Object Recognition In HADOOP Using HIPI

Ankit Kumar Agrawal, Prof. Sivagami M.

Abstract: The amount of images and videos being shared by the user is exponentially increasing but applications that perform video analytics is severely lacking or work on limited set of data. It is also challenging to perform analytics with less time complexity. Object recognition is the primary step in video analytics. We implement a robust method to extract objects from the data which is in unstructured format and cannot be processed directly by relational databases. In this study, we present our report with results after performance evaluation and compare them with results of MATLAB.

Keywords: Foreground Segmentation, Backgr-ound Subtraction, Object Extraction

I. INTRODUCTION

Most image processing and video analytics algorithms are applicable to large data sets. It is often desirable to run these algorithms on large data sets which can be efficiently performed on distributed systems. Big data in form of large data sets cannot be efficiently processed on sequential applications and need distributed platform for optimality. This applies to real-time data as well, which consists of images stored in sequential manner called frames and these frames also consist of data about associated time and frame information for each frame. Hadoop Map Reduce provides one such platform where these tasks can be divided among its nodes. Working on image processing in Hadoop Map Reduce can prove to be quite a task. HIPI is a tool which when integrated with Hadoop can reduce the complex nature of the processing to the programmer. If the data set consists of video files, it can be converted into set of frames using another tool called Hadoop Streaming. Later, these framesets can be passed as input to the HIPI for further processing. The proposed system can be slightly modified to work on object detection algorithm which can be used in surveillance video applications. Among the approaches for foreground object extraction, one of the most common techniques is the background subtraction technique. This technique has been implemented from several years in various vision systems like video surveillances, tele-conferencing, etc. We will discuss further about this technique in the later section. This paper is categorized as follows. Section II discusses about related works that have been done in field of object recognition. Section III explains the detailed procedure followed towards the achievement of the aim of this project. Section IV demonstrates a performance comparison between the system in HIPI and MATLAB.

II. RELATED WORKS

As we know, object recognition is an elementary process towards the achievement of video analytics. Recently, there have been quite a few researches going on in this field. Illumination-robust foreground detection algorithm proposed by Li et al. [1] proves to be robust against the noise as well as illumination changes. This algorithm updates the Gaussian mixture model using online expectation-maximization technique. Also, it uses a spherical K-means clustering method for updation in case of unstable illumination. It proves to be more effective for perturbations from illumination changes. Another algorithm which was proposed by Sun et al. [2] uses foreground feature point probability calculated via robust SIFT trajectories to detect foreground object. This algorithm can detect the foreground object using their proposed consensus foreground object template (CFOT). Codebook model proposed by Kim et al. [3] uses a real-time segmentation in which each pixel is quantized into compressed form of background model commonly known as codebooks. This method achieves robust detection among moving background or illumination variations. However, this algorithm doesn't work well with compressed videos. Algorithm proposed by Kim et al. [4] follows the same procedure to some extent. It uses Gaussian family model and multiple thresholds similar to our system for robust foreground extraction. It models the background as generalized Gaussian Family of Distributions and updates them by selective running average and static pixel observation while our system uses on-line approximation technique. This also needs to be noted that all the above mentioned systems were implemented in Matlab while our system will implement object recognition algorithm in HIPI tool on Hadoop framework.

III. PROPOSED WORK

In this paper, we propose a test-bed platform for the execution of a robust algorithm for foreground extraction for color-video sequences in Hadoop using [4] with few modifications in the algorithm. Here we are using on-line approximation technique instead of Gaussian distribution and instead of using of using morphological operations for eliminating smaller objects, we are using thresholding technique. In order to do so, each pixel is modeled as a Generalized Gaussian Family (GGF) distribution. We divide the processing into parts for making it optimal for the mappers and reducers to obtain the results. The process begins by passing each frame from the video to the mapper having ImageHeader and FloatImage as the key-value input pair. Then, various steps are taken to acquire the objects from the inputs. The mapper gives the output in form of IntWritable, FloatWritable as key-value pair which is passed upon to the reducer. The reducer is also responsible for conversion of the outputs back to image form and its

- *Ankit Kumar Agrawal, Master Of Computer Application, SCSE, VIT University, Chennai Campus, Chennai, India* ankit.grwl.123@gmail.com
- *Prof. Sivagami M. SCSE, VIT University, Chennai Campus, Chennai, India,* msivagami@vit.ac.in

storage on the HDFS. It can be seen by copying the images to local drives. The process is explained below in details:

A. Background Modeling

Gaussian distribution N(μ, σ) was earlier used to model the variance of a pixel. This modeling was carried out in static scene over time. However, recent advancement in the technologies has led to better quality cameras and surveillance devices. These devices provide steady images with reduced noise. Also, they prove to be better in case of indoor scenes where pixels' variations are smaller in compared to outdoor scenes because of lesser light dispersion and changes in illumination as well as rarer motions occurrence. The background is modeled from the initial frame using color components. The background modeling is carried out in two modes, first being luminance component while the other being the color component. Using luminance component becomes necessary because color components are affected by the presence of noise and lighting conditions changes. However, color component act as second background model because luminance component changes drastically in the presence of objects' shadows and reflections from lighting. Thus, luminance component is used for initial object segmentation while the color component helps to remove the false segmentations obtained due to shadows and reflections. Since we are working on HIPI tool in Hadoop, we extract the RGB component of each pixel of the input image/frame. Secondly, we use this RGB component to obtain the HSI model of the same. Then, we extract the hue component from the HSI model.

$$I = \max(R, G, B)$$

$$S = \begin{cases} 0 & \text{if } I = \min(R, G, B) \\ (I - \min(R, G, B))/I & \text{otherwise} \end{cases}$$

$$H = \begin{cases} 0 & \text{if } I = \min(R, G, B) \\ (G-B)*60/S & \text{if } I = R \\ 180 + (B-R)*60/S & \text{if } I = G \\ 240 + (R-G)*60/S & \text{if } I = B \end{cases}$$

If H<0 then H=H+360

B. Background Updation

It is not necessary that the background model will always be static. In other words, it is a common practice to accommodate changes in the background. The change can occur primarily due to two reasons: lighting conditions leading to gradual changes, and motion of camera or background objects resulting in sudden changes. To cope with these changes, we used on-line approximation technique [5] to update the background model. This technique will handle variations in lighting, motions, and multiple moving objects. In this technique, value of each pixel is modeled as a mixture of Gaussians and then Gaussians corresponding to background colors are determined based on their persistence and variance. Pixel values are matched with the background distributions and those that fit are considered as background while those that do not fit are considered as foreground. For the video processing, the following analysis uses the function *V(x,y,t)* which refers to a frame in the video sequence where t is the time dimension, x and y are the pixel location variables. Now, we apply the mean filter to calculate the background image/frame at an instance t,

$$B(x,y) = 1/N \left(\sum_{i=1}^{N} V(x,y,t-i) \right)$$

where N is the number of preceding images taken for averaging corresponding pixels in the given images and would depend on the number of images per second in the video and the amount of movement in the video. This background should take in considerations lighting changes, motion cluttering and long term scene changes. After calculating the background *B(x,y)* we can then subtract it from the image *V(x,y,t)* at time *t*=t. Updation of the background require a consistent frame as background acquired from mean of N previous frames. Background frame initially is the first frame. But afterward, it is calculated as mean of corresponding pixels in N consecutive frames, thus resulting in a frame with updated background.

C. Foreground Extraction

In HIPI, we calculate the mean and the variance of the current frame as well as the background frame. We also obtain the mean and variance of the hue component of the current and background frames. Once the background modeling and updation is obtained, extraction of foreground objects' silhouette is performed from the video sequences. Thus the foreground is

$$|V(x,y,t) - B(x,y)| > Th$$

where *Th* is threshold. The first step is performed by subtracting the luminance components of the current frame from that of the background model. This results in initial region classification. Now, we classify the initial object regions into four categories namely Reliable Background (a), Suspicious Background (b), Suspicious Foreground (c), Reliable Foreground (d). This classification is performed using multiple thresholds based on their reliability as shown in Eq. (1). We use multiple thresholds instead of a single threshold because it can lead to under-segmentation or over-segmentation in the suspicious regions.

$$D(p) = |L_I(p) - L_B(p)|$$

$$\begin{aligned} p &\Rightarrow \text{region (a)} && \text{if } D(p) < K_1\sigma(p) \\ p &\Rightarrow \text{region (b)} && \text{if } K_1\sigma(p) <= D(p) < K_2\sigma(p) \\ p &\Rightarrow \text{region (c)} && \text{if } K_2\sigma(p) <= D(p) < K_3\sigma(p) \\ p &\Rightarrow \text{region (d)} && \text{if } K_3\sigma(p) <= D(p) \end{aligned} \quad (1)$$

In Eq. (1), L_I refers to luminance components of the current frame and L_B refers to luminance components of the background model. Thresholds K_1, K_1, K_3 are determined by training data. However, we notice that large number of background is identified falsely in suspicious foreground region due to shadow of the objects. In order to eliminate these shadows, we use color component. As we know, a shadow only modifies the brightness of the background but does not alter their color property. Hence we update the regions containing shadows and categorized under region (c) to region (b). This is illustrated in Eq. (2).

$$p \Rightarrow \text{region(b)} \quad (2)$$

$$\text{if } (p \in \text{region(c)} \;\&\; |H_I(p) - H_B(p)| < K_1\sigma_H(p))$$

In Eq. (2), H refers to the color component of the frames/images and σ_H is the standard deviation of the color component in the background model.

D. Object Recognition

Once the regions are classified for all the pixels, we begin the labeling step where pixels belonging to region (c) and region (d) are labeled using the 8-neighbor labeling algorithm. In this algorithm, same labels are assigned to all the connected pixels belonging to the foreground regions using the region growing technique. After labeling all the foreground pixels, we produce the output file on the HDFS using the pixel setting method by providing RGB components of the obtained HSI model of the foreground regions. This will store the output as image to the output path on the HDFS.

IV. PERFORMANCE EVALUATION & COMPARISON WITH MATLAB

Time and memory consumption plays a major role in the performance evaluation. The number of targets with false alarms also acts as a good measure for performance evaluation. Lesser the false alarm rate, better is the algorithm. Counting the instructions in line code or using complexity analysis can easily measure the amount of time and memory consumed. On the other end, detection of alarm rate for performance evaluation uses three main approaches in general.

A. ROC Curve Evaluation

The evaluation using Receiver Operation Character (ROC) Curves require metrics. These ground truth based metrics are computed from the true positives (TP), true negatives (TN), false positives (FP), false negatives (FN). FP and FN are referred to as pixels which are falsely classified as foreground (FP) and background (FN) i.e. pixels belonging to region (c) and region (b) respectively while TP and TN are referred to as pixels that are accurately classified as foreground (TP) and background (TN) i.e. pixels belonging to region (d) and region (a) respectively. Now, various metrics such as the detection rate also known as the recall, the precision, the F-Measure can be computed using this classification of pixels. Detection rate illustrates the percentage of pixels accurately classified as foreground as compared with the total sum of misclassified background pixels and corrected foreground pixels in the ground truth:

$$DR= TP / (TP + FN)$$

Precision gives the percentage of pixels accurately classified as foreground as compared with the total number of pixels classified as foreground:

$$Precision= TP / (TP + FP)$$

Achieving good performance is marked by higher Detection Rate without change in the Precision. The effectiveness measure known as F-Measure is determined by:

$$F= (2 * DR * Precision) / (DR + Precision)$$

The performance of classification in Precision- Detection Rate is characterized by the F-Measure. The objective is to maximize F close to one to achieve more and more effectiveness. The performance comparison for the given system in HIPI and MATLAB has been achieved and are shown in the table below:

Metrics	**Applications**	
	MATLAB	***HIPI***
DR	0.03742	0.2112
Precision	0.29684	0.0368
F-Measure	0.06610	0.0626

Table. *Peformance Comparision*

From the table, we can observe that the results are better in HIPI as compared to MATLAB. This has been the case when we use single node cluster in Hadoop. Hence, we can say that these metrics results will get better in HIPI with increase in number of cluster nodes as more parallelism is achieved.

V. CONCLUSION

In this paper, we provide a test-bed for implementation of [4] in Hadoop that allows parallel processing for object recognition along with a few changes in the technique such as implementing on-line approximation technique using mean filters. The current applications of the system include feeds from images and can be taken forward to integrate video broadcasting websites such as YouTube object recognition in video streams. The current design of the proposed system in Hadoop works statically but can be implemented in real-time using real-time processing models in Hadoop such as Spark.

VI. ACKNOWLEDGMENTS

This research was supported by the School of Computer Science and Engineering (SCSE), Vellore Institute of Technology, Chennai Campus. Parts of this work have appeared in [4], [5].

VII. REFERENCES

[1] Dawei Li, Lihong Xu, Erik D. Goodman, "Illumination-Robust Foreground Detection in a Video Surveillance System", IEEE Trans. Pattern, vol. 23, no. 10, pp. October 2013.

[2] Shih-Wei Sun, Yu-Chiang Frank Wang, Fay Huang, Hong-Yuan Mark Liao, "Moving Foreground Object Detection Via Robust SIFT Trajectories", J. Vis. Commun. Image R. 24, pp. 232-243 December 2012.

[3] Kyungnam Kim, Thanarat H. Chalidabhongse, David Harwood, Larry Davis, "Real-time Foreground-Background Segmentation Using Codebook Model", Elsevier, pp. 2004.

[4] Hansung Kim, Ryuuki Sakamoto, Itaru Kitahara, Tomoji Toriyama and Kiyoshi Kogure, "Robust Foreground Extraction Technique Using Gaussian Family Model and Multiple Thresholds", ACCV 2007, pp. 758-768, 2007.

[5] Chris Stauffer, W.E.L. Grimson, "Adaptive Background Mixture Models For Real-time Tracking", MIT Cambridge.

[6] Hadoop image processing interface website: *hipi.cs.virginia.edu/*

[7] Chris Sweeney, Liu Liu, Sean Arietta, Jason Lawrence, "HIPI: A Hadoop Image Processing Interface for Image-based MapReduce Tasks", University of Virginia.

[8] Aruna U Gawde, Miraj Shah, Imaad Ukaye, Mihir Nanavati, "Object Detection Using Hadoop", Ijaret, Vol. 1, Issue XI, ISSN 2320-6802, pp. Dec 2013.