Proposal

Ran Lin,Rui Li,Jin Huang,Yuhan Liu

February 12, 2018

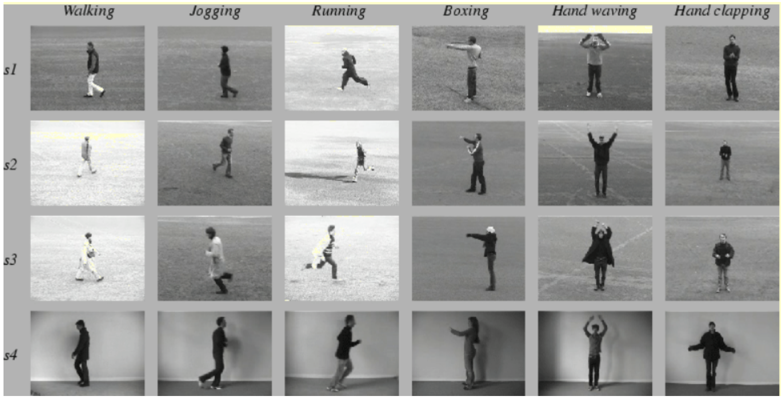
1 Team Members

Ran Lin(DS): Mathematical modeling and data analysis.
Rui Li(DS): Data analysis and data visualization.
Jin Huang(CS): Amazon S3 and EC2, data storage and cluster build.
Yuhan Liu(CS): Spark RDD and data preprocessing.

2 Data sets

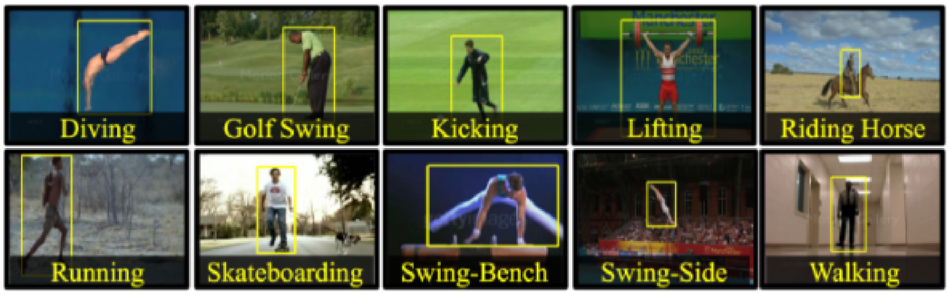
After careful discussion, we decided to change the dataset. We are going to use several different datasets together to finish our project. Including:

1 KTH dataset which contains six types of human actions (walking, jogging, running, boxing, hand waving and hand clapping) performed several times by 25 subjects in four different scenarios: outdoors, outdoors with scale variation, outdoors with different clothes and indoors. Currently the database contains 2391 sequences. All sequences were taken over homogeneous backgrounds with a static camera with 25fps frame rate.

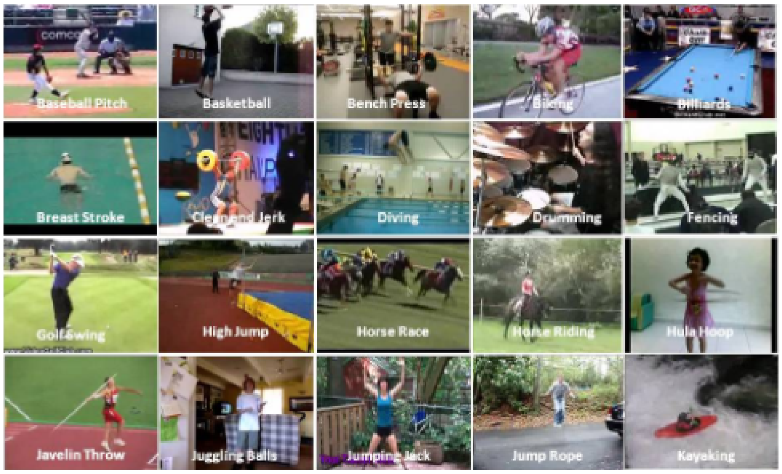


2 UFC dataset UCF Sports dataset consists of a set of actions collected from various sports which are typically featured on broadcast television channels such as the BBC and

ESPN. The video sequences were obtained from a wide range of stock footage websites including BBC Motion gallery and GettyImages.



3 UCF-50 dataset UCF50 is an action recognition data set with 50 action categories, consisting of realistic videos taken from youtube. This data set is an extension of YouTube Action data set (UCF11) which has 11 action categories.



4 UT-Interaction dataset The UT-Interaction dataset contains videos of continuous executions of 6 classes of human-human interactions: shake-hands, point, hug, push, kick and punch. Ground truth labels for these interactions are provided, including time intervals and bounding boxes. There is a total of 20 video sequences whose lengths are around 1 minute. Each video contains at least one execution per interaction, providing us 8 executions of human activities per video on average. Several participants with more than 15 different clothing conditions appear in the videos.

3 Overview and Introduction

Nowadays,human action recognition plays an important role in computer vision, multimedia research and other applications, such as human-machine interaction etc. Although a lot of researchers have already studied in this field for several years and got some significant results, due to the rapid growth of the data, the question became more difficult, the researchers not only need to improve the recognition algorithm to increase the accuracy, but also need to figure out a way to deal with the big amount of data. Since its impossible to download all the datasets(for example, in every minute almost 300 hours of video are uploaded) into one local machine, even its possible to do that, using the single machine to finish the data processing and data training will be a time-consuming job, therefore combining Hadoop ecosystem and mathematical modeling together to solve the problem will be a good choice. There are already several researchers has referred to this field, but they are still in the process of exploration.

4 Project Scope

We are going to upload all datasets into Amazon S3 buckets, and then set up a cluster with Apahce Spark, read the raw video data on RDD, extract the frame of the video, subtract the background and finally extract the feature using ALMD. After that,we will write back the result to our cluster,and then classify the data using Spark MLib Random Forest.Our goal is to recognize human actions by extracting and analyzing the motions in the videos. Since each of the videos is labeled with an action, like running, walking and so on. For each dataset, we break 70% of them as training set to train our model, then use the remaining 30% data to test the models accuracy. Finally, we will analyze the models accuracy of different dataset.

5 Approach to solve the problem

5.1 Terminology

Frames: still images which compose the complete moving picture.

Pixels: short for picture elements. Digital images can be modeled as simple two-dimensional matrices of intensity values, and each value in the matrices is a pixel. A color image would require three of these matrices, with one each denotes red, green and blue channel.

Intensity: Intensity: for most gray scale images, intensity is integer that ranges from 0 (black) to 255 (white).

5.2 Tools

For the storage since all the datasets we selected provide the link to download, we will create several Amazon S3 buckets, first download the dataset and upload them into buckets.

For the cluster we will deploy Spark on Amazons EC2 cloud computing services and use Spark RDD. Since MapReduce with Hadoop is based on acyclic data flow. And what we are going to do is a kind of iterative task. If we use MapReduce with HDFS, that means every time during an iterate algorithm, we need to read data from the HDFS and finish every stage and write back the results data to HDFS, that will be really slowed due to replication and disk I/O, but HDFS is necessary for fault tolerance. So according to the condition we are facing, Spark RDD is the best way to resolve the problem and provide us a good performance about iterative task, meanwhile with the fault tolerance.

For analyze we choose Spark MLib, MLib is a sparks scalable machine learning library, and its easy to use(have APIs with python and R, etc.) Also, MLib has a pretty good performance, which is almost 100x faster than MapReduce. Finally, Its really easy to deploy on the Hadoop cluster.

For analysis result visualization, we may use some mathematical tools, like matlab and R.

6 Design of the solution

6.1 Preprocessing

Loading data Turn input data into an RDD in parallel using each worker nodes, the following analysis are all done in parallel on each tuple of RDD.

Frame extraction Extract frames from videos and then resize the them into a fixed frame size.

Frame conversion Change the frames RGB color into gray scale.

Background subtraction Here, human is the foreground object, subtract each frame with the background frames will filter out the background.

6.2 Feature extraction using adaptive local motion descriptor (ALMD)

The idea of ALMD is based on detect the local intensity fluctuation between the successive frames, with a threshold selected to filter out noise which would be caused by the changing of shooting angle, illumination variation and other reasons.

Dynamic texture features extraction by using ALMD. Compute the upper and lower ALMD value of each pixel in each frame, which indicate the increase and decrease of pixels intensity, then record both of them into two vectors separately. Here, each video will be ensemble with two vectors of length equal to number of frames in that video minus one.

Collect vectors from the videos at the end of flatMap() function, combine the video names with the responding vectors to result RDD.

6.3 Classification with Random Forests

This model takes the vectors collected in step 2 as input, utilizing the Spark Mllib to implement that model. The elements in vectors are regarded as variables, and the video are regarded as instances. The idea of Random Forests is to build plenty of sub-tree, where each trees input is the bootstrap samples of the instances, and each node of the trees supposed to be split by the best one of the randomly chosen variables, then the prediction would be made by the majority vote of all the trees. Here sub-trees are trained in parallel and TreePoint structure are applied for memory saving.