Counting Calories: Application of Object

Recognition and OCR to Food

Samuel Lee   
Computer Science  
Northeastern UniversityBoston, USA  
lee.samuel5@northeastern.edu

Aaron Pan   
Robotics  
Northeastern UniversityBoston, USA  
pan.aa@northeastern.edu

Abhishek Uddaraju   
Robotics  
Northeastern UniversityBoston, USA  
uddaraju.ab@northeastern.edu

*Abstract*—Food plays a crucial role in determining the health of individuals. While both hunger and excess consumption are problems related to food, we observe that overindulgence of food, especially unhealthy foods, is becoming an increasingly important topic in the health community. Now, we observe more individuals making conscious decisions on what foods to eat depending on the macro-nutrients as well as total calorie intake. The main goal of our project is to enable individuals to count calories in a more efficient way by utilizing modern computer vision technology and methodology. Specifically, in this paper, we explore the use of YOLOv8, which is a deep learning model that uses convolutional neural networks (CNNs), to perform object recognition and classification efficiently and accurately. In addition, {SENTENCE ABOUT OCR}. Through transfer learning principles, we use the existing YOLOv8 pre-trained model and retrain it using our own training images. As a visual prototype for this project, we created a simple interactive GUI via Python’s library Tkinter and TkCalendar.

Keywords—YOLO, food classification, food recognition, OCR, convolutional neural network, object recognition

# Introduction

The computer vision field of study has witnessed an exponential growth of technological advancements and discoveries in the past decade alone. From incorporating machine learning principles with computer vision techniques to the recent growing public interest in augmented reality, computer vision has proved to be a blue ocean field of study that harnesses a lot of potential. A lot of this potential stems from opportunities to use computer vision techniques in other fields of study. Based on Amugongo et al. [1], we can notice the growing importance of computer vision in healthcare, especially in relation to food.

Though hunger remains a pressing issue today, the opposite extreme has become an increasingly important issue: overindulgence. Overeating increases the chance of an individual becoming overweight or obese, and studies have shown that obesity increases the risk of type 2 diabetes and heart disease [2]. As such, keeping track of daily calorie intake has become important for individuals, with mobile apps enabling individuals to do so [3]. Not only is total calorie intake important, but what kind of calories one consumes is also important [4]. However, listing all the foods eaten and total calories consumed every day is not an easy task, even with a mobile app. Given that attention spans are getting shorter year over year, this manual method of counting calories is not sustainable. To address the issue of convenience and efficiency of counting calories, we developed a prototype program via Python to demonstrate how simple counting calories can be using modern computer vision technology.

In this paper, we primarily focus on the use of You Only Look Once (YOLO) v8 for object detection. YOLO is a deep learning network model that is recognized for its speed and accuracy when detecting objects [5]. By using a single convolutional neural network (CNN) as well as non-maximum suppression (NMS) algorithm, YOLO can process images faster than most other object recognition model. In tandem with YOLO, we use Optical Character Recognition (OCR) and Tkinter and TkCalendar to enable our Python program to count calories of food images.

# Literature of related Works

Research related to food recognition in the computer vision community has become more prevalent in the past few years. There even is a systematic review of research papers that explore computer vision applications for food recognition with the purpose of calorie estimation [1]. Beyond that, here are some interesting papers that we delved into and were inspired by.

## Classification of Turkish Cuisine With Deep Learning on Mobile Platform [6]

Kayıkçı, Başol, and Dörter explore the use of multiple pre-trained models and transfer learning to evaluate each model’s accuracy and efficiency. The paper experiments with several activation functions – Softmax, Tanh, and ReLU – and several optimization algorithms – Stochastic Gradient Descent (SGD), Adam, and Adadelta. We found it particularly interesting in how this paper compares multiple models using different internal features like the activation function and optimization algorithm. Furthermore, when testing and training the models with images, the paper also uses multiple food image databases such as Food4, Food15, and Food24. This paper’s narrow scope of identifying Turkish foods helped us set a general direction for our methodology.

## A Computer Vision Based Food Recognition Approach for Controlling Inflammation to Enhance Quality of Life of Psoriasis Patients [7]

Hridoy et al. employs computer vision technology to a specific subset of individuals: patients with psoriasis. Recognizing that certain foods increase inflammation in patients, the authors seek to identify the top 15 inflammation inducing foods through deep learning coupled with transfer learning from pre-trained networks. This study was specifically applicable to our project’s goal because of its use of transfer learning to adapt the model for its own use.

## DeepFood: Deep Learning-Based Food Image Recognition for Computer-Aided Dietary Assessment [8]

Liu et al. seems to have been pioneers in using deep learning models for food recognition. This paper released in 2016 and proposes a solution to the cumbersome task of listing daily food intake by using a mobile application that takes images. It also proposes using transfer learning from already existing deep learning models that were widely used at the time. By using CNNs and bounding boxThe issue this paper addresses aligns very well with the purpose of our own project. However, we recognize that the technology Liu et al.’s paper uses may be outdated and relatively slow compared to advancements made since 2016. Nevertheless, we reviewed and referred to this paper the most out of all the literature we reviewed.

## Learning Program Representations for Food Images and Cooking Recipes [9]

Papadopoulos et al. proposed a solution beyond mere food recognition. By taking the food image or recipe image, the authors aimed to create a cooking program, which is a representation of the recipe via a program. They were then able to use StyleGANv2 to generate what the food would look like given the recipe represented by the cooking program. We reviewed this paper to explore further enhancements to our current project. For simplicity, we don’t explore the individual ingredients that went into the food that our program recognizes. For instance, if our system is able to discern pasta from an image, it doesn’t breakdown the ingredients that go into this pasta. Using the recipe database and finding the closest recipe based on the generated image and the actual image, it would be interesting to see how our system can more robustly recognize individual ingredients to then calculate calories.

## Vision-Based Approaches for Automatic Food Recognition and Dietary Assessment: A Survey [10]

Subhi, Ali, and Mohammad explore multiple methodoloigies for food recognition and assessing the food’s nutritional facts. The paper looks at using classic machine learning models as well as deep learning models to classify and identify what food is in the image. After identifying the object, the paper addresses volume and weight estimation to then get the food’s nutrient information. From this paper, we were mainly focusing on the section on getting the volume and weight estimation from a 2-D image. This was also something we struggled with given the scope and limited resources. The paper proposes some robust solutions to get the volume and weight: using another object as reference, virtual reality estimation, and size estimation using camera’s intrinsic qualities such as focal length. This paper helped us brainstorm our methodology for getting the weight and volume of the food item.

# Methodology

For our end goal of enabling users to count calories using images, there are 3 main components: (1) object recognition, (2) weight and volume estimation, and (3) an interface that users can interact with.

## *Object Recognition via YOLOv8*

First, for object recognition, we used the YOLOv8 deep learning model [11]. We took the pre-trained YOLOv8 model and retrained it using our own images for object recognition. To prepare data to use for training our model, we used the Food101 image database in Kaggle [12]. Due to limitations in computational resources and time, we limited the output of the model to 10 object categories: cheesecake, tacos, scallops, pizza, pho, donuts, pancake, spaghetti, strawberry, and banana. Some sample images are shown in Fig. 1. In addition, we trained our YOLO model to recognize LCD screens for OCR purposes as mentioned in section C of Methodology.

{INSERT IMAGE OF EACH FOOD ITEM}

1. Sample Images from Training Set

Using 20 images per category, we then created YOLO-style annotations of the bounding boxes in each image. By using an online tool called CVAT.ai, we were able load our images, draw annotated bounding boxes around the objects that we want recognized, and export .txt files for each image file uploaded that show the corresponding index and bounding boxes within each image.

A plate of pancakes and a cup of liquid

Description automatically generated

1. Bounding Box Tool CVAT

A number with numbers and a number

Description automatically generated with medium confidence

1. Example .TXT File with Annotated Data

Once we have the original images and the corresponding bounding box .txt files, we are now ready to train our YOLO model for our own classification purposes. First, we installed the packages required for training the model in our machines via “pip install ultralytics.” Then, we coded the network training process as shown in Fig. 4.

1. Code for YOLO Training
2. Folder Organization for YOLO Training
3. Log Messages in Terminal

The quality and overall running time of training the model are determined by the number of epochs we specify for training. For increased robustness, we opted to use an epoch value of 100, which took about \_\_\_\_ hours to train. Once training is finished, {SHOW WHAT THE OUTPUTS ARE}

1. Expected Output

## Weight and Volume Estimation via OCR

To measure the weight and volume of the identified food item, we initially evaluated two main options: (1) using a known object dimension as a size reference such as a credit card (2) using images from multiple angles to get the size estimation. Option (1) would still prove it difficult to get the depth of the food object, making it ultimately unreliable if the depth is estimation-based. Option (2) is feasible, but considering the setup and calculations required as well as our limitation on time, we thought that using an OCR approach would yield the highest returns for our time.

By using a scale, we would weigh the food item and take an image of the food on the scale. Using YOLO, we trained the model to read the LCD screen of the scale. Then, taking the bounding box area of the LCD screen, we use OCR to read the weight of the food that is being weighed. Finally, given the weight, we created a table of calories per gram [13], which we use to estimate the total calories of the food.

1. Calories per Gram Table

|  |  |
| --- | --- |
| Food | Calories Per Gram |
| Banana | 0.89 |
| Cheesecake | 3.21 |
| Strawberry | 0.36 |
| Pancake | 2.27 |
| Donut | 4.26 |
| Pho | 0.89 |
| Pizza | 2.82 |
| Scallop | 1.11 |
| Taco | 2.52 |
| Spaghetti | 1.54 |

{ INCLUDE IMAGES OF OCR}

## Interactive Interface via Tkinter and TkCalendar

As a prototype, we want to make our project accessible to users interested in counting calories. Using this prototype, users can understand the logical flow and accessibility of the application. For simplicity, we use Tkinter and TkCalendar, which are libraries available to be installed for Python. The application will start with a calendar view, where the user can select a specific date to open. Once the date window is opened, the user can add food items that they consumed via uploading images. If the image contains an identifiable food item, the user will have the option to add the food item to the daily calorie list for the selected date after reviewing the details on calorie as well as the identified food name.

A screenshot of a calendar

Description automatically generated

1. Calendar View GUI

A screenshot of a computer

Description automatically generated

1. Specific Date View GUI

{ ADD IMAGES }

1. Add Food Item View GUI

{ ADD IMAGES}

1. Unrecognized Food Item View GUI

The total calories, images uploaded, and individual food items per day are saved in a local database for tracking purposes. We used a local database instead of a cloud database for simplicity.

# Experiments and Results

Reliability of the trained YOLO network determines the integrity of our object detection application. To ensure that the network trained well, the first thing we checked is the loss graphs that were generated from the training run. Overall, we can see that there is a decreasing trend in the loss graphs, which is a positive signal for effective training.

{ ADD IMAGES}

1. Loss Graphs of Training YOLO

Next, we ran several images through the trained network and manually confirmed whether the bounding boxes and annotations generated by the network are accurate. Here is a summary of our results as well as some sample images of our testing.

1. Summary of Test Results

{ADD TABLE}

{ ADD IMAGE}

1. Test Samples of Trained YOLO Network

Specifically, we focused on testing to see if the LCD screen is correctly identified as shown in the figure below.

{ ADD IMAGE}

1. Bounding Box Around LCD Screen

To confirm if the content within the LCD bounding box is correctly read with OCR, {ADD DETAILS ABOUT THIS}.

# Discussion of Results

{FILL THIS OUT}

# Summary

In this paper, we explored the use of YOLOv8 and OCR to identify food items and calculate calorie estimations. We also used an interactive GUI for the purposes of counting calories and prototyping the application.

# Future Enhancements

Due to limited resources and scope, there are many things that we would like to have added into this research project. Here is a list of enhancements we would like to make to our research in the future:

* Comparing accuracy and speed using different deep learning models
* Creating a CNN model from scratch and training it and comparing the results with YOLO
* Training with more images
* Training more food categories
* Estimating volume using multiple cameras or multiple images of different angles

##### References

1. G. Eason, B. Noble, and I. N. Sneddon, “On certain integrals of Lipschitz-Hankel type involving products of Bessel functions,” Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955. *(references)*
2. J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.
3. I. S. Jacobs and C. P. Bean, “Fine particles, thin films and exchange anisotropy,” in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.
4. K. Elissa, “Title of paper if known,” unpublished.
5. R. Nicole, “Title of paper with only first word capitalized,” J. Name Stand. Abbrev., in press.
6. Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, “Electron spectroscopy studies on magneto-optical media and plastic substrate interface,” IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].
7. M. Young, The Technical Writer’s Handbook. Mill Valley, CA: University Science, 1989.

**.**