Final Project

Classification of Big Cats from Google Image Dataset

DATA 603 Platforms for Big Data Processing

University of Maryland – Baltimore County

Prof. James Klucar

Team 5

Kumra, Shouray (skumra1)

Mironov, Sanaa (has1)

Purushothaman, Govind Ram (gx54176)

Thakkar, Mahima Ashokbhai (db59735)

**INTRODUCTION**

For the project, we have chosen to classify the Big Cats such as Jaguar, Lynx, Tigers, Lions, Leopards and Cheetahs from the Google Image Dataset using the ResNet50 Model.

**DATASET**

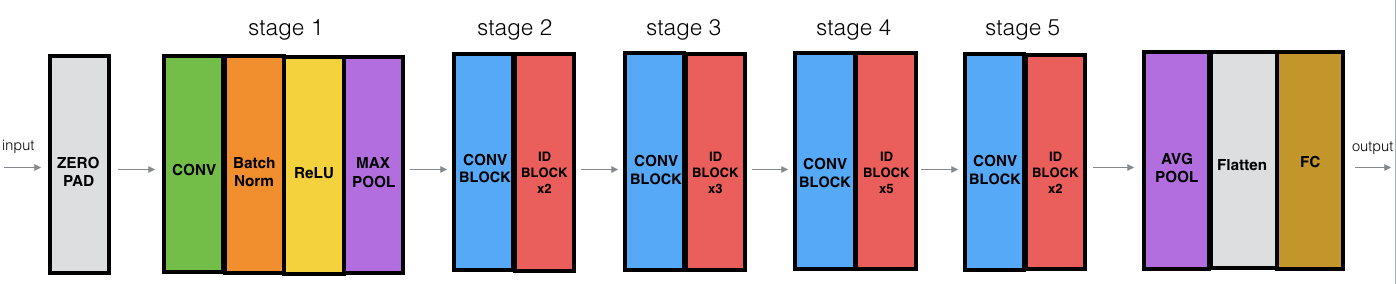
The Google Open Images V5 features segmentation masks for 2.8 million object instances in 350 categories. Unlike bounding-boxes, which only identify regions in which an object is located, segmentation masks mark the outline of objects, characterizing their spatial extent to a much higher level of detail. These masks cover a broader range of object categories and a larger total number of instances than any previous dataset.

The segmentation masks on the training set (2.68M) have been produced by an interactive segmentation process, where professional human annotators iteratively correct the output of a segmentation neural network. This is considered efficient than manual drawing alone, while at the same time delivering accurate masks. In addition to the masks, 6.4M new human-verified image-level labels are present, reaching a total of 36.5M over nearly 20,000 categories

Why ResNet50?

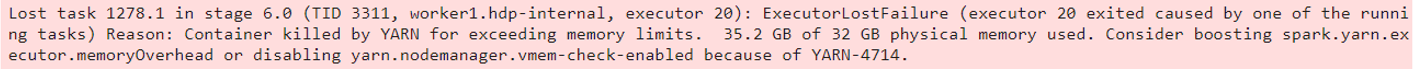
ResNet50 is 50 layers deep, pre-trained Deep learning model for image classification of the Convolutional Neural Network, mostly used to analyze visual imagery. Using a pre-trained model is a more effective approach than to collect loads of data and train the model our self. It is trained on a million images of 1000 categories from the ImageNet database, and the model has over 23 million trainable parameters, which indicates a deep architecture that makes it better for image recognition and classification. Also, ResNet50 has excellent generalization performance with fewer error rates when compared to other pre-trained models like AlexNet, GoogleNet or VGG16.

Further, in a deep convolutional neural network like VGG16, several layers are stacked and trained to the task, and the network learns several low/mid/high level features at the end of its layers. Whereas in a residual learning, instead of trying to learn some features, we try to learn some residual by subtracting the feature learned from the input of the layer. This is done by directly connecting the input of the nth layer to some (n+x) th layer. Training these kinds of networks is easier than training simple deep convolutional neural networks and the problem of degrading accuracy is also resolved.

Architecture of ResNet50

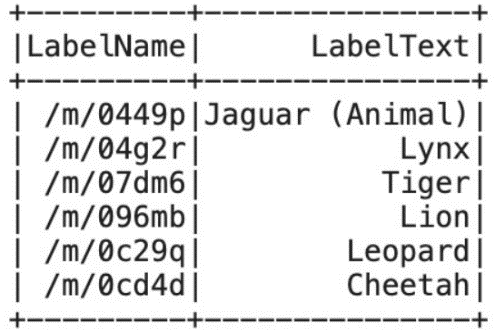
Errors faced

* Container killed by YARN for exceeding memory limits.

We could not upload our .parquet file as it exceeded the physical memory of 20GB initially allocated by us. We then discovered that this was not enough and increased the capacity to 50GB. 



Code breakdown

1. *Creating the .parquet file*

Created Dataframe with the Labels and Metadata from the Google Open Image Dataset to search and retrieve the required images with their corresponding labels.

2. *Evaluating*

Read the parquet file and checked for the count of images which contained the required labels.

3. *Resizing Images*

Created function evaluate\_chip to pass chipdata which is a column of our dataframe image\_chips which contains binary values.

The function was exectuted in 5 steps

a) Load the image – Resize the image in 224x224

b) Prepare image – Reshape the image into 1???

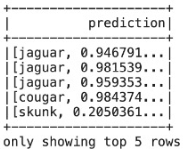
c) Load the data – Used .h5 file i.e. MobileNet\_1\_0\_224\_tf.h5 from MobileNet pre-trained model.

d) Prediction – The reshaped image from step 2 was passed to predict the result.

e) Decode prediction – The prediction was passed in the pre-defined library (decode\_predictions).

4. *Creating UDF*

The function evaluate\_chip is passed through a pre-defined function (UDF) to fetch the decoded predictions.

5. *Evaluating Image Chips*

The results of step 4 is added as a new column called ‘Prediction’ to the image\_chip dataframe.

Conclusion

We can conclude that MobileNet predicted good results for few labels.

Reference

<https://towardsdatascience.com/the-w3h-of-alexnet-vggnet-resnet-and-inception-7baaaecccc96>

<https://medium.com/@nina95dan/simple-image-classification-with-resnet-50-334366e7311a>