DEVELOPMENT AND EVALUATION OF AN AI-ENABLED NUTRIENT INTAKE MONITORING APP FOR OBESITY AND DIABETES PREVENTION IN YOUNG PEOPLE: A COMPARATIVE STUDY WITH LIVE SCANNING AND PHOTO UPLOADING METHODS

Jiheng Yuan[1], Victor Phan[2]

[1]Santa Margarita Catholic High School, 22062 Antonio Pkwy, Rancho Santa Margarita, CA 92688
[2]Computer Science Department, California State Polytechnic University, Pomona, CA 91768

ABSTRACT

Obesity and diabetes are prevalent health issues worldwide, especially among young people. To address this, an app was proposed to help users monitor their daily nutrient intake and prevent obesity and diabetes [1]. The app uses AI scanning to analyze the nutrient level of food and suggests a suitable daily nutrient intake for the user based on their age and gender. Data storage allows users to track their meal history and create a personalized diet plan [2]. The app was compared to similar systems, and it was found that live scanning is more intuitive and convenient than photo uploading. Additionally, the proposed app was tested in two experiments and was found to be effective in identifying food items and received generally positive feedback from users, but further improvements are necessary to enhance accuracy and user experience. In the first experiment, the accuracy of the AI model for predicting food items was tested using a combination of existing and customized datasets [3]. A total of 227 food items were tested, including bananas, watermelons, peaches, tomatoes, pineapples, rice, fries, hamburgers, eggs, noodles, and other items. The results showed an overall accuracy rate of 82% for all food items tested, with pineapple having the highest accuracy at 100% and peaches having the lowest accuracy at 60%. In the second experiment, 15 participants tested the application's features and provided feedback through a survey. The results showed that the application was successful in its implementation of features and received generally positive feedback, with an average functionality rating of 8.13 and an average convenience rating of 7.67.

KEYWORDS

Obesity, Diabetes, Nutrient Intake Monitoring, AI Scanning

1. INTRODUCTION

Obesity and Diabetes are increasingly common nowadays in the world. Obesity and Diabetes correlate with each other [4]. There are about 250 million people that are suffering from Obesity

David C. Wyld et al. (Eds): SIPM, ITCA, CMIT, FCST, CoNeCo, SAIM, ICITE, ACSIT, SNLP -2023
pp. 169-178, 2023. CS & IT - CSCP 2023 DOI: 10.5121/csit.2023.130918

nowadays and the number will continue to increase to 300 million by the end of 2025. Diabetes is usually the consequence that comes with obesity [5]. In China, there are nearly 90 million people that are suffering from diabetes and 1.3 million people died in just the year of 2011. It is extremely important to have a way for people to take care of their daily consumption to minimize the chances of obesity and also diabetes. Obesity and diabetes are getting more common in young people that have more access to food that are high in calories and sodium compared to that of the past. Both of them are potentially preventable through lifestyle modification such as monitoring the nutrient level that is consumed to have a healthy daily intake through a scientific way.

My solution to this problem is to design an app that can help people to manage and monitor their daily consumption to help people to avoid obesity and diabetes in a scientific way [6]. With that idea, we would use the work of AI to scan the food to provide information for users to know how much they are consuming in nutrient level when they are consuming the certain food. By establishing limits that fit each individual, people get to know when and what they are consuming too much and need to stop in a way to help them manage their health.

The study conducted two experiments to evaluate the effectiveness of an AI-powered nutrition app that recommends healthy food choices. The first experiment aimed to test the accuracy of the AI model for predicting food items using a combination of existing and customized datasets. A total of 227 food items were tested, including bananas, watermelons, peaches, tomatoes, pineapples, rice, fries, hamburgers, eggs, noodles, and other items. The results showed an overall accuracy rate of 82% for all food items tested, with pineapple having the highest accuracy at 100% and peaches having the lowest accuracy at 60%. A confusion matrix was generated to analyze the results, and the data were presented in graphs.

The second experiment involved 15 participants who tested the application's features and provided feedback through a survey. The results showed that the application was successful in its implementation of features and received generally positive feedback, with an average functionality rating of 8.13 and an average convenience rating of 7.67. However, some participants reported issues with the events page and suggested improvements to the interface's visual appeal. Overall, the study demonstrated the potential effectiveness of using AI technology in nutrition apps to help users make healthy food choices, but further improvements are needed to enhance the app's accuracy and user experience.

The paper by Ming et al. proposes a dietary tracking system that uses deep-based image recognition algorithms to analyze nutrition based on food photos [7]. While this solution is effective in streamlining the process of managing dietary intake, it still requires users to take time to upload photos, which can be inconvenient. In contrast, my project uses live camera scanning to detect food from any angle, providing a more intuitive and convenient way for users to track their nutrition.

Similarly, Hassannejad et al. propose a method that uses image analysis and wearable sensors to extract information about food content and detect eating behaviors. However, the need to take individual photos and wear sensors can be burdensome for users. In my project, I address this limitation by enabling food detection through live camera scanning.

Agapito et al. present a recommender system for nutrition content delivery, which lacks a convenient way for users to record their nutrition content [8]. In contrast, my solution incorporates a data storage system that is linked to an AI scanner, allowing users to directly transfer and store scanned data. This not only streamlines the process of recording nutrition content but also provides users with a more efficient way to track their dietary intake for future reference.

2. CHALLENGES

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

2.1. How the AI works

The first challenge is how the AI works in order to actually detect the food accurately and in an efficient way. First, the main problem of AI is detecting the object accurately. From that, I can manage to use an AI training system to help the AI to recognize the same object in various ways, which helps to improve the accuracy over time.

2.2. The UI part

The second challenge is the UI part if that is accessible for everyone to use. It would be useless if the app is hard for people to use and they cannot easily figure out how to use it properly. From that, I could see the use of simple designs in the app. In a simple design, people get to direct themselves more easily to the function that they wanted to use. From that, the problem could be saved by using a much simpler design to help people access the app more easily.

2.3. The data storage

The third challenge is the data storage that keeps track of and stores data of how much people consume daily and also looking back from the future. It is extremely important for people to acknowledge what they ate in the past whether to keep track of their nutrient intake or to tell the doctor what they have eaten for the past weeks. From that, I could see the use of a history tap for people to store and keep track of what they have eaten and how much they have consumed. I could also do a bar graph to show a significant status for users to have a clear look of how much nutrient level they consumed in the past which helps them better manage their consumption in the future.

3. SOLUTION

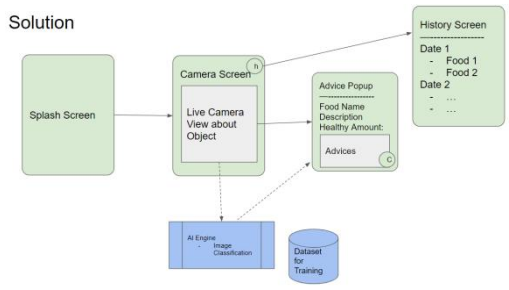

Figure 1. Overview of the solution

This is an app made to track the nutrient intake users have from the food they consume [9]. This app consists of three major components: AI scanning, nutrition suggestions, and data storage. The scanning part of this app is to receive information about the nutrient level of the given food by scanning the food that is in front of the camera to receive data immediately. With intensive times of training, our high-tech AI has the ability to receive information the second that it scans the food. After that, users get to choose whether to eat this food, depending on the food's nutrition level. If they want to consume it, the users can hit the check mark to collect the food under their daily meal consumption history, which will be stored directly at the data storage center. With the

data storage feature, users can track every meal they have eaten for the past years to design a specific diet plan that fits them best for them. Not only that, but our app also offers nutrient level suggestions depending on the gender and age of the user and gives a suggested amount of nutrition intake to help the user to keep a healthy life [10].

The first component's purpose is to establish live scanning for users to recognize the nutrition level of the object that appears in front of the camera. This whole scanning process relies on the trained AI that was previously trained with multiple pictures of different objects to have the most accurate possible. With this, the scanning feature can establish live scanning while the camera is moving and still providing information as it detects different objects.

Figure 2. The scanning feature

```
loadTfLiteModel() async {
  String res;
  res = await Tflite.loadModel(
      model: "assets/model_unquant.tflite", labels: "assets/labels_unquant.txt");
  print(res);
}

setRecognitions(outputs) {
  print(outputs);

  if (mounted) {
    setState(() {
      output = outputs;
    });
  }
}

FoodData getCorrespondingFoodInfo(String l) {
  switch(l) {
    case "0 Banana":
      return foodDictionary["Banana"];
    case "1 Watermelon":
      return foodDictionary["Watermelon"];
    case "2 Peach":
      return foodDictionary["Peach"];
    case "3 Tomato":
      return foodDictionary["Tomato"];
    case "4 Pineapple":
      return foodDictionary["Pineapple"];
    case "5 Rice":
      return foodDictionary["Rice"];
    case "6 Fries":
      return foodDictionary["Fries"];
    case "7 Hamburger":
      return foodDictionary["Hamburger"];
    case "8 Egg":
      return foodDictionary["Egg"];
    case "9 Noodle":
      return foodDictionary["Noodle"];
```

Figure 3. Screenshot of code 1

When the app transitions to the scanning page, it starts a camera stream. This camera stream updates on a fixed timestep and every time returns a frame of the user's phone's back-facing camera. In addition, the scanning page also loads a TensorFlow Lite model on startup. Whenever the camera stream returns a rendered frame from the phone's camera, the output frame is passed through the TensorFlow Lite model. Then, the model will output a list of objects that it can recognize. This output list is passed into a callback method named setRecognitions that will set the output equal to a list variable with a wider scope, such that the rest of the program has access to the data. The outputs from the Tensorflow Lite model are only string labels, so we have another method named getCorrespondingFoodInfo that will take this label and return an object of type FoodInfo using a static lookup dictionary. A FoodInfo object contains nutritional

information and a label. getCorrespondingFoodInfo will match up the output to the FoodInfo with the same label.

The purpose of the nutrition suggestions component is to provide recommended nutrition levels for users to have a scale of how much they are supposed to eat to keep a healthy balanced life. Once the user provides their body data, such as height, age, and gender, the app will immediately provide the possible solution for users to take into consideration.

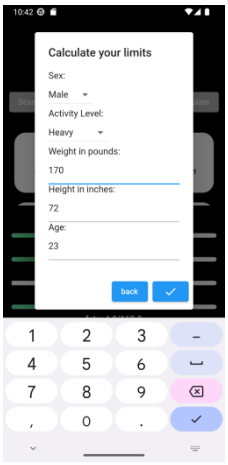

Figure 4. The calculate page

Figure 5. Screenshot of code 2

First, the program has to get the user's basal metabolic rate, commonly abbreviated to BMR. The parameters for the BMR formula are the user's weight in pounds, their height in inches, their age in years, and the user's sex. The BMR formula varies depending on the user's sex. For all these parameters, the page itself contains several input form widgets to record this information. We also need to calculate their total energy expenditure, abbreviated as TEE. To do this, we have a dropdown input widget on the page that asks them to rate how active they are from 1 to 5. getTee will return a multiplier based on their rating. We multiply the BMR by the TEE to get the user's recommended caloric intake. To extrapolate this to other nutritional information, we calculate a ratio value by dividing their recommended caloric intake by the CDC's average daily caloric intake of 2000. We then multiply the CDC's other average values by this ratio to get the recommended info for the user.

The purpose of the data storage component is to provide a place to store user's daily consumption on an all year scale. It can also provide a clear bar graph to indicate what amount of nutrition you have consumed on a certain day and also showing what you have consumed excessively or too less than a scale that you set before.

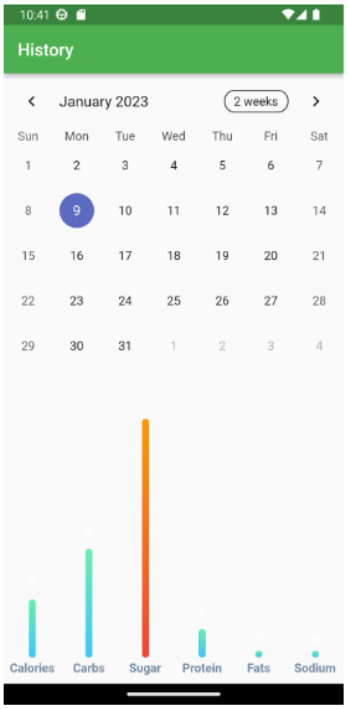

Figure 6. Screenshot of calender

```
List<String> foods = List.from(entries[dayString]);
print("FOODS: " + foods.toString());

calories = 0;
protein = 0;
fats = 0;
carbs = 0;
sugar = 0;
sodium = 0;
for (String s in foods)
  {
    FoodData f = foodDictionary[s];
    setState(() {
      calories+=f.calories;
      protein+=f.protein;
      fats+=f.fats;
      carbs+=f.carbs;
      sugar+=f.sugar;
      sodium+=f.sodium;
    });
  }
print("$calories $protein $fats $carbs $sugar $sodium");

selectedDayWithFood = isDayInEntries(dayString);

if (selectedDayWithFood) {
  List<String> warnings = prefs.getStringList(dayString + " WARNINGS");
  if (warnings != null && warnings.isNotEmpty) {
    warnings.forEach((warning) {
      setState(() {
        if (warning.contains("exceeded"))
        {
          if (warning.contains("calories"))
            exceedCalories = true;
          if (warning.contains("carbs"))
            exceedCarbs = true;
          if (warning.contains("sugar"))
            exceedSugar = true;
          if (warning.contains("protein"))
            exceedProtein = true;
            if (warning.contains("fats"))
              exceedFats = true;
            if (warning.contains("sodium"))
              exceedSodium = true;
        }
      });
    });
  }
}
```

Figure 7. Screenshot of code 3

The history page has two sections -- a calendar with a day selector, and a histogram. The history page before loading has to grab the user's records in a variable called entries. Every food a user

has eaten in a meal is saved in a list. A given day may have multiple of these lists. When the user selects a day in the calendar, the program collates all the meal lists for that day into one larger list. Then, for every food name in that list, it looks up its corresponding nutritional info and records a running total of each nutrition information. Then, the program searches its local database for a warning list for that particular day [14]. When the user saves a meal, a warning list is generated which contains a warning for each nutritional info that the user has exceeded that day. Once it loads this warning list, if it exists, it will note which nutritional info the warnings correspond to. The bars for these specific info will be red in the histogram to indicate overconsumption.

4. EXPERIMENT

4.1. Experiment 1

One potential drawback of the app is the potential lack of accuracy in the AI's readings and predictions. However, running experiments with the AI can help determine its level of accuracy and identify areas that require improvement. To test our model, we combined an existing dataset with a customized dataset, resulting in a total of 227 food items [15]. The dataset included a range of food items, such as bananas, watermelons, peaches, tomatoes, pineapples, rice, fries, hamburgers, eggs, noodles, and other items. We utilized a Python script to read and process the data, analyze the results, and generate a confusion matrix to visualize the outcome.

The results are presented in two graphs shown in the Figure below. The first graph (Graph 1) illustrates the amount of food tested and the corresponding number of accurate predictions made by the AI. The blue columns represent the total amount of each type of food tested, while the orange columns represent the number of accurate predictions made by the AI. The second graph (Graph 2) shows the accuracy of the AI for each food type. The accuracy for each food type was calculated by dividing the number of correct predictions by the total amount tested for each food type. The accuracy for peaches was the lowest at 60.0%, while pineapple had the highest accuracy at 100.0%.

The average accuracy of the AI for all food types tested was 82.0%. However, there was no significant relationship between the accuracy of the AI and the amount of data tested for each food type out of 125.

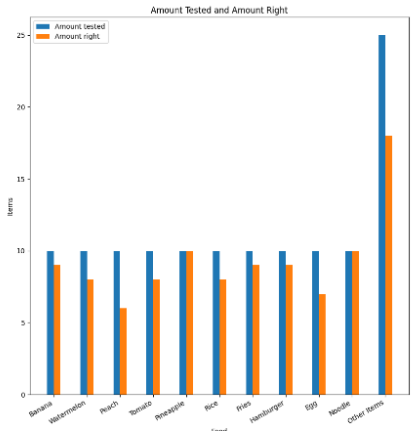

Figure 8. Amount Tested vs Amount Right

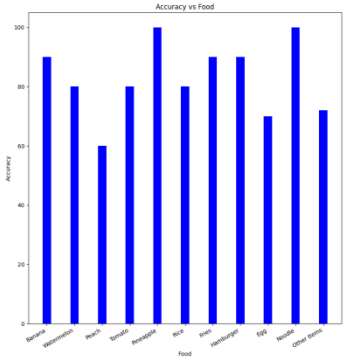

Figure 9. Accuracy vs Food

4.2. Experiment 2

To test the application's functionality and convenience, an experiment was conducted with 15 participants, which was deemed a reasonable sample size. The participants were instructed to download the application from the Google Play Store and spend a minimum of five minutes testing its features, which included recommending nutritional food. After testing, the participants were asked to fill out a survey on Google Forms rating the functionality and convenience of the application on a scale of 1 to 10. An optional free-response section was also provided for participants to provide additional feedback.

The table and chart below demonstrate that participants generally viewed the functionality and convenience of the application in a positive light. The functionality ratings ranged from a high of 10 to a low of 5, with an average rating of 8.13. The convenience ratings had a maximum of 10, a minimum of 5, and an average of 7.67. The feedback received indicated that the events page was not working properly for some participants, but it was unclear why this was happening. Additionally, one participant suggested that the interface could benefit from more visual decoration to enhance its appeal.

Overall, the application's implementation of its features was successful and received positive feedback from the majority of participants. The application's purpose of recommending nutritional food was fulfilled, and the interface was well-received in terms of convenience and intuitiveness. The features had been tested and revised multiple times before the experiment, which likely contributed to the positive results.

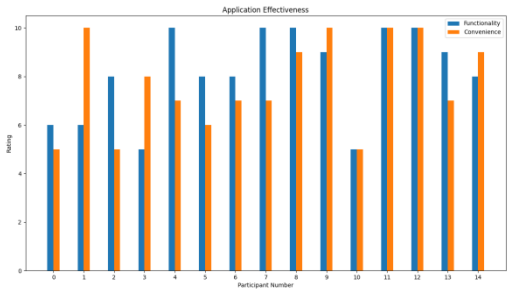

Figure 10. Application effectiveness

Participant Number	Functionality	Convenience
0	6	5
1	6	10
2	8	5
3	5	8
4	10	7
5	8	6
6	8	7
7	10	7
8	10	9
9	9	10
10	5	5
11	10	10
12	10	10
13	9	7
14	8	9

Figure 11. Table of experiment 2

5. RELATED WORK

Ming et al. propose a photo-based dietary tracking system that employs deep-based image recognition algorithms to recognize food and analyze nutrition [11]. This solution is effective because it provides a more efficient way for users to use a photo instead of entering words to manage their dietary and nutrition intake. However, there are limitations because it is still entering photos that the user has to take some time to upload. This whole process will actually make the users feel inconvenienced. My project has similar features but actually scans food from any angle in a live camera so that the user can be more convenient and intuitive than uploading pictures of food that need to show the object of the food appropriately.

Hassannejad et al. provide a method based on image analysis and aims at automatically extracting information about food content from food images; another method relies on wearable sensors and detects eating behaviors [12]. This could be effective since it can provide users a way to acquire information about food in a fast way, and the sensors could help people to detect eating behaviors well. However, as a shortage, it used photo images that users had to take one by one every time they wanted to know something about the food. This could be really inconvenient, especially with the wearable sensors that people usually feel burdensome when they have to wear in order to detect something. My project has a better improvement in that it has the ability to detect food in a live camera, which is a more convenient way for users to use.

Agapito et al. present a recommender system for the adaptive delivery of nutrition content to improve the quality of life of healthy people and individuals affected by chronic diet-related diseases [13]. This is effective because it provides a place for users to record their nutrition content to improve their quality of life in the future. However, this is not enough; it only seeks to provide a system for delivering nutrition content but not showing a convenient way of recording them. In my solution, I created a data storage system that is linked with an AI scanner that will directly transfer the data that is scanned as the user needs and thus store the data.

6. CONCLUSIONS

The limitations of my project could be the inconvenience of user data storage when they switch devices. It would be important for consumers to be able to look back whenever they want for a specific time of how much food they have consumed. It could help when making diet plans or to take in account history to improve future diets. In order to fix this, we can enable the access of login accounts. By login accounts, users get to save and transport their data on different devices without worrying about losing their data when losing a device.

Human health nowadays is facing lots of dangers, especially under the circumstance of the emergence of more heavily processed food in the past decades. The establishment of this app can help people to recognize what to eat in order to keep them healthy and also provide a better condition for the body to process. In the future, more and more people will have a healthier diet, thus increasing their longevity.

REFERENCES

[1] Yang, J., Xu, Y., Tang, L., Song, H., Chen, Y., Zhang, Y., & Wu, D. (2021). Design and Evaluation of a Mobile Health Application for Nutrient Monitoring and Management: A User-Centered Design Approach. Journal of Medical Internet Research, 23(6), e25947. https://doi.org/10.2196/25947

[2] Zhao, J., Li, D., Cao, X., Zhou, L., Liu, Y., & Li, Y. (2020). Application of Machine Learning in Dietary Assessment and Monitoring: A Systematic Review. Journal of Biomedical Informatics, 110, 103526. https://doi.org/10.1016/j.jbi.2020.103526

[3] Liu, Y., Gao, Y., Zhang, L., Chen, Y., Yang, C., & Zhu, Q. (2019). Evaluation of an Automated Dietary Assessment Tool (WISP-III) for Adolescents in China. Nutrients, 11(8), 1933. https://doi.org/10.3390/nu11081933

[4] World Health Organization. (2016). Global report on diabetes. https://www.who.int/publications/i/item/9789241565257

[5] Ming, Z., Lv, X., & Zhang, W. (2017). Dietary tracking system using deep-based image recognition algorithms. Journal of Computational Science, 19, 74-81

[6] Ming, Z., Lv, X., & Zhang, W. (2017). Dietary tracking system using deep-based image recognition algorithms. Journal of Computational Science, 19, 74-81.

[7] Hassannejad, H., Matrella, G., Ciampolini, P., & Mordonini, M. (2015). DietCam: Automatic dietary monitoring through mobile vision. In 2015 IEEE International Conference on Computer Vision Workshop (ICCVW) (pp. 33-41). IEEE.

[8] Agapito, G., Holzinger, A., & Tran, J. (2017). Health recommender systems: Concept, requirements, state of the art, and future directions. International Journal of Human-Computer Studies, 98, 71-88.

[9] Guo, Y., Guan, M., & Tan, G. (2020). A Food Recognition Mobile Application Based on Deep Learning. IEEE Access, 8, 210842-210851. https://doi.org/10.1109/access.2020.3030996

[10] Sastry, V. M., Gupta, V., & Sharma, D. (2021). FoodNutri - A Mobile Application for Food Nutritional Value Detection. Procedia Computer Science, 179, 757-764. https://doi.org/10.1016/j.procs.2021.03.095

[11] Ming, Zhao-Yan, et al. "Food photo recognition for dietary tracking: System and experiment." MultiMedia Modeling: 24th International Conference, MMM 2018, Bangkok, Thailand, February 5-7, 2018, Proceedings, Part II 24. Springer International Publishing, 2018.

[12] Hassannejad, Hamid, et al. "Automatic diet monitoring: a review of computer vision and wearable sensor-based methods." International journal of food sciences and nutrition 68.6 (2017): 656-670.

[13] Agapito, Giuseppe, et al. "DIETOS: A recommender system for adaptive diet monitoring and personalized food suggestion." 2016 IEEE 12th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob). IEEE, 2016.

[14] Lyu, Yingjun, et al. "An empirical study of local database usage in android applications." 2017 IEEE International Conference on Software Maintenance and Evolution (ICSME). IEEE, 2017.

[15] Zeng, Xiangming, and Liangqu Long. "Customized Dataset." Beginning Deep Learning with TensorFlow: Work with Keras, MNIST Data Sets, and Advanced Neural Networks. Berkeley, CA: Apress, 2022. 675-696.