

**EKLER****EK-1 Etik Kurul Kararı**

T.C.  
SELÇUK ÜNİVERSİTESİ  
TIP FAKÜLTESİ DEKANLIĞI

**GİRİŞİMSSEL OLMAYAN KLİNİK ARAŞTIRMALAR ETİK KURULU KARARLARI**

Toplantı Sayısı: 2014/18

Toplantı Tarihi : 04.11.2014

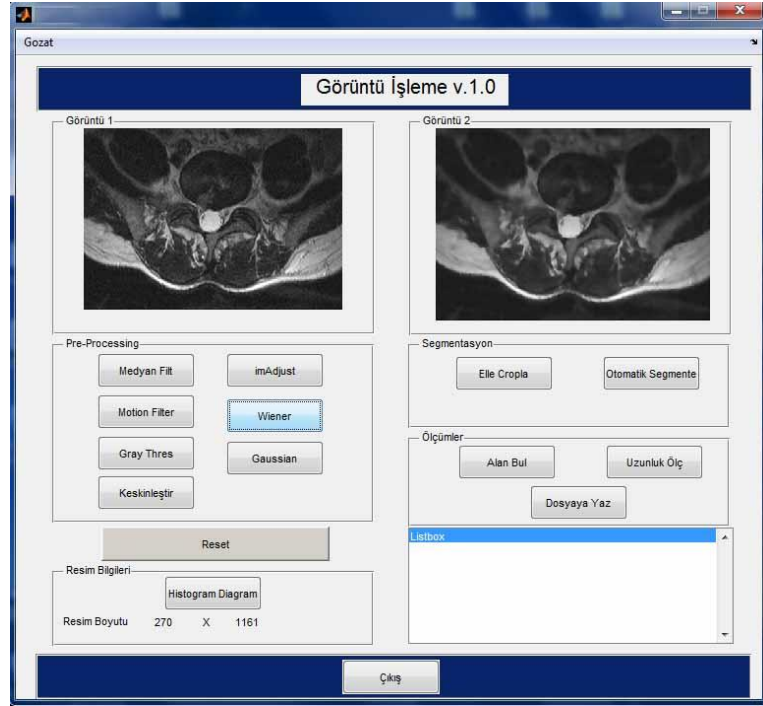
Karar Sayısı 2014/301 S.Ü. Teknoloji Fakültesi Öğretim Üyesi Yrd.Doç.Dr. H.Erdinç KOÇER'in, "Makine Öğrenmesi Yöntemleriyle Bel Bölgesi Rahatsızlıklarının Tanısı" adlı çalışmanın görüşülmesi, başlıklı araştırmasının değerlendirilme talebi ile ilgili 28.10.2014 tarihli dilekçesi ve ekleri görüşüldü.

Yapılan inceleme ve görüşmelerden sonra; Yrd.Doç.Dr. H.Erdinç KOÇER'in, "Makine Öğrenmesi Yöntemleriyle Bel Bölgesi Rahatsızlıklarının Tanısı" adlı araştırmanın kabulüne oy birliği ile karar verildi.

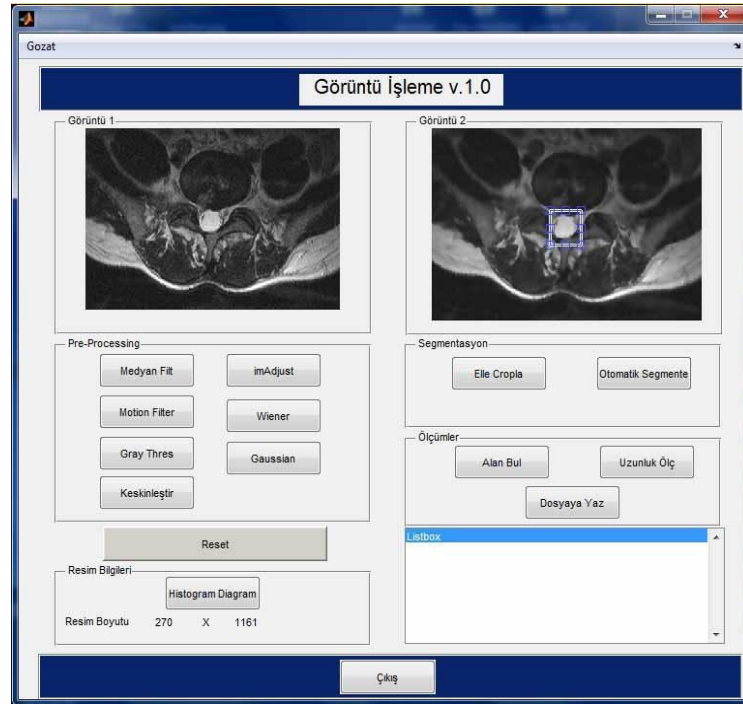
ASLI GİBİDİR  
04/11/2014  
  
Mehmet Kılıç  
Sekreterya  
S. Ü. D. E. K.

## EK-2 Tezde Kullanılan Yazılımların Ekran Görüntüleri

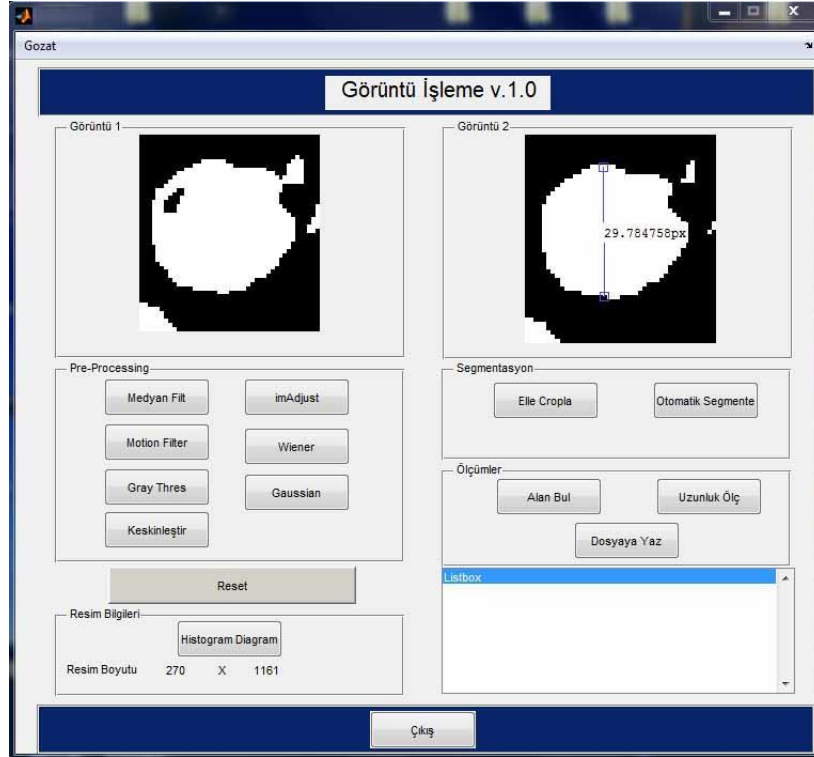
Tez çalışmasının ön işlem aşamasında MR görüntülerinin iyileştirilmesi (histogram eşitleme ve medyan filtreleme) ve kesme işlemleri için MATLAB GUI ortamında geliştirilen yazılımın ekran görüntüleri (Şekil 1, Şekil 2, Şekil 3) aşağıda verilmiştir.



Şekil 1. Ön işlem aşaması ana ekranı

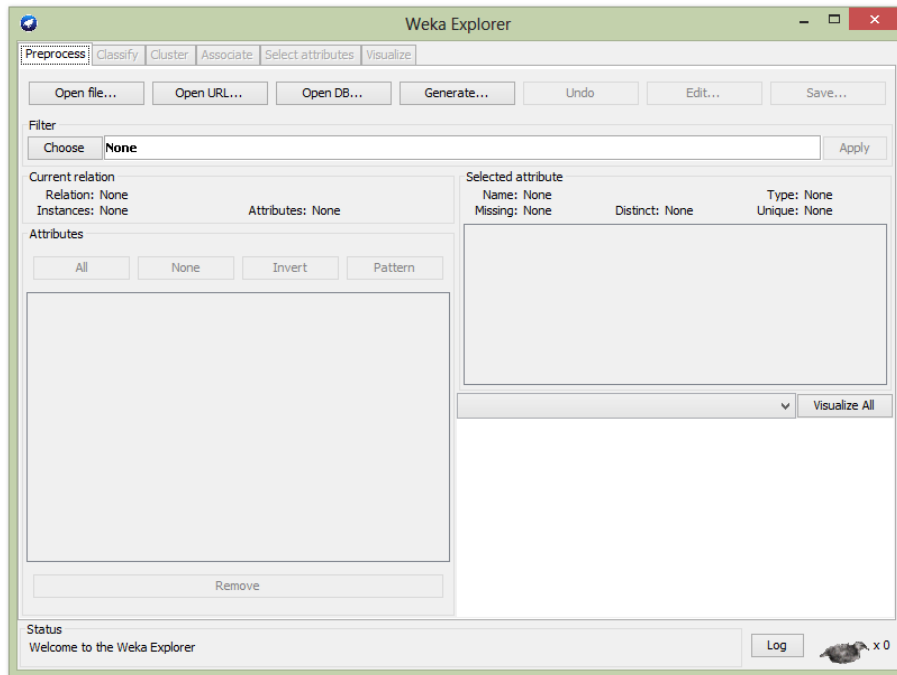


Şekil 2. İlgili bölgenin kesilmesi ekran görüntüsü

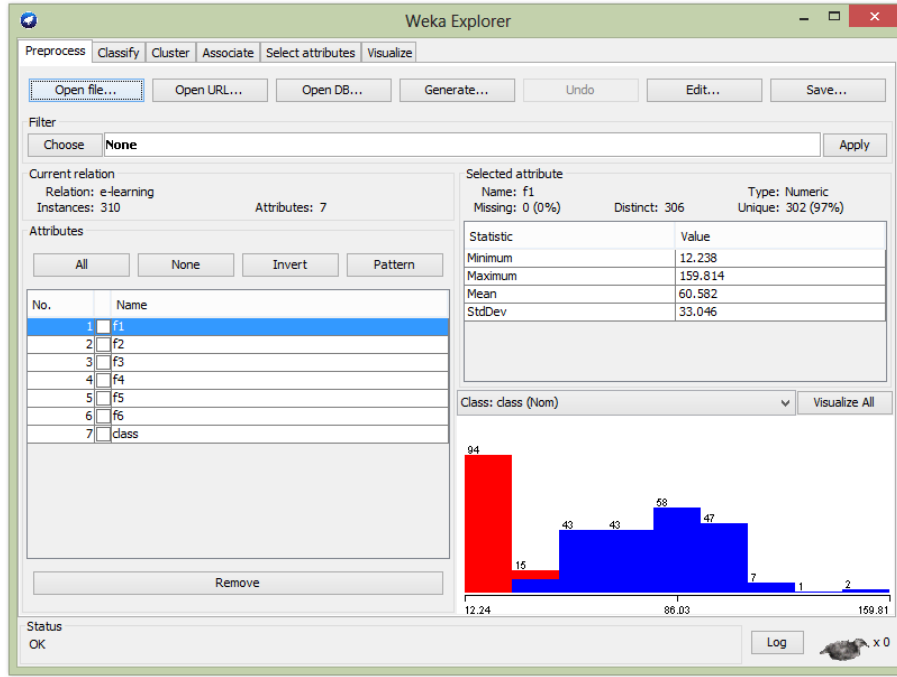


Şekil 3. İlgili bölge üzerinde ölçüm işlemi

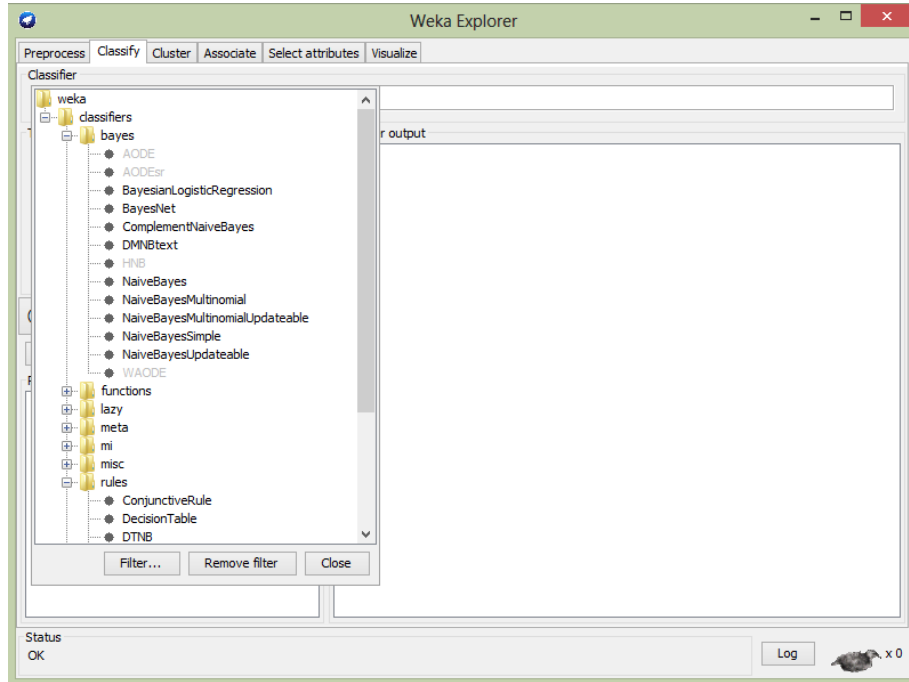
Tez çalışmasının sınıflandırma aşamasında özellik çıkarımından elde edilen özellik vektörleri, Waikato Üniversitesinde geliştirilmiş açık kaynak kodlu WEKA makine öğrenmesi yazılımında sınıflandırılmıştır. Sınıflandırma işleminin adımlarını gösteren ekran görüntüleri (Şekil 4, Şekil 5, Şekil 6, Şekil 7) aşağıda verilmiştir.



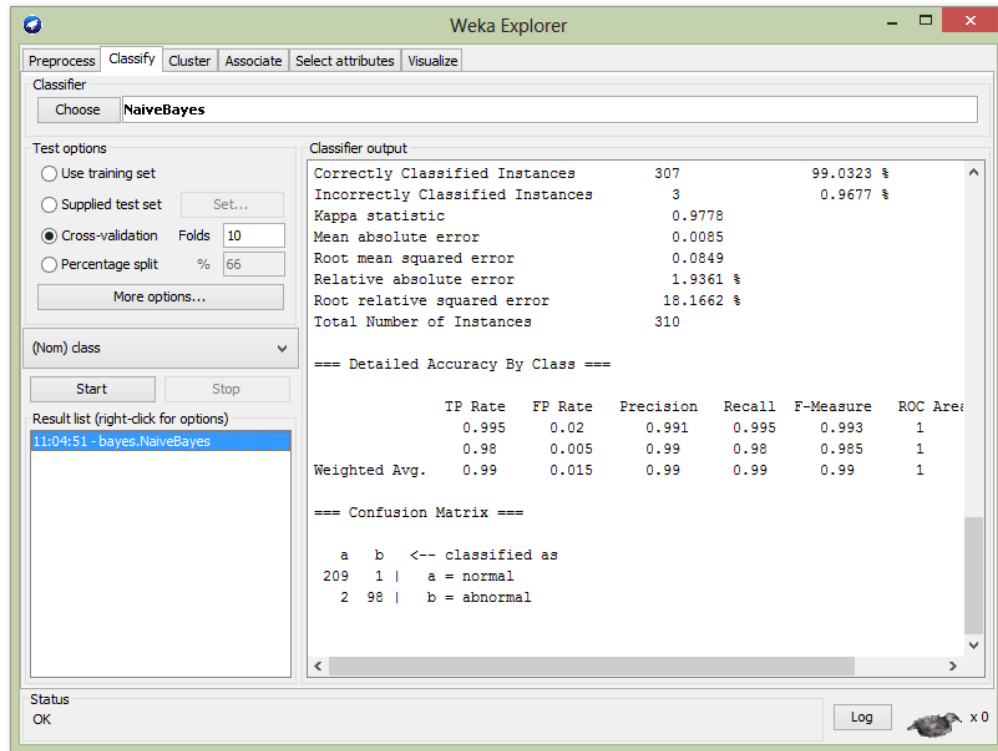
Şekil 4. Weka yazılımı ana ekran görüntüsü



Şekil 5. Açılan dosyanın istatistiki bilgiler ekran görüntüsü



Şekil 6. Weka yazılımı kütüphanesinde yer alan sınıflama algoritmaları



Şekil 7. Naive bayes algoritması ile sınıflama sonuçları

### **EK-3 Tezde Geliştirilen Ortalama Kayma Kümeleme Tabanlı Veri Ağırlıklandırma Yönteminin Matlab Kodları**

```

clc;

clear all;

load yavuz_3_class_full;

dataPts=yavuz_3_class_full(:,1:11);

dataPts=dataPts';

bandWidth=5.75;

[clustCent,data2cluster,cluster2dataCell] = MeanShiftCluster(dataPts,bandWidth);

function [clustCent,data2cluster,cluster2dataCell] =
MeanShiftCluster(dataPts,bandWidth,plotFlag);
%perform MeanShift Clustering of data using a flat kernel
%
% ---INPUT---
% dataPts      - input data, (numDim x numPts)
% bandWidth    - is bandwidth parameter (scalar)
% plotFlag     - display output if 2 or 3 D   (logical)
% ---OUTPUT---
% clustCent    - is locations of cluster centers (numDim x numClust)
% data2cluster - for every data point which cluster it belongs to (numPts)
% cluster2dataCell - for every cluster which points are in it (numClust)
%
% Bryan Feldman 02/24/06
% MeanShift first appears in
% K. Funkunaga and L.D. Hosteler, "The Estimation of the Gradient of a
% Density Function, with Applications in Pattern Recognition"

%*** Check input ****
if nargin < 2
    error('no bandwidth specified')
end

if nargin < 3
    plotFlag = true;
    plotFlag = false;
end

```

```

%**** Initialize stuff ***
[numDim,numPts] = size(dataPts);
numClust      = 0;
bandSq       = bandWidth^2;
initPtInds   = 1:numPts;
maxPos       = max(dataPts,[],2);           %biggest size in each dimension
minPos       = min(dataPts,[],2);           %smallest size in each dimension
boundBox     = maxPos-minPos;               %bounding box size
sizeSpace    = norm(boundBox);              %indicator of size of data space
stopThresh   = 1e-3*bandWidth;             %when mean has converged
clustCent    = [];                         %center of clust
beenVisitedFlag = zeros(1,numPts,'uint8'); %track if a points been seen already
numInitPts   = numPts;                     %number of points to posibaly use as
initialization points
clusterVotes = zeros(1,numPts,'uint16');    %used to resolve conflicts on cluster
membership

while numInitPts

    tempInd    = ceil( (numInitPts-1e-6)*rand); %pick a random seed point
    stInd      = initPtInds(tempInd);           %use this point as start of mean
    myMean     = dataPts(:,stInd);              % intilize mean to this points
location
    myMembers  = [];                           % points that will get added to this cluster

    thisClusterVotes = zeros(1,numPts,'uint16'); %used to resolve conflicts on
cluster membership

    while 1 %loop untill convergence

        sqDistToAll = sum((repmat(myMean,1,numPts) - dataPts).^2); %dist squared
from mean to all points still active
        inInds      = find(sqDistToAll < bandSq); %points within bandWidth
        thisClusterVotes(inInds) = thisClusterVotes(inInds)+1; %add a vote for all the in
points belonging to this cluster

        myOldMean = myMean; %save the old mean
        myMean    = mean(dataPts(:,inInds),2); %compute the new mean
        myMembers = [myMembers inInds]; %add any point within
bandWidth to the cluster
        beenVisitedFlag(myMembers) = 1; %mark that these points have
been visited

    %*** plot stuff ****
    if plotFlag
        figure(12345),clf,hold on
        if numDim == 2
            plot(dataPts(1,:),dataPts(2:),'')

```

```

        plot(dataPts(1,myMembers),dataPts(2,myMembers),'ys')
        plot(myMean(1),myMean(2),'go')
        plot(myOldMean(1),myOldMean(2),'rd')
        pause
    end
end

% **** if mean doesn't move much stop this cluster ***
if norm(myMean-myOldMean) < stopThresh

    %check for merge possibilities
    mergeWith = 0;
    for cN = 1:numClust
        distToOther = norm(myMean-clustCent(:,cN)); %distance from posible
new clust max to old clust max
        if distToOther < bandWidth/2 %if its within bandwidth/2 merge
new and old
            mergeWith = cN;
            break;
        end
    end

    if mergeWith > 0 % something to merge
        clustCent(:,mergeWith) = 0.5*(myMean+clustCent(:,mergeWith));
%record the max as the mean of the two merged (I know biased towards new ones)
        %clustMembsCell{mergeWith} = unique([clustMembsCell{mergeWith}
myMembers]); %record which points inside
        clusterVotes(mergeWith,:) = clusterVotes(mergeWith,:) + thisClusterVotes;
%add these votes to the merged cluster
    else %its a new cluster
        numClust = numClust+1; %increment clusters
        clustCent(:,numClust) = myMean; %record the mean
        %clustMembsCell{numClust} = myMembers; %store my
members
        clusterVotes(numClust,:) = thisClusterVotes;
    end

    break;
end

end

initPtInds = find(beenVisitedFlag == 0); %we can initialize with any of the
points not yet visited
numInitPts = length(initPtInds); %number of active points in set

end

```



```

[val,data2cluster] = max(clusterVotes,[],1);           %a point belongs to the cluster
with the most votes

%*** If they want the cluster2data cell find it for them
if nargout > 2
    cluster2dataCell = cell(numClust,1);
    for cN = 1:numClust
        myMembers = find(data2cluster == cN);
        cluster2dataCell{cN} = myMembers;
    end
end
end

```

## KİŞİSEL BİLGİLER

**Adı Soyadı** : Yavuz ÜNAL  
**Uyruğu** : T.C  
**Doğum Yeri ve Tarihi** : Tokat/Erbaa 05.02.1980  
**Telefon** : 03582123341  
**Faks** : 03582180104  
**e-mail** : yavuz.unal@amasya.edu.tr

## EĞİTİM

Derece	Adı, İlçe, İl	Bitirme Yılı
Lise	: Coşkun Önder Lisesi , Erbaa, Tokat	1997
Üniversite	: Selçuk Üniversitesi Teknik Eğitim Fakültesi	2002
Yüksek Lisans	: Selçuk Üniversitesi Elektronik ve Bilgisayar Eğt.	2006
Doktora	: Selçuk Üniversitesi Bilgisayar Müh.	2015

## İŞ DENEYİMLERİ

Yıl	Kurum	Görevi
2002-2012	Selçuk Üniversitesi Teknik Eğitim Fak.	Araştırma Gör.
2012-...	Amasya Üniversitesi Eğitim Fakültesi	Öğretim Gör.

## UZMANLIK ALANI

Görüntü İşleme, Sınıflama, Veri Ağırlıklandırma

## YABANCI DİLLER

İngilizce

## BELİRTMEK İSTEĞİNİZ DİĞER ÖZELLİKLER

## YAYINLAR

Yavuz ÜNAL, Kemal POLAT, H. Erdinç KOÇER , "Pairwise FCM based feature weighting for improved classification of vertebral column disorders", [Computers in Biology and Medicine](#), Vol:46, pp.61-70, 2014. (Doktora tezinden yapılmıştır)

Yavuz ÜNAL, Kemal POLAT, H. Erdinç KOÇER “Detection of Abnormalities in Lumbar Discs from Clinical Lumbar MRI with Hybrid Models”, *Applied Soft Computing*, 2015. (Doktora tezinden yapılmıştır)

Hakan IŞIK, Mustafa Cihat AVUNDUK, Yavuz ÜNAL.,(2010) “Design and Realization of an Improved Patalogy Equipment by Giving a Frozen Ability”, *E-Journal of New World Sciences Academy*, Volume: 5, Number: 2, Article Number: 1A0078, pp.215-225 (Yüksek lisans tezinden yapılmıştır)

Yavuz ÜNAL, Erdiñ KOÇER, H.E. AKKURT “A comparison of feature extraction techniques for diagnosis of lumbar intervertebral degenerative disc disease” Innovations in Intelligent Systems and Applications (INISTA), 2011 International Symposium on INnovations in Intelligent SysTems and Applications (Doktora tezinden yapılmıştır)

Y. Unal, H. E. Kocer and H. E. Akkurt, Automatic diagnosis of intervertebral degenerative disk disease using Artificial Neural Networks, 6th International Advanced Technologies Symposium (IATS'11), 16-18 May 2011, Elazığ, Turkey (Doktora tezinden yapılmıştır)

Y. Unal, E. Kocer, "Diagnosis of pathology on the vertebral column with backpropagation and Naive Bayes classifier", The International Conference on Technological Advances in Electrical, Electronics and Computer Engineering (TAEECE2013), 9-13 may. Konya/TURKEY. (Doktora tezinden yapılmıştır)