



**BULANIK KÜMELER İÇİN
UZAKLIK VE BENZERLİK ÖLÇÜLERİNİN
BAZI MATLAB UYGULAMALARI**

YÜKSEK LİSANS TEZİ

Özge KAYA

Danışman

Dr. Öğr. Üyesi Esra GÜLLE

MATEMATİK ANABİLİM DALI

Temmuz 2023

Bu tez çalışması 22.FEN.BİL.45 numaralı proje ile Afyon Kocatepe Üniversitesi
Bilimsel Araştırmalar Projeleri (BAP) tarafından desteklenmiştir.

AFYON KOCATEPE ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

YÜKSEK LİSANS TEZİ

**BULANIK KÜMELER İÇİN UZAKLIK VE BENZERLİK
ÖLÇÜLERİNİN BAZI MATLAB UYGULAMALARI**

Özge KAYA

Danışman

Dr. Öğr. Üyesi Esra GÜLLE

MATEMATİK

Temmuz 2023

TEZ ONAY SAYFASI

Özge KAYA tarafından hazırlanan “Bulanık Kümeler için Uzaklık ve Benzerlik Ölçülerinin Bazı Matlab Uygulamaları” adlı tez çalışması lisansüstü eğitim ve öğretim yönetmeliğinin ilgili maddeleri uyarınca 19/07/2023 tarihinde aşağıdaki jüri tarafından **oy birliği** ile Afyon Kocatepe Üniversitesi Fen Bilimleri Enstitüsü **Matematik Anabilim Dalı**’nda **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Danışman : Dr. Öğr. Üyesi Esra GÜLLE

Başkan : Prof. Dr. Erdiñ Dündar
Afyon Kocatepe Üniversitesi
Fen Edebiyat Fakültesi

Üye : Doç. Dr. Uğur Ulusu
Sivas Cumhuriyet Üniversitesi
Cumhuriyet Sosyal Bilimler MYO

Üye : Dr. Öğr. Üyesi Esra Gülle
Afyon Kocatepe Üniversitesi
Fen Edebiyat Fakültesi

Afyon Kocatepe Üniversitesi
Fen Bilimleri Enstitüsü Yönetim Kurulu’nun
..... /..... /..... tarih ve
..... sayılı kararıyla onaylanmıştır.

.....

Prof. Dr. İbrahim EROL
Enstitü Müdürü

BİLİMSEL ETİK BİLDİRİM SAYFASI
Afyon Kocatepe Üniversitesi

Fen Bilimleri Enstitüsü, tez yazım kurallarına uygun olarak hazırladığım bu tez çalışmada;

- Tez içindeki bütün bilgi ve belgeleri akademik kurallar çerçevesinde elde ettiğimi,
- Görsel, işitsel ve yazılı tüm bilgi ve sonuçları bilimsel ahlak kurallarına uygun olarak sunduğumu,
- Başkalarının eserlerinden yararlanılması durumunda ilgili eserlere bilimsel normlara uygun olarak atıfta bulunduğumu,
- Atıfta bulunduğum eserlerin tümünü kaynak olarak gösterdiğimi,
- Kullanılan verilerde herhangi bir tahrifat yapmadığımı,
- Ve bu tezin herhangi bir bölümünü bu üniversite veya başka bir üniversitede başka bir tez çalışması olarak sunmadığımı

beyan ederim.

19/07/2023

Özge KAYA

ÖZET

Yüksek Lisans Tezi

BULANIK KÜMELER İÇİN UZAKLIK VE BENZERLİK ÖLÇÜLERİNİN BAZI MATLAB UYGULAMALARI

Özge KAYA

Afyon Kocatepe Üniversitesi

Fen Bilimleri Enstitüsü

Matematik Anabilim Dalı

Danışman: Dr. Öğr. Üyesi Esra GÜLLE

Bu çalışma beş ana bölümden oluşmaktadır.

Birinci bölümde, tez konusunun önemini ve tarihi geçmişini anlatan genel bir literatür bilgisi verilmiştir. İkinci bölümde, çalışma için gerekli bulanık küme kavramı, bulanık kümeler için temel kavramlar, bazı uzaklık ve benzerlik ölçüleri verilmiştir. Üçüncü bölümde, Matlab programında Sobel, Prewitt, Laplacian, Canny ve Robert Cross filtreleri açıklanmış, daha sonra bu filtreler bulanık mantık kullanılarak uygulanmıştır. Ayrıca, kenar belirleme filtrelerinin Hamming, Öklid ve Minkowski uzaklık ölçüleri yardımıyla elde edilen benzerlik oranları verilmiştir. Dördüncü bölümde, yapılan çalışmada sunulan bilgiler değerlendirilmiştir.

Son bölüm olan beşinci bölümde ise, çalışma süresince yararlanılan literatürdeki kaynaklar listelenmiştir.

2023, viii + 40 sayfa

Anahtar Kelimeler: Bulanık küme, Üyelik fonksiyonları, Bulanık uzaklık ölçüleri, Bulanık benzerlik ölçüleri, Matlab.

ABSTRACT

M.Sc. Thesis

SOME MATLAB APPLICATIONS OF DISTANCE AND SIMILARITY MEASURES FOR FUZZY SETS

Özge KAYA

Afyon Kocatepe University

Graduate School of Natural and Applied Sciences

Department of Mathematics

Supervisor: Asst. Prof. Esra GÜLLE

This thesis consists of five main chapters.

In the first chapter, a general literature information about the importance and historical background of the thesis subject is given. In the second part, the concept of fuzzy set, the basic concepts of fuzzy sets, some measures of distance and similarity, that are required for the study, are given. In the third chapter, Sobel, Prewitt, Laplacian, Canny and Robert Cross filters are explained in Matlab program, then these filters are applied using fuzzy logic. Additionally, similarity ratios obtained using Hamming, Euclidean and Minkowski distance measures for edge detection filters are given. In the fourth chapter, the information presented in the study is evaluated.

In the fifth section, which is the last chapter, the sources in the literature that we use during our study are listed.

2023, viii + 40 pages

Keywords: Fuzzy set, Membership functions, Fuzzy distance measures,
Fuzzy similarity measures, Matlab.

TEŞEKKÜR

Yüksek Lisans tezimin araştırma konusunun verilmesi, çalışmaların yönlendirilmesi, sonuçların değerlendirilmesi ve yazımı aşamasında yapmış olduğu büyük katkılarından dolayı danışman hocam Sayın Dr. Öğr. Üyesi Esra GÜLLE'ye, araştırma ve yazım süresince her konuda öneri ve yardımlarını esirgemeyen Sayın Arş. Gör. Başak ALDEMİR'e teşekkür ederim.

Yüksek lisans eğitimine başladığım günden beri çalışmalarımın her aşamasında sevgi ve desteğiyle yanımda olan annem ve babama sonsuz teşekkürlerimi ve sevgilerimi sunarım.

Özge KAYA
Afyonkarahisar 2023

İÇİNDEKİLER DİZİNİ

	Sayfa
ÖZET	i
ABSTRACT	ii
TEŞEKKÜR	iii
İÇİNDEKİLER DİZİNİ.....	iv
SİMGELER DİZİNİ.....	vi
ŞEKİLLER DİZİNİ.....	vii
TABLolar DİZİNİ.....	viii
1. GİRİŞ.....	1
2. BULANIK KÜMELER.....	3
2.1 Bulanık Kümelerde Temel Küme İşlemleri ve Özellikleri.....	4
2.1.1 İki Bulanık Kümenin Kesişimi.....	4
2.1.2 İki Bulanık Kümenin Birleşimi.....	5
2.1.3 İki Bulanık Kümenin Farkı	6
2.1.4 İki Bulanık Kümenin Eşitliği	7
2.1.5 Bulanık Kümelerin Özellikleri.....	7
2.2 Bulanık Sayılar	8
2.2.1 Üçgen Bulanık Sayılar	8
2.2.2 Yamuk Bulanık Sayılar	9
2.3 Bulanık Kümeler için Uzaklık ve Benzerlik Ölçüleri.....	11
2.3.1 Bulanık Kümeler İçin Uzaklık Tabanlı Benzerlik Ölçüleri	12
3. BULANIK KENAR BELİRLEME	15
3.1 Matlab	15
3.2 Görüntü Türleri.....	16
3.2.1 İkili Görüntüler.....	16
3.2.2 Gri Seviyede Görüntüler	17
3.2.3 RGB (Renkli) Görüntüler.....	17
3.3 Görüntü İşleme ve Filtreleme	18
3.3.1 Kenar Belirleme Filtreleri	19
3.3.1.1 Sobel Filtresi.....	19
3.3.1.2 Prewitt Filtresi	22
3.3.1.3 Laplacian (Laplasyan) Filtresi.....	23
3.3.1.4 Canny Filtresi	24

3.3.1.5 Robert Cross (Roberts) Filtresi.....	27
3.3.2 Bulanık Mantık ile Kenar Belirleme (Filtreleme).....	29
3.3.3 Kenar Belirleme Filtrelerinin Benzerlik Oranları	32
4. TARTIŞMA ve SONUÇ	37
5. KAYNAKLAR.....	38
ÖZGEÇMİŞ.....	Hata! Yer işareti tanımlanmamış.



SİMGELER DİZİNİ

Simgeler	
$\in$	Eleman
$\cap$	Kesişim
$\cup$	Birleşim
$\setminus$	Fark
$\mathbb{R}$	Reel sayılar kümesi
$F(X)$	X kümesi üzerindeki tüm fuzzy ailesi
$\mathcal{P}(X)$	X kümesinin kuvvet kümesi
BMS	Bulanık mantık ile kenar belirme uygulanarak elde edilen görüntü



ŞEKİLLER DİZİNİ

	Sayfa
Şekil 2.1 İki bulanık kümenin kesişiminin grafiği	5
Şekil 2.2 İki bulanık kümenin birleşiminin grafiği	6
Şekil 2.3 İki bulanık kümenin farkının grafiği	7
Şekil 2.4 Üçgen bulanık sayı	9
Şekil 2.5 Yamuk bulanık sayı	10
Şekil 2.6 Üyelik dereceleri	10
Şekil 2.7 Bulanık sayıların denklemini kullanarak boy hesaplama	11
Şekil 3.1 Sobel filtresinin Matlab programındaki kodları	21
Şekil 3.2 Sobel filtresinin Matlab programındaki sonuçları	21
Şekil 3.3 Prewitt filtresinin Matlab programındaki kodları	22
Şekil 3.4 Prewitt filtresinin Matlab programındaki sonuçları	23
Şekil 3.5 Laplacian filtresinin Matlab programındaki kodları	24
Şekil 3.6 Laplacian filtresinin Matlab programındaki sonuçları	24
Şekil 3.7 Canny filtresinin Matlab programındaki kodları	26
Şekil 3.8 Canny filtresinin Matlab programındaki sonuçları	27
Şekil 3.9 Robert Cross filtresinin Matlab programındaki kodları	28
Şekil 3.10 Robert Cross filtresinin Matlab programındaki sonuçları	29
Şekil 3.11 Görüntüyü içe aktarma	30
Şekil 3.12 RGB görüntünün gri tonlamaya dönüştürülüp bulanıklaştırılması	30
Şekil 3.13 Gradyan hesaplanması	31
Şekil 3.14 Kenar belirleme ve üyelik derecelerinin oluşturulması	31
Şekil 3.15 Bulanık mantık kullanılarak elde edilen kenar görüntüsü	32

TABLÖLAR DİZİNİ

	Sayfa
Tablo 3.1 Klasik filtreler arasındaki benzerlik oranlarının elde edilmesi.....	33
Tablo 3.2 Klasik filtre ve BMS benzerlik oranlarının elde edilmesi	34
Tablo 3.3 Klasik filtreler arasındaki benzerlik oranlarının elde edilmesi.....	35
Tablo 3.4 Klasik filtre ve BMS benzerlik oranlarının elde edilmesi	36



1. GİRİŞ

Klasik bilimsel mantığa uyarlanmış bir matematiksel yaklaşım, en keskin sınırlarla kullanımı en kolay yöntemdir. İhtiyaca uygun çözümler üretmek için çeşitli matematiksel modeller oluşturulmuştur. Bu modeller, düşünceyi, dili ve duyguları modellemek için kapsamlı bir yaklaşım değildir. Doğada karşılaşılan zorluklar ve ikilemler, ortaya çıkan belirsizliklerin yeni yöntemlerle modellenmesine kapı aralamıştır. Bu yöntemlerin en önde geleni olan bulanık mantık kavramı, Aristoteles iki değerli mantığına karşı ortaya çıkmıştır. Aristoteles'in kümeler teorisine göre bir x elemanı kümeye dahildir veya değildir. Kümeye dahil olmayan eleman o kümenin elemanı olamaz. Dahil olan ve olmayan elemanların kesişimi boş küme, birleşimi evrensel kümedir. Bulanık mantık, yaşlı, genç, uzun, kısa, sıcak, soğuk, ılık, çok, az, biraz, fazla, çok az, çok fazla gibi konuşma diline uygun belirsizlik değişkenlerini ifade ederek bu değişkenler için "0" ve "1" değerleri yerine 0.24, 0.7 gibi ara değerli sonuçlar sağlamaktadır.

Zadeh (1965) orijinal makalesinde klasik mantık kurallarını kullanarak bulanık mantık ve bulanık küme teorisini geliştirip yayınladığında, belirsizliğe sahip sistemlerin ele alınması yeni bir boyut kazandı. Bulanık mantık ve bulanık küme teorisi farklı birçok disiplinde önemli bir yere sahiptir. Matematikte analiz ve fonksiyonlar teorisi bilim dalında Kamber (2017), Talo ve Yavuz (2021) gibi birçok araştırmacının bulanık sayılar için yakınsaklık ve toplanabilme çeşitleri üzerine, topoloji bilim dalında Çitli (2006), Karakaşoğlu (2008) gibi birçok araştırmacının ise bulanık kümeler için karar verme yöntemleri üzerine yaptıkları teorik çalışmaların yanında mühendislik, robotik gibi disiplinlerde de Yang ve Xu (2011), Utama vd. (2022) gibi araştırmacıların farklı programlar kullanarak yaptıkları uygulamalı çalışmalar mevcuttur.

Xuecheng (1992) uzaklık ve benzerlik ölçülerinin temel aksiyomlarını tanıtmış, bu ölçüler arasındaki ilişkiyi örneklerle incelemiştir. Setnes vd. (1998), Omran ve Hassaballah (2006), Beg ve Ashraf (2009), Verma (2015) başta olmak üzere birçok araştırmacı ise uzaklık ve benzerlik ölçülerini bulanık mantık kavramını kullanarak genelleştirmiştir.

Alshennawy ve Aly (2009), Aborisade (2010), Kushwaha vd. (2012), Couso vd. (2013), Kumawat ve Panda (2022) olmak üzere birçok araştırmacı tarafından bulanık mantıkta kenar belirleme yöntemlerini kullanarak çalışmalar yapılmıştır.

İki bulanık kümenin uzaklık ölçüsü, bulanık kümeler arasındaki farkı tanımlayan bir ölçüdür. Ayrıca, iki bulanık kümenin benzerlik ölçüsü bulanık kümeler arasındaki benzerliği gösterir. Uzaklık ve benzerlik fonksiyonları, kümeler arasındaki fark ve benzerlik derecelerini ölçmek için kullanılan matematiksel fonksiyonlardır.

Bu çalışmanın amacı, birçok alanda önemli bir yere sahip olan bulanık küme yapısını incelemek ve bulanık küme üzerinde tanımlanan farklı uzaklık ve benzerlik ölçülerini tanıtmaktır. Ayrıca matematiğin birçok dalında kullanılan Matlab programının bir uygulaması olan kenar belirleme yöntemleri ele alınarak tanıtılan uzaklık ve benzerlik ölçüleri yardımıyla bu yöntemler arasındaki benzerlikleri ortaya koymak amaçlanmıştır.

2. BULANIK KÜMELER

Bu bölümde ilk olarak Zadeh (1965) tarafından verilen ve bulanık küme teorisi için temel oluşturan bulanık küme kavramı tanıtılacaktır. Ayrıca, Klir ve Yuan (1995) tarafından verilen bulanık küme kavramının temel özellikleri incelenecektir. Daha sonra bulanık kümeler için uzaklık ve benzerlik ölçüleri kavramlarından bahsedilecektir.

Bulanık küme teorisi, tanımlanması veya anlaşılması zor olan, kesinlikten söz edilemeyen belirsizliğin olduğu durumları açıklamayı amaçlamaktadır. Modern mantıkta, bir kümenin elemanları belirli elemanlardır. Bu eleman kümenin üyesidir ya da değildir. Bu tür elemanlardan oluşan kümelere keskin (crisp) kümeler denir. “Ya hep ya hiç” mantığının ötesinde bir görüşe dayanan bulanık kümeler, bir elemanın kümeye ait olmasını derecelendiren üyelik fonksiyonu ile karakterize edilirler. Yani klasik kümeden farklı olarak bulanık kümede kümenin elemanları “0” veya “1” değerleri yerine 0 ile 1 arasında herhangi bir değer alabilir. Bu durumu bir örnekle açıklayalım: “hava soğuk” denildiğinde birçok kişi günlük hayattaki kullanımını anlamaktadır ancak “soğuk” kelimesinin ifade ettiği anlam göreceli olarak birbirinden farklı olabilmektedir. Afyonkarahisar’da bulunan bir kişinin soğukluk değeri -2°C algılamasına karşın, İzmir’de bulunan bir kişi için bu değer 10°C olabilir. Arada bulunan birçok kişinin görüşü olarak başka dereceler de olduğu söylenebilmektedir. Bu yüzden “soğuk” kelimesinin altında insanların ima ettiği sayısal anlayışın bir sonucu olarak belirsiz bir durum ortaya çıkar. Bu rastgele değildir, ancak belirsizdir. İşte bu şekilde kelimelerin ima ettikleri belirsizliklere bulanıklık (fuzzy) denir.

Tanım 2.1 X boştan farklı bir küme olmak üzere X kümesinde ki

$$\mu_A: X \rightarrow [0,1]$$

biçiminde tanımlanan üyelik fonksiyonu ile karakterize edilen bir A kümesine bulanık küme denir. Burada $\mu_A(x)$ değerine üyelik derecesi denir (Zadeh 1995).

Bir A bulanık kümesi

$$A = \{(x, \mu_A(x)): x \in X \text{ ve } \mu_A(x) \in [0,1]\}$$

biçiminde bir eleman ile bu elemanın ilgili kümeye üyelik derecesini gösteren sıralı

ikililer halinde gösterilir. Ayrıca, A bulanık kümesi

$$A = \left\{ \mu_A(x)/x : x \in X \text{ ve } \mu_A(x) \in [0,1] \right\}$$

biçimde de ifade edilebilir.

Örnek 2.1 Bir oteldeki odaları, bu odaların konforunu içinde bulunan yatak sayısı gösterecek biçimde bulanık küme teorisini kullanarak sınıflandırılım. $X = \{1,2, \dots, 10\}$ oteldeki yatakların sayısı ile tanımlanmış odaların bir kümesi olmak üzere “beş kişilik bir aile için rahat oda tipi” bulanık kümesi

$$A = \{(1, 0.2), (2, 0.4), (3, 0.6), (4, 0.8), (5, 1), (6, 0.7), (7, 0.3), (8, 0), (9, 0), (10, 0)\}$$

biçiminde tanımlanabilir.

Tanım 2.2 Her $x \in X$ için $\mu_A(x) = 0$ oluyorsa A bulanık kümesine boş küme denir.

2.1 Bulanık Kümelerde Temel Küme İşlemleri ve Özellikleri

Keskin (crisp) kümeler üzerinde tanımlanan temel küme işlemlerinden kesişim ve birleşim işlemleri her bir elemanın belirli bir derecede o kümeye ait olduğu bulanık kümelerde minimum ve maksimum fonksiyonları kullanılarak tanımlanmıştır. Ayrıca, bulanık kümelerin fark, kapsama ve eşitlik işlemleri ise elemanların üyelik derecesi kullanılarak ifade edilmiştir.

Bu bölümde, X bir evrensel küme ve A, B ve C bu kümenin bulanık alt kümeleri olarak seçilecektir.

2.1.1 İki Bulanık Kümenin Kesişimi

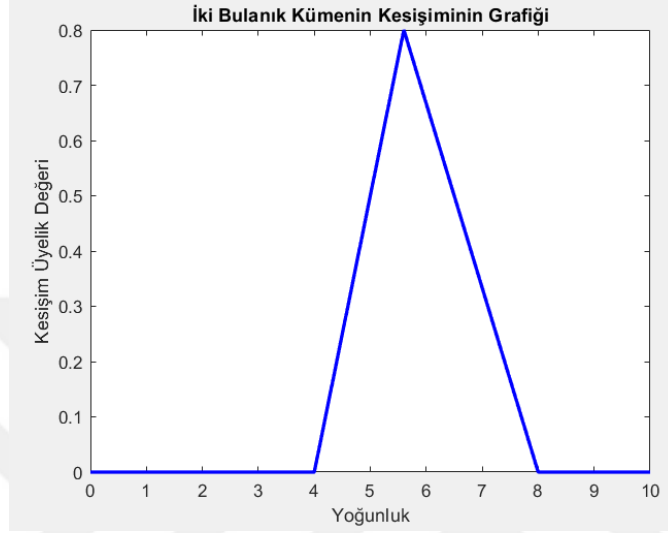
A ve B bulanık kümelerinin üyelik fonksiyonları sırasıyla μ_A ve μ_B olsun. Bu takdirde, A ve B bulanık kümelerinin kesişimi de bir bulanık kümedir ve $A \cap B$ ile gösterilir.

$A \cap B$ bulanık kümesinin üyelik fonksiyonu her $x \in X$ için

$$\min\{\mu_A(x), \mu_B(x)\}$$

biçiminde tanımlanır.

Bir başka deyişle iki bulanık kümenin kesişimi, her iki kümeye ait olan elemanların en küçük üyelik derecesini alan yeni bir bulanık kümeyi ifade eder. Eğer ortak eleman yoksa kesişim kümesi boş küme ve dolayısıyla üyelik derecesi sıfırdır.



Şekil 2.1 İki bulanık kümenin kesişiminin grafiği.

Örnek 2.1.1.1 $A = \{0.3/1, 0.5/3, 0.7/5\}$ ve $B = \{0.2/1, 0.4/3, 0.6/4\}$ bulanık kümelerinin kesişimi

$$A \cap B = \{\min\{0.3, 0.2\}/1, \min\{0.5, 0.4\}/3\} = \{0.2/1, 0.4/3\}$$

biçimindedir.

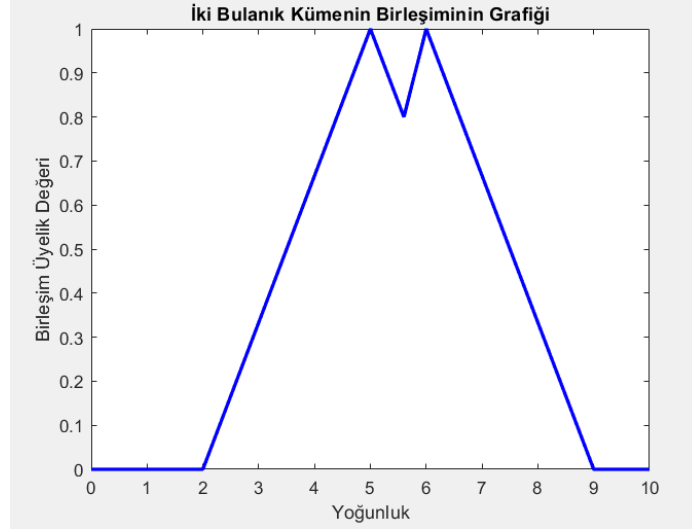
2.1.2 İki Bulanık Kümenin Birleşimi

A ve B bulanık kümelerinin üyelik fonksiyonları sırasıyla μ_A ve μ_B olsun. Bu takdirde, A ve B bulanık kümelerinin birleşimi de bir bulanık kümedir ve $A \cup B$ ile gösterilir.

$A \cup B$ bulanık kümesinin üyelik fonksiyonu her $x \in X$ için

$$\max\{\mu_A(x), \mu_B(x)\}$$

biçiminde tanımlanır. Bir başka deyişle iki bulanık kümenin birleşimi, her iki kümeye ait olan elemanların en büyük üyelik derecesini alan yeni bir bulanık kümeyi ifade eder.



Şekil 2.2 İki bulanık kümenin birleşiminin grafiği.

Örnek 2.1.2.1 $A = \{0.3/1, 0.5/3, 0.7/4\}$ ve $B = \{0.2/1, 0.4/3, 0.6/4, 0.5/5\}$ bulanık kümelerinin birleşimi

$$\begin{aligned} A \cup B &= \{\max\{0.3, 0.2\}/1, \max\{0.5, 0.4\}/3, \max\{0.7, 0.6\}/4, 0.5/5\} \\ &= \{0.3/1, 0.5/3, 0.7/4, 0.5/5\} \end{aligned}$$

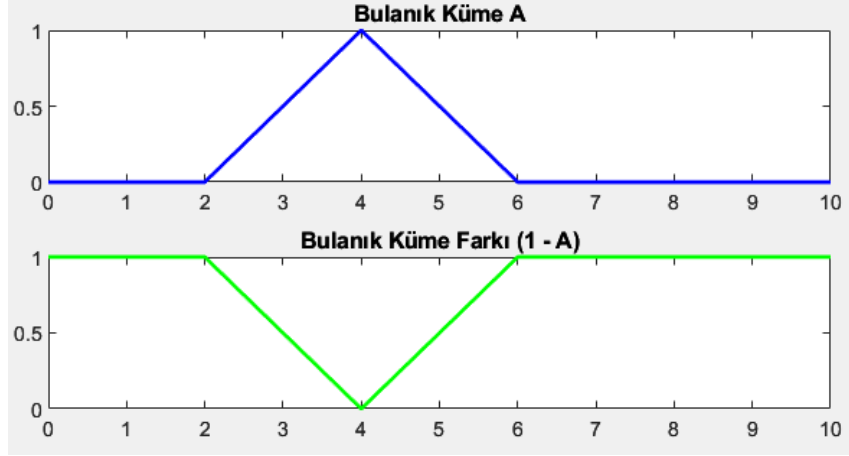
biçimindedir.

2.1.3 İki Bulanık Kümenin Farkı

A bulanık kümesinin üyelik fonksiyonu μ_A olsun. Bu takdirde, A bulanık kümesinin farkı da bir bulanık kümedir ve $X \setminus A$ ile gösterilir. $X \setminus A$ bulanık kümesinin üyelik fonksiyonu her $x \in X$ için

$$1 - \mu_A(x)$$

biçiminde tanımlanır.



Şekil 2.3 İki bulanık kümenin farkının grafiği.

Örnek 2.1.3.1 $A = \{0.3/1, 0.5/3, 0.7/5\}$ bulanık kümesinin farkı

$$\{0.7/1, 0.5/3, 0.3/5\}$$

biçimindedir.

2.1.4 İki Bulanık Kümenin Eşitliği

A ve B bulanık kümelerinin üyelik fonksiyonları sırasıyla μ_A ve μ_B olsun. Eğer her $x \in X$ için

$$\mu_A(x) \leq \mu_B(x)$$

oluyorsa A bulanık kümesi B bulanık kümesinin alt kümesidir denir. Eğer her $x \in X$ için

$$\mu_A(x) = \mu_B(x)$$

oluyorsa A bulanık kümesi B bulanık kümesine eşittir denir.

2.1.5 Bulanık Kümelerin Özellikleri

Bulanık kümeler, keskin (crisp) kümeler için bilinen küme işlemlerinin özelliklerinden iki tanesi hariç hepsini sağlamaktadırlar. Keskin kümelerde geçerli olan

$$A \cup A' = X \text{ ve } A \cap A' = \emptyset$$

özellikleri bulanık kümelerde geçerli olmayan ve bulanık küme teorisini ayırt edici karakterizasyonunu ortaya koyan özelliklerdir.

- $A \cup B = B \cup A$, $A \cap B = B \cap A$, (Değişme Özelliği)
- $A \cup (B \cap C) = (A \cup B) \cap C$, $A \cap (B \cup C) = (A \cap B) \cup C$ (Birleşme Özelliği)
- $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$,
 $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$ (Dağılma Özelliği)
- $\overline{(A \cup B)} = \bar{A} \cap \bar{B}$, $\overline{(A \cap B)} = \bar{A} \cup \bar{B}$ (De Morgan Özelliği)
- $A \cup A = A$, $A \cap A = A$ (Denk Güçlü Özelliği)

özellikleri sağlanır.

2.2 Bulanık Sayılar

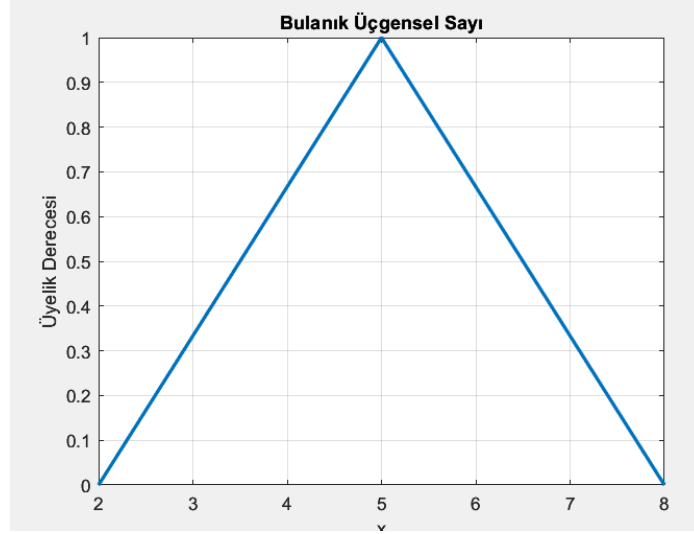
Bulanık sayılar, bulanık kümelerin bir alt kümesidir. Kaufmann ve Gupta (1988) tarafından üçgen ve bulanık sayılar için üyelik fonksiyonları ve temel özellikleri tanımlanmıştır.

2.2.1 Üçgen Bulanık Sayılar

Üçgen bulanık sayılar (n_1, n_2, n_3) sıralı üçlüsü ile ifade edilirler. n_1, n_2, n_3 sırasıyla en küçük değeri, en büyük değeri ve en geniş değeri göstermektedir. $N = (n_1, n_2, n_3)$ üçgen bulanık sayısı için üyelik fonksiyonu

$$\mu_N(x) = \begin{cases} 0, & x < n_1 \\ \frac{x - n_1}{n_2 - n_1}, & n_1 \leq x \leq n_2 \\ \frac{n_3 - x}{n_3 - n_2}, & n_2 \leq x \leq n_3 \\ 0, & x > n_3 \end{cases}$$

biçimindedir.



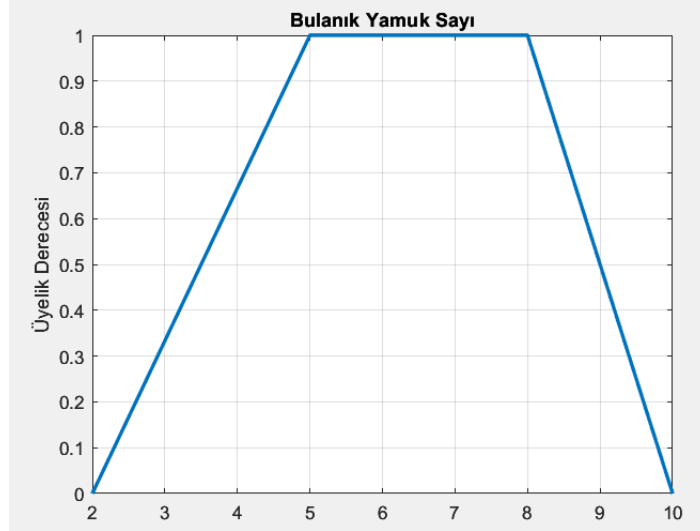
Şekil 2.4 Üçgen bulanık sayı.

2.2.2 Yamuk Bulanık Sayılar

Yamuk bulanık sayılar (n_1, n_2, n_3, n_4) sıralı dördlüsü ile ifade edilirler. Burada n_1 ve n_4 sırasıyla alt ve üst sınırlardır. $N = (n_1, n_2, n_3, n_4)$ yamuk bulanık sayısı için üyelik fonksiyonu

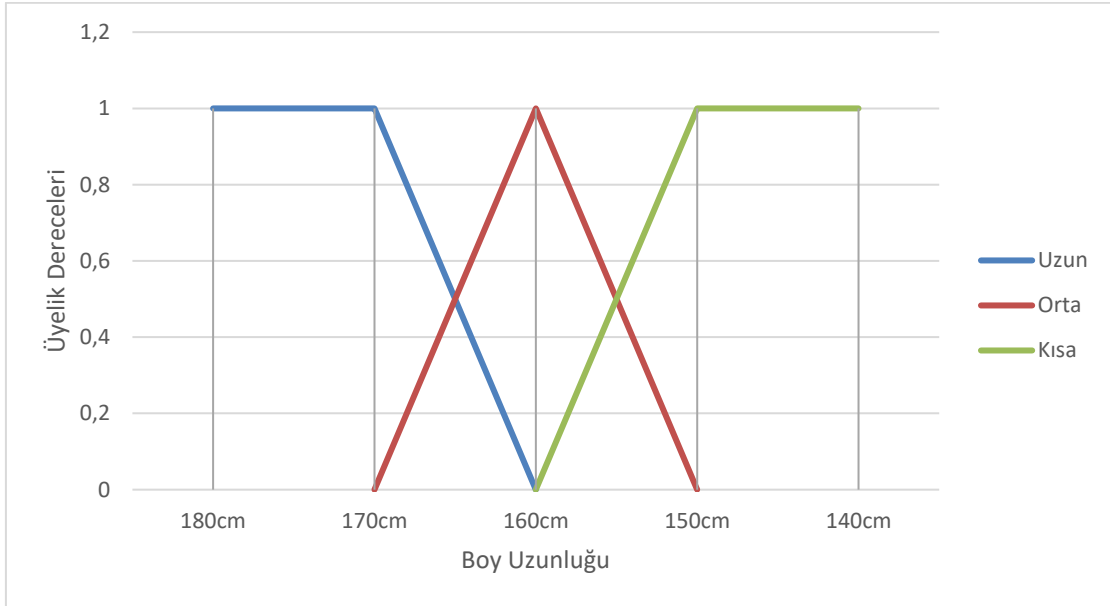
$$\mu_N(x) = \begin{cases} 0, & x < n_1 \\ \frac{x - n_1}{n_2 - n_1}, & n_1 \leq x \leq n_2 \\ 1, & n_2 \leq x \leq n_3 \\ \frac{n_4 - x}{n_4 - n_3}, & n_3 \leq x \leq n_4 \\ 0, & x > n_4 \end{cases}$$

biçimindedir.



Şekil 2.5 Yamuk bulanık sayı.

Bulanık mantığın dereceleri (üyelik dereceleri) vardır, örneğin klasik kümede 160 cm olan biri uzun insanlar kümesi içinde değildir, ayrıca 169 cm olan bir kişi de uzun boylu insanlar kümesinde değildir. Ancak bulanık mantıkta 160 cm boyunda olan bir kişiye kısa denemez çünkü orta boylu insanların kümesine de dahildir.



Şekil 2.6 Üyelik dereceleri.

Adı	Denklemi
Uzun	$\mu_{Uzun}(x) = \begin{cases} 0, & x < 160 \\ \frac{x - 160}{10}, & 160 \leq x \leq 170 \\ 1, & x > 170 \end{cases}$
Orta	$\mu_{Orta}(x) = \begin{cases} 0, & x < 150 \quad x > 170 \\ \frac{x - 150}{10}, & 150 \leq x \leq 160 \\ \frac{170 - x}{10}, & 160 \leq x \leq 170 \end{cases}$
Kısa	$\mu_{Kısa}(x) = \begin{cases} 1, & x < 150 \\ \frac{160 - x}{10}, & 150 \leq x \leq 160 \\ 0, & x > 160 \end{cases}$

Şekil 2.7 Bulanık sayıların denklemini kullanarak boy hesaplama.

Bulanık mantıkta 165 cm boyundaki bir kişi 0.5 üyelik derecesiyle, 169 cm boyundaki bir kişi 0.9 üyelik derecesiyle, 180 cm boyundaki bir kişi de 1 üyelik derecesi ile uzun boylu olabilir. Aynı şekilde 155 cm boyundaki bir kişi 0.5 üyelik derecesi, 160 cm boyundaki bir kişi 1 üyelik derecesi ile orta boy kümesine dahildir.

2.3 Bulanık Kümeler için Uzaklık ve Benzerlik Ölçüleri

Bu bölümde, birçok araştırmacı tarafından farklı açılardan geniş çapta araştırılan ve bulanık küme teorisinde önemli bir yer ihtiva eden uzaklık ve benzerlik ölçülerinin temel aksiyomlarından bahsedeceğiz.

Burada X kümesi üzerindeki tüm fuzzy kümelerinin ailesi $F(X)$ ile gösterilecektir.

Tanım 2.3.1 Bir $d : F(X) \times F(X) \rightarrow \mathbb{R}^+$ reel fonksiyonu

- i. Her $A, B \in F(X)$ için $d(A, B) = d(B, A)$,
- ii. Her $A \in F(X)$ için $d(A, A) = 0$,
- iii. Her $D \in \mathcal{P}(X)$ için $d(D, D^c) = \max_{A, B \in F(X)} d(A, B)$,
- iv. Her $A, B, C \in F(X)$ için eğer $A \subset B \subset C$ ise

$$d(A, B) \leq d(A, C) \text{ ve } d(B, C) \leq d(A, C)$$

özelliklerini sağlıyorsa d ye $F(X)$ üzerinde bir uzaklık ölçüsü denir.

Örnek 2.3.1 Her $A, B \in F(X)$ için $p \geq 1$ olmak üzere

$$d_p(A, B) = \left(\int_0^1 |\mu_A(x) - \mu_B(x)|^p \right)^{1/p}$$

biçiminde tanımlanan d_p fonksiyonu $F(X)$ üzerinde bir uzaklık ölçüsüdür.

Tanım 2.3.2 Bir $S : F(X) \times F(X) \rightarrow \mathbb{R}^+$ fonksiyonu

- i. Her $A, B \in F(X)$ için $S(A, B) = S(B, A)$,
- ii. Her $D \in \mathcal{P}(X)$ için $S(D, D^c) = 0$,
- iii. Her $C \in F(X)$ için $S(C, C) = \max_{A, B \in F(X)} S(A, B)$,
- iv. Her $A, B, C \in F(X)$ için eğer $A \subset B \subset C$ ise

$$S(A, B) \geq S(A, C) \text{ ve } S(B, C) \geq S(A, C)$$

özelliklerini sağlıyorsa S ye $F(X)$ üzerinde bir benzerlik ölçüsü denir.

Örnek 2.3.2 Her $A, B \in F(X)$ için $p \geq 1$ olmak üzere

$$S_p(A, B) = 1 - \left(\int_0^1 |\mu_A(x) - \mu_B(x)|^p \right)^{1/p}$$

biçiminde tanımlanan S_p fonksiyonu $F(X)$ üzerinde bir benzerlik ölçüsüdür.

2.3.1 Bulanık Kümeler İçin Uzaklık Tabanlı Benzerlik Ölçüleri

Diamond (1994) bulanık kümeleri içeren uzaklık ölçülerini tanıtmıştır. Uzaklık fonksiyonları, bulanık kümeler arasındaki benzerlik veya farklılığı ölçmek için kullanılan matematiksel fonksiyonlardır. Bulanık kümelerin benzerliğini hesaplamanın en belirgin yolu uzaklıklarına dayanmaktadır.

İki aşamalı olan bu hesaplamanın ilk aşamasında iki bulanık küme arasındaki uzaklık bir uzaklık ölçüsü ile elde edilir, ikinci aşamasında uzaklık ölçüsü yardımıyla benzerlik ölçüsü tanımlanır.

$X = \{x_1, \dots, x_n\}$ kümesi sonlu bir küme olsun. $i = 1, \dots, n$ olmak üzere X kümesinde herhangi iki bulanık A ve B kümesi için üyelik fonksiyonları $\mu_A(x_i)$ ve $\mu_B(x_i)$ olsun. Literatürde çeşitli uzaklık ölçütleri mevcuttur. En yaygın olarak kullanılan uzaklık ölçülerine aşağıda yer verilmektedir.

1. Hamming Uzaklığı

Hamming uzaklığı, iki bulanık küme arasındaki farklılık miktarını hesaplamak için kullanılır. Hamming uzaklığı

$$d_H(A, B) = \sum_{i=1}^n |\mu_A(x_i) - \mu_B(x_i)|$$

formülüyle hesaplanmaktadır.

2. Normalleştirilmiş Hamming Uzaklığı

Normalleştirilmiş Hamming uzaklığı, Hamming uzaklığına benzer bir ölçüdür. Normalleştirilmiş Hamming uzaklığı, Hamming uzaklığının toplam eleman sayısına bölünmesiyle elde edilmektedir. Normalleştirilmiş Hamming uzaklığı

$$d_{nH}(A, B) = \frac{1}{n} \sum_{i=1}^n |\mu_A(x_i) - \mu_B(x_i)|$$

formülüyle hesaplanmaktadır.

3. Öklid Uzaklığı

Öklid uzaklığı, iki bulanık kümenin elemanlarının üyelik dereceleri arasındaki değeri Euclid (Öklid) uzaklık formülüne göre hesaplanmaktadır.

Formülü ise

$$d_E(A, B) = \sqrt{\sum_{i=1}^n (\mu_A(x_i) - \mu_B(x_i))^2}$$

biçimindedir.

4. Minkowski Uzaklığı

Minkowski uzaklığı, bulanık kümeler arasındaki benzerlik veya farklılık derecesini ölçmek için kullanılan bir metriktir ve

$$d(A, B) = \left(\sum_{i=1}^n |\mu_A(x_i) - \mu_B(x_i)|^p \right)^{\frac{1}{p}}$$

formülüyle hesaplanmaktadır. Burada p , Minkowski uzaklığı formülünde kullanılan bir parametredir ve pozitif bir tamsayıdır. $p = 1$ için Minkowski uzaklığına Manhattan uzaklığı, $p = \infty$ için Minkowski uzaklığına Chebyshev uzaklığı denir. $p = 2$ için Minkowski uzaklığından Öklid uzaklığı elde edilir.

3. BULANIK KENAR BELİRLEME

Bu bölümde Matlab programı tanıtılmıştır. İlk olarak ikili, gri seviye ve RGB (Renkli) görüntü türleri açıklanmıştır. Daha sonra görüntü işleme ve filtrelemeden bahsedilerek kenar belirleme filtreleri anlatılmıştır. Ayrıca, bulanık mantıkla kenar belirleme yapılmıştır. Matlab programı ile elde edilen filtrelerin benzerlik oran değerleri tablolar halinde listelenmiştir.

3.1 Matlab

Matlab, 1984 yılında Cleve Moler tarafından öğrencilerinin matematiksel hesaplama problemlerini çözmelerine yardımcı olmak amacıyla geliştirilmiştir. Programın ismi, MATrix LABoratory (Matris Laboratuvarı) kelimelerinden türetilmiştir. Matlab'ın en önemli özelliklerinden biri, matris işlemleri için optimize edilmiş olmasıdır. Bu nedenle, büyük boyutlu matris işlemleri gibi hesaplama yoğunluğu yüksek işlemleri hızlı bir şekilde yapabilmektedir. Matlab

- Matematiksel hesaplama
- Fonksiyonlar oluşturma ve kullanma
- Grafik çizimi
- Veri analizi ve istatistik
- Yapay zekâ ve makine öğrenmesi uygulamaları
- Algoritma oluşturma

gibi birçok alanda kullanılan bir programdır.

Uyarı 3.1.1 Matlab'da proje oluşturmak için Matlab Ana Çalışma Alanı'na (Command Window) gelinir. “Current Folder” (Geçerli Klasör) bölümünde, yeni projeyi oluşturmak için istenilen klasöre gidilir veya yeni bir klasör oluşturulur. Bu klasör, projenin çalışma dizini olacaktır. Matlab Editor' u açmak için Ana Çalışma Alanı'ndan “File” (Dosya) menüsü seçilerek “New” (Yeni) seçeneğine tıklanır. Matlab, kendi komut dosyaları olan “.m” dosya uzantı formatını kullanır. Yeni açılan pencerede, “Script” (Komut Dosyası) veya “Function” (Fonksiyon) seçeneklerinden birisi seçilir. “Script” seçeneği, bir dizi Matlab komutunu içeren basit bir metin dosyası oluşturmaya

olarak sağlamaktadır. “Function” seçeneği ise parametreler alan ve herhangi bir değer döndüren Matlab fonksiyonu oluşturmaya olanak tanımaktadır. Seçtiğiniz seçeneğe tıkladıktan sonra, Matlab Editor' u yeni dosya için açılacak ve üzerinde çalışmaya başlayabileceğiniz bir metin düzenleyici görünecektir. Yeni dosyanın içirisine Matlab kodu yazılmaya başlanabilir. Dosyayı kaydetmek için “File” (Dosya) menüsünü seçilerek “Save” (Kaydet) seçeneğine tıklanır. Kaydedilecek ilgili klasörü seçilip dosyaya bir isim verilir. Proje kaydedilen dosyanın bulunduğu klasörde çalıştırılabilmektedir. Matlab Ana Çalışma Alanı'nda komutlarla ya da Matlab Editor üzerinde “Run” (Çalıştır) düğmesine tıklayarak proje çalışabilmektedir.

3.2 Görüntü Türleri

Görüntü türleri ikili, gri seviye ve RGB görüntüler biçimindedir.

3.2.1 İkili Görüntüler

İkili görüntüler, sadece siyah ve beyaz renklerinden oluşan görüntülerdir. Siyah renk genellikle nesnenin yokluğunu, beyaz renk ise nesnenin varlığını temsil etmektedir.

İkili görüntülerin oluşturulması için bir görüntünün uygun bir eşik değeri kullanılarak siyah ve beyaz renk tonlarına ayrılması gerekmektedir. Bu işlem, eşikleme işlemi olarak bilinmektedir. Eğer piksel değeri eşik değerinden büyükse, piksel beyaz olarak; eğer piksel değeri eşik değerinden küçükse, piksel siyah olarak işaretlenmektedir.

İkili görüntüler, özellikle morfolojik işlemler gibi temel görüntü işleme teknikleriyle işlenebilmektedir. Morfolojik işlemler ile ikili görüntülerdeki nesnelerin şeklini ve boyutunu değiştirmek, kenarları tespit etmek, nesne içindeki boşlukları doldurmak gibi işlemler gerçekleştirilebilmektedir. Bunun yanı sıra, ikili görüntülerde sınıflandırma (segmentasyon) ve nesne tespiti gibi diğer görüntü işleme uygulamaları da yapılabilmektedir.

3.2.2 Gri Seviyede Görüntüler

Gri seviyede görüntüler yalnızca siyah, beyaz ve aralarında grileşmiş tonları içeren görüntülerdir. Bu tonlar, görüntünün piksel değerlerinin belirli bir aralıkta olması ile oluşmaktadır. Her piksel bir gri tonunu temsil etmektedir. Ton değeri 0 (siyah) ile 255 (beyaz) arasında bir sayı ile ifade edilmektedir.

Bu görüntü türü, ikili (siyah-beyaz) görüntülere göre daha fazla bilgi içermektedir. Her bir piksel, yalnızca iki farklı değer yerine 256 (0-255) farklı değer alabilmektedir. Bu, görüntünün daha ince ayrıntılarının yakalanmasına ve daha iyi bir kontrast sunmasına izin vermektedir.

Sık sık görüntü işleme uygulamalarında gri seviyede görüntüler kullanılmaktadır. Örneğin kenar tespiti, görüntü sınıflandırma, filtreleme ve çeşitli görüntü analiz işlemleri için gri seviye görüntüler kullanılabilir.

3.2.3 RGB (Renkli) Görüntüler

RGB, kırmızı (R), yeşil (G) ve mavi (B) renk bileşenlerinden oluşan bir renk modelidir. RGB renk modeli, görüntülerin renklendirilmesi ve düzenlenmesi için kullanılan bir yöntemdir. Bu renk modeli renkli fotoğrafların, resimlerin, grafiklerin ve videoların doğru bir şekilde temsil edilmesini sağlamaktadır.

Görüntülerde her piksel RGB renklerinin bir kombinasyonunu temsil etmektedir. Görüntünün her bir pikselinin rengi, bu RGB değerlerine göre belirlenmekte ve bu pikseller bir araya gelerek tam bir görüntü oluşturmaktadır. Dijital görüntülerin yaygın dosya formatları arasında olan JPEG, PNG ve BMP dosya formatları RGB renk modelini desteklemektedir.

Gri seviyede görüntülerin oluşturulması, renkli bir görüntünün uygun bir yöntemle grileştirilmesiyle gerçekleştirilebilmektedir. Bu işlem, her pikselin RGB (Kırmızı, Yeşil, Mavi) kanallarındaki değerlerinin birleştirilmesiyle oluşmaktadır.

3.3 Görüntü İşleme ve Filtreleme

Görüntü işleme, dijital ortamda kaydedilmiş veya dijital hale getirilmiş bir görüntüyü işlemek için kullanılan bir bilgisayar teknolojisi dalıdır. Bu işlem, görüntüdeki belirli özelliklerin (renk, parlaklık, kontrast vb.) iyileştirilmesi, görüntüden gürültü giderilmesi, kenar ve nesne tespiti gibi birçok uygulamada kullanılmaktadır.

Görüntü işleme, görüntülerin işlenmesi için bir dizi teknik içermektedir. Bu teknikler, görüntü işleme algoritmaları olarak bilinmektedir. Bu işlemleri yapmak için matematiksel ve istatistiksel yöntemleri kullanmaktadır. Görüntü işleme tekniklerinden yaygın olarak kullanılan bazı uygulamalar şu şekildedir:

- Filtreleme: Görüntüyü belirli bir özellik veya kalite açısından iyileştirmek için kullanılmaktadır. Örnek olarak, kenar belirleme filtreleri, gürültü giderme filtreleri, bulanıklık giderme filtreleri vb. verilebilmektedir.
- Sınıflandırma (Segmentation): Görüntüde farklı nesneleri veya özellikleri ayırmak için kullanılmaktadır. Örneğin görüntü sınıflandırma, renk tabanlı segmentasyon vb. uygulamalar sınıflandırma uygulamalarıdır.
- Özellik Çıkarmı (Feature extraction): Görüntüdeki önemli özellikleri belirlemek için kullanılmaktadır. Işık yoğunluğu, köşeler, renk vb. örnekler verilebilmektedir.
- Görüntü Tanıma (Recognition): Görüntüdeki özellikleri tanımak ve sınıflandırmak için kullanılmaktadır. Örnek olarak, nesne tanıma, yüz tanıma uygulamaları verilebilmektedir.

Görüntü filtreleme, görüntü işleme sürecinde kullanılan bir tekniktir. Görüntü filtreleri, görüntüdeki gürültüyü azaltmak, kenarları belirginleştirmek, görüntüyü keskinleştirmek ve daha net bir görüntü elde etmek gibi amaçlar için kullanılmaktadır.

Görüntü filtreleri, genellikle bir matris veya çekirdek olarak ifade edilmektedir. Matrisin merkezindeki piksel, filtrelenen piksel olarak kabul edilmekte ve filtre matrisindeki diğer pikseller, bu pikselin etrafındaki diğer pikselleri temsil etmektedir. Filtre matrisindeki değerler, filtre tipine bağlı olarak değişkenlik göstermektedir.

3.3.1 Kenar Belirleme Filtreleri

Kenar belirleme filtreleri, bir görüntüdeki kenarları (yani, farklı piksel değerlerine sahip iki bölgenin birleştiği noktaları) tespit etmek için kullanılan filtrelerdir. Bu filtreler, bir piksel ve çevresindeki diğer piksellerin farkını hesaplayarak çalışmaktadır.

Matlab programı kenar belirleme filtreleri için özel işlevler sağlamaktadır. Bu işlevler, filtre matrisleri oluşturmak, filtre matrisini görüntüye uygulamak ve sonuçları analiz etmek için kullanılmaktadır. Matlab programında yaygın olarak kullanılan kenar belirleme filtreleri ve uygulamaları aşağıda verilmiştir.

3.3.1.1 Sobel Filtresi

Sobel filtresi, bir piksel ve çevresindeki diğer piksellerin farkını hesaplayarak çalışmaktadır. Her bir pikselin x ve y eksenlerindeki kenarlık yoğunluğunu hesaplamak için iki ayrı matris kullanılmaktadır. Bu matrisler, eksenlerindeki gradyanları hesaplamak için kullanılmaktadır.

Çekirdek matrisleri bir diğer tabirle konvolüsyon matrisleri köşegen, yatay ve dikey şekillerindeki kenarları bulmak amacıyla kullanılmaktadır. Sobel operatörü görselin ya da resmin kenarlarına denk gelen alansal ve yüksek frekansa karşılık bölgelerini yani keskin kenarları ortaya çıkarmaktadır. Bu operatördeki 3×3 boyutundaki konvolüsyon matrisleri aşağıda gösterildiği gibidir.

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, \quad G_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

G_x ve G_y matrisleri sırasıyla yatay ve dikey olarak görünen kenarları elde etmektedir. Matrisler girdiye yani giriş görüntüsüne bağımsız olarak uygulanabilmektedir. Böylelikle tüm eksenler için pikselin değeri ayrı ayrı ölçülmüş olmaktadır. Bu işlemlerden sonra çapraz duran kenarların mutlak değerini ve açısını bulmak için bu değerler aşağıdaki formüllerle birleştirilebilmektedir.

Piksel mutlak değeri

$$|G| = \sqrt{G_x^2 + G_y^2}$$

veya

$$|G| = |G_x| + |G_y|$$

formülleri yardımıyla hesaplanmaktadır. Kenarların açısı ise

$$\theta = \arctan\left(\frac{G_x}{G_y}\right)$$

formülü yardımıyla bulunur ve saat yönünün tersi yönündedir. Bu işlemin sonucunda eğer açı 90 derece ise dikey duran çizgileri, 0 derece ise yatay duran çizgileri gösterecektir. Eğik duran çizgilerde diğer açılar oluşturacaktır. Eğer açı 180 derecede ise üst kısım siyah bölgeden alt kısım beyaz bölgeye doğru geçişi gösterecektir.

G_x konvolüsyon matrisi yalnız kullanılırsa yatayda renk değişimini bulacağından dolayı çıkan çizgiler dikey olarak görülecektir. Benzer olarak G_y konvolüsyon matrisi de yalnız kullanılırsa aşağıdan yukarıya doğru renk değişimini siyahtan beyaza doğru göstereceğinden çıkan çizgiler yatay olacaktır. Resim üstünde belli açılardaki çizgiler göstermek istenirse verilen açı formülü ile hesaplama yapılarak istenilen çizgiler belirlenebilmektedir.

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad \text{ve} \quad G_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

konvolüsyon matrislerinin yaklaşık gradyan hesabı

$$P = \begin{bmatrix} P_1 & P_2 & P_3 \\ P_4 & P_5 & P_6 \\ P_7 & P_8 & P_9 \end{bmatrix}$$

Pseudo çekirdek matrisi yardımıyla

$$|G_x| = |-P_1 + P_3 - 2P_4 + 2P_6 - P_7 + P_9| \quad (3.1)$$

$$|G_y| = |P_1 + 2P_2 + P_3 - P_7 - 2P_8 - P_9| \quad (3.2)$$

formülleri ve dolayısıyla

$$|G| = |(P_1 + 2P_2 + P_3) - (P_7 + 2P_8 + P_9)| + |(P_3 + 2P_6 + P_9) - (P_1 + 2P_4 + P_7)|$$

formülü ile bulunabilmektedir.

Matlab programında kullanılan Sobel filtresinin algoritma kodları aşağıda Şekil 3.1’de verilmiştir.

```
% Sobel filtresi uygulama
sobel_x = [-1 0 1; -2 0 2; -1 0 1];
sobel_y = [-1 -2 -1; 0 0 0; 1 2 1];
img_sobel_x = imfilter(double(img), sobel_x);
img_sobel_y = imfilter(double(img), sobel_y);
img_sobel = sqrt(img_sobel_x.^2 + img_sobel_y.^2);
```

Şekil 3.1 Sobel filtresinin Matlab programındaki kodları.

RGB görüntülerin Matlab programında Sobel filtresi uygulanarak elde edilen çıktısı Şekil 3.2’de verilmiştir.



Şekil 3.2 Sobel filtresinin Matlab programındaki sonuçları.

3.3.1.2 Prewitt Filtresi

Prewitt filtresi, kenarların yönünü belirlemek için görüntüyü x ve y eksenlerinde taramaktadır. Bu filtre, genellikle kenar gücü veya gradyanın büyüklüğü hesaplaması için kullanılmaktadır. Piksel değerlerini komşu piksellerin değerleriyle karşılaştırarak kenarların yerini tespit etmektedir. Prewitt filtresinin çekirdek matrisleri

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \text{ ve } G_y = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

biçimindedir. Bu matrisler, filtre işlemi sırasında komşu piksellerin ağırlıklı toplamını hesaplamak için kullanılmaktadır. Sonuç olarak, x yönünde bir kenar görüntüsü ve y yönünde bir kenar görüntüsü elde edilmektedir. Bu iki görüntü, genellikle birleştirilerek son kenar görüntüsü ortaya çıkmaktadır.

Matlab programında kullanılan Prewitt filtresinin algoritma kodları aşağıda Şekil 3.3’de verilmiştir.

```
% prewitt|filtresi uygulama
prewitt_x = [-1 0 1; -1 0 1; -1 0 1];
prewitt_y = [-1 -1 -1; 0 0 0; 1 1 1];
img_prewitt_x = imfilter(double(img), prewitt_x);
img_prewitt_y = imfilter(double(img), prewitt_y);
img_prewitt = sqrt(img_prewitt_x.^2 + img_prewitt_y.^2);
```

Şekil 3.3 Prewitt filtresinin Matlab programındaki kodları.

RGB görüntülerin Matlab programında Prewitt filtresi uygulanarak elde edilen çıktısı Şekil 3.4’de verilmiştir.

Orjinal Resim



Prewitt Filtresi Çıktısı





Şekil 3.4 Prewitt filtresinin Matlab programındaki sonuçları.

3.3.1.3 Laplacian (Laplasyan) Filtresi

Laplacian filtresi, görüntüdeki keskin kenarları belirlemek için kullanılmaktadır. Laplacian operatörü, bir noktanın çevresindeki yoğunluk değerlerinin ikinci türevini hesaplamaktadır. Pozitif ağırlıklar yoğunluk artışını, negatif ağırlıklar ise yoğunluk azalışını temsil etmektedir.

Laplacian operatörü

$$\nabla^2(f) = \frac{\partial^2(f)}{\partial x^2} + \frac{\partial^2(f)}{\partial y^2}$$

biçiminde ifade edilmektedir. Burada, ∇^2 Laplacian operatörünü; f ise görüntüdeki yoğunluk fonksiyonunu ifade etmektedir. $\frac{\partial^2(f)}{\partial x^2}$, x yönünde ikinci türevi ve $\frac{\partial^2(f)}{\partial y^2}$, y yönünde ikinci türevi belirtmektedir. Laplacian filtresi, bu matematiksel ifadeyi ayırarak bir çekirdek formunda kullanılmaktadır. 3x3 boyutundaki bir Laplacian çekirdek matrisi

$$L = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

biçiminde ifade edilir.

Matlab programında kullanılan Laplacian filtresinin algoritma kodları aşağıda Şekil 3.5’de verilmiştir.

```
% Laplacian filtresi uygulama  
laplacian = [0 1 0; 1 -4 1; 0 1 0];  
img_laplacian = abs(imfilter(double(img), laplacian));
```

Şekil 3.5 Laplacian filtresinin Matlab programındaki kodları.

RGB görüntülerin Matlab programında Laplacian filtresi uygulanarak elde edilen çıktısı Şekil 3.6’da verilmiştir.



Şekil 3.6 Laplacian filtresinin Matlab programındaki sonuçları.

3.3.1.4 Canny Filtresi

Canny filtresi gürültüyü azaltma, keskinleştirme ve hassas kenar tespiti gibi işlemleri yapabilmektedir. Bu filtre bir görüntü üzerinde dört aşamalı bir işlem gerçekleştirmektedir:

1. Gürültü Azaltma:

- Görüntü üzerinde Gauss (blurlama) filtresi $3 \times 3, 5 \times 5, 7 \times 7$ vb. boyutlarındaki matrislerden birisini uygulamaktadır. Çekirdek boyutu, beklenen bulanıklaştırma etkisine bağlıdır. Temel olarak, çekirdek ne kadar küçük olursa, bulanıklık o kadar az görünür olur.
- Gauss filtresi, her bir pikselin komşu piksellerle ağırlıklı ortalamasını hesaplamaktadır. $(2k + 1) \times (2k + 1)$ boyutundaki bir Gauss filtre çekirdeğinin denklemi $1 \leq i$ ve $j \leq (2k + 1)$ olmak üzere

$$H_{ij} = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{(i - (k + 1))^2 + (j - (k + 1))^2}{2\sigma^2}\right)$$

biçiminde verilebilmektedir. Çoğunlukla görüntünün boyutuna göre değişim gösterse de $\sigma = 1$ iken 5×5 boyutunda Gauss filtre matrisi

$$\frac{1}{159} \begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix}$$

biçiminde kullanılmaktadır.

2. Kenar Gradyanlarının Hesaplanması:

- Görüntü üzerinde (3.1) ve (3.2) formüllerinin toplamı ile Sobel operatörü uygulanmaktadır.
- Gradyan değerleri, pikseldeki yoğunluk değişimini temsil etmektedir.

3. Eşikleme (Thresholding):

- Belirlenen alt ve üst eşik değerleri kullanılarak gradyan değerleri filtrelenmektedir.
- Eğer gradyan değeri üst eşik değerinden büyükse, piksel güçlü bir kenar pikseli olarak kabul edilmektedir. Eğer gradyan değeri alt eşik değerinden küçükse, piksel zayıf bir kenar pikseli olarak kabul edilmektedir.

4. Kenar Bağlantısı:

- Zayıf kenar pikselleri, güçlü kenar pikselleriyle bağlantılandırılmaktadır.
- Bu adımda ise zayıf kenar piksellerinin gerçek kenarlarla bağlantılı olmasını sağlanmaktadır.
- Genellikle piksel komşuluklarına dayalı bir bağlantı algoritması kullanılmaktadır.

Matlab programında kullanılan Canny filtresinin algoritma kodları aşağıda Şekil 3.7’de verilmiştir.

```
if iscell(strings)
    chars = cellfun(@(x)convertToChar(x), strings,...
        'UniformOutput', false);
elseif isstring(strings)
    chars = convertToChar(strings);
else
    % Do nothing
    chars = strings;
end

%-----
function c = convertToChar(s)
if isstring(s)
    if isscalar(s)
        c = char(s);
    else
        c = arrayfun(@(x) {char(x)}, s);
    end
elseif iscell(s)
    c = cellfun(@(x) convertToChar(x), s,...
        'UniformOutput', false);
else
    c = s;
end
```

Şekil 3.7 Canny filtresinin Matlab programındaki kodları.

RGB görüntülerin Matlab programında Canny filtresi uygulanarak elde edilen çıktısı Şekil 3.8’de verilmiştir.

Orijinal Resim



Canny Filtresi Çıktısı





Şekil 3.8 Canny filtresinin Matlab programındaki sonuçları.

3.3.1.5 Robert Cross (Roberts) Filtresi

Robert Cross (Roberts) filtresi, resimlerde 2 boyutlu geçişleri ölçmek için kullanılmaktadır. Keskin kenarları belirginleştirerek gri seviyede işlem yapmaktadır. Matris olarak 2×2 boyutunda matris kullandığından dolayı net olmayan resimlerde kenarları çoğunlukla bulamamaktadır. Bulunan kenarlar ise ince olarak gözükmemektedir. Aşağıda gösterilen 2×2 boyutundaki matrisler aralarında 90 derece açı bulunan iki matrisle işlemleri yürütmektedir.

$$\begin{bmatrix} P_1 & P_2 \\ P_3 & P_4 \end{bmatrix}, \quad G_x = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \quad G_y = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

Sobel filtresinde matrislerin yönleri x ve y eksenini boyunca kullanılmaktadır, fakat bu filtrede görüntü ızgarasına 45 ve 135 derecelik açılarda kullanılmaktadır.

$$|G_x| = |P_1 - P_4|$$

$$|G_y| = |P_2 - P_3|$$

biçiminde x ve y eksenini boyunca hesaplanan gradyanlar yardımıyla

$$|G| = |G_x| + |G_y|$$

veya

$$|G| = \sqrt{G_x^2 + G_y^2}$$

formülleri kullanılarak pikselin Roberts değeri elde edilebilmektedir. Oluşan kenarın resim ızgarasına olan açısı ise

$$\theta = \arctan\left(\frac{G_y}{G_x}\right) + 45$$

formülüyle hesaplanabilmektedir.

Matlab programında kullanılan Robert Cross filtresinin algoritma kodları aşağıda Şekil 3.9’de verilmiştir.

```
% robert cross fitresi uygulaması
rob_x = [0 0 0; 0 1 0; 0 0 -1];
rob_y = [0 0 0; 0 0 1; 0 -1 0];
img_rob_x = imfilter(img, rob_x);
img_rob_y = imfilter(img, rob_y);
rob= sqrt(img_rob_x.^2 + img_rob_y.^2);
```

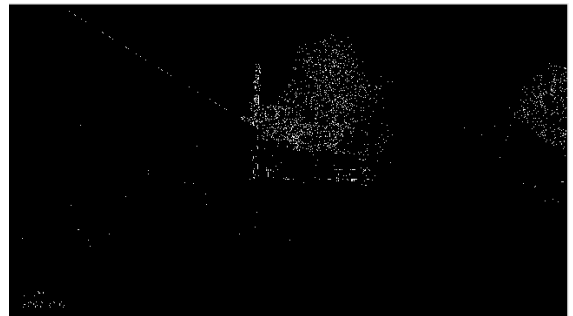
Şekil 3.9 Robert Cross filtresinin Matlab programındaki kodları.

RGB görüntülerin Matlab programında Robert Cross filtresi uygulanarak elde edilen çıktısı Şekil 3.10’da verilmiştir.

Orijinal Resim



Robert Cross Filtresi Çıktısı





Şekil 3.10 Robert Cross filtresinin Matlab programındaki sonuçları.

3.3.2 Bulanık Mantık ile Kenar Belirleme (Filtreleme)

Bulanık mantığın görüntüde kenar belirleme tekniklerinde kullanılması literatürde önemli bir yere sahiptir. Renkli görüntülerin bulanıklaştırılması, dijital görüntü işleme alanında sıkça kullanılan bir yöntemdir. Bulanıklaştırma, bir görüntüdeki detayları azaltarak görüntünün daha pürüzsüz ve yumuşak bir görünüm kazanmasını sağlamaktadır. Bu gürültüyü azaltmak, kenarları yumuşatmak veya özel efektler oluşturmak gibi birçok uygulamada kullanılmaktadır.

Bulanıklaştırma işlemi genellikle filtreleme işlemleri ile gerçekleştirilir. Bu filtreler, bir pikselin değerini, etrafındaki piksellerin değerleriyle birleştirerek yeni bir piksel değeri oluşturmaktadır.

Kenar belirleme için bulanık mantık kullanıldığında, keskin kenarlar yerine daha yumuşak, bulanık kenarlar elde edilebilmektedir. Bulanık mantık tabanlı kenar belirleme yöntemleri genelde bulanık kenarlık ölçüleri ve bulanık sınır çıkarma teknikleri kullanılmaktadır. Bu teknikler, görüntüdeki yoğunluk değişimlerini belirlemek ve kenar bölgelerini tanımlamak için bulanık kümeleri ve belirsizlik aralıklarını kullanmaktadır.

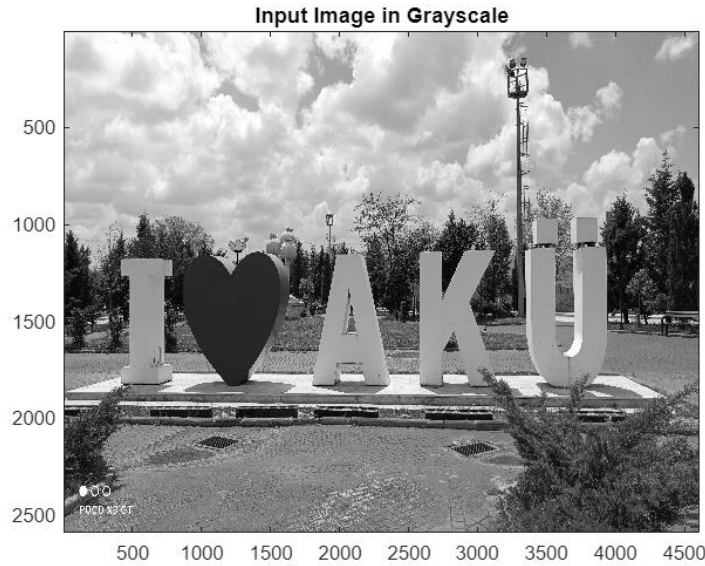
Bulanık kenar belirleme yöntemleri için genel adımlar aşağıda verilmiştir.

1. RGB görüntü seçilir ve içeri aktarılır.



Şekil 3.11 Görüntüyü içe aktarma.

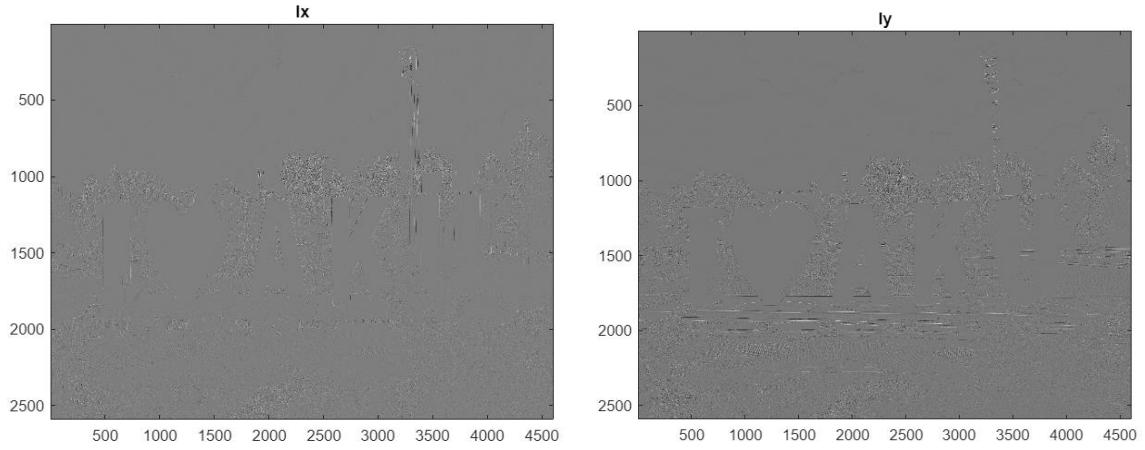
2. Giriş görüntüsünü bulanıklaştırma: Giriş görüntüsü, bulanıklaştırma işlemi kullanılarak bulanıklaştırılır. Bu, her pikselin etrafındaki piksellerin yoğunluk değerlerine dayalı olarak bulanık bir kenarlık ölçüsü elde edilmesini sağlamaktadır.



Şekil 3.12 RGB görüntünün gri tonlamaya dönüştürülüp bulanıklaştırılması.

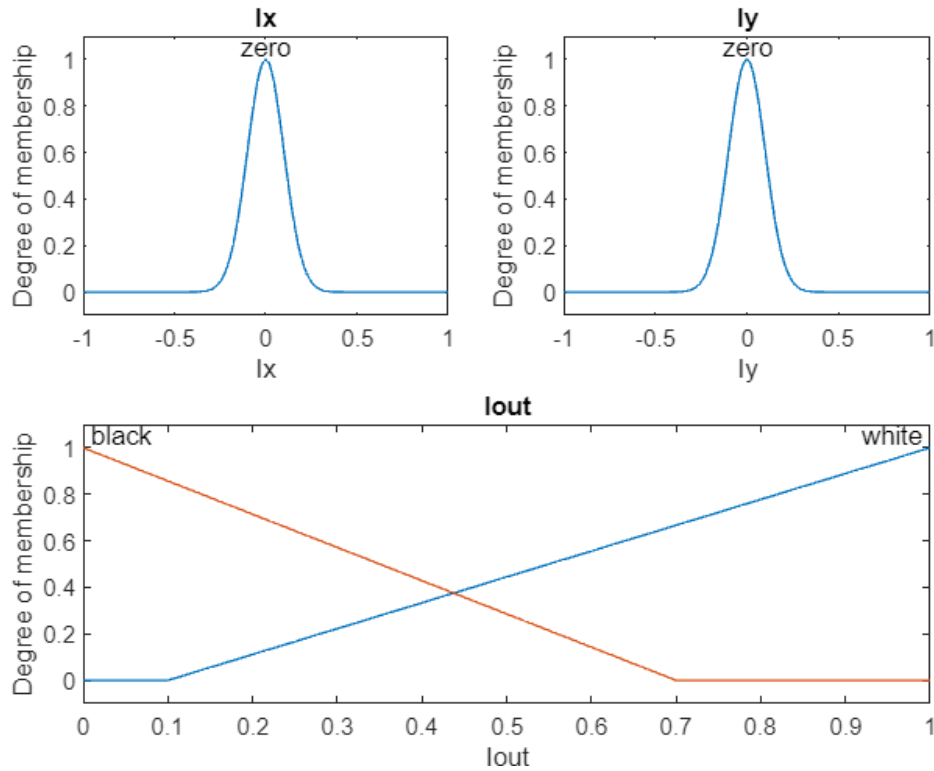
3. Gradyan (bulanık kenarlık) ölçümü: Bulanıklaştırılmış görüntü üzerinde kenarlık (gradyan) hesaplanmaktadır. Bu, kenarların yoğunluk veya keskinliklerini temsil

eden bir kenarlık haritasının oluşturulmasını sağlamaktadır.



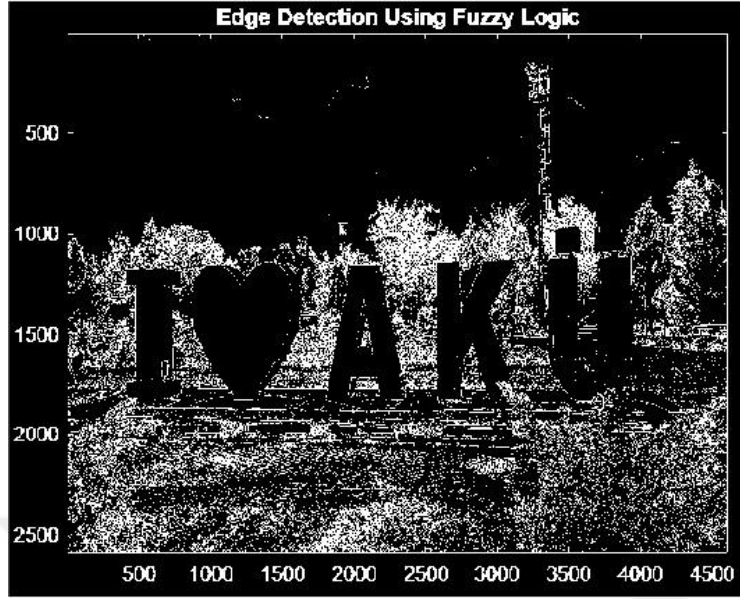
Şekil 3.13 Gradyan hesaplanması.

4. Bulanık kenarlık eşikleme (thresholding): Kenarlık haritası üzerinde belirli bir eşik değeri kullanılarak eşikleme yapılmaktadır. Bu, kenarların belirgin hale getirilmesi ve arka plana olan geçişlerin bastırılması için kullanılmaktadır.



Şekil 3.14 Kenar belirleme ve üyelik derecelerinin oluşturulması.

5. Sonuç elde etme: Bulanık sınır haritası kullanılarak, belirli bir eşik değeri üzerindeki pikseller keskin bir kenar olarak kabul edilebilmektedir. Bu, bulanık kenarları keskin kenarlara dönüştürmekte ve sonuç olarak kenar belirleme elde edilmektedir.



Şekil 3.15 Bulanık mantık kullanılarak elde edilen kenar görüntüsü.

Bulanık mantık ile kenar belirleme, geleneksel kenar belirleme yöntemlerine göre daha esnek ve toleranslı bir yaklaşımdır. Bu yöntem, görüntülerdeki belirsizlikleri ve bulanıklıkları ele almak için daha uygundur. Ancak, bulanık mantık tabanlı kenar belirleme yöntemleri, daha karmaşık hesaplama ve ayarlama işlemleri gerektirebilmekte ve bundan dolayı da işlem süresini arttırabilmektedir.

Matlab gibi görüntü işleme araçları, bulanık mantık tabanlı kenar belirleme yöntemlerini uygulamak için kullanılabilmektedir. Bu araçlar, bulanık mantık işlemlerini ve kenar belirleme algoritmalarını içeren fonksiyonları sağlamaktadır. Bu şekilde bulanık mantık tabanlı kenar belirleme işlemlerini gerçekleştirmeyi kolaylaştırmaktadır.

3.3.3 Kenar Belirleme Filtrelerinin Benzerlik Oranları

Bu bölümde Hamming, Öklid ve Minkowski uzaklık ölçüleri yardımıyla elde edilen benzerlik fonksiyonlarını kullanarak filtreler arasındaki benzerlik oranları hesaplanmıştır.

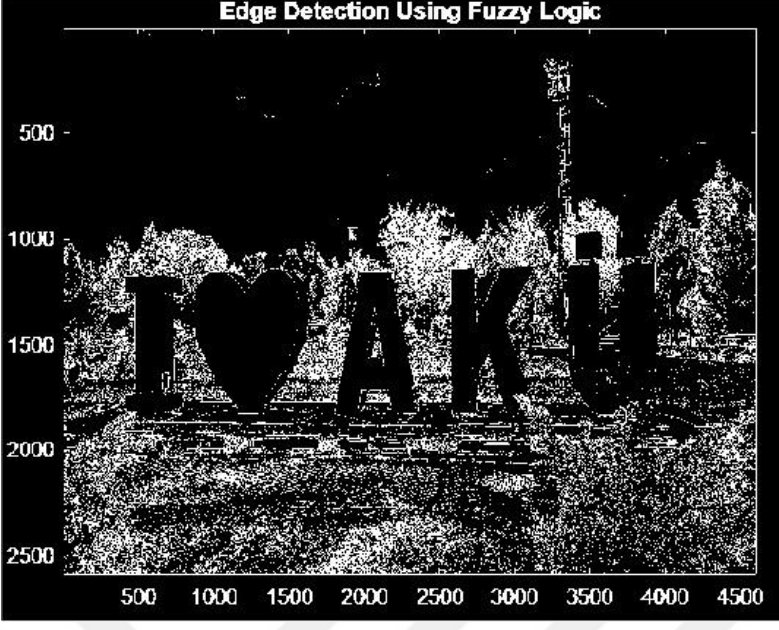
Tablo 3.1 ve Tablo 3.2 ile verilen tablolarda kullanılan Hamming, Öklid ve Minkowski

uzaklıklarına $\frac{1}{1+d}$ benzerlik fonksiyonu uygulanmıştır.

Gri Seviye Görüntü 			
Karşılaştırılan Filtreler	Hamming Uzaklığına Göre Benzerlik Oranı	Öklid Uzaklığına Göre Benzerlik Oranı	Minkowski Uzaklığına Göre Benzerlik Oranı
Canny-Sobel	0.0000201700496036	0.0096	0.0354
Prewitt-Sobel	0.0002408714161680	0.1613	0.2975
Laplacian-Sobel	0.0000405637390624	0.0186	0.0665
Roberts-Sobel	0.0000486013252296	0.0203	0.0677
Roberts-Prewitt	0.0000477278558286	0.0201	0.0673
Canny-Prewitt	0.0000202525729978	0.0095	0.0352
Roberts-Canny	0.0000184832317657	0.0090	0.0335
Laplacian-Roberts	0.0000419213343493	0.0184	0.0692
Laplacian-Prewitt	0.0000399613524173	0.0186	0.0658
Laplacian-Canny	0.0000185044423453	0.0085	0.0331

Tablo 3.1 Klasik filtreler arasındaki benzerlik oranlarının elde edilmesi.

Verilen tabloda Hamming, Öklid ve Minkowski uzaklıkları yardımıyla elde edilen benzerlik oranlarında en yüksek oran Prewitt ve Sobel filtreleri arasında çıkmıştır. Hamming uzaklığı yardımıyla elde edilen benzerlik oranına göre en düşük benzerlik oranı Roberts ve Canny, Öklid ve Minkowski uzaklıkları yardımıyla elde edilen benzerlik oranlarına göre en düşük benzerlik oranı ise Laplacian ve Canny filtreleri arasında çıkmıştır.

			
Karşılaştırılan Filtreler	Hamming Uzaklığına Göre Benzerlik Oranı	Öklid Uzaklığına Göre Benzerlik Oranı	Minkowski Uzaklığına Göre Benzerlik Oranı
Canny-BMS	0.0000039992	0.002032865	0.015933590
Laplacian-BMS	0.0000039855	0.002094396	0.016531473
Prewitt-BMS	0.0000039794	0.002092853	0.016532522
Roberts-BMS	0.0000039991	0.002100671	0.016583102
Sobel-BMS	0.0000039853	0.002095436	0.016549766

Tablo 3.2 Klasik filtre ve BMS benzerlik oranlarının elde edilmesi.

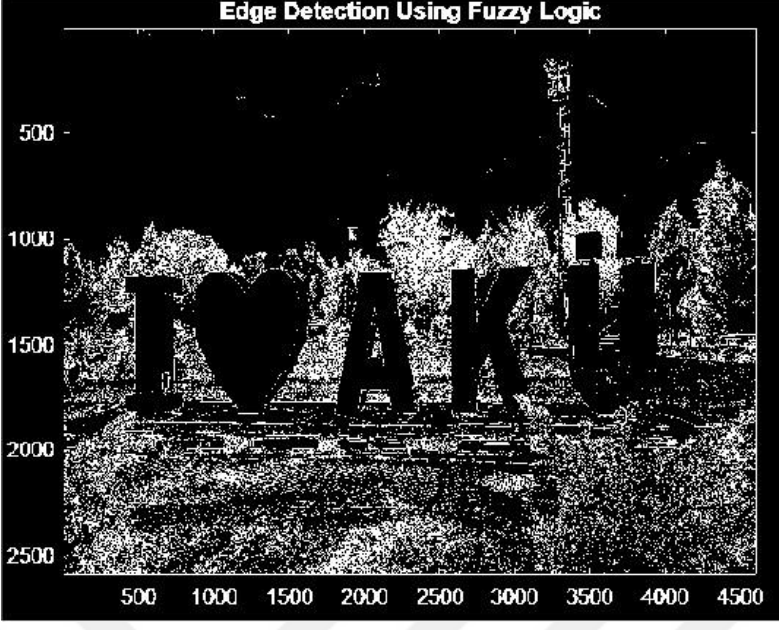
Verilen tabloda Hamming uzaklığına göre Canny filtresi ve BMS arasındaki benzerlik oranı en yüksek çıkmıştır. Öklid ve Minkowski uzaklığına göre belirlenen benzerlik oranı ise BMS ve Roberts filtresi arasında en yüksek değer olarak belirlenmiştir.

Tablo 3.3 ve Tablo 3.4 ile verilen tablolarda kullanılan Hamming, Öklid ve Minkowski uzaklıklarına e^{-d} benzerlik fonksiyonu uygulanmıştır.

Gri Seviye Görüntü 			
Karşılaştırılan Filtreler	Hamming Uzaklığına Göre Benzerlik Oranı	Öklid Uzaklığına Göre Benzerlik Oranı	Minkowski Uzaklığına Göre Benzerlik Oranı
Canny-Sobel	0.998508	0.8427	0.000000000001414
Prewitt-Sobel	0.999610	0.9858	0.094338521148113
Laplacian-Sobel	0.998649	0.9184	0.000000797210753
Roberts-Sobel	0.998583	0.9315	0.000001056525071
Roberts-Prewitt	0.999021	0.9302	0.000000966588515
Canny-Prewitt	0.998885	0.8433	0.000000000001258
Roberts-Canny	0.999130	0.8297	0.000000000000288
Laplacian-Roberts	0.907049	0.9210	0.000001445567338
Laplacian-Prewitt	0.999070	0.9173	0.000000687565407
Laplacian-Canny	0.999106	0.8299	0.000000000000209

Tablo 3.3 Klasik filtreler arasındaki benzerlik oranlarının elde edilmesi.

Tablo 3.3’de Hamming, Öklid ve Minkowski uzaklıkları yardımıyla elde edilen benzerlik oranlarında en yüksek oran Prewitt ve Sobel filtreleri arasında çıkmıştır. Hamming uzaklığı yardımıyla elde edilen benzerlik oranına göre en düşük benzerlik oranı Laplacian ve Roberts, Öklid uzaklığı yardımıyla elde edilen benzerlik oranlarına göre Roberts ve Canny, Minkowski uzaklığı yardımıyla elde edilen benzerlik oranlarına göre ise Laplacian ve Canny filtreleri arasında çıkmıştır.

			
Karşılaştırılan Filtreler	Hamming Uzaklığına Göre Benzerlik Oranı	Öklid Uzaklığına Göre Benzerlik Oranı	Minkowski Uzaklığına Göre Benzerlik Oranı
Canny-BMS	0.827448868818681	0.999088787886648	0.00000000000000062
Laplacian-BMS	0.890082638366111	0.999285477332226	0.000000000000051658
Prewitt-BMS	0.889008579664769	0.999281783624385	0.000000000000044773
Roberts-BMS	0.890064334368099	0.999285414261781	0.000000000000052620
Sobel-BMS	0.888846716830148	0.999281228237080	0.033979268908462

Tablo 3.4 Klasik filtre ve BMS benzerlik oranlarının elde edilmesi.

Tablo 3.4’de Hamming ve Öklid uzaklığına göre Laplacian filtresi ve BMS arasındaki benzerlik oranı en yüksek çıkmıştır. Minkowski uzaklığına göre belirlenen benzerlik oranı ise BMS ve Sobel filtresi arasında en yüksek değer olarak belirlenmiştir.

4. TARTIŞMA ve SONUÇ

Filtreler, bir sinyaldeki istenmeyen bileşenleri bastırmak, gürültüyü azaltmak, kenarları belirlemek veya diğer sinyal işleme görevlerini gerçekleştirmek için kullanılmaktadır. Filtrelerin benzerlik oranları, görüntü işlemede ve sinyal işlemede önemli bir konudur. Benzerlik oranları, iki veya daha fazla filtre arasındaki benzerlik düzeyini ifade eder ve bu oranlar, filtrelerin performansını değerlendirmek için kullanılmaktadır. Bu çalışmada uygulanan filtreler arasında Sobel, Canny ve Prewitt filtrelerinin, keskin kenarları doğru bir şekilde belirleme konusunda öne çıktığı gözlemlenmiştir. Ayrıca benzerlik oranı en yüksek filtreler Prewitt ve Sobel filtresi olarak belirtilmiştir.

Bu çalışmanın sonuçları, bulanık mantıkla kenar belirleme filtrelerinin görüntü işleme alanında etkili bir yaklaşım olduğunu doğrulamaktadır. Bulanık mantıkla filtrelemede, görüntüdeki kenarların belirlenmesi başarılı sonuçlar vermiştir. Geleneksel kenar belirleme teknikleri, keskin kenarları iyi tespit edebilirken, gürültülü ve düzensiz kenarları düzgün bir şekilde çıkaramamaktadırlar. Bu nedenle, bulanık kenar belirleme yöntemleri, daha karmaşık görüntü yapısına sahip nesneleri ve kenarları daha iyi ele almaktadır. Bulanık kenar belirleme, bulanık mantık temelinde çalışan yöntemlerle kenarlar arasında bulanık geçiş bölgeleri belirlemeyi amaçlamaktadır. Bu sayede, her pikselin sadece tamamen siyah veya tamamen beyaz olmadığı, aksine bu piksellerin arasında bulanık geçişlerin olduğu kabul edilmektedir. Bulanık kenar belirleme, belirli bir pikselin ne derece kenarda olduğunu ifade eden bir sayısal değer üretmektedir. Bu değer, pikselin kenara olan üyelik derecesini ifade etmektedir.

Elde edilen sonuçlar neticesinde bu çalışma görüntü işleme ve diğer alanlarda bulanık mantıkla kenar belirleme filtrelerinin etkili bir araç olduğunu göstermektedir. Bu çalışmanın sonuçları, ilgili araştırmacılar için temel bir kaynak teşkil edecektir.

5. KAYNAKLAR

- Aborisade D O, 2010, Fuzzy Logic Based Digital Image Edge Detection, Global Journal of Computer Science and Technology, 10, 78–83.
- Aksoy Y, Özkan E M, Karanfil S, 2003, Bulanık Mantığa Giriş, Yıldız Teknik Üniversitesi Vakfı Yayınları, İstanbul.
- Alshennawy A A, Aly A A, 2009, Edge Detection in Digital Images Using Fuzzy Logic Technique, International Journal of Computer and Information Engineering, 3, 540–548.
- Aybar E, 2006, Sobel Edge Detection Method for Matlab, Anadolu University, Porsuk Vocational School, Eskişehir.
- Beg I, Ashraf S, 2009, Similarity Measures for Fuzzy Sets, Applied and Computational Mathematics, 8, 192–202.
- Chen S M, Yeh M S, Hsiao P Y, 1995, A Comparison of Similarity Measures of Fuzzy Values, Fuzzy Sets and Systems, 72, 79–89.
- Couso I, Garrido L, Sánchez L, 2013, Similarity and Dissimilarity Measures Between Fuzzy Sets: A Formal Relational Study, Information Sciences, 229, 122–141.
- Çitli N, 2006, Bulanık Çok Kriterli Karar Verme, Yıldız Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, İstanbul.
- Diamond P, Kloeden P, 1994, Metric Spaces of Fuzzy Sets Theory and Applications, World Scientific, Singapore.
- Kamber E, 2017, Sezgisel Bulanık Normlu Uzaylarda Tanımlanan Bazı Yakınsaklık Çeşitleri, Sakarya Üniversitesi, Fen Bilimleri Enstitüsü, Doktora Tezi, Sakarya.
- Karakaşoğlu N, 2008, Bulanık Çok Kriterli Karar Verme Yöntemleri ve Uygulama, Pamukkale Üniversitesi, Sosyal Bilimler Enstitüsü, Yüksek Lisans Tezi, Denizli.
- Kaufmann A, Gupta M M, 1988, Fuzzy Mathematical Models in Engineering and Management Science, Elsevier Science Publishers, North-Holland, Amsterdam.
- Klir G, Yuan B, 1995, Fuzzy Sets and Fuzzy Logic Theory and Applications, Prentice Hall, New Jersey.
- Kumawat A, Panda S, 2022, A Robust Edge Detection Algorithm Based on Feature-Based Image Registration (FBIR) Using Improved Canny with Fuzzy Logic (ICWFL), The Visual Computer, 38, 3681–3702.

- Kushwaha N, 2012, Edge Detection Using Fuzzy Logic in Matlab. *International Journal of Advanced Research in Computer Science and Software Engineering*, 2.
- Mohammad E J, JawadKadhim M, Hamad W I, Helyel S Y, Alrsaak A A A, Al-Kazraji F K S, HadeeAbud A, 2014, Study Sobel Edge Detection Effect on the Image Edges Using Matlab, *International Journal of Innovative Research in Science, Engineering and Technology*, 3, 10408–10415.
- Omran S, Hassaballah M, 2006, A New Class of Similarity Measures for Fuzzy Sets, *International Journal of Fuzzy Logic and Intelligent Systems*, 6, 100–104.
- Setnes M, Babuska R, Kaymak U, Van Nauta Lemke H R, 1998, Similarity Measures in Fuzzy Rule Base Simplification, *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, 28, 376–386.
- Sharifi M, Fathy M, Mahmoudi M T, 2002, A Classified and Comparative Study of Edge Detection Algorithms, *International Conference on Information Technology*, Las Vegas.
- Suryakant N K, 2012, Edge Detection Using Fuzzy Logic in Matlab, *International Journal of Advanced Research in Computer Science and Software Engineering*, 2.
- Talo Ö, Yavuz E, 2021, Cesaro Summability of Sequences in Intuitionistic Fuzzy Normed Spaces And Related Tauberian Theorems, *Soft Computing*, 25, 2315–2323.
- Utama K M R A, Umar R, Yudhana A, 2022, Edge Detection Comparative Analysis Using Roberts, Sobel, Prewitt, and Canny Methods, *Jurnal Teknologi dan Sistem Computer*, 2, 67–71.
- Verma R, 2015, Cosine Entropy and Similarity Measures for Fuzzy Sets, *Konuralp Journal of Mathematics*, 3, 83–93.
- Weng Y, 2022, Use Edge Detection Method and Matlab for Time Measurement in Athletics Field, In *Proceedings of the 2022 6th International Conference on Electronic Information Technology and Computer Engineering*, Association for Computing Machinery, New York, 326–330.
- Xuecheng L, 1992, Entropy, Distance Measure and Similarity Measure of Fuzzy Sets and Their Relations, *Fuzzy Sets and Systems*, 52, 305–318.
- Yang G, Xu F, 2011, Research and Analysis of Image Edge Detection Algorithm Based on the Matlab, *Procedia Engineering*, 15, 1313–1318.