CS5543 Real-Time Big Data Analytics

Project Report 4

Project Title: VideoCeption

Team 2

Sri Praneeth Iyyapu - 8 Naresh

Goud Pogakula - 20 Vihari

Gorripati - 6 Mounika

Yalamanchili - 25

Project

Objectives

Significance:

The application that we proposed for this project is a video searching web-application. Our web-application is unique because it takes the video-search to another level. Then we will display the smart search results based on the actual content of video instead of names or tags associated with video. It is significant because currently there is no video-search application available that performs similar operation. There are few paid services, however are planning to make the project freely available with the open sourcing option as well.

Features:

1. **Smart Search:** This smart-search is based on the video content. The entire video is analyzed based on the visual and the audio content of the video uploaded to the application. So this is a smart way of searching the video instead of routine name basedsearch.
2. **Smart Tags:** The smart-tags can be user given while uploading the video or else our video-engine analyzes the video content and gives some smart-tags that pertain to the video. The video-search is dependent on the generated smart-tags that are generated.
3. **Spam Filtering:** Many spam videos are uploaded every day to YouTube and other websites. Currently there is no way to possibly identify whether a video is genuine or spam. We try to analyze the content of the video and label the video as spam.
4. **Video Recommendations:** This feature is similar to already existing video-recommendations. But we try to give smart recommendations based on oursmart-tags.
5. **Parental Control:** We can also categorize the videos based on content and generate a flag called “not suitable for children”. Hence if the parents decide to turn on a feature called “parental control”, all the inappropriate videos would be removed from thesearch.
6. **Content based Navigation:** You can navigate to the particular content/entity in the video using this feature.

Approach:

Data Sources: Our application needed videos particular to sports field. So we are collecting the videos from YouTube channels.

Analytic Tools:

- Apache Spark
- Apache Storm
- Apache Kafka
- OpenIMAJ

Analytical Tasks:

The Video that our application takes as input is analysed using OpenIMAJ. This tool will analyses each frame of the video and select the main frames which are completely different to other frames.

The main frames are analysed using deep machine algorithms which will be executed on Apache Spark with the help of Storm and Kafka.

Expected Inputs/Outputs:

The application takes the search words from user and based on the search word. It displays related videos. These videos are selected based on the content of the video.

Algorithms:

The algorithm that we will use for detecting the patterns, objects and actions and store that particular information so that our application can categorize the videos and display them according to the user search.

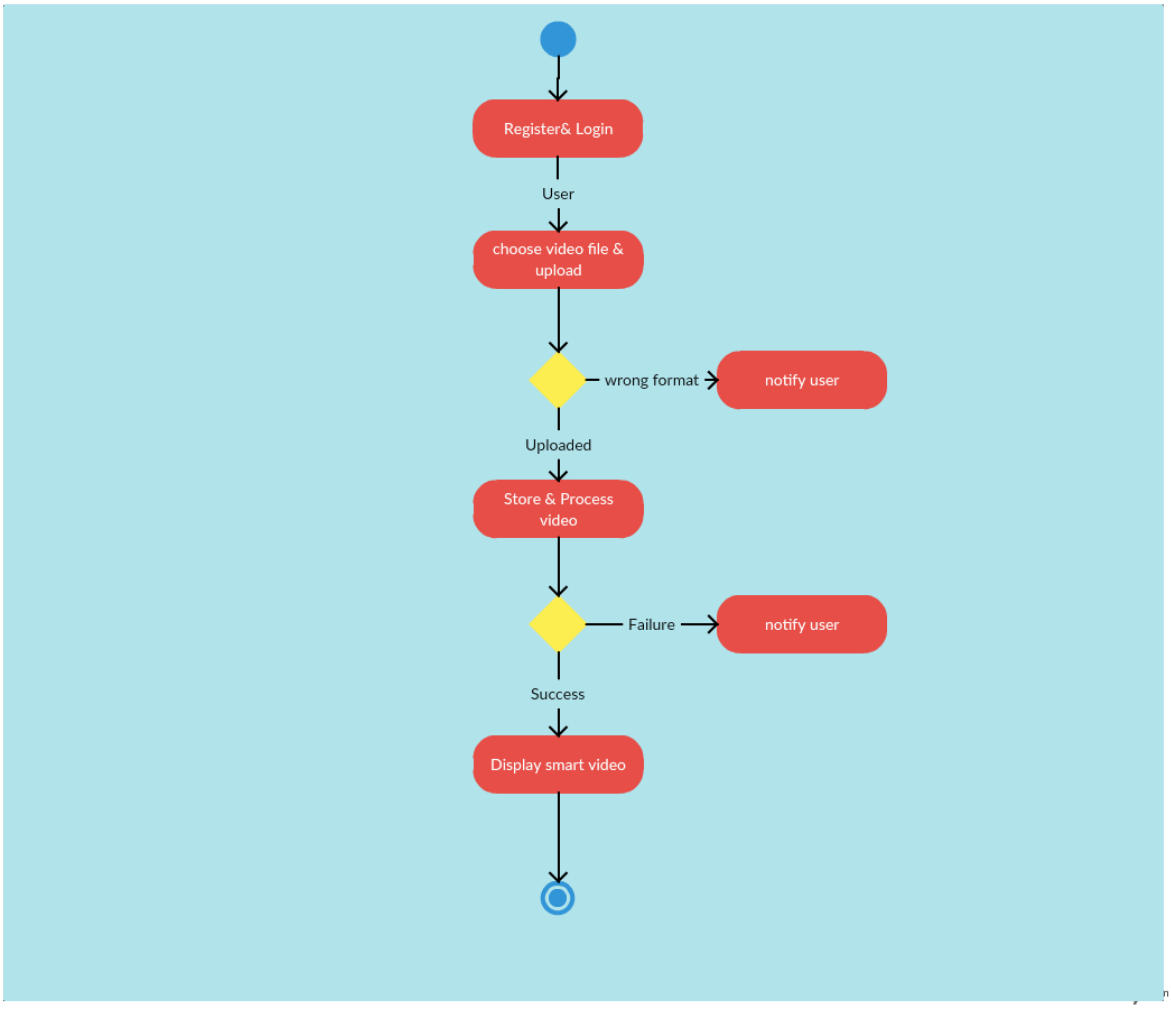
Related Work:

- Open Source Projects:
    - Dextro: https://www.dextro.co/
    - Clarifai: https://www.clarifai.com/

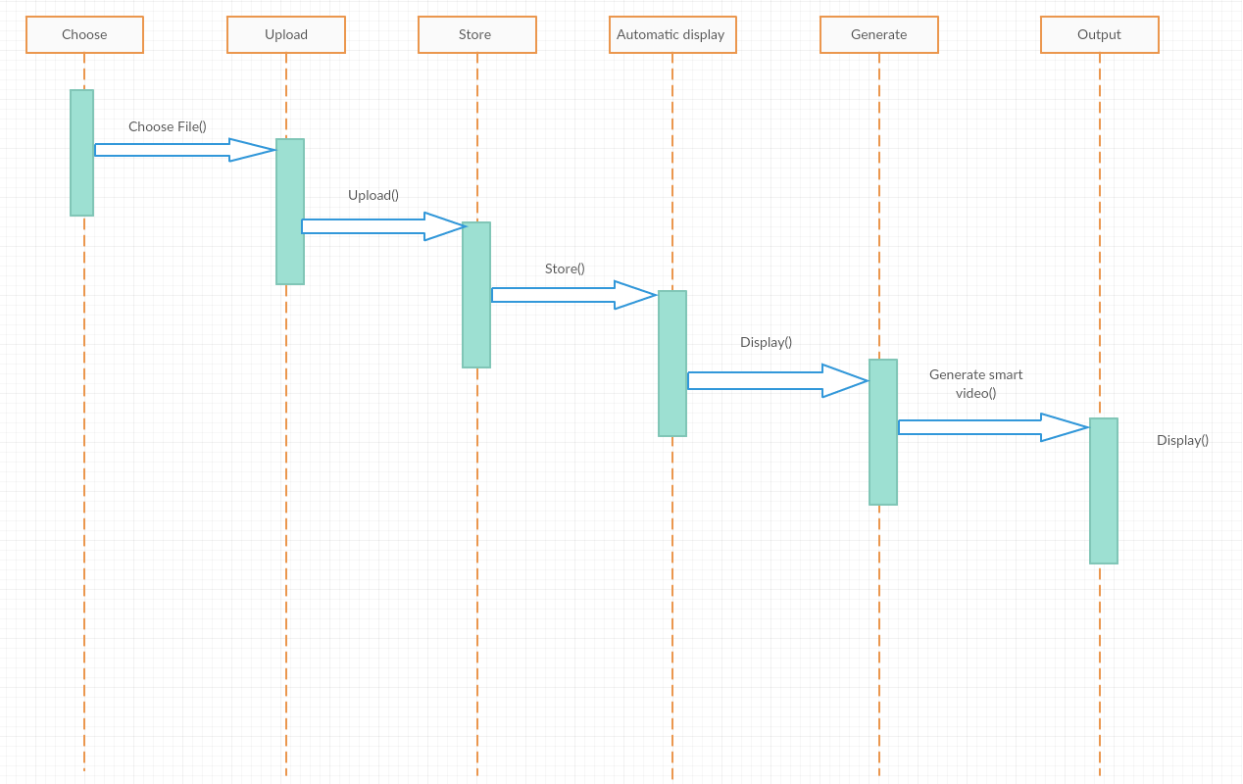
Application Specification:

- **System Specification**
  - **Software Architecture**

- **Design of Mobile Client(smartphone/web)**
  - **Features, GUI, Technologies**
    - **Activity Diagram (workflow, data, task)**

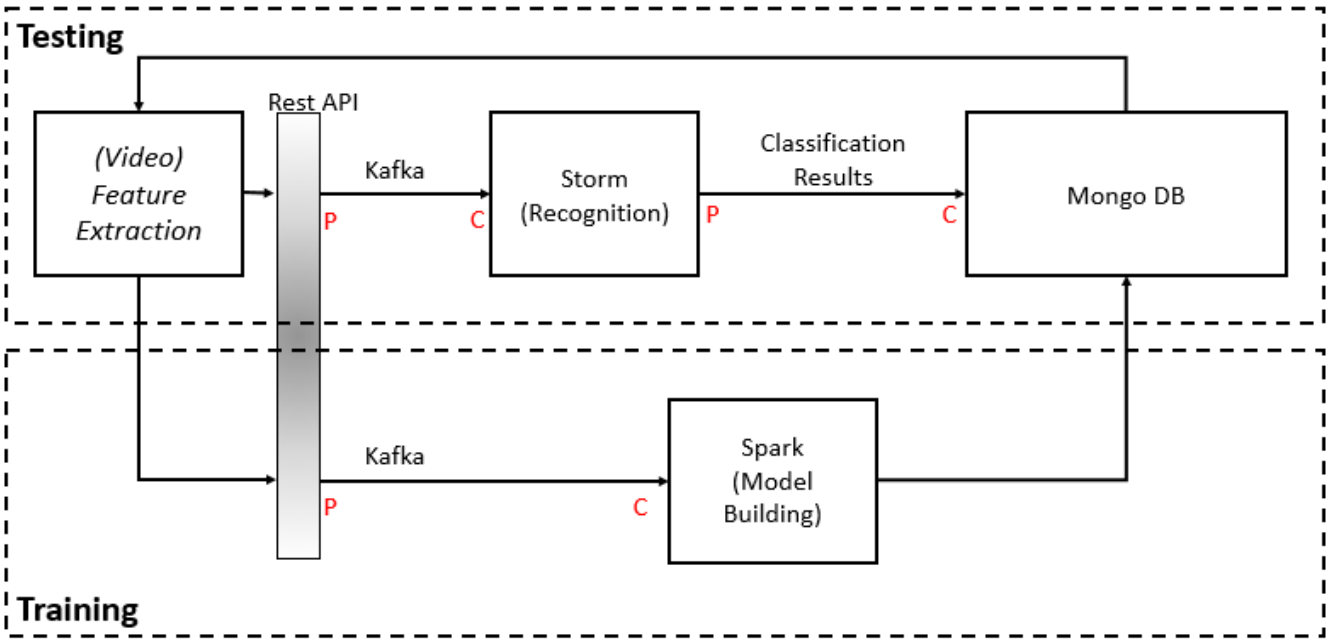


- **Sequence Diagram(interaction/collaboration)**



- **Existing Applications/Services Used:**
  - Dextro, it makes videos discoverable, searchable, and curatable using machine learning, https://www.dextro.co/
  - Clarifai, Powerful, secure, and affordable visual recognition API for developers and businesses, https://www.clarifai.com/
  - OpenIMAJ, Intelligent Multimedia Analysis, http://openimaj.org/

Implementation:

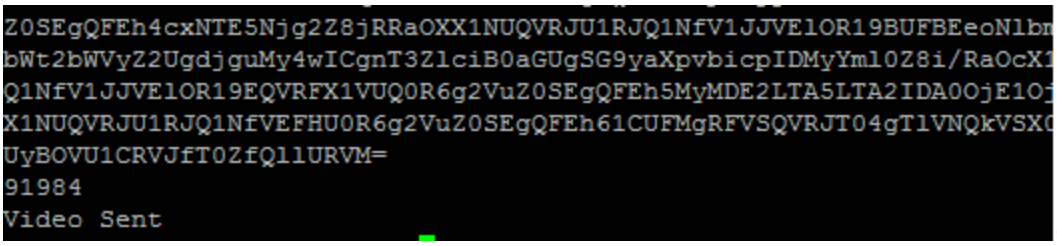


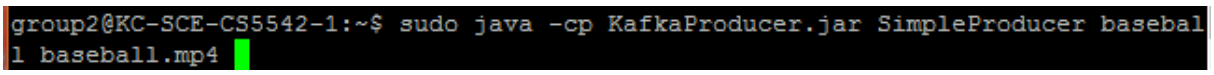
- The idea is to train the system to find out the videos related to sports field (especially baseball). To achieve this, we have created objects like ball, bat etc. and stored these models in mongo DB and these models are used to match with the objects found in other videos. If the same type of object present in the videos, we classify them as baseball videos. So these videos can be sent to user interface so that user can see these videos only. As this search is not based on the name or tags of the video and only based on the content of the video, user can't see spam videos.
- **Training:** First we have extracted the features and send those model to spark machine learning library. After that we have stored that model in mongo DB.
- **Testing:** Then other videos are searched using the model. These tasks are executed in storm. Kafka is used to send and receive the video in between them.

- Send the video to **Kafka Producer** and get the tags for eachframe.
  - For this increment we are sending the video from client side UI to server side.
  - Now various topics are created like baseball, cricket, ball, sports etc.

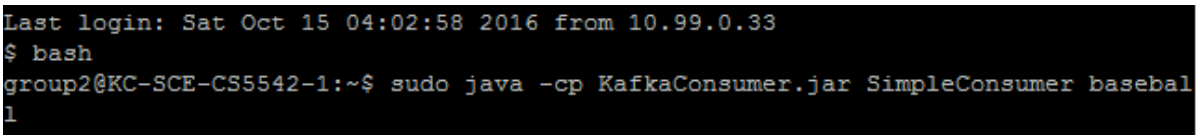
```
Last login: Sat Oct 15 02:49:54 2016 from 10.99.0.33
$ bash
group2@KC-SCE-CS5542-1:~$ /usr/local/bin/kafka_2.11-0.10.0.1/bin/kafka-topics.sh
 --create --zookeeper localhost:2181 --replication-factor 1 --partitions 1 --top
ic sports
Created topic "sports".
group2@KC-SCE-CS5542-1:~$
```

  - So the producer produces the video.





- The consumers are subscribed to the respective topics.



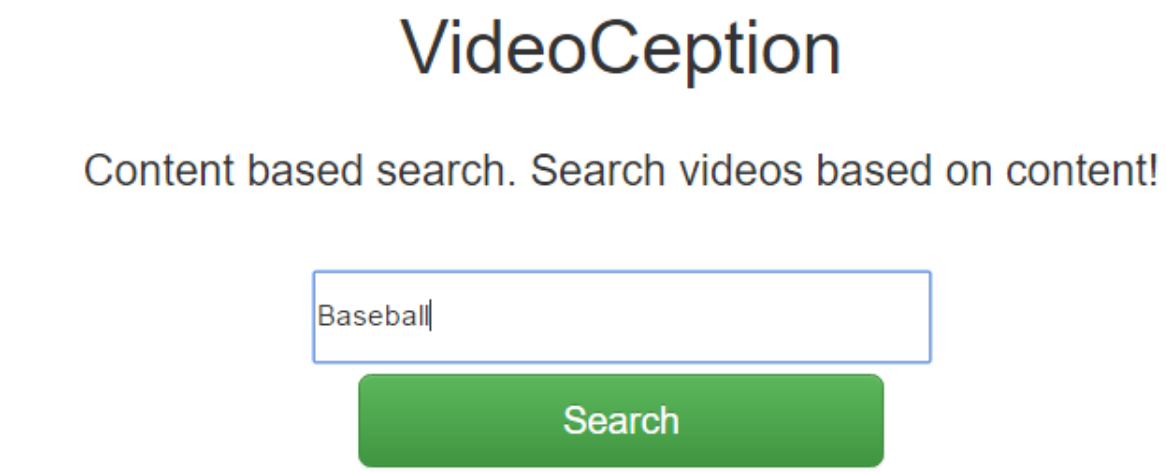
- We also create various bolts for the logic part. Each bolt pertains to handling thedata pertaining to the different topics.
- For example, for a topic called baseball we develop the logic in a bolt such that the bolt recognizes all the baseballs in the input based on say training data that it has.
- In the similar way we can develop logics for different topics in differentbolts.
- All these data are consumed by the consumer based on the topics that it subscribedto.
- So this is our basic topology for this increment.

- **Storm:**
  - Storm topology is created in such a way that it takes the input videos sent by Kafka and the storm bolt will execute on those videos to to find out the given models.
  - The videos which are having the given models are sent back to the user through Kafka. And this information can be logged into the Mongo DB.

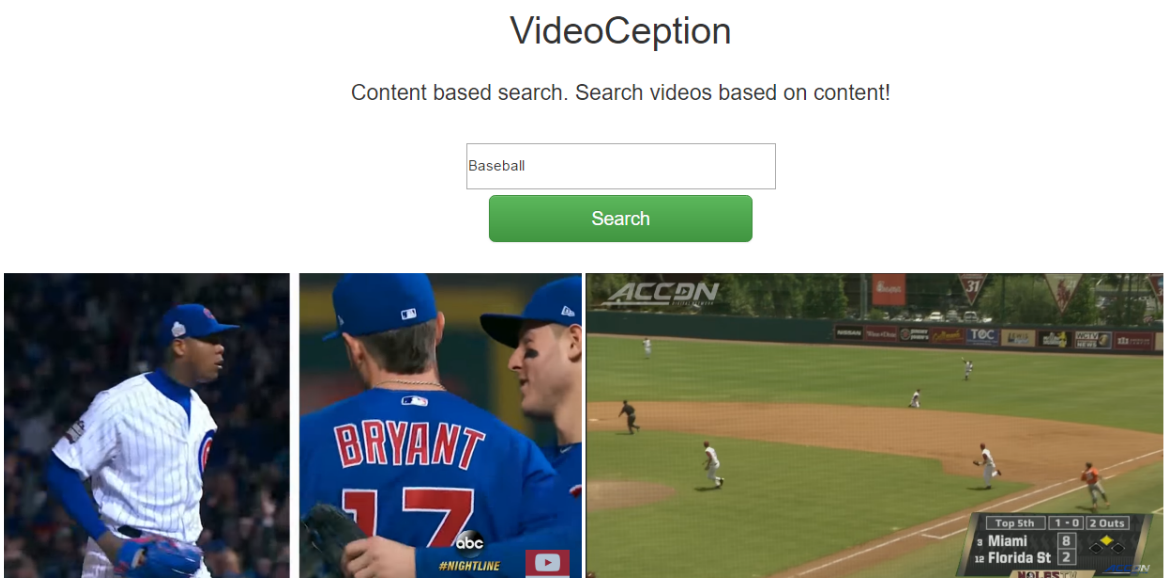
Documentation:

- The home screen of our application has threeoptions.
  - **Search:** User can search the videos by giving input.
  - Based on the search input, our system can build its own models and check if available videos are having the respected model pertaining to the user input.
  - Finally, it displays all the videos that are related to search keyword.

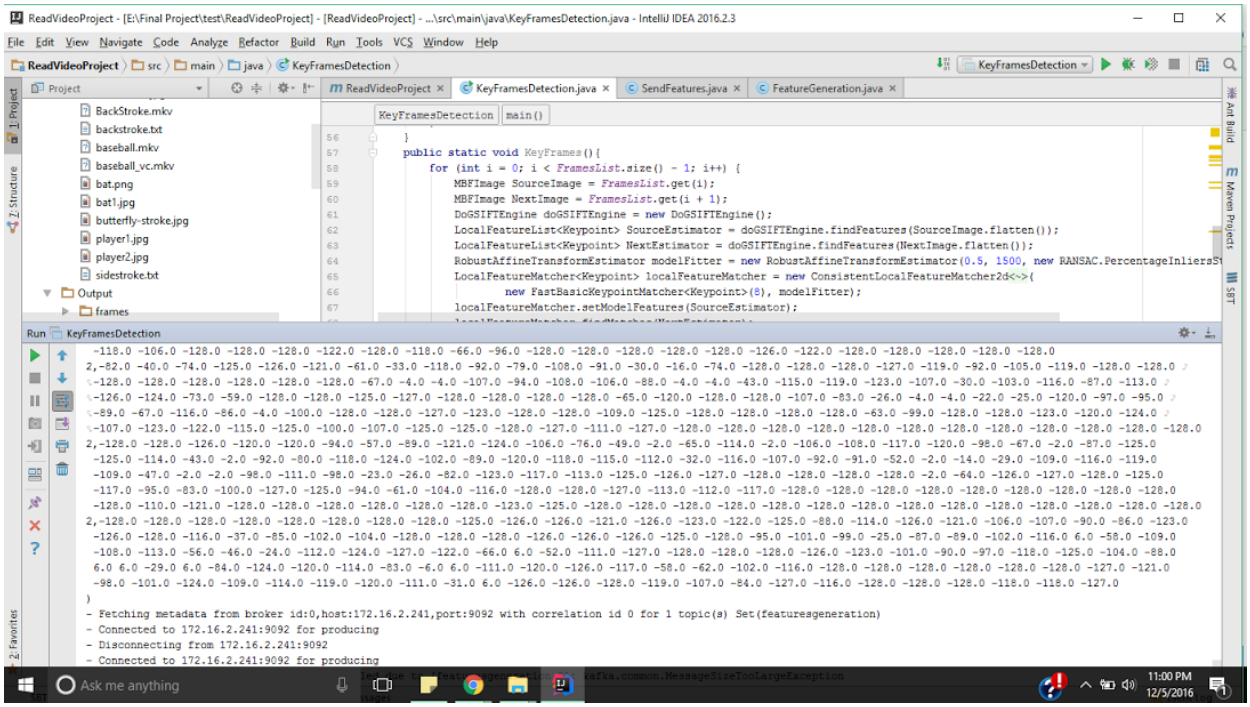
- Search for baseball:



- Videos that are related to baseball, ie if the content inside the video is of baseball, then that videos are displayed



- Feature Extraction:



- Model Detection:

Project Management:

- **Work Completed:** We have completed our entire application end-to-end.
- **Time taken:** It nearly took around 120 hours of our time.
- **Contribution:** Everyone has contributed equally.
- **Work to be completed:** None.
- **Time to be taken:** None
- **GitHub Repo:** https://github.com/niCEnANi/RealTimeBigData/tree/master/Project%20Report%204