References


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


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

if (A==1)
    \( BW_{del}(i,j)=1; \)
end
end
%if
end
%for

end

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

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

57
% 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\Signature\b','[B',num2str(i),'.jpg']));
    else
        imwrite(read(testSet,i),fullfile('C:\Users\HETC\Documents\MATLAB\Signature\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

[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));
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\test',['C',num2str(n),'.jpg']));

end

hold off