CalorAI: The Food Calorie Estimator

Yifei Wang
Student Number 1008822730
yifeii.wang@mail.utoronto.ca

Eric Miao
Student Number 1008009218
eric.miao@mail.utoronto.ca

Vhea He
Student Number 1009525202
vhea.he@mail.utoronto.ca

Abstract

In the digital age, it is has became increasingly important for people to track their food consumption. *CalorAI* addresses this need by using a CNN model to recognize and differentiate between different food items in an image of a meal and estimates the total calorie content of the meal. *CalorAI* encourages healthier eating habits for many people who don't know where to start and contribute to reducing the prevalence of conditions such as obesity. The codebase can be accessed at https://github.com/MiaoE/CalorAI and the dataset can be accessed at Google Drive .

1 Introduction

The global obesity epidemic is one of the most pressing public health challenges of today. According to the World Obesity Atlas 2023 report, 38% of the global population is currently overweight or obese [1], which is a concerning trend that is only projected to get worse. Obesity is heavily associated with a multitude of chronic conditions such as heart disease, diabetes, high blood pressure, liver disease, and cancer [2]. This rising prevalence of weight-related illnesses highlights the importance of tracking food consumption, as it is imperative that people are able to understand the caloric content and nutritional properties of their diets. Traditional diet tracker applications often require their users to input food items and portion sizes manually, which can be tedious and suffer from estimation inaccuracies. In this paper, we present *CalorAI*, a deep learning-based food calorie estimation system that processes meal images to predict their caloric content. This project can serve as a starting point for more advanced diet tracking methods, and help many individuals who aren't sure how to start practicing healthy food habits. *CalorAI* boasts a CNN-based pipeline for food classification and calorie estimation. We hope to take a novel approach to handling multiple food items in a single image, and contribute our open-source implementation for further research and potential real-world applications. This paper will detail the data preprocessing pipeline, the *CalorAI* model architecture and the model's current evaluation results.

2 Background

Tracking caloric intake is an essential step in addressing the global obesity epidemic and managing related health conditions like diabetes. For this reason, calorie estimation has garnered significant attention in recent years. With the widespread use of smartphones and easy access to the web, it should be simple for users to track their diets. However, many existing methods for calorie estimation are usually time-consuming, and suffer from estimation inaccuracies. Traditional diet tracking apps, such as *MyFitnessPal* [3] and similar apps like *Lose It!* or *Yazio* all follow a common design framework. These apps require users to manually log food items and portion sizes, which can quickly

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

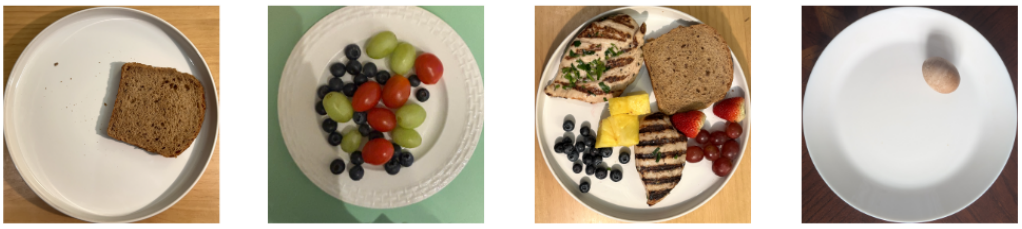

Figure 1: Sample images from our dataset

become tedious and prone to human error. Additionally, they rely heavily on user-generated databases of manually inputted calorie information and usually charge subscription fees, which creates an obstacle for many prospective users.

Many food and calorie related machine learning models have also attempted to explore AI-driven food recognition and calorie estimation. However, these models are either not very helpful for the common user, or are not publicly available. For instance, a paper published by IEEE Xplore looks into using DCNNs for food photo recognition [4]. Although their model was quite successful, with a 78.77% success rate, this type of model does not tell the user much nutritional information, which does not address the diet tracking issue we are tackling. Another paper explores topics much closer to this project, using images to estimate calorie content for dietary assessment [5]. With a 79% correctness within $\pm 40\%$ error, it does of much of what our project aims to accomplish. However, this model is not publicly available for everyday users, nor is a $\pm 40\%$ margin of error the best for accurate estimation.

Applications that are publicly available, such as *CalorieMama* [6], tend to not be the best at providing accurate calorie estimates. Instead of getting the total amount of estimated calories for a given image, CalorieMama focuses on food recognition but can sometimes misidentify items, leading to inaccurate calorie estimates. Additionally, the app only provides caloric information per serving, requiring users to manually calculate the total calorie content by measuring and converting their meals. This is even more time-consuming than traditional calorie tacking apps, and requires the user to perform their own calculations.

In this project, we hope to build a model that provides meaningful calorie estimation with minimal user effort, and is accessible and cost-effective for the general population.

Although there are a vast number of datasets that map food images to food categories and food categories to their calories, we found a lack of public datasets that map food images directly to the calories in the image. This creates an obstacle for our model, as there is no data from which our model can learn *portion estimation* of different types of food. Therefore, we decided to obtain our own dataset for training.

2.1 Data Collection

As our model is designed for everyday users to track their calorie intake, we chose to use cell phone images as our primary data source to simulate the real world. To collect these images, three people from our team took photos using cell phone cameras in different environments to mimic users. The dimensions of the images were ensured to be in 1:1 ratio, and the photos were centered around the 10 inch plates on which the different food items were placed. We also ensured that there would be some contrast between the plate and the image background so that the meal was clearly distinguishable.

A variety of different foods were then put on the plates to be photographed. The foods were weighed using a kitchen scale, and the weight (in grams) of every present food item was recorded for each image taken. These portion sizes were placed into a spreadsheet, and the total calories for each image was then calculated from a dataset that maps food to calories [7]. While photographing the food, we attempted to make each photo conductive to the model learning how to estimate portions directly from images. Firstly, we would place the same food in multiple orientations, taking actions such as rotating the plate, or moving the food to different locations on the plate. We also placed foods in

varying amounts, as well in combination with other foods. This ensured a varied photo database with 573 images of each food item in various different positions and combinations. Figure 1 shows some images from our dataset.

2.2 Data Pre-processing

In order to ensure our dataset was ready for training, we used multiple pre-processing steps to standardize the images and labels we collected before feeding them to our model. Since the raw images from cell phone cameras are high resolution, we resized the images to 400×400 pixels. The images were converted to tensors and normalized.

The labels were converted to multi-label one-hot encoding for training. Each food item present in a given image is assigned a positive value in a one-hot vector that represents the number of calories for a specific ingredient present in each of the items, which gives the model the ability to recognize multiple foods simultaneously.

Food Category	**Total**	**Train**	**Validation**	**Test**
Pineapple	188	151 (80%)	18 (10%)	19 (10%)
Blueberries	182	146 (80%)	20 (11%)	16 (9%)
Strawberries	166	135 (81%)	16 (10%)	15 (9%)
Chicken Breast	159	129 (81%)	15 (9%)	15 (9%)
Cantaloupe	156	126 (81%)	16 (10%)	14 (9%)
Egg	102	84 (82%)	9 (9%)	9 (9%)
Bread	98	77 (79%)	10 (10%)	11 (11%)
Grapes	93	74 (80%)	10 (11%)	9 (10%)
Cherry Tomato	91	75 (82%)	8 (9%)	8 (9%)
Mushrooms	90	71 (79%)	11 (12%)	8 (9%)
Jujube	86	65 (76%)	10 (12%)	11 (13%)
Broccoli	83	70 (84%)	8 (10%)	5 (6%)
Honeydew	81	64 (79%)	9 (11%)	8 (10%)
Cauliflower	80	65 (81%)	10 (13%)	5 (6%)
Raisins	75	61 (81%)	6 (8%)	8 (11%)
Sweet Potato	65	52 (80%)	7 (11%)	6 (9%)
Garlic	64	51 (80%)	5 (8%)	8 (13%)
Apple	51	39 (76%)	7 (14%)	5 (10%)
Carrot	50	40 (80%)	5 (10%)	5 (10%)
Clementine	41	31 (76%)	5 (12%)	5 (12%)
Pear	33	27 (82%)	2 (6%)	4 (12%)
Chives	30	23 (77%)	4 (13%)	3 (10%)
Orange	23	20 (87%)	2 (9%)	1 (4%)
Banana	21	16 (76%)	3 (14%)	2 (10%)
Potato	20	18 (90%)	0 (0%)	2 (10%)
Onion	14	11 (79%)	2 (14%)	1 (7%)
Total	**2432**	**1949 (80%)**	**270 (11%)**	**213 (9%)**

Table 1: Distribution of food categories across dataset splits.

Number of Food Types	**Total**	**Train**	**Validation**	**Test**
1	587	467 (80%)	56 (10%)	64 (11%)
2	212	172 (81%)	21 (10%)	19 (9%)
3	147	120 (82%)	16 (11%)	11 (7%)
4	82	63 (77%)	10 (12%)	9 (11%)
5+	65	53 (82%)	6 (9%)	6 (9%)
Total	**1093**	**895 (82%)**	**109 (10%)**	**109 (10%)**

Table 2: Image count and percentage by number of food types.

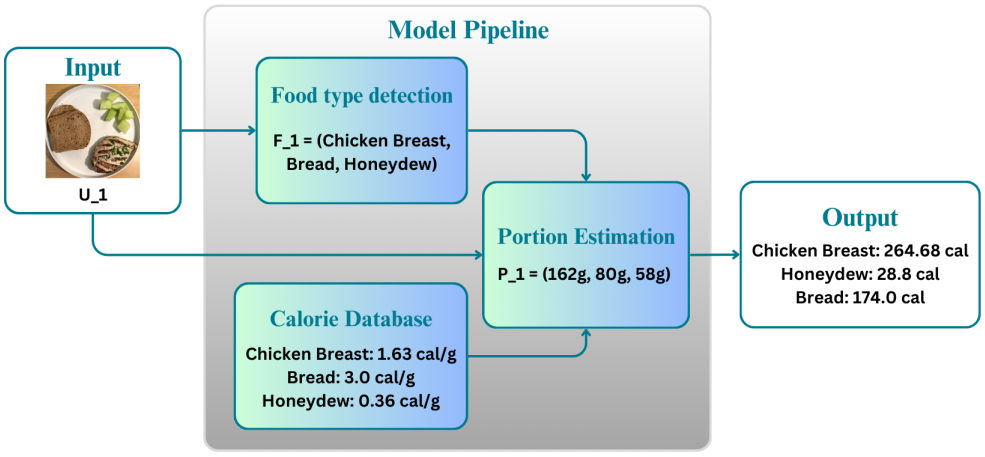

Figure 2: CalorAI Model Architecture

3 Model Design

3.1 Task Definition

The CalorAIze model solves the following task. Given an input plated food image U_i, the model should identify the j food items contained on the plate, denoted $F_i = (f_1, f_2, ..., f_j)$, and estimate the calories $C_i = (p_1, p_2, ..., p_j)$ of each food item f_j.

3.2 Baseline Testing

We performed baseline testing on various existing image classification models on our custom training and testing data. We determined to test various ResNet models due to its scalability and reputation in this field, as well as vision transformer (ViT) models due to its attention mechanisms in transformers.

Baseline testing was performed with 16GB RAM and Nvidia RTX 3060 Laptop GPU with CUDA 12.5, 6GB VRAM. As such, ResNet101, ResNet200, ViT Medium and ViT Large were not included in the testing due to memory limitations.

The results are shown in [Table X]

Model Name	Parameters	Test Accuracy (%)
ResNet18	11,189,850	88.779
ResNet26	13,999,450	90.085
ResNet34	21,298,010	88.567
ResNet50	23,561,306	92.449
ViT-Tiny	5,602,394	65.702
ViT-Small 16	21,821,594	69.442
ViT-Small 32	22,540,442	65.314

Table 3: Test accuracy of different models evaluated on the dataset, with corresponding model sizes.

3.3 Model Architecture

As seen in Fig 2, the CalorAI model consists of three main components: **food type classification**, **portion estimation**, and **calorie computation**. The food classifier is a multi-label classification model based on ResNet18[8]. Given an input image, the classifier identifies the food categories present by outputting a probability distribution over predefined food classes. The portion estimator is a CNN-based regression model that takes both the image and the detected food categories as input. It predicts the weight (in grams) of each identified food item by leveraging convolutional feature extraction followed by fully connected layers. Finally, the calorie computation module maps the

estimated portions to calorie values using a pre-existing database that provides calorie-per-gram information for each food type. The final calorie values are obtained by a simple linear transformation: multiplying the predicted weight by its corresponding calorie-per-gram value.

3.4 Special Considerations

3.5 Changes from Interim Report Code

4 Evaluation

4.1 Evaluation Metrics

The evaluation of the CalorAI model consists of two primary components: **food classification** and **portion estimation**. The food classification evaluation measures how accurately the model identifies the food types present in an image. Three key metrics are used: the F1 score, which balances precision and recall to indicate how well the model distinguishes different food categories; the Hamming loss, which quantifies the fraction of misclassified labels in a multi-label setting; and the exact match ratio, which calculates the proportion of test samples where the predicted food labels exactly match the ground truth.

For portion estimation evaluation, the model's ability to predict the weight of each detected food item is assessed using multiple error metrics. The Mean Absolute Error (MAE) computes the average absolute difference between the predicted and actual portion sizes, providing an intuitive measure of accuracy. The Root Mean Squared Error (RMSE), which penalizes larger errors more heavily, helps identify significant discrepancies in portion predictions. Additionally, the model's accuracy within a $\pm$ 10% range is reported, indicating the proportion of predictions that fall within 10% of the actual portion size. This metric is particularly relevant for practical applications, as small deviations in portion estimation can significantly impact calorie calculations. Together, these evaluation measures ensure that the model provides reliable predictions for food identification and calorie estimation.

4.2 Current Performance

4.3 Improvements in the Model

How did it perform compared to last time? Significant improvement? Why do we think this happened?

5 Future Next Steps

If we were to continue developing this project to prepare it for the user market, there are several areas we would focus on improving and expanding from what we currently have.

6 Expand Dataset

In order to improve the model's performance and scalability, we need to greatly expand the dataset it will be trained on. Currently, our dataset only includes 26 types of foods with images and labels, which is far from what a program that aims to be able to identify most foods should be trained on. We will need a much larger and diverse set of food images, which may involve collecting more photos, or integrating the model with internet resources and databases. This part will require significant time and resources, but is also crucial to improving the model's quality.

7 Improve Model Performance

With a larger and more diverse dataset, we would aim for improving the performance of the model. Although our current model shows potential, it doesn't perform at ideal levels to be used by the general public. This can be improved by training on better data. To do this, we plan to experiment with alternative pre-trained models, such as MobileNetV2 or even more transformer based models, like other, more robust ViT models. We could also manipulate the database with more advanced data augmentation techniques, such as random cropping, rotation, or colour adjustments. With a greater

image database to work with, diversifying the types of images available is very important for more generalization.

8 Prototype and User Interface

For the final product, we also plan to build a user-friendly interface that uses the CalorAI model. The application will be available for mobile use, and should allow users to upload images of their meals or take photos directly using their camera. The system will process the image, estimate the calorie content, and display both the detected food items and their corresponding calorie values. Users will also have the option of recording meals into daily logs, which compiles total calories consumed throughout the day. This feature will enable users to track their long-term eating habits effectively.

References

[1] Author(s) Unknown. Study on ai-based food recognition and calorie estimation. *PubMed Central*, 2023. Accessed: 2025-02-27.

[2] Mayo Clinic Staff. Obesity - symptoms and causes, 2024. Accessed: 2025-02-27.

[3] American Academy of Family Physicians. Myfitnesspal: An app review. *Family Practice Management*, 22(2):31–32, 2015.

[4] Yoshiyuki Kawano Keiji Yanai. Food image recognition using deep convolutional network with pre-training and fine-tuning. *IEEE Xplore*, 2015.

[5] Kiyoharu Aizawa Tatsuya Miyazaki, Gamhewage C. de Silva. Image-based calorie content estimation for dietary assessment. *IEEE Xplore*, 2011.

[6] CalorieMama. Caloriemama - ai-based food calorie estimation, 2025. Accessed: 2025-03-09.

[7] Kaggle Community. Calorie estimation discussion thread, 2021. Accessed: 2025-03-08.

[8] MathWorks. *ResNet-18*, 2024. Deep Learning Toolbox.