Low Level Design (LLD)

Brand Recognition

WRITTEN BY
SADASHIV NANDANIKAR

Revision Number: 1.0

Last data of revision: 27/04/2023

Contents

1. Introduction

1.1 What is Low-Level design document?

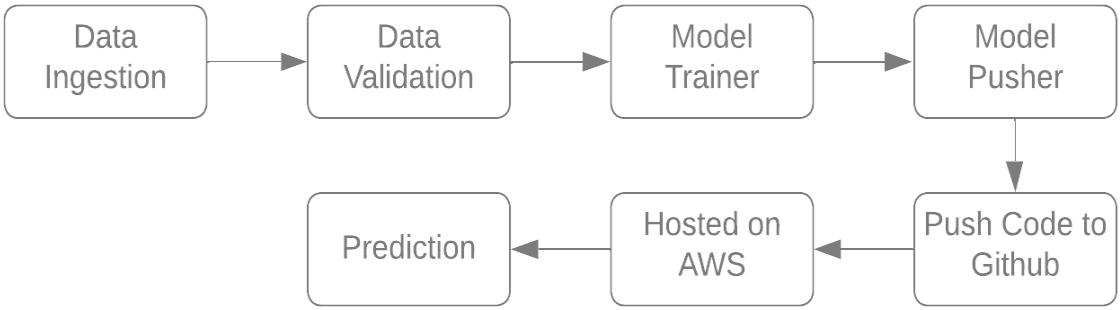
A low-level design document (LLD) is a detailed technical document that describes how a software system or application will be implemented at a low level. The purpose of an LLD is to provide a blueprint for developers and stakeholders to understand the technical details of the system, including the architecture, components, modules, interfaces, algorithms, data structures, and other implementation details.

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

2.1 Project Architecture:



3. Architecture Description

3.1 Data Gathering:

To collect data/images for the current project, web scraping techniques are being employed.

3.2 Data Description:

The data used in this project pertains to the ice cream brands Amul, Mother Dairy, and Kwality Walls. The data has been collected through web scraping techniques and in various contexts. To ensure a balanced dataset, appropriate measures have been taken. Annotation of the collected images has been performed using the LableImg tool.

3.3 Tool Used:

- The environment was created using Python 3.8.10.
- VS Code is used as IDE.
- GitHub serves as the code repository.
- To host the application, an instance of EC2 from the AWS cloud platform is being utilized.

Python programming language and other frameworks such as NumPy, PyTorch, YoloV8, labellmg are used in building the whole model.

3.4 Data Validation:

The Data Validation process ensures that the directory structure required for the YOLO algorithm is correct. Once the validation check is completed, a validation status file is generated and stored in the 'data_validation' artifact directory.

3.5 Model Building:

For this project, the YOLO v8 algorithm is utilized for both training and prediction purposes in model building.

3.6 Model Pusher:

The best model is uploaded to S3 bucket.

3.7 Prediction:

Flask web application takes user input images as input, which are pre-processed using the trained model to identify and label objects within the image. The prediction results are then displayed on the screen as output, with bounding boxes indicating the location and class of the detected objects.

3.8 Deployment:

This Model is deployed using an EC2 instance from the AWS cloud platform.