Low Level Design (LLD)

MASK WEAR DETECTION PROJECT

Revision Number: 1.8

Last date of revision: 05/09/2021

SOMAY

# Document Version Control

|  |  |  |  |
| --- | --- | --- | --- |
| Date Issued | Version | Description | Author |
| 18th May 2020 | 1.1 | First Draft | SOMAY |
| 20th May 2020 | 1.2 | Added Workflow chart | SOMAY |
| 20th May 2020 | 1.3 | Added Exception Scenarios Overall, Constraints | SOMAY |
| 21st May 2021 | 1.4 | Added KPIs | SOMAY |
| 26th May 2021 | 1.5 | Added user I/O flowchart | SOMAY |
| 26th May 2021 | 1.6 | Added EHR, LSTM model diagrams | SOMAY |
| 31st May 2021 | 1.7 | Added dataset overview and updated user I/O flowchart. | SOMAY |
| 02nd June 2021 | 1.8 | Restructure and reformat LLD | SOMAY |
|  |  |  |  |

**Contents**

[Document Version Control 2](#_gjdgxs)

[**Abstract** 4](#_30j0zll)

[1](#_1fob9te) Introduction 5

[1.1](#_3znysh7) Why this Low-Level Design Document? 5

[1.2](#_2et92p0) Scope 6

[1.3](#_tyjcwt) Constraints 6

[1.4](#_3dy6vkm) 1.3 Risks 6

[1.5](#_1t3h5sf) 1.4 Out of Scope 7

[2](#_4d34og8) Technical specifications 7

[2.1 Predicting Disease 7](#_2s8eyo1)

[2.2 Logging 7](#_17dp8vu)

[2.3 Database 7](#_3rdcrjn)

[3](#_26in1rg) Technology stack 8

[4](#_lnxbz9) Proposed Solution 9

[5](#_35nkun2) Model training/validation workflow 10

[6](#_1ksv4uv) User I/O workflow 12

[7](#_44sinio) Exceptional scenarios 13

**Abstract**

The new Coronavirus illness (COVID-19) has seriously affected the globe. By the end

of Nov 2020, the worldwide variety of recent coronavirus cases had already exceeded sixty million and the variety of deaths one,410,378 in keeping with info from the globe Health Organization (WHO). To limit the unfold of the illness, necessary face-mask rules area unit currently turning into common in public settings round the world. to boot, several public service suppliers need customers to wear face-masks in accordance with predefined rules (e.g., covering each mouth and nose) once using public services. These developments impressed analysis into automatic (computer-vision-based) techniques for face-mask detection that may facilitate monitor public behaviour and contribute towards constraining the COVID-19 pandemic. though existing analysis during this space resulted in economical techniques for face-mask detection, these typically operate beneath the idea that trendy face detectors offer excellent detection performance (even for covert faces) which the most goal of the techniques is to observe the presence of face-masks solely

# Introduction

## Why this Low-Level Design Document?

The purpose of this document is to present a detailed description of the Deep EHR System. It will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react to external stimuli. This document is intended for both the stakeholders and the developers of the system and will be proposed to the higher management for its approval.

This project shall be delivered in two phases:

Phase 1: All the functionalities with PyPi packages.

Phase2: Integration of UI to all the functionalities.

## Scope

1. The HLD documentation presents the structure of the system, such as the database architecture, application architecture (layers), application flow (Navigation), and technology architecture. The HLD uses non-technical to mildly-technical terms which should be understandable to the administrators of the system.

## Constraints

We will only be selecting a few of the chronic diseases.

## Risks

Document specific risks that have been identified or that should be considered.

## Out of Scope

Delineate specific activities, capabilities, and items that are out of scope for the project.

# Technical specifications

## 2.1 Dataset

## 2.1.1 Mask Wear DataSet overview

Data requirement completely depend on our problem statement.

* + - We need images data that is balanced and must have at least 1000 images.
    - We require at least 30- 40 images for each class label with annotation.
    - An image is nothing more than a two-dimensional array of numbers(pixels)
    - Pixel value ranging between 0 to 255
    - It is defined by the mathematical function (x, y), the value off (x, y) at any point is giving the pixel value at that point of an image
    - Original image is in the format of (width, height, no of RGB channels).

There are numerous image file formats out there so it can be hard to know which file type best suits your image needs (on your requirement).

a TIFF — Tagged image file format o BMP — Bitmap image file form

a JPEG - Joint photographic experts’ groups o GIF - graphics interchange format

a PNG — portable network graphics a EPS — encapsulated post script a RAW image files

* + - Tiffs are great for printing. These are lossless image files meaning they don’t need to compress or lose any image quality or information. These format images are high quality images.
    - bmp format developed by Microsoft for windows. There is no compression or information loss; this format is generally recommended for high quality scans.
    - JPEG is a lossy format meaning that the image is compressed to make a smaller file but this loss is not noticeable.
    - JPEG is a very popular format for digital cameras.
    - GIFs are widely used for web graphics because they are limited to only 256 colors, can allow for transparency and can be animated. These types of files are typically small in size and very portable.
    - PNG are a lossless image format; these files are able to handle up to 16 million colors unlike the 256 colors supported by GIF.
    - EPS is a common vector type file.
    - RAW images that are unprocessed that have been created by a camera or scanner. Digital cameras can shoot in raw, mostly used in photography.

If the data is in video format like (MP4) convert into images based on FPS (no. of frames displayed per second) in real time processing. There are number of tools to convert videos into images. Using cv we can convert video into images

## 2.1.2 Input schema

## 

## 2.3 Logging

We should be able to log every activity done by the user.

* The System identifies at what step logging required
* The System should be able to log each and every system flow.
* Developers can choose logging methods. You can choose database logging/ File logging as well.
* System should not be hung even after using so many loggings. Logging just because we can easily debug issues so logging is mandatory to do.

## 2.4 Database

System needs to store every request into the database and we need to store it in such a way that it is easy to retrain the model as well.

1. The User chooses the disease.

2. The User gives required information.

3. The system stores each and every data given by the user or received on request to the database. Database you can choose your own choice whether MongoDB/ MySQL.

**2.5 Deployment**

1. AWS



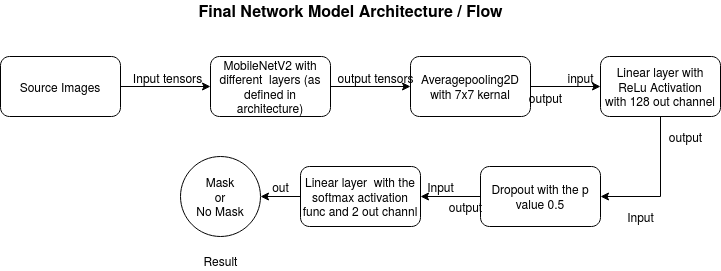
# Technology stack

|  |  |
| --- | --- |
| **Front End** | HTMl |
| **Backend** | Python |
| **Database** | MongoDB |
| **Deployment** | AWS |

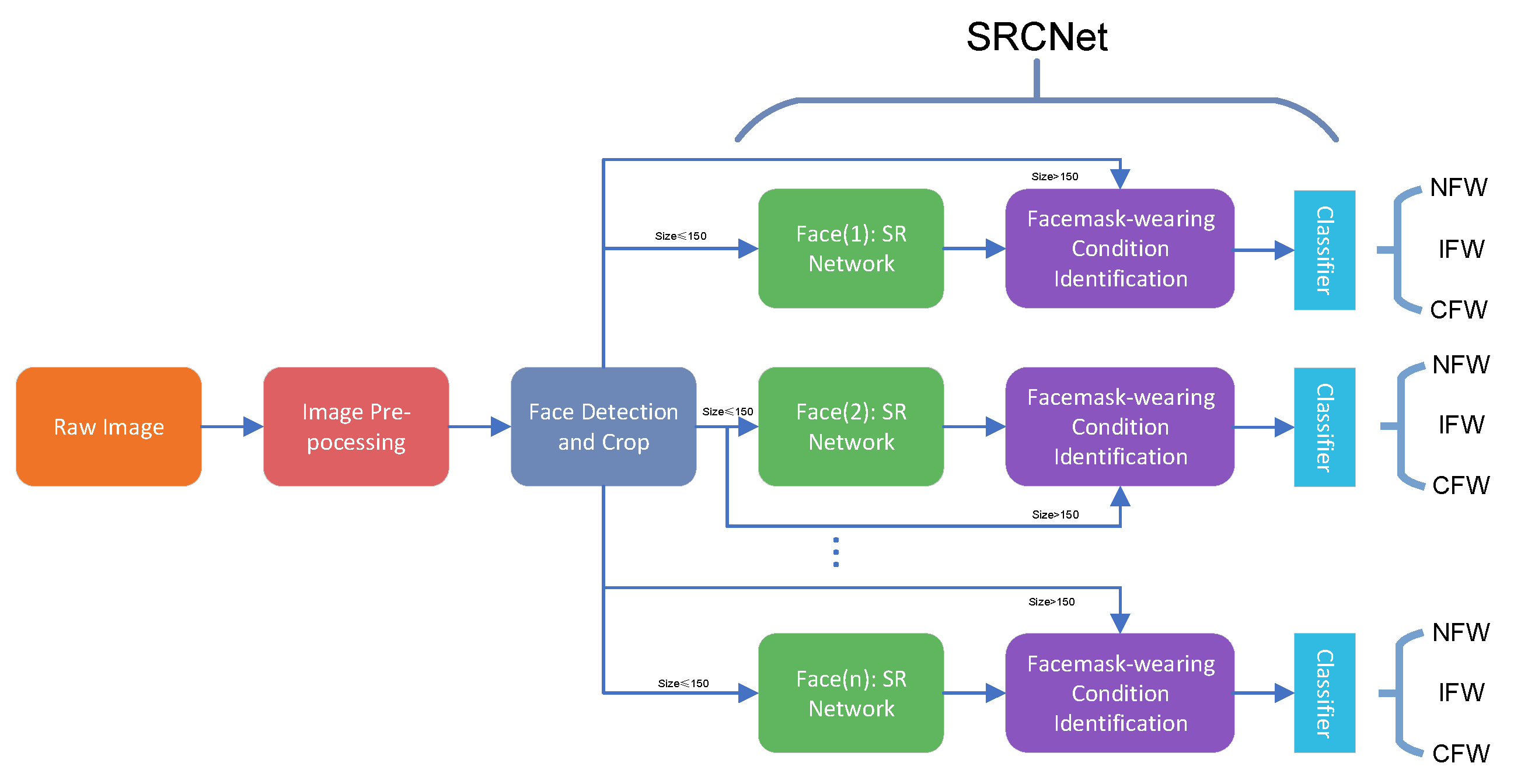
# Proposed Solution

1. The solution proposed here is a Mask Wear Detection can be implemented to perform use cases and to prevent the covid. if MWD detects Person wear mask or not. If wear person the mask, then he is entering the office gate otherwise person didn’t wear the mask then he or she is not entering the office.

# Model training/validation workflow



# User I/O workflow



# Exceptional scenarios

|  |  |  |  |
| --- | --- | --- | --- |
| Step | Exception | Mitigation | Module |
| 18th May 2020 | 1.1 | First Draft | Somay |
| 20th May 2020 | 1.2 | Added Workflow chart | Somay |

# Test cases

|  |  |  |  |
| --- | --- | --- | --- |
| Test case | Steps to perform test case | Module | Pass/Fail |
|  |  |  |  |

# Key performance indicators (KPI)

* Time reduction using the MWD model.
* Comparison of accuracy of MASK WEAR DETECTION
* Number of times a person visits the public places
* Immunity of person.
* Vaccines the persont has taken.