iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 6 **Issue:** IV **Month of publication:** April 2018

DOI: http://doi.org/10.22214/ijraset.2018.4465

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 6.887
Volume 6 Issue IV, April 2018- Available at www.ijraset.com

Hadoop Based Face Detection and Recognition

Tushar Moraye[1], Rushabh Nisher[2], Aniket Patil[3], Himanshu Warekar[4], Niti Desai[5]
[1, 2, 3, 4, 5] *Computer Engineering, Rajiv Gandhi Institute of Technology, Mumbai University*

Abstract: An application for face detection and recognition from surveillance cameras in public or commercial places is discussed in this paper. Designed to cross reference a particular face with that from the database. The System is designed to work with web cameras for the face detection and recognition system, which is based on open source platforms like OpenCV and Apache Hadoop and Spark and can be easily scaled further for commercial applications as the major chunk of processing is offloaded onto server. This system uses algorithms like HaarCascades and Local Binary Pattern Histogram (LBPH) for detection and recognition respectively.

Keywords: Face Detection; OpenCV; Apache Hadoop; Apache Spark; HaarCascades; Local Binary Pattern Histogram.

I. INTRODUCTION

Face detection and recognition is a process, which is used for the analysis of the input frame or continuous sequence of frames and to determine the number of detected faces and to recognize the detected faces. Face detection is the base for face recognition, whose results directly affect the process and accuracy of face recognition. Face recognition is an easy task for humans but it's entirely different task for a computer.

A very little is known about human recognition to date on how we analyse an image and how does the brain encode it. Face recognition is all about extracting those meaningful features from an image, putting them into a useful representation and performing classifications on them.

Face recognition based on the geometric features of a face is probably the most intuitive approach to Human identification. The whole process can be divided in three major steps:

Find a good database of faces with multiple images for each individual or to gradually develop a large database,

Use faces in the database and train the face recognizer algorithm,

Use frames to detect faces and recognize faces it was trained for.

A. Purpose

Face detection and recognition are often used in biometrics. It can also be used for video surveillance, human computer interface and image database management or measurement of another signal. The proposed methodology is basically a connection of two stages – face detection; which is done by Haar based Cascade classifier and face recognition using Local Binary Pattern Histogram (LBPH) algorithm. Various face detection and recognition methods have been used in the past. This paper also provides a solution for image detection and recognition in video surveillance. Implementation of face recognition is done using local binary pattern histogram analysis using eigen vectors. A system that uses different eigen vectors for each image will perform better than a system that only uses one.

B. Scope

The scope of the project is the system on which the software is installed, i.e. the project is developed as third-party general-purpose software, and it will work for particular institutes which can efficiently generate optimal solutions.

II. SYSTEM ANALYSIS

A. Existing System

Face localization is the first and foremost step in face recognition. Its purpose is to identify and localize the face. Face detection technology is imperative in order to support applications such as automatic lip reading, facial expression recognition and face recognition. The framework for both face detection and recognition is almost similar.

The framework mainly consists of two segments i.e. Face detector and a Face Recognizer. The Face detector searches for human faces from the image and localizes the faces from the background. After the detection or localization of a face takes place, the process of recognition will begin and it will try and determine the persons whose photo is taken by the face recognizer. Both face detector and recognizer, have a feature extractor and a pattern recognizer which have a task of transforming the pixels of the images into vector representation.

©IJRASET (UGC Approved Journal): All Rights are Reserved

B. Disadvantages of Existing System

1) *Processing and Storage:* High-definition video is very low in resolution with respect to camera images, despite of that it still occupies a significant amount of disk space. Processing of every frame of the video is a very big undertaking, so generally only a part (10-25%) is actually processed through a recognition system. Generally, to minimize total processing time, agencies make use of clusters of computers. But, adding computers involves considerable data transfer over a network, which can be bound by input-output restrictions, further limiting processing speed.
2) *Image Size:* In the event, when a face-detection algorithm finds a face in an image or a still from a video capture, the relative size of that face compared with the enrolled image size greatly affects how well the face will be recognized by the face recognition algorithm. An image, which is already small in size, coupled with the fact that a target is distant from the camera, means that the detected face is only 100 to 200 pixels on a side, which is very less. Further, having to scan an image for varying face sizes is a processor-intensive activity i.e. it takes up a lot of load on the processor.
3) *Face Angle:* The angle at which the target's face is captured highly influences the face recognition algorithm. Usually, when a face is added to the database of recognition software, multiple angles are used and captured (profile, frontal and 45-degree are common). Anything less than a frontal view highly affects the algorithm's capability to generate a template for the face.

III. PROPOSED SYSTEM

A. Characteristics of Proposed System

1) *Processing and Storage:* As the processing of data would take place on a central server, the processing can be increased by upgrading the server hardware overtime. To minimize total processing time, agencies can use clusters of computers as well. However, sending data to central server involves considerable data transfer over a network, which can be bound by input-output restrictions, further limiting processing speed which can be solved by using a better compression algorithm.
2) *Real Time Processing:* The processing of data would take place on a real time input stream that is sent from client pc to the server for recognition thus reducing the wait time for facial recognition. This would enable an organization to track people and get information about them instantly.
3) *Platform independent:* The proposed system is platform independent as it uses python/ java code, which can be run on any platform or device, thus making the application widely available and easy to access. This also means that the proposed system can have varied applications as it can be easily used on any platform.

IV. ARCHITECTURE DIAGRAM

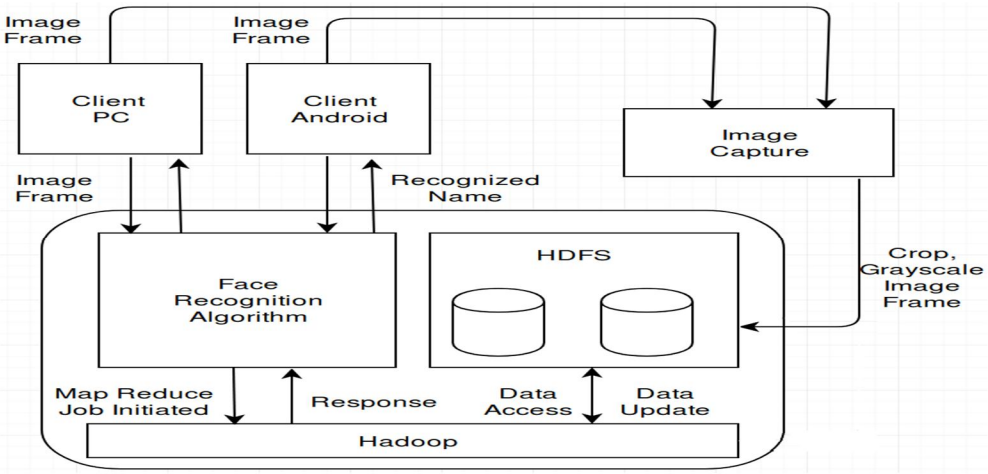


Fig. 1 Working of Proposed System

V. METHODOLOGY

A. LBPH Face Recognition Algorithm

Local Binary Pattern Histogram Algorithm is used because of its ability to update it with new faces easily then Eigenfaces and Fisherfaces.

Steps that are followed are as follows

©IJRASET (UGC Approved Journal): All Rights are Reserved

1) *Adding Person:* Whenever a new face is to be added to the algorithm, the face is captured and cropped and added to dataset and then simply using the update () function, the algorithm can be updated and trained for the new face.
2) *Train the algorithm:* Train the algorithm using the dataset collected overtime and save it as a yml file which would reduce the computations.
3) *Face Recognition:* To detect the face, we use HaarCascades frontal face detection and then detect eyes in the detected face using HaarCascades eye detection. Calculate the angle at which the face is tilted and then rotate the face to make align the eyes in a straight line. Equalize the histogram of the image and then predict the detected face using the LBPH algorithm.

B. Hadoop Processing

System is made up of two parts i.e. client side and server side. Components on client side are GUI or android application which will be used to capture images on the client. These images are the sent to the server. Components of server are Face recognition software: LBPH, Hadoop map-reducer and HDFS file system. The server side also consist of trained yml file in HDFS. this yml is later checked for recognition by LBPH. Now the image captured at client side will be forwarded to the server side. After which the map-reduce job will be initiated at server for recognition. After recognising the image, the details of image stored in database will be sent to client as a response.

VI. SOFTWARE ENVIRONMENT

A. Front End

1) The webcam captures an image or a stream of images which then the PC client uploads on the server. After the processing on server is done, the server will reply with a response. The response will be the details of the face. The PC App will display this response on the webcam feed itself.
2) In order to capture an image we use an android application. After the image is captured it is uploaded on the server. After the processing on server is done, the server will reply with a response. The response will be the details of the captured face image. The Android App will display this response on a new screen.

B. Back End

1) When the image or stream of images are uploaded on the server, a Python script will be called. This script will initiate the MapReduce job on Hadoop Cluster. After this processing, the image id of the captured image will be stored in the output directory on HDFS. This id is queried on MySQL and the result is sent to the PC Application.
2) When the image is uploaded on the server, a Python script will be called. This script will initiate the MapReduce job on Hadoop Cluster. After this processing, the image id of the captured image will be stored in the output directory on HDFS. This id is queried on MySQL and the result is sent to the Android Application.

VII. CONCLUSION

Prototype system for face detection and recognition using Hadoop is proposed. This paper shows the use of Image processing, by using openCV and Local Binary Pattern Histogram algorithm in real time is possible as the processing is offloaded to the main server which is also assisted by the yml generated by the algorithm.

1) *Future Work:* Live Face Recognition can be used with a combination of algorithms to decrease the chances of false positives.

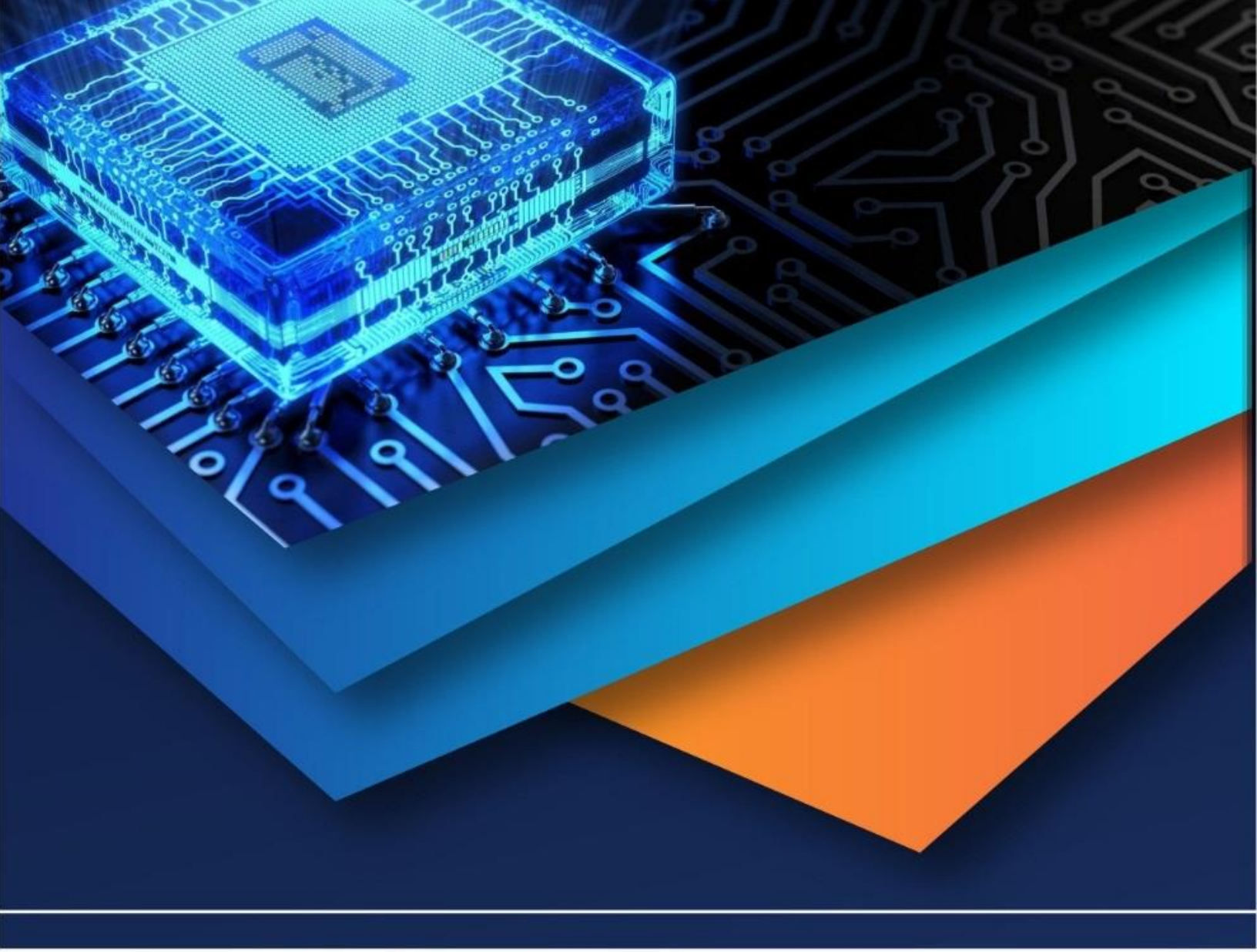
VIII. ACKNOWLEDGMENT

This work was supported by our project guide Prof. Niti Desai and also our sincere thanks to Rajiv Gandhi Institute of Technology.

REFERENCES

[1] Berlinska, J.; M. Drozdowski. (2011); Scheduling Divisible MapReduce Computations, Journal of Parallel and Distributed Computing, 71(3): 450-459.
[2] Dean, J.; S. Ghemawat. (2010); MapReduce: A Flexible Data Processing Tool, Communications of the ACM, 53(1): 72-77.
[3] Dean, J.; S. Ghemawat. (2008); MapReduce: Simplified Data Processing on Large Clusters,
[4] http://hadoop.apache.org/.
[5] http://opencv.org
[6] http://wiki.apache.org/hadoop/SequenceFile
[7] https:github.com/uvagfx/hipi
[8] hipi.cs.virginia.edu/
[9] www.edureka.com

©IJRASET (UGC Approved Journal): All Rights are Reserved



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)