A Survey on Live Video Stream using Distributed Technologies

Abhinav Pandey, Harendra Singh

***Abstract:** As data is growing in multiple dimensions with lots of verities and gradually becoming humungous so there is high demand of frameworks for processing Big Data. There were frameworks available for processing structured and semi-structured data but for processing un-structured data with real time analysis, very few options were availabe which can process only limited amount of data and high volume of data was a bottleneck for IT industries. It was not a big deal to work with 'data in rest' and only few frameworks available for analyzing data in motion like Apache Storm, got lots of motivation. Recently, for live streaming analysis and instant decision making Spark Streaming got introduced by Data Bricks and this is gaining lots of limelight due to its easy configuration and setup with loads of machine learning techniques and reliability at distributed platform. Using Storm prediction and face recognition were implemented. Through this study we will implement Real time video streaming analysis using Spark Streaming.*

***Keywords:** Distributed system, Hadoop, Spark, Spark Streaming, Opne CV*

I. INTRODUCTION

IT industries are surrounded with 3Vs (Volumn, Varity, Velocity) of big data, such voluminous data can come from myriad different sources, such as business sales records, the collected results of scientific experiments or real-time sensors used in the internet of things. Data may be raw or preprocessed using separate software tools before analytics are applied. Data may also exist in a wide variety of file types, including structured data, such as SQL database stores; unstructured data, such as document files; or streaming data from sensors. Further, big data may involve multiple, simultaneous data sources, which may not otherwise be integrated. Project may attempt to gauge a product's success and future sales by correlating past sales data, return data and online buyer review data for that product. Currently most of the innovations are dedicated to perform live streaming analysis and instant decision making like IoT. For live streaming analysis Apache Storm gained lots of focus but after getting introduced the Spark Streaming, most of the researches and developers are moving towards it and trying to develop a reliable and distributed system to perform real time event based decision making. A famous framework Strom gained popularity for real time analysis for face recognition of pedestrians by analyzing live video streaming, aims to identify movements and faces by performing image matching with existing images using Open CV and Storm CV, on distributed platform and deliver decisions to concerning departments. Since video processing and analysis from multiple resources become slow when using Cloud or even any single highly configured machine, hence for making quick decisions and actions, Apache Storm and Kafka have been used as a real time analysis frameworks, which delivers event based decisions making on Hadoop distributed environment. Apache Storm is a distributed real-time computation system for processing fast, large streams of data. Storm adds reliable real-time data processing capabilities to Apache Hadoop® 2.x. Storm in Hadoop helps capture new business opportunities with low-latency dashboards, security alerts, and operational enhancements integrated with other applications running in their Hadoop cluster. The storm is a distributed, reliable, fault-tolerant system for processing streams of data. The input stream of a Storm cluster is handled by a component called a spout. The spout passes the data to a component called a bolt, which transforms it in some way. A bolt either persists the data in some sort of storage, or passes it to some other bolt. You can imagine a Storm cluster as a chain of bolt components that each makes some kind of transformation on the data exposed by the spout.

For Image analysis a open-source image detection libraries, used to images extracted from the videos. There are two main open-source free vision libraries to process images. On the one hand, OpenCV (Open Computer Vision) is library written in C/C++ designed for computational efficiency and strongly focused on real-time applications. It is a well documented and widely used project. More than 500 functions are implemented in OpenCV to cover areas such as robotics, security, medical imaging or factory product inspection. It also includes a general-purpose Machine Learning Library (MLL) focused on statistical pattern recognition. On the other hand, OpenIMAJ (OPEN Intelligent Multimedia Analysis in Java) is a Java library and tool for scalable multimedia content analysis and image indexing. It includes broad state-of-the-art computer vision techniques. The distribution is made using a modular set of jar files under a BSD-style license. The design and implementation keep all the components modular to maximize code maintainability. Both libraries employ similar algorithms to detect images. With all these similarities, we selected OpenIMAJ to implement our code that use the Scale-Invariant Feature Transform (SIFT) to extract some interest pixels from images and describe them. To find analog pixels, we use Random

Manuscript published on 28 February 2018.
*Correspondence Author(s)

Abhinav Pandey, Department of Computer Science & Engineering, Lakshmi Narain College of Technology Excellence, Bhopal (M.P.), India. E-mail: abhinavpndy@gmail.com

Mr. Harendra Singh, Assistant Professor, Department of Computer Science & Engineering, Lakshmi Narain College of Technology Excellence, Bhopal (M.P.), India. E-mail: harendra.cse07@gmail.com

Sample Consensus (RANSAC) to fit a geometric model called an Affine Transform to the initial set of matches. After the pattern detection process, we obtain a number of matches or similarities. Closeness can be compared with a threshold. If the number is greater, we have a frame that contains the logotype. Notice that the goodness of the systems greatly depends on the goodness of the library. StormCV enables the use of Apache Storm for video processing by adding computer vision (CV) specific operations and data model. The platform enables the development of distributed video processing pipelines which can be deployed on Storm clusters.

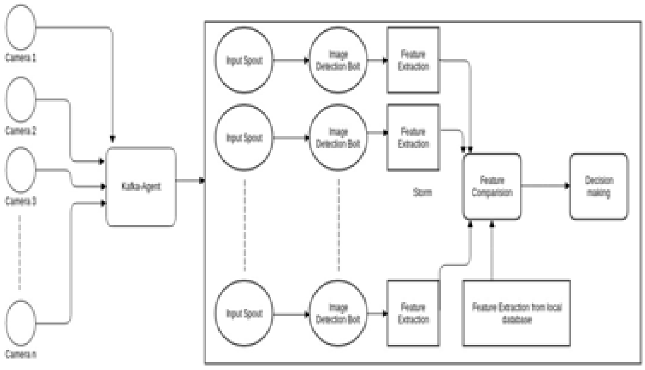
Spark Streaing also introduced which provide easy configuration, fast processing and efficeitn decision making.

II. BACKGROUND

Videos survilience is widly getting used for in security, health care, traffic cotrol, automobiles manufecturing and Airoplane fault detection. The new buzzing word arround us is IoT now a days, gaining lots of limelight and providing solution for various problems. In order to solve the live streaming analysis, researches worked with Apache Storm, Kafka and other frameworks. This work is derived by last uploaded worked on live streaming analysis, to provide more efficeint decision on the based of live video streaming analysis.

III. APACHE STORM FOR LIVE VIDEO STREAMING

This framework performs face recognition of pedestrians from live video streaming where The Kafka-Storm Topology is being designed to analyze Surveillance video. The system will extract Human faces from Surveillance video and compares with faces in Database to detect and the decision will be forwarded to concerned departments. The system is a real time face recognition for Surveillance video. The Kafka – Agent collects the images from various videos sources at the desired frequency. Frequency. It transmits the image string data to the Kafka Cluster. The entry level of the Storm Topology is the Spout which receives the String data from Kafka Cluster.

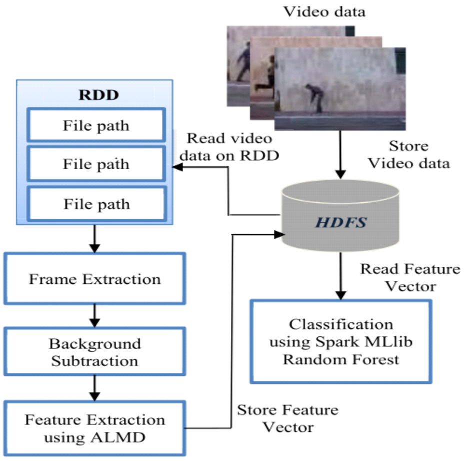


IV. VIDEO ANALYSIS USING APACHE SPARK

Recently, due to the rapid advancement of the Internet, social media video services and intelligent CCTV for video surveillance system, the multimedia data such as video is increasing rapidly. Moreover, understanding video context and identifying video types has an important significance in the management of massive video data. In order to manage

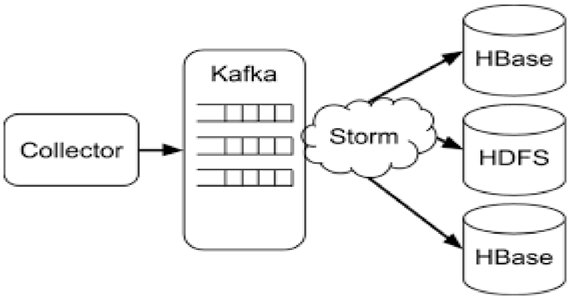
These videos and provide important services to the users, it is necessary to understand the human activities from the videos automatically. There are many applications, which focus on the action recognition, such as crowd behavior prediction, video surveillance, human-machine interaction, and sports game analysis. Human action recognition is an intricate field since static object characteristics, time, and motion features have to be considered. Moreover, due to the environmental variations including different viewpoints, moving backgrounds and large intra-class variations of different actions, the recognition of human actions is even more difficult. On the other hand, with the exponential growth of the multimedia data and videos from the different origins e.g. CCTVs, it increases the demand of distributed computing to provide the services efficiently. For example, in every minute almost 300 hours of video are uploaded. Existing video processing system uses the Hadoop platform in order to perform the distributed computing, however, it shows low efficiency in the iterative computation, which is essential in the machine learning. Moreover, it does not support the real time computation.

Videos can be stored in the Hadoop distributed file system (HDFS) and dataset is loaded into a Spark cluster, which is represented by resilient distributed datasets (RDDs). In RDD partition, we perform frame extraction from video data, background subtraction and finally, Adaptive Local Motion Descriptor (ALMD) is introduced to extract the motion feature. These RDD operations are done in parallel using each worker node and provides in-memory based computations. Adaptive Local Motion. Descriptor is inspired from Local Binary Pattern (LBP). and Local Ternary Pattern which are only able to extract the static texture information.



V. KAFKA FOR VIDEO BUFFERING: KAFKA IS PERFORMING TWO TASKS

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)



A. Stream Data Buffer

To process a huge amount of video stream data without loss, it is necessary to store the stream data in temporary storage. The Kafka broker works as a buffer queue for the data that the video stream collector produces. Kafka uses the file system to store the messages, and the length of time it retains these messages is configurable.

B. Video Stream Collector

The video stream collector works with a cluster of IP cameras that provide live video feeds. The component must read the feed from each camera and convert the video stream into a series of video frames. To distinguish each IP camera, the collector maintains the mapping of camera ID and URL with camera.url and camera.id properties in a stream-collector. properties file.

VI. SUGGESTIONS

The Apache Spark Streaming is better framework for reliable, easy configuration, fast processing and efficient decision making over the other real time analysis frameworks and performing pedestrain or motion detection and face recugnition will be more efficent using apache Spark Streaming. If OpenCV for image or video analysis and Spark Streaming as Engin for decision making and Kafka as a stream collector will be used for this purpose, so the prepared system will be more efficient then others.

VII. CONCLUSION

The system, prepared with the Apache Spark Streaming and Kafka will be more efficient and less complex, easy to enhacne and configure and in future, developer will get better support as a dedicated compony DataBriks working on this product.

REFERENCE

1. Human Action Recognition Using Adaptive Local Motion Descriptor in Spark, Md Azher Uddin, Joolekha Bibi Joolee, Aftab Alam, And Young-Koo Lee, Department of Computer Science and Engineering, Kyung Hee University, Yongin 1732, South Korea
2. Du-Hyun Hwang, Yoon-Ki Kim and Chang-Sung Jeong,"Real –Time Pedestrian D Etection Using A Pache S Torm In A D Distributed Environment".
3. Shreenath Dutt, Ankita Kalra, "A Scalable and Robust Framework for Intelligent Real-time Video Surveillance"
4. Lokesh Babu Rao, C. Elayaraja 2, "Image Analytics on Big Data In Motion - Implementation of Image Analytics CCL in Apache Kafka and Storm"
5. OpenCV http://docs.opencv.org/2.4/modules/contrib/doc
6. StormCV https://github.com/sensorstorm/StormCV
7. Jianbing Ding, Hongyang Chao, Mansheng Yang, "Real-Time Logo Recognition from Live Video Streams using an Elastic Cloud Platform".
8. Weishan Zhang, Liang Xu, Pengcheng Duan, Wenjuan Gong, Xin Liu, Qinghua Lu,"Towards a High Speed, Video Cloud Based on Batch processing Integrated with Fast processing"
9. Hongyang Chao, "Real-Time Logo Recognition from Live Video Streams Using an Elastic Cloud Platform"
10. Vaithilingam Anantha Natarajan, Subbaiyan Jothilakshmi, Venkat N Gudivada, "Scalable Traffic Video Analytics using Hadoop MapReduce"
11. Supun Kamburugamuve, Leif Christiansen, and Geoffrey Fox, "A Framework for Real Time Processing of Sensor Data in the Cloud"

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)
© Copyright: All rights reserved.