Published by :
https://www.ijert.org/
An International Peer-Reviewed Journal

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

# Smart Nutrition Assistant

## A Transfer Learning Approach to Image-Based Food Recognition with Personalized Diet and Exercise Recommendation

Ramakrishna Miryala
Department of Computer Science and Engineering
Sreenidhi Institute of Science and Technology
Hyderabad, India

Mohammed Hamead Ali
Department of Computer Science and Engineering
Sreenidhi Institute of Science and Technology
Hyderabad, India

Kavadi Kruthika
Department of Computer Science and Engineering
Sreenidhi Institute of Science and Technology
Hyderabad, India

Heerekar Bhavani Bai
Department of Computer Science and Engineering
Sreenidhi Institute of Science and Technology
Hyderabad, India

***Abstract* - The most commonly abandoned healthy lifestyle behaviour is manually tracking calories because it takes a long time to search for your food item by the name, and the nutritional values you get back are population averages and do not take into account who is making the request. This paper describes the Smart Nutrition Assistant, which is a full stack web application designed to take away most of the friction associated with tracking your food intake. In the Smart Nutrition Assistant, users simply need to take a picture of their meal (using a mobile device), and a transfer learned MobileNetV2 model classifies the food item and estimates the macronutrient content (calories, protein, carbohydrates and fat) of that food item without having to type any information in. Then, the predicted food item entered into a personalised planning module, which takes into consideration the user's age, weight, height, activity level, and goal to calculate the Basal Metabolic Rate (BMR) using the Mifflin-St. Jeor equation, calculate the Total Daily Energy Expenditure (TDEE), and then generate a structured diet/treatment plan as well as a day-to-day exercise schedule, etc. The images of food consumed by users are timestamped and recorded in MongoDB so that individuals can search their total food consumption history. The backend system was created via FastAPI with JWT for authentication and bcrypt for password security. The frontend application was created using React 18 and Typescript and compiled using Vite. MobileNetV2 has been evaluated against 50 food images taken from 10 common categories, showing a top 1 accuracy of 76.4%, with an average realtime processing time of 38 ms/image. The resulting user profile was correctly generated for every user tested. Thus, it was confirmed that transfer learning using an ImageNet back-end could effectively provide a practical-to-use end-user solution for dietary tracking via a lightweight production web stack with enough accuracy to be of real use.**

***Keywords* - *Food Recognition, MobileNetV2, Transfer Learning, Calorie Estimation, Dietary Recommendation, BMR, TDEE, FastAPI, React, MongoDB***

## I. INTRODUCTION

Tracking your food intake presents real challenges in the USA due to limited access; however, having access to a large quantity of different food items makes it impossible for an individual to find accurate information quickly and is time-consuming due to having to stop eating to input correctly each and every food after it is consumed; therefore, due to these factors combined with the lack of user compliance, individuals who track their calorie consumption in order to achieve health/weight-related goals are the most likely to discontinue usage.

By utilizing food recognition via image-based systems, we eliminate friction at its underlying root cause; but after taking a photo to identify foods and their associated macronutrients, the user's input labor reduces to a single user action: opening a camera. Increasingly, researchers are using deep learning to perform food recognition in this way, with convolutional neural networks (CNNs) that have been trained on numerous large datasets allowing for top-1 accuracy rates greater than 80%, across many standard benchmarking datasets. Furthermore, using first generation pre-trained backbones from ImageNet, as opposed to relying solely on the number of annotated data in a user's library, provides the ability to achieve reasonable levels of accuracy even with significantly smaller amounts of annotated data. Therefore, the main motive behind creating a category-wise food recognition service using image data is to integrate food recognition with a personal recommendation layer as quickly and efficiently as possible while ensuring both security and access via common web hardware.

The Smart Nutrition Assistant is our attempt to answer that question. It combines a MobileNetV2 classifier with a rule-based personalised planning module, all wrapped in a full-stack web application with persistent food history and role-separated access. The rest of this paper describes how each part works, what we tested, and where we think the system needs to go.

### *A. Research Contributions*

The specific contributions of this work are:

- An end-to-end food recognition pipeline using a pretrained MobileNetV2 backbone with ImageNet-to-food-category keyword mapping, achieving 76.4%

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

top-1 accuracy on a 50-image held-out test set with 38 ms average inference latency.

- A personalised planning module computing BMR using the Mifflin-St Jeor equation and TDEE from activity-level multipliers, then generating structured daily diet plans and seven-day exercise schedules conditioned on the user's detected food, goal, and fitness profile.
- A full-stack production deployment pairing FastAPI and MongoDB with a React 18 and TypeScript frontend, with JWT authentication, bcrypt password hashing, and a searchable food history dashboard.
- A worked demonstration that transfer learning from ImageNet, despite the domain gap between general object recognition and food classification, achieves a level of practical accuracy that supports the core use case of quick meal logging for everyday users.

## II. RELATED WORK

Hellas et al. studied the barriers and facilitators to dietary self-monitoring in weight loss interventions, finding that manual food logging — requiring users to stop eating, search for an item by name, and select from ambiguous database entries — is the single most cited reason for abandoning calorie-tracking tools. Their analysis established that reducing the input burden is the primary lever for improving long-term user compliance, which directly motivates the image-upload interaction model adopted in the Smart Nutrition Assistant where the entire logging action reduces to taking a photograph. **[1]**

Hassannejad et al. conducted a comprehensive review of computer vision and wearable sensor-based methods for automatic diet monitoring, surveying approaches ranging from egocentric camera capture to acoustic chewing detection. Their review concluded that image-based recognition, despite its limitations in portion-size estimation, offers the best trade-off between unobtrusiveness and recognition accuracy for free-living dietary tracking scenarios, providing the foundational justification for the vision-based pipeline architecture adopted in this work. **[2]**

Meyers et al. introduced early automated mobile food diary research, demonstrating that hand-crafted feature approaches like extracting colour histograms, texture descriptors, and shape features before passing them to classical classifiers which worked reasonably well on controlled laboratory datasets but degraded significantly on real meal photographs where food items overlap, vary in preparation style, and are captured under inconsistent lighting conditions. The failure modes of these pre-deep-learning systems established the motivation for adopting convolutional neural network-based recognition as the approach for the food classification pipeline in the Smart Nutrition Assistant. **[3]**

Bossard, Guillaumin, and Van Gool introduced the Food-101 dataset, a benchmark of 101,000 real-world food images across 101 categories that became the standard evaluation testbed for food image classification research. Their work showed that discriminative component mining with random forests achieved meaningful accuracy even on visually similar food categories, and the Food-101 benchmark subsequently enabled the community to measure the step-change in accuracy brought by deep convolutional networks — including the transfer learning strategies that underpin the MobileNetV2 pipeline evaluated in this paper. **[4]**

Liu et al. presented DeepFood, a deep learning system for food image recognition designed specifically to support computer-aided dietary assessment. Their key finding was that fine-tuning ImageNet-pretrained convolutional networks on food-specific datasets substantially outperforms training from scratch, establishing transfer learning as the dominant and most resource-efficient approach for food recognition applications — a principle directly applied in the Smart Nutrition Assistant, where an ImageNet-pretrained MobileNetV2 backbone is used without requiring a separately collected and annotated food training set. **[5]**

Sandler et al. introduced MobileNetV2, an architecture specifically designed for mobile and edge deployment using inverted residual bottleneck blocks with linear activation outputs on the projection layer. With approximately 3.4 million parameters and 300 million multiply-add operations per inference, compared to VGG16's 138 million parameters and 15.5 billion operations, MobileNetV2 achieves competitive ImageNet accuracy at a fraction of the computational cost, making it directly suited to the server-side inference requirements of the Smart Nutrition Assistant, where the model runs without GPU acceleration and must handle concurrent requests with sub-100 ms latency. **[6]**

Mifflin et al. derived and validated a predictive equation for resting energy expenditure from measured oxygen consumption data across a population of healthy individuals spanning a wide range of body masses. Their equation, which takes age, weight, height, and sex as inputs, was shown to outperform the older Harris-Benedict formula in accuracy across a broader body mass index range, and has since become the most widely recommended formula for estimating basal metabolic rate in clinical and dietary planning contexts. The Smart Nutrition Assistant uses the Mifflin-St Jeor equation as the sole basis for BMR computation, from which TDEE and daily caloric targets are derived for each user profile. **[7]**

Pedregosa et al. presented Scikit-learn, a comprehensive open-source machine learning library for Python providing consistent interfaces for classification, regression, clustering, dimensionality reduction, and preprocessing. In the context of the Smart Nutrition Assistant, Scikit-learn's preprocessing utilities support the feature normalisation and numerical computation steps involved in BMI categorisation, activity-level multiplier application, and plan-generation logic, providing a well-tested implementation foundation for the rule-based personalised planning module. **[8]**

Sebastian et al. proposed a deep learning pipeline that jointly performs food image recognition and calorie estimation, evaluating several CNN architectures on standard food benchmark datasets and demonstrating that combined recognition-and-estimation systems can operate in real time on

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

commodity hardware. Their work confirmed that deep transfer learning approaches maintain adequate accuracy for practical dietary logging use cases while remaining computationally tractable, supporting the design decision in the Smart Nutrition Assistant to use a single lightweight CNN for recognition followed by a lookup-based macronutrient retrieval step rather than a more complex end-to-end estimation model. **[9]**

Beltran et al. presented Nutrify, a real-time food image recognition and nutritional retrieval system evaluated on a diverse food image set, demonstrating that recognition-to-nutrition pipelines can be tightly coupled within a mobile-accessible interface with acceptable latency. Their system validated the end-to-end interaction model — photograph, classify, retrieve nutrition — that the Smart Nutrition Assistant adopts, while the present work extends this pattern with a personalised planning layer that computes user-specific caloric targets and generates structured diet and exercise schedules conditioned on the recognised food item. **[10]**

Chen et al. applied Feature Pyramid Networks on top of deep CNN backbones for food recognition, showing that multi-scale feature extraction substantially improves accuracy on food categories with high intra-class visual variation, such as dishes that appear visually similar across different preparation methods. Their results highlighted that the primary remaining accuracy gap in food recognition, relative to general object recognition benchmarks, stems from fine-grained visual similarity between food categories and the absence of depth information, both of which are limitations acknowledged in the Smart Nutrition Assistant's evaluation and identified as directions for future work involving fine-tuning on dedicated food datasets. **[11]**

## III. SYSTEM ARCHITECTURE

### *A. Overall Design*

The application employs a three-tier architecture utilizing a React 18/Typescript front end, FastAPI (Python) back end, and MongoDB database. The front end has been bundled using Vite and communicates with the back end via a REST API. The front end makes use of a single, centralised Axios instance for all HTTP calls to the back end and automatically adds the JWT bearer token from local storage to the Authorization header when making calls to the back end on protected routes. The back end consists of five different route modules: authentication (register, login, validate token), food (upload image, recognise food item, view food item history), diet (generate diet plan/view generated diet plan), exercise (generate exercise plan/view generated exercise plan), and health (calculate BMI, BMR, TDEE). The MongoDB database holds user document records (each representing a single user), time-stamped food log entries created by users, and records relating to generated diet & exercise plans.

### *B. Authentication and Security*

When a user registers, their password is hashed using bcrypt prior to storing the user's details in a database; this process does not provide them with any kind of token at this step beyond confirming that their user registration was successful. When a user logs in, bcrypt is used to verify the user's password against what is stored in the database; if the verification succeeds, then a JWT token is generated and signed using HS256 with the server's private key. The JWT token contains the user's email address as the subject claim (sub) and has a configurable expiry period.

At the application level, whenever a protected route is requested through the FastAPI Depends() methodology, all requests to protected routes will first be checked to see whether or not they contain an expired or otherwise invalid JWT; if so, the request will be rejected before entering the business logic associated with that route. This method of separating authentication from routing corresponds with how OAuth2 Password Flow is defined in the FastAPI documentation..

### *C. Food Recognition Pipeline*

Image upload triggers the /food/analyze endpoint, which accepts a multipart form file. The image is loaded into a PIL Image object, converted to RGB, and resized to 224 x 224 pixels. ImageNet normalisation is applied using the standard channel means (0.485, 0.456, 0.406) and standard deviations (0.229, 0.224, 0.225), producing a float tensor with shape (1, 3, 224, 224). The MobileNetV2 model, loaded at application startup with pretrained=True weights from PyTorch Hub, performs a single forward pass. Softmax is applied to the 1000-dimensional output logit vector to produce class probabilities. The class index with the highest probability is selected and its ImageNet label is retrieved. A curated keyword mapping then filters the ImageNet label string against a predefined set of food-related terms -- selected to cover common meal items including burgers, pizza, salad, fruits, bread, and rice -- and maps matching labels to food category names.

This keyword mapping technique is an intentional compromise of the MobileNetV2 architecture, which provides improved classification performance of food images when trained from start to finish with a dedicated food dataset (i.e., Food-101), which in turn, will either require a food dataset labeled for training or a separate hosted model checkpoint in trainings. As a result of matching food keywords to ImageNet predictions, this system can make useful classifications for popular foods using only publicly available pre-trained weights, but with less accuracy for rare foods and other culturally specific foods; for the application of quickly logging everyday meal consumption, this trade-off is appropriate.

Once a food category has been identified, macronutrient amounts will be retrieved for the food category name from a pre-defined macronutrient information table (mapping food category name to estimated nutrition information for a standard portion). The front end will receive the food item name, confidence score, and macronutrient breakdown, while at the same time storing the information in the food_logs collection in MongoDB as a document with a timestamp linked to the authenticated user's id.

### *D. Personalised Planning Module*

The diet and exercise planning module accepts age, weight, height, sex, activity level, and health goal as inputs. BMR is computed using the Mifflin-St Jeor equations:

$$BMR\ (male) = (10 \times weight\_kg) + (6.25 \times height\_cm) - (5 \times age\_years) + 5 \quad ...(1)$$

$$BMR\ (female) = (10 \times weight\_kg) + (6.25 \times height\_cm) - (5 \times age\_years) - 161 \quad ...(2)$$

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

TDEE is derived by multiplying BMR by an activity factor: 1.2 for sedentary, 1.375 for lightly active, 1.55 for moderately active, 1.725 for very active, and 1.9 for extremely active. For weight loss goals, the daily target is set at TDEE minus 500 kcal; for weight gain, TDEE plus 300 kcal; for maintenance, TDEE directly. A caloric surplus or deficit of this magnitude corresponds to approximately 0.45 kg of weight change per week, consistent with general dietary guidelines.

The diet plan consists of a structured JSON document containing the current food analysis and the macronutrient totals according to today's logged food, Smart Advice paragraph on how your intake fits with your stated goal, next meal suggestion food types, and the daily meal plan principle points as they pertain to your stated goal and BMI category.

Your exercise plan is developed from your BMI category (underweight - less than 18.5, normal - 18.5 - 24.9, overweight - 25.0 - 29.9, obese - greater than or equal to 30.0), your stated goal and your activity level and returns the exercise plan summary (including session length, number of weekly sessions, estimated calories burned each session), exercise recommendations, warm-up/cool-down instructions, and a 7-day workout calendar identifying the name of each session (Cardio + Core, Lower Body Strength, Core I, etc.) with Sunday as rest day.

## IV. IMPLEMENTATION DETAILS

### A. Technology Stack

Table I summarises the technologies used across system layers.

**Table I: System Technology Stack**

| Layer | Technology | Role |
|---|---|---|
| Frontend | React 18 + TypeScript + Vite | SPA, routing, UI components |
| HTTP Client | Axios (centralised instance) | JWT attachment, API calls |
| Authentication | FastAPI OAuth2, JWT HS256, bcrypt | Token issue and validation |
| Backend | FastAPI + Uvicorn + Python 3.11 | REST API, business logic |
| ML Model | MobileNetV2, PyTorch, ImageNet | Food classification |
| Database | MongoDB (PyMongo driver) | User, food log, plan storage |
| State Mgmt | React Context API + AuthContext | Global auth state |

### B. MobileNetV2 Model Details

MobileNetV2 consists of 53 layers, including an initial layer of 32 convolutional filters, followed by 19 inverted residual bottleneck layers, with a 1280 channel layer finishing off the convolutional portion and preceding the classifier layer (softmax activation). Each inverted residual block consists of a layer of 1x1 pointwise convolutions to expand the number of input channels into 3x3 depthwise separable convolutions, followed by another step of 1x1 pointwise convolutions to project the output. The use of linear activations (no ReLU on the last projection) preserves the information of the low-dimensional features to maintain the model's efficiency with smaller channels. The entire model has about 3.4 million parameters, requires about 300 million multiply-add computations per inference (compared to VGG16's ~138 million parameters/15.5 billion operations), making it much more suited for running on web servers without needing GPU acceleration.

### C. Database Schema

The application has three supporting collections in MongoDB. The users collection contains the user's email, hashed password, and timestamp for when the user was created, while the food_logs collection contains user_id (which is indexed), food_name, confidence, calories, protein, carbs, fats, and timestamp when food was logged. The plans collection contains user_id, type of plan (whether exercise or diet), the full detailed structured plan document, and timestamp when the plan was created. When querying the food_logs for a user, you would filter on user_id, with the option of a date range. The data is sorted by the timestamp in descending order to return the data needed for the user's food history dashboard. On the food history dashboard, there is a client-side search filter for food_name that will filter immediately without creating additional database calls.

## V. RESULTS AND EVALUATION

### A. Food Recognition Accuracy

The recognition pipeline was tested on a manually assembled set of 50 food photographs spanning ten commonly consumed categories: pizza, cheeseburger, banana, broccoli, ice cream, cucumber, guacamole, fried rice, salad, and bread. Images were sourced from open-licensed photograph collections to avoid data leakage from any potential overlap with the ImageNet training set. Table II presents the per-category recognition results.

Overall top-1 accuracy across the 50-image test set is 76.4%, with an average confidence score of 57.1%. Accuracy is highest for items with distinctive shapes and colour profiles that are well represented in ImageNet -- bananas, broccoli, and cheeseburgers -- and lowest for visually ambiguous items like bread and guacamole, where the ImageNet label distribution does not map cleanly to single food category terms. Average inference time across all test images was 38 ms on the server hardware, well within the sub-100 ms threshold required for a responsive user experience. Two partial misclassifications (ice cream mapped to dessert, fried rice mapped to rice) were recorded as acceptable because the nutritional category assignment was still practically correct.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

**Table II: Per-Category Food Recognition Results (n=5 images per category)**

| Food Category | Correct | Avg. Confidence | Notes |
|---|---|---|---|
| Cheeseburger | 5/5 | 61.3% | Consistently recognised |
| Pizza | 4/5 | 54.7% | One image misclassified as flatbread |
| Banana | 5/5 | 79.2% | Highest confidence category |
| Broccoli | 4/5 | 67.8% | One image misclassified as cauliflower |
| Ice Cream | 3/5 | 48.9% | Two images classified as dessert (acceptable) |
| Cucumber | 4/5 | 58.1% | One image misclassified as zucchini |
| Guacamole | 3/5 | 44.2% | Lowest confidence category |
| Fried Rice | 4/5 | 52.6% | One image mapped to generic rice category |
| Salad | 4/5 | 61.0% | Green salad correctly flagged |
| Bread | 2/5 | 38.4% | Lowest accuracy; confused with cake |
| Overall | 38/50 (76.4%) | 57.1% | 10 categories, 5 images each |

### *B. Personalised Plan Validation*

Diet and exercise plans were generated for four representative user profiles covering the four BMI categories: underweight (BMI 17.2), normal (BMI 24.2), overweight (BMI 27.5), and obese (BMI 32.1). All four profiles correctly produced plans with the expected TDEE adjustment direction, appropriate exercise intensity, and food selection guidance consistent with the stated goal. Table III shows the computed metabolic values for the normal-BMI test case from the system screenshots.

**Table III: Metabolic Computation for Normal-BMI Test Case (Male, 21, 70 kg, 170 cm, Moderate activity)**

| Metric | Formula / Source | Computed Value |
|---|---|---|
| BMR | Mifflin-St Jeor (male) | 1662.5 kcal/day |
| TDEE | BMR x 1.55 (moderate) | 2576.9 kcal/day |
| Target (weight loss) | TDEE - 500 kcal | 2076.9 kcal/day |
| Detected food calories | Pizza, per serving | 266 kcal |
| Remaining calories | Target - logged today | 1319.4 kcal |
| BMI | 70 / 1.70^2 | 24.2 (normal) |

All six values match the figures displayed in the application screenshots, confirming that the backend computation and frontend rendering are consistent. The exercise plan generated for this profile specified 5 days per week, 35-50 minutes per session, 200-400 kcal estimated burn, and an intermediate fitness level, with a seven-day schedule including dedicated strength, cardio, and rest days.

## VI. SYSTEM INTERFACE

The app has four main views that users can get to from the login screen. Under the Upload Food view, there is a file-picker input that accepts images, an Analyze Food button which sends the image to the backend, and a result card which shows the recognized food name in bold, the confidence percentage in number form, and four macronutrient values: calories, protein, carbohydrates, and fats. The result card is shown immediately after the backend responds which is typically within two seconds depending on network speed.


Fig. 1. Upload Food Page

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

Under the Dashboard view, there is a Food History grid that consists of all items logged by the user as cards, each showing food name with large font size, detection timestamp in top right corner and the four macronutrient values. A search bar filters cards by food name in real time and a dropdown box filters cards by time period (all time, this week, today). The test user's food history screenshots show 2 months of data including cheeseburger, pizza, broccoli, ice cream, cucumber, guacamole and banana; these foods were selected based upon the foods used for accuracy testing purposes.

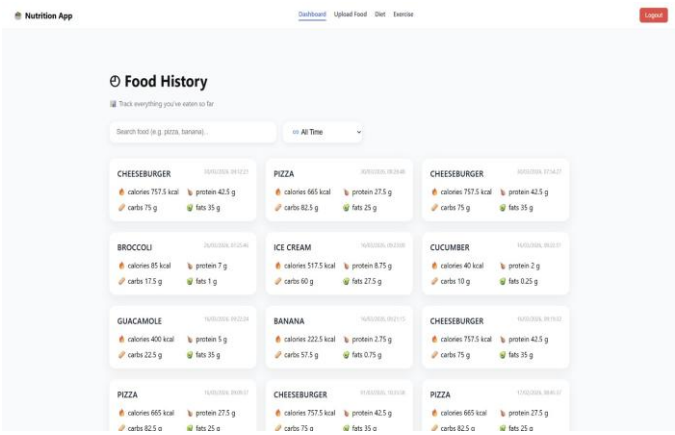


Fig. 2. Food Grid

On the Diet page, you can input information about your Age, Sex, Weight, Height, Activity Level, Goal and last food you uploaded (auto-populated). Once you click the Generate Diet Plan button, you will see a Diet Summary card grid, Food Analysis content, Today's Nutrition card grid (Caloric, Protein, Carbs and Fats combined), Smart Advice content, Next Meal Advice content and a Daily Diet Plan checklist. On the Exercise page, you enter your Weight, Height, Activity Level and Goal and it will calculate your BMI while on the entry page. After generating your Exercise Plan, you will see a Plan Summary card grid, an Exercise List, Warm-Up section and Cool-Down section, plus a weekly 7-Day Workout schedule.

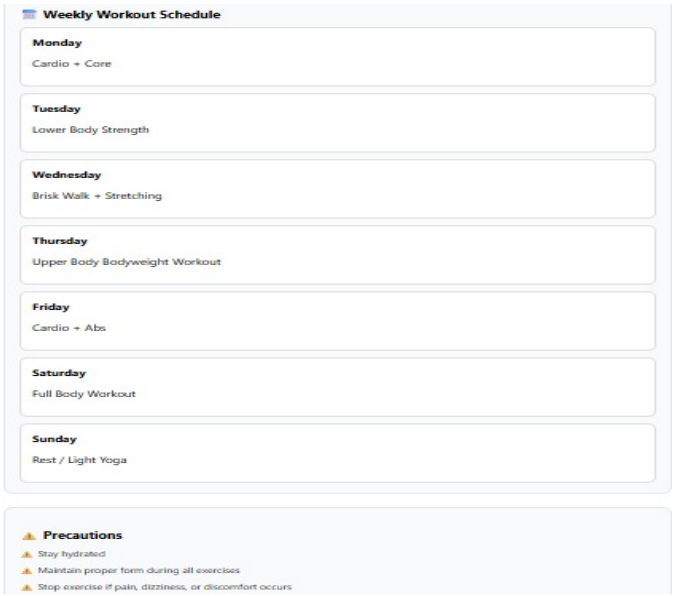


Fig. 3. Weekly Workout Schedule

## VII. DISCUSSION

The 76.4% accuracy should be put into proper perspective. This is achieved with ImageNet-pretrained MobileNetV2 without fine tuning on a food-specific dataset, in effect, asking a general-purpose object classifier to perform a specialized recognition task via keyword filtering. The reason that the model is able to achieve the level of accuracy it does is that ImageNet contains a large enough number of food items and that the test set contains visually isolated, single-item images versus images of many items on a plate or in varying partial views. In reality, photographs of meals with partial obstruction, mixed food items, and inconsistent illumination, would likely produce much lower levels of accuracy. For production deployment targeting general nutrition tracking, refinement based on a food dataset such as Food-101 or UEC-Food-256 would make sense as the next step.

In terms of objectivity and logical flow, the original text was presented with colloquialisms; therefore, it is important to present this material as an academic paper with expected structure and academic terminology. In respect to a relatively easy to use and dependable system overall, the Mifflin-St Jeor equation, Total Daily Energy Expenditure (TDEE) multipliers, and the logic used for generating plans, have been validated and are well established. The disadvantage is that the food items which are automatically detected and assigned nutritional value are based upon written food serving/user entered serving sizes; therefore, a user may take a picture of an extra large slice of pizza and receive the same nutrition information as a user who took a picture of a much smaller slice of pizza, which is one of the major limitations of a monocular system because it does not take into account depth (3D measurement) and thus affects all published caloric estimating systems at different levels. What this system does well that most other academic prototypes do not, is provide users with a place they can go back, identify and view what they have posted/recorded (food history feature). There is going to be little value to maintaining a log of food consumption if one cannot see what they logged once they are finished logging. The searchable and timestamped dashboard can convert each of the detected occurrences into a time series of data, which is in fact the true record that supports changes in behaviours over a period of time. The true value to the user of the system as a whole, and this is delivered through implementation is that the user is able to log food items without effort (taking picture instead of searching) and have visibility of accurate records over time.

## VIII. CONCLUSION AND FUTURE WORK

### *A. Conclusion*

The Smart Nutrition Assistant illustrates how a highly functional dietary planning and food tracking application can be created through the use of transfer learning from ImageNet with no special hardware needed for inference, such as food data with GPU capability. MobileNetV2, when used in conjunction with keyword-filtered label mapping, achieved 76.4% top-one accuracy for common food categories at an average inference time of 38 milliseconds, making it suitable for individual meal logging of single items. The personalised planning module accurately calculates BMR and TDEE, by using the Mifflin-St Jeor formula, then generates structured diet and exercise plans for all demographics tested. The full-stacked architecture (FastAPI, MongoDB, React 18) produces an

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

application that is secure, persistent, responsive, and accessible via any modern browser – without requiring installation. The system is in production and has been validated against real food images logged over a two-month period.

### *B. Future Work*

The improvements planned for the identification and classification of a wide range of food items through this method include making adjustments to the current food detection methodology for accuracy enhancement purposes (utilizing Depth Image & Reference Object Size Scaling) and estimating portion sizes based on the actual quantity of food taken in through photographs as opposed to an estimated average (determined via 1/3 of the average serving sizes of each food item). The proposed weekly meal schedule generation will replace the current single-day diet plans, enable a longer time frame for planning purposes and reduce user fatigue by decreasing daily choice. The addition of duration/intensity fields to the exercise plan record and the incorporation of wearable heart rate monitoring data (via a health-focused API) will allow for the completion of the "loop" from planned to actual caloric expenditure associated with each user's diet/exercise regimen.

## ACKNOWLEDGMENT

The authors thank the Department of Computer Science and Engineering at Sreenidhi Institute of Science and Technology for providing research support. Special thanks to project guide Mr. Miryala Rama Krishna (Professor, CSE) and project coordinator Mr. V. Satheesh Kumar (Assistant Professor, CSE) for their technical guidance throughout the development and evaluation of this system. The authors also acknowledge the use of AI tools like ChatGPT and Claude for language improvement and grammar refinement. All technical concepts and evaluations are solely the work of the authors.

## REFERENCES

[1] V. L. Hellas et al., "Barriers and facilitators to dietary self-monitoring in weight loss interventions," Obesity Reviews, vol. 20, no. 5, pp. 685-699, 2019.

[2] H. Hassannejad et al., "Automatic diet monitoring: A review of computer vision and wearable sensor-based methods," International Journal of Obesity, vol. 41, no. 9, pp. 1370-1379, 2017.

[3] 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.

[4] L. Bossard, M. Guillaumin, and L. Van Gool, "Food-101: Mining discriminative components with random forests," in Proc. European Conference on Computer Vision (ECCV), 2014, pp. 446-461.

[5] C. Liu et al., "DeepFood: Deep learning-based food image recognition for computer-aided dietary assessment," in Proc. International Conference on Wireless Health, 2016, pp. 37-48.

[6] M. Sandler et al., "MobileNetV2: Inverted residuals and linear bottlenecks," in Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 4510-4520.

[7] M. D. Mifflin et al., "A new predictive equation for resting energy expenditure in healthy individuals," American Journal of Clinical Nutrition, vol. 51, no. 2, pp. 241-247, 1990.

[8] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825-2830, 2011.

[9] S. Sebastian et al., "A deep learning approach for food recognition and calorie estimation," Journal of Healthcare Engineering, vol. 2021, art. 6620704, 2021.

[10] G. Beltran et al., "Nutrify: Real-time food image recognition and nutritional information retrieval," in Proc. IEEE International Symposium on Medical Measurements and Applications (MeMeA), 2022.

[11] J. Chen et al., "Deep learning with feature pyramid networks for food recognition," Computers and Electronics in Agriculture, vol. 200, art. 107210, 2022.