Assignment Machine Learning -1

Table of Contents

Ventrapragada Sai Shravani (17070123120)

Snigdha Labh (17070123105)

Satyaki Tatte (17070123112)

Vinayak Kuanr (17070123121)

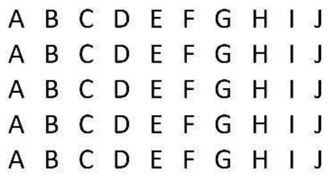
Batch: G3 (17-21)

Source code

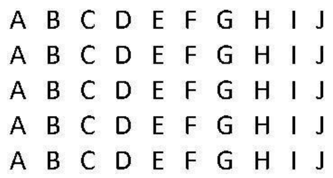
```
b1=load('bias1.mat');
b2=load('bias2.mat');
w1=load('weights1.mat');
w2=load('weights2.mat');
b1=b1.b1;
b2=b2.b2;
w1=w1.w1;
w2=w2.w2;
%Image Preprocessing
I = imread('C:\Users\DELL\Desktop\matlab\Handwrittern-Optical-Character-Recognition-master\TRA.png');
final_image = 'C:\Users\DELL\Desktop\matlab\Handwrittern-Optical-Character-Recognition-master\TRA.png';
%Convert to GrayScale
Igray =rgb2gray(I);
figure(1);
imshow(Igray);
% pause;
%Convert to binary image
Ibw = im2bw(Igray,graythresh(Igray));
figure(2);
imshow(Ibw);
% pause;
%Edge detection
Iedge = edge(uint8(Ibw));
figure(3);
imshow(Iedge);
% pause;
% Image Dilation
se = strel('square',2);
```

```
Iedge2 = imdilate(Iedge, se);
figure(4);
imshow(Iedge2);
% pause;
% Image Filling
Ifill= imfill(Iedge2,'holes');
figure(5);
imshow(Ifill);
% pause;
%Blob analysis
[Ilabel num] = bwlabel(Ifill);
disp(num);
Iprops = regionprops(Ilabel);
Ibox = [Iprops.BoundingBox];
Ibox = reshape(Ibox,[4 num]);
```

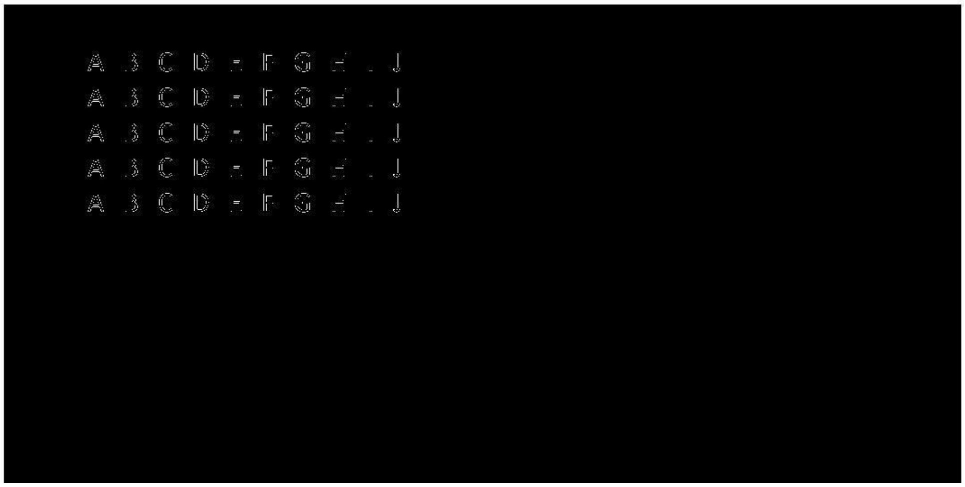
```
Warning: Image is too big to fit on screen; displaying at 50%
Warning: Image is too big to fit on screen; displaying at 50%
Warning: Image is too big to fit on screen; displaying at 50%
Warning: Image is too big to fit on screen; displaying at 50%
Warning: Image is too big to fit on screen; displaying at 50%
    50
```

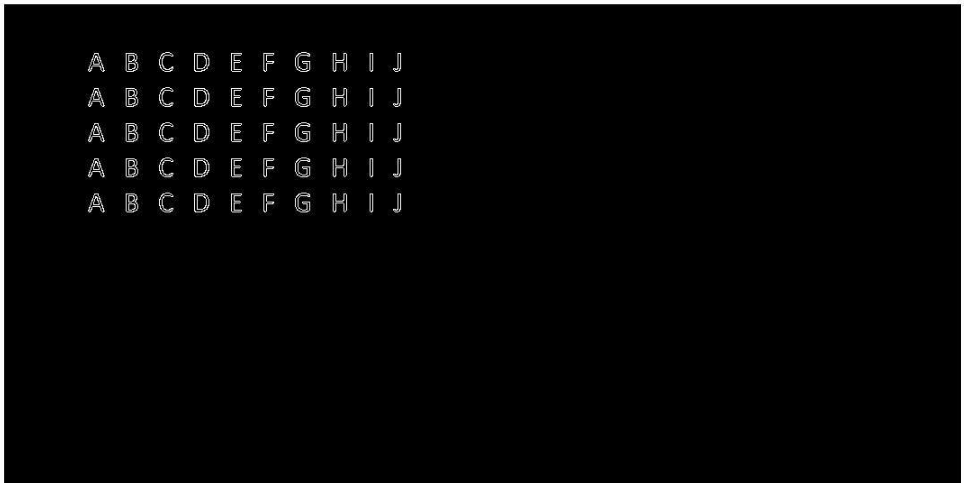


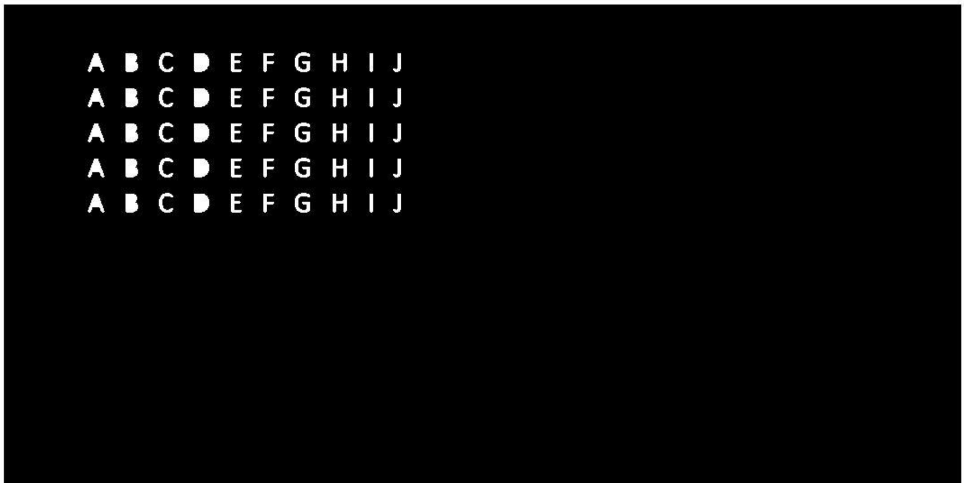
A B C D E F G H I J
A B C D E F G H I J
A B C D E F G H I J
A B C D E F G H I J
A B C D E F G H I J



A B C D E F G H I J
A B C D E F G H I J
A B C D E F G H I J
A B C D E F G H I J
A B C D E F G H I J





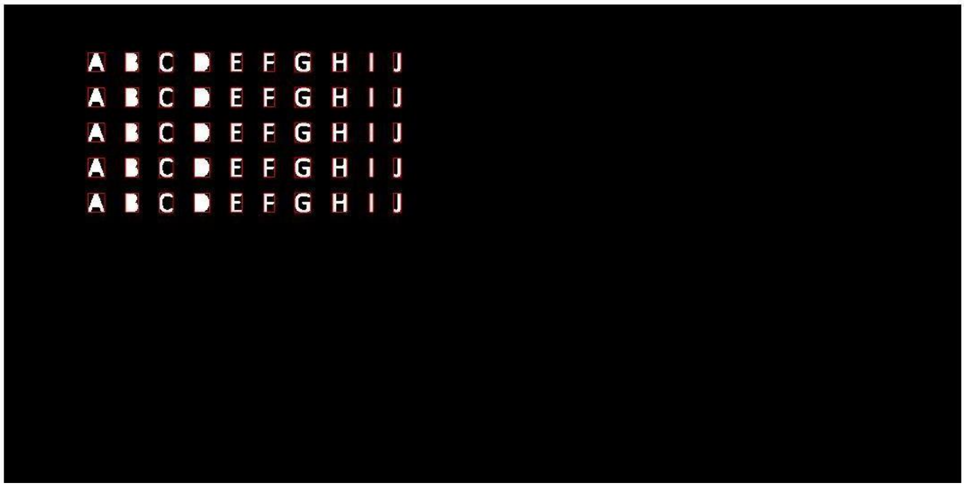


Plotting object location

```
hold on;
for cnt = 1:num
    rectangle('position',Ibox(:,cnt),'edgecolor','r');
end
% a=0.5;
% b=0.500;
% X=0;
% for x=1:24
%     G1=[Ibox(1)  Ibox(2)];
%     G2=[a b]
%     E_dist=sqrt(sum((G1-G2).^2,2))
%      if (a<2.000)
%          a=a+0.3750;
%      end
%      X=X+1;
%      r=rem(X,4);
%      if(r==0)
%          if (b<2.000)
%              b=b+0.2500;
%          end
%     end
% end
cnt=1;
data1=[];
row1=[];
row2=[];
row3=[];
row4=[];
row5=[];
p2=[];
for cnt1 = 1:num
    data=imcrop(Ifill,Ibox(:,cnt1));

    %Feature Extraction
    %Trimming Image
    data( ~any(data,2), : ) = []; %rows
    data( :, ~any(data,1) ) = [];  %columns
    %resizing Image
    final_data = imresize(data,[5 7]);
    %data to be sent to neural network
    data1=[data1 reshape(final_data',35,1)];
    if(rem(cnt1,5)==1)
        row1=[row1 reshape(final_data',35,1)];
    end
    if(rem(cnt1,5)==2)
        row2=[row2 reshape(final_data',35,1)];
    end
    if(rem(cnt1,5)==3)
        row3=[row3 reshape(final_data',35,1)];
    end
    if(rem(cnt1,5)==4)
        row4=[row4 reshape(final_data',35,1)];
    end
    if(rem(cnt1,5)==0)
        row5=[row5 reshape(final_data',35,1)];
    end

end
out=[row1 row2 row3 row4 row5];
j=1;
for i=41:50
Ptest = out(:,i);

answer = ffn(w1,w2,b1,b2,Ptest,35,1,30,10);

final(j,1) = char(answer*[49;50;51;52;53;'A';'B';'C';'D';'E']);
j=j+1;
end
```

```
final
```

```
final =

  10×1 char array

    'A'
    'B'
    'C'
    'E'
    'E'
    ' '
    'C'
    ' '
    '1'
    'E'
```



Conclusion-

Using imread function we read the image and then using rgb2gray the image was converted to gray scale, then using im2bw the image was converted to binary. Next, by doing so we made image in the form of ones and zeroes where 1 being black (background) and 0 being white (foreground). Then by using the function edge we detected the edge of the letter and with imdilate and imfill filled and dilated the image. Next we did blob analysis using regionprop and reshape function to form a bounding box around the letter. We tried calculating euclidean distance from line 50 to 67 but we couldnt extract one of the vectors to fulfil the eucalidean distance parameters. Hence we did trimming of image, resizing of image and hence extracted the features and then the data was sent to the neural network.

Published with MATLAB® R2018b