Image Object Recognition Using Apache Hadoop and Python

Author: Jaileen Del Valle Maldonado
Advisor: Prof. Nelliud Torres Batista Ph.D
Electrical & Computer Engineering and Computer Science Department

Abstract

The amount of data generated by people each day on social media platforms is increasing at an alarming rate. Studies performed show that approximately 1.5 billion images are uploaded to the internet each day. Applications that can use and analyze this data are not available to all users due to limitations in processing power or storage space required for the analysis of these large datasets. Apache Hadoop is an open-source framework that allows distributed processing and fault tolerance of Big Data with the use of commodity hardware using Hadoop Distributed File System (HDFS) and MapReduce. Using HDFS data is stored in a distributed manner across different machines (datanotes). The use of the MapReduce framework parallelized computing is available and manageable to be able to mine and analyze the image data available created by users. The focus of this article will be the analysis of image data in large datasets to create feature vectors using the k-means algorithm to group together images that contain similar objects inside them using Apache Hadoop, MapReduce, Apache Spark, Computer Vision, and the Python programming language.

Introduction

The term Big Data is used to describe the huge volumes of data generated by digital processes. Currently, the use of social media sites has increased the amount of image data being uploaded every day on sites like Facebook, WhatsApp, and Twitter. This increasing amount of information growth has created a new kind of problem for data analysts. Analyzing and processing such huge amounts of data could create bottlenecks caused by the use of a single computer, power concerns, and the amount of storage space needed. The present-day computer architectures are reaching their physical limitations and with this, the implementation of distributed systems is becoming more widespread. The main reason to which the popularity of distributed systems can be attributed are: i) physical limitations of processors, ii) scalability, iii) fault tolerance iv) latency. [1] With the use of distributed systems tasks are completed by dividing a task into multiple subtasks. Dividing system tasks is known as parallelization, this makes applications running on a distributed system more scalable and efficient. A widely known distributed system platform is Apache Hadoop and the Hadoop MapReduce algorithm.
Hadoop is being used as a system for processing huge datasets by using parallel and distributed computing. In addition, various studies have been performed using Apache Hadoop an example of these are: conducting analysis of text files, examining DNA sequencing data, converting images to PDF, and feature extraction and selection. These studies are performed by dividing the data across multiple features like algorithm parameters, images, or pixels. The k-means algorithm has been implemented within the MapReduce programming framework to analyze images and classify them based on their color.

Background & Related Work

Hadoop is an open-source framework that works as a distributed system with a scalable, fault-tolerant design. It is used for data storage and processing.

Apache Hadoop Distributed File System
HDFS is a filesystem designed for storing large files in a distributed manner with streaming data access patterns. These are usually run-on commodity hardware. The architecture of HDFS makes it scalable but there are a few drawbacks: 1. HDFS works better performing long sequential reads from files, but it is not used for random reads 2. Caching is not the best since files contain a big overhead and data would be re-read from the source. 3. HDFS only performs appends to files there is no updating functionality [2].
The master component or master node in the HDFS architecture is called the NameNode. The NameNode stores metadata information like where each block is stored, and how many times the file is replicated within the system and tracks the DataNodes. The DataNode is where the files and data are stored in the system. The NameNode is the one that administers all the DataNodes in the cluster. This includes DataNode failure and heartbeat messages. A heartbeat is a message that includes information about activity within the cluster and DataNode failures. These messages are configured to be sent every three seconds [3].

Apache Hadoop MapReduce
MapReduce can be defined as a programming model used for processing data. MapReduce considers the problems of distributing the data in a network of computers to always assure that all available memory, processor, and storage are used in the most optimized manner. MapReduce works with parallel data processing using the map phase and the reduce phase. Both phases have input and output key-value pairs. Hadoop performs exceptionally well when all the data being processed is contained within a single DataNode in HDFS [3].
The execution of a MapReduce program or job can be summed up in the following diagram:

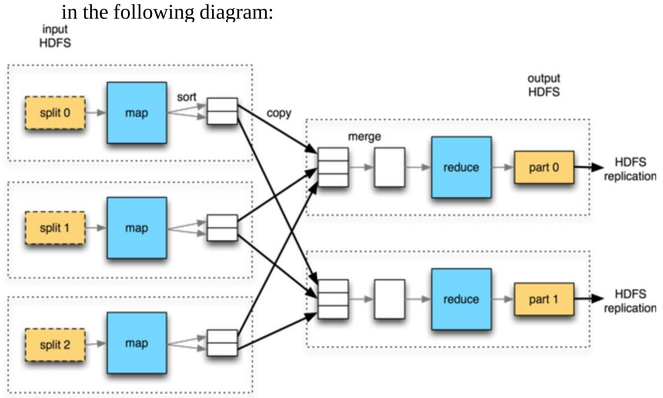

Figure 1
MapReduce Program Flow

Sequence Files
Sequence files are a Hadoop file format that stores binary key-value pairs in a sequential form. This type of file is splitable, supports compression, and can store arbitrary types using serialization frameworks [3].

Sequence files can be visualized as a container for multiple small files, see figure 2. In the case of images, the file name would be converted into the key of the sequence file and the value will be the binary content of the image or file. The creation of a sequence file must be performed using a MapReduce job.

Key	Value	Key	Value	Key	Value
file1.txt	file1 contents	file2.txt	file2 contents	fileN.txt	fileN contents

Figure 2
Visualization of a Sequence File

Feature Extraction using Descriptors
The SIFT was created by D.Lowe in 2004. The main goal of this algorithm is to extract features or descriptors from keypoints [6]. The features will be extracted and stored in HDFS as a sequence file having as the key the image name and the features as the value. The dimensions of the feature are separated by commas for simplicity of use. SIFT is a reliable algorithm since it is invariant to image scale and rotation. The descriptors of this algorithm were created for the sole purpose of image matching. Each feature vector descriptor is highly different which facilitates the process of matching with another feature vector within the file system.
The properties of an image that are commonly used for feature extraction are intensity, color, and texture. Consistency is also an important factor in feature detection and extraction since features must be detected even while an image has suffered changes like blurring, re-orientation, and re-escalation [7]. The SIFT algorithm can be divided into four steps: (1) keypoint localization, (2) orientation assignment, (3) keypoint descriptor, and (4) keypoint matching.

Feature Vector Clustering
Clustering allows the user to extract significant knowledge from the dataset. Clustering consists of partitioning the data of a dataset into different amounts of subsets or groups in a way that all the data that is similar end up together and the other unrelated data is grouped in other subgroups. K-means is defined as an unsupervised clustering algorithm [8]. The time complexity of the K-Means algorithm is O (nkt), where n refers to the number of datapoints, k is the number of clusters and t is the number of iterations [9].
The K-Means clustering algorithm can be divided into two main steps: (1) Cluster Assignment and (2) Move Centroid Step. These steps are repeated iteratively until one of the following conditions is reached; the centroids will not change their positions anymore or the iterations have gone through the maximum number.

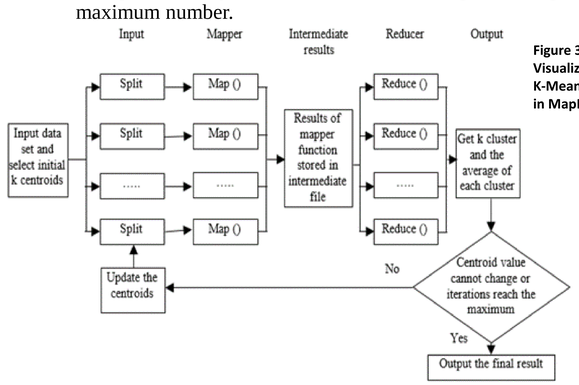

Figure 3:
Visualization of the K-Means algorithm in MapReduce

Methodology

Hadoop's principal components: Hadoop Common, HDFS, MapReduce, and YARN will be installed using a binary tarball which can be found on the Apache Software Foundation. Before installing Hadoop, the user must make sure they have Java installed on the computer. Additionally, another tool from the Hadoop ecosystem that needs to be installed is Apache Spark Enabling the use of Python with Spark requires installing Pyspark, this can be done using the command pip install pyspark [10]. The first step is to store in HDFS a text file that contains the paths of all the images that are going to be used for object recognition purposes.
After this file is stored in the distributed file system a MapReduce job must be performed to convert the images in the path into one SequenceFile. Afterward, another MapReduce job must be performed to read the image bytes from the images and save the image itself in HDFS After this process, the images are inputted to another function that creates and computes the keypoints and descriptors using the SIFT algorithm available in the computer vision library. Finally, the keypoints and descriptors created from the SIFT algorithm are filtered using a map function that groups the filenames with the features extracted.
A final pyspark script will cluster feature vectors by their similar properties using the K-Means Algorithm.

Conclusions

In conclusion, computer vision techniques and processes have been integrated with the use of the Hadoop environment to achieve a scalable algorithm capable of performing parallel tasks to manage Big Data applications such as image classification. This approach has been based on the Vector for Locally Aggregated Descriptors (VLAD) technique to generate an algorithm capable of recognizing image features. With the use of clustering algorithms like K-means through the Hadoop MapReduce function, the system can generate a dictionary of features or Bag of Visual Words (BoW) and then classify images based on this trained set for desired features.

References

1. N. Kumar, "Content Based Image Retrieval for Big Visual Data using Map Reduce - raiith", Raiith.iith.ac.in, 2015. [Online]. Available: https://raiith.iith.ac.in/1609/
2. C. Reggiani, "Scaling feature selection algorithms using MapReduce on Apache Hadoop", Claudioreggiani.com, 2013. [Online]. Available: http://claudioreggiani.com/pdf/masterthesis.pdf
3. T. White., Hadoop: The Definitive Guide, Second Edition. O'Reilly Media, Inc., 2010.
4. K. Potisepp, "Large-scale Image Processing Using MapReduce", semanticscholar, 2013. [Online]. Available: https://www.semanticscholar.org/paper/Large-scale-Image-Processing-Using-MapReduce-Potisepp/bb7ce436cc9e2b1c4c64516ae9f600dc517b7353
5. T. El-Sayed, A. El-Sayed and M. Badawy, "Impact of Small Files on Hadoop Performance: Literature Survey and Open Points", researchgate, 2019. [Online]. Available: https://www.researchgate.net/publication/337677872_Impact_of_Small_Files_on_Hadoop_Performance_Literature_Survey_and_Open_Points#:~:text=Hadoop%20performs%20well%20with%20files,of%20the%20MapReduce%20applications%20
6. D. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints", Cs.ubc.ca, 2004. [Online]. Available: https://www.cs.ubc.ca/~lowe/papers/ijcv04.pdf
7. W. Murphy, "Large Scale Hierarchical K-Means Based Image Retrieval With MapReduce", AFIT Scholar, 2014. [Online]. Available: https://scholar.afit.edu/etd/616/
8. S. Vemula and C. Crick, "Hadoop Image Processing Framework", ieeexplore, 2015. [Online]. Available: https://ieeexplore.ieee.org/document/7207264
[9] T. Habib and Z. Ansari, "An Analysis of MapReduce Efficiency in Document Clustering using Parallel K-Means Algorithm", researchgate, 2018. [Online]. Available: https://www.researchgate.net/publication/325208173_An_Analysis_of_MapReduce_Efficiency_in_Document_Clustering_using_Parallel_K-Means_Algorithm
[10] B. Chambers and M. Zaharia, Spark: The Definitive Guide: Big Data Processing Made Simple, 1st ed. Seoul: Hanbit Midieo, 2018.