Eater's Edge – Smart Food Prediction And Suggestion for Dining Decisions Using Content Based Approach and KNN

A PROJECT REPORT

Submitted by

ROSHINI V R [211420243046]

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY
IN
ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

PANIMALAR ENGINEERING COLLEGE

(An Autonomous Institution, Affiliated to Anna University, Chennai)

MARCH 2024

PANIMALAR ENGINEERING COLLEGE

(An Autonomous Institution, Affiliated to Anna University, Chennai)

BONAFIDE CERTIFICATE

Certified that this project report **“EATER’S EDGE – SMART FOOD PREDICTION AND SUGGESTION FOR DINING DECISIONS USING CONTENT BASED APPROACH AND KNN”** is the bonafide work of **“ROSHINI.V.R [211420243046]”** who carried out the project work under **, Mrs. R. PRIYA M.E.,** supervision.

SIGNATURE	**SIGNATURE**
Dr. S. MALATHI, M.E., Ph.D.	**Mrs. PRIYA, M.E.,**
HEAD OF THE DEPARTMENT	**ASSISTANT PROFESSOR**
DEPARTMENT OF AI&DS,	DEPARTMENT OF AI&DS,
PANIMALAR ENGINEERING COLLEGE,	PANIMALAR ENGINEERING COLLEGE,
NASARATHPETTAI,	NASARATHPETTAI,
POONAMALLEE,	POONAMALLEE,
CHENNAI-600 123.	CHENNAI-600 123.

Certified that the above-mentioned student was examined in End Semester project viva-voice held on_______________

INTERNAL EXAMINER **EXTERNAL EXAMINER**

DECLARATION BY THE STUDENT

I **ROSHINI V R** [211420243046], hereby declare that this project report titled **EATER'S EDGE – SMART FOOD PREDICTION AND SUGGESTION FOR DINING DECISIONS USING CONTENT BASED APPROACH AND KNN"**, under the guidance of **Mrs. R. PRIYA M.E.,** is the original work done by me and I have not plagiarized or submitted to any other degree in any university by us.

ROSHINI.V.R

ACKNOWLEDGEMENT

We would like to express our deep gratitude to our respected Secretary and Correspondent **Dr. P. CHINNADURAI, M.A., Ph.D.** for his kind words and enthusiastic motivation, which inspired us a lot in completing this project.

We express our sincere thanks to our directors **Tmt. C. VIJAYARAJESWARI**, **Dr. C. SAKTHI KUMAR, M.E., Ph.D. and Dr. SARANYASREE SAKTHI KUMAR B.E.,M.B.A., Ph.D.,** for providing us with the necessary facilities to undertake this project.

We also express our gratitude to our Principal **Dr. K. MANI, M.E., Ph.D**. who facilitated us in completing the project.

We thank the Head of the AI&DS Department, **Dr. S. MALATHI, M.E., Ph.D.,** for the support extended throughout the project.

We would like to thank my Project Guide **Mrs. R. PRIYA M.E.,** and all the faculty members of the Department of AI&DS for their advice and encouragement for the successful completion of the project.

ABSTRACT

The modern lifestyle often leaves individuals with limited time and information to make informed dietary choices. In response, the "Know-Before-You-Eat" web application is designed to empower users in understanding the nutritional content of their meals, identifying food items from images, and even suggesting recipes based on recognized ingredients. Leveraging machine learning techniques implemented through Scikit-Learn, the application employs a content-based approach to offer personalized diet recommendations. FastAPI ensures efficient communication between the user interface and the backend, while Streamlit provides an interactive and intuitive user experience. By amalgamating image recognition, nutritional analysis, and recipe prediction, Know-Before-You-Eat serves as a comprehensive tool for promoting healthier eating habits and enhancing nutritional awareness.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

Abbreviation	Meaning
CNN	convolutional neural networks
LBP	Local binary patterns
API	Application Programming Interface
SVM	Support Vector Machine
AU	Action unit
OpenCV	Computer Vision
KNN	K-Nearest Neighbor

CHAPTER 1
INTRODUCTION

INTRODUCTION

In today's fast-paced lifestyle, maintaining a balanced diet is increasingly challenging, with individuals often struggling to accurately assess the nutritional content of their meals. This lack of knowledge can lead to poor dietary choices, contributing to health issues such as obesity and malnutrition. Moreover, the abundance of food options makes it difficult for people to plan and prepare nutritious meals that align with their dietary goals and preferences. To address these challenges, we introduce Know-Before-You-Eat, an interactive web application designed to revolutionize the way individuals approach their dietary habits. Leveraging advanced image recognition technology, our platform allows users to effortlessly identify food items from images, providing detailed nutritional facts, including calorie counts. Additionally, utilizing a content-based approach powered by Scikit-Learn, Know-Before-You-Eat offers personalized diet recommendations tailored to each user's specific needs and preferences. Through seamless integration with FastAPI and Streamlit, our application ensures a user-friendly experience, enabling individuals to make informed decisions about their diet and access recipe predictions based on identified food names. By amalgamating image recognition, nutritional analysis, and recipe prediction, Know-Before-You-Eat serves as a comprehensive tool for promoting healthier eating habits and enhancing nutritional awareness. Know-Before-You-Eat represents a comprehensive solution to the problem of understanding and managing dietary intake, empowering users to prioritize their health and well-being through informed eating choices

1.1.Problem Definition:

The contemporary lifestyle is characterized by increasingly sedentary habits and the widespread availability of processed foods, leading to a surge in diet-related health issues such as obesity, diabetes, and cardiovascular diseases. Despite the abundance of nutritional information available, individuals often struggle to make informed dietary choices due to factors such as lack of time, knowledge, or resources. This project addresses the pressing need for effective dietary management solutions by leveraging technology to provide accessible, accurate, and personalized nutritional guidance.

The Field of Invention:

The field of invention encompasses the intersection of technology, nutrition science, and user-centric design. By harnessing advancements in ML, web development frameworks, and data analysis techniques, this project aims to revolutionize the way individuals interact with and manage their diets. Furthermore, it explores innovative approaches to translating complex nutritional data into actionable insights, thereby empowering users to make informed dietary decisions that align with their health goals and preferences.

Novelty of the Invention:

The novelty of this invention lies in its holistic approach to dietary management, which integrates state-of-the-art ML algorithms, intuitive web interfaces, and personalized recommendation systems. Unlike traditional dietary tracking tools or nutrition apps that offer generic advice, this project offers tailored solutions based on individual preferences, dietary restrictions, and health objectives. Moreover, the incorporation of image recognition technology for food identification adds a novel dimension to dietary management applications, enhancing user experience and usability.

1.2.Objectives of the Invention:

1. **Accurate Food Identification**: Develop a robust image recognition system capable of accurately identifying food items from images uploaded by users.

2. **Comprehensive Nutritional Analysis**: Provide users with detailed nutritional information, including calorie counts, macronutrient compositions, and micronutrient profiles, for informed dietary decision-making.

3. **Personalized Diet Recommendations:** Implement a content-based recommendation system that generates personalized diet plans based on user preferences, dietary restrictions, and health objectives.

4. **Seamless User Experience:** Create intuitive user interfaces for both applications using Streamlit, ensuring seamless navigation and interaction for users of all levels of technological proficiency.

5. **Promotion of Healthy Eating Habits:** Empower individuals to make healthier dietary choices by providing accessible, accurate, and personalized nutritional guidance, ultimately contributing to improved health outcomes and well-being.

By fulfilling these objectives, the invention aims to address the challenges associated with dietary management in modern society and pave the way for a healthier future.

CHAPTER 2
LITERATURE SURVEY

LITERATURE SURVEY:

Let us discuss several techniques adopted by various authors through this survey.

- Rominkumar Busa et al.[1] have introduced a robust textual recognition system that leverages the fusion of Optical Character Recognition (OCR) with a Convolutional Recurrent Neural Network (CRNN) model, employing Connectionist Temporal Classification (CTC) loss for classification. In this system, OCR initially extracts text from images, providing input to the CRNN model. The CRNN architecture combines Convolutional Neural Networks (CNNs) for feature extraction and Recurrent Neural Networks (RNNs) for sequential processing, allowing it to effectively capture complex patterns in text data. The utilization of CTC loss enables the model to handle variable-length sequences of characters without explicit segmentation, enhancing its adaptability to diverse text formats. During training, the CRNN model learns to map image features to corresponding text sequences, optimizing parameters through backpropagation. In inference, the trained model accurately predicts text from input images.

- K. Jayashree et al.[2] have introduced a novel Deep Learning technique for the detection of eye diseases leveraging computer vision methodologies. In this approach, an eye image serves as input to the system, which undergoes a series of processing steps to extract relevant features indicative of the presence of a disease. These processing steps typically involve pre-processing techniques such as image normalization, enhancement, and segmentation to isolate the region of interest within the eye image. Subsequently, Deep Learning models, such as Convolutional Neural Networks (CNNs) or more advanced architectures like Siamese networks or attention mechanisms, are employed to learn discriminative features from the pre-processed images. During training, the model learns to distinguish between healthy and diseased eye images by iteratively adjusting its parameters based on labeled data

- Kavitha Prithiviraj and S Prabakaran[3] have devised a sophisticated system for image analysis across multiple orientations, employing the Grey-Level Co-occurrence Matrix (GLCM) technique. Their approach involves rotating images at various angles to extract comprehensive features for analysis. The GLCM method quantifies the spatial relationships of pixel intensity values in an image, capturing textural information crucial for disease detection. By rotating images and applying GLCM, the system systematically explores different orientations, enhancing feature extraction robustness. Once features are extracted, a predictive model, such as a classifier or regression algorithm, is trained on labeled data to correlate extracted features with disease presence. During inference, the model evaluates new images, utilizing the extracted features to make predictions regarding the presence or severity of the disease.

- Jaewoo Park et al.[4] have introduced an innovative Optical Character Recognition (OCR) system tailored for multilingual characters, employing a segmentation-based approach combined with machine learning for feature analysis. In this system, textual images undergo segmentation into individual characters or glyphs, facilitating more precise feature extraction. Subsequently, machine learning techniques are utilized for comprehensive feature analysis, allowing the system to discern intricate patterns and variations present in multilingual characters. These features could include structural elements, stroke patterns, or spatial relationships within the characters. During training, the system learns to associate these extracted features with corresponding characters or linguistic elements across multiple languages, effectively capturing the diverse characteristics of different scripts. Once trained, the model can predict characters in unseen textual images with high accuracy, irrespective of the language or script used.

- Cristiano Premebida et al.[5] have introduced an advanced Pedestrian Detection System that integrates LiDAR (Light Detection and Ranging) technology with cameras to enhance pedestrian safety on roads. In this system, LiDAR sensors are employed to capture 3D spatial information about the surrounding environment, including pedestrians and other obstacles. The LiDAR data is processed using sophisticated algorithms to detect the presence of pedestrians accurately. When a pedestrian is detected within the vicinity of a road, the system triggers an alert mechanism to notify nearby civilians, particularly drivers, about the pedestrian's presence. This alert system serves as a proactive measure to mitigate the risk of pedestrian-related accidents, providing timely warnings to drivers and encouraging them to exercise caution while navigating through potentially hazardous areas. By combining LiDAR technology with camera-based pedestrian detection and alerting mechanisms.

- Chakaravarthi S et al.[6] have introduced an innovative detection and tracking technique leveraging Superpixel extraction. Superpixels are compact, perceptually meaningful image regions that group pixels with similar characteristics together. In their approach, images are first segmented into superpixels to reduce computational complexity while preserving relevant visual information. These superpixels serve as building blocks for subsequent detection and tracking tasks. Detection involves identifying objects or regions of interest within the image, while tracking involves following these objects over time or across frames. By operating on the level of superpixels rather than individual pixels, the proposed technique achieves greater efficiency and robustness in both detection and tracking tasks. Superpixel-based methods enable more effective feature extraction and representation, facilitating accurate object localization and trajectory prediction. This technique finds applications across various domains, including surveillance, autonomous navigation, and augmented reality, where real-time and reliable detection and tracking are essential.

- Charles R. Qi et al.[7] have introduced a groundbreaking neural network architecture named PointNet, designed specifically for analyzing point clouds—a data format characterized by its irregular structure. Point clouds represent spatial data as a collection of individual points in 3D space, often generated by sensors like LiDAR. PointNet revolutionizes the analysis of such data by directly consuming raw point sets, overcoming the challenges posed by their unordered and irregular nature. By employing a novel permutation-invariant function, PointNet is capable of capturing global features from point clouds regardless of their permutation, enabling robust and efficient processing. Furthermore, PointNet's architecture enables it to perform various tasks on point cloud data, including classification, segmentation, and object detection, with impressive accuracy and generalization capabilities.

- Joseph Redmon et al.[8] have spearheaded a pioneering effort in the realm of computer vision with the introduction of a unified real-time object detection framework, which has since become a cornerstone for rapid advancements in the field. This framework, characterized by its efficiency, accuracy, and speed, revolutionized the landscape of object detection by providing a unified architecture capable of detecting multiple objects in real-time. By integrating deep learning techniques with innovative algorithmic optimizations, the framework achieves remarkable performance on various datasets and across diverse application domains. Its versatility extends to accommodating different object sizes, shapes, and orientations, making it adaptable to a wide range of scenarios. Furthermore, the framework's real-time capabilities have significant implications for applications requiring timely decision-making, such as autonomous driving, surveillance, and augmented reality

- Wei Yang et al.[9] have made significant contributions to real-time vehicle detection algorithms, particularly through the utilization of frameworks like YOLOv2 (You Only Look Once version 2), which offer practical solutions for enhancing traffic management and safety measures. Their work addresses the critical need for efficient and accurate detection methods in dynamic environments such as roads and intersections. YOLOv2, known for its speed and effectiveness, enables rapid identification and localization of vehicles within video streams or images in real-time. By leveraging deep learning techniques and advanced architectural designs, these algorithms can reliably detect vehicles across various scales, orientations, and lighting conditions, crucial for ensuring robust performance in diverse traffic scenarios. Moreover, Yang and colleagues have conducted comprehensive comparisons among different detection models, shedding light on their respective strengths and weaknesses in terms of accuracy, speed, and computational efficiency.

- Tibor Trnovszky et al. have introduced a pioneering approach that harnesses the power of Convolutional Neural Networks (CNNs) for animal recognition systems, showcasing the transformative potential of deep learning in wildlife monitoring efforts. Their methodology represents a significant leap forward in the field, leveraging CNNs to automatically identify and classify animal species from visual data such as images or video footage. By training CNN models on large datasets of annotated animal images, Trnovszky and colleagues demonstrate the capability of deep learning algorithms to accurately recognize diverse species across different environmental settings and conditions. .Moreover, they have proposed various methods to implement CNN-based animal recognition systems, offering insights into model architectures, training strategies, and optimization techniques.

- Gyanendra et al. have introduced a pioneering CNN-based technique for wild animal detection, offering a comprehensive solution for monitoring and analyzing wildlife populations via camera trap networks. Their approach involves training Convolutional Neural Networks (CNNs) on extensive datasets of annotated wildlife images to enable precise identification and classification of various animal species. Leveraging the power of deep learning, the system can accurately detect and recognize wild animals captured in camera trap images or videos. Additionally, Gyanendra and colleagues have developed a suite of monitoring and analysis tools built upon their CNN-based detection technique. These tools facilitate a range of research activities, including population estimation, species distribution mapping, behavior analysis, and biodiversity assessments.

- Yuvaraj Munian et al. have introduced an innovative project aimed at advancing the field of wild animal auto-detection during active nocturnal hours using thermal image processing mounted on vehicle-based camera systems. This project represents a novel approach to wildlife monitoring, particularly in low-light conditions where traditional visual-based techniques may be limited. By utilizing thermal imaging technology, the system can detect heat signatures emitted by animals, enabling effective detection even in darkness or adverse weather conditions. The obtained radiometric images are processed by an intelligent system equipped with advanced algorithms for hot spot and moving object detection. Through transformation and analysis of thermal data, the system identifies potential wildlife presence with high accuracy, allowing for timely responses to mitigate human-wildlife conflicts or facilitate ecological research

- B. Karthikeya Reddy et al. have embarked on a significant research endeavor focusing on the application of the YOLOv3 model for animal identification within user-provided images. YOLOv3, an evolution of the YOLO (You Only Look Once) object detection algorithm, employs the darknet framework and leverages a pre-trained dataset to facilitate animal recognition tasks. The darknet architecture, renowned for its efficiency and effectiveness in object detection, serves as the backbone of the YOLOv3 model, enabling it to detect and localize animals within images with remarkable accuracy and speed. The research evaluates the overall performance of the YOLOv3 model across a diverse range of training and testing images drawn from the pre-trained dataset. By systematically analyzing the model's performance on various datasets, the study aims to assess its robustness, generalization capabilities, and potential limitations in accurately identifying different animal species across different environmental contexts and conditions..

CHAPTER 3
SYSTEM ANALYSIS

3.1 Existing System

Existing System:

Traditional methods of managing dietary habits relied on manual techniques such handwritten food journals or consultations with nutritionists. These methods w time-consuming, prone to errors, and lacked real-time feedback, making challenging for users to maintain accurate records and receive timely guidance.

Traditional Methods:

Before technology-driven solutions emerged, individuals typically relied conventional methods for dietary management, including handwritten food diaries consultations with nutritionists.

Communication Challenges:

Traditional methods faced communication barriers between users and healthc professionals, making it difficult to access personalized dietary advice. Schedul appointments and geographical constraints limited accessibility, while slow respo times hindered timely support and guidance.

Limited Sustainability:

Traditional approaches struggled to promote long-term adherence to healthy eat habits due to their tedious nature and lack of real-time feedback. Without sustaina support systems, users were more likely to revert to old habits, undermining effectiveness of traditional methods in fostering lasting behavior change.

.

3.2 Proposed system

Proposed System:

1. Computer Vision and Machine Learning for Food Identification:

- Integration of computer vision techniques to recognize food items fr uploaded images.
- Utilization of machine learning algorithms for accurate food identification a extraction of nutritional information.
- Implementation of Flask framework to develop an interactive web platform image-based food analysis.

2. Nutritional Analysis and Calorie Counting:

- Extraction of relevant nutritional information, including calorie counts, fr identified food items.
- Automation of dietary tracking and analysis, reducing the need for manual d entry.
- Provision of accurate nutritional insights to users, aiding in informed diet decision-making.

3. Recipe Suggestions and Culinary Inspiration:

- Incorporation of recipe recommendation functionality based on identified fo items.
- Enhancement of user engagement and exploration of diverse meal options.
- Promotion of balanced and healthy eating habits through culinary.

4. **Personalized Diet Recommendations:**

- Development of a content-based approach using machine learning algorithm
- Integration of FastAPI and Streamlit frameworks to create a user-frien interface for personalized diet planning.
- Customization of diet plans based on individual preferences, health goals, a dietary restrictions.

5. **Continuous Learning and Adaptation:**

- Incorporation of feedback mechanisms to capture user preferences a behavior.
- Adaptation of diet recommendations over time to better align with use evolving needs and preferences.
- Promotion of sustainable and long-term adherence to healthier eating hab through adaptive algorithms.

6. **Synergistic Integration of Technologies:**

- Synergy between computer vision, machine learning, and web developm technologies to address dietary management challenges comprehensively.
- Leveraging the strengths of each technology to enhance user experience a effectiveness of the system.
- Demonstration of interdisciplinary collaboration in developing innovat solutions for dietary management.33

3.3 Feasibility Study

3.3.1 Technical feasibility:

- The proposed system is technically feasible due to the availability of establisl libraries and frameworks like OpenCV, Scikit-Learn, Flask, FastAPI, a Streamlit. These tools provide robust support for computer vision, mach learning, and web development, with ample documentation and commun resources for streamlined development. The system operates efficiently standard hardware, eliminating the need for specialized equipment.

- Scalability and performance are ensured through optimized design practic including caching and load balancing techniques. Compatibility with exist platforms is facilitated by adherence to standard protocols and data form Security measures such as encryption and authentication mechanisms prot user data, ensuring compliance with privacy regulations.

- Rigorous testing procedures validate the system's functionality and reliabil with iterative development allowing for continuous improvement based on u feedback. Overall, the proposed system offers a comprehensive solution dietary management, prioritizing scalability, performance, security, a interoperability.

3.3.2 Economic Feasibility :

- The proposed system demonstrates strong economic feasibility as it levera widely available open-source libraries and frameworks, minimiz development costs associated with software tools and licenses. Additionally, system operates efficiently on standard hardware, reducing infrastruct expenses. By streamlining dietary management processes and promot healthier eating habits, the system has the potential to yield long-term c savings in healthcare expenses associated with diet-related illnesses. Moreov the scalable architecture of the system allows for gradual expansion as u demands increase, ensuring optimal resource allocation and return investment over time.

3.3.3 Market Feasibility :

- The proposed system addresses a growing market demand for diet management solutions by offering personalized features and user-frien interfaces. With scalability and potential partnerships, it is poised to captur significant share of the expanding health and wellness technology market.

3.3.4 Operational feasibility :

- The proposed system demonstrates operational feasibility through its seaml integration into users' routines. With user-friendly interfaces and accessibi across devices, it simplifies food identification, nutritional analysis, and c planning. Its scalability allows it to accommodate growing user demands, wh feedback mechanisms ensure continuous improvement.

3.4 Software Environment :

- The software environment for the proposed wildlife monitoring and alert system involves a combination of programming languages, libraries, and development tools to enable seamless integration and operation. Below are the key components of the software environment:

1. Computer Vision and Machine Learning:

- OpenCV: This library is utilized for image processing tasks like identifying food items from images.

- Scikit-Learn: It provides machine learning algorithms to classify food items and perform nutritional analysis based on the data extracted from images.

2. Web Development:

- Flask and FastAPI: These frameworks aid in building interactive web applications. They enable the creation of user-friendly interfaces where users can upload images, receive nutritional information, and access diet recommendations.

- Streamlit: This tool is used for data visualization, allowing users to interactively explore dietary insights and recommendations presented in an intuitive format.

3.5 Database Management:

- MySQL or MongoDB: These database management systems store user profiles, food items, and dietary preferences. They ensure efficient organization and retrieval of data necessary for providing personalized diet advice.

CHAPTER 4
SYSTEM DESIGN

4.1. Architecture Diagram

A content-based recommendation engine is a type of recommendation system that uses the characteristics or content of an item to recommend similar items to users. It works by analyzing the content of items, such as text, images, or audio, and identifying patterns or features that are associated with certain items. These patterns or features are then used to compare items and recommend similar ones to users.

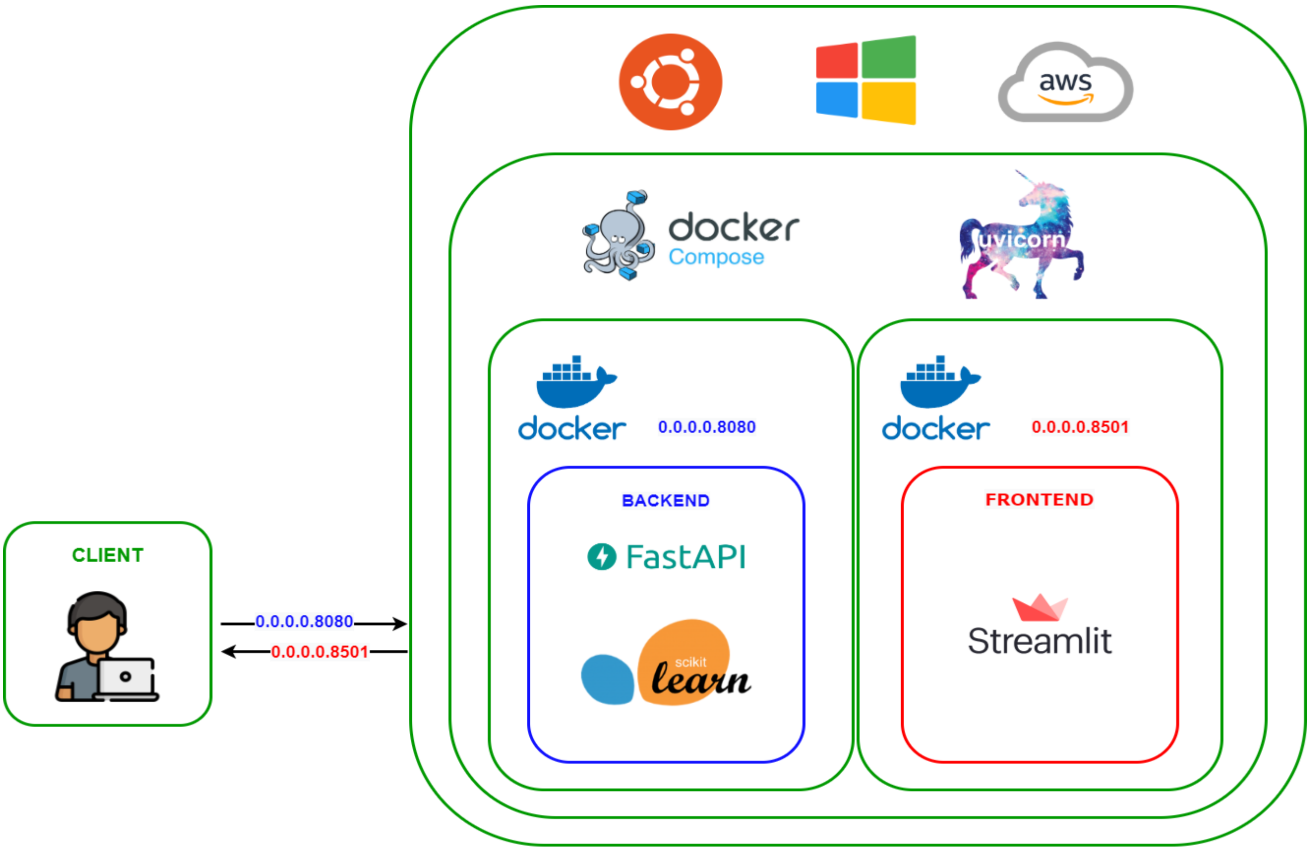

FIG 4.0

CHAPTER 5
SYSTEM ARCHITECTURE

5.1 Hardware Components:

a. Server Infrastructure:

- Physical or Virtual Servers: Host the backend components of the system, including application logic and databases.
- Storage Devices: Store application code, databases, and user data securely.
- Networking Equipment: Facilitate communication between servers, users, and external resources.

5.2 Software Components:

Computer Vision Libraries:

- OpenCV: Used for image processing tasks such as food item recognition and feature extraction.
- TensorFlow or PyTorch: Deep learning frameworks for advanced image analysis tasks, such as object detection or image classification.

Machine Learning Libraries:

- Scikit-Learn: Provides machine learning algorithms for food item identification and nutritional analysis based on extracted features.
- TensorFlow or PyTorch: Utilized for training and deploying deep learning models for more complex dietary analysis tasks.

Web Development Frameworks:

- Flask or FastAPI: Lightweight web frameworks for building the backend of the web application and handling HTTP requests.
- Streamlit : For creating interactive data visualization interfaces for presenting nutritional information and diet recommendations.

These software components work together to create a robust and scalable system for identifying food items, analyzing their nutritional content, and providing personalized diet recommendations to users. Depending on the specific requirements and technical preferences, different combinations of these components can be utilized to achieve the desired functionality and performance.

5.3 System Workflow:

The system workflow begins with users interacting with the application by uploading images of their meals and specifying dietary preferences or goals. These images undergo image processing using computer vision techniques to identify individual food items. Machine learning algorithms then analyze these items to calculate their nutritional content, including calories and macronutrients. Based on this analysis and user-input goals, personalized dietary recommendations are generated, suggesting suitable meal options and portion sizes. These recommendations, along with detailed nutritional information, are presented to users through the application interface. Users have the opportunity to provide feedback on the recommendations, which is used to refine future suggestions. User profiles and data are securely stored in the system's database for future reference. Over time, the system continuously learns and adapts based on user interactions and feedback, enhancing the accuracy and relevance of its recommendations for improved dietary management.

5.4 Security Measures:

1. Encryption:
   - Employ SSL/TLS encryption to secure data transmission.
   - Encrypt sensitive data at rest to prevent unauthorized access.

2. Authentication and Authorization:
   - Implement robust user authentication mechanisms.
   - Utilize role-based access control to limit data access.

3. Data Protection:
   - Ensure confidentiality of user profiles and dietary data.
   - Implement measures to protect against data breaches and unauthorized access.

4. Regular Audits and Updates:
   - Conduct security audits to identify vulnerabilities.
   - Promptly apply software updates and patches to mitigate risks.

5.5 Reporting and Analytics:

Reporting and analytics in the dietary management system play a pivotal role in understanding user behaviors, optimizing system performance, and providing actionable insights for users.

User Engagement Tracking:

- Monitoring user interactions such as uploads, searches, and viewed recommendations helps gauge user engagement levels and preferences. This data provides valuable insights into how users interact with the system and what features are most utilized.

Nutritional Analysis Reporting:

- Generating reports on dietary habits and nutrient intake allows users to gain a comprehensive understanding of their eating patterns. These reports highlight areas for improvement and adherence to recommended dietary guidelines, empowering users to make informed decisions about their nutrition.

Feedback Analysis:

- Analyzing user feedback helps identify trends, common issues, and areas for improvement within the system. By understanding user sentiments and preferences, the system can refine its functionalities and recommendations to better meet user needs.

Performance Monitoring:

- Tracking system performance metrics such as response time, uptime, and resource utilization ensures optimal system operation. By monitoring performance, the system can identify and address any performance bottlenecks or issues that may arise, ensuring a seamless user experience.

Data Visualization:

- Presenting analytical findings through interactive dashboards and visualizations enhances comprehension and user engagement. Visual representations of data such as charts, graphs, and heatmaps provide users with clear and actionable insights into their dietary habits and nutritional intake.

Continuous Improvement:

- Leveraging analytics insights and user feedback, the system can continuously evolve and improve its functionalities over time. Regular updates and enhancements ensure that the system remains relevant, effective, and capable of meeting the changing needs of its users.

5.6. Algorithm:

1. Image Upload:

- Users upload an image of their meal using the web interface of the application.

2.Image Preprocessing:

- Preprocess the uploaded image to enhance quality and remove any noise or artifacts. Common preprocessing techniques include resizing, cropping, and noise reduction.

3. **Object Detection:**

- Utilize computer vision techniques, such as convolutional neural networks (CNNs) or pre-trained models like YOLO (You Only Look Once), to detect and localize individual food items within the image.

4. **Feature Extraction:**

- Extract relevant features from the detected food items, such as color histograms, texture descriptors, and shape attributes. These features provide valuable information for subsequent analysis.

5. **Nutritional Data Retrieval:**

- Retrieve nutritional data for each identified food item from a predefined database or external APIs. This data typically includes information such as calories, macronutrients (carbohydrates, proteins, fats), micronutrients (vitamins, minerals), and serving sizes.

6. **Nutritional Analysis:**

- Utilize machine learning algorithms, such as regression or classification models, to analyze the extracted features and nutritional data. This analysis aims to calculate the overall nutritional content of the meal, considering the quantities and combinations of various food items.

7. **User Preferences Incorporation:**

- Incorporate user-provided dietary preferences or goals into the analysis process.

This may include factors such as dietary restrictions (e.g., vegetarian, gluten-free), nutritional targets (e.g., calorie intake, protein consumption), or health objectives (e.g., weight loss, muscle gain).

8. Dietary Recommendation Generation:

- Based on the nutritional analysis and user preferences, generate personalized dietary recommendations for the user. These recommendations may include suggesting alternative meal options, adjusting portion sizes, or modifying nutrient compositions to better align with the user's goals and preferences.

9. Presentation to User:

- Present the dietary recommendations and nutritional analysis results to the user through the web interface of the application. Ensure that the information is displayed in an easy-to-understand and visually appealing format, using charts, graphs, or textual descriptions as appropriate.

10. Feedback Collection:

- Allow users to provide feedback on the recommendations or analysis received. This feedback could include ratings, comments, or specific suggestions for improvement. Collect and store this feedback data for future analysis and refinement of the recommendation algorithms.

CHAPTER 6
SYSTEM IMPLEMENTATION

6.1 Program / Code

Python Code (Sender side):

```
import os
import json
import numpy as np
import pandas as pd

from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image

# Flask utils
from flask import Flask, redirect, url_for, request, render_template
from werkzeug.utils import secure_filename
from gevent.pywsgi import WSGIServer



from splinter import Browser
from bs4 import BeautifulSoup
import pandas as pd
import requests
import os


# Define a flask app
app = Flask(__name__)

# Model saved with Keras model.save()
#MODEL_PATH = os.path.join("models","keras_models", "model-mobilenet-
RMSprop0.0002-001-0.930507-0.647776.h5")
MODEL_PATH = os.path.join("models","keras_models", "model-mobilenet-
RMSprop0.0002-008-0.995584-0.711503.h5")

# Load your trained model
model = load_model(MODEL_PATH)
print("Model loaded successfully !! Check http://127.0.0.1:5000/")

with open(os.path.join("static","food_list", "food_list.json"), "r", encoding="utf8")
as f:
    food_labels = json.load(f)
```

```
class_names = sorted(food_labels.keys())
label_dict = dict(zip(range(len(class_names)), class_names))

food_calories = pd.read_csv(os.path.join("static","food_list", "Food_calories.csv"))

def prepare_image(img_path):
    img = image.load_img(img_path, target_size=(224, 224))
    # Preprocessing the image
    x = image.img_to_array(img) / 255
    x = np.expand_dims(x, axis=0)
    return x


@app.route("/", methods=["GET"])
def Home():
    # Main page
    #Food = mongo.db.collection.find_one()
    return render_template('Know_Before_You_Eat.html')




@app.route("/predict", methods=["GET", "POST"])
def upload():
    data = {}
    if request.method == "POST":
        # Get the file from post request
        f = request.files["image"]

        # Save the file to ./upload_image
        basepath = os.path.dirname(__file__)
        file_path = os.path.join(basepath, "upload_image",
secure_filename(f.filename))
        f.save(file_path)

        # Make prediction
        image = prepare_image(file_path)
        preds = model.predict(image)
        predictions = preds.argmax(axis=-1)[0]
        pred_label = label_dict[predictions]

        food_retrieve = food_calories[food_calories["name"]==pred_label]

        food_nutrional_min = food_retrieve["nutritional value min,kcal"]
```

```
food_nutrional_min=np.array(food_nutrional_min)
food_nutrional_min = str(food_nutrional_min)


food_nutrional_max = food_retrieve["nutritional value max,kcal"]
food_nutrional_max=np.array(food_nutrional_max)
food_nutrional_max = str(food_nutrional_max)

Unit = food_retrieve["unit"]
Unit=np.array(Unit)
Unit = str(Unit)

Calories = food_retrieve["average cal"]
Calories=np.array(Calories)
Calories = str(Calories)

data = pred_label

if data=="beef carpaccio":
   data="carpaccio"
elif data=="cheese plate":
    data="cheese"
elif data=="chicken quesadilla":
    data="quesadilla"
elif data=="chicken wings":
    data="Buffalo wing"
elif data=="grilled salmon":
    data="Salmon#As_food"
elif data=="lobster roll sandwich":
    data="lobster roll"
elif data=="strawberry shortcake":
    data="Shortcake#Strawberry_shortcake"

path={'executable_path':'/usr/local/bin/chromedriver'}
browser=Browser('chrome',**path,headless=False)
# browser=Browser('chrome',path,headless=True)

if data=="tuna tartare":
    url="http://ahealthylifeforme.com/tuna-tartare-recipe/"
    browser.visit(url)
    html=browser.html
    soup=BeautifulSoup(html,"html.parser")
    var=soup.select_one('div.entry-content')
    description=var.select('p')
```

```
        else:
            url="https://en.wikipedia.org/wiki/"
            browser.visit(url+data)
            html=browser.html
            soup=BeautifulSoup(html,"html.parser")
            var=soup.select_one('div.mw-parser-output')
            description=var.select('p')
            nutri=soup.select_one('table.infobox')

        if (data=="greek salad" or data=="oysters" or data=="smoked scallop" or
data=="paella"):
            output=description[1].text
        elif data=="mussels" :
            output=description[2].text
        elif data=="Salmon#As_food":
            output=description[3].text
        else:
            if description[0].text!='\n':
                output=description[0].text
            elif description[0].text=='\n' and description[1].text!='\n':
                output=description[1].text
            elif description[1].text=='\n' and description[2].text!='\n':
                output=description[2].text
        output
        description = output
        browser.quit()


        return "<center><i><h4>" + pred_label.title()+" </h4></i>
"+"<b><h3>Probability</h3></b><h4>"+str(preds.max(axis=-1)[0]) + '\n' +
"</h4><br><br><b><h4 class=\"desc\">" +\
        description + "</h4><br><br>" +\
        "<div class=\"heading-section\"><h2 class=\"mb-4\"><span>Nutrional
Facts</span></h2></div><hr></hr>" + \
        "<h5><b>Nutrional Value - Min (kcal)  :  </b>" +
food_nutrional_min + '\n' + "<br><br>" + \
        "<b>Nutrional Value - Max (kcal)  :  </b>" + food_nutrional_max
+ '\n' + "<br><br>" + \
        "<b> Avg Calories  :  </b>" + Calories + "<br><br>" + \
        "<b> Unit  :  </b>" + Unit + '\n' + "</h5></center> <br><br>" + \
        "<div id=\"Recipe\" class=\"heading-section\"><h2 class=\"mb-
4\"><span>Recipe - Cookbook </span></h2></div><hr></hr>" + \
        str(nutri)
  return None
```

```
if __name__ == "__main__":
    # Serve the app with gevent
    http_server = WSGIServer(("0.0.0.0", 5000), app)
    http_server.serve_forever()
```

Python Code (Diet recommendation):

Backend:

```
from fastapi import FastAPI
from pydantic import BaseModel,conlist
from typing import List,Optional
import pandas as pd
from model import recommend,output_recommended_recipes


dataset=pd.read_csv('../Data/dataset.csv',compression='gzip')

app = FastAPI()


class params(BaseModel):
    n_neighbors:int=5
    return_distance:bool=False

class PredictionIn(BaseModel):
    nutrition_input:conlist(float, min_items=9, max_items=9)
    ingredients:list[str]=[]
    params:Optional[params]


class Recipe(BaseModel):
    Name:str
    CookTime:str
    PrepTime:str
    TotalTime:str
    RecipeIngredientParts:list[str]
    Calories:float
    FatContent:float
    SaturatedFatContent:float
    CholesterolContent:float
    SodiumContent:float
```

```
    CarbohydrateContent:float
    FiberContent:float
    SugarContent:float
    ProteinContent:float
    RecipeInstructions:list[str]

class PredictionOut(BaseModel):
    output: Optional[List[Recipe]] = None


@app.get("/")
def home():
    return {"health_check": "OK"}


@app.post("/predict/",response_model=PredictionOut)
def update_item(prediction_input:PredictionIn):

recommendation_dataframe=recommend(dataset,prediction_input.nutrition_input,pr
ediction_input.ingredients,prediction_input.params.dict())
    output=output_recommended_recipes(recommendation_dataframe)
    if output is None:
        return {"output":None}
    else:
        return {"output":output}
```

Frontend:

```
import requests
import json

class Generator:
def__init__(self,nutrition_input:list,ingredients:list=[],params:dict={'n_ne
ighbors':5,'return_distance':False}):
        self.nutrition_input=nutrition_input
        self.ingredients=ingredients
        self.params=params

    def set_request(self,nutrition_input:list,ingredients:list,params:dict):
        self.nutrition_input=nutrition_input
        self.ingredients=ingredients
```

```
        self.params=params

    def generate(self,):
        request={
            'nutrition_input':self.nutrition_input,
            'ingredients':self.ingredients,
            'params':self.params
        }

response=requests.post(url='http://backend:8080/predict/',data=json.dump
s(request))
        return response
```

CHAPTER 7
SYSTEM TESTING

System testing is a critical phase in the development lifecycle of the dietary management project. It ensures that the application functions as intended, meets user requirements, and performs reliably under various conditions. Here's an overview of the testing process for the project:

1. Functional Testing:

- Image Processing: Verify that the system accurately identifies and extracts food items from uploaded images.
- Nutritional Analysis: Test the accuracy of nutritional calculations and ensure they align with expected values.
- Dietary Recommendations: Validate that personalized recommendations are generated based on user preferences and nutritional analysis results.
- User Interaction: Test the functionality of the web interface, including image upload, form submission, and feedback submission.

2. Performance Testing:

- Response Time: Measure the time taken for the system to process image uploads, perform nutritional analysis, and generate recommendations. Ensure that response times are within acceptable limits.
- Scalability: Test the system's ability to handle concurrent user requests and increasing loads without significant degradation in performance.
- Resource Utilization: Monitor CPU, memory, and disk usage during peak load scenarios to identify any resource bottlenecks.

3. Security Testing:

- Data Encryption: Verify that sensitive user data, such as login credentials and dietary information, is encrypted during transmission and storage.
- Authentication and Authorization: Test the effectiveness of authentication

mechanisms and access controls to prevent unauthorized access to user accounts and data.

- Vulnerability Assessment: Conduct penetration testing and vulnerability scanning to identify and address potential security vulnerabilities, such as SQL injection or cross-site scripting (XSS) attacks.

4. Usability Testing:

- User Interface: Evaluate the usability and intuitiveness of the web interface by soliciting feedback from users or conducting usability tests.
- Accessibility: Ensure that the application is accessible to users with disabilities, complying with accessibility standards such as WCAG (Web Content Accessibility Guidelines).

5. Compatibility Testing:

- Cross-Browser Compatibility: Test the application on different web browsers (e.g., Chrome, Firefox, Safari) to ensure consistent functionality and appearance.
- Mobile Responsiveness: Verify that the application is responsive and usable across various devices and screen sizes, including smartphones and tablets.

6. Integration Testing:

- External APIs: Test integration with external APIs for nutritional data retrieval, ensuring seamless communication and accurate data exchange.
- Database Integration: Verify the integrity of data stored in the database and test data retrieval and manipulation operations.

7. Regression Testing:

- Perform regression testing whenever code changes or updates are made to the application to ensure that existing functionality remains unaffected.

8. User Acceptance Testing (UAT):

- User Feedback: Solicit feedback from real users through beta testing or pilot programs to validate that the application meets their needs and expectations.
- Validation of Requirements: Verify that the application fulfills all specified user requirements and objectives.

9. Documentation Review:

- User Documentation: Review user manuals, guides, and help documentation to ensure completeness and accuracy.
- Technical Documentation: Validate technical documentation, including system architecture diagrams, API documentation, and code comments.

By rigorously conducting these types of testing, the dietary management project can ensure the reliability, security, and usability of the application, ultimately delivering a high-quality solution that meets the needs of its users.

RESULTS

1.After fine-tuning a pre-trained MobileNet model achieved about 99.03% Top-1 Accuracy on the Training set and about 73% accuracy on Valid & test data.
2.After fine-tuning a pre-trained VGG16 model achieved about 98.03% Top-1 Accuracy on the Training set and about 70% accuracy on Valid & test data.
3.Using KNN Algorithm achieved at score:0.404 at K=3
4.Using Random Forest Model achieved at score:0.2

RESEARCH & DISCUSSION :

The architecture works as follows:

- Resizes the input image into 448x448 before going through the convolutional network.
- A 1x1 convolution is first applied to reduce the number of channels, which is then followed by a 3x3 convolution to generate a cuboidal output.
- The activation function under the hood is ReLU, except for the final layer, which uses a linear activation function.
- Some additional techniques, such as batch normalization and dropout, respectively regularize the model and prevent it from overfitting.

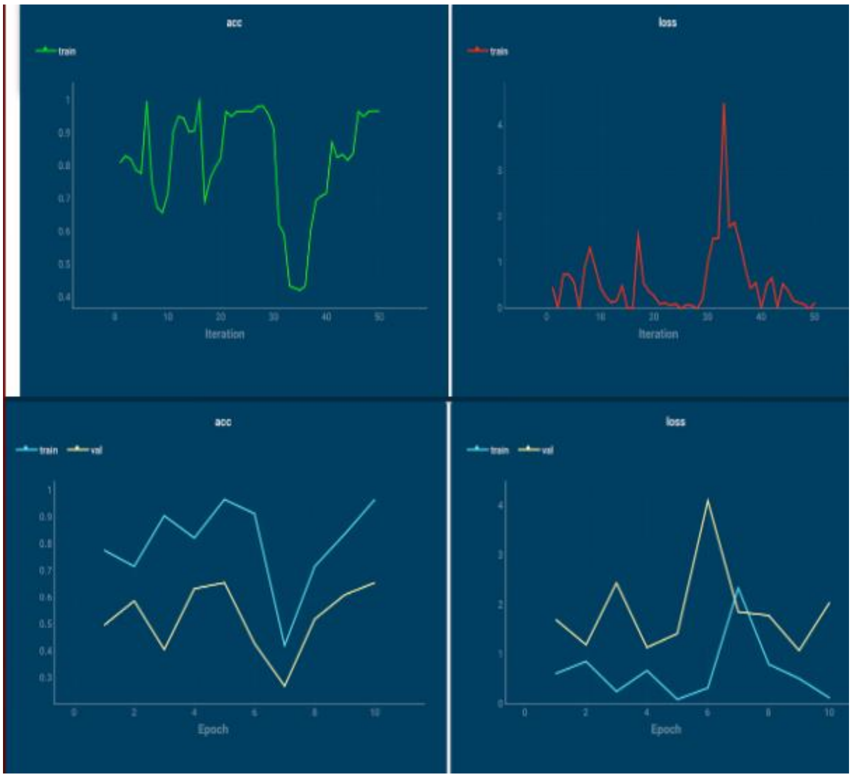

FIG 7.0

Object Detection Performance Comparison

The current and the latest yolov8 model that we use, has a mean average precision of about 91.7% and achieves a minimum of 40 to 45 fps, which makes it ideal for the real time animal monitoring and alerting. It is justified as follows:

Object Detection Performance Comparison (YOLOv8 vs YOLOv5)

Model Size	YOLOv5	YOLOv8	Difference
Nano	28	37.3	+33.21%
Small	37.4	44.9	+20.05%
Medium	45.4	50.2	+10.57%
Large	49	52.9	+7.96%
Xtra Large	50.7	53.9	+6.31%

*Image Size = 640

TABLE 7.1

Grayscale

In using KNN with grayscale images, the process involves converting images to grayscale, extracting features such as pixel values, and storing them along with labels during training. When classifying new images, distances to training data are calculated, and the k-nearest neighbors are determined. Majority voting among these neighbors assigns a class to the new image, which is then outputted.

KNN Model	K	Score
Grayscale=**False** 1 picture resolution=**512*512*3** Number of Training pictures=**606** (6 per each category) **Test_score.** (k=3, N_pictures=101) < 0.01	3	0.236
	5	0.165
	7	0.140
	9	0.122

KNN Model	K	Score
Grayscale=**True** 1 picture resolution=**512*512*1** Number of Training pictures=**1010** (10 per each category) **Test_score.** (k=3, N_pictures=101) = 0.01	3	0.404
	5	0.232
	7	0.163
	9	0.148

FIG 7.2

CHAPTER 8
CONCLUSION

CONCLUSION:

The development of the dietary management project represents a significant endeavor aimed at promoting healthier eating habits and facilitating personalized nutritional guidance for users. Throughout the project lifecycle, various challenges were addressed, and innovative solutions were implemented to create a robust and user-friendly application. As we conclude this project, several key points emerge:

1. Enhanced User Experience:

- The project prioritized the user experience by developing an intuitive web interface that allows users to easily upload meal images, receive personalized dietary recommendations, and track their nutritional progress. Usability testing and feedback from users were instrumental in refining the interface to meet user needs effectively.

2. Advanced Technology Integration:

- Leveraging cutting-edge technologies such as computer vision, machine learning, and web development frameworks, the project achieved accurate food item identification, nutritional analysis, and personalized recommendation generation. Integration with external APIs for nutritional data retrieval further enriched the application's capabilities.

3. Data Security and Privacy:

- Robust security measures were implemented to safeguard user data, including encryption of sensitive information, secure authentication mechanisms, and regular security audits. These measures ensure user privacy and protect against unauthorized access or data breaches.

4. Continuous Improvement and Adaptation:

- The project adopted an iterative development approach, continuously incorporating user feedback, analyzing system performance metrics, and refining recommendation algorithms. This iterative process ensures that the application remains relevant, responsive to user needs, and capable of adapting to emerging dietary trends.

5. Impact on Health and Wellness:

- Ultimately, the goal of the dietary management project is to empower users to make informed dietary choices, improve their nutritional intake, and achieve their health and wellness goals. By providing personalized recommendations based on individual preferences and nutritional analysis, the application contributes to promoting healthier lifestyles and reducing the risk of diet-related health issues.

In conclusion, the dietary management project represents a significant advancement in leveraging technology to address the complex challenges of dietary management and nutrition. Through collaboration, innovation, and a commitment to user-centric design, the project has succeeded in delivering a valuable tool for individuals seeking to improve their dietary habits and overall well-being. As the project evolves and continues to incorporate advancements in technology and nutritional science, it holds the promise of making a meaningful impact on public health and wellness in the years to come.

CHAPTER 9
REFERENCE

REFERENCES:

1. R. Shanthakumari, C. Nalini, , et.al, 2022 International Mobile and Embedded Technology Conference (MECON).

2. Jingrong Chen, Hoa Xu, Jianqing Wc, Rui Yue, Changwei Yuan and LuWang, " Deer Crossing Road Detection With Roadside LiDAR Sensor " ,IEE Access,2019.

3. Elizaveta Dubrovinskaya , Fraser Dalgleish ,Bing Ouyang, Paolo Casari, " Underwater LiDAR Signal Processing for Enhanced Detection and Localization of Marine Life ",OCEANS - MTS/IEEE Kobe Techno-Oceans (OTO) ,2018.

4. Cristiano Premebida, Oswaldo Ludwig,and Urbano Nunes, " LIDAR and Vision-BasedPedestrian Detection System ", Journal of Field Robotics, 2009.

5. Chakaravarthi S, Visu P, Ganesan R, "An Object Detection and Tracking Techniques Using Superpixel Extraction in Compressed Video Sequences", International Journal of Pure and Applied Mathematics , Volume 119, 2018.

6. Charles R. Qi, Hao Su, Kaichun Mo, Leonidas J. Guibas," PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation ",IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

7. Joseph Redmon, Santosh Divvala, Ross Girshick and Ali Farhadi, "You only look once: Unified real-time object detection", IEEE Conference on computer Vision and Pattern Recognition(CVPR), June. 2016.

8. Wei Yang, Ji Zhang, Hongyuan Wang and Zhongbao Zhang, "A vehicle real-time detection algorithm based on YOLOv2 framework", Proceedings on Real-Time Image and Video Processing, vol. 10670, 2018.

9. Tibor Trnovszky, Patrik Kamencay, Richard Orjesek, Miroslav Benco and Peter Sykora, "Animal Recognition System Based on Convolutional Neural Network", Advances in Electrical and Electronic Engineering, vol. 15, no. 3, 2017.

10. Gyanendra and Pragya Gupta, "Wild Animal Detection Using Deep Convolutional Neural Network", Proceedings of 2nd International Conference on Computer Vision & Image Processing, pp. 327-338, 2018.

11. Yuvaraj Munian, Antonio Martinez-Molina and Miltiadis Alamaniotis, "Intelligent System for Detection of Wild Animals Using HOG and CNN in Automobile Applications", 11th International Conference on Information Intelligence Systems and Applications, 2020.

12. B. Karthikeya Reddy, Shahana Bano, G. Greeshmanth Reddy and Rakesh Kommineni, "Convolutional Network based Animal Recognition using YOLO and Darknet", 6th International Conference on Inventive Computation Technologies (ICICT), 2021.

13. H. Nguyen, S. J. Maclagan, T. D. Nguyen, T. Nguyen, P. Flemons, K. Andrews, et al., "Animal recognition and identification with deep convolutional neural networks for automated wildlife monitoring", IEEE International Conference on Data Science and Advanced Analytics (DSAA), 2017.

14. Hongchao Niu, Xiaobing Hu and Hang Li, "Improved YOLOv5 network-based object detection for anti-intrusion of gantry crane", 2nd International Conference on Control Robotics and Intelligent System, August 2021.

15. M.S. Norouzzadeh, D. Morris, S. Beery, N. Joshi, N. Jojic and J. Clune, "A deep active learning system for species identification and counting in camera trap images", Methods Ecol. Evol, vol. 12, pp. 150-161, 2021.

16. A. Yilmaz, G.N. Uzun, M.Z. Gurbuz and O. Kivrak, "Detection and Breed Classification of Cattle Using YOLO v4 Algorithm", In Proceedings of the 2021 International Conference on INnovations in Intelligent SysTems and Applications (INISTA) Kocaeli 25-27, pp. 1-4, 2021.

17. X. Yu, J. Wang, R. Kays, P. A. Jansen, T. Wang and T. Huang, "Automated identification of animal species in camera trap images", EURASIP J. Image Video Process, 2013, [online] Available: https://doi.org/10.1186/1687-5281-2013-52.

18. A. Dhillon and G. K Verma, "Convolutional neural network: a review of models methodologies and applications to object detection", Progress in Artificial Intelligence, vol. 9, no. 7553, pp. 85-112, 2020.

19. C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, et al., "Going deeper with convolutions", Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1-9, 2015.

Diet-Right: A Smart Food Recommendation System

Roshini V R[1,*], Priya R[1]
[1] Department of Artififcial Intelligence and Data Science
[2] Panimalar Engineering College,Chennai

Abstract

Inadequate and inappropriate intake of food is known to cause various health issues and diseases. Due to lack of concise information about healthy diet, people have to rely on medicines instead of taking preventive measures in food intake. Due to diversity in food components and large number of dietary sources, it is challenging to perform real-time selection of diet patterns that must fulfill one's nutrition needs. Particularly, selection of proper diet is critical for patients suffering from various diseases. In this article, we highlight the issue of selection of proper diet that must fulfill patients' nutrition requirements. To address this issue, we present a cloud based food recommendation system, called Diet-Right, for dietary recommendations based on users' pathological reports. The model uses ant colony algorithm to generate optimal food list and recommends suitable foods according to the values of pathological reports. Diet-Right can play a vital role in controlling various diseases. The experimental results show that compared to single node execution, the convergence time of parallel execution on cloud is approximately 12 times lower. Moreover, adequate accuracy is attainable by increasing the number of ants.

Keywords: Recommender System, Food, eHealth, ACO, Cloud Computing, Pathological Reports

1. Introduction

One of the major factors for a healthy life is daily diet and food, specifically, for the people suffering from some minor or major diseases. eHealth initiatives and research efforts aim to offer various pervasive applications for novice end users to improve their health [1]. Various studies depict that inappropriate and inadequate intake of daily diet are the major reasons of various health issues and diseases. A study conducted by World Health Organization (WHO) estimates that around 30% of the total population of the world is suffering from various diseases, and 60% deaths each year in children are related to malnutrition [2, 3]. Another study by WHO reports that inadequate and imbalanced intake of food causes around 9% of heart attack deaths, about 11% of ischemic heart disease deaths, and 14% of gastrointestinal cancer deaths worldwide [4]. Moreover, around 0.25 billion children are suffering from Vitamin-A deficiency [5], 0.2 billion people are suffering from iron deficiency (anemia) [6], and 0.7 billion people are suffering from iodine deficiency [7].

Generally, a person remains unaware of major causes behind deficiency or excess of various vital substances, such as calcium, proteins, and vitamins, and how to normalize such substances through balanced diet. Several works [8-15] proposed different recommendation systems related to food. These systems can be categorized as: (a) food recommendation systems [8, 9], (b) menu recommendations [11], (c) diet plan recommendations [12], (d) health recommendations for different diseases like diabetes and cardiovascular [13, 14], and (e) recipe recommendations [10, 15]. All the aforementioned systems provide recommendations to either some specific disease or balance the diet without considering information about any disease or nutrition deficiency in the body. For instance, in [8], a food recommendation system is proposed for the patients of diabetes. The system recommends various foods for diabetic patients without considering the diabetes level that may fluctuate frequently. Similarly, the authors in [9] do not consider the nutrition factors that have significant importance for a balanced diet recommendation.

Keeping in view the above mentioned facts and figures, it is of critical importance to maintain a balanced intake of food. However, it is quite challenging for a common person to keep track of personal food requirements because of the massive diversity of dietary components and items. A systematic food recommendation system is desired to recommend the appropriate food considering the disease of the person. The major challenge in designing such a system is the handling of greater volumes of data in terms of ingredients, quantity, nutrition facts, people's preferences, and simultaneously taking into consideration a person's pathological reports. The system must be scalable enough to handle recommendation queries from all over the globe. A solution to the aforementioned challenges is the use of cloud computing. Cloud computing is an innovative and emerging platform that enables users to perform on-demand scalability of computing and storage resources [16].

In this article, we present a cloud based food recommendation system called Diet-Right that considers the users' pathological tests results, and recommends a list of optimal food items. To achieve optimal results, we developed an algorithm based on Ant Colony Optimization (ACO). We designed a database of 345 pathological test reports and their normal ranges. A database was created by performing a field survey and collecting the information about pathological reports from different laborities [17, 18, 19]. The collected data was verified by a pathalogist of a hospital. Moreover, a database of 3,400 food items with 26 entries for most common nutrition was taken from the official website of composition of foods integrated dataset

(CoFID) [20]. Based on the real-time input of user's parameters, the Diet-Right recommends top ranked food items to the user.

The rest of the paper is organized as follows. Section 2 discusses related work. Section 3 presents the system model and recommendation process. Section 4 presents the experimental setup and results, and Section 5 concludes the paper.

2. Related Work

Several works have been proposed for different recommendation systems related to diet and food. These systems are used for food recommendations, menu recommendations, diet plan recommendations, health recommendations for specific diseases, and recipe recommendations. Majority of these recommendation systems extract users' preferences from different sources like users ratings [21, 22], recipe choices [23, 24], and browsing history [25, 26, 27]. For instance, in [24], a recipe recommendation system is proposed using social navigation system. The social navigation system extracts users' choices of recipes and in return recommends the recipes. Similarly, in [27], a recipe recommendation system is proposed that is capable of learning similarity measure of recipes using crowd card-sorting. The above mentioned recommendation systems lack in solving a common problem known as cold start problem. All these system must wait for the users to enter enough data for the effective recommendations [28]. Some of the commercial applications like [29, 30] offer users for a quick survey in order to get users preferences in a short time. For instance, the survey used by [29] is specifically designed to match the lifestyle of the user i.e., healthy, sportsman, pregnant, etc. The survey also attempts to avoid various foods which do not match the user's lifestyle. Similarly, a questionnaire is used by [30] through which a user answers different questions about his/her lifestyle, food preferences, nutrient intake, and habits. The system once extracts all the basic information is then able to recommend different meals for daily and weekly basis.

A Food Recommendation System (FRS) [8] is proposed for diabetic patients that used K-mean clustering and Self-Organizing Map for clustering analysis of food. The proposed system recommends the substituted foods according to nutrition and food parameters. However, FRS does not adequately address the disease level issue because the level of diabetes may vary hourly in different situations of the patient and the food recommendations may also vary accordingly. Tags and latent factor are used for android based food recommender system [9]. The system recommends personalized recipe to the user based on tags and ratings provided in user preferences. The proposed system used latent feature vectors and matrix factorization in their algorithm. Prediction accuracy is achieved by use of tags which closely match the recommendations with users' preferences. However, the authors do not consider the nutrition factors in order to balance the diet of the user according to his needs. Content based food recommender system [10] is proposed which recommend food recipes according to the preferences already given by the user. The preferred recipes of the user are fragmented into ingredients which are assigned ratings according to the stored users' preferences. The recipes with the matching ingredient are recommended. The authors do not consider the nutrition factors and the balance in the diet. Moreover, chances of identical recommendation are also present because the preference of the user may not change on daily basis. In [31], knowledge based dietary nutrition recommendation system is proposed for obesity. The recommendations include dietary nutrition and diet menus for individuals using collaborative filtering technique. An application for mobile users is also developed in order to recommend the dietary nutrition and menus to the users. Similarly, a food recommender system is proposed in [32] for patients in care facilities. The application is designed for caregivers in the care facilities in order to offer the food according to the patient preferences.

The above mentioned food recommendation systems are specifically dealing with some diseases or related to balance the diet. In case of food recommendation for specific diseases, the systems recommend different foods for patients without knowing the level of disease which may vary in different cases. Similarly, in case of food recommendations to balance the diet, nutrition factors are ignored which are very much important to recommend food and balance diet.

In medical practice, sometimes, pathological tests are required to identify a particular disease. A pathological test report usually indicates deficiency or excess of certain compounds/parameters in human body (e.g., levels of iron, calcium, or RBC count, etc.) which may cause particular disease. In this article, we present a novel food recommendation system specifically dealing with the pathological tests results. Our system considers diseases related to pathological reports, and most common nutrition factors in recommending the food items to the users. For this purpose, we used a database of 345 pathological test reports to categorize various diseases that occur due to the deviation from the normal ranges of compounds/parameters. Moreover, we designed a system that allows users to input values for a specific parameter. Based on the deviations of the input parameter value from the normal ranges, the system generates a diet plan that aims to cover those abnormalities. Furthermore, we used ant colony algorithm to train the system with the values of various parameters' ranges and diseases.

3. Diet-Right Model

The main focus of this work is to provide dietary assistance to different people who are suffering from common diseases. The proposed model recommends various foods and nutrition to the people based on their pathological test reports. Every pathological report has some indicators that are calculated based on the nature of the tests. For instance, if a doctor advised a patient to take pathological test of blood, then the common test entries include the values of hemoglobin, red blood cells (RBC), white blood cells (WBC), plasma, and sugar. Normal ranges of the aforementioned indicators are usually given in the test reports. In this way, the patient can identify the abnormalities after comparing with the normal ranges. In our proposed system, a user is provided with the complete list of the test parameters to make selection from. The user inputs the specific values oftest report in the selected parameters. We gathered the data of normal ranges for tests including blood (plasma and serum), urine, stool, cerebrospinal fluid (CSF), and gastric and secretion tests. A matrix of 345 entries was constructed. Each individual components of a test (e.g., blood test) have normal ranges with lower and upper bound. The ranges of the same component may differ on the basis of gender, age groups, and fasting or no fasting. Our system is trained on various types of age groups and their respective ranges of parameters. This allows the system to suggest diets as per needs of the users.

3.1 Diet-Right Architecture

In majority of the existing food recommendation systems, centralized architecture is used [8-15]. The main disadvantage of using such systems is scalability, when dealing with the massive amount of data. We propose a cloud based solution to offer the scalability and pervasiveness, where the smart phone users can conveniently access the recommendation system (see**Fig. 1**).

The model takes the input values as a first step. User enters the demographic data including gender and age as well as the value of the pathological test reports. These values are sent to cloud infrastructure in second step and are compared with the normal ranges that are stored in our database. In the third step, the abnormality level of the pathological test reports is computed. In the next step, the weight assignments and matrix generation process is carried out. In the fifth step, ranks are calculated for each food item and are sorted in descending order. In the sixth step, the user is provided the recommended list of food items.

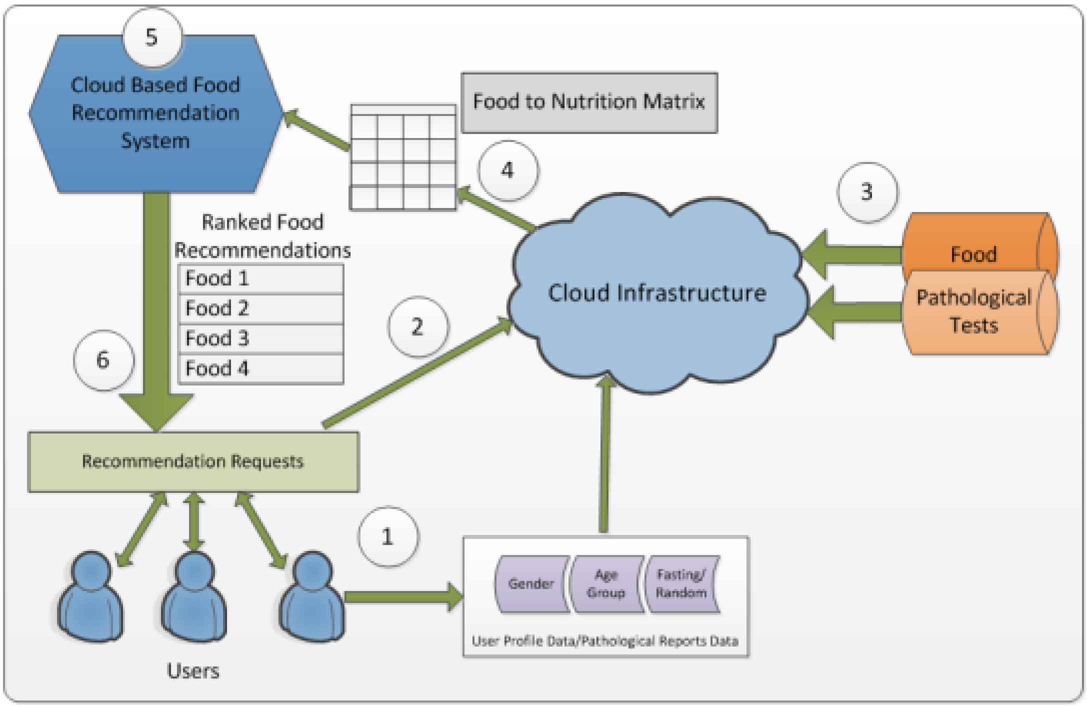

Fig. 1. Architecture of Diet-Right

3.2 Proposed Algorithm based on Ant Colony Optimization (ACO)

In this subsection, we present the food recommendation process using variant of ant colony approach on a graph of foods to generate the optimal food set for the users. In Diet-Right, we have used Ant Colony Optimization technique. ACO metahiuristic is a constructive and population based-approach which relies on the social behavior of ants. It is recognized as a most powerful approach for the solution of combinatorial optimization problems [33]. The main steps used in our proposed Algorithm are explained as follows:

Each food item is placed on nodes and a strongly connected graph is generated as shown in **Fig. 2.** Each link of graph has associated η and τ values, where τ is the randomly initialized pheromone, and η is the heuristic information initialized as the inverse of squared sum of difference of all the ingredients I. In Equation **(1),** k represents index for food ingredient, i and j represent ith and jth food item, I represents a single ingredient in a certain food item, and m is the total number of ingredients in a certain food item.

$$\eta_{ij} = \sum_{k=1}^{m} (I_{ik} - I_{jk})^2 \ . \tag{1}$$

Where, η is used to control exploration and exploitation of ACO and the values of $\eta \in (0,1)$.

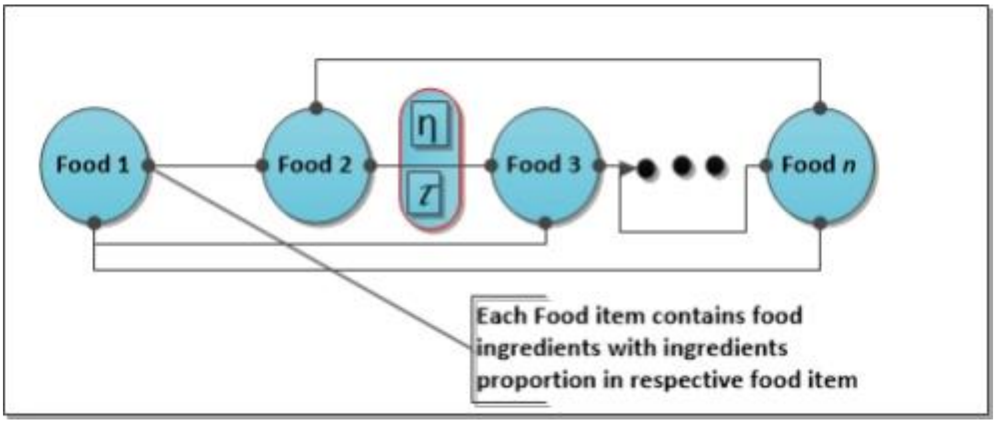

Fig. 2. Graph representation of the problem

After initialization, each ant constructs its local solution by visiting nodes which provide best cost in terms of low error compared to targets. Target vector represents the amount of food ingredients required against the particular disease. Target vector is predefined based on pathological reports, for instance, target vector for user with calcium deficiency may ranges from 9 to 10.5. The different nodes or food items are selected using transition rule which selects a path with highest transition probability. Transition probability is given by Equation **(2):**

$$p_i^k(t) = \begin{cases} \dfrac{[\tau i\ (t)]^{\alpha} \times [\eta i\ (t)]^{\beta}}{\sum_{r \in j^k} [\tau_r(t)]^{\alpha} \times [\eta_r\ (t)]^{\beta}}, & \text{if } i \in j^k \\ 0 & otherwise \end{cases} \tag{2}$$

Where, τi(t) represents the pheromone level at time t, ηi(t) is the heuristic information at time t , and a, β are the hyper parameters in the model used to weight heuristic information and pheromone level (used for fine tuning). Moreover, k represents ant, i represents initial node, j is the target node, r is index of current selected path, and j^k represents the solution.

When an ant selects a path among all existing paths (excluding the path in the solution), it updates the pheromone level locally as depicted in Equation **(3).**

$$\tau_{ij}^{k}(t+1) = \begin{cases} \tau_{ij}^{k}(t) \times \rho + \delta\tau^{k}, & \text{if } ij \in s_k(t) \\ (1-\rho)\,\tau_{ij}^{k}(t) & \text{otherwise} \end{cases} \tag{3}$$

Where, $\tau_{ij}^{k}(t+1)$ is new pheromone level that is increased by amount $\delta\tau_k$, and evaporation is governed by multiplication of pheromone decaying parameter p. Also, S is the solution of the k_{th} ant at time t.

Each ant provides locally optimized food set based on the nutrition expert recommendations, but here, we are interested in the globally optimized solution. As we are using supervised approach, we use Root Mean Square Error (RMSE) for the selection of globally best solution. To do so, we initialize global solution to EMPTY set and solve for each ant. Initially, solution returned by the first ant is considered as the global solution. Afterwards, when rest of the ants return with a solution, their RMSE is compared with the current global solution replacing the global solution with the solution having minimum RMSE value. For fast convergence of the solution, we update the pheromone level again using the same formula, but the update is only for the path that is globally optimal solution as depicted in Equation **(4).**

$$\tau_{ij}(t+1) = \begin{cases} \tau_{ij}^{g}(t) \times \rho + \delta\tau^{g}, & \text{if } ij \in s_g(t) \\ (1-\rho)\,\tau_{ij}^{g}(t) & \text{otherwise} \end{cases} \tag{4}$$

In food selection process, there is a need to select diversified foods to enhance the acceptance of foods among different people. We manage and update the heuristic information in such a way that the diversity among foods is maximized. For heuristic information update, we use Equation **(5).**

$$\eta_i = \frac{1}{m_i} \sum_{k=1}^{n} Y\left(s_k(t)\right)\left(1 + \varphi_i e^{\frac{-|s^k(t)|}{n}}\right), \text{ if } i \in s_t(t) \tag{5}$$

Where, m is the selected number of foods, Φ_i is the number of times a food is selected in whole iteration, and m, Φ_i are the parameters used to balance the solution in terms of local and global perspective. The used heuristic information faciliates in the selection of foods with minimal redundancy. Alorithm 1 presents food recommendation using ACO.

Algorithm 1: Food Recommendation using ACO

Input: Dataset (f, n) f foods, n nutrition, and T (n) prescribed nutrition plan

Output: Selection of Optimized Food Set R based T (n)

Set initial values of the heuristic information ηij, p and level of pheromone trails $\tau_{i,j}$ randomly and $Nants \leftarrow k$

$Sg \leftarrow \emptyset$

repeat

  Generate and randomly place ants at different nodes of the graph

  $k \leftarrow 1$

  for $k \leq Nants$ **do**

    Construct Solution sk for each Ant using transition and pheromone update

    while (all nodes are not visited) **do**

      Calculate transition probability using (2)

      With constraint $sk(t) \neq sk(t+1)$ **for** $k = 1, 2, 3, \ldots, n$

      Update pheromone locally τ_{ij}^{k} using (3)

    Add $\arg_{max}\{p\ (t)\}$ to $s^k(t)$

    end while

    if $(s^g = \emptyset)$ **then**

      $s^g(t) = s^k(t)$

    else if $(\mathrm{J}(T - \sum_{i=1}^{size(sg)} s_i^g)^2 \geq \mathrm{J}(T - \sum_{i=1}^{size(s^k)} s_i^k)^2)$ **then**

    $s^g(t) = s^k(t)$

    else

    do nothing

    end if

  end for

  Update pheromone globally τ_{ij}^{g} using (4)

  Update heuristic information n_i using (5)

  $E(T, sg) = \mathrm{J}(T - \sum_{i=1}^{size(sg)} s_i^g)^2$

until $E(T, s^g) \geq T_H$ or maximum iterations reached

Return s^g

3.3. Ranked List Generation

If a single item is selected, the foods are recommended based on parameters relevant to the item. For instance, in case of uric acid, purine quantity in the foods is considered. However, if multiple items are selected, then the following equation is used.

$$T_w = \sum_{i=1}^{n} p_i \ w_i \tag{6}$$

Where, T_w represents the total weight of the food item, p_i represents a single parameter (e.g., purine, carbohydrates, etc.), and w_i represents the importance of the food item (value lies between 0 and 1). The weight T_w of the food items facilitate to rank the foods based on the selected items. Food items consist of p_n nutrition and each pathological test indicates certain deficiencies of p_n in human body. Therefore, the weights w_i are introduced to compensate the tradeoff between different test substances. Weight assignment of different foods is stored in food-to-nutrition matrix where each row of the matrix represents a food plan. There are m number of food plans each having n numbers of food items in every food plan. The matrix generates ranked food plans which are used by the user.

4. Experimental Setup and Results

Extensive simulations are conducted to evaluate performance of the proposed system. The experimental setup and results are discussed as follows.

4.1. Experimental Setup

The details of the experimental setup and parameters used for evaluation are presented in **Table 1**.

Table 1. Experimental Setup

Parameters	**Values**
Total Number of Food Items	3400
Nutrition for Each Food Item	26
Total Number of Pathological Test Reports	345
Number of Ants used for Simulation	10-120
Maximum Iterations for each Ant	200
Simulation Tool	MATLAB
Single Node System Configuration	RAM 16 GB, Cores 4
Cloud Configuration	MATLAB Parallel Cloud, Cores 16

4.2. Results

We analyzed the behavior of proposed algorithm in terms of time complexity. We observed that increasing number of ants converge the solution to its minimum cost, but practically, it is not feasible to use high number of ants. Moreover, using high number of ants to contract a solution increases the time complexity as shown in**Fig. 3.**

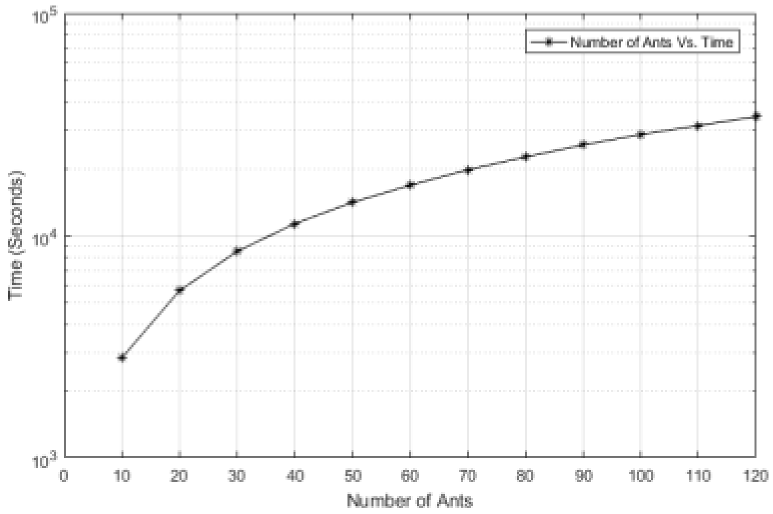

Fig. 3. Tradeoff between numbers of ants to time complexity

To select optimal number of ants for best results irrespective of time complexity, RMSE was estimated. It is observed that 110 ants produce lowest error rate. As our algorithm uses random initialization, it produced varied results. To address this abnormality, we used average of 10 executions with same settings. It can be observed that with increasing number of ants, RMSE is decreasing.**Fig. 4**shows average RMSE withrespect to increasing number of ants.

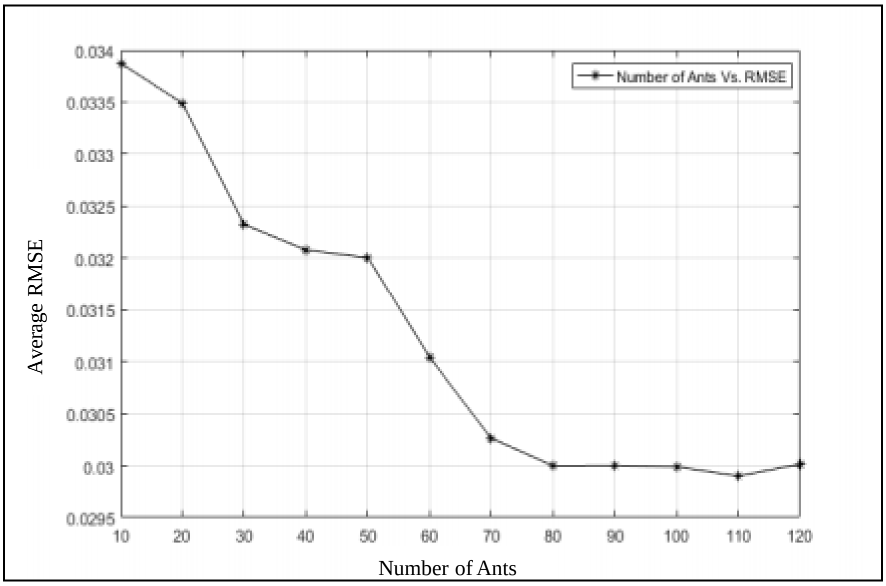

Fig. 4. Tradeoff between number of ants and RMSE

For selection of optimal number of ants, we performed best cost analysis for number of iterations versus number of ants. It is evident in**Fig. 5**that in our case, 80 ants provide best results in terms of converence of the solution.

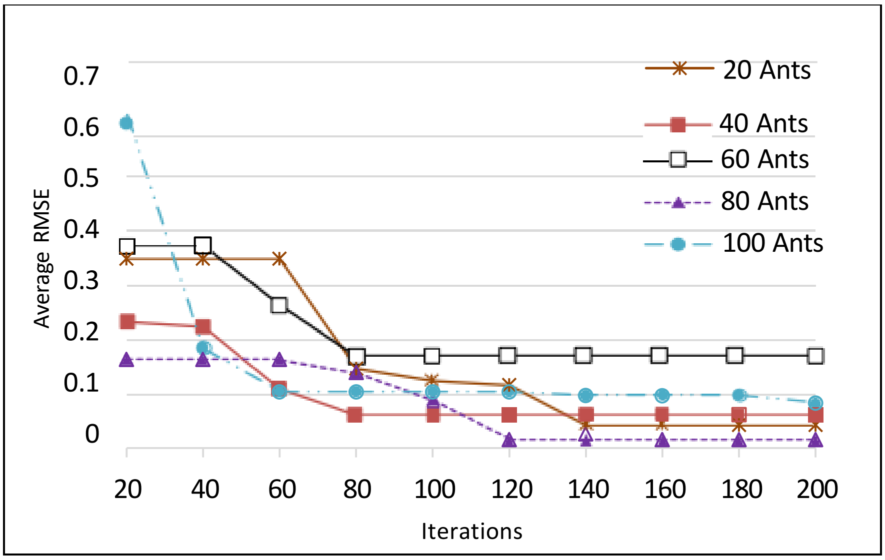

Fig. 5. Cost over Varying No. of Ants and Iterations

Fig. 6 presents the convergence time of different diseases. As shown in the previous result, 80 ants provide the best result in terms of convergence of the solution. Therefore, we have used 80 ants for the convergence time comparison between the most common diseases. The result shows that the convergence time for normal person is higher compared to persons with some disease. On the other hand, the convergence time to recommend foods for a hypertension patient is significantly lower compared to others. The reason for such variance is that the number of foods available for a normal/healthy person is much higher compared to the number of foods that are avilable for a patient.

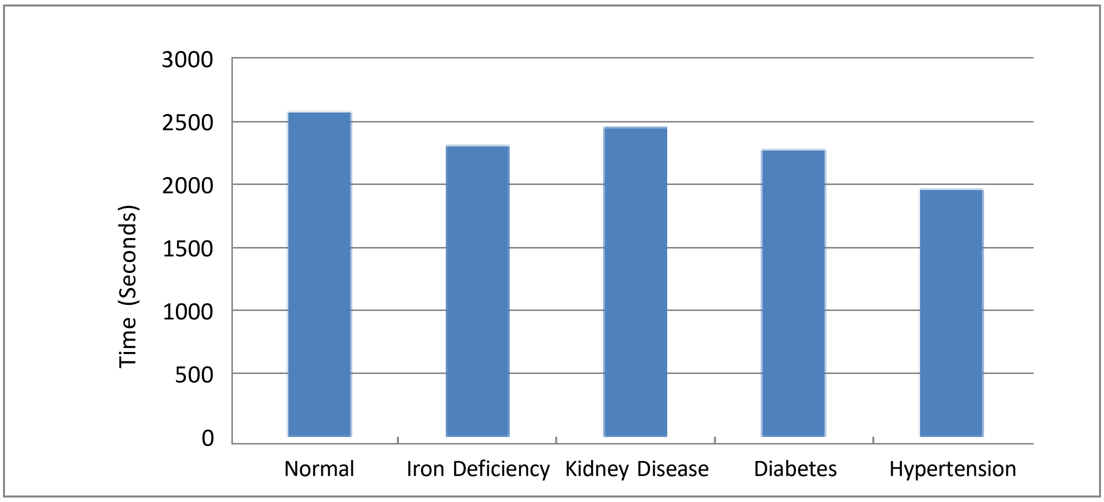

Fig. 6. Convergence Time of Different Diseases

Fig. 7 illustrates the cost comparison of common diseases. It can be seen that least cost is achieved for hypertension, whereas normal person exhibited highest cost compared to others. This shows that the dataset used in this study is more suitable for certain diseases, such as hypertension.

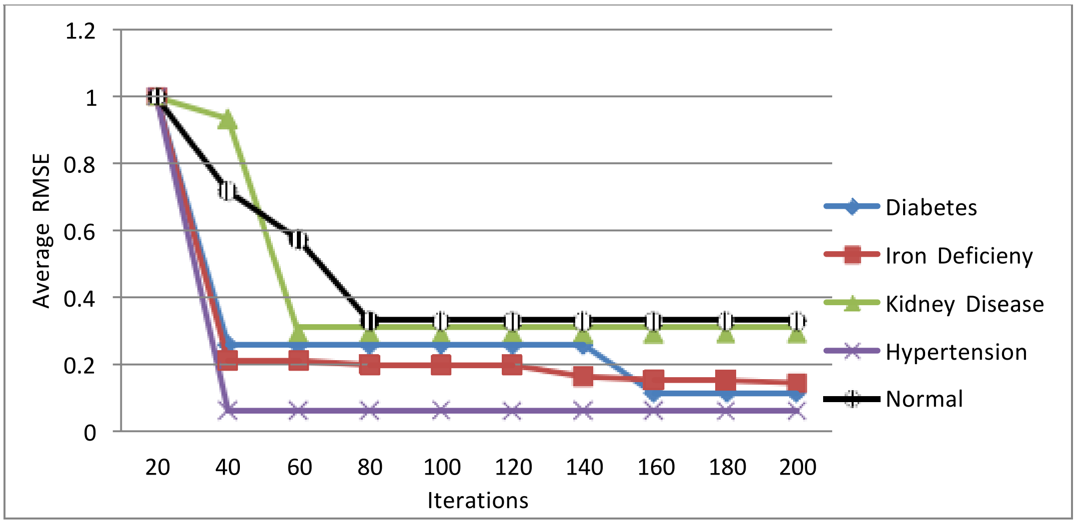

Fig. 7. Cost Comparisons of Diseases

Fig. 8 depicts the accuracy of recommendations relative to number of ants. The result shows that the highest accuracy is achieved with 110 ants. It is quite evident that when we increase the number of ants, the accuracy is also increased. Moreover, it is observed that the accuracy remains constant between 80 to 100 ants.

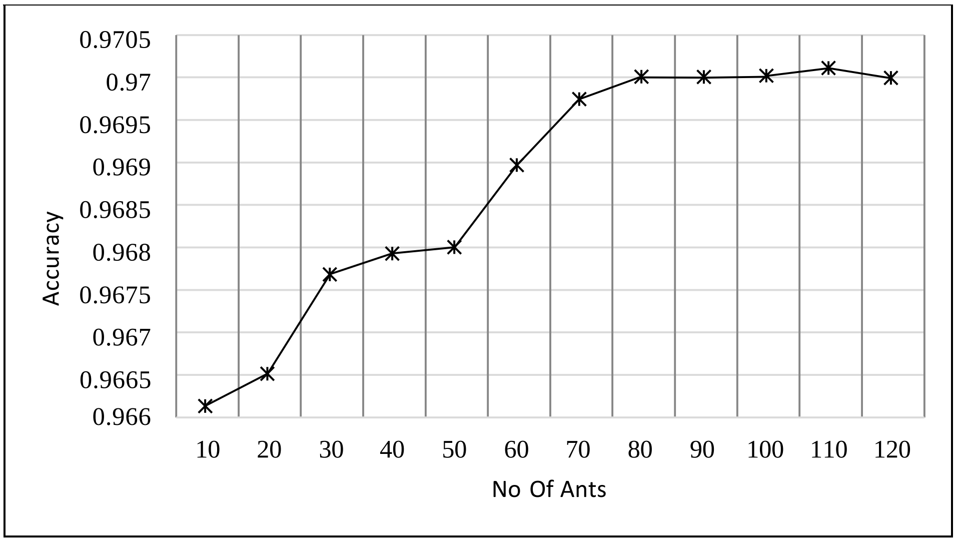

Fig. 8. Accuracy of Recommendations

Fig. 9 depicts the convergence time of single node and cloud-based execution. For this experiment, we executed our algorithm using Matlab's cloud framework [34]. It is evident from the result that the convergence time is significantly reduced with cloud-based execution. It is noteworthy that the convergence time of cloud based execution is approximately 12 times lower (on average) compared to single node execution.

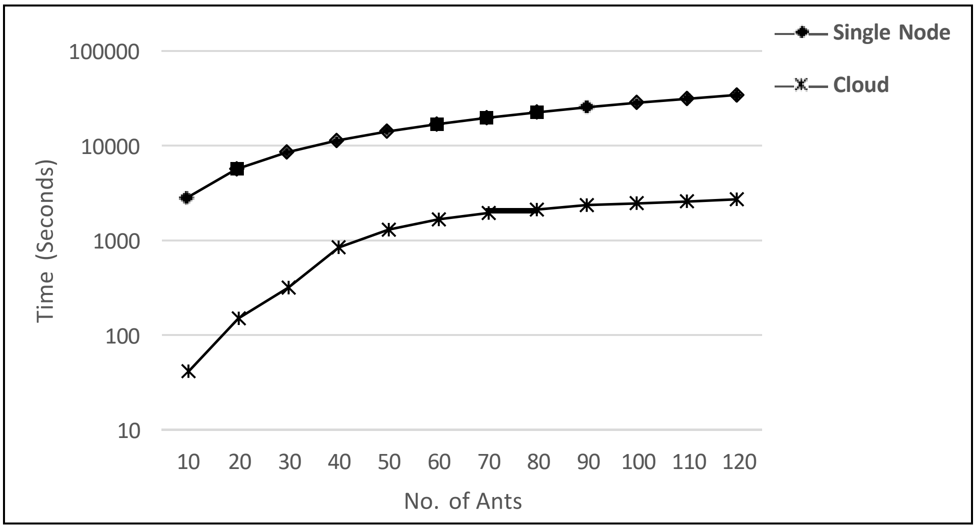

Fig. 9. Convergence Time of Single Node and Cloud based Execution

5. Conclusions

In this paper, we presented a cloud based food recommendation system called Diet-Right. Based on us er input, it recommends a list of optimal food items using an ACO model. Diet-Right manages and updates the heuristic information in such a way that the diversity among foods is maximized. Extensive expermentation was performed to check the cost, accuracy, convergence time, and performance gain. As a future research, we will focus on the recommendations breakdown for different timings of the day, such as breakfast, lunch, and dinner. Moreover, we will consider the amount of nutrition in different food items as per timing and daily needs of the patients. Furthermore, group food recommendation for family/friends is another interesting research area that can be explored.

References:

[1] Oh, Yoosoo, Ahyoung Choi, and Woontack Woo, "u-BabSang: a co ntext-aware food recommendation system," *The Journal of Supercomputing* 54, no. 1, 61-81, 2010. Article (CrossRef Link)

[2] A global agenda for combating malnutrition: progress report. Geneva, World Health Organization, 2000.Article (CrossRef Link)

[3] Childhood nutrition and progress in implementing the International Code of Marketing of Breast-milk Substitutes. Geneva, World Health Organization, 2002.Article (CrossRef Link)

[4] WHO, Joint, and FAO Expert Consultation, "Diet, nutrition and the prevention of chronic diseases," *World Health Organ Tech Rep Ser 916*, no. i-viii, 2003.Article (CrossRef Link)

[5] WHO/UNICEF. Global prevalence of vitamin A deficiency, MDIS Working Paper No. 2. Geneva, World Health Organization, 1995.Article (CrossRef Link)

[6] WHO/UNICEF/United Nations University. Iron deficiency anaemia assessment, prevention and control: a guide for programme managers, Geneva, World Health Organization, 2001. Article (CrossRef Link)

[7] WHO/UNICEF/International Council for the Control of Iodine Deficiency Disorders, Progress towards the elimination of iodine deficiency disorders (IDD). Geneva, World Health Organization, 1999.Article (CrossRef Link)

[8] Phanich, M., Pholkul, P., & Phimoltares, S., "Food recommendation system using clustering analysis for diabetic patients," in *Proc. of International Conference on Information Science and Applications*, pp. 1-8, IEEE, April 2010.Article (CrossRef Link)

[9] Ge, M., Elahi, M., Fernaández-Tobías, I., Ricci, F., & Massimo, D., "Using tags and latent factors in a food recommender system," in *Proc. of the 5th International Conference on Digital Health*, pp. 105-112, ACM., May 2015.Article (CrossRef Link)

[10] Freyne, J., & Berkovsky, S., "Evaluating recommender systems for supportive technologies," *User Modeling and Adaptation for Daily Routines*, pp. 195-217, Springer London, 2013. Article (CrossRef Link)

[11] Runo, M., "FooDroid: A Food Recommendation App for University Canteens," *Unpublished semester thesis*, Swiss Federal Institute of Theology, Zurich, 2011.Article (CrossRef Link)

[12] Su, C. J., Chen, Y. A., & Chih, C. W., "Personalized ubiquitous diet plan service based on ontology and web services," *International Journal of Information and Education Technology*, 3(5), 522, 2013. Article (CrossRef Link)

[13] Evert, A. B., Boucher, J. L., Cypress, M., Dunbar, S. A., Franz, M. J., Mayer-Davis, E. J., & Yancy, W. S., "Nutrition therapy recommendations for the management of adults with diabetes," *Diabetes care*, 37(Supplement 1), S120-S143, 2014.Article (CrossRef Link)

[14] LeFevre, M. L., "Behavioral counseling to promote a healthful diet and physical activity for cardiovascular disease prevention in adults with cardiovascular risk factors: US Preventive Services Task Force Recommendation Statement, *Annals of internal medicine*, 161(8), 587-593, 2014.Article (CrossRef Link)

[15] Teng, C. Y., Lin, Y. R., & Adamic, L. A., "Recipe recommendation using ingredient networks," in *Proc. of the 4th Annual ACM Web Science Conference*, pp. 298-307, ACM, June 2012. Article (CrossRef Link)

[16] Rittinghouse, J. W., & Ransome, J. F., "Cloud computing: implementation, management, and security," *CRC press*, 2016.Article (CrossRef Link)

[17] Shaukat Khanum Laboratory, https://shaukatkhanum.org.pk/, accessed December 13, 2016.

[18] Agha Khan Laboratory, https://www.aku.edu/labreports/Pages/default.aspx, accessed December 13, 2016

[19] Healthways Laboratory, http://www.healthwayslabs.net, accessed December 13, 2016.

[20] COFIDS Dataset, https://www.gov.uk/government/publications/composition-of-foods-integrated-dataset-cofid, accessed September 03, 2016.

[21] Forbes, P., & Zhu, M., "Content-boosted matrix factorization for recommender systems: experiments with recipe recommendation," in *Proc. of the fifth ACM conference on Recommender systems*, pp. 261-264, ACM, October 2011.Article (CrossRef Link)

[22] Freyne, J., & Berkovsky, S., "Intelligent food planning: personalized recipe recommendation," in *Proc. of the 15th international conference on Intelligent user interfaces*, pp. 321-324, ACM, February 2010.Article (CrossRef Link)

[23] Geleijnse, G., Nachtigall, P., van Kaam, P., & Wijgergangs, L., "A personalized recipe advice system to promote healthful choices," in *Proc. of the 16th international conference on Intelligent user interfaces*, pp. 437-438, ACM, February 2011.Article (CrossRef Link)

[24] Svensson, M., Höök, K., & Cöster, R., "Designing and evaluating kalas: A social navigation system for food recipes,"*ACM Transactions on Computer-Human Interaction (TOCHI)*, 12(3), 374-400, 2005.Article (CrossRef Link)

[25] Nutrino, URL: https://nutrino.co/, accessed October 10, 2016.

[26] Ueda, M., Asanuma, S., Miyawaki, Y., & Nakajima, S. "Recipe recommendation method by considering the user's preference and ingredient quantity of target recipe," in *Proc. of the International MultiConference of Engineers and Computer Scientists*, Vol. 1, 2014. Article (CrossRef Link)

[27] Van Pinxteren, Y., Geleijnse, G., & Kamsteeg, P., "Deriving a recipe similarity measure for recommending healthful meals," in *Proc. of the 16th international conference on Intelligent user interfaces*, pp. 105-114, ACM, February 2011. Article (CrossRef Link)

[28] Yang, L., Hsieh, C. K., Yang, H., Dell, N., Belongie, S., & Estrin, D., "Yum-me: Personalized Healthy Meal Recommender System," *arXiv preprint arXiv*, 1605.07722, 2016. Article (CrossRef Link)

[29] ShopWell, http://www.shopwell.com/, accessed October 10. 2016.

[30] Yummly, http://developer.yummly.com, accessed October 10. 2016.

[31] Jung, H., & Chung, K., "Knowledge-based dietary nutrition recommendation for obese management," *Information Technology and Management*, 17(1), 29-42, 2016. Article (CrossRef Link)

[32] De Pessemier, T., Dooms, S., & Martens, L., "A food recommender for patients in a care facility, in *Proc. of the 7th ACM conference on Recommender systems*, pp. 209-212, ACM, October 2013. Article (CrossRef Link)

[33] DeléVacq, A., Delisle, P., Gravel, M., & Krajecki, M., "Parallel ant colony optimization on graphics processing units," *Journal of Parallel and Distributed Computing*, 73(1), 52-61. 2013. Article (CrossRef Link)

[34] Matlab Parallel Cloud, http://www.mathworks.com/products/parallel-computing/matlab-parallel-cloud/, accessed October 10. 2016.

PANIMALAR ENGINEERING COLLEGE

An Autonomous Institution & Affiliated to Anna University, Chennai
Approved by AICTE, New Delhi, Accredited by National Board of Accreditation(NBA)
Approved by UGC for 2(f) & 12(B) Status
Bangalore Trunk Road, Poonamallee, Chennai-600123.

Certificate of Participation

PECTEAM 2K24

ICONIC

This is to certify that ~~Dr.~~/~~Mr.~~/Ms. ROSHINI V R of AI & DS from P.E.C has presented paper entitled In- Store Navigation using Open CV in 7th International Conference on Intelligent Computing(IConIC) held on 22nd & 23rd MARCH,2024.

FACULTY CO-ORDINATOR	CO - CONVENER	CO - PATRON
	Prof. S. VIMALA,M.TECH.,(Ph.D)	Dr.K.MANI,M.E.,Ph.D.,

IEEE MADRAS SECTION

IEEE COMPUTER SOCIETY