AI Lab: Final Project

RECIPE RECOMMENDER SYSTEM

1. Introduction

Motivation

Advancements in artificial intelligence and computer vision are transforming various sectors, including culinary practices. Efficient kitchen management is increasingly important in a fast-paced world, highlighting the need for systems that simplify meal preparation. Current recipe recommendation systems often require manual ingredient input, which is both error-prone and inconvenient.

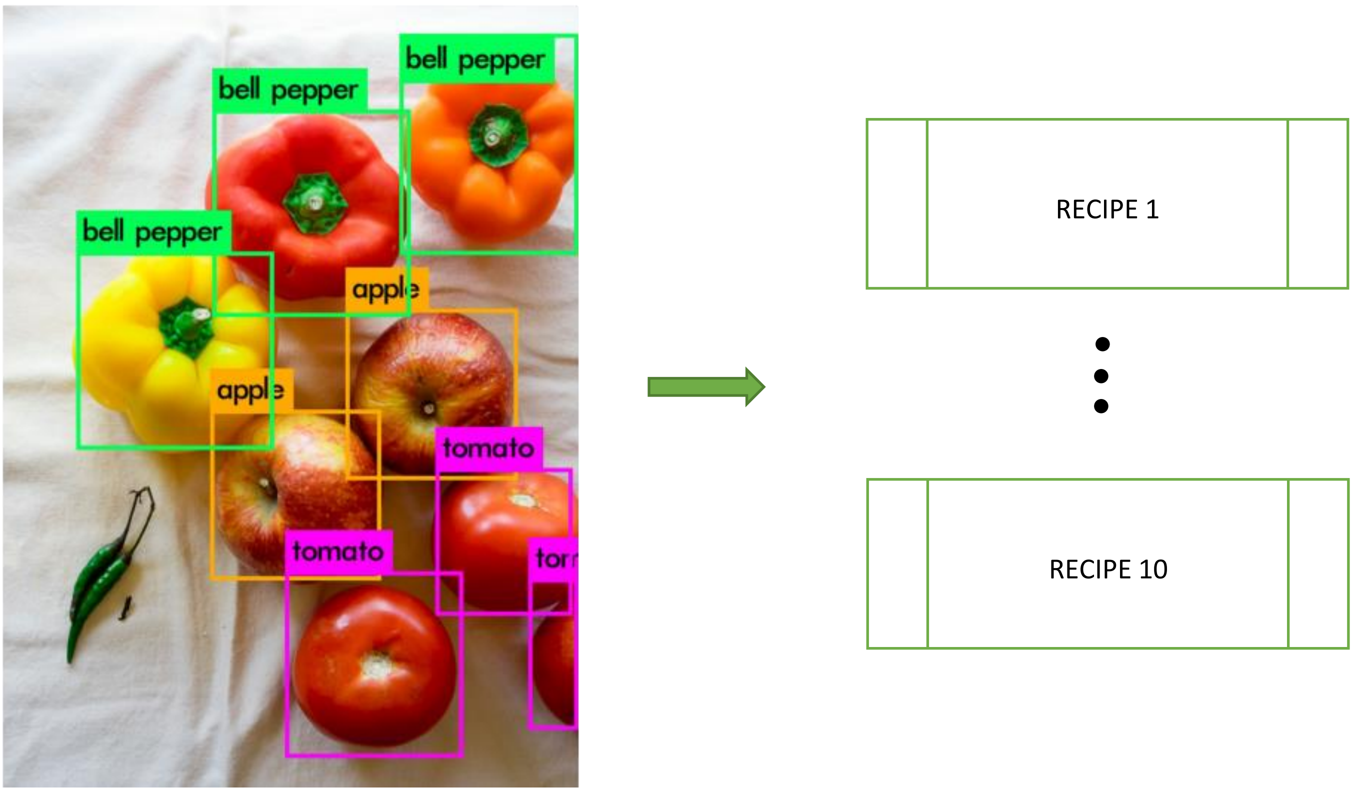
Problem Statement

Manual entry of ingredients into recipe systems is inefficient and can lead to inaccurate recommendations. Users frequently lack a precise inventory of their kitchen supplies, leading to suboptimal recipe suggestions. Thus, an automated solution that accurately detects ingredients in real-time and suggests appropriate recipes is needed to enhance user experience and reduce food waste.

This project aims to develop a recipe recommender system that utilizes real-time object detection to identify ingredients in a user's kitchen. The key objectives are to:

- Implement a camera-based system to recognize ingredients across different kitchen locations.
- Generate recipe suggestions based on detected ingredients to maximize resource use.
- Provide an intuitive interface for recipe discovery and meal planning.

By integrating real-time object detection with recipe recommendations, this system aims to ease the meal preparation and contribute to more sustainable cooking practices.



2. Methods

Data Description and Preprocessing

Groceries Images: The dataset for this project comprises images of fruits and vegetables, categorized into 36 distinct classes. Each class contains approximately 100 images, resulting in a diverse collection of approximately 3,600 images. To standardize the input for model training, all images were resized to a uniform resolution of 128x128 pixels. This resizing process ensures consistency and enhances the model's ability to learn features effectively from the data.

Recipe Database: The recipe dataset includes approximately 2.3 million recipes. This extensive collection serves as the foundation for generating recipe recommendations based on detected ingredients. The recipes were preprocessed to facilitate efficient retrieval and integration with the recommendation system.

Model Architecture

A Convolutional Neural Network was developed for the task of ingredient detection. The CNN architecture includes the following components:

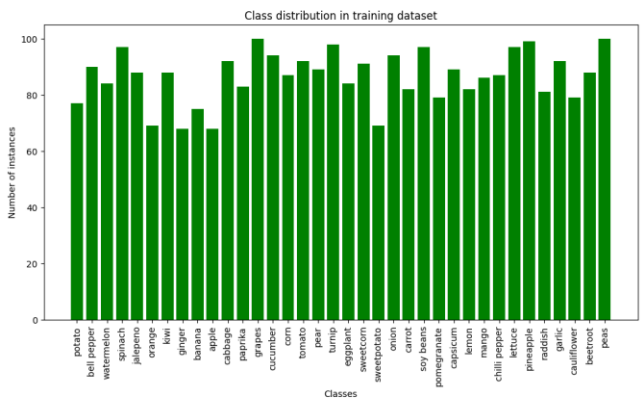
> Convolutional Layers: The network utilizes multiple convolutional layers to extract hierarchical features from the input images. These layers are followed by ReLU activation functions to introduce non-linearity.
>
> Pooling Layer: Max pooling is used to reduce the spatial dimensions of the feature maps, retaining essential features while reducing computational complexity.
>
> Fully Connected Layers: After feature extraction, the network uses fully connected layers to perform classification, mapping the extracted features to one of the 36 ingredient classes.

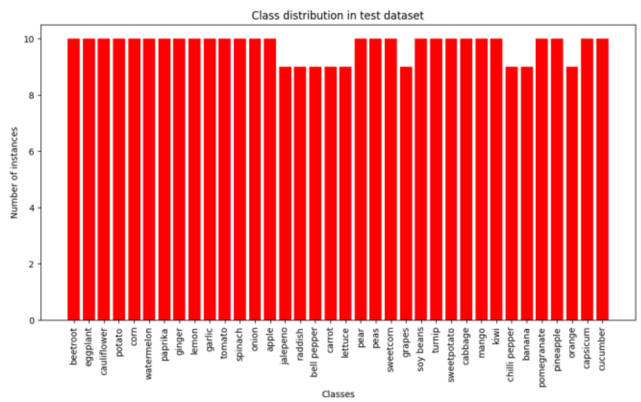
Training and Testing

The CNN model was trained for 15 epochs using the training dataset. The training process involved optimizing the model parameters to minimize classification error. The model's performance was evaluated periodically using the validation set to ensure that it generalized well to unseen data.

For testing, the trained model was applied to a video file instead of a live camera feed. This approach allowed for controlled conditions and easy reproducibility of results. The model processed each frame of the video to detect and classify ingredients, providing predictions that were subsequently analyzed.

A bar plot was created to visualize the distribution of different ingredient classes in the training and test datasets. Plots highlight the balance of class representation and helps identify any potential class imbalances. Below charts show Training and Test data distribution.





Recipe Handling and Integration

The recipe data was loaded from a CSV file into a DataFrame for processing. Essential preprocessing steps included cleaning the ingredient names extracted from the recipe data. Specifically, the NER (Named Entity Recognition) column, which contains ingredient lists, was stripped of extraneous characters and formatted for consistency. This involves removing brackets and quotation marks and replacing commas with spaces, making the data suitable for vectorization.

To facilitate efficient recipe retrieval based on detected ingredients, the TF-IDF vectorization technique was applied to the cleaned ingredient lists. This transforms the ingredient text data into numerical vectors that reflect the importance of each ingredient in the context of all recipes. A Nearest Neighbors model, using cosine similarity as the metric, was trained on these vectors to enable rapid searching of recipes based on ingredient matches.

When a set of ingredients is detected, these ingredients are vectorized using the same TF-IDF vectorizer. The resulting vectors are then compared against the precomputed recipe vectors to find the most similar recipes. The Nearest Neighbors algorithm retrieves the indices of the closest recipes based on cosine similarity. For each matched recipe, the title, ingredient list, and preparation directions are extracted and displayed. The system outputs top recommendations, providing users with a list of recipes they can prepare with their available ingredients.

The recommended recipes are printed along with their ingredients and cooking directions. This output is formatted for readability, and a limited number of top recipes (up to ten) are displayed to ensure relevance and manageability.

some results for input ingredient of `["flour","potato","pasta","fish","salt","oil"]`

```
Frieten  (Belgian French Fries)

ingredients: "oil", "potato", "salt"

"Cut the potatoes in the form of French fries and rinse them well with water.
make them dry with a clean dish cloth.", "Deep fry them 5 minutes at 150 degrees
Celsius,", "take them out of the oil and leave them to cool down.", "Heat the
oil to 180 degrees Celsius and deep fry the fries until they are crispy on the
outside and golden brown (about 5 minutes).", "Sprinkle with salt and serve with
mayonnaise."
==============================================================

Fresh Taglierini

ingredients: "flour", "pasta"

"Attach ribbon-pasta cutters to pasta machine and attach handle to thinnest
cutter.", "Line a tray with a dry kitchen towel.", "Feed first rolled out pasta
sheet, which will have dried out slightly but will still be soft, through cutter
and toss generously with flour.", "Form pasta loosely into nest and arrange on
kitchen-towel-lined tray.", "Make more taglierini with remaining dough in same
manner.", "Taglierini may be made 1 day ahead and chilled on towel-lined tray,
covered loosely with plastic wrap."
==============================================================

Jalapeno Butter Grilled Fish

ingredients: "fresh fish", "butter"

"Make a pan with heavy-duty tinfoil to fit on outdoor grill or inside grill.",
"Melt butter in pan over fire and add jalapeno juice. Add fish fillets and cook
until opaque."
```

3. Results

Validation Accuracy: The accuracy of the model on the validation dataset was plotted over epochs to illustrate the model's performance improvement during training. This plot provides a visual representation of how well the model learned to classify ingredients over time.

Validation accuracy: 0.9544

	precision	recall	f1-score	support
apple	1.00	0.70	0.82	10
banana	1.00	0.89	0.94	9
beetroot	1.00	1.00	1.00	10
bell pepper	0.89	0.89	0.89	9
cabbage	1.00	1.00	1.00	10
capsicum	0.90	0.90	0.90	10
carrot	0.89	0.89	0.89	9
cauliflower	1.00	1.00	1.00	10
chilli pepper	1.00	0.89	0.94	9
corn	0.80	0.80	0.80	10
cucumber	1.00	1.00	1.00	10
eggplant	0.91	1.00	0.95	10
garlic	1.00	1.00	1.00	10
ginger	1.00	0.90	0.95	10
grapes	1.00	1.00	1.00	9
jalepeno	1.00	1.00	1.00	9
kiwi	1.00	1.00	1.00	10
lemon	1.00	1.00	1.00	10
lettuce	1.00	1.00	1.00	9
mango	1.00	1.00	1.00	10
onion	0.83	1.00	0.91	10
orange	0.90	1.00	0.95	9
paprika	0.91	1.00	0.95	10
pear	1.00	1.00	1.00	10
peas	1.00	1.00	1.00	10
pineapple	1.00	1.00	1.00	10
pomegranate	1.00	1.00	1.00	10
potato	1.00	0.80	0.89	10
raddish	0.82	1.00	0.90	9
soy beans	1.00	1.00	1.00	10
spinach	1.00	1.00	1.00	10
sweetcorn	0.82	0.90	0.86	10
sweetpotato	1.00	0.80	0.89	10
tomato	0.83	1.00	0.91	10
turnip	1.00	1.00	1.00	10
watermelon	1.00	1.00	1.00	10
accuracy			0.95	351
macro avg	0.96	0.95	0.95	351
weighted avg	0.96	0.95	0.95	351

The CNN model achieved an impressive validation accuracy of approximately 96%. This high accuracy demonstrates the model's effectiveness in classifying images of fruits and vegetables, indicating that it can reliably recognize a wide range of ingredients when evaluated on the validation dataset. During training, the model consistently improved its performance, with accuracy metrics indicating robust learning. The final training accuracy showed the model's capability to generalize well on unseen data from the test set. Detailed metrics and accuracy values were logged and assessed throughout the training and testing phases.

Despite the high validation accuracy, real-time detection of ingredients presented challenges. The model faced difficulties due to varying conditions such as lighting, background clutter, and different angles of ingredients. This led to discrepancies between the expected and actual detection performance. Sample outputs from the real-time detection system were reviewed, highlighting cases where the model successfully identified ingredients as well as instances where the detection was inaccurate. This discrepancy underscores the need for further refinement in handling diverse real-world scenarios.

The TF-IDF vectorization approach effectively matched detected ingredients with recipes from the database. The system successfully identified relevant recipes based on the available ingredients, providing users with practical cooking suggestions. Analysis revealed issues with recipe data quality. Incomplete ingredient lists and inconsistencies between ingredient descriptions and meal preparation details were noted. Despite the effective matching of ingredients, these data quality problems impacted the overall reliability of the recommendations.