# main.m

* exists(‘data’, ‘var’) → Comprueba si existe o no la variable ‘data’.
* height(T) → Devuelve el número de filas en la tabla T.
* repmat → Concatena matrices.
* TreeBagger(100, Tb1, ResponseVarName) → Devuelve 100 arboles de clasificación entrenados usando los datos de nuestra de la tabla Tb1. ResponseVarName es el nombre de la variable respuesta (variable que se quiere predecir) en Tb1.
* predict → Predictor.
* num2str → Convertir números a array de caracteres.
* cellstr → Convertir a matriz de vectores de caracteres.

# read\_data.m

Parámetros: directorio de las imágenes, número de imágenes.

* strcat → Concatenar cadenas horizontales.
* dir(name) → Devuelve atributos acerca del directorio ‘name’.
* nargin → Devuelve el número de argumentos (parámetros) de entrada dados en la llamada a la función actualmente en ejecución.
* cell(1,n) → Devuelve una matriz de celdas de 1xn (1 fila y n columnas).
* imread → Leer imagen.
* rgb2gray → Convertir una imagen RGB a escala de grises.
* dlmread(filename, delimiter, R1, C1) → Lector de fichero. Comienza a leer en la fila de desplazamiento R1 (0 = primera fila) y la columna C1 (0 = primera columna). Delimiter puede ser: ‘,’ ‘;’ ‘ ‘, etc.
* reshape(A, [f c]) → Reformula A como un array de f x c

# read\_looking\_data.m

Parámetros: filename (‘Miram.xlsx’)

* xlsread(filename, sheet, range) → Lee los valores del fichero xlsx ‘filename’ especificados por ‘range’(celdas excel)

# generate\_eye\_data.m

Parámetros: data(los datos de las imágenes), crop\_size (para hacer el recorte de los ojos)

* struct() → Crea un struct sin campos.
* struct2table → Convierte un struct en una tabla.

# extract\_eye\_image.m

Parámetros: data{i} (un elemento de data), crop\_size.

* pdist → Distancia euclidiana entre dos valores (en este caso entre los dos centros de los ojos).
* ceil → Redondea al entero más cercano mayor o igual.

# generate\_features.m

Parámetros: im(imagen en cuestión).

* size(A,d) → Tamaño de la dimensión d de A. Si d = 1 filas, d =2 columnas.
* functions(f) → Retorna información de una función.
* int2str → convierte enteros a caracteres.
* extractHogFeatures(I, ‘Cellsize’, [X Y]) → Extrae carcateristicas HOG de la imagen I. Si [X Y] es más pequeño implica más detalle.

# generate\_non\_eye\_data.m

Parámetros: data(los datos de las imagenes), non\_eye\_per\_image (cuantas imagenes recortadas por cada imagen), crop\_size.

* randi(v\_max, num\_vals, ini) → Devuelve num\_vals valores aleatorios comprendidos entre ini y v\_max.

# split\_data.m

Parámetros: data\_eyes, data\_no\_eyes

* randperm(n) → Retorna un vector permutado de manera aleatoria entre 1 y n (ambos incluidos).
* floor → Redondea al entero más cercano menor o igual.

# eval\_prediction.m

Parámetros: prediction, class conocida.

* confusionmat(group, grouphat) → Devuelve la matriz de confusión determinada por los grupos conocidos y predichos. group = grupo conocido, grouphat = grupo predicho. (En el codigo creo que esta al reves).

# get\_looking\_vector.m

Parámetros: data(imagenes)

* cell2mat → Convierte matriz de celdas a matriz ordinaria del tipo de datos subyacente.
* extractfield(S, name) → Devuelve los valores del campo especificado por ‘name’ de la estructura ‘S’.
* kron → Duplica elementos (1 por 1) de un vector.