Franco Bueno Mattera
CS 777 O1
Big Data Analytics
Term Project Final Report

Image Classification using Apache pyspark

Project Description

The main purpose of this study is to attempt to perform image classification using pyspark mllib embedded classification algorithms and compare its performance with a classical image classification library and algorithm.

The second objective of this study Is to explore the feasibility in terms of configuration and code logic, of using pyspark mllib in order to perform image classification tasks

Methodology:

1. About the Dataset:

The dataset of choice was the Arboles de Chile Dataset from Kaggle Website. It contains over 6000 images of different trees present in all regions of chile. The original image size is 300, 300, 3 where the last number represents the colors.

For this particular study, only 3 classes were selected which were associated with 3 different species: a)Peumus boldus (Boldo), a native tree whose leaves are used to prepare tea and other beverages, b) Lithraea causita (Litre), a tree that is very similar to the one described before but it is extremely toxic and has a similar toxin to the one found on poison ivy c) Ulmus Americana (Ulmo) A highly priced tree by native and invasive bees who make a very special white honey. The images also came with label descriptions in csv files for both test and train sets.

2. Preparation of the dataset:

The images used for training and test sets were already separated into two folders. Additional transformations were necessary in order to make sure that the order of the images on the folders matched the order of the labeling.
The size of the images was reduced from (300,300,3) to (64, 64, 3) in order to fit the models in memory.
Due to the lack of proper methods on mllib to work directly with images, those were manually transformed into array, flattened and transformed into dense vectors (refer code for more details)

3. Training The model:

To train the dataset, two different Mllib algorithms were chosen: Multilayer perceptron classifier (a simple artificial neural network) and Naive Bayes Classifier. The latter was given a chance due to itsfast computational time and low memory usage.
Finally, a pre trained convolutional network was applied to the data and their results compared with the previous two methods in Keras/Tensorflow libraries. There is not much need to code the implementation of this step. Keras provides a very sophisticated API that is applicable to most cases with just replacing a few parameters. Unfortunately, this option is not available on pyspark

Results:

The classification metrics did not yield good results. This was also the case for the pre-trained model.
Additionally, attempting to imitate a well known pre-trained network was not possible due to performance limitations.

Multilayer perceptron classifier:

Performance Metrics: Multilayer Perceptron Classifier - Manual Tuning
Precision: 0.31333333333333335
Recall: 0.3
F1: 0.3047138047138047
Confusion Matrix: [[3. 4. 3.]
[3. 4. 5.]
[4. 2. 2.]]

Naive Bayes Classifier:

Precision: 0.12000000000000001
Recall: 0.13333333333333333
F1: 0.125
Confusion Matrix: [[0. 5. 5.]
[0. 1. 2.]
[0. 4. 3.]]

Transfer Learning: Vg16

Precision: 0.3
Recall: 0.3
F1: 0.26666666666666666

Discussions:

Transfer learning is a very powerful way to train a model with a small dataset and get good results, as it is suggested here.
One of the possible explanations for poor results is that their amount of data is very limited. There were approximately 6000 images available on the training set but only about 100 per species. This forces them to rotate images in order to increase the number of samples.

Another possibility is that image identification of plant species could be much more complicated than identifying a cat from a dog. It can take a good amount time for someone to learn how to differentiate between certain species of plants, and even with certain identification guides (taxonomy guides) it could be challenging. By taking a simple look at certain plant classification guides such as plant snap, it could be observed that their performance is far from ideal and probably (hopefully) no one will use for serious species classification tasks or to differentiate between an edible or poisonous plant.

Of the lessons learned and additional observations

The most important lesson learned in this study is the complexity of preparing the data without the proper support of a dedicated image classification library.

It is certainly possible to enhance pyspark to use GPU in some complex tasks but most of the documentation is aimed at distributed systems and not at local level.

UDF could be a promising way of implementing some of these algorithms and this was certainly attempted (with transfer learning). However, this approach was not possible to implement at this time, more research is required.

Memory limitations were a constant issue, even with powerful machines. CPU issues were also present but to a smaller degree. It would require an extremely big set up and a much greater train set in order to generate good results.

There is a library(sparkdl) that takes advantage of transfer learning for image classification. Unfortunately it has not been updated since 2018.