# Software Requirement Specifications

# Covid-19 SOP Enforcement System using mask detection and facial recognition

## Version: 1.7

| | |
|---|---|
| Project Code | CS491 |
| Supervisor | Mr. Shoaib Raza |
| Co Supervisor | Mr. Bahram K. Baloch |
| | |
| Project Team | Kaamla Salman<br>Rida Fatima<br>Zoha Yousuf |
| Submission Date | 14 December 2020 |

# Document History

| Version | Name of Person | Date | Description of change |
|---|---|---|---|
| 1.1 | Rida Fatima | 26-10-2020 | Document created<br>added chapter 1: Introduction |
| 1.2 | Kaamla Salman | 01-11-2020 | added chapter 2: Overall system description |
| 1.3 | Zoha Yousuf | 10-11-2020 | Added Context Diagram |
| 1.4 | Rida Fatima | 12-11-2020 | Added chapter 3: external interface requirements |
| 1.5 | Kaamla Salman | 18-11-2020 | Added 4.1: Functional hierarchy<br>Added Use case diagram |
| 1.6 | Rida Fatima | 20-11-2020 | Added Use case table |
| 1.7 | Zoha Yousuf | 21-11-2020 | Added chapter 5: Non-functional requirements<br>Added references and appendices |

# Distribution List

| Name | Role |
|---|---|
| Mr. Shoaib Raza | Supervisor |
| Mr. Bahram K. Baloch | Co- Supervisor |
| Computer Science Department | |

# Document Sign-Off

| Version | Sign-off Authority | Sign-off Date |
|---|---|---|
| 1.1 | Mr. Shoaib Raza | 30-11-2020 |
| 1.2 | Mr. Bahram K. Baloch | 9-11-2020 |
| 1.3 | Mr. Shoaib Raza | 11-11-2020 |
| 1.4 | Mr. Bahram K. Baloch | 17-11-2020 |
| 1.5 | Mr. Shoaib Raza | 20-11-2020 |
| 1.7 | Mr. Shoaib Raza | 23-11-2020 |

# Table of Contents

1. *8*
1.1. Purpose of Document 6
1.2. Intended Audience 6
1.3 Abbreviations
……..6
1.4. Document Convention 6
2. *9*
2.1. Project Background 7
2.2. Project Scope 7
2.3. Not In Scope 7
2.4. Project Objectives 7
2.5. Stakeholders 7
2.6. Operating Environment 8
2.7. System Constraints 8
2.8. Assumptions & Dependencies 8
3. *11*
3.1. Hardware Interfaces 10
3.2. Software Interfaces 10
3.3. Communications Interfaces 10
4. *13*
4.1. *13*
4.2. Use Cases 12
4.2.1. COVID-19 SOP Enforcement System 12
5. *16*
5.1. Performance Requirements 14
5.2. Safety Requirements 14
5.3. Security Requirements 14
5.4. User Documentation 14
6. *17*
7. *17*

# 1. Introduction

## 1.1. Purpose of Document

This document provides a detailed description of the environment and purpose of the software that is being developed. It completely describes what are the software's hardware and software requirements and how it is expected to perform while also discussing its user interface and target audience.

## 1.2. Intended Audience

This document can be used by several stakeholders such as, health-conscious employers, SOP enforcement authorities and various institutions and organizations. It will be utilized by the developers to keep track of their progress.

## 1.3 Abbreviations

- MTCNN: Multi-Task Cascaded Convolutional Neural Network
- YOLO: you only look once
- SMTP: Simple Mail Transfer Protocol
- MongoDB: mongo database
- OpenCV: Open-Source Computer Vision Library

## 1.4 Document Convention

The font Arial is used at size 11. The heading's font size is 16 in bold letters whereas the subheading font is size 12 in bold.

# 2. Overall System Description

## 2.1. Project Background

After the spread of COVID-19, companies are trying to update their existing infrastructure and operations in an attempt to make them safe and efficient. Since the pandemic is ongoing, some measures are required for the protection of the employees and customers. Hence, to minimize the risk factors, advanced applications and services are needed. Medical authorities have recommended the use of face masks and coverings for avoiding contamination along with social distancing.

Kaihan et al. proposed a face detection and segmentation method based on an improved Mask R-CNN, named G-Mask, which incorporates face detection and segmentation into one framework aiming to obtain more fine-grained information of face. They used the ResNet-101 for feature extraction. A Generalized Intersection over Union (GIoU) was used for improving accuracy. it was used as a bounding box loss function [1]. Another paper proposed a methodology for mask detection composing of two components: feature extraction using ResNet and classification based on decision making trees [2].

Mingjie Jiang et al. in their research proposed a one-stage detector consisting of a pyramid network for fusing high-level semantic information. It included multiple feature maps and a model that detects face masks [3].

S. Ge et al. made a customized dataset MAFA for detecting faces with masks. They used LLE-CNNs for it. The research was mainly composed of 3 modules. Firstly, it combined 2 pre-trained CNNs for extracting facial regions from images used as input and then represented it with HD descriptors. Secondly, to transfer descriptors to similarity-based descriptor embedding modules via incorporating embedding modules. This was done by a locally linear embedding algorithm (LLE). Thirdly, Incorporation of verification module was done for identification of candidates' facial regions and position refinery by collectively performing classification and regression analysis [4].

## 2.2. Project Scope

This system will assist authorities or employers in accurately identifying individuals who do not follow SOPs in a workplace or public area. It will recognize the violator if they are not wearing a mask and inform the authorities if there is a crowd gathered in close proximity. It also increases accountability by sending warnings via email. What would have been a strenuous job for a human is automatically made effortless. The transmission of COVID-19 can be lessened via the proper use of this system.

## 2.3. Not In Scope

- Sickness detection: Our system will not detect facial cues of sickness.
- Heat map: it will not represent areas with a high density of people in the form of heat maps.

## 2.4. Project Objectives

A system based on neural networks that imposes SOPs by detecting masks and groups who do not maintain social distancing while identifying violators. Various applications of deep learning and computer vision like landmark detection, object detection and face recognition are incorporated which gives more precision in

classification and identification. Furthermore, if an individual violates the SOPs, the management of that organization and the person will be notified of the violation via email and various analytics will be generated regarding violations.

## 2.5. Stakeholders

- Development team
- Supervisors
- University's final year jury and committee
- Health-conscious employers
- SOP enforcement authorities
- Institutions and organizations

## 2.6. Operating Environment

This system requires a computer with at least 8gb RAM with a processor equivalent to Core i5 or above for smooth operation on a Windows 10 or Linux operating system. The client should have an IP camera of minimum 2MP installed to get a steady clear stream. It also requires a strong internet connection if the database is on cloud or sufficient hard disk memory if a local database is used.

## 2.7. System Constraints

- **Software constraints:** This system is only available as a desktop application. MTCNN has been used for face detection which can be slow under stress.
- **Hardware constraints:** Camera should be good enough so that it is able to show the faces clearly.
- **Cultural constraints:** This system is only available for use in the English language and specifically works on the faces of the Pakistani population.
- **Legal constraints:** Only authorized users in the organization can use this system and they should not misuse any information from the database or generated by the system.
- **Environmental constraints:** The area should not be very heavily populated.
- **User constraints:** The user should have the relevant skills to operate this system and should be connected to the relevant authorities who can take actions against SOP violators.
- **Off the shelf components:** Cameras which give the appropriate size ad ratio of the frame should be installed in areas that require monitoring.

## 2.8. Assumptions & Dependencies

While designing it is assumed for the hardware that the user will have 8GB RAM, a processor equivalent to Core i5 or above. The organization should have an IP camera of minimum 2MP installed. It is also assumed that the user will have complete knowledge of the system. Another assumption made is that the people in the real time video will not be wearing caps. The system is dependent on whether the image of the employees shows an appropriate number of facial landmarks.

# 3. External Interface Requirements

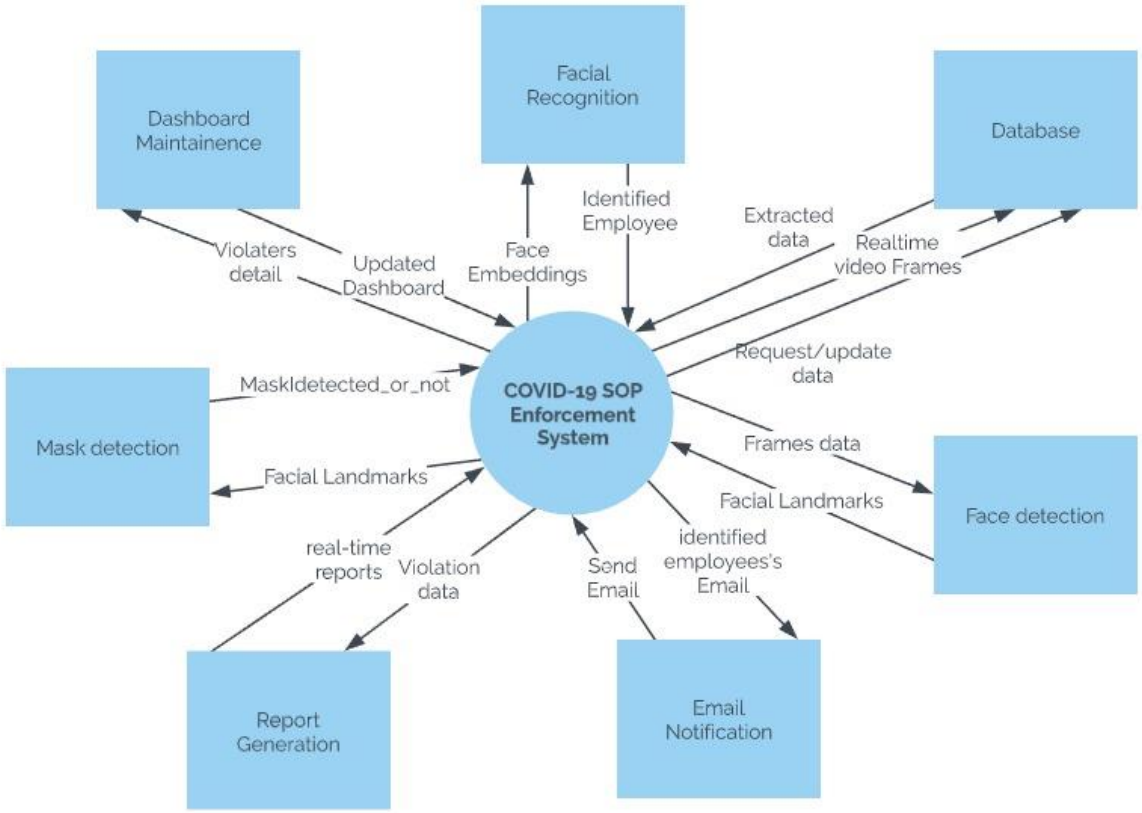


Figure 1: Context Diagram of COVID-19 SOP Enforcement System

## 3.1. Hardware Interfaces

Our system will get a steady stream of frames from an IP camera. The frames are transmitted over the network via WiFi.

## 3.2. Software Interfaces

The SOP enforcement system is operable on a Windows or Linux platform. Our system will access the MongoDB [5] database via the internet. The major libraries used are TensorFlow 2.0, OpenCV, NumPy for preprocessing the images and creating the models. PyQT tool is utilized in making the user interface for the desktop application. The data items(frames) are used and shared in all the detection procedures and facial recognition modules.

## 3.3. Communications Interfaces

- An SMTP server will be utilized in sending emails to the violators.
- MongoDB Atlas is used to communicate with the database.
- OpenCV provides us the interface to easily get frames from the camera [6].

# 4. Functional Requirements

## 4.1. Functional Hierarchy

- **Mask Detection:** Our system will include a module that will detect if an individual is wearing a mask or not. It will work for all types of masks ranging from surgical masks, N95 to cloth masks.
- **Face detection:** If a person is not wearing a mask, face detection will be used to determine if there are any faces in the image.
- **Face recognition:** After detecting the face in the frame, it will be compared to other faces in the database to see if there is a match and learn the person's identity.
- **Crowd detection:** To ensure that people are maintaining social distance, our system will detect groups of people.
- **Record generation:** A report will be generated containing graphs and information regarding a violation of SOPs.
- **Email notification:** An email will be sent to the person who violates the SOPs.
- **User interface:** A GUI interface to display the various outputs, violations and reports.

## 4.2. Use Cases

### 4.2.1. COVID-19 SOP Enforcement System

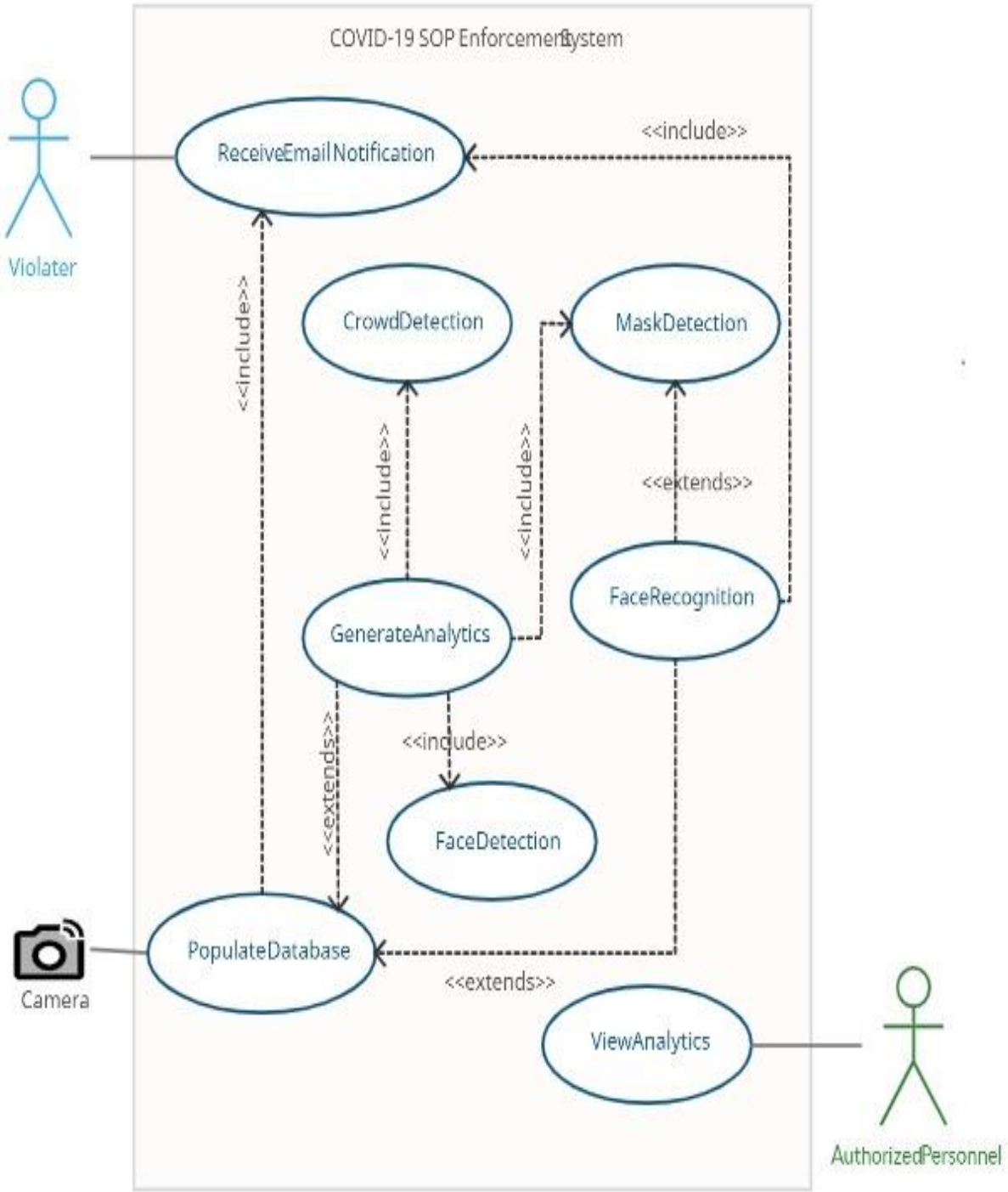


Figure 2: Use case diagram of COVID-19 SOP Enforcement System

There are 3 actors in this system: Violator, Authorized Personnel, and the camera. The Violator will Receive email notification. The authorized personnel will view Analytics. The camera will populate the database. The Populate database includes receive email notification as Email of the violator will be retrieved. The system will generate analytics which will extend to populate databases and will include Face Detection, Mask Detection, Crowd Detection. The Face recognition feature includes receive email notification and extends to populate database and mask detection.

| **01: COVID-19 SOP Enforcement System** | | |
|---|---|---|
| **Use case Id:** | 01 | |
| **Actors:** | Violators, Authorized Personnel | |
| **Feature:** | Identify the violators of the SOP's | |
| **Pre-condition:** | - system predicts the faces accurately.<br>- connection with the database is establishes | |
| **Scenarios** | | |
| **Step#** | **Action** | **Software Reaction** |
| **1.** | Images are captured using a camera. | Database is populated with the images. |
| **2.** | System detects the face of a person. | Face is detected. |
| **3.** | Crowd is detected. | An email is sent to the authorized personnel. |
| **4.** | Mask is not detected on the face of a person. | If a person is not wearing a mask, face of a person is recognized and matched with the other faces in the database and then an email is sent to the person violating the SOP's and to the authorized personnel. |
| **Alternate Scenarios:** | | |
| **4a:** A person is wearing a mask.<br>**4a:** Face of the person is not found in the database.<br>**4a**: An email will be sent only to the authorized personnel. | | |
| **Post Conditions** | | |
| **Step#** | **Description** | |
| **1.** | If a person is not wearing a mask, an email will be sent to the person violating the SOP's and to the authorized personnel. | |
| **2.** | Crowd is detected and an email is sent to the authorized personnel. | |
| **Use Case Cross referenced** | Not Applicable | |

Table 1: Use case COVID-19 SOP Enforcement System

# 5. Non-functional Requirements

## 5.1. Performance Requirements

- It should be fast enough to accurately predict every face that shows up in the frame.
- It should be able to handle the data stream of all the cameras.
- It should be reliable such that it records logs properly and makes predictions accurately.
- Graphs should be clear enough to aid in decision making.
- It should have an easy-to-use interface especially for non-tech users.
- The system should have the capability to notify the authorities and violators of the violations on time.

## 5.2. Safety Requirements

The system should be strong enough so that it does not crash mid operation.

## 5.3. Security Requirements

The connection with the database must be secure and the data should not be misused

## 5.4. User Documentation

- SDS (Software Design Specification Document)
- User Manual

# 6. References

[1] K. Lin, H. Zhao, J. Lv, C. Li, X. Liu, R. Chen and R. Zhao, "Face Detection and Segmentation Based on Improved Mask R-CNN," Discrete Dynamics in Nature and Society, Hindwai, Gunangzhou, China, 2020, pp. 1-11.

[2] M. Loey , G. Manogaran , M. H. . N. Taha and N. E. M. Khalifa, "A hybrid deep transfer learning model with machine learning methods for face mask detection in the era of the COVID-19 pandemic," *Measurement,* 2020.

[3] M. Jiang, X. Fan and H. Yan, "RetinaMask: A Face Mask detector," arXiv preprint arXiv:2005.03950, Hong Kong, 8 May 2020.

[4] S. Ge, J. Li, Q. Ye and Z. Luo, "Detecting Masked Faces in the Wild with LLE-CNNs," IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Honolulu, HI, 2017, pp. 426-434.

[5] Y. Gu, S. Shen, J. Wang and J. Kim, "Application of NoSQL database MongoDB," *2015 IEEE International Conference on Consumer Electronics - Taiwan*, Taipei, 2015, pp. 158-159, doi: 10.1109/ICCE-TW.2015.7216831.

[6] I. Culjak, D. Abram, T. Pribanic, H. Dzapo and M. Cifrek, "A brief introduction to OpenCV," *2012 Proceedings of the 35th International Convention MIPRO*, Opatija, 2012, pp. 1725-1730.

# 7. Appendices

- Important SOPs

  http://covid.gov.pk/guideline

- Mask detection

  https://www.pyimagesearch.com/2020/05/04/covid-19-face-mask-detector-with-opencv-keras-tensorflow-and-deep-learning/

- Facial Landmark detection

  https://www.pyimagesearch.com/2017/04/03/facial-landmarks-dlib-opencv-python/