

Software Architecture Design Document (SADD)

Face Mask Wear Detection

Revision Number: 1.0

Last date of revision: 26/06/2022

Architecture Document

Shashank Singh

DOCUMENT VERSION CONTROL

Date Issued	version	Description	Developed by
26/06/2022	1.0	First Draft	Shashank Singh

Table of Contents

Document Version Control	2
Abstract	4
1 Error! Bookmark not defined.	
1.1 Error! Bookmark not defined.	
1.2 Error! Bookmark not defined.	
1.3 Error! Bookmark not defined.	
1.4 Error! Bookmark not defined.	
1.5 Error! Bookmark not defined.	
2 Error! Bookmark not defined.	6
2.1 Face mask wear detector (data set)	
2.1.1 Dataset overview	
2.1.2 Input schema	
2.2 Predicting mask wear or not	8
2.3 Logging	
3 Error! Bookmark not defined.	9
4 Error! Bookmark not defined.	10
5 Error! Bookmark not defined.	11
6 Error! Bookmark not defined.	
7 Error! Bookmark not defined.	12
Error! Bookmark not defined.	12

Architecture Document

Abstract

COVID-19 pandemic has rapidly affected our day-to-day life disrupting the world trade and movements. Wearing a protective face mask has become a new normal. In the nearfuture, many public service providers will ask the customers to wear masks correctly to avail of their services. Therefore, face mask detection has become a crucial task to help global society.

1 Introduction

1.1 Why this Low-Level Architecture Design Document?

The Purpose of this document is to present a detailed of the Face mask wear detection system, A software architecture document is a map of the software. We use it **to see, at a glance, how the software is structured**. It helps you understand the software's modules and components without digging into the code. It's a tool to communicate with others—developers and non-developers—about the software.

Whenever possible, we make use of existing technology instead of reinventing the wheel and the usability of the system is taken into account as the #1 priority. The structure that the rest of the document will follow is: 1. A summarized description of the software architecture, including major components and their interactions. 2. Architectural constraints and decisions. 3. A detailed description of each component. 4. System functionality represented by use cases. 5. An outline description of the hardware and software platforms on which the system has been tested so far. Also, where preliminary tests and analysis show they could initially be deployed into. 6. A guide on how to run test cases of the application

1.2 Scope

The project team and developers are working intently on making this project a fully operational and viable technology system for the public market. The vital component was developed some time ago and it's in a quite mature state, but the project as a product is just ending its incubation state to begin its path to become a startup. As any project segues into a product, it is naturally lacking multiple components necessary to be successful in a competitive market, like documentation or project management and schedule.

This document applies to face mask wear detection system Web based application, which is still in the requirements gathering stage, thus making it hard if not impossible to take it into account in this first release of the document. Also, only a representative selection of use cases and tests is presented and no flow charts are provided due to the lack of previous documents like entities model and insufficient formal and in-depth requirements gathering and discussion about expectations on the system. In addition to that, the developer hasn't had the chance to talk to the other in-house developer to discuss general information about the original component of the system and any decisions made.

Thus, it is highly encouraged at least one future revision of the document to correct inaccuracies, extend it and insert additional feedback and decisions that were made in the past.

With these considerations in mind that you are going to read whole document, so this document has been made as accurate as possible, and as any developer-made artifact, it naturally contains a few, evident jokes to make the reader smile.

1.3 Constraints

The Face mask wear detector solution system must be user friendly, as automated as possible and user should not be required to know any of workings.

1.4 Risk

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

1.5 Out of scope

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

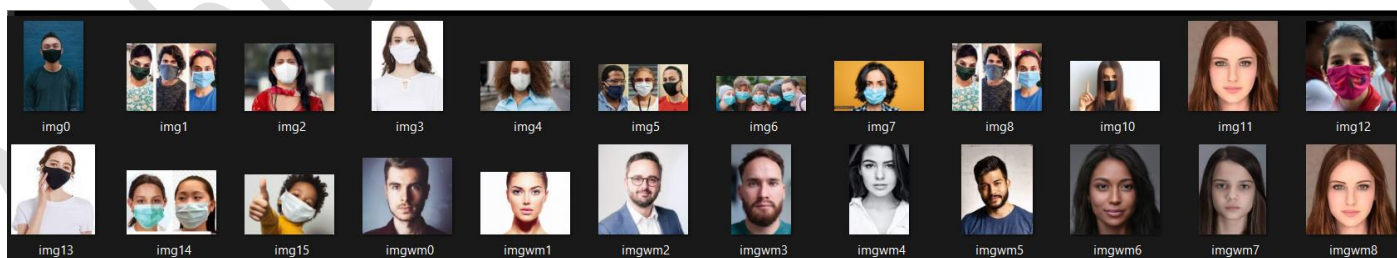
2 Technical Specification

2.1 Data set

Output	Finalized
Face detection (image)	Yes
Real time Face detection	yes

2.1.1 Dataset Overview

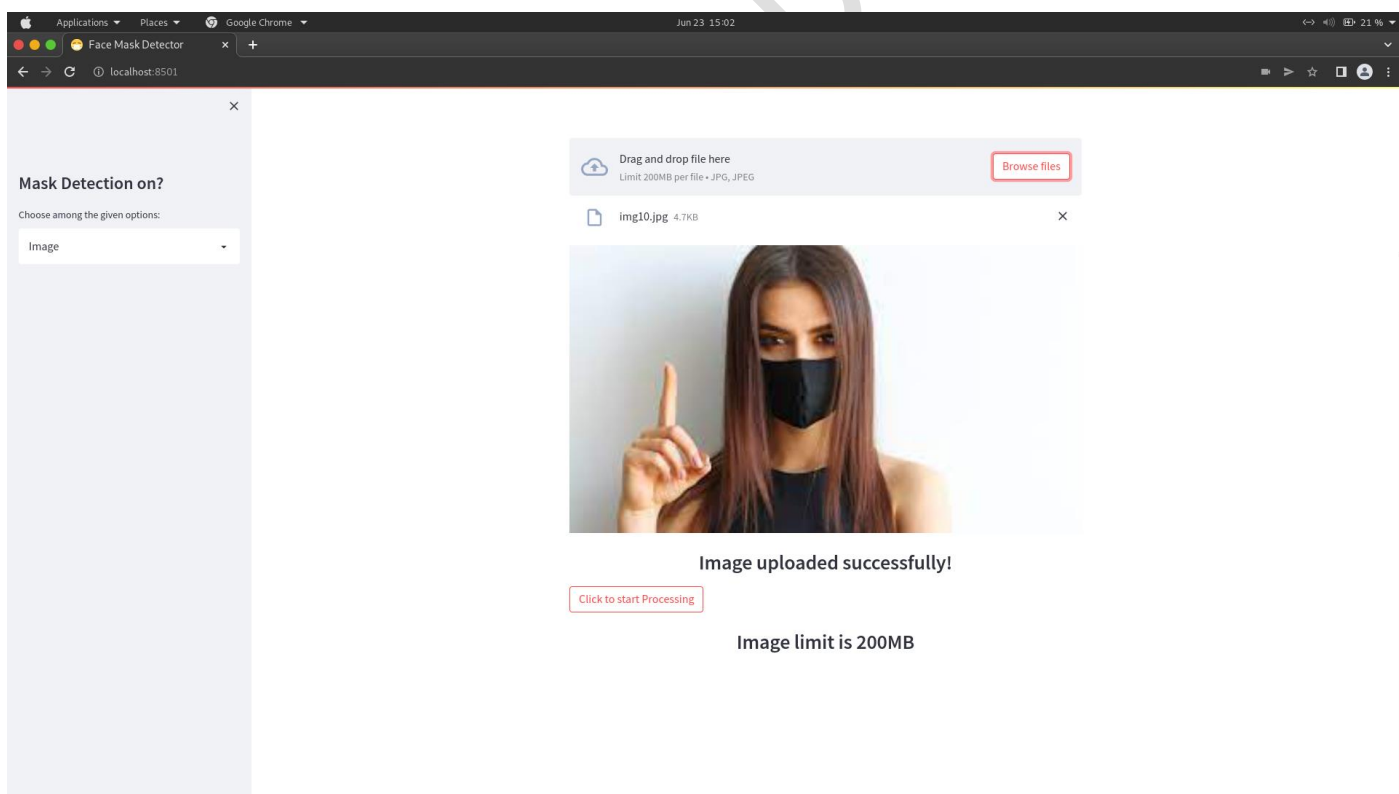
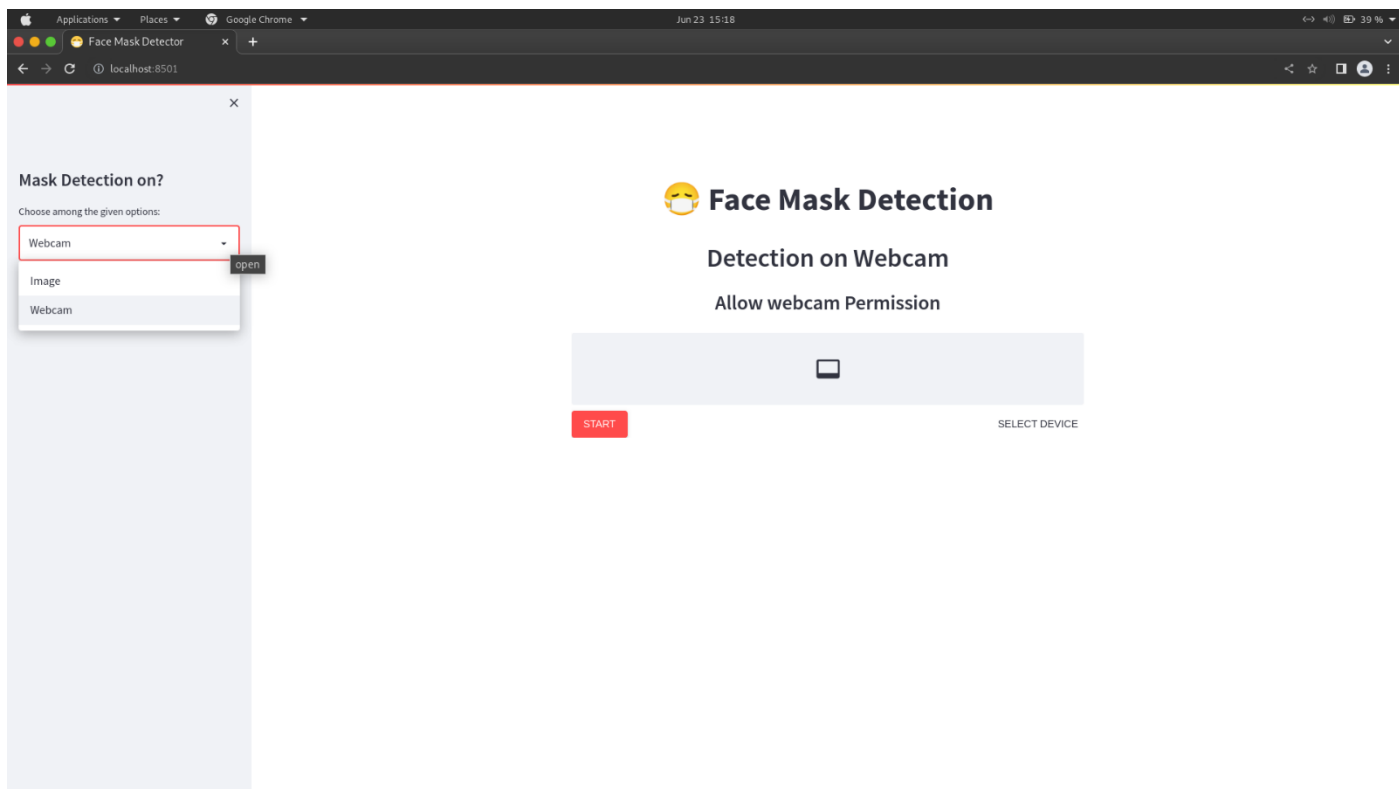
Image insertion into the input field where the image is in jpg format and is identifying whether person is wearing a mask or not .

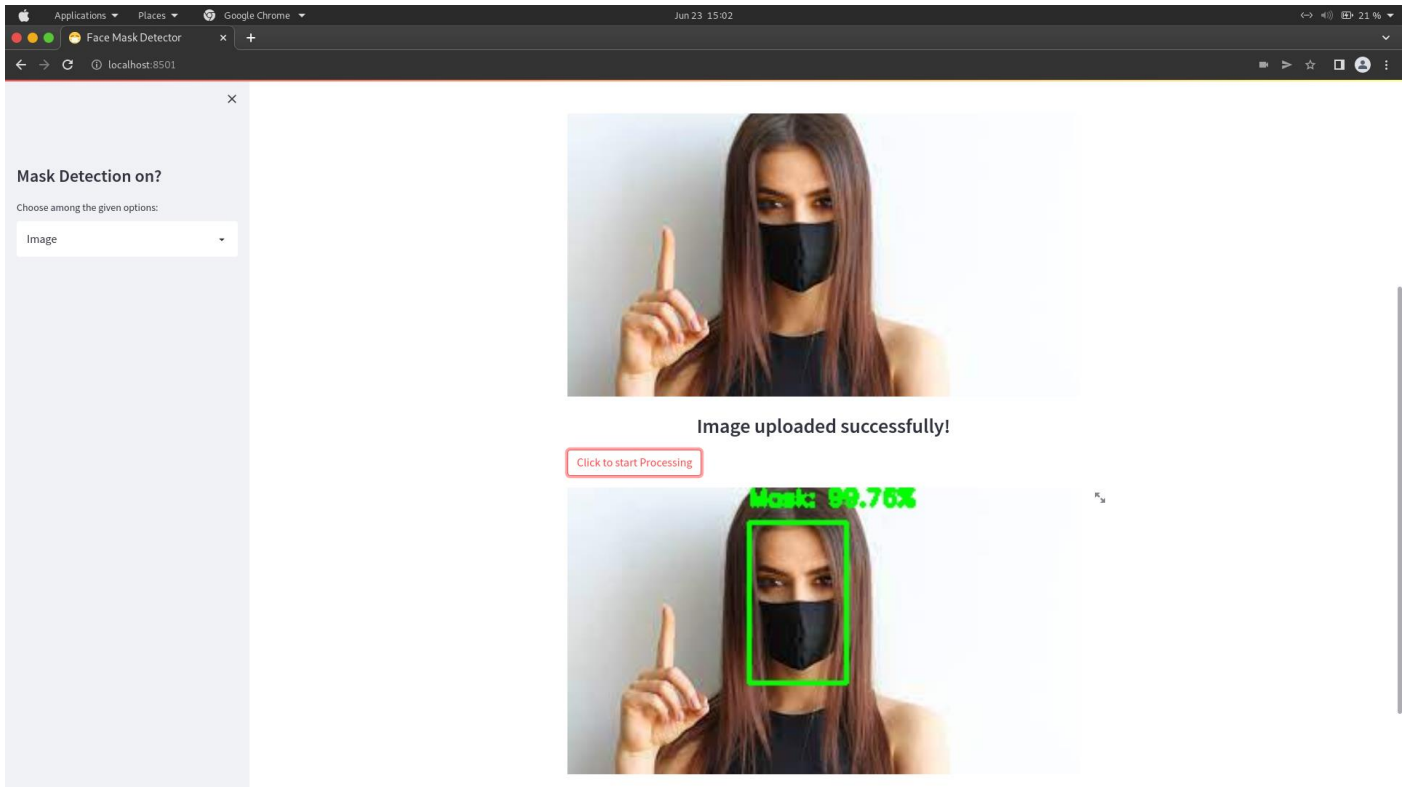


2.1.2 Input schema

*Note

image should be in jpg format only





2.2 Predicting Image

- System is display option for image masked face detection or real time face mask detection.
- The User chooses the target disease by clicking one of the available.
- The user gives required information.
- The system should able to predict whether the (image, video) wear a mask.

2.3 Logging

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

- The System identifies at what step logging required (Streamlit).
- 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.

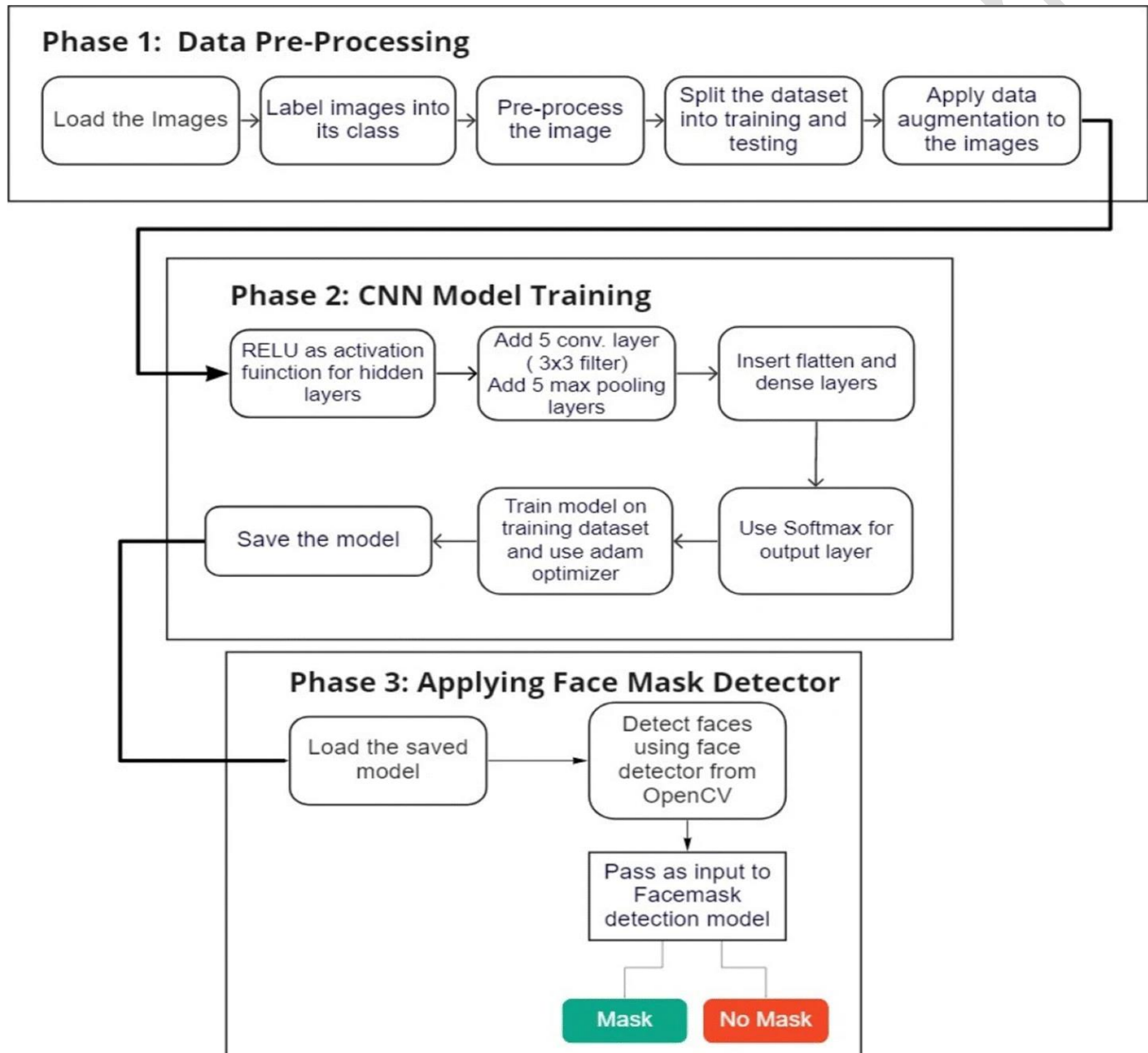
3 Technology Stack

Front End	HTML/CSS/Markdown
Back End	Streamlit (Python JavaScript Transcript)

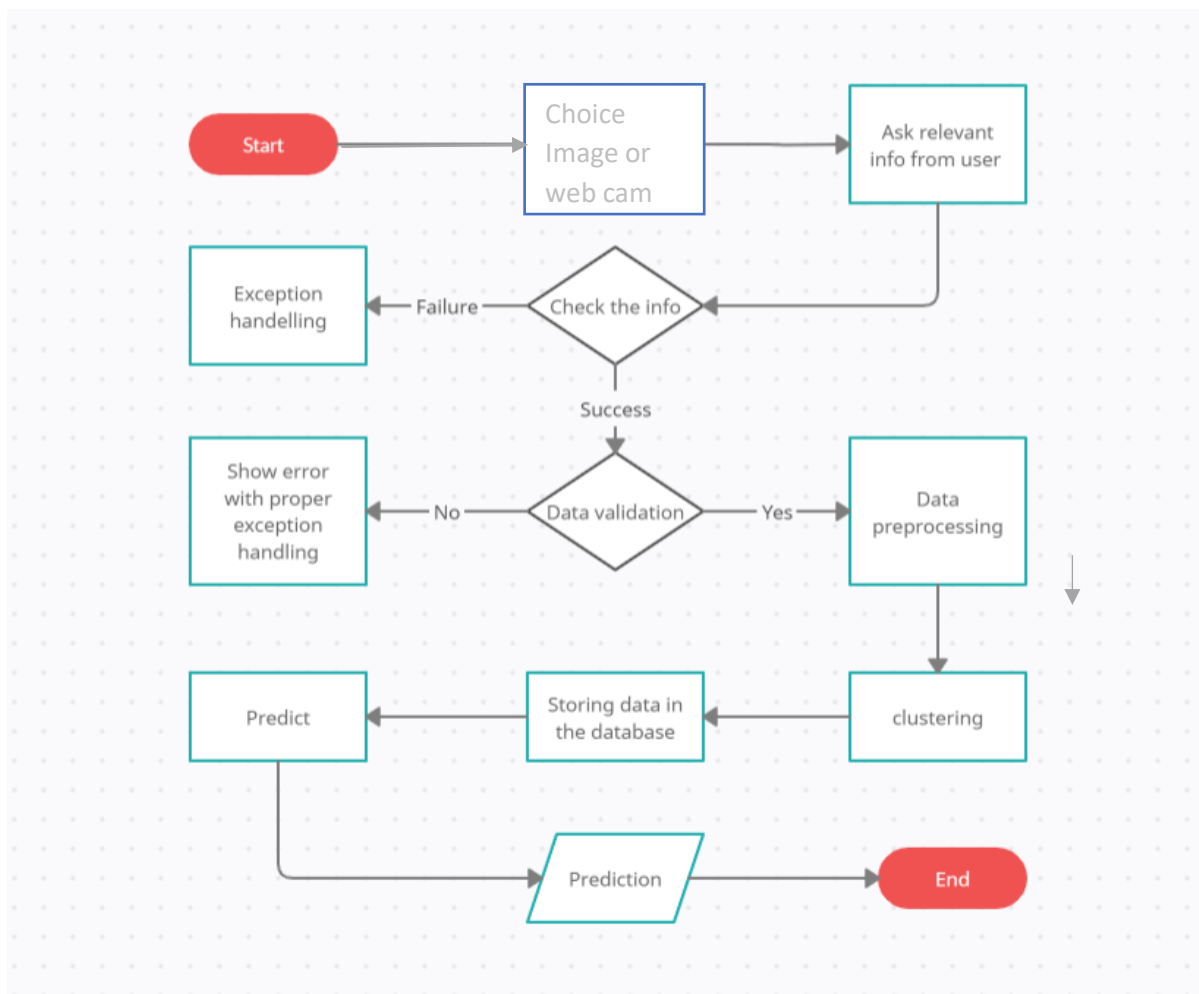
4 Proposed Solution

The solution proposed here is a face mask wear detection-based surveillance can prevent from above mention cases, if any person who don't wear a mask, then the person monitoring or police or any public safety department can take action immediately

5 Model Training validation workflow



6 User I/O workflow



7 Exceptional scenarios

Step	Exception	Mitigation	Module
26 th June 2022	1.0	First draft	Name

8 Test case

Test case	Step to perform test case	Module	Pass/fail
1 st	1.choice image/webcam 2. upload image /web cam 3. result	Image	Pass = mask 99.76%