Video and Image Processing Library using Computer Vision Techniques for Hadoop HDFS and MapReduce

Anvi Patel [1] , Ch Santoshi Madhuri [1] , Desu Lakshmi Kavya Mounika [1],

E K L Sushma [1] , K S K Manjusha [1], Noothalapati Tejasree [1] Raghavendra Kune [2]

[1]Gandhi Institute of Technology and Management (GITAM) University,Visakhapatnam,India.

[2] Advanced Data Processing Research Institute,Department of Space,Hyderabad,India.

Abstract - Hadoop has become defacto framework for handling large scale data and processing in Big Data technology. In Big Data on HDFS and MapReduce, there is a considerable amount of work has been done towards processing several types of textual data e.g weblogs, however, a little amount of work has been conducted on binary data like images and vidoes. Hence, there is a need to address the processing of images and videos on HDFS and MapReduce platform. In this paper, we present the processing of such binary data on HDFS using MapReduce with the use of standard computer vision techniques. We present the tools and discuss their implementation specific to the image/video domain in detail, followed by sample case studies using open source libray for computer vision such as opencv function. Here, image processing techniques are being implemented using standard Java libraries whereas video processing is being implemented using two types of Java libraries namely JavaCV and OpenIMAJ.

Keywords : Big Data , HDFS , MapReduce, Computer Vision Tools , OpenCV, OpenIMAJ.

1. Introduction

Big data [1] is a term that describes the large volume of data – both structured and unstructured– that evolves a business on a day-to-day basis. Big data can be analyzed for insights that lead to better decisions and strategic business moves. While the term "big data" is relatively new, the act of gathering and storing large amounts of information for eventual analysis is ages old. The concept gained momentum in the early 2000s when industry analysts articulated the now-mainstream definition of big data as the three Vs :- Volume, Velocity and Variety. Big Data is emerging as new field for distributed processing for both scientific and business needs.

1.1 Hadoop [2] is an open source, Java-based programming framework that supports the processing and storage of extremely large data sets in a distributed computing environment. It is part of the Apache project sponsored by the Apache Software Foundation. It can be used for processing really big data, storing a diverse set of data and for parallel data processing. Hadoop consists of three core components: a distributed file system, a parallel programming framework, and a resource/job management system.

1.2 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. It provides high throughput access to application data and is suitable for applications that have large data sets.We also describe our approach of

distributing tasks using MapReduce. MapReduce, As depicted in Figure 1 is a processing technique and a program model for distributed computing based on java. The MapReduce algorithm contains two important tasks, namely Map and Reduce.

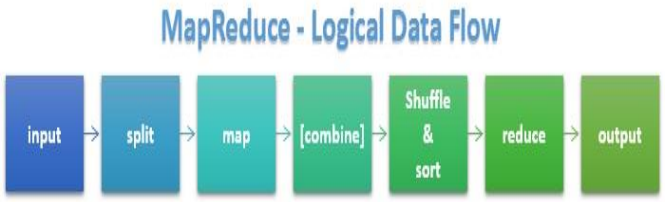


Figure 1. MapReduce Logical Data Flow

1.3 Image Processing [**3]** is a method to convert an image into digital form and perform some operations on it, in order to get an enhanced image or to extract some useful information from it. The Hadoop Image Processing Framework is largely a software engineering platform. Hadoop's complexity while providingusers with the ability to usethesystemforlargescale image processing without b ecoming crack Hadoop engineers.

1.4 Video Processing is a particular case of signal processing, which often employs video filters and where the input and output signals are video files or video streams.Video processing uses hardware, software, and combinations of the two for editing the images and sound recorded in video files. Extensive algorithms in the processing software and the peripheral equipment allow the user to perform editing functions using various filters. The desired effects can be produced by editing frame by frame or in larger batches.

1.4.1 OpenIMAJ[4] is set of libraries and tools for multimedia content analysis and content generation in java . OpenIMAJ is very broad and contains everything from state-of-the-art computer vision (e.g. SIFT descriptors, salient region detection, face detection, etc.) and advanced data clustering, through to software that performs analysis on the content, layout and structure of web pages.Development of OpenIMAJ is hosted by Electronics and Computer Science at the University of Southampton. Current development of OpenIMAJ is graciously funded by The European Union Seventh Framework under the ARCOMEM project.

1.4.2 Java OpenCV[5] (Open Source Computer Vision) is a library of programming functions mainly aimed at real-time computer vision. Originally developed by Intel, it was later supported by Willow Garage and is now maintained by Itseez. The library is cross-platform and free for use under the open-source BSD license.JavaCV uses wrappers from the JavaCPP Presets of commonly used libraries by researchers in the field of computer vision(OpenCV, FFmpeg, libdc1394, PGR FlyCapture, OpenKinect, librealsense, CLP S3 Eye Driver, videoInput, ARToolKitPlus,and flandmark), and provides utility classes to make their functionality easier to use on the Java platform, including Android.

2. Related Work

Hadoop which is under the license of Apache is accessible to researchers as an open source framework. For using MapReduce models, in addition to Hadoop, we can also use Twister [7] and Phoenix [6] or other MapReduce style frameworks[8]. These two frameworks, that are both open source implementations of MapReduce model, are designed for specific purposes. Twister for iterative calculations and Phoenix for multi-processor systems with shared memory can be suitable alternatives for Hadoop.

Li et al.[9] proposed a system for classifying ground imageries based on natural features that uses feature vectors approach and structured SVM in order to recognize natural features in any images. They used 6.5 million photos of Flickr and process them with the help of MapReduce programming model. Yan et al. [10] proposed the algorithm of extensibility based on MapReduce model which has been tested on 260,000 images. S. M. Banaei[11], H. K. Moghaddam 243 For Content Based Image Retrieval (CBIR), Shi et al. [12]proposed a model based on MapReduce method which has been tested on 400,000 images. Zhao, Li, and Zhou from Peking University conducted a research on the use of MapReduce model for satellite imagery documentation and management and spatial data processing[13]. They also proposed a system based on cloud computing. Yang and the associated team [14] conducted a research concerning the use of Hadoop in the field of medical images. They designed a system named MIFAS for fast and efficient access to medical images using Hadoop and Cloud Computing. For identification through cornea, Shelly and Raghava designed and implemented a system using Hadoop and Cloud

Computing[15]. They tested it up to 13.2 GB input file and Hadoop has shown about 80% efficiency in high volume of data compared to conventional image processing methods. Kucakulak and Temizel proposed a Hadoop-based system for pattern recognition and image processing of intercontinental missiles[16]. Almeer designed and implemented a system for remote sensing image processing systems with the help of Hadoop and cloud computing systems with 112 compute nodes [16]. HIPI is Hadoop Image bundle where a smaller number of images are bundled as HIB format and are processed using MapReduce implementation so as to overcome the probem of occupying smaller blocks and processing them using MapReduce implementations. 0. eXtended Hadoop And MapReduce for Image Processing (XHAMI) is another implementation of processing large scale images with the objective of overalapping of adjacent blocks for effectively processing the images using MapReduce on HDFS [17]

In this paper, we describe several mechanisms of Computer vision techinques integrating them with the Hadoop- Map Reduce technology and implemented image and video processing techniques with the help of Java, JavaCV and OpenIMAJ. The complete implementation with Computer Vision techniques is still growing and it helps to provide functionality very easily.

3. Image and Video Processing using Computer Vision Techniques using Hadoop MapReduce

The below sections depict the Image and Video processing implementations that are implemented using Hadoop Map Reduce techniques. The system configuration used for the development and experimental studies are given below.

- Operating System – Ubuntu 14.04 , 32 bit
- RAM : 8 GB
- Storage Disk – 500 GB.
- Apache Hadoop Version – 2.0

3.1 IMAGE PROCESSING LIBRARY

In the architecture we have created a map-reduce based model for image processing. This model consists of mapper and reducer which processes the image to and from HDFS. It is useful for big data analytics. Its objective is to deal with tendentious amount of data reliably, and compatibly. Our Hadoop database includes key ideas as mapper and reducer. Mapper works as change, filtering and conversion which means it performs the accompanying capacities Record peruse, Partioner, mapper and Combiner and Reducer incorporates the capacities as shuffling, sorting, reducing and yield design.

The algorithms implemented for image processing using MapReduce on HDFS are i) Rotation ii) Edge detection iii) Histogram as described below.

i. ROTATION

One of the technique we performed is rotation of images using MapReduce Technique in Hadoop, which is described in the class diagram in Figure 2

The rotator class takes input as an image in JPEG Format which shows rotated image in panel .The image is rescaled in rescale() function. It take image from the local and sends the image into hdfs Writer The image is moved to hdfs Writer and there is a collaboration between Hdfs write and rotator program .

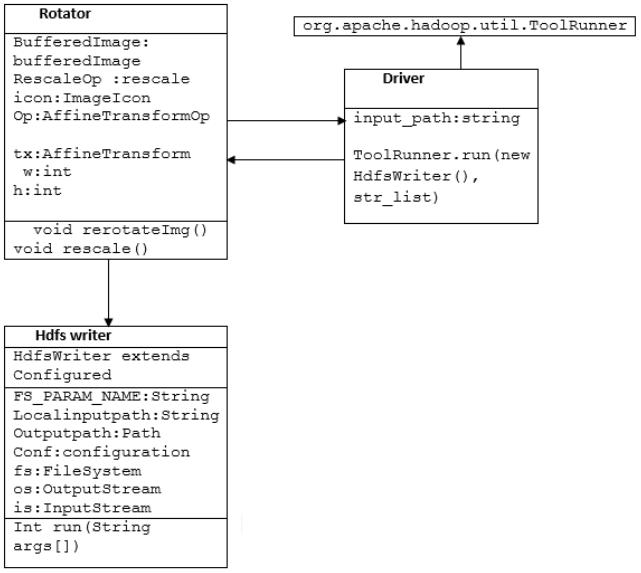


Figure 2 Rotation Class Diagram

ii. EDGE DETECTION

Edge detection is another implementation described in Figure 3 where,The edgeDetect sets the configuration and generic optionsparser and set themapperClass and setMapoutputkeyclass and mapoutputvalueclass ,setReducerClass.For edge detection sometimes not require a reducer ,as the resultant output of mapper function is directly written to the disk. ex: sobel edge detection .This mapper class forms a key value pair it adjust its desired parameter and apply it to a image .it sets the height ,width,pic size etc in cannyaedgeDetector().It return source image in getsourceImage().It set edges and return the low threshold valaues in getlowThreshold().the default size of high threshold is 7.5.The width and height of image is processed in void process().The reducer class create a new img path and wite the JPG format image on to the disk ImageIo.write().

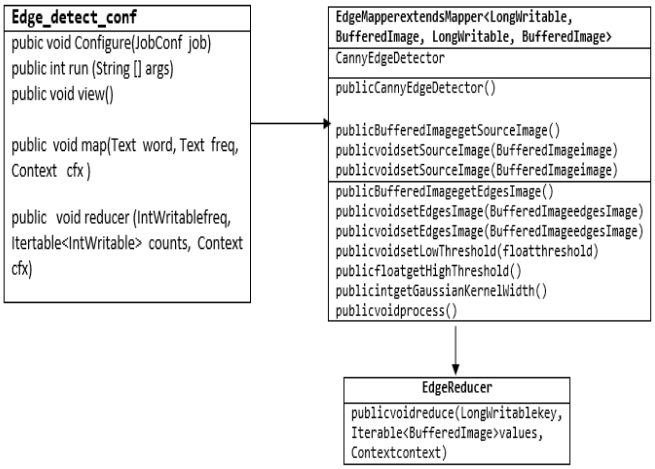


Figure 3 Edge Detection Class Diagram

iii. HISTOGRAM

Histogram is another implementation shown in Figure 4 where,This Histogram class is used to configure the job and take image as input and this built in input format read a "key/val" pair from each line and emitsthem and parse them into correct data types in the mapper given below.The mapper functions take Histogram operation from above block computes frequency count of the pixel in the image. The histogram is computed as follows, first, the block and length of the block is read, and each block is mapped to one map function.The Reducer Class will give the pixels value of an input image and display the histogram containing the pixel values of an input image.

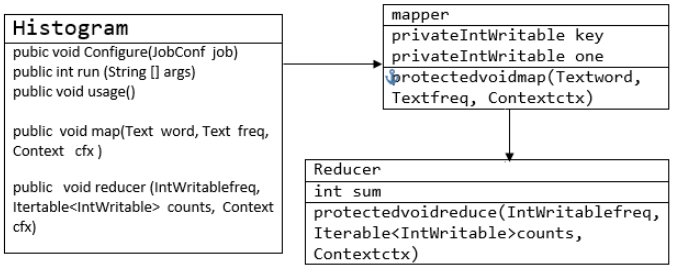


Figure 4 Histogram Class Diagram

3.2 VIDEO PROCESSING LIBRARY

In Video Processing, we implement Pattern matching also known as Template Matching, is a technique in digital image processing for finding small parts of an image which match a template image. It can be used in manufacturing as a part of quality control, a way to navigate a mobile robot, or as a way to detect edges in images. In, Pattern Matching a template(a small part) is taken as one input and a normal frames as another input, and the similarity between the two (if any) is matched through the pixel identification and the matched region is displayed quoted with a rectangle. But, the aim of this project is to handle multiple frames simultaneously and to display the count of the number of frames matched, which can be efficiently accomplished using the Map-Reduce algorithm.

Video processing has been implemented by integraging the OpenIMAJ and Java OpenCV libraries with HDFS and implementing the MapReduce functions as discussed below.

A. VIDEO PROCESSING USING OPENIMAJ

OpenIMAJ is a developing open source java package which is also growing towards correlation with Hadoop and is providing methods in java which focuses on Hadoop techniques of processing data. Thus, we applied the methods of OpenIMAJ in handling the video data. The OpenIMAJ implementation is linked up with the HDFSWriter program, which stores the frames directly into the Hadoop Distributed File System, where we apply the Map reduce for Pattern Matching to obtain the total count of the number of matches in Figure 6

The below design Figure 5 tells that we implement a single function called processVideo(), which takes care of the entire processing of video which mainly runs with the help of OpenIMAJ(Intelligent Multimedia Analysis in Java) and the Hadoop techniques (MapReduce). We will now look into the step by step process of the Implementation

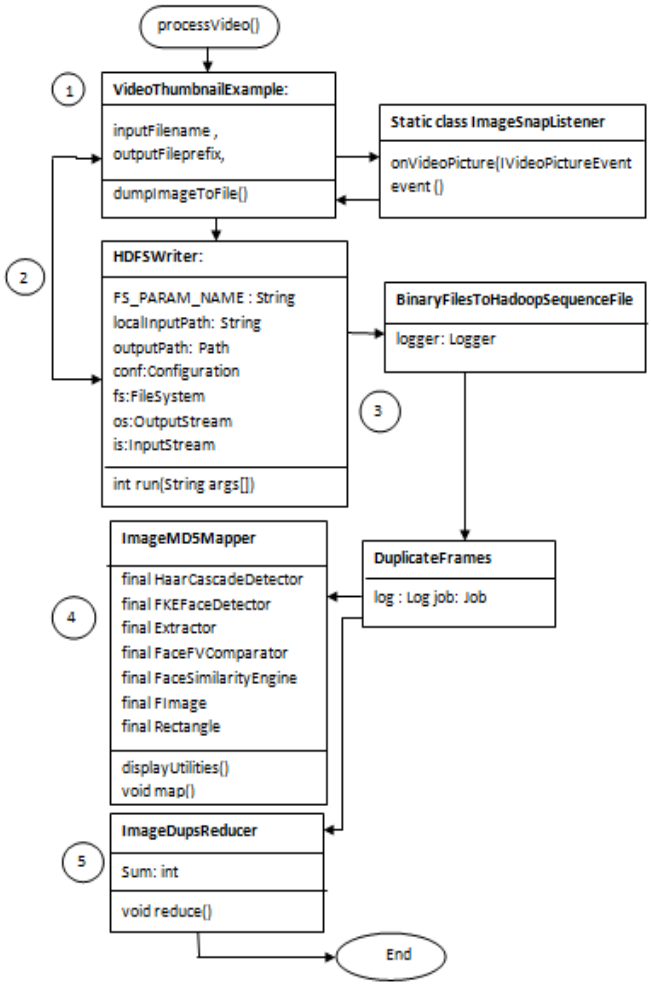


Figure 5 Pattern Matching (OpenIMAJ) Class Diagram

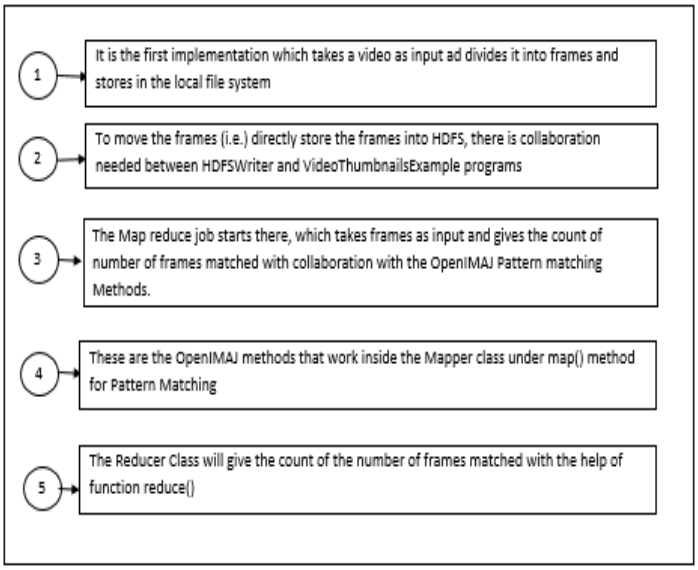


Figure 6 Description of Class Diagram

a) PATTERN MATCHING WITH OPENIMAJ LIBRARY

The Map Reduce functionality is as follows:

Basically, the implementation of the Pattern matching with the openIMAJ libraries is included in the **Mapper Class** and the **Reducer Class** plays the role of counting the number of frames matched and returns the sum of the count to the Mapper class again which is displayed.

ImageMD5Mapper Class

This Class extends **Mapper <LongWritable, Text, Text, IntWritable>** which says that the input key-value pair is LongWritable and Text whereas the output key value pair is Text and IntWritable. After the inputs for the template matching are taken, we provide the methods that help in the Pattern matching using OpenIMAJ.

The later implementation includes detection of matches shown in Table 1

```
for (final Entry<String, Double> matches :
e.getValue().entrySet())  {

if (matches.getValue() < bestScore) {
bestScore = matches.getValue();
best = matches.getKey();
final Rectangle r=
engine.getBoundingBoxes().get(best);
r.translate(image1.width, 0);
img.drawShape(r, 1F);
}
```

Table 1 : Mapper Class Implementation

The last part of Mapper Class is to display the number of matches as:

context.write(value.one);

context.write(new Text ("Number of matches = "), one);

ImageDupsReducer Class

This class extends the
Reducer<Text,IntWritable,Text,IntWritable>

where the count and sum of the matched patterns is found and returned to the mapper class shown in Table 2

```
int sum = 0
for (IntWritable count : counts) {
sum += count.get();
}
```

Table 2 Reducer Class Implementation

B. VIDEO PROCESSING USING OPENCV

OpenCV (Open Source Computer Vision) is a library of programming functions mainly aimed at real-time computer vision.Originally developed by Intel, it was later supported by Willow Garage and is now maintained by Itseez.The library is cross-platform and free for use under the open-source BSD license.We used this package to perform Pattern matching in integration with Hadoop Map Reduce shown in

Figure 7

Firstly we have the class Read which is used for extracting the frames from the MP4 video. Here various variables are used which are frameGrabber, paintConverter,BufferedImage, path ,frame which are all used in the execution of the program.Next we use the HdfsWriter which is uded for moving the extracted frames from the video into the Hadoop Distributed File System. Various variables along with the method run() are used to execute the program.

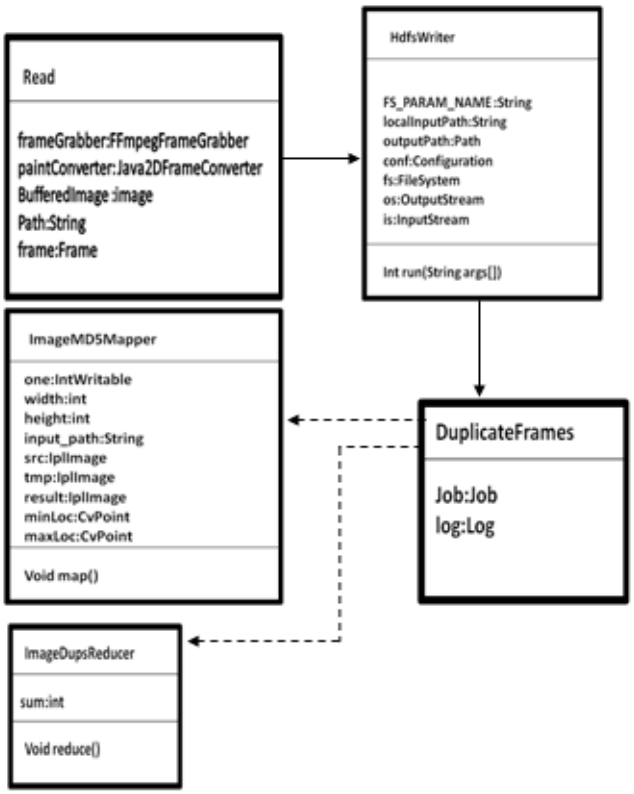


Figure 7 Pattern Mtching (OpenCV) Class Diagram

b) PATTERN MATCHING WITH OPENCV LIBRARY

DuplicateFrames is basically a MapReduce program which has two different classes for its task to be performed.

The two classes ImageMd5Mapper and ImageDupsReducer perform the jobs of mapper and

reducer respectively. Pattern matching is done as follows:

ImageMd5Mapper Class

The main function where the pattern is detected is:

cvMatchTemplate(src,tmp,result, CV_TM_CCORR_NORMED)

The matched part of the each frame is highlighted with a rectangle across its border using this code in Table 3

```
CvRect rect = new CvRect();
rect.x(maxLoc.x());
rect.y(maxLoc.y());
rect.width(tmp.width() + width);
rect.height(tmp.width() + height);
cvSetImageROI(src, rect);
IplImage imageNew =
cvCreateImage(cvGetSize(src), src.depth(),
src.nChannels());
```

Table 3 Mapper Class Implementation

The last class of mapper class to display matches are

```
context.write(value, one);
context.write(new Text("Matched Images"),
one);
```

ImageDupsReducer Class

This extends the **Reducer<Text,IntWritable,Text,IntWritable>**

where the count and sum of the matched patterns is found and returned to the mapper class shown in

Table **4**

```
int sum = 0;
for (IntWritable count : counts) {
sum += count.get();
}
```

Table 4 Reducer Class Implementation

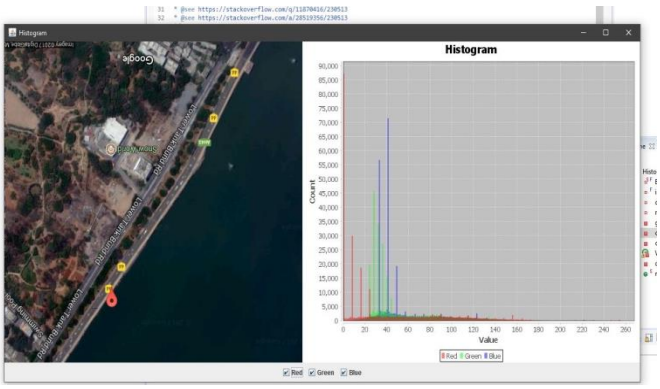
4. Results

a) IMAGE PROCESSING

ROTATION

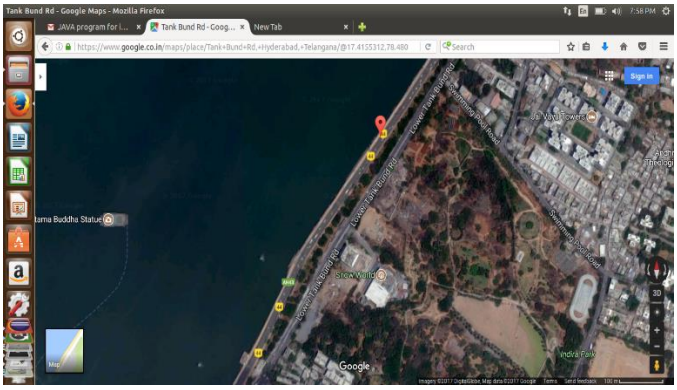
EDGE DETECTION

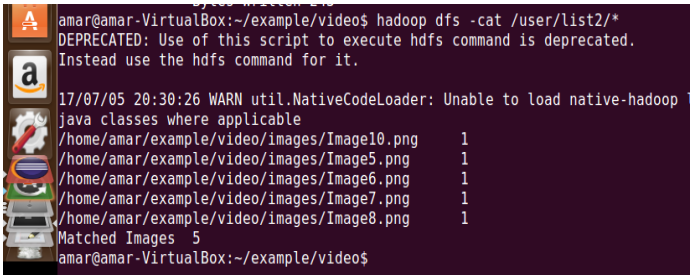
HISTOGRAM



b) VIDEO PROCESSING- OpenIMAJ & OpenCV:

PATTERN MATCHING





5. Conclusions and Future Work

Image and Video Processing is a promising area for distributed platforms as it demands lot of computing and storage resources. Hadoop and MapReduce is an evolving platform with enormous advantages for addressing large scale data and computing applications. In this, we use both the concepts of HDFS and MapReduce for addressing the binary data handling like images and videos. The main advantages of Map Reduce are Scalability, Flexible, Fast, Security and Authentication, Parallel Processing (etc).When it comes down the processing of large data sets, Hadoop's MapReduce programming allows for the processing of such large volumes of data in a completely safe and cost-effective manner. Here, we addressed the processing of images/videos using computer vision tools like Java OpenCV and OpenIMAJ. The several experiments with their implementations in Hadoop and MapReduce are done for the functions like edge detection, histogram and rotation. Similarly, MapReduce for videos are implemented for feature/pattern matching techniques using JavaCV and OpenIMAJ techniques. Implementation is done on a single node, however, it is proposed to extend the work and test the results on the multi node cluster. Also, it is proposed to implement some more functionalities of OpenCV libraries on MapReduce.

Acknowledgements

We would like to thank Sri. Desu Mallikharjuna Rao, Scientist 'SG', ADRIN, who helped to get the internship at ADRIN, and provided extensive cooperation throughout our intership and our stay at Hyderabad. We also would like to thank Sri. Amar Sharma, who helped us in understanding the basic concepts of Hadoop and MapReduce. We would like to thank Sri. Satya Rao Chowdary, who had made our stay comfortable at Hyderabad duirng the internship.

REFERENCES

[1] Big Data – Wikipedia https://en.wikipedia.org/wiki/Big_data

[2] XHAMI -Raghavendra Kune1, ,Pramod Kumar Konugurthi1 , Arun Agarwal2 , Raghavendra Rao Chillarige2 and Rajkumar Buyya3 (2015)The anatomy of big data computing .

[3] Image Processing: Almeer, M.H. (2012) Cloud Hadoop MapReduce For Remote Sensing Image Analysis. Journal of Emerging Trends in Computing and Information Sciences, 3, 637-644. S. M. Banaei, H. K. Moghaddam 245

[4] OpenIMAJ: Document http://openimaj.org/tutorial/index.html

[5] OpenCV : Reference https://en.wikipedia.org/wiki/OpenCV

[6] The Phoenix System for MapReduce Programming. http://mapreduce.stanford.edu/

[7] Twister. http://www.iterativemapreduce.org/

[8] Qura Question and Answer Website. http://www.quora.com/What-are-some-promising-open-source-alternatives-to-Hadoop-MapReduce-for-map-reduce

[9] Li, Y., Crandall, D.J. and Huttenlocher, D.P. (2009) Landmark Classification in Large-Scale Image Collections. ICCV, 1957-1964.

[10] L. Kenned., Slaney, M., K.Weinberger, Reliable Tags Using Image Similarity: Mining Specificity and Expertise from Large-Scale Multimedia Databases. In: proc. 1st Workshop on Web-Scale Multimedia Corpus, Beijing, 23-23 October 2009, pp.17-24.

[11] R.Yan, Fleury, M.-O Merler, M., Natsev, A. and Smith, J.R. (2009) Large-Scale Multimedia Semantic Concept Modeling Using Robust Subspace Bagging and MapReduce. Proceedings of the 1st ACM Workshop on Large-Scale Multimedia Retrieval and Mining, Beijing, 23-23 October 2009, pp.35-42.

[12] Shi, L.L., Wu, B., Wang, B. and Yan, X.G. (2011) Map/Reduce in CBIR Application. 2011 International Conference on Computer Science and Network Technology (ICCSNT), Vol. 4, Harbin, 24-26 December 2011, 2465-2468.

[13] Zhao, J.Y., Li, Q. and Zhou, H.W. (2011) A Cloud-Based System for Spatial Analysis Service. 2011 International Conference on Remote Sensing, Environment and Transportation Engineering (RSETE), Nanjing, 24-26 June 2011, pp.1-4. http://dx.doi.org/10.1109/RSETE.2011.5964031.

[14] Yang, C.-T. and Chen, L.-T., Chou, W.-L. and Wang, K.-C. (2010) Implementation of a Medical Image File Accessing System on Cloud Computing. 2010 IEEE 13th International Conference on Computational Science and Engineering (CSE), Hong Kong, 11-13 December 2010, 321-326. http://dx.doi.org/10.1109/CSE.2010.48

[15] Shelly and Raghava, N.S. (2011) Iris Recognition on Hadoop: A Biometrics System Implementation on Cloud Computing. 2011 IEEE International Conference on Cloud Computing and Intelligence Systems (CCIS), Beijing, 15-17 September 2011, pp.482-485. http://dx.doi.org/10.1109/CCIS.2011.6045114

[16] Kocakulak, H. and Temizel, T.T. (2011) A Hadoop Solution for Ballistic Image Analysis and Recognition. 2011 International Conference on High Performance Computing and Simulation (HPCS), Istanbul, 4-8 July 2011, 836-842. http://dx.doi.org/10.1109/HPCSim.2011.5999917

[17] XHAMI : Raghavendra Kune1 , Pramod Kumar Konugurthi1 , Arun Agarwal2 ,Raghavendra Rao Chillarige2 , and Rajkumar Buyya3(2011) XHAMI - Extended HDFS and MapReduce Interface for Image Processing Applications.

[18] HIPI Bundle: http://hipi.cs.virginia.edu/