ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 5, May 2015

Digital Image Classification and Clustering

Shashidhar.V, Aruna Kumara.B, Neelu L, Bharath J

Asst. Professor, Department of CSE, Raja Rajeswari College of Engineering, Bangalore, India

ABSTRACT-Digital images account for huge data in any industrial field such asInternet search, finance, etc. But in research area such as meteorology, genomics digital images play a crucial role, classification such of theimages which grows rapidly in terms of peta-bytes is a challenging task.Classifying the images against a category and processing those using clusters of computers will be designed and implemented in this. By making use of a software platform called hadoop which makes use ofdistributed file system which is used in data intensive applications. Data isprocessed in a distributed manner. Programs are developed using whichthe images are processed and categorized.

KEY WORDS: Digital image, Clusters, Hadoop, Data intensive and Web interface.

I. INTRODUCTION

Digital images represent numerous details about various subjects.Storing of the images and classifying them manually is a complex affair. Processing of digital images is data-intensive as well as process intensive. We can achieve optimal performance by distributing the job among several processors which can execute the tasks in parallel.

In imaging science, image processing is any form of signal processing for which the input is an image, such as a photograph or video frame; the output of image processing may be either an image or a set of characteristics or parameters related to the image. Most image-processing techniques involve treating the image as a two-dimensional signal and applying standard signal-processing techniques to it.

Image processing usually refers to digital image processing, but optical and analog image processing also are possible.An image defined in the “real world” is considered to be a function of two real variables, for example, a(x,y) with a as the amplitude (e.g. brightness) of the image at the real coordinate position (x,y).

Modern digital technology has made it possible to manipulate multi-dimensional signals with systems that range from simple digital circuits to advanced parallel computers. The goal of this manipulation can be divided into three categories:

```
Image Processing     (image in -> image out)
Image Analysis       (image in -> measurements out)
Image Understanding (image in -> high-level description out)
```

Closely related to image processing are computer graphics and computer vision. In computer graphics, images are manually *made* from physical models of objects, environments, and lighting, instead of being acquired (via imaging devices such as cameras) from *natural* scenes, as in most animated movies. Computer vision, on the other hand, is often considered *high-level* image processing out of which a machine/computer/software intends to decipher the physical contents of an image or a sequence of images (e.g., videos or 3D full-body magnetic resonance scans).

1.1 Classification Techniques

Classification of remotely sensed data is used to assign corresponding levels with respect to groups with homogenous characteristics, with the aim of discriminating multiple objects from each other within the image.

The level is called class. Classification will be executed on the base spectral or spectrally defined features such as density, texture, etc., in the feature space. It can be said that classification divides the feature space into several classes based on a decision rule.

In many cases, classification will be undertaken using a computer with the use of mathematical classification techniques.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 5, May 2015

Step 1: Definition of Classification Classes:-Depending on the objective and thecharacteristics of the image data, the classification classes should be clearly defined.
Step 2: Selection of Features- Features to discriminate between the classes should be established using multi-spectral and/or multi-temporal characteristics, textures etc.
Step 3: Sampling of training Data- Training data should be sampled in order to determine appropriate decision rules. Classification techniques such as supervised or unsupervised learning will then be selected on the basis of training data sets.
Step 4: Classification- Depending upon the decision rule, all the pixels are classified in a single class. There are two methods of pixel by pixel classification and per-field classification, with respect to segmented areas.Popular techniques are as follows:

- Multi-level slice classifier
- Minimum distance classifier
- Maximum likelihood classifier
- Other classifiers such as fuzzy set theories and expert systems

This provides an inroduction to MapReduce framework and a brief description. It contains various other information to enhance our knowledge about the topic. It lists all the user interfaces as well [1].

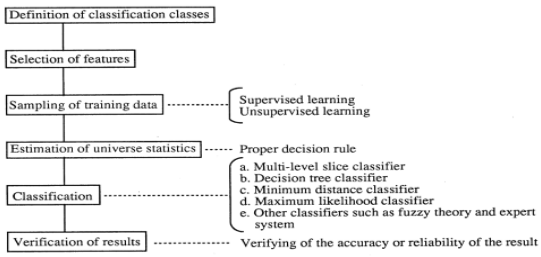


Figure 1.1: Procedures for Classification

The following methods are considered to determine a decision rule for classification:

1. **Supervised Classification**: In order to determine a decision rule for classification, it is necessary to know the spectral characteristics or features with respect to the population of each class. The spectral features can be measured using ground-based spectrometers. However due to atmospheric effects, direct uses of spectral features measured on the ground are not always available. For this reason, sampling of training data from clearly identified training areas, corresponding to defined classes is usually made for estimating the population statistics. This is called supervised classification. Statistically unbiased sampling of training data should be made in order to represent the population correctly.
2. **Unsupervised Classification**: In the case where there is less information in an area to be classified, only the image characteristics are used as follows.
   - Multiple groups, from randomly sampled data, will be mechanically divided into homogeneous spectral classes using a clustering technique.
   - The clustered classes are then used for estimating the population statistics. This classification technique is called unsupervised classification.

1.2 Objectives

The chief objective of the paper is to develop a digital image classifying application which runs on Hadoop software development framework and processes the image in a distributed manner and in parallel by utilizing the nodes present in the cluster. Image classification is interpreted as pixel classification, a process in which every pixel in an image is assigned to a class or category on the image. The following are the main activities which are carried out in order to achieve the objective of parallel execution:

- Create signatures for the images which describe the classes to which the pixel belongs as parallelepiped image classification is a supervised image Implementing the parallelepiped classification algorithm using the Hadoopframework in the mapper phase.
- Obtaining the classified image in its entirety from the map/reduce phase.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 5, May 2015

II. LITERATURE SURVEY

This literature survey represents the features and working methodologies which are already implemented in the existing system, their drawbacks, and the possible deficienies that could be rectified and improvised.

Most classification methods, both supervised and unsupervised rely on some distance measure that is calculated over the pixels' values and not its coordinates in the image.Regardless of the method used for classification, the idea is that if one pixel is assigned to a particular class, pixels similar to those should probably be assigned to the same class. The existing systems for digital image classification are implemented on standalone systems irrespective of the algorithms used for classification.

Its detailed guide to MapReduce and Hadoop framework. It contains information that can be used by beginners as well as intermediate developers to understand the framework, develop and implement some of the basic applications [2].

Its guide on setting up Apache-Hadoop on Ubuntu for a single node cluster and executing some sample programs such as word count and character count map/reduce programs. This has provided useful information in configuring and understanding basic Hadoop operations. Configuring our Hadoop systems was easy and effective using this [3].

Its a guide on setting up Apache-Hadoop on Ubuntu for a multi node cluster and executing some sample programs in the multiple nodes. All the systems that intend to run map/reduce tasks were configured. It provided us useful insight into the organisation of clusters and requirements for it [4].

Description of java packages are provided in this database. This contains the entire collection of Java built-in methods, collections, frameworks, API, etc. [5].

Description of the API for Hadoop and map/reduce operations are provided in this database [6].

Work at University of Washington, Astronomy Survey Science Group on Astronomical Image Processing with Hadoop discusses how Hadoop can be used to implement Image Coaddition of images taken from satellite cameras of 3.2 Gpixel. A total of 30 TB of information is collected every night. It is about multiple partially overlapping images and which images intersect based on the query bounds. This task was achieved by them using HadoopMapReduce [7].

They used a cluster of 700 nodes where each node consisted of two 2.8GHz Intel Xeon dual core processors, which makes it four cores per node. Each node had 8GB of RAM and two disks of 400GB each. High quality coadd images were obtained by them.

Another work by Mohamed H Almeer on HadoopMapRecude for Remote Sensing Image Analysis aims to provide an image processing interface for high volume images. This paper has compared the advantages of Hadoop image processing over the traditional non Hadoop approach to image processing [8].

III. ARCHITECTURE & WORKING METHODOLOGY

The Apache Hadoop project develops open-source software for reliable, scalable, distributed computing. This Paper includes these modules:

- Hadoop Common: The common utilities that support the other Hadoop modules.
- Hadoop Distributed File System (HDFS): A distributed file system that provides high-throughput access to application data.
- Hadoop YARN: A framework for job scheduling and cluster resource management.
- HadoopMapReduce: A YARN-based system for parallel processing of large data sets.

3.1 Hadoop Distributed File System

The Hadoop Distributed File System (HDFS) is a distributed file system designed to run on commodity hardware. It has many similarities with existing distributed file systems. However, the differences from other distributed file systems are significant. HDFS is highly fault-tolerant and is designed to be deployed on low-cost hardware. HDFS provides high throughput access to application data and is suitable for applications that have large data sets. HDFS

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 5, May 2015

relaxes a few POSIX requirements to enable streaming access to file system data. HDFS was originally built as infrastructure for the Apache Nutch web search engine project. HDFS is part of the Apache Hadoop Core project.

HDFS has a master/slave architecture as shown in Figure 1.2. An HDFS cluster consists of a single NameNode, a master server that manages the file system namespace and regulates access to files by clients. In addition, there are a number of DataNodes, usually one per node in the cluster, which manage storage attached to the nodes that they run on. HDFS exposes a file system namespace and allows user data to be stored in files. Internally, a file is split into one or more blocks and these blocks are stored in a set of DataNodes. The NameNode executes file system namespace operations like opening, closing, and renaming files and directories. It also determines the mapping of blocks to DataNodes. The DataNodes are responsible for serving read and write requests from the file system's clients.

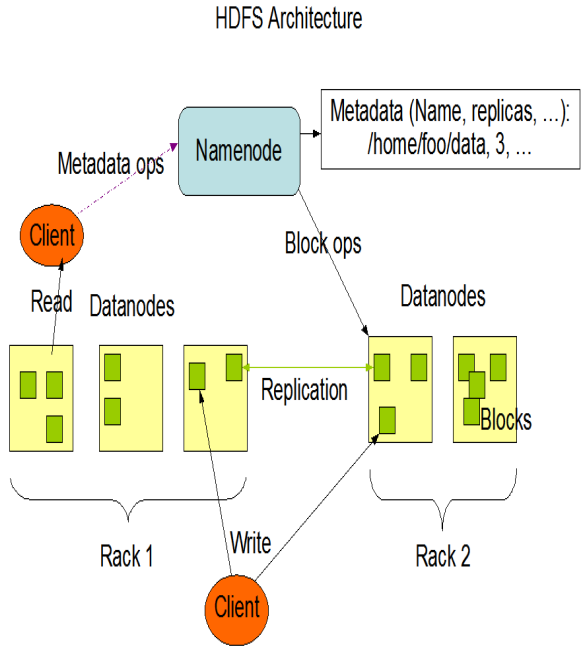


Figure 1.2: HDFS Architecture

The NameNode and DataNode are pieces of software designed to run on commodity machines. These machines typically run a GNU/Linux operating system (OS). HDFS is built using the Java language; any machine that supports Java can run the NameNode or the DataNode software. Usage of the highly portable Java language means that HDFS can be deployed on a wide range of machines. A typical deployment has a dedicated machine that runs only the NameNode software. Each of the other machines in the cluster runs one instance of the DataNode software. The architecture does not preclude running multiple DataNodes on the same machine but in a real deployment that is rarely the case.

The existence of a single NameNode in a cluster greatly simplifies the architecture of the system. The NameNode is the arbitrator and repository for all HDFS metadata. The system is designed in such a way that user data never flows through the NameNode.HDFS is designed to reliably store very large files across machines in a large cluster. It stores each file as a sequence of blocks; all blocks in a file except the last block are the same size. The blocks of a file are replicated for fault tolerance. The block size and replication factor are configurable per file. An application can specify the number of replicas of a file. The replication factor can be specified at file creation time and can be changed later. Files in HDFS are write-once and have strictly one writer at any time.

The system architecture is as shown in the Figure l.3. The main goal is to provide input to the mapper. The input data will be stored in HDFS which is a location aware file system.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 5, May 2015

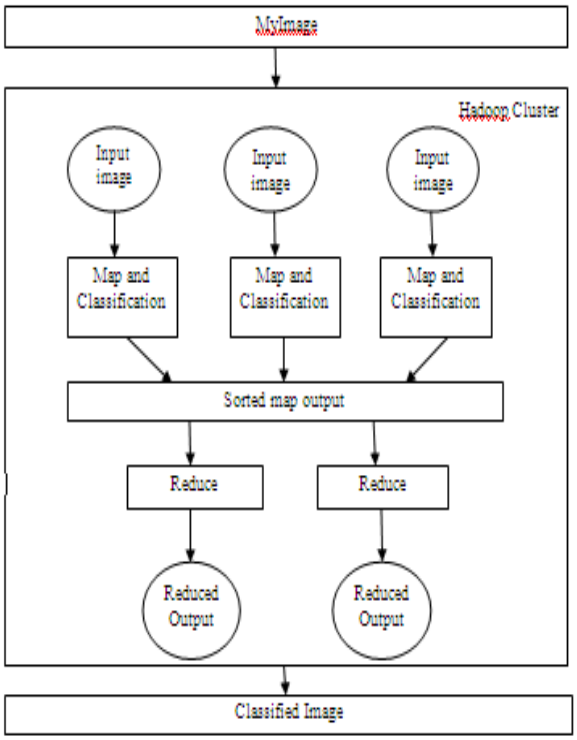


Figure 4.1: Architectural Design of Map/Reduce Image Classification

The map/reduce performs its job of classifying the image and writing the classified image into an output file. Hadoop performance is glaringly noticeable when the data that has to be processed is unorganized and the size of data is huge. The working methodology of parallelepiped supervised classifier is described as follows.

- **Obtain RGB values for the parallelepiped:** The prototypical image is used to identify regions which contain pixels for a particular class (the samples) and those pixels are used to calculate the signatures for the respective classes. This process is repeated for each class, for which one or more sample regions are used. The class definitions file contains, in each of its lines, a definition for a class in the classification task. This definition consists of a unique integer identifier, followed by three values which will be used as the reference colour for that class, followed by the class name. This reference colour will be used during classification to mark the classified pixels belonging to a class with their respective colours.

The samples definition file is used to declare which regions must be used for each class. It contains the coordinates of the regions with respect to the sample image. This file contains five numbers: the first is the unique identifier of the class, followed by the coordinates of the upper-left corner of the rectangle that contains the samples, followed by this rectangle's width and height.

- **Create parallelepiped with min and max values:** The above files are used to create the minimum and the maximum bound values for each class and it is ensured that these values are set appropriately. These are called the signatures, which can be considered as descriptors for the classes, often containing statistical information about the pixels used as samples.
- **Store each pixel in buffer:** The signatures thus obtained in the form of a parallelepiped when plotted on a 3-dimensional graph and each pixel is stored in the buffer.
- **Compare each pixel with class signature values:** The classification is carried out by taking as input the original image to be classified, the signatures and the class descriptor. The RGB values for each pixel in the image are obtained and are checked for the minimum and the maximum bound values in the class descriptor containing that value.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 5, May 2015

- **Classify:** The color of the pixel which is in the bound of minimum and maximum values is stored as a new class and that pixel is painted in the output image and a classified image is obtained.

IV. IMPLEMENATION

The implementation includes the use of various predefined classes of Hadoop and a custom data structure for image processing and java image processing interface to classify the image.The implementation began with identification of all classes that were needed for image processing using Hadoop. The methods that were to be overridden and the methods that had to be written entirely from scratch will be discussed as follows.

4.1 **Mapper**

It maps input key/value pairs to a set of intermediate key/value pairs. Maps are the individual tasks which transform input records into a intermediate records. The transformed intermediate records need not be of the same type as the input records. A given input pair may map to zero or many output pairs. The Hadoop Map-Reduce framework spawns one map task for each InputSplit generated by the InputFormat for the job.
map(Text key, MyImage value, Context context)

a. Reducer

- Shuffle: Output of the mapper is the input to the reducer. In the phase the framework, for each Reducer, fetches the relevant partition of the output of all the mappers.
- Sort: The framework groups Reducer inputs by keys (since different mappers may have output the same key) in this stage. The shuffle and sort phases occur simultaneously i.e. while outputs are being fetched they are merged.
- Reduce: In this phase the reduce(Object, Iterator, OutputCollector, Reporter) method is called for each <key, (list of values)> pair in the grouped inputs. The output of the reduce task is typically written to the FileSystem via OutputCollector.collect(Object, Object).

reduce(Text key, Iterable<MyImage> values, Context context)

4.3 **Job**

The default constructor is being used to configure the job. It is the job submitter's view of the Job. It allows the user to configure the job, submit it, control its execution, and query the state. The set methods only work until the job is submitted. Job class will be used in the mapreduce package which is an updated package compared to the original mapred.

4.4 **MyImage**

This is the data structure that is being used for the values for Mapper and Reducer. Writable interface provided by hadoop is being implemented by this class.

The methods readFields and write are being overridden to provide necessary functionality for the the image processing mapper and reducer. This class provides the basic means for image input/output while using Hadoop.Using the readFields method, the input image file is being read using an instance of the Java class BufferedImage. The input file name comes in from the inputstream.

4.5 **Writable**

Any key or value type in the Hadoop Map-Reduce framework implements this interface.Implementations typically implement a static read(DataInput) method which constructs a new instance, calls readFields(DataInput) and returns the instance. This interface is used for the input output of the image. The data structure for the image that has been used implements this interface.

4.6 **ImageInputFormat**

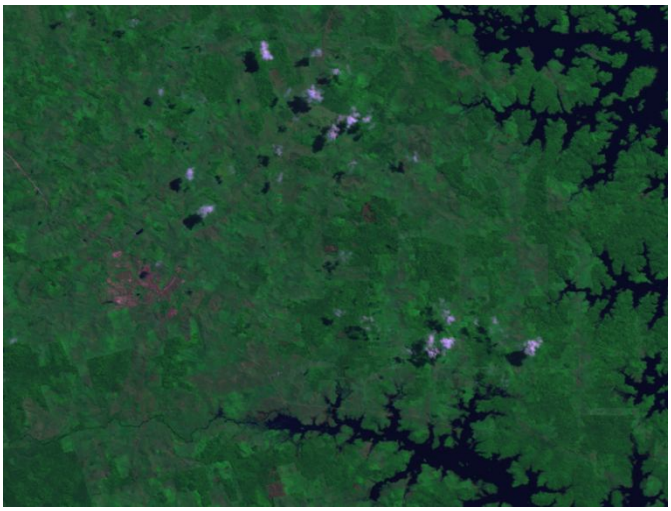
This class has been extended to incorporate how the input will be provided to the mapper.
This includes assigning key, value pairs and providing them to the mapper, setting required properties for input split. Input split can be in the form where parts of the files are given to different mappers or a whole file is given to a mapper.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

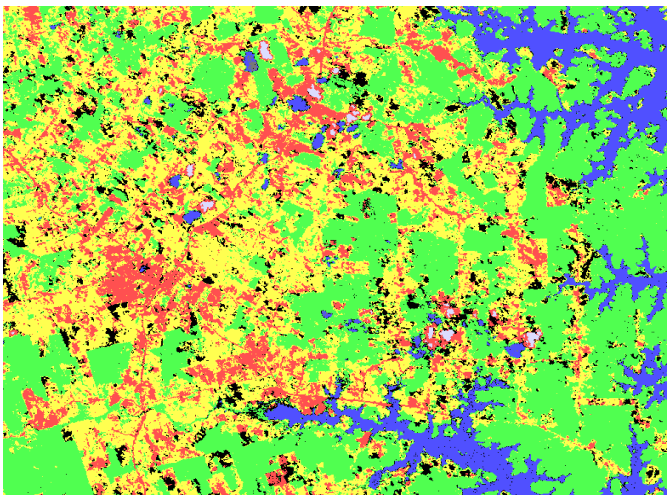
International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 5, May 2015



4.7 **RecordReader**
Main function of this class is to break data into key, value pairs for input to the Mapper.
4.8 **ImageOutputFormat**



This class has been extended to incorporate how the output will be provided to the mapper. FileOutputFormat is the base class for OutputFormat that read from FileSystem. It has methods that are responsible for output from the reducer. This class is also responsible for output that will be sent to the filesystem.

4.9 **RecordWriter**
It writes the output <key, value> pairs to an output file. RecordWriter implementations write the job outputs to the FileSystem. It has only two methods which will be override for image processing.

V. RESULTS AND DISCUSSION

On successful execution, the obtain images in various different classes that can be identified as:

- Water, Clouds, Shadow, Pasture, Urban and Forest.

Figure 1.5: Satellite Image for Classification
Figure1.6: Classified Image

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 5, May 2015

Figure 1.7: Satellite Image – 2

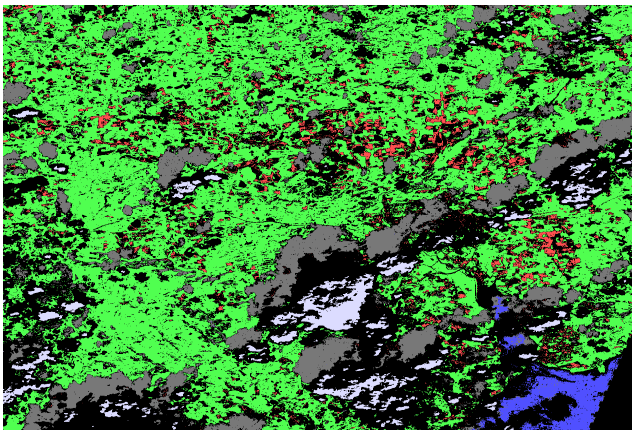


Figure 1.8: Classified Image – 2

5.1 Hadoop Web Interfaces

The web interfaces provide concise information about what's happening in the Hadoop cluster. Hadoop comes with several web interfaces which are by default available at these locations:

- http://localhost:50070/ – web UI of the NameNode daemon
- http://localhost:50030/ – web UI of the JobTracker daemon
- http://localhost:50060/ – web UI of the TaskTracker daemon

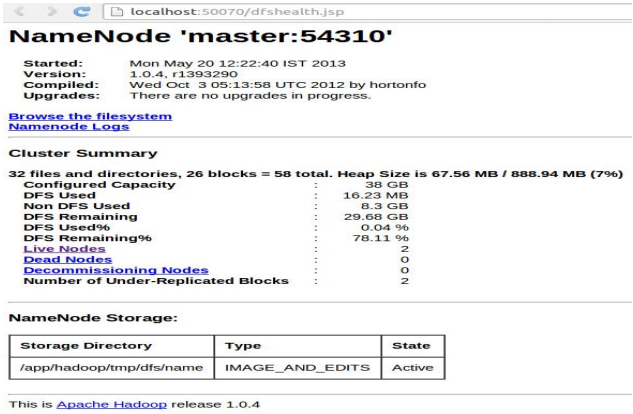
ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

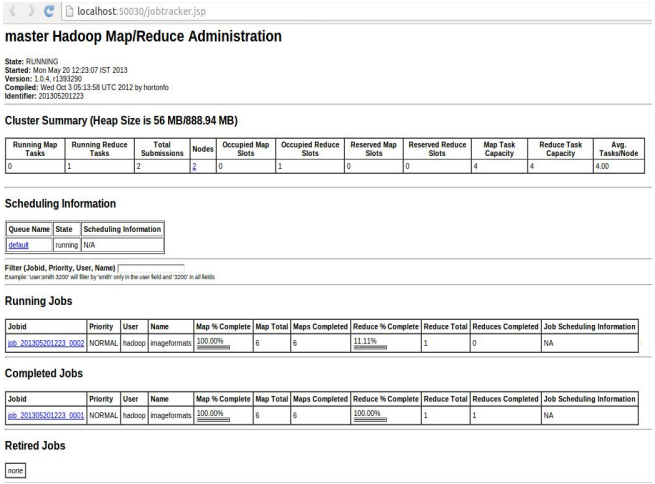
(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 5, May 2015

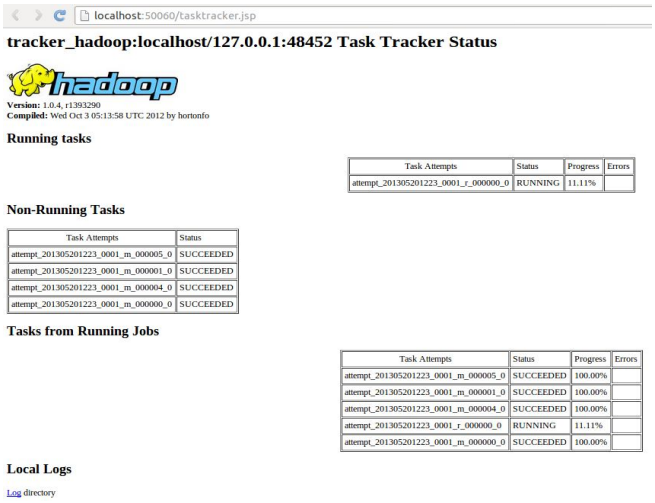
5.1.1 NameNode Web Interface



5.1.2 JobTracker Web Interface



5.1.3 TaskTracker Web Interface



ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 5, May 2015

VI. LIMITATIONS

- Implementing Map/Reduce phases is complex because it requires setting up of each cluster which runs Hadoop.
- Requires a location aware file-system because the data is distributed and will be in multiple copies.
- Lacks schema and other Database features.
- Signatures differ for different types of images
- Obtaining the signatures initially is a task which when completed will not be any challenge for MapReduce image classification.

VII. CONCLUSION AND FUTURE WORK

Digital Image classification using hadoop, is an approach towards gaining maximum efficiency in image processing. Most of the current approaches in image processing specifically in image classification which are process intensive and data intensive are implemented on a single computing systems which does not yield the optimum cost efficiency, time efficiency. As a solution to this issue the distributed image classification is developed and the objectives of implementing on a distributed framework such as hadoop is completed successfully.

As described in the objectives of the paper, image signatures are created, parallelepiped classification is implemented in hadoop and an entire classified image is obtained in its entireity from the mapreduce framework.

VIII. FUTURE WORK

Automated systems can be developed where no intervention of a human or other system is required except to start the system. Processing millions of images is a challenging task and automating these tasks will lead to faster and accurate processing. The burden on humans will also reduce. The only task for humans is to write programs that will take care of the automated processing.

Arial images obtained from the satellites are classified as required. This work can be modified according to the needs to implement it for other image processing tasks. This can vary simple image classification, image co addition, image analysis etc.,

REFERENCES

[1] MapReduce Tutorial-Apache Hadoop-hadoop.apache org
[2] Hadoop -The definitive Guide(Tom White)
[3] Running Hadoop on Ubuntu Linux (Single-Node Cluster)
[4] Running Hadoop on Ubuntu Linux (Multi-Node Cluster)
[5] Java Docs -docs.oracle.com/javase
[6] Hadoop Docs -hadoop.apache.org/docs/
[7] Work at University of Washington, Astronomy Survey Science Group on Astronomical Image Processing with Hadoop
[8] Mohamed H Almeer onHadoopMapRecude for Remote Sensing Image Analysis, April 2012
[9] Remote Sensing Note(Professor ShunjiMurai, President, Japan Association on Remote Sensing)

Websites

- http://www.infoq.com/articles/HadoopInputFormat
- https://groups.google.com/forum/?fromgroups=#!topic/spark-users/InBfB_lfP68
- http://stackoverflow.com/questions/3510201/hadoop-inputfile-as-a-bufferedimage
- http://mail-archives.apache.org/mod_mbox/hadoop-common-user/200805.mbox/%3CC449D1DF.3EE83%25tdunning@veoh.com%3E
- http://wiki.apache.org/hadoop/HadoopMapReduce?highlight=(inputf
- http://jerryjcw.blogspot.in/2009/10/on-hbase-table-join-dabblers.html
- http://stackoverflow.com/questions/3044050/image-processing-with-hadoop
- http://ro.uow.edu.au/cgi/viewcontent.cgi?article=1566&context=infopapers
- http://stackoverflow.com/questions/15470778/hadoop-and-different-format-of-inputs-like-image-audio-video
- http://www.slideshare.net/sh1mmer/upgrading-to-the-new-map-reduce-api

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 5, May 2015

- http://stackoverflow.com/questions/2115292/run-hadoop-job-without-using-jobconf
- http://www.slideshare.net/thecapacity/hadoop-overview-presentation
- http://geol.hu/data/online_help/ApplyingParallelepipedClassification.html
- http://pastebin.com/Y6gZZkmi
- http://mapreduce-specifics.wikispaces.asu.edu/Applications+and+Limitations+of+MapReduce