Face Recognition Using Hull Point Analysis with Qualcomm SDK Sanpdragon Processor

D.Rajapriya
Assistant Professor, Department of CSE, RVS Technical Campus – Coimbatore.

Sandhya
UG Scholar, Department of CSE, RVS Technical Campus – Coimbatore.

Pradeepa
UG Scholar, Department of CSE, RVS Technical Campus – Coimbatore.

Subadurga
UG Scholar, Department of CSE, RVS Technical Campus – Coimbatore.

Abstract – Relational over the last two decades, the advances in computer vision and pattern recognition power have opened the door to new opportunity of automatic facial expression recognition system. Face recognition is challenging due to wide variety of faces and complexity of noises and image backgrounds. Facial recognition is done through Hull Point Analysis for accuracy. Hull Point Analysis is a method of plotting six points in shape of hexagon with center point, then the points are connected with lines and the measurements are stored in database at the back end .When a stored user comes infront of the camera the analysis is done based on the measurements and validated with the existing line measurements that are stored in the database. The comparison is made and checked whether the image matches or not.

Index Terms – Face Recognition, Hull Point.

1. INTRODUCTION

1.1. FACE RECOGNITION IN IMAGE PROCESSING

The information age is quickly revolutionizing the way transactions are completed. Everyday actions are increasingly being handled electronically, instead of with pencil and paper or face to face. This growth on electronic transactions has resulted in a greater demand for fast and accurate user identification and authentication. Access codes for buildings, banks accounts and computer systems often use PIN's for identification and security clearances.

Using the proper PIN gain access, but the user of the PIN is not verified. When credit and ATM cards are lost and stolen , an unauthorized user can often come up with the correct personal codes. Despite warning, many people continue to choose easily guessed PIN's and passwords: birthdays, phone numbers and social security numbers. Recent cases of identity theft have heightened the needs for methods to prove that someone is truly who he/she claims to be.

Face recognition technology may solve this problem since a face is undeniably connected to its owner expect in the case of identical twins. It's nontransferable. The system can then compare scans to records stored in a central or local database or even on a smart card.

1.2 FACE RECOGNITION TECHNOLOGY

1.2.1 INTRODUCTION TO BIOMETRIC

A biometric is a unique, measurable characteristic of a human being that can be used to automatically recognize an individual or verify an individual's identity. Biometrics can measure both physiological and behavioral characteristics.

"Any automatically measurable, robust and distinctive physical characteristics or personal trait that can be used to identify an individual or verify the claimed identity of an individual.

This definition requires elaboration:-

Measurable means that the characteristic or trait can be easily presented to a sensor, located by it, and converted into a quantifiable, digital format. This measurability allows for matching to occur in a matter of seconds and makes it an automated process.

Biometric identifiers are the distinctive, measurable characteristics used to label and describe individuals. Biometric identifiers are often categorized as physiological versus behavioral characteristics. Physiological characteristics are related to the shape of the body. Examples include, but are not limited to fingerprint, palm veins, face

recognition, DNA, palm print, hand geometry, iris recognition, retina and odor/scent. Behavioral characteristics are related to the pattern of behavior of a person, including but not limited to typing rhythm, gait.

2. SYSTEM ANALYSIS

2.1 System Specification

2.1.1Qualcomm Snapdragon

Snapdragon is a suite of System-on-Chip (SoC) semiconductor products designed and marketed by Qualcomm for mobile devices. The Snapdragon central processing unit (CPU) uses the ARM RISC instruction set, and a single SoC may include multSiple CPU cores, a graphics processing unit (GPU), a wireless modem, and other software and hardware to support a smartphone's global positioning system(GPS), camera, gesture recognition and video. Snapdragon semiconductors are embedded in devices of various systems, including Google Android mobile and Windows Phone devices.

Benchmark tests of the Snapdragon 800's processor by *PC Magazine* found that its processing power was comparable to similar products from Nvidia. Benchmarks of the Snapdragon 805 found that the Adreno 420 GPU resulted in a 40 percent improvement in graphics processing over the Adreno 330 in the Snapdragon 800, though there was only slight differences in processor benchmarks.

Snapdragon processors enable next-level user experiences. Each one is a comprehensive all-in-one system, specifically designed to enable best-in-class mobile experiences with all-day battery life. Snapdragon processors also enable advanced connectivity, jaw-dropping graphics, and powerful and efficient processing and multitasking.

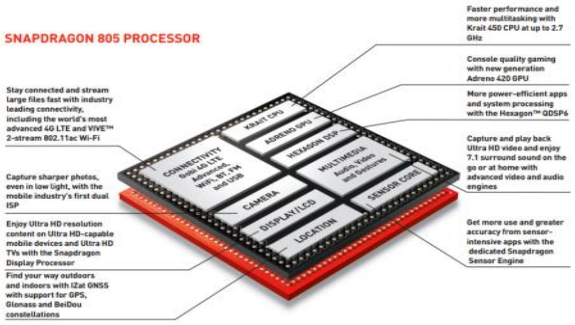


Fig 2.1 SnapDragon 805 Processor

2.1.2 Camera

A front-facing camera is a feature of cameras, mobile phones and similar mobile devices that allows taking a self-portrait photograph or video while looking at the display of the device, usually showing a live preview of the image.

A facial recognition system is a computer application capable of identifying or verifying a person from a digital image or a video frame from a video source. One of the ways to do this is by comparing selected facial features from the image and a facial database.It is typically used in security systems and can be compared to other biometrics such as fingerprint or eye iris recognition systems.

3. SYSTEM IMPLEMENTATION

3.1 Environment Development and Features

3.1.1 Android Studio

Android Studio is the official IDE for Android app development, based on IntelliJ IDEA. On top of Intellij's powerful code editor and developer tools, Android Studio offers even more features that enhance your productivity when building Android apps, such as:

- A flexible Gradle-based build system
- Build variants and multiple APK file generation
- Code templates to help you build common app features
- A rich layout editor with support for drag and drop theme editing
- Lint tools to catch performance, usability, version compatibility, and other problems
- Code shrinking with ProGuard and resource shrinking with Gradle
- Built-in support for Google Cloud Platform, making it easy to integrate Google Cloud Messaging and App Engine.

3.1.1.1 Developer Services

Android Studio supports enabling these developer services in your app:

- Ads using AdMob
- Analytics Google Analytics
- Authentication using Google Sign-in
- Notifications using Google Cloud Messaging

Enabling a developer service adds the required dependencies and, when applicable, also modifies the related configuration files. To activate the service, you must perform service-specific updates, such as loading an ad in the MainActivity class for ad display. To enable an Android developer service, select the File > Project Structure menu option and click a service under the Developer Services sub-menu. The service configuration page appears. In the service configuration page, click the service check box to enable the service and click OK. Android Studio updates your library

dependencies for the selected service and, for Analytics, updates the AndroidManifest.xml and other tracker configuration files.

3.1.1.2 Gradle

Android Studio uses Gradle as the foundation of the build system, with more Android-specific capabilities provided by theAndroid Plugin for Gradle. This build system runs as an integrated tool from the Android Studio menu and independently from the command line. You can use the features of the build system to:

Customize, configure, and extend the build process.

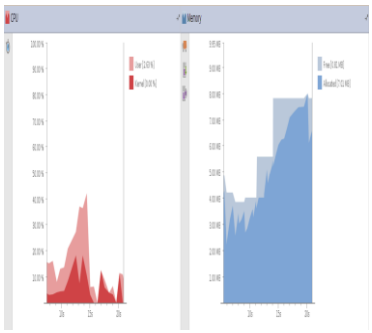
Create multiple APKs for your app with different features using the same project and modules.

Reuse code and resources across source sets.

The flexibility of Gradle enables you to achieve all of this without modifying your app's core source files.

3.1.1.3 Memory and CPU monitor

Android Studio provides a memory and CPU monitor view so you can more easily monitor your app's performance and memory usage to track CPU usage, find deallocated objects, locate memory leaks, and track the amount of memory the connected device is using. With your app running on a device or emulator, click the **Android** tab in the lower left corner of the runtime window to launch the Android runtime window. Click the **Memory** or **CPU** tab.



Memory and CPU Monitor Fig 3.1

3.1.1.4 Managing AVDs with AVD Manager

The AVD Manager is a tool you can use to create and manage Android virtual devices (AVDs), which define device configurations for the Android Emulator.

If you are creating a new AVD, you can specify the following hardware options for the AVD to emulate.

CHARACTERESTI	DESCRIPTION	PROPERTY
Device RAM size	Physical RAM on the device ,in megabytes. Default value is "96".	Hw.RAM Size
Touch-screen Support	Whether there is a touch scre or not on the device. Default value is "yes".	hw.touch Screen
Trackball support	Whether there kball on the device. Default value is "yes".	hw.trackBall
Keyboard Support	Whether the a QWERTY key ult value is "yes".	hw.keyboard
DPad support	Whether the device has DPad keys. Default value is "yes".	hw.dPad
GSM modem Support	Whether there is a GSM modem in thedevice. Default value is "ye	hw.gsmModem
Camera support	Whether the device has acamera. Default value is "no".	hw.camera
Maximum horizontal camera pixels	Default value is "640".	hw.camera. maxHorizontal Pixels
Maximum vertical camera pixels	Default value is "480".	hw.camera.max VerticalPixels
GPS support	Whether there is a GPS in the device. Default value is "yes".	hw.gps
Battery support	Whether the device can run on a b ult value is "yes".	hw.battery
Accelerometer	Whether there i lerometer in the device. Default value is "yes".	hw.accelerome
Audio recording support	Audio recording support	hw.audioInput
Audio playback support	Whether the device can play audio. Default value is	hw.audio Output

	"yes".	
SD Card support	Whether the orts insertion/ removal of virtual SD Cards. Default value is "yes".	hw.sdCard
Cache partition support	Whether we use a /cache partition on the device. Default value is "yes".	disk.cache Partition
Cache partition size	Default value is "66MB".	disk.cache Partition.size
Abstracted LCD density	Sets the gene ity characteristic used by the AVD's s ult value is "160".	hw.lcd.density

3.1.2 SQlite

SQLite is a software library that implements a self-contained, serverless, zero-configuration, transactionalSQL database engine. SQLite is the most widely deployed database engine in the world.

3.1.2.1 Features of SQLite

- Transactions are atomic, consistent, isolated, and durable (ACID) even after system crashes and power failures.
- Zero-configuration - no setup or administration needed.
- Full SQL implementation with advanced features like partial indexes and common table expressions. (Omitted features)
- A complete database is stored in a single cross-platform disk file. Great for use as an application file format.
- Supports terabyte-sized databases and gigabyte-sized strings and blobs. (See limits.html.)
- Small code footprint: less than 500KiB fully configured or much less with optional features omitted.
- Simple, easy to use API. Written in ANSI-C. TCL bindings included. Bindings for dozens of other languages available separately.
- Well-commented source code with 100% branch test coverage.
- Available as a single ANSI-C source-code file that is easy to compile and hence is easy to add into a larger project.
- Self-contained: no external dependencies.
- Cross-platform: Android, *BSD, iOS, Linux, Mac, Solaris, VxWorks, and Windows (Win32, WinCE, WinRT) are supported out of the box. Easy to port to other systems.
- Sources are in the public domain. Use for any purpose.

3.1.2.2 Uses for SQLite:

Database For The Internet Of Things. SQLite is popular choice for the

- Database For The Internet Of Things. SQLite is popular choice for the database engine in cellphones, PDAs, MP3 players, set-top boxes, and other electronic gadgets. SQLite has a small code footprint, makes efficient use of memory, disk space, and disk bandwidth, is highly reliable, and requires no maintenance from a Database Administrator.
- Application File Format. Rather than using fopen() to write XML, JSON, CSV, or some proprietary format into disk files used by your application, use an SQLite database. You'll avoid having to write and troubleshoot a parser, your data will be more easily accessible and cross-platform, and your updates will be transactional. (more...)
- Website Database. Because it requires no configuration and stores information in ordinary disk files, SQLite is a popular choice as the database to back small to medium-sized websites.
- Stand-in For An Enterprise RDBMS. SQLite is often used as a surrogate for an enterprise RDBMS for demonstration purposes or for testing. SQLite is fast and requires no setup, which takes a lot of the hassle out of testing and which makes demos perky and easy to launch.
- Zero-Configuration
- SQLite does not need to be "installed" before it is used. There is no "setup" procedure. There is no server process that needs to be started, stopped, or configured. There is no need for an administrator to create a new database instance or assign access permissions to users. SQLite uses no configuration files. Nothing needs to be done to
- tell the system that SQLite is running. No actions are required to recover after a system crash or power failure. There is nothing to troubleshoot.SQLite just works.
- Other more familiar database engines run great once you get them going. But doing the initial installation and configuration can be intimidatingly complex.
- Serverless

Most SQL database engines are implemented as a separate server process. Programs that want to access the database communicate with the server using some kind of interprocess communication (typically TCP/IP) to send requests to the server and to receive back results. SQLite does not work this

way. With SQLite, the process that wants to access the database reads and writes directly from the database files on disk. There is no intermediary server process.

There are advantages and disadvantages to being serverless. The main advantage is that there is no separate server process to install, setup, configure, initialize, manage, and troubleshoot. This is one reason why SQLite is a "zero-configuration" database engine. Programs that use SQLite require no administrative support for setting up the database engine before they are run. Any program that is able to access the disk is able to use an SQLite database.

On the other hand, a database engine that uses a server can provide better protection from bugs in the client application - stray pointers in a client cannot corrupt memory on the server. And because a server is a single persistent process, it is able control database access with more precision, allowing for finer grain locking and better concurrency. Most SQL database engines are client/server based. Of those that are serverless, SQLite is the only one that this author knows of that allows multiple applications to access the same database at the same time. An SQLite database is a single ordinary disk file that can be located anywhere in the directory hierarchy. If SQLite can read the disk file then it can read anything in the database. If the disk file and its directory are writable, then SQLite can change anything in the database. Database files can easily be copied onto a USB memory stick or emailed for sharing. Other SQL database engines tend to store data as a large collection of files. Often these files are in a standard location that only the database engine itself can access. This makes the data more secure, but also makes it harder to access. Some SQL database engines provide the option of writing directly to disk and bypassing the filesystem all together. This provides added performance, but at the cost of considerable setup and maintenance complexity. Single Database File, Stable Cross-Platform Database File.

The SQLite file format is cross-platform. A database file written on one machine can be copied to and used on a different machine with a different architecture. Big-endian or little-endian, 32-bit or 64-bit does not matter. All machines use the same file format. Furthermore, the developers have pledged to keep the file format stable and backwards compatible, so newer versions of SQLite can read and write older database files.

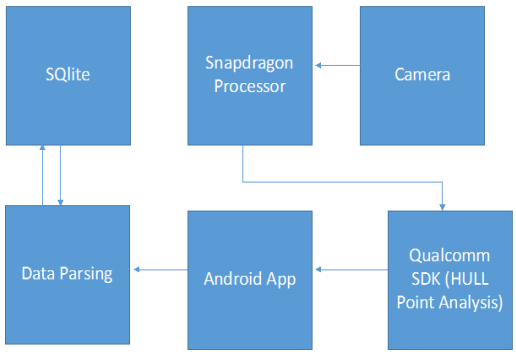
Most other SQL database engines require you to dump and restore the database when moving from one platform to another and often when upgrading to a newer version of the software.

Compact, When optimized for size, the whole SQLite library with everything enabled is less than 500KiB in size (as measured on an ix86 using the "size" utility from the GNU compiler suite.) Unneeded features can be disabled at compile-time to further reduce the size of the library to under 300KiB if desired.

Most other SQL database engines are much larger than this. IBM boasts that its recently released CloudScape database engine is "only" a 2MiB jar file - an order of magnitude larger than SQLite even after it is compressed! Firebird boasts that its client-side library is only 350KiB. That's as big as SQLite and does not even contain the database engine. The Berkeley DB library from Oracle is 450KiB and it omits SQL support, providing the programmer with only simple key/value pairs.

3.2 System Module

3.2.1 System Design Analysis:



Block Diagram Fig 3.2

Camera which is in our Smartphone is used to capture images live and used to recognize faces and store in database.The camera input is fed into Snapdragon processor which formulates the values with high processing power and GPU. The face in that live preview excluding the background and noise is done by Qualcomm SDK which basically uses Hull point Analysis algorithm. Hull point analysis is an algorithm where is marks the points on the outer surface of the face and each point distance is measured in pixels for accuracy and most deviated angle difference is noted and points are marked and connected to circum point of the face which is in middle and the distance are calculated and sent to our Android App for ease of use for user. Then data from android app are parsed and values are stored in SQlite under corresponding user with help of unique ID. Updating user takes the same step and the data obtained are updated with help of ID.

3.2.2 Facial Processing

Transform your apps with the ability to profile faces. The Snapdragon SDK makes it possible to detect a smile, determine where the eyes are looking, and detect blinking. Using this, you can create new interactions and enhance the user experience by doing things like integrating with real-time camera preview or analyzing photos in a photo album.

You can track a variety of facial properties with each frame:

Blink Detection – measure how open each eye is

Gaze Tracking – assess where the subject is looking

Smile Value – estimate the degree of the smile

Face Orientation – track the Yaw, Pitch and Roll of the head

These capabilities work with both real-time and stored images or videos, so your app can integrate them for different kinds of uses.

3.2.3 Facial Recognition

Go beyond face detection and perform real-time face analysis to identify people. You can use these Snapdragon SDK for Android capabilites to develop apps that can add users to an internal database through face registrion and then identify users based on facial analysis. Thes features do not use any cloud-based recognition and are done entirely offline.

These features allow your app to interact with a user in more ways:

Profiles – enable per-user settings and preferences

Turn-based gaming – allow players to take turns by setting up the UI specific to each player

Photo apps – run automated framing or facial processing to set up preferred users for picture taking

This feature set works with both real-time and stored images or videos.

Requirements

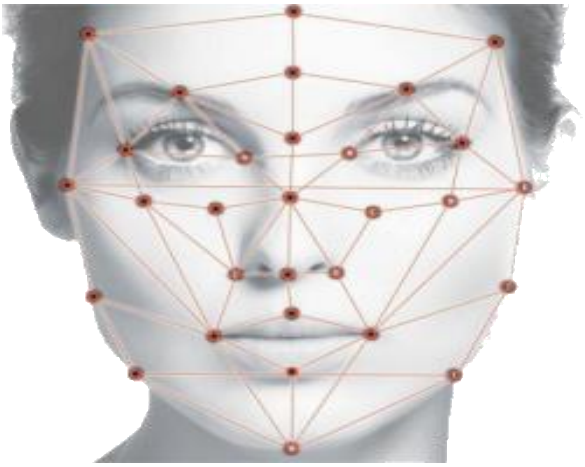
Hardware Requirements: Snapdragon S4, 200, 400, 600, or 800

Software Requirements: Android 4.0.3 and up

Sample Devices: Nexus 4, Samsung Galaxy S4 (Quad-core), LG Optimus G, HTC One, Sony Xperia Z Tablet,VIVO V1.

Introduction to Face detection from an image is a key problem in human computer interaction studies and in pattern recognition researches. Many researchers on automatic face detection have been proposed recently. The researchers of face detection are divided into a various of approaches.

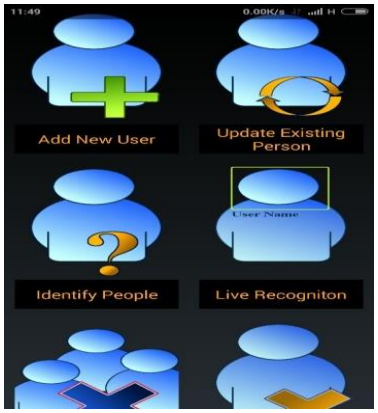
The feature-based approaches required the detection and measurement of salient facial points used geometrical distances and angles between primary facial features such as eyes, nose and mouth to classify faces using an economic representation of the face where the elements were based on their relative positions and sizes. A template-matching strategy was based on the earlier work of using feature-based templates of the mouth, eyes and nose, in addition to whole face templates. suggested that the expected shape of geometric features could be used to construct deformable templates in which templates could be translated, rotated and deformable to fit the best representation of their shape present in the image. However, low-level computer vision algorithms such as feature-based approaches were not powerful enough to find out all possible face regions and there were not likely to perform well in case of small faces or low quality images. The deformable templates were computationally expensive and not robust to everyday variation. Also, although the PCA was a very efficient designed specifically to characterize face region, it was not invariant to image transformations such as scaling, shift of rotation in its original form and requires complete relearning of the training data to add new individuals to the database. Although performance of pattern method approaches reported was quite well, and some of them could detect non-frontal faces, the approaches were extremely computationally expensive. We find the face candidate by skin and hair color like and the face by adapting intersection relationship(ICH) between a convex-hull of skin color regions(SCH) and a convex-hull of hair color regions(HCH).



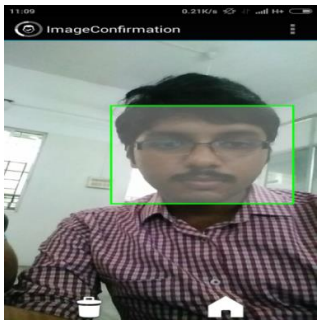
Computation of hull Fig 3.3

4. SCREENSHOTS

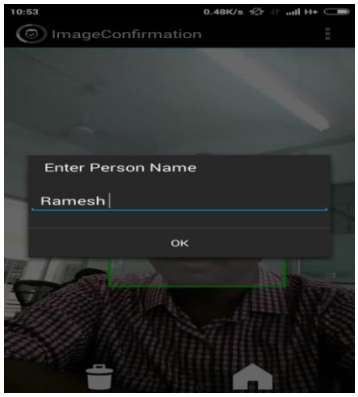
IMPLEMENTATION



ADD USER MODULE



ASSIGNING NAME FOR THE USER



IDENTIFICATION MODULE

5. CONCLUSION AND FUTURE ENHANCEMENT

5.1 CONCLUSION

This is an industrial project, it is being used currently for the attendance at various organizations and for payroll tracking system.This can be extended to allowance of respective individual through vault.This deals with accuracy and security when unauthorized persons try to unlock and is helpful for recovering the occurrence of uncertainities.This would be very useful when further enhanced and greatly improves accuracy and time.

5.2 FUTURE ENHANCEMENT

Another important scope is that the jailers photos are taken and stored in the database. In case if the jailers escape from the prison and mingle among the common people, with the already available hull point measurements of the particular person we can spot the person at any OF situation and can track him through the signal cameras. This can be done with accuracy and in a shorter time period as this hull point analysis helps in identifying the right person instantly.

Similarly another major advantage would be spotting of terrorists inside the country.We can spot the terrorists in a crowd even when they are in disguise because any external feature or appearance change would not affect the hull analysis as it takes into account only the internal points for maintaining the database at the back end.This would be a great help to the society as it safeguards the country from the encountered dangers that it ought to face.

REFERENCES

[1] Preparata, Shamos, *Computational Geometry*, Chapter "Convex Hulls: Basic Algorithms"

[2] Luc Devroye and Godfried Toussaint, "A note on linear expected time algorithms for finding convex hulls," *Computing*, Vol. 26, 1981, pp. 361-366.

[3] Barber, C. Bradford; Dobkin, David P.; Huhdanpaa, Hannu *(1December 1996). "The quickhull algorithm for convex hulls". ACM Transactions on Mathematical Software **22**(4)*

[4] Avis, David; Bremner, David; Seidel, Raimund *(1997), "How good are convex hull algorithms?", Computational Geometry: Theory and Applications **7** (5-6): 265–301,doi:10.1016/S0925-7721(96)00023-5.*

[5] *"Find Snapdragon Smartphones, Tablets and Smart Devices". ualcomm. 2015-12-08. Retrieved 2015-12-24.*

[6] *"Snapdragon seeds Qualcomm's future". Electronic Engineering Times. 4 June 2007. Retrieved 2 October 2014.*