```
#-----------------------------------------------------
#-----------------------------------------------------
#Ejemplo: Extracción de Características Geométricas
#En este ejemplo se muestra la extracción de características de geometría tales como:
#
#Características básicas
#Características elípticas
#Características basadas en momentos invariantes
#Los ejemplos 1,2 y 3 se realizan en radiografías de frutas.
#
#El ejemplo 4 consiste en clasificar tres tipos de flechas.
#-----------------------------------------------------
#-----------------------------------------------------


#-----------------------------------------------------
#0. Setup inicial
#
# Instalación de PyXvis
#-----------------------------------------------------
!wget https://www.dropbox.com/...../pyxvis.zip #CARGAR SU LIBRERIA
!unzip pyxvis.zip
!rm pyxvis.zip
print('PyXvis library downloaded.')
!pip install scipy==1.2
!pip3 install pybalu==0.2.5
!pip install ./pyxvis
print('PyXvis library installed.')
#=====================================================
import seaborn as sns
import numpy as np
import matplotlib.pyplot as plt
from   skimage.measure import label
from   pyxvis.processing.segmentation import seg_bimodal
from   pyxvis.features.extraction import extract_features
from   pyxvis.io.plots import plot_ellipses_image
from   sklearn.metrics import confusion_matrix, accuracy_score
#=====================================================

#-----------------------------------------------------
#1. Carga de Imágenes
#
#Se cargan dos imágenes N0001_0004b.png y N0006_0003b.png.
#-----------------------------------------------------
#=====================================================
!wget https://www.dropbox.com/...../N0001_0004b.png #CARGAR SU IMAGEN
!wget https://www.dropbox.com/...../N0006_0003b.png #CARGAR SU IMAGEN
#=====================================================

#-----------------------------------------------------
# Ejemplo 1: Características geométricas básicas
# En este ejemplo, mostramos cómo extraer las características geométricas básicas de
diez manzanas segmentadas en la imagen de rayos X N0001_0004b.png. La segmentación
en este ejemplo se realiza umbralizando la imagen de rayos X y seleccionando
aquellos objetos segmentados que presentan un tamaño (área) adecuado.
#-----------------------------------------------------
#=====================================================
# Input Imagen 1 para ejemplo 1
fig    = plt.figure()
ax     = fig.add_subplot(111)
img1    = plt.imread('N0001_0004b.png')
implot = plt.imshow(img1,cmap='gray')


# Segmentation
R      = img>0.27          # thresholding of light objects
L      = label(R)          # labeling of objects
n      = np.max(L)         # number of detected objects
T      = np.zeros((n,18)) # features of each object will stored in a row
```

```
67
68  # Analysis of each segmented object
69  t      = 0 # count of recognized fruits
70  for i in range(n):
71      R = (L == i)*1                          # binary image of object i
72      f = extract_features('basicgeo',bw=R)  # feature extraction for object i
73      area = f[4]
74      # recognition of fruits according to the size
75      if area>14000 and area<21000:
76          T[t,:] = f                          # storing the features of the fruit t
77          t = t+1
78          # labeling each recognized fruit in the plot
79          ax.text(f[1]-20, f[0]+10, str(t), fontsize=12,color='Red')
80
81  # Display and save results
82  plt.show()
83  F = T[0:t,:]
84  print('Basic Geo-Features:')
85  print(F)
86  np.save('GeoFeatures.npy',F)               # save features
87  #=====================================================
88
89  #-----------------------------------------------------
90  #Características extraidas con extract_features de PyXvis.
91  #f1 : Center of mass in  i  direction.
92  #f2 : Center of mass in  j  direction.
93  #f3 : Height.
94  #f4 : Width.
95  #f5 : Area.
96  #f6 : Perimeter.
97  #f7 : Roundness.
98  #f8 : Danielsson factor.
99  #f9 : Euler Number.
100 #f10 : Equivalent Diameter.
101 #f11 : Major Axis Length.
102 #f12 : Minor Axis Length.
103 #f13 : Orientation.
104 #f14 : Solidity.
105 #f15 : Extent.
106 #f16 : Eccentricity.
107 #f17 : Convex Area.
108 #f18 : Filled Area.
109 #Features divided by 1000 of the ten apples:
110 #-----------------------------------------------------
111 # EJERCICIO: CON LOS RESULTADOS ARROJADOS DEBE ARMAR UNA TABLA
112 #-----------------------------------------------------
113
114 #-----------------------------------------------------
115 # Ejemplo 2: Características Elípticas
116 #-----------------------------------------------------
117 #En este ejemplo, mostramos cómo extraer características elípticas del contorno de
    una fruta. Probamos este método en una radiografía de una cereza con forma elíptica
    como se muestra en el resultado.
118 #-----------------------------------------------------
119 #=====================================================
120 # Input Imagen 2 para ejemplo 2
121 fig    = plt.figure()
122 ax     = fig.add_subplot(111)
123 img2    = plt.imread('N0006_0003b.png')
124 implot = plt.imshow(img2,cmap='gray')
125 #=====================================================
126
127 #=====================================================
128 img    = img2                                  # input image with a fruit
129 R,_,   = seg_bimodal(img)                      # segmentation
130 fxell  = extract_features('ellipse',bw=R)      # extraction of elliptical features
131 print('Elliptical Features:')                  # show results
132 print(fxell)                                   # print elliptical features
133 plot_ellipses_image(img,fxell)                 # draw ellipse onto image
```

```
print('Note: The 7 values are: [Mass center: (io,jo) in pixels; Axis: (ae,be) in pixels; Orientation in grads; Eccentricity; Area in pixels]')
#=====================================================

#-----------------------------------------------------
# Ejemplo 3: Momentos invariantes
#-----------------------------------------------------
#En este ejemplo, mostramos cómo medir momentos invariantes que pueden utilizarse como característica de forma de los objetos de interés. Probamos este método con una radiografía que contiene 10 manzanas. Superponemos a esta imagen 4 rectángulos cuyo tamaño es de  a×b  píxeles (donde  b=3a ). Los rectángulos están situados en las direcciones horizontal y vertical, como se muestra en el resultado. Así, podemos simular una imagen de rayos X de entrada que contiene manzanas y rectángulos. La idea es separarlos. Vemos que el primer Hu-momento puede utilizarse para discriminar eficazmente las manzanas de los rectángulos.
#-----------------------------------------------------
#=====================================================
fig    = plt.figure()
ax     = fig.add_subplot(111)
img    = img1
img[100:399,750:849]  = 0.5
img[500:699,850:916]  = 0.75
img[20:119,100:399]   = 0.6
img[90:156,1000:1199] = 0.75
implot = plt.imshow(img,cmap='gray')
R      = img>0.27   # segmentation
L      = label(R)   # labeling
n      = np.max(L)  # number of segmented objects
t      = 0
T      = np.zeros((n,7))
for i in range(n):
    R = (L == i)*1                      # binary image of object i
    fx = ['basicgeo','hugeo']           # extraction of basic geometric features and Hu moments
    f = extract_features(fx,bw=R)       # feature extraction
    area = f[4]
    # recognition of fruits according to the size
    if area>10000 and area<31000:
        h      = f[18:]                 # hu moments
        T[t,:] = h
        t      = t+1
        x      = round(1000*h[0])       # first hu moment
        ax.text(f[1]-20, f[0]+10, str(int(x)), fontsize=12,color='Red')
plt.show()
F = T[0:t,:]
print('Hu Features:')
print(F)
np.save('HuFeatures.npy',F)             # save features
#=====================================================
#-----------------------------------------------------
#Note: In this example, the features (basic geometric features for centroid and area, and Hu moments) are computed by function extract_features of PyXvis with parameters ['basicgeo','hugeo'] and bw=R, where R is the binary image from which the features are extracted. The output of this function is a vector f computed by concatenation of two vectors, one for the basic geometric features (of 18 elements) and one for the Hu moments (of 7 elements). Thus, the first Hu moment is stored in f[18]. The reader can test Flusser and Gupta moments using functions parameters 'flusser' and 'gupta' respectively in function extract_features.
#-----------------------------------------------------


#-----------------------------------------------------
# Ejemplo 4: Reconocimiento de Flechas
#-----------------------------------------------------
#En este ejemplo se realiza un reconocimiento de tres tipos de flechas como las que se muestran en la figura. Las flechas son de diversos tamaños y diversas orientaciones.

#Clase 0: flecha curva unidireccional
```

```
185
186 #Clase 1: flecha recta bidireccional
187
188 #Clase 2: flecha recta unidireccional
189
190 #Para la solución se utiliza los momentos de Hu ya que son invariantes a la rotación
    y a la escala.
191 #-----------------------------------------------
192 #===============================================
193 # Datos de Training: 3 clases y 12 imágenes por clase
194
195 !wget https://www.dropbox.com/...../arrows_training.zip
196 !unzip arrows_training
197 #===============================================
198 #===============================================
199 # Datos de Testing: 3 clases y 10 imágenes por clase
200 !wget https://www.dropbox.com/...../arrows_testing.zip
201 !unzip arrows_testing
202 #===============================================
203
204 #===============================================
205 # Funciones necesarias para que se pueda cargar una imagen individual
206
207 def num2fixstr(x,d):
208   # example num2fixstr(2,5) returns '00002'
209   # example num2fixstr(19,3) returns '019'
210   st = '%0*d' % (d,x)
211   return st
212
213 def ImageLoad(prefix,num_char,num_img,echo='off'):
214   # img = ImageLoad('example/char_',1,3)   loads image 'example/char_01_003.png'
215   # img = ImageLoad('example/char_',2,15)  loads image 'example/char_02_015.png'
216   st   = prefix + num2fixstr(num_char,2) + '_' + num2fixstr(num_img,2) + '.png'
217   if echo == 'on':
218     print('loading image '+st+'...')
219   img    = plt.imread(st)
220   return img
221 #===============================================
222
223 #===============================================
224 # Extracción de momentos de Hu para el training
225
226 K = 3  # número de clases
227 N = 12 # número de imágenes por clase
228
229 fx = ['hugeo']                      # características a extraer (momentos de Hu)
230 M  = 7                              # los momentos de Hu son 7
231
232 # si se quiere usar momentos de Flusser se puede usar fx = ['flusser'] y M = 4
233
234 Xtrain = np.zeros((K*N,M))          # K x N muestras (filas), y M características
    (columnas)
235 ytrain_gt = np.zeros((K*N,))        # ground truth (clasificacion ideal)
236
237 t = 0
238 for j in range(K):                  # para cada clase
239   for i in range(N):                # para cada imagen de la clase
240     # Lectura de la imagen
241     img    = ImageLoad('arrows_training/arrow',j+1,i+1,echo='on')
242     # Extracción de características
243     R      = (img>0.5)*1            # segmentation
244     f = extract_features(fx,bw=R)   # feature extraction
245     Xtrain[t,:] = f
246     ytrain_gt[t] = j
247     t = t+1
248 #===============================================
249
250 #===============================================
251 # Extracción de momentos de Hu para el testing
```

```
252
253 Nt = 10 # número de imágenes por clase en el testing
254
255 Xtest = np.zeros((K*Nt,M))          # features
256 ytest_gt = np.zeros((K*Nt,))        # ground truth (clasificacion ideal)
257
258 t = 0
259 for j in range(K):                  # para cada clase
260   for i in range(Nt):               # para cada imagen de la clase
261     # Lectura de la imagen
262     img    = ImageLoad('arrows_testing/arrow',j+1,i+1,echo='on')
263     # Extracción de características
264     R      = (img>0.5)*1             # segmentation
265     f = extract_features(fx,bw=R)    # feature extraction
266     Xtest[t,:] = f
267     ytest_gt[t] = j
268     t = t+1
269 #=====================================================
270
271 #=====================================================
272 # Distribución de frecuencias por clase
273 # Los momentos de Hu son 7, y se han almacenados como 7 columnas en la matriz
274 # Xtrain, el primer momento de Hu es la columna cero:
275 hu = 0 # 0 es el primer momento, 1 es el segundo momento, etc.
276 sns.displot([Xtrain[0:12,hu],Xtrain[12:24,hu],Xtrain[24:36,hu]],bins=20)
277 #=====================================================
278
279 #=====================================================
280 # Clasificacion
281 # Claramente las clases se separan con dos umbrales que en este caso se escogen
282 # de manera manual:
283
284 th1 = 0.20
285 th2 = 0.28
286
287 # Clasificación en el Testing (K x Nt = 30 muestras, 10 por cada una de las tres
    clases)
288 xtest  = Xtest[:,hu]
289 n = K*Nt
290 ytest  = np.zeros((n,))
291 for i in range(n):
292   if xtest[i]<th1:
293     ytest[i] = 2
294   elif xtest[i]<th2:
295     ytest[i] = 1
296   # otro elif no es necesario porque ytest[i] se inicializó en cero
297 #=====================================================
298
299 #-----------------------------------------------------
300 # Evaluación de desempeño
301 #-----------------------------------------------------
302 #=====================================================
303 C = confusion_matrix(ytest_gt, ytest)
304 print('Matriz de Confusión:')
305 print(C)
306 print(' ')
307 Acc = accuracy_score(ytest_gt, ytest)
308 print('Accuracy:')
309 print(Acc)
310 #=====================================================
311
312 #=====================================================
313 #=====================================================
314 #=====================================================
```