

**T.C.
İNÖNÜ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**YAPAY SINIR AĞLARI İLE GELİŞTİRİLEN AKILLI KÖŞE BULMA
YÖNTEMİ**

FATMA ELZAHAR

**YÜKSEK LİSANS TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI**

ŞUBAT 2018

Tezin Bařlıđı: Yapay Sinir Ađları İle Geliřtirilen Akıllı Kõře Bulma Yöntemi

Tezi Hazırlayan: Fatma ELZAHAR

Sınav Tarihi: 23.02.2018

Yukarıda adı geçen tez jürimizce deđerlendirilerek Bilgisayar Mühendisliđi Ana Bilim Dalında Yüksek Lisans Tezi olarak kabul edilmiřtir.

Sınav Jüri Üyeleri

Tez Danıřmanı: Doç. Dr. Muhammed Fatih TALU
İnönü Üniversitesi

Prof. Dr. Ali Karcı
İnönü Üniversitesi

Doç. Dr. Bilal ALATAŐ
Fırat Üniversitesi

Prof. Dr. Halil İbrahim ADIGÜZEL
Enstitü Müdürü

ONUR SÖZÜ

Yüksek Lisans Tezi olarak sunduğum “Yapay Sinir Ağları İle Geliştirilen Akıllı Köşe Bulma Yöntemi” başlıklı bu çalışmanın bilimsel ahlak ve geleneklere aykırı düşecek bir yardıma başvurmaksızın tarafımdan yazıldığını ve yararlandığım bütün kaynakların, hem metin içinde hem de kaynakçada yöntemine uygun biçimde gösterilenlerden oluştuğunu belirtir, bunu onurumla doğrularım.

Fatma Elzahar

ÖZET

Yüksek Lisans Tezi

YAPAY SİNİR AĞLARI İLE GELİŞTİRİLEN AKILLI KÖŞE BULMA YÖNTEMİ

Fatma Elzahar

İnönü Üniversitesi
Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

49 + vii sayfa

2018

Danışman: Doç. Dr. Muhammed Fatih Talu

Bilgisayarla görmede öznitelik tanımlama önemli bir yer tutmaktadır. Farklı ölçek ve dönüşümlerde, değişen parlaklıklarda, küçük açı değişimlerinde, gürültülü, karışık ve yıpranmış görüntülerde değişmeyen öznitelikleri bulmak oldukça zordur. Ölçekten Bağımsız Öznitelik Dönüşümü metodu değişmeyen öznitelikleri bulmayı hedeflemektedir ve bunu yaparken dört aşamadan geçer. Anahtar noktaların bulunduğu aşamada köşe bulma yöntemini kullanır. Köşeler, görüntü işleme tekniklerinde çok kullanılan özelliklerdendir. Nesne tanıma, hareket izleme, stereo eşleştirme gibi yöntemler içinde sıklıkla kullanılmaktadır. Nesnenin tanımlanması, ayırt edilmesi ve belirlenebilmesi için köşelerden faydalanmak mümkündür. Gri-seviye görüntü içerisinde köşelerin tespitini gerçekleştiren yöntemler, kenar bulma tabanlı yöntemler ve yoğunluk tabanlı yöntemler olarak gruplandırılabilir. Kenar bulma tabanlı yöntemler, ön işlem olarak görüntünün kenarlarını bulduktan sonra köşe bulma işlemine devam eder. Köşe bulma işlemi öğreterek gerçekleştirmek için Yapay Sinir Ağları kullanılabilir. Yapay Sinir Ağları, öğrenme aşamasından sonra bilgiyi saklayıp genelleme yapar ve sistemin ürettiği çıktı ile beklenen değer arasında ilişki kurar.

Bu tez çalışmasında klasik yaklaşımlardan farklı olarak, Ölçekten Bağımsız Öznitelik Dönüşümü metodunda köşe bulma yöntemi için Yapay Sinir Ağları kullanılarak nesnelerin köşe bilgilerini öğrenen yapay bir köşe yakalayıcı geliştirilmiştir. Pozitif ve negatif görüntü parçaları oluşturularak sistem eğitilmiş, test uygulamalarıyla başarı elde edilmiş, geliştirilen akıllı köşe bulma yöntemi ile Moravec, Harris ve Susan Köşe Bulma Yöntemleriyle kıyaslanarak sonuçlar karşılaştırılmıştır.

ANAHTAR KELİMELER: Görüntü İşleme, Öznitelik Tanımlama, Ölçekten Bağımsız Öznitelik Dönüşümü, Köşe Bulma Yöntemleri, Yapay Sinir Ağları

ABSTRACT

M.S. Thesis

METHOD OF SMART CORNER DETECTION DEVELOPED WITH ARTIFICIAL NEURAL NETWORKS

Fatma Elzahar

İnönü University
Graduate School of Natural and Applied Sciences
Department of Computer Engineering

49 + vii pages

2018

Advisor: Assoc Prof. Muhammed Fatih Talu

Feature detection is an important part of computer vision. It is difficult to detect invariant features in different scales and transformations, in varying brightness, small angle changes, noisy, mixed and worn images. Scale Invariant Feature Transformation method aims to detect unchanging features and in doing so it goes through four stages. It uses the corner detection method in the stage where the key points are located.

The corners are very useful features in image processing techniques which are often used in object recognition, motion tracking and stereo matching methods. It is possible to use the corners to identify, distinguish and determine the object. Methods for performing corner detection in a gray-level image can be grouped into edge-based methods and density-based methods. The edge detection methods continue to detect the corner after finding the edges of the image as preprocessing.

Artificial Neural Networks can be used to perform teaching the corner detection process. Artificial neural networks store and generalize information after the learning phase and relate the output produced by the system to the expected value.

In this thesis study, an artificial corner catcher which learns corner information of objects by using Artificial Neural Networks for corner detection method in Scale Invariant Feature Transformation method is developed differently from classical approaches. The system was trained by creating positive and negative image parts and had high success results after test applications. Developed intelligent corner detection method was compared to other corner detection methods such as Moravec, Harris and Susan Corner Detection Methods and results were compared.

KEYWORDS: Image Processing, Feature Detection, Scale Invariant Feature Transform, Corner Detection Methods, Artificial Neural Networks

TEŐEKKÜR

Tez konusunun Őekillenmesinde ve alıŐmalarım boyunca gsterdiĐi anlayıŐ ve ilgi, verdiĐi her trl destek ve katkılarından tr danıŐman hocam Sayın Do. Dr. M. Fatih TALU'ya;

Ge baŐlayan ve ge biten yksek lisans eĐitimim boyunca beni destekleyen Sayın Prof. Dr. Ali KARCI' ya;

Btn Đrenim hayatım boyunca olduĐu gibi yksek lisans alıŐmalarım da gsterdiĐi sabır, destek ve ilgisinden tr deĐerli ANNEM'e;

alıŐmalarım sırasında kendilerinden aldıĐım zaman iin sabır gsteren eŐim Sherief ELZAHAR ve oĐlum Muhammed ELZAHAR'a

teŐekkr ederim.

İÇİNDEKİLER

ÖZET.....	i
ABSTRACT.....	ii
TEŞEKKÜR.....	iii
İÇİNDEKİLER.....	iv
SİMGELER VE KISALTMALAR.....	v
ŞEKİLLER DİZİNİ.....	vi
ÇİZELGELER DİZİNİ.....	vii
1. GİRİŞ.....	1
2. KÖŞE BULMA YÖNTEMLERİ.....	3
2.1 Moravec Yöntemi.....	4
2.2 Harris Yöntemi.....	5
2.3 Susan Yöntemi.....	8
3. ÖLÇEKTEN BAĞIMSIZ ÖZNİTELİK DÖNÜŞÜMÜ.....	12
3.1 Ölçek Uzayı Oluşturma.....	14
3.2 Anahtar Noktaların Bulunması.....	16
3.3 Yönelimlerin Atanması.....	20
3.4 Yerel İmge Tanımlayıcı.....	21
4. GELİŞTİRİLEN AKILLI KÖŞE BULMA YÖNTEMİ.....	25
4.1 Yapay Sinir Ağları.....	25
4.2 Eğitim Kümesinin İnşası.....	26
4.3 Ağ Mimarisi.....	31
4.4 Eğitim Sonuçları.....	32
4.5 Test Sonuçları.....	33
5. KARŞILAŞTIRMA SONUÇLARI.....	41
6. SONUÇLAR VE DEĞERLENDİRME.....	44
7. KAYNAKLAR.....	46

SİMGELER VE KISALTMALAR

BRIEF	İkili Sağlam Bağımsız Temel Öznitelikler
BRISK	İkili Sağlam Değişmez Ölçeklenebilir Anahtar Noktalar
DoG	Gauss Farkı
FAST	Hızlandırılmış Segment Testi Öznitelikleri
HOG	Yönlü Gradyan Histogramı
LoG	Gauss Laplası
MSE	Ortalama Hata Karesi
MSER	Maksimum Durağan Uç Bölgeler
ORB	Yönlü FAST ve Döndürülmüş BRIEF
PCA	Temel Bileşen Analizi
PSNR	Tepe Sinyal Gürültü Oranı
RIFF	Dönüşten Bağımsız Hızlı Öznitelikler
SIFT	Ölçekten Bağımsız Öznitelik Dönüşümü
SSIM	Yapısal Benzerlik Endeksi
SURF	Hızlandırılmış Gürbüz Öznitelikler
SUSAN	En Küçük Eşsiz Bölüm Benzetme Çekirdeği
USAN	Eşsiz Bölüm Benzetme Çekirdeği
YSA	Yapay Sinir Ağları

ŞEKİLLER DİZİNİ

Şekil 2.1	Harris yöntemine göre köşe olma durumu.....	7
Şekil 2.2	Basit bir görüntüde farklı yerlerde dört dairesel maske.....	9
Şekil 2.3	Benzer alanların aynı şekilde renklendirilmesi.....	9
Şekil 3.1	Ölçek uzayında gauss farklarının alınması	16
Şekil 3.2	Aynı oktavdaki pikselin komşu piksellerle karşılaştırılması	17
Şekil 3.3	Anahtar noktanın gauss ağırlık fonksiyonu ile işlenmesi	22
Şekil 3.4	Yönelimlerin histograma yerleştirilmesi.....	23
Şekil 3.5	Anahtar nokta etrafında seçilen pencerenin vektöre dönüşmesi.....	23
Şekil 4.1	Eğitim kümesi için oluşturulan kare görüntüsü	26
Şekil 4.2	Kenar bilgisi elde edilen kare görüntüsü	27
Şekil 4.3	Köşe olan parçanın 9x9 ebatlarında örneği.....	27
Şekil 4.4	Köşe görüntüsünün 45° çevrilmesiyle elde edilen örnekler	28
Şekil 4.5	Köşe görüntülerine gürültü eklenmesi sonucu elde edilen örnekler	29
Şekil 4.6	Köşe olmayan parçanın 9x9 ebatlarında örneği.....	29
Şekil 4.7	Köşe olmayan parçanın 15° çevrilmesiyle elde edilen örnekler	30
Şekil 4.8	Köşe olmayan görüntülere gürültü eklenmesiyle elde edilen örnekler	31
Şekil 4.9	Üç katmanlı yapay sinir ağı	32
Şekil 4.10	Matlab yapay sinir ağı karışıklık matrisi	33
Şekil 4.11	1. Sentetik görüntü ve köşe bulma yöntemleri sonuçları.....	34
Şekil 4.12	2. Sentetik görüntü ve köşe bulma yöntemleri sonuçları.....	36
Şekil 4.13	3. Sentetik görüntü ve köşe bulma yöntemleri sonuçları.....	38
Şekil 4.14	4. Sentetik görüntü ve köşe bulma yöntemleri sonuçları.....	39
Şekil 4.15	5. Sentetik görüntü ve köşe bulma yöntemleri sonuçları.....	40

ÇİZELGELER DİZİNİ

Çizelge 5.1	Köşe sayı bilgileri	42
Çizelge 5.2	Görüntü kalite metriklerine göre algoritma sonuçları.....	43

1. GİRİŞ

Bilgisayarla görme alanının bir alt dalı olan nesne tanıma, sayısal görüntü ve videolardaki nesnelerin ayırt edici özelliklerine göre tanımlanması işlemidir. Bilgisayarla görme konusunda geliştirilen yöntemler tanıma işlemlerinde yüz veya iris, optik karakter, barkod tanıma şeklinde uygulanırken, vücut veya aktivite takibi, web tabanlı sistemlerde görüntü araması, tıbbi görüntüleme işlemleri gibi alanlarda kullanılmaktadır. Dört adımdan oluşan bilgisayarla görme sisteminde sırasıyla önışleme adımında görüntü filtrelenip geliştirilir, bölütleme (segmentasyon) adımında nesnelere arka alandan çıkartılır, öznitelik çıkarımı adımında her nesnenin ayırt edici özellikleri tespit edilir ve sınıflandırma adımında nesnelere önceden tanımlanmış olan sınıf ya da kategorilere dâhil edilir.

Sınıflandırma adımının işlem süresinin artmaması ve sınıflandırma hassasiyetinin yüksek olması için gereksiz bilginin toplanmaması gerekmektedir. Bu nedenle öznitelik çıkarımı ve kullanılacak yöntem önemlidir. Örneğin, spektrum analiz bilgileri ses ve konuşma işleme uygulamalarında kullanılırken, çeşitli frekans bantlarındaki bilgileri öznitelik olarak kullanmakta [1], görüntü işleme uygulamalarında ise, görüntünün spektrum analizi bilgileri ve görüntüye ait renk, parlaklık, köşe ve kenar gibi bilgiler öznitelik kümesini oluşturmaktadır [2]. Metin sınıflandırma uygulamalarında, sınıflandırmayla oluşturulan birtakım anahtar kelime ve sembollerin metin içerisinde görülme sıklıkları öznitelik vektörü olarak kullanılmaktadır [3].

Pek çok öznitelik çıkarım yöntemi, nesnenin tanımlanması, ayırt edilebilmesi ve yerinin belirlenebilmesi için kullanılan köşe bulma yöntemlerini kullanmaktadır. Görüntüde yer alan nesnelere hakkında gerekli olan birçok bilgiyi veren köşeleri bulmak için kenar bulma tabanlı yöntemler ve gri-seviye görüntüler üzerinde morfolojik işlemler uygulayan yöntemler bulunmaktadır. Kenar bulma tabanlı yöntemler kenarları çıkardıktan sonra, yerel maksimum eğim derecesine sahip noktaları köşe olarak atayabilmek için bir eşik değeri kullanır. Kullanım alanı geniş olan bu yöntemlerin dezavantajı, aç ve yerleşim gibi köşelerin özelliklerini kaydetme yeteneğine sahip olmamalarıdır. Kitchen ve Rosenfeld'in geliştirmiş olduğu yöntem buna örnek olarak verilebilir [4]. Morfolojik işlemler uygulayan yöntemler ise köşeleri belirlemek için görüntü şablonları kullandığından fazla miktardaki hesaplamalar nedeniyle işlem performansı düşük olabilmektedir [5-6].

Bu tez çalışmasında, klasik köşe bulma yöntemlerine alternatif olarak, yapay sinir ağları kullanılarak öğrenme yoluyla yeni görüntülerdeki köşelerin ve nesnelerin bulunması hedeflenmiştir. Tez çalışmasında Ölçekten Bağımsız Öznitelik Dönüşümü (SIFT) metodunun anahtar nokta bulma aşamasında kullanılan köşe bulma yöntemi yerine, yapay sinir ağları ile geliştirilen akıllı köşe bulma yöntemi ile köşelerin bulunması sağlanmıştır.

Tez çalışmasının ikinci bölümünde literatürde yaygın bir kullanıma sahip olan bazı köşe bulma yöntemleri sunulmaktadır. Üçüncü bölümde Ölçekten Bağımsız Öznitelik Dönüşümü (SIFT) yöntemi anlatılmıştır. Dördüncü bölümde yapay sinir ağlarına değinilmiş olup, geliştirilen akıllı köşe bulma yönteminden bahsedilmiştir. Beşinci bölümde, kıyaslamalı sonuçlar ve değerlendirme sunulmaktadır.

2. KÖŞE BULMA YÖNTEMLERİ

Nesneyi görüntüden çıkarabilmek için kenar çizgileri veya noktasal özellikler kullanılır. Kenarlara nazaran daha güvenilir, anlamlı ve bağımsız olarak görülen noktasal özellikler ilgi noktaları veya köşelerdir. Görüntü işlemede önemli bir yeri olmasına rağmen, kesin bir “köşe” kavramı tanımı yapılamamıştır. Köşe; iki doğrunun kesiştiği nokta olarak veya yerel bir komşulukta iki farklı baskın kenarın kesişimi olarak tanımlanabilir. Görüntüde konum bakımından yeri doğru olarak tanımlanmış ve sabit bir şekilde beliren, nesnenin özelliğini gösteren ilgi noktası, bir köşe ya da bir eğrinin zirve noktası, bir çizginin sonu veya yerel yoğunluğu maksimum ya da minimum olan herhangi bir nokta olabilir [7]. Bulunan bu ilgi noktası gerçekte bir köşe olabilir ya da olmayabilir.

Köşe, iki doğrunun kesiştiği nokta olduğu gibi, şekillerin kesiştiği noktalar, nesne sınırlarının kesilerek yön değiştirdiği noktalar olabilir veya kenar sınır eğrisinin belirli bir eşik değeri üzerindeki değerleri köşe nokta olarak da tanımlanabilir [8].

Görüntü içindeki nesnelerin önemli birer özelliği olan köşeler, nesnenin tanımlanması, ayırt edilebilmesi ve yerinin belirlenebilmesi için yararlı olan bilgiyi sağlar[9]. Nesne tanımının yanı sıra, hareket izleme, stereo eşleştirme gibi köşeleri kullanan yöntemler için gerekli olan işlemlerden biri olmasından dolayı çok sayıda köşe bulma yöntemi geliştirilmiştir [10-16].

Standart bir köşe tanımı olmadığı gibi, görüntülerde farklı şekillerde köşeler oluştuğu için standart bir köşe bulma yöntemi de yoktur. Bu nedenle geliştirilen bir köşe bulma algoritmasının bütün köşe çeşitleri üzerinde doğru sonuçlar vermesi çoğu zaman sıkıntılıdır [17]. İyi bir köşe bulma algoritması bütün köşeleri doğru olarak tespit edebilmeli, köşe olmayan noktaları köşe olarak göstermemeli, köşeler tam olarak ayırt edilebilmeli, gürültülü resimlerde bile doğru sonuçlar vermeli ve bu koşulları sağlarken hızlı ve verimli bir şekilde çalışmalıdır [18]. Bunun yanı sıra değişik ışıklama, öteleme, döndürme ve diğeri dönüşümlere maruz kalmış benzer görüntülerde bile aynı köşeyi tespit edebilmelidir.

Literatürde köşe bulmak için farklı yaklaşımlar kullanan çok sayıda yöntem bulunmaktadır. Gri-seviyeli görüntüden köşelerin tespitini gerçekleştirirken kenar

bulma tabanlı yöntemler ve yoğunluk tabanlı yöntemler olmak üzere iki tür yöntem kullanılabilir [7].

Kenar bulma tabanlı yöntemler, adından da anlaşıldığı gibi öncelikle nesnelerin sınırlarını belirleyen kenarları çeşitli kenar bulma yöntemlerini uygulayarak bulmaktadır. Kenar çizgilerinin kesiştiği noktalar bulunarak veya eğriliğin maksimum olduğu noktalar belirlenerek köşelerin yerleri tespit edilmektedir. Bu yöntemde köşeleri belirleyebilmek için kenarları belirlenmiş bir görüntü üzerinde uygulama yapılmaktadır. En önemli dezavantajı, başlangıç olarak kenar bilgisine ihtiyaç duydukları için kenar bulma aşamasında oluşabilecek sapmalardan dolayı köşelerde kayba neden olabilirler. Kenar belirlemede ortaya çıkan hatalar veya sapmalar da köşeleri belirlemede hataya sebep olacaktır, bu bakımdan yeterince güvenilir değildirler.

Kenar bulma tabanlı yöntemler gibi daha önce başka bir işleme gerek duymadığı için güvenilirlikleri daha yüksek olan gri-seviye görüntüler üzerinde morfolojik işlemler uygulayarak köşe belirleyen yöntemler de vardır [11-12] .

Bu yöntemler, doğrudan görüntünün gri seviye yoğunluk değerleri üzerine uygulanırlar ve ilk işlem olarak başka bir yöntem uygulanmadığı için ondan kaynaklanabilecek hatalar oluşması söz konusu değildir.

Literatürde yer alan çok sayıda köşe bulma yönteminin birçoğu daha önce var olan bir yöntemin geliştirilmesine dayanır. Bu nedenle birçok köşe bulma yöntemine temel olan ve genellikle çok iyi sonuçlar üreten köşe bulma yöntemleri bu bölümde bahsedilmiştir.

2.1 Moravec Yöntemi

Hans P. Moravec tarafından 1977 yılında geliştirilmiş olan bu yöntemde [16] köşe araması yapılırken ilgili piksel merkez olarak kabul edilir, bu pikselin çevresindeki piksellerden ne kadar farklı olduğuna bakılır. Moravec, yoğunluk değişimlerini, 3×3 , 5×5 , 7×7 gibi küçük arama pencereleri kullanarak hesaplar. Bu yöntemde, arama penceresi 45° lik fark oluşturacak şekilde 4 farklı yöne kaydırılarak oluşan yeni arama pencereleri içindeki piksellerin parlaklıkları ile ilk arama penceresi içindeki piksellerin parlaklıkları arasındaki fark hesaplanır. I , gri tonlamalı dijital görüntünün piksellerinin

parlaklıklarını, $w(x, y)$ pencere fonksiyonunu, (u, v) ise kaydırma miktarını göstermek üzere yoğunluk değişim fonksiyonu Denklem 2.1'de gösterilmiştir.

$$E(u, v) = \sum_{x, y} w(x, y) [I(x + u, y + v) - I(x, y)]^2 \quad (2.1)$$

45° Fark ile oluşan kaydırma fonksiyonu değerleri Denklem 2.2'de gösterilmiştir.

$$(u, v) = (1, 0), (1, 1), (0, 1), (-1, 1) \quad (2.2)$$

Yoğunluk değişim fonksiyonunun minimum değerleri, köşe noktasını verir. Pencere fonksiyonunun sadece iki değer alması sonucu algoritmanın gürültüye karşı dayanıksız olması, sadece 4 farklı yöndeki 45°lik kayma grupları alındığı için yönden bağımsız olmaması, yoğunluk değişim fonksiyonu sonuçlarından sadece minimum olanın alındığı için kenarların da köşe gibi kabul edilmesi gibi dezavantajları bulunmaktadır.

2.2 Harris Yöntemi

Harris ve Stephens [15] 1988 yılında Moravec tarafından sunulan köşe bulma işlecinin eksikliklerini belirleyip, geliştirerek Plessey adıyla bilinen köşe bulma yöntemini ortaya koymuştur.

Pencere fonksiyonunun sadece iki değer alması sebebiyle gürültüye karşı dayanıksız olması durumunu engellemek için Denklem 2.3'deki Gauss fonksiyonunu kullanır.

$$w(x, y) = e^{-\frac{x^2 + y^2}{2\sigma^2}} \quad (2.3)$$

Moravec yönteminin aksine, Plessey yöntemi belirli bir pencerenin merkezindeki pikselin mümkün olan tüm küçük yönlerde yoğunluk değişimini hesaplaması Denklem 2.4'te gösterilmiştir.

$$E(\Delta u, \Delta v) = \sum_{x,y} w(x,y) [I(x + \Delta u, y + \Delta v) - I(x,y)]^2 \quad (2.4)$$

I_x ve I_y sırasıyla x ve y yönündeki kısmi türevler olmak üzere Taylor açılımı kullanılarak Denklem 2.5 elde edilir.

$$I(x + \Delta u, y + \Delta v) \approx I(x,y) + I_x(x,y)\Delta u + I_y(x,y)\Delta v \quad (2.5)$$

Yoğunluk değişim fonksiyonu, Taylor açılımında kullanılarak elde edilen denklem, Denklem 2.6'da gösterilmiştir.

$$E(\Delta u, \Delta v) = \sum_{x,y} w(x,y) [I_x(x,y)\Delta u + I_y(x,y)\Delta v]^2$$

$$A = \sum_{x,y} w(x,y) I_x^2(x,y)$$

$$B = \sum_{x,y} w(x,y) I_y^2(x,y)$$

$$C = \sum_{x,y} w(x,y) I_x(x,y) I_y(x,y)$$
(2.6)

Denklem 2.7'de kısaltılmış hali gösterilmiştir.

$$E(u, v) = Au^2 + 2Cuv + Bv^2 \quad (2.7)$$

Bu fonksiyon matris formunda yazıldığında Denklem 2.8 elde edilir.

$$(2.8)$$

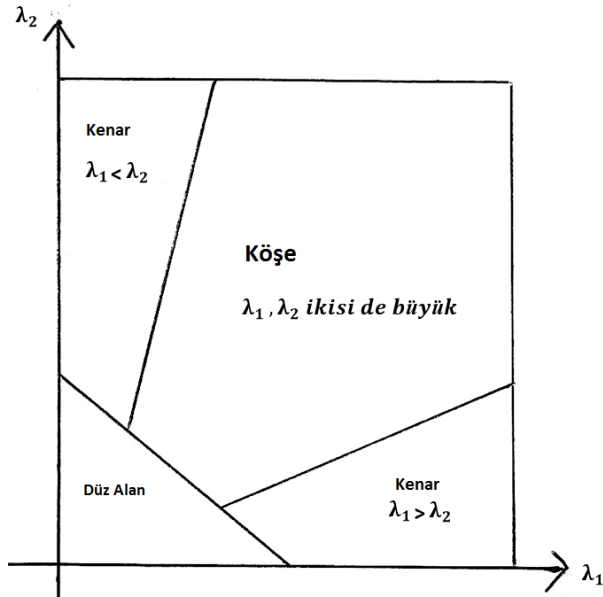
$$E(u, v) \cong [u, v] M \begin{bmatrix} u \\ v \end{bmatrix}$$

Bu ifadede M 2×2 'lik türev matrisi olmak üzere Denklem 2.9'da gösterilmiştir.

$$M = \sum_{x,y} w(x,y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \quad (2.9)$$

$$M = \begin{bmatrix} A & C \\ C & B \end{bmatrix}$$

Diagonal M matrisinin özdeğerleri ile yerel otokorelasyon fonksiyonunun kavışenmesi orantılıdır [15]. Karesel formda olan E fonksiyonu içerisindeki M matrisi simetriktir ve simetrik olan matrisin birimdik (ortonormal) özdeğerleri vardır. Şekil 2.1'de de gösterildiği gibi matrisin özdeğerlerinden (λ_1, λ_2) ikisi de eşik değerinin üzerinde ise o piksel köşe olarak kabul edilir, bir özdeğer diğerine göre daha büyükse kenar olarak, her iki özdeğer de küçük ise düz alan olarak kabul edilir. Böylece Moravec algoritmasındaki [16] 3.sorun da bu şekilde çözülmektedir.



Şekil 2.1 Harris Yöntemine Göre Köşe Olma Durumu

Harris yönteminde, M matrisinin özdeğerlerinin hesaplanmasının maliyetli olduğu görülmüş ve Denklem 2.10'da gösterildiği gibi yeni bir köşe fonksiyonu (R) önerilmiştir.

$$R = \det(M) - k \cdot (\text{Trace}M)^2$$

$$\det(M) = \alpha \times \beta \quad (2.10)$$

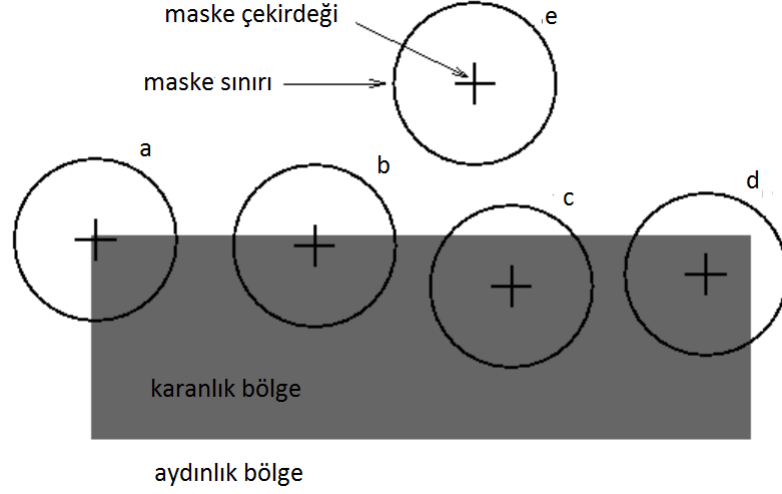
$$\text{Trace}M = \alpha + \beta$$

Harris ve Stephens k parametresinin uygulama testlerinde en doğru sonuçlar verdiği değer aralığını 0.04 ile 0.06 olarak önermiştir. R fonksiyonu negatif ise kenarları, pozitif ise köşe ve iç bölgeleri göstermektedir. Bu durumda köşe ve iç bölgeleri birbirinden ayırt edip köşeleri tespit edebilmek için belirli bir değere göre eşikleme yapılması gerekir, bu eşik değerinden daha küçük köşe değerleri iç bölgelere karşılık gelmektedir. Köşeler, köşe fonksiyonunun yerel maksimumu olarak belirlenir.

Plessey işlecinin genellikle doğru köşeleri bulan en iyi işleç olarak kabul edilmesine rağmen, işlem yoğunluğu fazla ve köşelerinin yerlerinin doğru tespiti konusunda eksiklikleri bulunduğu gözlenmiştir.

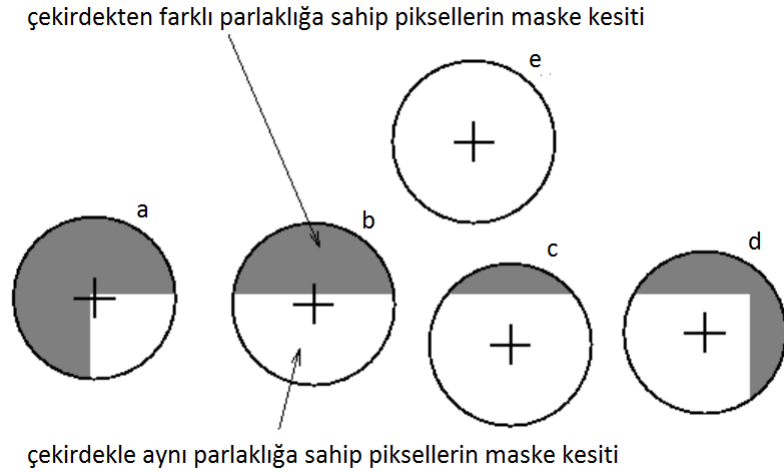
2.3 SUSAN Yöntemi

Smith ve Brady [10], dairesel bir maske içinde parlaklık karşılaştırmasına dayanan ve türev hesaplamadan köşelerin bulunabileceği bir yöntem önermişlerdir. Dairesel bir maskenin merkezi çekirdek olarak kabul edilir ve bu maske görüntüye Şekil 2.2'deki gibi uygulanır.



Şekil 2.2 Basit Bir Görüntüde Farklı Yerlerde Dört Dairesel Maske

Maskenin içindeki alanın parlaklığıyla çekirdeğin parlaklığı hesaplanarak karşılaştırılır, aynı veya benzerse USAN (Univalued Segment Assimilating Nucleus) olarak adlandırılır. USAN yoğunluk farkı olmayan düzgün bir görüntü bölümü üzerinde en geniş alana ulaşırken, bir kenar çizgisine yakın yer aldığında dairesel maskenin yaklaşık olarak yarısını kaplar. USAN'ın alanı maskenin yarısından daha az yer kapladığında ise merkez piksel bir köşe yakınlarında bulunur, Şekil 2.3'te gösterilmiştir.



Şekil 2.3 Benzer Alanların Aynı Şekilde Renklendirilmesi

Bu yöntem, kenarları ve köşeleri bulmak için USAN'ın büyüklüğünü, içindeki piksel sayısını, ikinci momentini kullanır. USAN haritasının yerel minimumu, görüntünün köşe noktalarını gösterir [10]. Görüntünün bütün pikselleri üzerinde dairesel maske uygulanarak merkez piksel ile maske içindeki pikseller karşılaştırılır ve benzer yoğunluktaki pikseller bulunarak USAN haritası oluşturulur. r_0 , maskenin merkez pikseli, r maske içindeki diğer pikseller, $I(r)$ piksellerin yoğunluk değeri, t eşik değeri olmak üzere ilgili pikselin USAN'ın üyesi olup olmadığı bilgisine Denklem 2.11 ile ulaşır.

$$c(r, r_0) = \begin{cases} 1, & |I(r) - I(r_0)| \leq t \\ 0, & |I(r) - I(r_0)| > t \end{cases} \quad (2.11)$$

Bu denklemi daha kararlı hale getirmek için Denklem 2.12 kullanılmalıdır.

$$c(r, r_0) = e^{-\left(\frac{|I(r) - I(r_0)|}{t}\right)g} \quad (2.12)$$

Her maskedeki USAN piksellerinin sayısına Denklem 2.13 ile ulaşılır.

$$n(r_0) = \sum_r c(r, r_0) \quad (2.13)$$

Bir sonraki aşamada n değeri geometrik eşik değeri olan g ile karşılaştırılır. n_{max} Değeri n 'nin alabileceği en büyük değer olmak üzere, g geometrik eşik değeri optimum gürültü reddetme oranı olan $3n_{max}/4$ değerine eşitlenerek Denklem 2.14 uygulanır.

$$R(r_0) = \begin{cases} g - n(r_0), & n(r_0) < g \\ 0, & \text{değilse} \end{cases} \quad (2.14)$$

Bazı noktalar köşe olmadığı halde köşe olarak belirlenebilir. Bunu engellemek için USAN'ın ağırlık merkezi ve çekirdekten ağırlık merkezine olan mesafe bulunur, bu mesafe yakın ise reddedilir. Bu basit yöntem algoritmayı daha kararlı hale getirir. Son olarak USAN haritası içinde yerel minimum hesaplanarak köşe noktaları tespit edilir. SUSAN yöntemi, gürültüye karşı dayanıklı oluşu ve görüntü dönüşlerinden çok fazla etkilenmemesi nedeniyle diğer köşe bulma yöntemlerine üstünlük sağlamaktadır.

3. ÖLÇEKTEN BAĞIMSIZ ÖZNETELİK DÖNÜŞÜMÜ

Nesnelerde ideal bir öznitelik perdelemelere, karışmalara, parazitlere ve bulanıklıklara dayanıklı iken aynı zamanda değişmez, ayırt edici, doğru ve verimli çalışır olmalıdır. Kullanılan yöntemin dayanıklılığı çıkarılan özniteliklerin dönüşüm ve ölçeklendirmeye karşı değişmez olmasıyla orantılıdır.

Her bir nesnenin özel görsel öznitelikleri bulunmaktadır. Bilgisayarlı görmede önemli bir yer tutan dönüşüme karşı değişimsizlik, nesne sınıflandırma uygulamalarında kullanılan görüntülerde ilgi bölgeleri bulunurken aranılan bir özelliktir. Farklı şartlarda elde edilen bir sahneye ait farklı görüntülerde bulunan ilgi bölgeleri aynı yüzey parçalarına denk gelerek bulunan ilgi bölgeleri sayesinde eşleşmeyi sağlar. Gauss Laplası (LoG The Laplacian-of-Gaussian) [19], Gaussların Farkı (DoG The Difference-of-Gaussian) [19], Maksimum Durağan Uç Bölgeler (MSER) [20] ilgi noktası bulma yöntemlerine örnek verilebilecek yöntemlerdir [21]. İlgi bölgelerinin yanı sıra nokta bulma ve eşleştirme işlemi de, bilgisayarla görmenin pek çok alanında kullanılmaktadır, bu amaçla geliştirilmiş çeşitli köşe bulma yöntemleri bulunmaktadır. 2. bölümde köşe bulma yöntemlerinden bahsedilmiştir.

Öznitelik tanımlamada çakışan görüntü bloklarından elde edilen eğim genlik değerleri büyüklüğüne dair histogram, ilk kez Dalal ve Triggs tarafından kullanılmıştır [22]. HOG (Histogramss of Oriented Gradients) olarak adlandırılan bu yöntem, yaya tanıma alanında kullanılmış ve daha sonra farklı nesne tanıma problemlerinde de yaygınlık kazanmıştır.

Özellikle sinyal işleme uygulamalarında sıkça kullanılan Temel Bileşen Analizi (PCA) yöntemi, sayısal görüntülerin de sinyal olarak yorumlanabilmesi sayesinde görüntü işleme uygulamalarında kullanılan bir yöntem olmuştur. PCA-SIFT [23] öznitelik çıkarma metodu ise, Ölçekten Bağımsız Öznitelik Dönüşümü (SIFT) metodunun dördüncü aşamasını değiştirmektedir. SIFT metodunda 4×4 parçalara histogram uygulanırken, PCA-SIFT metodunda 39×39 parça alınarak, yatay ve dikey genlik değerleri tek bir vektör olacak şekilde dizilir ve $2 \times 39 \times 39$ boyutunda bir vektör elde edilir. Bu vektöre Temel Bileşen Analizi (PCA) uygulanması sonucu vektörün boyutu düşer.

Bay ve Tuytelaars tarafından 2006'da önerilen SURF (Speeded Up Robust Features) [24] metodunda, SIFT metodundan daha iyi performans sağlayan bir yöntem geliştirilmeye çalışılmıştır. Bu amaçla Hessian matrisinin sadece determinantını hesaplamayı önermiştir. SIFT ile benzerlikler gösteren tanımlama algoritması, daha düşük işlemsel karmaşıklık değerine sahiptir. Bu yöntemin algılama ve tanımlama aşamaları SIFT yönteminden daha hızlı çalışmasına rağmen, görüş açısı ve parlaklık değişikliği durumlarında SIFT kadar iyi sonuçlar veremediği gözlenmiştir.

RIFF (Rotation Invariant Fast Features) yöntemi [25], ilgi algılayıcı ve öznitelik tanımlayıcı yöntem olarak kullanılmaktadır. Bu yöntemin gerçek zamanlı uygulamalar için hızlı, büyük ölçekli bilgi çekme görevleri için sağlam olduğu gözlenmiştir. Kutu filtre kullanması, filtreyi yeniden kullanması, düşük ölçek uzayı karmaşıklığına sahip olması, piksel yerine görüntünün tamamı üzerinden işlem yapması yöntemin hızlı olma sebeplerindedir.

SUSAN [10] köşe çıkarma yöntemine dayanan FAST (Features From Accelerated Segment Test) [26], FAST tabanlı çalışan BRISK (Binary Robust Invariant Scalable Keypoints) [27], bir anahtar bulma yöntemi ile birlikte çalışmak zorunda olan BRIEF (Binary Robust Independent Elementary Features) [28], FAST ve BRIEF birleşimi olan ORB (Oriented FAST and Rotated BRIEF) [29] yöntemi öznitelik çıkarma yöntemlerine örnek olarak verilebilecek diğer yöntemlerdir.

Bu tezde öznitelik çıkarım yöntemi olarak Ölçekten Bağımsız Öznitelik Dönüşümü (SIFT) kullanılmıştır. Görüntü eşleme, nesne ya da sahne tanımlama, çoklu imgelerden 3 boyutlu yapılandırma işlemlerinde hareketli nesne takibinde kullanılmakta olan yöntem, güçlü yöntemleri temel olarak kullanmasına ve çevrimiçi uygulamalarda ayırt edici ve nispeten hızlı olmasına rağmen eşleştirme aşamasında tanımlayıcının yüksek boyutlu olması (128 bit) büyük bir eksiklidir.

Görüntü eşleştirme tabanlı bir yöntem olan SIFT öznitelik metodunu, diğer öznitelik çıkarma metotlardan farklı ve üstün kılan en önemli özellik; farklı ölçek ve dönüşümlerde, değişen parlaklıklarda, görünüşteki küçük açı değişimlerinde, gürültü eklenmiş, karışık ve yıpranmış görüntülerde değişmeyen öznitelikleri bulmayı amaç edinmesidir.

SIFT yöntemi 4 aşamadan oluşmaktadır. İlk aşamada ölçek uzayı oluşturularak uç değer noktalar tespit edilir. İkinci aşamada anahtar noktalar bulunarak köşe olmayan noktalar elenir. Üçüncü aşamada bulunan anahtar noktalara yerel özelliklerine bağlı yönelim atanarak görüntü dönmelerinden kaynaklanan bağımlılık ortadan kaldırılır. Son aşamada anahtar noktalar bir Gauss penceresiyle ağırlıklandırılarak eğim büyüklüğü ve yönelim hesaplanır ve anahtar nokta tanımlayıcı bulunur.

3.1 Ölçek Uzayı Oluşturma

Ölçek, görüntünün yeniden boyutlandırılması için kullanılan katsayıdır. Aranılan nesneyi farklı koşullar altında tanıyabilmeye olanak sağlayan öznitelikler, görüntünün belirli oranlarda tekrar tekrar yeniden ölçeklenmesi ve filtrelenmesi ile oluşturulan ölçek uzayı ile çıkarılmaktadır. Farklı ölçeklenmiş ve filtrelenmiş görüntülerden oluşan ölçek uzayı, nesnenin içinde olduğu muhtemel arama ortamlarında ne gibi özniteliklere sahip olacağına dair bilgi sunar. Bu sayede ölçek farklılıkları sorun olmadan farklı ölçeklerde de tespit edilen noktaların konumlarının değişmemesi sağlanır. Ölçekten bağımsız sabit anahtar noktalar böylece elde edilebilir.

Koenderink [30] ve Lindeberg [31] in araştırmaları, bazı kabul edilebilir yaklaşımlar altında tek olası ölçek uzayı çekirdek matrisinin Gauss fonksiyonu olduğunu göstermiştir. Bu nedenle görüntünün ölçek uzayını oluştururken, görüntüyü Gauss ile katlamak gerekir. Gauss operatörü her piksele uygulanarak, bulanıklaşmış görüntü elde edilir.

Görüntünün Gauss ile filtrelenmesiyle elde edilen yeni görüntünün bulanıklık derecesini oktav belirler. Oktav, başlangıç değeri 2σ ise yarı değeri olan σ ile iki katı değeri olan 4σ arasındaki aralığı ifade eden seridir. İkinci seri, ilk serinin 2σ 'sı kadar alt örneklenmiş ve yarı oranına indirgenen görüntü ile başlar ve her seri için bu işlem bir önceki seri kullanılarak oktav sayısı kadar tekrarlanır. Bu şekilde başlangıç değeri değiştirilerek farklı oktavlar elde edilir.

Oktav ve ölçek sayısı, üzerinde işlem yapılacak olan orjinal resmin boyutuna bağlı olmasına rağmen, Lowe [19] bir oktav içerisinde $S + 3$ adet bulanıklaşmış görüntü kullanmaktadır ve bu görüntülerin çarpım faktörünü $k = 2^{1/s}$ olacak şekilde hesaplamaktadır. Bu değer 1'e yakınsaması teoride yaklaşım hatasının sıfıra

ulaşmasını sağlasa da pratikte $k = \sqrt{2}$ gibi bir değer için istikrarlı uç noktaların bulunmasında nerede ise etkisinin olmadığı gözlenmiştir [19]. Yaptığı çalışmalarda s değerini 2 alan Lowe, $k = 2^{1/2}$ çarpım faktörüne binaen dört oktav ve her bir oktav içinde beş bulanıklaştırmayı algoritma için ideal olarak tavsiye etmiştir. Denklem 3.1’de bir oktav içerisindeki ölçeklerin alacağı değerler gösterilmektedir.

$$\begin{aligned}
 \sigma_1 &= \sigma \\
 \sigma_2 &= k\sigma = \sqrt{2}\sigma \\
 \sigma_3 &= k^2\sigma = 2\sigma \\
 \sigma_n &= k^{n-1}\sigma = (\sqrt{2})^{n-1}\sigma
 \end{aligned} \tag{3.1}$$

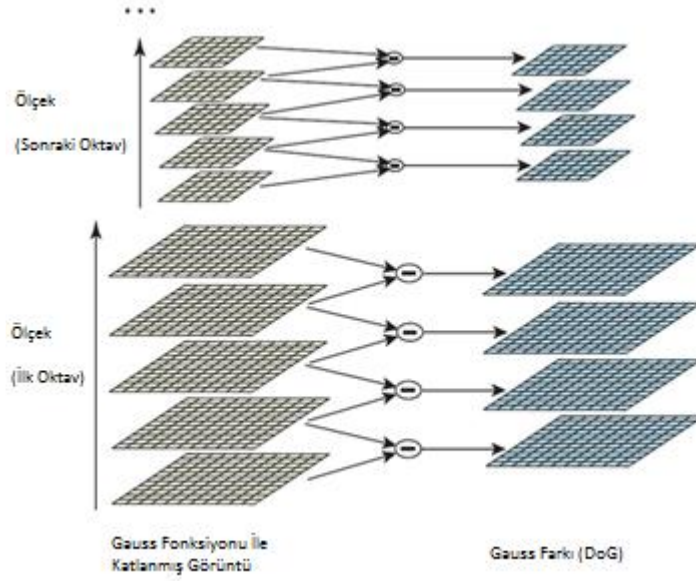
x, y Noktalarında σ ölçekli Gauss fonksiyonu Denklem 3.2’de gösterilmektedir.

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2} \tag{3.2}$$

$G(x, y, \sigma)$ Gauss fonksiyonu ile $I(x, y)$ görüntüsünün katlanması sonucu elde edilen $L(x, y, \sigma)$ bulanık görüntü Denklem 3.3’de ifade edilmiştir.

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \tag{3.3}$$

Kenar bulma algoritmalarında ve diğer görüntü işleme uygulamalarında ön işlem olarak, kenar iyileştirme amacıyla Gauss Laplası (LoG) veya Gauss Farkı (DoG) işlemleri kullanılmaktadır. Gauss Laplası işleminde bulanıklaştırılmış görüntünün ikinci türevi alınarak kenar ve köşeler bulunur. Ölçekten bağımsız olmaması, gürültüye karşı çok hassas olması ve türevden dolayı hesapsal maliyeti yüksek olması sebebiyle Gauss Laplası yerine Gauss Farkı tercih edilir[19]. Şekil 3.1’de oktav içerisinde Gauss Farkı alınmış görüntünün elde edilmesi gösterilmiştir.



Şekil 3.1 Ölçek Uzayında Gauss Farklarının Alınması

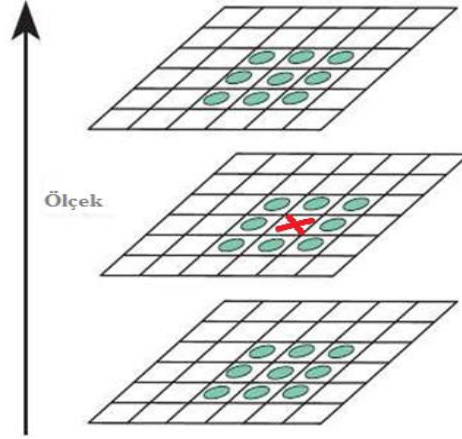
Gauss Farkı işlemi Gauss Laplası işlemine oldukça yakındır, ölçekten bağımsızdır, basit bir çıkarma işlemiyle yapılır ve Denklem 3.4’te gösterilmiştir. Bulanıklaştırma işlemi alçak geçiren filtre olarak kullanılırken, bulanıklaştırılmış görüntü orjinalinden çıkarıldığında, yüksek geçiren filtre gibi sonuç verir. Buna sebep olan, çıkarma işlemi sonucunda ani parlaklık değişimlerinin olduğu noktaların belirginleşmesidir[32].

$$D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) \quad (3.4)$$

$$D(x, y, \sigma) = L(x, y, k\sigma) - L(x, y, \sigma)$$

3.2 Anahtar Noktaların Bulunması

Ölçek uzayı belirlenen her oktav için Gauss Farkı hesaplandıktan sonra, uç noktaların bulunması gerekmektedir. Bu amaçla, aynı oktav içerisindeki Gauss Farkı görüntüsünün her pikseli, çevresindeki komşu 8 piksel, üst ve alt ölçeklerdeki 9’ar piksel olmak üzere toplamda 26 komşu piksel ile karşılaştırılır. Piksel o bölgenin yerel en büyük ya da en küçük değeriye *uçnokta* olarak atanırken arama işlemi en üst ve en alttaki ölçekler için yapılmaz. Şekil 3.2’de ilgili pikselin komşu piksellerle karşılaştırılması gösterilmektedir.



Şekil 3.2 Aynı Oktavdaki Pikselin Komşu Piksellerle Karşılaştırılması

Bulunan uçnokta değerleri tam olarak bir pikselin üzerinde bulunmayıp, genelde piksellerin arasında yer aldıkları için yaklaşık değerlerdir. Bu nedenle ölçek uzayında gerçek konumlarını belirlemek için ara değer kestirimi işlemi yapılır. Bu işlemin sonucunda düşük kontrastlı noktalar, gürültüye karşı hassas noktalar ve kenar üzerinde bulunan noktalar bulunur ve elenir.

Uç değerlerin aradeğer kestirim yerini belirlemek için Brown'un 3D ikinci derece fonksiyonunu yerel örnek noktalara uyarlamak gerekir[33]. Çalışmalar göstermiştir ki bu metot eşleştirme ve kararlılıkta büyük oranda başarı sağlamaktadır[19]. Burada D ve türevleri örnek noktada türetilir ve $x = (x, y, \sigma)^T$ bu noktadan olan uzaklıktır ve ofset olarak adlandırılır. Yöntem, $D(x, y, \sigma)$ ölçek uzayı fonksiyonunun Taylor açılımını kullanarak ara değer kestirimi yapmaktadır. Yerel örnek noktası x kullanılarak Denklem 3.5'deki gibi Taylor denklemi yazılır.

$$x = (x, y, \sigma)^T$$

$$D(x) = D + \frac{\partial D}{\partial x} x + \frac{1}{2} x^T \frac{\partial^2 D}{\partial x^2} x \quad (3.5)$$

Bu eşitliğin x 'e göre türevi alınır ve sifıra eşitlenirse, aranan uçdeğerler hesaplanarak ara piksellerin gerçek konumları \hat{x} Denklem 3.6'daki gibi bulunur.

$$\hat{x} = -\frac{\partial^2 D^{-1} \partial D}{\partial x^2} \frac{\partial D}{\partial x} \quad (3.6)$$

Brown'un önerdiği gibi, Hessian ve D 'nin türevi, komşu olan örnek noktaların farklarını kullanarak yakınlştırılır[33]. 3×3 Lineer sistem sonucu, en az maliyetle çözülebilir. \hat{x} Her hangi boyutta 0,5 ten büyükse, uç nokta başka bir örnek noktaya daha yakın demektir. Bu durumda, örnek nokta değiştirilir ve aradeğer kestirim işlemi bu nokta için oluşturulur. Bulunan \hat{x} , uç noktanın lokasyonunu kestirmek için örnek noktasının lokasyonuna eklenir. Uç noktalarda $D(\hat{x})$ değerinin eşitlikte yerine konularak hesaplanması Denklem 3.7'de gösterilmiştir. Bu sayede düşük kontrastlı bölgelerdeki kararsız uç noktaların elenmesi sağlanır.

$$D(\hat{x}) = D + \frac{1}{2} \frac{\partial D^T}{\partial x} \hat{x} \quad (3.7)$$

Elde edilen değer eşik değerinden düşükse düşük kontrastlı, dolayısı ile gürültüye karşı dayanıksız olduğu için elenir. Lowe yaptığı çalışmada, görüntü piksel değerlerini $[0, 1]$ aralığında kabul etmiştir ve hesaplamasında eşik değerini 0,03 olarak bu değerden küçük çıkan değerleri reddetmiştir[19].

Gauss Farkı görüntüleri, yapısı gereği kenarlara güçlü cevaplar verirler. Bu nedenle bu alanlarda kararlı olmayan anahtar noktaların oluşmasına sebep olabilmektedir. Burada Harris ve Stephens'in "Harris Köşe Bulma Yöntemi" kullanılır[15].

Harris Köşe Bulma Yöntemi'nde elde edilen köşe fonksiyonu Denklem 3.8'de gösterilmiştir.

$$R = \det(M) - k \cdot (\text{Trace}M)^2 \quad (3.8)$$

Köşenin bulunabilmesi için M matrisinin determinant ve iz'i bulunmalıdır. Bir matrisin determinantı, özdeğerlerinin çarpımına, iz'i ise toplamına eşit olduğundan

dolayı, köşe olma durumunun, her iki özdeğerin büyük olmasına bağlı olduğu görülmektedir.

Buna göre bir anahtar nokta etrafındaki birbirine dik olan iki eğim incelenir. Anahtar noktanın bulunduğu yere bağlı olarak 3 durum mümkündür. Bu yöntemde her iki eğim küçükse düz bir alan, eğimlerden biri küçük, diğeri büyükse kenar, her iki eğim de büyükse köşe olduğu kabul edilir[15]. Anahtar nokta adayları arasında en verimli olanlar köşe olduğu için diğeri elenir.

Gauss Farkı işleminde ayırt edilmesi zor olan bir zirve, kenar boyunca yüksek bir temel eğriye sahipken, dikey yönde bu değer küçüktür. Denklem 3.9'da da gösterildiği gibi Temel Eğriler 2×2 Hessian Matrisiyle hesaplanabilir. Komşu örnek noktaların farkları alınarak türevler hesaplanırken H matrisi anahtar noktanın ölçek ve lokasyonunda hesaplanır.

$$H = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix} \quad (3.9)$$

H matrisinin özdeğerleri, D görüntüsünün temel eğrisiyle orantılıdır. Sadece özdeğerlerin oranı değerlendirileceğinden, özdeğerleri hesaplamaya gerek yoktur. α , en büyük eğim büyüklüğüne sahip özdeğer; β , daha küçük eğim büyüklüğüne sahip özdeğer olmak üzere bu özdeğerlerin toplamı H matrisinin izi ile, bu özdeğerlerin çarpımı H matrisinin determinanı ile hesaplanabilir. Hessian Matrisi simetrik matris olduğundan, matrisin izi Denklem 3.10 ile ve determinanı Denklem 3.11 ile hesaplanır.

$$Tr(H) = D_{xx} + D_{yy} \quad (3.10)$$

$$Tr(H) = \alpha + \beta$$

$$Det(H) = D_{xx} \cdot D_{yy} - (D_{xy})^2 \quad (3.11)$$

$$Det(H) = \alpha \cdot \beta$$

Denklem 3.12'de r 'nin α ve β arasındaki oran olduğu ifade edilmiştir.

$$\alpha = r \cdot \beta \quad (3.12)$$

Denklem 3.13'te de özdeğerler yerine özdeğerler arasındaki orana ihtiyaç duyulduğu görülmektedir.

$$\begin{aligned} \frac{Tr(H)^2}{Det(H)} &= \frac{(\alpha + \beta)^2}{\alpha \cdot \beta} \\ \frac{Tr(H)^2}{Det(H)} &= \frac{(r \cdot \beta + \beta)^2}{r \cdot \beta^2} \\ \frac{Tr(H)^2}{Det(H)} &= \frac{(r + 1)^2}{r} \end{aligned} \quad (3.13)$$

Denklem 3.14'ün sağlandığı yerler köşe olarak adlandırılır ve diğer yerler de elenir. Lowe çalışmalarında $r = 10$ olarak almıştır[19].

$$\frac{Tr(H)^2}{Det(H)} < \frac{(r + 1)^2}{r} \quad (3.14)$$

Köşe dışındaki düz alanlar ve kenarların tespit edilip elenmesiyle anahtar nokta sayıları azaltılarak algoritmada verimlilik ve sağlamlık sağlanır.

3.3 Yönelimlerin Atanması

Her anahtar noktaya yerel özelliklerine bağlı yönelim atayarak görüntü dönmelerinden kaynaklanan bağımlılık ortadan kaldırılabilir. Seçilen anahtar noktanın ölçek bilgisi kullanılarak elde edilen bulanıklaştırılmış görüntünün, anahtar noktanın yöneliminin hesaplanması için kullanılması ile tüm hesaplamalar ölçekten bağımsız şekilde gerçekleştirilmiş olur. Belirlenen anahtar noktalar etrafındaki eğimlerin yönü ve

genlikleri kullanılarak en belirgin yönelim seçilir ve o anahtar noktaya ait yönelim olarak atanır. Anahtar noktanın belirlendiği ölçekteki tüm $L(x, y)$ görüntüleri için, eğim genlik değeri olan $m(x, y)$ Denklem 3.15'teki gibi ve yönelimi olan $\theta(x, y)$ Denklem 3.16'daki gibi hesaplanır.

$$m(x, y) = \sqrt{(L(x + 1, y) - L(x - 1, y))^2 + (L(x, y + 1) - L(x, y - 1))^2} \quad (3.15)$$

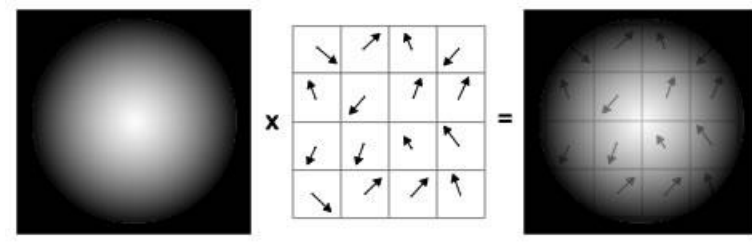
$$\theta(x, y) = \tan^{-1} ((L(x, y + 1) - L(x, y - 1)) / (L(x + 1, y) - L(x - 1, y))) \quad (3.16)$$

Anahtar nokta etrafındaki tüm piksellerin eğim yönelimleri kullanılarak, bir yönelim histogramı oluşturulur. Histogram, her biri diğerinden 10° açı farkına sahip olmak üzere 36 basamağa bölünmüş 360° 'lik alanı kapsar. Her nokta, eğim genliğine göre ağırlıklandırılarak histograma eklenir. Histogram, anahtar noktanın bulunduğu ölçek olan σ 'nın 1,5 katı kadar genişlikteki Gauss dairesel penceresi içindeki örnek noktaların eğim genliği eklenmesi ile oluşturulur. Örneğin pencere içerisindeki pikselin yönelimine en yakın basamağın değerine o pikselin genliği eklenir. Bu işlem pencere içerisindeki tüm pikseller için uygulanır. Oluşturulan histograma, bir veya birden fazla tepe noktası içeriyor olabilir. Bu durumda en yüksek tepe noktasına sahip pikselin yönelim açısı baskın değeri vermektedir, bu değer anahtar noktaya atanır, ancak bu değer %80'inden büyük diğer tepe noktalarına sahip yönelimler için aynı noktada fakat farklı yönelime sahip anahtar noktalar oluşturulur. Çoklu yönelime sahip anahtar noktalar, tüm anahtar noktaların %15'inden fazla olamaz. Son olarak, tepe noktasının yerini daha sağlam elde edebilmek için her tepe noktasına en yakın 3 adet histogram değerine bir parabol uygulanır.

3.4 Yerel İmge Tanımlayıcı

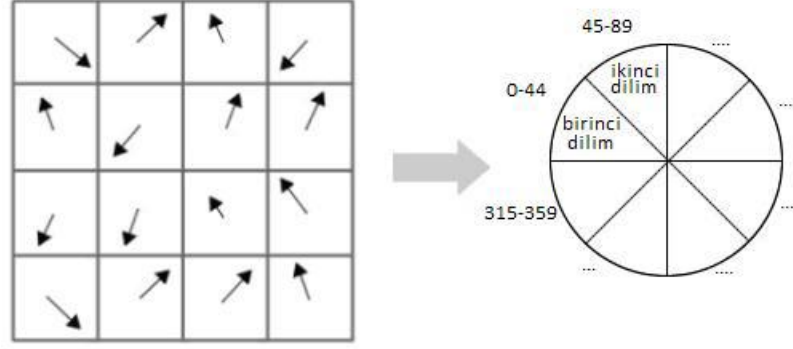
Her bir anahtar noktayı, diğer anahtar noktalardan ayırt edici olarak tanımlamak için, noktanın bulunduğu bölgenin özelliklerinden faydalanılır. Anahtar noktanın bulunduğu yerin etrafındaki bölge, bir Gauss penceresiyle ağırlıklandırılır ve bu bölgede yer alan her örnek nokta için eğim büyüklüğü ve yönelim hesaplanarak

anahtar nokta tanımlayıcı bulunur. Şekil 3.3'te anahtar noktanın gauss ağırlık fonksiyonu ile işlenmesi gösterilmiştir.



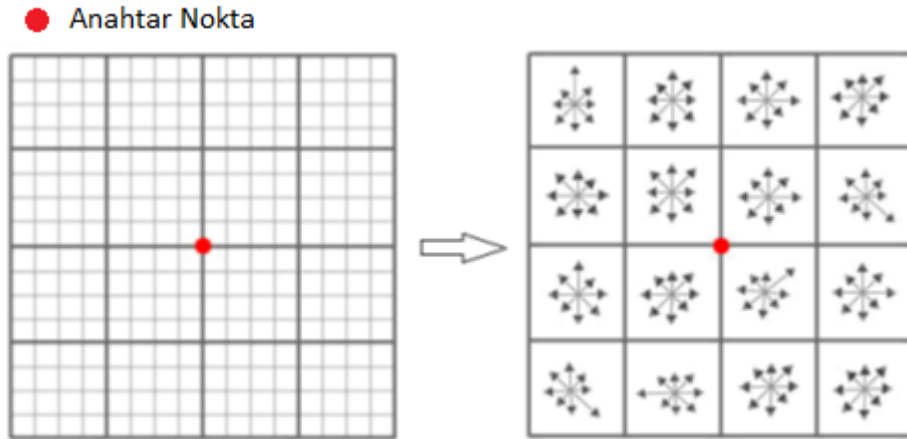
Şekil 3.3 Anahtar Noktanın Gauss Ağırlık Fonksiyonu İle İşlenmesi

Gauss penceresindeki σ ölçeği, daireyle çevrilerek gösterilen tanımlayıcının genişliğinin yarısına eşittir. Gauss penceresi kullanılmasının amacı, pencerenin konumunda oluşan küçük değişikliklerden dolayı meydana gelen tanımlayıcıdaki ani değişikliklerden kaçınmayı sağlamaktır ve tanımlayıcının merkezinden uzakta olan eğimlere daha az vurgu yapmaktır. Nokta, anahtar noktadan uzaklaştıkça histograma eklenen değeri de küçülür. Bu işlem “Gauss Ağırlık Fonksiyonu” ile yapılır. Anahtar nokta etrafında alınan 16×16 ’lık bölge, 4×4 ’lük çerçevelerden oluşan 16 alt bölgeye ayrılır. Her bir alt bölge içindeki bölgesel eğim yönelimlerinin histogramı hesaplanır. Her okun uzunluğu, bölge içindeki o yönün yanındaki eğim büyüklüğünün toplamına karşılık gelir. 16×16 Örnek dizisinden 4×4 tanımlayıcılar hesaplanır. 4×4 Boyutlu her bir çerçevede eğim büyüklük ve yönelimleri hesaplanır. Bu yönelimler 45° açı farkı olan 8 dilimlik bir histograma, genlikleri de dikkate alınarak Şekil 3.4’deki gibi yerleştirilir.



Şekil 3.4 Yönelimlerin Histograma Yerleştirilmesi

Böylece her anahtar nokta için $4 \times 4 \times 8 = 128$ elemanlı öznitelik vektörü oluşturulur ve şekil 3.5'teki gibi gösterilebilir.



Şekil 3.5 Anahtar Nokta Etrafında Seçilen Pencerenin Vektöre Dönüşmesi

Tanımlayıcının boyutu yüksek görünse de eşleştirme görevlerinde daha düşük boyutlu tanımlayıcılardan daha iyi çalıştığı ve eşleştirmedeki hesapsal maliyetin düşük olduğu gözlenmiştir[19].

Işık değişimindeki etkileri azaltmak için öznitelik vektörü birim uzunlukla normalize edilerek, kontrast değişimi gerçekleşmemiş olur. Bu işlem, her piksel değeri bir sabitle

arpılarak ve eđimler de yine aynı sabitle arpılarak yapılır. Her imge pikseline bir sabit eklenmesiyle oluşan parlaklık farkı, piksel farklarından hesaplandığı için eđim deęerlerini etkilemez.

Ancak, grntleme cihazının doyuma girmesi veya  boyutlu yzeylerin farklı ynelmelerde farklı deęişimler gstermesi ışık deęişiminin doęrusal olmamasına sebep olabilir. Bu gibi durumlarda eđim byklklerinde ynelim aılarına nazaran byk deęişikliklere sebep olabilir. Bu yzden eđim byklklerinin etkisini azaltmak için birim znelik vektrndeki deęerler 0,2 eşik deęeriyle karşılaştırılır, byk deęerler elenir ve yeniden birim uzunluęa normalize edilir.0,2 Eşik deęeri aynı  boyutlu nesne için farklı ışık deęerleri altında uygulama yapılarak deneyimsel olarak bulunmuştur[19]. nceki adımlarda her anahtar noktaya yer, lek ve ynelim atanarak yer, lek ve dnme işleme karşı baęımsız olması saęlanırken, bu blmde her anahtar nokta için bir tanımlayıcı vektr bularak ışık deęişimi, 3d bakış aısı gibi kalan zelliklere karşı baęımsız olması saęlanmıştır.

4. GELİŞTİRİLEN AKILLI KÖŞE BULMA YÖNTEMİ

4.1 Yapay Sinir Ağları

İnsan beyninin öğrenme özelliğini örnek alan, biyolojik sinir ağlarından esinlenen, benzer performans özellikleri gösteren Yapay Sinir Ağları (YSA) için bilgi işleme sistemidir denilebilir [34-35]. Biyolojik yapıdaki sinir hücreleri olan nöronlar, birbirlerine bağlanarak sinir sistemini oluştururlar. Sinir sistemi öğrenme, hafızaya alma ve veriler arasındaki ilişkiyi ortaya çıkarma kapasitesine sahip olan bir ağa benzetilebilir. Bu biyolojik yapıya benzeyen YSA, ağırlıklı bağlantılar olarak adlandırılan tek yönlü iletişim kanalları aracılığıyla birbirleriyle haberleşen, her elemanı kendi hafızasına sahip olan paralel ve dağıtık bilgi işlem yapılarıdır[36]. Öğrenme aşamasından sonra bilgiyi toplayıp, hücreler arasındaki ağırlıklar ile bilgiyi saklayan ve genelleme yaparak paralel dağıtık işlemci gibi çalışan YSA, kendisine gelen bir problemle ilgili sistemin ürettiği çıktı ile beklenen değer arasındaki benzerlik ilişkisini en güzel tanımlayacak uygun ağırlıkları bulur. Örneklerden faydalanarak öğrenmeye çalışır ve eğitim sırasında karşılaşmadığı test örnekleri için de istenen tepkiyi üretme kabiliyetine sahip olur.

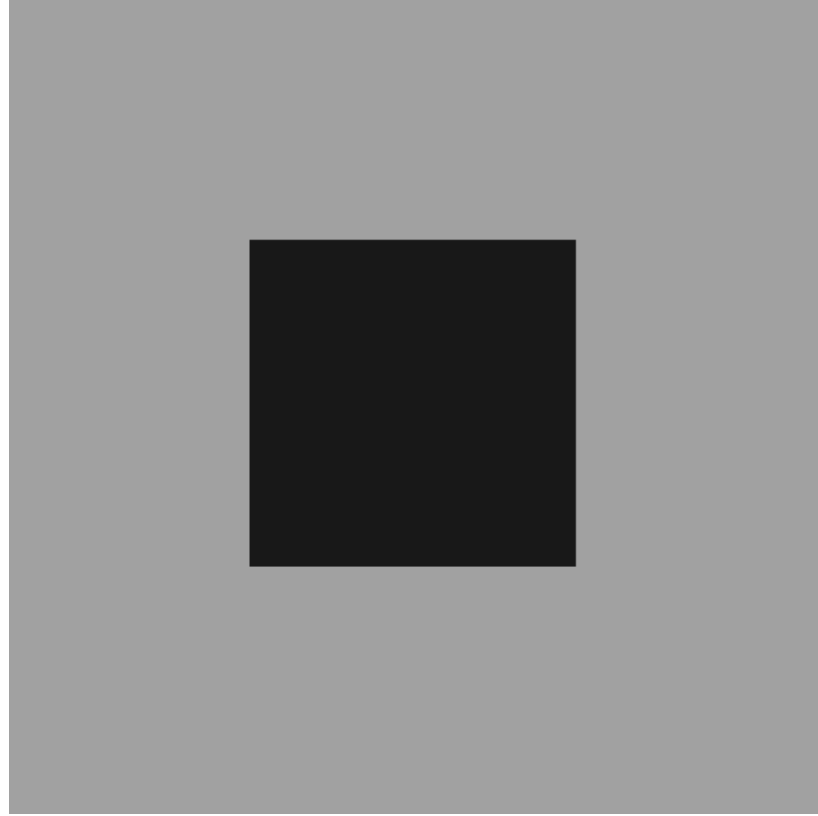
YSA, gürültüyü tolere edebilmektedir, veri setinde gürültü olsa bile, ağırlıkların üzerine dağıtarak gürültünün etkisi azaltılmaktadır. Bu nedenle hatayı tolere etme özelliği daha fazladır. Paralel dağıtık yapısı, öğrenme ve genelleme yeteneği ile hatayı tolere etme özelliği sayesinde YSA hesaplama ve bilgi işleme açısından güçlü bir yapıdadır. Bu sayede, YSA, hızlı bilgi işleme yeteneği gerektiren uygulamalarda ve örnek tanıma, işaret işleme, sistem kimliklendirme gibi gerçek zamanlı uygulamalarda kullanım imkanı bulur.

YSA üstün özellikleri ile sınıflandırma, kümeleme, vektör sayısallaştırılması, desen uygunluğu, fonksiyon yaklaşımı, tahmin yapmak, kontrol sorunlarını çözmek, en iyileme, arama çalışmaları gibi işlemleri yerine getirmektedir [37]. YSA mühendislik, imalat, tıbbi görüntü işleme ve tanı koyma gibi sağlık alanlarında, biyomedikal uygulamalarda, tarımsal alanda, askeri alan gibi birçok alanda kullanılmaktadır.

4.2 Eğitim Kümesinin İnşası

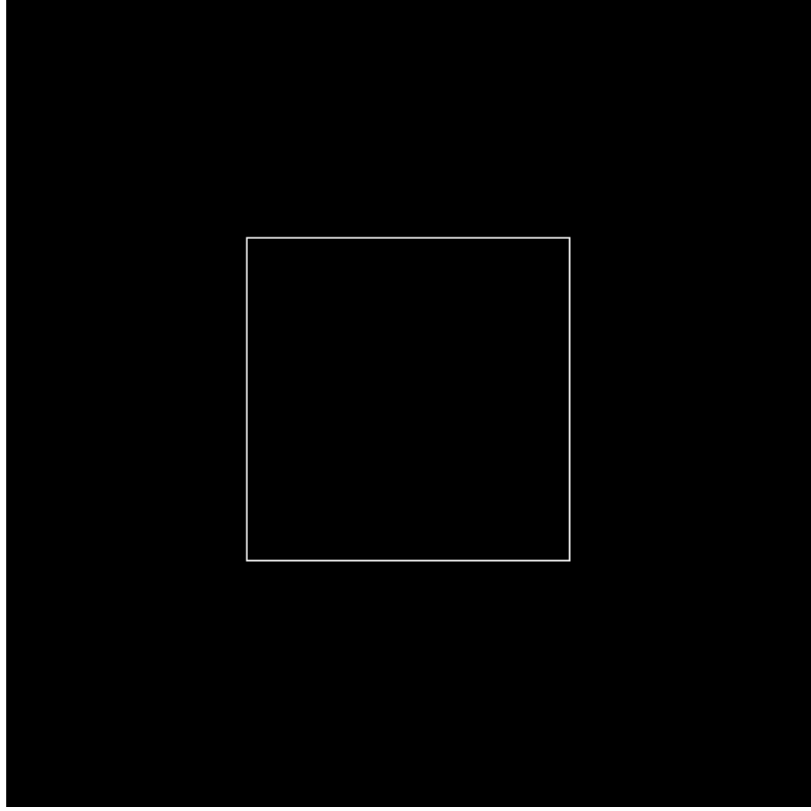
Köşe pikselinin yakalanabilmesi için köşe ve köşe olmayan görüntü parçalarının eğitim kümesinde tanımlanmış olması gerekir. Bu nedenle MATLAB ortamında hazırlanan bir yazılım yardımıyla belirli nesnelere üretilip, bu nesnelerin köşe olan (1) ve olmayan (0) parçaları eğitim kümesine kaydedilmiştir. Geliştirilen yazılım aşağıda ifade edilen aşamaları içermektedir:

1. 500x500 ebatlarında gri tonu rastgele belirlenen bir görüntü inşa et.
2. Şekil 4.1'deki gibi merkezine 200x200 ebatlarında gri tonu rastgele belirlenen kare yerleştir.



Şekil 4.1 Eğitim Kümesi İçin Oluşturulan Kare Görüntüsü

3. Şekil 4.2'deki gibi 200x200 Ebatlarındaki karenin kenar bilgisini elde et.



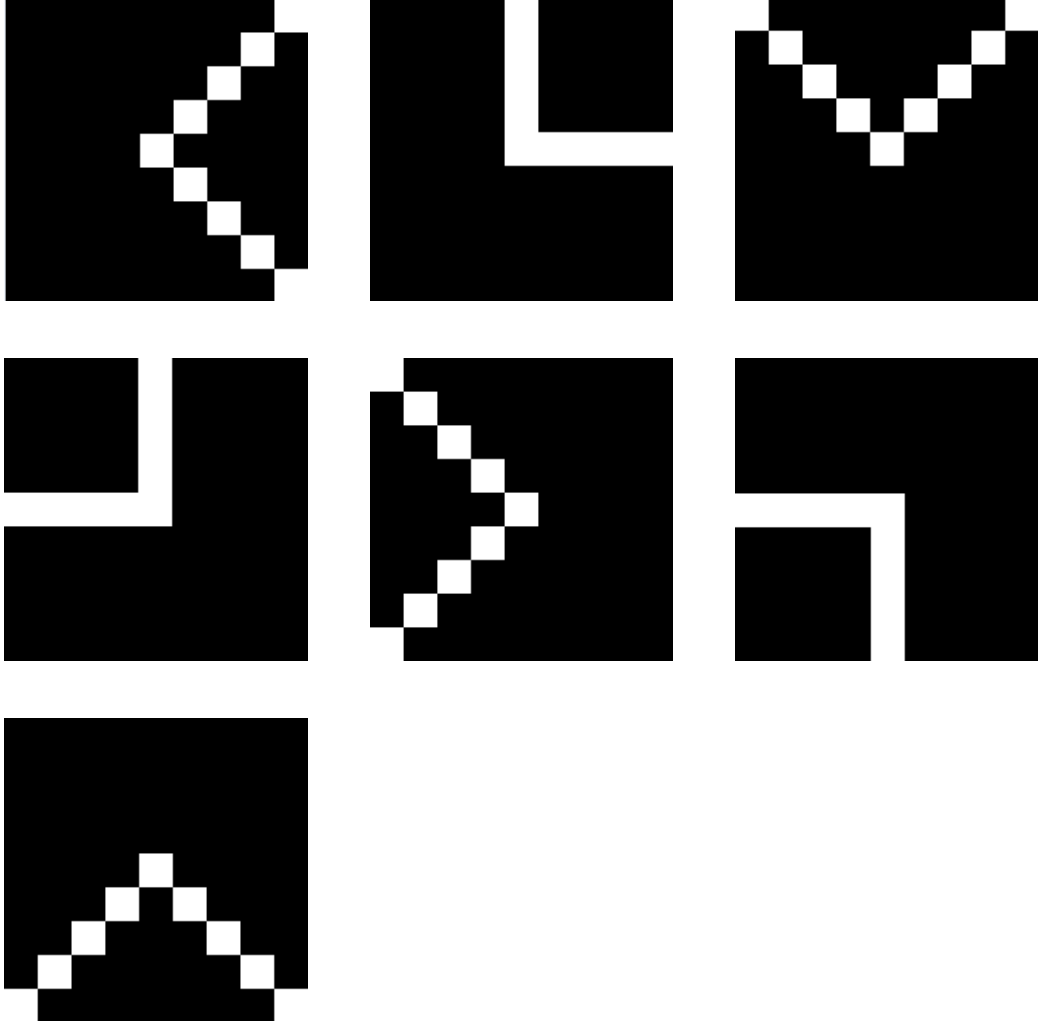
Şekil 4.2 Kenar Bilgisi Elde Edilen Kare Görüntüsü

4. Kenar bilgisi elde edilen kare üzerinde sol üst köşe parçanın (1) 9x9 ebatlarında örneğini oluştur ve kaydet. Bu örnek Şekil 4.3'te gösterilmiştir.



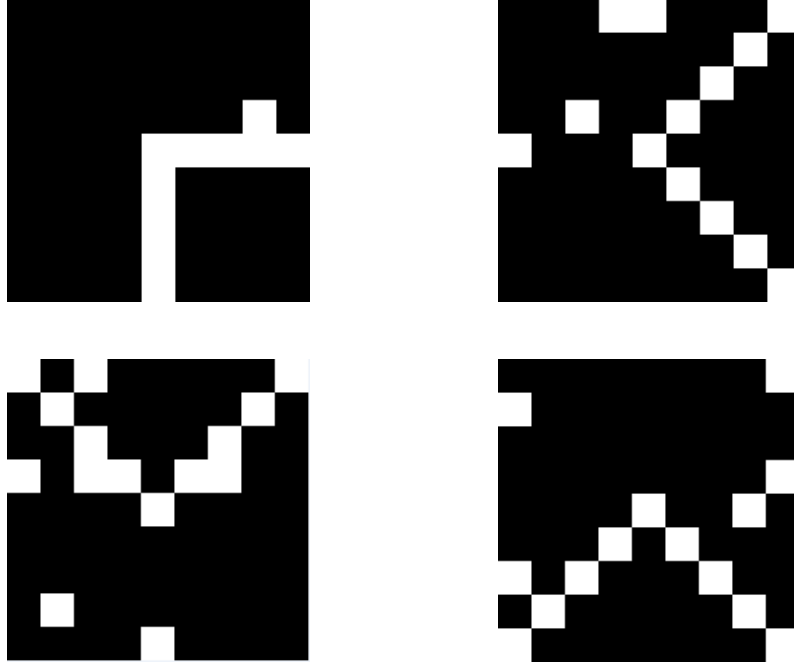
Şekil 4.3 Köşe Olan Parçanın 9x9 Ebatlarında Örneği

5. Köşe olan parçayı 45° çevirerek farklı örnekler elde et ve kaydet. Şekil 4.4'te köşe görüntüsünün 45° çevrilmesiyle elde edilen tüm örnekler gösterilmiştir.



Şekil 4.4 Köşe Görüntüsünün 45° Çevrilmesiyle Elde Edilen Örnekler

6. Köşe olan parçaya farklı gürültü seviyeleri ekleyerek yeni örnekler elde et ve kaydet. Gürültülü örneklerin birkaçı Şekil 4.5'te gösterilmiştir.



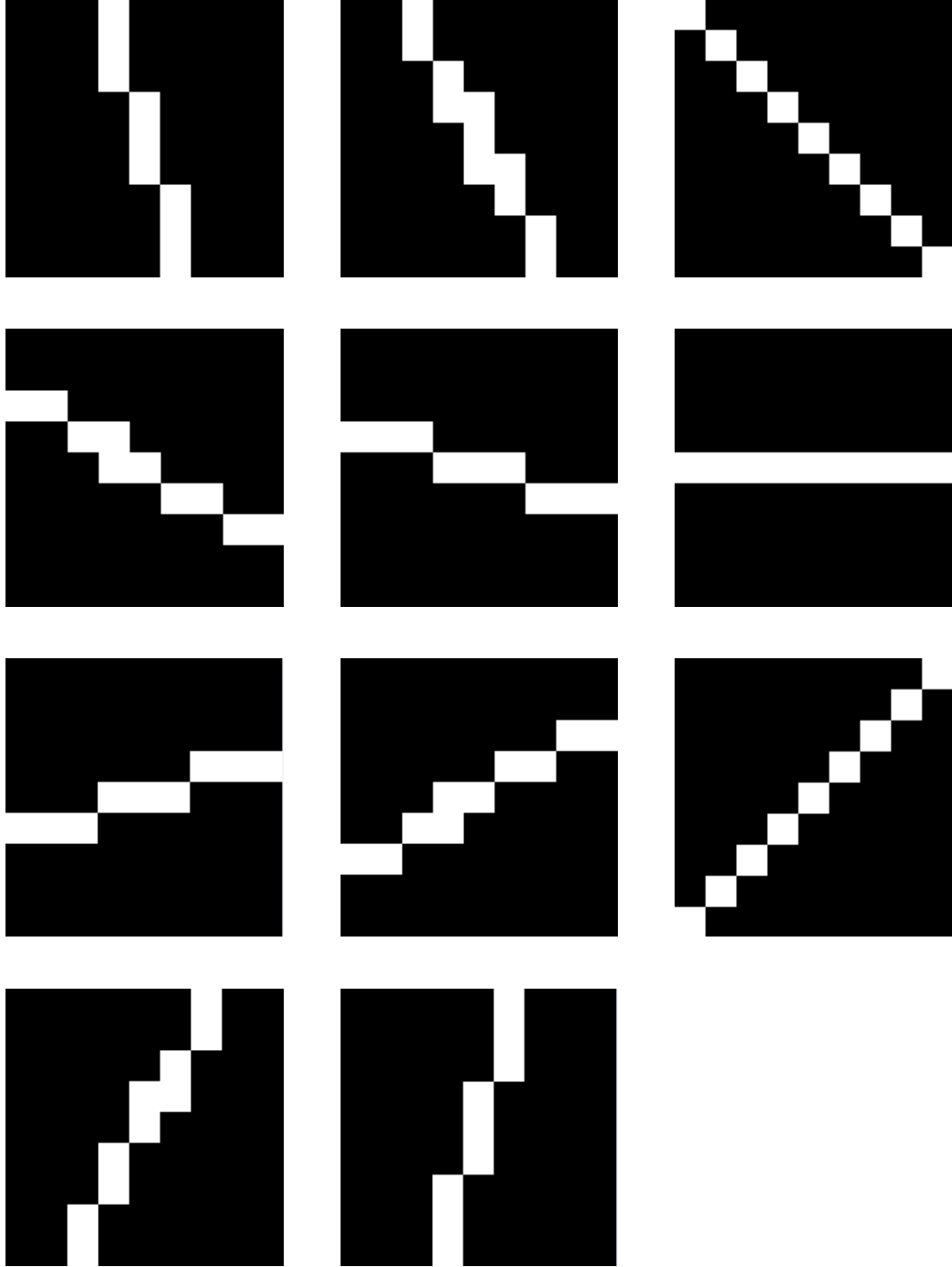
Şekil 4.5 Köşe Görüntülerine Gürültü Eklenmesi Sonucu Elde Edilen Örnekler

7. Kenar bilgisi elde edilen kare üzerindeki köşe olmayan parçanın (0 grubu) 9x9 ebatlarında örneğini oluştur ve kaydet. Bu örnek Şekil 4.6'da gösterilmiştir.



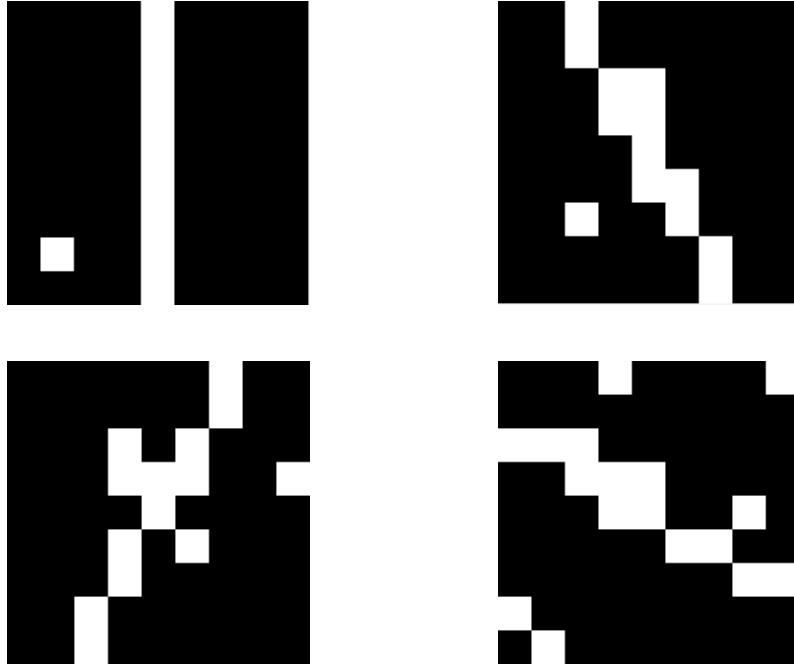
Şekil 4.6 Köşe Olmayan Parçanın 9x9 Ebatlarında Örneği

8. Köşe olmayan parçayı 15° çevirerek farklı örnekler elde et ve kaydet. Elde edilen örneklerin tümü Şekil 4.7’de gösterilmiştir.



Şekil 4.7 Köşe Olmayan Parçanın 15° Çevrilmesiyle Elde Edilen Örnekler

9. Köşe olmayan parçaya farklı gürültü seviyeleri ekleyerek yeni örnekler elde et ve kaydet. Köşe olmayan gürültülü örneklerin birkaçı Şekil 4.8’de gösterilmiştir.



Şekil 4.8 Köşe Olmayan Görüntülere Gürültü Eklenmesiyle Elde Edilen Örnekler

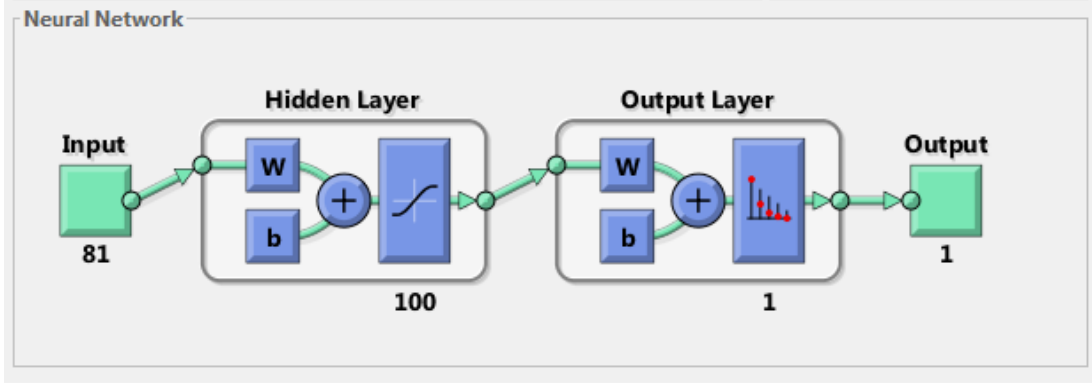
10. Daha fazla örnek oluşturmak amacıyla gürültü işlemlerini 10 defa tekrar et.

Bu işlemler 9x9 piksel boyutundaki kare görüntü için çalıştırılmış ve sonuç olarak 728 adet pozitif, 1092 adet negatif görüntü parçası olmak üzere 1820 adet görüntü parçası eğitim kümesine kaydedilmiştir.

4.3 Ağ Mimarisi

Eğitim işlemini gerçekleştirebilmek için üç katmanlı bir yapay sinir ağı kullanılmıştır. Giriş katmanı 81, gizli katman 100 ve çıkış katmanı tek hücre içermektedir. Elde edilen pozitif ve negatif görüntü parçaları (9x9) vektörel hale dönüştürüldükten sonra (81x1) ağına girişine verilmekte ve ilgili görüntü parçasının köşe olup olmasına göre çıkış değer ayarlanmaktadır. Rastgele ağırlık değerlerinin güncellenebilmesi için

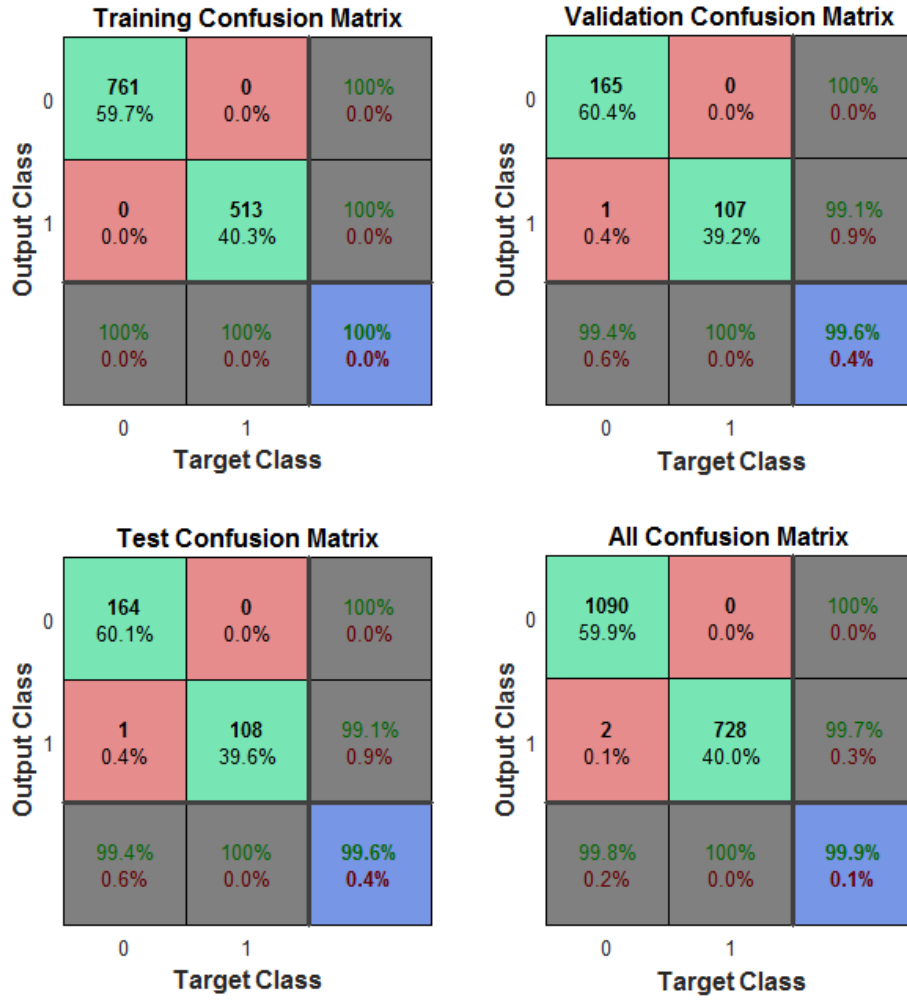
geriye yayılım yöntemi kullanılmıştır. Kullanılan üç katmanlı yapay sinir ağı Şekil 4.9'da gösterilmiştir.



Şekil 4.9 Üç Katmanlı Yapay Sinir Ağı

4.4 Eğitim Sonuçları

Eğitim kümesinde 1820 adet görüntü parçası bulunmaktadır. Bu kümenin %70'i eğitim örneği (1274 adet), %15'i doğrulama örneği (273 adet) ve %15'i test örneği olarak (273 adet) kullanılmıştır. Negatif görüntü parçaları 0, pozitif görüntü parçaları 1'i temsil etmek üzere sınıflandırılmış olan eğitim kümesi ilk olarak eğitime tabi tutulmuş, 0 olup çıktı olarak 0 sonucunu üreten 761 adet, 1 olup 0 sonucunu üreten 0 adet, 0 olup 1 sonucunu üreten 0 adet, 1 olup 1 sonucunu üreten 513 adet görüntü kaydedilmiş, eğitim sonucu % 100 başarılı olmuştur. Doğrulama adımında 273 adet örnek kullanılmış, bunlardan 0 olup 0 sonucunu üreten 165 adet, 1 olup 1 sonucunu üreten 107 adet, 0 olup 1 sonucunu üreten 1 adet ve 1 olup 0 sonucunu üreten 0 adet görüntü kaydedilmiş, doğrulama sonucu % 99,6 başarılı olmuştur. Test adımında 273 adet örnek kullanılmış, bunlardan 0 olup 0 sonucunu üreten 164 adet, 1 olup 1 sonucunu üreten 108 adet, 0 olup 1 sonucunu üreten 1 adet ve 1 olup 0 sonucunu üreten 0 adet görüntü kaydedilmiş, doğrulama sonucu % 99,6 başarılı olmuştur. Sonuç olarak yapay sinir ağı 0 olup 0 sonucunu üreten 1090 adet, 1 olup 1 sonucunu üreten 728 adet, 0 olup 1 sonucunu üreten 2 adet, 1 olup 0 sonucunu üreten 0 adet görüntü kaydedilmiş, yapay sinir ağı % 99,9 oranında başarılı olmuştur. Bu sonuçlar Şekil 4.10'daki Karışıklık Matrisi çizelgesinde gösterilmiştir.

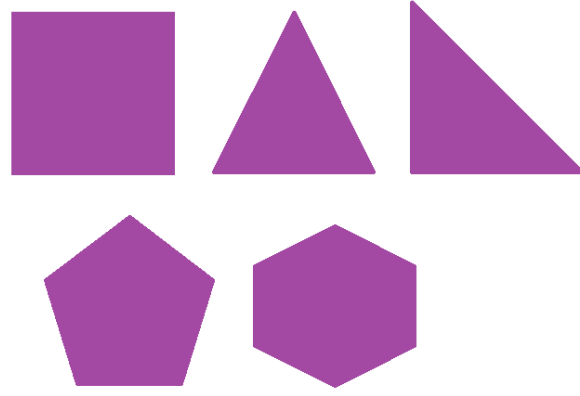


Şekil 4.10 Matlab Yapay Sinir Ağı Karışıklık Matrisi

4.5 Test Sonuçları

Köşe bulma yöntemlerinden Moravec, Harris, Susan yöntemleri ve kendi oluşturduğumuz yöntem 5 farklı sentetik görüntüde denenmiş, test sonuçları görüntü ile birlikte gösterilmiştir.

Şekil 4.11’de yer alan örnek sentetik görüntüde geometrik şekiller çizilmiş, Harris ve Uygulanan Yöntem ile iyi sonuçlar alındığı gözlenmiştir. Üçgen şekillerinin olması Moravec ve Susan Yöntemleri için köşe bulmayı zorlaştırmış, birden fazla noktanın köşe olarak algılandığı gözlemlenmiştir.



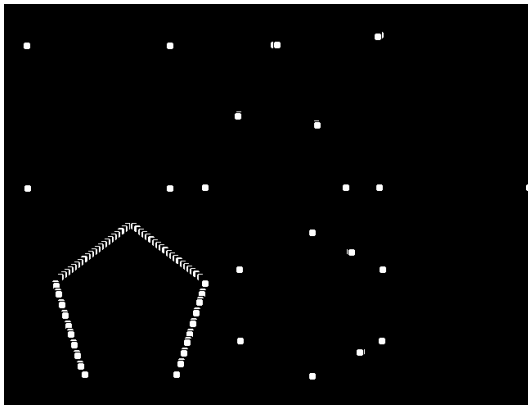
1. Sentetik Görüntü



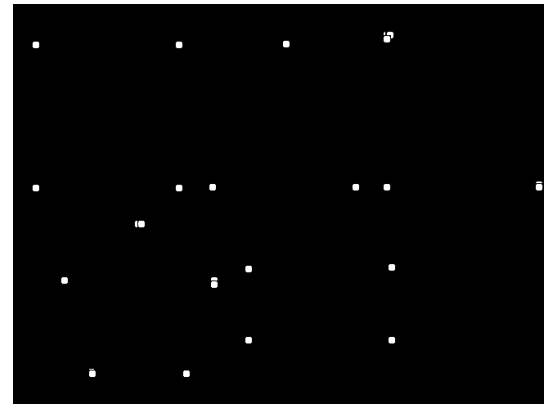
Moravec Yöntemi



Harris Yöntemi



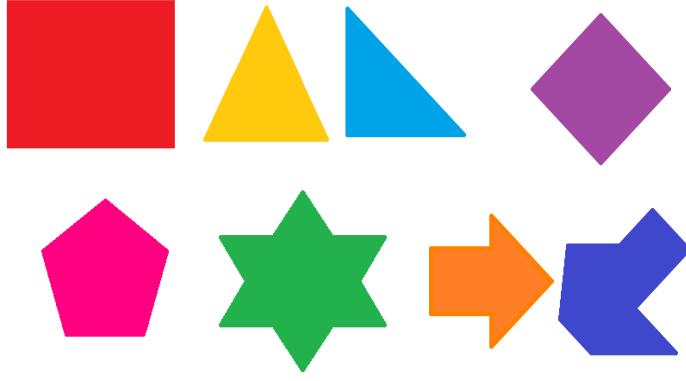
Susan Yöntemi



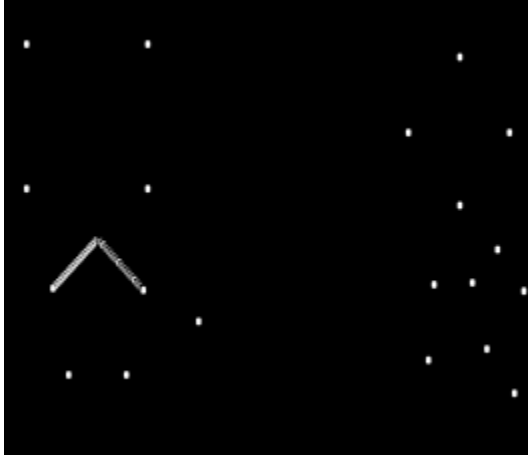
Uygulanan Yöntem

Şekil 4.11 1. Sentetik Görüntü ve Köşe Bulma Yöntemleri Sonuçları

Şekil 4.12’de geometrik şekiller ve bu şekillerin birleşimiyle elde edilen şekillerden oluşan sentetik görüntü oluşturulmuş, Moravec Yöntemi’nin yine üçgen şeklini tespit etmede yeterli olmadığı, ayrıca bazı şekillerin köşelerini hiç bulamadığı gözlemlenmiştir. Harris Yöntemi’nin bu görüntü için diğer yöntemlere göre daha başarılı olduğu görülmüş, Uygulanan Yöntem’in ise bazı kenarları da köşe olarak algıladığı tespit edilmiştir.



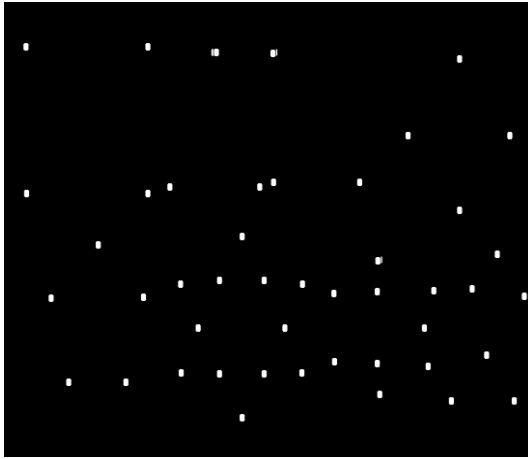
2. Sentetik Görüntü



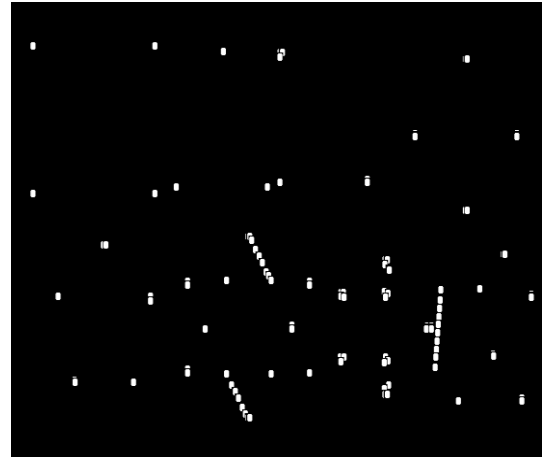
Moravec Yöntemi



Harris Yöntemi



Susan Yöntemi



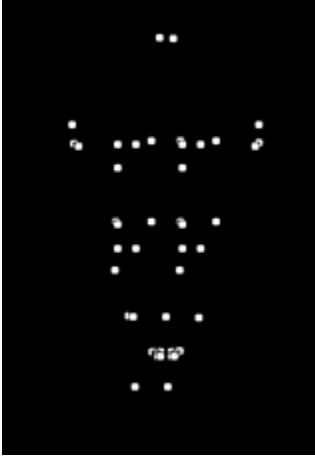
Uygulanan Yöntem

Şekil 4.12 2. Sentetik Görüntü ve Köşe Bulma Yöntemleri Sonuçları

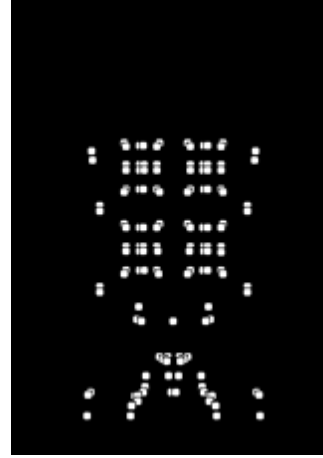
Şekil 4.13'te sentetik ev görüntüsü elde edilmiş, diğer geometrik şekillerde yer alan sivri köşeler yerine yuvarlak köşeler kullanıldığından yöntemlerin bu yuvarlak köşeleri algılaması incelenmiştir. Yuvarlak köşeler için bulunması gereken birden fazla nokta olması gerektiği öngörülmüş, Harris Yöntemi'nin yuvarlak köşeleri hiç bulmadığı, Moravec ve Susan Yöntemi'nin kısmi olarak tespit ettiği, Uygulanan Yöntem'in ise birden fazla nokta ile bulduğu gözlenmiştir.



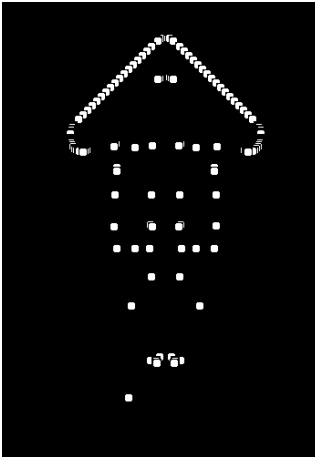
3. Sentetik Görüntü



Moravec Yöntemi



Harris Yöntemi



Susan Yöntemi



Uygulanan Yöntem

Şekil 4.13 3. Sentetik Görüntü ve Köşe Bulma Yöntemleri Sonuçları

Şekil 4.14'te köşeli kenarlara sahip ev görüntüsü örnek olarak alınmış, görüntü üzerinde çeşitli gürültüler bulunmasından dolayı yöntemlerin çok iyi sonuç vermediği gözlenmiştir.



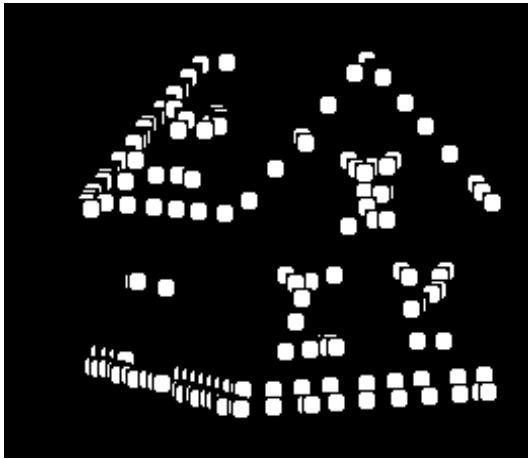
4. Sentetik Görüntü



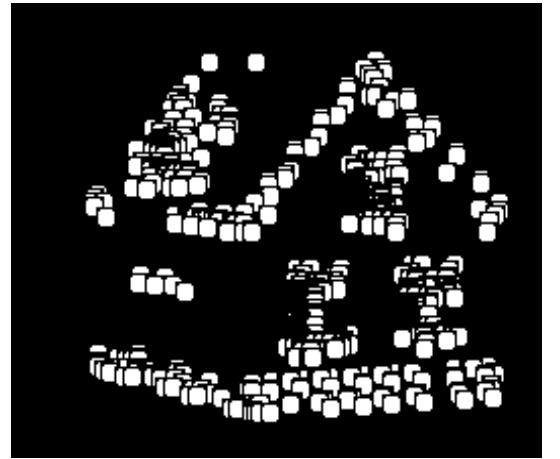
Moravec Yöntemi



Harris Yöntemi



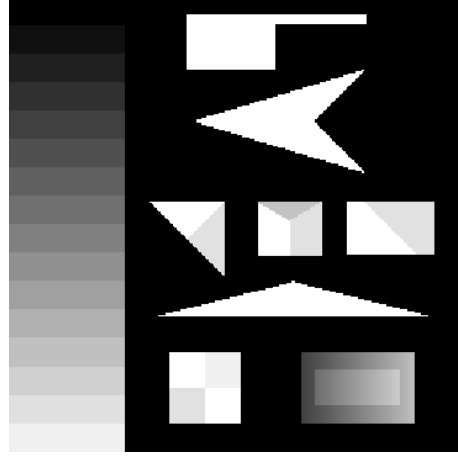
Susan Yöntemi



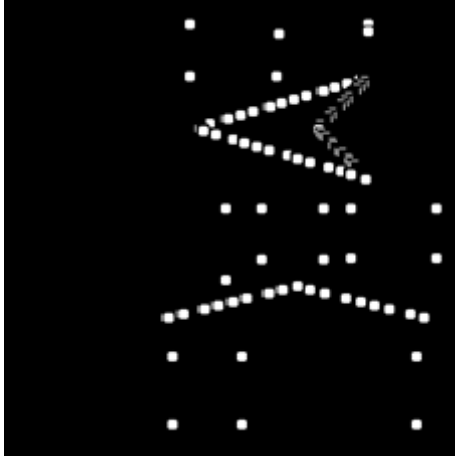
Uygulanan Yöntem

Şekil 4.14 4. Sentetik Görüntü ve Köşe Bulma Yöntemleri Sonuçları

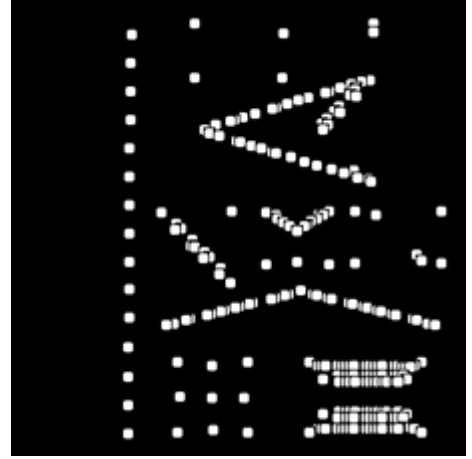
Şekil 4.15'te gri tonlamalı sentetik görüntü için Susan Yöntemi'nin en iyi sonucu ürettiği, diğer yöntemlerin de gri tonlamanın geçiş yaptığı yerlerde başarısız olduğu gözlenmiştir.



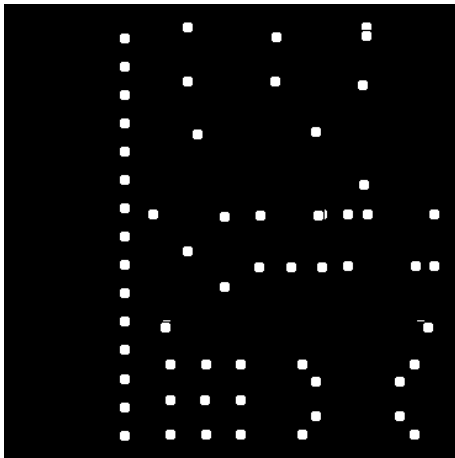
5. Sentetik Görüntü



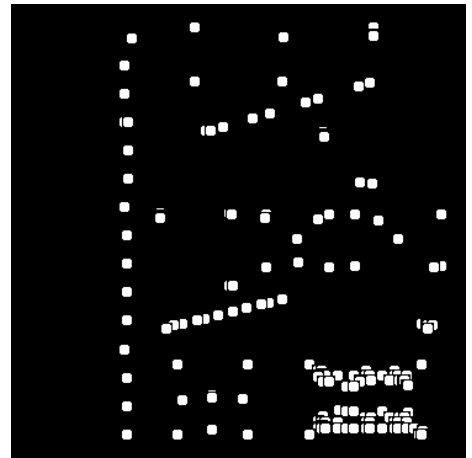
Moravec Yöntemi



Harris Yöntemi



Susan Yöntemi

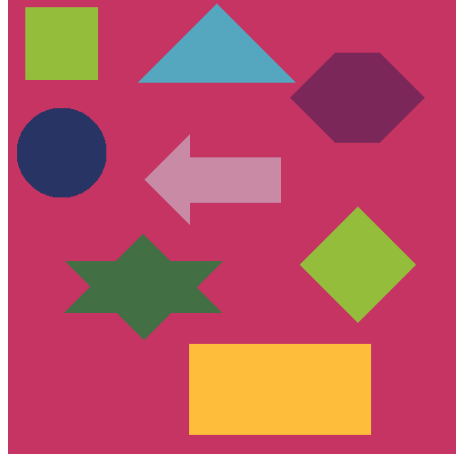


Uygulanan Yöntem

Şekil 4.15 5. Sentetik Görüntü ve Köşe Bulma Yöntemleri Sonuçları

5. KARŞILAŞTIRMA SONUÇLARI

Matlab ortamında köşe noktaları bilinen sentetik görüntü inşa edilerek köşe bulma yöntemleri ve oluşturduğumuz yöntem ile uygulandığında Şekil 5.1'deki görüntüler elde edilmiştir.



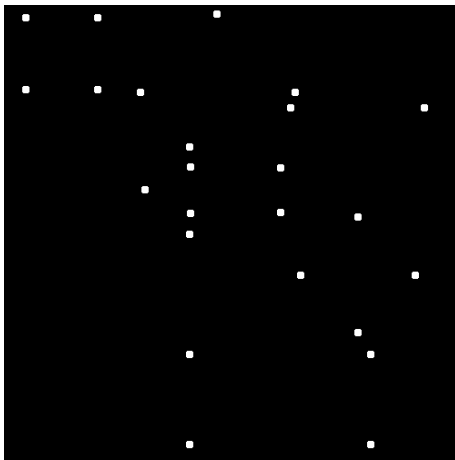
Örnek Sentetik Görüntü



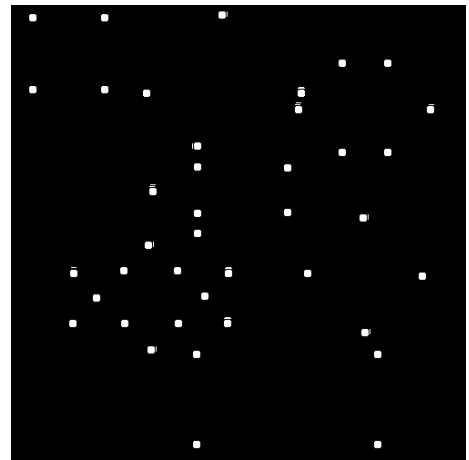
Moravec Yöntemi



Harris Yöntemi



Susan Yöntemi



Uygulanan Yöntem

Şekil 5.1 Örnek Görüntü ve Köşe Bulma Yöntemleri Sonuçları

Oluşturulan örnek sentetik görüntü için 44 adet köşe sayısı tespit edilmiştir. Algoritmalar uygulandığında Moravec Yöntemi ile 49 adet, Harris Yöntemi ile 36 adet, Susan Yöntemi ile 38 adet, geliştirdiğimiz yöntem ile 65 adet köşe bulunmuştur. Bulunan köşe sayılarından Moravec Yöntemi ile 10 adet, Harris Yöntemi ile 16 adet, Susan Yöntemi ile 24 adet, geliştirdiğimiz yöntem ile 23 adet doğru köşe noktası tespit edilmiştir. Bulunan köşe sayıları ve doğru köşe sayılarına ait çizelge aşağıda gösterilmiştir.

Çizelge 5.1 Köşe Sayı Bilgileri

	Bulunan Köşe Sayısı	Doğru Köşe Sayısı
Örnek Sentetik Görüntü	44	44
Moravec Yöntemi	49	10
Harris Yöntemi	36	16
Susan Yöntemi	38	24
Uygulanan Yöntem	65	23

Algoritmalar ile bulunan köşe bilgisinin doğruluğu hakkında bilgi almak amacıyla görüntü kalite metrikleri kullanılmaktadır. Görüntü kalite metrikleri, görüntü kalitesinin skoru hakkında objektif sonuç sağlamaktadır. Kullanılan yöntemlerle giriş görüntüsü ve algoritma sonucu elde edilen görüntü kıyaslanmaktadır. Bu yöntemlerde orijinal görüntü ile algoritma sonucu elde edilen görüntü arasındaki piksel sayıları, kontrast, parlaklık ve yapı özellikleri gibi bilgiler karşılaştırılmaktadır.

Sonuçların doğruluğunu ölçmede kullanılan yöntemlerden biri olan Dice Benzerlik Endeksi [38], sonuç kümesi olan X ile algoritma tarafından bulunan Y kümesinin kesişim alanını her bir kümenin alanının ortalama toplamına bölerek iki küme arasındaki benzerliği bulmaktadır. Dice benzerlik endeksi, $[0, 1]$ aralığında değerler almakta olup, 1 değeri X ve Y kümelerinin tamamen aynı olduğunu göstermektedir ve Denklem 5.1'deki formül ile hesaplanmaktadır.

$$D(X, Y) = \frac{2|X \cap Y|}{|X| + |Y|} \quad (5.1)$$

X ve Y kümesi piksel değerleri arasındaki hata oranı Ortalama Hata Karesi (Mean-Squared Error) yöntemiyle hesaplanabilmektedir. En küçük hata oranına sahip değer, ulaşılmak istenen değerdir.

Tepe Sinyal Gürültü Oranı (PSNR), iki küme arasındaki benzerlik hakkında bilgi vermektedir ve oranın büyüklüğü benzerliğin fazla olduğunu göstermektedir. Denklem 5.2’de de gösterildiği gibi bu yöntemde Ortalama Hata Karesi (MSE) kullanılmaktadır ve R ile ifade edilen değer görüntü piksellerinin alabileceği en büyük değeri göstermektedir [39].

$$PSNR = 10 \log_{10} \frac{R^2}{MSE} \quad (5.2)$$

Görüntü kalitesini anlamak için kullanılan diğer bir ölçü Yapısal Benzerlik Endeksi (Structural Similarity Index-SSIM) dir. SSIM, bir görüntünün algılanan görsel kalitesine ne kadar yaklaştığını gösterir. SSIM endeksi [0,1] arasında değer alır ve kalite arttıkça endeks değeri artar [40].

Dice Benzerlik Endeksi, MSE, PSNR ve SSIM ölçüleri algoritma sonuçları üzerinde kullanılmış ve Çizelge 5.2’deki sonuçlar elde edilmiştir. Bu çizelgeye göre kullanılan görüntü kalite metrik ölçüleriyle Susan Yöntemi en iyi sonuçları verirken, uyguladığımız yöntemin Moravec Yöntemi’nden daha iyi olduğu, Harris ve Susan Yöntemlerinden daha kötü sonuçlar elde ettiği görülmüştür.

Çizelge 5.2 Görüntü Kalite Metriklerine Göre Algoritma Sonuçları

	Dice	MSE	PSNR	SSIM
Moravec Yöntemi	0,2151	18,9873	35,35	0,9897
Harris Yöntemi	0,4	12,4848	37,17	0,9926
Susan Yöntemi	0,5854	8,8434	38,66	0,9941
Uygulanan Yöntem	0,422	16,3863	35,986	0,9918

6. SONUÇLAR VE DEĞERLENDİRME

Bu çalışmada Ölçekten Bağımsız Öznitelik Dönüşümü (SIFT) metodu araştırılmış, bu metoda yenilik katmak amacıyla köşe bulma yöntemleri ve yapay sinir ağları birlikte kullanılarak yeni bir akıllı köşe bulma yöntemi geliştirilmiştir. Görüntü işlemede kullanılan öznitelik çıkarım yöntemlerinden SIFT, güçlü yöntemleri temel olarak kullanmasına ve çevrimiçi uygulamalarda ayırt edici ve nispeten hızlı olmasına rağmen eşleştirme aşamasında tanımlayıcının yüksek boyutlu olması (128 bit) büyük bir eksiklik olarak görülmüştür. Bulunan her bir anahtar nokta için 128 elemanlı öznitelik vektörü oluşturulmaktadır. Anahtar noktanın azaltılması, SIFT metodunu geliştirmek için kullanılabilir. Bu amaçla SIFT metodundaki anahtar noktaların bulunması aşamasında köşe noktaların tespitinin yapıldığı yöntemin değiştirilmesiyle bu metoda yenilik katılacağı öngörülmüştür.

Görüntü işlemede önemli bir yeri olmasına rağmen köşe tanımının yapılamamış olması, bu alanda pek çok çalışmanın ortaya konulmasına sebep olmuştur. Köşe bulma yöntemlerinde, gri-seviye yoğunluk değerleri üzerinde çalışmak, ilk işlem olarak başka bir yöntem uygulanmadığı için yöntem kaynaklı hatalardan uzak olacağından avantaj olarak görülmektedir. Bunun yanısıra köşe bulma yönteminde kenar bulma yöntemlerini önışlem olarak kullanmak, kenar belirlemede ortaya çıkan hatalar veya sapmalar sebebiyle güvenilir bulunmamaktadır. Yapılan çalışmada renkli görüntü gri-seviyeli görüntüye dönüştürülmüş, daha sonra kenar bulma yöntemi kullanılarak görüntünün kenarları çıkarılmıştır.

Elde edilen gri-seviyeli kenarları çıkarılmış görüntü üzerinden yapay sinir ağları uygulaması çalıştırılmıştır. YSA, öğrenme aşamasından sonra bilgiyi toplayıp saklayarak genelleme yapar ve kendisine gelen problemle ilgili sistemin ürettiği çıktı ile beklenen değer arasındaki benzerlik ilişkisini kurar. Böylece örneklerden faydalanarak öğrenmeye çalışır ve eğitim sırasında karşılaşmadığı test örnekleri için istenen sonucu üretmeye çalışır. YSA eğitim kümesi için 9x9 piksel boyutunda 728 adet pozitif, 1092 adet negatif olmak üzere toplam 1820 adet görüntü parçası oluşturulmuş, kullanılan üç katmanlı YSA ile eğitim %99,9 oranında başarılı olmuştur.

Test görüntüsü oluşturularak, köşe noktaları manuel tespit edilmiş ve köşe matrisi oluşturulmuştur. Geliştirdiğimiz akıllı köşe bulma yönteminde, görüntünün kenarları tespit edilmiş, daha sonra YSA'dan geçirilerek köşe noktaları tespit edilmiştir. Elde ettiğimiz sonuçlar kalite metrik ölçüleriyle değerlendirilmiş, kıyaslama yapabilmemiz için aynı test görüntüsü Moravec, Harris ve Susan Köşe Bulma Yöntemleri kullanılarak köşe sonuçları elde edilmiş ve kalite metrik ölçüleriyle değerlendirilmiştir.

Geliştirilen akıllı köşe bulma yönteminin diğer yöntemlere göre daha yavaş çalıştığı, daha fazla köşe sayısı bulduğu, buna rağmen isabet eden doğru köşe sayısının az olduğu, eğitim kümesindeki negatif örneklerin çeşitlendirilmesiyle köşe sayısında önemli derecede azalma olabileceği gözlenmiştir.

KAYNAKLAR

- [1] L. Rabiner and B. H. Juang, *Fundamentals of Speech Recognition*, Prentice Hall International, New Jersey, 1993
- [2] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, Prentice Hall International, New Jersey, 2002
- [3] S. Günal, S. Ergin, M. B. Gülmezoğlu, Ö. N. Gerek, “On feature extraction for spam e-mail detection”, **Lecture Notes in Computer Science**, 4105 (2006) 635–642.
- [4] L. Kitchen and A. Rosenfeld, “Gray level corner detection,” **Pattern Recognition Letters**, 1:2 (1982) 95–102.
- [5] N. Nain, V. Laxmi, A. K. Jain, and R. Agarwal, “Morphological edge detection and corner detection algorithm using chain-encoding,” Proceedings of the 2006 International Conference on Image Processing, Computer Vision, and Pattern Recognition, IPCV'06, January 26-29, (2006), pp. 520-525.
- [6] X. Chang, L. Gao, and Y. Li, “Corner detection based on morphological disk element”, 2007 American Control Conference, New York, (2007), pp. 1994-1999.
- [7] D. Parks and J. P. Gravel, “Corner detection,” International Journal of Computer Vision, 2004.
- [8] R. Lin, “A modified morphological corner detector,” **Pattern Recognition Letters**, 19:3 (1998) 279–286.
- [9] X. C. He and N. H. C. Yung, “Curvature scale space corner detector with adaptive threshold and dynamic region of support,” Proceedings of the 17th International Conference on Pattern Recognition (ICPR), IEEE Computer Society, Washington DC, (2004), pp. 791–794.
- [10] S. M. Smith and J.M.Brady, “SUSAN - A New Approach to Low Level Image Processing,” Technical. Report. TR95SMS1c, Oxford University, 23:1, (1995), pp. 1–59..
- [11] F. Y. Shih, C. F. Chuang, and V. Gaddipati, “A modified regulated morphological corner detector,” **Pattern Recognition Letters**, 26:7 (2005) 931–937.
- [12] R. Laganieri, “A morphological operator for corner detection,” **Pattern Recognition Letters**, 31:11 (1998) 1643–1652.
- [13] C. Tomasi, “Detection and Tracking of Point Features Technical Report CMU-CS-91-132,” Image Rochester NY, 91:4, (1991), pp. 1–22.
- [14] R. Deriche and G. Giraudon, “A computational approach for corner and vertex

- detection*,” **International Journal of Computer Vision**, 10:2 (1993) 101–124.
- [15] C. Harris and M. Stephens, “A combined corner and edge detector,” In Proceedings of the Fourth Alvey Vision Conference, Manchester, (1988), pp. 147–151.
- [16] H. Moravec, “Towards automatic visual obstacle avoidance,” Proceedings of the 5th International Joint Conference on Artificial Intelligence, Cambridge, (1977), pp. 584
- [17] A. Sobania and J. P. O. Evans, “Morphological corner detector using paired triangular structuring elements,” **Pattern Recognition Letters**, 38:7 (2005) 1087–1098.
- [18] R. Mehrotra, S. Nichani, and N. Ranganathan, “Corner detection,” **Pattern Recognition Letters**, 23:11 (1990) pp. 1223–1233.
- [19] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” **International Journal of Computer Vision**, 60:2 (2004) 91–110.
- [20] J. Matas, O. Chum, M. Urban, and T. Pajdla, “Robust wide-baseline stereo from maximally stable extremal regions,” **Image and Vision Computing**, 22:10 (2004) 761–767.
- [21] M. Güney and N. Arıca, “Desen tabanlı ilgi bölgesi tespiti,” **J. Nav. Sci. Eng.**, 5: 1 (2009) 94–106.
- [22] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Washington DC, (2005), pp. 886–893.
- [23] Y. Ke and R. Sukthankar, “PCA-SIFT: a more distinctive representation for local image descriptors,” IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Washington DC, (2004), pp. 506–513.
- [24] H. Bay, T. Tuytelaars, and L. Van Gool, “SURF: Speeded up robust features,” **Lecture Notes in Computer Science**, 3951 (2006) 404–417.
- [25] G. Takacs, V. Chandrasekhar, S. Tsai, D. Chen, R. Grzeszczuk, and B. Girod, “Unified real-time tracking and recognition with rotation-invariant fast features,” IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Washington DC, (2010), pp. 934–941.
- [26] E. Rosten, R. Porter, and T. Drummond, “{FASTER} and better: A machine learning approach to corner detection,” **Pattern Anal. Mach. Intell.**, 32:1 (2010) 105–119.
- [27] S. Leutenegger, M. Chli, and R. Y. Siegwart, “BRISK : Binary Robust Invariant Scalable Keypoints,” IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Washington DC, (2011), pp. 2548–2555.
- [28] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, “BRIEF: Binary Robust

- Independent Elementary Features*,” European Conference on Computer Vision, Crete, (2010), pp. 778–792.
- [29] E. Rublee and G. Bradski, “*ORB - an efficient alternative to SIFT or SURF*,” IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Washington DC, (2011), pp. 2564–2571.
- [30] J. J. Koenderink, “*The structure of images*,” **Biol. Cybern.**, 50:5 (1984) 363–370.
- [31] T. Lindeberg, “*Scale-space theory: a basic tool for analyzing structures at different scales*,” **J. Appl. Stat.**, 21:1 (1994) 225–270.
- [32] M. Graves and B. Batchelor, *Machine Vision for the Inspection of Natural Products*. Springer Verlag, London, 2004.
- [33] M. Brown and D. G. Lowe, “*Invariant features from interest point groups*,” British Machine Vision Conference, Cardiff, Wales, (2002), pp. 656–665.
- [34] E. Öztemel, *Yapay Sinir Ağları*, Papatya Yayıncılık, İstanbul, 2003.
- [35] L. Fausett, *Fundamentals of Neural Networks*. Prentice Hall, USA, 2014.
- [36] J. A. Andersen, *Introduction to Neural Networks. Handbook of Brain Theory and Neural Networks*. MIT Press, Cambridge, MA, 1995.
- [37] Z. Şen, *Yapay Sinir Ağları İlkeleri*. Su Vakfı Yayınları, İstanbul, 2004.
- [38] L. R. Dice, “*Measures of the amount of ecologic association between species*,” **Ecology**, 26 (1945) 297–302.
- [39] A. Hore and D. Ziou, “*Image Quality Metrics: PSNR vs. SSIM*,” 20th International Conference on Pattern Recognition, Istanbul, (2010), pp. 2366–2369.
- [40] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “*Image quality assessment: from error visibility to structural similarity*,” **IEEE Transactions on Image Processing**, 13:4 (2004) 600–612.

ÖZGEÇMİŞ

Ad Soyad	Fatma Elzahar
Doğum Yeri ve Tarihi	Malatya 1979
Adres:	Malatya Su ve Kanalizasyon İdaresi Genel Müdürlüğü
E-Posta	fatmaelzahar@maski.gov.tr
Lisans	Fırat Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği 2002

Mesleki Deneyim

- Milli Eğitim Bakanlığı Şanlıurfa İl Milli Eğitim Müdürlüğü (2002-2009)
- İçişleri Bakanlığı Malatya İl Özel İdaresi (2009-2014)
- Malatya Büyükşehir Belediyesi Su ve Kanalizasyon İdaresi Genel Müdürlüğü (2009 – Devam Ediyor)

Yayın Listesi

1. Fatma Er (Elzahar), Muhammed Fatih Talu, Mehmet Gül, Performance Analysis of Segmentation Methods for Detecting and Qualifying of Mitochondrial Damage, International Symposium on Biomedical Science and Technology Symposium, 09 -11 Ağustos 2011, Ankara
2. Fatma Er (Elzahar), Suna İpek, Muhammed Fatih Talu, Bilal Alataş, Ali Karcı, Izgara Arabağlama Ağlarında Yönlendirme Algoritmalarının Uygulaması, 1.Ulusal Bilişim-Multimedya Konferansı, 16-20 Nisan 2001, Elazığ
3. Suna İpek, Fatma Er (Elzahar), Muhammed Fatih Talu, Bilal Alataş, Ali Karcı, Hiperküp Arabağlama Ağlarında Yönlendirme Algoritmalarının Uygulaması, 1.Ulusal Bilişim-Multimedya Konferansı, 16-20 Nisan 2001, Elazığ
4. Bilal Alataş, Muhammed Fatih Talu, Suna İpek, Fatma Er (Elzahar), Ali Karcı, Çizge Bölmeleme Probleminin Genetik Algoritma Çözümünde Yeni Bir Yaklaşım, 1.Ulusal Bilişim-Multimedya Konferansı, 16-20 Nisan 2001, Elazığ