Face Extraction and Recognition from Public Images Using HIPI

Conference Paper · November 2018

DOI: 10.1109/ICECCO.2018.8634718

3 authors:

Andrey Bogdanchikov
Suleyman Demirel University
18 PUBLICATIONS **110** CITATIONS
SEE PROFILE

Darmen Kariboz
Suleyman Demirel University
9 PUBLICATIONS **10** CITATIONS
SEE PROFILE

Meraryslan Meraliyev
Suleyman Demirel University
9 PUBLICATIONS **47** CITATIONS
SEE PROFILE

Face extraction and recognition from public images using HIPI

Andrey Bogdanchikov Darmen Kariboz Meraryslan Meraliyev

Department of Computer Sciences
Suleyman Demirel University
Kaskelen Kazakhstan
{andrey.bogdanchikov, darmen.kariboz, meraryslan.meraliyev}@sdu.edu.kz

***Abstract*—Social networking services with public data are widely used nowadays. Billions of images uploaded to the internet each day over the world. This paper proposes the idea of a system which is currently being developed. The system collects images from public sources by some specific criteria, applies face detection and recognition algorithms on collected images, and provides the results in readable form. The Apache Hadoop library is used to increase performance of the system, and images are downloaded by the help of HIPI library. For testing purposes Labeled Faces in the Wild benchmark is used as a database for images containing over 13233 images of 5749 identities. Each image is jpg format in 250*250 resolutions. Results were more than good after testing the system with this database for face detection and recognition.**

Keywords—image collection, face detection, face recognition, Hadoop, HIPI, Parallel image processing

I. INTRODUCTION

Social networking services are widely used nowadays, and most of the information there are public. So, this paper is about the system we are currently developing, that collects a set of public images and videos to provide statistical analysis by applying face recognition algorithm on it. On ordinary computers all these processes take some time for a lot of images, so in order to increase the performance Hadoop distributed systems used [1]. Hadoop is open source software for reliable, scalable, distributed computing for large data. Each component of the distributed system works on its own part of the set of images and videos and then gives results for its part. Then all of the results from each component are gathered together and combined into one. A result can be different depending on what kind of information you need, like how many different identities participated in a certain event, or detect known criminals.

The system consists of four main parts: image/video collection by given criteria, face detection, face recognition and presenting results in human readable form. The architecture of the complete system is shown in Fig. 1.

There are several face recognition systems that show good results. Tadas Baltrusaitis et al. [2] presented OpenFace 2.0, an open source tool for researchers interested in facial behavior analysis. They define facial behavior consisting of: facial landmark location, head pose, eye gaze, and facial expressions. Even if they do not use deep learning, their results are pretty competitive against those who do. Yaniv Taigman et al. [3] presented DeepFace, the system based on deep learning that use 3D face modeling to derive a face representation from a nine-layer deep learning network.

Similarly with other face recognition systems, we use deep learning to detect and recognize faces. Currently, the system uses face detection and recognition algorithm based on dlib of Davis E. King [4]. This system based on dlib has an accuracy of 99.38% on the Labeled Faces in the Wild benchmark (LZW) [5].

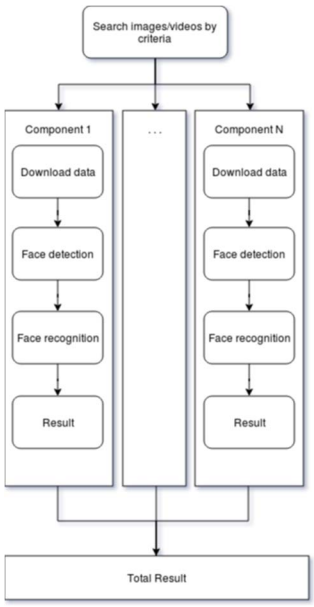

Fig. 1. The architecture of the complete system.

The rest of the paper is organized as follows: In Section II, a brief explanation of the system architecture. In Section III, experiments and results we obtained. In Section IV, conclusion and future work are described.

II. SYSTEM ARCHITECTURE

Fig. 1 shows architecture of the system. First of all, images from public sources must be collected by providing certain criteria. As a public source for data gathering, Instagram - a free social networking application for photo and video sharing, is used. As criteria hashtags are used to get list of images as a link, then distribute this list across several components. Each component works on its own part of the list, download some part of images from the link, and then perform face detection and recognition algorithms on this part. To download the images, the system uses HIPI [7]. HIPI is a framework that provides efficient and high-throughput image processing for parallel programs generally executed on a cluster. We convert all the images into a HipiImageBundle, shortly HIB. HIB is a collection of images represented as a single file on the HDFS. In this way, HIPI increases the speed of the system since working with single big file is much faster than working with a lot of small files. It also has several useful tools like filtering images without fully decoding it, thus saving some processing time.

The next step is face detection. In this step, face detection to the image is performed, and if there is any, we keep the image, otherwise we discard it. The output for the face detection is four numbers: top, right, bottom and left, location of face in pixels. Using this information, face can be cropped from the initial image if needed, like shown in Fig. 2.

Fig. 2. Face detection and cropping faces from image. (Image is taken from https://www.pexels.com/)

In addition to the whole set of images, the system has the set of known faces. After detecting faces, recognition step is done. Each face is compared with known faces. If the face is new i.e. the system didn't recognize it, it will be added to the set of known faces. In this way, we can be provided with the number of different identities. Fig. 3 illustrates face recognition part for known face.

When all the components complete their job, all results are gathered together to make up the overall result. For the current moment, result is a list of images followed by face coordinates and the name of a person, if this person is in the set of known people.

Fig. 3. Face recognition of known face in unknown image.

III. Experiments And Results

To test image collection part, the system was given some hashtags from Instagram. System parses all links to images with given hashtag. These links are divided into equal parts for each node in the Hadoop cluster. Each node downloads images from internet using predefined links. If the image is downloaded successfully, it is appended to the HIB.

For example:

- hashtag "jackiechanface" - found 250 images and videos, downloaded 193 samples
- hashtag "donnieyenfan" - found 151 images and videos, downloaded 124 samples
- hashtag "avengerstreet" - found 1180 images and videos, downloaded 1446 samples

In general, approximately 80% of the data is downloaded, other 20% cannot be downloaded because some links are corrupted.

After collecting data and making HIB from images, they are tested if they contain any face. Each image goes through face detection step, and if no face is detected, image is discarded, and kept otherwise. For face detection step, the system was tested with "Labeled Faces in the Wild" database. In this database each image has a person on it, i.e. none of the image is without face.

- In total 13233 images were processed
- In 13177 images 13856 faces were detected
- In 56 images no faces were detected

Because some images contain multiple faces, the number of detected faces is more than the number images. From these results, system has at least 0.42% inaccuracy for face detection. It can be higher since some images could contain several faces, but system found only one of them.

After all the images are filtered, face recognition algorithm is applied. Face recognition has "tolerance" parameter scaling from 0 to 1. This value represents distance of similarity between two faces, e.g. if the value equal to 0.45, it means the similarity of faces is 55%. Tolerance value and system accuracy for face recognition are reversely proportional, i.e. low tolerance – high accuracy, high tolerance – low accuracy. With the tolerance value higher than 0.7, the odds that the system will recognize two different people in one face is much higher.

To determine the optimal tolerance value, all the distances between faces in images and known faces were taken. As you can see from Fig. 4, the optimal value is between 0.3 and 0.5. In most cases similarity is between 50-70 percent.

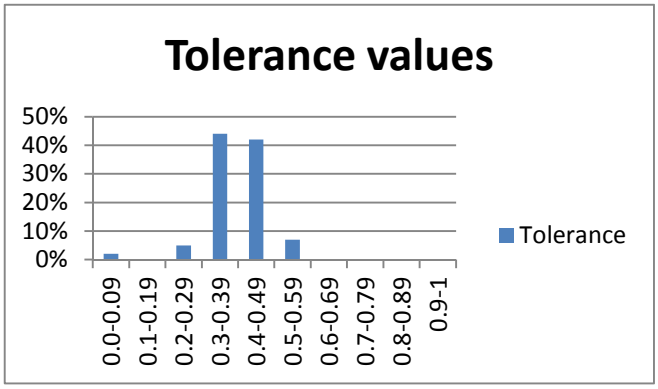

Fig. 4. Percentage of tolerance values for LZW

The system was given two sets of images, let's day set A and set B. Set A contains one image for one person. Set B is filled with images of people from set A; on this set face recognition is applied. As a result, in all images in set B, person should be recognized. Testing stage was given three tolerance values: 0.6, 0.5, and 0.4. Results are:

- For value 0.6, percentage of recognized faces is 94%
- For value 0.5, percentage of recognized faces is 84%
- For value 0.4, percentage of recognized faces is 56%

IV. Conclusion And Future Work

In this work, not complete system was presented, and results looks good so far. Data collection part shows 80 percent success, these results are good since some data in the internet is corrupted or links may be wrong. Face detection part has very good results detection faces in 99% cases in

LZW database. Face recognition part's results are worse than detection part's, showing success between 56 and 94 percent depending on tolerance value.

In the future, we are planning to improve face recognition model by applying incremental SVG algorithm which can update classification model during execution as Lufan Li et al. [7] has proposed. Also face recognition for videos has to be added. To do so, frames from videos must be extracted and processed in such way, that the system skips the frame if it very similar to the previous one. And all of these has to be done in distributed systems.

REFERENCES

[1] Chuck Lam. Hadoop in action. Manning Publications Co. 2011.

[2] Tadas Baltrušaitis, Amir Zadeh, Yao Chong Lim and Louis-Philippe Morency. OpenFace 2.0: Facial Behavior Analysis Toolkit. 2018 13th IEEE International Conference on Automatic Face & Gesture Recognition. Xi'an, China, 15-19 May 2018.

[3] Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato and Lior Wolf. DeepFace: Closing the Gap to Human-Level Performance in Face Verification. 2014 IEEE Conference on Computer Vision and Pattern Recognition. Columbus, OH, USA, 23-28 June 2014.

[4] Davis E. King. Dlib-ml: A Machine Learning Toolkit. Journal of Machine Learning Research 10, pp. 1755-1758, 2009.

[5] Gary B. Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. University of Massachusetts, Amherst, Technical Report 07-49, October, 2007.

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

[7] Lufan Li, Zhang Jun, Jiawei Fei and Shuohao Li. An Incremental Face Recognition System Based on Deep Learning. 2017 Fifteenth IAPR International Conference on Machine Vision Applications (MVA). Nagoya University, Nagoya, Japan, 8-12 May 2017.