

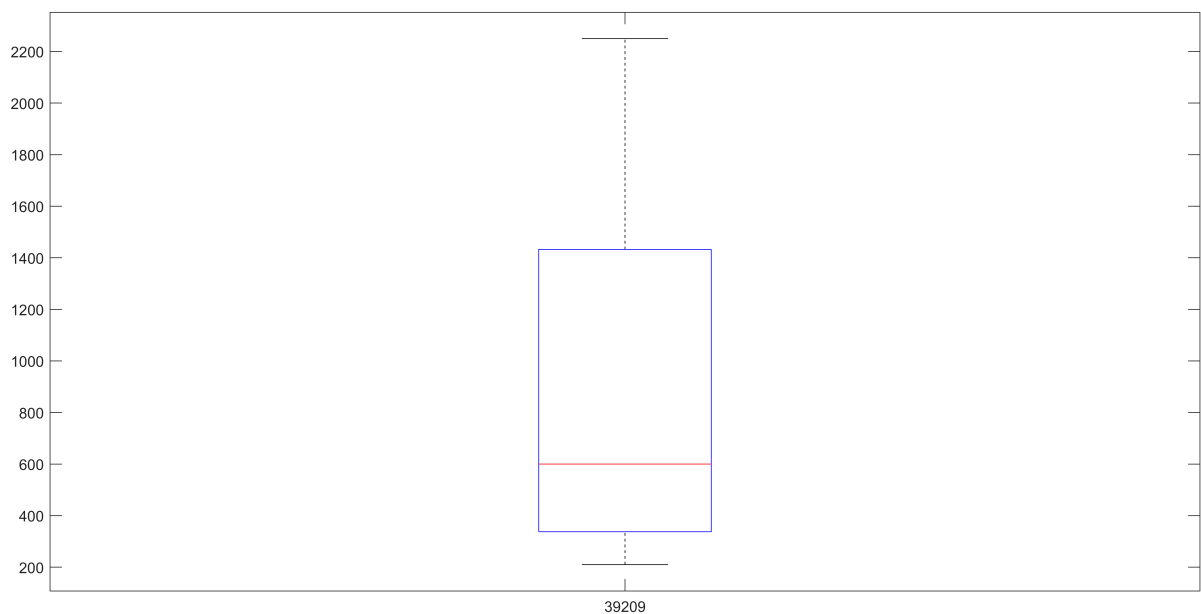
Lectura de datos

El imageDatastore permite manipular datos a gran escala, sin necesidad de cargar todas las imagenes a la vez.

```
dsModels = imageDatastore("Models\Meta\*.png");  
ds = imageDatastore(["Train1\Train1\","Train2\Train2\"], "LabelSource","foldernames", "Include  
nFiles = numel(ds.Files)
```

```
nFiles = 39209
```

```
%podemos mirar cuantos items de cada clase hay y la distribución:  
eachLable = countEachLabel(ds);  
boxplot(eachLable.Count, nFiles);
```



```
%De la misma manera, aplicar un ordenado aleatorio tambien es comun en los  
%preparativos para entrenar un modelo:  
ds = shuffle(ds);
```

Es interesante que de cada clase hayan aproximadamente la misma cantidad de representantes, de otra manera el modelo puede estar desviado. (Esto mas adelante).

Split Dataset

Ahora generaremos una particion en el datastore, un 80% sera para entrenar el modelo y el restante para test.

```
[train,test] = splitEachLabel(ds,0.8);  
nTrain = numel(train.Files)
```

```
nTrain = 31367
```

```
nTest = numel(test.Files)
```

```
nTest = 7842
```

```
%[train,test] = splitEachLabel(dsTrain,0.8,'randomized');  
% tmb se puede hacer con : subset
```

Calcular Características - Construir vector características

Un cop separades les imatges, queda extreure les característiques mes rellevants per poder entrenar els models.

S'ha definit una funció per poder filtrar les imatges (preProcesing):

```
trainTransform = transform(train, @preProcesing);
```

Amb les imatges filtrades, calculem les seves característiques:

```
T = table;  
%es util definir el size de la taula previament  
for n = 1:numel(train.Files)  
    [I,info] = read(trainTransform);  
    d = calcCaracteristicas(info.Label, I);  
    T = [T; d];  
end  
T.Properties.VariableNames = {'Label' 'Atb1' 'Atb2' 'Atb3'};  
save('calcs/dataSetVars.mat');
```