AI-Biometric-Driven Smartphone App for Strict Post-COVID Home Quarantine Management

Gaurav Jaswal
Indian Institute of Technology Delhi

Rohit Bharadwaj and Kamlesh Tiwari
Birla Institute of Technology and Science Pilani

Daksh Thapar, Piyush Goyal, and Aditya Nigam
Indian Institute of Technology Mandi

***Abstract*—COVID-19 has been announced as a Global Communal Health Extremity by WHO on January 2020. Meaningful preventive solutions have been taken with smartphone selfie/geofencing apps toward managing mandatory home quarantine and physical distancing. In the post-COVID world, fast screening and strict quarantine can play a crucial role in bringing back normality. Quarantine being offered at home can be a comfortable solution for both government and patients. On the other hand, it can be hazardous if not strictly followed and adequately realized. However, the existing geofencing/face selfie apps take static photographs and location data at certain time intervals that can allow patients to violate the rules between those periods, thus failing to ensure active user identity. To realize unbreached home quarantine policies, this article introduces a CUBA-HQM smartphone app that performs continuous user biometric authentication (CUBA) augmented with geofencing using AI technology. The purpose of continuous tracking is to strictly control the spread of infectious diseases in society by monitoring the individual move in/out in the quarantine zone.**

Digital Object Identifier 10.1109/MCE.2020.3039035
Date of publication 18 November 2020; date of current version 13 April 2021.

Published by the IEEE Consumer Technology Society

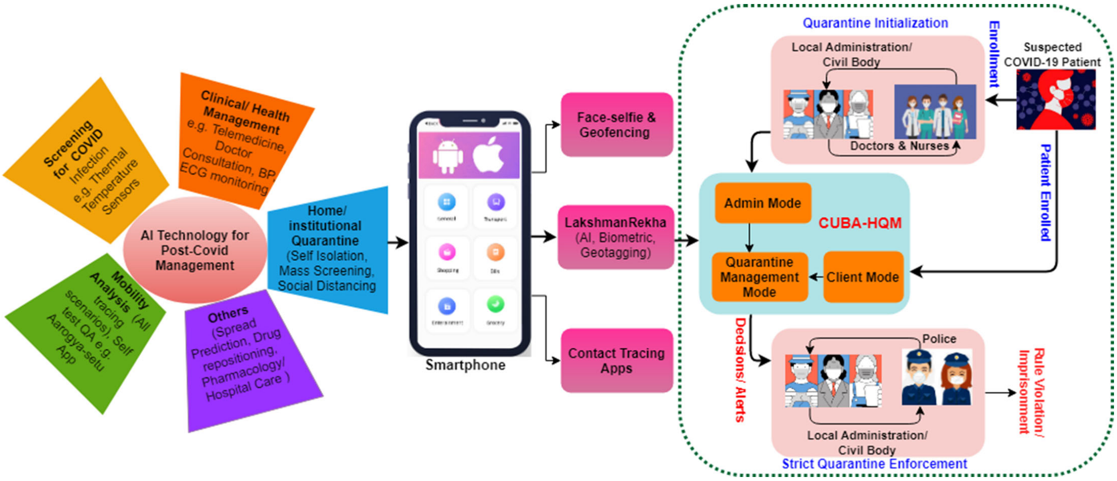
Screening for COVID Infection e.g. Thermal Temperature Sensors
Clinical/ Health Management e.g. Telemedicine, Doctor Consultation, BP, ECG monitoring
AI Technology for Post-Covid Management
Home/ institutional Quarantine (Self Isolation, Mass Screening, Social Distancing
Mobility Analysis (AI tracing scenarios), Self test QA e.g. Aarogya-setu App
Others (Spread Prediction, Drug repositioning, Pharmacology/ Hospital Care)
Smartphone
Face-selfie & Geofencing
LakshmanRekha (AI, Biometric, Geotagging)
Contact Tracing Apps
Quarantine Initialization
Local Administration/ Civil Body
Doctors & Nurses
Enrollment
Suspected COVID-19 Patient
Patient Enrolled
Admin Mode
CUBA-HQM
Quarantine Management Mode
Client Mode
Decisions/ Alerts
Police
Local Administration/ Civil Body
Rule Violation/ Imprisonment
Strict Quarantine Enforcement

Figure 1. Role of AI techniques for Post-COVID HQM and self-care using smartphone.

In order to help millions of people impacted by the "Severe Acute Respiratory Syndrome Coronavirus-2" virus, the government across the countries have been imposing lock-downs and closing borders. It has been aiming to flatten the epidemic curve impacting an estimated 5 billion people, the first global public health crisis of the 21st century. Very few countries have managed to keep it well-controlled. Unlike the EU and USA, lock-downs are not perceived sustainable at any level in countries like India as it is bankrupting the economy and destroying livelihoods. To cope-up with the situation, the governments are moving from lock-downs and guiding necessary precautionary measures such as maintaining social distance, mass screening, personalized screening, and imposing self/home quarantine. The home quarantine is home-based isolation, whereas institutional quarantine is enforced at a facility under the supervision of a civil body for those who have early coronavirus illness symptoms. Considering the fact that we do not know much about the disease and its effective pharmaceutical interventions are not expected to be available soon, it has become crucial to bring more technology-intensive strategies to combat such highly infectious communicable diseases. Automated tracking and contact tracing might be one of the key-tools to monitor the movements of every person with symptoms of coronavirus as well as to trace their contact profiles.

Need and motivation for HQM system: To restart the economy, countries have been entering into the phase of lifting lock-down. Nevertheless, it has been witnessing an alarming spike in coronavirus cases that can lead to a rapid and massive increase in demand for health facilities. The availability of beds and ventilators is abysmally low and expensive for many poor households in developing countries. Therefore, to prevent the health system from being overburdened, the government's immediate priority should be to demolish the expansion of the ubiquitous epidemic through mass examining, school/university closures, transport bans, and rigorous quarantines and social distancing. Among other containment measures, the government has to take decisive interventions to promote or impose social distancing, active monitoring, and prompt isolation of all cases. AI models, along with associated technologies, can trace individuals who may have come into the proximity of contaminated once and limit its spread in the community. Several promising initiatives, however, have been started by many countries to institutionalized AI-based home quarantined management (HQM) systems enforced by local authorities (health workers/ civic bodies/ police); an exemplary flow diagram is shown in Figure 1. However, no such mobile app yet exists for HQM that can continuously track location and ensure someone's identity by combining biometric verification[1], geofencing, and AI techniques.

Table 1. Comparison of post-COVID mobile apps for HQM and self-care.

App	Description	Features	Sensors	Platform
Aarogya-setu (India)	Trace proximity of app users with COVID patient	Monitor person's breathing	GPS, Bluetooth	Open Source
HaMagen (Israel)	Allow users to know last 14 days proximity with an infected	Send updates and allow users to review alerts	GPS, Bluetooth	Open Source
DataSpende (Germany)	Check COVID symptoms in smartwatch users	Collect pulse rate, body temperature	Smart-watch	Open Source
Self-quarantine (KR)	Mandatory for everyone staying in self-isolation	Instruct for self-diagnosis	GPS	Open Source
Home-quarantine (Poland)	Randomly collect selfies and geo-location	Regular alerts and risk with personal data	GPS, face selfie	Open Source
ENS (Austria, Belgium)	Facilitate contact tracing	Preserves the user identity/ location	Bluetooth	Open Source

QM MOBILE APPS: CURRENT STATUS AND CURBS

In the current COVID-19 crisis, smartphone-centric tracking apps such as Home Quarantine, Self Quarantine, etc., are already being used to locate quarantine violators and mark quarantined areas in various subdivisions of the world, as discussed in Table 1. In order to comply quarantining, individuals under self-isolation have been enforced to relay their instantaneous position routinely via geofencing technology or required to upload face selfies every hour or ten times a day. This approach requires significant efforts to any government for keeping track of infected persons particularly, with high numbers of infections. On the other hand, due to manual intervention, it brings huge trouble to individuals under home confinement. Also, geofencing apps fail to ensure the user identity throughout the time because individuals can leave cell phones at isolation zones and move in/ out quickly, leading to disobey the self-isolation rules. Similarly, the idea of uploading a face selfie every hour cannot guarantee his/her stay in a geofenced area during the intervals of time, he can also try to spoof/fool the system using a photograph containing its registered face called presentation attack. Also, countries have been employing mobile apps to enable contact tracing[2] that will alert if someone has come into closeness of any tests COVID-19 positive. Aarogya Setu (India), Exposure Notification System (Austria,, Belgium)[3], and Trace-Together (Singapore) are a few examples of mobile-centric contact tracing apps launched in 2020, as discussed in Table 1. These apps limit the duration of interactions that might help to prevent the broader spread of the virus. However, there are essential privacy inferences for the occurrence of COVID tracing apps. Thus, fully online technology is necessarily required to fix user identity and location while strictly ensuring user data privacy.

Scope of computer vision techniques for QM: Mobile phones have become an inseparable part of human lives in storing sensitive information and personal credentials. These factors can expository prioritize the need of a smartphone tool for keeping track the stay of COVID patients. Continuous user biometric authentication (CUBA) might be a frontier in this direction[1]. This approach diminishes the shortcomings of classic dynamic login procedures (fingerprint or password confrontation) by verifying the individuals identity persistently and latched the smartphone automatically if alteration in his/her identity has been reported. It becomes crucial for the app to collect information about user identity and location via in-build mobile sensors such as camera, accelerometer, gyroscope, touchpad, magnetometer, global positioning system (GPS), and microphones. To mitigate the risks of selfie attacks, the system will not use

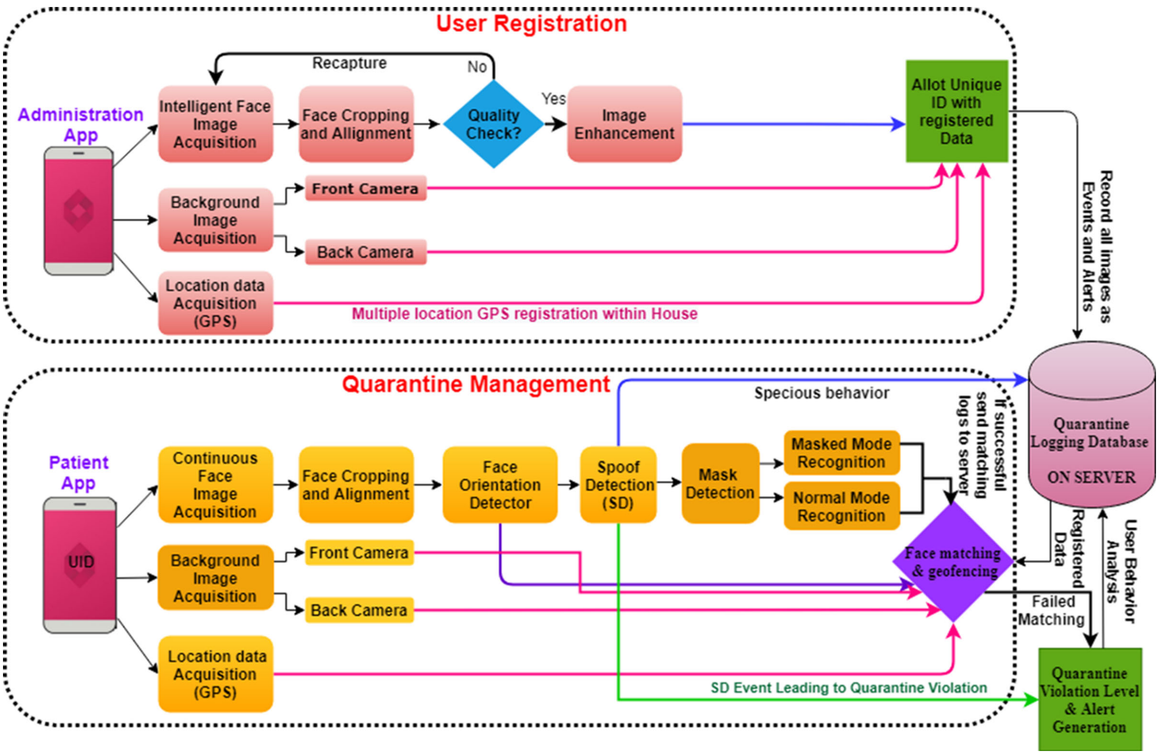
User Registration
Administration App
Recapture
No
Intelligent Face Image Acquisition
Face Cropping and Allignment
Quality Check?
Yes
Image Enhancement
Allot Unique ID with registered Data
Background Image Acquisition
Front Camera
Back Camera
Location data Acquisition (GPS)
Multiple location GPS registration within House
Record all images as Events and Alerts
Quarantine Management
Specious behavior
Patient App
UID
Continuous Face Image Acquisition
Face Cropping and Alignment
Face Orientation Detector
Spoof Detection (SD)
Mask Detection
Masked Mode Recognition
Normal Mode Recognition
If successful send matching logs to server
Quarantine Logging Database
ON SERVER
Face matching & geofencing
Registered Data
User Behavior Analysis
Failed Matching
Background Image Acquisition
Front Camera
Back Camera
Location data Acquisition (GPS)
SD Event Leading to Quarantine Violation
Quarantine Violation Level & Alert Generation

Figure 2. Proposed framework for LakshmanRekha.

physiological biometric traits like face, and/or periocular alone. The information extracted from primary traits can be augmented with the user's behavioral signatures such as gait, a touch gesture, and keystroke dynamics to make it difficult for an intruder under home isolation to spoof the multiple biometric traits simultaneously. Specifically, CUBA will match the quarantined location with the location from where the user uploaded the biometric data, but continually computes an authentication score that measures how certain it is that the quarantined user is also the one using the mobile. As we detect an action that indicates that the user identity changed, necessarily making an explicit log-out and will be notified to officials. Therefore, every quarantined user of the app will be fenced within some (say 300 m) radius of his quarantine. Considering the seriousness of the prevailing epidemic, we propose "LakshmanRekha: AI-based mobile tracking app" using geotagging and biometrics to control and monitor the individual move in/out of the quarantine zone. The proposed app can not only guarantee to locate the geofenced area but capable to continuously monitor and accurately detect the identity of a quarantined person so that no quarantined user can breach the geofence. It will be mandatory for people under quarantine to download and install the application on their mobile phones (which can be ensured by local authorities). Any unauthorized movement will be notified to the local administration (police or health department) as depicted in Figure 1. In addition to quarantine management, the proposed system serves a multipurpose: unbreachable mobile phone platform for normal (non-COVID) mobile users, curfew mode, or any national emergency mode for identifying the violators or lawbreakers.

LAKSHMANREKHA: IN-APP USER QUARANTINE MANAGEMENT

LakshmanRekha is a prototype app for post-COVID quarantine management consisting of two modules, as shown in Figure 2. In this section, we discuss our application framework and its realization.

Application Framework

LakshmanRekha consists of two modules, the front-end apps, and the proposed HQM backend. There are two front-end apps: one for administration to perform user registration and other for the patient quarantine management, as shown in Figure 2. The administration app is used for the patient's initial registration, which will include collecting necessary information about the patient, their biometric face data, and geo-location of their home. On the other hand, the patient app will continuously collect the biometric data and GPS location of the phone in use and sent it to the server. The server-based on the data from both apps will implement the HQM system, using a robust face verification biometric pipeline.

Front-End Apps The two front-end apps are as follows.

Administration app: It is used for patient enrollment and also receive/generate alarms if any patient violates the home quarantine. For registration, we take necessary information about the patient. Then, we utilize smart acquisition (fully automated data acquisition in background) for capturing the face images of the patient. Smart acquisition captures the face in three orientations: frontal, left profile, and right profile. It detects the facial landmarks, and registers it to a predefined image resolution. The whole process is automated without any human interaction. Moreover, GPS positions from three different rooms of the patient house are captured and designated as its geo-location. Enrollment step needs to be performed manually. After registration, a unique ID has been generated for the patient. If a patient violates the quarantine, the administration app will get alert with the patient ID. The administrative app will also have an option to adjust the granularity of continuity of the system, by adjusting the time duration for the authentication system.

Patient app: It should run in the background of the patient's mobile. When the phone is in usage, without interfering with the phone's regular usage, it periodically (after a fixed amount of time, which is less than 10 min) takes the photograph of the person using the front-camera and GPS location of the mobile and send it to the server as a query to be authenticated. The time interval is set by the administrator.

HQM System The proposed HQM system utilizes a biometric pipeline for implementing the quarantine, as shown in Figure 2. The biometric pipeline consists of two components: face detection and alignment, and face verification.

Face detection and alignment: Face detection and alignment is performed using a well-known state-of-the-art MTCNN[4] (https://github.com/ipazc/mtcnn). It uses cascaded convolutional neural networks trained to predict the bounding box of a face given an image. Once the face is detected, it regresses face landmarks and overlays them on a predefined face structure to align the face. Given a full image having a person's face, MTCNN provides us with the cropped and aligned face image (removing all unwanted background).

Face verification: The cropped and aligned face image is used to verify whether the person using the phone is the quarantined patient or not. This process is known as face verification. Given two face images, the verification system decides whether the images belong to the same person. We have used a benchmark face recognition system, Facenet[5] (https://github.com/davidsandberg/facenet), for face verification. It utilizes an inception network[6] for extracting facial features from an input image. The inception has been trained using triplet loss function to ensure that the same person's facial features lie close to each other as compared to different persons. During registration, the HQM stores the patient information, face images, and GPS locations of different rooms in a home on server. The query (Q) received consists of three things: an image (Q.img), GPS location (Q.gps), and id (Q.id). The HQM follows Algorithm 1. If no query has been captured for more than an hour, that could indicate that the person is violating quarantine without his phone, and will generate an alert. During nigh-time, no such alert will be generated, although any location variation is kept on checking silently.

Realization

The front-end apps have been built using Android Studio 4.0 and in Java language. The permissions required by the apps are listed in Table 2.

Algorithm 1. HQM

```
Input: Query(Q.id, Q.img, Q.gps)
 face_th,geo_th //Initializes a predefined thresholds
face_present,face_points=MTCNN(Q.img)
 //MTCNN returns a boolean indicating face presence and face
 localization points
db_img,db_gps=database(Q.id)
 //Retrieve the database image and gps location
geo_dist = square(db_gps - Q.gps)
 //Compute euclidean distance
if face_present then
  Q.img = localize_face(Q.img, face_points) //Crops face
  Q.img = allign_face(Q.img) //aligns the cropped face
  img_dist = facenet(Q.img,db_img) //Provides the facial
  distance between 2 images
  if $(img\_dist \geq face\_th) \vee (geo\_dist \geq geo\_th)$ then
    Send logs to server;
    Provide Alert;
  else
    Send logs to server;
  end
else
  if $geo\_dist \geq geo\_th$ then
    Send logs to server;
    Provide Alert;
  else
    Send logs to server;
  end
end
```

APPLICATION ANALYSIS AND DISCUSSIONS

To validate the utility of LakshmanRekha, we have performed three analysis: 1) face detection and recognition analysis, 2) geofencing performance analysis, and 3) server-load analysis.

Face detection and matching analysis: We have performed face detection and matching analysis using photographs taken from the front camera of a smartphone. We have created a small dataset of 18 subjects. Ten images have been taken for each subject in two different phases under changing illumination conditions and background making a total of 180 images. We have tested the performance of MTCNN[4] for face detection on these 180 images. It is important to note that neither MTCNN nor the Facenet has been trained on these 180 images. MTCNN can detect the face in 98% of the images. For validating face matching, we have performed two experiments: a) Interphase Matching: For the 18 subjects captured, phase 2 images have been taken as probe images and phase 1 as gallery images. We have achieved an equal error rate (EER) of 1.2% and correct recognition rate (CRR) of 91.6%. b) Scalability Analysis: We augmented the gallery data with 77 random subjects taken from test data of labeled faces in the wild (LFW) dataset[7]. Now, the total number of subjects is increased to 95 in the dataset. This has been done to analyze the scalability of the face matching system. We have achieved an EER of 0.03% and CRR of 91.6% The low EER is due to the fact that Facenet is already trained on LFW dataset. The Receiver operating characteristic (ROC) curves for both experiments are shown in Figure 3 and it is evident that there is no decrease in performance when the number of subjects increases, showing that the system is highly scalable in terms of number of subjects.

Table 2. Permissions required by the apps.

Permission	Description
Write External Storage Permissions	Store the registration images in the local app directory
System Alert Window Permissions	App runs in the background and capture photographs without interruption.
Receive Boot Completed Permission	When the device reboots the app is notified to start again
Camera and Internet Permission	To capture images and communicate with the server
Foreground Services	To make the app remain in the foreground as a service till the duration of taking the photograph
Access Fine Location	To get the precise GPS-based location data of the user
Read External Storage	To upload the images from device to the server

Geofencing analysis: The proposed HQM system relies on GPS for implementing geofencing. Merry and Bettinger[8] have shown that average accuracy of the mobile phone-based GPS is 9.9 m. As the average size of Indian households is around 23.9[9] m^2, our proposed system can efficiently implement the quarantine system for an average Indian household.

Server load analysis: The pricing of servers is based on two aspects, requests sent to server,

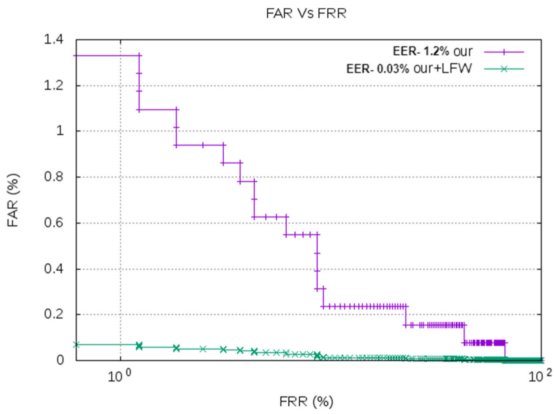

Figure 3. ROC-based performance analysis for face matching on in-house dataset, and in-house augmented with LFW dataset [7].

and space requirement. The proposed HQM system requires an average 2880 server requests per patient in one month. The biometric pipeline and algorithm utilize a one time-space of 5 GB and for each patient, 208 MB average space per month is required for logging and storing images.

SUMMARY AND CONCLUSION

CUBA has shown the potential to replace traditional biometric authentication techniques such as face selfie in consumer devices like smartphones. In this work, we have introduced AI-CUBA-HQM based smartphone app for the first time, ensuring strict post-COVID quarantining. It generates alerts to home confine as well as local administration if a person has jumped quarantine or violated isolation. Besides, various social and ethical concerns related to biometrics are covered as per government policy.

REFERENCES

1. P. Perera, J. Fierrez, and V. M. Patel, “Quickest multiple user active authentication,” in *Securing Social Identity in Mobile Platforms*. New York, NY, USA: Springer, 2020, pp. 179–196.
2. H. Cho, D. Ippolito, and Y. W. Yu, “Contact tracing mobile apps for COVID-19: Privacy considerations and related trade-offs,” 2020, *arXiv:2003.11511*.
3. K. Michael and R. Abbas, “Behind COVID-19 contact trace apps: The Google–Apple partnership,” *IEEE Consum. Electron. Mag.*, vol. 9, no. 5, pp. 71–76, Sep. 2020.
4. K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, “Joint face detection and alignment using multitask cascaded convolutional networks,” *IEEE Signal Process. Lett.*, vol. 23, no. 10, pp. 1499–1503, Oct. 2016.
5. F. Schroff, D. Kalenichenko, and J. Philbin, “FaceNet: A unified embedding for face recognition and clustering,” in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2015, pp. 815–823.
6. C. Szegedy *et al.*, “Going deeper with convolutions,” in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2015, pp. 1–9.
7. “Labeled faces in the wild: A database,” 2007. [Online]. Available: http://vis-www.cs.umass.edu/lfw/
8. K. Merry and P. Bettinger, “Smartphone GPS accuracy study in an urban environment,” *PloS One*, vol. 14, no. 7, 2019, Art. no. e0219890.
9. A. Thakur, “33% of Indians live in less space than US prisoners,” *The Times of India*, 2008.

Gaurav Jaswal is currently a Postdoctoral Fellow with the Indian Institute of Technology Delhi, New Delhi, India. His research interests include multimodal biometrics and deep learning. Contact him at gauravjaswal@ee.iitd.ac.in. He is the corresponding author of this article.

Rohit Bharadwaj is currently working toward the B.Tech degree with Birla Institute of Technology and Science Pilani, Pilani, India. His current research focuses on deep learning. Contact him at f20170633@pilani.bits-pilani.ac.in.

Kamlesh Tiwari is currently an Assistant Professor with the Birla Institute of Technology and Science Pilani, Pilani, India. His research interests include biometrics and deep learning. Contact him at kamlesh.tiwari@pilani.bits-pilani.ac.in.

Daksh Thapar is a Ph.D. Scholar with the Indian Institute of Technology Mandi, Mandi, India. His research interests include computer vision and deep learning. Contact him at d18033@students.iitmandi.ac.in.

Piyush Goyal is currently working toward the B.Tech. degree with the Indian Institute of Technology Mandi, Mandi, India. His current research focuses on deep learning. Contact him at b18077@students.iitmandi.ac.in.

Aditya Nigam is currently an Assistant Professor with the Indian Institute of Technology Mandi, Mandi, India. His research interests include biometrics, computer vision, and deep learning. Contact him at aditya@iitmandi.ac.in.