References

[1] Banka, R., Nourbakhsh, F., 2010. EXTRACTION OF SIGNATURE AND HANDWRITTEN REGIONS FROM OFFICIAL BINARY DOCUMENT IMAGES.

[2] Basavaraj, L., Samuel, R.S., 2009. Offline-line Signature Verification and Recognition: An Approach Based on Four Speed Stroke Angle. Int. J. Recent Trends Eng. 2.

[3] Bebis, G., Deaconu, T., Georgiopoulos, M., 1999. Fingerprint identification using delaunay triangulation, in: Information Intelligence and Systems, 1999. Proceedings. 1999 International Conference on. IEEE, pp. 452–459.

[4] Bhattacharyya, D., Kim, T., 2010. Design of artificial neural network for handwritten signature recognition. Int. J. Comput. Commun. 4, 59–66.

[5] Coetzer, J., 2005. Off-line signature verification. Stellenbosch: University of Stellenbosch.

[6] Cüceloğlu, İ., Oğul, H., 2014. Detecting handwritten signatures in scanned documents, in: Proceedings of the 19th Computer Vision Winter Workshop. pp. 89–94.

[7] Gautam, C.M., Sharma, S., Verma, J.S., 2012. A GUI for Automatic Extraction of Signature from Image Document. Int. J. Comput. Appl. 54.

[8] Gonzalez, R.C., Woods, R.E., 2002. Digital image processing. Pearson Education, Delhi, India.

[9] Gupta, C.S., Dixit, U.D., 2015. A REVIEW ON SIGNATURE DETECTION AND SIGNATURE BASED DOCUMENT IMAGE RETRIEVAL 4.

[10] Hafemann, L.G., Sabourin, R., Oliveira, L.S., 2015a. Offline handwritten signature verification-literature review.

[11] Kalera, M.K., Srihari, S., Xu, A., 2004a. Offline signature verification and identification using distance statistics. Int. J. Pattern Recognit. Artif. Intell. 18, 1339–1360.

[12] Karouni, A., Daya, B., Bahlak, S., 2011. Offline signature recognition using neural networks approach. Procedia Comput. Sci. 3, 155–161. .2010.12.027

[13] Khuwaja, G.A., Laghari, M.S., 2011. Offline handwritten signature recognition. World Acad. Sci. Eng. Technol. 59, 1300–1303.

[14] Kiani, V., Pourreza, R., Pourreza, H.R., 2009. Offline signature verification using local radon transform and support vector machines. Int. J. Image Process. 3, 184–194.

[15] Kumar, P., Singh, S., Garg, A., Prabhat, N., 2013a. Hand written signature recognition & verification using neural network. Int. J. Adv. Res. Comput. Sci. Softw. Eng. 3.

[16] Mandle, P., Shaligram, V., 2015. A Novel Image Matching Technique using SIFT and SURF 4.

[17] Mehra, R., Gangwar, R.C., 2014. A Survey: Enhanced Offline Signature Recognition Using Neuro-Fuzzy and SURF Features Techniques Vol 5(3), 4350–4353.

[18] Nguyen, V., Blumenstein, M., Muthukkumarasamy, V., Leedham, G., 2007. Off-line signature verification using enhanced modified direction features in conjunction with neural classifiers and support vector machines, in: Ninth International Conference on Document Analysis and Recognition (ICDAR 2007). IEEE, pp. 734–738.

[19] Özgündüz, E., Şentürk, T., Karslıgil, M.E., 2005. Off-line signature verification and recognition by support vector machine, in: Signal Processing Conference, 2005 13th European. IEEE, pp. 1–4.

[20] Piyush Shanker, A., Rajagopalan, A.N., 2007. Off-line signature verification using DTW. Pattern Recognit. Lett. 28, 1407–1414.

[21] Prashanth, C.R., Raja, K.B., Venugopal, K.R., Patnaik, L.M., 2012. DWT based Offline Signature Verification using Angular Features. Int. J. Comput. Appl. 52.

[22] Purohit, N. (2010). *OFFLINE HANDWRITTEN SIGNATURE VERIFICATION* .

[23] Pushpalatha, N., Gautam, A., 2014. Offline signature Verification using spatial domain feature sets and support vector machine. Int. J. Emerg. Technol. Adv. Eng. 4, 544.

[24] Ramachandra, 2009. *Signature Verification using Graph Matching*, International Journal of Recent Trends in Engineering.

[25] Shirdhonkar, M.S., Kokare, M.B., 2010. Discrimination between printed and handwritten text in documents. IJCA Spec. Issue On.

[26] Singh, N., Kaushal, S., 2015. INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY OFF-LINE SIGNATURE RECOGNITION USING MODIFIED NEURAL NETWORKS APPROACH.

[27] Sisodia, K., Anand, S.M., 2009. Off-line handwritten signature verification using artificial neural network classifier. Int. J. Recent Trends Eng. 2, 205–207.

[28] Srinivasan, H., Srihari, S., 2009. Signature-Based Retrieval of Scanned Documents Using Conditional Random Fields, in: Argamon, S., Howard, N. (Eds.), Computational Methods for Counterterrorism. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 17–32.

[29] Srinivasan, H., Srihari, S.N., Beal, M.J., 2006. Machine learning for signature verification, in: Computer Vision, Graphics and Image Processing. Springer, pp. 761–775.

[30] T.S. enturk. E. O¨ zgunduz. and E. Karshgil, "Handwritten Signature Verification Using Image Invariants and Dynamic Features," *Proceedings of the 13*th *European Signal Processing Conference EUSIPCO 2005, Antalya Turkey*, 4th-8th September, 2005.

[31] Zimmer, A., Ling, L.L., 2003. A hybrid on/off line handwritten signature verification system, in: Document Analysis and Recognition, 2003. Proceedings. Seventh International Conference on. IEEE, pp. 424–428.

APPENDIX

Some Sample Codes

Code for Thinning

```
continue_it = 1;
        while continue_it
            BW_old=BW;
            BW_del=zeros(size(BW));
            for i=2:size(BW,1)-1
                for j = 2:size(BW,2)-1
                    P = [BW(i,j) BW(i-1,j) BW(i-1,j+1) BW(i,j+1)
BW(i+1,j+1) BW(i+1,j) BW(i+1,j-1) BW(i,j-1) BW(i-1,j-1) BW(i-1,j)];
                        if P(2)*P(4)*P(6)==0 && P(4)*P(6)*P(8)==0 &&
sum(P(2:end-1))<=6 && sum(P(2:end-1)) >=2
                            A = 0;
                            for k = 2:size(P(:),1)-1
                                if P(k) == 0 && P(k+1)==1
                                    A = A+1;
                                end%if
                            end%for
                                if (A==1)
                                    BW_del(i,j)=1;
                                end%if
                        end%if
                end%for
            end%for



            BW(find(BW_del==1))=0;


            for i=2:size(BW,1)-1
                for j = 2:size(BW,2)-1
                    P = [BW(i,j) BW(i-1,j) BW(i-1,j+1) BW(i,j+1)
BW(i+1,j+1) BW(i+1,j) BW(i+1,j-1) BW(i,j-1) BW(i-1,j-1) BW(i-1,j)];
                    if P(2)*P(4)*P(8)==0 && P(2)*P(6)*P(8)==0 &&
sum(P(2:end-1))<=6 && sum(P(2:end-1)) >=2
                        A = 0;
                        for k = 2:size(P(:),1)-1
```

```
                        if P(k) == 0 && P(k+1)==1
                            A = A+1;
                        end%if
                    end%for
                        if (A==1)
                            BW_del(i,j)=1;
                        end%if
                end%if
            end%for
        end%for


        BW(find(BW_del==1))=0;

        if prod(BW_old(:)==BW(:))
            continue_it=0;
        end%if

    end%while
```

Code for SVM classification

```
Dataset = 'C:\Users\HETC\Documents\MATLAB\Signature\vv';
 Testset  = 'C:\Users\HETC\Documents\MATLAB\Signature\test';


 width=70; height=30;
 DataSet      = cell([], 1);

  for i=1:length(dir(fullfile(Dataset,'*.jpg')))

      % Training set process
      k = dir(fullfile(Dataset,'*.jpg'));
      k = {k(~[k.isdir]).name};
      for j=1:length(k)
         tempImage       = imread(horzcat(Dataset,filesep,k{j}));
         imgInfo         = imfinfo(horzcat(Dataset,filesep,k{j}));
```

```
          % Image transformation
          if strcmp(imgInfo.ColorType,'grayscale')
             DataSet{j}   = double(imresize(tempImage,[width
height])); % array of images
          else
             DataSet{j}   =
double(imresize(rgb2gray(tempImage),[width height])); % array of
images
          end
      end
  end
 TestSet =  cell([], 1);
   for i=1:length(dir(fullfile(Testset,'*.jpg')))

      % Training set process
      k = dir(fullfile(Testset,'*.jpg'));
      k = {k(~[k.isdir]).name};
      for j=1:length(k)
         tempImage       = imread(horzcat(Testset,filesep,k{j}));
         imgInfo         = imfinfo(horzcat(Testset,filesep,k{j}));

          % Image transformation
          if strcmp(imgInfo.ColorType,'grayscale')
             TestSet{j}   = double(imresize(tempImage,[width
height])); % array of images
          else
             TestSet{j}   =
double(imresize(rgb2gray(tempImage),[width height])); % array of
images
          end
      end
   end

% we have 30 images and we divided it into two label groups here.
 train_label               = zeros(size(30,1),1);
 train_label(1:15,1)    = 1;          % 1 = backgrounds
 train_label(16:30,1)   = 2;          % 2 = signatures

 % Prepare numeric matrix for svmtrain
```

```
 Training_Set=[];
 for i=1:length(DataSet)
     Training_Set_tmp   = reshape(DataSet{i},1, 70*30);
     Training_Set=[Training_Set;Training_Set_tmp];
 end

 Test_Set=[];
 for j=1:length(TestSet)
     Test_set_tmp   = reshape(TestSet{j},1, 70*30);
     Test_Set=[Test_Set;Test_set_tmp];
 end

 % Perform first run of svm
 SVMStruct = svmtrain(Training_Set , train_label, 'kernel_function',
'linear');
 Group       = svmclassify(SVMStruct, Test_Set);

 testSet = imageSet('test');

 for i=1:testSet.Count
     if(Group(i,1)==1)

imwrite(read(testSet,i),fullfile('C:\Users\HETC\Documents\MATLAB\Sign
ature\b',['B',num2str(i),'.jpg']));
     else

imwrite(read(testSet,i),fullfile('C:\Users\HETC\Documents\MATLAB\Sign
ature\s',['S',num2str(i),'.jpg']));
     end
 end
```

Sample Code for KS test

```
for x = 1:size(trainingFeatures,2)
        D(x,:) = pdist(trainingFeatures(1:7,x));
    end

    for x = 1:size(trainingFeatures,2)
        %for j = 1:size(queryFeatures,2)
```

```
            DT(x,:) = abs(queryFeatures(x) -
trainingFeatures(1:7,x));
        %end
    end

    [H, P] = kstest2(D(:),DT(:),'Alpha',0.01);
```

Code for signature extraction

```
clc;
x = imread('attendance.jpg');
x1 = rgb2gray(x);
figure;
imshow(x1);
title('gray image');

% binarization using Otsu method
threshold = graythresh(x1);
x2 =~ imbinarize(x1,threshold);
figure;
imshow(x2);
title('binarize image');
x2 = edge(x2,'Canny');
figure;
imshow(x2);
title('edge image');
x2 = bwareaopen(x2,10);

figure;
imshow(x);

% Label connected components
[L, Ne]=bwlabel(x2);

% Measure properties of image regions
prop = regionprops(L);
sign = cell(1,length(prop));



hold on
```

```
for n = 1:length(prop)

rectangle('Position',prop(n).BoundingBox,'EdgeColor','g','LineWidth',
2);
    rect = prop(n).BoundingBox;



    sign{n} = imcrop(x, rect);



imwrite(sign{n},fullfile('C:\Users\HETC\Documents\MATLAB\Signature\te
st',['C',num2str(n),'.jpg']));

end

hold off
```