Kathmandu University

Department of Computer Science and Engineering

Dhulikhel, Kavre

A Project Report
on
"Foodify"

[Code No: COMP 207]

(For partial fulfillment of Year II/Semester II in Computer Science/Engineering)

Submitted by

Sujan Ghimire (15)
Bigyan Kumar Piya (41)
Sewan Uprety (60)
Aagaman Bhandari (07)

Submitted to

Mr. Nabin Ghimire

Department of Computer Science and Engineering

January 1, 2024

Bona fide Certificate

This project work on

“FOODIFY”

is the bona fide work of

“SUJAN GHIMIRE”

“BIGYAN KUMAR PIYA”

“SEWAN UPRETY”

“AAGAMAN BHANDARI”

who carried out the project work under my supervision.

Project Supervisor

_____

Mr. Nabin Ghimire
Assistant Professor
Department of Computer Science and Engineering

Date: January 1, 2024

Abstract

The research-based approach project where we created a program that gave us the nutritional values of the food using food's image as a source. Our prime focus and intention were to learn the machine learning aspect in a bit more depth and embark the journey of deep learning. We took image-based data from various sources and used CNN based image classification model to get the best result with peak accuracy as much as possible. We used Streamlit and Flask web framework in order to create our website, thereby letting the users drag and drop their desired input image for extracting the nutritional information about it.

Keywords: *Food Image classification, CNN, MobileNetV2*

Table of Contents

List of Figures

Acronyms/Abbreviations

CNN	Convolutional Neural Network
DNN	Deep Neural Network
API	Application Program Interface
TF	TensorFlow
RTX	Ray Tracing Texel
AIML	Artificial Intelligence Markup language
GCN	Graph Convolutional Network
RNN	Recurrent Neural Network
Cal	Calorie
IDE	Integrated Development Environment

Chapter 1 Introduction

1.1 Background

In recent years, the software of deep learning in the realm of food popularity and dietary analysis has garnered vast attention. Foodify, a challenge leveraging deep learning algorithms, has emerged as a revolutionary approach to address the challenge of automatically recognizing meals gadgets and imparting complete dietary information. the arrival of convolutional neural networks (CNNs) and other deep studying architectures has revolutionized photograph recognition tasks, making them greater correct and green, thus laying the groundwork for advancements in meals reputation generation.

Foodify is a web-based program created with SteamLit and flask web framework. We have included 36 different vegetables and fruits. When a user uploads image of a vegetable or fruit (unknown), the program identifies the image and reports its calories. In the realm of web development, the choice of Streamlit and Flask as the frameworks reflected a commitment to creating an accessible and interactive interface. Streamlit, known for its simplicity and ease of use, enabled users to effortlessly interact with the system. Flask, a lightweight web framework, facilitated the integration of the machine learning model into the web application, ensuring a seamless experience for users. For Foodify, we used CNN for food recognition. CNNs are particularly useful for finding patterns in images to recognize objects, classes and categories. In Foodify, we are using mixed dataset combining dataset from various sources. There are various similar projects and research papers related to Foodify which are discussed in Chapter 2.

1.2 Objectives

Foodify aims to meet following objectives:

- Enable users to effortlessly upload food images, receiving instant and accurate nutritional information.
- To be aware and uplift quality of life.
- Food image classification with utmost accuracy.

1.3 Motivation and Significance

Nowadays, people are confined by the hectivity of their own work. As an outcome of which, they forget to treat themselves with proper and nutritious diet, which is a must. Observing this, we got the idea of creating the web based app “Foodify” which helps people to perceive what kind of food they are consuming which eventually leads to them balancing their diet in their daily life. We also thought of track of food being eaten by consumer, which means people can reminisce themselves the kind of food they are eating on regular basis.

Chapter 2 Related Works

2.1 Foodvisor

Foodvisor, an innovative nutritional application co-founded by Aurore Tran, Charles Boes, Gabriel Samain, and Yann Giret in 2015, employs advanced deep learning algorithms to conduct a sophisticated analysis of user-uploaded meal images. Operating on the principles of convolutional neural networks (CNNs) and recurrent neural networks (RNNs), the application performs intricate image recognition, overcoming challenges associated with varying camera perspectives, illumination conditions, and food presentation styles.

The algorithmic core of Foodvisor extends its capabilities beyond mere food identification to encompass the intricate task of quantity estimation. Tackling the complexities of this endeavor involves addressing issues related to object localization and image segmentation, demonstrating the application's prowess in computer vision.

At the crux of its functionality, Foodvisor conducts an exhaustive nutritional analysis utilizing mathematical models such as multivariate regression and neural network-based regression. This enables the application to not only ascertain caloric content but also precisely quantify the distribution of macronutrients—namely lipids, proteins, carbohydrates, and dietary fibers. Additionally, the algorithm delves into the realm of micronutrients, encompassing variables like cholesterol, sodium, potassium, magnesium, various vitamins, calcium, and iron.

Figure 3.1.1.1 Foodvisor Interface

User personalization is achieved through the incorporation of user-centric details, encompassing demographic factors (gender, age), anthropometric parameters (weight), lifestyle considerations, and individual dietary objectives. The application dynamically tailors its recommendations for daily nutritional intake, applying principles of reinforcement learning to adapt to evolving user preferences and goals.

Behind the computational curtain, Foodvisor's machine learning models undergo continual refinement via training on vast datasets enriched with meticulously annotated food images.

2.2 Computer Vision based Food Recognition with Nutrition Analysis

The authors of this paper are Asst. Prof Apoorva Busad, G S Suchitha, Darshan Gowda TS, Lakshmi S and Shalini K Gowda. It was published on 1st January, 2023. The paper discusses a number of methods for mobile visual food recognition, including the application of deep learning models like convolutional neural networks (CNNs) and graph convolutional networks (GCNs). In order to increase the precision and robustness of a food recognition system, the authors of the paper suggest a specific method that combines visual features of food images with semantic data. Additionally, the document explains the idea of machine learning, which entails training a model on a dataset to enable it to learn relationships and patterns in the data and form judgments or predictions based on fresh data. Machine learning comes in both supervised and unsupervised forms. The use of AIML, a programming language that enables creators to create chatbots that can have conversations with humans in natural language, is also briefly mentioned in the document. The document also discusses how difficult it is for consumers to understand how they perceive food products and how crucial it is to understand and evaluate consumers in order to predict a product's success.

2.3 The Food Recognition Benchmark: Using Deep Learning to Recognize Food in Images

The authors of this paper are Sharada Prasanna Mohanty, Gaurav Singhal, Eric Antoine Scuccimarra, Djilani Kebaili, Harris Héritier, Victor Boulanger and Marcel Salathé. It was published on 6th May, 2022. The document talks about the MyFoodRepo app, which is utilized in a number of medical cohort studies, as well as the significance of highly accurate data annotation. Instance segment annotations are redrawn, and class assignments are corrected, as part of the human assessment. The

document emphasizes how important it is for quality assurance to include a human correction and verification step. The benchmark depends on users of the MyFoodRepo app to report their daily dietary intake, and the Food Recognition API processes the images the app collects to produce instance segmentation and classification annotations for the images. The approach used to address the issue of accurate food image recognition is also discussed in the document. This approach is based on the idea that accurate food image recognition is possible but requires iterative advancements over time. The strategy is further based on the idea that a crowd-sourced approach with appropriately aligned incentives can efficiently leverage machine learning expertise around the world, providing a significantly wider intellectual focus on the problem than the traditional "single group" research approach.

Chapter 3 Design and Implementation

3.1 Data Preprocessing

3.1.1 Loading and Resizing Images:

The training data was sourced from a directory structure, with each class represented by a subdirectory. Images were loaded using OpenCV, and the script checked for image validity. Invalid images were excluded from the dataset.

3.1.2 Image Resizing and Labeling:

Images were resized to a uniform dimension of 224x224 pixels using OpenCV's resize function. This ensures consistency in input size for the convolutional neural network. Labels were assigned based on the class names, and the index of each class in the class_names array was used as the label.

3.1.3 Data Validation:

A check for the resized image's dimensions was implemented to ensure all images conform to the desired size (224x224). This step is crucial for maintaining the integrity of the dataset and compatibility with the model architecture.

3.1.4 Data Normalization:

Data normalization was performed by scaling pixel values to the range [0, 1]. This step is crucial to ensure consistent training across different images.

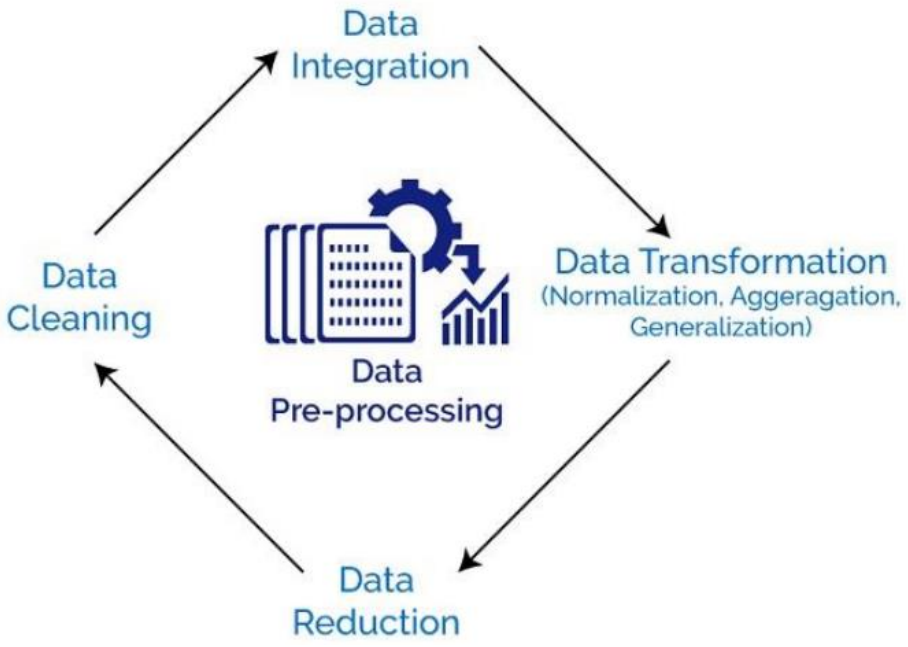

Figure 3.1.4.1 Data Preprocessing

This data preprocessing step involved loading, resizing, and labeling images for the training dataset. The resulting arrays (file_names, train_data_array, and train_data_labels_array) were then ready for further processing and model training.

3.2 Model Architecture:

3.2.1 MobileNetV2 Selection and Modification:

The MobileNetV2 architecture was chosen due to its efficiency and effectiveness in image classification tasks. The decision to freeze the pre-trained layers ensures that the model retains the general features learned from ImageNet while allowing customization for the specific food classification task.

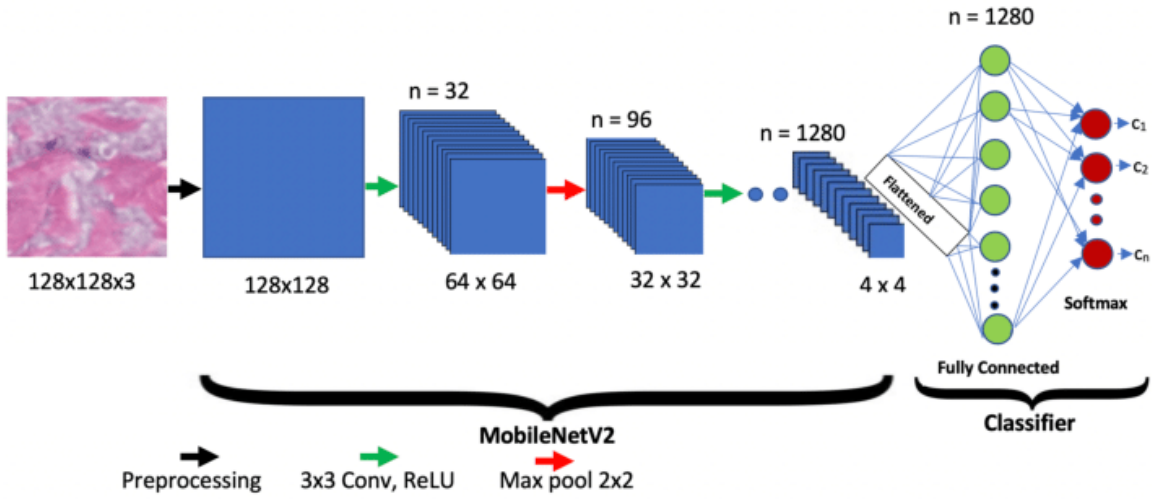

Figure 3.2.1.1 MobilenetV2 architecture

3.2.2 Custom Dense Layers:

Two custom dense layers were added on top of the MobileNetV2 base. The Global Average Pooling 2D layer was employed to reduce spatial dimensions and extract essential features. The subsequent dense layer with ReLU activation helps capture complex patterns. The final dense layer uses softmax activation to output probabilities for the 36 food classes.

Python code:

```
x = GlobalAveragePooling2D()(base_model.output)

x = Dense(128, activation='relu')(x)

predictions = Dense(36, activation='softmax')(x)

model = keras.Model(inputs=base_model.input, outputs=predictions)
```

3.3 Model Training:

The model was compiled using the Adam optimizer, which adapts learning rates during training. Categorical cross-entropy loss was chosen as it suits multi-class classification tasks. Accuracy was selected as the metric to monitor during training.

Python code:

```
model.compile(
    optimizer='adam',
    loss='categorical_crossentropy',
    metrics=['accuracy']
)
```

3.4 Web Application (Streamlit and Flask):

3.4.1 Streamlit Frontend:

Streamlit was employed for creating a user-friendly frontend. The application allows users to upload food images, which are then processed by the trained model. Streamlit components, such as file upload widgets and interactive elements, were utilized for a seamless user experience.

3.4.2 Flask Backend Integration:

Flask was used as the backend framework to handle requests from the Streamlit frontend. The Flask app serves as a bridge between the user interface and the machine learning model. It receives image uploads, processes them through the model, and returns the classification results. These integrated web application provides a practical interface for users to classify food images and obtain nutritional information.

This technical overview emphasizes the architectural decisions, model configuration, and the integration of Streamlit and Flask in your project. You can expand on each section with more detailed explanations and code snippets based on the specifics of your implementation.

Figure 3.4.2.1 ML deployment

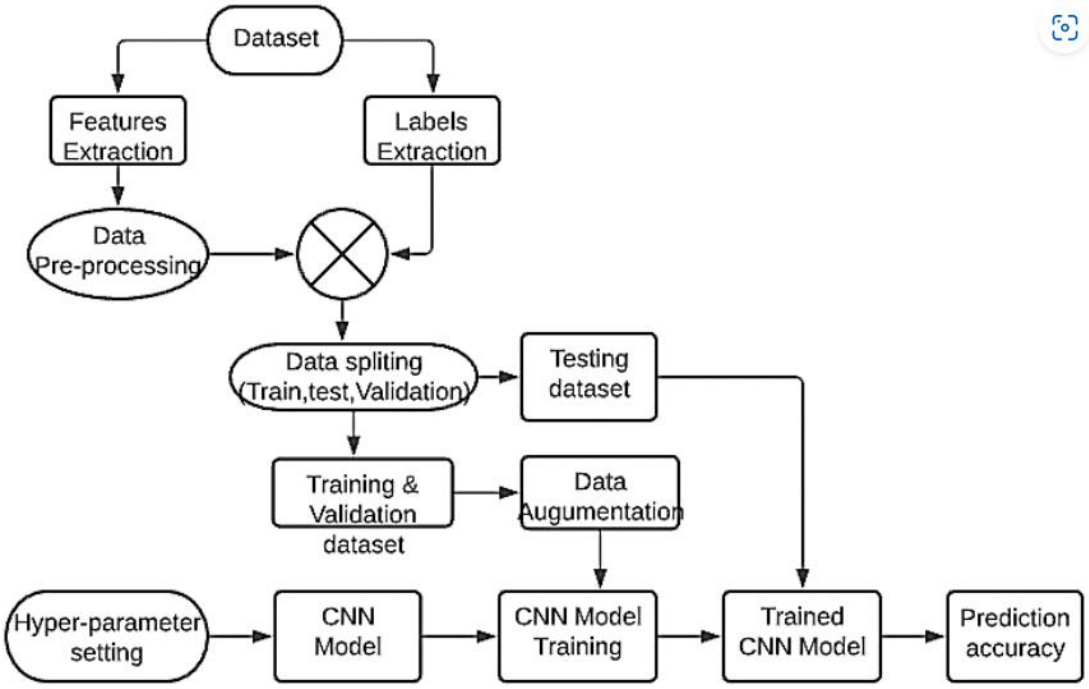

Figure 3.4.2.2 Block diagram of multi-class food image classifier using CNN

3.5 System Requirement Specifications

3.5.1 Software Requirements

• Python 3.8 in Jupyter Lab is used for data pre-processing, model training and prediction.

• Operating System: windows 7 and above or Linux based OS or MAC OS.

• IDE: Visual Studio code with android studio.

3.5.2 Functional requirements

In developing the software for Foodify, some of the functional requirements are:

• The software accepts the picture of food dataset as input.

• The software does pre-processing (like verifying for missing data) on input for model training.

• The software uses CNN models to classify image as main component of the software.

• The software needs ask user for permission to capture image and track user data.

3.5.3 Non-Functional requirements

• Usability: It defines the user interface of the software in terms of simplicity of understanding the user interface of foodify software.

• Efficiency: maintaining the possible highest accuracy in the classification of food type in shortest time with available data.

• Performance: It is a quality attribute of the food classification software that describes the responsiveness to various user interactions with it.

3.5.4 Hardware Specifications

Minimum hardware requirements are: CPU with 8 cores, 32 GB RAM, 1 TB hard drive, and an NVIDIA GeForce RTX 1080 (or 2080) Series 8GB GPU

Chapter 4 Discussion on the achievements

Our main objective was to detect the food item and extract the nutritional information about it. Since there were a lot of neural network models available, we trained our dataset on few of those. After training our dataset on various neural network models, MobileNetV2 emerged as the most efficient in accurately identifying food items, demonstrating minimal loss. The model's precision, recall, f1-score, and support for each food item are summarized below.

	precision	recall	f1-score	support
apple	1.00	0.80	0.89	10
banana	1.00	0.78	0.88	9
beetroot	1.00	1.00	1.00	10
bell pepper	1.00	0.90	0.95	10
cabbage	1.00	1.00	1.00	10
capsicum	0.91	1.00	0.95	10
carrot	1.00	0.90	0.95	10
cauliflower	1.00	1.00	1.00	10
chilli pepper	1.00	1.00	1.00	10
corn	0.82	0.90	0.86	10
cucumber	1.00	1.00	1.00	10
eggplant	1.00	1.00	1.00	10
garlic	1.00	1.00	1.00	10
ginger	1.00	1.00	1.00	10
grapes	1.00	1.00	1.00	10
jalepeno	1.00	1.00	1.00	10
kiwi	1.00	1.00	1.00	10
lemon	1.00	1.00	1.00	10
lettuce	1.00	1.00	1.00	10
mango	1.00	1.00	1.00	10
onion	1.00	1.00	1.00	10
orange	0.91	1.00	0.95	10
paprika	0.91	1.00	0.95	10

pear	0.91	1.00	0.95	10
peas	0.91	1.00	0.95	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
radish	0.83	1.00	0.91	10
soy beans	1.00	1.00	1.00	10
spinach	0.91	1.00	0.95	10
sweetcorn	0.89	0.80	0.84	10
sweetpotato	1.00	0.90	0.95	10
tomato	0.91	1.00	0.95	10
turnip	1.00	1.00	1.00	10
watermelon	1.00	1.00	1.00	10

Figure 3.5.4.1 Performance evaluation of the model MobileNetV2

Likewise, we evaluated the confusion matrix heatmap to give us a clear insight on how the model worked while identifying the food items. It let us know which food item was being mistaken for another one.

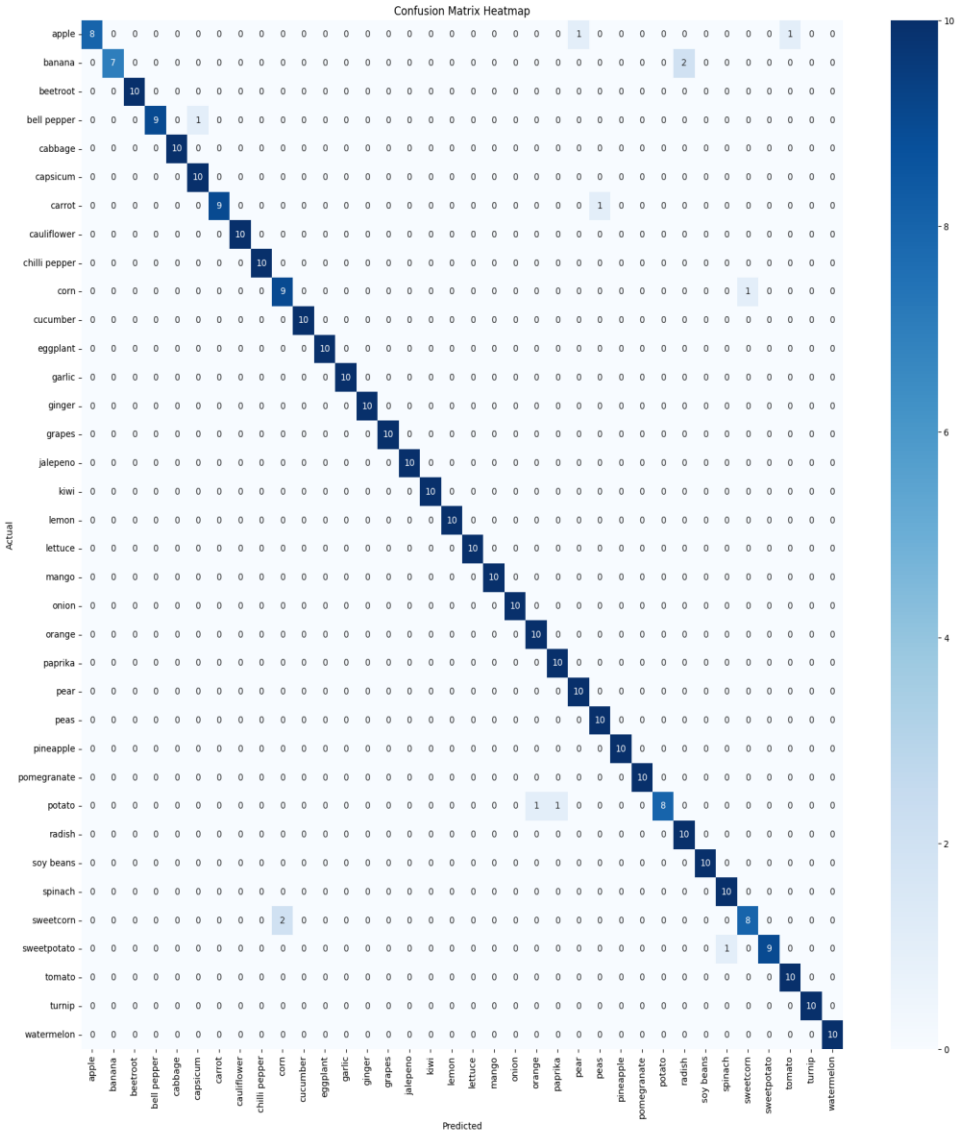

Figure 3.5.4.2 Confusion matrix heatmap of the model

Ultimately, we successfully deployed our Convolutional Neural Network (CNN) model, fine-tuned with MobileNetV2 architecture, onto our website. This implementation was achieved seamlessly through the integration of Flask and

Streamlit web frameworks. Leveraging the robust capabilities of Flask for web application development and Streamlit for interactive data visualization, our website now serves as a user-friendly platform for food item detection and nutritional information extraction. Users can conveniently upload images, and the deployed CNN model swiftly processes and identifies the food items, providing accurate and timely nutritional insights. This integration not only enhances the accessibility of our model but also offers a seamless and engaging user experience.

Chapter 5 Conclusion and Recommendation

In conclusion, while our nutrition app represents a significant stride in leveraging machine learning and deep learning techniques to derive nutritional information from food images, our project's culmination has unveiled critical limitations that must be addressed for further advancement. The focus on vegetables and fruits, while initially beneficial, limits the app's applicability across diverse dietary preferences and meal compositions. The dataset's restricted variety of images poses challenges in achieving robust and generalized image recognition, potentially leading to overfitting or underfitting issues in our model. Moreover, the app's current capacity to detect only one food item at a time restricts its practicality in real-world scenarios where meals typically encompass multiple components. Moving forward, our recommendations entail expanding the dataset to encompass a broader range of food items, enhancing the image recognition model to handle multiple items simultaneously, and incorporating a more diverse set of dietary options. This iterative process will be pivotal in improving the app's accuracy, versatility, and overall effectiveness in providing comprehensive nutritional assessments for users with varying dietary habits and preferences. As our project concludes, it sets the stage for future developments that prioritize inclusivity and precision in food recognition for enhanced user experience and healthier lifestyle choices.

5.1 Limitations

Despite the promising prospects of our nutrition app, certain limitations exist due to resource constraints and the nature of our dataset. The focus on specific food items, namely vegetables and fruits, has inherent drawbacks as it limits the app's versatility in recognizing a broader range of foods. The dataset's limited variety of images may result in challenges related to accuracy, potentially leading to overfitting or underfitting issues in our model. Additionally, the app's current capability to detect only one food item at a time restricts its practicality in real-world scenarios where enhancements should prioritize expanding the dataset to encompass a more diverse array of food items and refining the image recognition model to handle multiple items simultaneously. This approach will contribute to the app's overall effectiveness and ensure a more comprehensive and reliable nutritional assessment for users with varied dietary preferences and habits.meals often comprise multiple components. Recognizing these limitations, future

5.2 Future Enhancement

- User Profile and Preferences:

  The app could start by allowing users to create a profile with basic information such as age, gender, weight, height, and dietary preferences (e.g., vegetarian, vegan, keto). Users can set personal health and fitness goals, such as weight loss, muscle gain, or maintenance.

- Food Journal and Nutrient Tracking:

  Users can log their daily food intake by either manually entering food items or utilizing the image recognition feature to identify and log foods. The app should

provide a comprehensive breakdown of macronutrients (carbohydrates, proteins, and fats), micronutrients, and calories consumed based on the logged food items.

- Meal Planning and Recommendations:

The app could offer personalized meal plans based on the user's dietary goals and preferences. Users can receive suggestions for well-balanced meals that meet their nutritional requirements.

- Real-time Feedback and Alerts:

Users could receive real-time feedback on their nutritional choices, helping them make healthier decisions throughout the day. The app might send alerts or notifications to remind users to stay hydrated, consume enough fiber, or balance their macronutrient intake.

- Progress Tracking and Analytics:

Implementing visual representations of users' progress, such as charts and graphs, can motivate them to stay on track with their nutritional goals. Analytics could include trends over time, allowing users to identify patterns in their eating habits.

- Continuous Model Improvement:

Regularly updating and enhancing the image recognition model to recognize a broader range of foods and improve accuracy. Periodically incorporating user feedback to refine and optimize the app's features.

- Privacy and Security:

Ensuring robust privacy and security measures to protect users' sensitive health and dietary information.

References

Mohanty, P, S,. , Singhal, G,. , Scuccimarra, A, E,. , Kebaili, D,. , Héritier, H,. , Boulanger, V,. and Salathé, M(2022, May 06). *The Food Recognition Benchmark: Using Deep Learning to Recognize Food in Images.* Retrieved December 31, 2023, from https://www.frontiersin.org/articles/10.3389/fnut.2022.875143/full?fbclid=IwAR3xAwDg0L2bky3UygOp-y2Tp3bocP1dzsvDjk-uRMbY71b5izs59g3vlbs

Busad, A,. , Suchitha, S, G,. , TS, G, D,. , S, L,. , Gowda, K, S(2023, January 01). *Computer Vision based Food Recognition with Nutrition Analysis.* Retrieved December 31, 2023, from https://www.ijcrt.org/papers/IJCRT2301042.pdf

Wikipedia. (n, d). Foodvisor. Retrieved December 31, 2023, from https://fr.wikipedia.org/wiki/Foodvisor

Shen, Z,. , Shehzad, A,. , Chen, S,. and Sun, H,. , Liu, J(2020, June 11). *Machine Learning Based Approach on Food Recognition and Nutrition Estimation.* Retrieved December 31, 2023, from https://www.sciencedirect.com/science/article/pii/S1877050920316331

Jiang, M(2019). *Food Image Classification with Convolutional Neural Networks.* Retrieved December 31, 2023, from https://cs230.stanford.edu/projects_fall_2019/reports/26233496.pdf

Attokaren, J, D,. , Fernandes, G, I,. , Sriram, A,. , Koolagudi, G, S,. and Murthy, S, Y(2017, November 05). *Food Classification from Images UsingConvolutional Neural Networks.* Retrieved December 31, 2023, from https://www.researchgate.net/publication/322216381_Food_classification_from_images_using_convolutional_neural_networks

Foodvisor(n, d). Foodvisor App. Retrieved December 31, 2023, from https://www.foodvisor.io/en/Foodvisor

Singh, H,. , Singh, R,. and Goel, P(2023, June). *Block diagram of multi-class vegetable image classifier using CNN*. Retrieved December 31, 2023, from https://www.researchgate.net/figure/Block-diagram-of-multi-class-vegetable-image-classifier-using-

CNN_fig3_361990016?fbclid=IwAR2h_na2T1VWEampWkcnpDZcCI1hqhv3ivV07K1JUyzHyiSOcTDMwAmekLc

Seth, K(2022, January). *Fruits and Vegetables Image Recognition Dataset.* Retrieved December 31, 2023, from https://www.kaggle.com/datasets/kritikseth/fruit-and-vegetable-image-recognition?fbclid=IwAR3vPCg9eiHoIgV-54UBNeDCOB8L8Wr0Pbom_z-Sg8XCnTF7VJoJyRXwLKM

Streamlit(n.d.).Create an app. Streamlit Documentation. Retrieved December 31, 2023, from https://docs.streamlit.io/get-started/tutorials/create-an-app?fbclid=IwAR1FNBV4DTV2iS0dkSnI9G4fBtjiZsmcJO-A6nnKT1_WUM3ucXXO6Sv50vM

TensorFlow(n.d.). MobileNetV2. TensorFlow API Documentation. Retrieved December 31, 2023, from https://www.tensorflow.org/api_docs/python/tf/keras/applications/mobilenet_v2/MobileNetV2?fbclid=IwAR3_0u8ZJeJwOwtNqLbhgs76YHq1O6Wjy1ORVb9HrmeWeC--fObRIkjJsWc

Maillard, L. (n.d.). diet-vision-flutter. GitHub. Retrieve December 31, 2023, from

https://github.com/leopoldmaillard/diet-vision-flutter?fbclid=IwAR3ganolBr87BwpRxEVyDDr3EVtPVqmyT8WCtpsA-oHr7PIg7R3diTmKLyY

Google(n.d.). Search results for "calories." Google Search. Retrieved December 31, 2023, from https://www.google.com/search?&q=calories&fbclid=IwAR3_0u8ZJeJwOwtNqLbhgs76YHq1O6Wjy1ORVb9HrmeWeC--fObRIkjJsWc

APPENDIX

Figure 3.5.4.1 Snippet 1

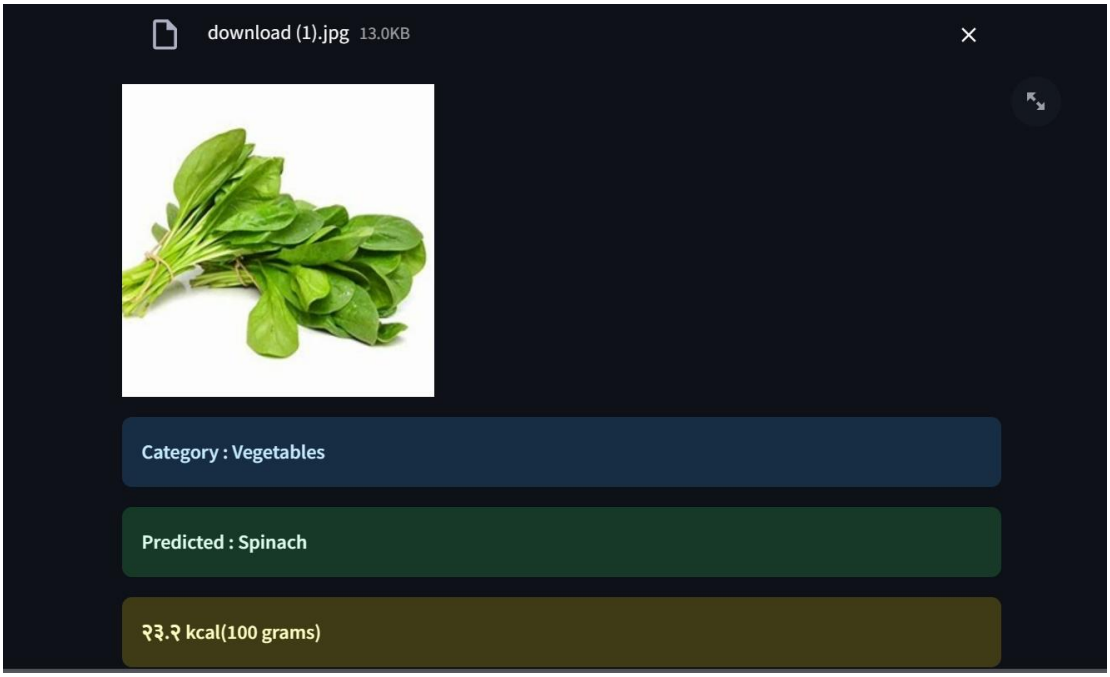

Figure 3.5.4.2 Snippet 2