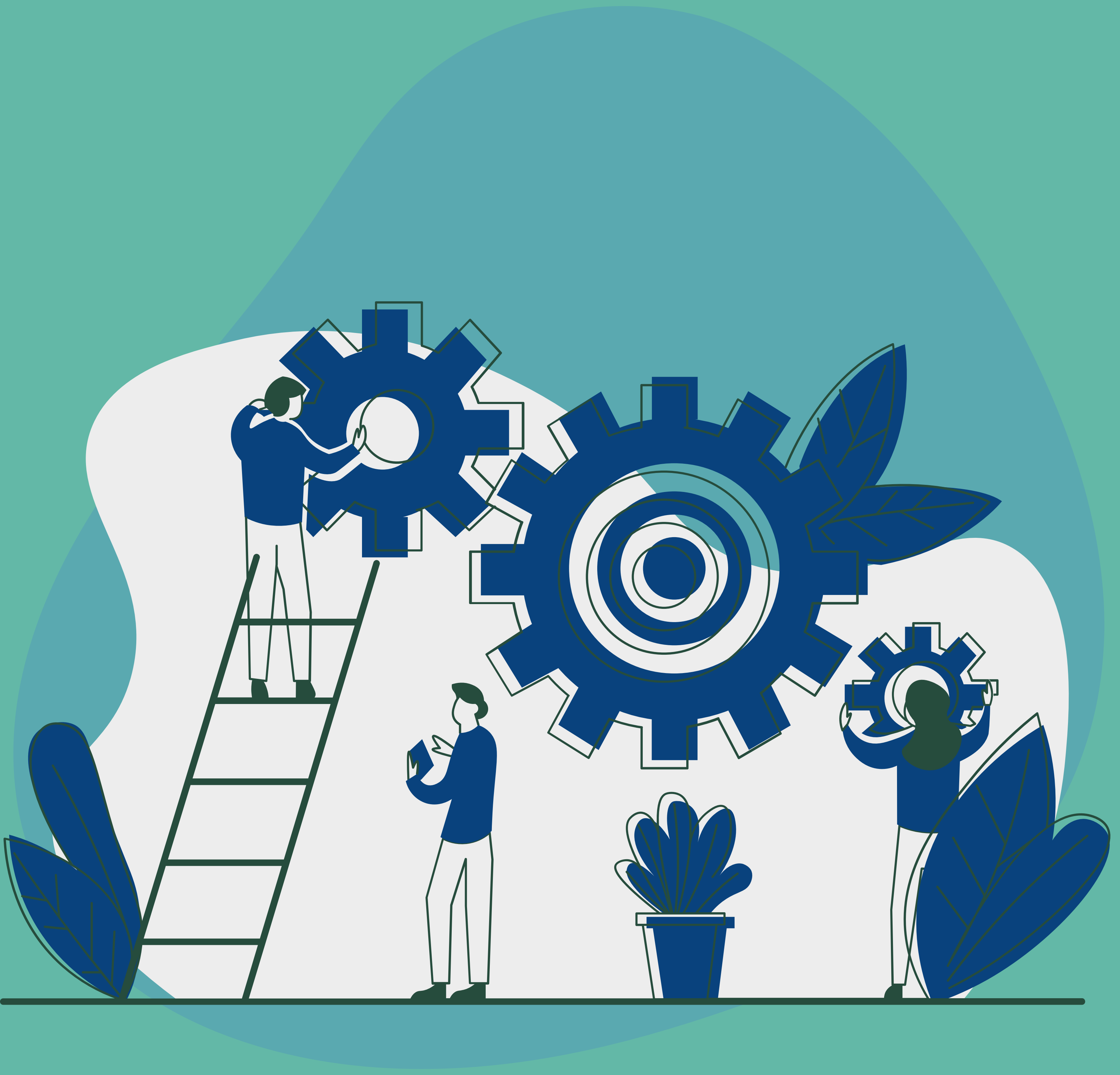
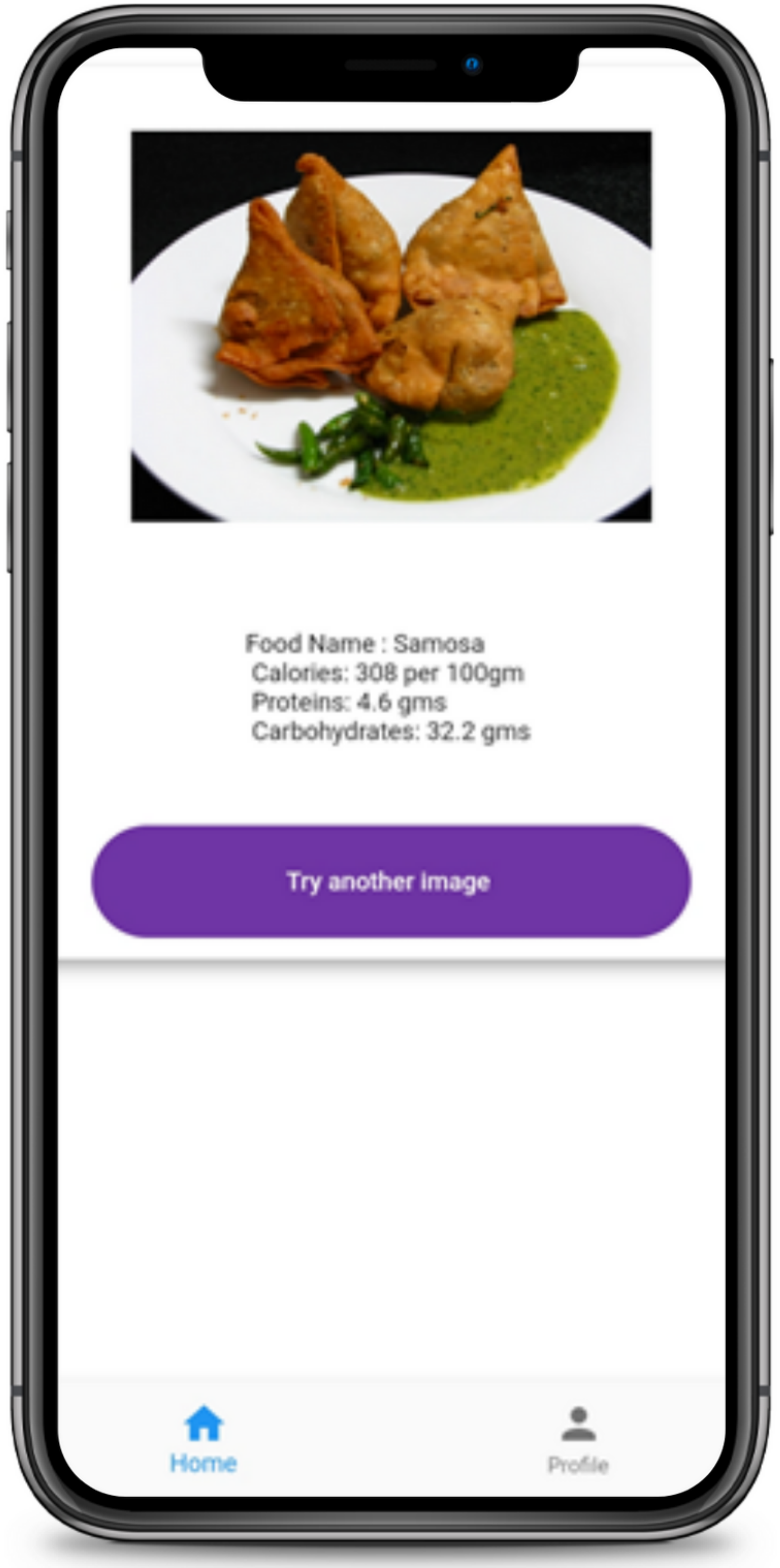


Problem Statement

To develop an image processing-based model to classify and calculate calories from food images. The model should be able to identify various classes of food images and estimate the calories.

NOVELTY IN PROJECT

- Existing applications like HealthifyMe, Calories Counter don't provide an option to input an image and track meal. A feature to track calories and nutrition from image along with existing drop-down menu will be highly beneficial.
- Previous work done in food image processing have issued conventional ML approach, which suffers from limitation of feature extraction.
- Our approach using deep learning aims at providing more accurate classification

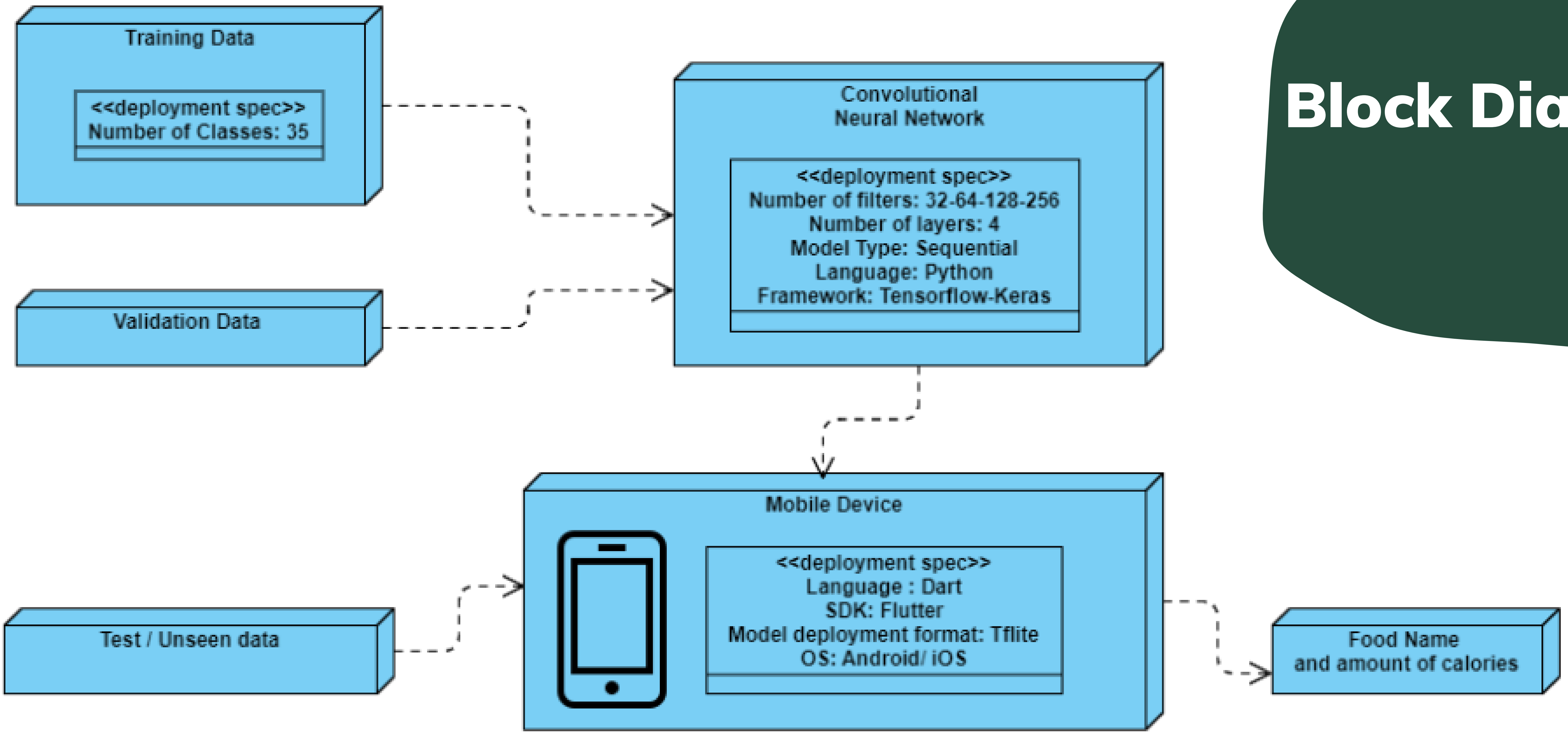


PROPOSED SOLUTION

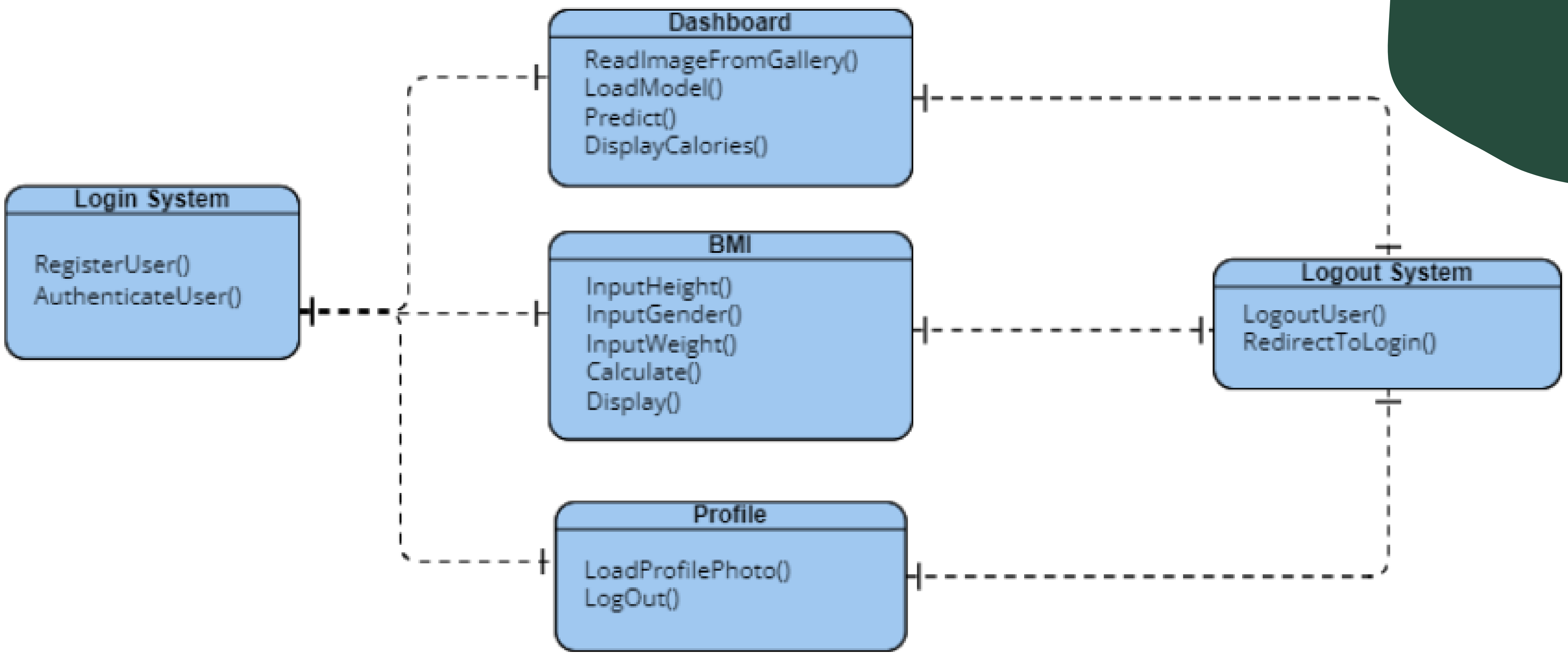
The existing systems are based on feature extraction and Machine Learning classification paradigm. Feature engineering is a process of putting domain knowledge into the creation of feature extractors to reduce the complexity of the data and make patterns more visible to learning algorithms to work. This process is difficult and expensive in terms of time and expertise. In Machine learning, most of the applied features need to be identified by an expert and then hand coded as per the domain and data type.

On other hand, Deep Learning algorithms try to learn high-level features from data. This is a very distinctive part of Deep Learning and a major step ahead of traditional Machine Learning. Therefore, deep learning reduces the task of developing new feature extractor for every problem. Like, Convolutional NN will try to learn low-level features such as edges and lines in early layers then parts of faces of people and then high-level representation of a face.

Block Diagram



Block Diagram

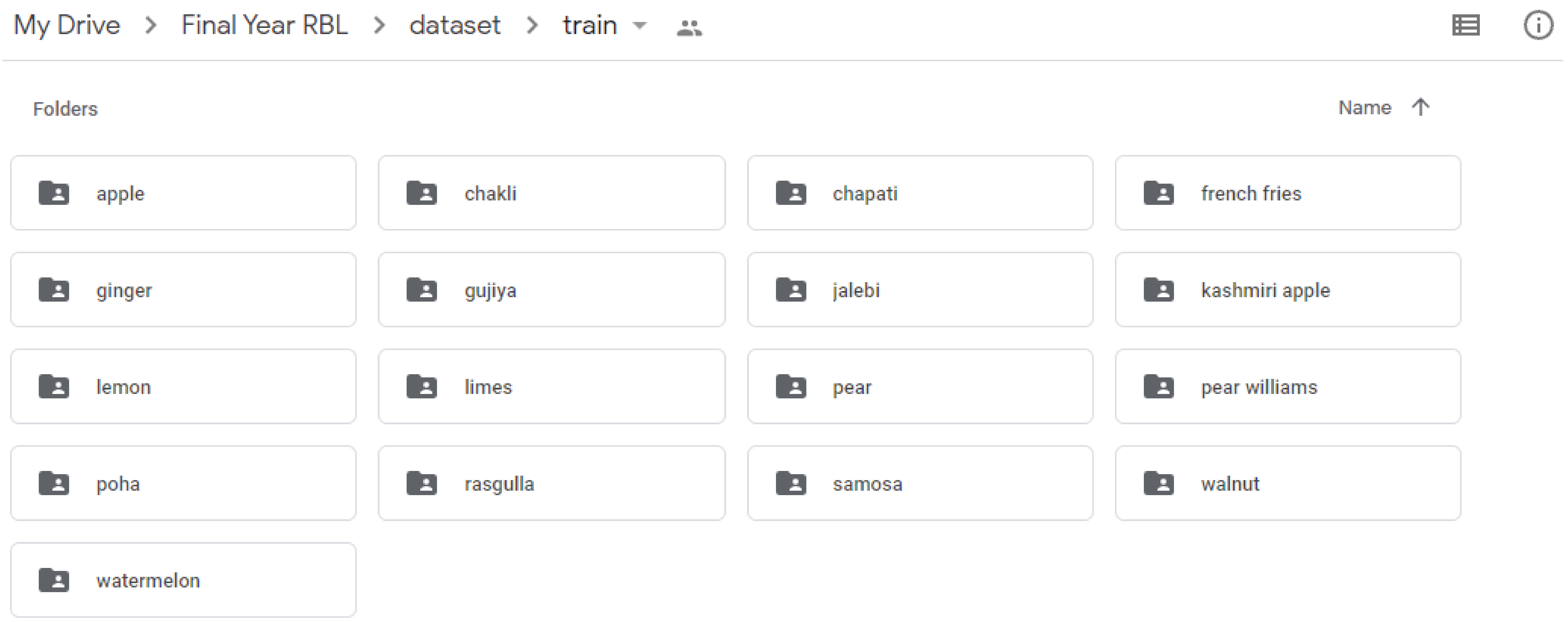


IMPLEMENTATION

Stage - I : Image Processing Model using CNN

ABOUT DATASET

The dataset contains 35 labels with 1875 training images and 795 validation Images. We have created a subset of Food 101 dataset and focused on incorporating Indian food items such as samosa, chakli, poha etc. for training the model.



ABOUT DATASET

Sr. No.	Food Name	Calories	Protein (gms)	Carbs (gms)	Fat (gms)
1	Apple	52	0.3	13.8	0.2
2	Chapati	70	3	15	0.4
3	Dabeli	278	7.23	42.35	9.4
4	Sev Puri	59	1.73	7.93	2.25
5	Vadapav	304	9.71	40.17	11.91
6	Samosa	308	4.67	32.21	17.86
7	Banana	105	1.1	23	0.3
8	Oranges	62	1.23	15.39	0.16
9	Modak	153	1.63	27.99	4.08
10	Rice	130	2.36	28.7	0.9
11	Dal	222	14	34	4.2
12	French Fries	323	3.4	43	15

Source: FatSecret API (https://www.fatsecret.com/) and NutritionX (https://www.nutritionix.com/food/)

CNN Parameter selection

We used hit-and-trail approach to select appropriate values for various CNN parameters for 17 labels.

Table – 3: Performance of different activation functions

Sr. No.	Activation Functions	No of Epochs	Training Accuracy	Training Loss	Validation Accuracy	Validation Loss
1	Sigmoid	10	0.7543	0.8978	0.678	0.8035
2	ReLu	10	0.8425	0.4626	0.8248	0.7345
3	TanH	10	0.6071	0.5144	0.751	0.8565

Table – 4: Number of filters in Hidden Layers of Convolutional Neural Network

Sr. No.	No of filters in Hidden Layers of Neural Network				Training Accuracy	Training Loss	Validation Accuracy	Validation Loss
	1	2	3	4				
1	8	8	8	8	0.5654	1.1468	0.6125	0.9834
2	16	16	16	16	0.5743	1.0226	0.6912	0.9541
3	8	16	32	64	0.7339	0.7960	0.8576	0.4392
4	32	64	128	256	0.8315	0.5117	0.9349	0.3176

CNN Parameter selection

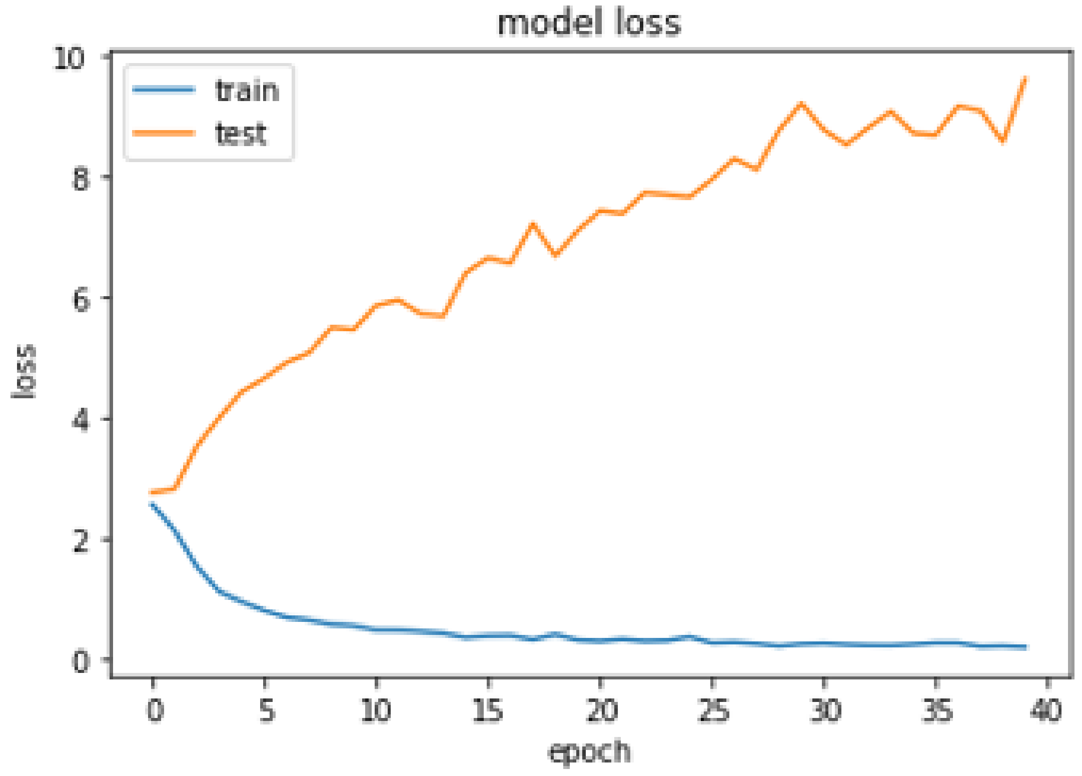
Table – 5: Performance of model with different epochs size

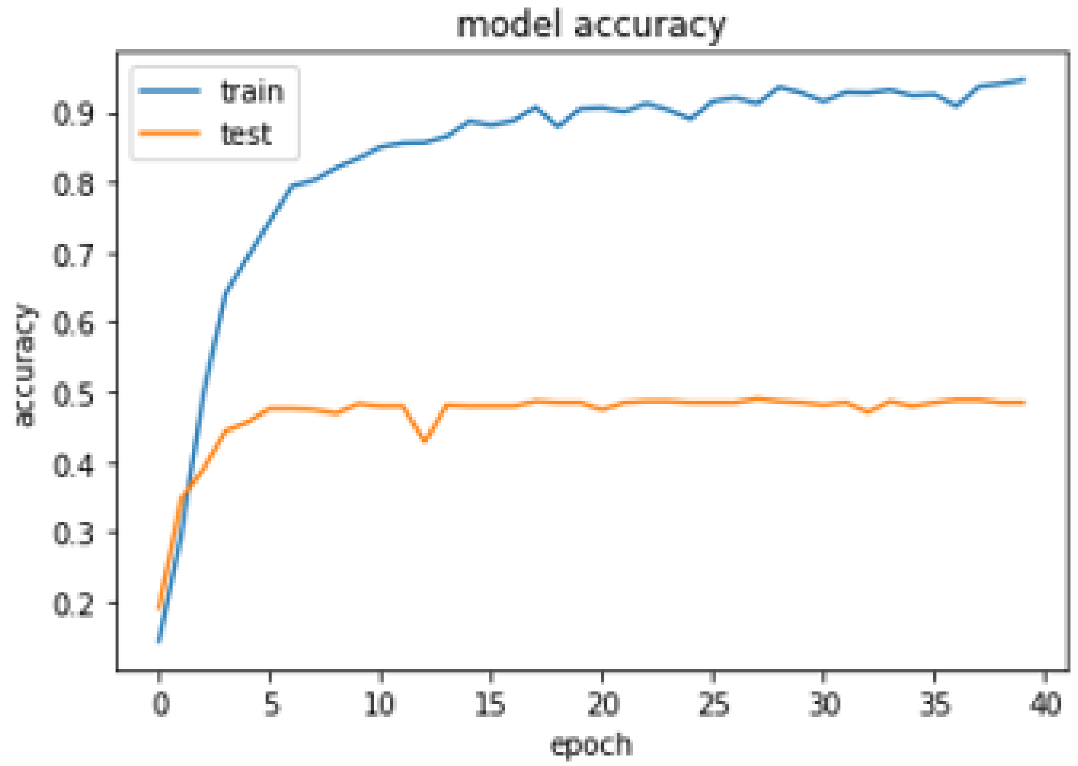
Sr. No.	Number of Epochs	Training Accuracy	Training Loss	Validation Accuracy	Validation Loss
1	10	0.8425	0.4626	0.8248	0.7345
2	20	0.9071	0.2780	0.9204	0.5103
3	30	0.9134	0.2202	0.9487	0.4027
4	40	0.9449	0.1888	0.9593	0.3803

Inferences:

1. We found that rectified linear activation function or ReLU gives high accuracy and low loss when validation dataset is provided to the model.
2. We decided to use configuration of 32 – 64 – 128 – 256 for number of filters in four hidden layers of CNN.
3. We choosed 40 epochs for fitting the dataset on model.

Performance of Model



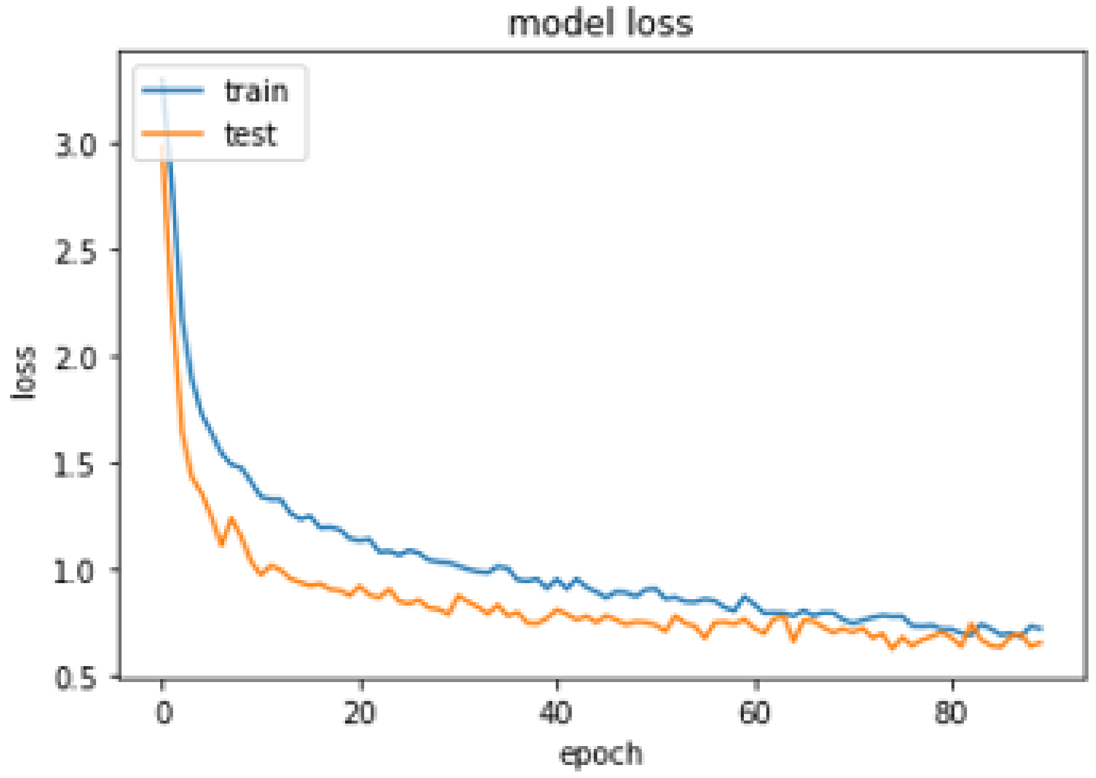


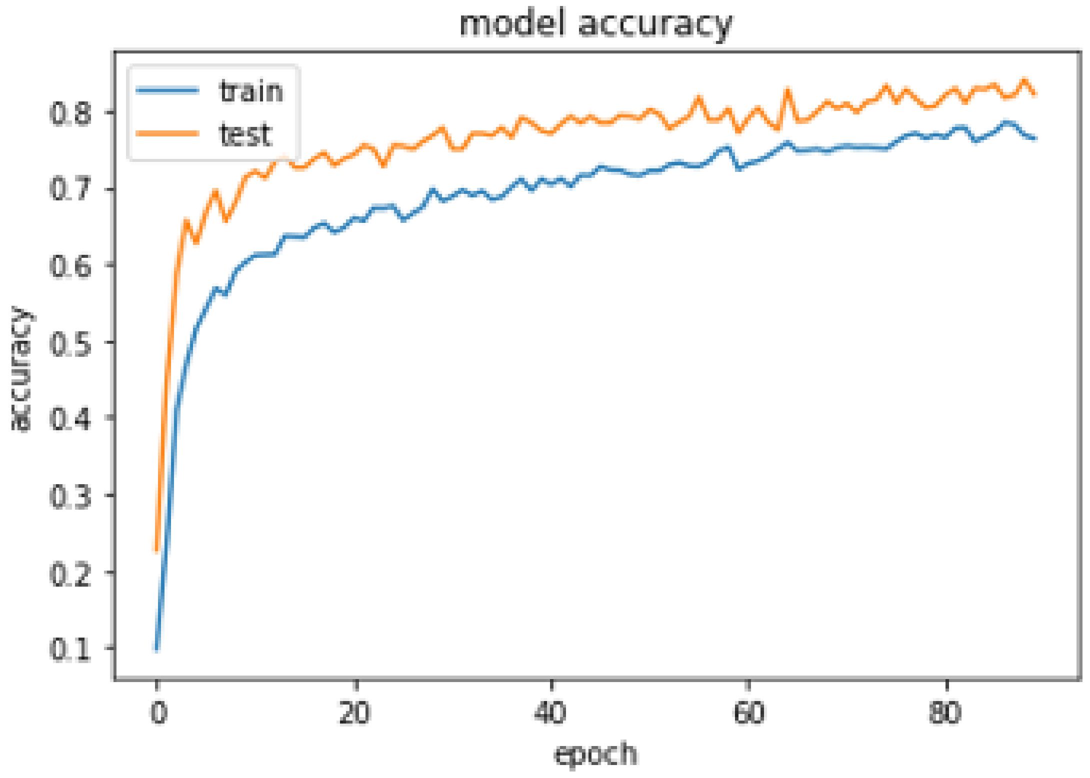
Epochs = 40

Low Train Loss = Low Bias

Low Test Loss = High Variance

UNDERFIT MODEL





Epochs = 90

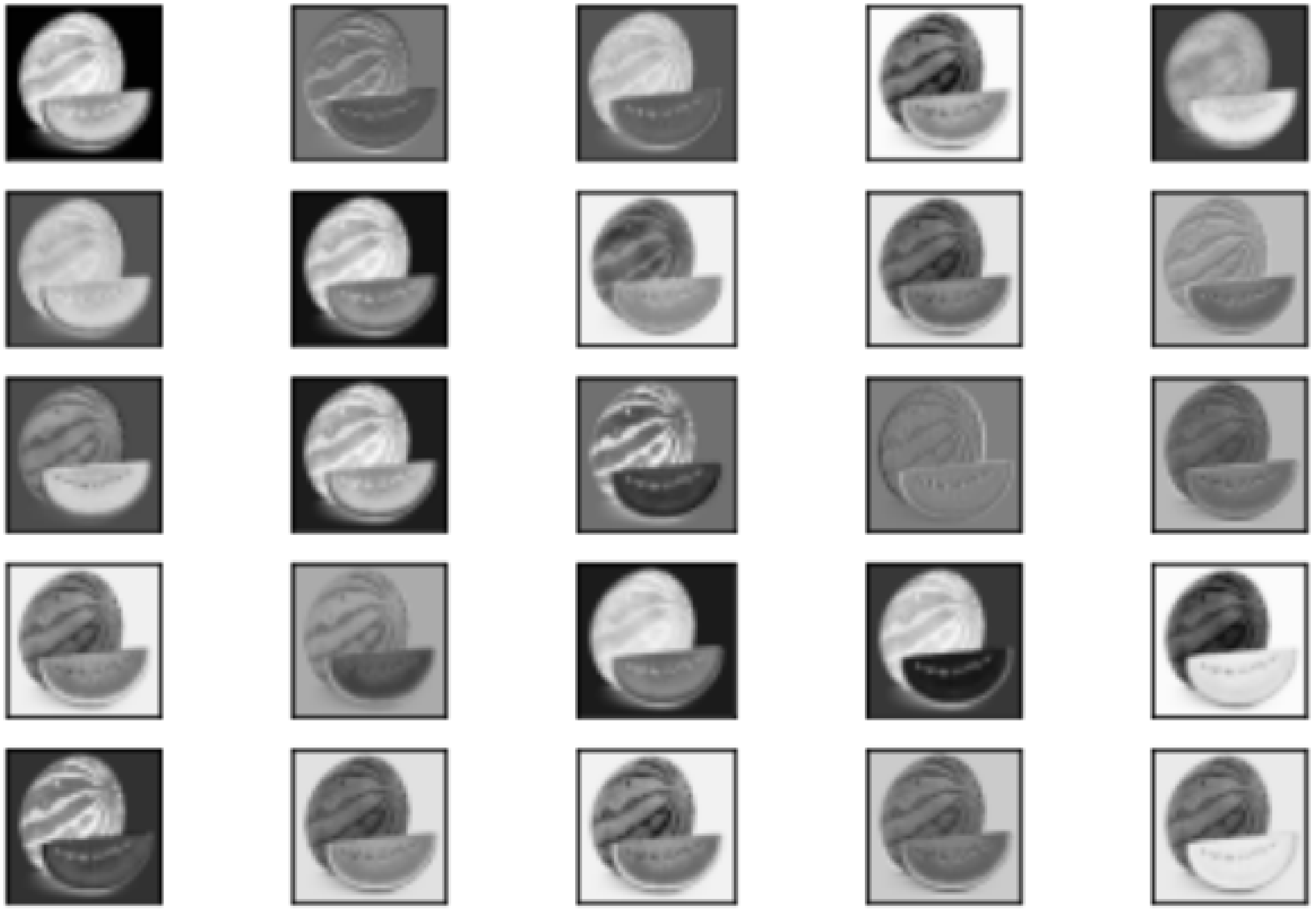
Low Train Loss = Low Bias

Low Test Loss = Low variance

BESTFIT MODEL

Features Extraction by CNN layers

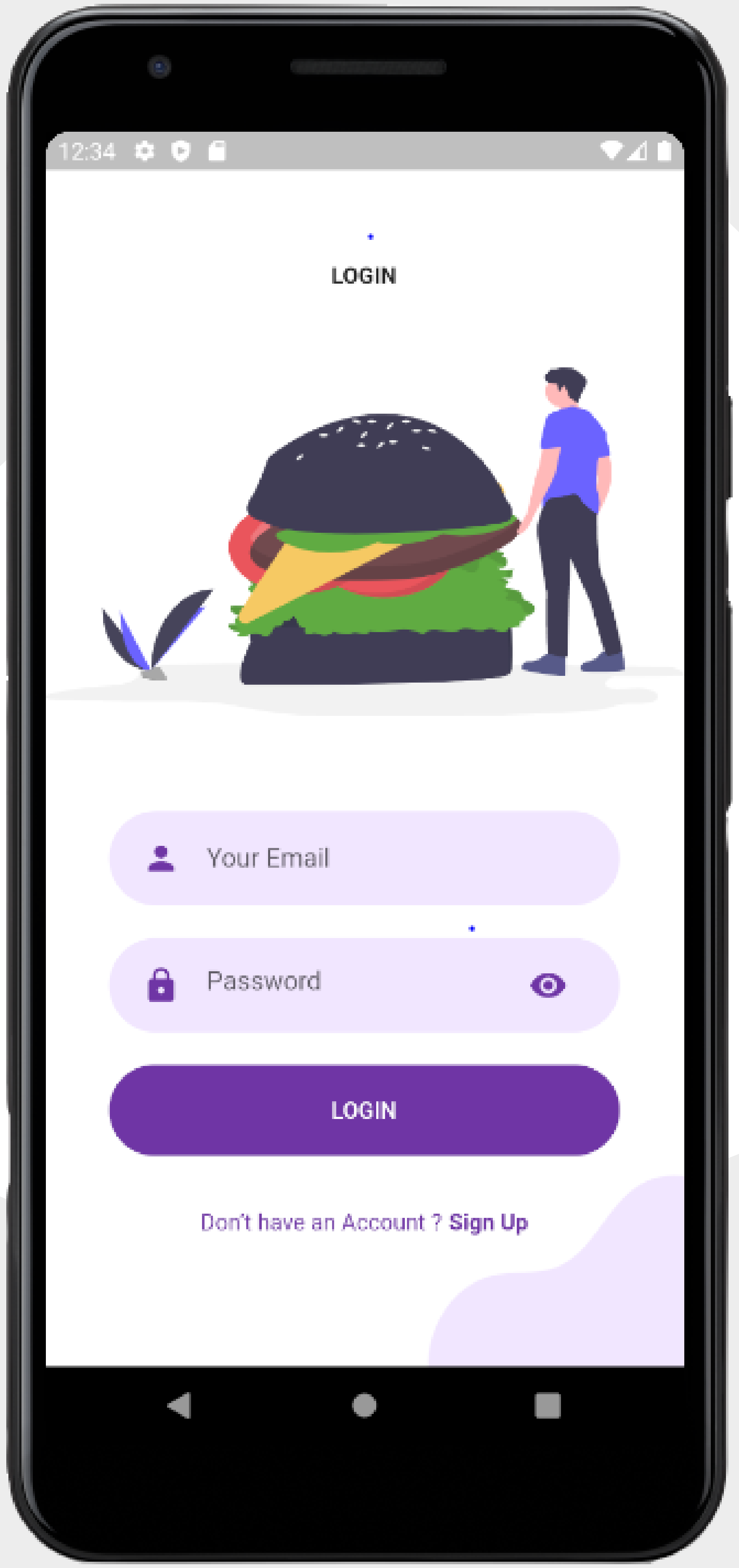
CNN’s have a strong ability to extract complex features that express the image in much more detail, learn the task specific features and are much more efficient.



Classification of test data using model / Output

```
image_url = '/content/drive/MyDrive/Final Year RBL/dataset/test4.jpg'
test_image = load_img(image_url, target_size = (150, 150))
test_image1 = img_to_array(test_image)/255
test_image = np.expand_dims(test_image1, axis = 0)
result = cnn_model.predict(test_image).round(3)
pred = np.argmax(result)
print(dic[pred])
plt.figure()
plt.imshow(test_image1)
plt.show()
```

poha



IMPLEMENTATION

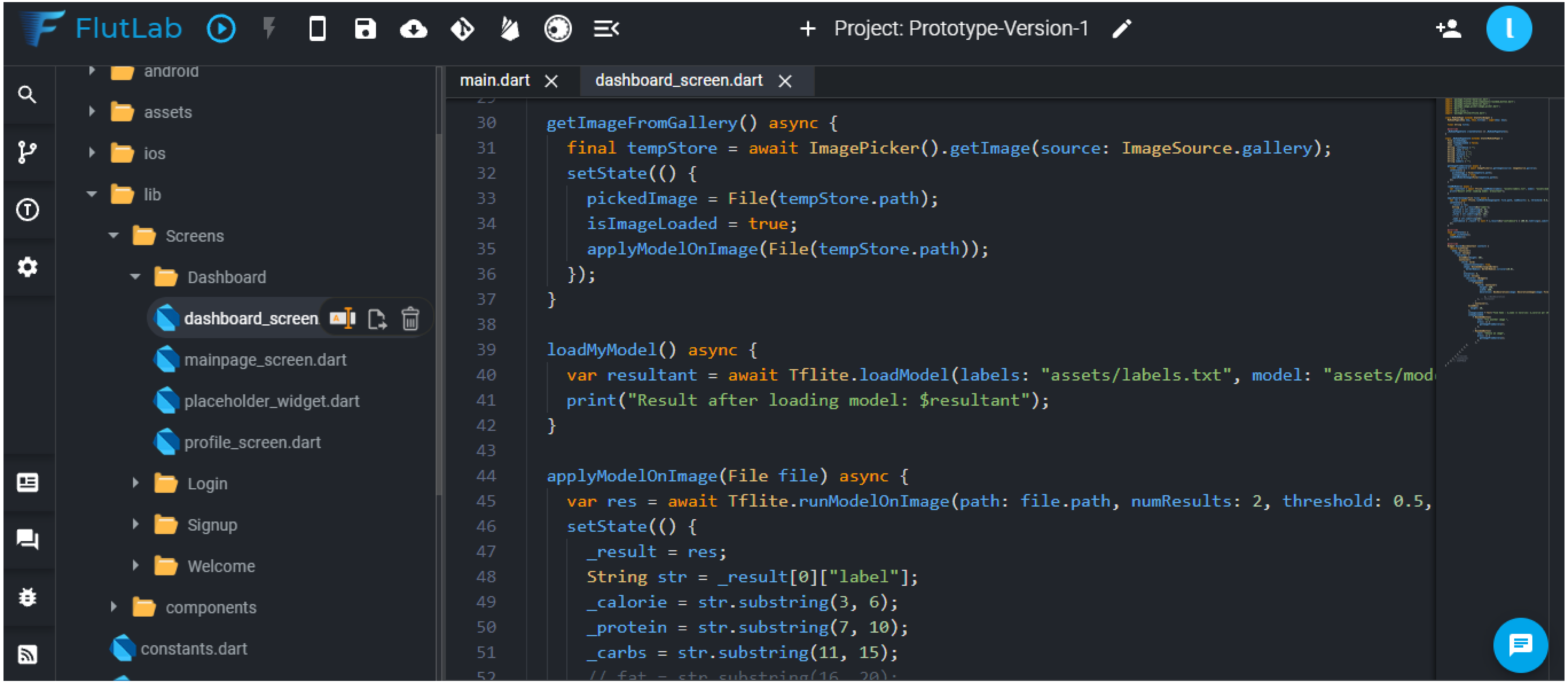
Stage - II : Development of Mobile Application

Mobile Application Development

IDE : FlutLab (www.flutlab.io/)

Language : Dart Programming / Flutter SDK

Platfrom : Andriod and iOS



FUTURE SCOPE

A functional prototype has been developed in this academic year. The project needs more refinement in terms of -

1. More food labels need to be included in model.
2. The mobile application can be connected by SQL/No-SQL database to store user-data and offer better fitness product experience
3. Object Detection using YOLO to detect multiple food label from same input image