

CS5542 : Lab Assignment #4

Due on Wednesday, February 17, 2016

Yugyung Lee

Sudhakar

1: Hadoop MapReduce Algorithm

Question:

Implement MapReduce algorithm for finding Facebook common friends problem and run the MapReduce job on Apache Hadoop. Write a report including your algorithm and result screenshots.

Description:

Given that the friends of a person are stored as $A \rightarrow (B, D, E)$ we can use this to feed the mapper, which will generate intermediate output of key-value pairs with key being friend along with the person. The value containing list of friends of that person.

Ex: Map($A \rightarrow BDE$)

(AB) \rightarrow (BDE)

(AD) \rightarrow (BDE)

(AE) \rightarrow (BDE)

Map($B \rightarrow ACDE$)

(AB) \rightarrow (ACDE)

(BC) \rightarrow (ACDE)

(BD) \rightarrow (ACDE)

(BE) \rightarrow (ACDE)

Grouping them will result as: $(AB) \rightarrow (BDE)(ACDE)$

Upon giving this to the reducer, which will intersect the values and output the same key. For Ex: $(AB) \rightarrow (BDE)(ACDE)$ will output $(AB) \rightarrow (CD)$

The final result will be:

(A B) \rightarrow (C D)

(A C) \rightarrow (B D)

(A D) \rightarrow (B C)

(B C) \rightarrow (A D E)

(B D) \rightarrow (A C E)

(B E) \rightarrow (C D)

(C D) \rightarrow (A B E)

(C E) \rightarrow (B D)

(D E) \rightarrow (B C)

Screenshot:

```

172.16.2.241 - PuTTY
$ hadoop fs -cat hadoopinput
A B C D
B A C D E
C A B D E
D A B C E
E B C D
$

```

Figure 1: Hadoop Input

```

CPU time spent (ms)=2880
Physical memory (bytes) snapshot=821559296
Virtual memory (bytes) snapshot=4148822016
Total committed heap usage (bytes)=989331456

Shuffle Errors
      BAD_ID=0
      CONNECTION=0
      IO_ERROR=0
      WRONG_LENGTH=0
      WRONG_MAP=0
      WRONG_REDUCE=0
File Input Format Counters
      Bytes Read=46
File Output Format Counters
      Bytes Written=57

$ hadoop fs -ls
Found 4 items
drwx-----   group6 supergroup          0 2016-02-18 02:58 .staging
-rw-r--r--   3 group6 supergroup        46 2016-02-18 02:43 hadoopinput
-rw-r--r--   3 group6 supergroup          0 2016-02-18 02:50 hadoopoutput
drwxr-xr-x   group6 supergroup          0 2016-02-18 02:58 out
$ hadoop fs -cat out
cat: 'out': Is a directory
$ hadoop fs -cat out/part-r-00000
AB      CD
AD      BC
BC      ADE
BE      CD
CD      ABE
DE      BC
$ vi input.txt
$ ls
input.txt  MutualFriendsMapReduce-1.0-SNAPSHOT.jar  output.txt
$ hadoop fs -ls out
Found 3 items
-rw-r--r--   3 group6 supergroup          0 2016-02-18 02:58 out/_SUCCESS
-rw-r--r--   3 group6 supergroup        38 2016-02-18 02:58 out/part-r-00000
-rw-r--r--   3 group6 supergroup        19 2016-02-18 02:58 out/part-r-00001
$ hadoop fs -cat out/part-r-00001
AC      BD
BD      ACE
CE      BD
$

```

Figure 2: Hadoop Output

2: Watch App

Question:

Implement asmartwatch/smartphone application using existing speech services/image services (e.g., IBM Alchemyapi, Face++) related to your project

Description:

In our project - Tour guide, ROboMe will be taking a series of images to map the interiors of an architecture. This requires image recognition technique. I have used the face detection service provided by the android FACE API. With this Google play service of face detection, the orientation of head and the rotation of head with respect to neck (left facing or right facing) can be detected. It also provides a function call called 'landmarks' which highlights the features identified in the face. To detect the facial statuses below method calls are available.

- `getIsLeftEyeOpenProbability()` - Returns a value between 0 and 1, giving probability that the left eye is open
- `getIsRighteyeOpenProbability()` - Returns a value between 0 and 1, giving probability that the right eye is open

- `getIsSmilingProbability()` - Returns a value between 0 and 1 giving a probability that the face is smiling

Screenshots:

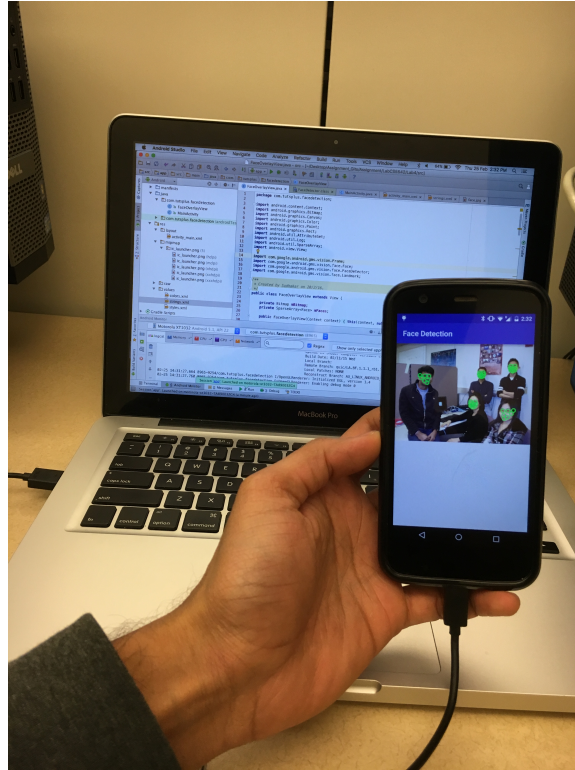


Figure 3: Face detection

Reference:

I have went through the below reference tutorial for using Google Face API. I have not included the same code but made use of the structure mentioned in the link - <http://code.tutsplus.com/tutorials/an-introduction-to-face-detection-on-android-cms-25212>

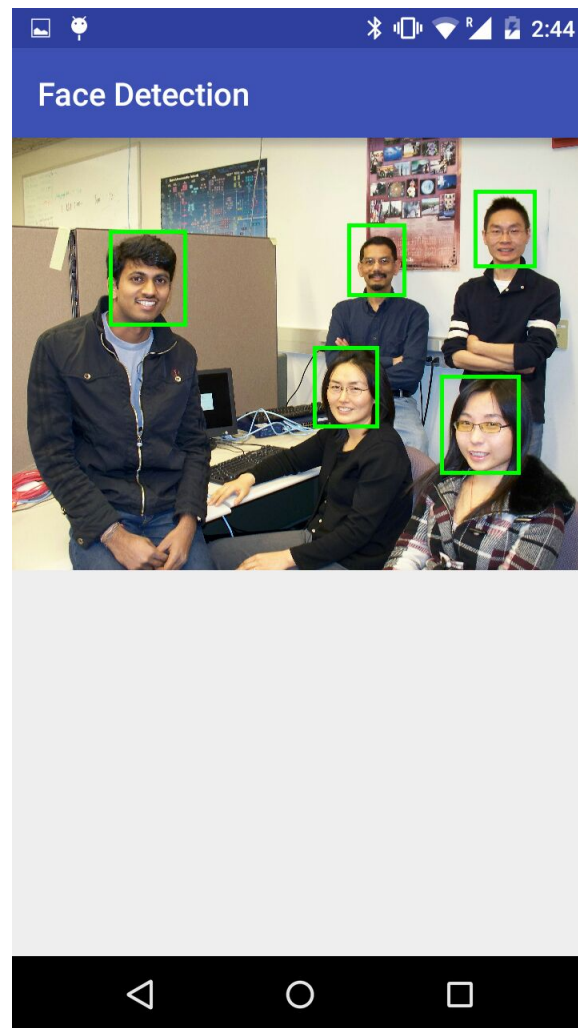


Figure 4: FaceBox

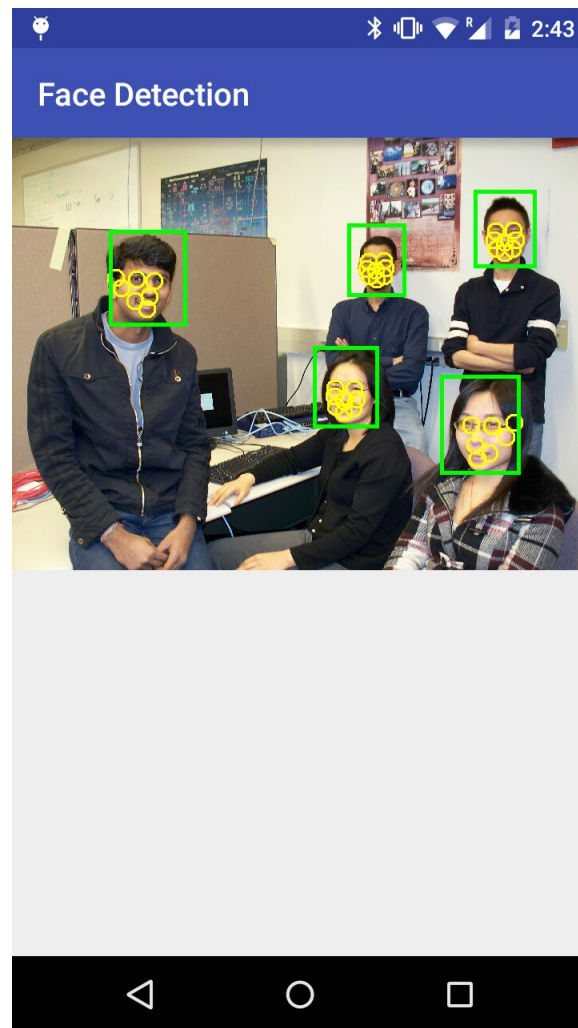


Figure 5: FaceBox with Landmarks