Milestone Report: Automatic Food Recognition and Calorie Estimation with Nutrition5k

Abdelrahman Aboelata
University of Maryland
aaboelat@umd.edu

Jaewook Kwon
University of Maryland
jkwon@umd.edu

Yufeng Zhan
University of Maryland
yxz2803@umd.edu

Abstract

We investigate automatic food recognition and calorie estimation from images using the Nutrition5k dataset. This milestone report summarizes our problem formulation, planned technical approach, and current implementation status, along with early qualitative observations and the next steps toward our final project.

1. Introduction

Accurate and accessible calorie estimation is an important problem for promoting healthy eating habits and preventing diet-related diseases such as obesity and diabetes. Manual calorie tracking is time-consuming, error-prone, and often discourages long-term user adherence. In contrast, an automated system that can recognize food items from images and estimate their calorie content based on visual cues such as portion size and composition has the potential to significantly improve dietary self-monitoring and public health awareness.

In this project, we aim to build a deep learning pipeline that takes one or more images of a meal as input, identifies and segments individual food items, and predicts the corresponding calories (and potentially macronutrients) for each item. Unlike simplified setups that only perform food categorization, we leverage the Nutrition5k dataset, which provides weight, volume, and nutritional annotations, to move toward end-to-end calorie estimation. This milestone report outlines our problem statement, technical approach, and preliminary progress.

2. Problem Statement

2.1. Dataset

Our primary dataset is Nutrition5k, a research dataset providing:

- RGB images of plated meals from multiple viewpoints;
- Ground-truth annotations for food components, including weight (grams), volume, and nutritional values (calories and macronutrients);
- Official scripts for standardized data splits and evaluation protocols.

We will initially follow the official train/validation/test splits and use single-view images for the core experiments. Multi-view information may be explored later if time permits.

2.2. Model Input and Output

The high-level input–output formulation is:

- **Input:** One or more RGB images of a meal, in our initial experiments a single image of a plate.
- **Intermediate output:** Instance segmentation masks for each food item, and a predicted food category label for each instance.
- **Final output:** For each segmented food item, predicted calorie content (kcal). Optionally, we may also estimate mass (g) and macronutrients (e.g. protein, fat, carbohydrate).

2.3. Evaluation Metrics

We plan to evaluate the system at multiple levels:

- **Classification:** Top-1 and Top-5 accuracy, macro-averaged F1 score for food category recognition.
- **Segmentation:** Mean Average Precision (mAP) and Intersection over Union (IoU) at standard thresholds for instance segmentation.
- **Calorie regression:** Mean Absolute Error (MAE) and Mean Absolute Percentage Error (MAPE) between predicted and ground-truth calories per item and per plate.

2.4. Expected Results

We expect that:

- Fine-tuning modern backbones such as EfficientNet or Vision Transformers (ViT) on Nutrition5k will provide strong baselines for food recognition.
- Incorporating instance segmentation masks and simple geometric features (e.g. segmented pixel area, inferred depth cues) will reduce calorie prediction error compared to purely category-based estimates.
- Lightweight or compressed variants of the models will offer a reasonable trade-off between accuracy and computational cost, making them more suitable for mobile or web deployment.

2.5. Difference from State of the Art / Baselines

Existing works often focus on either:

1. Food classification without explicit calorie estimation, or
2. Calorie estimation via simple category-to-calorie look-up tables without modeling portion size.

In contrast, our project:

- Uses Nutrition5k's weight and calorie annotations to supervise a regression head that models portion size and mass, rather than only predicting food types.
- Combines instance segmentation (Mask R-CNN) with modern classification backbones (EfficientNet/ViT) and a calorie regression module in a unified pipeline.
- Explicitly evaluates both recognition and calorie estimation performance under metrics that reflect the end-user goal of accurate dietary assessment.

3. Technical Approach

3.1. Data Preprocessing

We are implementing the following preprocessing steps:

- Parsing the official Nutrition5k metadata to extract image–component pairs, ground-truth calories, and masks or bounding boxes where available.
- Applying standard image transformations (random cropping, resizing, horizontal flipping, color jitter) for data augmentation during training.
- Normalizing images using ImageNet statistics to reuse pre-trained backbones such as EfficientNet and ViT.
- Constructing train/validation splits consistent with the dataset's recommended protocol to ensure comparability with future work.

3.2. Instance Segmentation

For instance segmentation, we plan to use Mask R-CNN:

- Initialize from a COCO-pretrained Mask R-CNN model.
- Replace the classification head to match Nutrition5k food component categories.
- Fine-tune on Nutrition5k using standard detection/segmentation losses (classification, bounding box regression, mask loss).

The segmentation masks will be used both for localizing each food item and for computing simple geometric features such as segmented area in image coordinates.

3.3. Food Classification Backbone

We will compare at least two backbone families:

- **EfficientNet:** A convolutional architecture with parameter-efficient scaling, pre-trained on ImageNet, then fine-tuned on Nutrition5k food items.
- **Vision Transformer (ViT):** A transformer-based model that may better capture global context and compositional cues in complex meals.

Both backbones will be integrated into the instance-level pipeline, either by:

1. Using cropped patches from the segmentation masks as input to classifiers, or
2. Sharing a common feature backbone within an end-to-end detection/segmentation framework.

3.4. Calorie Regression Head

On top of the instance-level features, we will add a regression head to predict calories:

- Inputs may include: pooled visual features from the classifier backbone, segmented pixel area, and simple depth or geometric proxies (e.g. relative scale on the plate).
- The regression head will consist of one or more fully connected layers with nonlinearities.
- We will train this head using an L1 or smooth L1 loss on ground-truth calories, possibly combined with auxiliary losses for mass or macronutrient prediction.

We are also considering multi-task training where the network jointly predicts category and calories, with a weighted combination of classification and regression losses.

3.5. Implementation Details

Our implementation will be based on PyTorch and existing open-source libraries for detection and segmentation (e.g. Detectron2 or torchvision models). We aim to:

- Start with a strong off-the-shelf Mask R-CNN implementation and adapt it to Nutrition5k.
- Reuse publicly available EfficientNet/ViT implementations for the classification backbone.
- Log training curves and evaluation metrics using tools such as TensorBoard or Weights & Biases for easier debugging and analysis.

4. Intermediate / Preliminary Results

At the time of this milestone, our progress is as follows:

- Implemented data loading pipelines for Nutrition5k, including parsing metadata and constructing train/validation splits.
- Set up a baseline image classification experiment by fine-tuning an ImageNet-pretrained EfficientNet on Nutrition5k food categories (without calorie regression yet).
- Integrated a Mask R-CNN model and verified that it can be trained on a small subset of Nutrition5k to produce reasonable instance masks for major food items.
- Designed the architecture of the calorie regression head and preliminary code to attach it to instance-level features.

Due to limited time before this milestone, we have focused primarily on:

1. Ensuring that the dataset preprocessing and model pipelines are correct and stable.
2. Running sanity-check experiments to confirm that pretrained backbones can successfully adapt to Nutrition5k.

Qualitatively, early visualization of Mask R-CNN outputs suggests that the model can correctly localize and segment dominant items on a plate (e.g. main protein vs. side dishes). This gives us confidence that the segmentation branch can support subsequent calorie estimation.

In the next phase, we will:

- Complete full-scale training of the classification and segmentation components on the official training split.
- Train and evaluate the calorie regression head using Nutrition5k's ground-truth calorie annotations.
- Report quantitative metrics (Top-1/Top-5 accuracy, mAP, IoU, MAE, MAPE) on the validation split, and compare our results to simple baselines such as category-wise average calorie lookup.
- Explore lightweight model variants or pruning/quantization techniques to assess feasibility for mobile or web deployment.

5. Conclusion and Next Steps

In summary, we have formulated the problem of image-based calorie estimation using Nutrition5k, designed a pipeline that combines instance segmentation, modern classification backbones, and a calorie regression head, and implemented the core components needed for end-to-end training. Our immediate next steps are to obtain comprehensive quantitative results, refine the architecture based on ablation studies, and analyze the trade-off between accuracy and efficiency for potential deployment.

References

[1] M. M. [Placeholder], "Nutrition5k: A dataset for food recognition and calorie estimation," in *Proceedings of [Conference]*, 20XX.

[2] X. Y. [Placeholder], "A systematic review of deep learning applications for food image recognition," *Journal / Conference*, 2023.

[3] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *ICCV*, 2017.

[4] M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *ICML*, 2019.

[5] A. Dosovitskiy *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," in *ICLR*, 2021.