Project Readme

Introduction

Food image recognition refers to identifying food items by seeing the food image. Different types of food can be recognized from different images. Our project has recognized around 25 types of food that include foods like ‘bruschetta’, ‘chocolate cake’, ‘French toast’, ‘chocolate mousse, etc. This has been achieved by training different models to accurately predict the distinct classes and thus come up with an app-based user interface that allows for checking recognized dish information and opt between checking the dish recipe or ordering the food based on the list of nearby venues and placing the order on the Zomato app linked to our app user interface.

Dataset

We have created a dataset from scratch for our project using 25 classes each for distinct food types. The unique classes identified in the dataset are as shown below:

```
array(['chocolate ', 'cup       ', 'donuts    ', 'dumplings ',
       'french    ', 'fried     ', 'garlic    ', 'gnocchi   ',
       'grilled   ', 'hamburger ', 'hot       ', 'ice       ',
       'macarons  ', 'mussels   ', 'nachos    ', 'pancakes  ',
       'pizza     ', 'red       ', 'samosa    ', 'spring    ',
       'strawberry', 'sushi     ', 'waffles   '], dtype='<U10')
```

This dataset is then stored in .mat format and then split into 80% train set and 20% test set. The count of the images for different classes in these train and test sets are shown below:

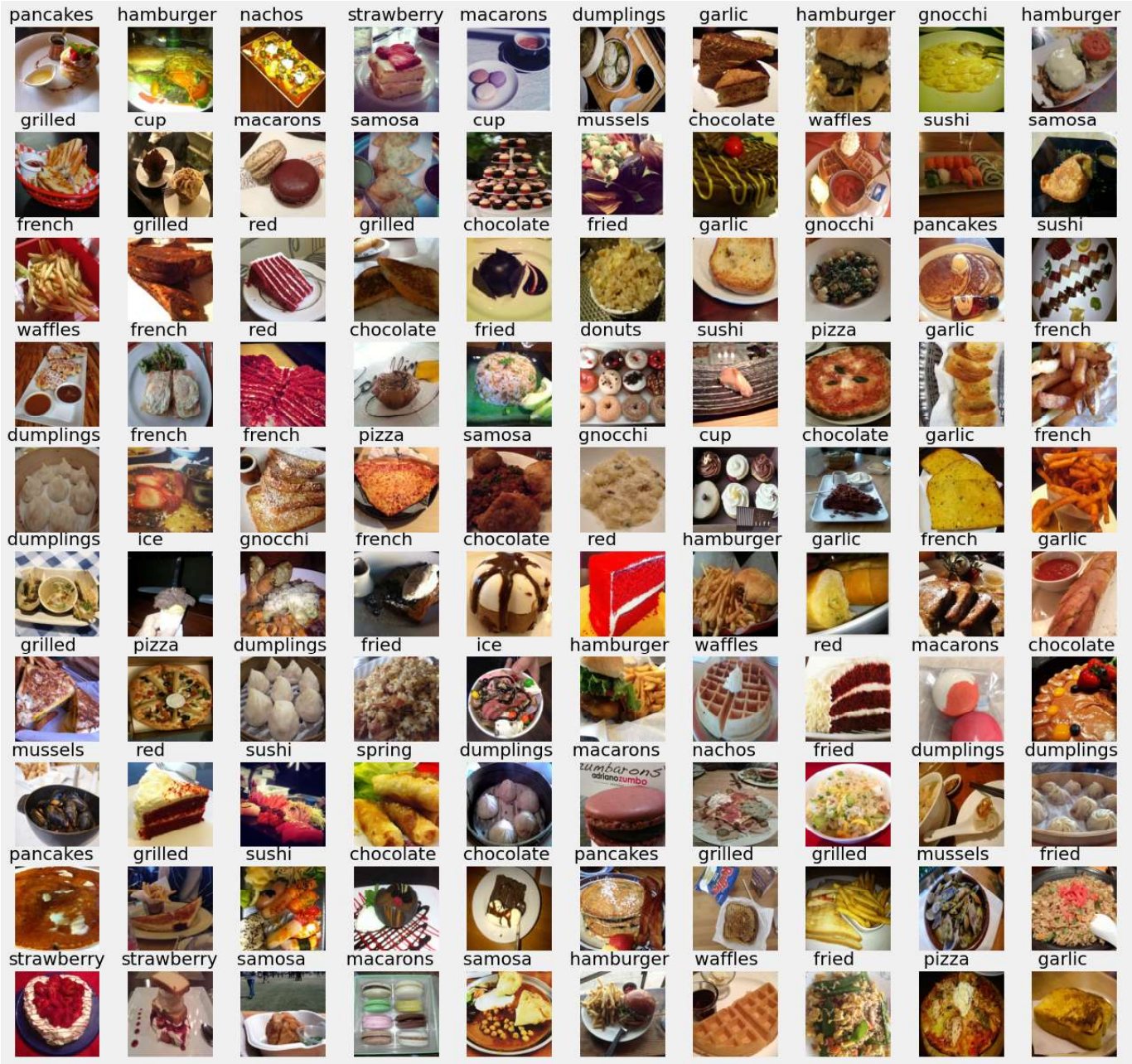
For train set:

```
{'chocolate ': 1432,
 'cup       ': 787,
 'donuts    ': 564,
 'dumplings ': 770,
 'french    ': 1412,
 'fried     ': 604,
 'garlic    ': 594,
 'gnocchi   ': 808,
 'grilled   ': 798,
 'hamburger ': 538,
 'hot       ': 709,
 'ice       ': 713,
 'macarons  ': 750,
 'mussels   ': 577,
 'nachos    ': 599,
 'pancakes  ': 653,
 'pizza     ': 562,
 'red       ': 630,
 'samosa    ': 630,
 'spring    ': 421,
 'strawberry': 587,
 'sushi     ': 538,
 'waffles   ': 635}
```

For test set:

```
{'chocolate ': 363,
 'cup       ': 173,
 'donuts    ': 143,
 'dumplings ': 210,
 'french    ': 363,
 'fried     ': 168,
 'garlic    ': 163,
 'gnocchi   ': 177,
 'grilled   ': 173,
 'hamburger ': 138,
 'hot       ': 189,
 'ice       ': 168,
 'macarons  ': 192,
 'mussels   ': 145,
 'nachos    ': 138,
 'pancakes  ': 172,
 'pizza     ': 142,
 'red       ': 146,
 'samosa    ': 162,
 'spring    ': 121,
 'strawberry': 154,
 'sushi     ': 130,
 'waffles   ': 148}
```

Images of different classes in the dataset are as shown in the form of grid view as below:



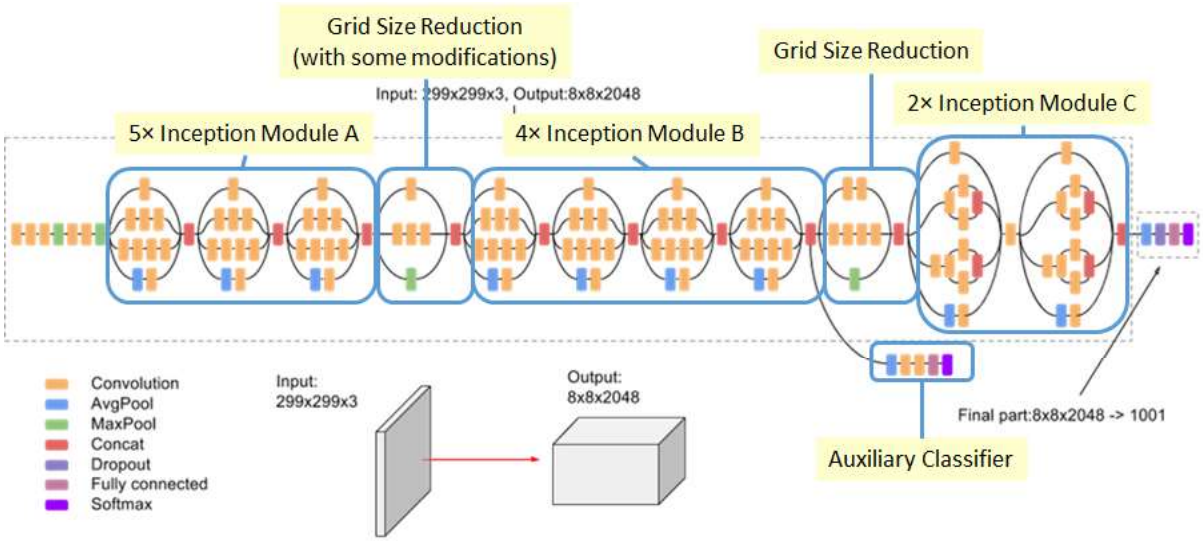
Dataset Link

https://drive.google.com/drive/folders/1dswZ_OWvMSFdQdfRKNw200d2rcKHAmuy?usp=sharing

Models

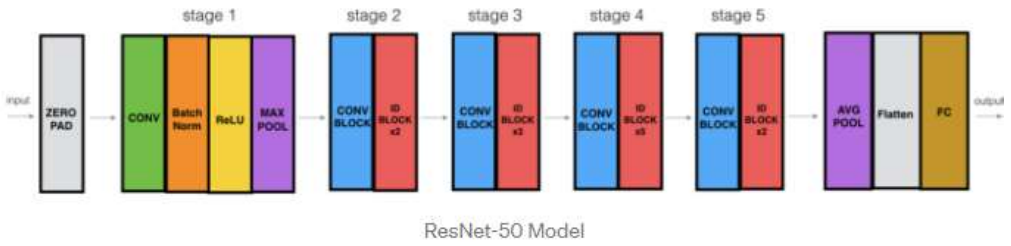
We have implemented four models for image classification-

- Inception Model
  Model Architecture



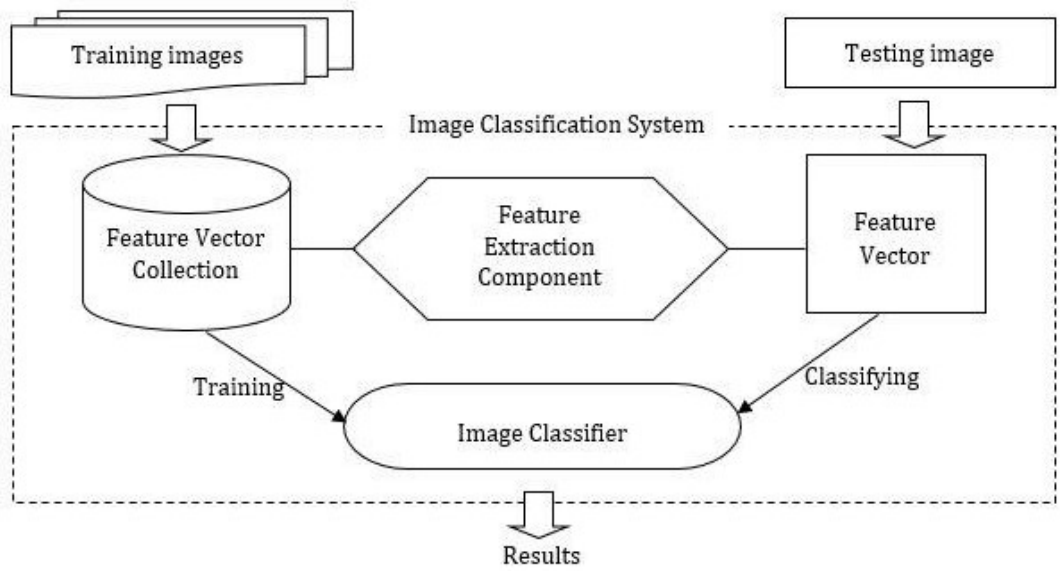
- VGG19 & VGG16
  Model Architecture

- ResNet-50
  Model Architecture



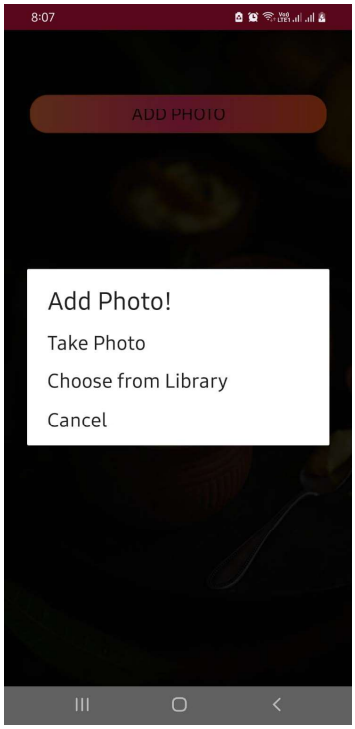
ResNet-50 Model

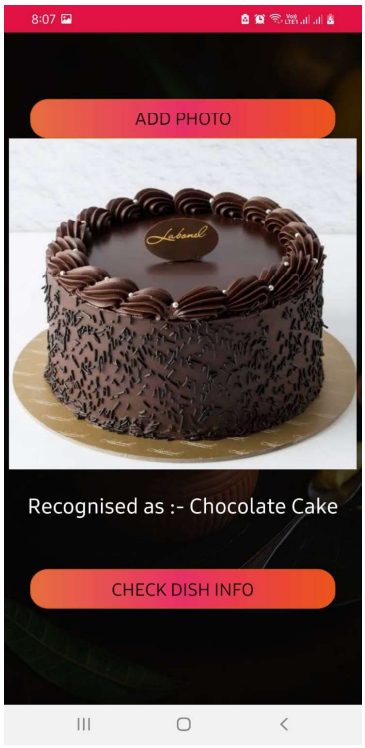
- Tensorflow Model Maker
  Model Architecture

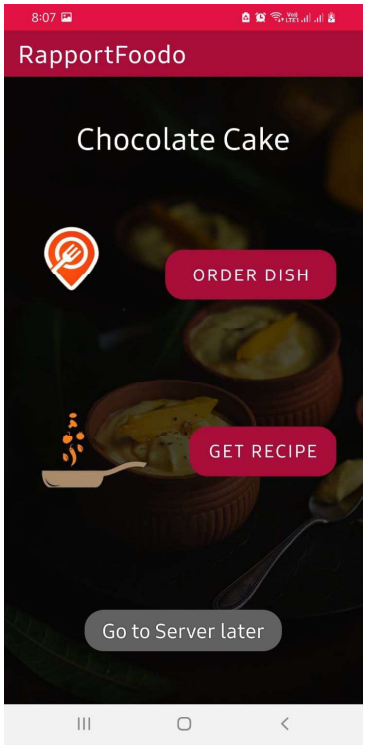


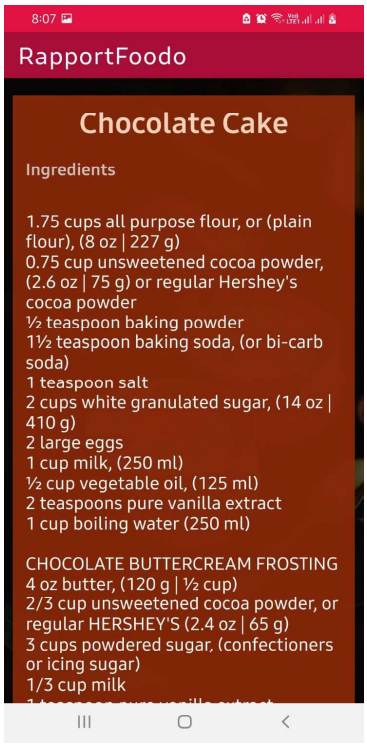
How to run the project

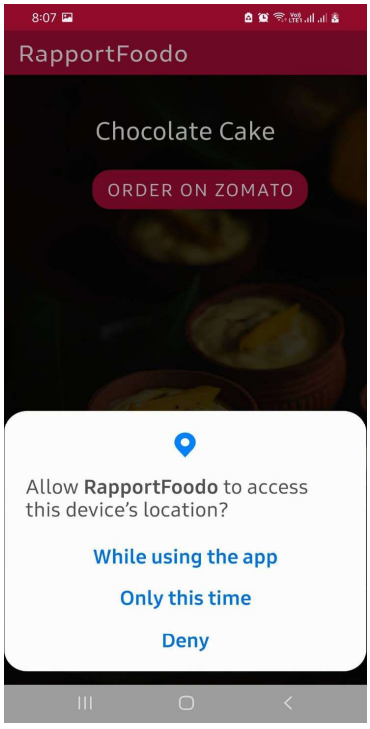
- **For running the application**- Go to app-debug.apk in the zip folder, install the apk file on the phone and then run the application. (Don't forget to give access to your location and kindly turn off the dark mode of your phone)
  Steps-











1. Firstly you will see the front cover for 2 secs and then you enter the home page where you will have the option to add the image.

2. Upload or click the food image. The model will recognize the dish and displays the dish name
3. Click on check info where you will get two options- order the dish and get the recipe
4. By clicking get the recipe, you will get the ingredients and instructions for cooking that recipe
5. By choosing order the dish option, firstly it will fetch the nearby restaurants that will be having that recipe and then you can directly place the order on zomato by clicking “Order on Zomato”.

- **For running the models-** We have submitted .ipynb files which will contain the models code and we have also saved our model in .pb format and one TensorFlow model in .tflite format. You can directly use it in deploying the model anywhere(app or web).

Challenges

- For implementing the large dataset and different neural networking models, we faced a computational challenge where we sampled our dataset and implemented models with fewer epochs.
- As we have to maintain dataset transparency we had to build the dataset of 25 classes from our own which was challenging. We also removed noise from the dataset manually.
- One of the main challenges was retrieving nearby restaurants by using Foursquare API.

Future Work

- Utilization of a larger dataset with more classes to recognize a better broad spectrum of food types.
- Training the data on a larger number of epochs on a platform that provides better computational features.
- Modification of the app user interfaces by providing in-app ordering features instead of providing options to switch to other food-ordering apps.