CalorieMate: AI-Powered Calorie and Nutrition Tracker

Ranjitha Durgasi, Nikhitha Reddy Nallanagula, Harshitha Korapati Murali,
Sai Bhargav Ram Koduru, Saikumar Gone, Kariveda Vikranth Reddy
Seidenberg School of Computer Science and Information Systems
Pace University, New York, USA

***Abstract*—CalorieMate is an AI-powered mobile application designed to simplify calorie and nutrient tracking through intelligent food recognition and data visualization. The application enables users to upload meal images, scan barcodes, or enter food manually to obtain real-time nutritional information. Using machine learning models such as InstructBLIP and leveraging cloud infrastructure, the app delivers a seamless and automated experience for users pursuing health goals.**

***Index Terms*—AI nutrition tracker, food recognition, InstructBLIP, calorie counter, mobile health app, PyTorch, Hugging Face, FastAPI, React Native**

I. Introduction

In recent years, there has been a growing awareness of the importance of balanced nutrition and healthy eating habits. However, the process of monitoring dietary intake remains a challenge for many individuals due to the reliance on manual data entry. Traditional nutrition tracking tools often require users to search for food items, estimate portion sizes, and input values into a database — a process that can be time-consuming, error-prone, and discouraging. These limitations significantly reduce user adherence, rendering such applications ineffective for long-term use, especially among individuals without prior knowledge of nutrition science.

CalorieMate was developed as a solution to these challenges by leveraging cutting-edge artificial intelligence (AI) and mobile technology. The app allows users to track their meals with minimal effort through automated image-based food recognition, barcode scanning, and real-time nutrition analysis. By integrating AI-powered models such as InstructBLIP, CalorieMate accurately identifies food items from photos and computes macronutrient content instantly. Coupled with an interactive dashboard that updates dynamically with each logged meal, the app empowers users to make informed dietary decisions aligned with their personalized health goals. This seamless user experience bridges the gap between technological capability and practical nutrition management.

II. Background and Objectives

The demand for digital health and wellness tools has surged in recent years, driven by the widespread adoption of mobile technologies and increased awareness of lifestyle-related health issues. Among the many challenges individuals face, maintaining a balanced diet and tracking nutritional intake are among the most persistent. Traditional methods, such as manually writing down meals or inputting data into apps, often prove to be inconvenient and unsustainable for daily use. Moreover, the accuracy of such methods heavily depends on the user's knowledge of food composition and portion sizes, which can vary significantly from person to person.

With the rise of artificial intelligence (AI) and machine learning (ML), opportunities have emerged to automate and enhance the process of dietary tracking. Leveraging these technologies, CalorieMate was developed to simplify and streamline the way users log meals and monitor their nutrition. The core objectives of the application are as follows:

- Automatically recognize food items from uploaded or captured images using computer vision models.
- Calculate nutritional information in real-time, including calories and macronutrients (carbohydrates, proteins, and fats).
- Provide barcode scanning functionality for packaged food items to ensure quick and accurate data retrieval.
- Display an intuitive daily dashboard that visualizes consumption patterns and remaining dietary goals.
- Enable manual data entry, food editing, and goal customization for enhanced flexibility and user control.

By combining AI capabilities with user-friendly design, CalorieMate aims to empower users with actionable insights that support healthier dietary choices and long-term wellness goals.

III. Related Work

Several mobile applications currently exist in the market that focus on nutrition tracking and healthy lifestyle management. Popular platforms such as MyFitnessPal, Yazio, and Lose It! allow users to log their daily meals, track calories, and set health-related goals. These applications have made strides in promoting nutritional awareness, but they are predominantly dependent on manual user input. Users are often required to search for food items from large databases, estimate portion sizes, and log entries manually—tasks that are repetitive and time-consuming. This reliance on user accuracy can result in inconsistent data and eventually lead to user fatigue or abandonment of the platform.

While some apps have introduced features like barcode scanning or importing meals from external sources, few have integrated artificial intelligence for automated food recognition and nutrient estimation. Emerging research in computer vision and natural language processing has shown that AI can substantially enhance the user experience by removing barriers to entry and increasing data accuracy. CalorieMate leverages this opportunity by using vision-language models for recognizing food items from images and offering real-time analysis. This novel approach not only reduces the cognitive burden on users but also enhances the precision and personalization of dietary feedback.

IV. Technology Stack

The development of CalorieMate incorporates a robust and scalable technology stack designed to optimize performance across multiple platforms and ensure seamless user experience. The architecture is modular and divided into three major components: frontend, backend, and machine learning systems.

Frontend: The mobile application is built using Expo, a framework based on React Native, which enables the development of cross-platform applications for both iOS and Android devices. This approach ensures code reusability and faster iteration, while also providing access to native device features such as the camera and file system for capturing or uploading food images.

Backend: The backend system is developed using FastAPI, a modern, high-performance web framework for building RESTful APIs. It is known for its speed, scalability, and ease of integration with Python-based ML services. The data layer is managed using MongoDB, a NoSQL database well-suited for storing user profiles, meal logs, and food metadata due to its flexible document schema. The entire backend infrastructure is deployed on Amazon Web Services (AWS), providing cloud-based scalability, high availability, and robust security.

Machine Learning: The core functionality of CalorieMate—food recognition from images—is powered by PyTorch, an open-source machine learning framework widely used for developing and deploying deep learning models. The system uses Hugging Face's InstructBLIP model, a state-of-the-art vision-language model trained to interpret visual inputs in natural language contexts. This enables CalorieMate to accurately identify a variety of food items, even in complex or mixed meal images. The ML component is hosted via a model-serving API that communicates with the backend for real-time inference and nutrient computation.

Together, this technology stack supports a powerful and intelligent ecosystem capable of transforming how users engage with nutrition tracking tools.

V. System Architecture and Workflow

The architecture of CalorieMate is designed to facilitate seamless communication between the user interface, backend services, and machine learning inference engines. It follows a modular and event-driven approach, enabling high responsiveness and scalability across devices.

The workflow proceeds as follows:

1) **User Input:** The user initiates interaction by either capturing a new meal image using the camera or uploading an existing photo from their device.
2) **Image Processing and Recognition:** The image is securely transmitted to the backend server, which then passes it to the machine learning inference module hosting the InstructBLIP model. This model analyzes the visual features of the meal and generates structured output in natural language describing the detected food items.
3) **Portion and Ingredient Estimation:** Once food items are identified, a custom estimation logic is applied to infer portion sizes based on visual cues or user input. These recognized items are then cross-referenced with a comprehensive nutrition database, allowing for accurate nutritional mapping.
4) **Nutritional Analysis:** The backend calculates macronutrient values (calories, proteins, fats, carbohydrates) and aggregates them into a structured nutrition profile. If the user has set dietary goals, the values are adjusted and compared against daily targets.
5) **Data Storage and Visualization:** Finalized nutritional data is saved in the user's personal database instance. The frontend retrieves this information and displays it on the dashboard in the form of visual graphs, summary tables, and progress indicators.

This architecture ensures real-time feedback with minimal user effort, providing a frictionless and highly personalized dietary tracking experience.

VI. Core Algorithms

The core intelligence behind CalorieMate lies in its AI-driven recognition pipeline and nutritional estimation logic. The following components form the foundation of the system's computational logic:

Food Recognition: The food recognition engine utilizes the InstructBLIP model from Hugging Face, which combines vision transformers and language models to interpret images through textual understanding. It is fine-tuned for food-related tasks and capable of identifying multiple ingredients within a single meal image, including complex dishes with diverse components. The model provides structured food item names, which are used as keys in subsequent processing.

Nutrition Estimation: After recognition, the food items are passed through a nutritional mapping algorithm. This logic references standard food databases (e.g., USDA FoodData Central) to fetch average nutrient values per 100g or standard serving size. Custom heuristics are applied to estimate portion sizes when not provided explicitly. The final calculation sums macronutrients and adjusts for user-specific modifiers, such as activity level or health goals.

Goal Customization: CalorieMate supports dynamic goal management. Users can specify dietary objectives—such as weight loss, muscle gain, or maintenance—which are translated into daily caloric targets using standard BMR (Basal Metabolic Rate) and TDEE (Total Daily Energy Expenditure) formulas. These targets influence how the dashboard visualizes nutrient consumption progress, helping users stay aligned with their personal plans.

Collectively, these algorithms form an intelligent backbone that transforms raw meal input into actionable, individualized health insights.

VII. Features and Implementation

CalorieMate includes a suite of features designed to provide a seamless, intelligent, and user-centric experience for nutrition tracking. Each feature was thoughtfully implemented to balance automation with user control:

- **User Authentication:** A secure authentication system was developed to handle user sign-up, login, and logout using encrypted API calls. JWT tokens are used to manage sessions and ensure data security.
- **Meal Logging:** Users can log meals in three ways: (1) by uploading or capturing a meal image, which is analyzed using AI for automatic food recognition; (2) by entering food items manually; and (3) by scanning barcodes on packaged foods. All methods are integrated into a unified meal log, which is stored in the cloud.
- **Real-Time Dashboard:** The app includes a visually interactive dashboard that presents real-time nutritional summaries. Users can view total calories consumed, remaining budget, and macronutrient breakdowns via progress rings and charts that update with each new log.
- **Settings Panel:** The settings interface allows users to set and adjust preferences such as unit systems (metric/imperial), health goals (e.g., weight loss or gain), and daily calorie targets. All updates are instantly applied and reflected across the app.
- **History and Weekly Reports:** Users can view their full meal history categorized by date. A summary report feature aggregates weekly data into charts and tables showing average calories and macronutrient trends, helping users analyze long-term dietary habits.

VIII. Development Approach

The CalorieMate development process followed an agile methodology, which provided flexibility in feature planning, iterative feedback, and continuous integration. The project was divided into three sprints, each lasting two weeks.

Sprint 1 focused on foundational features, including user authentication, manual food entry, image upload functionality, and the creation of the initial backend and MongoDB schema.

Sprint 2 introduced enhancements such as barcode scanning, personalized goal setting, macro breakdown logic, and a full history view with edit/delete capabilities.

Sprint 3 delivered AI-powered calorie and nutrient analysis, a weekly reporting system, and the user settings page. This sprint also included model optimization and frontend-backend integration testing.

User stories US1 through US13 were prioritized based on feasibility and user impact. Regular standups, sprint reviews, and retrospectives helped the team remain aligned and responsive to challenges.

IX. Results and Analysis

The final implementation of CalorieMate was evaluated through internal testing and peer feedback. Several important observations were made:

- **AI Performance:** The InstructBLIP model achieved high accuracy in identifying common meals, especially single-item dishes. Recognition of multi-item or region-specific meals posed occasional challenges, suggesting a need for dataset fine-tuning.
- **User Experience:** Test users appreciated the simplicity of the UI, especially the real-time dashboard. The ability to switch between manual and automated entry modes was identified as a key strength.
- **Challenges:** Accurately estimating portion sizes without user input proved difficult. Integrating ML model inference with backend logic while maintaining fast response times required optimization in both API design and model serving.

X. Conclusion

CalorieMate demonstrates a novel approach to nutrition tracking by combining artificial intelligence with intuitive mobile design. Through AI-driven food recognition, customizable goal tracking, and real-time analytics, the app simplifies the traditionally tedious process of meal logging. It empowers users to make informed dietary decisions with minimal effort.

The successful completion of the MVP marks a significant milestone, showcasing the feasibility and effectiveness of combining vision-language AI with mobile health applications. CalorieMate serves as a foundation for future enhancements, including real-world deployment, expanded food recognition capabilities, and integrations with wearable devices and smart assistants.

XI. Future Work

While the current version of CalorieMate fulfills its core objectives, several enhancements are planned to increase functionality, accuracy, and user engagement. The following areas represent the primary focus for future development:

- **Expand Training Dataset:** To improve the accuracy and diversity of food recognition, especially for multi-item dishes and international cuisines, we plan to augment the training dataset with more annotated food images. This will enable the AI model to better generalize across cultures and meal variations.
- **Integrate Wearable Devices and Voice Input:** Integration with smartwatches, fitness bands, and digital health platforms (e.g., Apple Health, Fitbit) will allow real-time activity tracking and passive calorie burn estimation. Voice input capabilities will further simplify logging by enabling hands-free interaction, enhancing accessibility and ease of use.
- **Enhance 3D Portion Size Estimation:** Current models rely heavily on user input or visual cues from 2D images for portion estimation. Incorporating 3D image processing or AR-based volume detection can provide more

accurate estimations, reducing user burden and increasing the reliability of nutrient calculations.

- **Add Social and Community Features:** Future versions will include social sharing and community engagement features, such as group challenges, recipe exchanges, and progress leaderboards. These additions aim to boost user motivation and retention by fostering a supportive health-focused community.

These enhancements will not only improve the app's technical capabilities but also create a more holistic, engaging, and intelligent nutrition-tracking experience for users.

ACKNOWLEDGMENT

This project was developed under the guidance of the Seidenberg School of Computer Science and Information Systems at Pace University, with special thanks to Professor Henry Wong for his mentorship and support throughout the project.

REFERENCES

[1] Hugging Face, "InstructBLIP: A Vision-Language Model for Visual Reasoning," [Online]. Available: https://huggingface.co/docs

[2] PyTorch, "PyTorch: An Open Source Machine Learning Framework," [Online]. Available: https://pytorch.org

[3] U.S. Department of Agriculture, "FoodData Central," [Online]. Available: https://fdc.nal.usda.gov/

[4] MongoDB, "MongoDB Documentation," [Online]. Available: https://www.mongodb.com/docs

[5] S. Tiangolo, "FastAPI: Web Framework for Building APIs with Python," [Online]. Available: https://fastapi.tiangolo.com/

[6] C. Xu, Y. He, and M. Gao, "A Smartphone-Based Food Recognition System Using Deep Learning," in *IEEE Access*, vol. 7, pp. 111467–111477, 2019.

[7] A. Meyers et al., "Im2Calories: Towards an Automated Mobile Vision Food Diary," in *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 1233–1241.

[8] A. D. Thomas and S. Narayanan, "AI-Based Nutritional Estimation from Food Images: A Survey," in *IEEE Reviews in Biomedical Engineering*, vol. 13, pp. 485–497, 2020.

[9] J. Long, E. Shelhamer, and T. Darrell, "Fully Convolutional Networks for Semantic Segmentation," in *Proc. IEEE CVPR*, 2015.

[10] L. Wang, D. Lee, and H. Lu, "BarCodeNet: Barcode Detection with Deep Learning," in *Sensors*, vol. 19, no. 3, pp. 512, 2019.

[11] J. Redmon and A. Farhadi, "YOLOv4: Optimal Speed and Accuracy of Object Detection," arXiv:2004.10934 [cs.CV], 2020.

[12] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet Allocation," in *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, 2003.

[13] R. Burke, "Hybrid Recommender Systems: Survey and Experiments," in *User Modeling and User-Adapted Interaction*, vol. 12, no. 4, pp. 331–370, 2002.

[14] T. Nguyen et al., "Food Image Recognition with Deep Convolutional Features," in *ACM International Conference on Multimedia*, 2017.

[15] Google, "ML Kit: Machine Learning for Mobile Developers," [Online]. Available: https://developers.google.com/ml-kit