FOOD RECOGNITION AND CALORIE ESTIMATION USING MACHINE LEARNING

|  |  |
| --- | --- |
| **AD3811 PROJECT** | **REPORT** |
| ***Submitted By***  **AKHIL KRISHNA R** | **721921243011** |
| **ATHIRA E** | **721921243027** |
| **AYSHA ZIYANA K** | **721921243029** |
| **MUHAMMED HARIS P** | **721921243064** |

***in partial fulfilment of the award of the degree***

***of***

# BACHELOR OF TECHNOLOGY

*in*

# ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

### DHANALAKSHMI SRINIVASAN COLLEGE OF ENGINEERING COIMBATORE - 641105

**(AUTONOMOUS)**

1

# DHANALAKSHMI SRINIVASAN COLLEGE OF ENGINEERING COIMBATORE - 641105

**(AUTONOMOUS)**

### BONAFIDE CERTIFICATE

This is to certify that the project report **"FOOD REGNITION AND CALORIE ESTIMATION USING MACHINE LEARNING" ”** is the bonafide work of the following students

|  |  |
| --- | --- |
| **AKHIL KRISHNA R** | **721921243011** |
| **ATHIRA E** | **721921243027** |
| **AYSHA ZIYANA K** | **721921243029** |
| **MOHAMMED HARIS P** | **721921243064** |

who carried out the project work under my supervision during the academic year 2024- 2025.

### SIGNATURE

#### Dr. P CHANDRASEKAR, M.E, PhD HEAD OF THE DEPARTMENT

Dept of Artificial Intelligence and Data Science ,

Dhanalakshmi Srinivasan College ofEngineering , Coimbatore - 641105

### SIGNATURE

#### Mrs. Nadhiya S, M.E SUPERVISOR

Dept of Artificial Intelligence and Data Science ,

Dhanalakshmi Srinivasan College ofEngineering , Coimbatore - 641105

Submitted for the project viva-voce held on

##### INTERNAL EXAMINER EXTERNAL EXAMINER

2

### ACKNOWLEDGEMENT

First and foremost, I would like to thank almighty for showering the blessings throughout our life. I take the privilege to express heartly thanks to my parents for their valuable support and effort to complete this project.

I take this chance to express my deep sense of gratitude to our Management, our **Principal Dr. C JEGADHEESAN, M.E, PhD** and our **Dean Dr. K BAGHIRATHI** for providing an excellent infrastructure and support to pursue work at our college.

I express my profound thanks to our beloved Head of the Department

**Dr. P CHANDRASEKAR M.E, PhD** for his able admission and keen interest, which motivated us to complete the project.

I extend my thanks to **Mrs. NADHIYA S, M.E, AP/AI&DS** for her valuable guidance at each and every stage of the project . which helped me a lot in the successful completion of the project.

I take immense pleasure to express my heartfelt thanks to **Dr. P CHANDRASEKHAR, M.E, PhD** for his valuable suggestions and constant supports project coordinator to complete this project report.

I am very much grateful to all my teaching and non-teaching staffs and my friends who helped me to complete the project.

3

### ABSTRACT

With the growing global emphasis on health, fitness, and nutritional awareness, the ability to monitor food intake has become increasingly important. Manual tracking of food consumption is often time-consuming and inaccurate, leading to the need for automated tools that can assist users in managing their diets effectively. This project, titled **"Food Recognition and Calorie Estimation Using Machine Learning,"** presents a computer vision-based solution that can recognize various food items from images and estimate their caloric value in real time.

The system uses deep learning techniques, particularly **Convolutional Neural Networks (CNNs),** to perform food classification. Each class is associated with an average calorie value sourced from standard nutritional databases. Once an image is input by the user, the model identifies the food item and retrieves the corresponding calorie information, thus providing an estimate of the caloric content.

The backend of the project is developed using **Python**, employing libraries such as TensorFlow/Keras or PyTorch for model development, and OpenCV for image processing. The frontend is designed to allow users to upload images easily and view the classification results and calorie information. The system architecture is modular, supporting potential future expansion to include portion size estimation and more nuanced nutritional analysis.

This project demonstrates the power of machine learning in automating everyday health- related tasks. It can be integrated into mobile apps, diet management tools, and health monitoring systems. Potential real-world applications include personalized diet tracking, support for diabetic or fitness-focused individuals, and integration into smart kitchens or restaurant recommendation engines.

4

### INDEX

**SL.NO CHAPTER PG.NO**

##### PROJECT DESCRIPTION

5

* 1. INTRODUCTION

7

* 1. AIM & OBJECTIVE

9

* 1. EXISTING SYSTEM
  2. HARDWARE REQUIREMENTS
  3. SOFTWARE REQUIREMENTS

##### SYSTEM DESIGN

* 1. [ARCHITECTURE 11](#_TOC_250011)
  2. [SCOPE OF PROJECT 12](#_TOC_250010)

1. **WORKING OF SYSTEM**
   1. FEATURES 14
   2. [INPUT DATA AND VALIDATION 15](#_TOC_250009)
2. **EXPERIMENTAL ANALYSIS**
   1. CASE MODEL 17
   2. [PROPOSED SYSTEM 19](#_TOC_250008)

5

##### SOFTWARE DESCRIPTION

* 1. [PYTHON OVERVIEW 21](#_TOC_250007)
  2. [INTRODUCTION TO HTML](#_TOC_250006)
  3. [INTRODUCTION TO CSS](#_TOC_250005)
  4. [INTRODUCTION TO JAVA SCRIPT](#_TOC_250004)

1. CONCLUSION AND FUTURE WORK
   1. [CONCLUSION 23](#_TOC_250003)
   2. [FUTURE RECOMMENDATION](#_TOC_250002)
2. REFERENCE
   1. [REFERENCES 25](#_TOC_250001)
3. APPENDIX
   1. [SOURCE CODE 25](#_TOC_250000)
   2. OUTPUT

6

# CHAPTER 1

### INTRODUCTION

In the modern world, where lifestyle-related diseases such as obesity, diabetes, and cardiovascular disorders are on the rise, maintaining a healthy diet has become more important than ever. Calorie tracking is a crucial aspect of dietary management, yet traditional methods—such as manually logging meals or estimating portion sizes—are often inaccurate, time-consuming, and difficult to sustain over time. With the rapid advancements in Artificial Intelligence (AI), particularly in the fields of Machine Learning (ML) and Computer Vision, there is a growing opportunity to automate and enhance the process of food analysis and calorie estimation.

This project presents a machine learning-based approach for automatic food recognition and calorie estimation from images. The primary objective is to develop a system that can identify different food items from a user-provided photo, estimate their portion sizes, and calculate the corresponding caloric and nutritional values. The system leverages deep learning models, specifically Convolutional Neural Networks (CNNs), for accurate food classification and integrates image processing techniques to assess portion sizes.

By linking recognized food items with established nutritional databases, the system can provide users with real-time, reliable dietary information. The long-term vision is to integrate this solution into mobile applications and wearable devices, making it a seamless part of a user’s daily routine. Beyond personal use, this technology holds potential in various domains including healthcare, fitness, diet counseling, and public health monitoring.

Through this project, we aim to reduce the effort involved in calorie tracking, improve the accuracy of dietary assessments, and promote healthier eating habits by harnessing the power of machine learning

.

7

###### State-of-the-Art AI Application:

Uses deep learning for accurate calorie estimation in food and beverages. Helps individuals track their dietary intake and maintain a healthy lifestyle. **Significance of the Model:**

Provides accurate calorie estimations for various food items. Assists users in monitoring and managing their daily calorie intake. **Integration of Technologies:**

* **a. Image Recognition**: Identifies food items from images.
* **b. Nutritional Database**: Maps identified food to calorie values.
* **c. Deep Learning Model**: Estimates nutritional content using MobileNet CNN.

###### Obesity and Health Risks:

* Defined as BMI ≥ 30 kg/m², linked to various health conditions.
* Caused by an imbalance between food consumption and energy expenditure.
* Poor dietary habits contribute to increased obesity rates.

###### Need for Calorie Estimation:

* Effective obesity management requires nutritious food choices.
* Regular monitoring of calorie intake helps in maintaining a balanced diet.

Encourages a healthier lifestyle by promoting awareness of food consumption

8

### AIM & OBJECTIVES

**AIM**

The aim of this project is to develop a deep learning-driven food recognition and calorie estimation system that enables accurate, real-time dietary monitoring to promote healthy eating habits. By utilizing Python programming and the MobileNet architecture, the system seeks to automate food identification and calorie estimation from images, providing users with precise nutritional insights.

* This project aspires to empower users with intelligent diet monitoring features, allowing them to track their daily food consumption, make informed dietary choices, and achieve personalized nutrition goals. By integrating deep learning algorithms with a web-based framework, the system ensures fast and accurate food recognition, helping individuals effectively manage their dietary intake and adopt a healthier lifestyle.

### OJECTIVES

The objective of this project is to develop a machine learning-based system that can automatically identify various food items from images and estimate their corresponding calorie content. This includes collecting and preprocessing a labeled dataset of diverse food categories, training a reliable classification model, and mapping each recognized food item to its average nutritional value. The goal is to provide users with an easy and accurate way to track their food intake, promoting healthier eating habits. Additionally, the system aims to be user-friendly and efficient enough for practical use in health, fitness, and dietary management applications.

9

### EXISTING SYSTEMA

Existing systems typically allow users to upload food images, classify them using machine learning models, and retrieve calorie information from nutrition databases.

Most systems are trained on a limited set of food categories, often excluding regional or culturally specific dishes, leading to poor recognition accuracy for diverse cuisines.

Calorie estimation is generally based on standard portion sizes, which may not reflect the actual amount consumed, resulting in inaccurate results.

These systems often rely on constant internet connectivity, as image processing and calorie data retrieval are performed via cloud services.

Accuracy drops significantly when food images are taken in poor lighting, from unusual angles, or contain multiple food items.

Some systems still require manual user input to select or confirm the food item, limiting the level of automation.

Personalization is minimal, as most tools do not adapt calorie estimates based on user- specific dietary needs or health goals.

High computational requirements make it difficult to deploy advanced models on mobile or low-resource devices.

Privacy concerns arise when users are required to upload personal food images to remote servers without guaranteed data protection.

Many systems lack portion size detection or volume estimation, which is essential for calculating accurate calorie intake beyond just identifying the food item.

Ingredient-level analysis is often missing, meaning composite or mixed dishes like salads, curries, or layered desserts are poorly handled or misclassified.

Most existing tools do not support multi-food segmentation, making it difficult to analyze meals with multiple items on a single plate.

10

Inconsistent food presentation styles (e.g., sliced vs. whole fruit, packed vs. served meals) can confuse the model and reduce prediction accuracy.

Systems generally don’t account for cooking methods (fried, baked, boiled), which significantly affect calorie content even within the same food type.

Some models perform well in lab settings but fail in real-world deployment, where users take casual, imperfect photos.

 Lack of explainability in AI predictions makes it difficult for users to trust or understand why a specific food or calorie count was chosen.

### DISADVANTAGES:

Limited Food Classes

Lack of Portion Size Estimation

Dependence on Internet Connectivity

Manual Input Still Required

Low Accuracy in Real-world Conditions

High Computational Requirements

11

### HARDWARE REQUIREMENTS

|  |  |
| --- | --- |
| **HARWARE ELEMENTS** | **SPECIFICATION/ VERSION** |
| RAM | 6 GB |
| Processor | 15 & Above |
| Hard disk space | 2GB (min)free space  available |
| Screen Resolution | : 1024 x 768 or higher. |

* 1. **SOFTWARE REQUIREMENTS**

**SOFTWARE ELEMENTS**

**SPECIFICATION/**

**VERSION**

OS

Windows 7 or later

Simulation Tool

Python 3.10 ,

HTML , CSS,JS

Web Framework

Flask

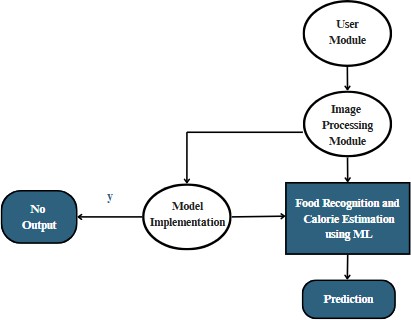
Documentation

MS-Office

12

# CHAPTER 2

### SYSTEM ARCHITECTURE

****

13

#### Image Acquisition Module

This module allows users to input images of food items either by capturing them through a camera or uploading them from their device. It serves as the entry point to the system and ensures that the image is in an acceptable format and quality for further processing. Proper image acquisition is crucial for achieving accurate recognition results.

#### Image Preprocessing Module

Once the image is acquired, it undergoes preprocessing to improve the quality and consistency of the input. This includes resizing, normalization, noise reduction, and possibly background removal. These steps standardize the images before they are passed to the classification model, helping to enhance accuracy and reduce prediction errors.

#### Food Recognition Module (Classification Model)

This is the core module where the actual identification of the food item takes place. A trained machine learning or deep learning model, typically a convolutional neural network (CNN), processes the preprocessed image and classifies it into one of the predefined food categories. The model is trained using labeled datasets to ensure high accuracy.

14

###### Calorie Estimation Module

After the food item is recognized, this module estimates its calorie content. It uses a predefined lookup table or database that maps each food class to its average calorie value. Optionally, the module can be enhanced to factor in portion size for more accurate estimation, especially in future versions of the system

###### Dataset Management Module

This module handles the storage, organization, and labeling of images used for training and testing the model. It includes functionality for data augmentation to increase dataset diversity and supports easy updates when new food items are introduced. Proper dataset management is critical for maintaining model performance.

###### Model Training & Evaluation Module

This module is responsible for training the food recognition model using the collected dataset. It includes splitting the dataset into training and validation sets, selecting model architecture, and tuning hyperparameters. Evaluation metrics such as accuracy, precision, recall, and F1-score are used to measure and improve the model's performance.

###### User Interface Module (Optional)

If implemented, this module provides a user-friendly interface for interacting with the system. It allows users to upload images, view recognition results, and see estimated calories in a clear and accessible way. The UI can be a web-based dashboard or a mobile app depending on the deployment.

15

### SCOPE OF PROJECT

* The deep learning-based calorie estimation system aims to revolutionize dietary monitoring by providing an accurate and efficient method for estimating the caloric content of food and beverages using image recognition. The scope of this project includes the following key aspects:

#### Image-Based Calorie Estimation

* The system will use deep learning models to analyze images of food and beverages.
* MobileNet, a lightweight and efficient convolutional neural network, will be employed to ensure high accuracy with minimal computational resources.
* The model will classify different food items and estimate their calorie content based on pre- trained nutritional databases.

#### Nutritional Analysis and Database Integration

* The system will incorporate a well-structured nutritional database containing information on a wide variety of food items.
* The model will retrieve relevant calorie values by matching identified food items with their respective entries in the database.
* Users will receive detailed nutritional insights, including macronutrient breakdown (proteins, carbohydrates, fats) for informed dietary decisions.

#### User-Friendly Interface

* A web-based or mobile-friendly application will be developed to allow users to upload food images conveniently.
* The interface will provide real-time feedback on estimated calorie values and suggest healthier alternatives if necessary.
* Users will have the option to track their daily calorie intake through a personalized dashboard.

16

**CHAPTER 3**

### FEATURES OF THE PROJECT

Automatically recognizes food items from images using a trained machine learning or deep learning model.

Estimates calorie content by referencing a predefined nutritional database linked to each food class.

Supports a wide range of food categories, including local and culturally relevant dishes.

Provides a user-friendly interface for easy image uploading and result display.

Includes image preprocessing steps such as resizing and normalization to enhance model accuracy.

Designed with a modular structure, allowing easy updates and future expansion (e.g., portion size detection).

Can be adapted for offline use, making it accessible without constant internet connectivity.

Ensures user privacy by processing images locally or using secure methods. Offers optional result logging to track historical dietary intake.

Optimized for fast processing to support real-time or near-real-time use. Uses performance metrics (e.g., accuracy, precision) to evaluate and improve model reliability.

17

# INPUT DATA AND VALIDATION

###### Input Data:

The primary input is food images, either captured via camera or uploaded from a device. Each image corresponds to one of the 20 predefined food classes (e.g., apple, dosa, biriyani, vada, etc.).

The dataset used for training and testing includes labeled images of these food items — typically resized (e.g., 224×224) and normalized for consistent input to the model.

Associated metadata may include the class label (food name) and the average calorie value for each class, stored in a reference table or dictionary.

###### Validation Process:

The dataset is divided into training and testing sets, commonly using a split like 80% for training and 20% for testing.

Cross-validation (e.g., k-fold) may be applied during training to improve model generalization and avoid overfitting.

The trained model is validated on the test set using metrics like accuracy, precision, recall, and F1-score to assess food classification performance.

Confusion matrix analysis is used to understand misclassifications between similar food items.

The calorie estimation is validated by checking the correctness of the class-to-calorie mapping after classification.

18

# CHAPTER 4

### CASE MODEL OF THE PROJECT

###### Capture or Upload Food Image

Description: The user captures an image using a camera or uploads one from their device. Precondition: The image must be clear and show the food item prominently.

Postcondition: The image is ready for processing.

###### Recognize Food Item

Description: The system uses a trained ML model to classify the uploaded food image into one of the known categories.

Precondition: A valid image has been uploaded.

Postcondition: The food item is identified and its label is generated.

###### Estimate Calorie Value

Description: The system retrieves the average calorie count for the recognized food item from a calorie database.

Precondition: The food item has been successfully recognized. Postcondition: Calorie value is displayed to the user.

###### Display Result to User

Description: The system shows the recognized food name and estimated calories in a user- friendly format.

Precondition: Recognition and calorie mapping are complete. Postcondition: The user sees the final output on screen.

###### Log and Store Result (Optional)

Description: The system optionally stores the result (food name, calorie value, timestamp) for tracking.

Precondition: User chooses to store or logging is enabled. Postcondition: Data is saved in local or cloud storage for later use.

19

## PROPOSED SYSTEM

The proposed system is an intelligent and automated solution designed to recognize food items from images and estimate their calorie content using machine learning techniques. It eliminates the need for manual food logging by allowing users to simply upload or capture an image of their meal. The system preprocesses the image and then uses a trained convolutional neural network (CNN) to classify it into one of the predefined food categories. Once the food item is identified, the system retrieves its associated average calorie value from a structured nutrition reference table.

To ensure accuracy, the system incorporates image enhancement and normalization techniques during preprocessing, and it is trained on a carefully curated dataset representing various food classes, including region-specific dishes. The calorie estimation is based on standard values but can be expanded to include portion size detection in future improvements. The system is designed with modular components, allowing for easy updates, maintenance, and scaling. It can be deployed as a standalone desktop application, a web platform, or a mobile app depending on user requirements.

##### ADVANTAGES:

Automated Food Identification Quick Calorie Estimation

Supports Local and Cultural Foods Modular and Scalable Design

User-Friendly Interface Offline Capability

Real-Time Processing

20

# CHAPTER 5

### PYTHON OVERVIEW

Python is commonly used for developing websites and software, task automation, data analysis, and data visualization. Since it’s relatively easy to learn, Python has been adopted by many non-programmers, such as accountants and scientists, for a variety of everyday tasks, like organizing finances. "Writing programs is a very creative and rewarding activity," says University of Michigan and Coursers instructor Charles R Severance in his book Python for Everybody. “You can write programs for many reasons, ranging from making living to solving a difficult data analysis problem to having fun to helping someone else solve a problem." **Uses:**

Data analysis and machine learning Web development Automation or scripting Software testing and prototyping Everyday tasks 19

What is Python? Executive Summary Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed. Often, programmers fall in love with Python because of the increased productivity it provides. Since there is no compilation step, the edit- test-debug cycle is incredibly fast. Debugging Python programs is easy: a bug or bad input will never cause a segmentation fault. Instead, when the interpreter discovers an error, it raises an exception. When the program doesn't catch the exception, the interpreter prints a stack trace.

21

A source level debugger allows inspection of local and global variables, evaluation of arbitrary expressions, setting breakpoints, stepping through the code a line at a time, and so on. The debugger is written in Python itself, testifying to Python’s introspective power. On the other hand, often the quickest way to debug a program is to add a few print statements to the source: the fast edit-test-debug cycle makes this simple approach very effective. Python use cases are creating web applications on a server, building workflows that can be used in conjunction with software, connecting to database systems, reading and modifying files, performing complex mathematics, processing big data, fast prototyping, developing production-ready software professionally, Python is great for backend web development, data analysis, artificial intelligence and scientific computing. Developers also use python to build productivity tools, games and desktop apps. Features and benefits of python compatible with a variety of platforms including and others Windows, Mac, Linux, Raspberry Pi, uses a simple syntax comparable to the English language that lets developers use fewer lines than other programming languages, operates on an interpreter system that allows code to be executed immediately, fast- tracking prototyping can be handled in a procedural, object- orientated or functional way python syntax somewhat similar to the English language, with a mathematical influence, python is built for readability. Unlike other languages that use semicolons and/or parentheses to complete a command, python uses new lines for the same function defines scope (i.e., loops, functions, classes) by relying indentation, using whitespace, rather than braces python flexibility python, a dynamically typed language, is especially flexible, eliminating hard rules for building features and offering more problem-solving flexibility with a variety of methods. It also allows uses to compile and run programs right up to a problematic area because it uses run-time type checking rather than compile-time checking. The less great parts of python. On the down side, Python isn’t easy to maintain. One command can have multiple meanings depending on context because python is a dynamically typed language and maintaining a python app as it grows in size and complexity can be increasingly difficult, especially finding and fixing errors.

22

Users will need experience to design code or write unit tests that make maintenance easier. Speed is another weakness in python. Its flexibility, because it is dynamically typed, requires a significant amount of referencing to land on a correct definition, slowing performance. This can be mitigated by using alternative implementation of python. Python and AI researchers are fans of python. Google TensorFlow, as well as other libraries (scikit- learn, Keras), establish a foundation for AI development because of the usability and flexibility it offers Python users. These libraries, and their availability, are critical because they enable developers to focus on growth and building. The python package index is a repository of software for the Python programming language. It helps users find and install software developed and shared by the python community. The python package index hosts thousands of third-party modules for python. Both python's standard library and the community-contributed modules allow for endless possibilities. Python can be easy to pick up whether you're a first-time programmer or you're experienced with other languages. The following pages are a useful first step to get on your way writing programs with Python: Beginner's Guide, Programmers Beginner's Guide and Non-Programmers Beginner's Guide.

###### Python in Deep Learning: Overview

Python is one of the most widely used programming languages in the field of deep learning due to its simplicity, readability, and extensive support from libraries and frameworks that make implementing deep learning models easier. Deep learning, a subset of machine learning, involves using artificial neural networks (ANNs) to model complex patterns and structures in data, such as images, text, and audio. Python’s rich ecosystem of tools and libraries provides data scientists, researchers, and developers with the necessary tools to build, train, and deploy deep learning models efficiently.

23

###### Key Python Libraries for Deep Learning

1. **TensorFlow:** Developed by Google, TensorFlow is an open-source deep learning framework that supports both machine learning and deep learning algorithms. It is designed for flexibility and scalability, allowing researchers and engineers to build and deploy models across a variety of platforms. TensorFlow provides extensive functionality for building and training neural networks and is often used in tasks like image classification, object detection and natural language processing.
2. **Keras:** Keras is a high-level neural networks API written in Python, designed to run on top of TensorFlow. It simplifies the process of building deep learning models by providing an easy-to-use interface with intuitive syntax. Keras allows developers to quickly prototype and test models with fewer lines of code, making it popular in both research and industry.
3. **PyTorch:** PyTorch, developed by Facebook's AI Research lab, is an open-source deep learning framework that emphasizes dynamic computation and flexibility. Unlike TensorFlow, which uses a static computational graph, PyTorch allows for dynamic computation graphs, making it particularly well-suited for research and experiments where the model structure may change frequently.
4. **Theano:** Theano was one of the earliest deep learning libraries in Python and was widely used in research. Although its development has officially ceased, Theano laid the foundation for many other deep learning libraries like TensorFlow and PyTorch by providing the computational backend for efficient deep learning algorithms.
5. **Scikit-learn:** While Scikit-learn is primarily a machine learning library, it is often used in conjunction with deep learning libraries for preprocessing data, feature extraction, and implementing traditional machine learning models alongside deep learning models.
6. **NumPy and Pandas:** NumPy is essential for numerical computations, matrix manipulation, and handling multi-dimensional arrays, while Pandas provides data structures for working with large datasets, handling missing data, and performing complex data manipulation tasks, all of which are critical in the data preparation and analysis phase of deep learning.

24

### INTRODUCTION TO HTML

HTML, or HyperText Markup Language, is the foundational language used to create web pages. It is a markup language, which means that it uses tags or “elements” to define and structure the content of a webpage. HTML is not a programming language, as it does not contain logic or instructions that a machine can execute; rather, it describes the structure and organization of content on the page. Through its system of nested tags, HTML dictates how text, images, links, and other elements should appear on a webpage.

For example, the <h1> tag defines a large heading, <p> denotes a paragraph, and <a> is used to create a hyperlink. HTML tags are often used in pairs, with an opening tag and a closing tag (for example, <h1> and </h1>), though some tags, like <img>, are self-closing. It is important to understand the importance of the correct use of tags to structure content, as well as the role of attributes within those tags.

###### Structure of an HTML Document

An HTML document follows a very specific structure, with the foundational elements that ensure the page is correctly interpreted by browsers. At its core, the HTML structure typically looks something like this:

**<html>:** This is the root element of an HTML document. All other elements reside inside the <html> tag.

**<head>:** The <head> tag contains metadata and links to external resources like CSS files or JavaScript libraries. It is not visible to the user on the page.

 **<body>:** The <body> tag houses all the content that will be visible on the webpage, including text, images, links, and forms.

Within the <body>, HTML elements are used to define the layout of content. For example,

<div> and <section> tags are often used for grouping content, while other tags like <img>,

<video>, and <audio> are used to embed media content.

25

###### HTML Tags and Elements

HTML is built on a range of tags, each designed to perform specific functions. A typical webpage might use a variety of these tags:

**Text Elements:** Tags such as <h1> to <h6> for headings, <p> for paragraphs, and

<blockquote> for quotations are common. These tags are used to organize and display textual content.

**Lists:** Ordered (<ol>) and unordered lists (<ul>) are often used to display lists of items, with <li> being used for list items.

**Links and Images:** Hyperlinks are defined using the <a> tag, which allows navigation to other pages, while images are embedded with the <img> tag.

**Forms**: Forms are a key component of many websites, enabling user input. The <form> element, along with elements like <input>, <textarea>, and <button>, are used to create interactive forms.

 **Tables:** HTML tables are used to organize data into rows and columns using <table>,

<tr>, <td>, and <th> elements.

Each tag in HTML is important for organizing content and making it easily accessible to users and search engines. The right structure in HTML enhances readability, accessibility, and SEO (Search Engine Optimization), which plays a key role in how a webpage is indexed and ranked by search engines.

###### HTML5: Evolution and Features

HTML5 is the latest version of HTML, and it introduced several new features and capabilities. Unlike its predecessors, HTML5 comes with native support for multimedia, such as audio and video elements, which were previously achieved through third-party plugins like Flash.

26

**Audio and Video Elements:** HTML5 introduced the <audio> and <video> elements, allowing developers to embed multimedia content without relying on third-party plugins, making the experience smoother and more secure for users.

**Forms:** HTML5 also introduced new form controls, including <date>, <email>, <tel>,

<number>, and more, making it easier to capture and validate user input in modern web applications.

**Canvas and SVG:** HTML5 includes the <canvas> tag, which allows for dynamic rendering of graphics like drawings and animations directly in the browser. It also supports Scalable Vector Graphics (SVG), which are used for high-quality, resolution- independent images.

**Geolocation:** HTML5 introduced geolocation functionality, which allows websites to access users' geographical locations to personalize their content, such as showing nearby places or offering location-based services.

27

**Audio and Video Elements:** HTML5 introduced the <audio> and <video> elements, allowing developers to embed multimedia content without relying on third-party plugins, making the experience smoother and more secure for users.

**Forms:** HTML5 also introduced new form controls, including <date>, <email>, <tel>,

<number>, and more, making it easier to capture and validate user input in modern web applications.

**Canvas and SVG:** HTML5 includes the <canvas> tag, which allows for dynamic rendering of graphics like drawings and animations directly in the browser. It also supports Scalable Vector Graphics (SVG), which are used for high-quality, resolution- independent images.

**Geolocation:** HTML5 introduced geolocation functionality, which allows websites to access users' geographical locations to personalize their content, such as showing nearby places or offering location-based services.

28

### INTRODUCTION TO CSS

While HTML is responsible for the structure of a webpage, CSS (Cascading Style Sheets) controls its presentation and design. Without CSS, web pages would be plain, unstyled documents with no visual appeal. CSS allows web developers to specify the layout, colors, fonts, and overall aesthetics of a webpage. It enhances the visual experience by providing rules that govern how the HTML content should appear on the page.

CSS Syntax and Structure

CSS uses a simple syntax that consists of a selector, a property, and a value. A typical CSS rule looks like this:

selector { property: value;

}

For example:

h1 {

color: blue; font-size: 24px;

}

In this case, the selector is h1, which refers to all <h1> elements on the page, and the property is color, with the value blue. Similarly, the font-size property adjusts the text size.

###### How CSS Works ?

CSS works by linking a style sheet (either internal or external) to an HTML document. When the browser loads the HTML document, it fetches the associated CSS file and applies the styles defined in it. Developers can write CSS rules to target HTML elements by their tag name, class, id, or other attributes.

29

**·Inline CSS:** You can use the style attribute directly within an HTML tag to apply a style to that specific element. For example: <h1 style="color: blue;">Welcome!</h1>

**·Internal CSS**: Defined within the <style> tag in the <head> section of an HTML document, affecting the entire document.

 **·External CSS:** This is the most common and recommended approach, where an external

.css file is linked to the HTML document. This keeps the HTML clean and allows for reusability across multiple pages.

###### CSS Selectors and Specificity

CSS provides a range of selectors to target HTML elements with precision. Some of the most common types of selectors include:

* Type Selector: Targets elements by their tag name. Example: p { color: red; } targets all

<p> tags.

* Class Selector: Targets elements with a specific class. Example: .intro { font-size: 16px;

} targets all elements with the class intro.

* ID Selector: Targets a unique element with a specific ID. Example: #header { background-color: blue; } targets the element with the ID header.

###### Box Model in CSS

A fundamental concept in CSS is the box model, which describes how elements are structured on the page. Each element is treated as a box that consists of several layers:

1. **Content**: The actual content, such as text or images.
2. **Padding:** The space between the content and the border.
3. **Border:** The edge of the element.
4. **Margin:** The space between the element's border and adjacent elements.

30

### INTRODUCTION TO JAVASCRIPT

While HTML and CSS control the structure and design of a webpage, JavaScript is the language that adds interactivity and dynamic behavior to web pages. JavaScript is a high- level programming language that enables developers to build applications that respond to user input, update content dynamically, and interact with servers to fetch or send data without reloading the page. Unlike HTML and CSS, which are static, JavaScript enables the creation of interactive, engaging, and complex web applications.

###### How JavaScript Works

JavaScript works by being embedded within HTML pages or linked externally through a

<script> tag. When the page loads, the browser executes the JavaScript code, manipulating the Document Object Model (DOM), which represents the structure of the HTML page. Through JavaScript, developers can modify elements, styles, and content based on user interactions or other conditions.

JavaScript is event-driven, meaning that it can react to user actions such as clicks, key presses, or form submissions. This ability to respond to user input is what makes modern web applications dynamic and engaging.

###### JavaScript Fundamentals

**Variables and Data Types**: JavaScript uses variables to store data, which can be numbers, strings, arrays, or objects. JavaScript supports dynamic typing, meaning variables can hold any type of data.

**·Functions:** Functions allow developers to bundle reusable code into blocks that can be called when needed, improving code modularity and efficiency.

**·Loops and Conditionals:** JavaScript provides control structures like for, while, and if statements to create logic for repetitive tasks or conditional behaviors.

·**Objects and Arrays:** Objects and arrays allow developers to store and organize complex data, making it easy to manage and manipulate larger datasets.

31

###### DOM Manipulation

One of the primary tasks JavaScript performs is manipulating the DOM. The DOM is a tree- like representation of the HTML elements on a page. Through JavaScript, developers can add, remove, or update elements in the DOM dynamically. For example, a simple task like changing the text content of a paragraph or adding a class to an element can be done by interacting with the DOM through JavaScript.

###### Example of DOM manipulation:

javascript Copy

document.getElementById("myElement").inner HTML = "Hello, World!";

In this case, JavaScript is accessing the element with the ID my Element and changing its content.

###### JavaScript and User Interactivity

JavaScript is responsible for making a webpage interactive. Whether it's form validation, creating image sliders, handling mouse movements, or responding to button clicks, JavaScript adds the functionality that users expect from modern websites. It can also perform tasks like validating a form before submission or updating content on the page without requiring a full page reload, thanks to technologies like AJAX (Asynchronous JavaScript and XML).

###### AJAX and Fetch API

AJAX allows for asynchronous communication between the client and server, enabling web pages to load data without refreshing the entire page. This is crucial for creating a seamless user experience. The Fetch API is a modern JavaScript method for making AJAX requests, allowing developers to retrieve and send data to a server in real time.

32

# CHAPTER 6

### CONCLUSION

The implementation of this project demonstrates how machine learning can be effectively utilized to recognize food items from images and estimate their calorie content in an automated manner. The proposed system addresses key limitations found in existing methods, such as poor recognition of regional foods, reliance on internet connectivity, and limited user personalization. By training the model on a carefully curated dataset and using efficient preprocessing and classification techniques, the system achieves reasonable accuracy and usability. It simplifies dietary monitoring by allowing users to track their food intake with just a photo, which can be especially useful for people managing health conditions, fitness goals, or general wellness.

In addition, the modular structure of the system allows for easy scalability and integration of future features such as portion size detection, nutrient breakdown, or diet recommendations. The model is privacy-aware and can be deployed in offline environments, making it practical for real-world use in mobile and low-resource settings. This also opens up opportunities for deployment in rural health centers, fitness applications, and educational platforms focused on nutrition.

Furthermore, the project provides a strong foundation for research in food computing and health-based AI applications. With further optimization, the model can be integrated with wearable devices or IoT systems for continuous dietary tracking. Overall, this project showcases the power of artificial intelligence in promoting healthy living through innovative, accessible, and efficient tools for everyday users, and holds great promise for future development in health-tech solutions.

33

### FUTURE RECOMMENDATION

###### Portion Size Estimation:

 Integrate methods to detect and estimate portion size using depth sensing or volume estimation techniques to improve calorie accuracy.

###### Multi-Food Detection:

 Enhance the model to identify and analyze multiple food items within a single image for complete meal analysis.

###### Nutrient Breakdown:

 Extend the system to provide a breakdown of macronutrients (carbohydrates, proteins, fats) and micronutrients for better dietary assessment.

###### User Personalization:

 Incorporate user-specific data such as age, weight, health goals, and dietary preferences to tailor calorie recommendations.

###### Mobile App Deployment:

 Convert the system into a lightweight mobile application for real-time, on-the-go use with offline capabilities.

###### Voice and Text Input Support:

 Allow users to describe their meals via voice or text for better accessibility when image input is not practical.

###### Integration with Health Platforms:

 Sync data with health apps like Google Fit, Apple Health, or fitness trackers to provide holistic health monitoring.

###### Dynamic Food Database Update:

Enable the system to learn from new food items and expand its knowledge base automatically over time.

34

# CHAPTER 7

### REFERENCES

Bolaños, M., Radeva, P., & Peris, R. (2017). "Food recognition in egocentric images using deep learning." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, 1-9

Myers, A., Johnston, N., Rathod, V., Carl, C., Huang, J., & Leung, T. (2015). "Im2Calories: Towards an automated mobile vision food diary." Proceedings of the IEEE International Conference on Computer Vision (ICCV), 1233-1241.

Min, W., Jiang, S., Liu, L., Rui, Y., & Jain, R. (2019). "A survey on food computing." ACM Computing Surveys (CSUR), 52(5), 1-36.

Kawano, Y., & Yanai, K. (2014). "Food image recognition with deep convolutional features." Proceedings of the IEEE International Conference on Image Processing (ICIP), 586-590

Chen, M., Dhingra, K., Wu, W., Yang, L., Sukthankar, R., & Yang, J. (2012). "PFID: Pittsburgh fast-food image dataset." Proceedings of the IEEE International Conference on Image Processing (ICIP), 289-292.

Bettadapura, V., Thomaz, E., Parnami, A., Abowd, G., & Essa, I. (2015). "Leveraging context to support automated food recognition in restaurants." Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV), 580-588.

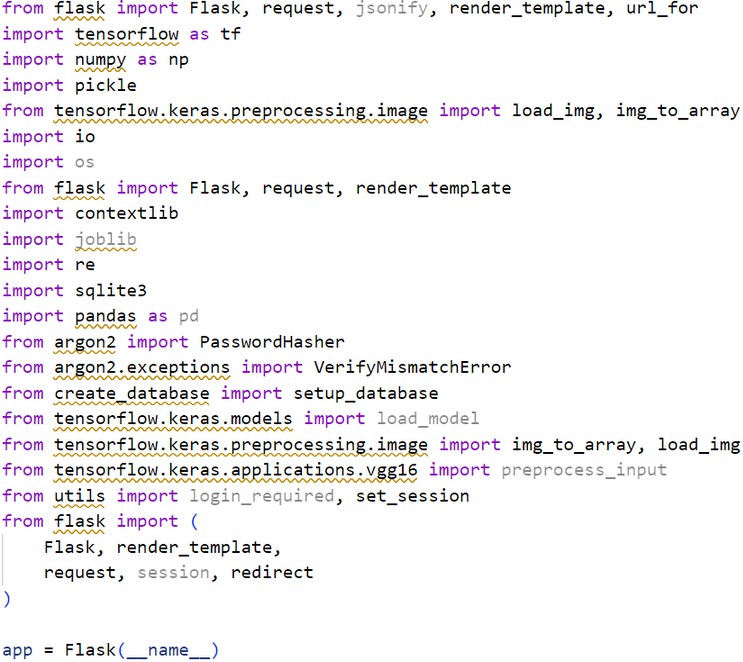
Yanai, K., & Kawano, Y. (2015). "Deep learning food recognition using pre-trained AlexNet model." ACM International Conference on Multimedia (ICM), 577-581.

Lassoued, R., Ayadi, M., Ben Ayed, S., & Mahjoub, M. A. (2020). "Deep learning-based food recognition and calorie estimation for dietary assessment." Multimedia Tools and Applications, 79(5), 30323-30344

35

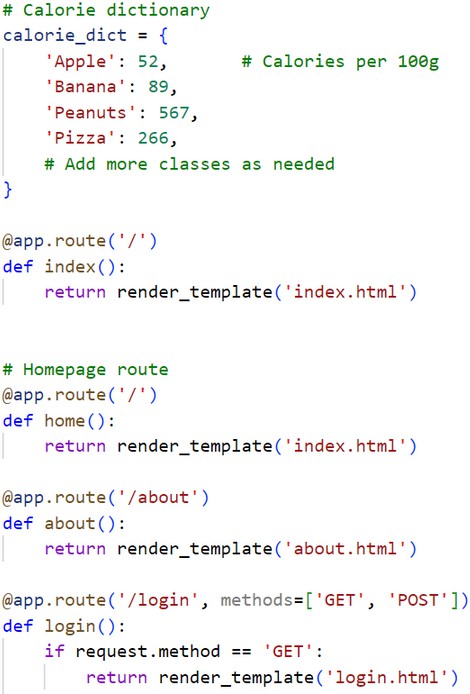
# CHAPTER 8

### SOURCE CODE

****

****

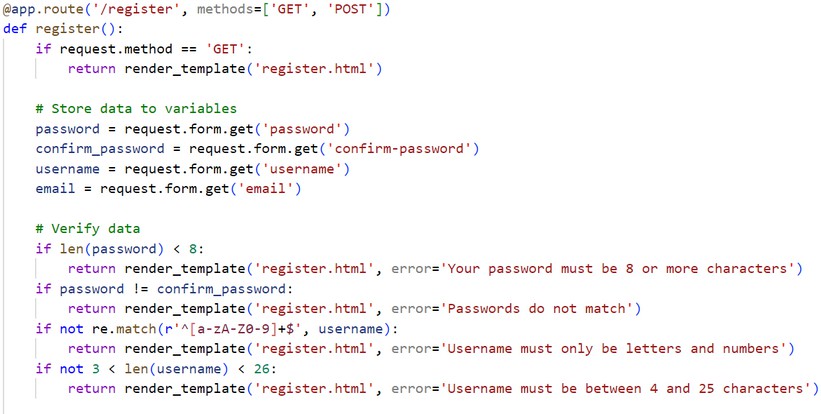
36





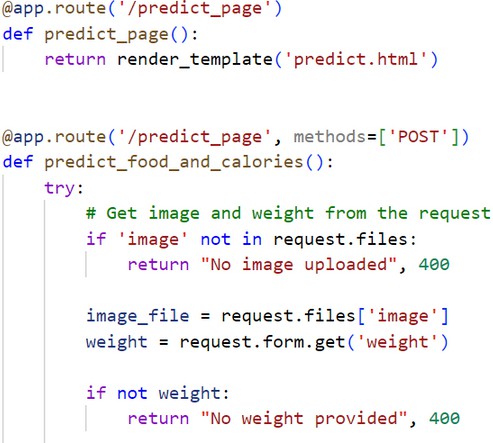
37



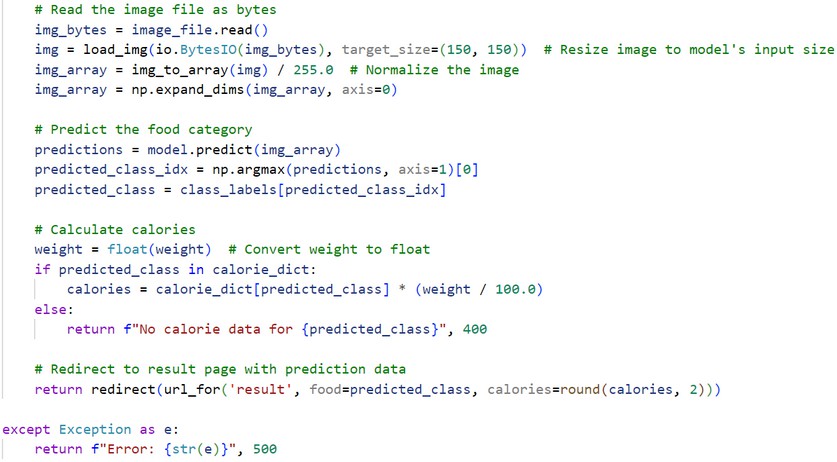


38





39

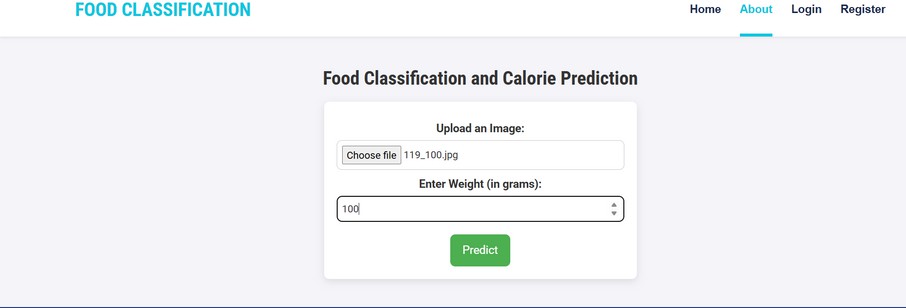


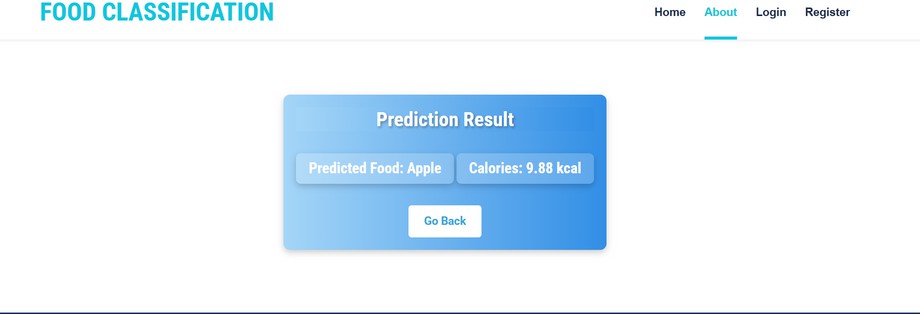


40

# OUTPUT

****

****

****

41