AI-Powered Food Discovery and Recommender App

Authors: Abhishek Tibrewal (MT23009), Aruba Sood (MT23022), Mayank (MT23045), Purbasha Barik (MT23068), Mehakdeep Kaur (MT23126), Rajith Ramchandran (MT23132)

Problem Statement

Develop an AI-powered Food Discovery and Recommender Application that revolutionizes the culinary experience by leveraging image recognition technology. The application should be capable of scanning an image of a food item, accurately identifying it, and providing a comprehensive suite of features including:

- **Complementary Food Recommendations:** Suggesting additional food items that pair well with the identified dish to enhance the dining experience.
- **Ingredient List Generation:** Displaying a list of ingredients required to prepare the identified dish.
- **Recipe Provision:** Offering a detailed recipe for creating the dish at home.
- **Instructional Video Access:** Providing a direct link to a YouTube video tutorial for step-by-step cooking instructions.

The application aims to cater to food enthusiasts who wish to explore new flavors, aspiring chefs seeking culinary inspiration, and anyone interested in learning about diverse cuisines through an interactive and informative platform. The challenge lies in creating an intuitive and user-friendly interface that delivers accurate food identification and relevant recommendations, thereby enriching the user's gastronomic journey.

Progress (Mid -sem Review 1)

We have made some substantial advancements in our Project. We have done thorough literature review, dataset construction, and preprocessing efforts. The literature review phase provided a comprehensive understanding of existing approaches in culinary applications, identifying gaps and opportunities for innovation. Drawing upon insights from diverse sources, we formulated a clear roadmap for the development of our artificial intelligence-powered food discovery and recommender app.

Subsequently, we have constructed a significant dataset representative of real-world culinary scenarios. This dataset encompasses a varied range of food items captured from various perspectives and conditions to facilitate robust training of image recognition algorithms. This dataset consists of a total of 3,160 images from 63 different classes of food, with each class containing 50 images. Moreover, several preprocessing steps were implemented to enhance the dataset's quality and ensure compatibility with our proposed methodologies. These steps include resizing, grayscaling, flipping, brightness & contrast adjustments, rotation etc.

Overall, the progress achieved in these initial phases sets a solid foundation for the subsequent stages of app development and evaluation, positioning us favorably to deliver a cutting-edge solution to address the identified challenges in culinary applications.

User Interface

The Android app lets users do several main things easily. First, when they open the app, they see a friendly welcome screen. They can then pick photos from their phone's picture gallery and see them on the screen. If something is loading, like a picture or information, there's a little spinning circle to show that it's happening. The app also shows visuals to let users know when things are loading, so they stay informed and engaged while using it.

Recommender System

Content-based filtering is a recommendation technique that suggests items (such as movies, articles, products, etc.) to users based on the features or attributes of those items and the user's preferences. It works by analyzing the characteristics of items and finding similarities between them. Here's how content-based filtering typically works:

Item Representation: Each item is represented by a set of features or attributes. For example, in a movie recommendation system, attributes could include genres, actors, directors, and plot keywords.

User Profile Creation: A user's preferences or profile is also represented using the same set of features. This profile can be constructed based on the user's interaction history, explicit ratings, or stated preferences.

Similarity Calculation: Similarity measures, such as cosine similarity or Pearson correlation, are used to quantify the similarity between items and the user profile. These measures assess how closely the features of an item match the features in the user profile.

Recommendation Generation: Based on the similarity scores, the system identifies items that are most similar to the user's profile. Items with the highest similarity scores are recommended to the user.

Filtering and Ranking: The recommended items may go through additional filtering or ranking steps to ensure relevance and diversity in the recommendations. For example, popular items or items already consumed by the user may be filtered out, and the remaining items may be ranked based on their similarity scores.

Content-based filtering is particularly useful when there is sufficient information available about the items' attributes and when users' preferences can be inferred from their past interactions with the system. It is commonly used in various domains such as e-commerce, content streaming platforms, and information retrieval systems. One of its advantages is that it can provide personalized recommendations even for new or less popular items, as long as their attributes are known.

This Projects defines a function get_recommendations that takes a title (presumably the name of a food item) and a cosine similarity matrix cosine_sim as inputs. Here's a breakdown of what the code does:

It first finds the index of the input title in the indices dictionary. It computes the cosine similarity scores between the input food item and all other food items in the dataset. The similarity scores are then sorted in descending order.It selects the top 5 most similar food items (excluding the input itself) based on their similarity scores. Finally, it returns the names of these 5 most similar food items from the DataFrame df. This code essentially implements a content-based recommendation system for food items based on cosine similarity between their features or attributes

For advance content based filtering we are using;-

Cosine similarity is a measure used to determine how similar two vectors are in a multi-dimensional space by measuring the cosine of the angle between them. It is commonly used in information retrieval, recommendation systems, and natural language processing.

Here's how cosine similarity is calculated and interpreted:

Vector Representation: Each item (document, user profile, etc.) is represented as a vector in a high-dimensional space, where each dimension corresponds to a feature or attribute. For example, in a document recommendation system, each document may be represented by a vector where each dimension represents the frequency of a specific word in the document.

Calculation: Cosine similarity is calculated using the dot product of the two vectors divided by the product of their magnitudes. Interpretation: Cosine similarity ranges from -1 to 1. If the cosine similarity is 1, it indicates that the two vectors are in the same direction (i.e., perfectly similar). If it is -1, it means the vectors are in opposite directions (i.e., perfectly dissimilar).

A value of 0 indicates orthogonality (i.e., the vectors are perpendicular), meaning there is no similarity between them. In recommendation systems, cosine similarity is often used to measure the similarity between items or between user profiles and items. Higher cosine similarity values suggest greater similarity, which can be interpreted as items being more closely related or suitable for recommendation.