Meal Nutrition Analysis

1st Ellika Mishra
Dept. of Computer Science
Texas A & M University
College Station, USA
ellikamishra@tamu.edu

2nd Mahima Bhatt
Dept. of Computer Science
Texas A & M University
College Station, USA
mahima19@tamu.edu

3rd Yash Honrao
Dept. of Computer Science
Texas A & M University
College Station, USA
yash.honrao@tamu.edu

***Abstract*—This study presents a novel multimodal deep learning approach for predicting calorie intake using continuous glucose monitoring (CGM) data, demographic information and meal images. We have implemented a comprehensive model that is integrating convolutional neural networks for image processing, bidirectional long-short term memory networks with attention mechanisms for CGM data analysis, and fully connected layers for demographic features. The model architecture is incorporating residual connections and dropout layers for enhancing performance and prevent overfitting. We have implemented a systematic hyperparameter tuning process using grid search to optimize the model's configuration. The effectiveness of our approach is demonstrated through various performance metrics, with the best model achieving a Root Mean Square Relative Error (RMSRE) of 0.35. This research contributes to the field of personalized nutrition by leveraging diverse data sources and advanced machine learning techniques, offering potential applications in dietary management and health monitoring.**

***Index Terms*—component, formatting, style, styling, insert**

I. Introduction

[1] In the recent past, nutrition research has shifted focus from individual nutrients to dietary patterns, meal composition and demographical information and data acquired from sensors, thereby taking a wholistic approach. By using Continuous Glucose Monitor, coupled with advancements in microbiome analysis and machine learning, we have opened new avenues for personalized nutrition research. Despite availability of plenty of data, accurately predicting an individual's calorie intake is still challenging. Traditional methods of dietary assessment often fall short due to recall bias and the complexity of human eating behaviors. [2] While previous studies have studied factors like demographics, glucose levels and food intake, there is a lack of comprehensive models that integrate multiple data sources and predict caloric consumption. The purpose of this study is to study various factors like CGM data, demographic information, and viome analysis an develop a multimodal Machine Learning model that predicts lunch calories intake. We have utilized continuous glucose monitor data, demographic information and breakfast meals along with their nutritional content (calories, carbohydrates, proteins) to train a Machine Learning model. This model aims at predicting the caloric content of subsequent lunch meals. The study's findings have the potential to significantly advance the research in the field of personalized nutrition. This approach could have far-reaching implications for diabetes prevention, weight management, and overall health improvement.

II. Data and Pre-processing

This work addresses the prediction of calorie consumption using multimodal data that includes continuous glucose monitoring (CGM), demographic data, and before-meal images of breakfast and lunch.

A. Continuous Glucose Monitoring (CGM) data

CGM is a method of tracking glucose levels in real-time using a small sensor inserted under the skin. CGM provides a continuous stream of glucose readings over time, enabling the analysis of patterns and trends in glucose fluctuations. These readings are crucial in understanding how food intake affects glucose levels, making CGM data an important predictor of calorie consumption.
The CGM data in this project is represented as sequences of glucose values collected for each subject across different days. These sequences are processed into fixed-length input vectors using padding or truncation.

B. Demographic data

This data represents quantitative clinical data, specifically patient-level health information collected in a structured format. It includes demographic information, anthropometric measurements, and various biomarkers related to metabolic health. It consists of demographic information such as Subject ID, Age, Gender, Race. There are several anthropometric measurements like weight, height, BMI, Other data consists of metabolic health markers like diabetes status, baseline fasting glucose, Insulin and other Lipid profile and Insulin Resistance measurements of individuals. It also contains microbiome analysis data.
Its significance lies in its ability to provide a holistic view of a patient's health status, enabling healthcare providers to assess various health risks, manage chronic conditions like diabetes, evaluate cardiovascular health, and develop personalized treatment plans. The data is invaluable for clinical research, epidemiological studies, and population health management. By offering a detailed snapshot of a patient's metabolic health, it supports evidence-based clinical decision-making, allows for the monitoring of treatment effectiveness, and contributes to improved patient outcomes.

C. Before-Breakfast and Before-lunch Image data

This dataset represents a longitudinal study of meal consumption patterns, capturing visual information about the food choices of 40 participants over a 10-day period. The data structure includes a Subject ID to uniquely identify each participant, a Day indicator to track the temporal aspect of the study, and two image fields: one for the meal before breakfast and another for the meal before lunch. This type of dataset is particularly valuable for nutrition research, allowing us to analyze dietary habits, portion sizes, and food composition visually without relying solely on self-reported data, which can provide more accurate insights into participants' actual eating behaviors and potentially reveal patterns or trends in food choices over time.

D. Data Pre-processing

Preprocessing of Continuous Glucose Monitoring (CGM) Data

For the preprocessing of CGM data, we used a custom function, `preprocess_cgm`. This function takes the raw CGM data as input to perform several operations. We first convert the `'CGM data'` column from string representation to a list of tuples, facilitating easier manipulation. The function then iterates through each row of the dataset, extracting individual glucose readings and organizing them into sequences. Concurrently, it maintains a record of the corresponding Subject ID and Day for each sequence.

The output of this function is twofold: a numpy array containing the processed CGM sequences, and a list of tuples holding the associated Subject ID and Day information. This preprocessing step is crucial for preparing the CGM data for subsequent analysis, ensuring that the temporal nature of the glucose readings is preserved while maintaining the ability to associate each sequence with its respective subject and day.

The `safe_eval` Function

The `safe_eval` function is designed to robustly handle the conversion of CGM data from various input formats to a consistent list representation. This function employs a multi-tiered approach to data processing:

- For string inputs, it attempts to evaluate the string as a Python expression using the `eval` function, which is particularly useful for converting string representations of lists to actual list objects.
- If the evaluation fails, the function catches the exception, logs an error message, and returns `None` to gracefully handle malformed data.
- For inputs that are already in list format, the function simply returns the input unchanged, preserving the existing structure.
- Finally, for any other data types, including `NaN` values or unexpected inputs, the function returns `None`.

This comprehensive approach ensures that the CGM data is uniformly processed, regardless of its initial format, thereby enhancing the robustness and reliability of subsequent data analysis steps.

Preprocessing and Visualization of CGM Data

The preprocessing and visualization of Continuous Glucose Monitoring (CGM) data is accomplished through two key functions: `preprocess_plot_cgm` and `plot_interactive_cgm`.

The `preprocess_plot_cgm` Function

The `preprocess_plot_cgm` function prepares the CGM data for visualization by applying the `safe_eval` function to the `'CGM Data'` column, ensuring robust handling of various input formats. It then groups the data by Subject ID, facilitating subject-specific analysis.

The `plot_interactive_cgm` Function

The `plot_interactive_cgm` function generates interactive time-series plots using `Plotly`, a powerful visualization library. For each subject, it creates a separate figure containing multiple traces, each representing a day's worth of CGM readings. The function iterates through the grouped data, extracting glucose values for each day and plotting them as separate lines. This approach allows for easy comparison of glucose patterns across different days for the same subject.

Interactive Plot Features

The resulting interactive plots feature customized layouts, including informative titles, axis labels, and legends. The use of Plotly's white template ensures a clean, professional appearance. These visualizations provide an intuitive and detailed representation of glucose fluctuations over time, enabling researchers to identify patterns, anomalies, or trends in the CGM data for each subject.

After pre-processing the CGM sequences, we mapped them with their corresponding nutritional labels. This process effectively filters the CGM sequences, retaining only those with matching nutritional data, and pairs each retained sequence with its respective lunch calorie information.

Preprocessing of Demographic data

First, we identified which columns in the dataset contain categorical information and which contain numerical information. Since the 'Viome' column contained multiple values separated by commas, we calculated an average value, representing the overall Viome score for each individual. We addressed the issue of missing data in the numerical columns by replacing any missing values with the average value for that particular column, ensuring that no data points are lost due to incomplete information.

Since the 'Race' column had multi-class categorical data, we performed One-hot encoding to transform the categorical information into a set of binary columns. Each new column represents a specific race category , with a '1' indicating that an individual belongs to that category and a '0' indicating they do not. This transformation makes the categorical data suitable to be used in further operations. Lastly, we applied scaling using StandardScaler to all the numerical data. This process adjusts the values in each column so that they have a similar scale, which is important for our Machine Learning model to work efficiently.

By performing these steps, the result is a fully processed demographic dataset where all the information is represented numerically and scaled appropriately.

Preprocessing of Meal Image data

For each row in the dataset, we extract the image information and append it to a list of sequences. Concurrently, we maintain a record of the corresponding Subject ID and Day, which are essential for maintaining the contextual integrity of the data. This approach ensures that each image is correctly associated with its respective subject and the day it was captured. The function returns two key elements: a pandas DataFrame containing the extracted image sequences, and a list of tuples holding the Subject ID and Day information for each image. We individually process the breakfast and lunch images. The resulting imgbrk and imglunch DataFrames contain the preprocessed image data for breakfast and lunch, respectively, while the subjectsdaysimg list maintains the corresponding subject and day information. We then mapped the pre-processed image data to the corresponding nutritional labels for both breakfast and lunch data, effectively creating aligned datasets where each image sequence is directly associated with its nutritional outcome.

Missing and corrupted image data was handled by a seperate Imputation class discussed in the later sections.

III. Methodology

For estimating the calorie consumption in lunch using multimodal data, we used three different models for each type of data. For images, we processed the images using CNNs. For the CGM Sequential data, we used Long-Short Term Memory model. And for the demographic data we used the FNN.

A. Convolutional Neural Network

We took a deep learning based approach for estimating calorie content from meal images using Convolutional Neural Networks (CNNs). The methodology consists of several key components:

Image Imputation Network

We defined a simple CNN to handle missing or corrupted data. The network consists of two convolutional layers, the first expands the input to 64 channels, applies a ReLU activation function. The second layer compresses it back to original number of channels. This imputation step helps to fill in or reconstruct missing parts of the input images, improving the robustness of the overall system.

Main Calorie Estimation Network

The CalorieNetWithImputation class contains the core architecture logic for calorie estimation. It performs feature extraction by adding a more complex CNN structure and by incorporating the imputation network. [4] The network processes two images: one taken before the meal and one after. Here's how it works:

1) Both images are first fed to the imputation network to handle any data inconsistencies.
2) The imputed images are then passed through a series of convolutional and pooling layers, gradually extracting higher-level features. We included dropout layers to prevent overfitting.
3) The features from both images are then flattened and concatenated, combining information from before the meal.
4) A final fully connected layer processes these combined features, producing a rich representation of the meal images.

In this way, the network learns the relevant features from the visual appearance of meals.

Dataset Handling

The CalorieMatrixDatasetWithSplit class handles the dataset, preparing it for training the neural network. It does the following:

1) It loads image data from a DataFrame of images stored as string representations of arrays.
2) Then these string representations are converted into actual image arrays, reshaping them to a consistent size and format.
3) If any image is missing, we replace it with a blank (zero) image to maintain consistency.
4) The images are transposed to the format expected by PyTorch (channels first).
5) We also mapped the corresponding calorie labels for each meal.

This dataset class makes sure the image data and labels are properly formatted and mapped for training the neural network.

B. Attention Mechanism

In the `attention_block` function, we implement an attention mechanism, a crucial component in modern deep learning models. Attention allows the model to focus on the most relevant parts of the input sequence, in our case, the CGM data. It works by assigning importance weights to different time steps in the sequence. We compute the attention scores for each time step, normalize these scores using a softmax activation, and then create a context vector by taking a weighted sum of the input based on these attention weights. This mechanism helps the model identify which parts of the CGM sequence are most informative for predicting calorie intake.

C. Model Architecture

The `build_combined_model` function defines the overall architecture of the multimodal deep learning model. The model is designed to process and integrate three types of input data:

1) **CGM Sequence Data**: We processed this time-series data using a combination of Bidirectional Long-Short Term Memory (LSTM) layers. We used two stacked

Bidirectional LSTM layers, each followed by a dropout layer to prevent overfitting. We implemented a residual connection between these LSTM layers, allowing the model to learn both simple and complex temporal patterns in the CGM data.
2) **Demographic Data**: We then processed Demographic information through a dense (fully connected) layer with ReLU activation. This allows the model to learn relevant features from the demographic data.
3) **Image Features**: Lastly, the Pre-processed features from meal images are being passed through a dense layer with ReLU activation function. This component enables the model to incorporate visual information about the meals.

The output that we got from these three streams, we combined them using a concatenation layer. Then we further processed it using additional dense layers with dropout for regularization. The final layer then outputs a single value, representing the predicted calorie of the lunch meal.

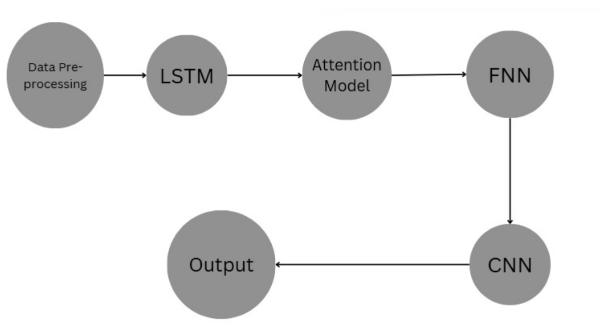

Fig. 1. Model Architecture.

D. Model Compilation

We compiled the model using the Adam Optimizer, which is known for its effectiveness in training deep neural networks. We considered learning rate as 0.01, and applied gradient clipping (clipnorm = 1.5) to prevent exploding gradients. Weight decay (1e-5) is also used for additional regularization. The model minimizes the Mean Squared Error (MSE) loss, which is appropriate for regression tasks like calorie prediction. We also used Mean Absolute Error (MAE) as an additional metric to evaluate the model's performance.

IV. Experimentation

A. Hyperparameter Tuning and Model Training

For optimizing our model's performance, we implemented a systematic approach for hyperparameter tuning and model training. This process, encapsulated in the `tune_and_train` function, allows us to explore a range of model configurations and training parameters efficiently.

B. Hyperparameter Grid Search

For exploring various combinations of hyperparameters we utilized a grid search methodology. We employed the ParameterGrid from scikit-learn to generate all possible combinations from a predefined parameter grid. This approach ensures a thorough exploration of the hyperparameter space, that allowed us to identify the most effective configuration for our specific task of calorie prediction.

C. Training Process

In order to efficiently train our model:

1) **Early Stopping**: We implemented an early stopping mechanism to prevent overfitting. Using this callback we monitored the validation loss and halted training if no improvement was observed over a specified number of epochs. Then the best weights are restored, ensuring we retain the model's optimal state.
2) **Learning Rate Scheduling**: We used `ReduceLROnPlateau` callback for dynamically adjusting learning rate. This technique reduces the learning rate when the validation loss plateaus, allowing for fine-tuning of the model's parameters in later stages of training.

D. Model Evaluation and Selection

For every configuration of hyperparameters, we trained the model on the training data and evaluated its performance on the validation set. Based on the validation loss, we selected the best model, by keeping a track of the best-performing model, its associated hyperparameters, and training history.

After exploring all hyperparameter combinations, we identified the configuration that yielded the lowest validation loss. We then evaluated the best-performing model on the test set.

In the table 1 you can see, few of the hyperparameters tested and the corresponding Validation Loss and RMSE.

V. Results

A. Performance Metrics

To comprehensively evaluate model performance, we used:

1) **Mean Squared Error (MSE)**: Used as the primary loss function during training.
2) **Mean Absolute Error (MAE)**: Provides an interpretable measure of prediction accuracy.
3) **Root Mean Square Error (RMSE)**: Offers a scale-dependent measure of prediction accuracy.
4) **Root Mean Square Relative Error (RMSRE)**: Provides a scale-independent measure of relative prediction accuracy, particularly useful for our calorie prediction task where the range of true values can vary significantly.

The final RMSRE value obtained for the best model was 0.35.

Implementation

Ellika Mishra: Designed an image pre-processing pipeline, including resizing, normalization, and augmentation, to ensure high-quality inputs for the model. Mapped image features to `Subject_ID` and `Day` columns to corresponding lunch calorie labels. Developed a Convolutional Neural Network (CNN) to extract relevant visual features from meal photographs and aligned these features sequentially with data

TABLE I
MODEL TESTING PARAMETERS AND RESULTS

Testing Parameters	Validation Loss	RMSE (1)	RMSE (2)
'batch_size: 32, clipnorm: 0.5, epochs: 50, factor_lr: 0.5, learning_rate: 0.1, min_lr: 1e-06, patience_es: 10, patience_lr: 5'	77792.74	278.79	0.4487
'batch_size: 32, clipnorm: 0.5, epochs: 50, factor_lr: 0.8, learning_rate: 0.1, min_lr: 1e-06, patience_es: 10, patience_lr: 5'	49366.41	222.10	0.3164
'batch_size: 32, clipnorm: 0.5, epochs: 100, factor_lr: 0.8, learning_rate: 0.1, min_lr: 1e-06, patience_es: 15, patience_lr: 10'	76262.97	276.07	0.3704
'batch_size: 32, clipnorm: 0.5, epochs: 100, factor_lr: 0.8, learning_rate: 0.01, min_lr: 1e-06, patience_es: 10, patience_lr: 5'	71707.13	267.78	0.4440

TABLE II
VALIDATION LOSS FOR DIFFERENT MODELS

Model Configuration	RMSRE
With breakfast data	0.29
Without breakfast data	0.45
With lunch data	0.36
With attention model, residual neural networks, hyperparameter tuning, and cross-validation using Image data and without breakfast labels	0.35

from LSTM layers and demographic inputs. Enhanced the LSTM architecture by incorporating attention mechanisms to dynamically focus on key aspects of the sequential data and introducing residual connections to improve gradient flow and support deeper network architectures. Experimented with various layer configurations, like activation functions and regularisers, and implemented a hyperparameter tuning process to optimize learning rates, dropout rates, batch sizes, and callback parameters. Applied cross-validation techniques to ensure robustness and generalization across diverse datasets.

Mahima Kardam Bhatt: Pre-processed CGM data and plotted interactive graphs on a per-subject, per-day basis to explore the data. Additionally, created padded CGM sequences and mapped these sequences to their corresponding lunch calorie labels based on Subject ID and Day. Designed and implemented BiLSTM layers to process CGM sequences for the model. Mapped CGM sequences with nutritional data and ran the model both with and without breakfast data, using different hyperparameters. Incorporated learning rate scheduling and early stopping conditions during training. Pre-processed demographic data by implementing one-hot encoding for categorical features (e.g., race) and standardization for numerical features to ensure consistent scaling. Handled missing values by filling gaps with the mean or default values. Have also implemented dropout layer and tuned with several batch size, learning rate and dropout rates. The demographic data was further expanded, integrated with the BiLSTM layers, and mapped with lunch calories per Subject ID and Day. Finally, implemented an RMSRE curve to plot interactive training and validation RMSRE values over the number of epochs.

Yash Milind Honrao: Conducted a thorough Literature survey to study the existing methods to process multimodal data using Machine Learning models. Studied the status of research for personalized nutrition using meal data. Conducted a thorough Exploratory Data Analysis of the given datasets by doing correlation analysis by using heatmaps to identify relationships between numerical features. Learned and tried pair plots to visualize relationships between multiple numerical features. Cleaned the data and pre-processed it to get it in a suitable structure and valid format for training Machine Learning model. Conducted thorough Feature Engineering by doing Categorical Encoding to convert the categorical variables into numerical forms by doing one-hot encoding. Applied feature scaling by using Standard Scaler for better comparability of features. Combined and organised the data, and the corresponding mappings to create a multimodal dataset, and further split it into training and validation sets. Implemented LSTM model for processing CGM Time Series Encoder. Implemented CNN for Image Encoder. Experimented with different models for different modalities (for example also experimented using MLP for Demographical data), to find out which works the best for our case. Implemented RMSRE Loss function and performed hyper-parameter tuning. Solely meticulously compiled and finalized a comprehensive research report detailing our project's methodology, findings, and implications and creating diagrams as well.

CONCLUSION

In this study, we present an approach to predict calories using multi-modal deep learning model by considering Continuous Glucose Monitoring (CGM) data, demographic information, and meal images. We demonstrate the potential of combining diverse data sources to improve the accuracy and reliability of caloric estimation in real-world settings.

Key findings of our study:

1) The effectiveness of our multimodal approach in capturing complex relationships between physiological responses (CGM data), individual characteristics (demographic data), and visual meal properties (image features).
2) The significance of attention mechanisms in processing CGM time-series data, allowing the model to focus on the most relevant glucose fluctuations for calorie prediction.
3) The importance of comprehensive hyperparameter tuning and model selection, as evidenced by our systematic grid search methodology.
4) The value of custom evaluation metrics, particularly the Root Mean Square Relative Error (RMSRE), in providing scale-independent assessments of model performance in calorie prediction tasks.
5) A multimodal model to estimate calories with an RMSRE of 0.35

Our results suggest that rather than considering only traditional single-modality methods, we should consider a more nuance and wholistic approach for estimating calorie intake. The model's ability to learn from multiple data sources simultaneously addresses the complexities and individual variations inherent in calorie estimation.

In conclusion our study contributes to the field of personalized nutrition by leveraging advanced machine learning techniques and diverse data sources. The proposed model promises us for applications in dietary management, health monitoring, and personalized nutrition recommendations.

FUTURE WORK

We acknowledge certain limitations in our study. Future work could explore more data modalities, such as physical activity levels or more detailed nutritional information.

REFERENCES

[1] K. Stefanidis et al., "Artificial Intelligence and Machine Learning Technologies for Personalized Nutrition: A Review," Informatics, vol. 11, no. 3, p. 62, Mar. 2024, doi: 10.3390/informatics11030062.
[2] Theodore Armand, T.P.; Nfor, K.A.; Kim, J.-I.; Kim, H.-C. Applications of Artificial Intelligence, Machine Learning, and Deep Learning in Nutrition: A Systematic Review. Nutrients 2024, 16, 1073.
[3] G. Hu, M. Ahmed, and M. R. L'Abbé, "Natural language processing and machine learning approaches for food categorization and nutrition quality prediction compared with traditional methods," Journal of the Academy of Nutrition and Dietetics, vol. 123, no. 3, pp. 523-533.e4, Mar. 2023, doi: 10.1016/j.jand.2022.11.013.
[4] S. Saha, S. Saha, S. Bhattacharyya and S. Konar, "FoodieCal: A Convolutional Neural Network Based Food Detection and Calorie Estimation System," 2021 5th International Conference on Computing Methodologies and Communication (ICCMC), Erode, India, 2021, pp. 1209-1213, doi: 10.1109/ICCMC51019.2021.9428820.
[5] L. Shi et al., "High-throughput analysis of hazards in novel food based on the density functional theory and multimodal deep learning," Food Chemistry, vol. 442, p. 138468, 2024, doi: 10.1016/j.foodchem.2024.138468.
[6] H. Khan, X. Wang, and H. Liu, "Handling missing data through deep convolutional neural network," Information Sciences, vol. 595, pp. 278-293, 2022, doi: 10.1016/j.ins.2022.02.051.
[7] R. Agarwal, T. Choudhury, N. J. Ahuja, and T. Sarkar, "Hybrid Deep Learning Algorithm-Based Food Recognition and Calorie Estimation," Journal of Food Quality, vol. 2023, Article ID 6612302, 2023, doi: 10.1155/2023/6612302.