USER MANUAL

Image Recognition with Machine Learning on Python

Created by Bîscă Emanuel (emanuel.bisca@stud.ubbcluj.ro) on November 2020

MACHINE LEARNING

TO DO

Step 1 [Time required 10 minutes]

Make sure you have installed on your computer:

- Python 3.5 or a recent version.
- Keras API.
- Anaconda open-source distribution of the Python.
- Pycharm as an IDE (I personally use this)

Step 2 [Time required 20 minutes]

Details

- Correctly importing the libraries is a crucial phase.
- Check if every image is of the size 224 / 224.
- Check the data format, so the input shape will be feeded accordingly.

Step 3 [Time required 60 minutes]

The network

- We will need an activation function.
- *Conv2D* would be the layer to convolve the one image into multiple images.
- *Flatten* is the one that flatten the dimensions of the image obtained after the convolving step.
- *Dense* is used to make a fully connected model and is the hidden layer. The output layer would contain only one neuron which decide to which category image belongs.
- *Dropout* helps to avoid unwanted overfitting on the considered dataset.

Step 4 [Time required 25 minutes]

Using *DataGenerator*

- *ImageDataGenerator* is extremely important because rescales the image, applies shear in some range, zooms the image and does horizontal flipping with the image. This *ImageDataGenerator* includes all possible orientation of the image.

Step 5 [Time required 20 minutes]

Interpret the results

- Give attention to the accuracy values and to the loss ones as well.
- Additionally, the model can be saved.

NECESSARY LIBRARIES

Keras TensorFlow

from **keras.preprocessing.image** import *ImageDataGenerator*

from **keras.models** import *Sequential*

from **keras.layers** import *MaxPooling2D* import *Activation* import *Dropout* import *Flatten* import *Dense*

WHY KERAS?

Keras is an open-source library that provides a Python interface for artificial neural networks.

Keras acts as an interface for the TensorFlow library.

Up until version 2.3 Keras supported multiple backends, including TensorFlow.

PERFORMANCE

Overall Accuracy Loss