HIGH LEVEL DOCUMENT

MASK DETECTION

Written By	Mohammad Sohail Parvez
Document Version	0.1
Last Revision Date	

Content

Abstract

For this project, we examine the “Mask Detection” dataset taken from Roboflow Universe. Our goal is to detect the people who are wearing mask or not.

1. Introduction

1.1 Why this High-Level Design Document?

The purpose of this High-Level Design (HLD) Document is to add the important details about this project. Through this HLD Document, I’m going to describe every small and big thing about this project.

2.General Description

2.1 Product Perspective

Computer vision based project using Supervised Learing Techniques .

2.2 Problem statement

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 near future, 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. COVID-19 mask detector could potentially be used to help ensure your safety and the safety of others

2.3 Proposed Solution

The dataset does not contain any more preprocessing. We have to encode the data to numerical and pass it to the model.

2.4 Technical Requirements

In this project the requirements to detect mask detection various platforms. For that, in this project we are going to use different technologies. Here are some requirements for this project.

- Model should be exposed through API or User Interface, so that anyone can test model.
- Model should be deployed on the cloud..

2.5 Data Requirements

The dataset is downloaded from Roboflow Universe.

```
mask_detection
  - train
        - images
        - labels
  - test
        - images
        - labels
  - valid
        - images
        - labels
  - data.yaml
```

2.6 Tools Used

1. Python
2. Tensorflow
3. Yolov5
4. Flask
5. Roboflow Universe
6. Google Colab
7. Vs Code
8. Github
9. Docker

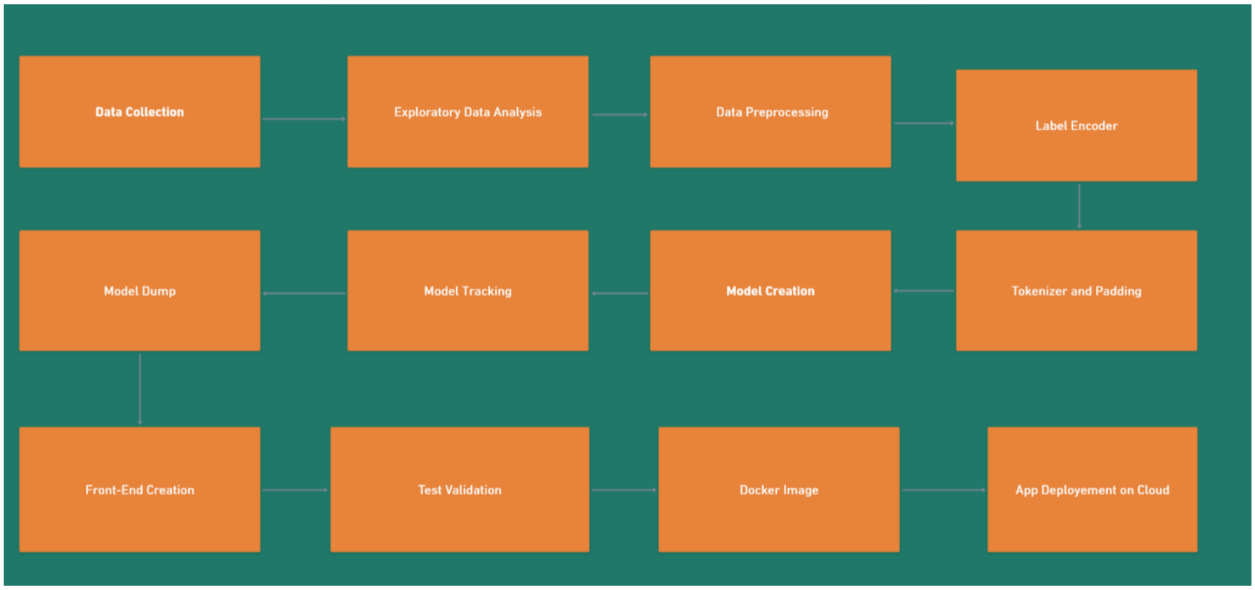
10. AWS
11. Anaconda

2.7 Constraints

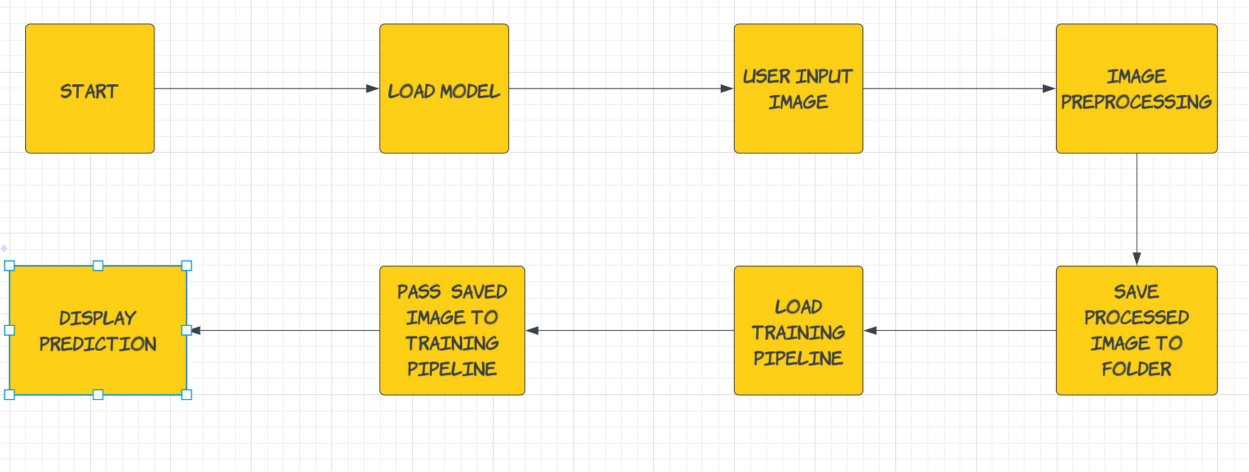
The mask detect system must be user friendly, error free and users should not be required to know any of the back-end working

3 Design Details

3.1 Process Flow



3.2 Deployment Process



3.3 Event Log

In this project we are logging every process so that the user will know what process is running internally.

Step-By-Step Description:

● In this project we defined logging for every function.

● By logging every function in the Exploratory Data Analysis.

● Then logs all the data preprocessing functions and code.

● Logs every model algorithm to our data.

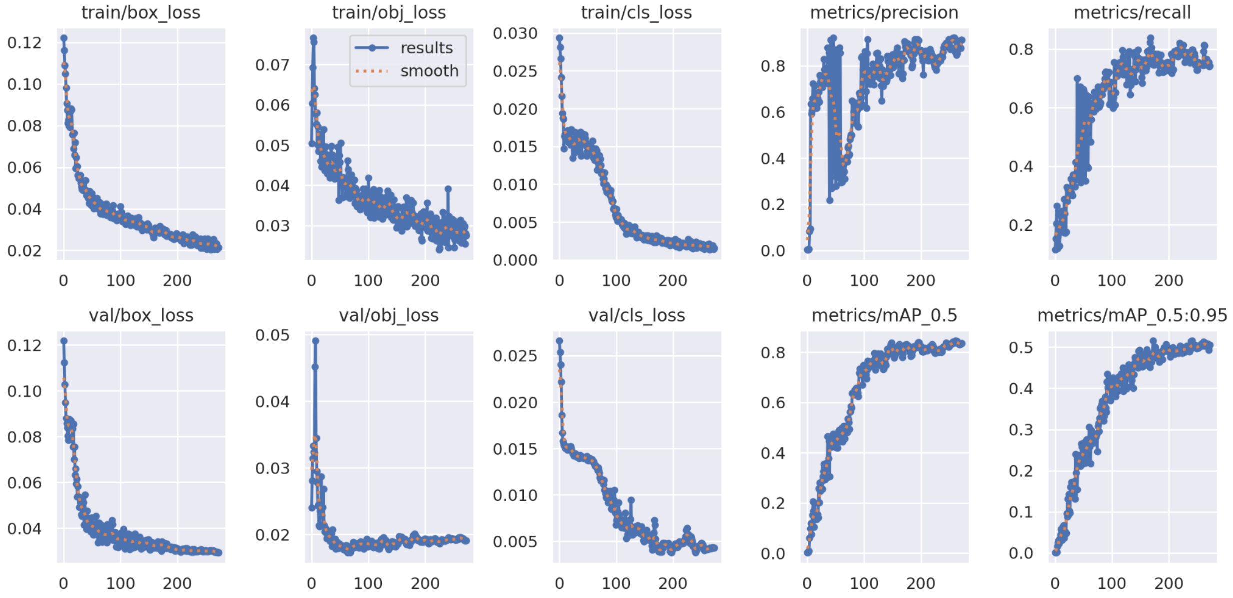
3.4 Error Handling

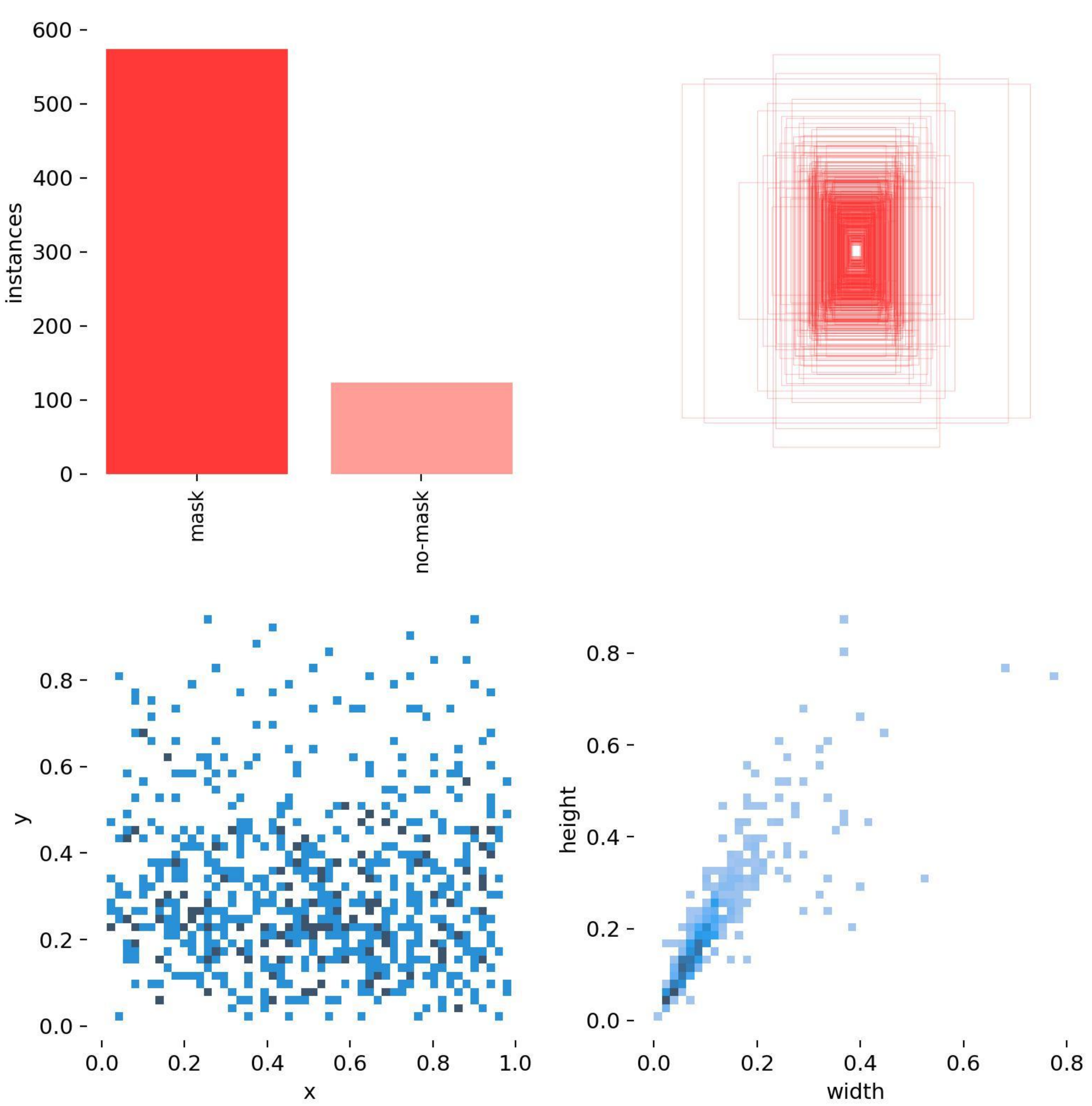
The project is designed in such a way that, at any step if error occurs then our application should not terminate rather it should catch the error and display that error with proper explanation as to what went wrong during process flow.

4. Performance

Solution of Machine Learning is used to detect in advance, so it should be as accurate a s possible so that it should give as much as possible accurate classification.

- We download the data from Roboflow and edited the data.yaml as per our need.
- We trained our model using YOLOv5 with edited configuration using data.yaml with 2 classes should be trained.
- The model is trained on 300 epochs with 82.9% mAP50 score





4.1 Re-usability

The programming is done in such a way for this project that it should be reusable. So that anyone can add and contribute without facing any problems.

4.2 Application Compatibility

The difference module of this project is using Python as an interface between them. Each module has its own job to perform and it is the job of the Python to ensure the proper transfer of information.

4.3 Deployment

We have deployed this on cloud and also dockerized this.

5. Conclusion

The Machine Learning model will detect the mask