



**YEREL İKİLİ ÖRÜNTÜ TABANLI UÇ ÖĞRENME YAKLAŞIMI KULLANAN AKILLI
ÖRÜNTÜ TANIMA SİSTEMİ İLE MERMER SINIFLANDIRMA UYGULAMASI**

Erhan TURAN

**Yüksek Lisans Tezi
Mekatronik Mühendisliği Anabilim Dalı
Danışman: Prof. Dr. Beşir DANDIL**

TEMMUZ-2018

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

**YEREL İKİLİ ÖRÜNTÜ TABANLI UÇ ÖĞRENME YAKLAŞIMI KULLANAN AKILLI
ÖRÜNTÜ TANIMA SİSTEMİ İLE MERMER SINIFLANDIRMA UYGULAMASI**

YÜKSEK LİSANS TEZİ

Erhan TURAN

(142134102)

Anabilim Dalı: Mekatronik Mühendisliği

Programı: Elektronik

Danışman: Prof. Dr. Beşir DANDIL

Tezin Enstitüye Verildiği Tarih: 13/06/2018

TEMMUZ-2018

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

**YEREL İKİLİ ÖRÜNTÜ TABANLI UÇ ÖĞRENME YAKLAŞIMI KULLANAN AKILLI
ÖRÜNTÜ TANIMA SİSTEMİ İLE MERMER SINIFLANDIRMA UYGULAMASI**

YÜKSEK LİSANS TEZİ

Erhan TURAN

(142134102)

**Tezin Enstitüye Verildiği Tarih : 13/06/2018
Tezin Savunulduğu Tarih : 06/07/2018**

Tez Danışmanı : Prof. Dr. Beşir DANDIL (F.Ü)

Diğer Jüri Üyeleri : Doç. Dr. Murat KARABATAK (F.Ü)

Dr. Öğr. Üyesi Ömer Faruk ALÇİN (B.Ü)

TEMMUZ-2018

ÖNSÖZ

Önerilen bu çalışmada değerli vaktini, tecrübesini ve bilgisini benden esirgemeyen değerli hocam Prof. Dr. Beşir DANDIL'a, tez sürecinde değerli görüşlerini paylaşan sayın Arş. Gör. Ferhat UÇAR'a ve seminer sürecinde desteğini esirgemeyen değerli hocam Doç. Dr. Koray Şener PARLAK'a saygı ve teşekkürlerimi sunarım. Ayrıca tez çalışmam süresince desteğini eksik etmeyen değerli arkadaşlarım Orhan YAMAN ve Mehmet Hüseyin YILDIRIM'a teşekkür ederim.

Hayatımın her anında desteğini üzerimden esirgemeyen babama, anneme ve kardeşlerime teşekkür ederim.

Erhan TURAN
ELAZIĞ - 2018

İÇİNDEKİLER

	<u>Sayfa No</u>
ÖNSÖZ	II
İÇİNDEKİLER.....	III
ÖZET	VI
SUMMARY	VII
ŞEKİLLER LİSTESİ	VIII
TABLolar LİSTESİ	X
SİMGELER LİSTESİ.....	XI
KISALTMALAR LİSTESİ	XIII
1. GİRİŞ	1
1.1. Mermer Sınıflandırma İşlemi	1
1.2. Literatür Taraması	3
1.3. Önerilen Çalışma	10
1.4. Amaç	14
2. ÖZELLİK ÇIKARIMI	15
2.1. LBP'den Elde Edilen Özellikler	16
2.1.1. LBP Uygulama Adımları.....	16
2.2. Çoklu Özellik Çıkarımı	20
2.2.1. Histogramdan Elde Edilen Özellikler.....	20
2.2.1.1. Birinci Derece Histogramdan Elde Edilen Özellikler	21
2.2.1.2. İkinci Derece Histogramdan Elde Edilen Özellikler	22
2.2.2. SIFT Algoritması.....	24
2.2.2.1. SIFT Uygulama Adımları	25
2.2.2.2. SIFT Tabanlı Görsel Kelime Çantasının Oluşturulması	31
2.3. Bölüm Değerlendirmesi	32

3. SINIFLANDIRMA İŞLEMİ.....	33
3.1. Uç Öğrenme Makinesi.....	33
3.1.1. ELM Modeli	33
3.1.2. ELM Sınıflandırıcısı	34
3.1.3. Parametrelerin Belirlenmesi	36
3.2. Toplu Sınıflandırıcı	37
3.2.1. Karar Ağacı Yöntemiyle Sınıflandırma.....	37
3.2.1.1. Karar Ağacı Algoritmaları ve Seçimi	37
3.2.1.2. C4.5 Tabanlı Karar Ağacı Oluşturma	40
3.2.2. Yapay Sinir Ağıyla Sınıflandırma	42
3.2.2.1. Yapay Sinir Ağı Modeli	43
3.2.2.2. Yapay Sinir Hücresinin Modeli	44
3.2.2.3. YSA Katman Mimarisi	47
3.2.2.4. Yapay Sinir Ağının Eğitilmesi	48
3.2.2.5. Parametrelerin Belirlenmesi	50
3.2.3. Destek Vektör Makinesiyle Sınıflandırma	52
3.2.3.1. DVM Modeli	53
3.2.3.2. Doğrusal Üstündüzlem Sınıflandırıcı	54
3.2.3.3. Doğrusal Olmayan Çekirdek Sınıflandırıcı	57
3.2.3.4. Çekirdek Fonksiyonları	57
3.2.3.5. Parametrelerin Belirlenmesi	58
3.3. Bölüm Değerlendirmesi	45
4. DENEYSEL SONUÇLAR VE GUI'NIN GELİŞTİRİLMESİ.....	60
4.1. Deneysel Sonuçlar	60
4.1.1. Sınıflandırma Performansının İncelenmesi	60
4.1.2. ELM Deneysel Sonuçlar.....	61
4.1.2.1. Aktivasyon Fonksiyonu ve Hücre Sayısına göre Başarım Performansının	

Karşılaştırılması.....	62
4.1.2.2. ROC Analizi	63
4.1.3. Karar Ağacı Deneysel Sonuçlar	64
4.1.3.1. Sınıflandırma Başarılarının Karşılaştırılması	65
4.1.3.2. ROC Analizi	67
4.1.4. YSA Deneysel Sonuçlar	68
4.1.4.1. Sınıflandırma Başarılarının Karşılaştırılması	70
4.1.4.2. Çevrim Sayısına göre Başarım Oranının Karşılaştırılması	72
4.1.4.3. ROC Analizi	73
4.1.5. DVM Deneysel Sonuçlar.....	68
4.1.5.1. Karmaşıklık Değerine göre Başarım Oranının Karşılaştırılması.....	75
4.1.5.2. ROC Analizi	77
4.1.6. Sınıflandırma Başarımlarının Karşılaştırılması	78
4.2. Grafiksel Arayüzün Geliştirilmesi	78
4.2.1. Özellik Çıkarımında Kullanılan Teknolojiler.....	79
4.2.2. Grafiksel Arayüzde Kullanılan Teknolojiler	80
4.2.3. GUI Çalışma Mimarisi	80
4.3. Bölüm Değerlendirmesi	89
8. SONUÇLAR	90
KAYNAKLAR	91
ÖZGEÇMİŞ	96

ÖZET

Mermerleri desen kalitesine göre ayırma işlemi uzman kararına göre yapılan bir işlemdir. Mermerin sınıflandırma safhası ekonomik değerinin en kritik bölümdür. Bu çalışmada, mermerlerin sınıflandırmasını hızlı ve yüksek başarımla gerçekleştiren ve bunu yaparken öğrenen bir sistem önerilmiştir. Bu sistem kameradan veya hafızadan elde ettiği görüntülere özellik çıkarım işlemi uygulamaktadır.

Mermer görüntülerini sınıflandırmak için Yerel ikili örüntüleri (Local Binary Pattern-LBP) öznitelik tabanlı Uç öğrenme makinesi (Extreme Learning Machine-ELM) kullanılmıştır. Görüntü kümesine LBP özellik çıkarım işlemi uygulanarak eğitim kümesi elde edilir. ELM sınıflandırıcısı LBP tabanlı eğitim kümesini kullanarak öğrenme işlemini gerçekleştirir. ELM sınıflandırıcısının sınıflandırma başarısını ölçmek için Toplu (Ensemble) sınıflandırıcı kullanılmıştır. C4.5 tabanlı karar ağacı, ileri beslemeli statik danışmanlı yapay sinir ağı ve destek vektör makinesi sınıflandırıcıları Toplu sınıflandırıcıyı oluşturmaktadır. ELM'nin sınıflandırma başarımı Toplu sınıflandırıcıyla karşılaştırılmıştır.

LBP tabanlı eğitim kümesinin başarımla performansını karşılaştırmak için Çoklu özellik çıkarımından yararlanılmıştır. Çoklu özellik çıkarımında Histogram ve Ölçekten Bağımsız Öznitelik Dönüşümü (Scale Invariant Feature Transform-SIFT) tabanlı öznitelikler mevcuttur. SIFT tabanlı özelliklerin eğitim kümesine dönüştürülmesi için Kelime çantası (Bag of Words-BOW) modelinden yararlanılmıştır. Kelime çantasında görsel kelimeleri ayırmak için k-ortalama kümeleme yöntemi kullanılmıştır.

Hazırlanan modellerin özellik çıkarım ve sınıflandırma işlemleri geliştirilen Grafiksel arayüz (Graphical User Interface-GUI) aracılığıyla yapılmaktadır. Kullanıcı uygulama üzerinden sistemi eğiterek, sınıfı bilinmeyen mermerin sınıfını belirleyebilir. Çalışma kapsamında geliştirilen GUI endüstriyel kullanıma uygundur.

Anahtar Kelimeler: Mermer Sınıflandırma, Uç Öğrenme Makinesi, Karar Ağacı, Yapay Sinir Ağı, Destek Vektör Makinesi, Yerel İkili Örüntü, Histogram Özellik Çıkarımı, SIFT, BOW, K-Ortalama

SUMMARY

Local Binary Pattern Based Marble Classification Application Using Intelligent Pattern Recognition System Using the Extreme Learning Approach

Separation of the marbles according to the pattern quality is an operation made according to expert decision. The marble classification stage is the most critical part of the economic value. In this study, a system was proposed which performs the classification of marbles quickly and with high performance and learns in doing so. This system applies feature subtraction to the images obtained from the camera or memory.

The Local Binary Pattern (LBP) feature-based Extreme Learning Machine (ELM) was used to classify marble views. The training set is obtained by applying LBP feature extraction to the image set. The ELM classifier performs learning using the LBP-based training set. The Ensemble classifier was used to measure the classification success of the ELM classifier. C4.5 based decision tree, forward feed static advisory artificial neural network and support vector machine classifiers constitute collective classifier. The classification performance of the ELM is compared with the collective classifier.

Multiple feature extraction was used to compare the performance performance of the LBP-based training set. Histogram and Scale Invariant Feature Transform (SIFT) based attributes are available for multiple feature extraction. The Bag-of-Words (BOW) model was used to transform SIFT-based features into a training set. The k-means clustering method is used to distinguish visual words in the vocabulary.

Graphical interface (Graphical User Interface-GUI) is used to develop the feature extraction and classification processes of the prepared models. By training the system through the application, the user can determine the class of marble that does not know the class. The GUI developed within the scope of the study is suitable for industrial use.

Keywords: Marble Classification, Extreme Learning Machine, Decision Tree, Artificial Neural Network, Support Vector Machine, Local Binary Pattern, Histogram Feature Extraction, SIFT, BOW, K-Means

ŞEKİLLER LİSTESİ

Sayfa No

Şekil 1.1 Genel mermer sınıflandırma sistemi	2
Şekil 1.2 Önerilen mermer sınıflandırma sistemi akış şeması	13
Şekil 2.1 Özellik çıkarım işlemi	15
Şekil 2.2 LBP detektörü akış şeması	17
Şekil 2.3 LBP detektörü doku belirleme işlemi	17
Şekil 2.4 P bitlik LBP noktasının hesaplanması	18
Şekil 2.5 A sınıfına ait örnek bir mermerin LBP histogramı	19
Şekil 2.6 Histogram oluşturma işlemi	23
Şekil 2.7 SIFT detektörü akış şeması	25
Şekil 2.8 Octave gruplarının yumuşatma işlemi	26
Şekil 2.9 DoG işlemi	27
Şekil 2.10 Anahtar nokta tespiti	28
Şekil 2.11 Yerel imge tanımlayıcısı akış şeması	30
Şekil 3.1 Tek katmanlı ileri beslemeli YSA yapısı (ELM Modeli)	34
Şekil 3.2 Karar ağacı oluşturma işlemi ön aşamaları	40
Şekil 3.3 LBP eğitilmiş C4.5 tabanlı karar ağacı	41
Şekil 3.4 Histogram eğitilmiş C4.5 tabanlı karar ağacı	41
Şekil 3.5 SIFT eğitilmiş C4.5 tabanlı karar ağacı	41
Şekil 3.6 Biyolojik sinir hücresi	44
Şekil 3.7 YSA hücre modeli	44
Şekil 3.8 Tek gizli katmanlı YSA yapısı	44
Şekil 3.9 DVM sınıflandırma düzlemi	53
Şekil 4.1 ELM eğitim parametreleri	61
Şekil 4.2 Hücre sayısına göre başarı oranının karşılaştırması	62

Şekil 4.3 ELM ROC alanlarının karşılaştırılması	64
---	----

Sayfa No

Şekil 4.4 Karar ağacı eğitim parametreleri	65
Şekil 4.5 Güven değerine göre eğitim başarısının karşılaştırılması	66
Şekil 4.6 Güven değerine göre test başarısının karşılaştırılması	66
Şekil 4.7 Karar ağacı ROC alanlarının karşılaştırılması	68
Şekil 4.8 YSA eğitim parametreleri	69
Şekil 4.9 LBP tabanlı YSA	69
Şekil 4.10 LBP tabanlı YSA'nın çapraz doğrulamalı başarı grafiği	70
Şekil 4.11 Histogram tabanlı YSA'nın çapraz doğrulamalı başarı grafiği	71
Şekil 4.12 SIFT tabanlı YSA'nın çapraz doğrulamalı başarı grafiği	72
Şekil 4.13 Çevrim sayısına göre başarı oranının karşılaştırması	72
Şekil 4.14 YSA ROC alanlarının karşılaştırılması	74
Şekil 4.15 DVM eğitim parametreleri	75
Şekil 4.16 LBP tabanlı sınıflandırma başarısının karşılaştırması	75
Şekil 4.17 Histogram tabanlı sınıflandırma başarısının karşılaştırması	76
Şekil 4.18 SIFT tabanlı sınıflandırma başarısının karşılaştırması	76
Şekil 4.19 DVM ROC alanlarının karşılaştırılması	78
Şekil 4.20 GUI akış şeması	81
Şekil 4.21 GUI anasayfa	82
Şekil 4.22 CSV tabanlı özellik çıkarım menüsü	83
Şekil 4.23 CSV dosyasının arff dosyasına dönüştürülmesi	84
Şekil 4.24 LBP tabanlı özellik çıkarımı	85
Şekil 4.25 Histogram tabanlı özellik çıkarımı	85
Şekil 4.26 SIFT tabanlı özellik çıkarımı	86
Şekil 4.27 Sınıflandırma ayar menüsü	86
Şekil 4.28 Sonuç sayfası	87
Şekil 4.29 Sonuç karar algoritması	88

TABLÖLAR LİSTESİ

Sayfa No

Tablo 2.1. LBP histogramından elde edilen özellikler.....	19
Tablo 2.2. LBP tabanlı eğitim kümesi oluşturma işlemi sözde kodu	20
Tablo 2.3. Histogramdan elde edilen özellikler	23
Tablo 2.4. Histogram özellik çıkarımı sözde kodu	24
Tablo 2.5. SIFT tabanlı eğitim kümesi sözde kodu	31
Tablo 3.1. ELM katsayı atama sözde kodu	36
Tablo 3.2. Karar ağacı oluşturma işlemi sözde kodu	42
Tablo 3.3. YSA'da kullanılan en yaygın toplama fonksiyonları	45
Tablo 3.4. YSA da kullanılan en yaygın aktivasyon fonksiyonları	46
Tablo 3.5. YSA katsayı atama sözde kodu	51
Tablo 3.6. DVM çekirdekleri	57
Tablo 3.7. DVM katsayı atama sözde kodu	58
Tablo 4.1. Karmaşıklık matrisi	61
Tablo 4.2. ELM 10-kat çapraz doğrulama	63
Tablo 4.3. ELM tabanlı karmaşıklık matrisi	63
Tablo 4.4. Karar Ağacı 10-kat çapraz doğrulama	67
Tablo 4.5. Karar Ağacı karmaşıklık matrisi.....	67
Tablo 4.6. LBP tabanlı YSA'nın çapraz doğrulamalı başarı oranı (Fold=10).....	70
Tablo 4.7. Histogram tabanlı YSA'nın çapraz doğrulamalı başarı oranı (Fold=10)	71
Tablo 4.8. SIFT tabanlı YSA'nın çapraz doğrulamalı başarı oranı (Fold=10)	71
Tablo 4.9. YSA10-kat çapraz doğrulama	73
Tablo 4.10. YSA tabanlı karmaşıklık matrisi	73
Tablo 4.11. DVM 10-kat çapraz doğrulama	77
Tablo 4.12. DVM tabanlı karmaşıklık matrisi	77
Tablo 4.13. Sınıflandırma başarımlarının karşılaştırılması	78

SİMGELERİN LİSTESİ

P	: LBP nokta sayısı
R	: LBP yarıçap
A	: Görüntünün x eksenindeki piksel sayısı
B	: Görüntünün y eksenindeki piksel sayısı
G	: Gri renk seviyesi
P_i	: Gri renk seviyesi yoğunluğu
μ	: Ortalama
σ^2	: Varyans
μ^3	: Çarpıklık
μ^4	: Basıklık
E	: Enerji
H	: Entropi
G	: Gauss fonksiyonu
I	: İşlenmemiş görüntü
L	: Yumuşatılmış görüntü
D	: Octave gruplarının farkı
$\hat{x}$	: SIFT uç noktaların konumu
H	: Hessian Matrisi
α	: En büyük gradyan değerine sahip öz değer
β	: En küçük gradyan değerine sahip öz değer
m	: Gradyan büyüklüğü
θ	: Gradyan açısı
w	: Giriş katmanı ile ara katman arasındaki ağırlık değeri
β	: Ara katman ile çıkış katmanı arasındaki ağırlık değeri
x_i	: ELM giriş değeri
y_i	: ELM çıkış değeri
$g(x)$	: Aktivasyon fonksiyonu
y_i	: ELM çıkış değeri
O_j	: ELM istenen çıkış

$H^{\dagger}$	: Moore-Penrose matrisi
C_i	: Karar ağacındaki örneklerin sınıfı
T	: C_i lerden oluşan sınıf kümesi
P_t	: Herhangi bir sınıfa ait örneklerin bütün örnekler arasında bulunma olasılığı
$H(T)$	: Entropi
$H(X_k)$	: Eğitim kümesinin sınıflara göre genel entropisi
$H(X, T)$	: Özelliklerin sınıfa göre ayrı ayrı entropileri
(X, T)	: Kazanç
V_i	: Sıralanmış özelliklerin orta sıradaki değeri
t_i	: Eşik değeri
w_i	: YSA ağırlık değeri
E_m	: YSA çıktı hatası
y_m	: YSA çıkış değeri
λ	: Öğrenme katsayısı
α	: Momentum katsayısı
m	: En yakın örnekler arası mesafe
d	: Sınıflandırma düzlemi ile en yakın örnekler arası mesafe
X	: Eğitim kümesi
x^t	: A sınıfına ait örnekler
r^t	: B sınıfına ait örnekler
W_n	: Ağırlık vektörleri
p	: Eşik değeri
L_p	: İkinci dereceden iyileme işlemi
$g(x)$	: Test fonksiyonu

KISALTMALAR LİSTESİ

YSA	: Yapay Sinir Ağı
PCA	: Temel Bileşen Analizi (Principal Component Analysis)
DVM	: Destek Vektör Makinesi (Support Vector Machine)
CART	: Sınıflandırma ve Regresyon Ağacı (Classification and Regression Tree)
LBP	: Yerel İkili Desen (Local Binary Pattern)
KNN	: K En Yakın Komşu (K Nearest Neighbour)
SIFT	: Ölçekten Bağımsız Öznitelik Dönüşümü (Scale Invariant Feature Transform)
SURF	: Hızlandırılmış Sağlam Öznitelikler (Speeded Up Robust Features)
BOF	: Görsel Kelime Çantası (Bag of Words)
ORB	: Odaklanmış Hızlı ve Döndürülmüş Öz (Oriented FAST and Rotated BRIEF)
ELM	: Uç Öğrenme Makinesi (Extreme Learning Machine)
DQLK	: Derin Quasi-Lineer Çekirdeği
HRBFN	: Hierarchical Radial Basis Function Network (Hiyerarşik radyal temel fonksiyon ağı)
PLC	: Programmable Logic Unit (Programlanabilir Mantıksal Denetleyici)
RGB	: Red-Green-Blue
SDH	: Toplam-Fark Histogramı (Sum-Difference Histogram)
GUI	: Graphical User Interface (Grafiksel Kullanıcı Arayüzü)



1. GİRİŞ

Günümüzde başta inşaat sektörü olmak üzere birçok farklı sektörde kullanılan mermer, kalker ve dolomitik bileşenlerin uzun yıllar içerisinde bir araya gelerek oluşturduğu doğal ve eşsiz bir taştır. Bileşiminin yaklaşık %90-98 kısmını CaCO_3 (Kalsiyum Karbonat) oluşturur; geriye kalan kısmı da değişken oranda MgCO_3 (Magnezyum Karbonat) içerir. Ayrıca düşük miktarda da olsa değişken oranlarda feldspat, silis, florin, demiroksit, silika gibi organik bileşenlerden oluşur. Mermeri traverten, granit gibi doğal taşlardan üstün kılan en önemli özellik, değişken miktarlarda barındırdığı organik ve inorganik bileşenler sayesinde eşsiz dokulara sahip olmasıdır [1]. Mermerin uygulandığı alanda, aynı dokuya sahip plakaların kullanılması tercih edilmektedir. Aynı dokuya sahip mermerleri tespit etmek uzmanlık gerektiren insan kaynaklı bir işlemdir. Mermer, sayısız tür ve eşsiz dokuya sahip olduğundan sınıflandırma ölçütleri değişkenlik gösterir. Sayısız çeşit ve dokulara sahip mermer plakalarını sınıflandırmak oldukça zor ve uzmanlık gerektiren bir işlemdir.

Mermer, uzun ve zorlu işlemler sonucunda ürün haline dönüşmektedir. Mermer bloklarının ürüne dönüşme sürecindeki son aşaması dokularına göre sınıflandırma işlemidir. Sınıflandırma işlemi, mermerin ekonomik değerinin belirlendiği aşamadır. Sanayide mermer sınıfını belirleme işlemi uzman personel sayesinde gerçekleştirilir. Dolayısıyla personel kaynaklı hatalar, işletmenin maddi ve ticari itibar kaybına neden olabilir. Bu çalışmada mermer dokularını sayısal özelliklere dönüştürerek sınıflandırabilen bir sistem geliştirilmesi amaçlanmıştır.

1.1. Mermer Sınıflandırma İşlemi

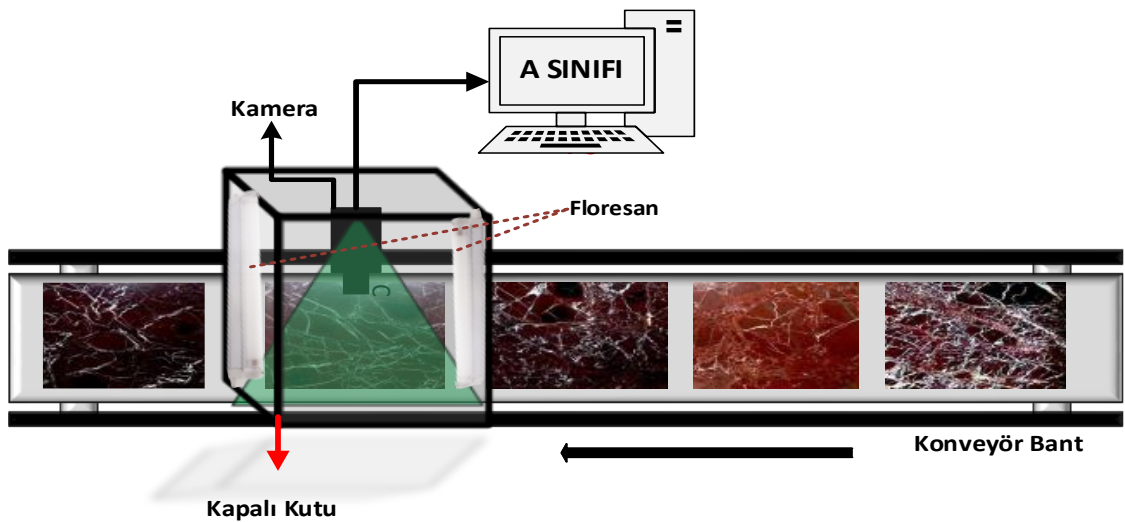
Mermer sınıflandırma işlemi aynı tür içinde ve farklı türler arasında yapılmaktadır. Farklı türler arasındaki sınıflandırma işlemi, örneğin ‘Hazar bej’, ‘Elazığ vişne’, ‘Traverten’ gibi türlerin aynı zaman ve ortamda sınıflandırılmasıdır. Aynı tür içindeki sınıflandırma işlemi ise renk, homojenlik, damar yapısına göre değişkenlik gösteren dokuları ayırma işlemidir. Farklı türdeki mermerlerin renk ve dokuları belirgin şekilde ayrılabilirdiğinden bu alandaki sınıflandırma işlemi daha yüksek başarımla sağlanır. Aynı tür içindeki sınıflandırma işleminde renk, homojenlik ve damar yapısı benzerlik gösterdiğinden sınıflandırılması daha zordur.

Mermerlerin dokularına göre sınıflandırma işlemi uzman personelin öznel kararına göre yapılan bir işlemdir. Uzman personel işletme içinde belirlenen ölçütlere göre mermeri renk, homojenlik ve dokusuna göre sınıflandırmaktadır. Personel kaynaklı yanlış sınıflandırma işleminde firma ekonomik ve ticari itibar kaybına uğrayabilir. Mermerin yüksek ekonomik değeri göz önüne alındığında nesnel bir yapı kullanılarak sınıflandırılması gerekmektedir.

Üreticiler mermer dokularını farklı ölçütlere göre ayırmaktadır. Büyük boyuttaki mermer plakalarında sınıflandırma için kullanılan fosil oranının eşik değeri esnetilebilmektedir [2]. Genelleştirilmiş bir sistemde, doku çeşidi ve renk yoğunluklarına göre sınırsız olan mermer çeşitleri için ayrı ayrı sınıflandırma özellikleri belirlemek mümkün değildir [2].

Dünyadaki mermer türleri arasında renk ve damar yapısı özellikleri ile “Elazığ Vişne” küresel adıyla “Rosso Levanto” mermeri önemli bir yere sahiptir. “Elazığ Vişne” mermeri kendi içerisinde 12 farklı dokuya ayrılmaktadır. Dünyada sadece Elazığ’da çıkarıldığından yüksek ekonomik değere sahiptir [2]. Yüksek sayıda doku sınıfı içeren bu mermerin sınıflandırma süreci, düşük sayıda doku türüne sahip olan mermerlere göre daha maliyetli ve karışıktır.

Mermer sınıflandırma işleminde, sistemin genel işleyiş yapısı Şekil 1.1’de verildiği gibidir. Şekil 1.1 ile verilen genel mermer sınıflandırma birimi, aynı zamanda bu tez çalışmasında kullanılan sınıflandırma işlem sürecini de ifade etmektedir.



Şekil 1.1. Genel mermer sınıflandırma sistemi

1.2. Literatür Taraması

Mermer dokularını otonom bir şekilde sınıflandırmak için çeşitli çalışmalar yapılmıştır. Bu çalışmada Yerel İkili Örüntü (Local Binary Pattern-LBP) özellik çıkarım tabanlı Uç Öğrenme makinesi (Extreme Learning Machine-ELM) kullanılarak mermer doku sınıflandırması amaçlanmıştır. LBP ve ELM'nin sınıflandırma başarımını karşılaştırmak için özellik çıkarımında histogram ve Ölçekten Bağımsız Özellik Dönüşümü (Scale Invariant Feature Transform -SIFT) yöntemleri, sınıflandırıcı da ise karar ağacı, Yapay Sinir Ağı (Artificial Neural Network-YSA) ve Destek Vektör Makinesi (Support Vector machine-DVM) yöntemleri kullanılmıştır. Önerilen yöntemler özellik çıkarımı ve sınıflandırma işlemi olmak üzere iki aşamada gerçekleştirilir. Literatür taramasında bu iki aşamada kullanılmış yöntemlere ait genel çalışmalar sıralanmıştır.

Şekil 1.1'de verilen genel sınıflandırma sistemindeki kamera, konveyör üzerinde bulunan kapalı kutu içerisine yerleştirilmiştir. Bu şekildeki genel yapı, deneysel test ortamında [3] ve gerçek fabrika ortamında [4] denenmiştir.

Denetimsiz eğitim setinin, kümelenerek sınıflandırılmasını sağlayan bir çalışma önerilmiştir [3]. Bu çalışmada mermer görüntülerinin özellikleri çıkarılarak, k-Ortalama ve C-Ortalama kümeleme algoritmalarıyla eğitim setinin sınıflarına gerek duyulmadan, kümeleme yapılması sağlanmıştır. Denetimli eğitim setinin kullanılmamasının nedeni, sayısız mermer grubunun eğitim verisini elde etmenin zor ve uzun süreçler gerektirmesidir. Bu çalışmada ayrıca Şekil 1.1'de verilen genel mermer sınıflandırma sistemine ek olarak, sınıflandırılan mermeri ilgili alana yerleştiren PLC kontrollü pnömatik ayırıcı geliştirilmiştir.

Genel mermer sınıflandırma sisteminin, bir mermer üretim tesisinde kullanıldığı bir çalışma gerçekleştirilmiştir [4]. RGB, XYZ, YIQ, K-L renk uzaylarında, özellik çıkarım işlemi yapılarak YSA ile sınıflandırma işlemi gerçekleştirilmiştir. YSA'nın öğrenme işleminde denetimli eğitim seti kullanılmıştır. Yeni mermer grubunun sınıflandırılabilmesi için eğitim setinin sisteme öğretilmesi gerekmektedir. Bu çalışmada kullanılan eğitim seti, çalışma haricinde elde edilerek sisteme tanıtılmıştır. Test kümesi, sınıflandırma algoritması üzerinden test edilerek gerçek sınıfları ile karşılaştırılmış ve bu karşılaştırma sonrasında sınıflandırma algoritmasının başarı oranı belirlenmiştir. Böylece, sınıflandırma işlemlerini farklı renk uzaylarında gerçekleştirerek, başarı oranlarını karşılaştırmıştır.

Ardışıl özellik çıkarımı ve sınıflandırma işlemlerinin birlikte kullanıldığı [5] bir çalışma yapılmıştır. Hiyerarşik radyal temel fonksiyon ağı (Hierarchical Radial Basis Function

Network-HRBFN) sınıflandırıcısı kullanılarak her adımda veri seti sınıflandırılır. Sınıflandırılmayan küme için ayrı bir özellik çıkarım işlemi uygulanmıştır. Sırasıyla Toplam-Fark Histogramı (Sum-Difference Histogram-SDH), Dalgacık ve Morfolojik özellikleri elde edilerek sınıflandırma işlemi gerçekleştirilmiştir. Ancak bu çalışmadaki, eğitim kümesinde bulunan mermer grupları, renk ve desen olarak gözle birbirinden kolayca ayrılabilen türlerdir.

Nesneleri analiz etmek Temel Bileşenler Analizi (Principal Component Analysis-PCA) ile Karhunen-Loève dönüşümü kullanılmıştır [6]. Değişken sayısının fazla olduğu veri kümelerinin analizinde kullanılan ve doğrusal bir algoritma kullanan PCA yöntemiyle; veriler sadeleştirme, azaltma, modelleme, aykırılık tespiti, değişken seçme, sınıflandırma ve tahmin işlemlerinden geçirilir. PCA işleminden sonra, veriler analiz edilebilecek sadelikte ve işlem maliyeti açısından verimli hale dönüştürülmüştür.

Toplam-Fark ve Eşdizilimsel (Co-Occurence) histogramından özellik çıkarımını sağlayan bir çalışma yapılmıştır [7]. Bu çalışmada görüntü dokusunu analiz edebilmek için ‘ortalama’, ‘varyans’, ‘enerji’, ‘korelasyon’, ‘belirsizlik’, ‘zıtlık’, ‘homojenlik’, ‘gölge’, ‘çıkıntı’ özelliklerinin hesaplaması verilmiştir. Bu özellik çıkarım yöntemlerinin performans karşılaştırması yapılmıştır.

Mermerin ocaktan ürün haline gelene kadar kalite kontrol süreçleri incelenmiştir [8]. Mermelerin desen kalitesiyle ilgili en önemli faktörün renk homojenliği olduğu belirtilmiştir. Homojenlik ve damar yapısının karakteri incelenecek en önemli faktörlerdir. Mermer beyaz ise sarı ve gri damar istenmeyen bölgelerdir. Renkli mermerde ise renklerin soluk, heterojen ve belirgin olmayan kısımları istenmeyen bölgeler olarak kabul edilir. Bu şekilde bir desen kalite kontrol yöntemi önerilmiştir. Ayrıca mermerlerin fiziksel ve kimyasal bileşenleri açıklanmıştır.

DVM sınıflandırma algoritması, mermerlerin doku sınıflandırmasında kullanıldığı bir çalışma yapılmıştır [9]. Mermerlerin özellikleri SDH yöntemiyle elde edilmiştir. DVM’de sınıfları ayırmak için polinomal çekirdek kullanılmıştır. Polinomal çekirdek, sınıfları birbirinden ayırmayı sağlayan polinomsal bir fonksiyondur. DVM’de, kolayca ayrılabilen sınıflar için doğrusal fonksiyon daha karmaşık sınıflarda Gauss veya Sigmoid çekirdekler kullanılmıştır.

Görüntü histogramı kullanılarak özellik çıkarım işlemi gerçekleştirilmiştir [10]. Bu özellikler PCA yöntemi kullanılarak daha verimli ve sadeleştirilmiş özellikler elde edilmiştir. PCA’dan elde edilen özellikler ile saf özelliklerin sınıflandırma başarısı karşılaştırılmıştır.

Karar ağacı sınıflandırma yöntemi olan, belirsizlik (entropi) tabanlı “ID3” adlı algoritması önerilmiştir [11]. ID3 algoritması karar ağaçlarında kategorik verilerde kullanılan en yaygın

algoritmadır. Bu algoritma, özelliklere ait olasılıklardan entropiyi hesaplayarak düğümün hangi özellikten bölüneceğini belirler. Her düğüm sadece bir sınıfa ait örnekleri içermelidir. Eğer düğüm farklı sınıftaki örnekleri içeriyorsa düğüm saf değildir ve bölünebilir.

Sayısal verilere cevap veremeyen ID3 algoritmasının, gelişmiş sürümü olan C4.5 algoritması geliştirilmiştir [12]. C4.5 algoritmasında, sayısal veriler sıralanarak ortadaki iki değer ortalamasına göre verileri 'büyük', 'küçük', 'büyük eşit', 'küçük eşit' şeklinde sınıflara ayrılmaktadır. Bu şekilde kategorilere ayrılan veriler ID3 algoritması üzerinden işlem görmektedir. ID3 algoritmasında özelliklere ait entropiler hesaplanarak düğümü oluşturan özellikler tespit edilir.

Sınıflandırma ve Regresyon (Classification and Regression Tree-CART) tabanlı karar ağacı oluşturma yöntemi açıklanmıştır [13]. CART tabanlı karar ağacında, kategorik eğitim kümesi kullanılmıştır.

Çoklu özellik çıkarım işlemi eşzamanlı uygulanarak, genişletilmiş çok yapılı yerel LBP ile yüksek çözünürlüklü görüntülerin sınıflandırılması gerçekleştirilmiştir [14].

Dantel görüntülerini sınıflandırmak için Otsu modeli ile arka plan devre dışı bırakıldıktan sonra LBP özellik çıkarımı uygulanmıştır [15]. Dantel dokusu sanayideki birçok alanda kullanılabilir. Esnek olması ve kolayca deforme olması, dantel gibi değişkenlik gösteren dokuların sınıflandırılmasını zorlaştırır. Bu çalışmada, dairesel LBP'nin yarıçap ve nokta sayısı değiştirilerek, dantel dokularının sınıflandırma başarısına etkisi incelenmiştir.

LBP tabanlı özellikler kullanılarak nesne tanıma işlemi gerçekleştirilmiştir [16]. LBP tabanlı özellikler, noktalar arası ilişkilerin bilgisini taşıdığı için sınıflandırmada başarılı sonuçlar üretmektedir.

Desen ve renk olarak birçok çeşitliliğe sahip kelebeklerin kanat dokularının sınıflandırılması üzerine bir çalışma yapılmıştır [17]. Türlerin ayrımında çoğu zaman desen ve renk özellikleri önemli sonuçlar vermektedir. Bu çalışmada kelebek türlerinin sınıflandırılması için LBP özelliklerinden yararlanılmıştır. Ayrıca LBP özellik çıkarımı için ($nLBPd$) operatörü önerilmiştir. Bu operatör klasik LBP operatörüne göre kelebek görüntülerin sınıflandırılmasında daha yüksek başarı göstermiştir.

Görüntüdeki kişinin yaşını tespit edebilen LBP tabanlı bir çalışma gerçekleştirilmiştir [18]. LBP özellikleri, Öklid uzaklık yöntemi ve Ki-Kare istatistik test yöntemini kullanan k-En Yakın Komşu (K Nearest Neighbour-KNN), tarafından sınıflandırılmıştır.

Görüntü eşleştirmeleri için döngüsel, boyutsal, parlaklık, ölçek ve gürültüye dayanıklı SIFT adlı özellik çıkarım yöntemi duyurulmuştur [19]. SIFT algoritmasının amacı en fazla ayırt edici

özelliği olan köşe noktaları tespit etmektir. En fazla renk geçişi köşe noktalarda olduğundan ayırt edici niteliktedir.

İris görüntülerinin SIFT ve Hızlandırılmış Sağlam Özellikler (Speeded Up Robust Features-SURF) özelliklerinin kullanıldığı, Skor Seviye Füzyonu önerilmiştir [20]. SIFT ve SURF algoritmalarıyla özellik noktalar tespit edilerek, görüntüler üzerinde eşleşme yapılmıştır. Skor Seviye Füzyonu ile elde edilen özellikler birleştirilerek en uygun eşleşme özellikleri tespit edilmiştir.

Çevrimiçi görüntü veritabanlarına saldırı yapıldığında, otomatik olarak özellikleri değiştirebilen bir yöntem önerilmiştir [21]. Görsel Kelime Çantası (Bag of Words-BOW) yöntemi kullanarak özel görüntülere doğrudan erişim sağlanabilir. Günümüzde, çevrimiçi durumun yaygınlaşması ile görüntü veritabanlarına erişme ve arama imkânı artmıştır. Bu veritabanları kötü niyetli saldırılara hedef haline gelebilir.

Dijital kameralar saklama alanlarındaki, farklı balık türlerinin varlığını ve miktarını tespit etmek için kullanılmıştır [22]. Sucul uzmanlar balıkları sınıflandırmak için geleneksel yöntemleri kullanmaktadırlar. Nil nehrindeki Tilapia balıklarını sınıflandırmak için geleneksel yönteme göre daha hızlı ve maliyetsiz SIFT ve SURF özelliklerinden yararlanılmıştır. Bu çalışmada balıkları sınıflandırmak için k-NN, YSA ve DVM kullanılarak sınıflandırma işlemi gerçekleştirilmiştir.

Protein yapıları arasında benzerliklerin tespit edilmesi için, SIFT ve SURF tabanlı özellik çıkarım işleminden yararlanılmıştır [23]. Protein yapılarını anlamak, işlevlerini bulmak için çok önemlidir. Yapısal hizalama, benzerlik ve yapısal fragmanların kullanımı gibi birçok yöntem benzer protein yapılarını bulmak için geliştirilmiştir.

SIFT yöntemi kullanılarak panoramik görüntü için yeni bir harmanlama yöntemi önerilmiştir [24]. Görüntü mozaigi, aynı poza ait sıralı görüntülerin birleştirilerek elde edilen panoramik görüntüdür. Bu yöntemde, birleştirme için çakışan bölgeler etrafında oluşan renk farklılıkları ve yoğunluk değerleri kullanılmıştır.

BOW modeli HEP-2 hücrelerinin sınıflandırılması için kullanılmıştır [25]. Bu model SIFT ve SURF tabanlı özelliklerle birlikte yaygın bir şekilde kullanılmaktadır.

SURF, MSER, RSPE detektörleri insan görüntüsünü tespit etmek için kullanılarak, performans karşılaştırılması yapılmıştır [26]. Sisteme giriş olarak verilen video üzerinden ilgili noktalar tespit edilerek insan görüntüsüyle eşleştirilmiştir.

Histogram, SIFT ve SURF tabanlı özelliklerin kullanıldığı yüz tanıma uygulamasının geliştirildiği bir çalışma yapılmıştır [27]. Bu çalışmada SIFT tabanlı özellikler eşleşme için daha doğru sonuç verirken SURF tabanlı işlemlerin hızlı olduğu gösterilmiştir.

Detektörlerin ve tanımlayıcılardan elde edilen özelliklerin sıkıştırmaya uygunluğu incelenmiştir [28]. Görsel arama ve sınıflandırma uygulamalarında genellikle resim gösterimi ve açıklaması için yerel özellikler kullanılır. Bu çalışmada özellik sıkıştırma işlemi için Fisher vektörü kullanılmıştır. Özelliklerin sıkıştırılabilmesi mobil uygulamalar başta olmak üzere donanım bağımlı sistemlerde verimliliği arttırdığı gösterilmiştir.

SIFT ve SURF detektörleri sahne sınıflamasında kullanılarak performans karşılaştırması yapılmıştır [29]. Sahne sınıflamasında dönme, aydınlatma, ölçek, bakış açısı, ses gibi durumlardan etkilenmeyen görüntü özelliklerine ihtiyaç duyulur. SIFT'in 128 vektörlü modeline göre 64 ve 96 vektörlü modeli ve SURF detektörü işlem süresi açısından daha performanslı sonuçlar verdiği gösterilmiştir.

Wikipedia resimlerinin sınıflandırılmasını sağlayan bir çalışma yapılmıştır [30]. Nesne tanıma, içerik temelli sınıflandırma işleminde önemli bir görevdir. Bu çalışmada nesneleri tanımak için SIFT, SURF ve Odaklanmış Hızlı ve Döndürülmüş Öz (Oriented FAST and Rotated BRIEF-ORB) detektörleri kullanılmıştır.

Özellik çıkarım sürecinde belirli bir miktar da alakasız bilgi yok edilerek boyutsallığın azaltıldığı bir çalışma yapılmıştır [31]. Örüntü tanıma işlemi, özellik çıkarma, özellik seçimi ve sınıflandırma sürecinden oluşan temel unsurları içerir. Bu çalışmada, PCA tabanlı yerel tanımlayıcının, sınıflandırma için yararlı olan özellikleri yakalaması sağlanmıştır.

Uç Öğrenme Makinesinin (Extreme Learning Machine-ELM) düğüm değerleri ve ağırlık değerlerinin nasıl hesaplandığı açıklanmıştır [32]. Huang ve ekibi tarafından önerilen ELM tek katmanlı ileri beslemeli bir sinir ağı modeli için kullanılan öğrenme algoritmasıdır. ELM yönteminin sınıflandırma ve regresyon analizinde yüksek performans verdiği gösterilmiştir.

ELM sınıflandırıcısı, hiperspektral görüntülerin sınıflandırılmasında kullanılmıştır [33]. Ayrıca ELM, Loopy Güven Yayılması ile birlikte kullanılarak başarısı artırılmıştır.

Gerçek zamanlı belgelerin sınıflandırma işleminde, eğitim ve test işlemi aynı anda gerçekleştirilmektedir [34]. Bu çalışmada, Konvolüsyonel Sinir Ağına göre ELM sınıflandırıcısının gerçek zamanlı uygulamada daha başarılı sonuçlar verdiği gözlenmiştir.

Doğrusal ayrılmayan hiperspektral görüntülerin sınıflandırma işlemi gerçekleştirilmiştir [35]. Bu çalışmada ELM sınıflandırıcısı YSA'ya göre yüksek performans sağlamıştır.

Farklı aktivasyon fonksiyonları içeren ELM modellerinin hisse senedi tahmin performansı üzerindeki etkisi incelenmiştir [36]. Modellerin performansı geleneksel YSA ile karşılaştırılmıştır.

Karar ağaçlarında derinliği azaltmak için bir düğümde birden fazla özellik kullanılabildiği bir çalışma yapılmıştır [37]. Birden fazla özellik kullanıldığında karar ağacının boyutu ve derinliği düşeceğinden test işlemi hızlandırdığı verilmiştir.

Karar ağacı üzerindeki kök ve iç düğümlerin çok değişkenli doğrusal bir fonksiyon şeklinde kullanıldığı bir çalışma gerçekleştirilmiştir [38]. Tek değişkenli bir karar ağacı, kökünde ve iç düğümlerinde özelliklerden sadece biriyle sınıflandırma yapmaktadır.

Karar ağacının büyüklüğünün düğüm fonksiyonuna ve sayısına göre hesaplanması verilmiştir [39]. Büyüklük değeri, eşik değerine göre budama işleminde kullanılır.

Karar ağacında bir düğümün bölünüp bölünmeyeceği katışıklık ölçütü kullanılarak hesaplanmıştır [40]. Bu çalışmada bir düğümdeki bütün sınıflar aynı ise o düğüm katışıksız olduğu dikkate alınarak katışıklık ölçütü tanımlanmıştır.

YSA'ların çalışma algoritması ve kullanım alanlarını açıklanmıştır [41]. YSA'lar giriş ve çıkışı olan, sınıflandırma ve kümeleme algoritması olarak kullanılır. Gelişmiş öğrenme yeteneği sayesinde birçok alanda kullanılmaktadır.

Görüntü oluşturmak için çarpma yayılma algoritması kullanan yer radarı (Ground Penetrating Radar-GPR) cihazında, görüntülerin sınıflandırma için YSA kullanılmıştır [42]. YSA'lar GPR ile elde edilen yeraltı jeolojik özelliklerinin simüle görüntülerini sınıflandırmak için kullanılabilir.

Rough-Set YSA modeliyle özellik seçimini geliştirerek, sınıflandırma performansını arttıran hibrit bir yaklaşım önerilmiştir [43]. Mekânsal görüntüleri hibrit yaklaşım ile sınıflandırmak için, çok bantlı bilgi sınıfları çıkarılmıştır.

Sınıflandırma başarısını ve kaynak performansını doğrudan etkileyerek verimliliği arttıran özellik seçim işlemi uygulanmıştır [44]. Özellik çıkarımı görüntü sınıflandırması için çok önemli bir adımdır. Bu çalışmada YSA öğrenme işlemine başlamadan önce özellikler Saliency ısı haritalama seçim işlemine tabii tutulur. Bu seçim işlemiyle görüntü deki en belirgin noktalar üzerinde işlem yapılmıştır.

Tıbbi görüntüler üzerinde YSA, k-Ortalama ve Adaboost algoritmalarının sınıflandırma performansı karşılaştırılmıştır [45]. Görüntü sınıflandırması, uzaktan algılayan tıbbi işlemler için hayati bir öneme sahip olduğundan, yüksek performans sağlayan sınıflandırma işlemlerine ihtiyaç duyulmuştur.

YSA algoritması, uydu görüntülerinde arazi türünü sınıflandırmak için kullanılmıştır [46]. Bu çalışmada denetimli ve denetimsiz eğitim setleri kullanılarak YSA ve k-Ortalama performansı karşılaştırılmıştır.

YSA algoritması radyografik kaynak görüntüleri üzerindeki kusurları sınıflandırmak için kullanılmıştır [47]. Bu çalışmada görüntülerin, histogram tabanlı özellikleri kullanılarak eğitim kümesi oluşturulmuştur.

Nesnelerin doku sınıflandırma işlemi YSA algoritmasıyla gerçekleştirilmiştir [48]. Görüntüyü oluşturan noktalar, YSA ile kümelenerek daha basit ve belirgin dokuya dönüştürülmüştür.

DVM nin algoritmik olarak çalışma adımları detaylı bir şekilde verilmiştir [40]. DVM, modeli oluşturan parametre etkilerinin bir toplamı olarak ifade edilir. Sınıf yapısı ve sayısına göre başarılı bir model oluşturmak için farklı çekirdekler kullanılabilir. Ayrıca bu çalışmada doğrusal ve doğrusal olmayan DVM fonksiyonu açıklanmıştır.

Görüntüler sınıflandırma işlemine sokulmadan önce, ilgili bölgenin daha net algılanması için kümeleme işlemine tabi tutulur [49]. Görüntü kümelenmesinden sonra elde edilen sınıflandırma sonuçları doğrudan sınıflandırmaya göre daha başarılı sonuçlar vermiştir. Bu çalışmadaki denetimli eğitim seti DVM ile sınıflandırılmıştır.

Denetimli görüntülerin, uzaktan algılama ve sınıflandırılması için kesin, hızlı ve verimli makine öğrenme modellerine ihtiyaç duyulmuştur. Temel olarak DVM, parametrik olmayan ikili sınıflandırıcı gibi davranan öğrenme tabanlı bir modeldir [50]. DVM üstün makine öğrenme algoritmalarının bir grubunu temsil eder. Bu çalışmada farklı kernel tiplerine göre DVM nin başarı oranının değişebildiği bir çalışma gerçekleştirilmiştir.

DVM sınıflandırıcısı, hiperspektral görüntüleri sınıflandırmak için kullanılmıştır [51]. DVM temelde iki sınıfı ayırmak için geliştirilse de çoklu sınıflarda da kullanılır. Ancak sınıf sayısı arttıkça işlem maliyeti ve başarı oranı düşebilir. Bu çalışmada aktif öğrenme ile çoklu sınıf içeren görüntü setlerini daha başarılı bir şekilde sınıflandırıldığı görülmüştür.

Alüminyum tel bağlantılarının, ultrasonik görüntüleri üzerinden kalite kontrolünü sağlayan uygulama DVM sınıflandırıcısı kullanılarak gerçekleştirilmiştir [52]. Görüntülere morfolojik işlemler ve görüntü histogram eşitlemesi uygulanarak, daha başarılı özellik çıkarımı için altyapı hazırlanır. Bu çalışmada kaynak noktalarının kalitesi, RBF çekirdeği kullanan DVM ile tespit edilmiştir.

Büyük ölçekli veri kümesinin DVM ile hızlı bir şekilde eğitildiği bir çalışma gerçekleştirilmiştir [53]. DVM’de kernel kullanımı doğrudan başarı ve performansı etkileyen

unsurdur. Bu çalışmada, DVM’de Derin Quasi-Linear Çekirdeği (DQLK) kullanılarak geniş ölçekli resim sınıflandırması yapılmıştır.

DVM ile görüntü işleme tabanlı toprak sınıflandırma işlemi gerçekleştirilmiştir [54]. Toprak türünün belirlenmesi, tarımsal ürün kaybının önüne geçilmesine yardımcı olur. Toprağın sınıflandırılması toprak setlerinin benzer özelliklere göre ayrılmasıdır. Bu işlem laboratuvar ortamında uzun sürede gerçekleştirilir. Ancak görüntü işleme tabanlı kısa sürede ve verimli bir şekilde yapılabilir.

DVM tabanlı sınıflandırma işleminde, paralel programlamanın, ardışıl programlamaya göre daha başarılı sonuçlar verdiği görülmüştür [55]. DVM sınıflandırıcısında kullanılan çekirdekler değiştirilip paralel bir şekilde çalıştırılarak en doğru sonuç elde edilir. Bu çalışmada hiperspektral görüntülerdeki darboğazlar analiz edilerek, paralel işlemle verimli bir şekilde sınıflandırması sağlanmıştır.

1.3. Önerilen Çalışma

Görüntü tabanlı sınıflandırma yapabilmek için iki temel işleme ihtiyaç duyulur. Görüntüleri sayısal olarak ifade edebilmek için özellik çıkarım işlemi birincil işlem olarak yapılır. Özellik çıkarım işlemi sonucunda elde edilen özellikler, sınıflandırma algoritmasını oluşturmak üzere ikinci aşamayı oluşturur. Görüntüleri sınıflandırmak için birçok özellik çıkarım yöntemi ve sınıflandırma algoritması mevcuttur.

Özellik çıkarım aşamasında, görüntülere sayısal değerler atanarak eğitim kümesi oluşturulur. Bu tez çalışmasında LBP tabanlı özellik çıkarım yöntemi kullanılarak eğitim kümesi oluşturulmuştur. LBP özellik vektörü, tekdüze olmayan noktaları bir bütün olarak işleyip, tekdüze noktaları ise ayrı ayrı hesaplar. Bu durum gereksiz noktaların getirdiği işlem maliyetini azalttığından bellek ve işlem süresi açısından performans artışı sağlar. Ayrıca noktaların komşularıyla olan ilişki bilgisi hesaplandığından sınıflandırma işleminde yüksek başarıma sahiptir. Bu avantajlarından dolayı yaygın bir şekilde kullanılmaktadır. LBP özellik çıkarımının sınıflandırmadaki başarısını ölçmek için SIFT ve histogram özellik çıkarım yöntemleri, çoklu özellik çıkarımı altında kullanılmıştır.

Görüntüleri eşleştirmek için ayırt edici özelliklerin tespit edilmesi gerekmektedir. Özellik noktaları tespit edilirken, görüntülerin aynı koşullarda elde edilmesi başarıyı artırır. Genellikle farklı koşullarda elde edilen görüntüler üzerinde eşleştirme işleminin çalışması istenir. SIFT

tabanlı özellikler döngüsel, boyutsal, parlaklık, ölçek ve gürültüye dayanıklı olduğundan sınıflandırmada yaygın bir şekilde kullanılır.

Doku sınıflandırmasında kullanılan en temel özellik çıkarım yöntemlerinden biri Histogram tabanlı özellik çıkarımıdır. Bu yöntem LBP özelliklerinin sınıflandırma performansını karşılaştırmak için kullanılmıştır. Histogram tabanlı özellik çıkarım işleminden görüntünün 'Ortalama', 'Varyans', 'Çarpıklık', 'Basıklık', 'Enerji', 'Entropi', 'Zıtlık', 'Homojenlik' ve 'Korelasyon' değerleri hesaplanır. Histogram özellikleri birbirleriyle bağlantılı olarak hesaplandığından işlem süresi kısadır.

LBP'nin sınıflandırma başarısını ölçmek için kullanılan SIFT ve Histogram özellikleri, geliştirilen GUI aracılığıyla elde edilmiştir. GUI, özellik çıkarım ve sınıflandırma işlemi için deneysel test ortamı sunar. Görüntü sınıflandırma işleminin, özellik çıkarımından sonra gelen ikinci adımı sınıflandırma işlemidir. Bu çalışmada mermer görüntülerini sınıflandırmak için ELM sınıflandırıcısı kullanılmıştır. ELM'nin sınıflandırma başarısı toplu (Ensemble) sınıflandırıcıyla karşılaştırılmıştır. Toplu sınıflandırıcı Karar Ağacı, YSA ve DVM sınıflandırıcılarından oluşmaktadır.

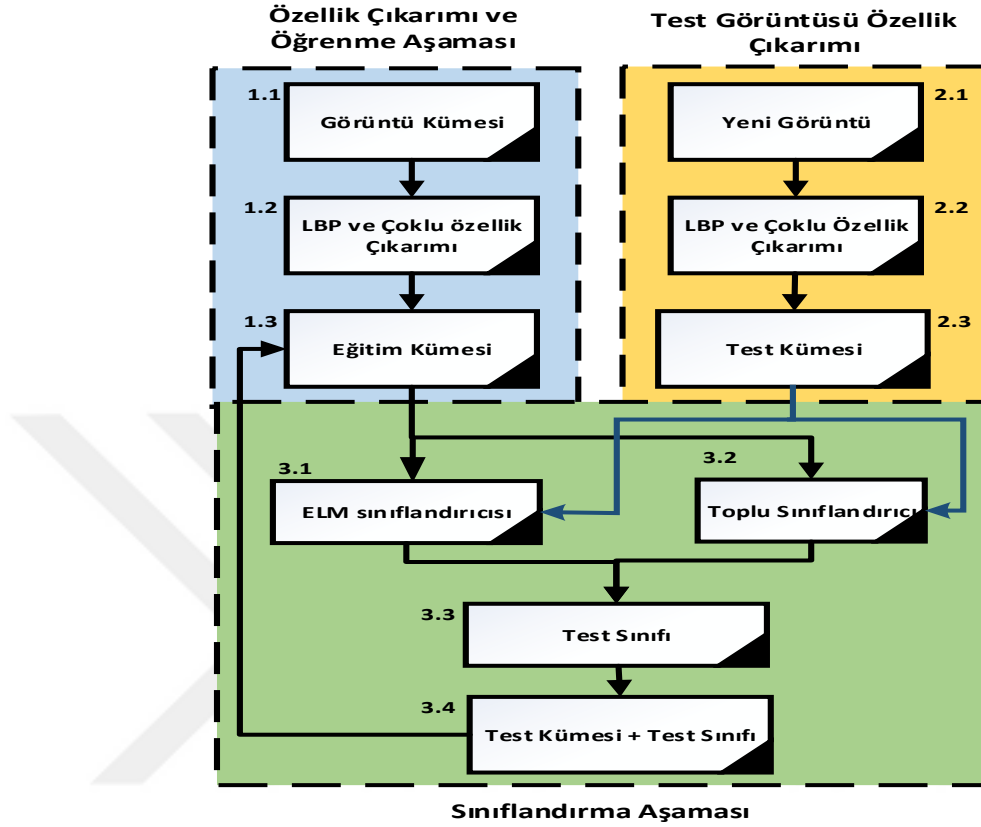
ELM sınıflandırıcısı, tek katmanlı ileri beslemeli bir YSA'dır. ELM, tek bir çevrimde ağırlık noktalarını hesaplayarak sonuca ulaşmaktadır. Bu durum döngüsel çalışan YSA'ya göre daha kısa sürede öğrenme işleminin gerçekleşmesini sağlar. ELM'de ara katmandaki hücre sayısı YSA'daki gibi değiştirilebilir. Böylece karmaşık eğitim setlerinin yüksek başarı ile sınıflandırılması sağlanır. Ayrıca YSA'nın aksine, kullanıcı bağımlı az sayıda parametreye ihtiyaç duyulması, karmaşıklığı azaltır.

Toplu sınıflandırıcı içerisinde kullanılan Karar ağacının hesapsal maliyeti ve hafıza tüketimi düşük olduğundan birçok alanda kullanılır. Bu çalışmada sayısal eğitim kümesi üzerinde çalışan, C4.5 tabanlı Karar ağacı kullanılmıştır. Toplu sınıflandırıcı içerisinde kullanılan DVM sınıflandırıcısı, karmaşık olmayan eğitim kümelerinde yüksek hızlı sonuçlar üretir. Sınıflandırma fonksiyonu sınıfların birbirine en yakın noktaları üzerinden işlem yaptığından, gereksiz noktaların getirdiği hafıza kaybı ve işlem yükü elimine edilir. Ancak çoklu sınıfa sahip eğitim kümelerinde, birden fazla doğrusal fonksiyon ile sınıflandırması işlem yükünü ve süresini arttırır. Toplu sınıflandırıcıda kullanılan bir diğer sınıflandırıcı YSA'dır. Bu sınıflandırıcıda katman sayısı ve hücre sayısı değiştirilerek karmaşık eğitim verilerinin sınıflandırılma başarısı arttırılır. Bu çalışmadaki YSA, yapısal olarak "*İleri Beslemeli*", öğrenme yöntemine göre "*Danışmanlı Öğrenme*" ve öğrenme zamanına göre "*Statik Öğrenme*" yapısındadır.

Geliştirilen GUI üzerinde sınıflandırma işleminin iki temel adımı olan özellik çıkarımı ve sınıflandırma işlemi gerçekleştirilir. GUI, deneysel test ortamını sağlamanın yanı sıra son kullanıcıya yönelik endüstriyel bir yazılım hizmeti sunar. Özellik çıkarım işlemi ve sınıflandırma işlemi alt birimler şeklinde geliştirilmiştir. Kullanıcı görüntüleri GUI üzerinden sisteme tanıtılarak LBP, Histogram ve SIFT tabanlı özellik çıkarım işlemini gerçekleştirir. Eğitim kümesi oluşturulan mermer türü, sınıflandırma işlemine hazırdır. GUI üzerindeki sınıflandırma işleminde, sınıfı bilinmeyen bir test görüntüsü ELM ve toplu sınıflandırıcı kullanılarak sınıflandırılır. Sınıflandırma algoritmalarına ait parametreler, GUI üzerinde değiştirilerek sınıflandırma başarımına olan etkileri izlenir. ELM'nin toplu sınıflandırıcıya karşı sınıflandırma başarısı, sınıflandırma süresi ve sınıflandırma sonucu karşılaştırılır. Sınıflandırıcılar GUI üzerinde paralel bir şekilde çalıştırılır. ELM ve toplu sınıflandırıcının sınıflandırma işlemi tamamlandığında, GUI sonuçları kullanıcıya göstermek için sonuç menüsüne yönlendirir. Sonuç sayfasında, sınıflandırıcıların çapraz doğrulamalı başarı oranları, sınıflandırma süreleri ve sınıflandırma sonucu verilir. ELM ve toplu sınıflandırıcının sonuçlarına göre test görüntüsünün sınıfı, “Kazanan Alır” algoritmasına göre belirlenir. Bu algoritmaya göre, test görüntüsünün sınıfı, sınıflandırıcıların verdiği sınıfların frekansına ve sınıflandırma başarısının büyüklüğüne göre aşamalı bir şekilde belirlenir. Test sınıfını belirlemenin birincil aşamasında, sınıf sayısı fazla olan test sınıfı olarak atanır. Sınıf sayısında eşitlik mevcut ise sınıflandırma başarıları toplanarak, yüksek başarı oranına ait sınıf test görüntüsünün sınıfı olarak atanır. GUI'nin sonuç sayfasında, eğitim kümesini geliştirme işlemi de gerçekleştirilir. Test görüntüsünün sınıfı kullanıcı tarafından sakıncalı bulunduğu durumda değiştirilerek özellikleriyle birlikte eğitim kümesine eklenmesi sağlanır. Bu sayede sınıflandırma algoritmaları güncellenmiş eğitim kümesini kullanarak daha başarılı bir sınıflandırma yapabilmektedir.

Geliştirilen GUI ile mermer sınıflandırma işlemini gerçekleştirmek için kullanıcının uzman olmasına gerek yoktur. GUI Java ve Python programlama dilleri kullanılarak geliştirildiğinden işletim sisteminden bağımsız bir şekilde çalışabilmektedir. Ancak GUI'nin sanayide daha verimli bir şekilde hizmet edebilmesi için, gömülü sistem üzerinde çalışması gerekmektedir. Geliştirilen GUI yazılım mimarisi bakımından Linux ve Windows tabanlı gömülü sistemlerde çalışabilir. Gömülü sistemlerin kişisel bilgisayarlara göre işlem kapasitesi ve hızı daha düşük kaldığından, geliştirilen GUI'nin gömülü sistemde çalıştırılması performansını etkiler. Bu tez çalışmasının sonraki sürecinde, gömülü sistem üzerinde çalışabilen daha verimli GUI yazılım mimarisi geliştirilebilir.

Önerilen mermer sınıflandırma sisteminin akış şeması Şekil 1.2’de verildiği gibi üç aşamada gerçekleşir.



Şekil 1.2. Önerilen mermer sınıflandırma sistemi akış şeması

Şekil 1.2’deki akış şeması adımlarının ayrıntılı açıklaması ilerleyen bölümlerde, özet açıklaması aşağıda verilmiştir.

Özellik çıkarım ve öğrenme aşaması: Görüntü kümesine LBP ve çoklu özellik çıkarım işlemi uygulanır. Özellik çıkarım işlemi sonucunda görüntülerin özellikleri dosyaya kaydedilerek eğitim kümesi oluşturulur. Çoklu özellik çıkarımı Histogram ve SIFT tabanlı özellik kümelerini kapsar.

Test görüntüsü özellik çıkarımı: Denetimsiz görüntünün sınıfını belirlemek için gerçekleştirilen özellik çıkarma işlemi bu aşamada uygulanır. Denetimsiz görüntüye GUI üzerindeki tercihe göre LBP, Histogram veya SIFT tabanlı özellik çıkarım işlemi uygulanır. Test kümesi sadece bir görüntünün LBP, Histogram veya SIFT tabanlı özelliklerini içerir.

Sınıflandırma aşaması: Eğitim kümesini kullanan sınıflandırma algoritmalarının, denetimsiz görüntünün sınıfını belirlediği aşamadır. Bu aşamada, ELM ve toplu sınıflandırıcı kullanılmıştır. Toplu sınıflandırıcı Karar Ağacı, YSA ve DVM’den oluşmaktadır.

Sınıflandırıcılar, eğitim kümesini kullanarak sınıflandırma fonksiyonlarını üretir. İkinci aşamada oluşturulan denetimsiz görüntünün özellikleri sınıflandırma fonksiyonları üzerinden teste tabii tutulur. Denetimsiz görüntünün sınıfı bu test sonucunda belirlenir. Denetimsiz görüntünün sınıfı ve özellikleri birleştirilerek eğitim kümesine eklenmesi sağlanabilir. Bu ekleme işlemi ile eğitim kümesinin kendi kendini geliştirmesi sağlanmıştır.

Tez çalışmasının bundan sonraki bölümlerinde sırasıyla özellik çıkarımı, sınıflandırma işlemi, deneysel sonuçlar ve GUI geliştirme aşamaları ile sonuçlar verilmiştir.

1.4. Amaç

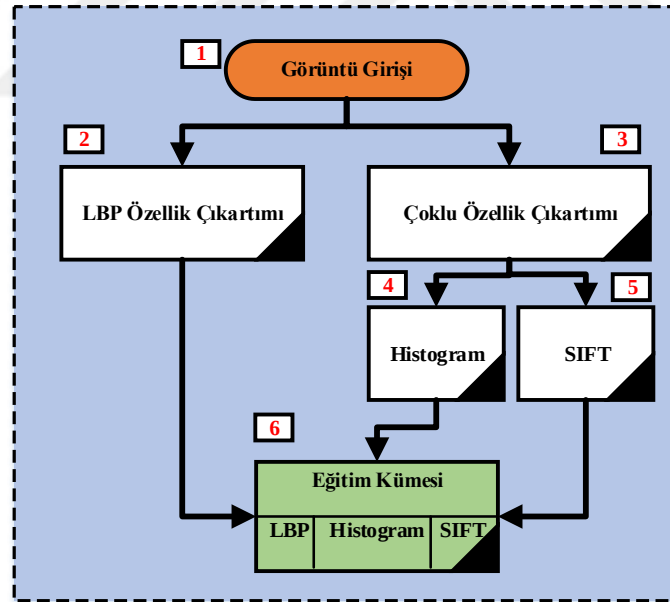
Tez çalışmasında, mermer sınıflandırma işlemini öznel yapıdan çıkarıp nesnel yapıya dönüştüren deneysel çalışma mimarisinin geliştirilmesi amaçlanmıştır. Ayrıca deneysel ortamın sağlanması ve endüstriyel kullanıma yönelik grafiksel arayüzün geliştirilmesi hedeflenmiştir.

2. ÖZELLİK ÇIKARIMI

Özellik çıkarım işlemi görüntüyü tanımlayan dokuları tespit etmek için kullanılan genel bir yöntemdir. Görüntü üzerindeki dokuları ayırt etmek için birçok özellik çıkarım yöntemi mevcuttur. Görüntü kümesinin sınıflandırma ölçütlerine göre özellik çıkarım yöntemi belirlenirse, sınıflandırma için daha yüksek başarımlar elde edilebilir.

Görüntüyü oluşturan renklerin geometrik bir şekilde diziliminden meydana gelen dokular, görüntü sınıflandırma işlemlerinde görüntüleri birleştirici ve ayırıcı etkiye sahiptir. Dokuyu oluşturan renkler, görüntünün frekans, faz, rastgelelik, tekdüzelik, yoğunluk, doğrusallık özelliklerini içerir.

Özellik çıkarım aşamasında mermer görüntülerine sayısal özelliklerden oluşan etiketler atanır. Görüntünün özelliklerinden oluşan eğitim kümesi sınıflandırma ve kümeleme işlemlerinde kullanılır. Özellik çıkarım işleminin giriş ve çıkış birimlerini içeren akış şeması Şekil 2.1’de verildiği gibidir.



Şekil 2.1. Özellik çıkarım işlemi

Şekil 2.1’de verilen özellik çıkarım işlemi, Şekil 1.2’deki genel akış şemasının birinci ve ikinci aşamasını oluşturmaktadır. Şekil 1.2’deki akış şemasının birinci bölümünde özellik çıkarım işlemi eğitim kümesi için kullanılır. Bu bölümde özellik çıkarım işlemine uygulanan girişler denetimli görüntülerdir. İkinci aşama da ise denetimsiz test görüntüsü giriş verisi olarak

özellik çıkarım işlemine uygulanır. Özellik çıkarım işleminin aşamaları, Şekil 2.1’de verilen notasyona göre aşağıda özetlenmiştir:

1. Özellik çıkarım işlemine girdi olarak denetimli görüntü kümesi veya denetimsiz test görüntüsü verilir.
2. Yerel İkili Örüntü (Local Binary Pattern-LBP) tabanlı özellik çıkarım işlemi, görüntü kümesi veya test görüntüsüne uygulanır. Bu işlemin sonucunda, standart şekle getirilmiş LBP histogramı elde edilir.
3. Çoklu özellik çıkarım işlemi başlatılır. Bu işlem, kendi içerisinde bulunan Histogram ve Ölçekten Bağımsız Özellik Dönüşümü(Scale Invariant Feature Transform -SIFT) tabanlı özellik çıkarım işlemlerini başlatır.
4. Histogram tabanlı, 'ortalama', 'varyans', 'enerji', 'rassallık', 'bağılılaşım', 'zıtlık', 'homojenlik', 'çıkıntı' ve 'gölge' özellikleri hesaplanır.
5. SIFT tabanlı özelliklere kelime çantası modeli uygulanır. Bu model ile görüntüler en çok barındırdığı görsel kelimelere göre adlandırılır.
6. LBP, Histogram ve SIFT tabanlı özellikler ayrı ayrı dosyalara kaydedilerek eğitim kümesi oluşturulur.

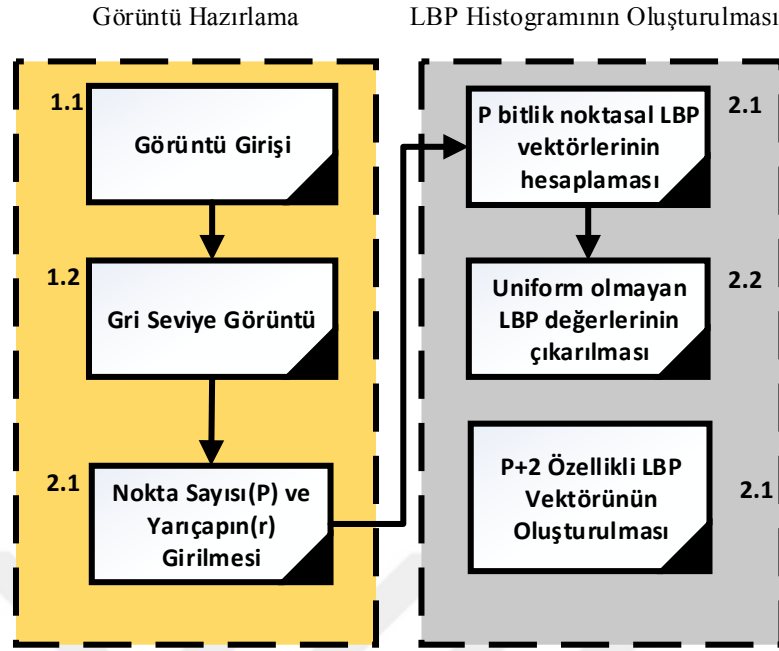
2.4. LBP’den Elde Edilen Özellikler

LBP detektörü yüksek performanslı bir doku tanımlayıcısı olması nedeniyle doku sınıflandırma ve kümeleme işlemlerinde kullanılmaktadır. LBP, dokunun ayırt edici önemli özelliklerini belirler. Önemli noktalar histogram üzerinde birleştirilerek, özellik vektörü oluşturulur.

2.4.1. LBP Uygulama Adımları

LBP detektörü görüntünün her bir noktası için komşuluk vektörü oluşturmaktadır. Komşuluk vektörü, merkez noktanın etrafında bulunan noktalarla ilişkisini belirtir.

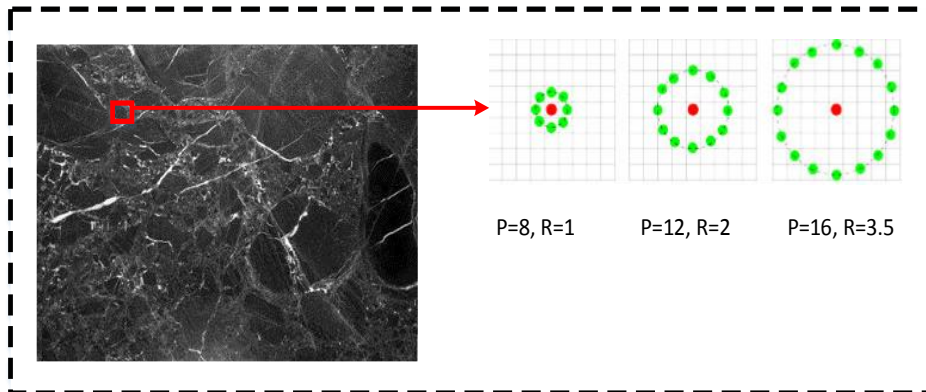
LBP özellik histogramının oluşturma adımları Şekil 2.2’de verildiği gibidir.



Şekil 2.2. LBP detektörü akış şeması

Şekil 2.2’de verilen akış şemasında, LBP özelliklerinin oluşturma işlemi iki genel aşamada açıklanmıştır.

Görüntü Hazırlama: Bu aşamada görüntü girişi gerçekleştirilir. Görüntü gri seviyeye dönüştürülerek LBP özellik çıkarımı için hazır hale getirilir. Gri görüntü üzerinde noktalar arası ilişkinin hesaplanacağı nokta sayısı ve yarıçap bilgileri girilir.



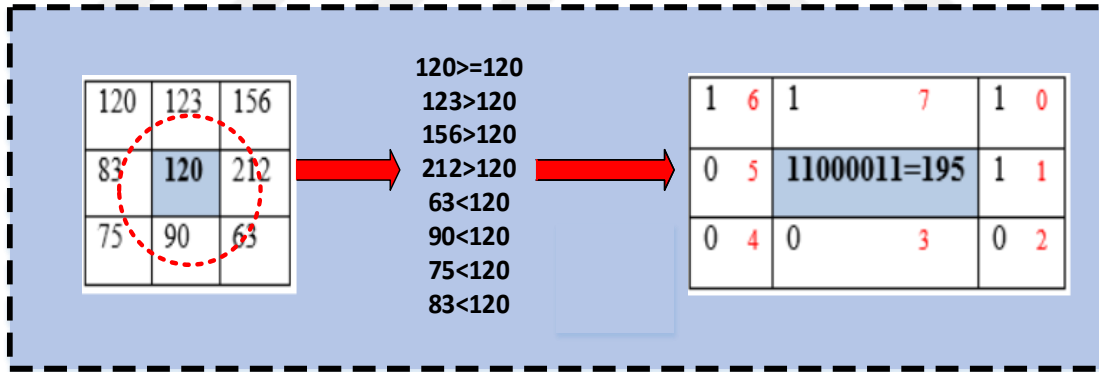
Şekil 2.3. LBP detektörü doku belirleme işlemi

Şekil 2.3'te, P (Nokta Sayısı) ve R (Yarıçap) değerlerine göre LBP dizisi hazırlanır. P değerinin büyük olması histogramı büyütürken aynı zamanda işlem maliyetini de artırır. P değerinin küçük girilmesi ise doku analizinde bilgi kaybına neden olur. R komşu noktaların merkez noktaya olan uzaklığıdır. Farklı R değerinin girilmesi farklı boyuttaki dokuların analizini mümkün kılmaktadır.

LBP Histogramının Oluşturulması: Bu aşamada P ve R değerine göre merkez nokta çevre noktalarla karşılaştırılır. Denklem 2.1'deki y değeri kendisi ve komşuları arasındaki farkın basamak fonksiyonunun, ikili tabana dönüştürülmesiyle etiketlenir.

$$LBP_{P,R}(x_c) = \sum_{p=0}^{P-1} u(x_p - x_c) 2^p, u(y) = \begin{cases} 1 & y \geq 0 \\ 0 & y < 0 \end{cases} \quad (2.1)$$

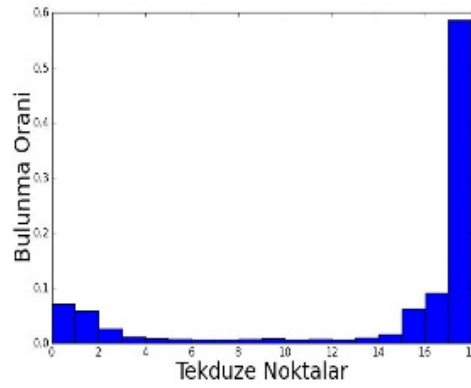
Denklem 2.1'de x_c değeri LBP etiketi üretilen merkez noktadır. x_p değeri ise merkez noktanın komşularıdır. Her bir LBP noktası, P bitlik ikili sayı ile temsil edilir. $P=8$ ve $R=1$ değerleri için LBP nokta tanımlayıcısı Şekil 2.4'deki gibi hesaplanır.



Şekil 2.4. P bitlik LBP noktasının hesaplanması

Şekil 2.4'te verilen LBP nokta tanımlayıcısı görüntü üzerindeki bütün noktalara ayrı ayrı uygulanarak LBP histogramı oluşturulur. Histogramdaki noktaların tamamı doku tanımlamada kullanılamaz. Doku yüzeyini oluşturan noktalar tekdüze noktalardır. Tekdüze noktalar, LBP örüntüsünde 0-1 yada 1-0 geçiş sayısı en fazla 2 olanlardır. '0000000' ve '1111111' örüntüleri "0" geçişe sahiptir. 00011100 ve 11110011 örüntüleri 2 geçiş içerdiği için tekdüzedir. LBP örüntüsünde 2 den fazla geçiş içeren noktalar tekdüze olmayıp histogram da tek bir bölmede toplanır. Bu çalışmadaki tekdüze olmayan noktalar histogramın son bölümünde toplanmıştır.

LBP örüntüsünde 8 nokta için 256 farklı renk düzeyi oluşmaktadır. Bu renk örüntülerinden sadece 58 tanesinin tekdüze olduğu görülmektedir. Bunlar ‘00000000’, ‘00000001’, ‘00000011’, ‘00000111’, ‘00001111’, ‘00011111’, ‘00111111’, ‘01111111’, ‘11111111’ ve bu örüntülerin dairesel kombinasyonlarıdır. LBP histogramında 58 adet tekdüze, 1 adet tekdüze olmayan örüntüler için toplam da 59 adet nokta bulunur. Tekdüze noktaların sayısı $P*(P-1)+2$ şeklinde hesaplanabilir [17]. LBP histogramının boyutu $P+2$ olarak hesaplanır. Histogramı oluşturan tekdüze noktalar standartlaştırılarak yüzdelik değerleri histograma eklenir. Şekil 2.5’te $P=16$ ve $R=4$ değerlerine sahip bir LBP histogramı verilmiştir.



Şekil 2.5. A sınıfına ait örnek bir mermerin LBP histogramı

Şekil 2.5’teki histogramda, x düzlemindeki her bir nokta görüntünün LBP tabanlı özelliğini oluşturmaktadır. Her x değerine karşılık gelen y değeri ise tekdüze noktalar içerisinde o gruba ait noktaların bulunma sıklığını gösterir. Şekil 2.5’teki LBP histogramından elde edilen özellikler, Tablo 2.1’de verilmiştir.

Tablo 2.1. LBP histogramından elde edilen özellikler

1. dereceden histogram özellikleri			
X0	0.0359	X9	0.0635
X1	0.0359	X10	0.0295
X2	0.0233	X11	0.0219
X3	0.0184	X12	0.0166
X4	0.0188	X13	0.0183
X5	0.0229	X14	0.0232
X6	0.0312	X15	0.0406
X7	0.0511	X16	0.0441
X8	0.0867	X17	0.4173
İşlem Özellikleri			
İşlenme süresi	0.359 sn Tekdüze olmayan küme: X17		

Tablo 2.1’de verilen LBP tabanlı özellikler yüzdelik değerleri içerir. Bütün değerler toplandığında sonuç birdir. X17’nin en büyük değere sahip olmasının nedeni tekdüze olmayan bütün noktaları içermesidir. Bütün görüntülere yukarıdaki işlemler ayrı ayrı uygulanarak eğitim kümesi oluşturulur. LBP tabanlı, eğitim seti oluşturma işleminin sözde kodu Tablo 2.2’de verildiği gibidir.

Tablo 2.2. LBP tabanlı eğitim kümesi oluşturma işlemi sözde kodu

```

1: INPUT; // Görüntü,nokta sayısı ve çapın girilmesi
2: Image <- image
3: numPoints <- noktasyısı
4: radius <- çap
5: FUNCTION describe(image, numPoints, radius eps=1e-7):
6:     lbp <- feature.local_binary_pattern(image, numPoints, radius, method="uniform")
7:     (hist, _) <- np.histogram(lbp)//LBP özelliklerinin histograma dönüştürülmesi
8:     hist <- hist.astype("float")
9:     hist <- hist /= (hist.sum() + eps) //LBP histogramının standardizasyonu
10:    //LBP histogramının görselleştirilmesi
11:    fig,ax4 <- plt.subplots(nrows=1, ncols=1,figsize=(9, 6))
12:    ax4.hist(refs['brick'].ravel(), normed=True, bins=18, range=(0, 18))
13:    ax4.set_ylabel('Percentage',fontsize=24)
14:    cwd <- os.getcwd()//dosya konumunun alınması
15:    yol <- cwd+"\dist\image\hist.jpg" //Histogramın kayıt yolunun oluşturulması
16:    resized1 <- plt((320,240))//Histogram görüntüsünün ölçeklenmesi
17:    savefig(yol) <- plt//Histogram görüntüsünün kaydedilmesi
18:    RETURN hist
19: END FUNCTION

```

2.5. Çoklu Özellik Çıkarımı

Eğitimi kümesinin sınıflandırma başarımını karşılaştırmak için Histogram ve SIFT özellikleri çoklu özellik çıkarımının içerisinde kullanılmıştır.

2.5.1. Histogramdan Elde Edilen Özellikler

Mermerin dokusunu sayısal olarak tanımlayabilmek için kullanılan özellik çıkarım işlemlerinden biri de histogram tabanlı özelliklerdir. Birinci dereceden histogram tabanlı özellikler 'Ortalama', 'Varyans', 'Çarpıklık', 'Basıklık' 'Enerji' ve 'Entropi' dir. İkinci dereceden histogram tabanlı özellikler ise 'Zıtlık', 'Enerji', 'Homojenlik' ve 'Korelasyon' 'dur. Histogram tabanlı özellik çıkarımı için görüntü gri seviyeye çevrilir. Gri seviye Histogramı renk

düzeylerini içeren toplam nokta sayısını verir. Histogramın uygulama aşamaları, aşağıda adım adım verilmiştir.

1. Görüntünün boyutu $A \times B$ şeklinde ifade edilir.
2. 2 boyutlu görüntü fonksiyonu $f(x, y)$ ve $x=0, 1, 2 \dots A, y=0, 1, 2 \dots B$ şeklinde tanımlanır.
3. Her $f(x, y)$ fonksiyonu için $i=0, 1, 2 \dots G$ 'ye kadar gri renk seviyesini belirtir.
4. Her piksel 8 bit ile ifade edildiğinden, maksimum $G=2^8=256$ şeklinde işlem görür.
5. Histogramda herhangi bir renk düzeyinin görüntüde bulunan toplam sayısı $h(i)$ şeklinde gösterilir. Örnek olarak " $h(137)=60$ " işlemi 137. gri seviyesinden, görüntüde toplam 60 noktanın olduğunu belirtir.
6. Histogram üzerindeki gri renk seviyesinin yoğunluğunu ifade eden $p(i)$ değeri denklem 2.2'de verilmiştir.

$$p(i) = h(i)/(A \times B) \quad (2.2)$$

2.5.1.1. Birinci Derece Histogram Özellikleri

İşlenmemiş görüntünün birinci derece histogram özellikleri aşağıdaki gibi hesaplanır.

Ortalama: Görüntüdeki gri renk seviyelerinin ortalamasının toplamıdır.

$$\mu = \sum_{i=0}^{G-1} ip(i) \quad (2.3)$$

Varyans: Gri renk düzeyinin ortalaması üzerindeki yoğunluk değişimidir.

$$\sigma_2 = \sum_{i=0}^{G-1} (i - \mu)^2 p(i) \quad (2.4)$$

Çarpıklık: Simetrikliğin ölçüsü olarak işlem görmektedir.

$$\mu_3 = \sigma - 3 \sum_{i=0}^{G-1} (i - \mu)^3 p(i) \quad (2.5)$$

Basıklık: Histogramın sivrililiği ve basıklığı hakkında bilgi verir.

$$\mu_4 = \sigma - 4 \sum_{i=0}^{G-1} (i - \mu)^4 p(i) - 3 \quad (2.6)$$

Enerji: Gri renk düzeylerinin ortalamalarının kareleri toplanarak hesaplanmaktadır.

$$E = \sum_{i=0}^{G-1} [p(i)]^2 \quad (2.7)$$

Entropi: Histogramın tekdüzeliğini ölçmek için kullanılır.

$$H = - \sum_{i=0}^{G-1} p(i) \log_2 [p(i)] \quad (2.8)$$

2.5.1.2. İkinci Derece Histogram Özellikleri

İkinci dereceden histogram özellikleri, birinci dereceden histogramın Eşdizimli (Co-Occurrence) matristen elde edilen özelliklerdir. Eşdizimli matris, noktaların birbirleriyle olan ilişkisi üzerinden işlem yaptığından birinci dereceden özelliklere göre daha belirleyicidir.

Enerji: Histogramın benzerliğini ölçmek üretilen bir özelliktir.

$$Energy = \sum_i \sum_j P^2[i, j] \quad (2.9)$$

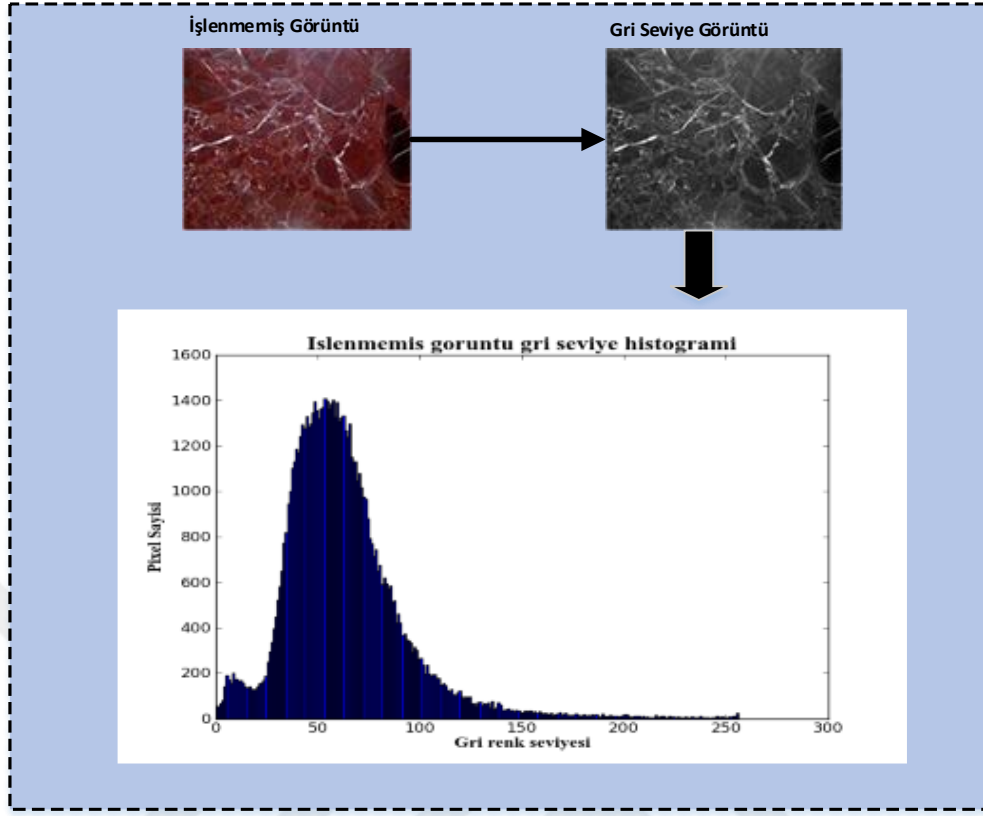
Zıtlık: Görüntünün renk düzeyine göre, en açık ve en koyu noktaları arasındaki farktır.

$$Contrast = \sum_i \sum_j (i - j)^2 P[i, j] \quad (2.10)$$

Homojenlik: Görüntü içerisinde dokuları oluşturan renk dağılımının ölçüsüdür.

$$Homogeneity = \sum_i \sum_j \frac{P[i, j]}{1 + |i - j|} \quad (2.11)$$

İşlenmemiş görüntü gri seviye görüntüye dönüştürüldükten sonra histogram alma işlemi gerçekleştirilir. Histogram oluşturma işlemi aşamalı olarak Şekil 2.6'da verilmiştir.



Şekil 2.6. Histogram oluşturma işlemi

Şekil 2.6'daki histogramdan elde edilen özellikler Tablo 2.3'te verilmiştir.

Tablo 2.3. Histogramdan elde edilen özellikler

1. dereceden histogram özellikleri	
Ortalama	62.14578125
Varyans	29.7982508385
Çarpıklık	1.74027596089
Basıklık	9.39548737006
Enerji	0.0122886352539
Entropi	-6.70870635827
2. dereceden histogram özellikleri	
Zutluk	0.526084117032
Enerji	0.455327864299
Homogeneity:	0.814151903982
Korelasyon	0.71027473763
İşlem Özellikleri	
Kalite Sınıfı	A kalite
İşlenme süresi	0. 327 sn

Birinci derece Histogram özelliklerini, alma işleminin sözde kodu Tablo 2.4’te verilmiştir. İkinci dereceden özellik çıkarım işlemi, Opencv tabanlı kütüphane aracılığıyla alındığından sözde kodu verilmemiştir.

Tablo 2.4. Histogram özellik çıkarımı sözde kodu

```

1:  INPUT; // Görüntü,nokta sayısı ve çapın girilmesi
2:  hst <- histogram
3:  NM <- görüntü boyutu
4:  eachCount <- hst[0] // Gri renk seviyesi nokta sayısı
5:  FUNCTION featureExtraction(hst, NM):
6:    for num <- to (0, len(eachCount)) do
7:      mean <- mean + (num*eachCount[num] // Denklem 2.3
8:    endfor
9:    for num1 <- to (0, len(eachCount)) do
10:     varyans <- varyans + ((num1-mean)**2)*(eachCount[num1] / NM) // Denklem 2.4
11:   endfor
12:   varyans <- math.sqrt(varyans)
13:   for num2 <- to (0, len(eachCount)) do
14:     skw <- skw + ((num2-mean)**3)*(eachCount[num2] / NM)
15:     krt <- krt+ ((num2-mean)**4)*(eachCount[num2] / NM)-3
16:     energy <- energy + (eachCount[num2] / NM)**2 // Denklem 2.7
17:     if (eachCount[num2] != 0) then
18:       // Denklem 2.8
19:       entropy <- entropy + (eachCount[num2] / NM)*math.log(eachCount[num2] / NM) /math.Log(2)
20:     endif
21:   endfor
22:   skewness <- skw / (varyans**3) // Denklem 2.5
23:   kurtosis <- krt / (varyans**4) // Denklem 2.6

```

2.2.2. SIFT Algoritması

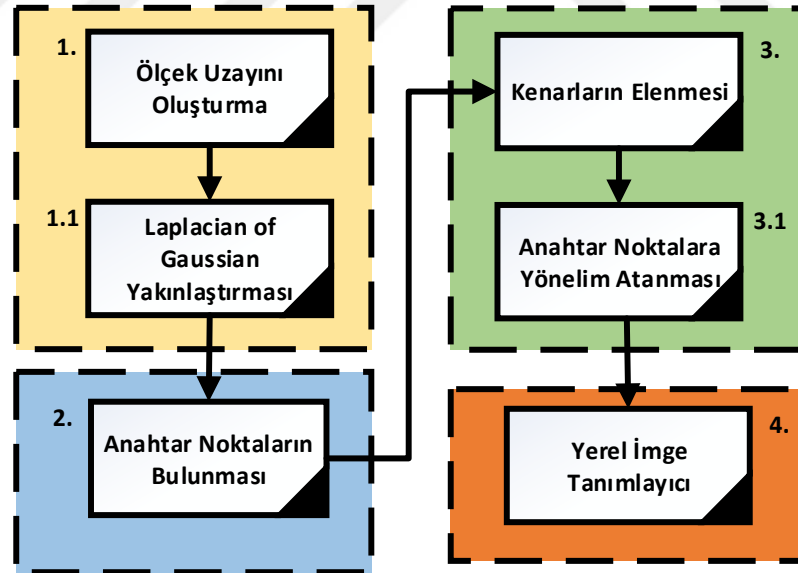
Günümüzde resimleri eşleştirmek için çeşitli yöntemler bulunmaktadır. Görüntüler kendisini tanımlayan benzer sayısal özellikler üzerinden karşılaştırılır. Dolayısıyla görüntünün benzer alanlara bölünmesi gerekmektedir. Her bölümlene algoritması iki temel sorunu iyileştirmeyi hedefler. Bunlar, bölümlmeyi gerçekleştirmek için uygun ölçüt seçimi ve bölümlmeyi etkin bir şekilde gerçekleştiren yöntem seçimidir. Görüntü üzerinde verimli bir bölümlene için ayırt edici özellikli noktalar gereklidir. Özellikli noktaları tespit eden en yaygın nokta tanımlayıcılarından biri de SIFT algoritmasıdır.

2.2.2.1. SIFT Uygulama Adımları

SIFT detektörü farklı görünüm arasında güvenilir eşleştirme gerçekleştirmek için kullanılabilecek bir anahtar nokta tanımlayıcısıdır. Bu anahtar nokta tanımlayıcısı döngüsel, boyutsal, parlaklık, ölçek ve gürültüye karşı dirençli olduğundan yaygın bir şekilde kullanılmaktadır. SIFT detektörü aşamalı bir şekilde gerçekleşen dört adımdan oluşur.

- 1) Ölçek uzayındaki uç değer noktaların tespiti
- 2) Anahtar noktaların bulunması
- 3) Yönelimlerin belirlenmesi
- 4) Anahtar nokta tanımlayıcıları

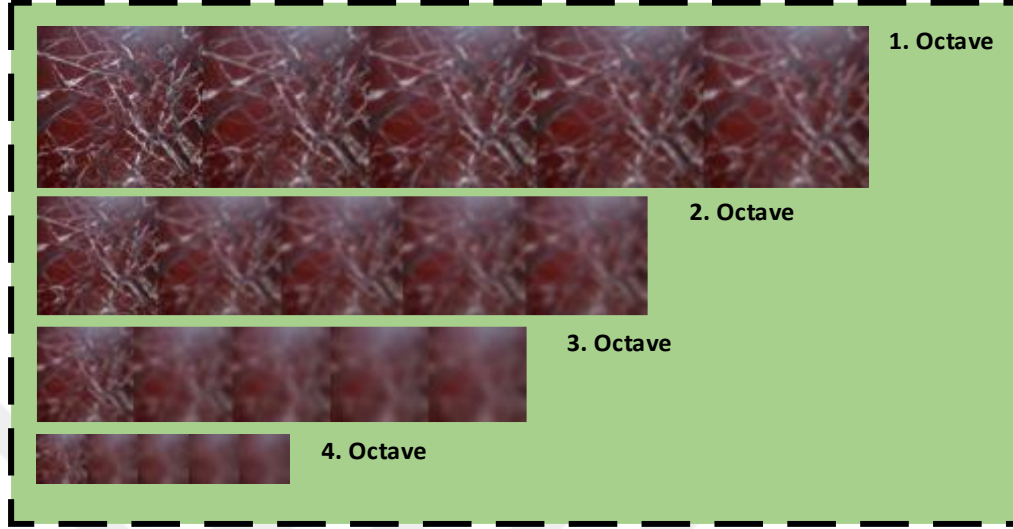
Yukarıda ardışıl gerçekleşen dördüncü adımın sonunda özellik tanımlayıcıları elde edilir. Bu adımları gerçekleştiren akış şemasının adımları farklı renklere gelecek halinde Şekil 2.7’de verilmiştir.



Şekil 2.7. SIFT Detektörü Akış Şeması

Ölçek Uzayını Oluşturma: Görüntünün boyutu yarıya düşürülerek ikinci seviye görüntü elde edilir. Her seviyedeki görüntü kümesi octave olarak adlandırılır. Görüntü boyutu yarıya düşürülerek 4 octave elde edilir. Octave içindeki her görüntü kademeli olarak arttırılarak

bulanıklaştırılır. Görüntüden elde edilen octave gruplarının bulanıklaştırılması Şekil 2.8’deki gibidir.



Şekil 2.8. Octave gruplarının yumuşatma işlemi

Literatürde yapılan çalışmalarda en uygun octave sayısı ve bulanıklaştırma sırasıyla 4 ve 5 olarak önerilmektedir [12]. İşlenmemiş görüntü, Gauss fonksiyonuyla çarpılarak bulanıklaştırılır. Gauss fonksiyonu denklem 2.12’de verilmiştir.

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2} \quad (2.12)$$

Yumuşatılmış Görüntü ise Gauss Gaus fonksiyonu ve Ham Görüntünün çarpımı olarak denklem 2.13’te verilmiştir.

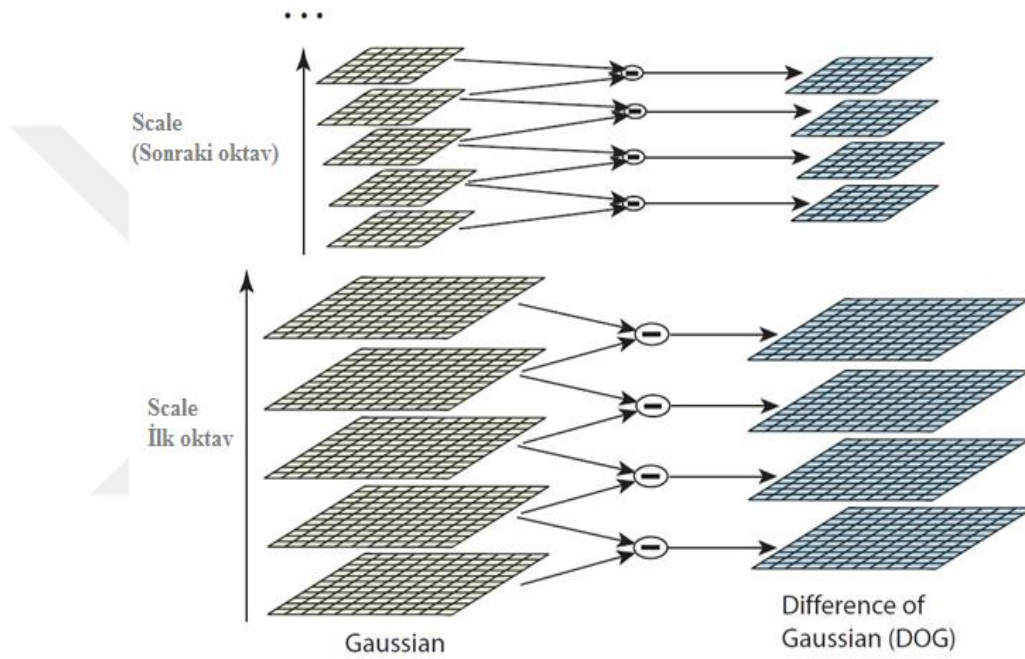
$$L(x, y, \sigma) = G(x, y, \sigma) \times I(x, y) \quad (2.13)$$

Laplacian of Gaussian Yakınlaştırması: Görüntü bulanıklaştırıldıktan sonra kenar ve köşelerin bulunması gerekmektedir. Bulanıklaştırılan görüntülerin ikinci türevi alınarak kenar ve köşeler bulunur. Bu işlem Laplacian of Gaussian işlemi olarak adlandırılır. Görüntüler oktavlara ayrıldığından, bütün grup üyelerine türev işlemi uygulamak hesapsal maliyeti artırır. Dolayısıyla türev işlemi yerine bu işleme denk kabul edilen DoG (Difference of Gaussian) işlemi uygulanır.

DoG işlemi basit bir çıkarma işlemi ile yapılarak, $D(x, y, \sigma)$ şeklinde ifade edilir. DoG fonksiyonu biri diğerine göre k kadar orantılı büyüklüğe sahip Gauss ile yumuşatılmış iki görüntü arasındaki farktır.

$$D(x, y, \sigma) = L(x, y, k\sigma) - L(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) \times I(x, y) \quad (2.14)$$

DoG işlemindeki octave gruplarının çıkarılması Şekil 2.9’da verilmiştir.

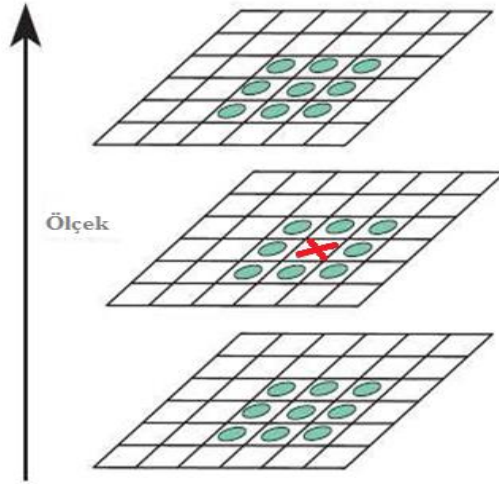


Şekil 2.9. DoG işlemi [19]

Anahtar Noktaların Bulunması: Anahtar noktalar görüntüyü tanımlayan ayırt edici noktalardır. Belirli bir boyuttaki bir alanın görüntü üzerinde gezdirilmesiyle elde edilir. Aşağıdaki adımlar izlenerek anahtar noktalar tespit edilir.

- 1) X noktası octave grubu içerisinde bulunan bütün görüntüler üzerindeki 26 komşu piksel ile karşılaştırılır.
- 2) X noktası bu 26 piksel içerisinde en büyük veya en küçük değer ise anahtar nokta olarak kabul edilir.
- 3) Yukarıdaki kontrol işlemleri bütün pikseller için ayrı ayrı yapılır.

Yukarıdaki işlem adımlarının çalışmasını açıklayan görsel Şekil 2.10’daki gibidir.



Şekil 2.10. Anahtar nokta tespiti [19]

Anahtar Noktaların Elenmesi: Anahtar noktaların bulunması işleminde kararlı anahtar noktaların yanında kararsız noktalar da tespit edilir. Bu kısımda gürültüden daha çok etkilenen ve kenarlarda bulunan zayıf noktalar bulunarak elenir. Kararsız noktalar düşük zıtlığa sahiptir. Dolayısıyla DoG fonksiyonun anahtar nokta referans alınarak ikinci dereceden Taylor serisi açılımı gerçekleştirilir.

$$D(x) = D + \frac{\partial D^T}{\partial x} + \frac{1}{2} x^T \frac{\partial^2 D}{\partial x^2} x \quad (2.15)$$

Uç noktaların konumunu içeren $\hat{x}$, $D(x)$ fonksiyonunun türevi alınıp sıfıra eşitlenerek bulunur.

$$\hat{x} = - \frac{\partial^2 D^{-1}}{\partial x^2} \frac{\partial D}{\partial x} \quad (2.16)$$

Anahtar noktaların konumu hesaplandıktan sonra düşük zıtlığa sahip noktalar eşik değerine göre elenir.

$$D(\hat{x}) = D + \frac{1}{2} \frac{\partial D^T}{\partial x} \hat{x} \quad (2.17)$$

Denklem 2.17'deki $D(\hat{x})$ değeri, belirlenen eşik değerinden büyükse elenir. Geriye kalan noktaların zıtlığı yüksektir. Ayrıca gürültüden daha az etkilenirler.

Kenarların Elenmesi: Kenarlar üzerinde düşük zıtlığa sahip anahtar noktalar bir önceki adımda elimine edilmiştir. Etkin anahtar noktaların tespiti için, kenarlar üzerinde bulunan anahtar noktaların tespit edilip elenmesi gerekmektedir. Bu işlemi SIFT algoritması Harris Corner kenar çıkarma detektörünü kullanarak gerçekleştirir [19].

Gradyan, anahtar nokta etrafındaki değişimin en büyük değeridir. Bir anahtar nokta incelenirken birbirine dik olan iki gradyan dikkate alınır. Anahtar noktanın köşe olup olmadığı 2x2 Hessian matrisi kullanılarak belirlenir.

$$H = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix} \quad (2.18)$$

α en büyük gradyan değerine sahip öz değerdir. β ise en küçük değere sahip öz değerdir. Bu öz değerlerin toplamı $Tr(H)$, çarpımı $Det(H)$ şeklinde ifade edilir.

$$Tr(H) = D_{xx} + D_{yy} = \alpha + \beta \quad (2.19)$$

Denklem 2.19'daki öz değerler hesaplandığında $Det(H)$ negatifse nokta elenir ve anahtar nokta olarak kabul edilmez. α öz değeri β öz değerinin r katıdır. Bu parametrelere göre Denklem 2.20'yi sağlayan noktalar, anahtar nokta olarak kabul edilir. Sağlamayan noktalar ise elimine edilir.

$$\frac{Tr(H)^2}{Det(H)} < \frac{(r+1)^2}{r} \quad (2.20)$$

Anahtar Noktalara Yönelim Atanması: SIFT algoritması bu aşamada anahtar noktalara, 2 boyutlu eksen etrafındaki dönmelere karşı dayanıklılık kazandırır. Her anahtar nokta için en az bir tane döngüsel atama işlemi yapılmaktadır. Gauss filtresinden geçirilmiş olan görüntüde, $L(x,y,\sigma)$ kullanılarak gradyan büyüklüğü $m(x,y)$ ve açısı $\theta(x,y)$ hesaplanır.

$$m(x,y) = \sqrt{(L(x,y+1) - L(x,y-1))^2 + (L(x+1,y) - L(x-1,y))^2} \quad (2.21)$$

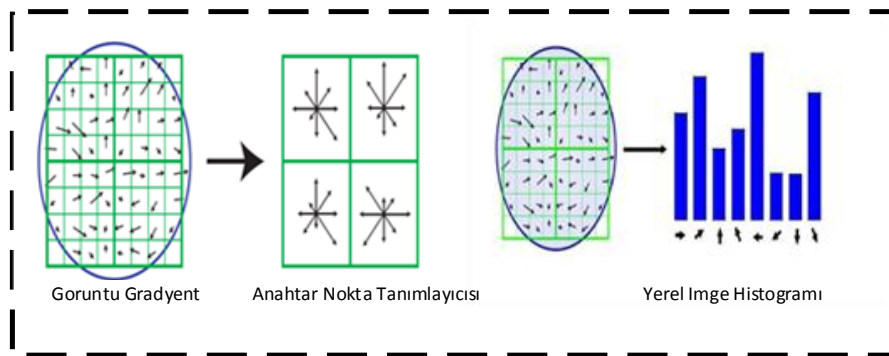
$$\theta(x,y) = \tan^{-1}\left(\frac{L(x,y+1) - L(x,y-1)}{L(x+1,y) - L(x-1,y)}\right) \quad (2.22)$$

Anahtar bölge etrafındaki bütün pikseller için m ve θ değerleri hesaplanarak histogram oluşturulur. Bu histogramda, 360 derecelik gradyan yönelimleri 36 sütuna bölünür ve her birine 10 derece aralığı verilir. Bu aşamaya kadar anahtar nokta bulunma işlemi ve gradyan yönelimlerinin belirlenmesi işlemleri gerçekleştirilmiştir. Böylece konum, ölçek ve yönelimden bağımsız özellik vektörleri elde edilmiştir. Her anahtar nokta için bir tanımlayıcı vektör (descriptor vector) bulunması gerekmektedir.

Yerel İmge Tanımlayıcı: Bu aşamada diğer değişimlere parlaklık değişimleri, 3D bakış açısına karşı dayanıklılık kazandırılır. Verilen adımlar takip edilerek yerel imge tanımlayıcıları hesaplanır.

- 1) Anahtar nokta etrafında 16x16 boyutunda bölgeler oluşturulur. Bu bölgeler 4x4 lük parçalara ayrılır. Bu şekilde 16 adet parça elde edilir.
- 2) Her bir parça içerisinde, gradyan yönelimleri 45 derecelik 8 farklı açı aralığına etiketlenmiş histogramlar oluşturulur. 8 açı aralığına etiketlenmiş 4x4=16 histogramlı özellik vektörünün, 128 elemanı bulunmaktadır.
- 3) Elde edilen histogram, normalize işlemine tabi tutulur.
- 4) Lineer olmayan aydınlatma etkilerini azaltmak için 0,2'lik eşikleme uygulanır ve vektör tekrar normalize işlemine tabi tutulur.

Yukarıdaki işlemlerin akış şeması Şekil 2.11'de verilmiştir.



Şekil 2.11. Yerel imge tanımlayıcısı akış şeması [19]

2.2.2.2. SIFT Tabanlı Görsel Kelime Çantasının Oluşturulması

SIFT algoritması görüntüyü tanımlayan anahtar noktalar üretir. Bu anahtar noktaların sınıflandırma algoritmalarında kullanılabilmesi için eğitim kümesinin oluşturulması gerekmektedir. Anahtar noktalar, görsel kelimelere dönüştürülerek sınıflandırma için uygun hale getirilir. Eğitim kümesi aşağıdaki işlemlere göre oluşturulur.

- 1) SIFT detektörü kullanılarak anahtar noktaların algılanması ve tanımlanması
- 2) Eğitim kümesini oluşturan her bir görüntü için özellik nokta histogramının oluşturulması
- 3) Her bir görüntü için oluşturulan histogramların, birleştirilerek kelime çantası histogramının oluşturulması
- 4) Kelime çantası histogramının, k-Ortalama kelime gruplarının oluşturulması
- 5) Kelime gruplarında oluşan eğitim kümesinin standardizasyonu
- 6) Sınıflandırılmamış görüntü özelliklerinin, standardizasyonu
- 7) Oluşturulan mesafe özelliklerinin sınıflandırma algoritması üzerinde test edilmesi

Kelime çantası (Bag of Words-BOW) görüntüleri oluşturan görsel kelimelerden oluşur. SIFT tabanlı eğitim kümesi, her bir görüntünün kelimelere olan ilişkisiyle elde edilir. SIFT tabanlı özellik çıkarım işleminin sözde kodu Tablo 2.5’de verilmiştir.

Tablo 2.5. SIFT tabanlı eğitim kümesi sözde kodu

```
1: INPUT; // Görüntü,nokta sayısı ve çapın girilmesi
2: image_paths <- görüntüYolu
3: FUNCTION featureExtraction(image_paths):
4:   fea_det <- cv2.FeatureDetector_create("SIFT") //Özellik detektörü tanımlama
5:   des_ext <- cv2.DescriptorExtractor_create("SIFT") //Anahtar nokta tanımlama
6:   destination_list <- [] //Görüntü listesinin tanımlanması
7:   for image_path <- to(image_paths) do //Görüntü yolundaki bütün görüntüleri hedef listesine ekleme
8:     im <- cv2.imread(image_path)
9:     kpts <- fea_det.detect(im)
10:    kpts, des <- des_ext.compute(im, kpts)
11:    destination_list.append((image_path, des))
12:   endfor
13:   // Özellik tanımlayıcılarının liste haline getirilmesi
14:   descriptors <- des_list[0][1]
15:   for image_path <- to(descriptor in des_list[0:]) do
16:     descriptors <- np.vstack((descriptors, descriptor))
17:   endfor
18:   k <- 10 // K-Means ile görüntüleri temsil eden görsel kelime gruplarının çıkarılması
19:   voc, variance <- kmeans(descriptors, k, 1)
```

```
20: im_features <- np.zeros((len(image_paths), k), "float32")
21: for i <- to (xrange(len(image_paths))) do
22:   words, distance <- vq(des_list[i][1],voc) //Görüntü özelliklerinin kelime gruplarına uzaklığının hesaplanması
23:   for w <- to (words) do
24:     im_features[i][w] += 1
25:   endfor
26: endfor
27: stdSlr <- StandardScaler().fit(im_features) // Görüntüyü temsil eden mesafelerin standardizasyon işlemi
28: im_features <- stdSlr.transform(im_features)
```

2.3. Bölüm Değerlendirmesi

Özellik çıkarım işlemi bütün görüntü sınıflandırma işlemlerinin temelini oluşturmaktadır. Bu bölümde LBP ve çoklu özellik çıkarım işlemleri kullanılarak eğitim kümesi oluşturulmuştur. Eğitim kümesi, görüntü kümesindeki her bir görüntünün ayrı ayrı LBP, Histogram ve SIFT özelliklerinden oluşmaktadır. LBP, Histogram ve SIFT özellikleri ayrı ayrı çalışan bağımsız uygulamalardan elde edilir. Sistemin alt birimlerinin bağımsız uygulamalar halinde geliştirilmesi, kontrol ve geliştirme işlemlerini verimli kılar. Özellik çıkarım işlemi, Grafıksel Arayüz (Graphical User Interface-GUI) üzerinde kullanılmak üzere, geliştirilen menüler aracılığıyla gerçekleştirilir. Bu menülerde mermer görüntülerinin dosya yolu GUI üzerinden seçilerek özellik çıkarım işlemleri başlatılarak Weka tabanlı dosya biçimine dönüştürülür. Eğitim setinin Weka tabanlı dosya biçimine dönüştürülmesinin sebebi, sınıflandırıcılar eğitilirken okuma fonksiyonları aracılığıyla doğrudan erişilebilmesidir.

3. SINIFLANDIRMA İŞLEMİ

Özellik çıkarım bölümünde elde edilen özelliklerin, sınıflandırılması için “*Uç Öğrenme Makinesi*” (*Extreme Learning Machine-ELM*) kullanılmıştır. ELM’nin sınıflandırma başarımını ölçmek için toplu sınıflandırıcı kullanılmıştır. Karar Ağacı, Yapay Sinir Ağı (YSA) ve Destek vektör makinesi (DVM) toplu sınıflandırıcıyı oluşturmaktadır.

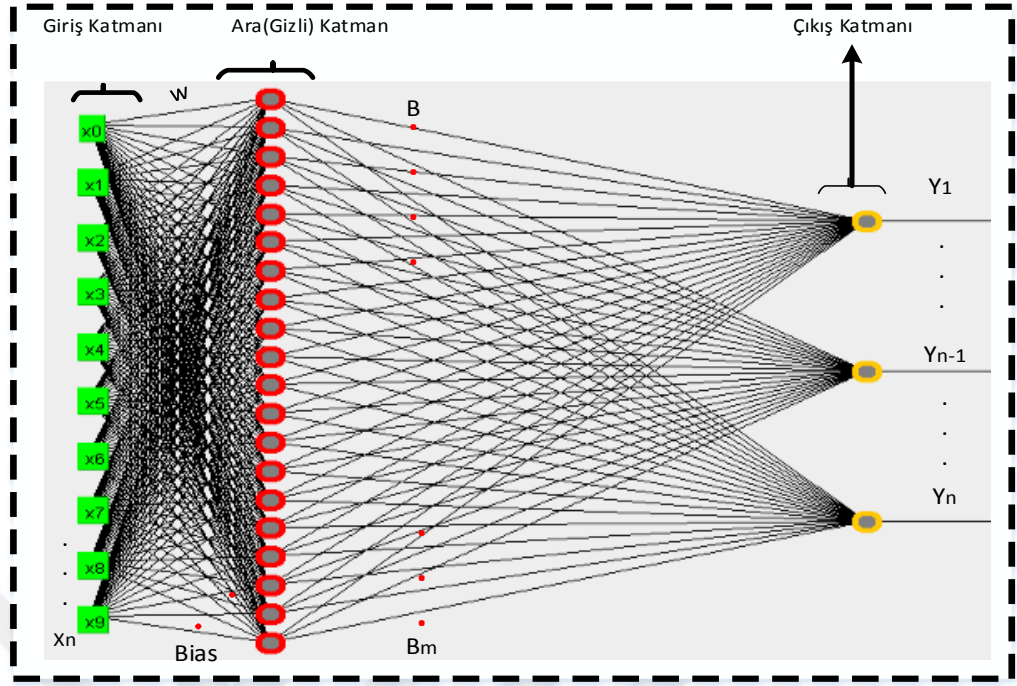
3.1. Uç Öğrenme Makinesi

ELM tek gizli katmanlı ileri beslemeli bir yapay sinir ağıdır. Bu çalışmada kullanılan ELM, test edilen görüntünün özelliklerini kullanabilen, ileri beslemeli statik öğrenmeli bir yapı içerir. Geliştirilen GUI aracılığıyla test edilen mermer görüntüsünün özellikleri ve sınıfı birleştirilerek eğitim kümesine eklenebilir. ELM her test işleminde güncellenmiş eğitim setini kullandığından, daha gelişmiş bir sınıflandırma yapabilmektedir. ELM sınıflandırıcısı tek çevrimde öğrenme işlemini gerçekleştirebildiğinden, YSA’ya göre hızlıdır [34].

ELM yapısı bakımından tek gizli katmana sahip yapay sinir ağına benzer. Ancak YSA’ya göre daha az hafızaya gereksinimi duyarken daha hızlı öğrenme işlemini gerçekleştirir. Sistemin eğitilmesi için YSA’nın aksine öğrenme oranı ve momentum katsayısına ihtiyaç duymadan, daha az parametreyle çalışır. Bu durum geliştiriciler için kolaylık sağlar.

3.1.1. ELM Modeli

ELM yapısı bakımından yapay sinir ağına benzese de çalışma ilkesi farklılık gösterir. ELM’de giriş özellikleri ile ara katman arasındaki (w) ağırlık değeri rasgele atanır; ayrıca ilerleyen aşamada güncellenmez. Ara katman ile çıkış katmanı arasındaki ağırlık değeri (β) analitik olarak hesaplanır.



Şekil 3.1. Tek katmanlı ileri beslemeli YSA yapısı (ELM Modeli)

Şekil 3.1’de tek katmanlı YSA mimarisi verilmiştir. Ara katmandaki hücre sayısı kullanıcının modeline göre değiştirilebilir.

3.1.2. ELM Sınıflandırıcısı

YSA’ların aksine, ELM’de giriş ağırlıkları rastgele üretilirken, çıkış ağırlıkları analitik yöntemler ile hesaplanır. Bu yöntem ile eğitim süreci YSA’ya göre hızlandırılmış olur. Ayrıca YSA’ya göre daha iyi genelleme yapmaktadır. Şekil 3.1’de tek katmanlı YSA modeline göre giriş ve çıkış değerleri olan x_i ve y_i , denklem 3.1 ve denklem 3.2’de verilmiştir

$$x_i = [x_1, x_2, \dots, x_N]^T \in R^M \quad (3.1)$$

$$y_i = [y_1, y_2, \dots, y_N]^T \in R^M \quad (3.2)$$

Şekil 3.1’de verilen N tane gizli hücreye sahip ve $g(x)$ aktivasyon fonksiyonunu kullanan YSA, matematiksel olarak denklem 3.3’teki gibi ifade edilir.

$$\sum_{i=1}^N \beta_i g(w_i \cdot x_j + b_i) = O_j, \quad j = 1, \dots, M \quad (3.3)$$

Denklem 3.3'te $w_i = [\omega_1, \omega_2, \dots, \omega_N]^T$ değeri i 'ninci ara hücre ile giriş hücreleri arasındaki ağırlık vektörüdür. $\beta_i = [\beta_1, \beta_2, \dots, \beta_N]^T$ ise i 'ninci ara hücre ile ve çıkış hücresi arasındaki ağırlık vektörüdür. " b_i ", i 'ninci ara hücrenin eşik değeridir. ELM ağının istenen çıkışı $o_j = [o_1, o_2, \dots, o_j]^T$ şeklinde ifade edilir. " $w_i \cdot x_j$ " değeri " w_i " ve " x_j " nin içsel çarpımıdır. N tane gizli sinir hücresine ve $g(x)$ aktivasyon fonksiyonuna sahip standart Tek Katmanlı YSA'nın ağ hatası sıfıra yaklaşabilir. Dolayısıyla ağın verdiği çıkış ile istenen çıkış arasındaki ilişki $\sum_{j=1}^N \|o_j - y_j\| = 0$ şeklinde ifade edilir. β_i, w_i ve b_i değerleri kullanılarak ELM'nin ürettiği çıkış değeri denklem 3.4'teki gibi belirlenir.

$$\sum_{i=1}^N \beta_i g(w_i \cdot x_j + b_i) = y_j, \quad j = 1, \dots, M \quad (3.4)$$

Denklem 3.4'te verilen N tane eşitlik, denklem 3.5'deki gibi kısaltılabilir:

$$Y = H \cdot \beta \quad (3.5)$$

Denlem 3.5'teki H matrisi denklem 3.6'da verildiği gibidir.

$$H = \begin{bmatrix} g(w_1 \cdot x_1 + b_1) & \dots & g(w_N \cdot x_1 + b_N) \\ \vdots & \dots & \vdots \\ g(w_1 \cdot x_M + b_1) & \dots & g(w_N \cdot x_M + b_N) \end{bmatrix}_{N \times M} \quad (3.6)$$

$$\beta = \begin{bmatrix} \beta_1 \\ \vdots \\ \beta_M \end{bmatrix}_{M \times 1} ; Y = \begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix}_{N \times 1} \quad (3.7)$$

Denklem 3.7'de sırasıyla çıkış katmanı ile ara katman arasındaki ağırlıklar ve çıkış vektörü verilmiştir. w_i giriş katmanı ağırlıkları ve b_i ara katman eşik değerleri rastgele seçilir ve H analitik olarak hesaplanır. Geleneksel Tek Katmanlı YSA'da ağ eğitilirken, ELM sınıflandırıcısı denklem 3.5'in çözülmesiyle tamamlanır. Çıkış ağırlıkları olan β değerleri denklem 3.8'deki gibi hesaplanır.

$$\beta = H^+ Y \quad (3.8)$$

β değerleri, denklem 3.8'den elde edilir. Denklem 3.8'deki H^+ , matrisi H çıkış matrisinin genelleştirilmiş tersidir ve “Moore-Penrose matrisi” olarak ifade edilir.

3.1.3. Parametrelerin Belirlenmesi

ELM sınıflandırma fonksiyonunu üretmeden önce sistemde gerekli değerlerin tanımlı olması gerekmektedir. Bu tez çalışmasında, kullanıcı ELM'nin girişlerini, geliştirilen GUI üzerinden değiştirebilmektedir. Bu durum farklı mermer gruplarında en iyi sonucu alabilmek için kullanıcıya test ortamı sunar. GUI üzerinde, ELM ve karşılaştırma amaçlı kullanılan toplu sınıflandırıcının giriş değerlerini ayarlanabilir.

Aktivasyon fonksiyonu, ara katman ve çıkış katmanındaki değerin çıkışını belirleyen fonksiyondur. Bu tez çalışmasındaki ELM sınıflandırıcısında, sigmoid, sinüs ve radbas fonksiyonları kullanılmıştır. İstenilen çıkış verilerine göre aktivasyon fonksiyonun seçimi ELM'nin performansını etkiler.

ELM'nin giriş değerlerinden biri de hücre sayısıdır. Hücre sayısının fazla olması ELM'nin ağı daha iyi öğrenmesini sağlar. Ancak belirli bir hücre sayısından sonra ELM maksimum öğrenme düzeyine çıkmaktadır. Dolayısıyla ELM işlem süresini düşürmek için en uygun hücre sayısının belirlenmesi gerekmektedir. ELM'de hücre sayısını belirlemek için kesin kurallar yoktur. Bu uygulamada oluşturulan GUI aynı zamanda görüntü sınıflandırma alanında deneysel test ortamı sunmaktadır. GUI aracılığıyla hücre sayısının başarı ve zamanlama performansı izlenerek en uygun değerler belirlenir. Uygulama üzerindeki katsayı atama işleminin sözde kodu Tablo 3.1'deki verilmiştir.

Tablo 3.1. ELM katsayı atama sözde kodu

```
1: INPUT; //Eğitim kümesi ve test verisi
2: TestImage <- testverisi
3: Train <- egitimseti
4: FUNCTION;
5:   ELM elm <- new ELM();
6:   setActiveFunction <- aktivasyon fonksiyonu
7:   setNumberOfHiddenNeurons <- noronSayısı
8:   setNumDecimalPlaces <- hassaiyet
9:   // ELM Eğitim ve Test İşlemi
10:  elm <- Train
11:  className <- elm(TestImage)
12: END FUNCTION
```

3.2. Toplu Sınıflandırıcı

Karar Ağacı, Yapay Sinir Ağı (YSA) ve Destek vektör makinesi (DVM) toplu sınıflandırıcıyı oluşturmaktadır.

3.2.1. Karar Ağacı Yöntemiyle Sınıflandırma

Karar ağacı, belirsizliğin en düşük olduğu kuralları tespit ederek, eğitim kümesinin sınıflandırılmasını sağlar. Hesapsal işlem maliyeti düşük olduğundan birçok alanda kullanılmaktadır [40].

Bu çalışmadaki uygulamada kullanılan C4.5 tabanlı karar ağacı, statik öğrenmeli bir karar ağacıdır. Geliştirilen Grafiksel Arayüz (Graphical User Interface-GUI) sayesinde test edilen görüntünün özellikleri, eğitim kümesine eklenebildiğinden, karar ağacı kendi kendini geliştirebilmektedir.

Karar ağaçları kural oluşturma algoritması olarak böl ve yönet yaklaşımını kullanır. Bu yaklaşımda bölme işlemi, problemin alt problemlere bölünerek basit yapılar haline dönüştürülmesini ve bu basit yapıların çözümünü sağlar. Yönetme işlemi ise alt problemlerin çözümü birleştirilerek problemin bütün çözümünün elde edilmesi temeline dayanır. Karar ağacında kullanılan eğitim setindeki bütün veriler sınıfların ve özelliklerin dağılımından bağımsızdır [40]. Bu nedenle karar ağacının çalışması için, verilerin normalizasyona tabi tutulması zorunlu değildir. Karar ağaçlarında düğümleri oluşturan kural çıkarımıyla ilgili birçok çalışma yapılmıştır. Bu çalışmada en yaygın kullanılan ID3, C4.5 ve Sınıflandırma ve Regresyon Ağacı (Classification and Regression Tree-CART) tabanlı karar ağacı algoritmaları incelenmiştir. Geliştirilen uygulama üzerinde, ELM ile başarı performansının karşılaştırılması için, C4.5 tabanlı karar ağacı oluşturulmuştur.

3.2.1.1. Karar Ağacı Algoritmaları ve Seçimi

Karar ağaçları çalışma mimarisi bakımından eğitim kümesindeki özelliklerden kurallar oluşturarak sınıflandırma işlemini gerçekleştirir. Kural çıkarım yöntemi bakımından Karar ağacı üçe ayrılır [39].

1. Entropi tabanlı
 - ID3
 - C4.5
2. Cart tabanlı
 - Gini
 - Twoing
3. Bellek tabanlı sınıflandırma
 - Knn

Sayısal değerler üzerinden çalışma, verimsiz dalları budama ve normalizasyon özelliklerinden dolayı C4.5 tabanlı karar ağacı bu çalışmada kullanılmıştır. Kural çıkarım yapılarına göre ID3, C4.5 ve GINI tabanlı Karar ağaçları aşağıdaki gibi açıklanmıştır.

ID3: Bu algoritmada entropi, nominal değerlerin olasılıklarından hesaplanır. Her örneğin, sınıfının belli olduğu eğitim setine, denetimli (Supervised) eğitim seti denilmektedir. Denetimli ve nominal değerlerden oluşan eğitim seti, ID3 algoritması ile sınıflandırılır. ID3 algoritmasında düğümü oluşturan kurallar sırasıyla aşağıdaki adımlar takip edilerek hesaplanır [11].

- 1) Bir veri kümesindeki belirsizliğin (rassallık) ölçüsü entropidir. Denetimli eğitim kümesindeki sınıf kümesi T 'dir. T kümesini oluşturan her bir sınıf ise C_i olsun. T sınıfını oluşturan C_i sınıflarının P_t olasılık değeri aşağıdaki gibidir.

$$P_t = \left(\frac{|C_1|}{|T|}, \frac{|C_2|}{|T|}, \frac{|C_3|}{|T|}, \dots, \frac{|C_k|}{|T|} \right) \quad (3.12)$$

- 2) $H(T)$, T sınıfı için entropi değeridir.

$$H(T) = - \sum_{i=1}^n p_i \log_2 p_i \quad (3.13)$$

- 3) Sınıflara göre eğitim kümesinin genel entropisi denklem 3.14'teki gibi özellik değerlerinin sınıflara göre entropileri ise denklem 3.15'teki gibi hesaplanır.

$$H(X_k) = - \sum_{i=1}^n \frac{|T_i|}{|X_k|} \log \frac{|T_i|}{|X_k|} \quad (3.14) \quad H(X, T) = \sum_{k=1}^n \frac{|X_k|}{|X|} H(X_k) \quad (3.15)$$

- 4) Kazanç genel entropiden sınıfa ait entropi çıkarılarak elde edilir. Kazanç düğümü oluşturacak özelliği belirler. Kazancı büyük olan özellik karar ağacına düğüm olarak eklenir.

$$Kazanç(X, T) = H(T) - H(X, T) \quad (3.16)$$

- 5) Yukarıdaki adımlar bütün özelliklere iteratif bir şekilde uygulanır. İterasyon saflık ölçütü 1 veya 0 olana kadar devam eder.

C4.5: Bu algoritma denetimli sayısal verilere sahip, eğitim kümesi üzerinden işlem görmektedir. C4.5 algoritması sayısal verileri sınıflandırılarak nominal değerlere dönüştürdükten sonra ID3 yöntemine göre çalışmasını tamamlar [12]. Aşağıdaki adımlar takip edilerek hesaplanmaktadır.

- 1) Eşik değeri bilgi kazancını belirlemek için girilir.
- 2) Herhangi bir özelliğin sayısal değerlerini V değeri belirtmektedir. Eşik değeri t_i şeklinde belirtilir. V değerleri sıralanarak ortanca elaman bulunur ve i değeri ortanca elamanın indisidir.

$$t_i = \frac{V_i + V_{i+1}}{2} \quad (3.17)$$

- 3) Sayısal değerler eşik değeri olan t_i ye göre, kategorilendirilerek nominal değerlere dönüştürülür. Bu aşamadan sonra kategorilendirilmiş eğitim kümesine, ID3 algoritması uygulanarak karar ağacı oluşturulur.

GINI: Bölümleme yapılabilmesi için özellik değerlerini ikiye ayırır. Gini algoritması aşağıdaki adımlar takip edilerek hesaplanmaktadır. [13].

- 1) Sağ ve sol değerler her bölümleme için ayrı ayrı hesaplanır.

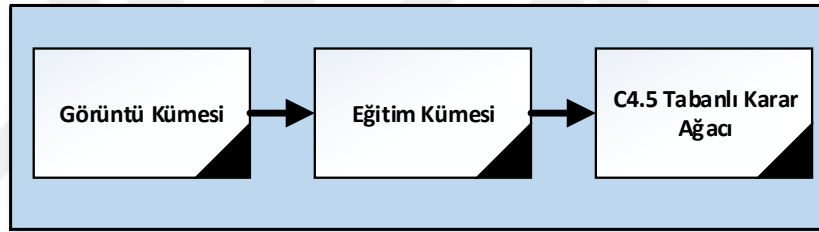
$$Ginisol = 1 - \sum_{i=1}^k \left(\frac{|T_{sinf_i}|}{|B_{sol}|} \right)^2 \quad (3.18) \quad Gini sağ = 1 - \sum_{i=1}^k \left(\frac{|T_{sinf_i}|}{|B_{sağ}|} \right)^2 \quad (3.19)$$

- 2) Sağ ve sol bölümdeki ayrı ayrı sınıf değerini $Tsınıf_i$, bölümdeki değer sayısı ise $|B_{sol}|$ ve $|B_{sağ}|$ ile ifade edilir.
- 3) En küçük GINI değerine sahip özellik düğümün kuralını oluşturur.

$$Gini_j = \frac{1}{n} (|B_{sol}| Gini_{sol} + |B_{sağ}| Gini_{sağ}) \quad (3.20)$$

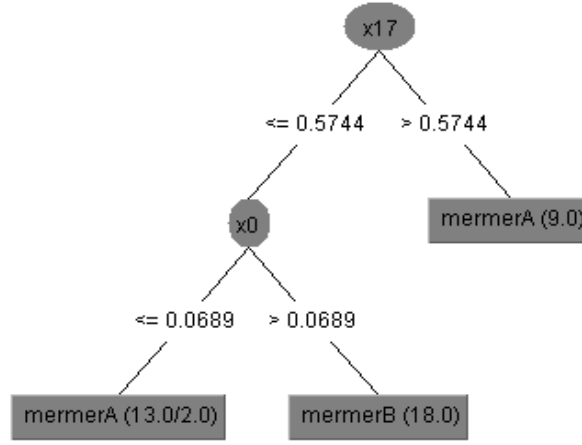
3.2.1.2. C4.5 Tabanlı Karar Ağacı Oluşturma

C4.5 tabanlı karar ağacı oluşturma işlemi, Şekil 1.2'deki önerilen mermer sınıflandırma işleminin, 3.2. adımındaki toplu sınıflandırıcının içerisinde yer almaktadır. Karar ağacı denetimli eğitim kümesinde çalışabilen, bir sınıflandırma yöntemidir. Karar ağacının oluşturma sürecini gösteren akış şeması Şekil 3.2'de verilmiştir.

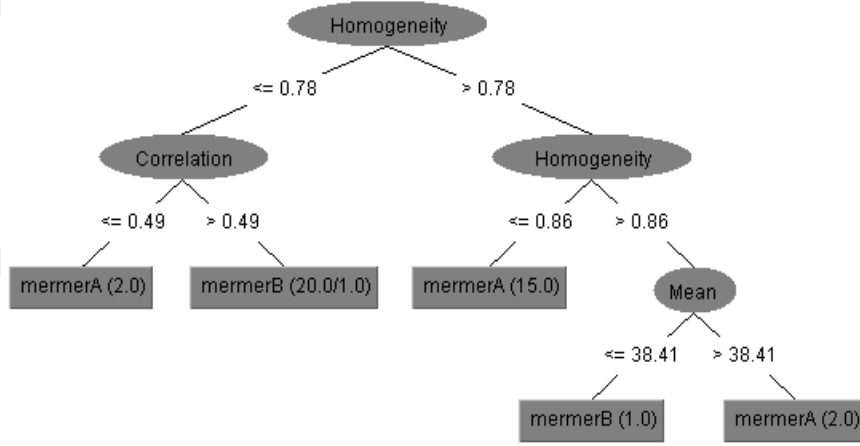


Şekil 3.2. Karar ağacı oluşturma işlemi ön aşamaları

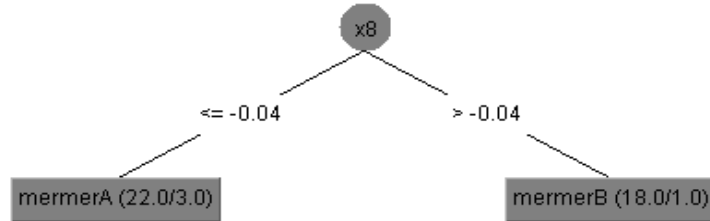
Eğitim kümeleri kullanılarak Yerel İkili Örüntü (Local Binary Pattern-LBP), Histogram ve Ölçekten Bağımsız Özellik Dönüşümü (Scale Invariant Feature Transform-SIFT) tabanlı eğitim kümesinin C4.5 tabanlı karar ağaçları oluşturulmuştur. Karar ağacı eğitilirken eğitim verisinin tamamı kullanılmıştır. Eğitim başarısı hesaplanırken eğitim kümesinin tamamı eğitildiğinden test başarısı şeklinde yorumlanmamalıdır. Karar ağaçlarının test başarısı çapraz doğrulama yöntemine göre deneysel sonuçlar bölümünde verilmiştir. LBP ve Histogram tabanlı karar ağaçlarının SIFT tabanlı karar ağacına göre daha yüksek başarıma sahip olduğu görülmektedir. Şekil 3.3, Şekil 3.4 ve Şekil 3.5'te sırasıyla LBP, Histogram ve SIFT tabanlı karar ağaçları verilmiştir. Bu karar ağaçlarının görselleri GUI tabanlı yazılımın bir çıktısıdır.



Şekil 3.3. LBP eğitimli C4.5 tabanlı karar ağacı



Şekil 3.4. Histogram eğitimli C4.5 tabanlı karar ağacı



Şekil 3.5. SIFT eğitimli C4.5 tabanlı karar ağacı

C4.5 tabanlı karar ağacı oluşturma işlemi ve bu karar ağacı üzerinden test görüntüsünün sınıfını belirleme işleminin sözde kodu Tablo 3.2’de verildiği gibidir.

Tablo 3.2. Karar ağacı oluşturma işlemi sözde kodu

```
1: INPUT; //Eğitim kümesi ve test verisi
2: TestImage <- testverisi
3: Train <- eğitimseti
4: FUNCTION;
5:   //Karar ağacı ayarları
6:   C45 tree <- new C45();
7:   setNumDecimalPlaces(2) <- hassasiyet
8:   setMinNumObj <- minimumDugumSayısı
9:   setDoNotCheckCapabilities <- datasetKontrol
10:  setConfidenceFactor <- güvenFaktörü
11:  setUnpruned(true) <- budama
12:  // Karar Ağacını Eğitim ve Test İşlemi
13:  Treebuild <- Train
14:  className <- Treebuild(TestImage)
15:  // Karar Ağacı Görselleştirme İşlemi
16:  jf <- JFrame("Karar Ağacı");
17:  jf.setSize <- (500,400);
18:  tv <- TreeVisualizer(Treebuild.graph)
19:  jf.getContentPane <- tv;
20:  jf.setVisible <- true
21:  tv.fitToScreen()
22: END FUNCTION
```

Karar ağacı açgözlü sınıflandırma algoritmasıdır. Bu algoritma her düğümde en fazla sayıda örneği sınıflandırmayı hedefler. Eşik değeri girilmemişse, sınıflandırıcı bütün örnekleri sınıflandıran kadar karar ağacına düğüm ve yaprak eklenmeye çalışır. ‘*Minimum Yaprak Sayısı*’ gibi eşik değerleri girilirse karar ağacının boyutu ve oluşturulma süresi değişebilir. Bu çalışmada karar ağacı işleme süresinin, doğruluğa getireceği etki, uygulama üzerinden izlenerek değiştirilebilir. Toplu sınıflandırıcıdaki algoritmalar paralel çalıştırıldığından en hızlı sonuçlandırma süresi, en yavaş sınıflandırıcıya bağlıdır. Dolayısıyla karar ağacının işlem süresi, diğer sınıflandırıcılara göre daha kısa olacağından sınıflandırma başarısını düşürecek eşik değerleri sınıflandırma başarısını düşürebilir.

3.2.2. Yapay Sinir Ağıyla Sınıflandırma

Özellik çıkarım bölümünde elde edilen özelliklerin, sınıflandırılması için kullanılan yöntemlerden biri de yapay sinir ağlarıdır. Yapay sinir ağları-YSA (Artificial Neural Network) paralel işlem yapabilme yeteneğinden dolayı, eşzamanlı çalışarak karmaşık işlemleri yüksek başarı oranıyla işleyebilir [40]. YSA çalışma yöntemi olarak kendi kendisini eğitebilmektedir. Geliştirilen GUI aracılığıyla test edilen mermer görüntüsünün özellikleri ve sınıfı birleştirilerek

eğitim kümesine eklenebilir. YSA her test işleminde güncellenmiş eğitim setini kullandığından, daha gelişmiş bir sınıflandırma yapabilmektedir.

YSA, insan beyninin öğrenme, çözümüleme, regresyon analizi gibi yeteneklerinin, matematiksel yapıya aktarılmasını amaçlayan sistemlerdir. YSA ile insan beyninin matematiksel olarak modellenmesi amaçlanmıştır. İnsan beyninin temel birimi olan hücrelerin kavramsal bir yapı halinde modellenmesiyle YSA çalışmaları başlamıştır. Bilgisayar teknolojisindeki işlemcilerin hızı, insan beyninden yaklaşık 10^6 kat daha hızlı olmasına rağmen YSA'lar insan beyninin işlevselliğinin çok altındadır. YSA sınıflandırıcısı kendi kendini geliştirebilme, sabit kaynak tüketimi, işlevsellik ve karar hızı bakımından verimli olduğundan birçok alanda kullanılmaktadır [41].

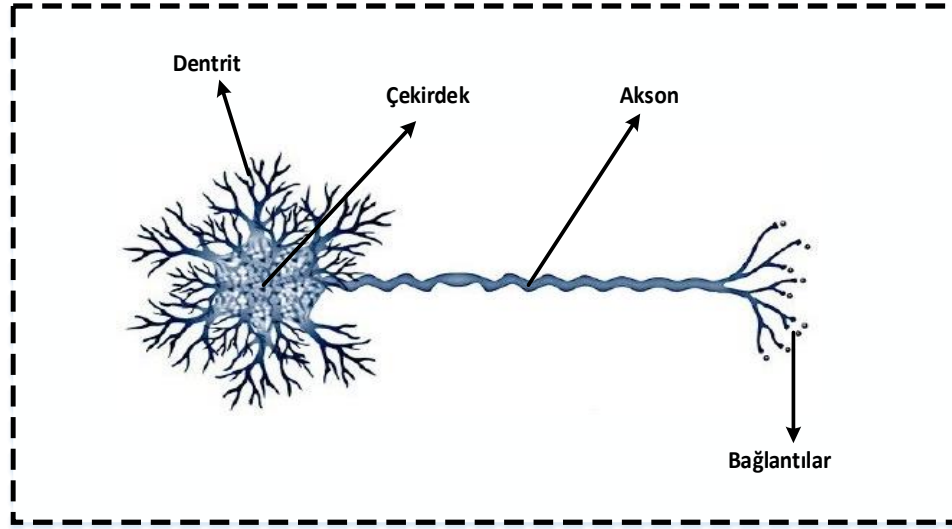
Yapay sinir ağları kendi kendini eğitebilen sistemlerdir. Eğitim işlemi sırasında ağ yapısına göre ayırt edici davranışlar sergileyebilirler. YSA modeli oluşturulmadan önce problemin iyi analiz edilmesi gerekir. Problemin girişi ve çıkışı arasındaki bağlantı iyi analiz edilirse, ağa verilecek parametreler başarı için belirleyici olabilmektedir. YSA'nın avantajları ve dezavantajları aşağıda verilmiştir [41].

3.2.2.1. Yapay Sinir Ağı Modeli

YSA'lar insan beyninin modellenmesi fikrinden ortaya çıktığından, biyolojik sinir ağı yapısının anlaşılması gerekir. Hücreler biyolojik sinir ağının temel bileşenidir. Bir hücre dört alt birimden meydana gelir. Dendrit, çekirdek, akson ve bağlantılar hücrenin alt birimleridir.

- *Dendrit*, duyu organları ve bağlantılı hücrelerden gelen sinyalleri çekirdeğe iletir.
- *Çekirdek*, dendritlerden gelen sinyalleri birleştirilerek aksona iletir.
- *Akson*, çekirdekten gelen sinyalleri işleyerek ilgili bağlantılara gönderir.
- *Bağlantılar*, akson tarafından üretilen sinyalleri diğer hücre veya organlara gönderme işini yürütür. YSA'lar ise bu alt birimler modellenerek geliştirilmiştir.

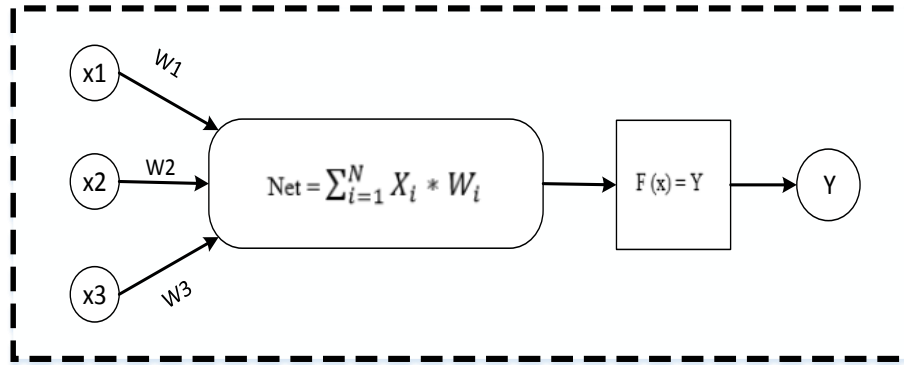
Dört alt birimden oluşan biyolojik sinir yapısı Şekil 3.6'da verildiği gibidir.



Şekil 3.6. Biyolojik sinir hücresi

3.2.2.2. Yapay Sinir Hücresinin Modeli

Yapay sinir hücreleri, biyolojik sinir hücresi referans alınarak geliştirildiğinden benzer alt birimlere sahiptirler. Hücreler aralarında bağ kurarak yapay sinir ağı modelini oluşturur. YSA'lar beş alt birimden oluşur. Bu birimler girişler, ağırlıklar, toplama fonksiyonu (birleştirme fonksiyonu), aktivasyon fonksiyonu ve hücre çıkışı olarak ifade edilir. YSA hücre modeli Şekil 3.7'de verilmiştir.



Şekil 3.7. YSA hücre modeli

Girişler: YSA sınıflandırıcısında hücelere gelen verilerdir. Şekil 3.7'de x_i değerleriyle gösterilmiştir. Bu veriler doğrudan gelebileceği gibi bir hücreden de gelebilir. Girişler, Net fonksiyonunun hesaplanabilmesi için hücrenin çekirdeğine iletilir.

Ağırlıklar: YSA’da hücrelere gelen veriler çekirdeğe iletilmeden önce ağırlıklandırılarak çekirdeğe iletilir. Bu ağırlıklar Şekil 3.14’teki w_i değerleriyle gösterilmiştir. Bu değerler, girişlere ağırlık atayarak giriş değerlerinin çıkışa olan etkisini ayarlar.

Toplama Fonksiyonu (Birleştirme Fonksiyonu): Bu alt birim kendisine gelen girdilerin ağırlıklandırılmış değerlerini toplayarak, her bir hücrenin net girişini hesaplar. Uygulamada toplam işlevinden farklı olarak kullanılan Net fonksiyonları mevcuttur. Ayrıca hücrelerin Net fonksiyonları farklı olabilir. En uygun toplama fonksiyonunu belirlemek için kesin kurallar yoktur. Bu işlem genelde deneme yanılma yoluyla ve tecrübeyle belirlenir. YSA sınıflandırıcısında kullanılan en genel toplama fonksiyonları [41] Tablo 3.3’te verilmiştir.

Tablo 3.3. YSA’da kullanılan en yaygın toplama fonksiyonları

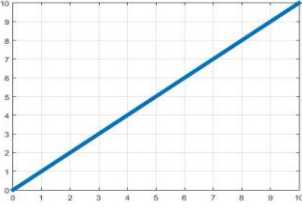
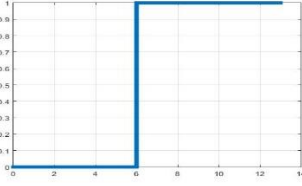
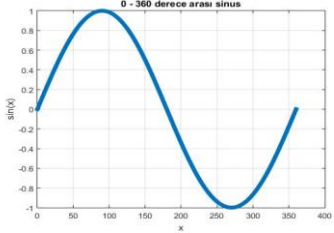
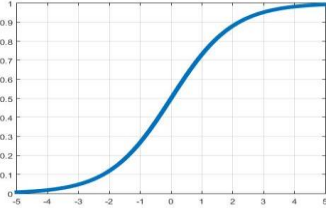
Toplam $Net = \sum_{i=1}^N x_i * w_i$	Girişler ağırlık değerleriyle çarpılır ve bütün değerler toplanarak Net değer hesaplanır.
Çarpım $Net = \prod_{i=1}^N x_i * w_i$	Girişler ağırlık değerleriyle çarpılır ve bütün değerler birbirleriyle çarpılarak Net değer hesaplanır.
En Büyük Değer $Net = \text{Max} (x_i * w_i)$	Giriş ve ağırlıklar çarpılarak en büyük değer seçilir ve Net değeri olarak atanır.
En Küçük Değer $Net = \text{Min} (x_i * w_i)$	Giriş ve ağırlıklar çarpılarak en küçük değer seçilir ve Net değeri olarak atanır.
Seçim $Net = \sum_{i=1}^N \text{Sgn}(x_i * w_i)$	N sayıda giriş ve ağırlıklar çarpılır. Pozitifler ile negatiflerin ayrı ayrı toplam sayısı bulunur. Büyük olan sayı Net değer olarak atanır.
Kümülatif Toplam Değer $Net = \text{Net (eski)} + \sum_{i=1}^N x_i * w_i$	Giriş ve ağırlıklar çarpılarak toplanır ve Net değeri olarak atanır. Bir sonraki Net değeri hesaplanırken giriş ve çarpımların toplamına ilaveten eski net değeriyle de toplanır. Bu işlem kümülatif bir şekilde devam eder.

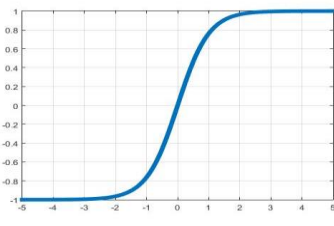
Aktivasyon Fonksiyonu: Bu birim hücrelerin net girişlerini alarak, belirli bir fonksiyonla işler ve hücrenin çıkışını belirler.

Çalışmalarda sıkça tercih edilen bazı aktivasyon fonksiyonları Tablo 3.4’te verilmiştir. Aktivasyon fonksiyonun seçiminde, türevi kolay alınabilir doğrusal olmayan fonksiyonlar tercih edilmelidir. Türev alma işlemi hesapsal maliyetli olduğundan kaynak tüketimi ve zaman

kaybına yol açar. YSA sınıflandırıcısında kullanılan en yaygın aktivasyon fonksiyonu Sigmoid fonksiyondur. Çok katmanlı ve doğrusal olmayan YSA'lar+da Sigmoid fonksiyonun kullanılması başarılı sonuçlar elde edilmesini sağlar. Ancak her ne kadar doğrusallık YSA'nın yapısına aykırı olsa da doğrusal bir eğitim kümesi kullanılmayacağı anlamına gelmez [41]. Bunun gibi çeşitli örneklerde eğitim kümesinin girişi ve çıkışı arasındaki bağ iyi bir şekilde analiz edildikten sonra aktivasyon fonksiyonu seçilmelidir.

Tablo 3.4. YSA'da kullanılan en yaygın aktivasyon fonksiyonları

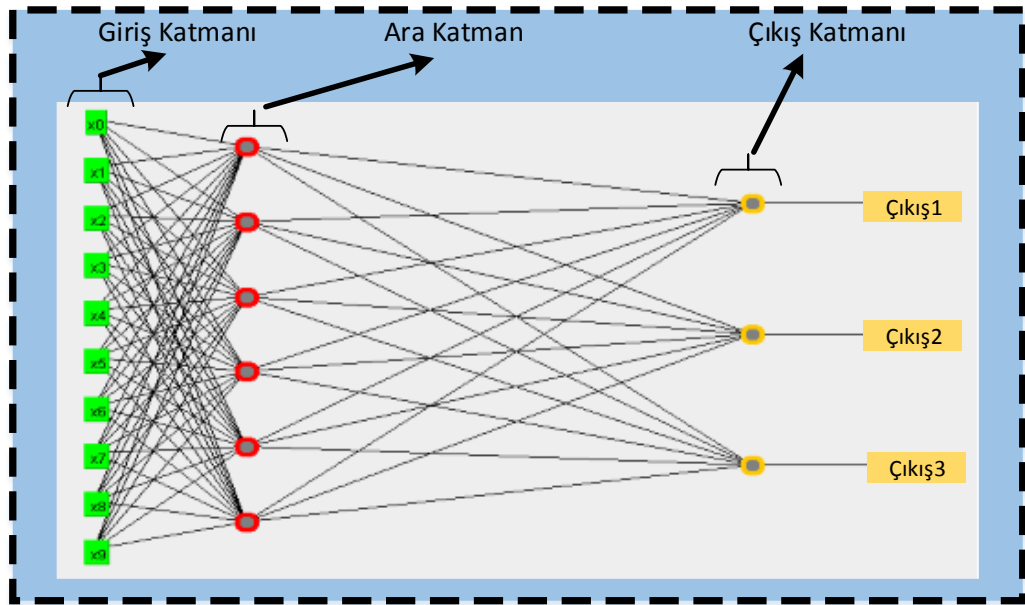
Doğrusal Aktivasyon Fonksiyonu 	$F(Net) = A * Net$ (A sabit değerdir.)	YSA'nın çalışma mantığına uygun olmamasına rağmen doğrusal problemlerin çözümü için kullanılabilir. Toplama işleminden çıkan değer, sabit bir değer ile çarpılarak çıkış elde edilir.
Adım (Step) Aktivasyon Fonksiyonu 	$F(Net) = \begin{cases} 1 & \text{if } Net > \text{Eşik} \\ 0 & \text{if } Net < \text{Eşik} \end{cases}$	Net değeri eşik değeri ile karşılaştırılarak, 0 veya 1 çıkışını verir.
Sinüs Aktivasyon Fonksiyonu 	$F(Net) = \sin(Net)$	Eğitim kümesi sinüs tabanlı bilgiler içeriyorsa bu aktivasyon fonksiyonu kullanılır.
Sigmoid Aktivasyon Fonksiyonu 	$F(Net) = \frac{1}{1 + e^{-Net}}$	Doğrusal bir fonksiyon olmadığından, YSA'nın yapısına uygundur. Sürekli ve türevi alınabilir bir fonksiyondur. 0 ve 1 arası çıkış verir. YSA'da en çok tercih edilen aktivasyon fonksiyonlarının başında gelir.

Tanjant Hiperbolik Aktivasyon Fonk. 	$F(Net) = \frac{e^{Net} + e^{-Net}}{e^{Net} - e^{-Net}}$	Sigmoid fonksiyona benzer. Ancak bu fonksiyonun çıkış değeri -1 ile 1 aralığındadır. Sigmoid fonksiyona göre işlem maliyeti ve kaynak tüketimi fazla olduğundan daha az tercih edilir.
Eşik Değer Fonksiyonu	$F(Net) = \begin{cases} 0 & Net \leq 0 \\ Net & 0 < Net < 1 \\ 1 & Net \geq 1 \end{cases}$	Net değerini 0 ve 1 arasında karşılaştırarak çıkış değerini üretir

Hücre Çıkışı: Aktivasyon fonksiyonunun çıkışı, aynı zamanda hücrenin de çıkışıdır. Bu değer, YSA'nın çıkış değeri olabileceği gibi ileri veya geri beslenerek, ağ içinde de kullanılabilir. Her hücrenin bir veya daha fazla girişi olabilir ancak tek çıkışı vardır. Bir hücrenin çıkışının bağlanacağı ve verileceği hücrelerle ilgili herhangi bir kısıt yoktur.

3.2.2.3. YSA Katman Mimarisi

YSA hücrelerin birbirine bağlanmasıyla giriş, ara ve çıkış katmanlarından oluşur. Bir YSA'nın yapısı Şekil 3.8'de verilmiştir.



Şekil 3.8. Tek gizli katmanlı YSA yapısı

Giriş Katmanı: Eğitim kümesinin her bir elemanının YSA'ya verildiği katmandır. Eğitim kümesindeki özellik sayısı kadar giriş katmanında hücre mevcuttur. Bu katmanda ki değerler genel olarak doğrudan ara katmana iletilir.

Ara Katman: Giriş katmanı verileri doğrudan bu katmandaki hücrelere iletir. Ara katman sayısı ve hücre sayısı değişkendir. Ayrıca bu katmanda bulunan hücre sayıları, giriş ve çıkış sayısından bağımsızdır. Ancak bazı uygulamalarda giriş sayısı ve çıkış sayısı toplamının yarısı kadar hücreye sahip tek bir ara katman oluşturulduğu görülmüştür. Genel olarak ara katman ve hücre sayısı tamamen tasarımcıya bağlıdır. Ara katmanların ve hücrelerin sayısının artması karmaşıklığı ve problemin çözümünü uzatabilir. Ancak karmaşık problemler için YSA'nın boyutunun büyütülmesi gereklidir.

Çıkış Katmanı: Bu katmandaki veriler işlenerek sistemin çıkışı olarak verilir. Ayrıca dinamik ağlarda tekrar ağı verilerle ağı ağırlıklarının güncellenmesi sağlanır.

3.2.2.4. Yapay Sinir Ağı'nın Eğitilmesi

YSA algoritması, insan beyni gibi dışarıdan gelen verileri işleyerek istenilen sonucu elde etmeye çalışır. İnsan beyninde öğrenme zamanla ve tecrübeyle daha fazla gelişir. Gelişen beyin aynı olaylar karşısında vereceği tepkiyi bilerek hareket eder. Farklı olaylar karşısında ise tecrübelerine dayanarak çıkarım yapar. YSA, verilerden çıkışlar elde ettikten sonra, bu çıkışları istenilen sonuç ile karşılaştırır. Eğer sonuçlar istendiği gibi değilse hatayı ağı tamamına yayarak ağırlıkların güncellenmesini sağlar. Bu şekilde hata eşik değerinin altına düşene kadar öğrenme işlemi devam ettirilir. YSA'daki bu işlem, insan beyninin zamanla tecrübe kazanma özelliğinden gelmektedir [41].

YSA'nın ağırlıklarını, hedeflenen çıkışa ulaşana kadar güncellemesi, öğrenme süreci olarak ifade edilir. Genelde eğitim kümesinin %80'i eğitim için %20'si ise test için ayrılır. Eğitim için ayrılan veriler ile öğrenme işlemi gerçekleştirilir. Test kümesiyle de ağı başarısı ölçülür. Sınıflandırma başarısını ölçmek için çapraz doğrulama ve yüzdelik bölme gibi yöntemler kullanılabilir. YSA'nın eğitilme süreci aşağıda verilen dokuz adımda sıralanabilir.

1. Örneklerin Toplanması: YSA'nın ilgili problemi öğrenerek çözebilmesi için eğitilmesi gerekmektedir. Bu eğitim işlemi örüntü özelliklerinin öğrenilmesidir. Bu özellikler ağı sırasıyla tanıtılır. YSA örnekleri öğrendikten sonra, hiç görmediği örnekler karşısında yaptığı çıkarımların doğruluğu ağı başarısını yansıtır. YSA'nın iyi bir şekilde eğitilebilmesi için

eğitim kümesinin düzgün bir biçimde oluşturulması gerekmektedir. Eğitim kümesi sayısal değerlerden oluşmalıdır. Eğer nominal değer içeriyorsa bu özellikler normalize edilmelidir. Doğru elde edilmiş bir eğitim kümesi ağın başarısında en büyük paya sahiptir. Eğitim kümesinin bütün elemanlarının aynı standartlar altında elde edilmiş olması gerekmektedir [40].

2. Ağ Mimarisinin Belirlenmesi: YSA sınıflandırıcısındaki giriş sayısı genellikle eğitim kümesindeki özellik sayısıdır. Ancak bazı durumlarda Temel Bileşen Analizi (Principal Component Analysis-PCA) uygulanarak ağırlıksız özellikler elimine edilebilir. Bu aşamada giriş sayısı, ara katman sayısı, ara katmandaki hücre sayısı ve çıkış sayısı belirlenir.

3. Öğrenme Parametrelerinin Belirlenmesi: Bu işlem adımında, ağa harici olarak verilen öğrenme katsayısı, toplama fonksiyonu, aktivasyon fonksiyonu ve momentum katsayısı belirlenir.

4. Ağırlıkların Başlangıç Değerinin Belirlenmesi: Hücreleri birbirine bağlayan ağırlık değerlerinin başlangıç değerleri ve öğrenmenin sonlanması için çıkıştaki hatanın eşik değeri belirlenir. Bu değerler rastgele atanır. Ağ eğitildikçe bunların yerine uygun değerler güncellenir.

5. Eğitim ve Test Kümesinin Belirlenmesi: Sisteme verilen örneklerin %80 i eğitim için ayrılır. Geriye kalan % 20 lik kısım ise test işlemi için ayrılır. Bu örnekler rastgele seçilmelidir.

6. İleri Hesaplamaların Yapılması: Verilen girişler için ağın çıkışı hesaplanır. Bu işlem bütün örnekler en az bir defa denenene kadar devam eder.

7. Çıkış Hatasının Tespiti: İstenilen sonuç ile ağın verdiği çıkış arasındaki farktır.

8. Ağırlıkların Güncellenmesi: Çıkış hatası eşik değerinden büyükse oluşan hata miktarı ağa geri verilir. Bu işlem boyunca ağırlıklar değiştirilerek hatanın azalması beklenir.

9. Öğrenmenin Bitirilmesi: Ağın çıkışı ile istenilen değer arasındaki fark eşik değerinden az olursa ağ öğrenme işlemini tamamlamış demektir.

YSA yukarıdaki adımları gerçekleştirerek öğrenme işlemini gerçekleştirir. Eğitim kümesindeki bütün değerler sırasıyla ağa tanıtılarak hedeflenen çıkışa ulaşılması beklenir. Hedeflenen değer ile YSA çıkışı arasındaki fark hata olarak tanımlanır. Bu hata kullanılarak ağ ağırlıkları güncellenir. Hata, eşik değerinin çok üzerinde olduğunda ağın öğrenmesi gecikebilir. YSA'nın öğrenmesini hızlandırmak için parametrelerde değişiklikler yapılabilir. Başlangıç değerleri, katman sayısı, katmanlardaki hücre sayısı, eğitim kümesindeki verilerin gösterim şekilleri ve örnek sayısı YSA tarafından değiştirilebilir.

YSA'nın eğitilmesindeki önemli sorunlardan biri de zamandır. YSA'ların öğrenme sürecinde yerel minimumlara takılmaması için momentum katsayısı eklenebilir. Ağırlık

değerlerinin başlangıçta büyük atanması ağın yerel sonuçlara düşmesini sağlar. Küçük olarak atanması ise öğrenme süresinin çok uzun olmasına yol açar. Momentum katsayısı kullanılarak ağırlıklara müdahale edilebilir [40].

YSA'da ağın öğrenme durumu hata grafiği çizilerek görsel bir şekilde izlenebilir. Böylece her çevrimdeki hata bir önceki hatayla karşılaştırılabilir. Hata miktarının uygun görülen bir değerinden sonra ağın yeterince öğrendiği kabul edilir ve eğitim sonlandırılır. Eğitimin sonlandırıldığı andaki ağ ağırlıkları, eğitilmiş ağırlıklar olarak saklanır ve test işleminde bu ağırlıklar kullanılır.

3.2.2.5. Parametrelerin Belirlenmesi

YSA eğitime başlamadan önce öğrenme oranı, momentum katsayısı ve başlangıç ağırlık değerleri atanmalıdır. Bu değerleri atamak için kesin kurallar yoktur. Ancak deneme yanılma ve tecrübelerle göre bazı varsayımlar elde edilmiştir. Ağırlık değerlerinin genelde 0.1- 1.0 aralığında seçilmesiyle başarılı sonuçlar elde edildiği görülmüştür [41].

Öğrenme katsayısı, ağırlıkların değişim miktarının belirlenmesini sağlar. Öğrenme katsayısının büyük seçilmesi ağın yerel minumumlar arasında atlmasına, küçük seçilmesi iterasyon sayısının artmasına neden olur. Literatürde momentum katsayısının seçilmesinde 0.2- 0.6 aralığının başarılı sonuçlar verdiği belirtilmektedir [41].

Öğrenme katsayısı gibi Momentum katsayısı da öğrenme başarısını etkiler. Momentum katsayısı, değişimin belirli bir oranının yeni değişime eklenmesidir. Tecrübeler Momentum katsayısının 0.6-0.8 aralığında başarılı sonuçlar verdiğini gösterir [41]. Uygulama üzerinde ki katsayı atama işleminin sözde kodu Tablo 3.5'te verilmiştir.

Tablo 3.5. YSA katsayı atama sözde kodu

```

1: INPUT;//Eğitim kümesi ve test verisi
2: TestImage <- testverisi
3: Train <- egitimseti
4: FUNCTION;
5:   YSA ysa <- new YSA();
6:   setMomentum <- momentumKatsayısı
7:   setLearningRate <- ogrenmeOranı
8:   setHiddenLayers <- "a" //auto (Otomatik Hücre Sayısı)=(Özellik sayısı + Sınıf sayısı)/2
9:   setNumDecimalPlaces <- hassaiyet
10:  setTrainingTime <- çevrimSayısı
11:  // YSA Eğitim ve Test İşlemi
12:  ysa <- Train
13:  className <- ysa(TestImage)
14: END FUNCTION

```

Ağın eğitilmesi için hata miktarının ağırlıklara dağıtılması gerekmektedir. Ağın yeni ağırlıkları denklem 3.21, denklem 3.22, denklem 3.23, denklem 3.24, denklem 3.25 ve denklem 3.26'daki ifadelerle göre hesaplanır.

$$E_m = \text{istenilen çıkış} - \text{ağ çıkışı} \quad (\text{Çıkış hatası}) \quad (3.21)$$

$$\delta_m = y_m (1 - y_m) * E_m \quad (3.22)$$

$$\Delta A_{jm}(t) = \lambda * \delta_m * y_{jm} + \alpha * \Delta A_{jm}(t-1) \quad (\text{Çıkış ile ara katman arası değişim}) \quad (3.23)$$

$$A_k(t) = A_k(t-1) + \Delta A_k(t) \quad (\text{Ara katman ile çıkış arasındaki yeni ağırlıklar}) \quad (3.24)$$

$$\delta_j^a = y_j^a (1 - y_j^a) \sum_m A_{jm}^a (t-1) \delta_m \quad (\text{Ara Katman Hata oranları ve Değişim miktarları}) \quad (3.25)$$

$$A_k(t) = A_k(t-1) + \Delta A_k(t) \quad (\text{Ara katman ile giriş arasındaki yeni ağırlıklar}) \quad (3.26)$$

Denklem 3.21'de verilen E_m çıktı hatasıdır. Denklem 3.23'te verilen ΔA_{jm} değeri ara katman ile çıkış katmanı arasındaki değişim miktarıdır. λ öğrenme oranını ve α momentum katsayısını temsil eder. Denklem 3.23'te verilen çıkış ile ara katman arasındaki değişim miktarı ara katmandaki bütün hücreler için ayrı ayrı hesaplanmaktadır. Denklem 3.23'te m hücre sırasını, y_{jm} değeri ise m inci ara hücreye ait çıkış değeridir.

Denklem 3.24'te verilen $A_k(t)$ değeri ara katman ile çıkış arasındaki yeni ağırlık değeridir. Denklem 3.23'te bulunan değişim miktarı ile ara katman ve hücre arasındaki ağırlık değeri toplanarak güncellenmiş ağırlık değeri bulunur.

Denklem 3.24'te ara katman ile çıkış değeri arasındaki ağırlık hesaplandıktan sonra ara katmandaki hata oranları ve değişim miktarlarının hesaplanması işlemine geçilir. Denklem 3.25'te verilen δ_j^a ifadesi ara katmana ait çıkış hatasını verir. Bu çıkış hatası denklem 3.23'e uygulanarak ara katman ile giriş katmanı arasındaki hata değeri bulunur. Denklem 3.26'da, bulunan hata, giriş katmanı ile ara katman arasındaki ağırlık değeriyle toplanarak güncellenmiş ağırlık değerini oluşturur. Bu işlemler yapıldıktan sonra hata ağı yayılmıştır. YSA üzerinde belirlenen çıkış hatası eşik değerine göre veya çevrimsayısına göre öğrenme işlemi devam eder.

3.2.3. Destek Vektör Makinesiyle Sınıflandırma

Özellik çıkarım bölümünde elde edilen özelliklerin, sınıflandırılması için kullanılan yöntemlerden biri de Destek Vektör Makinesi (Support Vector Machine-DVM)'dir. DVM sınıflandırıcısı, karmaşık olmayan eğitim kümeleri için düşük hafıza tüketimli ve düşük işlemci yüküne sahip çözümler sunar. Sınıflandırma fonksiyonu oluşturulurken sınıfların birbirine en yakın noktaları üzerinden işlem yapıldığından, gereksiz noktaların getirdiği hafıza kaybı ve işlem yükü elimine edilir. Tek çevrimde sınıflandırma fonksiyonunu oluşturmasından dolayı, genellikle hızlıdır. Geliştirilen Grafiksel Arayüz (Graphical User Interface-GUI) aracılığıyla test edilen mermer görüntüsünün özellikleri ve sınıfı birleştirilerek eğitim kümesine eklenebilir. DVM her test işleminde güncellenmiş eğitim setini kullandığından, daha gelişmiş bir sınıflandırma yapabilmektedir.

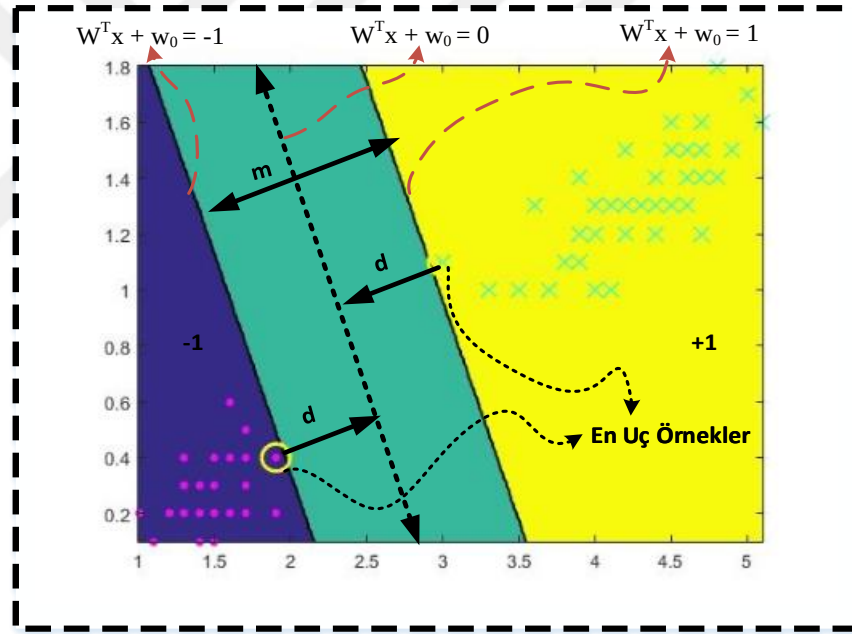
DVM sınıflandırıcısı verileri sınıflandırırken, sınıflar arası birbirine en yakın noktaları tespit ederek, bu noktaları ayıran fonksiyona dik uzaklığı maksimum yapmayı hedefler. Ayrıca sınıfları ayıran fonksiyonu oluştururken, bütün sınıflara eşit mesafede olmasını amaçlar. Doğrusal olarak ayıramayan verilerde çekirdek tipi değiştirilerek doğrusal olmayan veriler sınıflandırılabilir. Temelinde ikili sınıflandırma olmasına rağmen çoklu sınıflandırma işlemini de gerçekleştirebilir [40].

DVM sınıflandırıcısı, basit eğitim kümeleri sınıflandırılırken, karmaşık problemlerin çözülmemesi amaçlanarak geliştirilmiştir. Veriler sınıflandırılırken, bütün küme sınırlarına eşit mesafede en uzak fonksiyon aranır. Sınıflandırma fonksiyonu, veri tipine göre doğrusal veya doğrusal olmayan şekilde olabilir. Çoklu sınıfa sahip eğitim kümelerinde, birden fazla doğrusal

fonksiyon ile sınıflandırma işlemi gerçekleştirilebilir. Ancak çoklu doğrusal fonksiyon işlem maliyetini de beraberinde getirir. Ayrıca dağınık verilerden oluşan eğitim kümelerinde doğrusal fonksiyonlar sınıflandırma işlemi için yetersiz kalabilir. Doğrusal fonksiyonların yetersiz kaldığı durumlarda doğrusal olmayan çekirdekler kullanılabilir [40,51].

3.2.3.1. DVM Modeli

DVM, doğrusal ve doğrusal olmayan verileri ayırmak için çözümler sunar. Verilerin doğrusallığına göre farklı yöntemler kullanılarak sınıflandırma fonksiyonu üretilir. DVM'nin uç örnekler arası maksimum mesafeye göre oluşturduğu doğrusal fonksiyon Şekil 3.9'da verilmiştir.



Şekil 3.9. DVM sınıflandırma düzlemi

Şekil 3.9'da verilen sınıfları birbirinden ayıran düzlem, iki sınıfa da eşit mesafede olmayı amaçlar.

3.2.3.2. Doğrusal Üstündüzlem Sınıflandırıcı

Veri kümesi doğrusal bir fonksiyon ile ayrıldığında bu fonksiyon aşağıdaki adımlar izlenerek oluşturulur [40].

1. Doğrusal üstündüzlem eğitim kümesini iki sınıfa ayırır. Bu sınıfları -1 ve $+1$ etiket değerleriyle temsil edilir. Eğitim kümesi $X = \{x^t, r^t\}$ şeklinde temsil edilir.

- $x^t \in C1 \rightarrow r^t = +1$ ve $x^t \in C2 \rightarrow r^t = -1$ şeklinde sınıflara atanır.
- $W(W1, W2, \dots, Wn)$ ağırlık vektörleridir.(n niteliklerin sayısıdır.)
- w_0 Ağırlık vektörlerine göre değişen sabit bir sayıdır.

2. Sınırları belirlemek için W ve w_0 değerleri denklem 3.27 ve denklem 3.28'deki koşulları sağlamalıdır.

$$\bullet \quad (r^t = +1) \rightarrow (W^T x^t + w_0 \geq +1) \quad (3.27)$$

$$\bullet \quad (r^t = -1) \rightarrow (W^T x^t + w_0 \geq -1) \quad (3.28)$$

- Denklem 6.1 ve 6.2'deki iki eşitsizlik, denklem 6.3'deki gibi birleştirilir.

$$\bullet \quad r^t(W^T x^t + w_0) \geq 1 \quad (3.29)$$

- Şekil 3.24'te verilen d uzaklığı kenar payı olarak adlandırılır. İyi bir genelleme fonksiyonu üretmek için kenar payı olabildiğince büyük olması istenir. Böylece yeni veriler gürültülü olsa bile doğru sınıflandırma yapılır.

3. Şekil 3.24'te verilen d uzaklığını hesaplamak için düzlemin her iki tarafını kapsayan genelleme yapılması gerekmektedir.

$$\bullet \quad r^t \in \{-1, +1\} \text{ olduğuna göre } d = \frac{r^t(W^T x^t + w_0)}{\|W\|} \text{ şeklinde ifade edilir.}$$

- Bu değer en az p olması gerekmektedir.

$$d = \frac{r^t(W^T x^t + w_0)}{\|W\|} \geq p \quad (3.30)$$

4. P değeri olabildiğince büyük tutulmaya çalışılır. Ancak bu işlemi W vektörü ölçeklenerek gerçekleştirilemez. Aynı sonuç için $p\|W\| = 1$ olarak sabitlenmesi gerekir.

- $(r^t(W^T x^t + w_0) \geq +1)$ koşulunu sağlayan $(\min \frac{1}{2} \|W\|^2)$ değeri bir iyileme işlemidir. Bu çözüm en iyi W ve w_0 değerini verir. Dolayısıyla uç örnekler, sınıflandırma fonksiyonuna en az $1/\|W\|$ ve marjini ise $2/\|W\|$ dir.
5. Eğitim kümesini oluşturan sınıflar doğrusal ayrılamıyorsa, yeni bir uzaya taşınması gerekir. Genelde bu uzayda boyut sayısı ilkinin göre daha fazladır. İlk uzaydaki iyileme işlemi d ye göre gerçekleştirilir. Ancak örnek sayısının, N ye göre yazılması gerekmektedir. Bu yeni tanımlama işlemi için Lagrange çarpanlarından yararlanılır.
6. Dördüncü adımdaki kısıt ve iyileme işlemleri, Lagrange çarpanları kullanılarak kısıtlanmamış şekilde denklem 3.31'deki gibidir.

$$\begin{aligned}
 L_p &= \frac{1}{2} \|W\|^2 - \sum_{t=1}^N a^t [r^t(W^T x^t + w_0) - 1] \\
 &= \frac{1}{2} \|W\|^2 - \sum_t a^t r^t [(W^T x^t + w_0) + \sum_t a^t]
 \end{aligned} \tag{3.31}$$

- Bu işlemler ile W değerinin W_0 değerine göre en küçük, a^t değişkenine göre en büyük değeri aranır. L_p problemi ikinci dereceden bir iyileme işlemidir. Bundan dolayı Karush-Kuhn-Tucker (KKT) koşulları kullanılarak eşik problemi çözülür. KKT doğrusal olmayan problemlerin çözümünde optimizasyon yapmak yani alternatifler içerisinde en uygun değeri bulmak için kullanılır. KKT için amaç ve kısıt fonksiyonu sırasıyla denklem 3.32 ve 3.33'de verilmiştir.
- $$\max. z = f(x) \tag{3.32}$$
- $$g(x) \leq 0 \tag{3.33}$$
7. L_p problemi W ve W_0 değerlerine göre eğim $= 0$ ve $a^t \geq 0$ koşullarını sağlaması gerekmektedir. Bu koşullar sağlandığında L_p değerinin büyük olması istenir. Eğim L_p nin türevi alınarak hesaplanmaktadır.

$$\frac{\partial L_p}{\partial W} = 0 \rightarrow W = \sum_t a^t r^t x^t \tag{3.34}$$

$$\frac{\partial L_p}{\partial w_0} = 0 \rightarrow w_0 = - \sum_t a^t r^t \tag{3.35}$$

8. Denklem 3.34 ve 3.35, denklem 3.31'e yerleştirilerek L_d değeri hesaplanır.

- $$\begin{aligned}
L_d &= \frac{1}{2}(W^T W) - W^T \sum_t a^t r^t x^t - W_0 \sum_t a^t r^t + \sum_t a^t \\
&= -\frac{1}{2}(W^T W) + \sum_t a^t \\
&= -\frac{1}{2} \sum_t \sum_s a^t a^s r^t r^s (x^t)^T x^s + \sum_t a^t
\end{aligned}
\tag{3.36}$$

9. L_d değeri a^t ye göre büyürken, kısıtlar denklem 3.37’de verilmiştir.

- $$\sum_t a^t r^t = 0 \text{ ve } a^t \geq 0 \text{ şeklindedir.} \tag{3.37}$$

10. N tane a^t değeri hesaplandığında çoğunun sıfır olduğu görülür. Denklem 3.34’deki koşulu sağlayan x^t örneklerine destek vektörleri denir. Bulunan x^t örnekleri denklem 3.38’deki koşulu sağlar; ve sınır vektörüdür.

- $$r^t(W^T x^t + w_0) = 1 \tag{3.38}$$

11. Denklem 3.38’deki eşitlik kullanılarak herhangi bir destek vektörü ile W_0 değeri hesaplanır.

- $$W_0 = r^t - W^T x^t \tag{3.39}$$

12. Bütün destek vektörleri için bu işlemler gerçekleştirilip ortalaması alınır. Bu şekilde ortaya çıkan sınıflandırma fonksiyonuna, destek vektör makinesi denir. Onuncu adımda görüldüğü gibi DVM, sadece $a^t \geq 0$ bu koşulu sağlayan vektörleri kullanır. Koşulu sağlamayan vektörler sınıflandırma fonksiyonuna etki etmez. Bu vektörler atılarak karmaşıklık azaltılır.

13. Sınıflandırma işleminde, sınıfı bilinmeyen bir örnek test edildiğinde d değişkenine bakılmaksızın denklem 3.40 ve denklem 3.41 hesaplanır.

- $$g(x) = W^T x + w_0 \text{ değeri hesaplanır.} \tag{3.40}$$

- $$g(x) > 0 \rightarrow C_1 \text{ değilse } C_2 \text{ sınıfına aittir.} \tag{3.41}$$

3.2.3.3. Doğrusal Olmayan Çekirdek Sınıflandırıcı

Doğrusal olmayan eğitim kümesini doğrusal olmayan bir fonksiyon ile çözmek işlem yükünü artırır. Doğrusal olmayan eğitim kümesi, taban işlevleriyle yeni bir uzaya çıkarılıp doğrusal bir model şeklinde çözülür. Yeni oluşturulan doğrusal uzay, doğrusal olmayan uzaya denk olur. DVM, vektör çekirdekleri üzerinde işlem yaptığından genel olarak çekirdekleme işlemi olarak da adlandırılır.

Çekirdek makinelerinin amacı, yeni uzayın boyutlarından oluşan iç çarpımı, bir çekirdek fonksiyonu şeklinde uygulamaktır. Bu işlem yerine çekirdek fonksiyonu, doğrudan ilk uzaydaki gösterim fonksiyonlarına uygulanır. Çekirdek fonksiyonu olarak $K(x^t, x^s)$ tanımlanır ve ilk uzaydaki gösterimlere genel olarak denklem 3.42 ve 3.43'deki gibi uygulanır.

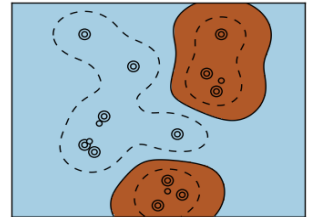
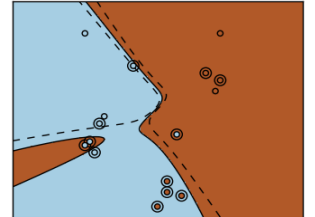
$$L_d = \sum_t a^t - \frac{1}{2} \sum_t \sum_s a^t a^s r^t r^s K(x^t, x^s) \quad (3.42)$$

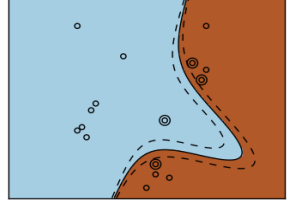
$$g(x) = \sum_t a^t r^t K(x^t, x) \quad (3.43)$$

3.2.3.4. Çekirdek Fonksiyonları

Doğrusal ayrılmayan eğitim kümelerinde çekirdek makineleri kullanılır. Sıkça kullanılan çekirdek makineleri Tablo 3.6'da verilmiştir [40].

Tablo 3.6. DVM çekirdekleri

Gauss Çekirdeği (RBF)	$\gamma=1/2\sigma^2$, $K(x_i, x_j)=\exp(-\gamma\ x_i - x_j\ ^2)$	
Hiperbolik Tanjant Çekirdek:	$K(x^t, x) = \tanh(k \cdot x_i \cdot x_j + c)$	

Polinomal Çekirdek:	$K(x_i, x_j) = (x_i * x_j)^d$	
Puk Çekirdek:	$K(x_i, x_j) = \left[1 + \left(2\sqrt{\ x_i - x_j\ ^2} \sqrt{2^{(1/w)} - 1} / \sigma \right)^2 \right]^w$	

3.2.3.5. Parametrelerin Belirlenmesi

DVM sınıflandırma fonksiyonunu üretmeden önce sistemde katsayıların tanımlı olması gerekmektedir. Yapılan bu çalışmada, kullanıcı DVM girişlerini GUI üzerinden değiştirebilmektedir. Bu durum farklı mermer gruplarında en iyi sonucu alabilmek için kullanıcıya test ortamı sunar.

DVM sınıflandırma fonksiyonunun doğrusallığı çekirdek tipine göre belirlenir. Veri kümesi doğrusal olarak ayrılabilirse doğrusal çekirdek kullanılmalıdır. Veri kümesi doğrusal olarak ayrılamıyorsa Gauss, Hiperbolik, Polinomal ve Puk gibi doğrusal olmayan çekirdekler ile sınıflandırma yapılır.

Filtre tipi, eğitim kümesini düzenleyerek daha başarılı sınıflandırma fonksiyonlarının üretilmesini sağlar. Normalize filtre tipinde eğitim kümesindeki veriler belli bir aralığa çekilerek sınıflandırma fonksiyonu oluşturulur. Standardize filtre tipinde ise veriler olasılıksal değerleri üzerinden işlem görür.

Karmaşıklık değeri ise sınıflandırma fonksiyonunun yumuşaklığını ayarlar. Bu çalışmada karmaşıklık değeri 3 en keskin değeri, 0 ise en yumuşak değeri temsil eder. Katsayıları GUI üzerinden sisteme veren sözde kod Tablo 3.7’de verilmiştir.

Tablo 3.7. DVM katsayı atama sözde kodu

```

1: INPUT; //Eğitim kümesi ve test verisi
2: TestImage <- testverisi
3: Train <- eğitimseti
4: FUNCTION;
5:   SVM svm=new SVM();
6:   setKernel<- PolyKernel
7:   setFilter <- standardize
8:   setNumDecimalPlaces <- hassaiyet
9:   setKarmaşıklık <- complexity
10:  // SVM Eğitim ve Test İşlemi
11:  svm <- Train
12:  className <- svm(TestImage)
13: END FUNCTION

```

3.3. Bölüm Değerlendirmesi

Bu bölümde LBP, Histogram ve SIFT tabanlı eğitim kümeleri kullanılarak ELM ve toplu sınıflandırıcı içerisinde bulunan Karar ağacı, YSA ve DVM eğitilmiştir. Sınıflandırma parametreleri geliştirilen GUI aracılığıyla kullanıcı tarafından belirlenebilmektedir. Bu sayede çeşitli eğitim kümelerinde katsayılar değiştirilerek başarı ve işlem süresine göre karşılaştırma yapılabilir. Ayrıca sınıflandırıcıların başarı oranına göre ROC alanları belirlenerek performans karşılaştırmaları verilmiştir.



4. DENEYSEL SONUÇLAR VE GUI'NIN GELİŞTİRİLMESİ

Sınıflandırma algoritmalarının eğitim kümeleri üzerinde sınıflandırma başarımları girilen parametrelere göre değişkenlik göstermiştir. ELM ve toplu sınıflandırıcının başarımları girilen parametreler göre LBP ve çoklu özellik kümesinde test edilmiştir. Bu test işleminin akış süreci geliştirilen GUI üzerinde toplanmıştır.

4.1. Deneysel Sonuçlar

Bu bölümde sınıflandırıcıların eğitim kümeleri üzerindeki çapraz doğrulama başarımları ve ROC analizleri verilmiştir.

4.1.1. Sınıflandırma Performansının İncelenmesi

Tez çalışması kapsamında sınıflandırıcının doğruluk ve hassasiyetini ölçmek için Alıcı İşletim Özellik Eğrisi (Receiver Operating Characteristic-ROC) kullanılmıştır. ROC sınıflandırma işlemi sonuçlarının, gerçek değerleriyle karşılaştırılarak performansının ölçülmesini sağlayan bir yöntemdir. Duyarlılığın özgülüğe karşı gösterdiği tepkiye göre incelenir. ROC eğrisinde duyarlılığın istenilen değeri daima 1'dir. Özgülüğün ise sınır değeri 1 olduğundan eğri altında kalan alan en fazla 1 olabilir. Sınıflandırma performansının artırılmasını izlemek için ROC alanı incelenir.

Tahmin edilen değer pozitif (P) ve gerçek değer de pozitif (P) olduğu durumlar doğru pozitif (TP) olarak adlandırılır.

Tahmin edilen değer pozitif (P) ve gerçek değer negative (N) olduğu durumlar yanlış pozitif (FP) olarak adlandırılır.

Tahmin edilen değer negatif (N) ve gerçek değer de negatif (N) olduğu durumlar doğru negatif (TN) olarak adlandırılır.

Tahmin edilen değer negatif (N) ve gerçek değer pozitif (P) olduğu durumlar yanlış negatif (FN) olarak adlandırılır. ROC grafiğinde eğri TPR eksenine yaklaşırsa doğruluğu artar. Eğri altında kalan alan ne kadar büyükse sistem o derece güvenlidir. Gerçek ve tahmini değerlere göre karmaşıklık matrisinin dizilimi Tablo 4.1'de verilmiştir.

Tablo 4.1. Karmaşıklık matrisi

Tahmin \ Gerçek	P	N
	TP	FP
P		
N	FN	TN

Duyarlılık, gerçek sınıfa ait örneklerin test tarafından hangi oranda saptanabildiğini belirten bir olasılıktır. Özgüllük, bir testin gerçek sınıfa ait olmayanları ayırabilme yeteneğini belirten orandır. Doğruluk, gerçekte testin doğru ve yanlış olarak ayrılma oranıdır. Duyarlılık, Özgüllük ve Doğruluk değerleri denklem 4.1, denklem 4.2, denklem 4.3'teki gibi hesaplanır.

$$\text{Duyarlılık (Sensitivity)} = \frac{TP}{TP+FN} * 100 \quad (4.1)$$

$$\text{Özgüllük (Specificity)} = \frac{TN}{TN+FP} * 100 \quad (4.2)$$

$$\text{Doğruluk (Accuracy)} = \frac{TN+TP}{TN+FP+FN+TP} * 100 \quad (4.3)$$

4.1.2. ELM Deneysel Sonuçlar

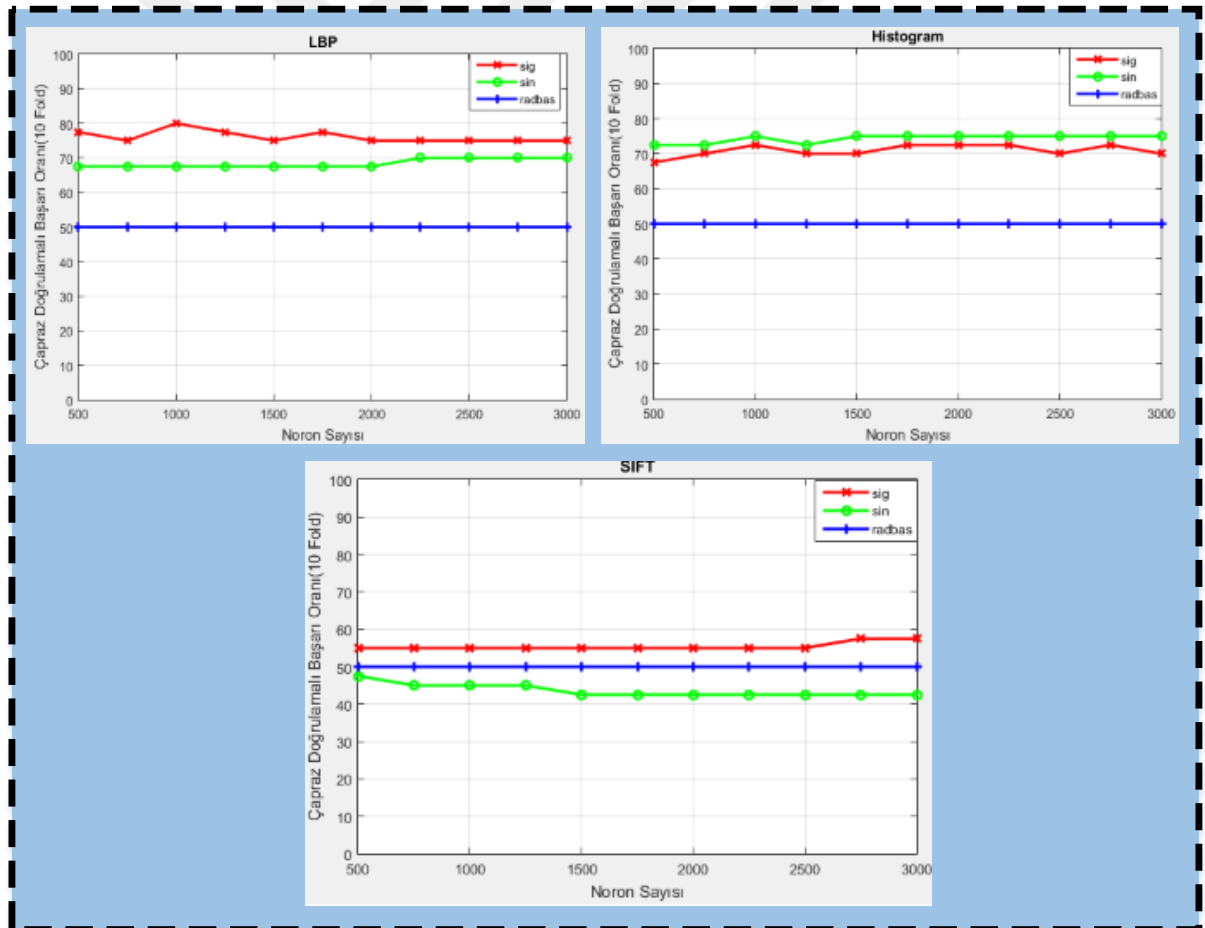
ELM, özellik çıkarım aşamasında elde edilen LBP, Histogram ve SIFT tabanlı özellikleri kullanır. Her bir görüntü kümesinin özellikleri ayrı ayrı dosyalarda tutulur. ELM eğitilirken parametrelerin girildiği menü Şekil 4.1'de verilmiştir. Bu şekildeki veriler örnek olarak verilmiştir.

Şekil 4.1. ELM eğitim parametreleri

Şekil 4.1’de verilen GUI menüsünde, ELM sınıflandırıcısına eğitilmeden önce değerler atanır. Deneysel ortamda elde edilen en uygun performans değerleri “Önerilen Ayarlar” bölümüne arka planda kaydedilir. Dolayısıyla kullanıcı bu kontrol kısmını işaretlerse, kendi belirlediği ayarlara göre eğitim işlemi başlatılır.

4.1.2.1. Aktivasyon Fonksiyonu ve Hücre Sayısına göre Başarım Performansının Karşılaştırılması

ELM’de kullanılan aktivasyon fonksiyonu ve hücre sayısı, sınıflandırma başarısını doğrudan etkileyen faktörlerdir. Tablo 3.4’te verilen aktivasyon fonksiyonlarına göre hücre sayısının 10-kat çapraz doğrulamaya göre sınıflandırma başarımı Şekil 4.2’de verilmiştir.



Şekil 4.2. Hücre sayısına göre başarı oranının karşılaştırması

Şekil 4.2’deki grafiklerde görüldüğü gibi en yüksek çapraz doğrulamalı başarı oranına, LBP tabanlı ELM’de ulaşılmıştır. Bu tez çalışmasında geliştirilen GUI aracılığıyla yüksek başarı sağlayan parametreler tespit edilebilir. Sınıflandırma başarısının en yüksek olduğu parametreler sisteme tanıtılarak sürekli kullanılması sağlanır. ELM’de sigmoid aktivasyon fonksiyonunun diğerlerine göre daha yüksek başarıma ulaştığı görülmektedir.

4.1.2.2. ROC Analizi

En yüksek başarı oranına sahip 10-kat çapraz doğrulamalı ELM’de, örneklerin gerçek sınıflarına karşılık gelen tahmini sınıflar Tablo 4.2’de verilmiştir.

Tablo 4.2. ELM 10-kat çapraz doğrulama

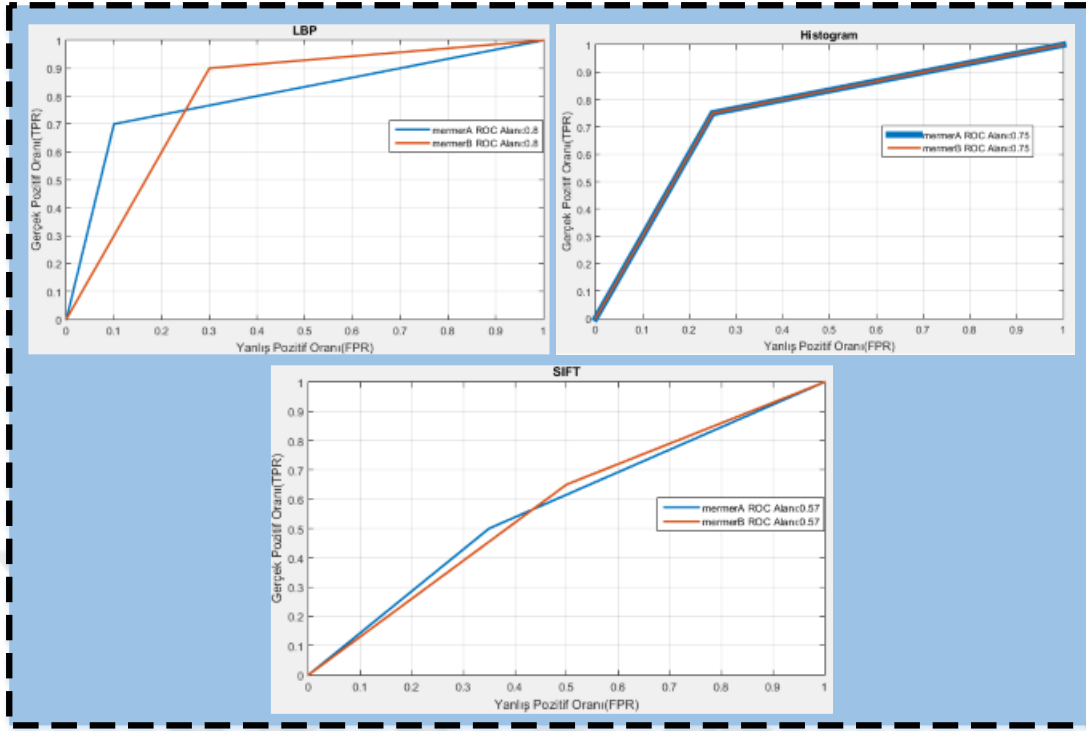
	LBP		Histogram		SIFT	
Gerçek Tahmin	A	B	A	B	A	B
A	14	6	15	5	10	10
B	2	18	5	15	7	13

Karmaşıklık matrisinde sırasıyla bir sınıf “True” olarak işlem gördüğünde diğer bütün sınıflar ayrı sınıf şeklinde değerlendirilip “False” olarak işlem görür. Tablo 4.2’deki değerler kullanılarak oluşturulan karmaşıklık matrisleri Tablo 4.3’teki gibidir.

Tablo 4.3. ELM tabanlı karmaşıklık matrisi

	LBP		Histogram		SIFT	
Gerçek Tahmin	TP Oranı	FP Oranı	TP Oranı	FP Oranı	TP Oranı	FP Oranı
A	0.7	0.1	0.75	0.25	0.5	0.35
B	0.9	0.3	0.75	0.25	0.65	0.5

Tablo 4.3’te verilen karmaşıklık matrislerinin ROC eğrileri Şekil 4.3’te verilmiştir.

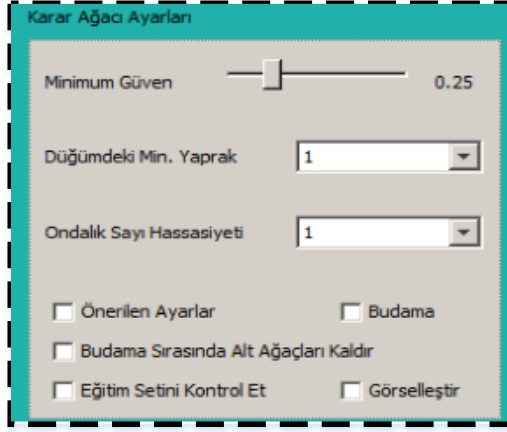


Şekil 4.3. ELM ROC alanlarının karşılaştırılması

4.1.3. Karar Ağacı Deneysel Sonuçlar

GUI üzerindeki test görüntüsünün sınıfı belirlenirken, eğitim verisinin tamamı kullanılarak eğitilen karar ağacı üzerinde sınıflandırma işlemi gerçekleştirilir. Şekil 3.3, Şekil 3.4 ve şekil 3.5'te verilen karar ağaçlarının eğitim başarıları sırasıyla % 95, 97.5 ve 90'dır.

Bu çalışmada C4.5 tabanlı karar ağacı kullanılmıştır. Karar ağacının başarı oranları mermer çeşitlerine göre değişkenlik gösterebilir. Bu nedenle kullanıcının parametrelerini değiştirebileceği, bir karar ağacı eğitilmesi için GUI işlem menüsü geliştirilmiştir. Parametre ayarlarının yapıldığı işlem menüsü Şekil 4.4'te verildiği gibidir.

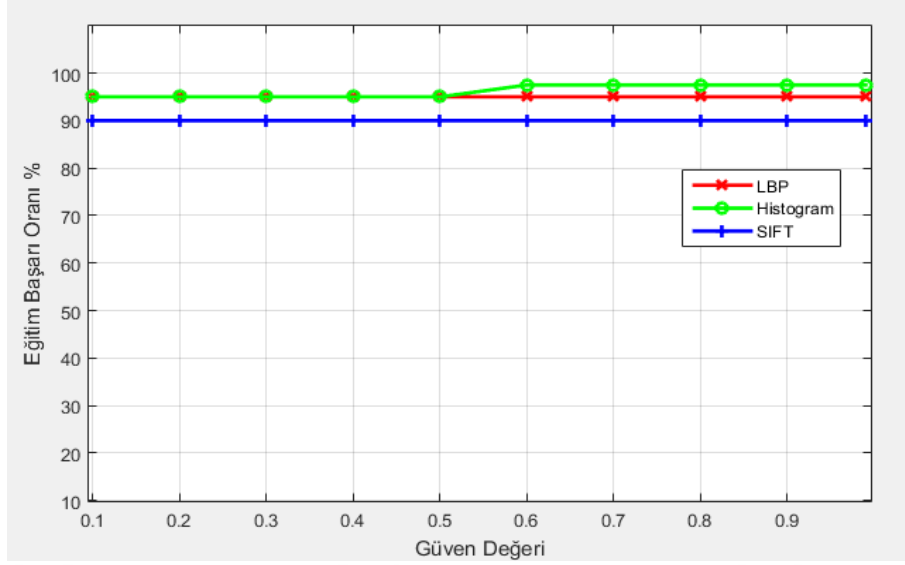


Şekil 4.4. Karar ağacı eğitim parametreleri

Yukarıdaki menüde karar ağacının güven değeri, düğümdeki en az yaprak sayısı, budama işleminin olup olmasını karar verilebilir. Düğüm değerleri hesaplanırken, ondalık hassasiyet belirlenebilir. Kullanıcı çeşitli mermer gruplarında bu ayarlamaları değiştirerek en başarılı sınıflandırma yöntemini tespit edebilir.

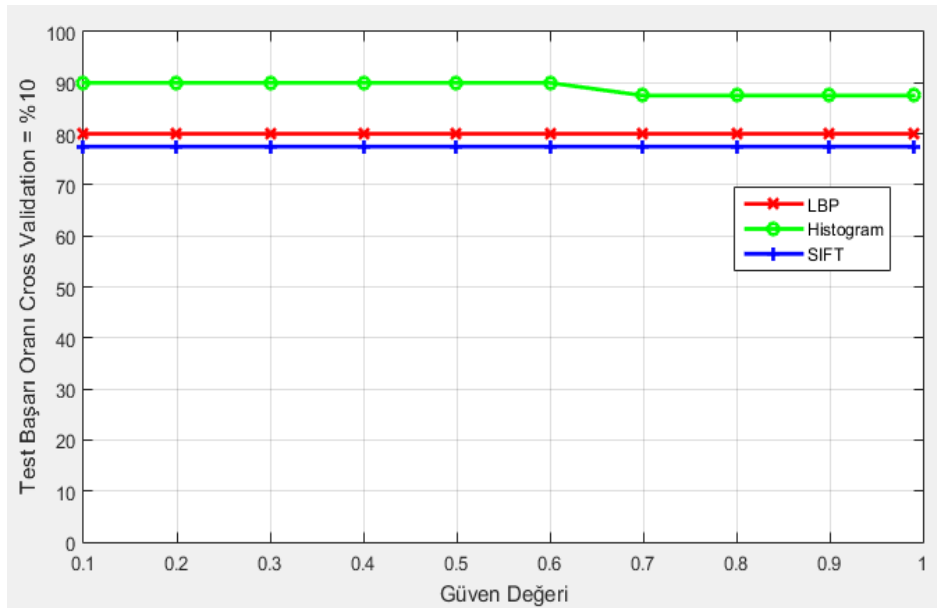
4.1.3.1. Sınıflandırma Başarımının Karşılaştırılması

Karar ağaçlarında düğümlerin ağaca eklenmesi için minimum güven değerini sağlaması gerekir. Güven değeri, karar ağacında budama boyutunu belirler. *Ondalık sayı hassasiyeti=2* ve *Minimum yaprak sayısı=1* kabulleri yapılarak güven değerine göre eğitim başarısı değişkenlik gösterebilir. LBP, Histogram ve SIFT tabanlı eğitim kümeleri kullanılarak ayrı ayrı karar ağaçları elde edilmiştir. Karar ağacı oluşturulurken eğitim kümesinin tamamı kullanılır ve bu eğitim kümesinin tamamı karar ağacı üzerinden test edilirse ortaya çıkan doğruluk oranı eğitim başarısıdır. Şekil 4.5'te güven değerinin eğitim başarısına etkisi özellik çıkarımına göre ayrı ayrı verilmiştir.



Şekil 4.5. Güven değerine göre eğitim başarısının karşılaştırılması

Yukarıdaki şekilde görüldüğü gibi güven değerinin sadece Histogram tabanlı Karar Ağacının eğitim başarısını arttırmıştır. Karar ağacının test başarısını ölçmek için çapraz doğrulama (Cross Validation) kullanılmıştır. Çapraz doğrulama, verilen orana göre eğitim kümesini eğitim ve test şeklinde bölerek sınıflandırma işlemi gerçekleştirir. Bütün eğitim seti test edildikten sonra ortalama test başarısı çapraz doğrulamanın başarısını vermektedir. Karar ağaçlarının güven değerine göre 10-kat çapraz doğrulamalı test başarısı Şekil 4.6’da verildiği gibidir.



Şekil 4.6. Güven değerine göre test başarısının karşılaştırılması

Şekil 4.6’da görüldüğü gibi histogram tabanlı çapraz doğrulama LBP ve SIFT tabanlı çapraz doğrulamaya göre daha yüksek test başarısına sahiptir.

4.1.3.2. ROC Analizi

“Güven değeri=0.6 ve Minimum Yaprak Sayısı=1 kabullerine göre sınıflandırma işleminde, örneklerin gerçek sınıflarına karşılık gelen tahmini sınıfları Tablo 4.4’te verilmiştir. Karmaşıklık matrislerinde A sınıfı pozitif değer olarak kabul edilirlen B sınıfı ise negatif değer olarak işlem görmektedir.

Tablo 4.4. Karar ağacı 10-kat çapraz doğrulama

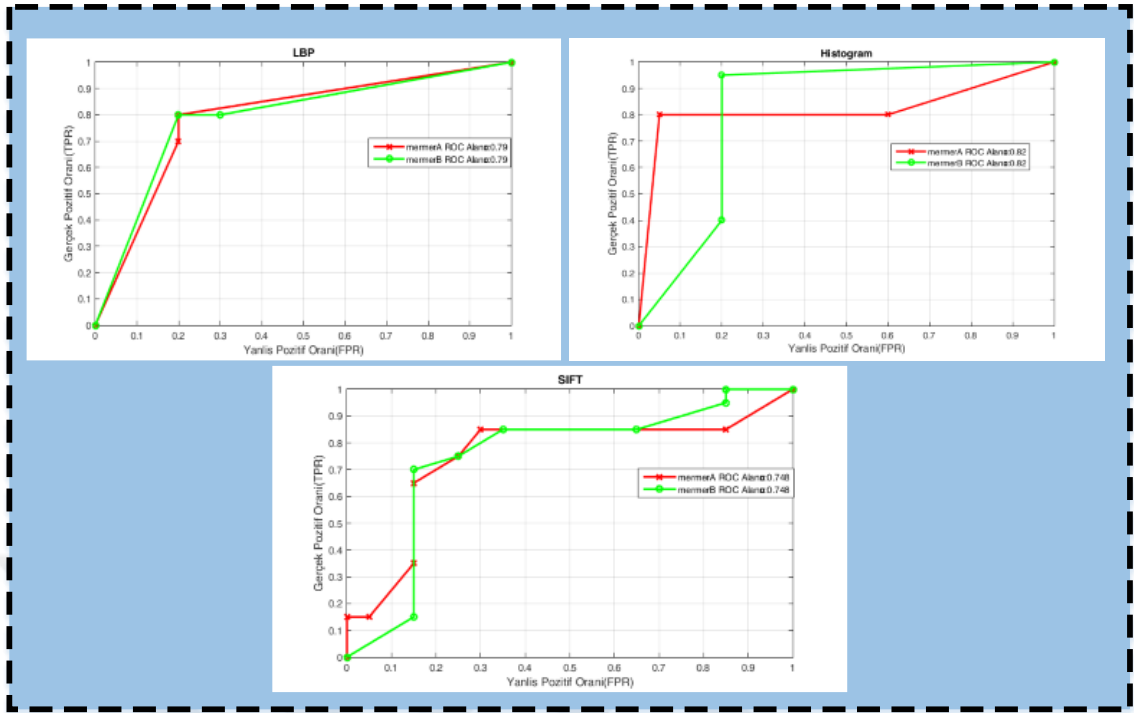
Gerçek \ Tahmin	LBP		Histogram		SIFT	
	A	B	A	B	A	B
A	16	4	16	4	17	3
B	4	16	1	19	6	14

Karmaşıklık matrisinde sırasıyla bir sınıf “True” olarak işlem gördüğünde diğer bütün sınıflar ayrı sınıf şeklinde değerlendirilip “False” olarak işlem görür. Tablo 4.4’teki değerler kullanılarak oluşturulan karmaşıklık matrisleri Tablo 4.5’te verilmiştir.

Tablo 4.5. Karar ağacı karmaşıklık matrisi

	LBP		Histogram		SIFT	
	TP Oranı	FP Oranı	TP Oranı	FP Oranı	TP Oranı	FP Oranı
A	0.8	0.2	0.8	0.05	0.85	0.3
B	0.8	0.2	0.95	0.2	0.7	0.15

Tablo 4.5’te verilen karmaşıklık matrislerinin ROC eğrileri Şekil 4.7’de verilmiştir.



Şekil 4.7. Karar ağacı ROC alanlarının karşılaştırılması

Şekil 4.7’de sınıfların, ROC eğrileri altında kalan alanları verilmiştir. Bu alan duyarlılığın özgülüğe karşı oranını vermektedir. ROC eğrisi “mermerA” sınıfına göre hesaplanırken, “mermerA” pozitif, “mermerB” negatif olur. “mermerB” için ROC eğrisi çizilirken tam tersi kabuller yapılır. Bu kabuller çoklu sınıflar içinde geçerlidir. Örneğin, üç sınıflı bir eğitim kümesinde, bir sınıf pozitif olarak kabul edildiğinde diğer iki sınıf bir bütün olarak düşünülüp negatif şeklinde işlem görür. Sınıfların genel TP, TN, FP ve FN değerleri hesaplandığından ROC alanları aynı olur.

4.1.4. YSA Deneyisel Sonuçlar

Özellik çıkarım aşamasında mermer görüntülerinin özelliklerinden eğitim kümesi elde edilmiştir. Eğitim kümesi Yerel İkili Örüntü (Local Binary Pattern-LBP), Histogram ve Ölçekten Bağımsız Özellik Dönüşümü (Scale Invariant Feature Transform -SIFT) olmak üzere üç çeşittir. Her bir görüntü kümesinin LBP, Histogram ve SIFT özellikleri ayrı ayrı dosyalarda tutulur. İstenilen özellik tipine göre, YSA sınıflandırma işlemini yapabilme esnekliği, GUI üzerinde kullanıcıya verilmiştir. Bu çalışmadaki YSA’da, yapısına göre “İleri Beslemeli”, öğrenme yöntemine göre “Danışmanlı Öğrenme” ve öğrenme zamanına göre “Statik Öğrenme” kullanılmıştır. Aktivasyon fonksiyonu olarak Sigmoid fonksiyon kullanılmıştır.

“*Rossolevanto*” adlı eğitim kümesi eğitilirken kullanılan parametre ve özellikler Şekil 4.8’de verilmiştir.



Yapay Sinir Ağı Ayarları

Momentum: 0.6

Öğrenme Oranı: 0.6

Katmanlar: a

Ondalık Sayı Hassasiyeti: 2

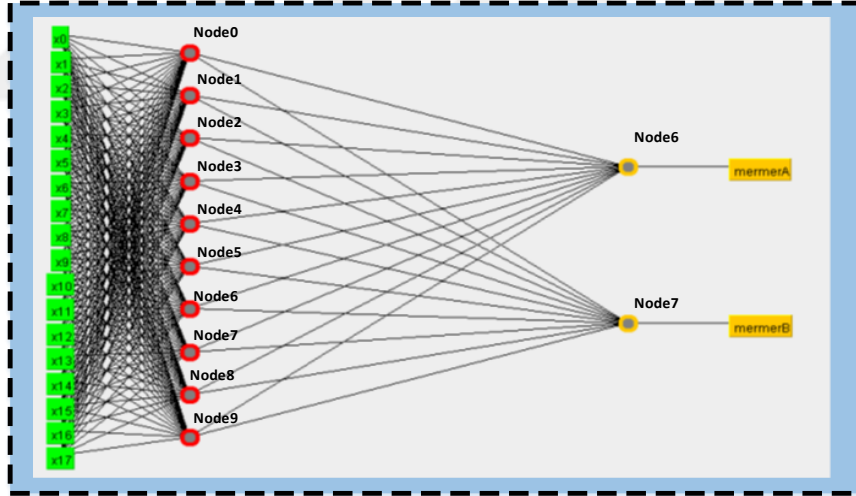
Eğitim Devir Sayısı: 3000

☐ Önerilen Ayarlar

☐ Eğitim Setini Kontrol Et

Şekil 4.8. YSA eğitim parametreleri

Şekil 4.8’de verilen ayarlar kullanılarak oluşturulan LBP tabanlı örnek bir YSA, Şekil 4.9’da verilmiştir



Şekil 4.9. LBP tabanlı YSA

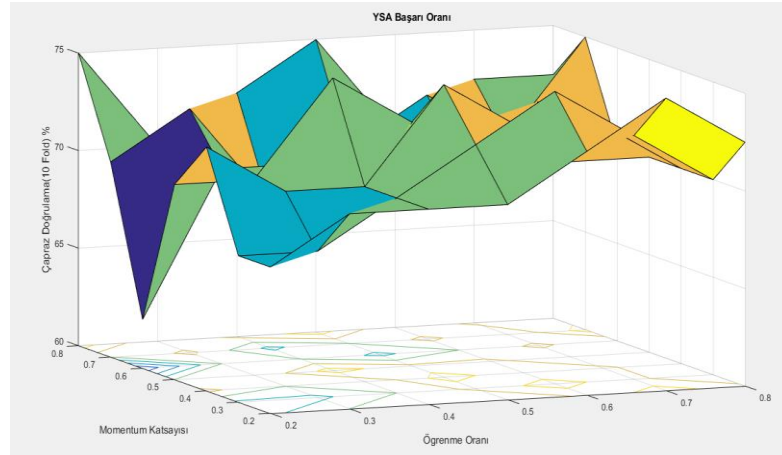
Şekil 4.9’da verilen görsel Weka tabanlı bir görseldir. Sınıflandırma algoritmaları Weka kütüphanesi kullanılarak oluşturulmaktadır.

4.1.4.1. Sınıflandırma Başarılarının Karşılaştırılması

YSA’da momentum katsayısı ve öğrenme oranının sınıflandırmaya etkisi test yöntemiyle belirlenir. Çevrim sayısının 500 olduğu YSA’nın, momentum katsayısı ve öğrenme oranına göre çapraz doğrulamalı başarı oranları Tablo 4.6, Tablo 4.7 ve Tablo 4.8’de başarı oranlarını gösteren grafik ise Şekil 4.10, Şekil 4.11 ve Şekil 4.12’de verilmiştir.

Tablo 4.6. LBP tabanlı YSA’nın çapraz doğrulamalı başarı oranı (Fold=10)

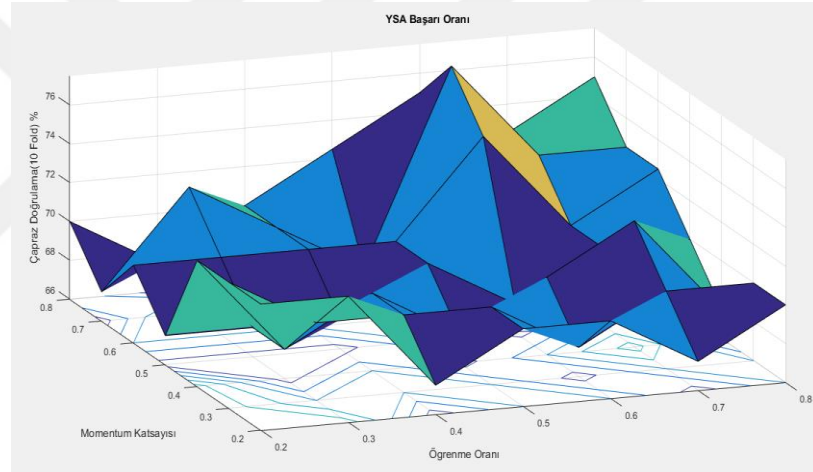
	M L	Momentum Katsayısı						
		0.2	0.3	0.4	0.5	0.6	0.7	0.8
Öğrenme Oranı	0.2	67.5	70.0	70.0	70.0	72.5	75.0	72.5
	0.3	67.5	67.5	70.0	72.5	75.0	72.5	70.0
	0.4	72.5	70.0	70.0	75.0	72.5	72.5	70.0
	0.5	70.0	70.0	75.0	72.5	72.5	70.0	70.0
	0.6	62.5	70.0	70.0	67.5	70.0	72.5	70.0
	0.7	70.0	72.5	67.5	70.0	72.5	70.0	75.0
	0.8	75.0	70.0	72.5	75.0	70.0	72.5	72.5



Şekil 4.10. LBP tabanlı YSA’nın çapraz doğrulamalı başarı grafiği

Tablo 4.7. Histogram tabanlı YSA'nın çapraz doğrulamalı başarı oranı (Fold=10)

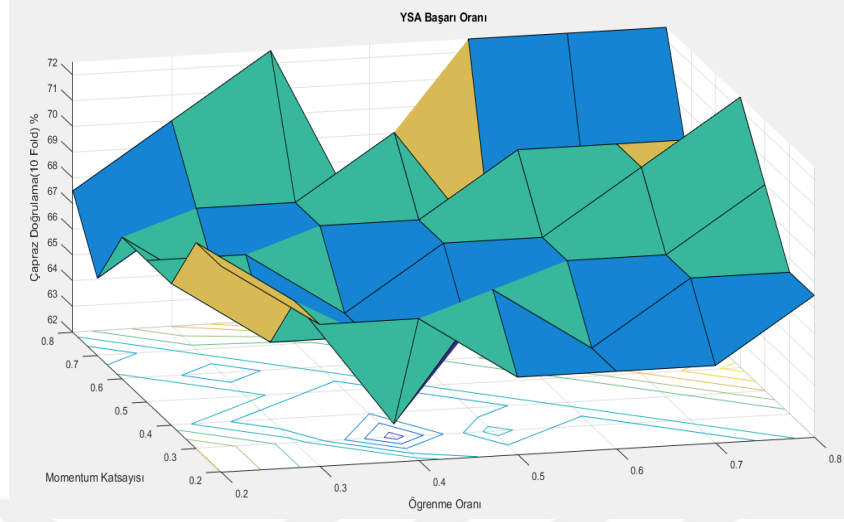
Öğrenme Oranı	M \ L	Momentum Katsayısı						
		0.2	0.3	0.4	0.5	0.6	0.7	0.8
	0.2	72.5	72.5	67.5	70.0	70.0	67.5	70.0
	0.3	72.5	70.0	70.0	70.0	67.5	70.0	70.0
	0.4	72.5	67.5	70.0	67.5	70.0	72.5	67.5
	0.5	67.5	67.5	67.5	70.0	67.5	70.0	70.0
	0.6	70.0	70.0	70.0	70.0	75.0	70.0	72.5
	0.7	67.5	72.5	70.0	67.5	77.5	72.5	72.5
	0.8	70.0	67.5	70.0	72.5	75.0	72.5	75.0



Şekil 4.11. Histogram tabanlı YSA'nın çapraz doğrulamalı başarı grafiği

Tablo 4.8. SIFT tabanlı YSA'nın çapraz doğrulamalı başarı oranı (Fold=10)

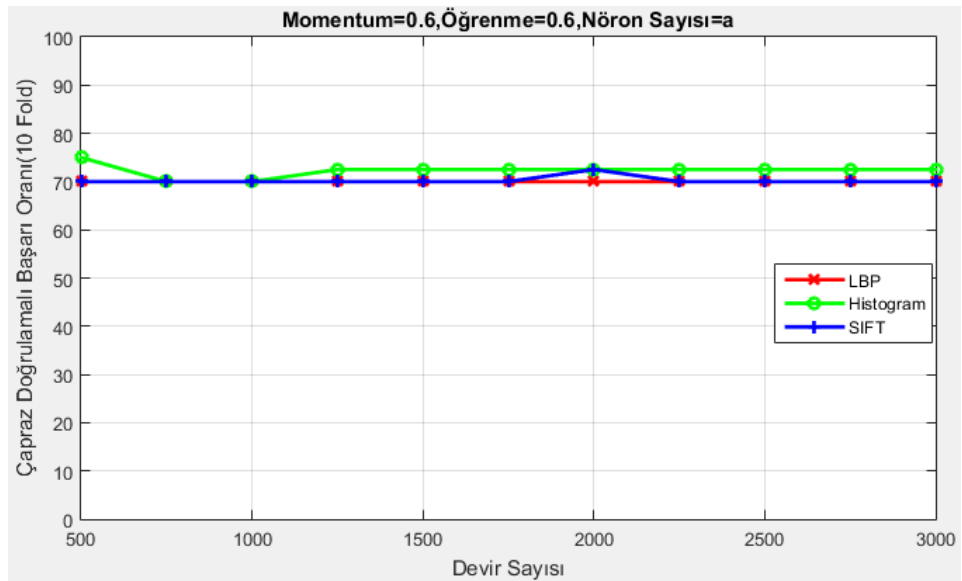
Öğrenme Oranı	M \ L	Momentum Katsayısı						
		0.2	0.3	0.4	0.5	0.6	0.7	0.8
	0.2	70.0	67.5	67.5	65.0	65.0	65.0	67.5
	0.3	70.0	67.5	62.5	67.5	65.0	67.5	67.5
	0.4	67.5	65.0	65.0	65.0	67.5	67.5	70.0
	0.5	67.5	67.5	65.0	67.5	67.5	70.0	72.5
	0.6	67.5	65.0	67.5	67.5	70.0	70.0	70.0
	0.7	65.0	67.5	67.5	70.0	65.0	65.0	67.5
	0.8	67.5	70.0	72.5	65.0	72.5	72.5	72.5



Şekil 4.12. SIFT tabanlı YSA'nın çapraz doğrulamalı başarı grafiği

4.1.4.2. Çevrim Sayısına göre Başarı Oranının Karşılaştırılması

YSA'da çevrim sayısı hata oranını azaltmada etkili olduğundan sınıflandırma başarısını doğrudan etkiler. Sabit öğrenme oranı ve momentum katsayısı kullanılarak başarı oranı belirli bir noktaya kadar arttırıldıktan sonra genelde sabit kalır. Şekil 4.13'te sabit öğrenme oranı ve momentum katsayılarına göre sınıflandırma başarı grafiği verilmiştir.



Şekil 4.13. Çevrim sayısına göre başarı oranının karşılaştırması

YSA sınıflandırıcısında belirli bir çevrim sayısından sonra oluşan hata değeri ağırlıklara fazla etki etmediğinden başarı oranını çok fazla etkilemez ve ağıın ezberlemesine neden olabilir. Şekil 4.13’te verilen grafikte görüldüğü gibi belirli bir çevrimden sonra grafik sabit devam etmiştir. Başarı oranının en yüksek ve çevrim sayısının en düşük olduğu nokta fayda/maliyet oranının en yüksek olduğu noktadır. Mermer sınıflandırma işleminde türlere göre sabit parametreler farklı başarı oranları verebilir. Başarı oranı eğitim kümesinin karmaşıklığına göre değişeceğinden YSA’daki sabit parametreler en başarılı sınıflandırmaya ulaşmak için değiştirilir.

4.1.4.3. ROC Analizi

Momentum katsayısı=0.6, Öğrenme Oranı =0.6, Çevrim = 2000 ve Katman sayısı=auto kabullerine göre oluşturulan çapraz doğrulamalı YSA’da, örneklerin gerçek sınıflarına karşılık gelen tahmini sınıfları Tablo 4.9’da verilmiştir.

Tablo 4.9. YSA 10-kat çapraz doğrulama

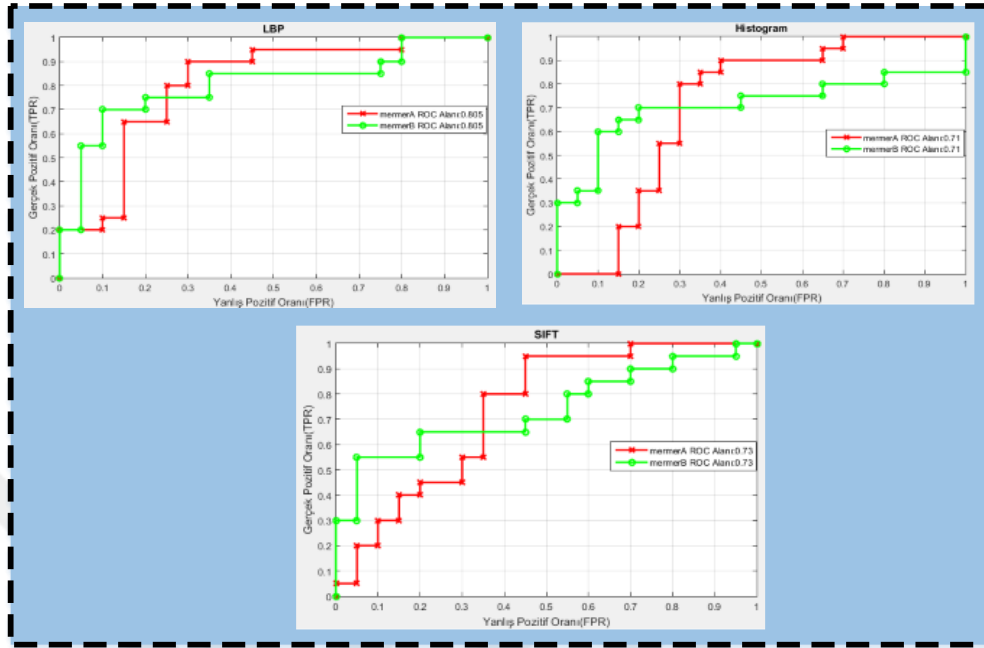
	LBP		Histogram		SIFT	
	A	B	A	B	A	B
Gerçek Tahmin						
A	14	6	15	5	16	4
B	5	15	6	14	7	13

Karmaşıklık matrisinde sırasıyla bir sınıf “True” olarak işlem gördüğünde diğer bütün sınıflar ayrı sınıf şeklinde değerlendirilip “False” olarak işlem görür. Tablo 4.9’a göre oluşturulan karmaşıklık matrisleri Tablo 4.10’da verilmiştir.

Tablo 4.10. YSA karmaşıklık matrisi

	LBP		Histogram		SIFT	
	TP Oranı	FP Oranı	TP Oranı	FP Oranı	TP Oranı	FP Oranı
A	0.7	0.25	0.75	0.3	0.72	0.038
B	0.75	0.3	0.7	0.25	1	0.14

Tablo 4.10’da verilen karmaşıklık matrislerinin, ROC eğrileri Şekil 4.14’te verilmiştir.

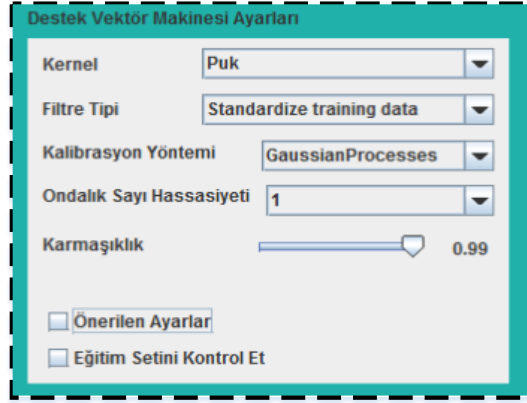


Şekil 4.14. YSA ROC alanlarının karşılaştırılması

Şekil 4.14’te sınıfların ROC eğrileri altında kalan alan verilmiştir. Bu alan bize duyarlılığın özgülüğe karşı oranını vermektedir. ROC alanı büyük olan sınıfın hatalara karşı daha dayanıklı olduğu söylenebilir. ROC alanını arttırmak için o sınıfa ait daha fazla örnek eğitim kümesine tanıtılabilir. Bu sayede sınıflandırmanın başarı oranı artar.

4.1.5. DVM DeneySEL Sonuçlar

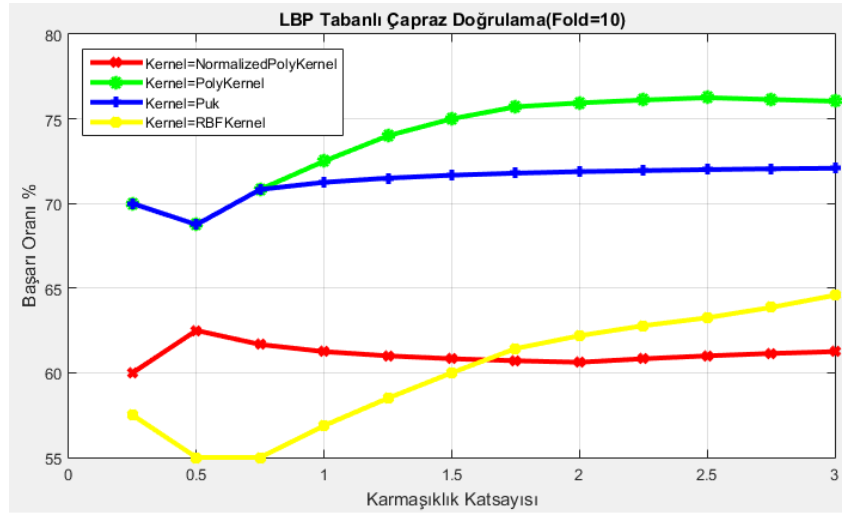
Eğitim kümesi Yerel İkili Örüntü (Local Binary Pattern-LBP), Histogram ve Ölçekten Bağımsız Özellik Dönüşümü (Scale Invariant Feature Transform -SIFT) olmak üzere üç çeşittir. Her bir görüntü kümesinin LBP, Histogram ve SIFT özellikleri ayrı ayrı dosyalarda tutulur. İstenilen özellik tipine göre sınıflandırma işlemini yapabilme esnekliği GUI üzerinde kullanıcıya verilmiştir. DVM eğitilirken parametrelerin girildiği menü Şekil 4.15’te verilmiştir.



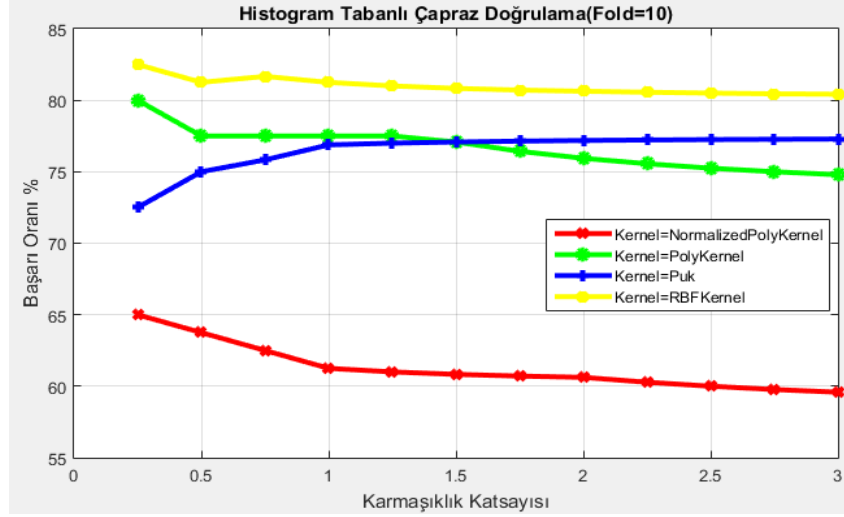
Şekil 4.15. DVM eğitim parametreleri

4.1.5.1. Karmaşıklık Değerine göre Başarım Oranının Karşılaştırılması

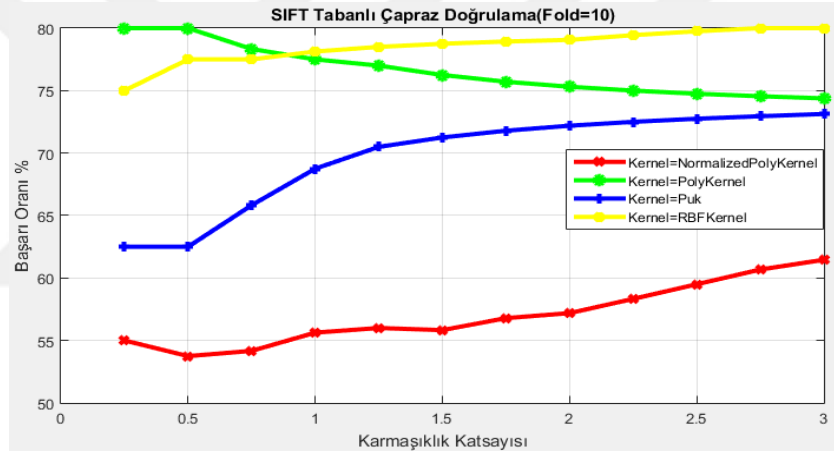
DVM’de karmaşıklık katsayısı, sınıflandırma başarısını doğrudan etkileyen bir değerdir. Şekil 4.16, Şekil 4.17 ve Şekil 4.18’de sabit filtre tipi ve işlem hassasiyeti kullanılarak, çekirdek tiplerine göre karmaşıklık katsayısının sınıflandırma başarısı verilmiştir. Test başarısı, 10 parçalı çapraz doğrulama yöntemine göre hesaplanmıştır.



Şekil 4.16. LBP tabanlı sınıflandırma başarısının karşılaştırması



Şekil 4.17. Histogram tabanlı sınıflandırma başarısının karşılaştırması



Şekil 4.18. SIFT tabanlı sınıflandırma başarısının karşılaştırması

Şekil 4.16, Şekil 4.17 ve Şekil 4.18’deki grafiklerde görüldüğü gibi karmaşıklık sayısı arttığında genellikle çekirdeklerin başarı oranının arttığı gözlemlenmiştir. Bu tez çalışmasında geliştirilen GUI aracılığıyla yüksek başarı sağlayan parametreler tespit edilebilir. Bu parametreler sisteme GUI üzerinde bulunan önerilen ayarlar kısmında tanıtılarak, otomatik kullanılması sağlanabilir. LBP, Histogram ve SIFT tabanlı çapraz doğrulamalı test işlemine göre, genellikle “PolyKernel” diğer çekirdeklere göre daha yüksek başarı oranlarına çıkmıştır. Dolayısıyla GUI üzerinde önerilen ayarlar kısmına tanımlanmıştır.

4.1.5.2. ROC Analizi

En yüksek başarımlı orana sahip çapraz doğrulamalı DVM’de, örneklerin gerçek sınıflarına karşılık gelen tahmini sınıflar Tablo 4.11’de verilmiştir.

Tablo 4.11. DVM 10-kat çapraz doğrulama

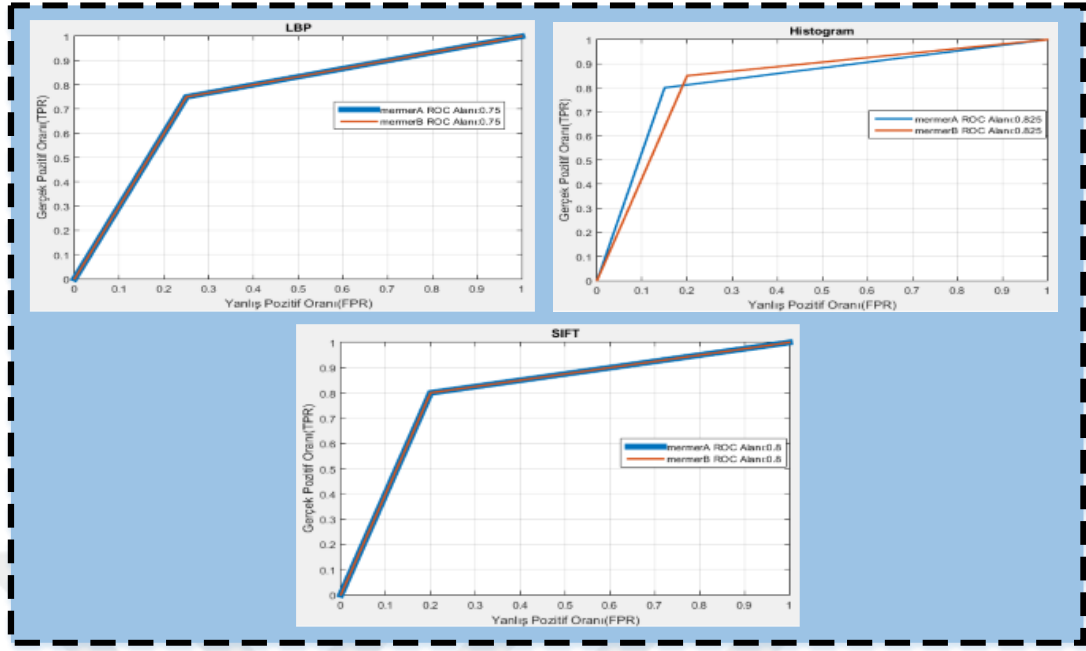
Gerçek Tahmin	LBP		Histogram		SIFT	
	A	B	A	B	A	B
A	15	5	16	4	15	5
B	5	15	3	17	3	17

Karmaşıklık matrisinde sırasıyla bir sınıf “True” olarak işlem gördüğünde diğer bütün sınıflar ayrı sınıf şeklinde değerlendirilip “False” olarak işlem görür. Tablo 4.11’deki değerler kullanılarak oluşturulan karmaşıklık matrisleri Tablo 4.12’de verilmiştir

Tablo 4.12. DVM karmaşıklık matrisi

	LBP		Histogram		SIFT	
	TP Oranı	FP Oranı	TP Oranı	FP Oranı	TP Oranı	FP Oranı
A	0.75	0.25	0.8	0.15	0.75	0.15
B	0.75	0.25	0.85	0.2	0.85	0.25

Tablo 4.12’de verilen karmaşıklık matrislerinin ROC eğrileri Şekil 4.19’da verilmiştir.



Şekil 4.19. DVM ROC alanlarının karşılaştırılması

4.1.6. Sınıflandırma Başarımlarının Karşılaştırılması

Özellik çıkarım yöntemlerine göre sınıflandırıcıların 10-kat çapraz doğrulamalı başarımlar oranları Tablo 4.13’te verilmiştir.

Tablo 4.13. Sınıflandırma Başarımlarının Karşılaştırılması

Sınıflandırıcı		ELM (%)	Toplu Sınıflandırıcı (%)		
			Karar Ağacı	YSA	DVM
Çoklu Özellik	LBP	80	80	75	77.5
	Histogram	75	90	77.5	82.5
	SIFT	57.5	77.5	72.5	80

4.2. Grafiksel Arayüzün Geliştirilmesi

Grafiksel Arayüz (Graphical User Interface-GUI), çalışmanın ana düşüncesinin ortaya koyulduğu ve yönetildiği birimdir. Mermer görüntülerinden eğitim kümelerinin oluşturulması ve bu eğitim kümeleri kullanılarak sınıflandırma algoritmalarının eğitilmesi, geliştirilen GUI aracılığıyla sağlanır. GUI özellik çıkarım yöntemlerinin ve sınıflandırıcıların kontrol edildiği

merkezi işlem birimidir. Görsel menülerden oluşan GUI, sınıflandırıcılar için deneysel test ortamını sağlamanın yanısıra endüstriyel kullanıma yönelik bir ortam sunar.

Tez çalışması kapsamında geliştirilen GUI birimi, mermer sınıflandırma işlemini genel geçer bir yapıya taşıyan yazılımsal bir üründür. Geliştirilen GUI yazılımı sayesinde mermer sınıflandırma alanında herhangi bir tecrübesi olmayan personel bile, sisteme tanıtılmış eğitim verilerini kullanarak mermer sınıflandırma işlemini yapabilmektedir. Sınıflandırıcıların test işlemini gerçekleştirmek için geliştirilen deneysel GUI birimi, kullanım yöntemi bakımından sanayinin ihtiyacına da cevap verecek şekilde tasarlanmıştır. Merkezi yönetim birimi olan GUI, özellik çıkarımı ve sınıflandırma işlemlerini gerçekleştiren, birbirleriyle bağlantılı alt birimlerden oluşur.

4.2.1. Özellik Çıkarımında Kullanılan Teknolojiler

Özellik çıkarım işlemi, eğitim kümesini oluşturmak için GUI üzerinden yönetilir. GUI Java programlama dili kullanılarak geliştirilmiştir. Ancak özellik çıkarım işlemini gerçekleştiren alt birimler Python tabanlıdır. Özellik çıkarım birimlerinde kullanılan teknolojiler aşağıda verilmiştir.

- **Matplotlib:** Bu teknoloji Python programlama dili için geliştirilmiş dizi tabanlı bir grafik oluşturma kütüphanesidir. Histogram ve Ölçekten Bağımsız Özellik Dönüşümü(Scale Invariant Feature Transform-SIFT) tabanlı özellik çıkarım birimlerinde görüntü histogramının grafiğini görselleştirmek için kullanılmıştır.
- **Skimage:** Python için geliştirilmiş bu kütüphaneyle eşdizimli matris(Co-Occurence) tabanlı özellikler elde edilir.
- **Opencv:** C tabanlı görüntü işleme kütüphanesidir. Java, Python, C# ve C++ gibi programlama dillerine derlenmiş kütüphanesi mevcuttur. Kameradan görüntü alma, görüntü ölçekleme, görüntü renk uzayı değiştirme ve SIFT tanımlayıcılarının tespiti işlemlerinde kullanılmıştır.
- **Scikit-Learn:** Python tabanlı makine öğrenme kütüphanesidir. Sınıflandırma, kümeleme, regresyon, karar ağacı gibi algoritmaların fonksiyonlar aracılığıyla kullanılmasını sağlar. Bu kütüphane uygulamada sınıflandırma için kullanılmamıştır. SIFT tabanlı özellik çıkarım biriminde veri depolamak ve görsel kelimeleri ölçeklendirmek için kullanılmıştır.

- **CSV Dosya Birimi:** Tablo tabanlı bir dosya sistemidir. Veriler tablodaki satırlar ve sütunlara kaydedilir. Temel düzeyde, eğitim kümelerinin tutulduğu veri tabanı olarak kullanılmıştır.

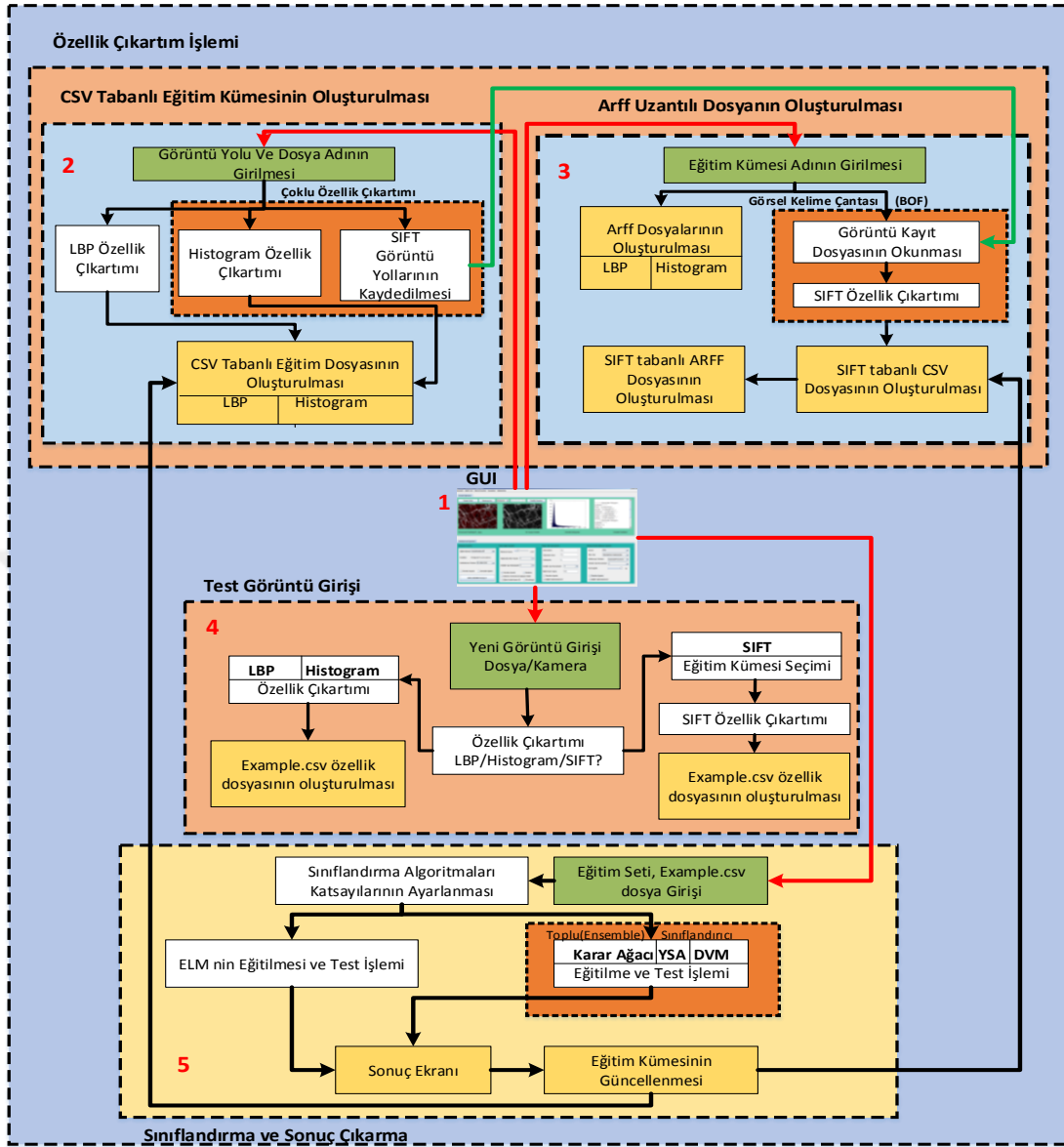
4.2.2. Grafiksel Arayüzde Kullanılan Teknolojiler

GUI son kullanıcının doğrudan etkileşimde olduğu birimdir. Bu birim genel olarak Java programlama dili kullanılarak geliştirilmiştir. GUI geliştirilirken kullanılan teknolojiler aşağıda verilmiştir.

- **Swing API:** Bu kütüphane Java tabanlı Grafiksel arabirimler geliştirmek için kullanılan bir kütüphanedir. Bu çalışmadaki GUI'nin tamamı Swing kütüphanesinin araç çantası (Toolbox) kullanılarak geliştirilmiştir.
- **IDE:** Uygulama geliştirilirken Eclipse editörü tercih edilmiştir. Uygulamanın Eclipse te geliştirilmesinin sebebi, istikrarlı, hata oranının düşük, açık kaynak kodlu olması ve Java ile Python derleyicisinin aynı platform üzerinde bulundurmasıdır.
- **Weka:** Bu kütüphane Java için geliştirilmiş açık kaynak bir makine öğrenme kütüphanesidir. Kendi adıyla kullanılan bir arayüze sahiptir. Sınıflandırma, kümeleme, regresyon analizi, karar ağacı gibi birçok algoritmayı barındırır. Sınıflandırma algoritmalarını oluşturma, eğitme ve test işlemi Weka tabanlı geliştirilmiştir.
- **Matlab:** Matris tabanlı özel bir programlama diline sahip, matematiksel hesaplamaların yapıldığı bir çatıdır. GUI ve Weka'dan elde edilen verileri anlamlı grafiklere dönüştürme işleminde kullanılmıştır.

4.2.3. GUI Çalışma Mimarisi

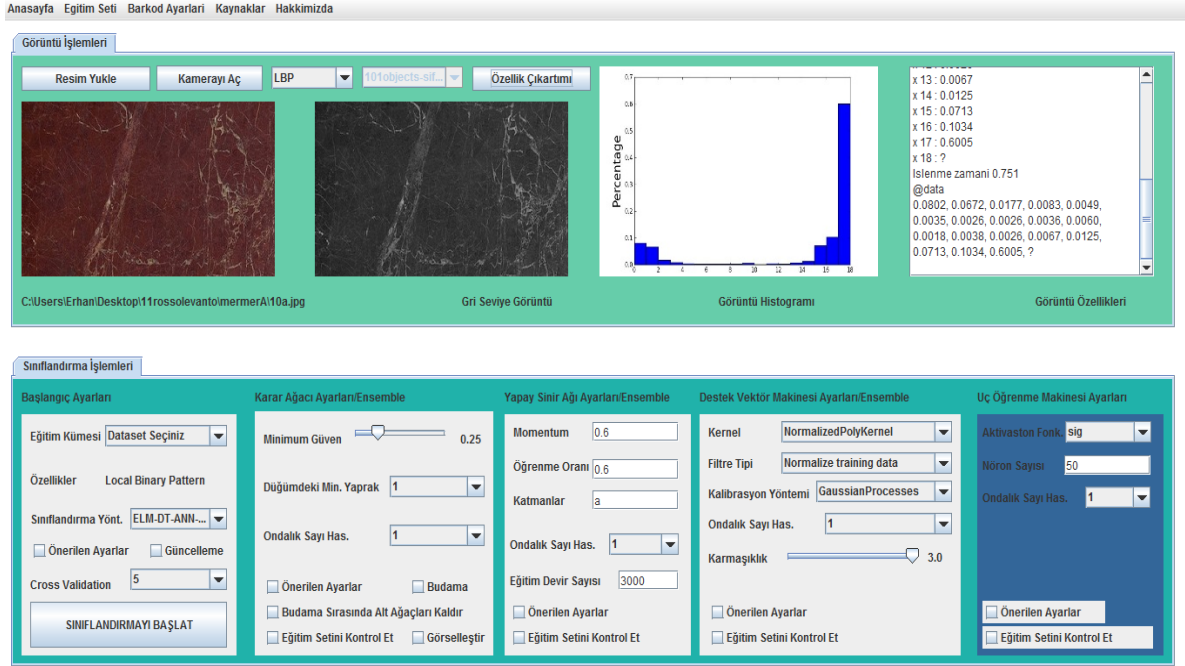
GUI alt birimler tarafından yapılan işlerin kullanıcıyla etkileştirildiği birimdir. Alt birimler ayrı ayrı bölümler halinde geliştirilmiştir. Böylece sistem üzerinde yapılacak geliştirme ve bakım maliyetleri daha düşük olmaktadır. GUI mimarisinin, her bir alt birimi numaralandırılarak Şekil 4.20'de verilmiştir.



Şekil 4.20. GUI akış şeması

Bu tez çalışmasında, sistemin tamamı GUI üzerinden yönetilebilir. GUI kullanıcıyla görsel iletişimin yanı sıra alt birimler arasındaki iletişimi de sağlar. Alt birimler GUI üzerinden dosyalar ve değişkenler şeklinde tanımlanan hafıza birimleri üzerinden haberleşir. Birimlerin çalışma yöntemleri adım adım aşağıda açıklanmıştır.

1) GUI: Uygulamanın merkezi işlem birimi olarak kabul edilir. Sadece alt birimleri yönetmenin yanı sıra kendi üzerinde bulunan sanal hafıza birimleriyle akış kontrolü sağlar. GUI başlangıç ekranı Şekil 4.21’de verilmiştir.



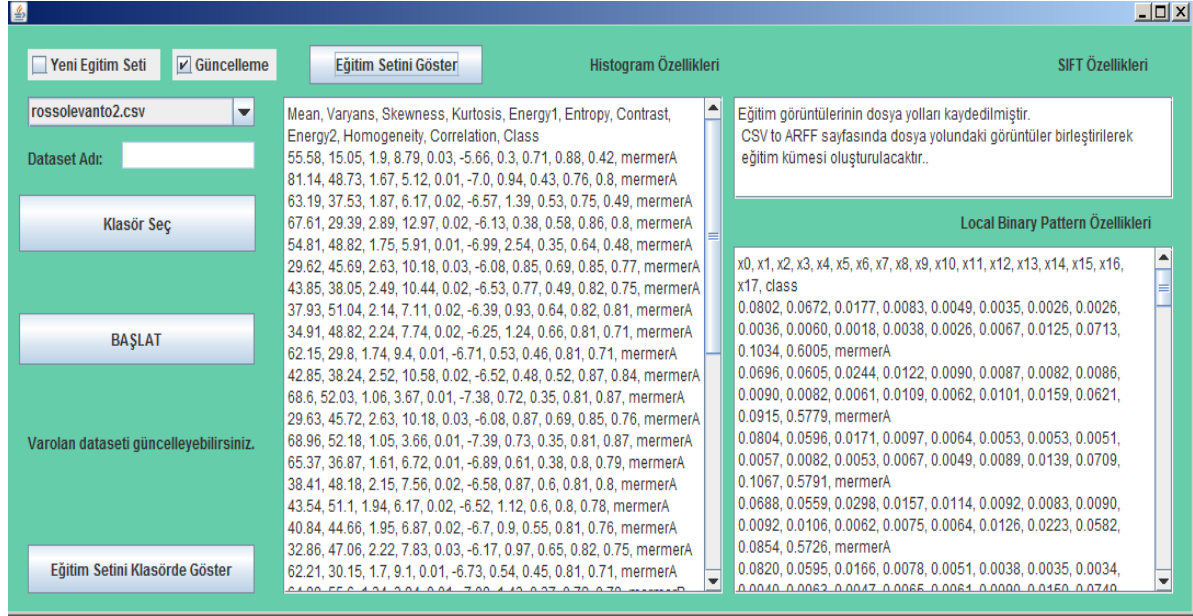
Şekil 4.21. GUI ana sayfa

Şekil 4.21’deki görüntü işlemleri bölümünde, kameradan veya hafızadan görüntü yükleme işlemi gerçekleştirilebilir. Görüntü yüklendikten sonra Yerel İkili Örüntü (Local Binary Pattern-LBP), histogram veya SIFT tabanlı özellik çıkarımı yapılması için seçim yapılır. Yapılan özellik seçimine göre, test görüntüsünün özellikleri sınıflandırma algoritmaları için uygun dosya biçimine dönüştürülür.

2) CSV Tabanlı Eğitim Kümesinin Oluşturulması: Sisteme mermer görüntülerinin tanıtılması gerekmektedir. Bu işlem ikinci bölümde açıklanan özellik çıkarım işlemlerinin yapıldığı menüdür. Şekil 4.20’deki bütün işlemler GUI tarafından başlatılarak denetimi sağlanır. Bu alt birim görüntünün dosya yolunu ve özelliklerin kaydedileceği dosya adını parametre olarak alır.

Görüntülerin bulunduğu klasörün adı, görüntülerin sınıfı olarak kabul edilir. Dosya yolundaki görüntüler sırasıyla “*feature-lbp.py*” ve “*feature-hist.py*” adlı uygulamalarda okunarak, özellikleri “*csv*” uzantılı dosyalara kaydedilir. Bu uygulamalar ile sadece LBP ve histogram tabanlı özellikler elde edilir. SIFT tabanlı özellikler, kelime çantası modeline dönüştürülmesi gerektiğinden, bu menüde oluşturulamaz. Eğitim kümesindeki bütün elamanların, anahtar noktalarından oluşan SIFT tabanlı histogram üzerinden kümeleme yapılarak kelime çantası modeli oluşturulur. Dolayısıyla bütün sınıflara ait örneklerin aynı anda okunması gerekmektedir. Bu işlem bu menüde yapılamayacağından sınıflara ait görüntülerin

yolları bir sonraki menüde kullanılmak üzere dosyaya kaydedilir. Bu işlemlerin gerçekleştirildiği menü Şekil 4.22’de verilmiştir.



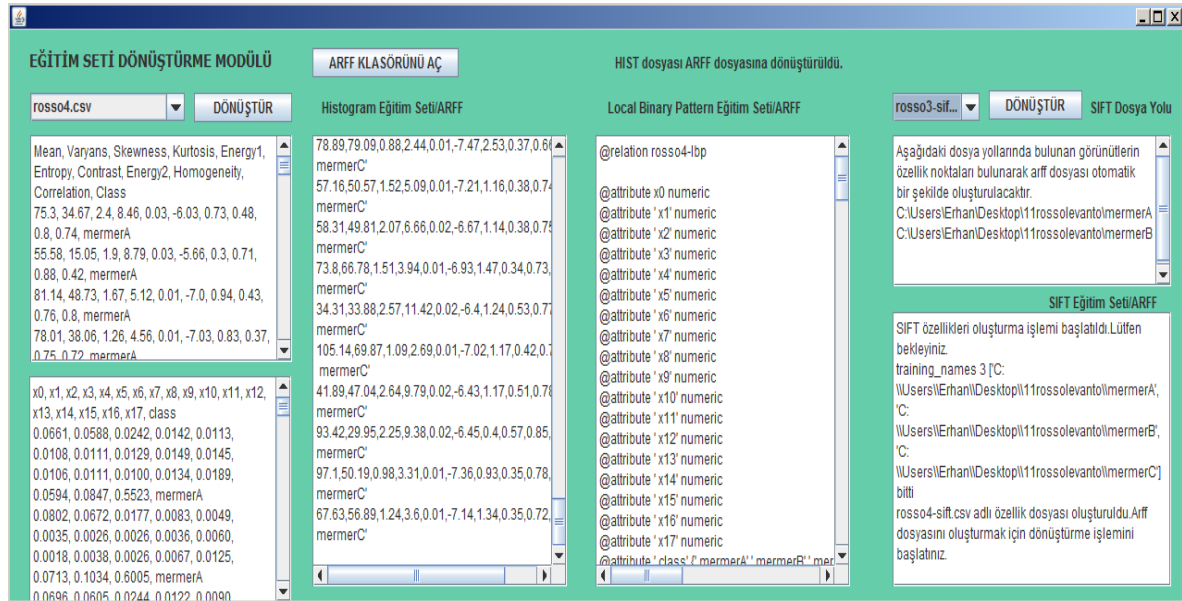
Şekil 4.22. CSV tabanlı özellik çıkarım menüsü

Şekil 4.22’de, LBP ve histogram tabanlı özelliklerin elde edilmesi gösterilmiştir. Uygulamada yeni bir eğitim seti oluşturma veya eğitim seti güncelleme işlemi yapılabilir. Yeni eğitim seti oluşturulduğunda eğitim seti adının girilmesi gerekir. Güncelleme yapıldığında ise istenilen eğitim seti seçilmelidir. “Klasör Seç” butonu ile görüntü yolu seçilir. “Başlat” butonuna basıldığında, seçili görüntü yolundaki görüntülerin özellikleri elde edilerek csv dosyasına kaydedilir. Daha sonra aynı eğitim kümesine farklı bir sınıf eklemek için güncelleme işlemi yapılması gerekmektedir. Bu işlemlerden sonra, “Başlat” butonuna tekrar basıldığında aynı eğitim setine yeni sınıflardan oluşan görüntülerin özellikleri eklenerek güncellenecektir. Aynı zamanda SIFT için tutulan dosya yollarının bulunduğu dosyada güncellenecektir. “Eğitim Setini Göster” butonuna basıldığında ise eğitim setinin tamamı okunarak kullanıcıya gösterilir.

3) Arff Uzantılı Dosyanın Oluşturulması: Uygulamanın eğitim kümesini şekil yönünden tanınması için “arff” uzantılı dosya biçimine dönüştürülmesi gerekmektedir. Bu dosya tipi Weka tabanlı bir dosyadır. Doğrudan Weka kütüphanesinin fonksiyonları kullanılarak okunur. LBP, histogram ve SIFT tabanlı özellikler ayrı ayrı dosyalarda tutulur. Dolayısıyla bu dosyaların ayrı ayrı “arff” dosya tipine dönüştürülmesi gerekmektedir. CSV tabanlı eğitim kümesinin oluşturulması işleminde sadece LBP ve histogram tabanlı özellikler elde edilmiştir. SIFT

tabanlı eğitim kümesinin, “csv” dosyası bu bölümde oluşturulacaktır. SIFT özelliklerinin “csv” dosyası oluşturulduktan sonra “arff” dosya tipine dönüştürülecektir.

LBP ve histogram tabanlı “csv” dosyasının, “arff” dosyasına dönüştürülmesi ve SIFT özelliklerinin çıkarılarak “csv” dosyasının oluşturulması, paralel yapılır. Bu işlemler birbirinden bağımsız olduğundan Java’nın iş parçacığı (Thread) yapısı kullanılarak eşzamanlı çalıştırılır. Dolayısıyla işlem süresi kısalmır. Bu işlemlerin gerçekleştirildiği menü Şekil 4.23’te verilmiştir.

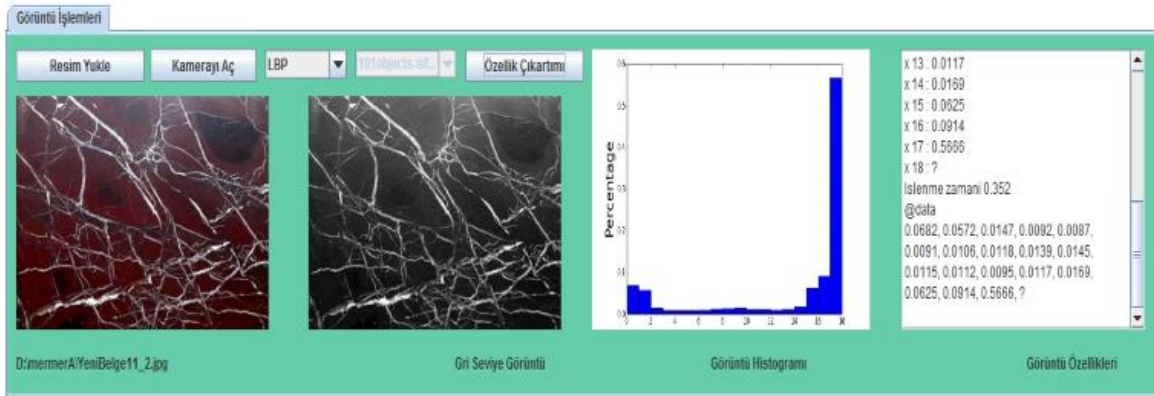


Şekil 4.23. CSV dosyasının arff dosyasına dönüştürülmesi

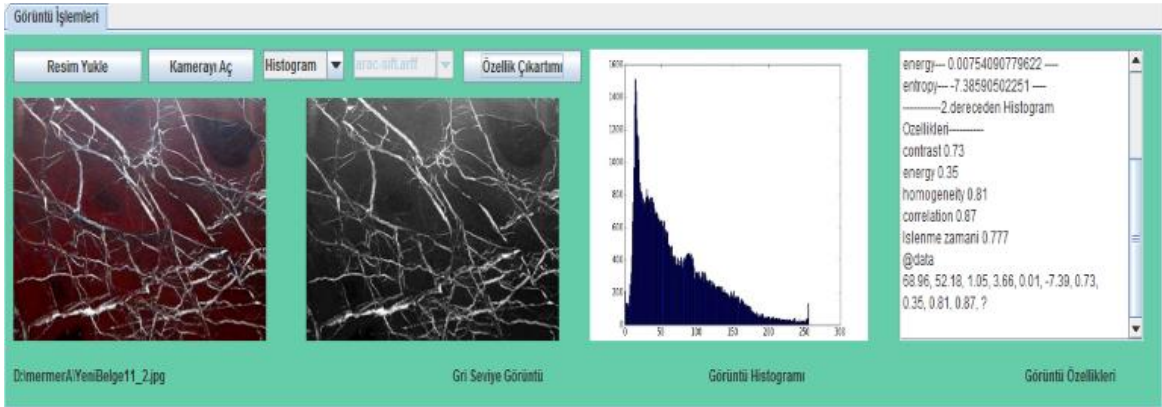
Şekil 4.23’te verilen menü açılırken sistemde kayıtlı “csv” dosyaları taranarak seçenek kutusuna otomatik bir şekilde yerleştirir. Kullanıcı “Dönüştür” butonuna basmadan önce seçenek kutusundan eğitim setini seçmelidir. Dönüştürme işlemi başladıktan sonra LBP ve Histogram dosyasının “arff” dosyası ile aynı eğitim setinin SIFT tabanlı csv dosyası oluşturulur. SIFT tabanlı “csv” dosyası oluştuktan sonra menünün sağındaki seçenek kutusunda aktifleşir. SIFT tabanlı “csv” dosyası, menünün sağındaki “Dönüştür” butonuyla “arff” dosyasına dönüştürülür.

4) Test Görüntü Girişi: Yeni görüntü girişi Şekil 4.20’de verilen akış şemasının dördüncü adımında gerçekleşir. Uygulamaya hafızadan veya kameradan görüntü girişi yapılabilir. Kameradan görüntü girişi deneysel olarak geliştirilmiştir. Yeni görüntü girişi yapıldıktan sonra üç tip özellik çıkarım işlemi uygulanabilir. LBP tabanlı özellik çıkarımında, gri seviye

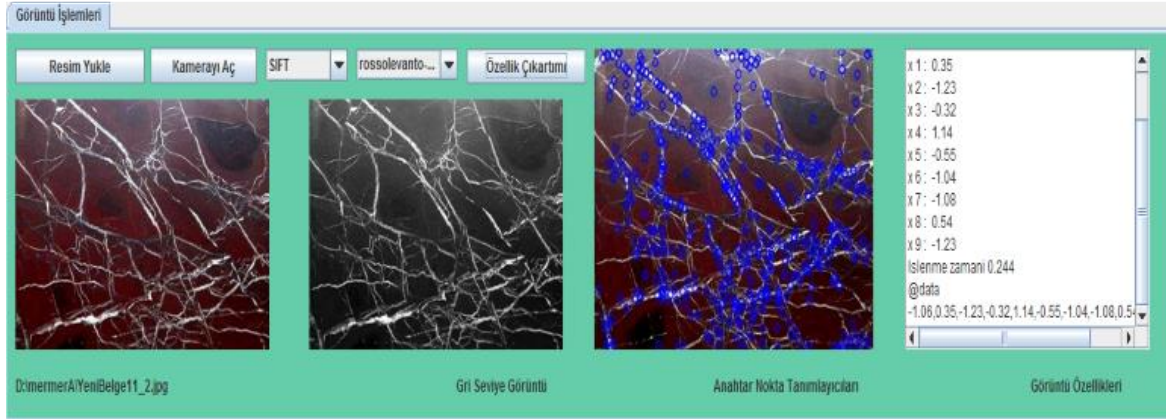
görüntüdeki tekdüze noktaların histogramı çıkarılır. Histogram tabanlı özellik çıkarım işleminde, görüntü gri seviyeye çevrildikten sonra histogramı elde edilir. Histogram üzerinden sayısal özellikler hesaplanır. SIFT tabanlı özellik çıkarımı için görsel kelime çantasının oluşturulması gerekmektedir. Bu görsel kelime çantası k-Ortalama ile kümelenecek on kümeye ayrılır. Yeni görüntünün özellikleri bu on kümeye olan vektörel uzaklıklarla ifade edilir. Bu yüzden test görüntüsünün SIFT tabanlı özellikleri için SIFT tabanlı eğitim setinin seçilmesi gerekir. Test görüntüsünün özellik çıkarım işlemi Şekil 4.24'te verilmiştir.



Şekil 4.24. LBP tabanlı özellik çıkarımı



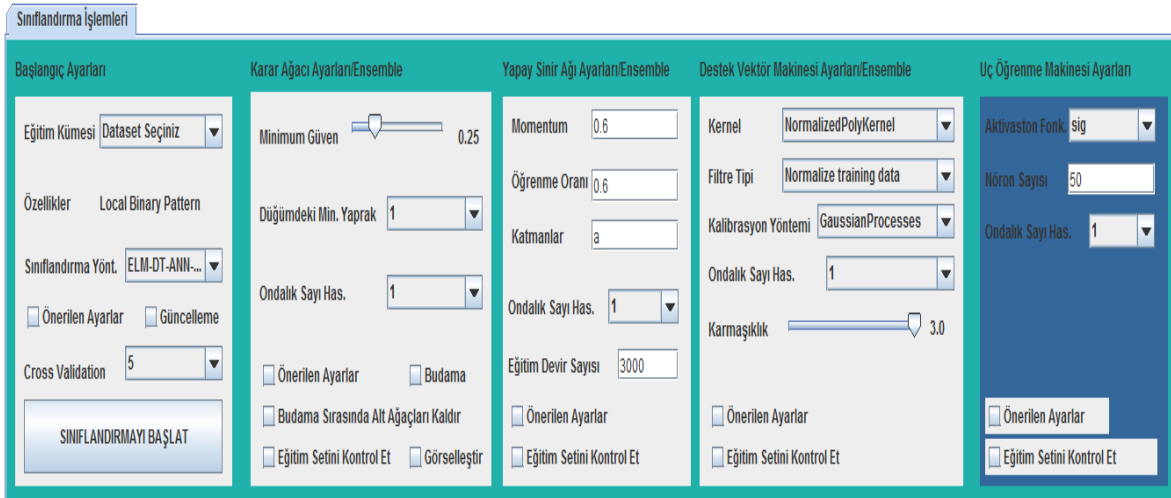
Şekil 4.25. Histogram tabanlı özellik çıkarımı



Şekil 4.26. SIFT tabanlı özellik çıkarımı

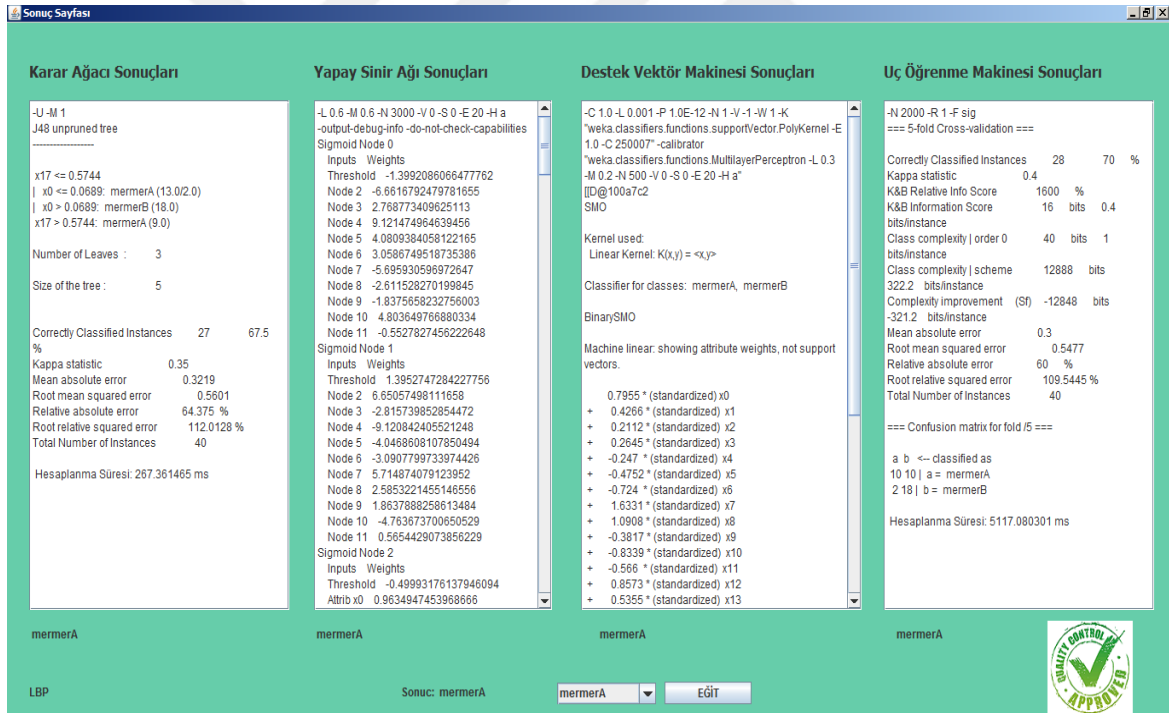
Şekil 4.24, Şekil 4.25 ve Şekil 4.26’da görüldüğü gibi test görüntüsünün özellik çıkarım işlemleri ayrı ayrı gerçekleştirilir. LBP, histogram ve SIFT tabanlı özellik çıkarımının birlikte denenmesinin amacı doğruluk derecesini arttırmaktır. Mermer doğal bir taştır. Bütün mermer yüzeyleri benzersizdir. Bu yüzden LBP ve histogram tabanlı özellikler sınıflandırma için elverişlidir.

5) Sınıflandırma ve Sonuç Çıkarma: Eğitim kümesi ve test kümesinin özellikleri kullanılarak, sınıflandırma işlemi gerçekleştirilir. Eğitim kümesi, sınıflandırma algoritmalarına verilerek öğrenme işlemi gerçekleştirilir. Sınıflandırma algoritmaları üzerinden test görüntüsünün özellikleri test edilir. Uç öğrenme makinesi, Karar ağacı, Yapay sinir ağı ve Destek vektör makinesi kullanılarak sınıflandırma işlemi gerçekleştirilir. Bu sınıflandırma algoritmaları paralel bir şekilde çalışarak sonuç üretir. Sınıflandırma işlemi ayarlarının yapıldığı menü Şekil 4.27’de verilmiştir.



Şekil 4.27. Sınıflandırma ayar menüsü

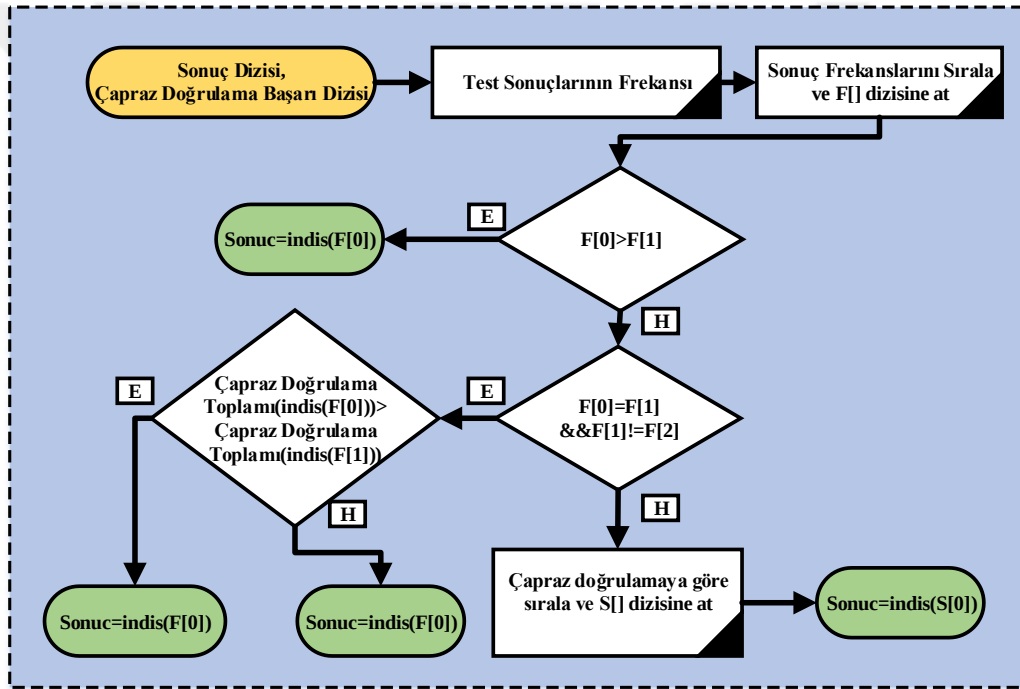
Şekil 4.27’de verildiği gibi sınıflandırma algoritmalarının ayarları aynı menü üzerinden yapılır. “Başlangıç Ayarları” kısmında eğitim kümesi seçim alanı mevcuttur. Bu alan LBP ve histogram tabanlı özelliklerin kullanım işleminde aktifleşir. SIFT tabanlı eğitim işlemi için, test görüntüsünün özellikleri elde edilirken kullanılan eğitim kümesi kullanılır. “Sınıflandırmayı Başlat” butonuna basıldığında girilen ayarlarla birlikte eğitim kümesi ve test görüntüsünün özellikleri sınıflandırma algoritmalarına verilir. Sınıflandırma algoritmaları işleme aynı anda başlayarak sonuç üretmeye çalışır. Uygulama bütün sınıflandırma algoritmalarının sonuçlarını ürettiğini algıladığında sonuç gösterim sayfasını başlatır. Sınıflandırma algoritmalarının “Önerilen Ayar” bölümü zaman kazancını göz ardı ederek en doğru sonucu elde etmek için otomatik bir şekilde sınıflandırma algoritmalarını ayarlar. Kullanıcının isteği doğrultusunda “Önerilen Ayarlar” işlemi sınıflandırma algoritmalarının tamamına veya istenildiği kadar algoritmaya uygulanabilir. Sonuç menüsü Şekil 4.28’de verildiği gibidir.



Şekil 4.28. Sonuç sayfası

Şekil 4.28’de verilen sonuç sayfasında sınıflandırma algoritmalarının fonksiyon değerleri ve test görüntüsünün sonuçları hesaplanmıştır. Sonuç sayfası, test görüntüsünün sınıfını, sınıflandırma algoritmalarının hesaplama süresini ve çapraz doğrulamalı başarı oranını gösterir.

Sınıflandırma algoritmaları test görüntüsü için farklı sonuçlar üretebilir. Sistem, sınıflandırma algoritmaları arasında yapılacak oylamaya göre genel sonuca karar verir. Bu oylama sonucunda hangi sınıf sayısı fazla ise uygulama ona göre seçim yapar. Uygulamada sadece bütün algoritmaların sonucu aynı çıkarsa, kalite kontrol logosu gösterilir. Kesin sonuç çıkarımı için farklı senaryolar mevcuttur. Birden fazla sınıfa ait eğitim kümesinde her bir sınıflandırma algoritması farklı sonuç verebilir. Bu durumda doğruluk oranı fazla olan algoritmanın verdiği sonuç dikkate alınır. Aynı zamanda bütün sınıflandırma algoritmalarının farklı sonuçları aynı başarıyla verdiği düşünülürse içlerinden birine öncelik atanabilir. Sonuç çıkarma işleminin algoritması Şekil 4.29’da verildiği gibidir.



Şekil 4.29. Sonuç karar algoritması

LBP özellikleriyle, sınıflandırma işlemi gerçekleştirildikten sonra kesin olmayan sonuçlar ortaya çıkabilir. Kesin sonuca ulaşmak için histogram ve SIFT algoritmalarına göre aynı işlemler tekrarlanır.

LBP, histogram ve SIFT tabanlı sınıflandırma sonuçları belirsizlik içeriyorsa bu durum uygulama tarafından kullanıcıya bildirilir. Kullanıcı, mermerin sınıfını çıkan sonuçlara göre yanlış olduğuna karar verirse, bu mermeri sisteme tanıtarak öğrenmesini sağlayabilir. Sonuç sayfasında, eğitim kümesinde bulunan herhangi bir sınıf test görüntüsüne atanabilmektedir. “EĞİT” butonuna basıldığında görüntünün özellikleri ile birlikte sınıf bilgisi eğitim kümesine

eklenir. Böylece aynı görüntü tekrar sisteme geldiğinde sınıflandırma algoritmaları bu görüntüyü tanıyarak doğru sonucu üretebilecektir.

4.3. Bölüm Değerlendirmesi

Bu bölümde GUI birim menülerinin çalışma yapısı anlatılmıştır. Uygulama eğitim seti oluşturma işlemi, yeni görüntü girişi ve sınıflandırma işlemlerini içerir. Önerilen bu çalışma, mermer gruplarında, sınıflandırma algoritmalarının performans testi ve ayarlamaların yapılabilmesi için deneysel ortam sunmaktadır. Son kullanıcı etkileşimi dikkate alındığından, olabildiğince otonom bir yapı geliştirilmeye çalışılmıştır.



5. SONUÇLAR

Mermer sınıflandırma işlemi uzman personel görüşüne dayanan öznel bir işlemdir. Sınıflandırma işleminin personel kaynaklı olması zaman ve ekonomik açıdan kayıplara neden olabilmektedir. Mermer ihracatı ülkemizin cari açığını azaltmada önemli bir etkiye sahiptir. 2017 senesinde işlenmiş ve blok mermer ihracatı yaklaşık 1.7 milyar dolardır. Bu rakamın 684 milyon doları işlenmiş mermer olarak ihraç edilir. İşlenmiş mermer, blok mermere göre daha yüksek kazanç sağlar. Mermerlerin yanlış sınıflandırılması durumunda iç satışlarda firmaya, ihracatta ise ülkemizin sektörel itibarına zarar verebilir. Belirtilen nedenlerden dolayı bu tez çalışması ile mermer sınıflandırma işlemini, öznel yapıdan çıkararak genel bir yapıya dönüştüren bir çalışma gerçekleştirilmiştir. Çalışma kapsamında LBP tabanlı ELM sınıflandırıcısı kullanılmıştır. ELM sınıflandırıcısının başarımını değerlendirmek için toplu sınıflandırıcı kullanılmıştır. LBP özellik çıkarımının başarısını ölçmek için ise çoklu özellik çıkarımından yararlanılmıştır. Toplu sınıflandırıcı Karar ağacı, YSA ve DVM'den oluşur. Çoklu özellik çıkarımında ise histogram ve SIFT tabanlı özellikler elde edilir.

Tez çalışması kapsamında sınıflandırma işlemi ve özellik çıkarımının gerçekleştirilmesi için deneysel test ortamına ihtiyaç duyulduğundan GUI geliştirilmiştir. Geliştirilen GUI deneysel test ortamı sağlamanın yanı sıra sanayinin ihtiyaçlarına da cevap verecek şekilde tasarlanmıştır. Mermer türlerine göre sınıflandırma kriterleri değişkenlik gösterdiği için, her işletme sınıflandırma yapacağı mermer türünü GUI üzerinden tanıtarak sistemi eğitebilmektedir.

Bu çalışma sonucunda işlenmiş mermerlerin sınıflandırılmasında, personel kaynaklı sınıflandırma hatalarının önüne geçebilmek amacıyla, akıllı sınıflandırıcılardan oluşan bir GUI ortaya çıkmıştır. LBP tabanlı özelliklere göre ELM ve Karar ağacı % 80, YSA % 72.5 ve DVM % 77.5 başarı oranına sahiptir. Bu başarı oranlarının artırılması için GUI üzerinde eğitim kümesinin geliştirilmesi sağlanmıştır. Test edilen mermerin sınıfı, özellikleri, sınıflandırma fonksiyonu ve sınıflandırma başarısı sonuç ekranında kullanıcıya gösterilmektedir.

Tez çalışması kapsamında geliştirilen GUI ilerleyen süreçte gömülü sistem üzerine uygulanabilir. Gömülü sistem üzerinde çalışan bir mermer sınıflandırma uygulaması sanayi kullanımı açısından daha verimli olması düşünülmektedir.

KAYNAKLAR

- [1] **Arıkan, M.**, 1962. Türkiyede Mermercilik, Bilimsel Madencilik Dergisi, Cilt 2,Sayı 7.
- [2] **Akkoyun, Ö., Toprak, Z.F.**, 2008, Mermer Blok Kalitelerinin Bulanık Mantık Yöntemi ile Belirlenmesi, BMYS2008 ESKİSEHIR
- [3] **Selver, M.A., Akay, O., Alim, F., Bardakcı, S. ve Olmez, M.**, 2011. An Automated Industrial Conveyor Belt System Using Image Processing And Hierarchical Clustering For Classifying Marble Slabs, Robotics And Computer-Integrated Manufacturing, **27**, 164–176.
- [4] **Alajarin, J.M., Delgado, J.D.L., Balibrea, L.M.T.**, 2005. Automatic System for Quality-Based Classification of Marble Textures, Ieee Transactions On Systems, Man, And Cybernetics, Vol.35, No.4.
- [5] **Selver, M.A., Akay, O., Ardalı, E., Yavuz, A.B., Onal, O. ve Ozden, G.**, 2009. Cascaded and Hierarchical Neural Networks for Classifying Surface Images of Marble Slabs, Ieee Transactions On Systems, Man, And Cybernetics, Vol.39, No.4.
- [6] **Wold, S.**, 1987, Principal Component Analysis, Chemometrics and Intelligent Laboratory Systems, **2**, 37-52
- [7] **Unser, M.**, 1986. Sum and Difference Histograms for Texture Classification, IEEE Transactions On Pattern Analysis And Machine Intelligence, Vol.8, No.1,
- [8] **Akçakoca, H., Uysal, Ö., Topal, I.**, 2003, Mermerlerin Kalite Kontrol Süreci Açısından Tekno-Mekanik Özelliklerinin Önemi, Türkiye IV Mermer Sempozyumu, Bildiriler Kitabı.
- [9] **Alajarin, J.M.**, 2004, Supervised Classification Of Marble Textures Using Support Vector Machines, Electronics Letters, Vol.40, No.11.
- [10] **Malinga, B., Raicu, D., Furst, J.**, 2006, Local vs. Global Histogram-Based Color Image Clustering, Depaul University School Of Computer Science Chicago.
- [11] **Quinlan, J.R.**, 1986, Induction of Decision Trees. Mach. Learn, 81–106
- [12] **Quinlan, J.R.**, 1993, C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers.
- [13] **Breiman L., Friedman J., Stone C.J., Olshen R.A.**, 1984, Classification-Regression(Wadsworth-Statistics-Probability)
- [14] **Bian X., Chen C., Du Q., Sheng Y.**, 2016, Extended Multi-Structure Local Binary Pattern For Highresolution Image Scene Classification, *IGARSS*

- [15] **Hoang V.T., Porebski A., Vandenbroucke N.**, 2016, Hamad D., LBP Parameter Tuning For Texture Analysis Of Lace Images, *IEEE IPAS*
- [16] **Jiang Y., Wang Y.**, 2016, Image Recognition and Classification Based on REM with LBP Feature, *WCICA*
- [17] **Kaya Y., Kaycı L.**, 2013, Kelebek Görüntülerin Sınıflandırılmasında Yeni Yerel İkili Örüntüler
- [18] **Nabiyev V., Gunay A.**, 2011, LBP Yardımıyla Görüntüdeki Kişinin Yaşının Bulunması, *Cankaya University Journal Of Science And Engineering*
- [19] **Lowe D. G.**, 2004, Distinctive Image Features from Scale-Invariant Keypoints, *International Journal Of Computer Vision*.
- [20] **Bakshi S., Das S., Mehrotra H., Sa P.K.**, Score Level Fusion Of Sift And Surf For Iris
- [21] **Melloni A., Bestagini P., Costanzo A., Barni M., Tagliasacchi M., Tubaro S.**, 2013, Attacking image classification based on Bag-of-Visual-Words, *WIFS GUANGZHOU*.
- [22] **Fouad M.M.M., Zawbaa H.M., El-Bendary N., Hassanien A.E.**, 2013, Automatic Nile Tilapia Fish Classification Approach using Machine Learning Techniques, *International Conference On Hybrid Intelligent Systems (His)*
- [23] **Hayashida M., Koyano H., Akutsu T.**, 2014, Measuring the Similarity of Protein Structures Using Image Local Feature Descriptors SIFT and SURF, *International Conference On Systems Biology (Lsb) Qingdao, China*
- [24] **Varol A.H.**, 2014, Görüntü Mozaikleme İle Panoramik Görüntü Üretimi, *Başkent Üniversitesi Fen Bilimleri Enstitüsü Yüksek Lisans Tezi*
- [25] **Ensafi S., Lu S., Kassim A.A., Tan C.L.**, 2014, A Bag of Words Based Approach for Classification of HEP-2 Cell Images, *Workshop On Pattern Recognition Techniques For Indirect Immuno- Fluorescence Images*
- [26] **Dave A., Dubey R.**, 2012, Do Humans Fixate on Interest Points, *International Conference On Pattern Recognition Tsukuba, Japan*
- [27] **Liu Y., Li C., Su B., Wang H.**, 2013, Evaluation of feature extraction methods for face recognition, *Sixth International Symposium On Computational Intelligence And Design*
- [28] **Heisterklaus I., Grabel P.**, 2015, Feature Detectors and Descriptors for Fisher Vectors, *IEEE 5th International Conference On Consumer Electronics Berlin (Icce-Berlin)*

- [29] **Khan N.Y., McCane B., Wyvill G.**, 2011, SIFT and SURF Performance Evaluation Against Various Image Deformations on Benchmark Dataset, International Conference On Digital Image Computing: Techniques And Applications
- [30] **Tomasev N., Pracner D., Brehar R.**, 2013, Object Recognition in WIKImage Data Based on Local Invariant Image Features, International Conference On Intelligent Computer Communication And Processing (ICCP)
- [31] **Kurt Z., Özkan K., Işık Ş.**, 2015, The Principal Component Analysis Method Based Descriptor for Visual Object Classification, International Journal Of Intelligent Systems And Applications In Engineering
- [32] **Huang, G.-B., Zhu, Q.-Y. and Siew, C.-K.**, 2006. Extreme learning machine: theory and applications, *Neurocomputing*, 70(1), 489–501.
- [33] **Cao F., Yang Z., Jiang M., Chen W., Ye Q.**, 2017, Spectral-Spatial Classification of Hyperspectral Image Using Extreme Learning Machine and Loopy Belief Propagation, IEEE International Conference On Internet Of Things
- [34] **Kolsch A., Afzal M.Z., Ebbeck M., Liwick M.**, 2017, Real-Time Document Image Classification using Deep CNN and Extreme Learning Machines, 14th IAPR
- [35] **Ergul U., Bilgin G.**, 2017, Hiperspektral Görüntülerin Melez Çekirdek Aşırı Öğrenme Makinesi ile Sınıflandırılması, IEEE
- [36] **Özçalıcı M.**, 2017, Aşırı Öğrenme Makineleri İle Hisse Senedi Fiyat Tahmini, Hacettepe Üniversitesi İktisadi Ve İdari Bilimler Fakültesi Dergisi Cilt 35/1
- [37] **Yildiz, O.T., Alpaydin, E.**, 2000, Linear Discriminant Trees, *ICML2K* Stanford Unı. USA.
- [38] **Yildiz, O.T., Alpaydin, E.**, 2000, Omnivariate Decision Trees, *IEEE Transactions On Neural Networks*, 12(6), 1539-1546
- [39] **Rokach, L., Maimon, O.**, 2008, Data mining with decision trees: theory and applications, World Scientific Pub Co Inc.
- [40] **Alpaydin, E.**, 2009, Introduction to Machine Learning, second edition, Mit Press.
- [41] **Öztemel, E.**, 2012, Yapay Sinir Ağları, Papatya Yayıncılık.
- [42] **Elkhetali, S.**, 2003, Application of ANN'S to the Classification of Radar Images, Telsiks.
- [43] **Vasundara, D.N., Seetha, M.**, 2016, Implementation of Hybrid RS-ANN for Spatial Image Classification, Ieee International Conference On Computational Intelligence And Computing Research

- [44] **Greenhow K.A., Johnson C.G.**, 2015, Regioned Downsample for ANN Image Classification , Sai Intelligent Systems Conference
- [45] **Periyasamy, N., Thamilselvan, P., Sathiaselvan J. G. R.**, 2015, A Comparative Study of ANN, K- Means and Adaboost Algorithms for image classification, IEEE Sponsored 2nd International Conference On Innovations In Information Embedded And Communication Systems.
- [46] **Mahmon, N.A., Ya'acob N.**, 2014, A Review on Classification of Satellite Image Using Artificial Neural Network (ANN), IEEE 5th Control And System Graduate Research Colloquium
- [47] **Kumar, J., Anand, R.S., Srivastava S.P.**, 2014, Flaws Classification using ANN for Radiographic Weld Images, International Conference On Signal Processing And Integrated Networks (Spin)
- [48] **Ahmed, S.A., Dey, S., Sarma, K.K.**, 2011, Image Texture Classification Using Artificial Neural Network (ANN), National Conference On Emerging Trends And Applications In Computer Science
- [49] **Park, S., Lee, H.S., Kim J.**, 2017, Seed Growing for Interactive Image Segmentation Using SVM Classification with Geodesic Distance, Electronics Letters Vol. 53 NO. 1 PP. 22–24
- [50] **Mankar, N.V., Khobragade, A., Raghuwanshi, M.M.**, 2016, Classification of Remote Sensing Image Using SVM Kernels, IEEE Sponsored World Conference On Futuristic Trends In Research And Innovation For Social Welfare (WCFTR'16)
- [51] **Sun, S., Zhong, P., Xiao, H., Liu, F., Wang, R.**, 2015, An Active Learning Method Based On Svm Classifier For Hyperspectral Images Classification, Workshop On Hyperspectral Image And Signal Processing: Evolution In Remote Sensing (WHISPERS)
- [52] **Rui, W., Zhili, L., Xing, Z.**, 2017, Recognition and Classification of Ultrasonic Aluminum Wire Joint Based on Image Morphology and C-SVM, 18th International Conference On Electronic Packaging Technology
- [53] **Liang, P., Li, W., Liu, D., Hu, J.**, 2017, Large-Scale Image Classification Using Fast SVM with Deep Quasi-Linear Kernel, International Joint Conference On Neural Networks (IJCNN)

- [54] **Srunitha, K., Padmavathi, S.**, 2016, Performance of SVM Classifier For Image Based Soil Classification, International Conference On SIGNAL Processing, Communication, Power And Embedded System (SCOPEs)
- [55] **Madroñal, D., Lazcano, R., Fabelo, H., Ortega, S., Callicó, G.M., Juárez, E., Sanz, C.**, 2016, Hyperspectral Image Classification Using A Parallel Implementation of The Linear SVM on A Massively Parallel Processor Array (MPPA) Platform, Conference On Design And Architectures For Signal And Image Processing (DASIP)



ÖZGEÇMİŞ

Erhan TURAN 1991 yılında Elazığ’ da doğdu. İlk, orta ve lise öğrenimini Elazığ’da tamamladı. 2009 yılında Mehmet Koloğlu Anadolu Lisesini bitirdikten sonra 2013 yılında Fırat Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği bölümünden mezun oldu. 2013 yılında Sanayi Bakanlığı teknogirişim sermayesi desteğiyle Havsan Robotik Ltd. Şti. adlı firmayı kurdu ve halen çalışmaktadır. 2015 yılının Ocak ayında Fırat Üniversitesi, Fen Bilimleri Enstitüsü, Mekatronik Mühendisliği Elektronik Ana Bilim dalında yüksek lisans eğitimine başladı.

