OneClickBite: Food Image Classification & Recipe Retrieval using Web Scraping

Raj Kumar Aggarwal
raj22053@iiitd.ac.in

Devansh Nautiyal
devansh22097@iiitd.ac.in

Gunjan Panda
gunjan22099@iiitd.ac.in

Shubhangi Agrawal
shubhangi22126@iiitd.ac.in

Shubhanshu Gupta
shubhanshu22127@iiitd.ac.in

1. INTRODUCTION

1.1 PROBLEM STATEMENT & MOTIVATION

For people to understand the world, images are considered to be the visual basis which help them obtain, express and transmit information. In recent times social media has gained popularity where a stronger information carrying capacity is contributed by images itself. Over 500 million posts may be found by simply searching #food on Instagram, while another 250 million posts can be found by simply searching #foodie, and around a 100 million posts for #foodphotography.

Food images are not just complicated but quite often deceive one's sense of taste by their sense of sight. Social networking connects people from all over the world drawing different cultures and their food together thus offering different visual representations of food. This lures people to try out new dishes, but there might be times where the dishes are not known in the first place. This will propose higher requirements on food image recognition. The project aims to identify a food using image classification algorithm(s) and provide options to order or cook the identified food as per user's preference.

1.2 DATASET

We used the Food-101 Dataset, which has a variety of food images, of 101 classes, having 1001 images each. However, we finalized on going for 45 classes. For the classification model, all 45 classes were fed as it is, however, for the application, out of these classes, similar classes like greek salad, caprese salad and caesar salad were integrated into one main class i.e. salad on reviewing manually. More such integrated classes were formed eg. pasta, cake, soup, pizza and ice cream, keeping the rest of the classes the same.

2. LITERATURE REVIEW

Google Lens (Dining)- It can scan a menu of food images and rightly classify them and further provide information related to that dish. However the retrieved set of search results include anything and everything associated with that, but doesn't provide streamlined search results. Rather the user ends up spending more time on exploring through the provided results.

[1] discusses the challenges in automatically recognizing and classifying food images using mobile imaging technologies, and introduces a new dataset of food images. The authors propose a new method for image representation using the perceptual concept of Anti-Textons, which outperforms other approaches. However, the proposed method doesn't work well with large datasets and lacks evaluation methods to test and compare different solutions. We propose to use CNN for food image classification that will be trained on the Food 101 dataset which is huge thus overcoming the shortcomings of the said literature.

[2] proposes a web search engine that allows any mobile device to send a query image and retrieve the most relevant recipes from the UPMC Food-101 dataset. The protocol followed was to split out examples having both image and text and randomly select some samples for each category to train a one-vs-rest linear SVM model. The authors set a future scope to enhance their retrieval system using machine learning techniques. Since our project is also based on food image recognition models, we can implement their enhanced model for image classification.

[3] proposes a food image retrieval and classification method based on Faster R-CNN, which detects the food area of the picture and extracts visual features for retrieval and classification tasks. The proposed method is

evaluated on the Dish-233 dataset. The author proposed three image classification and retrieval algorithms using CNN features, color features, and multi-core learning to improve retrieval accuracy, image matching and outperform single-core classification. The underlying model works on sequential data and doesn't perform well on large datasets due to slow convergence. Our model will focus on image classification incorporating various layers that would result in better accuracy even for large datasets.

[4] gives a general view on all aspects of Information retrieval and Web scraping. It covers different types of web scrapers, such as traditional web scrapers that extract information from HTML pages, and modern web scrapers that use machine learning techniques to extract information from unstructured data along with their advantages and disadvantages. However, the paper involves more discussion than practical implementation. We plan to implement the scraping technique using the jSoup module of Android Studio and extract the list of ingredients and recipe from the internet via the HTML tags.

[5] presents a convolutional neural network(CNN) architecture that uses a pre-trained VGG-16 network as a feature extractor, followed by a fully connected layer for classification. The dataset used in the study consists of 1000 images of food items belonging to 10 classes, which include burgers, pizza, sushi, and other commonly consumed foods. Overall, the paper presents a method for automatically classifying food items from images. As our proposed model will be trained on the Food 101 dataset, which has nearly 100 classes of food items, the data will consist of diversified images, resulting in a much better accuracy and hence more precise classification.

3. NOVELTY

Overcoming the shortcomings of the literature review, the novelty of our project lies in its uniqueness to provide a one stop solution to the users looking to explore new food varieties. The user will be given an option to upload a food image, which will be identified using classification algorithm(s). Once identified, the user can either order the food from a food aggregator (like Zomato) or cook from the recipe so extracted using web scraping technique(s). Users will have the flexibility to choose over recipe instructions or a video recipe. The web scraper will also extract the list of ingredients required and redirect the user to buy them from an online instant delivery application like BlinkIt. To the best of our knowledge, there isn't any similar one stop solution for the users looking to explore new food varieties.

4. METHODOLOGY

4.1 OVERALL WORKING OF APPLICATION

The proposed application allows users to upload a photo of their food, identify it and be able to cook or order the identified dish. The app is backed by tflite_model_maker model for food classification. Once identified, the user can either choose to order or cook the identified food. The overall working of the application is explained using wireframes through Figure 1 and the sample video of its working can be referred using: https://www.youtube.com/watch?v=uxTss2Gtp8w

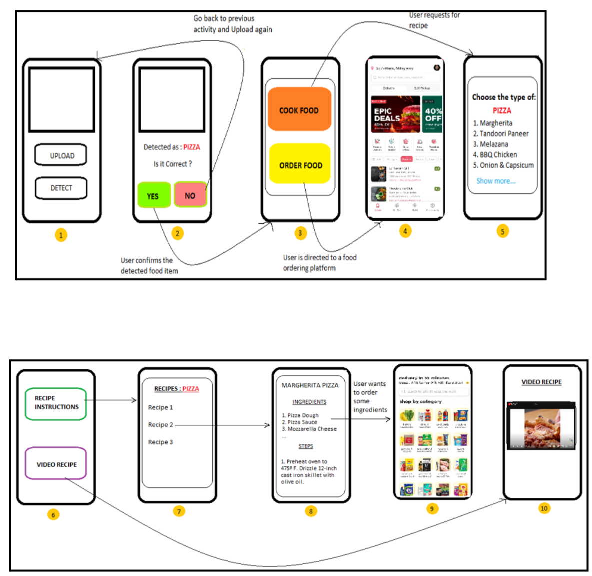

Figure 1 : *Working of proposed application*

4.2 MODEL

The experimentation began with TensorFlow Model Maker, which enables fine-tuning of pre-trained models on specific datasets through transfer learning. By using pre-trained models, the training process becomes quicker and requires less computational resources compared to more complex models. Later, we experimented with MobileNetV3, which is a lightweight model suitable for mobile apps and more complex than Model Maker. However, due to its smaller memory footprint, it does not produce as good results in image classification as more complex models. Finally, InceptionV3 was implemented, a widely used convolutional neural network architecture known for its high accuracy in image classification tasks. Though it is a relatively complex architecture and requires high computational power, it resulted in the best accuracy out of the three, with training at 81% and testing at 77%. Details of the metrics are discussed later. Ultimately, InceptionV3 was chosen to detect food images in our app, given its superior performance.

4.3 WEB SCRAPING VIA JSOUP

To be able to cook the identified food, we have scraped 8 websites where for every class 3 URLs were scraped for recipe and ingredient extraction. For this JSoup library was used. The multiple URL option enable users to be flexible around choosing recipes of their choice. The scrapped information includes all necessary information required by one to cook like the recipe title, cuisine type, ingredients, detailed instructions, etc. Figure 2 shows the architecture of web scraping implemented in the project.

The scraping model takes the URL of any recipe from a number of websites (currently 8) as an input, and fetches the scraped information listed earlier (Figure 3). For this asynchronous task in the application over the network connection, we used the volley library provided by Google. It manages the processing and caching of network requests and it saves developers valuable time from writing the same network call/cache code again and again.

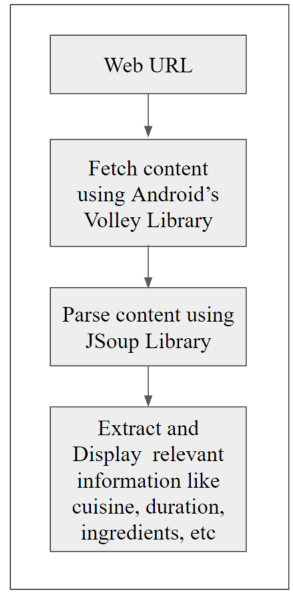

Figure 2 : *Architecture of web scraping*

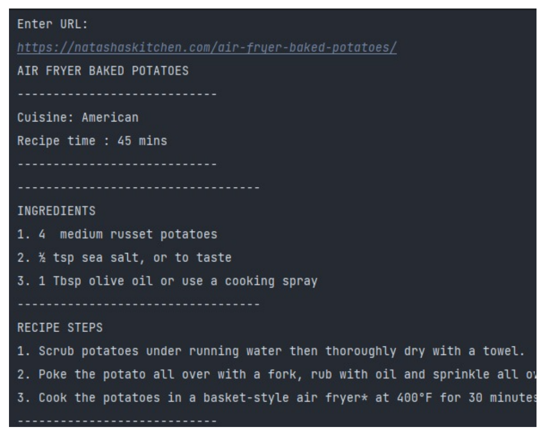

Figure 3: *Web Scraping via Jsoup*

4.4 OTHER FEATURES

In addition to written recipes, provision of video recipes is also provided via YouTube integration where for every class videos in two main languages namely English and Hindi will be displayed.

Grocery Delivery apps like BlinkIt and Zepto has also been integrated for the users to shop the ingredients.

In case the user wishes to order the identified food, he is redirected to a food delivery app (Zomato and Swiggy), also integrated in the application.

5. RESULT EVALUATION AND ANALYSIS

5.1 TRAINED NEURAL NETWORK MODEL ON DATASET

After settling in for 45 classes, and restraining our dataset to having more of those food items that can be found in India, we used 3 neural network models namely, Model Maker, MobileNetV3 and InceptionV3 to train the data and evaluated the performance of these on test data. InceptionV3 outperformed the rest, and was chosen as the final model for image classification.

5.2 EVALUATION

The models were trained on 80% of the data from the food101 dataset, considering 45 food classes, and the rest 20% was used for testing the performance of the model.

5.2.1 Comparison with baselines: Initially, the classification task was carried out on TensorFlow's Model Maker for 45 food classes out of the 101 classes present in the dataset, for 5 epochs with default parameters set.Table 1 displays the various model's accuracy and loss for both training and testing data over several epochs. The first attempt resulted in an accuracy of 72.77% and 72.37% for training and testing data, respectively, with a decreasing loss in each epoch. In the subsequent phase, the same model was trained for 30 epochs with a batch size of 32, resulting in a significant improvement in training accuracy from 72% to 78%. However, there was not much improvement in the test accuracy. MobileNet and InceptionV3 were then implemented, but MobileNet achieved only 60% accuracy for the training set and 51% for the testing set. On the other hand, InceptionV3, with its 48-layer deep network and stricter modelling, produced the best results with 81% accuracy on the training set and 77% on the testing set. Due to the minimized loss and significant increase in accuracy, this model was selected for classifying uploaded food images.

Loss is a measure of how well the model predicts the correct class label for each image, while accuracy measures the proportion of correctly classified images. As accuracy increases and loss decreases, it shows that the model is improving in image classification.

	Accuracy	Loss
Training Data	ModelMaker (5 epochs) 72.77% ModelMaker (30 epochs) 78.03% MobileNetV3 60.23% InceptionV3 81.13%	ModelMaker (5 epochs) 1.60 ModelMaker (30 epochs) 1.49 MobileNetV3 1.35 InceptionV3 0.79
Testing Data	ModelMaker (5 epochs) 72.37% ModelMaker (30 epochs) 74.16% MobileNetV3 51.63% InceptionV3 77.75%	ModelMaker (5 epochs) 1.61 ModelMaker (30 epochs) 1.57 MobileNetV3 1.89 InceptionV3 0.95

Table 1: *Accuracy and Loss for all models*

5.2.2 System's performance on existing data: The final model was evaluated on existing data to determine its performance on the specific task of food image classification. The precision, recall, and F1-score are the evaluation metrics for each class, and the support indicates the number of instances of each class in the test set. The overall accuracy of the model is 0.82, which means it correctly predicted the class of 82% of the food items. The weighted average of precision, recall, and F1-score is also 0.82, which indicates that the model performs similarly across all classes.

However, looking at the individual class metrics, we can see that the performance of the model varies significantly across different classes. Some classes, such as caesar_salad, fish_and_chips, and macarons, have high precision, recall, and F1-score, while others, such as spaghetti_bolognese and sushi, have relatively lower scores. Based on the three classification reports(in the figures below) for the three models, it can be concluded that MobileNetV3 had the poorest performance, exhibiting low accuracy, precision, recall, and F1-score for most food classes. ModelMaker's performance was decent, while InceptionV3 delivered the most favorable results with high accuracy, precision, recall, and F1-score.

	precision	recall	f1-score	support
apple_pie	0.52	0.52	0.52	210
breakfast_burrito	0.69	0.76	0.72	204
bruschetta	0.64	0.59	0.62	178
caesar_salad	0.78	0.80	0.79	200
caprese_salad	0.68	0.76	0.72	186
carrot_cake	0.69	0.65	0.67	191
cheesecake	0.51	0.45	0.48	177
chicken_curry	0.78	0.65	0.71	200
chicken_wings	0.89	0.79	0.84	204
chocolate_cake	0.64	0.58	0.61	219
churros	0.84	0.80	0.82	184
club_sandwich	0.78	0.75	0.77	195
cup_cakes	0.73	0.76	0.74	198
donuts	0.74	0.70	0.72	218
dumplings	0.95	0.86	0.90	200
falafel	0.57	0.70	0.63	180
fish_and_chips	0.72	0.78	0.75	195
french_fries	0.86	0.82	0.84	209
french_toast	0.57	0.67	0.62	190
fried_rice	0.79	0.83	0.81	196
garlic_bread	0.74	0.68	0.71	187
greek_salad	0.69	0.79	0.74	204
grilled_cheese_sandwich	0.62	0.62	0.62	191
hamburger	0.75	0.81	0.78	213
hot_and_sour_soup	0.91	0.89	0.90	210
hot_dog	0.86	0.74	0.80	226
ice_cream	0.71	0.80	0.75	213
lasagna	0.81	0.60	0.69	187
macaroni_and_cheese	0.75	0.77	0.76	186
macarons	0.88	0.93	0.91	192
miso_soup	0.95	0.89	0.92	211
omelette	0.66	0.56	0.60	198
onion_rings	0.88	0.89	0.88	205
pancakes	0.71	0.69	0.70	191
pizza	0.71	0.87	0.78	187
ramen	0.79	0.90	0.84	205
red_velvet_cake	0.72	0.71	0.72	222
samosa	0.76	0.81	0.79	207
spaghetti_bolognese	0.91	0.90	0.91	221
spring_rolls	0.78	0.77	0.78	188
strawberry_shortcake	0.63	0.77	0.69	193
sushi	0.84	0.64	0.73	218
tacos	0.67	0.64	0.65	193
tiramisu	0.62	0.68	0.65	212
waffles	0.71	0.71	0.71	206
accuracy			0.74	9000
macro avg	0.74	0.74	0.74	9000
weighted avg	0.75	0.74	0.74	9000

Figure 4: Evaluation metrics for ModelMaker model

	precision	recall	f1-score	support
apple_pie	0.33	0.21	0.26	308
breakfast_burrito	0.45	0.62	0.52	277
bruschetta	0.39	0.33	0.36	288
caesar_salad	0.55	0.52	0.54	285
caprese_salad	0.37	0.51	0.43	302
carrot_cake	0.42	0.33	0.37	307
cheesecake	0.35	0.23	0.28	291
chicken_curry	0.38	0.38	0.38	319
chicken_wings	0.49	0.71	0.58	292
chocolate_cake	0.38	0.52	0.44	310
churros	0.58	0.56	0.57	299
club_sandwich	0.52	0.51	0.52	286
cup_cakes	0.54	0.59	0.56	282
donuts	0.54	0.53	0.53	295
dumplings	0.76	0.75	0.75	296
falafel	0.42	0.45	0.44	335
fish_and_chips	0.52	0.39	0.44	312
french_fries	0.66	0.68	0.67	294
french_toast	0.33	0.33	0.33	287
fried_rice	0.65	0.59	0.62	288
garlic_bread	0.53	0.46	0.49	314
greek_salad	0.45	0.51	0.48	282
grilled_cheese_sandwich	0.33	0.23	0.27	311
hamburger	0.66	0.68	0.67	309
hot_and_sour_soup	0.65	0.73	0.68	313
hot_dog	0.72	0.66	0.69	298
ice_cream	0.38	0.68	0.49	281
lasagna	0.56	0.33	0.42	312
macaroni_and_cheese	0.66	0.56	0.61	302
macarons	0.73	0.73	0.73	289
miso_soup	0.81	0.72	0.76	340
omelette	0.27	0.22	0.25	301
onion_rings	0.60	0.61	0.60	282
pancakes	0.50	0.38	0.43	287
pizza	0.75	0.77	0.76	304
ramen	0.57	0.57	0.57	282
red_velvet_cake	0.53	0.60	0.56	330
samosa	0.49	0.47	0.48	286
spaghetti_bolognese	0.71	0.83	0.77	314
spring_rolls	0.44	0.49	0.47	306
strawberry_shortcake	0.43	0.56	0.49	294
sushi	0.46	0.50	0.48	304
tacos	0.38	0.49	0.43	312
tiramisu	0.45	0.24	0.31	282
waffles	0.60	0.47	0.53	313
accuracy			0.52	13501
macro avg	0.52	0.52	0.51	13501
weighted avg	0.52	0.52	0.51	13501

Figure 5: Evaluation metrics for MobileNetV3 model

	precision	recall	f1-score	support
breakfast_burrito	0.88	0.66	0.76	750
bruschetta	0.87	0.66	0.75	750
caesar_salad	0.97	0.76	0.85	750
caprese_salad	0.64	0.96	0.77	750
carrot_cake	0.89	0.73	0.80	750
cheesecake	0.77	0.81	0.79	750
chicken_curry	0.98	0.57	0.72	750
chicken_wings	0.82	0.87	0.85	750
chocolate_cake	0.80	0.76	0.78	750
churros	0.91	0.87	0.89	750
club_sandwich	0.86	0.85	0.86	750
cup_cakes	0.70	0.93	0.80	750
donuts	0.71	0.92	0.80	750
dumplings	0.81	0.94	0.87	750
falafel	0.77	0.81	0.79	750
fish_and_chips	0.95	0.84	0.89	750
french_fries	0.91	0.92	0.91	750
french_toast	0.98	0.56	0.71	750
fried_rice	0.93	0.84	0.88	750
garlic_bread	0.90	0.80	0.85	750
greek_salad	0.80	0.87	0.83	750
grilled_cheese_sandwich	0.88	0.76	0.81	750
hamburger	0.93	0.85	0.89	750
hot_and_sour_soup	0.99	0.82	0.90	750
hot_dog	0.87	0.83	0.85	750
ice_cream	0.83	0.79	0.81	750
lasagna	0.97	0.45	0.61	750
macaroni_and_cheese	0.91	0.83	0.87	750
macarons	0.84	0.97	0.90	750
miso_soup	0.91	0.98	0.94	750
nachos	0.91	0.86	0.89	750
omelette	0.79	0.78	0.78	750
onion_rings	0.89	0.95	0.92	750
pancakes	0.95	0.76	0.84	750
pizza	0.89	0.84	0.87	750
ramen	0.73	0.94	0.82	750
red_velvet_cake	0.81	0.87	0.84	750
samosa	0.93	0.76	0.84	750
spaghetti_bolognese	0.62	0.99	0.76	750
spring_rolls	0.83	0.92	0.87	750
strawberry_shortcake	0.86	0.78	0.82	750
sushi	0.47	0.95	0.63	750
tacos	0.81	0.77	0.79	750
tiramisu	0.93	0.77	0.84	750
waffles	0.77	0.90	0.83	750
accuracy			0.82	33750
macro avg	0.85	0.82	0.82	33750
weighted avg	0.85	0.82	0.82	33750

Figure 6: Evaluation metrics for InceptionV3 model

5.2.3 SOTA on different evaluation metrics: The final model then detected the food image and further recipes were extracted from the internet using web scraping technique. It can be evaluated on different metrics such as data completeness, data quality, scraping efficiency, and data relevance. The scraping of recipes from websites using Volley successfully fetched all relevant text from the recipe title, ingredients, time taken, and steps by targeting specific HTML class tags. This indicated completeness, while data quality was validated by the accuracy of the scraped data matching the actual data, fulfilling the use case of fetching recipe instructions for the detected food item. The scraping process was highly efficient, taking no more than 5 seconds to display the fetched recipe on the screen, provided the internet connection is active.

5.2.4 Performance on new data and handling different cases: The final model was evaluated on new data to determine its ability to generalize to unseen images, wherein it detected images with 77% accuracy. The system's performance was tested on different use cases, such as images with poor lighting and from different angles.

6. CONCLUSION

We have proposed a novel product that provides a one stop solution to foodies from classification & identification to order or cook a given food item. By implementing our proposed models, we were able to achieve significant improvements over the baseline model, as evidenced by the increased accuracy, precision, recall, and F1-score, as well as reduced loss.

7. FUTURE WORK

By looking at the results and the prototype, we infer that the product can be potentially used to solve the need of the user and can be implemented to work at scale with many more classes covering both major and minor food categories i.e. from the widely known to region specific classes thus making even the rarest of existing food items popular. The dataset can be further added with more augmented images to train the models better on real-life like submissions done by users. The classification can be improved using other complex models and at par algorithms, thus, aiming to improve the overall performance metrics, resulting in more accurate identification of food items. Additional features like calorie count and macro and micro nutrient distribution could be added by extending it to a completely new domain- food segmentation. These are some of the possible extensions to the proposed formulated problem statement.

7. REFERENCES

1. Farinella, G. M., Allegra, D., Moltisanti, M., Stanco, F., & Battiato, S. (2016). Retrieval and classification of food images. *Computers in biology and medicine*, *77*, 23-39.

2. Wang, X., Kumar, D., Thome, N., Cord, M., & Precioso, F. (2015, June). Recipe recognition with large multimodal food dataset. In 2015 IEEE International Conference on Multimedia & Expo Workshops (ICMEW) (pp. 1-6). IEEE.

3. Wei, P., & Wang, B. (2020). Food image classification and image retrieval based on visual features and machine learning. *Multimedia Systems*, 1-12.

4. Dastidar, B. G., Banerjee, D., & Sengupta, S. (2016). An intelligent survey of personalized information retrieval using web scraper. *International Journal of Education and Management Engineering*, *6*(5), 24-31.

5. Attokaren, D. J., Fernandes, I. G., Sriram, A., Murthy, Y. S., & Koolagudi, S. G. (2017, November). Food classification from images using convolutional neural networks. In *TENCON 2017-2017 IEEE Region 10 Conference* (pp. 2801-2806). IEEE.