**Adani Wilmar Project**

**Report**

**Chaitanya Sharma**

**Vellore Institute of Technology,**

**Bhopal**

1. OBJECTIVE :-

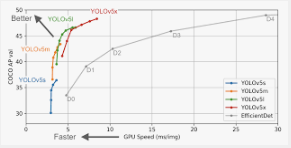
The company wants to get the data and in order to give services to the customers. This business problem was encountered and solved. The data pipeline to collect pictures ,recognize the company product ,get details which product and how many are there , also to get the product authenticated by a system.

2.0 Methodology:-

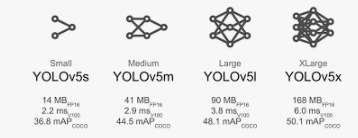
2.1 Literature Review :-

2.1.1 Research:-

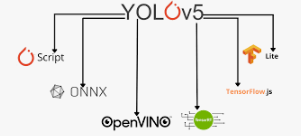
Firstly the detection part of the system was based on the YOLOv5 model which is the similar model of today’s YOLOv4 actual model based on the darknet library written and coded in C language. Pytorch provided the YOLOv5 model a sophisticated model for custom training and accurate detection in python language that can be used with a backend.



There are different neural network versions for every kind of problem



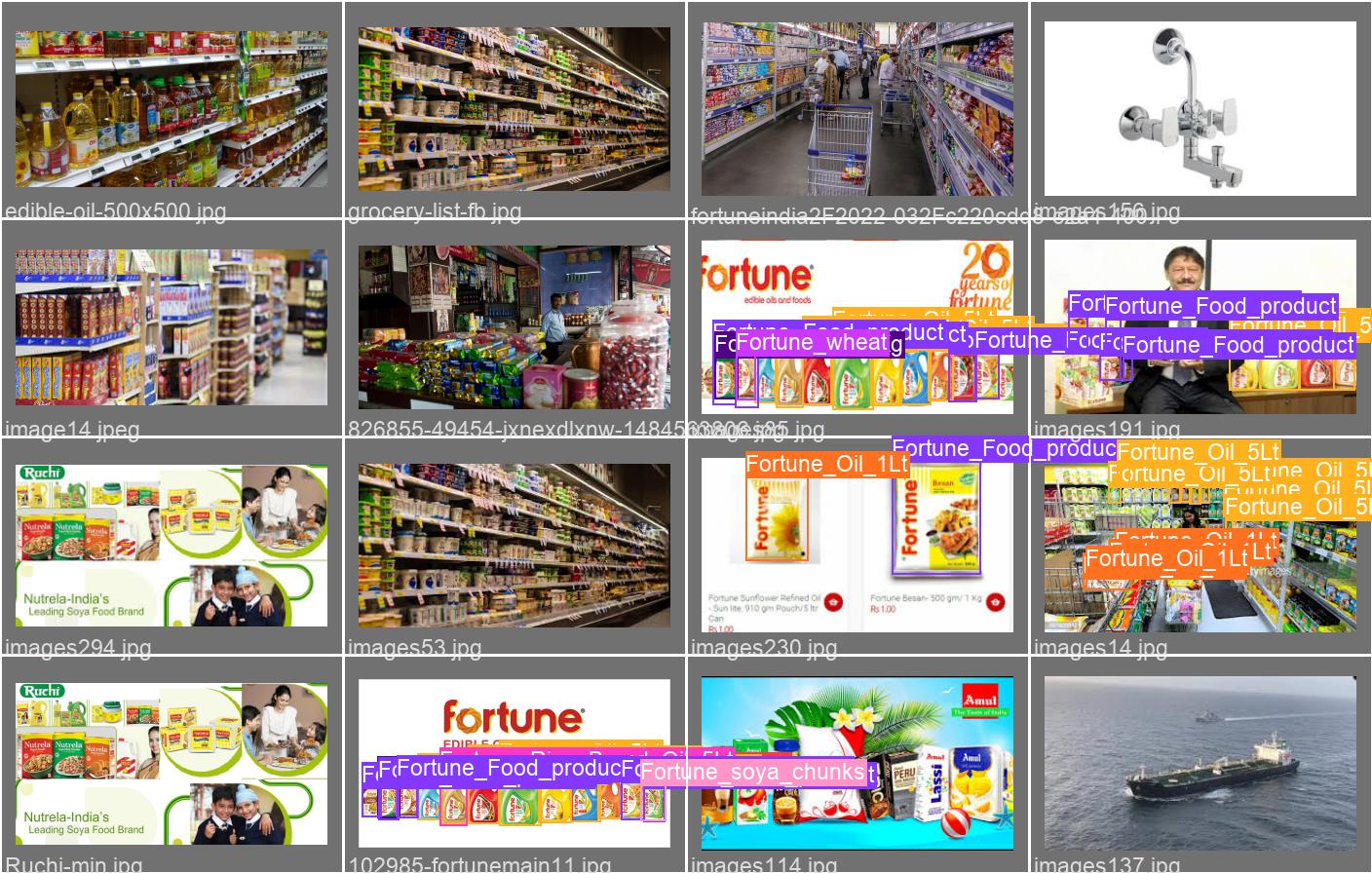
Available on every platform



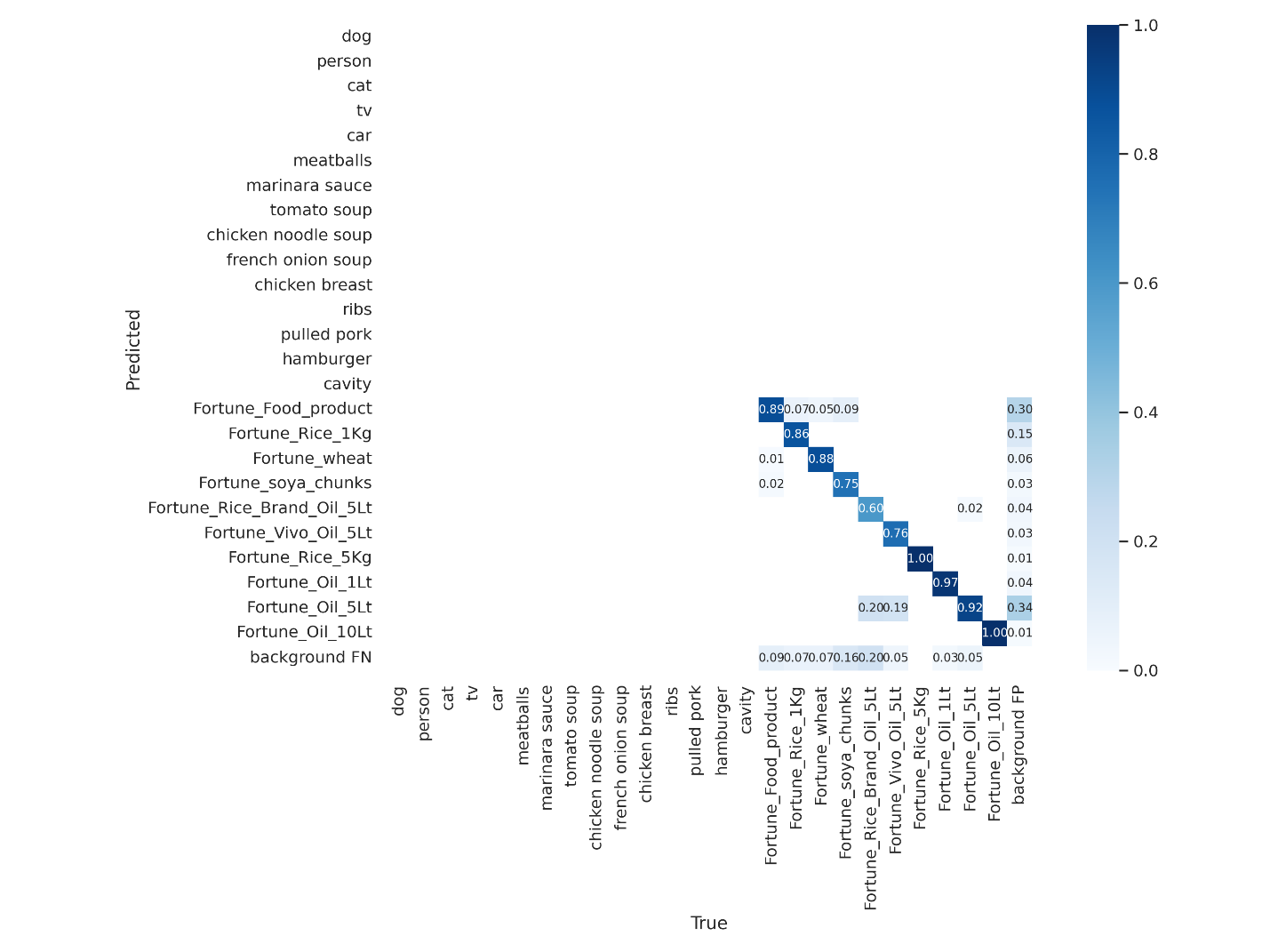
So, this was the best and easy to use solution of the machine learning part.

2.1.2 Experimentation and results:-

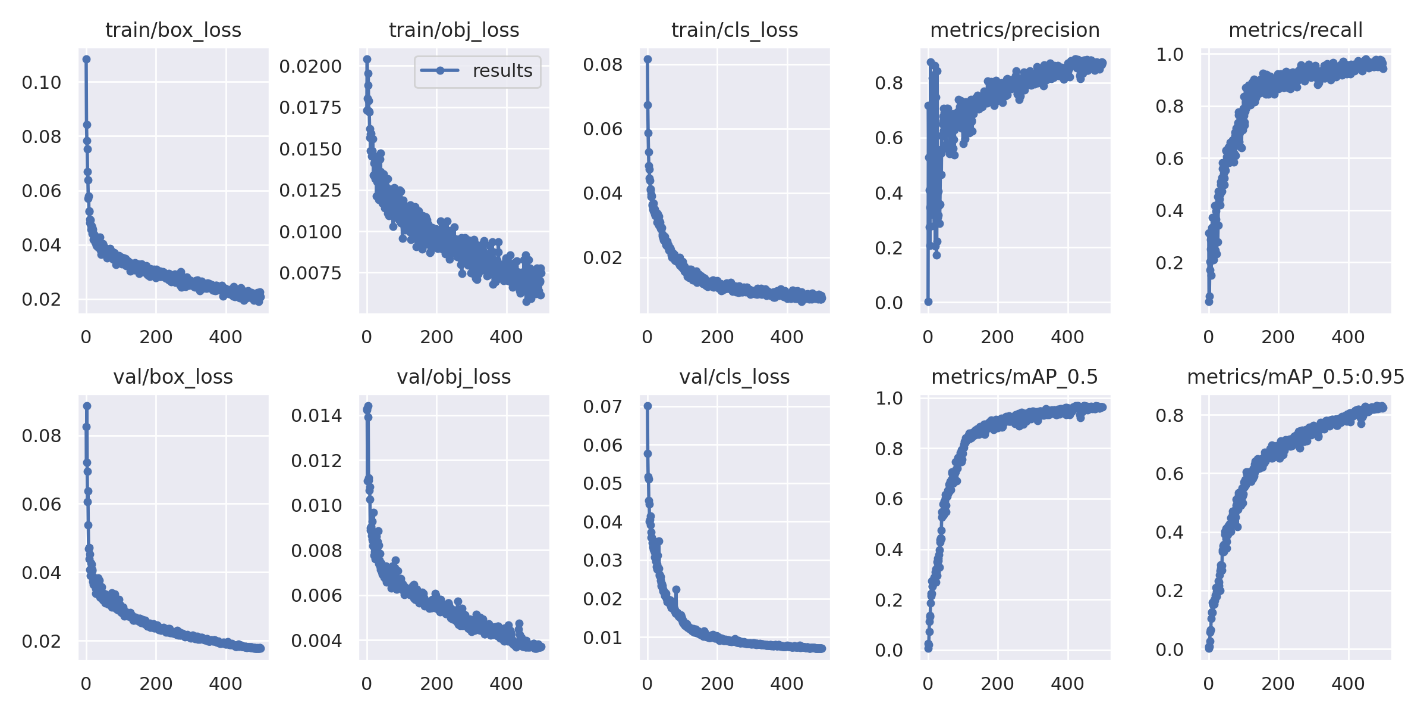
We used the multiclass classification model the yolov5s neural network. And Scrapped the images using a web scrapper to create the training and testing data sets. How to start the training testing and the labeling of the images the full package is provided in the final\_project folder in “Imglabel” which will help to label the images and as for the training the model ,there demos and the written functions which can be directly be used to scan , detect, train the model inside the helper\_fuction folder.



We labeled the images created a dataset for the custom model training. Trained the model on 500 epochs to get the result. It’s giving 87 to 97 percent accuracy rate in the results for every class refer to the confusion matrix.



The results were great the neural network is working extra ordinary amazing.



3.0 Solution :-

The solution we used to solve this problem is to make an API in FLASK which is a small web framework for python. The Idea is to implement the YOLOv5 algorithm to detect the product in the image from an flutter based app and a database to store the data.

Documentation

1. API (Overview ):-

The Adani Wilmar API can be found in the BACK-END folder the project and there are two endpoints available for the data processing which are “/api” and “/text” .The API is based on flask framework which is a small frame work for the web applications in python.

* 1. Setup :-

The BACK-END folder contains the venv a virtual environment

For the API to work. Open CMD or Command Prompt and route to the BACK-END folder then follow the command lines to enter in virtual environment:



The venv will start and then we need to download all the dependencies and tools in the venv in order for the API to work.

Install all the requirements from requirements.txt



And some following packages in the server:

1. Pip install pytesseract :- an library to detect characters in a image and converting those characters to text. We are using this to authenticate the pack image and get its Batch. No. with the Database.
2. Pip install flask\_pymongo :-A library for Flask web framework to communicate with MongoDB database.
3. Pip install pillow :- This is used to read image.
4. Pip install flask :- The Frame work on which the API is based we need to download and import it too.
5. Pip install matplotlib :- This library we are using to plot the results we get from the machine learning model.

1.2 Start :-

To start the API its very simple just need to do the following command lines :-

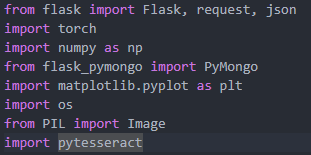






The server.py will load the API will start working.

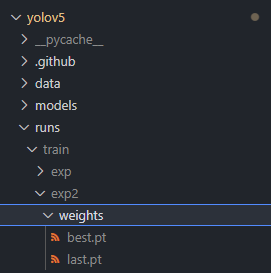
* 1. Imports:-



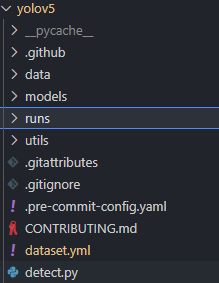
1. Firstly we are importing flask from Flask to set up the framework and importing requests to perform GET and POST requests. Json for the decoding and encoding of the data.
2. Import torch for the system to use machine learning model.
3. Importing NumPy as np , NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.
4. From the flask\_pymongo we installed earlier in the start now we are importing the PyMongo lib from it in order for the API to communicate with the Database.
5. Importing PIL , pytesseract and matplotlib.pyplot for the same reasons as explained in the start.
   1. End-Points :-
      1. For product detection “/api”:-



The working of the endpoints is quite simple and easy to understand. The end point receive a file from the form submission from user end. And at line 43 it checks that if the request contains a file or not if it contains then it proceeds to the line 44 loading the weight of the machine learning model and the YOLOv5 neural network using torch.hub.load(“this is the nueral network we want from the torch hub” , “type ,custom trained” , “the weights of the custom train model ” ). The weights of every experiment or every time we train the model are available in .pt format in yolov5 folder.



After loading the model we proceed to specify only the classes we need in the line 46. We give a array of the index of that particular class name present the dataset.yml file. Dataset.yml will contains all the requirements and paths to the train , val, labels list and when custom training the model give the paths in this file. It can be found in yolov5 folder.



Then the file we received in the request is stored in the static directory for the process. Then in line 48 we give path to the file and pass it in the neural network model. We store the output as results.

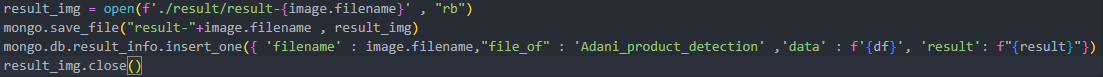


Using matplotlib we render the results in the form of an image , axis off and then we save it in the results directory.

In line 54 and 55 we are fetching the results in the form of pandas data frame , in line 55 we axis the name attribute which gives us information which index of detection has what labels and how many detections are there.



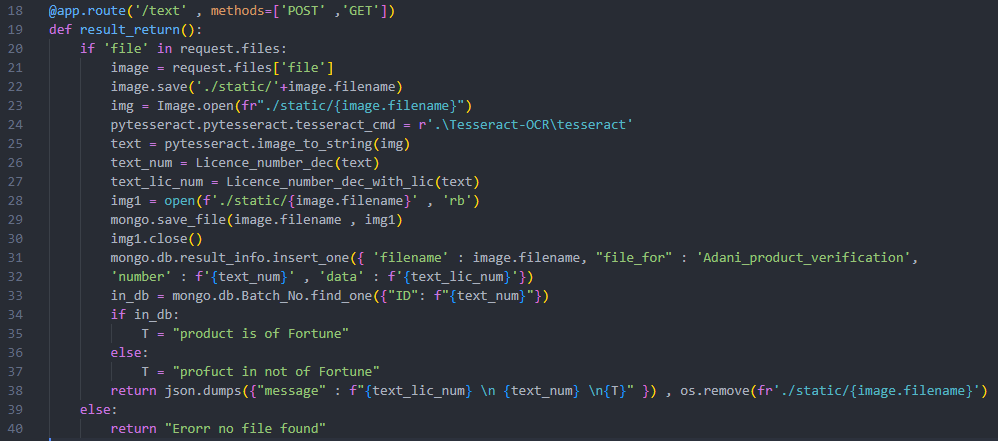
The we are opening and reading the result file in binary , then we sending the information and the data to the database. Then we are closing the file.



Lastly we are send the result to the user end and deleting the file from the static and the result folder in return.

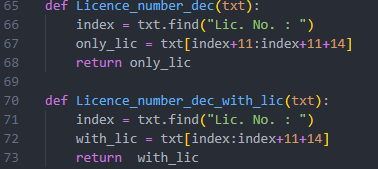


* + 1. For the Product Authentication “/text”:-



This is the endpoint where you have to POST the image of the back label details on the product pack this will return the product in authenticated or not.

The working is very simple and similar to the detect API endpoint. Check the request and if there is a file it save it in the static directory. Then open the image file and using pytesseract.pytesseract.tesseract\_cmd =r”specify the path to the tesseract application folder.” We using the tesseract OCR application in the code. Then using the function image\_to\_String(“passing the image”) and storing the string output in the text variable.

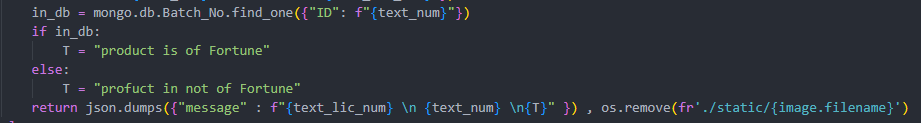


Using these two function which is just finding the “Lic. No. : ”and then Slicing it get only the required part from the text.

Similar way we did earlier in the /api endpoint we are opening the image sending it to the database with data



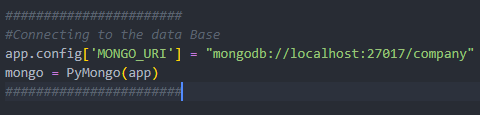
Then we are checking if the Lic no. we extracted is available in the database or not. If it is available then we are sending a message that “this is a Fortune product “ else “this is not a Fortune product”. And at the end we are deleting that file from the static folder.



2.0 Database :-

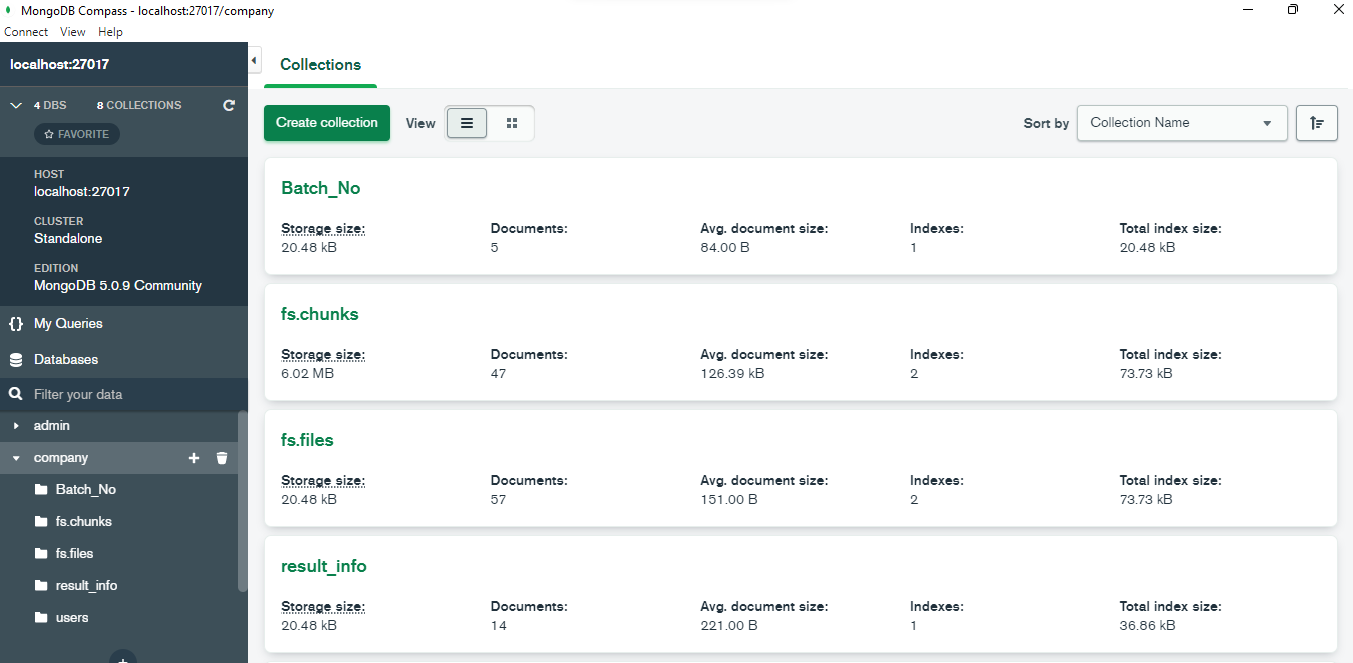
The Database used in the solution is MongoDB because its and unstructured database and can be used to store the images and data in it. Also it is easy to use and connect with Our flask api very easily.

2.1 Connection :-



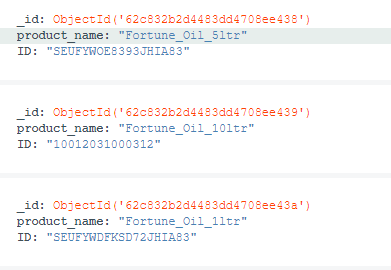
Here we are connecting to the database named “company” on the local host port 27017 which the default for the database.

2.2 Structure:-



Batch\_No :-

This has the data of all the products name , Object Id and ID. For the authentication of the product.



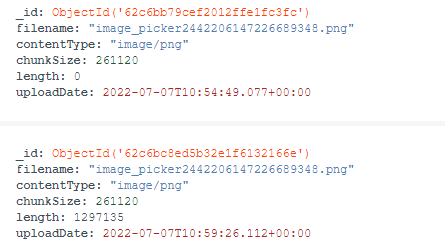
Fs.chunks:-

This stores the binary encoding of the files saved in the database. This files are in the form of chunks.



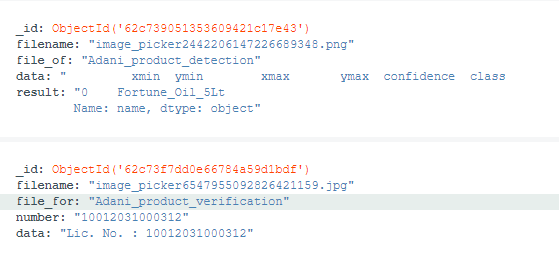
Fs.files:-

This contains the meta Data of the files stored in the database. Contains the name , object ID and the file name.



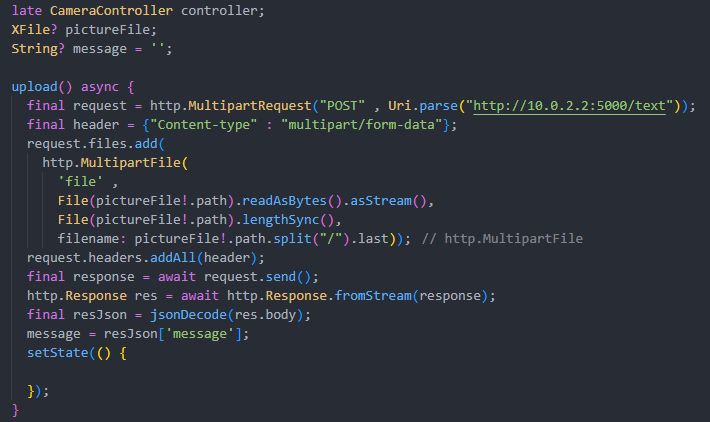
Result\_info:-

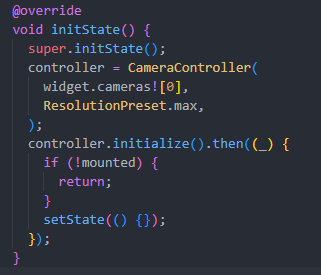
This contains all the data about the images , there purpose and the result data from the API.



3.0 Flutter App:-

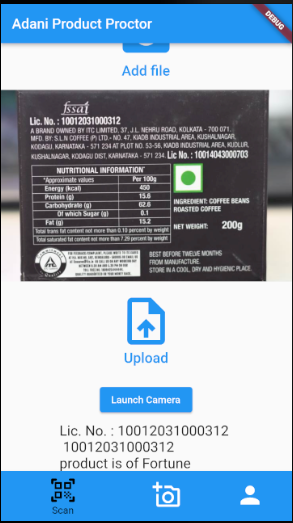
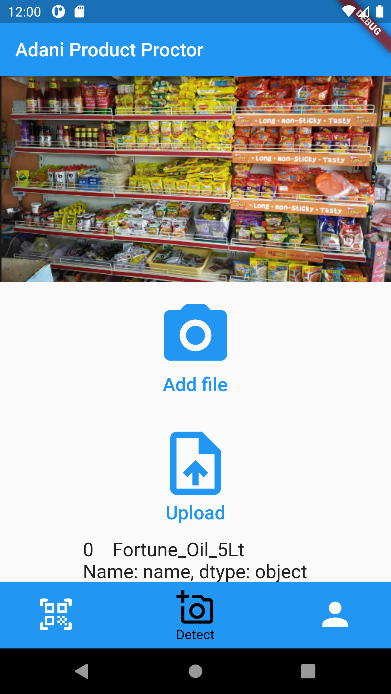
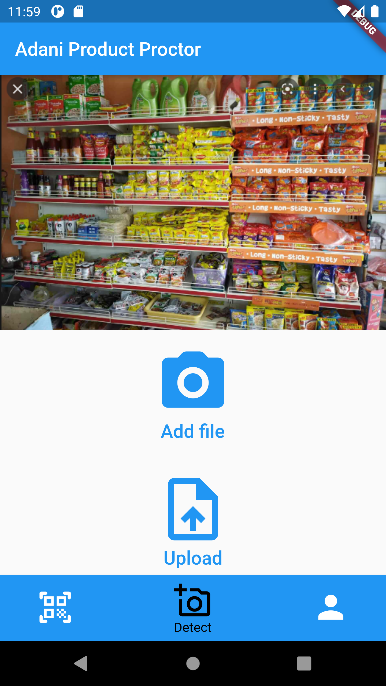
The name of the app in adani product proctor. We need to install flutter on the system in order for this to work. The app only contain simple dart coding for the widgets and some functions to perform GET and POST requests and render the results in the App. The app post the requests to the url which is a local host by default .Any one change if the url is changed.

this is upload file function that will send the selected file to the API and return a string message got from the API.



This is the camera controller the zero index in this is of the back camera. If we chose 1 then it’s the front camera for the mobile phones. Depending on the number cameras in the device this index can be changed.

Results:-

The result of the product Authentication page in the flutter app and the response as shown.