50,00	50,00	22,84
d4-f3	22,22	44,44	22,22	61,11	22,22	22,22	44,44	<b>72,22</b>	77,78	43,21
d4-f4	22,22	38,89	16,67	61,11	22,22	22,22	44,44	<b>72,22</b>	77,78	41,98
d4-f5	33,33	<b>94,44</b>	<b>72,22</b>	<b>100,0</b>	38,89	<b>100,0</b>	<b>88,89</b>	50,00	61,11	<b>70,99</b>
d5-f1	27,78	<b>94,44</b>	<b>66,67</b>	<b>100,0</b>	33,33	83,33	50,00	22,22	55,56	59,26
d5-f2	<b>72,22</b>	11,11	0,00	11,11	0,00	5,56	0,00	55,56	66,67	24,69
d5-f3	<b>88,89</b>	50,00	11,11	83,33	22,22	44,44	27,78	66,67	<b>88,89</b>	53,70
d5-f4	<b>94,44</b>	50,00	11,11	61,11	<b>55,56</b>	33,33	22,22	61,11	<b>88,89</b>	53,09
d5-f5	27,78	<b>94,44</b>	<b>66,67</b>	<b>100,0</b>	33,33	83,33	50,00	22,22	55,56	59,26
d6-f1	22,22	72,22	61,11	<b>100,0</b>	<b>44,44</b>	<b>100,0</b>	<b>88,89</b>	50,00	61,11	<b>66,67</b>
d6-f2	5,56	11,11	5,56	38,89	22,22	0,00	11,11	55,56	66,67	24,07
d6-f3	11,11	33,33	16,67	55,56	22,22	33,33	38,89	66,67	<b>94,44</b>	41,36
d6-f4	11,11	27,78	16,67	61,11	22,22	27,78	44,44	<b>72,22</b>	<b>94,44</b>	41,98
d6-f5	22,22	11,11	61,11	<b>100,0</b>	<b>44,44</b>	<b>100,0</b>	<b>88,89</b>	50,00	61,11	59,88
Ort.	24,67	44,89	29,11	<b>68,67</b>	27,11	44,44	45,11	<b>50,44</b>	<b>69,78</b>	<b>44,91</b>

Çizelge 4.2’de tüm uzaklık hesaplama teknikleri için yüzdesel olarak doğru tahmin etme oranları verilmiştir. Bu sayede her uzaklık hesaplama tekniğinin detaylı olarak incelenebilmesi söz konusudur. Her bir uzaklık hesaplama tekniğinin tüm sınıflara göre ortalama olarak başarıımı ve her bir veri sınıfı için tüm tekniklerin ortalama başarıımı Çizelge 4.2’de görülmektedir.

### **Hibrit - Medyanlı Sonuçlar**

Her bir veri sınıfı için hibrit yöntem ayrı olarak hesaplanmıştır. Her sınıf içerisinde toplam 18 farklı obje barındırmaktadır. Her bir objenin tüm medyanlarla karşılaştırılması tüm hesaplama teknikleri için gerçekleştirilmiştir. Elde edilen sonuçlara göre en fazla doğruluk oranına sahip sınıf, verilen objenin sınıfı olarak tahmin edilmiştir. Çizelge 4.3 ile “brd” veri sınıfı için tüm sonuçlar takip edilebilmektedir. “brd” veri sınıfına ait her bir obje için detaylı olarak yüzde oranında ne kadar doğru sonuç verildiği takip edilebilmektedir. “brd010” objesi için 25 farklı hesaplama tekniğinin %4,00 oranında “drb”, %8,00 oranında “kam”, %56,00 oranında “brd” ve %32 oranında “çyd” sınıfına ait sonuçlar alındığı görülmektedir. Bunların içerisinde en fazla doğruluk oranına sahip olan sınıf “brd” sınıfı olduğu için “brd010” objesi için tahmin edilen veri sınıfı “brd” sınıfı olacaktır. “brd” sınıfı için toplam 18 objeden 15’i için doğru sonuç verildiği Çizelge 4.3 ile görülmektedir.

Çizelge 4.3 Medyan Kullanılarak Gerçekleştirilen Hibrit Örneği

	drb	kam	arb	snd	saat	brd	ütü	tlf	çyd
<b>brd010</b>	4,00	8,00	0,00	0,00	0,00	<b>56,00</b>	0,00	0,00	32,00
<b>brd020</b>	0,00	8,00	0,00	0,00	20,00	<b>44,00</b>	0,00	16,00	12,00
<b>brd030</b>	20,00	0,00	0,00	0,00	0,00	<b>44,00</b>	0,00	16,00	20,00
<b>brd040</b>	8,00	44,00	0,00	0,00	0,00	<b>48,00</b>	0,00	0,00	0,00
<b>brd050</b>	16,00	0,00	0,00	0,00	0,00	<b>44,00</b>	0,00	8,00	32,00
<b>brd060</b>	12,00	32,00	0,00	0,00	0,00	<b>48,00</b>	0,00	0,00	8,00
<b>brd070</b>	12,00	32,00	0,00	0,00	0,00	<b>48,00</b>	0,00	0,00	8,00
<b>brd080</b>	4,00	24,00	0,00	0,00	0,00	<b>68,00</b>	0,00	0,00	4,00
<b>brd090</b>	4,00	40,00	0,00	0,00	0,00	<b>52,00</b>	0,00	0,00	4,00
<b>brd100</b>	8,00	<b>44,00</b>	0,00	0,00	0,00	<b>40,00</b>	8,00	0,00	0,00
<b>brd110</b>	0,00	0,00	0,00	0,00	0,00	<b>40,00</b>	0,00	20,00	40,00
<b>brd120</b>	8,00	8,00	0,00	0,00	24,00	<b>40,00</b>	0,00	8,00	12,00
<b>brd130</b>	16,00	12,00	0,00	0,00	0,00	<b>28,00</b>	8,00	12,00	24,00
<b>brd140</b>	0,00	8,00	0,00	0,00	0,00	<b>24,00</b>	8,00	16,00	<b>44,00</b>
<b>brd150</b>	8,00	44,00	0,00	0,00	0,00	<b>48,00</b>	0,00	0,00	0,00
<b>brd160</b>	0,00	12,00	0,00	0,00	4,00	<b>44,00</b>	0,00	40,00	0,00
<b>brd170</b>	0,00	0,00	0,00	0,00	0,00	<b>52,00</b>	4,00	28,00	16,00
<b>brd180</b>	0,00	0,00	0,00	0,00	<b>36,00</b>	<b>32,00</b>	0,00	20,00	12,00

#### Hibrit ile Ortalama Yaklaşımları Arasındaki İlişki (Medyan Hesaplamalı)

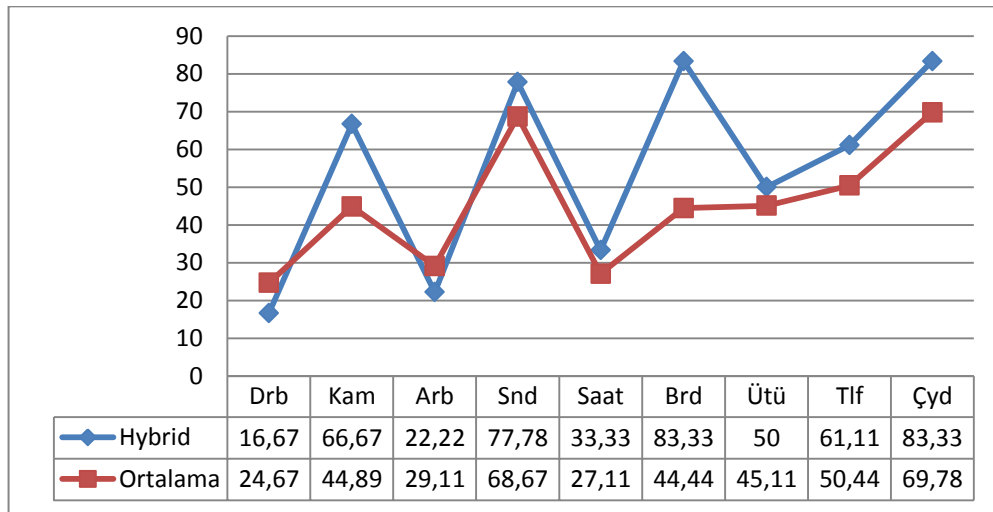
Her bir veri sınıfı için Çizelge 4.2’de elde edilen ortalama değerler ile, “brd” sınıfı için bir örneği Çizelge 4.3’de verilen her sınıf için hibrit yaklaşımının sonuçları Çizelge 4.4 ile karşılaştırılmıştır. Örneğin “brd” sınıfı için Çizelge 4.3’de gözlemlenebildiği gibi 18 objeden 15 obje için doğru sonuç verilmektedir. Bu doğruluk oranı olarak %83,33 ile ifade edilirken, Çizelge 4.2’de elde edilen ortalama oran ise %44,44 olarak gözlemlenmiştir. Hibrit ve Ortalama yaklaşımları için

ortalama doğruluk oranları ise %54,94 ve %44,91 olmaktadır. Bu durumda hibrit yaklaşımının doğruluk oranını artırıcı özelliği olduğu söylenebilmektedir.

Çizelge 4.4 Medyan Kullanılarak Hibrit ile Ortalama Karşılaştırması

	Hibrit	Ortalama
<b>drb</b>	16,67	24,67
<b>kam</b>	66,67	44,89
<b>arb</b>	22,22	29,11
<b>snd</b>	77,78	68,67
<b>saat</b>	33,33	27,11
<b>brd</b>	83,33	44,44
<b>ütü</b>	50,00	45,11
<b>tlf</b>	61,11	50,44
<b>çyd</b>	83,33	69,78
<b>Ortalama</b>	54,94	44,91

Çizelge 4.4'e ait grafik görüntüsü Şekil 4.7'de gösterilmiştir. Verilen grafiğe göre hibrit yaklaşım, Ortalama yaklaşıma göre daha doğru sonuçlar vermektedir.



Şekil 4.7 Medyanlı Hibrit ve Ortalama Sonuçların Karşılaştırılması

x eksenini veri sınıflarını, y eksenini yüzde olarak

doğru sonuç oranını göstermektedir.

### **Ortalama - Medyansız Sonular**

Toplam 25 farklı uzaklık hesaplama tekniđi iin her bir obje tm objelerle karřılařtırılmıřtır ve sorgulana objenin en yakın uzaklıktaki objenin sahip olduđu sınıfa ait olduđu tahmin edilmektedir. Her bir uzaklık hesaplama tekniđi iin toplam  $162 * 162 = 26244$  adet karřılařtırma iřlemi yapılmıřtır. En yksek ortalama dođru tahmin oranı “d6-f2” metriđi iin ıkmıřtır. Tm veriler iin dođru sonu verme oranı %49,51 olarak gzlenmiřtir. “saat” sınıfı %94,22 ortalamaıyla en dođru sonucun tahmin edilebildiđi sınıftır.

Çizelge 4.5 Medyan Kullanılmadan Karşılaştırma Sonuçları

	drb	kam	arb	snd	saat	brd	ütü	tlf	çyd	Ortalama
d2-f1	66,67	0,00	5,56	11,11	88,89	22,22	0,00	55,56	61,11	34,57
d2-f2	<b>83,33</b>	33,33	11,11	44,44	<b>100,00</b>	55,56	11,11	55,56	<b>83,33</b>	53,09
d2-f3	77,78	33,33	5,56	38,89	<b>100,00</b>	55,56	5,56	55,56	<b>83,33</b>	50,62
d2-f4	77,78	33,33	5,56	38,89	<b>100,00</b>	55,56	11,11	55,56	<b>83,33</b>	51,23
d2-f5	44,44	44,44	5,56	50,00	88,89	55,56	5,56	44,44	<b>88,89</b>	47,53
d3-f1	61,11	0,00	5,56	11,11	<b>100,00</b>	22,22	0,00	44,44	22,22	29,63
d3-f2	50,00	50,00	16,67	50,00	<b>100,00</b>	66,67	27,78	<b>77,78</b>	77,78	57,41
d3-f3	77,78	38,89	16,67	38,89	<b>100,00</b>	72,22	22,22	72,22	77,78	57,41
d3-f4	72,22	44,44	16,67	44,44	<b>100,00</b>	72,22	22,22	72,22	77,78	<b>58,02</b>
d3-f5	27,78	<b>55,56</b>	16,67	55,56	<b>100,00</b>	66,67	11,11	44,44	33,33	45,68
d4-f1	55,56	0,00	5,56	11,11	88,89	22,22	0,00	44,44	38,89	29,63
d4-f2	66,67	50,00	<b>33,33</b>	<b>61,11</b>	88,89	72,22	<b>55,56</b>	<b>77,78</b>	77,78	<b>64,81</b>
d4-f3	<b>83,33</b>	27,78	16,67	38,89	94,44	72,22	16,67	<b>77,78</b>	72,22	55,56
d4-f4	<b>83,33</b>	38,89	16,67	50,00	94,44	72,22	22,22	<b>77,78</b>	72,22	58,64
d4-f5	11,11	<b>55,56</b>	11,11	55,56	88,89	72,22	38,89	22,22	27,78	42,59
d5-f1	44,44	0,00	0,00	11,11	88,89	44,44	5,56	38,89	33,33	29,63
d5-f2	61,11	33,33	11,11	50,00	88,89	<b>83,33</b>	<b>72,22</b>	55,56	66,67	<b>58,02</b>
d5-f3	77,78	22,22	5,56	27,78	88,89	<b>88,89</b>	22,22	55,56	72,22	51,23
d5-f4	<b>83,33</b>	27,78	11,11	33,33	88,89	<b>88,89</b>	44,44	55,56	72,22	56,17
d5-f5	16,67	<b>72,22</b>	0,00	38,89	88,89	77,78	27,78	33,33	5,56	40,12
d6-f1	50,00	0,00	11,11	11,11	88,89	27,78	0,00	44,44	33,33	29,63
d6-f2	77,78	50,00	<b>27,78</b>	<b>61,11</b>	<b>100,00</b>	<b>83,33</b>	<b>50,00</b>	<b>77,78</b>	77,78	67,28
d6-f3	<b>83,33</b>	38,89	16,67	38,89	<b>100,00</b>	<b>83,33</b>	33,33	<b>77,78</b>	<b>83,33</b>	61,73
d6-f4	<b>83,33</b>	38,89	<b>27,78</b>	44,44	<b>100,00</b>	<b>83,33</b>	33,33	<b>77,78</b>	<b>83,33</b>	<b>63,58</b>
d6-f5	5,56	50,00	11,11	<b>61,11</b>	88,89	77,78	22,22	38,89	38,89	43,83
<b>Ortalama</b>	60,89	33,56	12,44	39,11	<b>94,22</b>	<b>63,78</b>	22,44	57,33	<b>61,78</b>	<b>49,51</b>

## **Hibrit - Medyansız Sonular**

Her bir veri sınıfı iin hibrit ynlem ayrı olarak hesaplanmıřtır. Her sınıf ierisinde toplam 18 farklı obje barındırmaktadır. Her bir objenin tm objelerle karřılařtırılması tm hesaplama teknikleri iin gerekleřtirilmiřtir. Tm hesaplama teknikleri sonularına gre en fazla doėruluk oranına sahip sınıf verilen objenin sınıfı olarak tahmin edilmiřtir. izelge 4.6 ile “brd” veri sınıfı iin tm sonular takip edilebilmektedir. “brd” veri sınıfına ait her bir obje iin detaylı olarak yzde oranında ne kadar doėru sonu verildiėi takip edilebilmektedir. “brd010” objesi iin 25 farklı hesaplama tekniėinin %12,00 oranında “cam”, %40,00 oranında “brd”, %20,00 oranında “t”, %4 oranında “tlf” ve %24 oranında “yd” sınıfına ait sonular alındıėı grlmektedir. Bunların ierisinde en fazla doėruluk oranına sahip olan sınıf “brd” sınıfı olduėu iin “brd010” objesi iin tahmin edilen veri sınıfı “brd” sınıfı olacaktır. “brd” sınıfı iin toplam 18 objeden 15’i iin doėru sonu verildiėi izelge 4.6 ile grnmektedir. Medyan kullanılması ve kullanılmaması ile alakalı yorumlar karřılařtırma kısmında yapılacaktır.

Çizelge 4.6 Medyan Kullanılmadan Gerçekleştirilen Hibrit Örneği

	drb	kam	arb	snd	saat	brd	ütü	tlf	çyd
<b>brd010</b>	0,00	12,00	0,00	0,00	0,00	<b>40,00</b>	20,00	4,00	24,00
<b>brd020</b>	20,00	0,00	0,00	0,00	0,00	<b>80,00</b>	0,00	0,00	0,00
<b>brd030</b>	12,00	0,00	0,00	0,00	0,00	<b>72,00</b>	0,00	16,00	0,00
<b>brd040</b>	16,00	0,00	0,00	0,00	12,00	<b>72,00</b>	0,00	0,00	0,00
<b>brd050</b>	16,00	0,00	0,00	0,00	0,00	<b>68,00</b>	0,00	16,00	0,00
<b>brd060</b>	0,00	0,00	0,00	0,00	0,00	<b>100,00</b>	0,00	0,00	0,00
<b>brd070</b>	0,00	0,00	0,00	0,00	0,00	<b>100,00</b>	0,00	0,00	0,00
<b>brd080</b>	8,00	0,00	0,00	0,00	0,00	<b>92,00</b>	0,00	0,00	0,00
<b>brd090</b>	8,00	0,00	0,00	0,00	0,00	<b>92,00</b>	0,00	0,00	0,00
<b>brd100</b>	36,00	0,00	0,00	0,00	0,00	<b>48,00</b>	0,00	16,00	0,00
<b>brd110</b>	0,00	0,00	0,00	0,00	0,00	<b>12,00</b>	<b>52,00</b>	4,00	32,00
<b>brd120</b>	20,00	0,00	0,00	0,00	12,00	<b>52,00</b>	16,00	0,00	0,00
<b>brd130</b>	28,00	8,00	4,00	0,00	0,00	<b>60,00</b>	0,00	0,00	0,00
<b>brd140</b>	20,00	0,00	0,00	0,00	8,00	<b>52,00</b>	12,00	0,00	8,00
<b>brd150</b>	16,00	0,00	0,00	0,00	0,00	<b>84,00</b>	0,00	0,00	0,00
<b>brd160</b>	20,00	0,00	0,00	0,00	0,00	<b>80,00</b>	0,00	0,00	0,00
<b>brd170</b>	16,00	0,00	0,00	0,00	16,00	<b>28,00</b>	4,00	4,00	<b>32,00</b>
<b>brd180</b>	0,00	4,00	0,00	0,00	<b>52,00</b>	<b>16,00</b>	8,00	16,00	4,00

#### Tüm veri sınıfları için hibrit ile ortalamanın farkı (medyansız)

Her bir veri sınıfı için Çizelge 4.5’de elde edilen ortalama değerler ile “brd” sınıfı için bir örneği Çizelge 4.6’da verilen her sınıf için hibrit yaklaşımının sonuçları Çizelge 4.7 ile bütünleştirilmiştir. Örneğin “brd” sınıfı için Çizelge 4.6’da gözlemlenebildiği gibi 18 objeden 15 obje için doğru sonuç verilmektedir. Bu doğruluk oranı olarak %83,33 ile ifade edilirken, Çizelge 4.5’de elde edilen ortalama oran ise %63,78 olarak gözlemlenmiştir. Hibrit ve Ortalama yaklaşımları için



ortalama doğruluk oranları ise %61,11 ve %49,50 olmaktadır. Bu durumda hibrit yaklaşımının doğruluk oranını artırıcı özelliği olduğu söylenebilmektedir.

Çizelge 4.7 Medyan Kullanılmadan Hibrit ile Ortalama Karşılaştırması

	<b>Hibrit</b>	<b>Ortalama</b>
<i>drb</i>	77,78	60,89
<i>kam</i>	44,44	33,56
<i>arb</i>	11,11	12,44
<i>snd</i>	44,44	39,11
<i>saat</i>	100,00	94,22
<i>brd</i>	83,33	63,78
<i>ütü</i>	27,78	22,44
<i>tlf</i>	77,78	57,33
<i>tea</i>	83,33	61,78
<b>Ortalama</b>	61,11	49,50

### **Karşılaştırmalı Sonuçlar**

Medyan kullanılarak elde edilen sonuçlar ile medyan kullanılmadan elde edilen sonuçların her biri ayrı çizelgeler halinde ve grafiklerle verilmiştir. Bu iki yaklaşımın kendi aralarındaki rekabeti yeni yaklaşımın performansını verecektir.

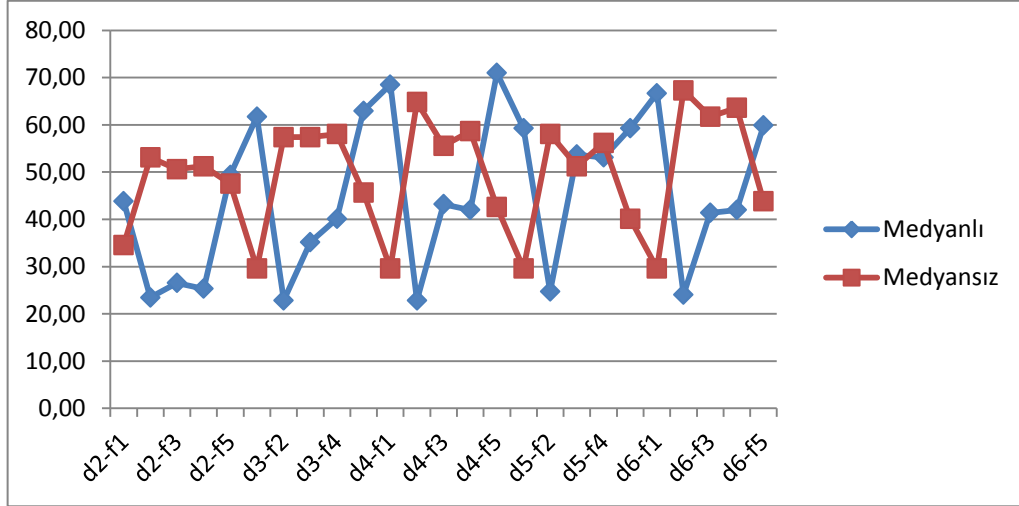
### **Ortalama Yüzde karşılaştırması**

Her bir mesafe uzaklığı ölçüm tekniğinin Medyanlı-Medyansız karşılaştırmalı sonucu Çizelge 4.8'de verilmiştir. Medyansız karşılaştırmanın daha doğru oranı yüksek sonuç vereceği tez çalışmasına başlama aşamasında öngörülmüştür. Aradaki farkın olabildiğince az ve yapılan işlemin daha hızlı geliştirilebilmesinin önünün açılması için çalışmalar yapılmıştır.

Çizelge 4.8 Medyanlı ve Medyansız Ortalama Başarımın Karşılaştırılması

<b>ORTALAMA YÜZDE</b>		
	<b>Medyanlı</b>	<b>Medyansız</b>
<b>d2-f1</b>	43,83	34,57
<b>d2-f2</b>	23,46	53,09
<b>d2-f3</b>	26,54	50,62
<b>d2-f4</b>	25,31	51,23
<b>d2-f5</b>	49,38	47,53
<b>d3-f1</b>	61,73	29,63
<b>d3-f2</b>	22,84	57,41
<b>d3-f3</b>	35,19	57,41
<b>d3-f4</b>	40,12	58,02
<b>d3-f5</b>	62,96	45,68
<b>d4-f1</b>	68,52	29,63
<b>d4-f2</b>	22,84	64,81
<b>d4-f3</b>	43,21	55,56
<b>d4-f4</b>	41,98	58,64
<b>d4-f5</b>	70,99	42,59
<b>d5-f1</b>	59,26	29,63
<b>d5-f2</b>	24,69	58,02
<b>d5-f3</b>	53,70	51,23
<b>d5-f4</b>	53,09	56,17
<b>d5-f5</b>	59,26	40,12
<b>d6-f1</b>	66,67	29,63
<b>d6-f2</b>	24,07	67,28
<b>d6-f3</b>	41,36	61,73
<b>d6-f4</b>	41,98	63,58
<b>d6-f5</b>	59,88	43,83
<b>Ortalama</b>	44,91	49,50

Medyan kullanılan yaklaşımda 5 numaralı uzaklık formülü seçilmesi, medyan kullanılmayan yaklaşımda ise 6 numaralı uzaklık formülü seçilmesi ortalama doğruluk oranını yukarı çeken faktörlerden biri olarak göze çarpmaktadır. Çizelge 4.8 ile verilen tablonun grafiksel gösterimi Şekil 4.8’de sunulmuştur.



**Şekil 4.8 Medyanlı/Medyansız Sonuçların Karşılaştırılması**

x eksenini uzaklık hesaplama yöntemlerini, y eksenini doğru

sonuç verme oranını göstermektedir

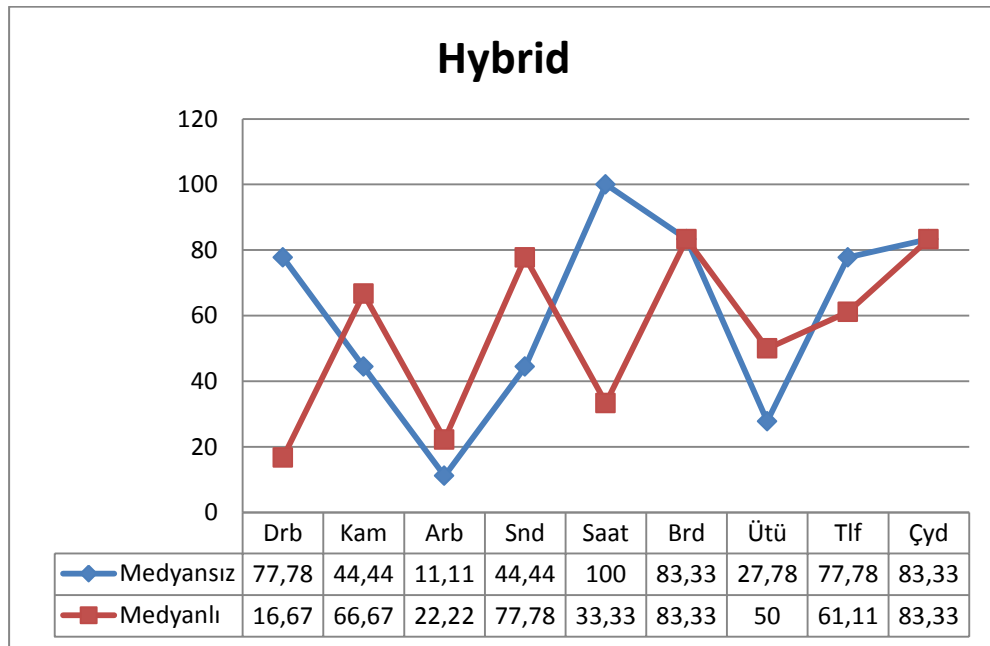
### Hibrit - Ortalama karşılaştırması

Mevcut iki yaklaşımdan birincisi hibrit yaklaşım ikincisi ise Ortalama yaklaşımıdır. Her iki yaklaşım için de Medyanlı ve Medyansız testler gerçekleştirilmiştir. Hibrit yaklaşımın Ortalama yaklaşımına göre daha iyi sonuçlar verdiği, Çizelge 4.9’da verilen tabloda görülmektedir.

Çizelge 4.9 Medyanlı ve Medyansız Başarımın Karşılaştırılması

	Hibrit		Ortalama	
	Medyansız	Medyanlı	Medyansız	Medyanlı
<i>drb</i>	77,78	16,67	60,89	24,67
<i>kam</i>	44,44	66,67	33,56	44,89
<i>arb</i>	11,11	22,22	12,44	29,11
<i>snd</i>	44,44	77,78	39,11	68,67
<i>saat</i>	100,00	33,33	94,22	27,11
<i>brd</i>	83,33	83,33	63,78	44,44
<i>ütü</i>	27,78	50,00	22,44	45,11
<i>tlf</i>	77,78	61,11	57,33	50,44
<i>çyd</i>	83,33	83,33	61,78	69,78
<i>ortalama</i>	61,11	54,94	49,51	44,91

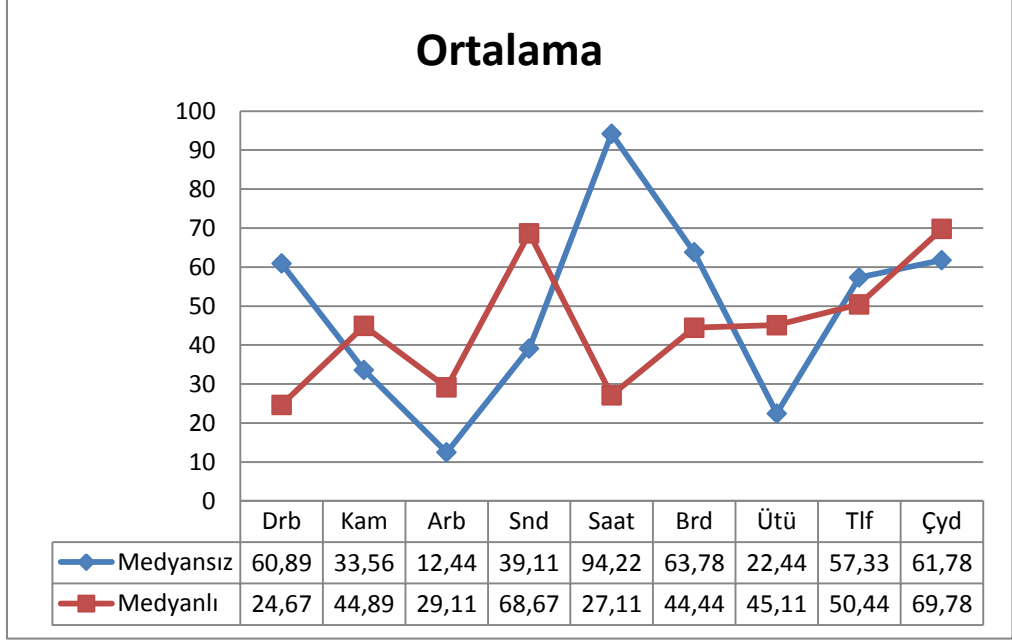
Çizelge 4.9 yardımıyla Şekil 4.9 ve Şekil 4.10'da Medyanlı ve Medyansız yaklaşımın probleme nasıl bir etkide bulunacağı belirtilmektedir.



Şekil 4.9 Hibrit Yöntemi İçin Medyanlı/Medyansız Karşılaştırması

x eksenini veri sınıflarını, y eksenini doğru

sonuç verme oranını göstermektedir



**Şekil 4.10 Ortalama Yöntemi İçin Medyanlı/Medyansız Karşılaştırması**  
 x eksenini veri sınıflarını, y eksenini doğru  
 sonuç verme oranını göstermektedir

Şekil 4.9 ve Şekil 4.10'da görüldüğü üzere medyanlı sonuç zaman zaman daha iyi sonuç veriyor olsa da genel olarak medyansız yaklaşım daha doğru sonuçlar vermektedir. Medyan kullanılması veya kullanılmaması arasında doğruluk oranı açısından çok büyük farklılıklar olmadığı gözlemlenmiştir. Bununla birlikte, iki teknik arasında çok ciddi manada görülen zaman farkı, yeni yaklaşımın en büyük avantajı olmaktadır.

#### **Zaman karşılaştırılması**

Medyan kullanılmaksızın, yapılan işlem sayısına her bir uzaklık hesaplama tekniği için geçen süre Çizelge 4.10 ile verilmiştir. Yaklaşık olarak her bir işlemin gerçekleştirilmesi ve bir sonuç vermesi 2 saat civarında bir süre almaktadır.

Çizelge 4.10 Medyan Kullanılmadan Karşılaştırma Süresi

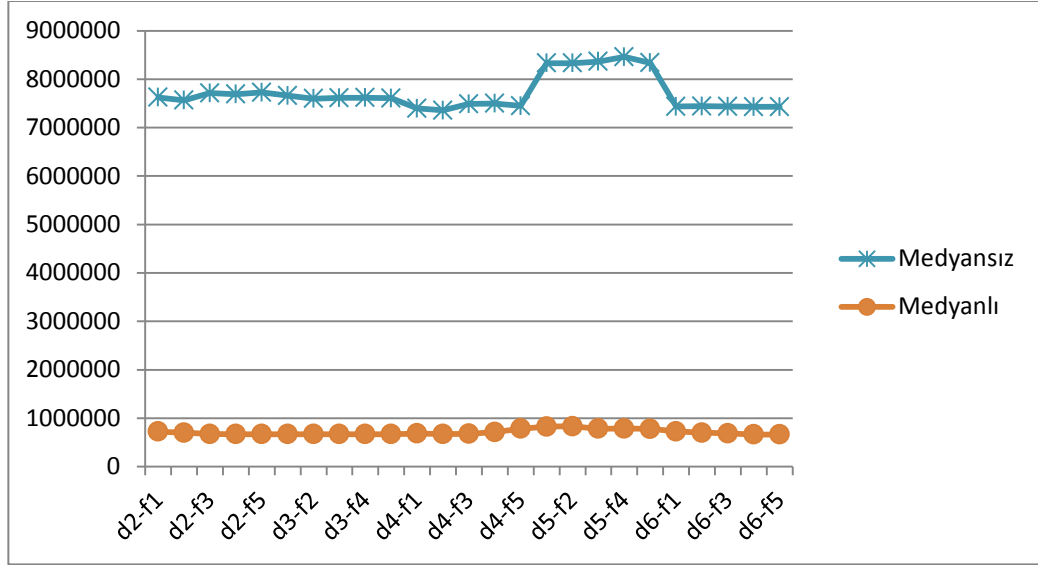
<b>Ms</b>	
<b>d2-f1</b>	7623933
<b>d2-f2</b>	7564381
<b>d2-f3</b>	7712399
<b>d2-f4</b>	7690598
<b>d2-f5</b>	7727274
<b>d3-f1</b>	7659989
<b>d3-f2</b>	7598000
<b>d3-f3</b>	7614561
<b>d3-f4</b>	7617326
<b>d3-f5</b>	7606214
<b>d4-f1</b>	7395895
<b>d4-f2</b>	7354568
<b>d4-f3</b>	7487072
<b>d4-f4</b>	7497677
<b>d4-f5</b>	7447354
<b>d5-f1</b>	8331006
<b>d5-f2</b>	8331478
<b>d5-f3</b>	8365124
<b>d5-f4</b>	8459200
<b>d5-f5</b>	8336862
<b>d6-f1</b>	7434707
<b>d6-f2</b>	7440826
<b>d6-f3</b>	7436413
<b>d6-f4</b>	7427706
<b>d6-f5</b>	7429089

Çizelge 4.11 ise medyan kullanıldığı durumda her bir uzaklık hesaplama tekniği için sonuçların çıkarılması için geçen süreyi vermektedir.

Çizelge 4.11 Medyan Kullanılarak Karşılaştırma Süresi

	<b>ms</b>
<b>d2-f1</b>	723613
<b>d2-f2</b>	702035
<b>d2-f3</b>	674706
<b>d2-f4</b>	670348
<b>d2-f5</b>	670320
<b>d3-f1</b>	670690
<b>d3-f2</b>	670426
<b>d3-f3</b>	670586
<b>d3-f4</b>	670297
<b>d3-f5</b>	670442
<b>d4-f1</b>	685575
<b>d4-f2</b>	674503
<b>d4-f3</b>	679017
<b>d4-f4</b>	714824
<b>d4-f5</b>	782239
<b>d5-f1</b>	825785
<b>d5-f2</b>	833712
<b>d5-f3</b>	784764
<b>d5-f4</b>	785877
<b>d5-f5</b>	779869
<b>d6-f1</b>	728841
<b>d6-f2</b>	702034
<b>d6-f3</b>	688111
<b>d6-f4</b>	664364
<b>d6-f5</b>	666076

Çizelge 4.10 ve Çizelge 4.11 incelendiğinde medyan kullanıldığı durumda geçen sürenin 2 saatlerden 12 dakikalara civarına indiği gözlemlenmiştir Bunlara ek olarak doğruluk oranında kayıp %5-%6 civarında gerçekleşmektedir.



**Şekil 4.11 Süre Bazlı Medyanlı/Medyansız Karşılaştırması**  
x eksenini kullanılan hesaplama yöntemini, y eksenini ms  
cinsinden geçen süreyi göstermektedir

Şekil 4.11, Çizelge 4.10 ve Çizelge 4.11 içindeki verilerin grafiksel gösterimidir. Görüldüğü gibi medyan kullanıldığı takdirde sistemin çalışma hızı oldukça düşmektedir. Bir başka göze çarpan nokta ise “d5” uzaklık mesafesi ölçüm formülü diğer uzaklık ölçütlerine nazaran daha uzun sürede çalışmaktadır.



## BÖLÜM 5

### 5. SONUÇ

#### 5.1. Değerlendirme

Tez çalışması kapsamında obje eşleştirme işlemini yeni yaklaşımlarla hızlandırmak ve doğruluk oranını olabildiğince yüksek tutmak amaçlanmıştır. Daha önce yapılan çalışmalar da göz önünde bulundurularak obje eşleştirme işlemine, öznitelik eşleştirme yöntemiyle yaklaşmıştır. Her bir objenin gösterimi öznitelikleri doğrultusunda çizge yöntemiyle yapılmış ve çizgelerin karşılaştırılması üzerinde durulmuştur.

Sorgusu yapılan objenin hangi veri sınıfına ait olduğunun belirlenebilmesi için veri sınıflarında bulunan tüm objelerin medyanının çıkarılması ve karşılaştırma işleminin sadece bu medyanlar ile yapılmasının nasıl sonuç verdiği araştırılmıştır. Veri sınıfı içinde bulunan çok sayıda çizgenin ortak bir temsilci yardımıyla ifade edilmesi sağlanmaya çalışılmıştır. Çizgeler üzerinde medyan hesaplama işlemi NP-Tam bir problem olduğu için çizgeler vektör uzayında ifade edilmiş ve vektör uzayında medyanı temsil edebilecek bir nokta kümesi hesaplanması hedeflenmiştir.

Çizgelerin vektör uzayına geçiş işlemi Tırtıl Ayrışması yardımıyla gerçekleştirilmiştir. Tırtıl Ayrışması, çizgelerin yaprak düğüm-root düğüm arasındaki yollarla ifade edilmesini sağlamaktadır. Bu yöntemin uygulanması sonucu elde edilen çıktı yardımıyla çizge, vektör uzayında bir vektör olarak ifade edilmektedir. Gömme işlemi neticesinde bir çizge, vektör uzayında, bulundurduğu düğüm sayısı kadar nokta ile gösterilmektedir.

Vektör uzayında bulunan noktaların tamamı veri sınıfı içinde bulunan tüm çizgelerin gömme tekniğiyle elde edilmiş vektörel karşılıklarıdır. Tüm noktalar üzerinde uygulanan K-Means algoritmasıyla tüm noktaların temsil edildiği yeni nokta kümesi oluşturulmuştur. Elde edilen bu nokta kümesi, olması gereken çizge medyanının vektör uzayında ifade edilmiş halidir. Karşılaştırma işlemi vektör uzayında gerçekleştirildiği için medyan çizgenin oluşturulmasına gerek kalmamaktadır.

Nokta kümeleri arasındaki mesafenin belirlenmesi için Hausdorff Mesafesi, Earth Mover's Distance (EMD) gibi algoritmalar kullanılmaktadır. Tez çalışmasında iki

nokta kümesi arasındaki mesafeyi hesaplamak için Hausdorff Mesafesi kullanılmıştır. Olası hata miktarını azaltmak için Hausdorff Mesafesi için yeniden düzenlemesi yapılan 25 farklı hesaplama metriği kullanılmıştır.

Yeni yaklaşım için iki ana bölüm mevcuttur. Birinci bölüm veri kümesinde bulunan objelerin kullanılmasıyla çizge medyanlarının, çizge gömme teknikleri kullanılarak hesaplanmasıdır. İkinci bölüm ise hesaplanan çizge medyanının başarımının tespit edilmesidir.

Çizge medyanı hesaplanması, eğitim kümesi içinde bulunan objelerin kullanılmasıyla offline olarak hesaplanmaktadır. Eğitim kümesi değişmediği müddetçe sorgu esnasında tekrar medyan hesabı yapılmasına gerek olmamaktadır. Veri sınıfına yeni obje eklendiği zaman yeni medyan hesaplanması için tüm sistemin tekrar çalıştırılmasına gerek kalmamaktadır. Yeni eklenen objenin ve veri sınıfında bulunan tüm objelerin vektör uzayına gömülmesi ve K-Means başlangıç merkezlerinin de mevcut medyan olarak sisteme verilmesi durumunda eski yapılan çalışmalar kaybolmadan yeni medyan hesaplanmış olacaktır. Sistemin ölçeklenmeye açık olması ve yeni medyanın en az masrafla hesaplanabiliyor olması yeni yaklaşımın önemli özelliklerinden biridir.

Çizge medyanı hesaplanması için geçen süre sorgu esnasında harcanmadığı için ayrı değerlendirilmesi gerekmektedir. Karşılaştırma işlemleri yapılırken medyan hesaplanmasının analizi ayrıca yapılmış olup, obje karşılaştırma süreleri üzerinden performans değerlendirmesi yapılmıştır.

Çizge medyanı hesaplanması ve karşılaştırma işlemleri vektör uzayında gerçekleştirilmektedir. Çizgelerin vektör uzayına geçirilmesi esnasında hangi gömme tekniğinin kullanılacağı önemli bir noktadır. Çizgelerin gömülmesi esnasında mümkün olduğunca az öznitelik kaybı yaşanması gerekmektedir. Çizgelerin vektör olarak ifade edilebilmesi için tez çalışmasında Tırtıl Ayrışması yöntemi kullanılmıştır. Bu sayede her bir düğümün özellikleri sabit kalmaktadır fakat vektör uzayında her bir nokta bir düğümü ifade etmektedir. Daha önce çizge medyanı hesaplanması için yapılan çalışmalarda, gömme işlemi, her bir çizge, vektör uzayında bir nokta olacak şekilde tasarlanmıştır. Bu durumda gömme işlemi izometrik olarak yapılamamaktadır. Ufak öznitelik kayıpları karşılaştırma işlemlerinin hata oranını artırmaktadır.

Gömme işlemi için uygulanan Tırtıl Ayırışması yönteminde, bir çizge, yaprak sayısı boyutunda vektör ile ifade edilebilmektedir. Bu sebeple her bir çizge, yaprak sayısına bağlı olarak vektör uzayında farklı boyutlarda temsil edilmektedir. Kurgulamış olduğumuz çözüm yaklaşımı ise her çizgenin aynı boyutta temsil edilmesini gerektirmektedir. Çözüm yolu olarak Sıfır Ekleme yapılması kullanılmıştır. Bu sayede öznitelik kaybı olmayacak, her bir noktaya yeni öznitelik verilmesi sonucu oluşabilecek hata ise tolere edilebilir seviyede olacaktır.

Çizgelerde bulunan her bir düğüm ağırlık değerine sahiptir. Bu değer iskeletleme yöntemi esnasında düğüme kazandırılan bir özniteliktir. Tırtıl Ayırışması kenarlar üzerinden işlem yaptığı için düğümlerin ağırlıkları hakkında bir nitelik içermemektedir. Düğüm ağırlıklarının da kaybedilmemesi için her bir düğüm için vektör uzayın ağırlığı kadar nokta eklenmektedir. Bu sayede aynı ağırlık vektör uzayına taşınmış olmaktadır.

Bir diğer önemli nokta ise medyan hesaplanması esnasında her çizgenin aynı oranda etki ediyor olması gereğidir. Tırtıl Ayırışması sonucu her bir çizge, bulundurduğu düğüm sayısı kadar nokta ile temsil edilmiş olacaktır. Bu durumda K-Means uygulaması esnasında düğüm sayısı fazla olan çizgeler oluşturulacak çıktının kendi çizgelerine yakın olmasına sebebiyet verecektir. Çözüm yolu olarak Düğüm Ekleme yapılması kullanılmıştır. Vektör uzayına nokta eklenirken ağırlık oranı kaybedilmeyecek şekilde fazla nokta eklenmektedir.

Vektör uzayında çizge medyanı hesaplanması için K-Means algoritması uygulanmıştır. Elde edilecek her bir merkez nokta etrafında bulunan tüm noktaların ortak bir temsilcisi olmaktadır. Çizgenin içermesi gereken düğüm sayısı kadar ortak nokta bulunması çizge medyanı oluşturulmasını sağlamaktadır. Çizge medyanında bulunan her bir nokta tüm noktaların katkısıyla oluşturulmuş olmaktadır. Bu nedenle K-Means algoritması kullanılmıştır.

Bölüm 4 Deneysel Çalışmalar bölümünde medyan hesaplanmasının analiz çalışmaları verilmiştir. Çizelge 4.1 ile K-Means esnasında geçen süre, adım sayısı ve noktaların özellikleri hakkında bilgiler görülmektedir. Adım sayısı noktaların dağılımıyla alakalı bir değişkendir. Diğer değişkenler çizgelerin durumlarına göre Tırtıl Ayırışması neticesinde ortaya çıkan sonuçlardır. Şekil 4.6 ile gösterilen uygulama görüntüsü örnek bir veri sınıfı için her bir adımın gerçekleştirilmesi için geçen süreyi ve her bir adımda merkezlerin kaç birim değiştiğini göstermektedir.

Yapılan deneysel çalışmalar sonucunda merkezler arasındaki deęişimin “200” birimin altında olması durumunda K-Means uygulamasını sonlandırmanın uygun olduęu belirlenmiştir.

Medyan hesaplanması nokta sayısı ve boyut sayısının fazla olması durumunda oldukça uzun sürede sonuçlar üretmektedir. Ancak bu işlemler bir defaya mahsus olarak yapılan işlemler olduęu için sisteme aşırı yük yüklememektedir. Ayrıca sistemin ölçeklenebilir olması ve her defasında yapılan işlemlerin ve adımların tekrarlanmasına gerek olmaması sistemin başarılı yönlerindedir.

Çizge medyanı hesaplanmasının ardından, çizge medyanı kullanılmasının performansa etkisi ölçülmüştür. Her bir sorgulanan obje için hem tüm objeler ile karşılaştırması hem de sadece her veri sınıfı için çizge medyanı ile karşılaştırması yapılmıştır. Çıkan sonuçlar doğrultusunda medyan kullanımının performansa etkisi belirlenmiştir.

İki nokta kümesinin karşılaştırılması esnasında Hausdorff Mesafesi kullanılmıştır. Bu yaklaşımın kullanıldığı durumda yeterince hassas sonuçların alınmadığı geçmiş çalışmalarda incelenmiş ve çeşitli düzenlemelerle daha hassas sonuçların üretilebildiği 24 farklı yöntem geliştirilmiştir. Bu yöntemlerin 20 tanesi kullanılmış ve 5 farklı yöntem de tarafımızca eklenerek toplam 25 farklı yöntemle Hausdorff Mesafesi kullanılmıştır.

Bu nedenle her karşılaştırma işlemi için 25 farklı sonuç bildirimini yapılmaktadır. Bu nedenle her uzaklık ölçüm yöntemi ayrı olarak değerlendirilmektedir. Her sınıf için doğruluk oranı ise tüm ölçüm yöntemlerinin ortalaması alınarak karar verilmektedir. Bu karşılaştırma yöntemi hem medyan kullanılarak hem de medyan kullanılmadan gerçekleştirilmiştir. Çizelge 4.2 medyan kullanılarak elde edilen doğruluk oranlarının tümünü, Çizelge 4.5 ise medyan kullanılmadan elde edilen doğruluk oranlarının tümünü göstermektedir. Medyan kullanılarak gerçekleştirilen karşılaştırma işlemlerinde en yüksek doğru tahmin oranı “d4-f5” yöntemiyle elde edilmektedir. En yüksek doğruluk oranı “çyd” sınıfıyla elde edilmekte ve tüm sistem için doğruluk oranı %44,91 olmaktadır. Medyan kullanılmadığı durumda ise en yüksek doğru tahmin oranı “d6-f2” yöntemiyle elde edilmektedir. En yüksek doğruluk oranı “saat” sınıfıyla elde edilmekte ve tüm sistem için doğruluk oranı %49,51 olmaktadır. Genel ortalamada medyan kullanılması doğruluk oranında %5-%6 seviyesinde bir gerilemeye neden olmaktadır. Çizelge 4.8 ile tüm uzaklık hesaplama yöntemleri için

iki yaklaşım arasındaki farkın nasıl gerçekleştiği ve Şekil 4.8 ile de aradaki farkın grafiği görülmektedir. Buna karşın Şekil 4.11’de gösterildiği üzere medyan kullanılan yöntemle yaklaşık olarak 1/8 oranında daha hızlı sonuç üretilmektedir. Tolere edilebilir seviyede hata payıyla beraber 8 kat hızdan kazanç yeni yaklaşımın performans açısından oldukça iyi olduğunu göstermektedir.

Tez çalışmasında verilen bir objenin hangi sınıfa ait olduğu sonucunun verilmesi için ikinci bir yaklaşım olan hibrit yaklaşımı kullanılmıştır. Kullanılan 25 farklı Hausdorff Mesafesi yöntemiyle üretilen sonuçlar ışığında nihai kararı verebilmek için tüm yöntemler içinde sonuç olarak en fazla verilen sınıfın bulunması kullanılmaktadır. En fazla sonucu alan sınıf sorgulanan objenin sınıfı olarak döndürülmektedir. Bu yaklaşımın kullanılması için sistemin tüm uzaklık ölçütleri için çalışması gerekmektedir. Çizelge 4.3 ile medyan kullanılan durumda “brd” sınıfı için her bir objenin hangi sınıfa ait olduğunun nasıl hesaplandığı aktarılmıştır. Aynı şekilde Çizelge 4.6 ile medyan kullanılmayan durum için bilgiler aktarılmıştır. Çizelge 4.4 ile medyan kullanılan durumda hibrit yaklaşımın sınıflar bazında verdiği sonuçların diğer yöntemin ortalamasıyla karşılaştırılması verilmiştir. Hibrit yöntem %54,94 ile diğer yöntemin ortalaması olan %44,91’den daha iyi sonuçlar vermektedir. Çizelge 4.7’ye göre bu durum medyansız karşılaştırmada daha belirgin bir şekilde görünmektedir. Medyansız karşılaştırmada hibrit yöntem %61,1 ile diğer yöntemin ortalaması olan %49,50’den daha iyi sonuçlar vermektedir.

Tüm sonuçların net bir şekilde görüldüğü Çizelge 4.9’da tüm sınıfların kullanılan tüm yöntemlere göre ortalamaları verilmektedir. Hibrit yönteminin kullanılması daha iyi sonuçları beraberinde getirirken, medyan kullanılması ve kullanılmaması arasında %6-%7 civarında bir farklılık bulunmaktadır. Şekil 4.9 hibrit kullanıldığı durumda medyan kullanılması ve kullanılmaması arasındaki performans farkını göstermektedir. Hibrit yöntemi kullanılmaması durumunda ise medyan kullanılması ile kullanılmaması arasındaki farklılık %5-%6 aralığındadır. Şekil 4.10 ile hibrit kullanılmadığı durumda medyan kullanılması ve kullanılmaması halinde performans farkının nasıl değişeceğini göstermektedir.

Tüm bu sonuçlar ve grafikler ışığında, oluşturulan medyan ile karşılaştırmaların yapılması Şekil 4.11’de net bir şekilde görüldüğü gibi yaklaşık olarak 1/8 oranında zaman kazancı sağlamaktadır. Bunun yanında hata oranındaki azalma göz ardı edilebilir bir seviye olan %5-%7 aralığındadır. Bu sonuçlar oluşturulan medyan

nokta kümesinin başarısının oldukça yüksek olduğunu vermektedir. Yapılan hız çalışmalarında medyan hesaplanması için geçen süre göz ardı edilmektedir. Bunun sebebi medyan hesaplanmasının sorgulama esnasında çalışmaması ve veri kümesinde olabilecek değişikliklere hızlı bir şekilde adapte olabilir olmasıdır.

## 5.2. Gelecek Çalışmalar

Yapılan çalışma neticesinde elde edilen kazanç oldukça iyi seviyededir. Ancak bu performansın başarımının ölçülebilmesi için bir başka medyan hesaplama analizi ile karşılaştırma işlemi yapılması gerekmektedir. Daha önce yapılan çalışmalardan birinin kullanılmasıyla elde edilecek olan çizge medyanı ile aynı veri kümesi kullanılarak karşılaştırma işlemlerinin yapılması, yeni oluşturulan yaklaşımın diğer yöntemlerden farkını gösterebilecektir.

Objelerin çizgelere çevrilmesi aşamasında iskeletleme yöntemi kullanılmaktadır. İskeletleme yöntemi kapsamında minimum kapsama ağacı uygulaması yapılmaktadır. Mevcut algoritmaların çıktıları tekil olmadığı için oluşturulabilecek her bir çizge birbirine çok uzak iki çizge olabilmektedir. Tekil çıktı verebilen minimum kapsama ağacı oluşturulabilecek yeni bir yaklaşım bu tarz problemlerin önüne geçebilecektir. Bu sayede çok daha hassas ve doğru sonuçlar elde edilebilecektir.

Tüm boyutların aynı seviyeye çekilmesi esnasında kullanılan Sıfır Ekleme yöntemi ile aslında verilen vektörlere yeni bir öznitelik katılmış olmaktadır. Dolayısıyla bir çeşit veri kaybı yaşanmış olmaktadır. Buradaki kayıp tolere edilebilecek seviyede olsa da kullanılabilir başka bir yöntem ile hata oranı düşürülebilecektir.

Çizge medyanı oluşturulurken en fazla düğüme sahip ağaç sayısı kadar merkez olduğu öngörülmektedir. Dinamik merkez sayısı yöntemiyle tüm noktaların genel davranışları doğrultusunda oluşturulacak çizgede kaç merkez olması gerekiyorsa ona göre geliştirilen bir medyan hesabı yapılabileceği düşünülmektedir. Bu sayede daha hassas sonuçlar alınabilecektir.

Ayrıca düğümlerin vektör uzayında ifade edilebilir olması için her bir düğüm vektör uzayına geçirilirken ağırlığınca nokta olarak vektör uzayına gömülmektedir. Bu işlem K-Means esnasında yapılan işlemleri negatif yönde etkilemekte ve sisteme aşırı yük bindirmektedir. Bu yöntemin yerine iki nokta karşılaştırılırken ağırlıklar

göz önünde bulundurularak yapılacak olan bir karşılaştırma ile performans artırımı sağlanabilecektir.

Kullanılan Hausdorff Mesafesi yerine iki nokta kümesi arasındaki mesafenin tespit edilebileceği başka bir algoritma kullanılabilir. Bu sayede iki veya üç farklı algoritma ile de medyan hesaplanmasının performansı desteklenmiş olacaktır.

## KAYNAKLAR

- [1] **Demirci** M. F., Osmanlioglu Y., Shokoufandeh A. & Dickinson S. “Efficient many-to-many feature matching under the  $\ell_1$  norm”. Computer Vision and Image Understanding, 2011.
- [2] **Ferrer** M., Valveny E., Serratoso F., Riesen K. & Bunke H. “An approximate algorithm for median graph computation using graph embedding”. Pattern Recognition, 2008.
- [3] **Dubuisson** M.P. and Jain A.K. “A Modified Hausdorff Distance for Object Matching”. Internation Conference on Pattern Recognition, Jerusalem, Israel : 565-568, 1994.
- [4] **Belongie** S., Malik J. & Puzicha J. "Shape matching and object recognition using shape contexts." Pattern Analysis and Machine Intelligence, IEEE Transactions on 24(4): 509-522, 2002.
- [5] **Demirci** F., Osmanlioglu Y. “Many-to-many matching under the  $\ell_1$  norm” Proceedings of the 15th International Conference on Image Analysis and Processing, Springer-Verlag, Berlin, Heidelberg, pp. 787–796, 2009.
- [6] **Bunke** H. & Shearer K.. "A graph distance metric based on the maximal common subgraph." Pattern recognition letters 19(3): 255-259, 1998.
- [7] **Deo** N. “Graph Teory with Applications to Engineering and Computer Science” ISBN-81-203-0145-5, 1974.
- [8] **Aittokallio** T. and Schwikowski B. “Graph-based methods for analysing networks in cell biology”. Brief. Bioinform. 7:243-255, 2006.
- [9] **Airola** A., Pyysalo S., Björne J., Pahikkala T., Ginter F. and Salakoski T. “All-paths graph kernel for protein-protein interaction extraction with evaluation of cross-corpus learning”. BMC Bioinformatics 9(Suppl 11):S2, 2008.
- [10] **Applegate** D. L., Bixby R. M., Chvátal V., Cook W. J. “The Traveling Salesman Problem”, ISBN 0-691-12993-2, 2006.



- [11] **Siddiqi** K., Shokoufandeh A., Dickinson S. J. and . Zucker S. W. "Shock graphs and shape matching." International Journal of Computer Vision 35(1): 13-32, 1999.
- [12] <http://www.ics.uci.edu/~eppstein//161/960206.html>
- [13] **Prim**, R. C. "Shortest connection networks and some generalizations." Bell system technical journal 36(6): 1389-1401,
- [14] **Kruskal**, J. B. "On the shortest spanning subtree of a graph and the traveling salesman problem." Proceedings of the American Mathematical society 7(1): 48-50, 1956.
- [15] **Gallager** R. G., Humblet P. A. & Spira P. M. "A distributed algorithm for minimum-weight spanning trees." ACM Transactions on Programming Languages and systems (TOPLAS) 5(1): 66-77, 1983.
- [16] **Jiang** X., Munger A. & Bunke H. An median graphs: properties, algorithms, and applications." Pattern Analysis and Machine Intelligence, IEEE Transactions on 23(10): 1144-1151, 2001.
- [17] **Tan** P. N., Steinbach M. & Kumar V. "Introduction to data mining", Pearson Addison Wesley Boston, 2006.
- [18] **Deza** M.M. and Deza E. "Encyclopedia of Distance", ISBN 978-3-642-00233-5, Springer,Verlag, Berlin Heidelberg, 2009
- [19] **Zadeh**, L. A. "Fuzzy sets." Information and control 8(3): 338-353, 1965.
- [20] **Chuang**, K. S., Tzeng H. L., Chen S., Wu J. & Chen T. J. "Fuzzy c-means clustering with spatial information for image segmentation." Computerized Medical Imaging and Graphics 30(1): 9-15, 2006.
- [21] **Hartigan** J. A. and Wong M. A. "Algorithm AS 136: A K-Means Clustering Algorithm". Journal of the Royal Statistical Society. Series C (Applied Statistics) Vol. 28, No. 1: 100-108, 1979.
- [22] **Shapiro** L. G. & Haralick R. M. "Structural descriptions and inexact matching." Pattern Analysis and Machine Intelligence, IEEE Transactions on(5): 504-519, 1981.

- [23] **Ullmann**, J. R. "An algorithm for subgraph isomorphism." *Journal of the ACM (JACM)* 23(1): 31-42, 1976.
- [24] **Williams**, M. L., Wilson R. C. & Hancock E. R. "Deterministic search for relational graph matching." *Pattern Recognition* 32(7): 1255-1272, 1999.
- [25] **Bunke**, H. "Error correcting graph matching: On the influence of the underlying cost function." *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 21(9): 917-922, 1999.
- [26] **Wang** Y. K., Fan K. C. & Horng J. T. "Genetic-based search for error-correcting graph isomorphism." *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on* 27(4): 588-597.
- [27] **Kittler** J., Christmas W. and Petrou M. "Probabilistic relaxation for matching of symbolic structures." *Advances in Structural and Syntactic Pattern Recognition*: 471-480, 1992.
- [28] **Almohamad** H., and S. Duffuaa. "A linear programming approach for the weighted graph matching problem." *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 15(5): 522-525, 1993.
- [29] **Bengoetxea** E., Larranaga P., Bloch I., Perchant A. and Boeres C. "Inexact graph matching by means of estimation of distribution algorithms" *Pattern Recognition* 35 (14): 2867–2880, 2002.
- [30] **Demirci** F., Shokoufandeh A., Keselman Y., Bretzner L. and Dickinson S. "Object recognition as many-to-many feature matching" *International Journal of Computer Vision* 69(2): 203–222, 2006.
- [31] **Linial** N., London E. and Rabinovich Y. "The geometry of graphs and some of its algorithmic applications" *Proceedings of the 35th Annual Symposium on Foundations of Computer Science, Washington, DC, USA, IEEE Computer Society*, pp. 577–591, 1994.
- [32] **Rubner** Y., Tomasi C. and Guibas L.J. "The earth mover's distance as a metric for image retrieval" *International Journal of Computer Vision* 40 (2): 99–121, 2000.

- [33] **Matoušek** J. “On embedding trees into uniformly convex Banach spaces” *Israel Journal of Mathematics* 237 : 221–237, 1999.
- [34] **Ferrer** M., **Serratos** F. and **Sanfeliu** A. “Synthesis of median spectral graph” *IbPRIA 2005*, volume 3523 of LNCS, pages 139–146. Springer, 2005.
- [35] **Hlaoui** A. and **Wang** S. “Median graph computation for graph clustering”. *Soft Comput.*, 10(1): 47–53, 2006.
- [36] **M“unger** A. “Synthesis of prototype graphs from sample graphs” In *Diploma Thesis, University of Bern (in German)*, 1998.
- [37] **Riesen** K., **Neuhaus** M. and **Bunke** H. “Graph embedding in vector spaces by means of prototype selection” *GbRPR 2007 Proceedings*, volume 4538 of LNCS, pages 383–393. Springer, 2007.
- [38] **Bunke** H. and **G“unter** S. “Weighted mean of a pair of graphs” *Computing*, 67(3):209–224, 2001.
- [39] **Huttenlocher** D. P., **Klanderman** G. A. and **Rucklidge** W. J. “Comparing images using the Hausdorff distance”, *IEEE Trans. PAMI*, vol. 15, pp. 850–863, 1993.
- [40] **Gupta** A. “Embedding tree metrics into low-dimensional euclidean spaces” *Discrete & Computational Geometry* 24 (1):105–116, 2000.
- [41] **Shafeeq** A. and **Hareesha** K. S. “Dynamic Clustering of Data with Modified K-Means Algorithm” *International Conference on Information and Computer Networks* vol. 27, 2012