



Notification Based on Face Detection

By:

Parth Desai

HardikKumar Dudhat

Milan Dabhi

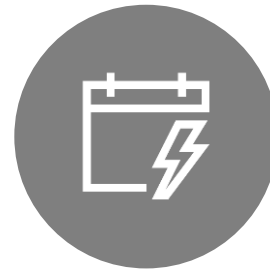
Outline



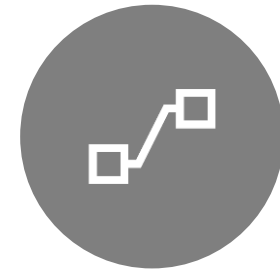
Introduction



Methodology



Implementation



Conclusion

Introduction

Face Recognition?

Why Cloud Computing?

- **A**ccessibility
- **B**oost throughput
- **C**ost-effective
- **D**ependable
- **E**vent-driven

Objectives

1. Image recognition algorithm.
2. Train the model
3. Update data into firebase
4. Cloud Function
5. On successful trigger, send notification

Methodology

Firestore Realtime database



Firestore Cloud function

Python libraries

- cmake
- dlib
- opencv
- face_recognition




Firestore Realtime Database

The screenshot shows the Firebase Realtime Database console for a project named "Face Detection Notification". The "Data" tab is selected, displaying a tree view of the database structure. The root node is a JSON object with two main branches: "metadata" and "notification". The "notification" node contains an "email" field with the value "p142desai@gmail.com" and an "unknown" field with the value "0".

Face Detection Notification ▾ Go to docs  



Realtime Database

[Data](#) [Rules](#) [Backups](#) [Usage](#)

<https://face-detection-notification-default-rtdb.firebaseio.com>   



```
https://face-detection-notification-default-rtdb.firebaseio.com/  
├── metadata  
│   └── lc: "p142desai@gmail.com"  
└── notification  
    ├── email: "p142desai@gmail.com"  
    └── unknown: "0"
```

Firestore Cloud Function

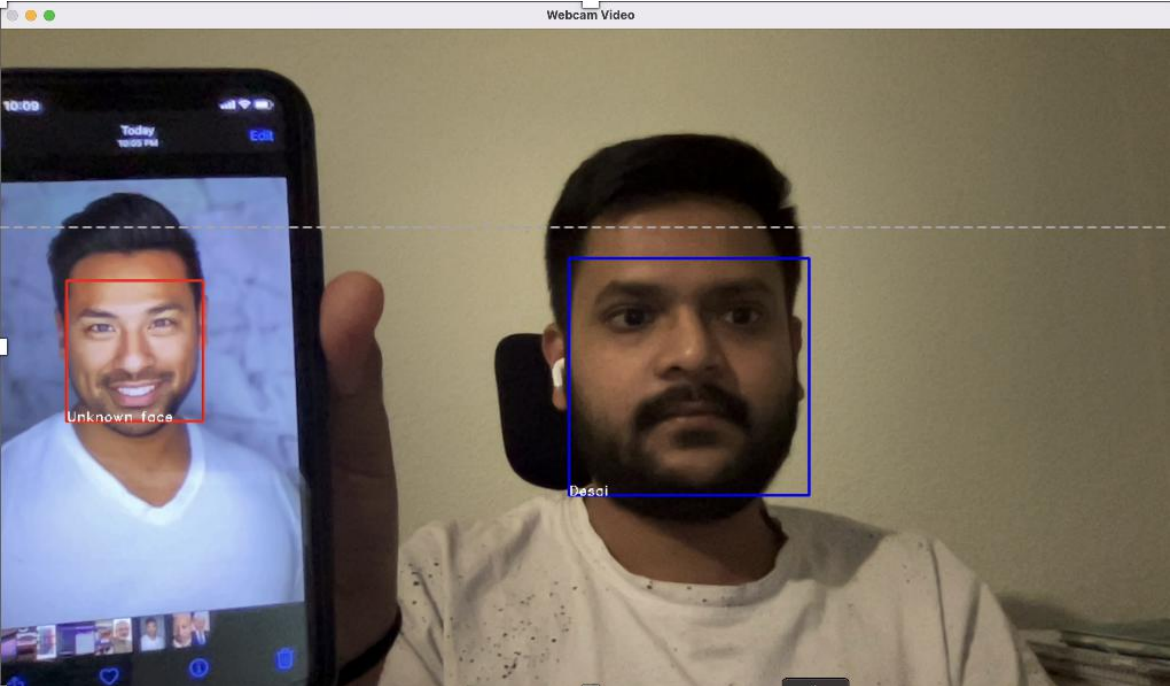
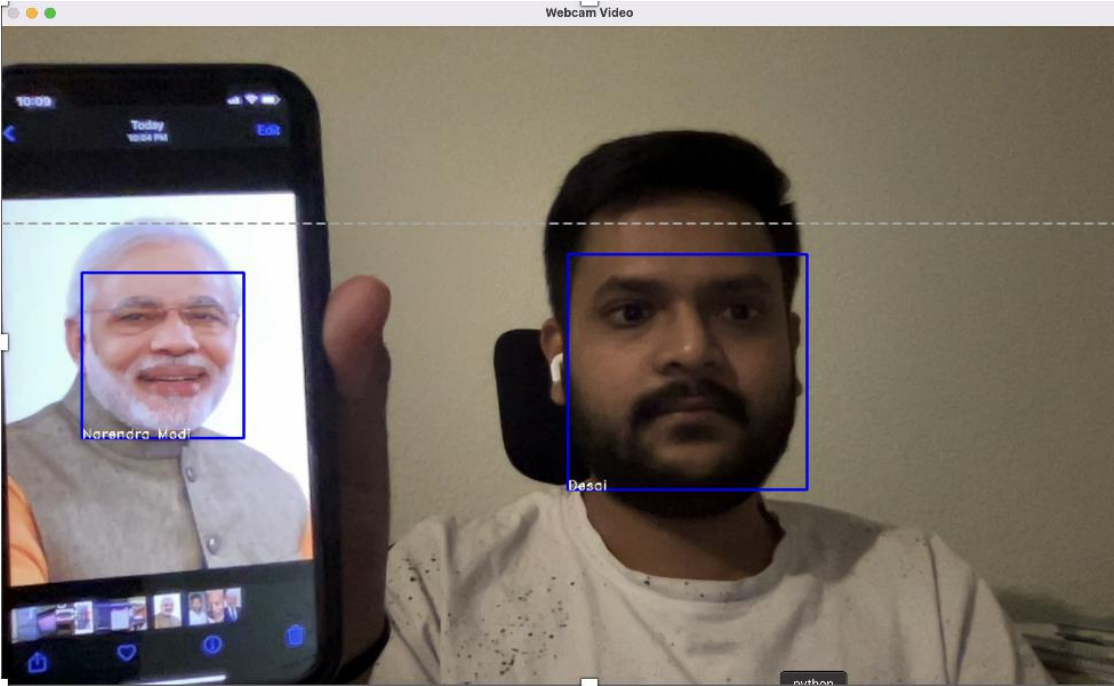
Face Detection Notification ▾ Go to docs  

Functions

[Dashboard](#) [Health](#) [Logs](#) [Usage](#)

Function	Trigger	Version	Requests (24 hrs)	Min / Max Instances	Timeout
sendEmail us-central1	 ref.update notification/unknown	v1	0	0 / —	1m 

Realtime Face-Recognition : OpenCV



Thank you