

# Original

```
I = rgb2gray(imread('Joc_de_caracters.jpg'));  
imshow(I)
```

0 1 2 3 4 5 6 7 8 9 B C D F G H J K L M N P R S T V W X Y Z

```
BW = I<126;  
imshow(BW)
```

0 1 2 3 4 5 6 7 8 9 B C D F G H J K L M N P R S T V W X Y Z

```
CC = bwconncomp(BW);  
props = regionprops(CC, 'Area', 'MajorAxisLength', 'MinorAxisLength');  
X = [props.Area; props.MajorAxisLength; props.MinorAxisLength;];  
Y = ['0'; '1'; '2'; '3'; '4'; '5'; '6'; '7'; '8'; '9'; 'B'; 'C'; 'D'; 'F'; 'G'; 'H'; 'J'; 'K';  
Classifier = TreeBagger(100,X,Y);  
[Ypredicted, certainty] = predict(Classifier,X);  
table(Classifier.ClassNames,Ypredicted,max(certainty,[],2), 'VariableNames',{'Name','Label','Score'})
```

ans = 30×3 table

	Name	Label	Score
1	'0'	'0'	0.6900
2	'1'	'1'	0.6200
3	'2'	'2'	0.6800
4	'3'	'3'	0.7300
5	'4'	'4'	0.6700
6	'5'	'5'	0.5800
7	'6'	'6'	0.6400
8	'7'	'7'	0.5400
9	'8'	'8'	0.6000
10	'9'	'9'	0.6500
11	'B'	'B'	0.6300
12	'C'	'C'	0.7100
13	'D'	'D'	0.6500
14	'F'	'F'	0.6200
15	'G'	'G'	0.5100
16	'H'	'H'	0.6600

	Name	Label	Score
17	'J'	'J'	0.7000
18	'K'	'K'	0.6300
19	'L'	'L'	0.6500
20	'M'	'M'	0.6600
21	'N'	'N'	0.6200
22	'P'	'P'	0.6400
23	'R'	'R'	0.6300
24	'S'	'S'	0.6000
25	'T'	'T'	0.7200
26	'V'	'V'	0.6900
27	'W'	'W'	0.6000
28	'X'	'X'	0.6400
29	'Y'	'Y'	0.6600
30	'Z'	'Z'	0.6300

```
PL = cell2mat(Ypredicted);
TL = cell2mat(num2cell(Y));
encerts = sum(PL == TL)
```

```
encerts = 30
```

```
percentatge_encerts = encerts/ size(Y,1)
```

```
percentatge_encerts = 1
```

```
%com era d'esperar, el percentatge d'encerts es del 100% buscant sobre
%l'imatge original
```

## Resize

```
Ig = imresize(I,15);
BW = Ig<126;
CC = bwconncomp(BW);
props = regionprops(CC,'Area','MajorAxisLength','MinorAxisLength');
X = [props.Area; props.MajorAxisLength; props.MinorAxisLength];
[label,score] = predict(Classifier,X);
% mostrem el resultat mes probable i la seva puntuació (score)
table(Classifier.ClassNames,label,max(score,[],2),'VariableNames',{'Name','Label','Score'})
```

```
ans = 30x3 table
```

	Name	Label	Score
1	'0'	'Z'	0.2900

	Name	Label	Score
2	'1'	'Z'	0.2900
3	'2'	'Z'	0.2900
4	'3'	'Z'	0.2900
5	'4'	'Z'	0.2900
6	'5'	'Z'	0.2900
7	'6'	'Z'	0.2900
8	'7'	'Z'	0.2900
9	'8'	'Z'	0.2900
10	'9'	'Z'	0.2900
11	'B'	'Z'	0.2900
12	'C'	'Z'	0.2900
13	'D'	'Z'	0.2900
14	'F'	'Z'	0.2900
15	'G'	'Z'	0.2900
16	'H'	'Z'	0.2900
17	'J'	'Z'	0.2900
18	'K'	'Z'	0.2900
19	'L'	'Z'	0.2900
20	'M'	'Z'	0.2900
21	'N'	'Z'	0.2900
22	'P'	'Z'	0.2900
23	'R'	'Z'	0.2900
24	'S'	'Z'	0.2900
25	'T'	'Z'	0.2900
26	'V'	'Z'	0.2900
27	'W'	'Z'	0.2900
28	'X'	'Z'	0.2900
29	'Y'	'Z'	0.2900
30	'Z'	'Z'	0.2900

```

PL = cell2mat(label);
TL = cell2mat(num2cell(Y));
encerts = sum(PL == TL)

```

```
encerts = 1
```

```
percentatge_encerts = encerts / size(Y,1)
```

```
percentatge_encerts = 0.0333
```

```
%% com podem veure el Classifíer original no funciona molt bé quan fem un resize de la imatge,  
% per tant intentem crear-ne un que funcioni independentment de la mida de  
% l'imatge  
props = regionprops('table',BW,'Area','Perimeter','BoundingBox');  
C1 = props.Area./(props.Perimeter.*props.Perimeter);  
C2 = props.Perimeter./(props.BoundingBox(:,3)); %W  
C3 = props.Perimeter./(props.BoundingBox(:,4)); %H  
C4 = props.BoundingBox(:,3)./props.BoundingBox(:,4); %W/H  
X = [C1 C2 C3 C4];  
BetterClassifier = TreeBagger(100,X,Y);  
[Ypredicted, certainty] = predict(BetterClassifier,X);  
table(BetterClassifier.ClassNames,Ypredicted,max(certainty,[],2),'VariableNames',{'Name','Label'})
```

```
ans = 30x3 table
```

	Name	Label	Score
1	'0'	'0'	0.6000
2	'1'	'1'	0.5200
3	'2'	'2'	0.5700
4	'3'	'3'	0.6500
5	'4'	'4'	0.6000
6	'5'	'5'	0.6400
7	'6'	'6'	0.6200
8	'7'	'7'	0.6500
9	'8'	'8'	0.6900
10	'9'	'9'	0.7500
11	'B'	'B'	0.6400
12	'C'	'C'	0.6300
13	'D'	'D'	0.6500
14	'F'	'F'	0.6200
15	'G'	'G'	0.7200
16	'H'	'H'	0.6100
17	'J'	'J'	0.5600
18	'K'	'K'	0.6600
19	'L'	'L'	0.6200
20	'M'	'M'	0.6300
21	'N'	'N'	0.6200

	Name	Label	Score
22	'P'	'P'	0.6100
23	'R'	'R'	0.6700
24	'S'	'S'	0.7500
25	'T'	'T'	0.6900
26	'V'	'V'	0.5600
27	'W'	'W'	0.6200
28	'X'	'X'	0.6600
29	'Y'	'Y'	0.6600
30	'Z'	'Z'	0.6600

```
PL = cell2mat(Ypredicted);
TL = cell2mat(num2cell(Y));
encerts = sum(PL == TL)
```

```
encerts = 30
```

```
percentatge_encerts = encerts/ size(Y,1)
```

```
percentatge_encerts = 1
```

```
%%fent que es tingui en compte la proporció en comptes de la magnitud,
%%creem un classifier que funciona per a totes les mides de imatge
```

## Caràcters deformats

```
I = rgb2gray(imread('Joc_de_caracters_deformats.jpg'));
BI = I < 128; % separem el background
imshow(BI);
```

**0 1 2 3 4 5 6 7 8 9 B C D F G H J K L M N P R S T V W X Y Z**

```
CC = bwconncomp(BI);
props = regionprops(CC, 'Area', 'MajorAxisLength', 'MinorAxisLength');
X = [props.Area; props.MajorAxisLength; props.MinorAxisLength];
[label,score] = predict(Classifier,X);

% mostrem el resultat mes probable i la seva puntuació (score)
table(Classifier.ClassNames,label,max(score,[],2), 'VariableNames',{'Name','Label','Score'})
```

```
ans = 30x3 table
```

	Name	Label	Score
1	'0'	'0'	0.6900

	Name	Label	Score
2	'1'	'1'	0.4500
3	'2'	'2'	0.5400
4	'3'	'K'	0.1400
5	'4'	'4'	0.6500
6	'5'	'5'	0.5000
7	'6'	'6'	0.5400
8	'7'	'7'	0.5400
9	'8'	'8'	0.6000
10	'9'	'9'	0.5500
11	'B'	'B'	0.5900
12	'C'	'C'	0.5300
13	'D'	'D'	0.6500
14	'F'	'F'	0.6200
15	'G'	'G'	0.4500
16	'H'	'H'	0.6600
17	'J'	'J'	0.7000
18	'K'	'K'	0.5800
19	'L'	'L'	0.3900
20	'M'	'M'	0.5900
21	'N'	'H'	0.2600
22	'P'	'K'	0.4100
23	'R'	'B'	0.4900
24	'S'	'S'	0.1800
25	'T'	'T'	0.3000
26	'V'	'V'	0.4900
27	'W'	'W'	0.4300
28	'X'	'F'	0.2600
29	'Y'	'4'	0.2300
30	'Z'	'Z'	0.6300

```

PL = cell2mat(label);
TL = cell2mat(num2cell(Y));
encerts = sum(PL == TL)

```

encerts = 24

```
percentatge_encerts = encerts / size(Y,1)
```

```
percentatge_encerts = 0.8000
```

```
%% amb el classfier original ens dona resultats força bons, anem a provar amb el proprcional
```

```
props = regionprops('table',BW,'Area','Perimeter','BoundingBox');
```

```
C1 = props.Area./(props.Perimeter.*props.Perimeter);
```

```
C2 = props.Perimeter./(props.BoundingBox(:,3)); %W
```

```
C3 = props.Perimeter./(props.BoundingBox(:,4)); %H
```

```
C4 = props.BoundingBox(:,3)./props.BoundingBox(:,4); %W/H
```

```
X = [C1 C2 C3 C4];
```

```
[Ypredicted, certainty] = predict(BetterClassifier,X);
```

```
table(BetterClassifier.ClassNames,Ypredicted,max(certainty,[],2),'VariableNames',{'Name','Label'})
```

```
ans = 30x3 table
```

	Name	Label	Score
1	'0'	'0'	0.6000
2	'1'	'1'	0.5200
3	'2'	'2'	0.5700
4	'3'	'3'	0.6500
5	'4'	'4'	0.6000
6	'5'	'5'	0.6400
7	'6'	'6'	0.6200
8	'7'	'7'	0.6500
9	'8'	'8'	0.6900
10	'9'	'9'	0.7500
11	'B'	'B'	0.6400
12	'C'	'C'	0.6300
13	'D'	'D'	0.6500
14	'F'	'F'	0.6200
15	'G'	'G'	0.7200
16	'H'	'H'	0.6100
17	'J'	'J'	0.5600
18	'K'	'K'	0.6600
19	'L'	'L'	0.6200
20	'M'	'M'	0.6300
21	'N'	'N'	0.6200
22	'P'	'P'	0.6100
23	'R'	'R'	0.6700

	Name	Label	Score
24	'S'	'S'	0.7500
25	'T'	'T'	0.6900
26	'V'	'V'	0.5600
27	'W'	'W'	0.6200
28	'X'	'X'	0.6600
29	'Y'	'Y'	0.6600
30	'Z'	'Z'	0.6600

```
PL = cell2mat(Ypredicted);
TL = cell2mat(num2cell(Y));
encerts = sum(PL == TL)
```

```
encerts = 30
```

```
percentatge_encerts = encerts/ size(Y,1)
```

```
percentatge_encerts = 1
```

```
%%sembla que també funciona millor que l'original, pero creiem q es deu a
%%que la deformació no es gaire pronunciada
```

## Descriptors de forma

```
%% Problem a fer servir els descriptors que creiem que no tenen en compte l'area, si no la forma
%% en el nostre cas: circularity, eccentricity, eulerNumber, extent.
```

```
%%problem primer sobre l'imatge original
I = rgb2gray(imread('Joc_de_caracters.jpg'));
BW = I<126;
CC = bwconncomp(BW);
props = regionprops(CC,'Circularity','Extent','Eccentricity','EulerNumber');
X = [props.Circularity; props.Extent; props.Eccentricity; props.EulerNumber]';
Y = ['0'; '1'; '2'; '3'; '4'; '5'; '6'; '7'; '8'; '9'; 'B'; 'C'; 'D'; 'F'; 'G'; 'H'; 'J'; 'K'];
FormClassifier = TreeBagger(100,X,Y);
[Ypredicted, certainty] = predict(FormClassifier,X);
table(BetterClassifier.ClassNames,Ypredicted,max(certainty,[],2),'VariableNames',{'Name','Label'})
```

```
ans = 30x3 table
```

	Name	Label	Score
1	'0'	'0'	0.5800
2	'1'	'1'	0.6500
3	'2'	'2'	0.6600
4	'3'	'3'	0.7200
5	'4'	'4'	0.6700



	Name	Label	Score
6	'5'	'5'	0.6600
7	'6'	'6'	0.6400
8	'7'	'7'	0.7000
9	'8'	'8'	0.5300
10	'9'	'9'	0.6200
11	'B'	'B'	0.6800
12	'C'	'C'	0.6300
13	'D'	'D'	0.6900
14	'F'	'F'	0.6200
15	'G'	'G'	0.6400
16	'H'	'H'	0.6900
17	'J'	'J'	0.5900
18	'K'	'K'	0.6800
19	'L'	'L'	0.6550
20	'M'	'M'	0.6100
21	'N'	'N'	0.6200
22	'P'	'P'	0.5600
23	'R'	'R'	0.6300
24	'S'	'S'	0.6200
25	'T'	'T'	0.5950
26	'V'	'V'	0.7300
27	'W'	'W'	0.6500
28	'X'	'X'	0.6000
29	'Y'	'Y'	0.6800
30	'Z'	'Z'	0.6000

```
PL = cell2mat(Ypredicted);
TL = cell2mat(num2cell(Y));
encerts = sum(PL == TL)
```

```
encerts = 30
```

```
percentatge_encerts = encerts/ size(Y,1)
```

```
percentatge_encerts = 1
```

```
%sembla que dona bons resultats sobre l'original, anem a provar amb els
%%caràcters deformats
```

```

I = rgb2gray(imread('Joc_de_caracters_deformats.jpg'));
BW = I<126;
CC = bwconncomp(BW);
props = regionprops(CC,'Circularity','Extent','Eccentricity','EulerNumber');
X = [props.Circularity; props.Extent; props.Eccentricity; props.EulerNumber]';
Y = ['0'; '1'; '2'; '3'; '4'; '5'; '6'; '7'; '8'; '9'; 'B'; 'C'; 'D'; 'F'; 'G'; 'H'; 'J'; 'K'; 'L'; 'M'; 'N'; 'P'; 'R'; 'S'; 'T'; 'V'; 'W'];
[Ypredicted, certainty] = predict(FormClassifier,X);
table(FormClassifier.ClassNames,Ypredicted,max(certainty,[],2),'VariableNames',{'Name','Label','Score'})

```

ans = 30x3 table

	Name	Label	Score
1	'0'	'0'	0.5800
2	'1'	'1'	0.5200
3	'2'	'Y'	0.2700
4	'3'	'V'	0.2900
5	'4'	'4'	0.6600
6	'5'	'5'	0.5200
7	'6'	'6'	0.5400
8	'7'	'7'	0.7000
9	'8'	'8'	0.5300
10	'9'	'9'	0.5000
11	'B'	'B'	0.4500
12	'C'	'C'	0.5300
13	'D'	'D'	0.6900
14	'F'	'F'	0.6200
15	'G'	'G'	0.6100
16	'H'	'H'	0.6900
17	'J'	'J'	0.5900
18	'K'	'K'	0.6800
19	'L'	'L'	0.5050
20	'M'	'M'	0.3100
21	'N'	'N'	0.5100
22	'P'	'P'	0.5600
23	'R'	'R'	0.5800
24	'S'	'S'	0.3600
25	'T'	'L'	0.5050
26	'V'	'V'	0.4400
27	'W'	'W'	0.4700

	Name	Label	Score
28	'X'	'7'	0.2700
29	'Y'	'Y'	0.5500
30	'Z'	'Z'	0.6000

```
PL = cell2mat(Ypredicted);
TL = cell2mat(num2cell(Y));
encerts = sum(PL == TL)
```

```
encerts = 26
```

```
percentatge_encerts = encerts/ size(Y,1)
```

```
percentatge_encerts = 0.8667
```

```
%% fent servir els descriptors de forma ens dona millors resultatats que amb el proporcional,
%% creiem que el FormClassifier donara resultats molt millors que l'altre en imatges amb deform
```

## Experimentació amb matricules

```
I = rgb2gray(imread('matricula1.jpg'));
BW = I<40;
imshow(BW);
```



```
CC = bwconncomp(BW);
%% borrem el primer element que trobem(escaneja columna per columna, de esquerra a dreta)
%% es a dir, la part blava de la matricula
BW(CC.PixelIdxList{1})=0;
imshow(BW)
```

# 3547 NXB

```
CC = bwconncomp(BW);
props = regionprops('table',BW,'Area','Perimeter','BoundingBox');
C1 = props.Area./(props.Perimeter.*props.Perimeter);
C2 = props.Perimeter./(props.BoundingBox(:,3)); %W
C3 = props.Perimeter./(props.BoundingBox(:,4)); %H
C4 = props.BoundingBox(:,3)./props.BoundingBox(:,4); %W/H
X = [C1 C2 C3 C4];
Yexpected = {'3';'5';'4';'7';'N';'X';'B';}
```

```
Yexpected = 7x1 cell
'3'
'5'
'4'
'7'
'N'
'X'
'B'
```

```
[Ypredicted, certainty] = predict(BetterClassifier,X);
Ypredicted
```

```
Ypredicted = 7x1 cell
'5'
'G'
'4'
'4'
'N'
'3'
'p'
```

```
percentatge_encerts = sum(strcmp(Ypredicted,Yexpected))/7
```

```
percentatge_encerts = 0.2857
```

%% podem veure que no funciona gaire bé (nomes encerta 2/7) , anem a provar amb l'altre Classificador

```
props = regionprops(CC,'Circularity','Extent','Eccentricity','EulerNumber');
X = [props.Circularity; props.Extent; props.Eccentricity; props.EulerNumber]';
Yexpected = {'3';'5';'4';'7';'N';'X';'B';}
```

```
Yexpected = 7×1 cell
'3'
'5'
'4'
'7'
'N'
'X'
'B'
```

```
[Ypredicted, certainty] = predict(FormClassifier,X);
Ypredicted
```

```
Ypredicted = 7×1 cell
'5'
'G'
'Y'
'Y'
'N'
'S'
'D'
```

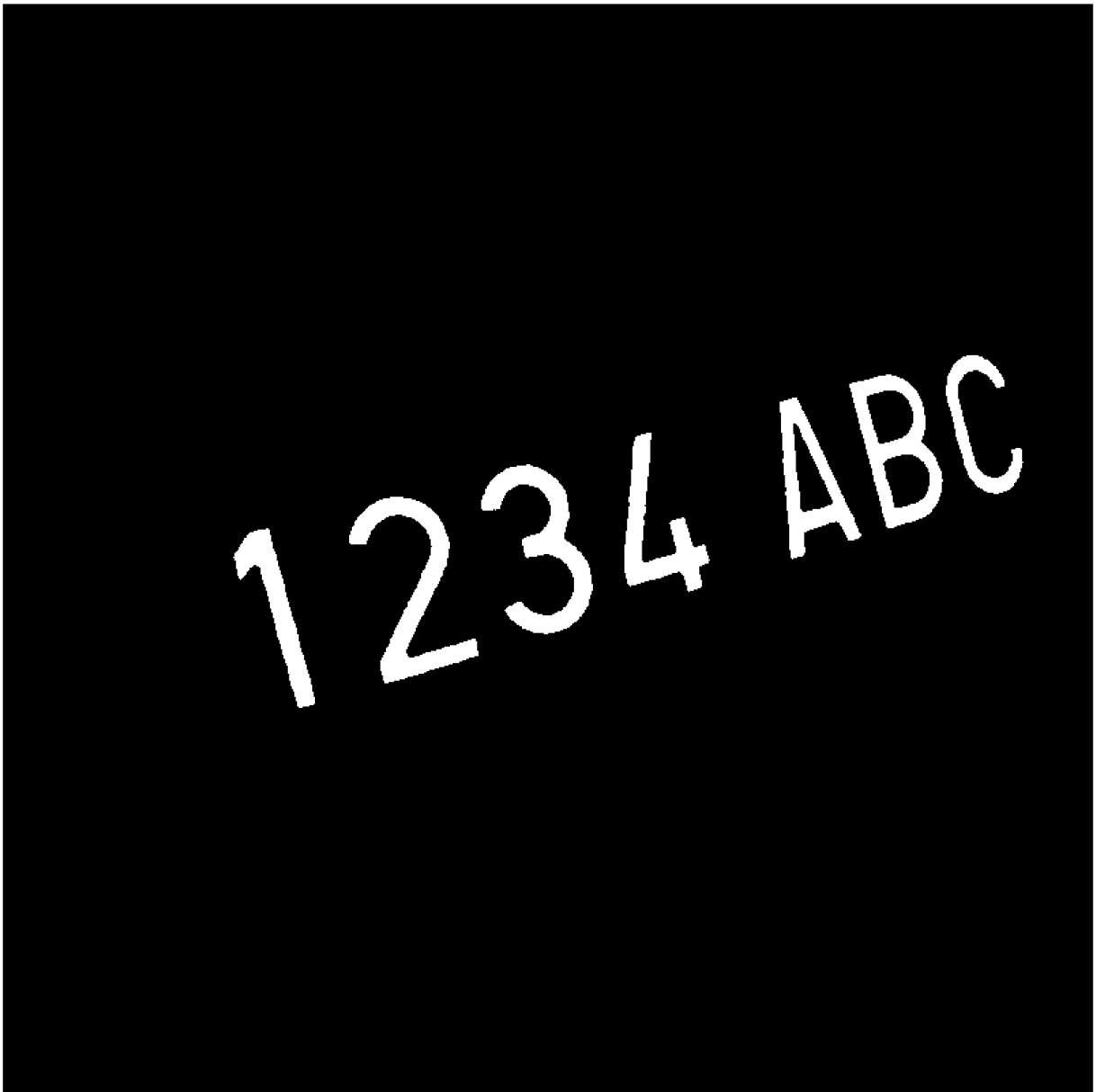
```
percentatge_encerts = sum(strcmp(Ypredicted,Yexpected))/7
```

```
percentatge_encerts = 0.1429
```

```
%sembla que el formClassifier dona encara pitjors resultats que l'anterior
```

```
%porbem amb una segona matriucla
```

```
I = rgb2gray(imread('matricula2.jpg'));
BW = I<40;
imshow(BW);
```



```
CC = bwconncomp(BW);
props = regionprops('table',BW,'Area','Perimeter','BoundingBox');
C1 = props.Area./(props.Perimeter.*props.Perimeter);
C2 = props.Perimeter./(props.BoundingBox(:,3)); %W
C3 = props.Perimeter./(props.BoundingBox(:,4)); %H
C4 = props.BoundingBox(:,3)./props.BoundingBox(:,4); %W/H
X = [C1 C2 C3 C4];
Yexpected = {'1';'2';'3';'4';'A';'B';'C';}
```

```
Yexpected = 7x1 cell
'1'
'2'
'3'
```

```
'4'  
'A'  
'B'  
'C'
```

```
[Ypredicted, certainty] = predict(BetterClassifier,X);  
Ypredicted
```

```
Ypredicted = 7×1 cell  
'1'  
'6'  
'6'  
'C'  
'P'  
'0'  
'G'
```

```
percentatge_encerts = sum(strcmp(Ypredicted,Yexpected))/7
```

```
percentatge_encerts = 0.1429
```

```
%% funciona pitjor que en el cas anterior, ja que la matricula no esta recta, anem a provar amb  
%% el qual abans a donat millors resultats amb les deformacions
```

```
props = regionprops(CC,'Circularity','Extent','Eccentricity','EulerNumber');  
X = [props.Circularity; props.Extent; props.Eccentricity; props.EulerNumber]';  
Yexpected = {'1';'2';'3';'4';'A';'B';'C'};
```

```
Yexpected = 7×1 cell  
'1'  
'2'  
'3'  
'4'  
'A'  
'B'  
'C'
```

```
[Ypredicted, certainty] = predict(FormClassifier,X);  
Ypredicted
```

```
Ypredicted = 7×1 cell  
'J'  
'Y'  
'Y'  
'7'  
'Y'  
'7'  
'J'
```

```
percentatge_encerts = sum(strcmp(Ypredicted,Yexpected))/7
```

```
percentatge_encerts = 0
```

```
%% el de forma no consegueix encertar cap de dels caràcters
```