# VISIÓ PER COMPUTADOR

Sessió 8 de Laboratori

**Facultat d'Informàtica de Barcelona**

**Manel Frigola Joan Climent**

**Barcelona, Maig de 2022**

1. **Continguts de la sessió**

L’objectiu de la sessió és el de conèixer les possibilitats dels descriptors de forma d’objectes binaritzats i, sense entrar en gaires detalls, utilitzar un dels molts classificador que té el Matlab. En concret es treballaran els següents conceptes:

* + Caracterització de regions: *regionprops* i altres
  + Construcció d’un classificador d’objectes: *TreeBagger*.
  + Classificació d’objectes: *Predict*.

Els objectes a classificar que s’utilitzaran en aquesta pràctica són els joc de caracters de les matrícules espanyoles que apareixen en la imatge *Joc\_de\_caracters.jpg* disponible en la carpeta de l’assignatura.



També hi han disponibles, en la mateixa carpeta, imatges de test per comprovar que tal acurada és la classificació. Aquestes imatges han estat deformades en mida i angle, tal com es mostra en la figura següent.



**2. Exercici de la sessió**

1. Obrim la imatge i binaritzem amb un llindar fix:

I = rgb2gray(imread('Joc\_de\_caracters.jpg'));

BI = I < 128; % lletres estan en negre

1. Obtenim la llista dels píxels que conformen cadascun dels caràcters segmentats:

CC = bwconncomp(BI);

1. Mesurem algunes propietats descriptores de forma amb *regionprops*.

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

Comentar que **aquestes propietats són fortament dependents de la mida** de l’objecte, i que funcionaran més o menys bé per a lletres de la mateixa mida, però deixaran de funcionar en casos reals on no es conegut el zoom de la imatge.

1. Cal passar de llista a taula, per això construïm una taula de mesures o taula d’observacions:

X = [props.Area props.Perimeter props.BoundingBox(:,3) props.BoundingBox(:,4)];

1. Construïm el output o resultat que voldríem obtenir a partir de les entrades X (taula de mesures):

OUT = ['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'];

1. Construcció del classificador.

Sense entrar en gaires detalls formals, farem servir un mètode de classificació anomenat *TreeBagger* que ens retorna una variable del tipus *Classifier.* La variable *Classifier* s’utilitzara en una posterior funció de predicció per predir el resultat en altres imatges. El vector OUT és el resultat que el *TreeBagger* haurà d’aprendre a “endevinar” a partir de les observacions X.

Classifier = TreeBagger(100,X,OUT');

Nota: El primer paràmetre de la funció *TreeBagger*, 100 en l’exemple anterior, indica el nombre d’arbres de decisió que es construiran per classificar les observacions. Normalment aquest tipus de classificadors comencen a funcionar correctament amb un nombre elevat d’observacions diferents (variables descriptores o predictores) per classe. **Comentar que amb quatre mesures per lletra, i només una observació per classe, construir 100 arbres de desició és un despropòsit**, doncs amb tant poques observacions no es poden construir 100 arbres de decisió tots diferents.

1. Ara cal comprovar que tan bo és el classificador, repetint el procediment de mesura amb una nova imatge de test.

I = rgb2gray(imread('Joc\_de\_caracters\_deformats.jpg'));

BI = I < 128; % separem el background

figure;imshow(BI);

CC = bwconncomp(BI);

props = regionprops(CC,'Area','MajorAxisLength','MinorAxisLength');

X = [props.Area; props.MajorAxisLength; props.MinorAxisLength;]';

1. Amb la variable *Classifier* i la funció *Predict* intentem endevinar quin caràcter és cadascun (*label*) i quina puntuació té la classificació (*score*).

[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'})

**3. Es demana**

1. Mesureu el resultat de la classificació amb el joc de caràcters ampliat de mida (*imresize*). Introduir columnes (predictors/descriptors) a la taula d’observacions que siguin força invariants al zoom i elimineu les que siguin no-invariants o combineu mesures per a que el resultat sigui proper al 100%.
2. Mesureu el resultat amb la imatge del joc de caràcters deformats.
3. Proveu el resultat amb descriptors de forma.
4. Opcionalment, proveu el resultat amb dues imatges de matricules.