Classification en utilisant Convolutional Neural Network

Ce projet a pour but de comprendre le CNN à travers l'échange avec les participants du workshop si vous voulez d'avantage de ressources je vous invite à regarder le github lié à ce projet. (https://github.com/quentin-ducheix/CNN)

I. Téléchargeais les images que tu veux entrainer

Utiliser des adons Google chrome pour télécharger les images de Google image ou utiliser le dataset que je vous mets à disposition dans le lien ci-contre : https://drive.google.com/drive/folders/1XaFM8BJFligrqeQdE-_5Id0V_SubJAZ

II. Import et Initialisation

```
# Importing openCV
import cv2

# Importing the Keras libraries and packages
import numpy as np
from keras.preprocessing import image
from keras.models import Sequential
from keras.layers import Conv2D
from keras.layers import MaxPooling2D
from keras.layers import Flatten
from keras.layers import Dense

# Initialising the CNN
classifier = Sequential()
```

III. Convolution

```
classifier.add(Conv2D(32, (3, 3), input_shape = (64, 64, 3), activation = 'relu'))
```

IV. Pooling

```
classifier.add(MaxPooling2D(pool_size = (2, 2)))
```

V. Deuxième layer de convolution

```
classifier.add(Conv2D(32, (3, 3), activation = 'relu'))
classifier.add(MaxPooling2D(pool_size = (2, 2)))
```

VI. Flattening

```
 classifier.add(Flatten())
```

VII. Full connection

```
 classifier.add(Dense(units = 128, activation = 'relu'))
classifier.add(Dense(units = 1, activation = 'sigmoid'))
```

VIII. Compiler le CNN

```
 lassifier.compile(optimizer = 'adam', loss = 'binary_crossentropy', metrics =
['accuracy'])
```

IX. Preparation des images pour notre CNN

```
from keras.preprocessing.image import ImageDataGenerator

train_datagen = ImageDataGenerator(rescale = 1./255,
                                   shear_range = 0.2,
                                   zoom_range = 0.2,
                                   horizontal_flip = True)

test_datagen = ImageDataGenerator(rescale = 1./255)

training_set = train_datagen.flow_from_directory('dataset/training_set',
                                                 target_size = (64, 64),
                                                 batch_size = 32,
                                                 class_mode = 'binary')

test_set = test_datagen.flow_from_directory('dataset/test_set',
                                            target_size = (64, 64),
                                            batch_size = 32,
                                            class_mode = 'binary')

classifier.fit_generator(training_set,
                         steps_per_epoch = 100,
                         epochs = 2,
                         validation_data = test_set,
                         validation_steps = 50)
```

X. Faire de nouvelles prédictions

```
image_tested = 'dataset/single_prediction/cat_or_dog_1.jpg'

test_image = image.load_img(image_tested, target_size = (64, 64))
test_image = image.img_to_array(test_image)
test_image = np.expand_dims(test_image, axis = 0)
result = classifier.predict(test_image)
training_set.class_indices

if result[0][0] == 1:
    prediction = 'dog'
else:
    prediction = 'cat'

#image dispaly
print("its a " + prediction)
img = cv2.imread(image_tested)
cv2.imshow(prediction, img)
cv2.waitKey()
```