

RECONOCIMIENTO DE FLORES

**Carlos Eduardo Hincapié
Juan Manuel Sánchez Pareja
Alejandro Agudelo Toro**

TIPOS DE FLORES



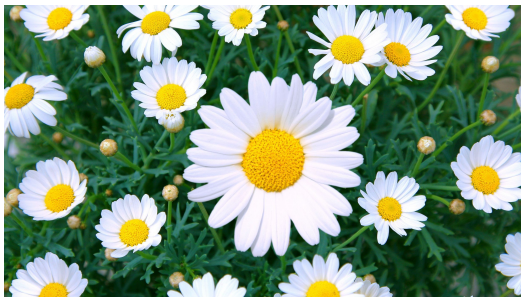
sunflower



dandelion



Rose



Daisy



Tulip

LIBRERÍAS

```
import sys
import os
from tensorflow.python.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras import optimizers
from tensorflow.python.keras.models import Sequential
from tensorflow.python.keras.layers import Dropout, Flatten, Dense, Activation
from tensorflow.python.keras.layers import Convolution2D, MaxPooling2D
from tensorflow.python.keras import backend as K
from tensorflow.keras.optimizers import Adam
K.clear_session()
```

PARÁMETROS DE ENTRENAMIENTO

```
17  """
18  Parameters
19  """
20  epocas = 10
21  longitud, altura = 150, 150
22  batch_size = 32
23  pasos = 100
24  validation_steps = 30
25  filtrosConv1 = 32
26  filtrosConv2 = 64
27  tamaño_filtro1 = (3, 3)
28  tamaño_filtro2 = (2, 2)
29  tamaño_pool = (2, 2)
30  clases = 5 #Numero de clases de flores
31  lr = 0.0004 #Learning Rate
```

CLASIFICACION DE IMAGENES

```
34  ##Preparamos nuestras imagenes
35
36  entrenamiento_datagen = ImageDataGenerator(
37      rescale=1. / 255,
38      shear_range=0.2,
39      zoom_range=0.2,
40      horizontal_flip=True)
41
42  test_datagen = ImageDataGenerator(rescale=1. / 255) #Validacion datagen
43
44  entrenamiento_generador = entrenamiento_datagen.flow_from_directory(
45      data_entrenamiento,
46      target_size=(altura, longitud),
47      batch_size=batch_size,
48      class_mode='categorical')
49
50  validacion_generador = test_datagen.flow_from_directory(
51      data_validacion,
52      target_size=(altura, longitud),
53      batch_size=batch_size,
54      class_mode='categorical')
```

RED NEURONAL

```
58 cnn = Sequential()
59 cnn.add(Convolution2D(filtrosConv1, tamaño_filtro1, padding = "same", input_shape=(longitud, altura, 3), activation='relu'))
60 cnn.add(MaxPooling2D(pool_size=tamaño_pool))
61
62 cnn.add(Convolution2D(filtrosConv2, tamaño_filtro2, padding = "same")) #activation='relu'
63 cnn.add(MaxPooling2D(pool_size=tamaño_pool))
64
65 cnn.add(Flatten())
66 cnn.add(Dense(256, activation='relu'))
67 cnn.add(Dropout(0.5))
68 cnn.add(Dense(clases, activation='softmax'))
69
70 <
71 > cnn.compile(loss='categorical_crossentropy',
72               optimizer=optimizers.Adam(lr=lr),
73               metrics=['accuracy'])
74
75 > cnn.fit_generator(
76     entrenamiento_generador,
77     steps_per_epoch=pasos,
78     epochs=epocas,
79     validation_data=validacion_generador,
80     validation_steps=validation_steps)
```

PREDICCIÓN

```
12 def predict(file):
13     x = load_img(file, target_size=(longitud, altura))
14     x = img_to_array(x)
15     x = np.expand_dims(x, axis=0)
16     array = cnn.predict(x)
17     result = array[0]
18     answer = np.argmax(result)
19     if answer == 0:
20         print("La imagen predecida fue un margarita")
21     elif answer == 1:
22         print("La imagen predecida fue un diente de leon")
23     elif answer == 2:
24         print("La imagen predecida fue un rosa")
25     elif answer == 3:
26         print("La imagen predecida fue un girasol")
27     elif answer == 4:
28         print("La imagen predecida fue un tulipan")
29     return answer
```


PRUEBAS

