We need to stop for a while and really understand why image has to be preprocess?

Images are composed of pixel for which values are between [0-255]. Most of signal processing and Machine Learning algorithms prefer data between [0-1] to avoid numerical overflows. There are also specific reasons for Deep Learning where some activation functions are not compatible with all kinds of input.

Furthermore, an RGB image is composed of 3 channels. We need to make sure that pixels from all these channels are in the same range of values. Otherwise, it will have some performances impact during our training such as lowest values would be discarded.

To make it short, we can not feed a Deep Neural Network with raw data, data has to be prepared and we call that step **Data Preprocessing**.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

We need to stop for a while and really understand why image has to be preprocess?

Images are composed of pixels for which values are between [0-255]. Most of signal processing and Machine Learning algorithms prefer normalized data (with ranges such as [0-1] or [-1,1]) to avoid numerical overflows. There are also specific reasons for Deep Learning where some activation functions are not compatible with all kinds of input.

~~Furthermore, an RGB image is composed of 3 channels. We need to make sure that pixels from all these channels are in the same range of values. Otherwise, it will have some performances impact during our training such as lowest values would be discarded.~~ RGB channels are already on the same range

To make it short, It is better to feed DNN with preprocessed normalized data ~~we can not feed a Deep Neural Network with raw data,~~ data has to be prepared and we call that step **Data Preprocessing**.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

That's why in Keras, when you load a model from keras.applications.mobilenet import MobileNet, preprocess\_input, you load the preproocessing too!

The preprocessing function is provided to the ImageDataGenerator so that it knows how to prepare data for the model.

The transformed data are no more looking conventional like images. That's why we need to unpreprocess the data to look like again to images between [0-255] and to be then displayed.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

That's why in Keras, when you load a model you load the preproocessing too!

(from keras.applications.mobilenet import MobileNet, preprocess\_input)

The preprocessing function is provided to the ImageDataGenerator so that it knows how to prepare data for the model. ImageDataGenerator generates batches of preprocessed data that’s why data formatted in batches do not match the expected format by usual image display function

~~The transformed data are no more looking conventional like images.~~ That's why we need to unpreprocess the data back to [0-255] range if we want to display related images ~~to look like again to images between [0-255] and to be then displayed.~~

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

TWO Things to remember :

1 When feeding a pretrained net you need to preprocessed data the same way it has been done during training

2 Mean and standard deviation used in the preprocessing function are the ones computed on the training dataset

When using pretrained nets from keras model zoo,