A HEALTH-FOCUSED SYSTEM THAT BLENDS MACHINE LEARNING AND FLUTTER TO ASSIST USERS IN ACCOMPLISHING THEIR DIET/FITNESS GOALS ALONG WITH PROVIDING HELPFUL ADVICE

Morgan Li[1], Yangxuezhe Sun[2] and Rayyan Zaid[2]

[1]Los Osos high school, 6001 Milliken Ave, Rancho Cucamonga, CA 91737
[2]Rancho Cucamonga High School, 7873 Chablis Pl, Rancho Cucamonga, CA 91739
[3]Computer Science Department, California State Polytechnic University, Pomona, CA 91768

ABSTRACT

We built an AI Fitness Application in hopes of addressing the problem of obesity. Obesity and lack of fitness awareness are major problems in the United States that lead to unhealthy lifestyles and health risks. To solve this problem, we developed an app that bundles together essential components to help a person keep up with their fitness. Our app includes a calorie logging system where you can log your food and exercise, a fitness tip section, and a collection of exercise videos. Our technology uses Flutter Dart as the main programming language for the User Interface. We also use Python from our AI system and Firebase for our database [11].

In our research, we tested out our AI Food Classification Model and Calorie Tracking system. After extensive testing, we realize the need to fine tune our calorie tracking system for exercises and also to expand our AI model's dataset to recognize more common foods. Some challenges we faced were:

1. *Integrating our AI Food Classification Model to the frontend*
2. *Create a Calorie Calculation Function and connect that with the frontend.*

Overall, our system provides a simple and friendly interface that anyone can pick up instantly. Our solution is not a large-scale solution to obesity, but rather it's an option that the everyday person can use quickly.

KEYWORDS

Fitness, Python (AI), Flutter(front-end), Firebase(back-end)

1. INTRODUCTION

Problem

In response to the pressing public health crisis highlighted by the latest research from the JAMA network, which reveals a concerning rise in obesity rates from 27.5% in 1999 to a startling 43.0%

David C. Wyld et al. (Eds): SEAU, NLMLT, ITCS, CST, DMS -2023
pp. 105-122, 2023. CS & IT - CSCP 2023 DOI: 10.5121/csit.2023.132008

in 2018, as well as an increase in severe obesity from 3.1% to 6.9% over the same period [1]. Our commitment to addressing this issue through a fitness app is resolute.

Side Effects of Obesity

The ramifications of being overweight are extensive and severe, as it can lead to a multitude of health problems, including cardiovascular issues, high blood pressure, elevated cholesterol levels, and an increased risk of diabetes. Given this dire situation, it is imperative to provide individual [6].

ls with a practical and effective toolbox to combat obesity and its associated health risks.

Who Does Obesity Affect in the Long Run

It affects individuals of all ages, genders, socioeconomic backgrounds and races. However, the long-term consequences are far-reaching and can affect every aspect of everyone's life and health. A reason why people are overweight is food intake, in other words, the relationship between calorie intake and expenditure. When a human individual consumes more calories through food than the body uses through physical activity and metabolic processes, the excess calories are stored as fat, causing weight gain over time [7]. It is convenient to have a tool to record the number of calories burned and consumed throughout the day [5].

Method A:

The solution aims to address complex public health issues by bringing experts and groups together under national leadership, involving the general public through various channels. However, it has proven ineffective in reducing obesity rates [12]. The limitations include not addressing the challenges of mobilizing people and the need for international collaboration. Our application proposes improvements by incorporating data analytics and conducting experiments with an individualized approach to combat obesity without relying on professionals.

Method B:

The solution aims to highlight the importance of self-monitoring, physical activity, and diet in successful weight loss maintenance. It identifies that participants who exercise daily and maintain a balanced diet are more likely to sustain weight loss. However, the study's effectiveness is limited due to survey biases. Our application seeks to improve by offering a self-improvement app that doesn't require sharing data for research.

Method C:

In this experiment, the experimental team conducted experiments on multiple experimental subjects, observing and summarizing the reactions of the experimental subjects facing the weight loss plan. However, the experimental results were not very ideal. Because the planned control of exercise and diet is not continuous, the weight loss effect of the experimental subjects is not ideal. To cope with these limitations, our application can accommodate the need for continuous tracking by continuously monitoring calorie records over a period of time. It provides a more stable, engaging, and personalized way of managing your diet [4].

We developed a health-focused app with calorie tracking and other features to combat the growing issue of obesity in the United States.

How it solves the problem:

Personalization

Our app allows customization for calorie goals based on user profiles, including parameters such as height, weight, age, and gender ensuring a tailored experience.

Calorie tracking

The Fitness app simplifies calorie tracking allowing users to log their daily food intake effortlessly.

-Nutrient insights: Users receive insights on their eating habits to help them to make more informed and healthy choices.

Effectiveness:

- Customization is pivotal to our Fitness app's approach to address the problem of growing obesity. By tailoring calorie goals we can help encourage users to embark on their weight loss/ fitness journey that is not only effective but safe.

- Data Analytics: Beyond Calorie tracking, we utilize the Machine Learning framework called Tensorflow Lite to help users identify their nutrients [13]. Users can use the "add food" log and take or submit an image in which the Ai will "somewhat" accurately identify the substance.

Advantages over other methods:

- Accessibility: With a free easy to use interface, the app is accessible to a wide audience

- Major goal: by focusing on calorie tracking and personal fitness, it directly addresses the crucial aspects of obesity.

- While many other apps consider basic user information, we take personalization to another level by taking in account the user's weight, height and age to aid the users in their fitness journey.

In Experiment 1, we assessed the Ai's proficiency in food recognition. The experiment was organized into a test group and a control group. In the test group we selected ten random popular food items from the internet and evaluated the Ai's capability to identify them. Meanwhile for the control group, we employed data from our pre-existing training dataset. Our primary discovery revealed that the Ai is still within the early developmental stages and does not demonstrate high accuracy in recognizing foods both in the test and control groups.

In Experiment 2, we focused on testing the accuracy of the in-app calorie calculation and comparing it with real-life experimental results. Our goal is to identify, summarize, and correct any differences caused by the system's universal motion intensity input. We conducted four experiments and summarized the experimental results. The findings indicate discrepancies between system predictions and actual calorie expenditure, emphasizing the need for sophisticated algorithms and comprehensive parameters (exercise intensity) to more accurately adjust results.

2. CHALLENGES

In order to build the project, a few challenges have been identified as follows.

2.1. Display the Same Tips

When the user accesses the learn screen, they are confronted with a tip screen that provides tips for exercise and diet. The current problem of the app is that it will display the exact same tips for exercise and food to users. This problem may diminish user experience with the app so a random tip generator needs to be created in order to add diversity to the tips. To address the problem, we will need to create a tip list, which will contain random tips, along with a function that will take random tips from said list. A void initState will also be used to refresh the page with new tips.

2.2. The Newly Added Calorie Data

When we add more foods, the newly added calorie data is not immediately updated on the main page. This occurs when the newest data is not immediately reflected on the progress bar after it is recorded in the backend database. To solve this problem, we might need to reload the screen every time the user enters data. We first need to create a mechanism that refreshes the home page every single time the user presses the home button. This mechanism can be achieved by Flutter's built in InitState function and using navigator's push replacement [8]. This function will be called every time the screen is loaded. This will re-initialize the data and recalculate the calories.

2.3. Directly Display a Value on the Progress Bar

Since it is impossible to directly determine the amount of calories consumed from exercise in the input entered by the user, we cannot directly display a value on the progress bar. We use user data and exercise data to calculate the approximate calories consumed during an exercise. We use factors such as height, weight, age, gender, exercise name, duration, and heartbeat.

We need to use metabolic equivalents, metabolic constants that estimate calorie expenditure during physical activity based on oxygen consumption. Each activity is assigned a MET value that represents its intensity compared to rest.

You can use the MET of an exercise to estimate calories burned per minute using the following formula:

$$\frac{MET \times 3.5 \times (body\,weight\,kg)}{200} = kcal / min$$

We need to put the user's data into the formula and get the final result.

3. SOLUTION

SignUp/Login:

The Signup screen is crucial for the project to work. Users can either sign up for a new account or login with their existing credentials. This part uses flutter and Firebase to handle user data and authentication.

The Login screen is where users provide their email and password which are validated, and they are granted access to the system.

(Flutter – Firebase)

Log Screens: (Flutter – Python – Firebase)

The main purpose of this part is to collect user data and give feedback through calculations. This part is the core of the program.

For food logging, the user has to submit the name and calories of their meal. Alternatively, the user can take a picture of the food. This picture is sent to the AI model, and the AI model returns a predicted name for the food.

Once the user confirms the information, they press submit which updates their database information.

Likewise, when users record exercise, the calories they burn is also recorded in the profile and fed back into the visual representation. Through this function, the app provides real-time feedback on the user's caloric increase or decrease.

User services: Calories In/Out. Video/Exercises Tips

The Calories In/out tracks the user's daily caloric intake and excess. Users can put in their meals which are included in daily calories. Users can also set calorie goals, in which the system will alert if they achieved their goals. The Exercise videos also provide users with exercises they can follow. Finally, there is a Exercise/Food Tips Screen that gives users helpful advice in their fitness journey.

How they connect:

Our frontend and backend are tightly connected. The main purpose of this connection is to store accurate food and exercise data for each user. Here are some instances in our application when the frontend and backend work together:

1. When the user checks their food log, it queries our database for that user's food data.

2. When the user adds a new food or exercise, it adds it to their log on firebase.

3. When the user opens the home screen, the calorie progress bars access the database to fill out the total number of calories by adding up each food and exercise instance.

4. When the user takes a picture of their food, the picture is sent to our Python Image Classification model. The model takes in the image and sends back a list of predictions to the frontend.

Overall, this seamless connection between these two parts of our application ensure that the user has a smooth experience.

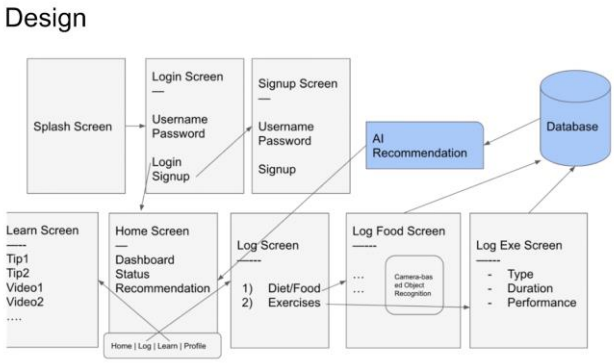


Figure 1. Overview of the solution

Frontend:

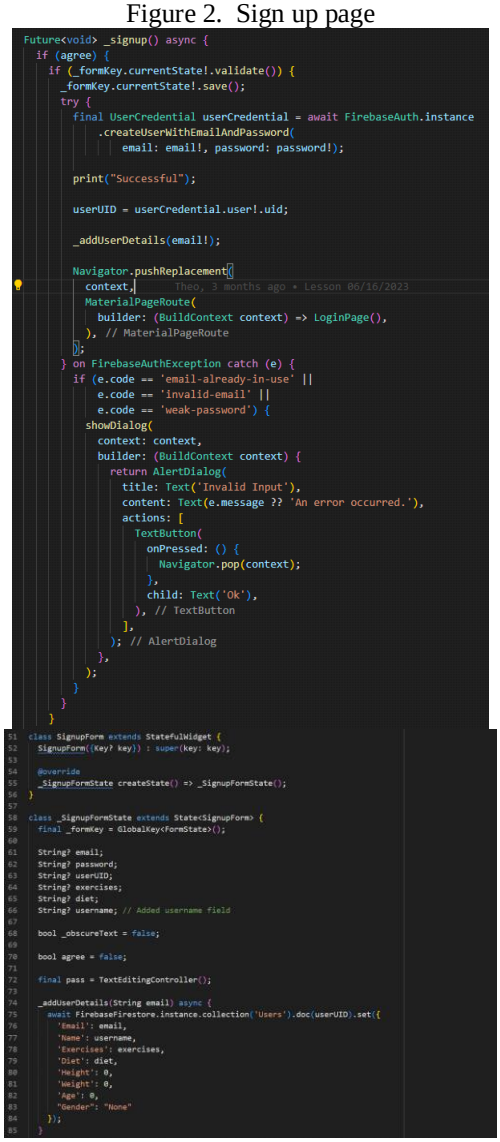
- Login Page
- Signup Page
- Home Page
- Log Page
- Learn Page
- Profile Page

Backend:

- Firebase (FirebaseAuth, Firestore)
- Tflite Classification Model

One of the key components of the system is the "Register/Login" screen. It aims to manage the verification and control of user identities. It allows users to sign up for a new account or log in with existing credentials. This component uses the Flutter language as the user interface and Firebase as the backend service to handle user data storage, authentication, and verification of email and password credentials to ensure secure access to the system.

Figure 2. Sign up page



```
Future<void> _signup() async {
  if (agree) {
    if (_formKey.currentState!.validate()) {
      _formKey.currentState!.save();
      try {
        final UserCredential userCredential = await FirebaseAuth.instance
            .createUserWithEmailAndPassword(
                email: email!, password: password!);

        print("Successful");

        userUID = userCredential.user!.uid;

        _addUserDetails(email!);

        Navigator.pushReplacement(
          context,       Theo, 3 months ago • Lesson 06/16/2023
          MaterialPageRoute(
            builder: (BuildContext context) => LoginPage(),
          ), // MaterialPageRoute
        );
      } on FirebaseAuthException catch (e) {
        if (e.code == 'email-already-in-use' ||
            e.code == 'invalid-email' ||
            e.code == 'weak-password') {
          showDialog(
            context: context,
            builder: (BuildContext context) {
              return AlertDialog(
                title: Text('Invalid Input'),
                content: Text(e.message ?? 'An error occurred.'),
                actions: [
                  TextButton(
                    onPressed: () {
                      Navigator.pop(context);
                    },
                    child: Text('Ok'),
                  ), // TextButton
                ],
              ); // AlertDialog
            },
          );
        }
      }
    }
```

```
class SignupForm extends StatefulWidget {
  SignupForm({Key? key}) : super(key: key);

  @override
  _SignupFormState createState() => _SignupFormState();
}

class _SignupFormState extends State<SignupForm> {
  final _formKey = GlobalKey<FormState>();

  String? email;
  String? password;
  String? userUID;
  String? exercises;
  String? diet;
  String? username; // Added username field

  bool _obscureText = false;

  bool agree = false;

  final pass = TextEditingController();

  _addUserDetails(String email) async {
    await FirebaseFirestore.instance.collection('Users').doc(userUID).set({
      'Email': email,
      'Name': username,
      'Exercises': exercises,
      'Diet': diet,
      'Height': 0,
      'Weight': 0,
      'Age': 0,
      "Gender": "None"
    });
  }
```

Figure 3. Screenshot of code 1

Authentication

This code is used for user registration and login, and runs when the user enters information and presses the submit button. It is associated with the _SignupFormState class, which manages the state of the signup form. It begins by validating the _Formkey by checking if the emails are already in use, or if the password is too weak. If the _Formkey is valid the system will save the state and will save the credentials variables. Then an userCredential object is created by the class UserCredential. This object will access the database on firebase, along with trying to create a new user by the CreateUserWithEmailAndPassword function. The _addUserDetails method interacts with Firebase Firestore. It will add the user's input information to the "Users" collection, such as email, username, and some initial values.

The _signup method handles the registration process. It tries to create the user through Firebase Authentication, and if successful it retrieves the user's UID and adds the user's details into the personal database located in Firestore and navigates to the login page. This code snippet controls

user registration, validating form data, creating a Firebase Authentication user account, and storing additional user details in Firebase Firestore (personal database) upon successful registration.

Logging

As the core function of the program, the logging component focuses on collecting user data and providing feedback through calculations. It plays a key role in enabling users to track their food consumption and exercise activities. In summary, this component helps users track their caloric intake and expenditure, and leverages artificial intelligence for food recognition. This logging feature provides real-time feedback on the user's fitness progress. It relies on Firebase's Firestore Database to store and manage user data, making it an essential and foundational component of AI Fitness Tracker.

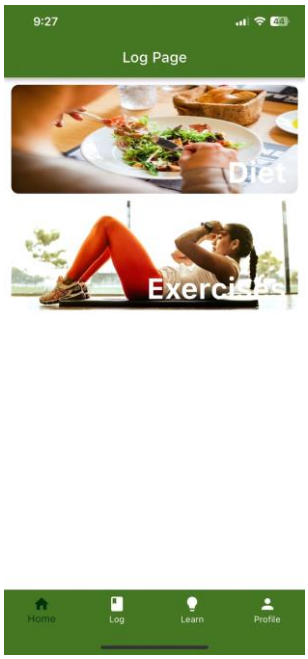


Figure 4. Log page

```
double getBMRForExerciseDuration(double weightI , double heightInInches
    int age, String gender, double exerciseDurationInMinutes) {
  double weightInKg = poundsToKilograms(weightInLbs);
  double heightInCm = inchesToCentimeters(heightInInches);

  double bmr;

  if (gender == "Male") {
    bmr = 88.362 + (weightInKg * 13.397) + (4.799 * heightInCm) - (5.677 *
  } else {
    bmr = 447.593 + (weightInKg * 9.247) + (3.098 * heightInCm) - (4.330 *
  }

  double BMRForExerciseDuration = (exerciseDurationInMinutes / 14400.0) *
  return BMRForExerciseDuration;
}

double getActiveCalories(
    double heartRate, double weightInLbs, int age, double time) {
  double weightInKg = poundsToKilograms(weightInLbs);
  double calories =
      (((heartRate - 80) * 1.0) / ((220 - age) * 1.0)) * weightInKg;
  double totalCalories = calories * time;
  return totalCalories;
}

double exerciseToCalories(double heightInInches, double weightInLbs,
    double minutesExercised, double bpm, int age, String gender) {
  double calories;

  calories = getBMRForExerciseDuration(
          weightInLbs, heightInInches, age, gender, minutesExercised) +
      getActiveCalories(bpm, weightInLbs, age, minutesExercised);
```

```
saveFood() {
  FocusScopeNode currentFocus = FocusScope.of(context);

  if (!currentFocus.hasPrimaryFocus) {
    currentFocus.unfocus();
  }

  String foodName;

  if (isPredictionCorrect) {
    foodName = _recognitions![0]['label'];
  } else {
    foodName = foodNameController.text;
  }
  Map food = {
    'date': DateTime.now().toString().substring(0, 16),
    'food': foodName,
    'calories': int.parse(caloriesController.text),
  };

  buildLoading();
  addFoodLog(food).then((value) {
    Navigator.of(context).pop();
    snapBarBuilder('Food log was added');
  });
}

buildLoading() {
  return showDialog(
    context: context,
    barrierDismissible: false,
    builder: (BuildContext context) {
      return const Center(
        child: CircularProgressIndicator(
          valueColor: AlwaysStoppedAnimation<Color>(Colors.blue),
        ), // CircularProgressIndicator
      ); // Center
    },
```

Figure 5. Screenshot of code 2

Exercise Functions Explanation

First, we will talk about how our exercise tracking system works.

The purpose of the code in the first part is to calculate users' caloric expenditure during exercise using input on various input parameters. These different inputs can be seen in the “exerciseToCalories()” function.

The `poundsToKilograms` and the `inchesToCentimeters` functions convert weight from pounds to kilograms and height from inches to centimeters.

The `getBMRForExerciseDuration` function calculates Basal Metabolic Rate (BMR) adjusted for exercise duration. It estimates the BMR for both males and females based on the Harris-Benedict equation, taking as inputs weight, height, age, gender, and duration of exercise.

The getActiveCalories() function uses heart rate, weight, age and exercise time as inputs to calculate calories burned during exercise. It estimates the calories burned above the resting BMR. The exerciseToCalories() function combines the BMR for exercise duration and active calories to estimate the total calories burned during exercise. It takes height, weight, exercise duration, heart rate (bpm), age, and gender as inputs.

Calorie Functions Explanation

Now we will talk about how our food tracking system works with the Machine Learning Model. There are also different methods that are used in the program: "saveFood()". This method is called when the user wants to save a food log entry. It collects data like the food name, date, and calories, then adds this data to Firestore database for users using the addFoodLog() function. It also displays a loading dialog while the operation is being processed, and displays a snack bar notification when it is finished. The getImageFromCamera() and getImageFromGallery(): These functions allow the user to select an image either from the camera or the gallery and set _image accordingly. The main part of our food Machine Learning system is the classifyObject() function. This method uses TensorFlow Lite to run a model on the selected image and returns a

list of recognitions, which include labels and confidence scores. It sets _recognitions with the results. It interacts with a local database through the addFoodLog() function to save food log entries.

The User interface, another core function of the program, focuses on giving the user visual details which help them understand their health information. We use flutter functions and progress bars, along with text widgets to help users monitor their diets. There are basic calculations made in order to fill the progress bars to certain extents. This component is crucial to the user's visual experience within the app [9].

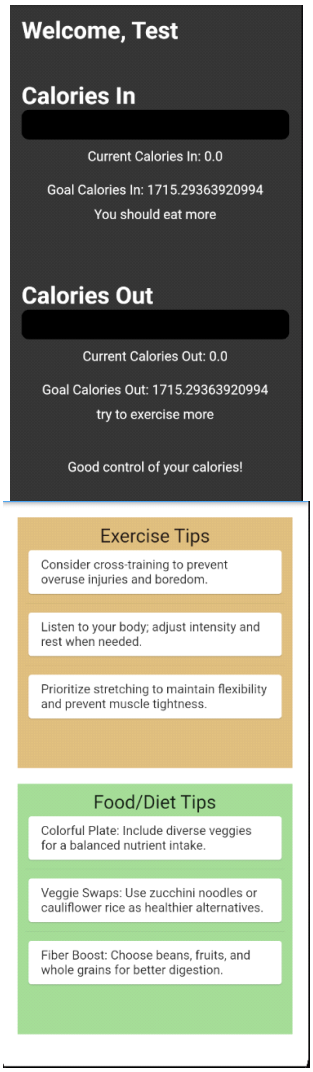


Figure 6. Test tips page

```dart
import 'package:flutter/material.dart';

class CustomProgressBar extends StatelessWidget {
  final double currentLevel;
  final double goal;

  CustomProgressBar({required this.currentLevel, required this.goal});

  double calculateProgress(double currentLevel, double goal) {
    double progressPercentage = currentLevel / goal;
    return progressPercentage.clamp(
        0.0, 1.0); // Clamp the value between 0.0 and 1.0
  }

  @override
  Widget build(BuildContext context) {
    double progressPercentage = calculateProgress(currentLevel, goal);

    return Container(
      height: 40.0,
      width: goal.toDouble(), // Set a fixed width for the progress bar
      decoration: BoxDecoration(
        color: Colors.black,
        borderRadius: BorderRadius.circular(10.0),
        border: Border.all(color: Colors.black, width: 2.0),
      ), // BoxDecoration
      child: FractionallySizedBox(
        widthFactor: progressPercentage,
        alignment: Alignment.centerLeft,
        child: Container(
          decoration: BoxDecoration(
            color: Colors.red,
            borderRadius: BorderRadius.circular(10.0),
          ), // BoxDecoration
        ), // Container
      ), // FractionallySizedBox
    ); // Container
  }
}
Map<String, List<String>> generateExerciseTipMap() {
  List<int> randomIndices = generate3UniqueRandomNumbers(exerciseList.length);
  List<String> randomExerciseTips =
      randomIndices.map((index) => exerciseList[index]).toList();
  return {'exercise': randomExerciseTips};
}

// Function to generate a map with 3 random food tips
Map<String, List<String>> generateFoodTipMap() {
  List<int> randomIndices = generate3UniqueRandomNumbers(foodList.length);
  List<String> randomFoodTips =
      randomIndices.map((index) => foodList[index]).toList();
  return {'food': randomFoodTips};
}

List<String> randomExerciseTips = [];
List<String> randomFoodTips = [];

@override
void initState() {
  super.initState();
  // Generate the random exercise and food tips when the widget initializes.
  randomExerciseTips = generateExerciseTipMap()['exercise']!;
  randomFoodTips = generateFoodTipMap()['food']!;
}
```

Figure 7. Screenshot of code 3

Progress bar explanation:

First we will talk about how our progress bars work for Calories gained / burned

The purpose of this code is to calculate how much of the progress bar would need to fill compared to the total calorie out goal. This is shown in the ‘calculateProgress' function.
It sets the “fractionally sizedbox” to the progressPercentage which is calculated by the “CalculateProgress” function.

The ‘CalculateProgress’ function uses variables stored in Firestore Database from the exerciseToCalories function. The exerciseToCalories function iterates over the exercises performed on the current day and sums up the calories. It does the same thing for the foods eaten.
Random tip generation explanation:

Next, we talk about the tip generation that uses a random number generator.

This function generates a map with a key 'exercise' and a list of three random exercise tips. It first calls ‘generate3UniqueRandomNumbers’ with the length of the exercise List -1 to get three unique random indices. It then uses these indices to select three exercise tips from the exercise List. The selected tips are stored in the random Exercise Tips list. The function returns a map with the 'exercise' key and the list of selected exercise tips.

The food tips are the exact same as the exercise tips with the names of the functions changed to food instead of exercise.

The method is called in the 'initState' which will run every time the page is refreshed or reopened. This ensures different tips are shown to the user every time the tip screen is opened.

4. EXPERIMENT

4.1. Experiment 1

Accuracy of AI Model

1. On varied foods (closest to user experience)

2. Foods from our model (model accuracy)

In Experiment A, we will evaluate the accuracy of the artificial intelligence model. Due to the limitations of the machine learning training, the model is likely to make errors when faced with different types of food. Because of this, it is crucial to further improve the accuracy of AI models. The increased accuracy makes it easier for users to record food samples without errors, thus enhancing the user experience.

On random foods

Random Foods: A dataset where we randomly select different sets of foods from the real world to represent situations that users may encounter in their daily lives. The foods will be chosen to mimic the unpredictability and randomness of user input. By conducting multiple experiments, we will summarize and record the AI predictions for each food product.

For random foods, we follow these steps:

1. Arbitrarily select 10 different foods from this website. Google search and save 2 pictures of each food. Website: https://www.rd.com/article/america-favorite-food/

2. Input each picture into the model and tally how many the model predicts correctly.

3. Calculate the percentage by dividing # correct / 20.

Food in our model: This set of data will be used to test the accuracy of the AI model on one or more sets of food images on which the model was trained and recorded during development. The purpose of this data is to evaluate the impact of AI recognition model performance on foods it has already seen and identified [2].

For food in our model, we will follow these steps:

1. Arbitrarily select 10 different foods from our Kaggle dataset. Google search and save 2 pictures of each food.

2. Input each picture into the model and tally how many the model predicts correctly.

3. Calculate the percentage by dividing # correct / 20.

On model foods

1.donuts 2/2
2.chicken_curry1/2
3.pho1/2
4.sushi 2/2
5.french_toast 1/2
6.chocolate_cake 2/2
7.apple_pie 1/2
8.sashimi 1/2
9.hot_dog 1/2
10.crab_cakes 0/2
12/20 = 60%

Random foods

1. Chili : 0%
2. Chicken tenders : 0%
3. Thai iced tea : 0%
4. Tacos : 0%
5. Doughnuts : 0%
6. Crab rangoon : 0%
7. Chicken sandwich : 50%
8. Macaroni and cheese : 0%
9. Churros : 100%
10. Cheeseburger : 100%

5/20 = 25%

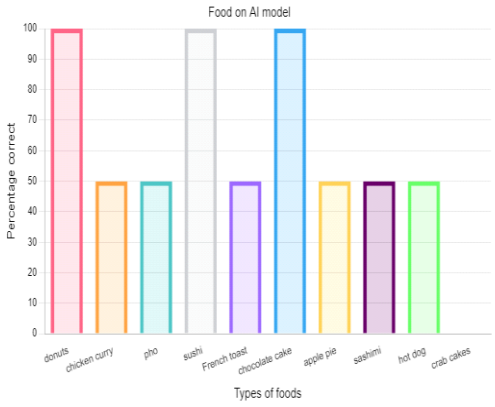


Figure 8. Food on AI model

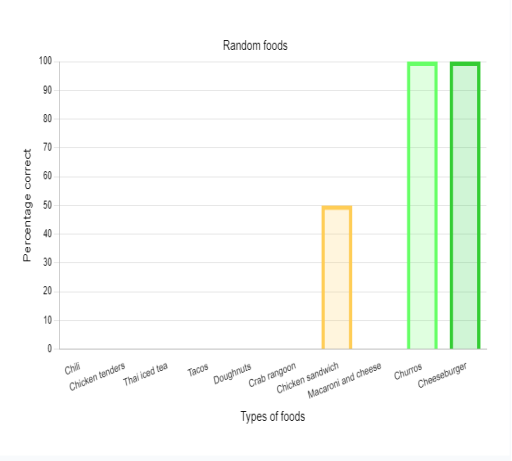


Figure 9. Random foods

AI model result analysis:
Mean: 60%
Median: 50%
Lowest Value: 0%
Highest Value: 100%
Random foods result analysis:
Mean : 25%
Median : 0%
Lowest value: 0%
Highest value: 100%

The Data did not surprise us as our ai model is still in the stages of infancy. The biggest effect on the results is the previous training on the food items.

4.2. Experiment 2

Accuracy of Calories burned calculations:

1. On real time experiments (Closest to user experience)

2. Calculations in the app (system accuracy)

In Experiment 2, we will evaluate the accuracy of calorie calculations used within the system and compare it to the actual data presented by real world experiments. Due to the limitations of variables included in the system's equation, the system is generalizing the data of exercises when they are inputted. With this issue, we will need to improve the accuracy of the equations making it closer to experimental values.

On Experimental Data:

Real life experiments steps:

1. Record weights of participants before running 30 minutes at 90 BPM

2. Make sure people do not drink water or use the restrooms before / during / after the experiment until their values are recorded before and after the experiment.

3. Calculate the data by reweighing the participants after the run, divide the amount of lbs by half so both fat and muscle loss are evaluated and multiplying them by their respective calories per pound. (3500 calories per 1 lb of fat, 800 calories per 1 lb of muscle) [10].

Systems Calculation steps:

1. Input the exercise in the exercise log page. (Exercise: running, Bpm 90+, Duration: 30 mins)

2. Now, you have the actual calories (from real life) and the predicted calories (from the algorithm).

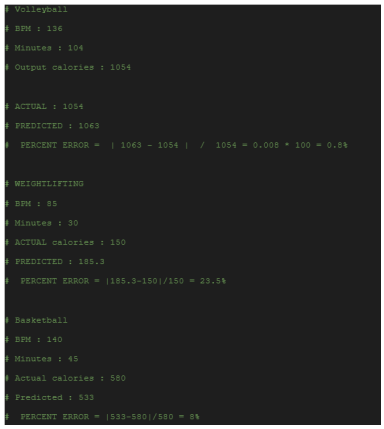


Figure 10. Figure of experiment 2

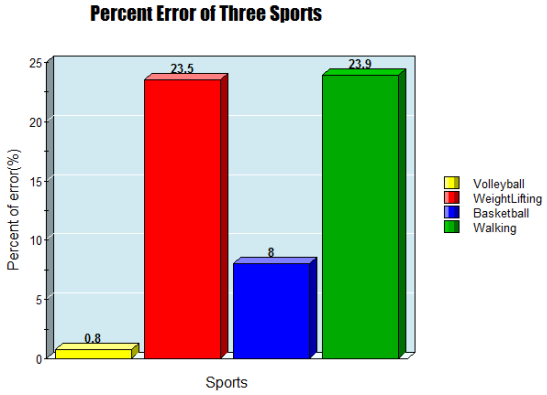


Figure 11. Percent error of three sports

Based on Percent Error
Mean: 14.05
Median: 15.75
Lowest value: 0.8
Highest value: 23.9

5. RELATED WORK

Solution Summary:

The solution involves bringing together experts and various groups under national leadership to address complex public health issues effectively. This collaborative approach aims to create lasting positive changes in society [3].

Explanation of How the Solution Works:

This approach involves engaging with the general public through diverse channels, including media outlets, medical consultations, and personal interactions.

Effectiveness Assessment:

Unfortunately, this solution has proven to be highly ineffective in curbing obesity rates, as evidenced by the threefold increase in obesity rates in the United States throughout the 20th century.

Limitations:

Both sources acknowledge the complexity of the obesity problem but fail to address the specific challenges associated with mobilizing people to combat the growing obesity issue. Additionally, the global nature of the problem, which requires international collaboration, is overlooked.

Proposed Improvements:

To enhance effectiveness, we propose incorporating more data analytics and calculations within the context of the obesity predicament. Additionally, we plan to conduct experiments to gather evidence on the impact of our app in reducing obesity. Our app will focus on individualized approaches, eliminating the need to recruit professionals to address the obesity problem [3].

Solution summary: self monitoring, physical activity and diet are important factors in successful weight loss maintenance.

Explain how the solution works: During the testing phase of the experiment, participants were categorized into two groups based on their ability to keep weight loss. The participants who were able to keep their weight after their weight loss programs tended to exercise daily (30 mins). They were also less likely to consume over the top diets.

Effectiveness assessment: This study may not be effective as it seems, since the main source of data collection is through the usage of surveys. These surveys often contain biases such as sampling issues, conformity or acquiescence bias.

Limitations: the effectiveness on physical activity and dieting on maintaining weight loss depend both on the mental strength of the participants and their ability to tell the truth within the surveys.

Proposed improvements: Our app improves on the basis of self improvement without the fear of giving away data for research projects.

In this study, people attempted to curb obesity by implementing a lifestyle-changing weight loss program. Such programs focus on treating cognitive impairment through a low-calorie diet, exercise program, and therapy. Over the course of the program, participants undergo intensive daily monitoring and participate in high-intensity physical activity. This allows them to achieve their weight loss goals in a very short period of time and immediately observe a weight loss of over 12% [4].

After summarizing the data, it was found that only a small proportion of subjects lost 10% or more of their weight within 2 to 4 years. Only 10% of participants achieved significant results in the first two years, but these effects diminished over time. This program is experimentally limited by its almost complete reliance on a specific diet and physical activity regime, and it cannot provide a comprehensive assessment of long-term weight stability, which may be influenced by an individual's internal and external circumstances. Our app can minimize the negative impact of this effect on weight loss by continuously monitoring calorie intake over time.

6. CONCLUSIONS

1. AI not accurate
    a. Get more pictures of each food and retrain the model in python
2. AI isn’t relevant
    a. Add more common foods to our dataset
3. App login needs to have error feedback
    a. Firebase already gives us feedback (password is wrong, email doesn’t exist). All we need to do is take the Firebase feedback and display it on the screen
4. Create a more accurate calorie calculation system
    a. Using the research paper (MET) and coming up with a better formula.

In summary, one of the main issues with our project is the accuracy of our AI model, which currently remains inaccurate, especially when dealing with certain foods. By using an expanded dataset and retraining the AI, we can improve its accuracy as much as possible. Another limitation is the relevance of our artificial intelligence. Our AI database does not represent the typical foods eaten [14]. To make it more applicable to users' daily lives, we need to incorporate a wider range of common foods into our dataset, aligning the AI with users' dietary choices and preferences. We can solve this issue by investing more time to obtain larger and more diverse food image datasets.

In addition, the login process of our application lacks user-friendly error feedback, and our frontend needs more humane feedback and prompt effects. In order to enhance the user experience, we can pop up a prompt every time an incorrect password is entered so that users can quickly solve and clarify the login problem. Also, this is an easy aspect to fix because Firebase automatically gives us the feedback [15]. All we need to do is implement it in the frontend.

Finally, for exercise intensity measurement, the current calculation formula is inaccurate. This is because the numerical value representing exercise intensity is a constant. To solve this problem, we should achieve more accurate intensity measurement by leveraging heart rate data and developing a formula that targets the corresponding intensity level based on the user's exercise and heart rate. It will accurately calculate calorie consumption based on user activity and intensity level (adjusted). This improvement will increase the accuracy and usefulness of our calorie counting system.

We designed this project to help the average person keep track of their health and diet. While the app provides a simple interface, we discovered that it may have some limitations. These include: artificial intelligence accuracy and relevance, inconvenient logins, and inaccurate calorie counting. Addressing these issues by improving accuracy will significantly improve the controllability, usability, and appeal of our applications to users.

REFERENCES

[1] Moy, Karen, et al. "Metabolic equivalent (MET) intensities of culturally-specific physical activities performed by New Zealanders." NZ Med J 119 (2006): 6.

[2] Page, David. Food Americana: The Remarkable People and Incredible Stories behind America's Favorite Dishes. Mango Media Inc., 2021.

[3] Kruger, Judy, Heidi Michels Blanck, and Cathleen Gillespie. "Dietary and physical activity behaviors among adults successful at weight loss maintenance." International Journal of Behavioral Nutrition and Physical Activity 3.1 (2006): 1-10.

[4] Christiansen, Tore, et al. "Weight loss maintenance in severely obese adults after an intensive lifestyle intervention: 2-to 4-year follow-up." Obesity 15.2 (2007): 413-420.

[5] Ogden, Cynthia L., et al. "Trends in obesity prevalence by race and Hispanic origin—1999-2000 to 2017-2018." Jama 324.12 (2020): 1208-1210.

[6] GBD 2015 Obesity Collaborators. "Health effects of overweight and obesity in 195 countries over 25 years." New England journal of medicine 377.1 (2017): 13-27.

[7] Swinburn, B. A., and E. Ravussin. "3 Energy and macronutrient metabolism." Baillière's clinical endocrinology and metabolism 8.3 (1994): 527-548.

[8] Palumbo, Daniele. The Flutter framework: Analysis in a mobile enterprise environment. Diss. Thesis, Turin, IT: Politecnico di Torino, accessed May 17, 2022, https://webthesis. biblio. polito. it/19111/1/tesi. pdf, 2021.

[9] Ortega, Francisco B., et al. "Physical fitness in childhood and adolescence: a powerful marker of health." International journal of obesity 32.1 (2008): 1-11.

[10] Katan, Martijn B., and David S. Ludwig. "Extra calories cause weight gain—but how much?." Jama 303.1 (2010): 65-66.

[11] Van de Poel, Ibo. "Embedding values in artificial intelligence (AI) systems." Minds and Machines 30.3 (2020): 385-409.

[12] Sturm, Roland, and Aiko Hattori. "Morbid obesity rates continue to rise rapidly in the United States." International journal of obesity 37.6 (2013): 889-891.

[13] Elgendy, Nada, and Ahmed Elragal. "Big data analytics: a literature review paper." Advances in Data Mining. Applications and Theoretical Aspects: 14th Industrial Conference, ICDM 2014, St. Petersburg, Russia, July 16-20, 2014. Proceedings 14. Springer International Publishing, 2014.

[14] Zhou, Xuanhe, et al. "Database meets artificial intelligence: A survey." IEEE Transactions on Knowledge and Data Engineering 34.3 (2020): 1096-1116.

[15] Khawas, Chunnu, and Pritam Shah. "Application of firebase in android app development-a study." International Journal of Computer Applications 179.46 (2018): 49-53.