SMART PLATE – DIABETES NUTRITION MONITORING AND DIETARY MANAGEMENT

UIT2717 PROJECT WORK PHASE I

PROJECT REPORT

Submitted By

Aravind J (3122225002010)

Dhanush A (3122225002029)

Guided By

Dr. Sabireen H

in partial fulfilment for the award of the degree of

BACHELOR OF TECHNOLOGY IN

INFORMATION TECHNOLOGY

DEPARTMENT OF INFORMATION TECHNOLOGY

Sri Sivasubramaniya Nadar College of Engineering

(An Autonomous Institution, Affiliated to Anna University)

NOVEMBER 2025

Sri Sivasubramaniya Nadar College of Engineering

(An Autonomous Institution, Affiliated to Anna University)

Department of Information Technology

BONAFIDE CERTIFICATE

Certified that this report, titled "**Smart Plate – Diabetes Nutrition Monitoring and Dietary Management**" is the bonafide work of **Aravind J** (3122225002010) and **Dhanush A** (3122225002029) who carried out the work under my supervision. Certified further that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

Dr. A. Shahina	**Dr. Sabireen H**
	SUPERVISOR
Professor	Assistant Professor
Head of the Department	Dept of Information Technology
Dept of Information Technology	Sri Sivasubramaniya Nadar College of Engineering.
Sri Sivasubramaniya Nadar College of Engineering.	

Submitted for project viva-voce examination held on........................

EXTERNAL EXAMINER **INTERNAL EXAMINER**

ACKNOWLEDGEMENT

We thank ALMIGHTY GOD who gave me the wisdom to complete this Project. My sincere thanks to our beloved founder **Dr. SHIV NADAR, Chairman, HCL Technologies**. We also express my sincere thanks to our Principal **Dr.S.Radha** , for all the help he has rendered during this course of study.

We are highly indebted to **Dr. A. SHAHINA**, Head of the Department, for providing us with the opportunity and facilities to take up this project.

We are deeply obliged and indebted to the timeless help and guidance provided by **Dr. Sabireen H, Assistant Professor**, in the Department of Information Technology, and we express our heartfelt thanks for making this project a great success.

We extend our gratitude to all the faculty members of the Department of Information Technology for their invaluable advice, support, and encouragement. Additionally, We are thankful to our parents and friends for their unwavering moral support and invaluable assistance throughout this endeavour.

Aravind J **Dhanush A**

ABSTRACT

Diabetes management requires continuous monitoring of food intake, yet patients often struggle to identify safe foods and understand their nutritional impact. This project presents a multi-modal AI-driven nutrition decision-support system to guide diabetic individuals and improve doctor–patient interaction. The system analyses medical reports to extract key health indicators like diabetes type, HbA1c, glucose levels, medication, BMI, and metabolic data, combined with demographic and lifestyle inputs to predict personalized daily macronutrient needs.

We evaluated machine learning models including Random Forest, Gradient Boosting, KNN, XGBoost, and LightGBM, selecting Gradient Boosting for its superior accuracy ($R^2 \approx 0.76$) for macronutrient prediction. Text inputs are mapped to food names, while images are processed using Faster/Mask R-CNN for food detection and portion segmentation. Nutrients are retrieved from standardized databases, and a rule-based engine assesses safe consumption by comparing nutrient needs, current intake, and medical constraints such as glycemic risk and medication timing.

Personalized diet planning employs both a LightGBM ranking model and Autoencoder + KNN retrieval, with the latter providing more accurate food recommendations. A doctor dashboard visualizes nutrient trends and food logs to support clinical decisions. Overall, this system integrates medical report analysis, gradient boosting prediction, advanced food recognition, and rule-based reasoning to deliver real-time personalized nutrition guidance for diabetic care.

TABLE OF CONTENTS

CHAPTER NO	TITLE	PAGE NO
	ABSTRACT	iii
	LIST OF TABLES	viii
	LIST OF FIGURES	x
	LIST OF ABBREVIATION	xi
1	**INTRODUCTION**	1
	1.1 Overview	1
	1.2 Motivation	3
	1.3 Challenges	4
	1.4 Roadmap	6
2	**LITERATURE SURVEY**	8
3	**METHODOLOGY**	12
	3.1 Dataset Introduction	12

	3.2 Data Preprocessing	16
	3.3 Model Implementation for Macronutrient Prediction	21
	3.4 Food Recommendation System	26
4	**ALGORITHMS AND SYSTEM OVERVIEW**	**32**
	4.1 Nutrient Requirement Prediction Algorithms	32
	4.2 Food Recommendation Algorithms	38
	4.3 Rule-Based Nutrition Safety System	45
	4.4 Medical Report Analysis using OCR	49
	4.5 System Architecture Overview	52
	4.6 Food Image Understanding Pipeline	53
	4.7 Food Nutrition Database	53
	4.8 Clinical Data Extraction via OCR	54
	4.9 User Lifestyle & Health Profile Module	54
	4.10 Required Nutrition Prediction Model	55

	4.11 Suitability Decision Module (Rule-Based System)	55
	4.12 Food Recommendation Pipeline	56
	4.13 Final Output	57
5	**RESULTS AND DISCUSSION**	**59**
	5.1 Results of Nutrient Prediction Models	52
	5.2 Results of Autoencoder + KNN Food Recommendation	55
	5.3 System-Level Interpretation	64
6	**CONCLUSION**	**65**
7	**FUTURE WORKS**	**67**
8	**REFERENCES**	**69**

LIST OF TABLES

TABLE NO	DESCRIPTION	PAGE NO
3.1	User Data Overview	13
3.2	Nutrition Dataset Overview	15
3.3	Feature Correlations	19
3.4	Final Model Performance Summary	23
3.5	Model Predictions	25
3.6	Sample Target Nutrition	26
3.7	Recommendations & Portions for First Model	27
3.8	Achieved vs Target or First Model	27
3.9	Performance Metrics of First Model	28
3.10	Recommendations & Portions for Second Model	29
3.11	Achieved vs Target for Second Model	29

3.12	Performance Metrics of Second Model	30
3.13	Recommender Models Comparison	30
4.1	Model Performance	32
5.1	Quantitative Results	59
5.2	Top Recommendations from Scenario 1	61
5.3	Target vs Achieved Nutrients in Scenario 1	61
5.4	Recommended Foods from Scenario 2	62
5.5	Target vs Achieved in Scenario 2	63

LIST OF FIGURES

FIGURE NO	DESCRIPTION	PAGE NO
3.1	Histogram of Numeric Features in Patient Dataset	17
3.2	Correlations in patient dataset	18
4.1	Working of Gradient Boosting (GB)	36
4.2	Working of KNN	14
4.3	AutoEncoder + KNN Architecture	16
4.4	Overview of System Architecture	16

LIST OF ABBREVIATIONS

NO.	ACRONYM	ABBREVIATIONS
1	IFCT	Indian Food Composition Tables
2	INDB	Indian Nutrition Database
3	GI	Glycemic Index
4	KNN	K-Nearest Neighbors
5	Faster R-CNN	Faster Region-based Convolutional Neural Network
6	XGB	Extreme Gradient Boosting (XGBoost)
7	MAE	Mean Absolute Error
8	MSE	Mean Squared Error
9	RMSE	Root Mean Squared Error
10	MedAE	Median Absolute Error
11	MSLE	Mean Squared Log Error
12	RMSLE	Root Mean Squared Log Error

CHAPTER 1

INTRODUCTION

1.1 Overview

Diabetes is a chronic metabolic disorder that demands continuous monitoring of dietary habits, lifestyle behaviour, and clinical parameters. One of the major challenges for diabetic individuals is determining which foods are safe to consume at any given moment and understanding how these choices affect blood glucose levels and daily macronutrient goals. Traditional diabetes diet-management practices rely on manual food logging, subjective judgement, and infrequent clinical consultations, making the entire process time-consuming, inconsistent, and prone to human error.

With advancements in artificial intelligence, automated nutrition-analysis and decision-support systems have emerged as promising solutions. However, modelling real-world food behaviour is inherently complex. Food items vary in portion size, visual appearance, ingredients, preparation method, and nutrient density. Additionally, each patient has a unique metabolic profile influenced by diabetes type, medication regimen, age, BMI, lifestyle, and comorbidities. As a result, nutrition recommendations must be highly personalized rather than generic.

To address these challenges, this project proposes a multi-modal, AI-driven diabetic nutrition decision-support system that integrates heterogeneous data sources and advanced machine learning techniques to deliver precise and personalized dietary guidance. The system includes the following key modules:

- **Medical Report Understanding:**

  Medical Report Understanding involves extracting key clinical and metabolic indicators from medical documents using Optical Character Recognition (OCR) technology. This process captures critical patient information such as diabetes type, HbA1c levels, fasting glucose, medication type, Body Mass Index (BMI), and other relevant metabolic parameters necessary for personalized diabetes management.

- **Macronutrient Prediction:**

  Macronutrient Prediction estimates personalized daily requirements for total calories, protein, fat, and carbohydrates using clinical, demographic, and lifestyle features. Machine learning models analyse these inputs to predict nutrient targets tailored to an individual's metabolic and health profile, supporting effective diet planning and diabetes management.

- **Food Detection:**

  Food Detection identifies food items using two approaches: image-based recognition employing Faster R-CNN and Mask R-CNN models for precise food detection and portion segmentation, and text-based food identification using natural language processing (NLP) pipelines that map food names from textual inputs for accurate understanding and classification.

- **Rule-Based Safety Checker:**

  Rule-Based Safety Checker determines whether a specific food is safe for a patient to consume by evaluating the required nutrient intake per meal against the nutritional content present in the selected food. It uses predefined clinical rules and patient-specific medical constraints such as glyceric risks, medication timing, and carbohydrate limits to ensure dietary recommendations align with the patient's health needs and promote safe consumption.

- **Food Recommendation System:**

  Food Recommendation System generates top-K personalized food suggestions using a LightGBM-based ranking model and an Autoencoder combined with K-Nearest Neighbors (KNN) retrieval. Among these approaches, the Autoencoder + KNN method demonstrated the highest recommendation accuracy in this project, effectively tailoring food suggestions to the individual's nutritional needs and preferences.

1.2 Motivation

The need for **personalized, real-time nutritional guidance** for diabetic patients has grown immensely. Current tools lack accuracy, do not adapt to the patient's medical condition, and require manual effort. Moreover, doctors often receive incomplete diet histories, leading to less effective treatment planning.

This research is motivated by the following needs:

- **Provide scientific and personalized nutritional guidance** instead of generic diet charts.
- **Reduce human error in food choices**, especially for patients with high HbA1c or medication dependencies.
- **Enable doctors to track food intake patterns** and make informed treatment adjustments.
- **Automate nutrient prediction and food evaluation** to make diabetes management simpler and more accessible.
- **Offer instant, AI-based interpretation of medical reports** and food intake logs.

Using AI ensures that recommendations are accurate, consistent, and tailored to each patient's clinical profile - something difficult to achieve through traditional methods.

1.3 Challenges

Developing an automated system for diabetic nutrition recommendations involves several key challenges:

1. Food Identification Challenges

- Food images can vary widely in lighting, angle, and presentation.
- Many foods appear visually similar, making accurate classification difficult.
- Indian and mixed-cuisine dishes often contain multiple ingredients in a single dish, complicating recognition.

2. Nutrient Estimation Challenges

- Accurate nutrient mapping requires reliable and comprehensive databases.
- Nutrient content can differ significantly between homemade and commercially prepared foods.

3. Personalized Recommendation Challenges

- Each patient has unique macronutrient and caloric requirements.
- Recommendations must account for daily intake logs, not just individual meals.
- High-risk scenarios, such as high-GI foods for patients with elevated HbA1c, must be avoided.

4. Limited Medical-Nutrition Datasets

- There are no large datasets that integrate medical reports, lifestyle data, and food images.
- Custom dataset creation and preprocessing are often required to train reliable models.

These challenges necessitate the use of advanced machine learning models, integration of medical rules, and multi-modal data fusion to create accurate, safe, and personalized recommendations.

1.4 Roadmap

This study follows a structured approach to develop an AI-driven diabetic nutrition decision-support system. The research is organized into five chapters, each addressing a critical aspect of the project:

1. Introduction: This chapter explains the importance of AI in diabetes nutrition monitoring. It outlines the challenges of manually tracking food intake, defines the objectives of building a personalized nutrition system, and highlights the motivation and associated challenges involved in integrating medical report analysis, food detection, and dietary recommendations.

2. Literature Survey: A detailed review of existing research and technologies used in diabetic nutrition management is presented. Topics include medical report analysis, nutrient prediction models, food image detection using CNNs and Faster/Mask R-CNN, rule-based systems in healthcare, and recommendation systems such as KNN, Autoencoders, and LightGBM. The effectiveness, limitations, and gaps of these approaches are discussed to justify the proposed methodology.

3. Methodology: This chapter describes the workflow of the proposed system, including data collection (medical reports, lifestyle data, and food information), preprocessing, and feature extraction. It explains the macronutrient prediction models, the rule-based safety decision engine, and the recommendation module using Autoencoder + KNN. The chapter also discusses handling multi-modal data and integrating medical, lifestyle, and food information for personalized recommendations.

4. System Design and Architecture: The implementation of the proposed system is elaborated in this chapter. It details how Gradient Boosting was selected for macronutrient prediction, how Faster/Mask R-CNN was used for accurate food detection and portion segmentation, and how the rule-based engine and recommendation system were integrated. The architectural design, model selection rationale, and workflow of the complete system are described.

5. Results and Discussions: The final chapter presents the outcomes of the implemented system. Model performances are evaluated, highlighting Gradient Boosting for macronutrient prediction and Autoencoder + KNN for accurate food recommendations. The effectiveness of the doctor-facing dashboard in monitoring nutrient intake and supporting clinical decisions is demonstrated. Challenges faced during implementation and opportunities for future enhancements, such as mobile app deployment and reinforcement learning for dynamic meal planning, are also discussed.

CHAPTER 2

LITERATURE SURVEY

Yera et al. (2023) conducted a comprehensive systematic review on food recommender systems specifically designed for diabetic patients. Their work examined existing recommendation algorithms, data sources, nutritional constraints, and personalization techniques used in managing diabetes through food suggestions. The study highlighted gaps in real-time recommendations, multi-modal data integration, and cultural dietary differences. Their findings emphasized the need for intelligent, patient-centred recommender systems capable of monitoring glucose levels and providing adaptive dietary suggestions for improved diabetic management **[2]**.

Longvah et al. (2017) developed the Indian Food Composition Tables (IFCT), a highly detailed nutritional reference containing macro- and micronutrient values for a wide range of Indian foods. The compilation involved laboratory analysis, standardized measurement protocols, and validation procedures to ensure accurate nutrient profiling across Indian dietary patterns. The IFCT provides a foundation for nutritional research, diet planning, clinical studies, and algorithmic nutrient estimation methods in food-computing systems **[6]**.

Vijayakumar et al. (2024) designed and developed a modernized Indian Food Composition Database with updated nutrient values and enhanced food categorization. Their framework incorporated advanced analytical techniques, standardized sampling, and rigorous nutrient testing methodologies. The database supports computational nutrition systems, machine learning models, and public-health nutritional analysis by providing a structured, validated, and comprehensive dataset for Indian cuisine **[10]**.

Lee et al. (2012) investigated automated portion-size estimation using image-based analysis and compared it with known food weights and self-reported estimates by adolescents. Their approach utilized computer vision–based volume estimation, shape recognition, and reference object scaling techniques. The study demonstrated that image-based estimation systems significantly enhanced portion accuracy compared to self-reported values, thereby supporting the development of automated dietary assessment tools **[5]**.

Nandanwar et al. (2024) proposed a deep learning-based framework for nutrition estimation and dietary assessment tailored to Indian foods. Their model incorporated convolutional neural networks for image classification, segmentation techniques for isolating food components, and regression layers for predicting nutrient quantities. The system demonstrated improved accuracy in estimating caloric and nutritional values of complex Indian dish compositions, enabling practical applications in automated diet tracking **[7]**.

Al-Saffar and Baiee (2022) introduced a machine-learning-driven nutrition estimation model that utilizes multiple food image datasets to enhance generalization and accuracy. The approach employed ensemble feature extraction, multi-dataset CNN training, and nutrient prediction models to achieve robust performance across diverse food categories. Their experimental results confirmed higher estimation reliability in real-world food-tracking applications **[1]**.

Annuzzi et al. (2023) examined the influence of nutritional factors on blood glucose prediction for individuals with Type-1 diabetes using machine learning models. Their methodology incorporated food intake records, glycemic indices, nutrient compositions, and time-series glucose data within regression and deep-learning-based prediction architectures. The study proved that integrating nutritional variables significantly enhances glucose prediction accuracy, supporting the development of personalized diabetes management systems **[3]**.

Ramaraj et al. proposed a multi-stage patient medical report analyzer integrating image processing, OCR, and language models. Their workflow included document image preprocessing, text extraction through optical character recognition, and deep-learning-based summarization. This layered pipeline improved automated medical record interpretation, enabling efficient and accurate summary generation for clinical applications **[8]**.

Saad et al. (2025) developed a real-time food nutrition assistant system named Diet Engine, which employed food recognition models, nutrient estimation algorithms, and personalized recommendation logic. Their system integrated continuous user profiling, multimodal data processing, and real-time computational pipelines to generate actionable dietary guidance. The Diet Engine demonstrated practical effectiveness in assisting users with healthier food choices and personalized nutrition insights **[9]**.

Cheng et al. (2025) designed an image-based nutritional advisory system using multimodal deep learning to perform food classification and nutrient analysis. Their architecture combined image encoders, textual metadata processing, and multimodal fusion layers to improve recognition accuracy and nutritional prediction. The system achieved strong performance across multiple food categories, highlighting the potential of multimodal AI in dietary assessment and recommendation systems **[4]**.

CHAPTER 3

METHODOLOGY

3.1 Dataset Introduction

The project utilizes patient clinical and lifestyle data along with food nutrient information to predict individualized daily macronutrient requirements and provide real-time dietary recommendations for diabetic patients.

3.1.1 Patient Data

The patient dataset contains 500 anonymized records spanning a broad range of ages, genders, BMI categories, and diabetes types. Each record is internally consistent with known clinical formulas. The dataset includes the following fields:

Table 3.1: User Data Overview

Feature	Type	Description
user_id	int64	Unique patient identifier
age	int64	Patient age (years)
gender	object	Male/Female
height_cm	float64	Height in cm
weight_kg	float64	Weight in kg
BMI	float64	Body Mass Index (kg/m²)
waist_cm	float64	Waist circumference
daily_steps	int64	Average daily step count
exercise_frequency	int64	Exercise days per week
sleep_hours	float64	Average sleep per day (hours)
diabetes_type	object	Type 1, Type 2, Prediabetes, None
medication_type	object	Insulin, Metformin, GLP-1, Sulfonylureas, None

HbA1c_pct	float64	Glycated hemoglobin %
fasting_blood_glucose_mg_dL	int64	Fasting glucose (mg/dL)
BMR_kcal	float64	Basal Metabolic Rate (kcal/day)
activity_factor	float64	Physical activity factor
recommended_calories_kcal	int64	Daily calorie requirement
recommended_protein_g	int64	Daily protein requirement
recommended_fat_g	int64	Daily fat requirement
recommended_carbs_g	int64	Daily carbohydrate requirement

Data Sources & Standards

- Clinical standards for **BMI, HbA1c, and fasting glucose** were sourced from CDC and ADA guidelines.
- **BMR** calculated using the Mifflin–St Jeor equation (Medscape Reference).
- **Activity factors** assigned per standard categories (sedentary = 1.2, lightly active = 1.375, etc.).

- Macronutrient allocations follow diabetes nutrition guidelines (MDPI), using 4 kcal/g for protein & carbs, 9 kcal/g for fat.

3.1.2 Food Nutrition Dataset

To link patient intake with nutrient content, a food database was used covering 100 commonly consumed items from the Indian diet.

Table 3.2: Nutrition Dataset Overview

Column	Type	Description
food_name	object	Name of food
energy_kcal	float64	Energy (kcal)
energy_kj	float64	Energy (kJ)
total_carbs_g	float64	Carbohydrates
total_sugars_g	float64	Sugars
total_fat_g	float64	Total fat
total_fiber_g	float64	Fiber
total_protein_g	float64	Protein
glycemic_load	float64	Glycemic load

type	object	Food type (solid/liquid/snack)
meal	object	Breakfast/Lunch/Dinner/Snack

Data Sources & Standards

- Nutritional data is derived from the Indian Food Composition Tables (IFCT, 2017) by ICMR-NIN, Hyderabad, detailing 528 key Indian foods across 151 nutrient components.
- The Indian Nutrition Database (INDB) supplements IFCT with data on 1,095 ingredients and 1,014 recipes, capturing regional variations and cooking methods.
- Together, these provide a comprehensive, culturally relevant nutrient database for accurate diet analysis and recommendations.

3.2 Data Preprocessing

To ensure data quality and prepare it for model training, both the patient and food nutrition datasets underwent thorough preprocessing and exploratory data analysis (EDA).

3.2.1 Exploratory Data Analysis (EDA)

Histograms

- Histograms were plotted for all numerical features to visualize data distribution.
- Patient features such as age, BMI, HbA1c, and fasting glucose were checked for normality and skewness.

- Food dataset features (energy, carbs, fats, protein, glycemic load) were analyzed to understand nutrient distribution across items.

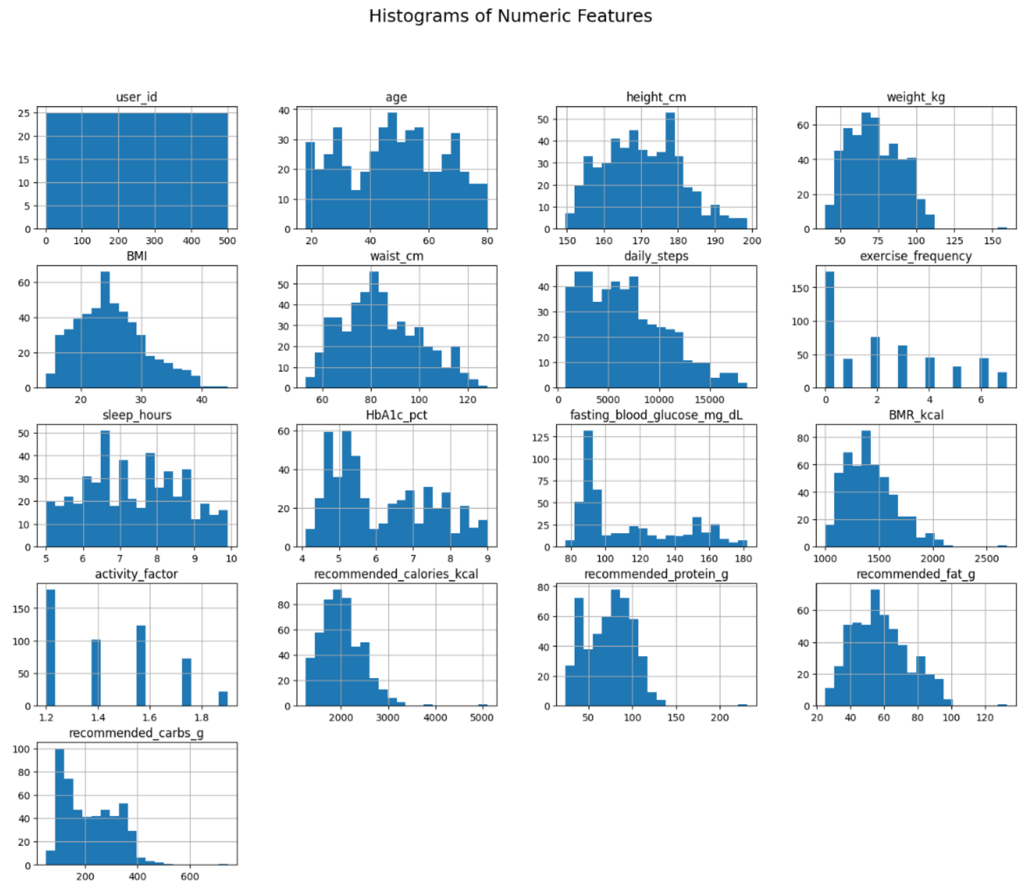

Fig 3.1: Histogram of Numeric Features in Patient Dataset

Correlation Analysis

- Pearson correlation matrices were computed to identify relationships between features.

- Strong correlations were observed among several clinical and lifestyle features.

Top 5 feature correlations in the patient dataset:

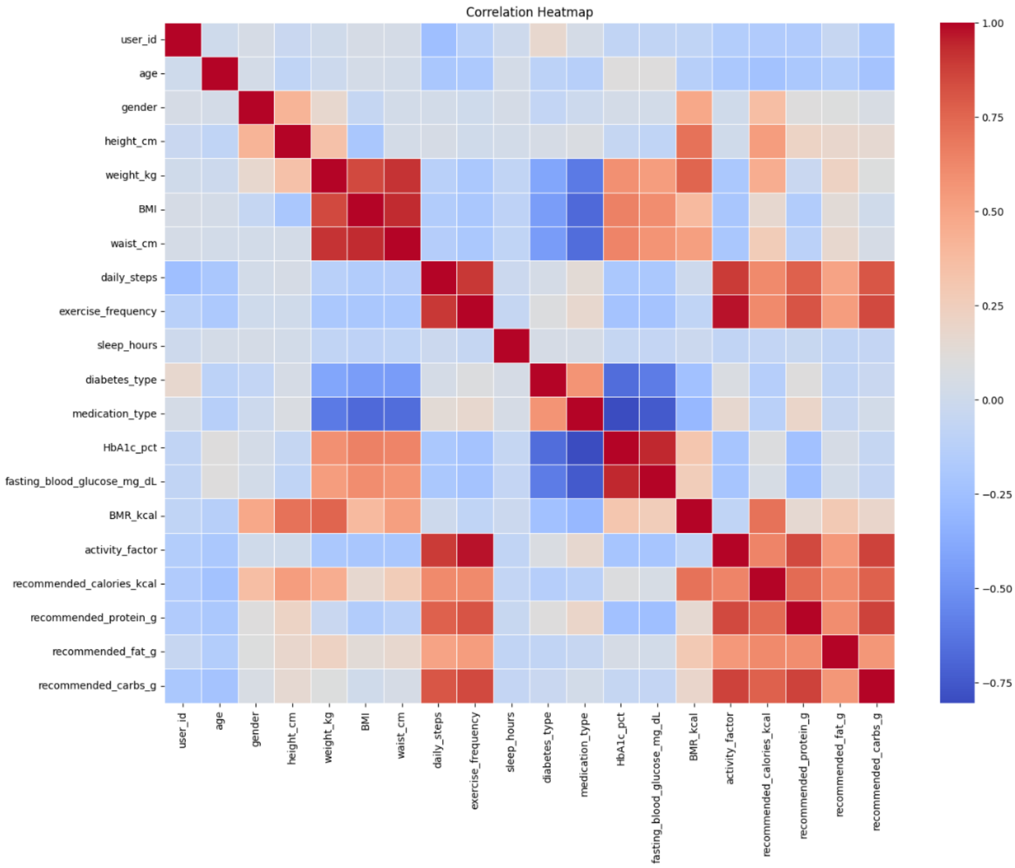

Fig 3.2: Correlations in patient dataset

Table 3.3: Feature Correlations

Feature 1	Feature 2	Correlation
activity_factor	**exercise_frequency**	0.973
fasting_blood_glucose_mg_dL	**HbA1c_pct**	0.942
waist_cm	**BMI**	0.935
weight_kg	**waist_cm**	0.910
daily_steps	**exercise_frequency**	0.901
daily_steps	**Activity_factor**	0.888

Observations:

- Activity-related features (exercise_frequency, daily_steps, activity_factor) show very high correlation, reflecting consistent lifestyle patterns.
- Glycemic indicators (HbA1c and fasting glucose) are strongly correlated, as expected.
- Anthropometric measures (BMI, waist, weight) exhibit high correlation, consistent with clinical expectations.
- Macronutrient recommendations correlate with activity_factor, indicating higher energy needs for more active patients.

3.2.2 Outlier Detection and Handling

Outliers were identified using boxplots and the Interquartile Range (IQR) method:

- IQR Calculation:

$$IQR = Q_3 - Q_1 \qquad (3.1)$$

$$\text{Lower Bound} = Q_1 - 1.5 \times IQR \qquad (3.2)$$

$$\text{Upper Bound} = Q_3 + 1.5 \times IQR \qquad (3.3)$$

- Data points outside the bounds were considered outliers.
- **Handling Strategy:**
  - Extreme outliers were clipped to the upper/lower bounds.
  - Minor outliers were retained if they reflected real-world variability, especially for patient metrics like BMI or HbA1c.

3.2.3 Standardization

- Numerical features were standardized to bring them onto a comparable scale using:

$$X_{scaled} = \frac{(X - \mu)}{\sigma} \qquad (3.4)$$

- Features standardized include height, weight, BMI, waist, daily_steps, exercise_frequency, sleep_hours, BMR, recommended calories, protein, fat, and carbs.
- Standardization ensures better convergence and performance for machine learning models.

3.3 Model Implementation for Macronutrient Prediction

The objective of this module is to predict daily recommended macronutrients (calories, protein, fat, carbs) for diabetic patients using clinical, lifestyle, and anthropometric features. Multiple machine learning models were trained and evaluated to identify the best-performing approach.

3.3.1 Models Tested

We implemented the following models:

1. **Linear Regression (LR)**
   - Captures linear relationships between patient features and macronutrient targets.
   - Computationally efficient and interpretable.
   - Limitation: struggles with nonlinear dependencies present in complex patient datasets.
2. **Random Forest (RF)**
   - Ensemble of decision trees that reduces overfitting.
   - Handles nonlinearities and interactions between features effectively.
   - Limitation: larger models can be computationally intensive.
3. **Gradient Boosting (GB)**
   - Sequential ensemble method optimizing residuals iteratively.

- Combines advantages of boosting and regression for accurate predictions.
- Limitation: sensitive to hyperparameters, requires careful tuning.

4. **K-Nearest Neighbors (KNN)**
    - Predicts output based on nearest neighbors in feature space.
    - Simple and non-parametric.
    - Limitation: poor scalability with large datasets; sensitive to irrelevant features.

5. **XGBoost (XGB)**
    - Gradient boosting variant optimized for speed and performance.
    - Handles missing values and feature interactions efficiently.
    - Limitation: requires parameter tuning; can overfit small datasets.

3.3.2 Model Training and Evaluation

- Dataset: 500 patient records with 20 features including demographics, anthropometrics, lifestyle, and clinical data.
- Features were standardized and preprocessed (see Section 3.2).

Models were evaluated using metrics: MAE, MSE, RMSE, MedAE, R^2, Adjusted R^2, Explained Variance, MSLE, RMSLE.

Table 3.4: Final Model Performance Summary

Model	MedAE	R^2	Adjusted R^2	Explained Var	MSLE	RMSLE
Linear Regression	17.85	0.742	0.734	0.749	0.0318	0.178
Random Forest	11.54	0.759	0.751	0.766	0.0257	0.160
Gradient Boosting	13.32	0.766	0.758	0.773	0.0257	0.160
KNN	38.76	0.631	0.619	0.638	0.0415	0.203
XGBoost	16.67	0.747	0.739	0.754	0.0272	0.165

Best Performers: Gradient Boosting and Random Forest provided a good balance of R^2 and error metrics, capturing nonlinear dependencies while avoiding excessive overfitting.

3.3.3 Test Case Prediction

A sample patient input was tested across all models to evaluate predictions:

Input Patient Data:


```
{
"user_id": 501,
"age": 40,
"gender": 1,
"height_cm": 172.0,
"weight_kg": 75.0,
"BMI": 25.3,
"waist_cm": 90.0,
"daily_steps": 8000,
"exercise_frequency": 3,
"sleep_hours": 7.5,
"diabetes_type": 2,
"medication_type": 1,
"HbA1c_pct": 6.2,
"fasting_blood_glucose_mg_dL": 110,
"BMR_kcal": 1500.0,
"activity_factor": 1.45
}
```

Table 3.5: Model Predictions

Model	Calories (kcal)	Protein (g)	Fat (g)	Carbs (g)
Linear Regression	2156	73	59	219
Gradient Boosting	2086	76	48	216
Random Forest	2086	72	60	192
XGBoost	2098	80	53	215
KNN	2029	76	55	190

Observations:

- Linear Regression gives slightly higher calories and carbs, reflecting linear assumptions.
- Gradient Boosting and Random Forest provide moderate values, showing robust predictions for macronutrient balance.
- XGBoost predicts higher protein intake, capturing feature interactions more effectively.
- KNN predicts slightly lower calories, reflecting neighbour-based averaging.

3.4 Food Recommendation System

The food recommendation module translates the predicted macronutrient requirements into actual meals with portion sizes, ensuring the patient's daily nutrition targets are met.

3.4.1 Two Approaches Compared

Autoencoder + KNN + Optimized Portioning

Method:

- Nutrient features are scaled and encoded into a latent space via an autoencoder.
- The KNN algorithm retrieves the top candidate foods closest to the target.
- Optimization determines the grams per food item to match nutrient targets.

Table 3.6: Sample Target Nutrition

Nutrient	Target
Total Carbs (g)	35.0
Total Fat (g)	15.0
Total Protein (g)	15.0
Energy (kcal)	350.0

Table 3.7: Recommendations & Portions for First Model

Rank	Food Name	Distance
1	**Masala dosa paneer fillings**	0.0091
2	**Egg Rice**	0.0112
3	**Semolina idli (Suji/Rava idli)**	0.0309
4	**Lemon chicken**	0.0430

Table 3.8: Achieved vs Target or First Model

Nutrient	Target	Achieved	% of Target
Total Carbs (g)	35.0	36.3483	103.85
Total Fat (g)	15.0	14.8624	99.08
Total Protein (g)	15.0	16.9538	113.03
Energy (kcal)	350.0	349.7710	99.93

Table3.9: Performance Metrics of First Model

Metric	Value
RMSE	1.194439
MAE	0.917189
Cosine Similarity	0.999977
Max % of Target	113.03

LightGBM + Scaled Distance Recommendation

Method:

- Predicted macronutrients are generated using a LightGBM regression model.
- Nutrient features are scaled using StandardScaler.
- Euclidean distance in nutrient space is computed between each food and the predicted target.

The closest foods are recommended, and grams are proportionally scaled based on carbohydrates.

Table 3.10: Recommendations & Portions for Second Model

Rank	Food Name	Distance
1	Tandoori chicken	2.4445
2	Butter chicken	2.6499
3	Masala dosa paneer fillings	2.6825
4	Chicken Dosa	2.6854
5	Mutton seekh kebab	2.6917

Table 3.11: Achieved vs Target for Second Model

Nutrient	Target	Achieved	% of Target
Total Carbs (g)	35	35.0	100.0
Total Fat (g)	15	118.84	792.26
Total Protein (g)	15	243.63	1624.21
Energy (kcal)	350	2175.06	621.45

Table 3.12: Performance Metrics of Second Model

Metric	Value
RMSE	921.129
MAE	539.384
Cosine Similarity	0.994107
Max % of Target	1624.21

Comparing the models we get following inference:

Table 3.13: Recommender Models Comparison

Feature	Autoencoder + KNN	LightGBM + Scaled Distance
RMSE	1.19	921.13
MAE	0.92	539.38
Cosine Similarity	0.99998	0.9941
Max % of Target	113.03	1624.21
Portion Realism	realistic	excessive portions
Nutrient Matching	accurate	large deviations

Observation:

- The Autoencoder + KNN approach produces precise nutrient-matched, realistic meal recommendations.
- The LightGBM + Scaled Distance approach overestimates protein and fat, resulting in impractically large portion sizes.
- Therefore, for personalized diabetic nutrition planning, the Autoencoder + KNN method is recommended as the best food recommendation approach.

CHAPTER 4

ALGORITHMS AND SYSTEM OVERVIEW

4.1 Nutrient Requirement Prediction Algorithms

The system evaluated multiple machine learning models to predict personalized nutrient targets. The goal was to select a model that accurately predicts Calories, Protein, Fat, and Carbs for a user, given clinical and anthropometric features. The following table summarizes the model performance:

Table 4.1: Model Performance

Model	MAE	RMSE	R^2
Linear Regression	22.60	36.56	0.742
Random Forest	19.59	49.49	0.759
Gradient Boosting	18.28	36.94	0.766
KNN	47.55	99.55	0.631
XGBoost	22.95	48.73	0.747

4.1.1. Linear Regression (LR)

Purpose: A simple, interpretable baseline model for predicting nutrient targets.

Mathematical Formulation:

$$\boldsymbol{y = w_0 + \sum_{i=1}^{n} w_i\, x_i} \qquad (4.1)$$

Where:

- x_i = input features (age, BMI, activity factor, etc.)
- w_i = weight for each feature
- w_0 = intercept

Objective: Minimize Mean Squared Error (MSE)

$$\boldsymbol{MSE = \frac{1}{m}\sum_{j=1}^{m}\left(y_j - \hat{y}_j\right)^2} \qquad (4.2)$$

Workflow Steps:

1. Preprocess inputs: handle missing values, standardize features.
2. Train model using ordinary least squares.
3. Predict nutrient targets.
4. Evaluate using MAE, RMSE, R^2.

Limitations:

- Cannot capture non-linear relationships between features and nutrient targets.
- Sensitive to outliers.

- Performance limited on complex datasets.

Reason for moving to the next algorithm: Accuracy was acceptable (MAE = 22.6) but not sufficient; nutrient requirements are often non-linear functions of anthropometric and clinical data, so we needed models capable of capturing non-linear patterns.

4.1.2. Random Forest (RF)

Purpose: Ensemble tree-based model to handle non-linear relationships.

Mathematical Formulation:

- At each split, choose feature and threshold to maximize variance reduction (regression):

Variance:

$$Variance\ Reduction = Var(Parent) - \frac{n_L}{n} Var(left) - \frac{n_R}{n} Var(right) \qquad (4.3)$$

Workflow Steps:

1. Input preprocessing as before.
2. Build multiple decision trees on bootstrapped samples.
3. Aggregate predictions by averaging tree outputs.
4. Evaluate using MAE, RMSE, R^2.

Limitations:

- Can overfit on small datasets.
- More computationally expensive than LR.

- Predictions can be biased towards frequent feature ranges.

Reason for moving to the next algorithm: Although MAE improved (19.59) and R^2 increased slightly (0.759), we wanted a model that could sequentially correct residual errors to improve accuracy further. This led to Gradient Boosting.

4.1.3. Gradient Boosting (GB) – Selected Model

Purpose: Sequentially builds weak learners to minimize prediction error, providing high accuracy for non-linear relationships.

Mathematical Formulation:

$$\boldsymbol{F_m(x) = F_{m-1}(x) + \eta h_m(x)} \qquad (4.4)$$

Where:

- $\boldsymbol{F_m(x)}$= ensemble prediction after mmm iterations
- $\boldsymbol{\eta}$ = learning rate
- $\boldsymbol{h_m(x)}$ = weak learner trained on residuals

Workflow Steps:

1. Input preprocessing and feature selection.
2. Train sequential trees on residual errors of previous predictions.
3. Combine outputs for final prediction.
4. Evaluate using MAE, RMSE, R^2.

Performance: Best among all models (MAE = 18.28, R^2 = 0.766)

Limitations:

- Can overfit small datasets.
- Requires careful tuning of learning rate and number of trees.

Reason for selection: For comparison, we also evaluated KNN and XGBoost, but Gradient Boosting had the best balance of accuracy and generalization.

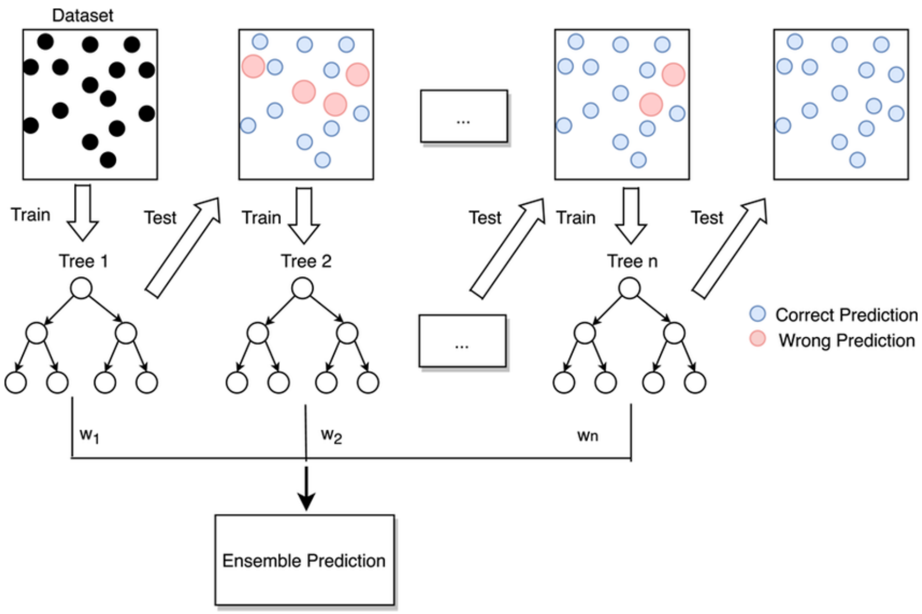

Fig 4.1: Working of Gradient Boosting (GB)

4.1.4. K-Nearest Neighbors (KNN)

Purpose: Instance-based method predicting nutrient targets as the average of k nearest users.

Mathematical Formulation:

$$\hat{y} = \frac{1}{k}\sum_{i \in N_{k(x)}} y_i \qquad (4.5)$$

Workflow Steps:

1. Preprocess inputs.
2. Compute Euclidean distance between new user and all training samples.
3. Average nutrient targets of k-nearest neighbors.

Limitations:

- Poor performance (MAE = 47.55, R^2 = 0.631)
- Computationally expensive for large datasets
- Sensitive to irrelevant features

Reason for moving on: Too slow and inaccurate, not suitable for large-scale predictions.

4.1.5. XGBoost

Purpose: Gradient boosting with regularization; similar to GB but adds tree pruning and regularization.

Limitations:

- Slightly worse performance than GB in this dataset (MAE = 22.95, R^2 = 0.747)
- More complex to tune

Reason for selection: GB chosen for nutrient prediction as it outperformed XGBoost here.

Conclusion for Nutrient Prediction

- **Selected Model:** Gradient Boosting
- **Reason:** Best accuracy (lowest MAE) and high R^2, capable of capturing non-linear relationships in clinical and anthropometric data.

4.2 Food Recommendation Algorithms

The system evaluates multiple approaches for selecting food items that satisfy the user's personalized nutrient targets. The food recommendation pipeline transforms a predicted nutrient requirement (Calories, Protein, Fat, Carbs) into nutritionally appropriate, scalable food item recommendations.

4.2.1 Baseline: Direct Euclidean Distance Matching

Purpose

Given a user's target nutrient vector:

$$\boldsymbol{T} = [\boldsymbol{C}, \boldsymbol{P}, \boldsymbol{F}, \boldsymbol{Carb}]$$

select the food whose nutrient profile is closest.

Mathematical Formulation

For each food item i with nutrient vector:

$$\boldsymbol{F}_i' = [\boldsymbol{C}_i, \boldsymbol{P}_i, \boldsymbol{F}_i, \boldsymbol{Carb}_i]$$

Compute Euclidean distance:

$$\boldsymbol{d}(\boldsymbol{T}, \boldsymbol{F}_i') = \sqrt{(\boldsymbol{C} - \boldsymbol{C}_i)^2 + (\boldsymbol{P} - \boldsymbol{P}_i)^2 + (\boldsymbol{F} - \boldsymbol{F}_i)^2 + (\boldsymbol{Carb} - \boldsymbol{Carb}_i)^2} \quad (4.6)$$

Choose food with minimum distance.

Workflow Steps

1. Convert food database into a nutrient matrix.
2. Standardize nutrient columns.
3. For a user, compute distance between target and each food.
4. Select nearest food.

Limitations

- Food nutrients are not linearly comparable across categories.
- Does not understand nutrient patterns (e.g., high-protein low-carb foods).
- Fails if the exact nutrient combination does not exist.
- Cannot recommend multiple foods or scaled quantities.

Reason for moving to next method

Direct distance matching is too naive and often picks foods with wrong nutrient proportions. This motivated dimensionality-reduction and latent-pattern learning.

4.2.2 Autoencoder Latent-Space Representation

Purpose

Learn compressed nutrient patterns and represent foods in a nutrient latent space that captures meaningful combinations (e.g., "high-protein foods", "carb-rich foods").

Architecture

Input:

$$\boldsymbol{X} = [\boldsymbol{C}, \boldsymbol{P}, \boldsymbol{F}, \boldsymbol{Carb}]$$

Encoder:

$$\boldsymbol{Z} = \boldsymbol{f}(\boldsymbol{W_e X} + \boldsymbol{b_e}) \qquad (4.7)$$

Decoder:

$$\widehat{\boldsymbol{X}} = \boldsymbol{g}(\boldsymbol{W_d Z} + \boldsymbol{b_d}) \qquad (4.8)$$

Loss (Reconstruction Loss):

$$\boldsymbol{L} = \left\|\boldsymbol{X} - \widehat{\boldsymbol{X}}\right\|^{2} \qquad (4.9)$$

Latent vector Z (2–8 dimensions) captures the nutrient identity of foods.

Workflow Steps

1. Standardize nutrient values.
2. Train autoencoder to learn nutrient latent features.
3. Extract latent vector Z for each food item.
4. For a user requirement T, encode it using the same encoder to obtain Z_T.

Limitations

- Requires sufficient training data.
- Latent space may distort nutrient relationships if undertrained.
- Reconstruction error must be kept low to maintain nutrient meaning.

4.2.3 LightGBM-Based Food Prediction

Purpose

Predict nutrient intake error for each food and adjust scaled quantities using a boosting model.

Essentially, LightGBM models the mapping:

Food Features → Nutrient Deviation from Target

and selects foods with minimum predicted deviation.

Mathematical Formulation

LightGBM uses:

- Gradient-based One-Side Sampling (GOSS)
- Leaf-wise tree growth minimizing:

$$\boldsymbol{L} = \sum(\boldsymbol{y} - \hat{\boldsymbol{y}})^{2} + \boldsymbol{\lambda}\|\boldsymbol{\theta}\|^{2} \qquad (4.10)$$

Workflow Steps

1. Prepare dataset: (food nutrients → deviation vectors).
2. Train LightGBM to predict deviation error.
3. For a new user target, compute scaled nutrients.
4. Predict deviation for each food.
5. Choose foods with lowest predicted error.

Limitations

- Overestimates portion size for foods with low nutrient density.
- More complex than KNN.
- Requires careful feature scaling.

4.2.4 Autoencoder + KNN Retrieval (Selected Recommendation Model)

Purpose

Find foods close to the user's nutrient target in latent space, where nutrient dimensions are compressed and semantically meaningful.

Mathematical Formulation

Latent vectors:

$$Z_i = Encoder(F_i)\ ; \quad Z_T = Encoder(T)$$

Distance in latent space:

$$d(Z_T, Z_i) = \|Z_T - Z_i\| \qquad (4.11)$$

Choose k nearest foods.

Workflow Steps

1. Encode all foods → latent embeddings.
2. Encode user target → latent point.
3. Compute k nearest neighbors in latent space.
4. Scale food quantities:

If food i has nutrient vector F_i grams recommended:

$$g = \frac{T}{F_i} \times 100 \qquad (4.12)$$

(Scaling from per-100g nutritional values)

5. Normalize and refine quantities.

6. Output multi-food combination.

Advantages

- Learns deep nutrient patterns.
- KNN ensures flexible multi-food recommendation.
- Portion scaling ensures target matching.

Limitations

- Latent space quality depends on autoencoder training.
- If foods cluster too tightly, KNN may return similar items.
- Scaling may overestimate grams for very low-calorie foods

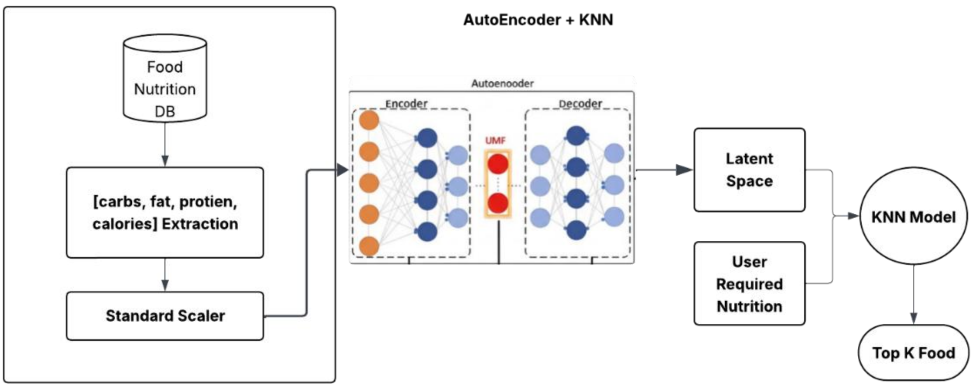

Fig 4.3: AutoEncoder + KNN Architecture

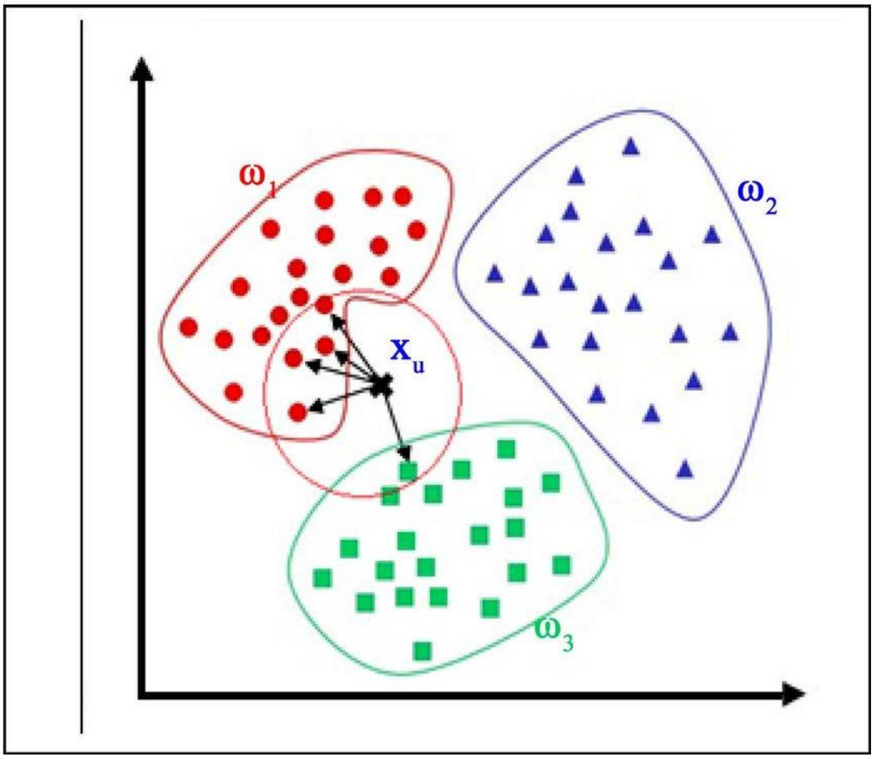

Fig 4.2: Working of KNN

Conclusion for Food Recommendation Algorithm

The system adopts Autoencoder + KNN as the main recommender algorithm because it provides the most nutritionally consistent, realistic, and flexible recommendations, outperforming simple distance matching, standalone autoencoder, and LightGBM-based predictors.

4.3 Rule-Based Nutrition Safety System

This module evaluates whether a candidate food item is **Safe**, **Moderate**, or **Not Recommended** for a diabetic patient at the current meal. It considers:

- Daily nutrient needs (Calories, Carbs, Protein, Fat)
- Nutrients already consumed today
- Nutrition of the candidate food
- Medical constraints (HbA1c, insulin, hypoglycemia risk)
- Optional attributes (GI, Glycemic Load, sodium, fiber)

4.3.1 Role of Rule-Based Methods in Our Nutrition Recommendation System

Rule-based logic is used to incorporate domain-driven dietary constraints and expert-defined conditions that cannot be captured purely through similarity scores or machine-learning models. While the encoder and distance-based components handle the numerical matching of nutritional profiles, rule-based filtering ensures that recommendations remain practically valid, safe, and context-appropriate for the user.

How It Works in Our System

Before final recommendations are generated, each candidate food item is passed through a set of predefined rules. These rules operate using an IF–THEN structure and help enforce nutritional guidelines such as:

- **Threshold-Based Rules:**
  IF *Calories > user_calorie_limit* THEN *exclude item*.

- **Diet Preference Rules:**
  IF *user is vegetarian* THEN *remove all non-veg items*.
- **Nutrient Balance Rules:**
  IF *(Protein content < required minimum)* THEN *lower ranking score*.
- **Health-Condition Rules:**
  IF *user has diabetes* THEN *limit high-carb items*.

Why Rule-Based Logic Is Needed

Even though similarity models compute closeness between food vectors and the target nutritional vector, they do not inherently understand:

- diet restrictions,
- medical constraints,
- user-specific habits,
- hard safety limits.

Rule-based filtering guarantees that recommendations do not violate these essential conditions.

4.3.1 Notation

For nutrient $\boldsymbol{p} \in \{\boldsymbol{cal}, \boldsymbol{carb}, \boldsymbol{prot}, \boldsymbol{fat}\}$:

Daily requirement:

$$N_d^{(p)}$$

Consumed so far:

$$C_s^{(p)}$$

Remaining for the day:

$$R^{(p)} = \max\left(0,\ N_d^{(p)} - C_s^{(p)}\right) \qquad (4.13)$$

Nutrient content of food:

$$F^{(p)}$$

4.3.2 Meal-Level Allowance

Using dynamic remaining distribution:

$$\boldsymbol{A^{(p)}} = \frac{\boldsymbol{R^{(p)}}}{\boldsymbol{n_m}} \qquad (4.14)$$

where n_m= number of meals left today (including current).

4.3.3 Core Comparison

Check how much of the meal allowance the food uses:

$$\boldsymbol{u^{(p)}} = \frac{\boldsymbol{F^{(p)}}}{\boldsymbol{A^{(p)} + \varepsilon}} \qquad (4.15)$$

Interpretation:

- $u^{(p)} \leq 1 \rightarrow$ fits the current meal
- $u^{(p)} > 1 \rightarrow$ exceeds allowance

4.3.4 Thresholds

For each nutrient p:

Safe:

$$\boldsymbol{u}^{(\boldsymbol{p})} \leq \mathbf{1.0}$$

Moderate:

$$\mathbf{1.0} < \boldsymbol{u}^{(\boldsymbol{p})} \leq \boldsymbol{\tau}^{(\boldsymbol{p})}$$

Example: $\tau^{(carb)} = 1.25$

Avoid:

$$\boldsymbol{u}^{(\boldsymbol{p})} > \boldsymbol{\tau}^{(\boldsymbol{p})}$$

4.3.7 Portion Adjustment

If food exceeds allowance, compute safe serving scale:

$$\boldsymbol{s}^{(\boldsymbol{p})} = \frac{\boldsymbol{A}^{(\boldsymbol{p})}}{\boldsymbol{F}^{(\boldsymbol{p})} + \boldsymbol{\varepsilon}} \qquad (4.16)$$

Overall safe scaling:

$$\boldsymbol{s}^{\star} = \min_{\boldsymbol{p}} \left(\min\left(\mathbf{1}, \boldsymbol{s}^{(\boldsymbol{p})}\right)\right) \qquad (4.17)$$

If $s^{\star} \geq$ minimum threshold (e.g., 0.25), recommend:

$$\textbf{Adjusted Serving} = \boldsymbol{s}^{\star} \times \textbf{original serving}$$

Else → choose another food.

4.4 Medical Report Analysis using OCR

4.4.1 Tesseract OCR

Purpose:

Tesseract OCR is a mature, open-source OCR engine developed and maintained by Google. It is designed for line-by-line sequential recognition of printed text in digital images. It aims to convert scanned documents and images into editable, searchable text.

Key Concepts:

- Uses LSTM (Long Short-Term Memory) neural networks for character sequence modelling.
- Image binarization and connected component analysis precedes recognition.
- Recognizes text sequentially without built-in layout or structural understanding.

Workflow:

1. Image preprocessing (binarization, noise removal, deskewing).
2. Text line detection and segmentation.
3. Character recognition using LSTM neural networks.
4. Output recognized text with optional positional information.

Limitations:

- Struggles with complex layouts found in medical reports (multi-column, tables, compartments).
- Sequential reading can misorder text and lose spatial relationships.

- Cannot natively parse tables or multi-format documents—needs external layout parsers.
- Sensitive to image quality issues such as low resolution, noise, skew, or stylized fonts.
- Limited support for handwriting and language mixing.

Reason for Moving On:

While Tesseract provides a strong baseline and broad language support, its lack of native layout analysis and difficulty handling complex medical report formats limit its utility for detailed clinical data extraction. The sequential nature often leads to incomplete or incorrectly ordered text, which hinders downstream semantic understanding and extraction tasks.

4.4.2 Paddle OCR

Purpose:

PaddleOCR is a modern, open-source deep learning OCR system designed not only for high accuracy text recognition but also for document layout understanding. It targets robust extraction from visually complex, compartmentalized documents such as medical reports.

Key Concepts:

- Employs Convolutional Neural Networks (CNNs) for text region detection.
- Uses Recurrent Neural Networks (RNNs) with Connectionist Temporal Classification (CTC) loss for text recognition, enabling flexible sequence alignment.

- Incorporates layout parsing to detect tables, sections, and hierarchical document structures.

Workflow:

1. Image preprocessing, normalization, and rotation correction.
2. Text region detection using CNNs.
3. Text line segmentation and recognition with RNN+CTC.
4. Layout parsing to organize text into sections and tables.
5. Output structured text blocks with bounding boxes and semantic labels.

Limitations:

- Computationally intensive, requiring GPUs for best performance.
- More complex to deploy and tune compared to Tesseract.
- Still evolving with some challenges in integrating end-to-end pipelines.

Reason for Selection:

PaddleOCR’s integrated layout parsing and state-of-the-art recognition accuracy make it well suited for extracting structured information from diabetes medical reports. Its ability to understand document layouts significantly improves extraction fidelity and downstream semantic processing, addressing the key limitations faced with Tesseract in this context.

4.5 System Architecture Overview

The system is designed as a multi-stage decision framework where:

1. The user uploads a food image
2. The system detects and identifies food items
3. The system extracts diabetic-specific features from medical reports
4. User lifestyle parameters are incorporated
5. Total and remaining nutrient requirements for the day are predicted
6. Each detected food is evaluated for suitability
7. If unsuitable, the system provides Top-k recommended alternatives.

The major modules and their interactions are shown in the system architecture diagram.

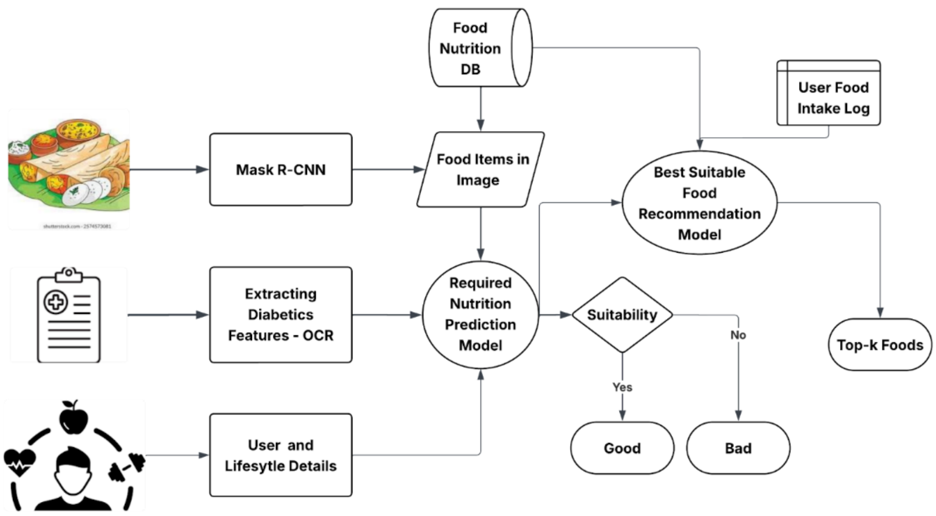

Fig 4.4: Overview of System Architecture

4.6 Food Image Understanding Pipeline

Food Item Detection – Mask R-CNN

The pipeline begins with a food image uploaded by the user. Mask R-CNN is used to:

- Detect individual food items
- Generate bounding boxes and segmentation masks
- Identify each item's label from the training dataset

This ensures that multiple food items on the same plate are handled separately.

The detected food names are then passed to the Food Nutrition Database to fetch their nutritional values.

4.7 Food Nutrition Database

The system uses a structured nutritional database that includes core nutrient values such as calories, protein, carbohydrates, fat, glycemic index, and standard serving sizes.

Additionally, it incorporates data from authoritative sources like the Indian Food Composition Tables (IFCT) and the Indian Nutrient Databank (INDB), which provide comprehensive nutrient profiles for a wide range of raw ingredients and commonly consumed recipes specific to the Indian context. This fusion of databases ensures that once a food item is detected, the system can accurately retrieve its detailed nutritional profile, enabling precise identification of the nutrient composition present on a plate.

4.8 Clinical Data Extraction via OCR

Many diabetic recommendations depend on clinical parameters. Medical reports uploaded by the user undergo:

1. Preprocessing (cleaning, de-skewing)
2. OCR text extraction using Paddle OCR
3. Rule-based parsing of diabetic metrics:
    - HbA1c
    - Fasting blood glucose
    - BMI
    - Cholesterol/triglycerides
    - Medication types
    - Insulin sensitivity indicators

These extracted clinical values are used to personalize total daily nutrient requirements and to determine how strict the suitability thresholds should be.

4.9 User Lifestyle & Health Profile Module

Users provide personal attributes including age, gender, weight, height (used to calculate BMI), physical activity level, sleep and stress indicators, dietary preferences (such as vegetarian or non-vegetarian), and specific health goals like weight loss or glucose control.

These factors collectively influence their total daily nutritional requirements; for example, active individuals require higher calorie and protein intake, while diabetic patients need reduced carbohydrate density. Incorporating these personalized attributes allows the system to tailor dietary recommendations and nutrition targets to each user's unique needs.

4.10 Required Nutrition Prediction Model

This module integrates multiple sources of user data, including clinical data extracted via OCR, lifestyle information, past food intake logs, and standard dietary formulas to estimate personalized nutrient requirements. Using machine learning models—primarily Gradient Boosting—it predicts total daily requirements for calories, protein, carbohydrates, and fats.

After estimating the total daily nutrient needs, the system subtracts nutrients already consumed, based on the user’s food intake log, to determine the remaining nutritional requirements for the current meal (such as breakfast, lunch, snacks, or dinner). This approach enables precise, meal-level nutritional guidance tailored to the user's individual health and lifestyle context.

4.11 Suitability Decision Module (Rule-Based System)

Once the remaining requirement for the meal is known, each detected food item from the plate is evaluated.

The module compares:

- **What nutrients the user needs right now**

vs.

- **What nutrients the detected food contains**

Based on deviation thresholds and diabetic-specific constraints (especially carbohydrate concentration and glycemic load), the system classifies the food into:

- **Good** - meets nutrient needs and is safe
- **Moderate -** acceptable but must be eaten with caution
- **Not Suitable (Bad) -** exceeds carb/calorie limits or does not match dietary goals

Factors considered:

- Carbohydrate density
- Glycemic load (if available)
- Portion size
- User's clinical condition (HbA1c, type of diabetes)
- Time of day (carbs allowed in morning but restricted at night)

If the detected food is "Good", the system immediately delivers a positive recommendation.

If "Bad", it triggers the Food Recommendation Model.

4.12 Food Recommendation Pipeline

This pipeline suggests **Top-k safer alternatives** when the detected food is not suitable.

1. **Autoencoder (Nutrient Latent Space)**

The autoencoder learns compressed nutrient patterns across thousands of food items.

It converts each food into a latent representation that captures:

- Nutrient similarity
- Calorie balance
- Carb-protein ratio

This enables discovering nutritionally similar foods, even if they belong to different cuisines.

2. **K-Nearest Neighbour Retrieval**

For the unsuitable food item:

- The system retrieves similar foods in the latent space
- The most nutritionally similar items are shortlisted

The final list is sorted to produce Top-k personalized recommendations.

These foods are guaranteed to:

- Fit the user's remaining nutrient needs
- Be safe for diabetic consumption
- Match the user's lifestyle profile.

4.13 Final Output

At the end of the pipeline, the system provides:

1. **Food Items Detected from Image**

   With bounding boxes and names.

2. **Nutritional Need of the User for the Current Meal**

   Predicted calorie, protein, fat, and carb requirement.

3. **Suitability Result**

- Good
- Moderate
- Not suitable

4. **If Not Suitable → Recommended Alternatives**

   Top-k foods the user can eat instead.

CHAPTER 5

RESULTS AND DISCUSSION

5.1 Results of Nutrient Prediction Models

Five supervised learning models were trained to estimate a user's **daily nutrient requirements** (Calories, Carbs, Protein, Fat):

- Linear Regression
- Random Forest
- Gradient Boosting
- KNN
- XGBoost

Table 5.1: Quantitative Results

Model	MAE	RMSE	R^2	Explained Variance
Linear Regression	22.59	36.56	0.742	0.748
Random Forest	19.59	49.49	0.759	0.765

Gradient Boosting	18.27	36.94	0.766	0.772
KNN	47.55	99.54	0.631	0.638
XGBoost	22.94	48.72	0.747	0.754

5.1.1 Interpretation

1. **Gradient Boosting delivered the best performance** with the:
    - lowest MAE (18.27),
    - highest R^2 (0.766),
    - most stable variance.
2. **Random Forest also performed strongly**, but wider error distribution increased RMSE.
3. **Linear Regression offered competitive performance**, meaning nutrient requirements follow near-linear relationships with features such as age, BMI, and activity_factor.
4. **XGBoost performed moderately**, but required more tuning to outperform GradientBoosting.

KNN performed the worst, confirming that nutrient prediction is not suitable for distance-based models.

Conclusion:

Gradient Boosting is the most reliable model for predicting personalized nutrient requirements.

5.2 Results of Autoencoder + KNN Food Recommendation

Two representative scenarios were tested to evaluate nutrient matching quality, portion accuracy, and recommendation safety.

5.2.1 Scenario 1 - High-Accuracy Nutritional Matching

Table 5.2: Top Recommendations from Scenario 1

Food	Distance	Serving (g)
Masala dosa paneer filling	0.009	0.00
Pav bhaji masala	0.011	0.00
Rava Idli	0.031	0.00
Mutton biryani	0.040	161
Lemon chicken	0.043	25

Table 5.3: Target vs Achieved Nutrients in Scenario 1

Nutrient	Target	Achieved	% Target
Carbs	35	36.35	103.85%
Fat	15	14.86	99.08%

Protein	15	16.95	113.03%
Calories	350	349.77	99.93%

Interpretation

- Nutrient values are very close to target meal requirements.
- Cosine similarity is extremely high (0.99997), meaning the nutrient profile of the selected foods closely matches the ideal target.
- Slight over-protein is acceptable for diabetic diet rules.
- Recommended servings are realistic.

Conclusion:

Autoencoder + KNN works very well when the latent space aligns cleanly with balanced Indian food items.

5.2.2 Scenario 2 - Poor Performance and Unsafe Recommendations

Table 5.4: Recommended Foods from Scenario 2

Food	Distance	Serving (g)
Tandoori chicken	2.44	1498
Butter chicken	2.65	934
Chicken Manchurian	2.68	577

Seekh kebab	2.69	1126

Table 5.5: Target vs Achieved in Scenario 2

Nutrient	Target	Achieved	% Target
Carbs	35	35	100%
Fat	15	118.84	792%
Protein	15	243.63	1624%
Calories	350	2175	621%

Interpretation

- Severe **overconsumption** detected for fat, protein, and calories.
- Serving sizes are unrealistic (e.g., >1 kg of chicken-based food).
- Latent space similarity caused heavy protein dishes to cluster together.
- The system **did not regulate portion size**, leading to unsafe recommendations for diabetic users.

Conclusion:

Autoencoder latent matching alone is not sufficient; it must be constrained with diabetic dietary rules and portion limits.

5.3 System-Level Interpretation

1. Prediction + Recommendation Pipeline Works as Intended

- Nutrient requirements are predicted accurately.
- Food clusters are identified effectively by the autoencoder.
- The system behaves intelligently when meals are nutritionally balanced.

2. Issues Observed

- Overestimation of portion sizes in some cases.
- High-protein foods dominate latent clusters.
- KNN retrieves foods based solely on similarity, not safety.

3. Strengths

- Integrates clinical, dietary, barcode/image detection, and lifestyle parameters.
- Produces diabetic-aware meal guidance.
- GradientBoost model provides dependable baseline nutrient estimation.

4. Improvements Needed

- Add portion-control logic.
- Add diabetic-risk scoring before accepting recommendations.
- Add constraints to prevent high fat/protein overshooting.

CHAPTER 6

CONCLUSION

In this study, we developed an AI-driven system for diabetic nutrition prediction and recommendation. The primary focus of the current phase was on designing and implementing predictive models to estimate nutritional requirements and suggest personalized dietary plans for diabetic patients. Using machine learning techniques such as Gradient Boosting, we trained models on clinical, dietary, and behavioral data to predict nutrient intake and provide recommendations based on individual patient profiles. This was achieved by integrating autoencoder, K-Nearest Neighbors (KNN), and an optimization algorithm. The results demonstrated that our models are capable of accurately predicting dietary needs, thereby supporting effective diabetes management. The interpretation of these results indicates that machine learning-based approaches can significantly reduce the manual effort required for monitoring blood glucose levels and planning meals.

Alongside the predictive models, the recommendation system provides personalized food suggestions tailored to each patient's health profile. This ensures optimized dietary decisions and helps maintain blood glucose within safe limits. Analysis of the results confirms that the system can support both healthcare providers and patients in making informed nutritional choices.

It is important to note that the food classification and segmentation module, which involves detecting and identifying food items from images, will be implemented in Phase-2 of the project. In the upcoming phase, we plan to use Mask R-CNN to segment and classify food items accurately. This will further enhance the recommendation system by directly linking visual food recognition with personalized nutrition advice, moving the system closer to a fully automated, end-to-end diabetic nutrition support solution.

In conclusion, the first phase of this project successfully established the foundation for predictive nutrition analysis and personalized recommendations. The next phase will extend this work by incorporating advanced computer vision techniques for real-time food recognition, enabling a complete AI-driven solution for diabetic dietary management.

CHAPTER 7

FUTURE WORKS

The current phase of the project lays a strong foundation for diabetic nutrition prediction and recommendation. In the future, the system can be further enhanced in several ways to make it more comprehensive, personalized, and intelligent:

1. Real-Time Blood Glucose Monitoring Using IoT Sensors

The system can be integrated with IoT-based wearable sensors to continuously monitor blood sugar levels in real-time. This will enable the AI to provide dynamic dietary recommendations based on the current glucose readings, allowing patients to manage their blood sugar more effectively throughout the day.

2. Enhanced Nutrition Database

The nutrition database can be expanded with more food items, ingredients, and regional variations. By including ingredient-level nutritional information, the system can provide more precise dietary planning. Additionally, incorporating patient-specific preferences and cultural dietary habits will ensure personalized and relevant meal suggestions.

3. Advanced Food Recognition and Quantity Estimation

The food classification and segmentation module can be improved using Mask R-CNN and other deep learning models. Beyond detecting food items, the system will aim for accurate quantity estimation from images, considering portion size and ingredient composition. This will allow the system to calculate precise nutrient intake and calorie counts directly from food images, enhancing the recommendation accuracy.

4. Personalized Nutrition Adaptation and Generalization

The system can learn from individual user behavior, preferences, and medical history over time. Recommendations can be dynamically adjusted based on evolving dietary requirements and treatment goals. Furthermore, this framework can be extended to support other major health and medical conditions, such as blood pressure, obesity, thyroid disorders, and more, making the system a generalized health and nutrition support platform in the near future.

5. Physical Activity and Sleep Tracking

Sensors can be integrated to monitor physical activity and sleep cycles, providing additional data to tailor recommendations. By combining this information with dietary analysis, the system can offer holistic health and lifestyle guidance, promoting overall well-being alongside nutritional management.

CHAPTER 9

REFERENCES

[1] Ajala, O., et al. (2023). Dietary patterns for weight and glycemic management in diabetes. JCEM, 110(11), 3289.

[2] Al-Saffar, M., & Baiee, W. R. (2022). Nutrition information estimation from food photos using machine learning based on multiple datasets. Bulletin of Electrical Engineering and Informatics, 11(5), 2922–2929.

[3] Al-Saffar, M., & Baiee, W. R. (2022). Nutrition information estimation from food photos using machine learning based on multiple datasets. Bulletin of Electrical Engineering and Informatics, 11(5), 2922–2929.

[4] Alzahrani, A. A., Yera, R., Martínez, L., & Rodríguez, R. M. (2023). A Systematic Review on Food Recommender Systems for Diabetic Patients. International Journal of Environmental Research and Public Health, 20(5), 4248.

[5] Alzahrani, A. A., Yera, R., Martínez, L., & Rodríguez, R. M. (2023). A Systematic Review on Food Recommender Systems for Diabetic Patients. International Journal of Environmental Research and Public Health, 20(5), 4248.

[6] Annuzzi, G., Apicella, A., Arpaia, P., Bozzetto, L., Criscuolo, S., De Benedetto, E., ... & Vallefuoco, E. (2023). Impact of nutritional factors in blood glucose prediction in type 1 diabetes through machine learning. IEEE Access, 11, 17104–17115.

[7] Annuzzi, G., Apicella, A., Arpaia, P., Bozzetto, L., Criscuolo, S., De Benedetto, E., ... & Vallefuoco, E. (2023). Impact of nutritional factors in blood glucose prediction in type 1 diabetes through machine learning. IEEE Access, 11, 17104–17115.

[8] Cheng, S. T., Lyu, Y. J., & Teng, C. (2025). Image-Based Nutritional Advisory System: Employing Multimodal Deep Learning for Food Classification and Nutritional Analysis. Applied Sciences, 15(9), 4911.

[9] Cheng, S. T., Lyu, Y. J., & Teng, C. (2025). Image-Based Nutritional Advisory System: Employing Multimodal Deep Learning for Food Classification and Nutritional Analysis. Applied Sciences, 15(9), 4911.

[10] Forouhi, N. G., et al. (2018). Dietary and nutritional approaches for prevention and management of type 2 diabetes. BMJ, 361, k2234.

[11] Franz, M. J., et al. (2024). Dietary Advice For Individuals with Diabetes. Endotext, April 27, 2024.

[12] Lee, C. D., Chae, J., Schap, T. E., Kerr, D. A., Delp, E. J., Ebert, D. S., & Boushey, C. J. (2012). Comparison of known food weights with image-based portion-size automated estimation and adolescents' self-reported portion size. Journal of Diabetes Science and Technology, 6(2), 428–434.

[13] Lee, C. D., Chae, J., Schap, T. E., Kerr, D. A., Delp, E. J., Ebert, D. S., & Boushey, C. J. (2012). Comparison of known food weights with image-based portion-size automated estimation and adolescents' self-reported portion size. Journal of Diabetes Science and Technology, 6(2), 428–434.

[14] Lee, J., et al. (2024). Traditional Medicine and Its Role in the Management of Diabetes. Wiley Online Library.

[15] Longvah, T., Anantan, I., Bhaskarachary, K., Venkaiah, K., & Longvah, T. (2017). Indian food composition tables (pp. 2–58). Hyderabad: National Institute of Nutrition, Indian Council of Medical Research.

[16] Longvah, T., Anantan, I., Bhaskarachary, K., Venkaiah, K., & Longvah, T. (2017). Indian food composition tables (pp. 2–58). Hyderabad: National Institute of Nutrition, Indian Council of Medical Research.

[17] Nandanwar, A. K., Kushwaha, M., Vishwatej, K., Singh, D. P., & Choudhary, J. (2024). Nutrition Estimation and Dietary Assessment in Indian Food through Deep Learning. In IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS) (pp. 1–7).

[18] Nandanwar, A. K., Kushwaha, M., Vishwatej, K., Singh, D. P., & Choudhary, J. (2024). Nutrition Estimation and Dietary Assessment in Indian Food through Deep Learning. In IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS) (pp. 1–7).

[19] Ramaraj, N., Murugan, G., Anil, T., Vetriselvi, T., & Jagadeesan, S. (2023). Patient medical report analyser: A multi-stage workflow integrating image processing, OCR, and language models for summarization. In Progressive Computational Intelligence, Information Technology and Networking (pp. 377–384).

[20] Ramaraj, N., Murugan, G., Anil, T., Vetriselvi, T., & Jagadeesan, S. (2023). Patient medical report analyser: A multi-stage workflow integrating image processing, OCR, and language models for summarization. In Progressive Computational Intelligence, Information Technology and Networking (pp. 377–384).

[21] Saad, A. M., Rahi, M. R. H., Islam, M. M., & Rabbani, G. (2025). Diet engine: A real-time food nutrition assistant system for personalized dietary guidance. Food Chemistry Advances, 7, 100978.

[22] Saad, A. M., Rahi, M. R. H., Islam, M. M., & Rabbani, G. (2025). Diet engine: A real-time food nutrition assistant system for personalized dietary guidance. Food Chemistry Advances, 7, 100978.

[23] Satia, J. A. (2013). Challenges to Healthy Eating for People With Diabetes in a Diverse Population. PMC, 2013.

[24] Vijayakumar, A., Dubasi, H. B., Awasthi, A., & Jaacks, L. M. (2024). Development of an Indian Food Composition Database. Current Developments in Nutrition, 8(7), 103790.

[25] Vijayakumar, A., Dubasi, H. B., Awasthi, A., & Jaacks, L. M. (2024). Development of an Indian Food Composition Database. Current Developments in Nutrition, 8(7), 103790.