Entrega 9 - 28/04

Reconeixement (classificació)

Buscar un patró mentres es va iterant en tota la imatge (matching per correlació).

Linear classifier

Tenint una eq de la recta: ax + by +c = 0. Si agafem un x y de la recta la distancia serà 0. Però si agafem x y de fora la recta ja no serà 0. Serà ax + by + c != 0 (major o menor depenent del punt). I aquell valor serà la distancia respecte la recta.

[ax + by + c = d]

Calcular dist: manhattan, euclidean, scaled euclidean, mahalanobis ... (diapos)

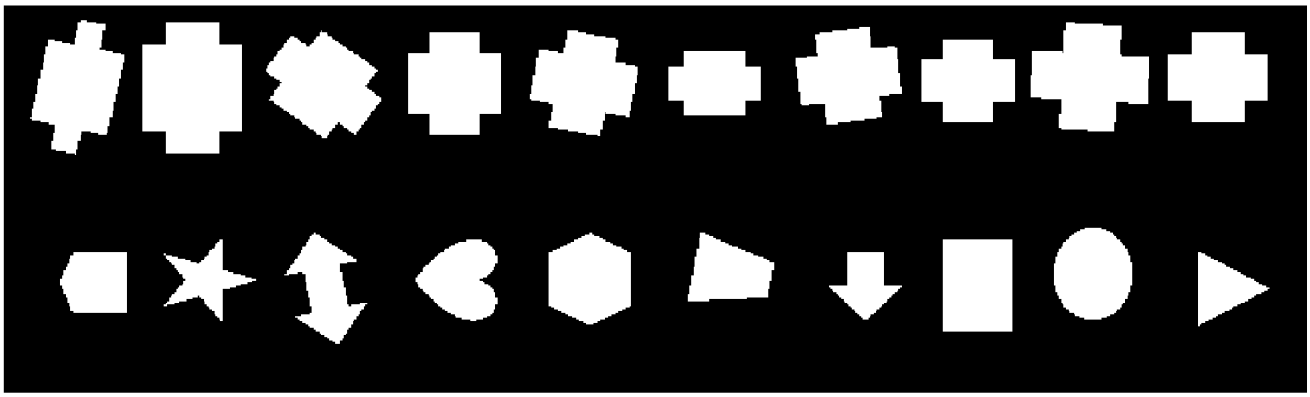
K-NN classifier

Els k veins més propers. Preferible amb poques dades.

Support Vector Machines (SVM)

Trobar la millor recta com a boundary. Trobar els punts més característics per després poder separar amb una recta. Punts de suport en fer la classificació.

```
BW = rgb2gray(imread("creus_no_creus.png")) < 200;
imshow(BW);
```



```
% trobar una funcio que classifiqui

BWU = BW;
BWU(end/2:end,:) = 0;
BWD = BW;
BWD(1:end/2,:) = 0;
imshow(BWU);
```

```
imshow(BWD);
```

```
CCU = bwconncomp(BWU);
CCD = bwconncomp(BWD);

propsU = regionprops('table', CCU, 'Centroid', 'BoundingBox', 'Perimeter', 'Circularity', 'Soli
propsD = regionprops('table', CCD, 'Centroid', 'BoundingBox', 'Perimeter', 'Circularity', 'Soli

NumObj = CCU.NumObjects;

FU = [propsU.Perimeter./propsU.MinFeretDiameter,propsU.Circularity,propsU.MinorAxisLength./prop
FD = [propsD.Perimeter./propsD.MinFeretDiameter,propsD.Circularity,propsD.MinorAxisLength./prop

Features = [FU;FD];
Output = false([2*NumObj,1]);
Output(1:NumObj) = true;

NeuralTrainedClassification.predictFcn(Features) % 95% neural classification (wide)
```

```
ans = 20×1 logical array
   1
```

```
1
1
1
1
1
1
1
1
1
1
  ⋮
```

```
imshow("Captura.PNG");
```