

```

1  #-----
2  #-----
3  #Ejemplo: Extracción de Descriptores de Fourier
4  #En este ejemplo se muestra cómo usar los Descriptores de Fourier
5  #para separar figuras geométricas con formas distintas.
6  #-----
7  # NOTA: ESTE NO ES UN EJEMPLO DE CLASIFICADOR (se sugiere desarrollar)
8  #-----
9
10 #-----
11 #0. Setup inicial
12 #
13 # Instalación de PyXvis
14 #-----
15 !wget https://www.dropbox.com/...../pyxvis.zip #CARGAR SU LIBRERIA
16 !unzip pyxvis.zip
17 !rm pyxvis.zip
18 print('PyXvis library downloaded.')
19 !pip install scipy==1.2
20 !pip3 install pybalu==0.2.5
21 !pip install ./pyxvis
22 print('PyXvis library installed.')
23 #=====
24 #-----
25 # Setup de librerías
26 #-----
27 #=====
28 import numpy as np
29 import matplotlib.pyplot as plt
30 from pyxvis.features.extraction import extract_features
31 from pyxvis.io.plots import plot_features
32 #=====
33
34 #-----
35 # Ejemplo: Reconocimiento de 4 caracteres
36 # En este ejemplo se realiza la extracción de características necesarias
37 # para realizar un reconocimiento de los siguientes cuatro caracteres con
38 # diversos tamaños y diversas orientaciones.
39 #
40 # Clase 0: punto '.'
41 #
42 # Clase 1: tilde '~'
43 #
44 # Clase 3: asterisco '*'
45 #
46 # Clase 4: coma ','
47 #
48 # Para la solución se utiliza los descriptores ya que son invariantes a la rotación
49 # y a la escala.
50 # En este ejemplo no se realiza la clasificación, sólo se muestra la extracción de
51 # descriptores de Fourier.
52 #-----
53
54 #-----
55 #1. Carga de Imágenes
56 #
57 #La base de datos consiste en 4 clases y 30 imágenes por clase
58 #-----
59 #=====
60 # Carga de base de datos
61 !wget https://www.dropbox.com/...../chars.zip
62 !unzip chars.zip
63 #=====
64
65 #-----
66 # Funciones necesarias
67 #-----

```

```

68 #=====
69 def num2fixstr(x,d):
70     # example num2fixstr(2,5) returns '00002'
71     # example num2fixstr(19,3) returns '019'
72     st = '%0*d' % (d,x)
73     return st
74
75 def ImageLoad(prefix,num_char,num_img,echo='off'):
76     # img = ImageLoad('example/char_',1,3) loads image 'example/char_01_003.png'
77     # img = ImageLoad('example/char_',2,15) loads image 'example/char_02_015.png'
78     st = prefix + num2fixstr(num_char,2) + '_' + num2fixstr(num_img,3) + '.png'
79     if echo == 'on':
80         print('loading image '+st+'...')
81     img = plt.imread(st)
82     return img
83 #=====
84
85 #-----
86 # Extracción de Descriptores de Fourier
87 #-----
88 #=====
89 K = 4 # número de clases
90 N = 30 # número de imágenes por clase
91
92 fx = ['fourierdes'] # características a extraer
93 M = 16 # los descriptores son 16
94
95 # si se quiere usar momentos de Flusser se puede usar fx = ['flusser'] y M = 4
96
97 X = np.zeros((K*N,M)) # K x N muestras (filas), y M características
98 # (columnas)
99 y = np.zeros((K*N,)) # ground truth (clasificación ideal)
100
101 t = 0
102 for j in range(K): # para cada clase
103     for i in range(N): # para cada imagen de la clase
104         # Lectura de la imagen
105         img = ImageLoad('chars/char_',j+1,i+1,echo='on')
106         # Extracción de características
107         R = (img>0.5)*1 # segmentation
108         f = extract_features(fx,bw=R) # feature extraction
109         f = f/np.linalg.norm(f) # normalización
110
111         X[t,:] = f
112         y[t] = j
113         t = t+1
114
115 #-----
116 # Visualización de distribución de clases
117 #-----
118 #=====
119 f1 = 3 # Descriptor de Fourier f1
120 f2 = 5 # Descriptor de Fourier f2
121
122 print('Descriptores de Fourier: '+str((f1,f2)))
123
124 Xs = X[:,[f1,f2]]
125
126 plot_features(Xs,y,'Descriptores Fourier',show=1)
127 #=====
128
129 #-----
130 #Del gráfico anterior, se puede pensar que un clasificador podría ser el siguiente:
131 #
132 # si x2 es mayor que 0.6 entonces es clase 2
133 #
134 #en caso contrario, si x1 es menor que 0.3 entonces es clase 3
135 #

```

```

136 #en caso contrario, si x1 es mayor que 0.6 entonces es clase 0
137 #
138 #en caso contrario es clase 1
139 #-----
140
141 #-----
142 # Matriz de similitud
143 #-----
144 #Como cada caracter ha sido descrito como un vector (de 16 elementos) de norma uno,
se puede realizar un producto punto de todos con todos. De esta manera los productos
que sean cercanos a 1 indican que esos vectores son similares.
145 #
146 #En esste ejemplo, en la matriz X de 120x16 elementos, existen 30 vectores por cada
caracter, de esta manera los productos punto entre vectores de la misma clase deben
dar cercanos a 1 (toda la diagonal es 1.0), mientras que los productos punto de
vectores entre distintas clases deben dar valores mucho menores que 1. Se observa
que la clase 3 con la clase 1, sin embargo, entrega vectores similares (el producto
punto es aprox. 0.84).
147 #-----
148 #=====
149 D = np.dot(X,X.T)
150 fig = plt.figure()
151 ax = fig.add_subplot(111)
152 implot = plt.imshow(D,cmap='jet')
153 #=====
154
155 #-----
156 #En este mapa de calor, colores parecidos al rojo indican valores cercanos a 1,
mientras que colores cercanos al azul indican valores cercanos a 0.
157 #
158 #Un clasificador podría diseñarse usando vecinos más cercanos, es decir para una
nueva imagen de testing, se extrae el vector de Fourier normalizado en 1, se hace el
producto punto con todas las filas de X y se escoge aquella clase de la mayoría de
votos de las k muestras más similares de X.
159 #-----
160
161 #=====
162 #=====
163 #=====

```