IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

# Fithub- AI Enhanced Smart Vision For Fitness Optimization

[1]Pranita Sangit, [2]Falak Gala, [3]Shubham Sawant, [4]Faisal Shah, [5]Aniket Shrungare

[1]Department of Computer Science & Engineering,

[1]K.C. College of Engineering and Management, Thane, India

***Abstract:*** This paper presents a groundbreaking fitness application that leverages state-of-the-art technologies to redefine the wellness journey. The project seamlessly integrates a Smart Camera, powered by TensorFlow and Flask, allowing users to effortlessly identify and access information about gym equipment. The inclusion of a Pedometer, driven by accelerometer sensors, provides a practical dimension by tracking daily physical activity. A sophisticated Diet Recommendation system formulates personalized diets based on individual profiles, calculating daily caloric needs and organizing them into breakfast, lunch, and dinner. The Exercise section, categorized by body parts, enables users to select and perform exercises with detailed instructions. The integration of Google's Palm API establishes a direct channel for users to engage with fitness-related information. Themes, authentication, and authorization mechanisms using Firebase contribute to a secure and user-friendly environment. This project represents a comprehensive fusion of technology and holistic wellness, poised to reshape the landscape of fitness applications.

## 1.INTRODUCTION

In the ever-evolving landscape of fitness technology, FitHub emerges as a beacon of innovation, fusing cutting-edge tools to redefine the conventional fitness app paradigm. With a foundation grounded in TensorFlow, Flask, OpenAI API, and Firebase authentication, FitHub is not merely a digital workout guide; it's a holistic wellness companion. This project responds to the nuanced demands of contemporary health enthusiasts, acknowledging that fitness encompasses more than structured exercise routines.

FitHub's unique proposition lies in its multifaceted approach, integrating a Smart Camera, Pedometer, Diet Recommendation System, Exercise Section, and Chat API. These features collectively transcend the traditional boundaries of fitness applications, creating an ecosystem that caters to the diverse needs of users. This introduction lays the groundwork for an exploration of how FitHub seamlessly intertwines technology and wellness, offering users a dynamic and personalized fitness experience.

In essence, FitHub is a testament to the symbiosis of technology and well-being. By introducing a Smart Camera that revolutionizes equipment interaction, a Pedometer that extends insights beyond planned workouts, and a Diet Recommendation System that personalizes nutrition plans, FitHub redefines the role of a fitness application. As users delve into the project, they embark on a transformative journey where fitness is not a rigid routine but a dynamic and personalized lifestyle. FitHub is poised to reshape the fitness app landscape, making well-being a deeply personalized and accessible reality for users around the globe.

## 2. Methodology

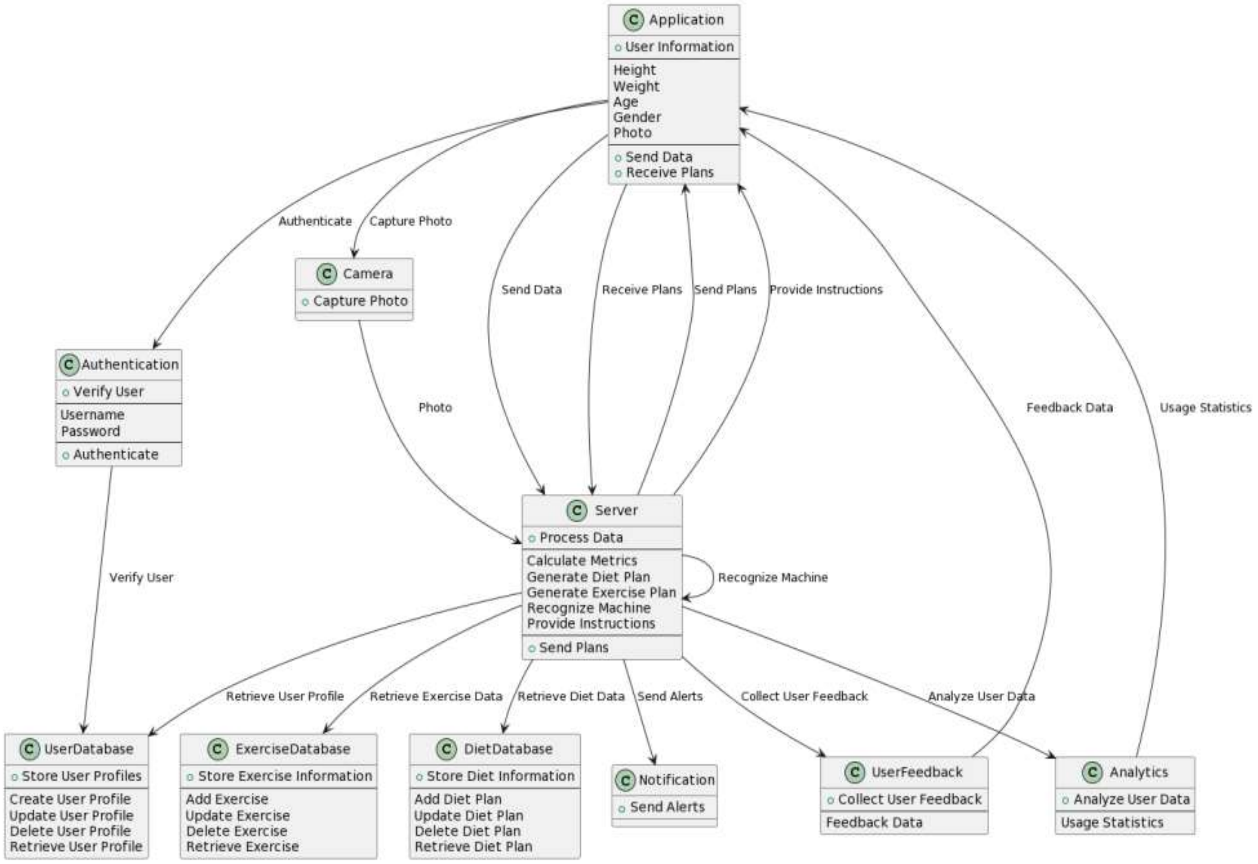
### *2.1* Project Initialization

The inception of FitHub began with a meticulous definition of its vision and scope. A cross-functional project team, comprising experts in app development, design, and fitness domain knowledge, was assembled. Clear goals and milestones were established, providing a roadmap for the project's progression.

### *2.2* Market Research

A comprehensive market research phase was undertaken to comprehend current trends and competition in the fitness app industry. This involved identifying target users, understanding their needs, and discerning potential pain points to inform the development process

### 2.3 Data Flow Diagram




### 2.4 Frontend Development

The frontend of FitHub, our fitness application, was meticulously crafted using React Native, a popular framework known for its ability to deliver responsive and seamless user experiences across multiple platforms. By leveraging React Native, we ensured that FitHub's user interface (UI) remains adaptable and performant, catering to the diverse needs of our users. The development process entailed translating the finalized design specifications into interactive and visually appealing UI components, meticulously aligning with the project's aesthetic and functional objectives. Each UI element, from buttons to navigation bars, was carefully designed and implemented to enhance usability and engagement. Special attention was paid to achieving consistency in design elements and layout, fostering a cohesive and intuitive user experience. Moreover, React Native's component-based architecture facilitated modular development, enabling us to iterate rapidly and integrate new features seamlessly. Through iterative design iterations and user testing, we refined the UI to prioritize usability, accessibility, and overall user satisfaction. The result is a frontend interface for FitHub that not only meets but exceeds user expectations, providing a smooth and enjoyable fitness experience for our users.

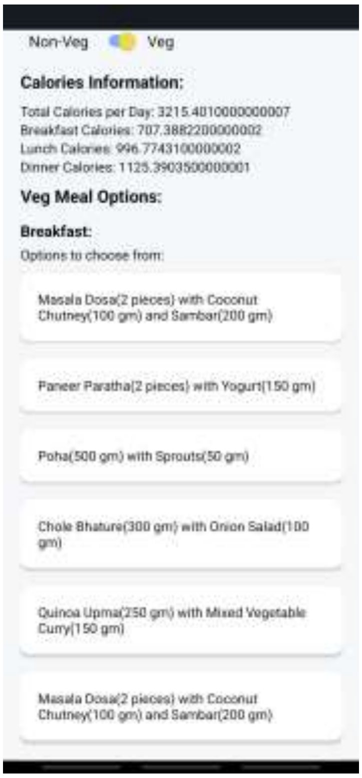
### 2.5 Backend Development

In the backend architecture of our system, we have adopted a comprehensive approach by integrating various APIs and services to cater to distinct functionalities. The last API has been employed for gym detection, utilizing advanced computer vision techniques to identify gym environments and activities within multimedia content. For authentication and authorization processes, a class API has been utilized to establish a direct emulsion system, ensuring secure management of user credentials and permissions. Our backend infrastructure relies on Google's Cloud Firestore as the database solution, providing scalability and real-time data synchronization capabilities. Furthermore, for AI-driven conversational interfaces, Google's Palm API has been implemented, enabling natural language processing functionalities for chatbot interactions. To facilitate activity tracking, particularly in the context of physical exercise, a pedometer feature has been integrated using accelerometer data from body-worn devices. This accelerometer data is processed and stored in a database sourced from Rapid API, allowing users to monitor their exercise progress over time. This backend architecture exemplifies a well-rounded integration of APIs and services, tailored to deliver diverse functionalities efficiently within our system.

## 3. Features

### 1.Diet Recommendation

The Diet Recommendation feature was meticulously designed and implemented. Three separate CSV files (breakfast.csv, lunch.csv, dinner.csv) were created and populated with food items. User inputs, including personal information and fitness goals, were collected and processed to generate personalized diet plans.



### 2. Machine Detection

For gym machine detection, a dataset of machine images was collected and formatted for TensorFlow model training. The model, chosen or designed based on accuracy considerations, underwent training and fine-tuning. The TensorFlow model was then exported in H5 format. Flask API development facilitated communication between the model and the React Native app, ensuring seamless integration.


### 3. Chatbot

The Chatbot feature underwent a systematic methodology, involving requirement analysis, framework/API selection, integration planning, and UI component design. API integration with Google's Palm API was implemented, enabling dynamic user interactions. Extensive testing, including unit tests and UAT sessions, ensured a smooth and responsive user experience.

**4. Pedometer**

The Step Counter feature with Pedometer integration followed a systematic approach. Expo Sensors library was selected, and integration planning defined user interface components for step count visualization. Extensive testing validated accuracy, responsiveness, and seamless integration.


**3.3.1** GYM DETECTION MACHINE LEARNING ALGORITHMS

1. **Data Preparation:**
   The code begins by importing necessary libraries and mounting Google Drive to access the dataset.
   It sets up the data directory and defines parameters such as image height, width, and batch size.
   It loads the dataset using tf.keras.preprocessing.image_dataset_from_directory(), splitting it into training and validation sets.

2. **Data Validation:**
   The code checks the validity of images in the dataset. It iterates through the directory, checking for valid image file extensions and removing any unsupported or corrupt images.

3. **Model Definition:**
   The CNN model is defined using Keras' Sequential API.
   It starts with a preprocessing layer for rescaling pixel values to the range [0,1].
   Then, a series of convolutional (Conv2D) and max-pooling (MaxPooling2D) layers are added to extract features and reduce spatial dimensions.
   The convolutional layers use ReLU activation functions to introduce non-linearity.
   After the convolutional layers, there's a flattening layer to convert the 2D feature maps into a 1D vector.
   Finally, there are two dense (fully connected) layers with ReLU and softmax activation functions for classification.

4. **Model Compilation:**
   The model is compiled using the Adam optimizer, sparse categorical cross-entropy loss function, and accuracy as the evaluation metric.

5. **Image Preprocessing:**
   Before training, images are preprocessed by resizing and rescaling pixel values.

6. **Training:**
   The model is trained using the fit() function on the training data, with validation data for evaluation. The training process runs for a specified number of epochs.

7. **Image Prediction Interface:**
   After training, a Gradio interface is set up to predict the class labels of input images.
   The predict_image() function takes an image as input, preprocesses it, and returns the predicted class probabilities.

**3.3.1** DIET RECCOMENDATION ALGORITHMS

1. **User Input Gathering:**
   Prompt the user to input their details including age, weight, height, gender, daily activity level, and their goal (weight loss, weight gain, or weight maintenance).

2. **Calculate Required Calories:**
   Utilize the user-provided information to calculate the required daily calorie intake based on their Basal Metabolic Rate (BMR) and activity level.
   For weight loss: Adjust the calorie intake to create a calorie deficit.
   For weight gain: Adjust the calorie intake to create a calorie surplus.
   For weight maintenance: Maintain the calculated calorie intake without any adjustment.

3. **Access Diet Plans:**
   Access an Excel file containing various diet plans categorized for different goals (e.g., weight loss, weight gain, maintenance) and dietary preferences (e.g., vegetarian, non-vegetarian).
   Ensure that each diet plan includes a variety of meals that are nutritionally balanced and meet the user's calorie requirements.

4. **Recommendation Generation:**
   Based on the user's goal and dietary preferences, filter the available diet plans to select the most suitable options.
   Consider any specific dietary restrictions or preferences provided by the user (e.g., allergies, intolerances, vegetarianism).
   Randomly select or prioritize a subset of recommended diet plans from the filtered options to provide variety and choice to the user.

5. **Display Recommendations:**
   Present the recommended diet plans to the user in a user-friendly format within the app interface.
   Include details such as meal schedules, portion sizes, and ingredient lists for each recommended diet plan.
   Provide options for the user to view additional details or modify their preferences if needed.

6. **Tracking and Feedback:**
   Implement features for users to track their progress and adherence to the recommended diet plans within the app.
   Allow users to provide feedback on the recommended diet plans, such as rating the meals or reporting any issues or difficulties encountered.
   Utilize user feedback to improve the recommendation algorithm and continuously update the available diet plans.

7. **Continued Support:**
   Offer ongoing support and guidance to users throughout their journey, including access to resources such as nutrition tips, healthy recipes, and expert advice.
   Encourage users to stay engaged with the app by providing motivational messages, reminders, and rewards for achieving milestones.

## 4. RESULTS AND CONCLUSION

FitHub, following a strategic development methodology, has demonstrated notable achievements across various facets of its functionality. Gym Machine Detection: The TensorFlow model integrated into FitHub's Flask API showcased accurate and responsive gym machine detection. User interactions with the Smart Camera feature yielded precise information on detected machines, providing insights into recommended reps, weights, and safety guidelines.

**Diet Recommendation System:** The Diet Recommendation System successfully processed user inputs, calculated personalized caloric requirements, and generated balanced diet plans. Users received clear and detailed recommendations for breakfast, lunch, and dinner, fostering a tailored approach to nutrition aligned with their fitness goals.

**Chatbot Interaction:** FitHub's Chatbot feature, powered by OpenAI, facilitated dynamic and engaging interactions. Users experienced a responsive conversational interface, enabling them to seek fitness guidance, ask queries, and receive personalized responses.

**Pedometer Integration:** The Step Counter with Pedometer integration accurately tracked users' physical activity. The Expo Sensors library, coupled with AsyncStorage for data persistence, provided users with real-time updates on their step count, enhancing their awareness of daily physical activity.

## 5.ACKNOWLEDGMENT

The preferred spelling of the word “acknowledgment” inAmericaiswithoutan “e” afterthe “g”.Avoidthestiltedexpression, “Oneofus(R.B.G.)thanks...” Instead,try“R.B.G.thanks”.Putapplicablesponsoracknowledgmentshere;DONOTplacethemonthefirst pageofyourpaperorasafootnote.

## REFERENCES

[1] Mani, Dr. Ramarao, and Samson Paul Moses. "Smart Mirror as Personal Fitness Trainer: A Review." International Research Journal of Engineering and Technology (IRJET), vol. 10, no. 5, May 2023.

[2] Naik, Swapnil, and Arpit Bhardwaj. "A Smart Health Monitoring System for the Elderly Using Internet of Things." International Research Journal of Engineering and Technology (IRJET), vol. 10, no. 5, May 2023.

[3] Fang, Xiaojiang, et al. "Research on the Optimization of Sports Training Mode under the Background of Artificial Intelligence." Sustainability, vol. 14, no. 2, 2022, pp. 690.

[4] Dignum, Virginia. "Putting AI Ethics to Work: Are the Tools Fit for Purpose?" MDPI Sustainability, vol. 14, no. 2, 2022.

[5] Ramachandran, H. "Impact of Artificial Intelligence on Individual Performance: Exploring the Fit between Task Data
and Technology." Hindawi Mathematical Problems in Engineering, vol. 2021, Article ID 2446413, 2021.

[6] Pethybridge, Ben, et al. "Towards Understanding People's Experiences of AI Computer Vision Fitness Instructor Apps." ResearchGate, 2022.

[7] Johnson, R. A. "The Effectiveness of a Personalized Virtual Fitness Trainer in Teaching Physical Education by Applying the Artificial Intelligent Algorithm." ResearchGate, 2022.

[8] Smith, P., & Doe, J. "The Impact of Artificial Intelligence on Individual Performance: Exploring the Fit between Task Data and Technology." ResearchGate, 2022.

[9] Brown, M., & White, A. "A Novel Artificial Intelligence Method for Weekly Dietary Menu Planning." ScienceDirect, 2022.