Bengaluru's Favourite Car Colour

Rhythm Girdhar - 01FB16ECS302- ridhigirdhar3@gmail.com

Rohan Mohapatra- 01FB16ECS307- rohan.ron14@gmail.com

Rohan Rajesh Talesara- 01FB16ECS309- rtalesara77@gmail.com

Saloni Govil- 01FB16ECS333- salonigovil04@gmail.com

Overview

Our assignment was to **Find the most common car colour** in Bangalore Urban. We collected a sample of 1000+ images from the streets/roads and extracted the images of cars and classified them based on their colour using tools in Python and segregated them into their identified colours. We finally used Hadoop map and reduce algorithm to find the frequency of each colour.

Learning Outcome

The assignment completely aims at giving us a hands-on experience with various extraction tools, image processing and clustering Algorithms and using Hadoop to work on large amount of data.

Goals

Our main aim was to find the total number of cars of each color using Hadoop and OpenCV. These are the goals we followed:

- To collect 1000+ images of cars around us.
- To extract the cars from the collected dataset.
- To remove background clutter and find the dominant car colour.
- To run a clustering algorithm and separate the cars based on the colour identified.
- To compile them into text files and run map and reduce algorithm on Hadoop.
- To get the final frequencies of all the given colours obtained.

Tools Required

Programming Languages used:

- Python - For Image Extraction and clustering into different colors.
- Java - For applying the Map - Reduce Algorithm to easily find the frequency of each color from the given files.

Technologies used:

- Hadoop for handling huge amount of Data (map and reduce)
- K-Means Clustering Algorithm for quantization and color clustering.
- Numerous Python Libraries - PIL(Image), SciPy, Numpy, SKLearn, os

Detailed Description:

Dataset collection:

- We used our phone cameras to collect photos of cars on the streets/roads.
- We targeted Bangalore Urban.
- The images were all resized to 500px X 288px

Image extraction:

- Haar Cascades were used to extract the cars from the above images.
- The extracted images were resized to 60px X 60px
- The extracted images had background clutter (such as roads). To remove the same we further cropped the image such that only the cars were visible.
- The images so obtained contained colours apart from the car's colour, such as the number plate colour, shadows etc.

Identifying the colors:

- In order to reduce the effect of these extra colours, we used k-means clustering to reconstruct the images using the 4 most dominant colours of the car.
- We further applied K-means clustering on all these images to cluster them into different colours. Raw pixel values were used as the feature set. The initial centroids were initialised manually.

Segregation:

- The data points were segregated into different clusters based on the initialised centroids.
- The result of the segregation was written onto a text file, i.e., the colour obtained was recorded in the text file.

Frequency calculation:

- The Map-reduce algorithm, written in Java, is compiled.
- Mapreduce: Map procedure is executed by mapper function. All the text in the input file (one color one word) is converted into key-value pairs, where key is the color and value is a counter which is one in this place. Reduce procedure is executed by reducer function. It gives the frequency count based on key-value pairs.
- The code is run with the final-general and final-specific text files as inputs on Hadoop.
- The Output files are downloaded and the value with highest frequency is recorded and declared as the most frequent color.
- Bar plots summarising the data were also generated.

Key points

- Completely automated: All steps in the procedure are carried out by a python scripts. The only part where human interference is needed is while initialising the centroids.
- Space efficient: The final colour quantized images have only 4 colours, and are of dimension 50px X 50px. Consequently, 1080 images take only ~400kb.
- Fast: The entire process runs under 5 seconds (including resizing, extraction, cropping, quantization, and clustering). This does not include Hadoop map and reduce.

References

- https://www.pyimagesearch.com/2014/05/26/opencv-python-k-means-color-clustering/
- http://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html
- https://mubaris.com/2017/10/01/kmeans-clustering-in-python/
- http://scikit-learn.org/stable/tutorial/statistical_inference/unsupervised_learning.html