

Implementation of Mobile-Cloudlet-Cloud Architecture for Face Recognition in Cloud Computing using Android Mobile

Priya Bhatnagar
R.C.E.W, Jaipur
Rajasthan, India

Abstract: Real time face recognition applications use face recognition algorithms, which extract information such as shape, size, and position of facial features (eyes, nose, ear, lips) and then compare this extracted feature to stored featured database to locate matching images. Face recognition application used in airport security, in passport office, in R.T.O office, in government organization, in development and research department. In this paper we implement a application which recognize a face and minimize computation time using our mobile-cloudlet-cloud architecture named MOCHA. And also develop an algorithm for percent variance. Our experimental result demonstrate that we can communicate anywhere from the world by using jelastic cloud. We will run application on mobile devices using jelastic cloud as the backend server.

Keywords-Face detection, Face recognition, Jelastic cloud, Cloudlet, mobile devices

1. INTRODUCTION

Face recognition applications that automatically identify an individual from captured images or videos are everywhere, for applications such as airport, passport office, R.T.O office, government organization, and research center. Face recognition algorithms analyze images, extract information such as the shape, size and position of the facial features (e.g., eyes, nose, mouth), and then use these extracted features to search a facial database to locate matching images. The rapid adoption of smart phones, net books and tablets in the consumer market, multimedia content is becoming the dominant form of information. These devices are becoming increasingly more powerful, with more processing power, storage, and sensing capabilities. In addition, it is now possible to rent computing, storage, and network resources as needed via cloud computing, in which data is processed and stored remotely at large-scale compute and data centers [1], [2].

This paper describes our work with the design and implementation of the face recognition on the MOCHA (Mobile Cloud Hybrid Architecture) cloud computing platform, which provides a mobile-cloudlet-cloud architecture. And we are using Jelastic cloud server and deploy application so that we can communicate anywhere from the world. And the second goal of this system that we are using GPS coding for checking the communication network and time monitoring and GPS class is monitoring time scheduling for one request and if it cross more than 1sec then send communication error. We have written our own logic where communication happens within 0.2sec approx for one image. And the third goal of this system that We developed our own algorithm which compares the percent matching faces with respect to given face name in the database. It displays % matching face with corresponding face and database id also.

2. LITERATURE SURVEY

With the ubiquity of new information technology and media, more effective and friendly methods for human computer interaction (HCI) are being developed which do not rely on traditional devices such as keyboards, mice, and displays.

Furthermore, the ever decreasing price/ performance ratio of computing coupled with recent decreases in video image acquisition cost imply that computer vision systems can be deployed in desktop and embedded systems. The rapidly expanding research in face processing is based on the premise that information about a user's identity, state, and intent can be extracted from images, and that computers can then react accordingly, e.g., by observing a person's facial expression. In the last five years, face and facial expression recognition have attracted much attention though they have been studied for more than 20 years by psychophysicists, neuroscientists, and engineers. Many research demonstrations and commercial applications have been developed from these efforts. A first step of any face processing system is detecting the locations in images where faces are present. However, face detection from a single image is a challenging task because of variability in scale, location, orientation (up-right, rotated), and pose (frontal, profile). Facial expression, occlusion, and lighting conditions also change the overall appearance of faces. Facial expression recognition concerns identifying the affective states (happy, sad, disgusted, etc.) of humans [3]. Evidently, face detection is the first step in any automated system which solves the above problems. It is worth mentioning that many papers use the term "face detection," but the methods and the experimental results only show that a single face is localized in an input image. In this paper, we differentiate face detection from face localization since the latter is a simplified problem of the former. Meanwhile, we focus on face detection methods rather than tracking methods. While numerous methods have been proposed to detect faces in a single image of intensity or color images, we are unaware of any surveys on this particular topic. A survey of early face recognition methods before 1991 was written by Samal and Iyengar . Chellapa et al. wrote a more recent survey on face recognition and some detection methods. Among the face detection methods, the ones based on learning algorithms have attracted much attention recently and have demonstrated excellent results. Since these data driven methods rely heavily on the training sets, we also discuss several databases suitable for this task. A related and important problem is how to evaluate the performance of the proposed detection methods. Many recent face detection papers compare the performance of

several methods, usually in terms of detection and false alarm rates. It is also worth noticing that many metrics have been adopted to evaluate algorithms, such as learning time, execution time, the number of samples required in training, and the ratio between detection rates and false alarms. Evaluation becomes more difficult when researchers use different definitions for detection and false alarm rates. In this paper, detection rate is defined as the ratio between the number of faces correctly detected and the number faces determined by a human. An image region identified as a face by a classifier is considered to be correctly detected if the image region covers more than a certain percentage of a face in the image. In general, detectors can make two types of errors: false negatives in which faces are missed resulting in low detection rates and false positives in which an image region is declared to be face, but it is not. A fair evaluation should take these factors into consideration since one can tune the parameters of one's method to increase the detection rates while also increasing the number of false detections. In this paper, we discuss the benchmarking data sets and the related issues in a fair evaluation. With over 150 reported approaches to face detection, the research in face detection has broader implications for computer vision research on object recognition [5]. Nearly all model-based or appearance-based approaches to 3D object recognition have been limited to rigid objects while attempting to robustly perform identification over a broad range of camera locations and illumination conditions. Face detection can be viewed as a two-class recognition problem in which an image region is classified as being a "face" or "nonface."

Existing system

There exist solutions that provide face recognition in wireless communication using mobile-cloudlet-cloud architecture on cloud. In [1] Tologa Soyata, Rajani muraleedharan and Colin Funai, present a face recognition application on cloud (Amazon web server) using mobile-cloudlet-cloud architecture. And present a algorithm which minimize overall response time. And used SPREAD (Secure protocol for reliable data delivery) scheme to statistically enhance the data confidentiality service in an ad hoc network. SPREAD is based on secret sharing and multi-path routing. Which have some disadvantage like

- a) Mobile nodes might be compromised themselves (e.g., nodes be captured in a battle field scenario) and subsequently be used to intercept secret information relayed by them.
- b) Data security applied on whole data which can be trapped in between and decrypted the entire data.
- c) Data security percent is less.

There are no existing model which utilizes all the three components (mobile, cloudlet, jelastic cloud) working together with specific algorithms, applications and initial result.

3. PROPOSED SYSTEM

In this paper we present a application for face recognition and percent variance in wireless communication using mobile-cloudlet-cloud architecture on jelastic cloud.

With the help of this application user can communicate anywhere from the world.

3.1 Related Work

Android development SDK provides API libraries and developer tools which are necessary to build, test, and debug apps for android. For a new developer, recommends to download the ADT Bundle to quickly start developing apps. It

includes the Android SDK mechanism and a version of the Eclipse IDE with built-in ADT(Android Developer Tools).

- Eclipse+ADT plug-in
- Android SDK tools
- Android platform-tools
- The latest Android platform

3.2 System architecture

The mobile-cloudlet-cloud architecture broken into three subsystem: 1) mobile devices 2) cloudlet 3) Jelastic cloud

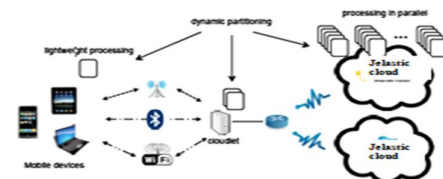


Fig. 1. The MOCHA Architecture: mobile devices interact with the cloudlet and the cloud via multiple connections and use dynamic partitioning to achieve their QoS goals (e.g., latency, cost).

Figure [1] The MOCHA architecture[1]

3.2.1 Mobile device

In our work we use mobile devices, such as Smartphone's. The main task of our mobile device is to capture image and send them to the cloudlet in raw form for processing. After pre processing captured image sent to the cloud to perform real time template matching over a large database located in the cloud.

3.2.2 Cloudlet

The cloudlet is a special-purpose inexpensive compute box with the capability of massively parallel processing MPP. In our approach we can use till 16 cloudlet [4].



	Cloudlet	Cloud
State	Only soft state	Hard and soft state
Management	Self-managed; little to no professional attention	Professionally administered, 24x7 operator
Environment	"Datacenter in a box" at business premises	Machine room with power conditioning and cooling
Ownership	Decentralized ownership by local business	Centralized ownership by Amazon, Yahoo!, etc.
Network	LAN latency/bandwidth	Internet latency/bandwidth
Sharing	Few users at a time	100s-1000s of users at a time

(a) Cloudlet Concept
 (b) Key Differences: Cloudlet vs. Cloud
 Figure : What is a Cloudlet?

Figure 2. The cloudlet concept

3.2.3 Cloud

Cloud computing provides computing and storage resources in a pay-as-you-go manner. In our architecture, a client program which is running on the cloudlet sends a request to the servers on Jelastic cloud where the actual program runs on virtual instances in parallel and the results are sent back to the mobile device or cloudlet.

3.3 System requirement

3.3.1 Hardware Requirement

- Internet connection required in mobile and laptop

- Android 2.3/4.0 based mobile

3.3.2 Software Requirement

- Jdk1.6
- Android sdk4.0
- Web server : Tomcat 6.0
- IDE : Net Beans 7.0.1

3.4 Face recognition using MOCHA architecture

Face recognition application in wireless communication using mobile-cloudlet-cloud architecture divided in two stages: First is **face detection**: In which we use Viola-Jones and Adaboost algorithm for face detection[1] and percent variance we try all possible rectangle locations, and classify if rectangle contains a face (and only the face) and Select Features $i=1..n$, Learn thresholds θ_i , Learn weights α_i . Features can be used to represent both the statistically close facial information and sparsely related background data in a sample image. In order to be successful a face detection algorithm must possess two key features, accuracy and speed. There is generally a trade-off between the two. Through the use of a new image representation, termed integral images, Viola and Jones describe a means for fast feature evaluation, and this proves to be an effective means to speed up the classification task of the system. The integral image computes a value at each pixel (x,y) that is the sum of the pixel values above and to the left of (x,y) , inclusive. This can quickly be computed in one pass through the image. Let A, B, C, D be the values of the integral image at the corners of a rectangle. Then the sum of original image values within the rectangle can be computed:

$$\text{Sum} = A - B - C + D$$

Only 3 additions are required for any size of rectangle. This is now used in many areas of computer vision.

Adaboost algorithm takes a feature set and a training set of positive and negative images, any number of machine learning approaches could be used to learn a classification function. AdaBoost learning algorithm is used to boost the classification performance of a simple learning algorithm (e.g., it might be used to boost the performance of a simple perceptron). It does this by combining a collection of weak classification functions to form a stronger classifier. In the language of boosting the simple learning algorithm is called a weak learner.

Second stage is **face recognition** where we use Eigen face approach Face Images are projected into a feature space ("Face Space") that best encodes the variation among known face images. The face space is defined by the "eigenfaces", which are the eigenvectors of the set of faces. Acquire the training set and calculate eigenfaces (using PCA projections) which define eigenspace. In eigenface approach we done this steps

- When a new face is encountered, calculate its weight.
- Determine if the image is face.
- If yes, classify the weight pattern as known or unknown.
- (Learning) If the same unknown face is seen several times incorporate it into known faces.

Applications

- AirPort
- Passport Office
- RTO Office

- Government Organization
- Research & Development Center

3.5 Flow diagrams

3.5.1 Face detection/recognition flow diagram

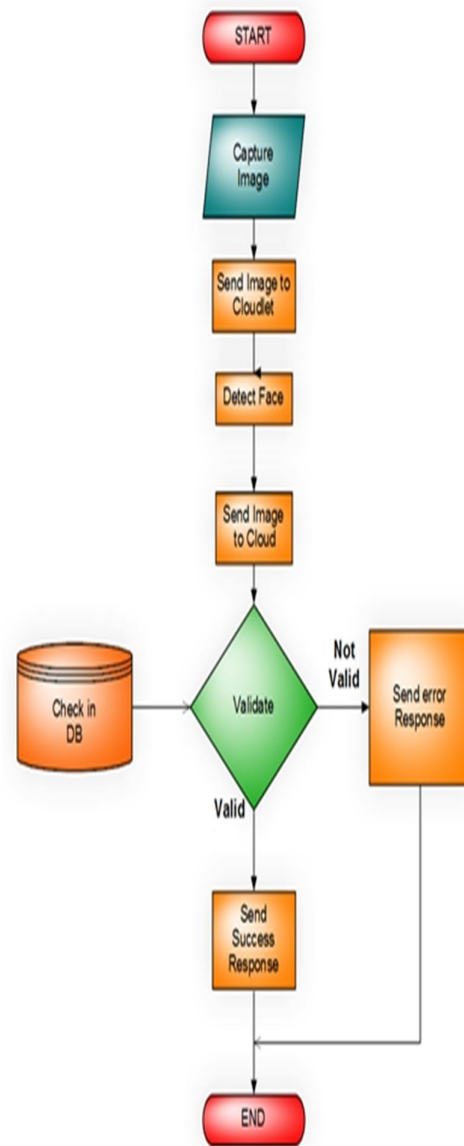


Figure 3. face detection/recognition flow diagram

3.5.2 Flow diagram for cloudlet

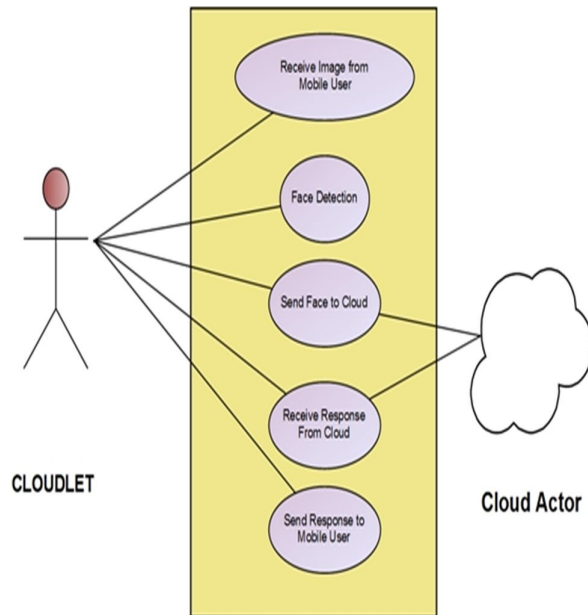


Figure 4. Flow diagram for cloudlet

3.5.3 Flow diagram for cloud

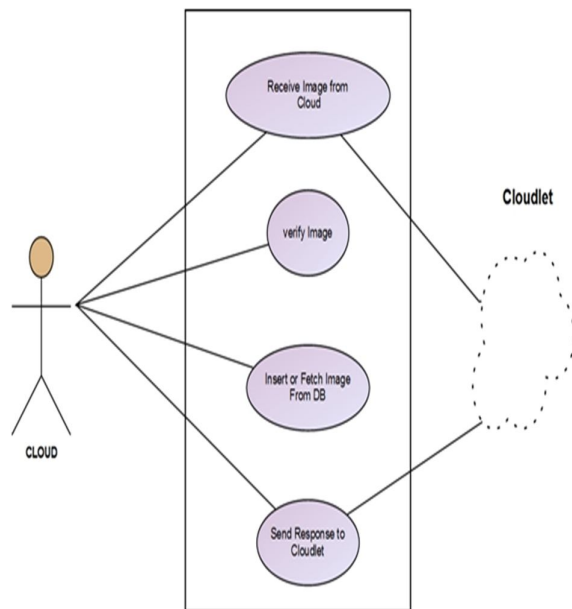


Figure 5. Flow diagram for cloud

This table presents a test case to check the integration of Cloud Vision with Face Recognition in cloud using android phone.

Table 1. Test case to check the integration of cloud vision

Step number	Description	Input	Expected output
1.	Select setting in android application		It should establish the connection with cloud server else display error in communication
2	Enter valid user id and password and click on submit button		Main page should be displayed with different face detection and identification
3	Select face detection and user id		It should identify the face on cloud server and display verify or not
4	Enter valid user id and password and click on submit button		Main page should be displayed with different face detection and identification
5	Select face detection		It interact with cloudlet and detect that the face is existing or not
6	Select face detection		Error in communication if no proper network
7	Select face identification		Error in communication if no proper network
8	Select face identification		It should identify the faces in cloud server and display verify or not

3.6 Testing

During software development testing is the major quality control measure employed its basic function is to detect the errors in the software. For this, different levels of testing are used, which performs different tasks on the aim to test different aspects of the system.

This below table present test functionality of Cloud Server Connection

Table 1. Test case functionality of cloud server

STEP NUMBER	DESCRIPTION	INPUT	EXPECTED OUTPUT
1.	Deploy the application to Jelastic Cloud		it should verify the cloud environment else display the error
2.	Deploy the application to Jelastic Cloud		Cloud server verify the jelastic environment as SAAS and process the request
3.	Deploy the application to Jelastic Cloud		it should verify to port number on which application will be running and deploy the application to server
4.	Select project run as server		Application should start run if all the connection is proper on server

5. FUTURE GOALS

In future more sophisticated synchronization algorithms which permitting cloud to cloud communication has to be extended rather than multiple cloudlet-cloud communication links. And also develop application for 3D face recognition on MOCHA architecture.

6. CONCLUSION

In this paper we presented an approach for face detection which minimizes computation time while achieving high detection accuracy. The approach was used to construct a face detection system which is approximately 15 faster than any previous approach. Preliminary experiments, which will be described elsewhere, show that highly efficient detectors for other objects, such as pedestrians, can also be constructed in this way. New algorithms, representations, and insights were presented which are quite generic and may have broader application in computer vision and image processing. and also

present an algorithm for percent variance. We have designed MOCHA to integrate mobile devices, the cloudlet, and using jelastic cloud for better development. In our approach GPS class is monitoring time scheduling for one request and if it cross more than 1sec then send communication error. We have written our own logic where communication happens within 0.2sec approx for one image.

7. REFERENCES

- [1] Ming-Hsuan Yang, David Kriegman, and Narendra Ahuja, "Detecting faces in images : A survey," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 1, pp. 34–58, 2002.
- [2] Paula Viola and Michael J. Jones, "Robust real time face detection," International Journal of Computer Vision, vol. 57, no. 2, pp. 137–154, 2004.
- [3] Matthew Turk and Alex Pentland, "Eigenfaces for recognition," Journal of Cognitive Neuroscience, vol. 3, no. 1, pp. 71–86, 1991.
- [4] Paula Viola and Michael J. Jones, "Robust real time face detection," in Second International Workshop on Statistical and Computational Theories of Vision - Modeling, Learning, Computing, and Sampling, July 2001, pp. 1–25.
- [5] Tolga Soyata, Eby G. Friedman, and J. H. Mulligan, "Integration of Clock skew and register delays into a retiming algorithm," in Proceedings of the International Symposium on Circuits and Systems, May 1993, pp.1483–1486.
- [6] Tolga Soyata and Eby G. Friedman, "Retiming with non-zero clock skew, variable register and interconnect delay," in Proceedings of the IEEE Conference on Computer-Aided Design, Nov 1994, pp. 234–240.
- [7] B. Chun and P. Maniatis, "Dynamically Partitioning Applications between Weak Devices and Clouds," in Proceedings of ACM Mobile Cloud Computing and Services (MCS), June 2010.
- [8] P. Stuedi, I. Mohamed, and D. Terry, "Where Store: Location-based Data Storage for Mobile Devices Interacting with the Cloud," in Proceedings of ACM Mobile Cloud Computing and Services (MCS), June 2010.
- [9] A. Miettinen and J. Nurminen, "Energy Efficiency of Mobile Clients in Cloud Computing," in Proceedings of Usenix Hot Cloud, June 2010.
- [10] E. Jung, Y. Wang, L. Prilepov, F. Maker, X. Liu, and V. Akella, "User-Profile-Driven Collaborative Bandwidth Sharing on Mobile Phones," in Proceedings of ACM Mobile Cloud Computing and Services (MCS), June 2010.
- [11] A. Shieh, S. Kandula, A. Greenberg, and C. Kim, "Seawall: Performance