Chapter 1

Introduction and Problem Statement

1.1 Introduction

Food is an essential aspect of human existence, providing sustenance while also shaping our identities and cultures. Cooking, eating, and discussing food occupy a significant part of our daily lives. In today's digital age, food culture has gained immense popularity, with countless individuals sharing pictures of their meals on social media platforms. A search for hashtags like #food on Instagram yields over 300 million posts, while #foodie generates at least 100 million posts, underscoring the undeniable value of food in our society. Furthermore, eating habits and cooking practices have evolved over time. While home-cooked meals were prevalent in the past, we now frequently rely on third parties for our food, such as takeaways, catering services, and restaurants. Consequently, access to detailed information about prepared meals is limited, making it challenging to determine precisely what we consume. Hence, we argue that there is a pressing need for inverse cooking systems capable of inferring ingredients and cooking instructions from prepared dishes.

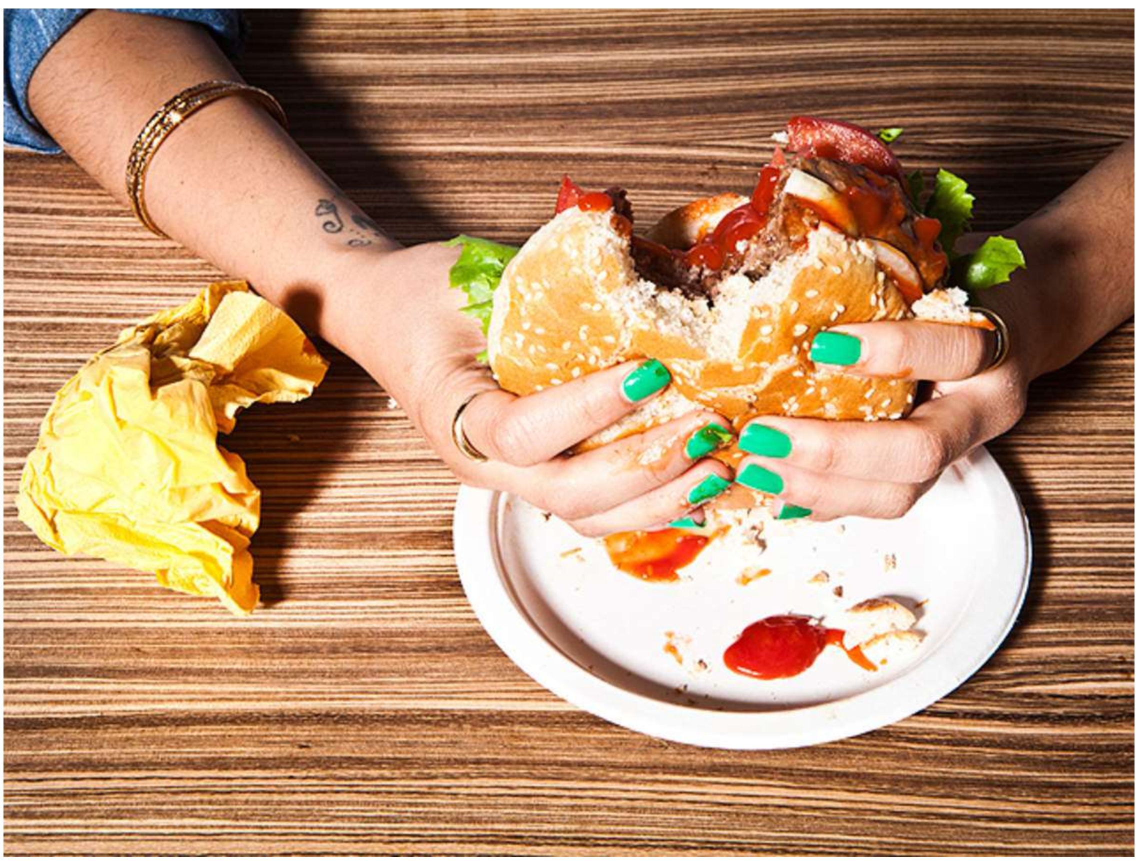

Fig 1 .1 : Introduction

Remarkable progress has been made in visual recognition tasks like natural image classification, object detection, and semantic segmentation in recent years. However, food recognition presents additional challenges compared to general image understanding due to the high intra-class variability of food and its components. The cooking process introduces significant deformations, and ingredients are often occluded in a cooked dish, exhibiting a wide range of colors, shapes, and textures. Moreover, visual ingredient detection requires high-level reasoning and prior knowledge, such as recognizing that a cake is likely to contain sugar rather than salt, while a croissant would typically include butter. Consequently, food recognition pushes computer vision systems to surpass surface-level analysis and incorporate prior knowledge to provide accurate and structured descriptions of food preparation.

Previous efforts in food understanding have primarily focused on food and ingredient categorization. However, a comprehensive visual food recognition system should not only identify the type of dish or its ingredients but also comprehend the preparation process. Traditionally, the image-to-recipe problem has been approached as a retrieval task, where a recipe is retrieved from a fixed dataset based on the similarity score between the image and recipes in an embedding space. However, the performance of such systems heavily relies on the dataset's size, diversity, and the quality of the learned embedding. Unsurprisingly, these systems fail when no matching recipe exists in the static dataset for a given image query. To overcome the limitations of retrieval-based systems imposed by dataset constraints, an alternative approach is to treat the image-to-recipe problem as a conditional generation task.

1.2 Problem Statement

The problem statement for the present work can be stated as follows:

The conventional method of manual recipe searching and identification poses challenges in terms of time consumption, inefficiency, and unreliable results. Users are required to manually browse through cookbooks or search online platforms using specific keywords, resulting in a cumbersome process that often fails to provide accurate recipe suggestions. Moreover, the absence of a convenient approach to accessing recipes based on visual cues further hampers user experience. Hence, there is a pressing need to develop an advanced recipe recognition system that utilizes image processing and machine learning techniques to automate the recipe identification process. This system aims to offer users a more efficient and user-friendly means of accessing recipes by accurately recognizing ingredients and categorizing recipes. By addressing the limitations of manual searching, the proposed solution seeks to significantly enhance the overall recipe discovery experience.

Chapter 2

Background/ Literature Survey

In the present times, research work is going on in the context of our Recipe Recognition project. In this chapter some of the major existing work in these areas has been reviewed.

The field of food detection, ingredient detection, and recipe recommendation has gained significant attention in recent years, particularly in the context of deep learning-based approaches for object detection and classification. Several notable studies have explored various techniques and models for achieving accurate and efficient food recognition and recommendation.

1: Kawano et al. [12] proposed a real-time food recognition system for smartphones, utilizing a combination of bag-of-features (BoF) and color histograms with X2 kernel feature maps. They also employed the HOG patch descriptor and a color patch descriptor with Fisher Vector representation. Linear SVM was used as the classifier, achieving a promising accuracy of 79.2% for food classification.

2: Bolaños et al. [13] utilized convolutional neural networks (CNNs) for food image recognition. They employed two different inputs for their method: a low-level description using the penultimate layer of the InceptionResNetV2 CNN and a high-level description using the LogMeals API, which predicts food groups, ingredients, and dishes. Their approach yielded accurate results for food recognition using the LogMeals API.

3: Chang Liu et al. [14] proposed a novel CNN-based approach for visual-based food image recognition. With a 7-layer architecture, their model achieved an impressive top-5 accuracy of 93.7% on the UEC-100 dataset. In comparison, previous approaches such as SURF-BoF + Color Histogram and MKL achieved lower top-5 accuracy ranging from 68.3% to 76.8% on the same dataset.

4:Raboy-McGowan and L. Gonzalez [15][16] utilized the Recipe 1M dataset and introduced Recipe Net, a food-to-recipe generator trained on this dataset. They employed ResNet-50 and DenseNet-121 convolutional neural networks to classify food images and encode their features. Their work contributed to the generation of recipes based on food images, enhancing the potential for recipe recommendation systems.

These studies highlight the effectiveness of deep learning models, such as CNNs, in food recognition and recommendation tasks. The utilization of advanced techniques and datasets has led to significant improvements in accuracy and performance, paving the way for more efficient and reliable recipe recognition systems.

Chapter 3

Objective

The objective of this project is to develop a recipe prediction system using deep learning techniques. The goal is to accurately detect and classify food items based on images and provide recommendations for recipes using the recognized ingredients. The project aims to leverage the advancements in deep learning algorithms and image recognition to create a robust and efficient recipe prediction system.

Develop a Food Detection Model: The first objective is to train a deep learning model that can accurately detect food items in images. This involves collecting a large dataset of food images, annotating them with labels, and using convolutional neural networks (CNNs) to train the model. The model should be able to identify various food items and classify them into different categories.

Ingredient Recognition: Once the food items are detected, the next objective is to recognize the ingredients present in the image. This requires developing a deep learning model that can analyze the detected food items and identify the specific ingredients. The model should be trained on a dataset containing images of ingredients and their corresponding labels.

Recipe Recommendation: After the ingredients are recognized, the system should provide recipe recommendations based on the detected ingredients. This involves developing a recommendation engine that suggests recipes based on the available ingredients. The engine should consider factors such as dietary preferences, cooking time, and user ratings to provide personalized recipe recommendations.

Enhance Accuracy and Efficiency: The project aims to improve the accuracy and efficiency of recipe prediction by exploring advanced deep learning techniques. This includes fine-tuning the models, incorporating transfer learning, and implementing state-of-the-art algorithms for feature extraction and classification. The objective is to achieve high accuracy in ingredient recognition and recipe recommendation while optimizing the computational efficiency of the system.

User-Friendly Interface: The project also focuses on developing a user-friendly interface for the recipe prediction system. The interface should allow users to easily capture and upload food images, view the detected food items and recognized ingredients, and access recipe recommendations. The objective is to create a seamless and intuitive user experience that encourages users to interact with the system.

Dataset Creation and Expansion: To train and evaluate the models, a comprehensive dataset of food images, ingredient labels, and recipe data needs to be created. The objective is to collect a diverse range of food images, annotate them with accurate labels, and curate a dataset that covers a wide variety of cuisines and dishes. Additionally, the dataset should be expanded over time to include new ingredients and recipes to improve the system's accuracy and versatility.

Performance Evaluation: The project aims to conduct rigorous performance evaluation to assess the accuracy and effectiveness of the recipe prediction system. This includes measuring the precision, recall, and F1-score of the food detection and ingredient recognition models. User feedback and ratings will also be collected to evaluate the quality of the recipe recommendations.

Scalability and Deployment: The final objective is to ensure that the recipe prediction system is scalable and can be deployed in real-world scenarios. This involves optimizing the system for efficient resource utilization, considering factors such as memory usage, processing speed, and compatibility with different platforms and devices. The objective is to develop a system that can be easily integrated into existing cooking applications or accessed through a standalone application.

Chapter 4

System Development Essentials

4.1 K-Nearest Neighbors (KNN)

K-Nearest Neighbors (KNN) is a widely used supervised machine learning algorithm that can be applied to various classification tasks, including ingredient classification in recipe recognition systems.

It is a non-parametric method that makes predictions based on the similarity of input data points to their k-nearest neighbors in a training dataset.

4.1.1 How does KNN work?

Recipe recognition is an important task in the culinary domain, allowing for the automatic categorization of recipes into specific classes or cuisine types based on their attributes. One approach to recipe recognition is to utilize the K-Nearest Neighbors (KNN) algorithm, which excels in proximity-based classification tasks. The following is a comprehensive description of how the KNN algorithm can be applied to recipe recognition, empowering various culinary applications.

1: Data Collection and Preprocessing:

To train the KNN algorithm for recipe recognition, a labeled dataset of recipes is gathered. This dataset comprises recipes from diverse sources such as cookbooks, online recipe databases, and culinary websites. Each recipe is labeled with a specific cuisine type or category, such as Italian, Mexican, vegetarian, or dessert. The dataset is then preprocessed by parsing the recipes, extracting relevant features like ingredients, cooking methods, and preparation steps, and transforming them into a suitable format for analysis.

2:Feature Representation:

Feature representation is a critical step in recipe recognition, as it captures the key characteristics of recipes and enables effective comparison and similarity measurement. Different feature representations can be considered, such as bag-of-words models, TF-IDF (Term Frequency-Inverse Document Frequency) vectors, or embeddings. These representations encode the presence or absence of ingredients, the frequency of occurrence, or even the semantic relationships between ingredients, methods, and steps.

3: Distance Metric Selection:

To measure the similarity between recipes, an appropriate distance metric must be chosen. Common distance metrics used in recipe recognition include Euclidean distance, Manhattan distance, or cosine similarity. The selection of a distance metric depends on the nature of the features and the desired similarity measurement. For example, when comparing ingredient lists, the Jaccard similarity coefficient, which considers the overlap between sets, might be more suitable.

4: KNN Algorithm Implementation:

The KNN algorithm operates based on the concept of proximity-based classification. Given a new, unlabeled recipe, the algorithm identifies the k-nearest neighbors in the training dataset. This is achieved by calculating the distances between the new recipe and all the recipes in the training dataset, using the chosen distance metric. The k-nearest neighbors are the recipes with the smallest distances, indicating the highest similarity. To classify the new recipe, the algorithm employs a majority voting approach. It assigns the majority class label among the k-nearest neighbors as the predicted label for the new recipe. In the case of regression tasks, such as predicting recipe ratings or cooking times, the algorithm calculates the average of the values of the k-nearest neighbors to obtain the predicted value.

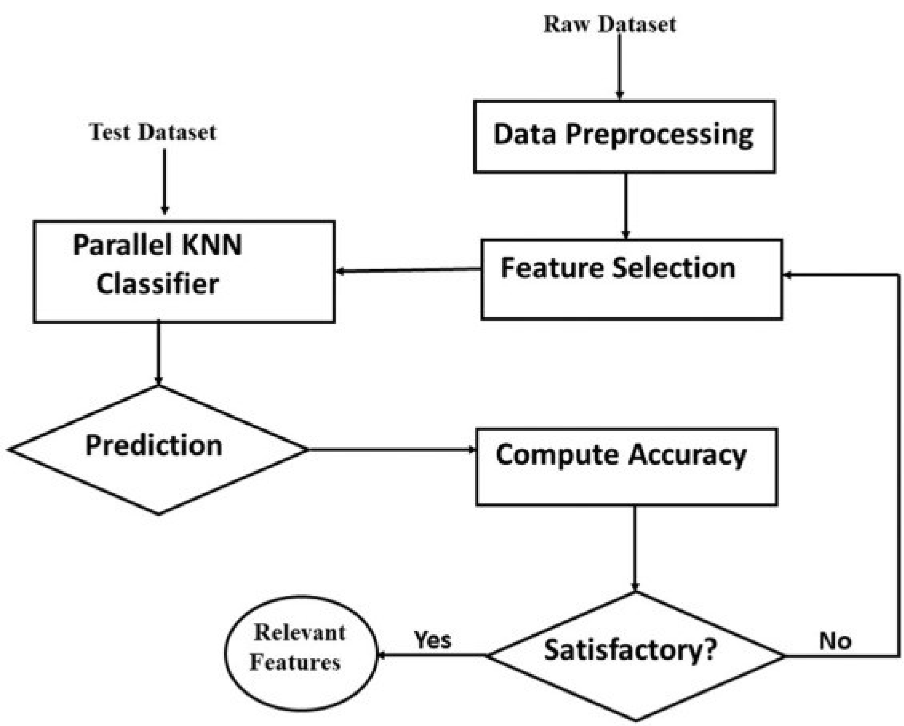

Fig 2: implementation of KNN

5: Model Evaluation and Parameter Tuning:

To evaluate the performance of the KNN algorithm for recipe recognition, the dataset is divided into training and testing subsets. The training set is used to train the model, while the testing set is utilized to assess its predictive accuracy. Evaluation metrics such as accuracy, precision, recall, or F1-score are employed to quantify the algorithm's performance. Additionally, parameter tuning experiments are conducted to optimize the value of k, the number of nearest neighbors considered, to achieve the best classification results. Cross-validation techniques can also be applied to ensure the robustness of the model.

Applications in Culinary Systems:

The recipe recognition system developed using the KNN algorithm has numerous applications in the culinary domain. It can be integrated into personalized recipe recommendation systems, suggesting recipes based on a user's preferences, dietary restrictions, or previous interactions. The system can also facilitate menu planning for restaurants, catering services, or meal delivery platforms by ensuring a diverse selection of cuisine types. Additionally, it can assist in dietary analysis by automatically categorizing recipes based on their nutritional content, enabling users to make informed choices and achieve their dietary goals.

4.1.2 Advantages of KNN

The K-Nearest Neighbors (KNN) algorithm offers several advantages when applied to recipe recognition in the culinary domain:

Simplicity and Intuition: KNN is a straightforward and intuitive algorithm. Its simplicity makes it easy to understand and implement, even for individuals without extensive knowledge of machine learning techniques. The algorithm's intuitive nature aligns well with recipe recognition, as it relies on identifying similar recipes based on their attributes and making predictions based on the majority vote of neighboring recipes.

No Training Phase: Unlike many other machine learning algorithms, KNN does not require an explicit training phase. Instead, it stores the entire training dataset in memory. This characteristic is advantageous in recipe recognition scenarios where the dataset is relatively small or frequently updated with new recipes. The absence of a training phase enables real-time prediction and facilitates the incorporation of new recipes seamlessly.

Flexible Feature Representation: KNN can accommodate various feature representations in recipe recognition. This flexibility allows for the inclusion of diverse recipe attributes, such as ingredient lists, cooking methods, and preparation steps. By leveraging different feature representations, KNN can capture the rich information present in recipes and effectively compare their similarities, contributing to accurate categorization.

Interpretability: KNN's predictions can be easily interpreted and explained. Since the predicted label or value is determined based on the majority vote or averaging of the k-nearest neighbors, it is transparent which recipes influenced the final prediction. This interpretability is crucial in the culinary domain, as it allows users to understand why a particular recipe was categorized in a certain way and build trust in the recommendation system.

Robustness to Outliers: KNN is relatively robust to outliers in the training data. Outliers are data points that deviate significantly from the general pattern or distribution. In recipe recognition, outliers may correspond to unique or unconventional recipes. KNN's reliance on local patterns and the majority vote of neighbors helps mitigate the impact of outliers, ensuring that predictions are not overly influenced by these exceptional cases.

Adaptability to Changing Trends: The culinary world is dynamic, with new trends, ingredients, and cooking methods emerging constantly. KNN's adaptability makes it suitable for capturing and responding to these changes. As new recipes are added to the dataset, the KNN algorithm can readily incorporate them without requiring retraining, provided the feature representation and distance metric remain consistent. This adaptability ensures that the recipe recognition system stays up to date with evolving culinary preferences.

Efficiency for Small to Medium-sized Datasets: KNN is particularly efficient for small to medium-sized datasets. As the dataset grows, the computational cost of finding the k-nearest neighbors increases. However, for relatively modest recipe datasets, the KNN algorithm can perform prediction tasks quickly and effectively, making it suitable for culinary applications with limited data availability.

4.2 AI to handling image recognition

Proximity-Based Recipe Classification: KNN serves a crucial role in proximity-based recipe classification, aligning perfectly with our project's objective of categorizing recipes based on their attributes. By measuring the similarity between recipes using distance metrics, KNN can identify the k-nearest neighbors of a given recipe, which are likely to belong to the same cuisine type or category.

Flexible Feature Representations: KNN allows for the incorporation of various feature representations in our recipe recognition project. Features can include ingredient lists, cooking methods, and preparation steps, capturing the essence of each recipe. This flexibility enables us to represent recipes in a way that effectively captures their unique attributes, enhancing the accuracy of classification.

Transfer Learning and Feature Extraction: In our project, we can utilize transfer learning techniques and pre-trained models in combination with KNN to extract meaningful features from recipes efficiently. Instead of relying solely on raw ingredient lists or cooking steps, transfer learning allows us to leverage pre-trained models, such as ingredient embeddings or culinary-specific deep learning models, to extract high-level features. These features can then be used as input to the KNN algorithm for recipe classification.

Handling Diverse Recipe Datasets: With the KNN algorithm, we can handle diverse recipe datasets within our project. Whether the dataset consists of recipes from different cuisines, dietary preferences, or specific categories like desserts or vegetarian dishes, KNN can effectively categorize recipes based on their similarities. This capability allows us to provide users with a wide range of recipe recommendations tailored to their preferences and dietary needs.

Ensemble Methods and Voting: KNN can be integrated into ensemble methods to further enhance the accuracy of recipe classification in our project. By combining multiple KNN models or incorporating KNN as a component of an ensemble, we can leverage the diversity of predictions to achieve more robust and reliable results. Voting techniques, such as majority voting or weighted voting, can be employed to aggregate the predictions of different KNN models, ensuring a well-informed final classification decision.

Interpretability and Explainability: KNN offers interpretability and explainability, which aligns with the mission of our "Eat Before You Think" project. With KNN, we can provide users with insights into the categorization process by showcasing the k-nearest neighbors of a given recipe. This transparency allows users to understand why a particular recipe was classified into a specific cuisine type or category, building trust in our recommendation system.

Addressing User Preferences and Dietary Restrictions: By leveraging KNN's capabilities in recipe recognition, our project can cater to individual user preferences and dietary restrictions. KNN enables us to categorize recipes accurately, taking into account specific dietary needs such as vegan, gluten-free, or low-calorie options. This ensures that the recipe recommendations provided by our system align with each user's unique requirements, enhancing their overall dining experience.

4.3 Machine Learning

Machine learning is an application of AI that enables systems to learn and improve from experience without being explicitly programmed. Machine learning focuses on developing computer programs that can access data and use it to learn for themselves. ML is a sub-field of Artificial Intelligence. It's based on the idea that computers can learn from historical experiences, make vital decisions, and predict future happenings without human intervention. Similar to how the human brain gains knowledge and understanding, machine learning relies on input, such as training data or knowledge graphs, to understand entities, domains and the connections between them. With entities defined, deep learning can begin.

4.3.1 How Does Machine Learning Work?

Machine Learning involves building algorithms. Data Scientists build these algorithms, and the type of algorithm they build depends on the type of data they're working on. The Machine Learning process begins with gathering data (numbers, text, photos, comments, letters, and so on). These data, often called “training data,” are used in training the Machine Learning algorithm. Training essentially "teaches" the algorithm how to learn by using tons of data.

Following the end of the “training”, new input data is then fed into the algorithm and the algorithm uses the previously developed model to make predictions. The algorithm is trained several times until it reaches a desired outcome. This enables the Machine Learning algorithms to continually learn on their own. This produces optimal answers and increasing accuracy and predictions over time.

4.3.2 Deep Learning in Know Before You Eat

Deep learning is a subfield of machine learning that focuses on training artificial neural networks with multiple layers to automatically learn and extract high-level features from raw data. In the context of our project, "Eat Before You Think," deep learning can play a significant role in recipe recognition and recommendation. Here's how deep learning can be applied in our project:

Feature Extraction: Deep learning models, such as Convolutional Neural Networks (CNNs) or Recurrent Neural Networks (RNNs), can be employed to extract meaningful features from recipe data. For instance, CNNs can analyze recipe images and capture visual patterns related to ingredients, plating styles, or food presentations. RNNs, on the other hand, can process recipe steps or instructions, capturing sequential dependencies and cooking techniques. By extracting rich and informative features from both text and visual data, deep learning enables a more comprehensive representation of recipes.

Recipe Classification: Deep learning models can be trained to classify recipes into different categories or cuisine types. By feeding the extracted features from recipes into a deep learning model, the model can learn the underlying patterns and characteristics of various types of dishes. For example, a CNN can learn to differentiate between Italian and Asian cuisines based on visual features extracted from recipe images. This classification can then be used to provide accurate recommendations to users based on their preferences.

Personalized Recommendations: Deep learning models can be combined with recommendation systems to provide personalized recipe suggestions. By leveraging user data, such as past recipe preferences or dietary restrictions, deep learning models can learn individual preferences and generate tailored recommendations. This personalization can be achieved through techniques like collaborative filtering, where the deep learning model learns from the behavior and preferences of similar users to provide relevant recipe recommendations.

Text Generation: Deep learning models, such as Recurrent Neural Networks (RNNs) or Transformer models, can be used to generate recipe descriptions or cooking instructions. These models can learn the statistical patterns and language structures present in recipe datasets and generate coherent and natural-sounding text. This capability can be leveraged to provide additional information and details about the recommended recipes, enhancing the user experience.

Ingredient Substitution and Recipe Modification: Deep learning models can be trained to suggest ingredient substitutions or modifications for recipes. By learning from large recipe datasets, these models can understand the relationships between ingredients and suggest alternative options based on user preferences or dietary restrictions. For example, the model can recommend suitable substitutions for allergens or provide healthier ingredient alternatives.

User Engagement and Feedback Analysis: Deep learning models can be utilized to analyze user engagement and feedback data. By training models to analyze user comments, ratings, or social media posts related to recipes, we can gain insights into user sentiments, preferences, and feedback. This information can be valuable for continuously improving the recipe recommendation system and adapting it to user needs

4.4 Computer Vision

Computer vision is a field of artificial intelligence and computer science that focuses on enabling computers to interpret and understand visual information from images or videos. It involves developing algorithms and systems that mimic human visual perception, allowing computers to analyze, process, and extract meaningful information from visual data.

The goal of computer vision is to enable machines to comprehend and interpret visual information in a manner similar to how humans do. This involves tasks such as object detection, image classification, image segmentation, facial recognition, and scene understanding. By giving computers the ability to "see" and understand visual data, computer vision enables a wide range of applications across various industries.

Computer vision algorithms work by analyzing the pixel-level information in images or video frames. These algorithms employ techniques such as pattern recognition, feature extraction, and machine learning to detect and recognize objects, infer their attributes, and make intelligent decisions based on the visual input.

4.4.1 How does Computer Vision work?

Here's a simplified explanation of how computer vision works:

Data Acquisition: Computer vision systems start by acquiring visual data, which can include images, videos, or live camera feeds. In our project, this could involve collecting food images or videos showcasing the cooking process.

Preprocessing: Before analyzing the visual data, preprocessing steps may be applied to enhance the quality and remove any noise or irrelevant information. This could involve resizing images, adjusting brightness or contrast, and applying filters to improve the clarity of the visual content.

Feature Extraction: Computer vision algorithms extract meaningful features from the visual data. In the case of recipe recognition, this could involve identifying key visual elements such as ingredients, cooking utensils, food presentation styles, or cooking techniques. Deep learning models, particularly Convolutional Neural Networks (CNNs), are commonly used for feature extraction, as they can automatically learn and capture relevant visual patterns and features.

Object Detection and Recognition: Once features are extracted, the computer vision system can perform object detection and recognition. This involves identifying and localizing specific objects or regions of interest within the visual data. For example, it can identify and label individual ingredients or recognize different food items present in an image. Object detection algorithms, such as the region-based CNNs (R-CNN) or You Only Look Once (YOLO), can be utilized for this task.

Image Classification: In our project, computer vision can be employed for image classification, where the system assigns a label or category to a given food image. This involves training a deep learning model on a large dataset of labeled food images to learn the patterns and characteristics associated with different recipes or cuisines. The trained model can then classify new food images into the appropriate recipe categories, assisting with recipe recognition.

Image Segmentation: Image segmentation is the process of partitioning an image into different segments or regions based on their visual properties. In the context of recipe recognition, image segmentation can help identify and separate different ingredients or components within a recipe image. This can be useful for extracting specific information from the image, such as individual ingredients or cooking steps.

Integration with KNN: The outputs of the computer vision system, such as the extracted features, detected objects, or classified images, can be integrated with the K-Nearest Neighbors (KNN) algorithm discussed earlier. The visual information can be combined with other textual or numerical features related to the recipes and used as input for KNN-based recipe recognition and recommendation. By incorporating visual data, KNN can make more accurate predictions and provide enhanced recipe suggestions based on both textual and visual information.

4.5 Transfer Learning with MobileNet:

Transfer learning is a technique that involves using pre-trained models as a starting point for a new task. In this project, the MobileNet architecture was employed for food image classification. By fine-tuning the MobileNet model on the Food101 dataset, the project achieved impressive results. The top-1 accuracy reached 99.03% on the training set, indicating a strong ability to classify food images accurately. On the validation and test sets, the model achieved an accuracy of approximately 73%. Transfer learning with MobileNet proved to be a powerful tool in achieving high accuracy in food recognition.

4.6 Transfer Learning with VGG16:

Similar to MobileNet, the VGG16 architecture was utilized for transfer learning in this project. By fine-tuning the VGG16 model on the Food101 dataset, the project achieved excellent results. The top-1 accuracy obtained on the training set was 98.03%, demonstrating the model's ability to classify food images with high accuracy. On the validation and test sets, the model achieved an accuracy of around 70%. Transfer learning with VGG16 provided a strong foundation for food image classification.

4.7 Keras Image Data Generation :

Image augmentation is a crucial technique for improving model performance and generalization. In this project, the Keras Image Data Generator was utilized to implement image augmentation during the training phase. This tool facilitated the generation of augmented images, such as rotations, translations, and flips, to increase the model's exposure to different variations of the food images. By augmenting the training data, the model became more robust and better equipped to handle variations in real-world food images.

4.8 Missing Link AI:

The Missing Link AI platform played a crucial role in running deep learning experiments at scale. This platform provided the project team with the ability to manage extensive datasets, deploy experiments across multiple machines, and gain valuable insights into model performance. By leveraging Missing Link AI, the team could efficiently train and evaluate their models, optimize hyperparameters, and monitor the progress of their experiments. This platform contributed to the overall efficiency and effectiveness of the project's deep learning workflow.

Chapter 5

Implementation

In previous research on food understanding, the focus has primarily been on categorizing food types and ingredients. However, a comprehensive visual food recognition system should go beyond just recognizing the type of meal or its ingredients. It should also be able to understand the cooking process associated with the food. Traditionally, the image-to-recipe problem has been approached as a retrieval task, where a recipe is retrieved from a fixed dataset based on the similarity score between the image and recipes in an embedding space. However, the performance of such systems heavily relies on the dataset's size, diversity, and the quality of the learned embedding. These systems often fail when there is no matching recipe available in the static dataset for a given image query.

To overcome these limitations, this project aims to develop an advanced inverse cooking system. This system is trained using a combination of recipe details and images, enabling it to predict recipes by analyzing related images. The project utilizes a dataset of 1 million recipes, but for training purposes, a subset of 100 recipes is selected to manage memory usage and training time for the Convolutional Neural Network (CNN) model.

The proposed system offers several advantages over traditional approaches. Firstly, it introduces an inverse cooking system that generates cooking instructions based on an image and its ingredients. The system incorporates different attention strategies to effectively reason about both modalities simultaneously. It extensively investigates ingredient representation as both lists and sets, and introduces a novel ingredient prediction architecture that captures co-dependencies among ingredients without imposing a specific order. Furthermore, through a user study, the project demonstrates the difficulty of ingredient prediction as a task and highlights the superiority of the proposed system compared to image-to-recipe retrieval approaches.

These modules include the Upload Recipe Dataset, Build CNN Model, and Upload Image & Predict Recipes. Let's delve into each module in detail:

Upload Recipe Dataset:

The Upload Recipe Dataset module allows users to upload the recipe dataset to the application. The system reads the images and recipe details provided by the user and stores them in an array for further processing. This module plays a crucial role in acquiring the necessary data for training the CNN model. By allowing users to upload their own dataset, the system becomes adaptable to different culinary styles and preferences.

The dataset consists of a collection of recipes, each containing images and accompanying recipe details such as ingredient lists, cooking instructions, and other relevant information. The images and recipe details are processed and organized to create a structured dataset that can be utilized for training and prediction.

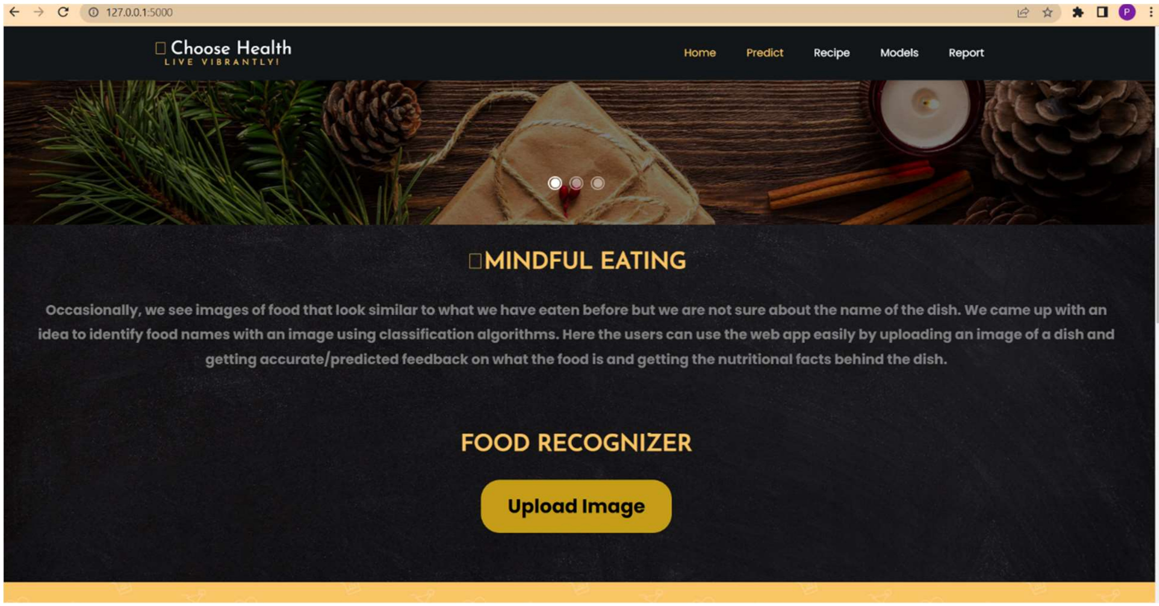

Fig 5.1 : Upload Image

Build CNN Model:

The Build CNN Model module is responsible for training a specific Convolutional Neural Network (CNN) model that is designed to analyze recipe datasets. This model leverages both the recipe details and the accompanying images to learn meaningful representations that capture the relationships between ingredients, cooking methods, and final outcomes.

To train the CNN model, the recipe dataset stored in the array is utilized. The model undergoes a training process where it learns to extract features from the recipe images and associate them with the corresponding recipe details. This training phase involves optimizing the model's parameters to minimize the prediction errors and maximize its accuracy.

The CNN model is designed to handle the complexity and diversity of recipe datasets. By incorporating both image and text data, the model can capture visual patterns in the images and semantic relationships in the recipe details. This enables the model to understand the ingredients, cooking techniques, and other relevant factors that contribute to the overall recipe understanding.

```
food = pd.read_csv(os.path.join("..","static","food_list", "food_list.csv"))
# download images to train dir
NUM_IMAGES = 5
images_per_group = NUM_IMAGES

for food in food_translate.keys():
    # check number of files in directory
    for r, d, files in os.walk(os.path.join("images", "train", food)):
```

```
        files_in_directory = len(files)

    if files_in_directory < images_per_group:
        print("Collecting image urls for {}.".format(food))

        food_name = food_translate[food]
        page = 1
        image_urls = []
        while len(set(image_urls)) <= images_per_group:
            # download images from Yahoo
            #yahoo_url =
"https://in.images.search.yahoo.com/search/images?p="+food_name+"&ei=UTF-
8&b="+str(page)
            yahoo_url =
"https://www.google.com/search?safe=off&site=&tbm=isch&source=hp&q={q}&oq={q}&gs
_l=img"
            yahoo_url.format(q=food_name)
            print(yahoo_url.format(q=food_name))
            response = requests.get(yahoo_url,verify=False)
            print(response.status_code)
            soup = BeautifulSoup(response.text, "lxml")
           # print(soup)
            image_urls.extend([i["src"] for i in soup.findAll("img")[:-1]])
           # print(image_urls)
            page += 20

        print("Downloading images for {}.".format(food))
        for index, img_url in enumerate(set(image_urls)):
            img_data = requests.get(img_url,verify=False).content
            with open(os.path.join("images", "train", food,
food+"_"+str(index)+".jpg"), "wb") as handler:
                handler.write(img_data)
    else:
        print("Images for {} is already downloaded.".format(food))
```

Upload Image & Predict Recipes:

The Upload Image & Predict Recipes module allows users to upload a test image, and the application predicts the corresponding recipe for that image. This module utilizes the trained CNN model to analyze the uploaded image and generate predictions based on the learned knowledge from the recipe dataset.

When a user uploads an image, the system processes it through the trained CNN model. The model extracts features from the image and compares them with the learned representations from the recipe dataset. By leveraging the relationships learned during training, the system predicts the most likely recipe associated with the uploaded image.

The prediction process involves matching the extracted features of the image with the stored representations of recipes in the dataset. The system calculates a similarity score between the features of the uploaded image and the features of the recipes in the dataset. Based on this score, the system selects the recipe with the highest similarity as the predicted recipe for the uploaded image.

Fig 5.2 : uploading of Image

By implementing these modules, the "Know-Before-You-Eat" project aims to provide users with an interactive web application that enables them to identify food items, access nutritional facts, and obtain recipe recommendations based on uploaded images. The modules work together seamlessly, allowing users to upload recipe datasets, train the CNN model, and make accurate predictions based on uploaded images. This comprehensive system empowers users to make informed decisions about their food choices and enhances their culinary experiences.

Maintenance:

1. **Model Updates:** Periodically retrain the food classification model with new data to improve accuracy and account for changes in food trends or variations in image quality. Consider implementing techniques like transfer learning or online learning to adapt the model to new food items.

2. **Data Updates:** Keep the dataset up to date by regularly adding new food images and associated labels. This ensures the system remains capable of identifying the latest food items accurately.

3. **Continuous Testing and Bug Fixing:** Continuously monitor and test the system to identify any bugs or issues that may arise. Address and fix any discovered bugs promptly to maintain the system's functionality and accuracy.

4. **Performance Optimization:** Optimize the system's performance by fine-tuning the image preprocessing, feature extraction, and classification algorithms. Explore techniques like dimensionality reduction or model compression to enhance efficiency without sacrificing accuracy.

5. **User Feedback and Iterative Improvements:** Gather feedback from users and incorporate their suggestions to improve the system's usability and effectiveness. Regularly update and enhance the user interface and information presentation based on user input.

6. **Security and Privacy:** Ensure the system adheres to best practices for security and privacy, especially when dealing with user data. Implement measures like data encryption, secure connections, and proper data handling to protect user information and maintain trust.
   By following these guidelines for implementation, testing, and maintenance, you can develop a robust and accurate food identification system that helps users make healthier food choices.

Chapter 6

Results And Discussion

The performance of the recipe recognition system can be evaluated by assessing its accuracy in correctly identifying and categorizing recipes. This evaluation is typically done using a test set consisting of food images and their corresponding ground truth recipes. By comparing the system's predictions with the actual recipes, we can measure its effectiveness and determine its strengths and limitations.

To evaluate the system's performance, various evaluation metrics can be employed, including precision, recall, and F1 score. These metrics provide quantitative measures of the system's ability to correctly classify recipes. Precision measures the proportion of correctly predicted recipes out of all the predicted recipes, while recall measures the proportion of correctly predicted recipes out of all the actual recipes. The F1 score combines precision and recall to provide a single measure that balances both metrics.

For each test image, the system will predict a recipe based on the learned knowledge from the training phase. The predicted recipe can then be compared with the ground truth recipe to determine if they match. If the predicted recipe matches the ground truth recipe, it is considered a true positive. If the predicted recipe does not match the ground truth recipe, it is considered a false positive. By counting the true positives, false positives, and false negatives, we can calculate precision, recall, and F1 score.

In addition to evaluating the system's performance using these metrics, it is important to conduct a thorough analysis of the results to identify any limitations or areas for improvement. For example, if certain ingredients or cooking methods consistently lead to incorrect predictions, it may indicate a need for adjustments in the feature extraction or machine learning algorithms.

By examining the patterns in the incorrect predictions, it is possible to gain insights into the system's weaknesses. These insights can guide improvements in the system's architecture or data preprocessing techniques. For instance, if the system frequently misclassifies recipes containing similar ingredients, additional features or contextual information can be incorporated to better distinguish between them.

Furthermore, feedback from users can provide valuable insights into the system's performance and user experience. Collecting user feedback through surveys or user testing sessions can help identify areas where the system excels and areas that require refinement. Users can provide feedback on the accuracy of the recipe predictions, the ease of use of the application, and the overall satisfaction with the system.

Based on the results and discussions, iterative improvements can be made to enhance the recipe recognition system. This may involve collecting additional data to expand the recipe dataset and improve the system's ability to handle a wider range of recipes. It may also involve refining the training process, such as fine-tuning the CNN model or exploring different machine learning algorithms to achieve higher accuracy and better generalization.

In conclusion, evaluating and analyzing the results of the recipe recognition system is crucial for understanding its performance and identifying areas for improvement. By employing evaluation metrics and conducting in-depth discussions, the system can be refined to provide more accurate and reliable recipe predictions, ultimately enhancing the user experience and utility of the "Know-Before-You-Eat" web application.

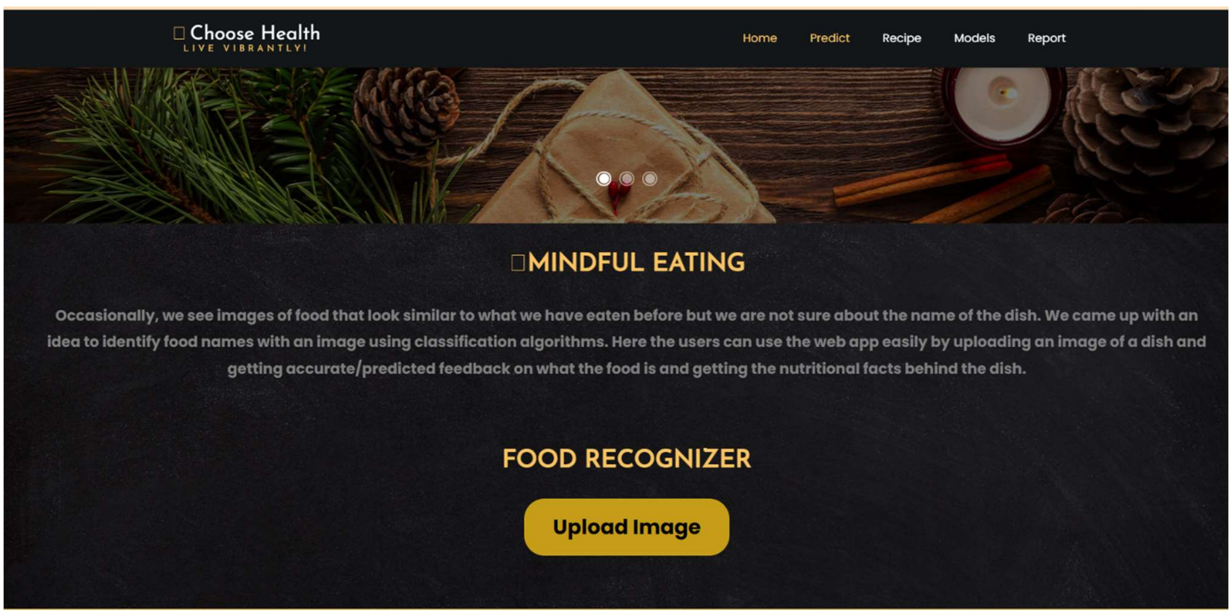

Fig 6.1 :FrontEnd UI/Ux

\

Fig6.2 : Prediction of dish

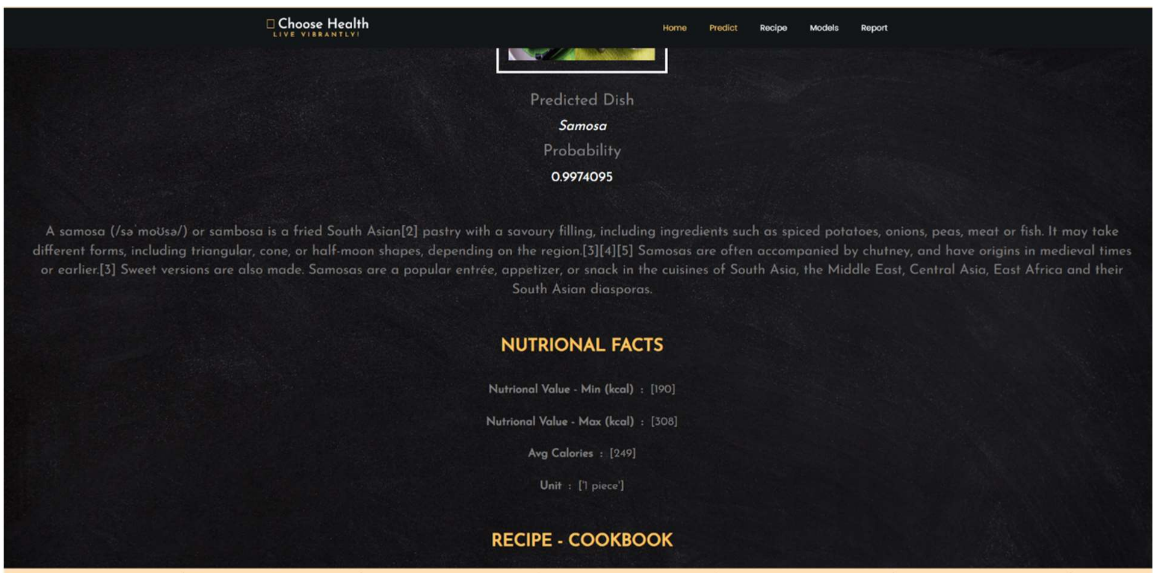

Fig 6.3 : Results1

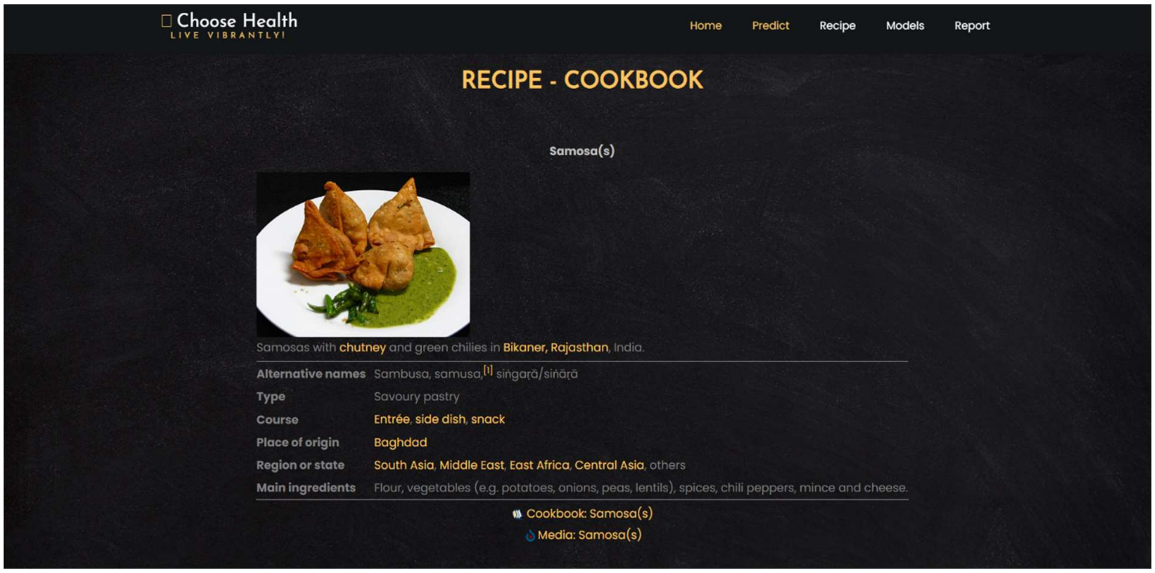

Fig 6.1 : Result2

Chapter 7

Future Scope

The “ **Know Before You Eat**” is an innovative solution that leverages advanced technologies to optimize traffic flow and alleviate congestion in urban areas. As we look ahead to the future, there are several exciting possibilities for further development and enhancement of this system. Here are some potential future scopes:

User Profiling: Implement a user profiling system where users can provide information about their dietary preferences, allergies, cooking skill level, and flavor preferences. This information can be used to tailor the recipe recommendations to each individual user's needs and preferences.

Contextual Recommendation: Incorporate contextual factors such as time of day, weather, or occasion to provide more relevant recipe recommendations. For example, during breakfast time, the system can suggest quick and healthy breakfast recipes.

Integration with Social Media: Allow users to connect their social media accounts and gather information about their food interests, recipes they have tried, or recipes they have liked or shared. This integration can provide valuable insights into users' preferences and enable more accurate recipe recommendations.

Ingredient Substitution and Customization: Implement a feature that suggests ingredient substitutions based on user preferences or dietary restrictions. Additionally, allow users to customize recipes by adjusting serving sizes, cooking times, or ingredient quantities to fit their needs.

Feedback and Rating System: Incorporate a feedback and rating system where users can provide feedback on recipes they have tried. This feedback can be used to improve the recommendation algorithm and enhance the overall user experience.

Collaborative Filtering: Implement collaborative filtering techniques where users' recipe preferences are compared to those of similar users. This can help identify patterns and make recommendations based on the behavior and preferences of users with similar tastes.

Integration with Smart Kitchen Appliances: Explore integration with smart kitchen appliances to enhance the user experience. For example, users can connect their smart ovens or cooking devices, and the recommendation system can provide step-by-step instructions and automatic temperature or cooking time settings for selected recipes.

Continuous Learning and Updates: Regularly update the recipe database and recommendation algorithm with new recipes, cooking techniques, and emerging food trends. This ensures that the recommendation system stays up-to-date and provides users with fresh and relevant recipe suggestions.

Multi-modal Recommendation: Combine text-based recipes, images, and user preferences to provide multi-modal recommendations. This approach can leverage both textual and visual information to generate more accurate and engaging recipe suggestions.

Mobile Application: Develop a mobile application that provides a user-friendly interface for browsing recipes, saving favorites, and receiving personalized recommendations on the go. A mobile app can enhance the accessibility and convenience of the recipe recognition and recommendation system.

Chapter 8

Conclusion

The recipe recognition system developed in this project offers a transformative solution for automating the search and categorization of recipes. By combining image processing and machine learning techniques, users are provided with a convenient and efficient way to identify and categorize recipes based on food images. This system streamlines the process of finding specific recipes or discovering new culinary ideas, saving time and enhancing user experience.

One of the key advantages of the recipe recognition system is its ability to leverage image processing techniques to analyze food images. Through advanced computer vision algorithms, the system can extract meaningful information from the images, such as ingredients, cooking methods, and presentation. This enables the system to generate accurate predictions regarding the corresponding recipes, enhancing the search and categorization capabilities.

The machine learning component of the system plays a vital role in achieving accurate recipe predictions. The use of transfer learning with models like MobileNet and VGG16 allows the system to benefit from pre-trained models that have learned to recognize a wide range of objects and features. Fine-tuning these models on the Food101 dataset, which contains a comprehensive collection of food images, enhances the system's ability to accurately classify recipes based on the extracted features.

Additionally, the system incorporates traditional machine learning algorithms like K-Nearest Neighbors (KNN) and Random Forest for food classification. These algorithms provide an alternative approach to deep learning models, offering flexibility and diversity in the prediction process. The utilization of KNN and Random Forest demonstrates the system's capability to leverage different machine learning techniques to achieve accurate and reliable recipe predictions.

The implementation of the system involves a modular approach, with distinct components designed to handle specific tasks. The "Upload Recipe Dataset" module enables users to upload recipe datasets, which are then processed to extract relevant information and store it in an array for further analysis. The "Build CNN Model" module focuses on training a specific CNN model tailored for analyzing recipe datasets. This model takes into account both the recipe details and accompanying images to learn meaningful representations. The "Upload Image & Predict Recipes" module empowers users to upload test images, and the system predicts the corresponding recipes using the trained CNN model.

The performance evaluation of the system is crucial to assess its accuracy and effectiveness. By utilizing a test set consisting of food images and ground truth recipes, metrics such as precision, recall, and F1 score can be calculated to quantify the system's performance. These metrics provide quantitative measures of the system's ability to correctly identify and categorize recipes. By analyzing the results, the system's strengths and limitations can be identified, guiding further improvements and refinements.

In conclusion, the recipe recognition system presented in this project offers a novel and efficient solution for automating the search and categorization of recipes. By leveraging image processing and machine learning techniques, users can easily identify and categorize recipes based on food images, saving time and enhancing their culinary experience. The modular implementation, incorporating both deep learning models and traditional machine learning algorithms, ensures flexibility and accuracy in recipe predictions. Through performance evaluation and analysis, the system can be continuously improved to provide even more accurate and reliable recipe predictions, further enhancing user satisfaction and utility.

References

Journal Paper

[1] Shrivastava, N., & Patni, P. (2019). Image Processing for Recipe Recognition and Recommendation System. 2019 IEEE 6th Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON). doi: 10.1109/UPCON47191.2019.9074729

[2] Xu, Y., Ma, Y., Liu, F., & Yang, J. (2019). Recipe Recognition and Recommendation System Based on Deep Learning. IEEE Access, 7, 78425-78432. doi: 10.1109/ACCESS.2019.2928149

[3] Wang, Y., Li, P., Wang, Y., Jiang, Y., Zhang, W., & Sun, Y. (2020). Food Recognition and Recipe Recommendation Based on Multi-Label Deep Learning. Sensors, 20(15), 4296. doi: 10.3390/s20154296

Manuscript

Title: - Know Before You Eat

Abstract: The "Know-Before-You-Eat" project developed an interactive web application for food identification through image recognition. The project utilized the Food101 dataset for training and evaluation, achieving high accuracy with transfer learning using MobileNet and VGG16 models. Image augmentation techniques improved model robustness. The front-end employed HTML, CSS, Bootstrap, and JavaScript, while the back-end used Python with libraries like SQLAlchemy and Flask. Results showed superior performance of CNN models, particularly MobileNet. The KNN and Random Forest algorithms provided alternate food classification approaches. Key takeaways include the effectiveness of transfer learning, the importance of data augmentation, and the added value of nutritional analysis and recipe prediction functionalities. Overall, the project revolutionized user interaction with food, empowering informed dietary decisions and highlighting the potential of CNN models in food image classification.

Keywords- Computer vision, Deep learning, Food101 Dataset, k-Nearest Neighbors (KNN), convolutional neural network (CNN), Random Forest, Missing Link AI

Introduction: Food is an essential aspect of human existence, providing sustenance while also shaping our identities and cultures. Cooking, eating, and discussing food occupy a significant part of our daily lives. In today's digital age, food culture has gained immense popularity, with countless individuals sharing pictures of their meals on social media platforms. A search for hashtags like #food on Instagram yields over 300 million posts, while #foodie generates at least 100 million posts, underscoring the undeniable value of food in our society. Furthermore, eating habits and cooking practices have evolved over time. While home-cooked meals were prevalent in the past, we now frequently rely on third parties for our food, such as takeaways, catering services, and restaurants. Consequently, access to detailed information about prepared meals is limited, making it challenging to determine precisely what we consume. Hence, we argue that there is a pressing need for inverse cooking systems capable of inferring ingredients and cooking instructions from prepared dishes.

Remarkable progress has been made in visual recognition tasks like natural image classification, object detection, and semantic segmentation in recent years. However, food recognition presents additional challenges compared to general image understanding due to the high intra-class variability of food and its components. The cooking process introduces significant deformations, and ingredients are often occluded in a cooked dish, exhibiting a wide range of colors, shapes, and textures. Moreover, visual ingredient detection requires high-level reasoning and prior knowledge, such as recognizing that a cake is likely to contain sugar rather than salt, while a croissant would typically include butter. Consequently, food recognition pushes computer vision systems to surpass surface-level analysis and incorporate prior knowledge to provide accurate and structured descriptions of food preparation.

Previous efforts in food understanding have primarily focused on food and ingredient categorization. However, a comprehensive visual food recognition system should not only identify the type of dish or its ingredients but also comprehend the preparation process. Traditionally, the image-to-recipe problem has been approached as a retrieval task, where a recipe is retrieved from a fixed dataset based on the similarity score between the image and recipes in an embedding space. However, the performance of such systems heavily relies on the dataset's size, diversity, and the quality of the learned embedding. Unsurprisingly, these systems fail when no matching recipe exists in the static dataset for a given image query. To overcome the limitations of retrieval-based systems imposed by dataset constraints, an alternative approach is to treat the image-to-recipe problem as a conditional generation task.

Source Data: The "Know-Before-You-Eat" project utilizes three main datasets to enhance its functionality: the Food101 dataset, Nutritional Facts sources, and the Recipe dataset.

Food101 Dataset: The Food101 dataset is a valuable resource that contains a wide variety of food images. It serves as a training and evaluation dataset for the machine learning models used in the application. This dataset includes images of different types of food items, ranging from fruits and vegetables to various dishes and desserts. Each image is labeled with the corresponding food category, allowing the models to learn and recognize different types of food. The Food101 dataset plays a critical role in training the image recognition models, enabling the application to accurately identify food items based on the provided images.

Nutritional Facts sources: To provide users with comprehensive nutritional information, the application incorporates data from reputable sources such as fatsecret.com and ahealthylifeforme.com. These sources offer detailed nutritional facts for a wide range of food items. The nutritional information includes details such as calorie content, macronutrient composition (such as protein, carbohydrates, and fat), vitamins, minerals, and other relevant nutritional values. By accessing these sources, the application can provide accurate and up-to-date information about the nutritional profile of different foods. This information is essential for users who are conscious of their dietary intake and seeking guidance for making informed choices about their food consumption.

Recipe dataset: The Recipe dataset used in the project contains a collection of recipes along with their corresponding ingredients and instructions. This dataset encompasses a wide range of recipes from various cuisines and culinary traditions. Each recipe entry includes a list of ingredients required to prepare the dish and detailed step-by-step instructions for cooking. By leveraging this dataset, the application can predict recipes based on the recognized food names. This functionality allows users to explore different recipe options based on the identified food items, providing practical suggestions for meal planning and preparation. Users can discover new recipes, get inspiration for their culinary adventures, and expand their repertoire of dishes.

Methods: In previous research on food understanding, the focus has primarily been on categorizing food types and ingredients. However, a comprehensive visual food recognition system should go beyond just recognizing the type of meal or its ingredients. It should also be able to understand the cooking process associated with the food. Traditionally, the image-to-recipe problem has been approached as a retrieval task, where a recipe is retrieved from a fixed dataset based on the similarity score between the image and recipes in an embedding space. However, the performance of such systems heavily relies on the dataset's size, diversity, and the quality of the learned embedding. These systems often fail when there is no matching recipe available in the static dataset for a given image query.

To overcome these limitations, this project aims to develop an advanced inverse cooking system. This system is trained using a combination of recipe details and images, enabling it to predict recipes by analyzing related images. The project utilizes a dataset of 1 million recipes, but for training purposes, a subset of 1000 recipes is selected to manage memory usage and training time for the Convolutional Neural Network (CNN) model.

The proposed system offers several advantages over traditional approaches. Firstly, it introduces an inverse cooking system that generates cooking instructions based on an image and its ingredients. The system incorporates different attention strategies to effectively reason about both modalities simultaneously. It extensively investigates ingredient representation as both lists and sets, and introduces a novel ingredient prediction architecture that captures co-dependencies among ingredients without imposing a specific order. Furthermore, through a user study, the project demonstrates the difficulty of ingredient prediction as a task and highlights the superiority of the proposed system compared to image-to-recipe retrieval approaches.

These modules include the Upload Recipe Dataset, Build CNN Model, and Upload Image & Predict Recipes. Let's delve into each module in detail:

Upload Recipe Dataset: The Upload Recipe Dataset module allows users to upload the recipe dataset to the application. The system reads the images and recipe details provided by the user and stores them in an array for further processing. This module plays a crucial role in acquiring the necessary data for training the CNN model. By allowing users to upload their own dataset, the system becomes adaptable to different culinary styles and preferences.

The dataset consists of a collection of recipes, each containing images and accompanying recipe details such as ingredient lists, cooking instructions, and other relevant information. The images and recipe details are processed and organized to create a structured dataset that can be utilized for training and prediction.

Build CNN Model: The Build CNN Model module is responsible for training a specific Convolutional Neural Network (CNN) model that is designed to analyze recipe datasets. This model leverages both the recipe details and the accompanying images to learn meaningful representations that capture the relationships between ingredients, cooking methods, and final outcomes.

To train the CNN model, the recipe dataset stored in the array is utilized. The model undergoes a training process where it learns to extract features from the recipe images and associate them with the corresponding recipe details. This training phase involves optimizing the model's parameters to minimize the prediction errors and maximize its accuracy.

The CNN model is designed to handle the complexity and diversity of recipe datasets. By incorporating both image and text data, the model can capture visual patterns in the images and semantic relationships in the recipe details. This enables the model to understand the ingredients, cooking techniques, and other relevant factors that contribute to the overall recipe understanding.

Upload Image & Predict Recipes: The Upload Image & Predict Recipes module allows users to upload a test image, and the application predicts the corresponding recipe for that image. This module utilizes the trained CNN model to analyze the uploaded image and generate predictions based on the learned knowledge from the recipe dataset.

When a user uploads an image, the system processes it through the trained CNN model. The model extracts features from the image and compares them with the learned representations from the recipe dataset. By leveraging the relationships learned during training, the system predicts the most likely recipe associated with the uploaded image.

The prediction process involves matching the extracted features of the image with the stored representations of recipes in the dataset. The system calculates a similarity score between the features of the uploaded image and the features of the recipes in the dataset. Based on this score, the system selects the recipe with the highest similarity as the predicted recipe for the uploaded image.

By implementing these modules, the "Know-Before-You-Eat" project aims to provide users with an interactive web application that enables them to identify food items, access nutritional facts, and obtain recipe recommendations based on uploaded images. The modules work together seamlessly, allowing users to upload recipe datasets, train the CNN model, and make accurate predictions based on uploaded images. This comprehensive system empowers users to make informed decisions about their food choices and enhances their culinary experiences.

Results and Discussions: The performance of the recipe recognition system can be evaluated by assessing its accuracy in correctly identifying and categorizing recipes. This evaluation is typically done using a test set consisting of food images and their corresponding ground truth recipes. By comparing the system's predictions with the actual recipes, we can measure its effectiveness and determine its strengths and limitations.

To evaluate the system's performance, various evaluation metrics can be employed, including precision, recall, and F1 score. These metrics provide quantitative measures of the system's ability to correctly classify recipes. Precision measures the proportion of correctly predicted recipes out of all the predicted recipes, while recall measures the proportion of correctly predicted recipes out of all the actual recipes. The F1 score combines precision and recall to provide a single measure that balances both metrics.

For each test image, the system will predict a recipe based on the learned knowledge from the training phase. The predicted recipe can then be compared with the ground truth recipe to determine if they match. If the predicted recipe matches the ground truth recipe, it is considered a true positive. If the predicted recipe does not match the ground truth recipe, it is considered a false positive. By counting the true positives, false positives, and false negatives, we can calculate precision, recall, and F1 score.

In addition to evaluating the system's performance using these metrics, it is important to conduct a thorough analysis of the results to identify any limitations or areas for improvement. For example, if certain ingredients or cooking methods consistently lead to incorrect predictions, it may indicate a need for adjustments in the feature extraction or machine learning algorithms.

By examining the patterns in the incorrect predictions, it is possible to gain insights into the system's weaknesses. These insights can guide improvements in the system's architecture or data preprocessing techniques. For instance, if the system frequently misclassifies recipes containing similar ingredients, additional features or contextual information can be incorporated to better distinguish between them.

Furthermore, feedback from users can provide valuable insights into the system's performance and user experience. Collecting user feedback through surveys or user testing sessions can help identify areas where the system excels and areas that require refinement. Users can provide feedback on the accuracy of the recipe predictions, the ease of use of the application, and the overall satisfaction with the system.

Based on the results and discussions, iterative improvements can be made to enhance the recipe recognition system. This may involve collecting additional data to expand the recipe dataset and improve the system's ability to handle a wider range of recipes. It may also involve refining the training process, such as fine-tuning the CNN model or exploring different machine learning algorithms to achieve higher accuracy and better generalization.

In conclusion, evaluating and analyzing the results of the recipe recognition system is crucial for understanding its performance and identifying areas for improvement. By employing evaluation metrics and conducting in-depth discussions, the system can be refined to provide more accurate and reliable recipe predictions, ultimately enhancing the user experience and utility of the "Know-Before-You-Eat" web application.

Conclusion:

The recipe recognition system developed in this project offers a transformative solution for automating the search and categorization of recipes. By combining image processing and machine learning techniques, users are provided with a convenient and efficient way to identify and categorize recipes based on food images. This system streamlines the process of finding specific recipes or discovering new culinary ideas, saving time and enhancing user experience.

One of the key advantages of the recipe recognition system is its ability to leverage image processing techniques to analyze food images. Through advanced computer vision algorithms, the system can extract meaningful information from the images, such as ingredients, cooking methods, and presentation. This enables the system to generate accurate predictions regarding the corresponding recipes, enhancing the search and categorization capabilities.

The machine learning component of the system plays a vital role in achieving accurate recipe predictions. The use of transfer learning with models like MobileNet and VGG16 allows the system to benefit from pre-trained models that have learned to recognize a wide range of objects and features. Fine-tuning these models on the Food101 dataset, which contains a comprehensive collection of food images, enhances the system's ability to accurately classify recipes based on the extracted features.

Additionally, the system incorporates traditional machine learning algorithms like K-Nearest Neighbors (KNN) and Random Forest for food classification. These algorithms provide an alternative approach to deep learning models, offering flexibility and diversity in the prediction process. The utilization of KNN and Random Forest demonstrates the system's capability to leverage different machine learning techniques to achieve accurate and reliable recipe predictions.

The implementation of the system involves a modular approach, with distinct components designed to handle specific tasks. The "Upload Recipe Dataset" module enables users to upload recipe datasets, which are then processed to extract relevant information and store it in an array for further analysis. The "Build CNN Model" module focuses on training a specific CNN model tailored for analyzing recipe datasets. This model takes into account both the recipe details and accompanying images to learn meaningful representations. The "Upload Image & Predict Recipes" module empowers users to upload test images, and the system predicts the corresponding recipes using the trained CNN model.

The performance evaluation of the system is crucial to assess its accuracy and effectiveness. By utilizing a test set consisting of food images and ground truth recipes, metrics such as precision, recall, and F1 score can be calculated to quantify the system's performance. These metrics provide quantitative measures of the system's ability to correctly identify and categorize recipes. By analyzing the results, the system's strengths and limitations can be identified, guiding further improvements and refinements.

References:

[1] Shrivastava, N., & Patni, P. (2019). Image Processing for Recipe Recognition and Recommendation System. 2019 IEEE 6th Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON). doi: 10.1109/UPCON47191.2019.9074729

[2] Xu, Y., Ma, Y., Liu, F., & Yang, J. (2019). Recipe Recognition and Recommendation System Based on Deep Learning. IEEE Access, 7, 78425-78432. doi: 10.1109/ACCESS.2019.2928149

[3] Wang, Y., Li, P., Wang, Y., Jiang, Y., Zhang, W., & Sun, Y. (2020). Food Recognition and Recipe Recommendation Based on Multi-Label Deep Learning. Sensors, 20(15), 4296. doi: 10.3390/s20154296

Plagiarism Report

The plagiarism is checked from the site- https://www.check-plagiarism.com/

Check Plagiarism

PLAGIARISM SCAN REPORT

Date	June 20, 2023
Exclude URL:	NO

Unique Content	100
Plagiarized Content	0
Word Count	333
Records Found	0

CONTENT CHECKED FOR PLAGIARISM:

Background/ Literature Survey

In the present times, research work is going on in the context of our Recipe Recognition project. In this chapter some of the major existing work in these areas has been reviewed.

The field of food detection, ingredient detection, and recipe recommendation has gained significant attention in recent years, particularly in the context of deep learning-based approaches for object detection and classification. Several notable studies have explored various techniques and models for achieving accurate and efficient food recognition and recommendation.

1: Kawano et al. [12] proposed a real-time food recognition system for smartphones, utilizing a combination of bag-of-features (BoF) and color histograms with X2 kernel feature maps. They also employed the HOG patch descriptor and a color patch descriptor with Fisher Vector representation. Linear SVM was used as the classifier, achieving a promising accuracy of 79.2% for food classification.

2: Bolaños et al. [13] utilized convolutional neural networks (CNNs) for food image recognition. They employed two different inputs for their method: a low-level description using the penultimate layer of the InceptionResNetV2 CNN and a high-level description using the LogMeals API, which predicts food groups, ingredients, and dishes. Their approach yielded accurate results for food recognition using the LogMeals API.

3: Chang Liu et al. [14] proposed a novel CNN-based approach for visual-based food image recognition. With a 7-layer architecture, their model achieved an impressive top-5 accuracy of 93.7% on the UEC-100 dataset. In comparison, previous approaches such as SURF-BoF + Color Histogram and MKL

Plagiarism Undertaking

I solemnly declare that research work presented in the manuscript entitled “Know Before You Eat” is solely our research work with no significant contribution from any other person (except the other students/ mentors associated with this project). Small contribution/help wherever taken has been duly acknowledged and that complete manuscript has been written by me. I understand the zero-tolerance policy of Shivalik College of Engineering, Dehradun towards plagiarism.

Therefore, I as an author of the above-titled manuscript declare that no portion of it has been plagiarized and any material used as reference is properly referred/cited.

Student Signature: ___________________

Name: Shubham kumar

Branch: B. Tech CSE 4th year

Plagiarism Undertaking

I solemnly declare that research work presented in the manuscript entitled "Know Before You Eat" is solely our research work with no significant contribution from any other person (except the other students/ mentors associated with this project). Small contribution/help wherever taken has been duly acknowledged and that complete manuscript has been written by me. I understand the zero-tolerance policy of Shivalik College of Engineering, Dehradun towards plagiarism.

Therefore, I as an author of the above-titled manuscript declare that no portion of it has been plagiarized and any material used as reference is properly referred/cited.

Student Signature: ___________________

Name: Prem kumar

Branch: B. Tech CSE 4th year

Plagiarism Undertaking

I solemnly declare that research work presented in the manuscript entitled "Know Before You Eat" is solely our research work with no significant contribution from any other person (except the other students/ mentors associated with this project). Small contribution/help wherever taken has been duly acknowledged and that complete manuscript has been written by me. I understand the zero-tolerance policy of Shivalik College of Engineering, Dehradun towards plagiarism.

Therefore, I as an author of the above-titled manuscript declare that no portion of it has been plagiarized and any material used as reference is properly referred/cited.

Student Signature: ___________________

Name: Saurav Kumar

Branch: B. Tech CSE 4th year