



Automated Video Surveillance System Using Python And A Mobile App.

FINAL YEAR PROJECT

Dr. Senthil Jayavel

SUBMITTED BY-

Satvik Dhandhanian(11BCE0431)

Sahil Govel(11BCE0472)

Rishi Verma(11BCE377)

Introduction

- ▶ The primary purpose of this project is to free the owner of the extensive task of manually reviewing the video to identify the person who is present in front of the Video Surveillance System.
- ▶ In recent times there has been an alarming increase in number of crimes. With an automated Surveillance system the need for a human to manually review the surveillance footage is not required. Thus it reduces labor cost as well as saves time. Also, it is able to send notifications to the user's mobile phone as soon as it detects a face.

Aim & Objective

- ▶ The objective is to monitor the behavior, activities, or other changing information, usually of people for the purpose of influencing, managing, directing, or protecting them.
- ▶ There are two main purposes to a video surveillance system. The first and most important one is to deter crime. The second is to help catch criminals when a crime has been committed.

Viola /jones face detection algorithm

- ▶ This algorithm is extremely fast and efficient as compared to other face detection algorithms.
- ▶ It has a very high detection rate.
- ▶ Haar feature selection helps in the detection of facial features from non facial features.
- ▶ Adaboost training algorithm is the part of this , which enables the algorithm to learn from its existing result and gives a better result .

Features

- ▶ Person tracking and identification.
- ▶ Abnormal event detection.
- ▶ Key point tracking.
- ▶ Fast and efficient detection.
- ▶ High detection rate.

Features

- ▶ Low power requirement.
- ▶ Automated supervision needed.
- ▶ Lower Cost.

Software Specifications

► **Android Studio**

Android Studio is the official IDE for Android application development, based on IntelliJ IDEA. We use this software to develop our Android application which receives the message from the Google Cloud Messaging Server.

► **Google App Engine**

Google App Engine (often referred to as GAE or simply App Engine) is a platform as a service (PaaS) cloud computing platform for developing and hosting web applications in Google-managed data centers. Applications are sandboxed and run across multiple servers. App Engine offers automatic scaling for web applications -

Software Specifications

- as the number of requests increases for an application, App Engine automatically allocates more resources for the web application to handle the additional demand.

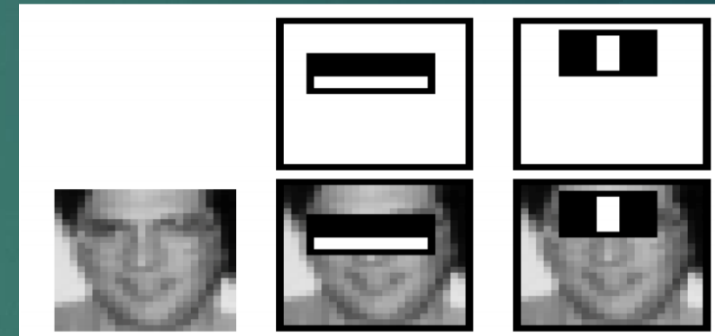
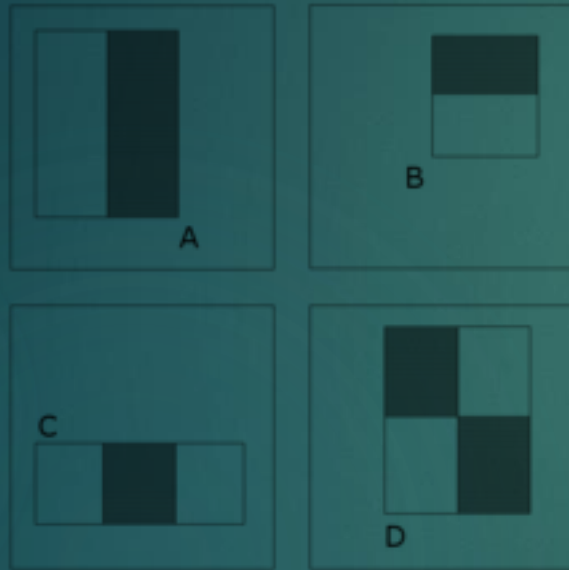
► **OpenCV**

OpenCV (Open Source Computer Vision) is a library of programming functions mainly aimed at real-time computer vision, developed by Intel Russia research center in Nizhny Novgorod. It is free for use under the open-source BSD license. The library is cross-platform. It focuses mainly on real-time image processing.

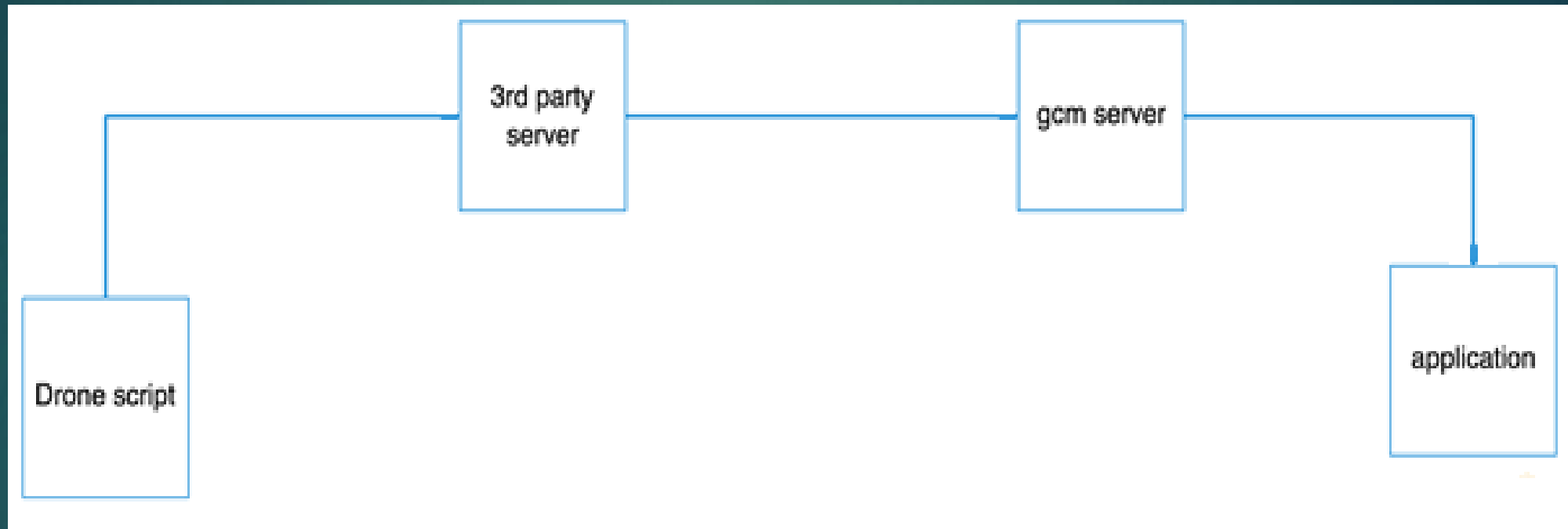


Design Approach

Haar classifiers – facial feature detection

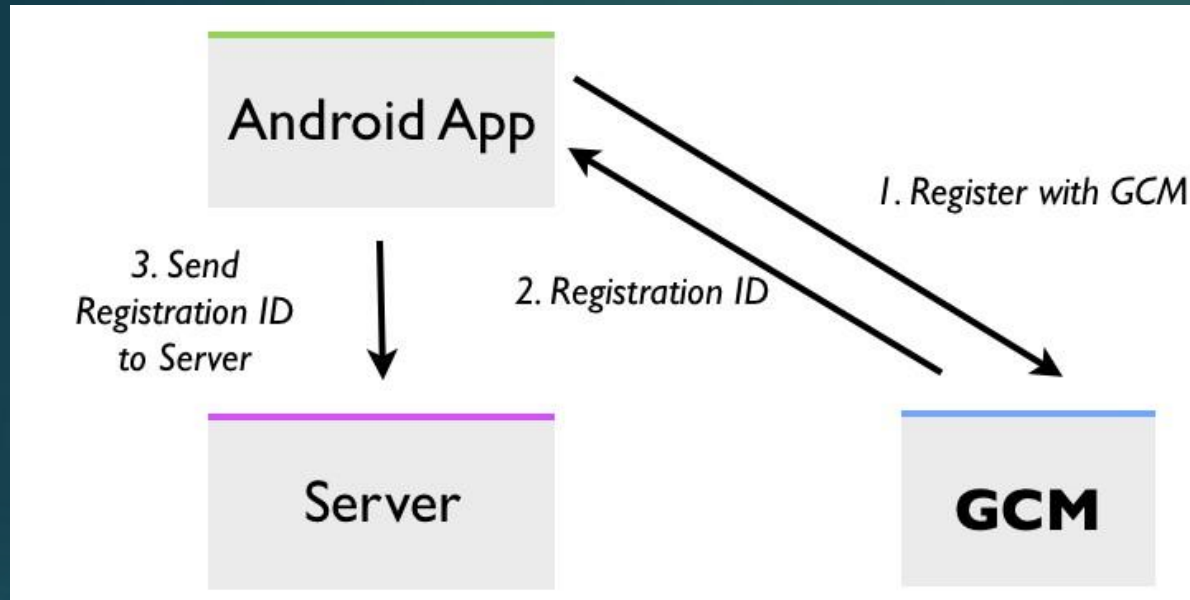


Block Diagram

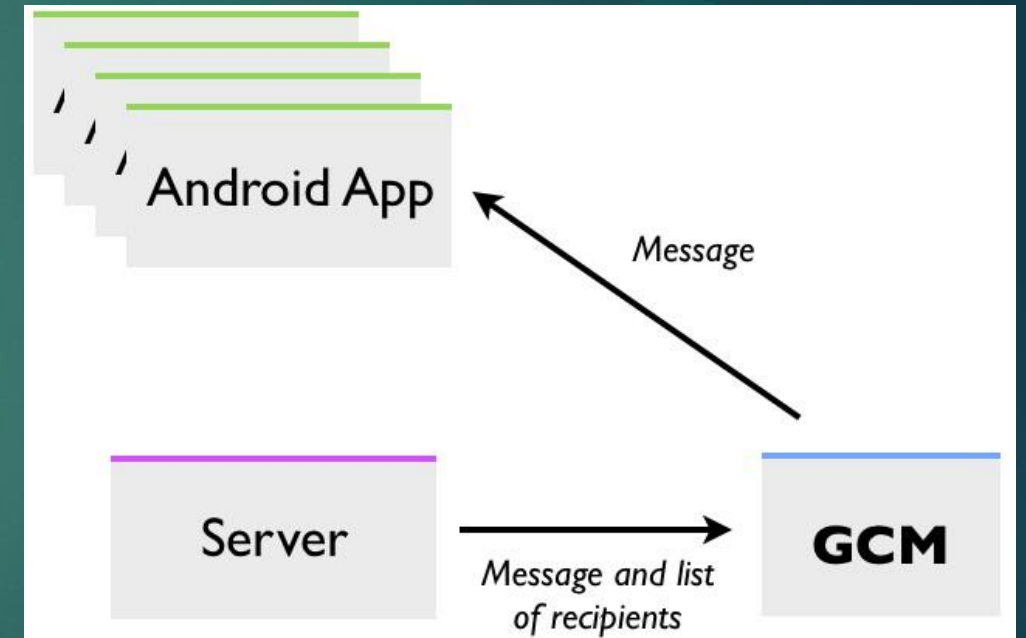


GCM Architecture



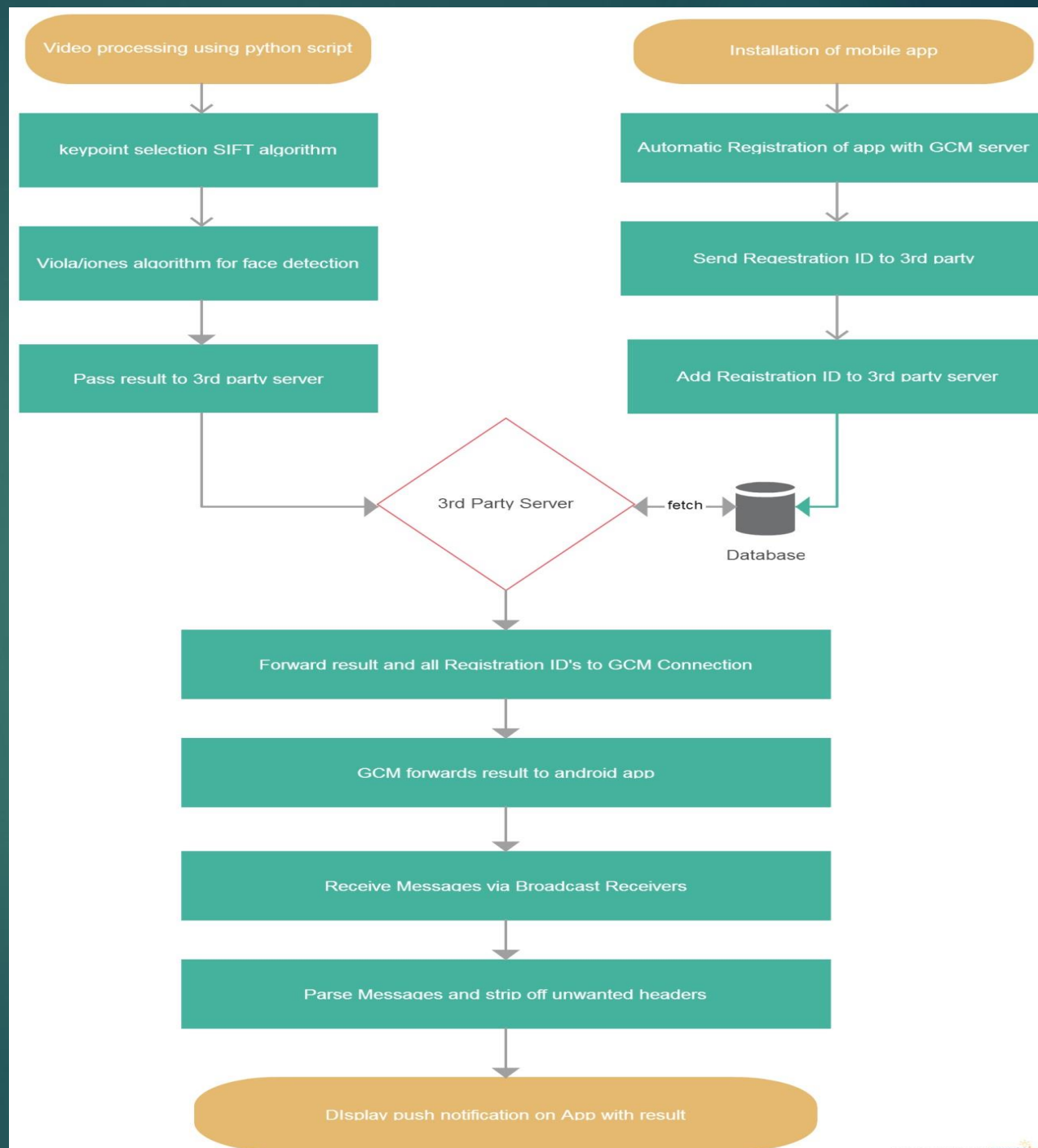


Registration With GCM Server



Sending Messages to Android Apps

Flow Diagram



Algorithm

1. Installing and setting up all the OpenCV libraries and functions and android application on the phone.
2. Running of program that detects the face in a video feed.
3. Facial feature detection.
4. Sending the result to the 3rd party server.
5. Data forwarded to GCM connection server along with all the registered device's IDs.

Algorithm

6. GCM Connection server sends details to applications using the registration IDs received from Application Server.
7. Device broadcast receiver receives the details as a bundle.
8. This bundle is scrapped off of unwanted headers and the result string is extracted. And displayed as a notification in the Notification tray.
9. Next this result string and current timestamp and stored in database.
10. Every time an Activity starts the app fetches result and timestamp from database and displays it as a list View. Also, tapping on the notification causes the application to start if it is inactive.
11. Stop.

Design Constraints

- ▶ Economic Constraint: The system will require the deployment of the video surveillance system at every entrance.
- ▶ Social Constraint: The usage of such a system can be deemed as violation of privacy in public areas.
- ▶ Sustainability Constraint: The usage of this system has to be sustained for large campuses and needs to be scaled accordingly; the sustainability of such an expansive system needs to be tested.
- ▶ Logistical Constraint: The system cannot take large amount of data's and it requires active internet connection all the time.

Design Constraints

- Environment Constraint: The system may not function properly in certain environments like in dim lights or weak internet connection zones.
- Delivery Constraint: The application completely depends on the Google Cloud Messaging platform which may add delay to send the notifications to users since it is a free and open service.

Marketing Analysis

- ▶ The existing method used in most places is limited to a person who can see footage of camera to do the surveillance. This method is limited and susceptible to human error. This method also varies from place to place and is not very efficient and requires crowd sourcing.
- ▶ Our product is not susceptible to manual error since the face detection is completely autonomous this helps the system to be efficient and fast. It is easy to use and portable and the application created specifically for the system can be used by any person with an android phone.

Summary

- ▶ Our video surveillance system provides an efficient and robust system with very low failure rate.
- ▶ In this project, we have added features like object tracking , multiple face tracking and features like Adaboost.
- ▶ Currently, we are incorporating multiple face identification system and trying to make the system more efficient in facial identification.
- ▶ The system is flexible as it can be further customized to extract suitable data required to improve the efficiency of the existing system.



Thank you

