Personalized Nutritional Recommendations from Food Images Using a Multimodal GUI

Chase Clemence & Tyler Blue

Invalid Date

Table of contents

1 Abstract

Accurate dieting requires manual input from the individual, requiring them to carefully measure the portion, macros, and quality of their food. Recent developments in the deep learning space attempt to automate this process, leveraging convolutional neural network (CNN) architectures to predict food types with semantic image segmentation as well as food volumne using pixel distances from the camera. However, no one has accurately solved this issue yet. We add the body of research by building a multimodal graphical user interface (GUI) that

leverages a CNN to predict portion sizes and macros as well as a LLM wrapper that gives personalized dietary advice based on the food's content. The CNN is trained on the MM-Food-100K dataset - a state of the art food dataset contianing information about ingredients used, portion size, nutritional profile, cooking method, and more. The developed CNN used ResNet18 pretrained weights, a regression head size of 512, and a dropout rate of **. The LLM wrapper used leverages OpenAI's** model to generate personalized dietary advice based on food macros, physical characteristics, and dietary goals. This is integrated into a simple GUI that consolodates the process into a simple, easy to use platform. The trained model performed well, achieving a MSE of ___. With this GUI, users are able to get real-time improvements to their diet, helping them achieve their goals faster.

2 1. Introduction

Automated food recognition and nutritional estimation are emerging research areas with potential to substantially improve public health. Today, accurately measuring dietary macronutrients still requires individuals to manually weigh ingredients, look up nutritional values, and log foods by hand. This process is time-consuming, error-prone, and often frustrating, causing many people to abandon nutrition tracking altogether. If automated solutions become widely accessible, dietary monitoring could become far more efficient, lowering the barrier to healthy eating and enabling more consistent long-term adherence. Deep learning — particularly convolutional neural networks (CNNs) — provides a promising pathway toward this goal. CNNs excel at extracting latent visual features from images, giving them the capacity to identify dishes, estimate portion sizes, and infer nutritional content directly from pixel information. When paired with an intuitive user interface, such systems could offer real-time macro predictions from a single food photograph, drastically simplifying the nutrition-tracking workflow. Motivated by these opportunities, our project aims to develop a comprehensive, Python-based graphical user interface (GUI) capable of predicting portion size along with protein, fat, and carbohydrate content using a CNN trained on images of food. The GUI also integrates a large language model (LLM) to provide personalized feedback based on the predicted macros. Together, these components form a prototype system for fast, automated nutritional assessment.

3 2. Related Work

Tyler should do this part since he is more familiar with the literature.

4 3. Methods

4.1 3.1 Dataset and Preprocessing

The models were trained on MM-Food-100K, a large-scale dataset consisting of 100,000 high-quality images spanning a wide variety of food categories. In addition to the photos themselves, the dataset includes rich metadata such as nutritional information, portion size, cooking method, and ingredient descriptions. Prior research indicates that models trained on MM-Food-100K consistently achieve higher performance on both regression and classification tasks compared to models trained on smaller or less diverse food datasets. The original schema of the dataset is shown below:

Schema Here

Several preprocessing steps were required before model development. Numerous metadata fields were removed due to irrelevance to the project's prediction targets. Dish names were normalized to ensure consistent labeling across the dataset, and the nutritional profile was parsed so that protein, fat, and carbohydrate content each occupied its own column. Because the dataset provides image URLs rather than raw image files, we downloaded all images locally via HTTP requests. Given storage and compute constraints, we further reduced the dataset by selecting the 300 most frequently occurring food categories and retaining 50 images per category, resulting in a final working dataset of 15,000 images. We then used a conventional 80/10/10 train–validation–test split to support robust model evaluation.

4.2 3.2 CNN Architecture

Our model architecture builds upon ResNet-18, a well-studied convolutional neural network with pretrained weights derived from ImageNet, a dataset containing more than one million labeled images. These pretrained weights provide rich, general-purpose visual representations—including edges, textures, shapes, and color gradients—giving the model a strong starting point for fine-tuning on food imagery. Leveraging this prior knowledge significantly accelerates training and reduces computational cost. The final model uses a hidden dimension of 512 with a dropout rate of ___ (to be filled) to prevent overfitting. We initially incorporated a dish-ID embedding layer to encode contextual information about the food category; however, this component was later removed to ensure that the GUI could generalize to unseen foods without requiring a predefined dish label.

4.3 3.3 Training Procedure

The CNN was trained for 20 epochs to obtain an optimal set of parameters. Because macro and portion estimation constitute a regression task, we used Mean Squared Error (MSE) as the loss function. Optimization was performed using Adam, which generally outperforms basic

Stochastic Gradient Descent by adapting learning rates during training. The initial learning rate was set to 1e-4, but we also implemented a scheduler that reduced the learning rate by a factor of 0.1 whenever validation loss failed to improve. To prevent unnecessary computation and reduce overfitting, we used early stopping with a patience value of 3. This ensured that the best-performing weights were preserved if validation loss began to diverge from training loss.

4.4 3.4 LLM Wrapper

Tyler taks about this since he implements this

5 4. GUI System Design

The GUI was implemented in Python using the PyQt5 framework to provide an intuitive, user-friendly front end for real-time nutritional prediction. Upon launching the application, the user is presented with a large image display panel, input fields for height, weight, and age, a dietary-goal selector, and an optional free-text box for additional context. The interface includes two primary controls: an Upload Image button for selecting a local food photograph and an Analyze Meal button that triggers the prediction pipeline. Once an image is selected, it is previewed within the GUI, and the regressor model is loaded onto either a CPU or GPU depending on hardware availability. When the user initiates analysis, the application applies the same transformation pipeline used during training, converts the input image to a PyTorch tensor, and performs inference in evaluation mode to generate predicted portion weight and macronutrient values. From there, the portion weight and macronutrient values are fed into the LLM wrapper that generates dietary advice based on the information given in the display panel. The output formatted into a human-readable report and displayed within a dedicated, read-only panel. The layout is organized using a combination of horizontal and vertical PyQt5 containers, and custom stylesheets are applied throughout to ensure a polished, modern visual aesthetic. Overall, the GUI tightly integrates the trained CNN model with a responsive interface, enabling seamless meal analysis without requiring the user to manually interact with the underlying code or model artifacts.

6 5. Results

7 6. Discussion & Future Work