

LOW LEVEL DOCUMENT

Mask Detection Prediction

Written By	Mohammad Sohail Parvez
Document Version	0.1
Last Revised Date	

Contents

1. Introduction.....	1	1.1
What is Low-Level Design Document?.....	1	1.2
Scope.....	1	
2. Architecture.....	2	3.
Architecture Description.....	3	3.1
Data Collection.....	3	3.2
Exploratory Data Analysis.....	3	3.3 Data
Preprocessing.....	3	3.4 Model
Creation.....	3	3.5 Model
Dump.....	3	3.6 Data from
User.....	3	3.7 Model Call/.pkl
file loaded.....	4	3.8
Docker.....	4	3.9
Deployment.....	4	4. Unit
Test Cases.....	5	

1. Introduction

1. 1 What is a Low-Level Design Document?

The goal of LLD or a low-level design document(LLDD) is to give the internal logical design of the actual program code for the adult income prediction estimation system. LLD describes the class diagrams with the methods and relations between classes and program specs. It describes the modules so that the programmer can directly code the program from the document.

1.2 Scope

Low-level design (LLD) is a component-level design process that follows a step-by-step refinement process. This process can be used for designing data structures, required software architecture, source code and ultimately, performance algorithms. Overall the data organization may be defined during requirement analysis and then refined during data design work.

2. Architecture



3. Architecture Description

3.1 Data Collection

The dataset is downloaded from Roboflow Universe.

3.2 Exploratory Data Analysis

No EDA is need for this project.

3. 3 Data Preprocessing

Annotating data and saving the images and labels for model training.

3. 4 Model Creation.

In model training we used YOLOv5

3.5 Model Dump

After comparing all accuracies and checking all regression metrics, The model is dumped used .pt format

3.6 Data From User

Here we will collect user's requirements to detect mask.

3.7 Model Call

Based on the User input, it will be thrown to the backend in the dictionary format.

3.8 Docker

Implemented docker in our project so that anyone can run our docker image which is uploaded as an open source on a docker hub. So it is useful for a developer to pull images from the docker and run on any platform.

3.9 Deployment

The model is deployed on Cloud

4. Unit Test Cases

Test Cases Description	Prerequisite Expected Result
Verify whether the Application URL is accessible to the user.	Application URL should be should be defined accessible to the user

Verify whether the Application loads completely for the user when the URL is accessed	The Application should accessible load completely for the Application is user when the URL is deployed accessed	
Verify whether the user is giving standard input.	Handled test cases .Users should be able toat backends. see successfully valid results.	
Verify whether user is able to see input fields	Application is User should be able to see accessible input fields	
Verify whether user is able to edit all input fields	2.Application is accessible. User is logged in to the application	User should be able to edit all input fields
Verify whether gets predict or classify news into categories	1. Application isaccessible. 2. User is logged in tothe application	Users should get the Submit button to submit the inputs.
Verify whether the user is presented with recommended results on clicking submit.	1.Application is accessible. 2.User is logged in to the application	User should be presented with recommended results on clicking submit
Verify whether the recommended results are in accordance to the selections user made	1.Application is accessible. 2.User is logged in to the application	The recommended results should be in accordance to the selections user made

Verify whether user has options to filter the recommended results as well	<ol style="list-style-type: none">1. Application is accessible2. User is logged in to the application	User should have options to filter the recommended results as well.
---	--	---