Ingredient Detection, Title and Recipe Retrieval from Food Images

Bhavikha Chopra
Dept. of CSE
PES University
Bengaluru, India
bhavikachopra1821@gmail.com

Gitika Jain
Dept. of CSE
PES University
Bengaluru, India
gitikavinaykiya@gmail.com

Mahendra N
Dept. of CSE
PES University
Bengaluru, India
mahendranerella2002@gmail.com

Snigdha Sinha
Dept. of CSE
PES University
Bengaluru, India
snigdhasinha0811@gmail.com

Dr. S. Natarajan
Dept. of CSE
PES University
Bengaluru, India
natarajan@pes.edu

***Abstract*—With advancements in the machine learning community and an increase in social media usage in our daily lives, we see a steady growth in the number of food pictures shared online by users on a daily basis. Unfortunately, the picture does not adequately convey the intricate recipe that was used to prepare it. Thus, the image-recipe association problem is addressed in this study. Large datasets like Recipe1M+ and Food101 were accessible for our use, however, they do not rank highly for Indian dishes. As a result, a unique dataset that is entirely devoted to Indian cuisine has been developed and utilized. The aim is to develop a website that can take an image of an Indian dish as input, identify its ingredients, and retrieve the dish's title. Further, this title is used to retrieve the recipe from the web. The task can be extremely challenging for a computer since it must understand the data well enough to produce accurate and precise ingredients. Both the dataset and the code will be made publicly available.**

***Keywords*—Indian dataset, recipe retrieval, ingredients detection, transformers, transfer learning, resNet-50, denseNet-100**

I. Introduction

Food has a unique ability to bring people together, connecting us to different cultures and places. In today's digital age, food traditions are more widespread than ever, with influencers sharing pictures and videos of the dishes they're consuming on social media. A search for food-related content on Instagram reveals an impressive number of posts, with over 500 million tagged with 'food' and more than 236 million tagged with 'foodie'. This demonstrates the undeniable significance of food in our communities. Behind every meal, there is a complex recipe and a unique backstory, but simply looking at a picture of food on social media does not provide access to the cooking process.

In modern times, we frequently consume food prepared by third-party entities such as restaurants, catering services, and takeaways, often through online delivery apps for added convenience. As a result, access to cooking recipes and ingredient information is limited, leaving us unaware of what we are consuming. Our project's primary goal is to educate people about the food they eat, the ingredients used, and the complex processes involved in its preparation. This knowledge can help individuals with allergies or dietary restrictions to avoid potentially harmful foods. However, this task is challenging due to the vast range of food images available and the significant differences between the ingredients used in various dishes. For example, mashed potatoes and wheat dough may appear similar in pictures but are entirely different. Additionally, even when following the same recipe, dish images may vary significantly, and nearly identical dish images may be associated with different ingredients and cooking instructions. To overcome these challenges, we employ transformers (encoders-decoders) and transfer learning. By using these frameworks, users can identify ingredients and cooking instructions from dish images without having to place an order or physically observe the cooking process.

Indian cuisine stands apart from the rest of the world in terms of its unique taste and preparation methods. It is a fusion of various cultures and regions, resulting in a flawless combination of flavors and textures. Indian food has been influenced by different civilizations, each contributing to its overall development and current form. This diversity is what allows most Indians to appreciate a wide array of flavors in their food. One of the hallmarks of Indian cuisine is the abundant use of spices in every dish. Each spice carries some nutritional and medicinal value, making it an integral part of Indian cooking. Given the enormous variety of Indian ingredients and spices, we recognized the need for an exclusive dataset that focuses on Indian food. Despite finding large datasets like Recipe1M+ and Food101 in our extensive literature survey, we discovered that they did not feature many Indian dishes. This realization motivated us to create a dataset by scraping various websites that contain recipes for Indian food.

The paper's main contributions are as follows:
(1) The creation of an Indian dataset called "Indian Flavours" by scraping cooking websites like Archana's Kitchen and Bawarchi Indian. It contains approximately 16,000 distinct Indian dishes and 1,60,000 images. The dataset is publicly available at Indian Flavours.

(2) The use of transformer architecture, incorporating a ResNet50 encoder and ingredient embeddings, to predict the ingredient list. The ingredient prediction problem was formulated as a set prediction, without imposing any order. The predicted ingredient list was then used to retrieve the dish title, followed by the recipe, from the web.
(3) The creation of a new niche dataset comprising 100 food categories (breakfast and snacks) with 100 images in each category. Transfer learning was used, using the DenseNet model, to train these images with their corresponding classes and ingredients.

II. Related Work

Amaia Salvador et. al.[1], describes an inverse cooking process such that a given dish image generates cooking recipes. It trains the model with the help of image embeddings and ingredient lists. The ingredient list used does not impose any order and is considered a set prediction model. The cooking recipe production process takes a food image as input and produces a series of instructions for the recipe, which were produced with the help of an instruction decoder that takes the ingredient embedding list and image embeddings as input. The image embeddings are extracted with a ResNet-50 encoder, and an ingredient embedding is obtained with the help of decoder architecture and then passed onto one layered embedding which maps each and every food ingredient to a static-size vector. At max 20 food ingredients for every recipe were kept and the food instructions were truncated at a max of 140 words. Adam optimizer was used for training the model until the criteria of early stopping has been met.

Mikhail Fain et. al.[2], introduces Cross-Modal k-Nearest-Neighbours (CKNN). The input of the encoder is taken as one continuous, long document. To train the model, we initialize the word embeddings randomly into the layer of 300 dimensions. The average of all the embeddings is computed in the next layer, then add a linear layer with the sigmoid activation function above for multi-label classification. Cross-entropy (binary) is applied as the function of loss. They refer to this model as the AWE-Encoder. They observed that improvement in performance and results of the independent transformers, text or image, is the same across distinct patterns and in order with results in performance on the Recipe1M dataset, substantiating our framework making use of CKNN.

I. Shchuka et. al.,[3] introduce a method that preprocesses the image given by the user and extracts ingredients from it. The algorithm used is split up into two consequential steps:1) A CNN model to generate characteristics of the image 2) A classifier that uses generated features to extract ingredients For stage one they experimented with already trained models like Resnet-101, Resnet-50, and Densenet161. The best results were obtained using the Densenet161 architecture in the last stage of the pipeline as an encoder. If the logit correlating to the ingredient is greater than the threshold of 0.25, then the ingredient is appended to the prediction list. The next step is to find relevant recipes which is a ranking-based task, where the model accepts the expected list of components as input and searches the database for the best recipes.

Hao Wang et. al.,[4] propose a Structure-aware Generation Network to deal with this problem statement of the generation of food recipes. A recipe2tree module is proposed to catch the hidden sentence-level structured trees for food recipes, which is learned through an unsupervised approach. The tree structures that are obtained are used to supervise the img2tree module. The img2tree module has been proposed for the purpose of recipe tree structure generation from food images, where an RNN is used for conditional tree generation The tree2recipe module is used to encode the deduced structured trees. It was carried out with the help of graph attention networks and improves the performance of recipe generation. On the benchmark Recipe1M dataset, their suggested model could develop the best standard and coherent recipes and attain state-of-the-art performance.

III. Dataset

A. Overview

Data is a crucial component in creating an inverse cooking system that can identify ingredients and cooking instructions from an image of a prepared meal. Deep learning and computer vision techniques can be used to accomplish this task of recipe retrieval from food images, but they require a large amount of pre-processed data to produce optimal results. This data should ideally include the dish title, the ingredients used to prepare the dish, and a picture of the dish.

B. Available Datasets

There are several publicly available datasets that can be used for research in the field of recipe retrieval and food image analysis, including Recipe1M+, Food101, and the IIITD CulinaryDB dataset. However, these datasets have limitations when it comes to Indian cuisine. Recipe1M+ and Food101 mostly contain foreign dishes, and the IIITD CulinaryDB dataset lacks images. Therefore, to address the need for an Indian food dataset, we created the 'Indian Flavours' dataset by scraping cooking websites such as Archana's Kitchen and Bawarchi Indian. This dataset contains over 16,000 unique Indian dishes and 160,000 images, making it a valuable resource for research on Indian cuisine.

C. Dataset Creation and Cleaning

The cooking websites Archana's Kitchen and Bawarchi Indian were scraped using the Selenium module. The websites had multiple pages, with each page containing a fixed number of dishes and a 'NEXT' button for navigation. The URLs of all the pages were first extracted and stored in a text file. This file was used to fetch each page and retrieve the necessary elements for all the dishes on the page. The list of ingredients, image URL, and title of each dish was obtained using specific XPaths. The image URLs were further used to extract the images. The collected data required significant processing, including handling unimportant phrases, special characters, numbers, casing, duplicates, etc. For example, a lady's finger was referred to as bhindi on one website and as okra on another. All

such variations were grouped into a single ingredient class called lady's finger. The ingredients were lemmatized to ensure consistency among tokens; for example, leaf and leaves were changed to leaf. The dish titles were cleaned by removing irrelevant phrases such as "Hyderabadi" from "Hyderabadi Biryani" to obtain precise titles. Duplicate ingredients differing only in casing were merged into a single word. These tasks ensured that the ingredient vocabulary was concise and precise for optimal model performance.

D. Ingredient Vocabulary Creation

After the dataset was created, a standardized and robust ingredient vocabulary was developed, which formed the basis for the multi-label classifier to make predictions. To achieve this, a comprehensive set of ingredients was compiled from all the dishes. Various text processing techniques were applied, such as removing special characters, numbers, and duplicates, and handling different variations of the same ingredient name. The resulting ingredient vocabulary was reduced from an initial 7793 to a final 235 classes, which ensured a concise and precise vocabulary to be used for the model.

TABLE I
STATISTICS OF THE DATASET

Number of ingredients	235
Number of Indian dishes	16,000
Number of images	160,000

E. Formatting the Dataset

The process of formatting large datasets with complex attributes is crucial to ensuring efficient access to the data and files required for training the model. Here, the approach used is to format the attributes of each dish (title, list of ingredients, and image) into two JSON files, with an 8-digit ID assigned to each dish to maintain consistency in title length. The first JSON file maps the dish ID to its attributes, such as ingredients, title, URL, and set, while the second JSON file includes the dish IDs and URLs of the dish images. We limit the number of images per class to 10 so that we have only the images that are relevant to our dish name. The images were then stored in a hierarchical manner on the hard disk using Python libraries like PIL and OS.

IV. PROPOSED METHODOLOGY

A. Transfer Learning

Dataset Creation: A dataset consisting of 100 food categories with 100 images per category, mostly from Indian Breakfast and Snacks categories, was created by scraping images from Google and storing them in separate folders on a local disk. After creating the dataset, two text files were generated: Classes.txt containing all food categories and Ingredients.txt containing the ingredients for each food category.

Transfer learning was applied to the dataset using DenseNet as the model, Adadelta optimizer, and cross-entropy loss and BCEwithlogits loss to store the best model. The classes and their ingredients were one-hot encoded to train the model.

Creation of Text Files: The dataset was then divided into three parts - train, validation, and test data sets, with 70%, 20%, and 10%, respectively. The model was trained on the train set, validated on the validation dataset, and tested on the test dataset.

Application of Transfer Learning: Transfer learning was applied on the dataset using DenseNet as the model, Adadelta optimizer, and cross-entropy loss, and BCEwithlogits loss to store the best model. The classes and their ingredients were one-hot encoded to train the model.

B. Transformers

The architecture consists of a frontend interface where a user uploads a food image, and the model processes the image data to provide an output on the same interface. To generate an ingredient set from an image, the task was divided into three sub-tasks, namely ingredient prediction, title retrieval, and recipe retrieval. Before training the model, the image representation was extracted using the pre-trained model ResNet-50, and the ingredient embeddings were obtained using the decoder architecture to predict the ingredient list. A single layer of embedding was used to map each ingredient to fixed-size vectors. The ingredient decoder consisted of transformer blocks, each block containing two layers of attention followed by a linear layer. The transformer model had multiple blocks of transformers, one linear layer, and one nonlinear layer softmax that gave the distribution of ingredients. The model required two types of sources: the features of the image and the ingredient embeddings. To achieve this, the images and ingredient embeddings were concatenated, and attention was applied to the combined embeddings.

It was noted that there were several common ingredients, such as cloves and chili, that were present in almost every dish. Training the model with these common ingredients would result in a bias towards these ingredients, causing the model to predict them for every image, whether they were present or not. To address this issue, a fixed number of ingredients for every dish were passed, ensuring that each dish had the same number of ingredient embeddings. This number was stored in maxnum labels.

However, there was a risk that the main ingredient could be missed out while training the model if it was not present in the maxnum labels of the dish. To mitigate this problem, a priority list was created consisting of all the main ingredients from their ingredient vocabulary. A script was then written to prioritize these ingredients at the beginning of the ingredient vocabulary, followed by the common ingredients. This way, the vectors of the important ingredients from the ingredient vocabulary were given more priority while training the model, improving the accuracy of ingredient prediction.

Ingredient decoder: During the training process, the order of ingredients was removed since it is not significant for the model. A max-pooling operation was used to combine the total outputs of various time steps.

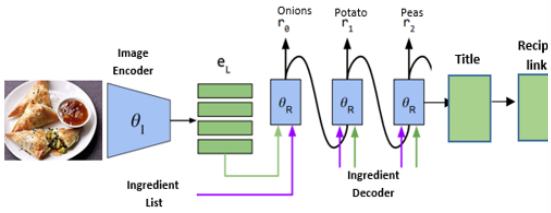

Fig. 1. The model architecture

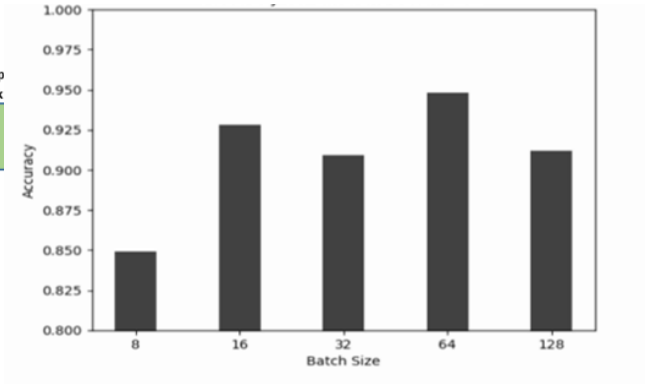

Fig. 2. Graph depicting accuracy of various batch sizes

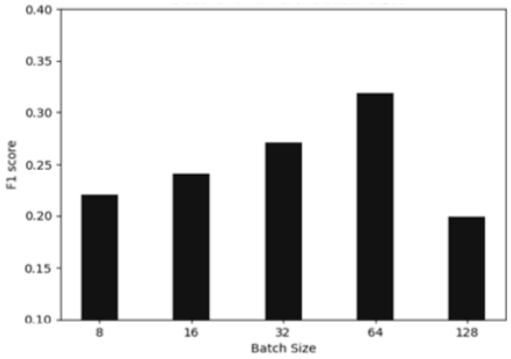

Fig. 3. Graph depicting F1 score of various batch sizes

To avoid ingredient repetition, pre-activation was forced for previously selected ingredients. Additionally, to minimize binary cross-entropy between predicted ingredients, eosLoss was added as a stopping criteria for the prediction of ingredients. To prevent loss of information, eosLoss was not included during the pooling operation and L1 penalty was added empirically. The previously trained model was loaded, the losses of the new epoch was compared to the previous epochs, and stored in the best model parameters to a checkpoint file on the drive. To ensure improved performance, a metric such as the F1 score and accuracy for each epoch was calculated and the model's best checkpoint was saved. The current model was also saved, and if the patience threshold mentioned in the args file was exceeded, training was stopped.

V. Results

We created a website with the use of HTML, ReactJS, and Flask, on which users could upload an image of a dish. After processing the image in the backend, the users get the predicted ingredients of the dish, its title, and a recipe link as an output on the frontend. For the transfer learning architecture, after training the model for different Epochs, the best accuracy was 0.68, yielded by running 32 epochs of the model.

TABLE II
Results on Various Batch Sizes

Batch Size	Accuracy	F1 Score
8	0.849	0.221
16	0.928	0.241
32	0.908	0.271
64	0.948	0.319
128	0.912	0.241

For the transformer architecture, we calculated the accuracy and F1 score by comparing these predicted ingredients with the ingredients of the retrieved title. We trained the model for various epochs, the results of which have been tabulated in Table II. The benchmark model had an F1 score of 0.49. Since the size of our dataset was only 10% of the benchmark dataset, we had initially assumed that we would achieve an F1 score less than half of that achieved by the benchmark model i.e. around 0.25. But, beyond our expectations, we were able to achieve an F1 score of 0.319. As depicted in Fig. 2 and Fig. 3, the best results were achieved for batch size 64. It yielded an accuracy of 0.948 while the F1 score achieved was 0.319.

The ingredients predicted by our model have been compared with the actual ingredients of the dish in Fig. 4. The title predicted by our model has also been shown.

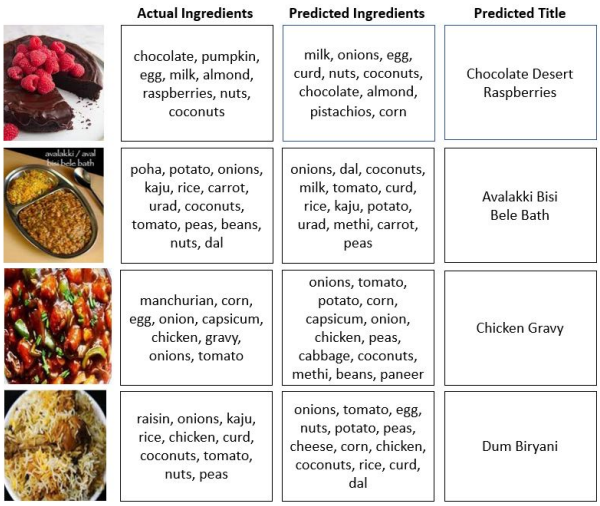

Fig. 4. A comparison of the actual ingredients in the dish with the ingredients predicted by the model. The predicted title has also been mentioned

VI. Conclusion

In this project, we have introduced a new dataset exclusively for Indian dishes. We have discussed an image-to-

recipe retrieval architecture that takes the image of a dish as input and produces its ingredients, title, and recipe link as output. We have formulated the ingredient prediction problem as a set prediction, without imposing any order. With the predicted ingredient list, the title is retrieved, which is further used to retrieve the recipe from the web. We argue that the recipe generation pipeline benefits from an intermediary step of predicting the list of ingredients instead of getting the recipe directly from the image.

Two techniques have been employed to approach the problem of recipe generation from food images: transformer architecture and transfer learning. On one hand, our transformer architecture yielded an F1 score of 0.319 and an accuracy of 0.948. On the other hand, our transfer learning approach achieved an accuracy of 0.68.

In conclusion, our recipe generation model presents notable advancements over prior models in the field. Our use of the top 10 ingredients for both training and evaluation, larger hidden size of dh = 512 for both the encoder and decoder embeddings, and incorporation of transfer learning have led to enhanced performance and accuracy in recipe generation tasks. Our image-to-recipe retrieval architecture, set prediction approach for ingredient prediction, and use of transfer learning provide promising directions for future research in this field. Our work can be extended to other cuisines and has the potential to improve the accessibility of diverse cuisines to individuals worldwide.

VII. Future Work

This study can be extended in multiple directions in the future both in terms of data and the model. As we are making our dataset publicly available with a detailed description about its format, someone having more sources of Indian food data can easily extend the dataset which will give better results. The ingredients detected from this model were used to retrieve the recipe link. It can also be extended to generate the instructions and get the calorie count of the predicted dish.

References

[1] Amaia Salvador et. al., "Inverse Cooking: Recipe Generation From Food Images," IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019

[2] Mikhail Fain et. al., "Dividing and Conquering Cross-Modal Recipe Retrieval: from Nearest Neighbours Baselines to SoT", arXiv:1911.12763v2 [cs.CV] 13 Jul 2021

[3] I. Shchuka, S. Miftakhov, V. Patrushev, M. Tikhonova and A. Fenogenova, "Dish-ID: A neuralbased method for ingredient extraction and further recipe suggestion," 2020 International Conference Engineering and Telecommunication (EnT), 2020

[4] Hao Wang et. al., "Learning Structural Representations for Recipe Generation and Food Retrieval," arXiv:2110.01209v1 [cs.CV] 4 Oct 2021

[5] A. Channam et. al., "Extraction of Recipes from Food Images by Using CNN Algorithm," 2021 Fifth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), 2021.

[6] P. Pandey et. al., "FoodNet: Rec- ognizing Foods Using Ensemble of Deep Networks," in IEEE Signal Processing Letters, vol. 24, no. 12, pp. 1758-1762, Dec. 2017

[7] Amaia Salvador et. al.,"Revamping Cross-Modal Recipe Retrieval with Hierarchical Transformers and Self-supervised Learning," arXiv:2103.13061v1 [cs.CV], 24 Mar 2021

[8] Zhu, Z. et. al., "Food Ingredients Identification from Dish Images by Deep Learning," Journal of Computer and Communications, 9, 85-101, 2021

[9] Ciocca et. al., "Food recognition: a new dataset, experiments, and results," IEEE journal of biomedical and health informatics, p. 588-598, 2016

[10] Sinno Jialin Pan and Qiang Yang, "A survey on transfer learning," IEEE Transactions on Knowledge and Data Engineering, 22(10):1345–1359, 2010.

[11] Rokon et. al., "Food Recipe Recommendation Based on Ingredients Detection Using Deep Learning," 2022

[12] Maheshwari S. et. al., "Recipe Recommendation System using Machine Learning Models," International Research Journal of Engineering and Technology (IRJET), 6(9): p. 366-369, 2019

[13] http://pic2recipe.csail.mit.edu/

[14] https://www.kaggle.com/datasets/dansbecker/food-101

[15] https://cosylab.iiitd.edu.in/culinarydb/

[16] https://www.archanaskitchen.com/

[17] https://bawarchiindian.com/