CalorieCapture: Calorie Counting Website

Aiden Walmer

Elizabethtown College  
360 W Bainbridge St   
Elizabethtown, PA 17022  
+1 717-364-7575

Walmera1@etown.edu

**ABSTRACT**

In recent years there has been an influx of people tracking their dietary needs through the use of online applications. More specifically there is a stark increase in the number of people utilizing apps for counting calories, in the hopes of reaching a healthier lifestyle. Calorie counting is the process in which an individual tracks how much energy is getting refueled into your body when consuming food and beverage. This is borderline essential when attempting to maintain the same weight or reach fitness goals. However, human error is often responsible for miscounting caloric intake leading to skewed calorie count results. These inaccuracies cause improper dieting habits and can induce abnormal gaining and loss of weight and in serious cases lead to eating disorders such as anorexia and the seconds leading cause of preventable death in the United States—obesity. I, family, friends, and countless other individuals online have expressed problems with accurately counting their calories. Most nutrition apps and dieting websites require the user to manually type in their caloric intake, which can result in a number of inaccuracies. These inaccuracies include but are not limited to the apps catalog not having a desired food option, tapping on the wrong food product in their catalog, incorrect calorie totals for specific food items, or even the user typing in the wrong number when tracking it manually. To prevent these inaccuracies I am developing CalorieCapture, a website that utilizes AI to accurately compute the total number of calories present in a single photo uploaded by the user. This will greatly reduce the inaccuracies of human error mentioned previously and will produce more accurate calorie counting results.

**ACM Computing Classification System**

**A diagram of a computer system

Description automatically generated**Figure 1. Flowchart of CSS Concepts

**Keywords**

Calorie counter; calorie counting; weight-watching

## 1. Introduction

In the following proposal I will describe CalorieCapture, a calorie counting website that utilizes AI and image detection software to calculate the number of calories in a meal from a single photo uploaded by the user. Human error is often the cause of miscounting calories, which can lead to inaccurate results when trying to lose or gain weight. But through the power of machine learning, CalorieCapture will be able to efficiently and accurately compute the total number of calories present in a single snapshot. Therefore eliminating the chance of incorrectly inputting inaccurate caloric information when tracking one’s calorie intake.

## 1.1 Existing Problems

According to a study conducted in 2020, 36.2% of adults residing in the United States of America are reported obese (ProCon.org). This statistic becomes all the more alarming when considering the fact that obesity reigns the second largest cause of preventable death in the United States, with a mean estimate of deaths clocking in at 300,000 per year due to obesity (Allison D, Fontaine K, Manson J, Stevens J, VanItallie T, 2). When individuals try to track their own caloric intake, a common problem that arises is underestimating the amount they consume and then subsequently underreporting their calorie intake. This can lead to people gaining more weight than they were previously expecting and in severe cases result in developing eating disorders such as obesity.

## 1.2 Existing Solutions Shortcomings

A method still used today for calorie counting is checking food labels and totaling up the calories without any use of any software whatsoever. While this method is accurate, it is grossly inefficient and time consuming. Which leads into quite possibly the most prominent solution to calorie counting, which is done through the use of nutrition apps. While this fixes the problem of efficiency, these apps tend to force the user to both manually input their meals and caloric intake, which results in many inconsistencies. The user could manually enter the wrong amount, accidentally select the wrong food product, or the applications catalog might just have incorrect calorie totals to begin with.

**1.4 My Solution and Expected Results**

The solution I have come up with is CalorieCapture, a calorie counting website that utilizes the power of AI to track caloric intake efficiently and accurately. A user will simply upload a snapshot of their meal and the website will detect and calculate the total number of calories present in the image. This website limits the inconsistency of human error while simultaneously executing at an efficient rate, removing the need to tediously check labels for an accurate calorie count.

## 1.5 Following Sections

In the subsequent sections of this proposal I will start by covering the background of others who have researched this field and what achievements they have made. After covering the background, I will describe my own personal design for solving this problem, including a flowchart and webpage design concept. Followed by an explanation of the experiments I will be conducting to evaluate the effectiveness of my design. Including possible data points, metrics to be used for comparison, and possible results. Then ultimately finalize this proposal with a complete timeline estimating my assumed completion of major goals until the final product is fully implemented.

## 2. Background

Hokuto Kagaya, Kiyoharu Aizawa, and Makoto Ogawa sought out to see how well food could be detected through the use of CNN’s. In their case study “Food Detection and Recognition Using Convolutional Neural Network” they wanted to test how the application of a convolutional neural network (CNN) would perform when given the task of identifying and recognizing food items. Due to the vast variety of types of food available, image recognition of various food items can prove to be very difficult. Although, through utilizing a deep learning algorithm to handle image detection it becomes much mor feasible. According to the results depicted here, CNN showed a significantly higher accuracy rate than traditional image detection methods. Furthermore, they concluded that color dominates the feature extraction process and is the leading characteristic for accurate image detection.

After the correct food item has been identified there still leaves the question of associating that given food item to its corresponding accurate calorie count. “Calorie Estimation from Pictures of Food” by Jun Zhou, Dane Bell, Sabrina Nusrat, Melanie Hingle, Mihai Surdeanu, and Stephen Kobourov strives to solve just that.

“Food Calories Estimation Using Image Processing” by Vinayak Sable

DeepFood: Deep Learning-based food image recognition for Computer-aided Ditary Assessment:

Ultralytics github repository / Yolo v8: GitHub (classify for food detection)

Deep Neural Networks for image detection:

## 3. Design

CalorieCapture website will be designed using a mix of HTML, CSS, JavaScript, and PHP to handle all the front-end development. The initial version of the website will only have a few pages: home, credit, login, secure image uploader, input window, and a results page. If I finish my timeline before the designated due date, I plan on adding a few more quality of life pages and website optimizations.

-Food data set ( Food 101 dataset on tensorflow)

-Describe design process (UI description, image uploader, the dataset I chose to incorporate, dataset implementation using roboflow, training using roboflow, deployment/hosting the server)

-Implementation and training of previously mentioned dataset for incorporating AI using roboflow

To handle the back-end development I will be utilizing a SQL database. The original dataset will include 5-10 fruits of various shapes, colors, and sizes. Each will be paired with their respective calorie count (Calories) for one serving, food identification number (FoodID), and an empty servings field (Servings) to be implemented later.

Table 1. Example Dataset of Fruit Database

|  |  |  |  |
| --- | --- | --- | --- |
| **Fruit** | **Calories** | **FoodID** | **Servings** |
| Apple | 95 | 1 | 1 |
| Orange | 45 | 2 | 1 |
| Banana | 105 | 3 | 2 |
| Grape | 2 | 4 | 3 |
| Strawberry | 6 | 5 | 1 |
| TOTAL | 362 | - | 8 |

## 4. Experiments

The bulk of my testing will come from the image detection software and accuracy of the calorie counter. To conduct the image detection tests, I will be taking photos of the food items on a standard white plate and will be taken roughly 1 foot above the food item from a top-down angle (See Fig. 2) . This standardization will greatly increase the consistency as well as accuracy of the image detection software. Then by following that image standard I will upload a variety of food-based images, some present in the original dataset and others not, to test the accuracy of my image detection software.

A red apple on a white plate

Description automatically generated

Figure 2. Top-Down Apple Photo Example

Secondly, I will be conducting experiments to validate the accuracy of my calorie counter, both with and without the use of AI/convolutional neural networks, to others currently used on the web. Then I will compare them to see which ones prove to be more precise (accuracy percentage) and efficient (time to produce a result in milliseconds).

**5.** **Timeline**

To kick off my timeline I will start by creating the main webpage. Then I will create the back end by using a SQL database. After linking the database back to my website I will then create a secure image uploader on my website. Next, I will tackle the processes of image detection and calorie counting through the aforementioned secure image uploader. Once I have a functioning website that is able to accurately calculate the number of calories based off the foods in my dataset, I will begin the AI implementation to increase the calorie counters efficiency. Finally through the use of convolutional neural networks, my AI will be able to continually learn and expand its original dataset exponentially. I expect to have a functional version of this website completed by the end of this semester, December 8th, 2023. This includes a webpage that allows users to upload images of their meals, which are then detected and calibrated to the corresponding database producing an estimated caloric count of the items detected in the image. The projected semester progression timeline is displayed below (Fig. 3), as well as a full year’s progression timeline WIP (Fig. 4).

A diagram of a diagram

Description automatically generated

Figure 3. Semester Progression Timeline

(Work In Progress)

Figure 4. Yearly Progression Timeline

## 6. REFERENCES

1. Anon. 2022. Most obese countries in the world: Procon.org. (February 2022). Retrieved September 8, 2023 from https://obesity.procon.org/global-obesity-levels/#:~:text=Global%20Obesity%20Levels&text=Among
2. David B. Allison. 1999. Annual deaths attributable to obesity in the United States. *JAMA* 282, 16 (1999), 1530. DOI:http://dx.doi.org/10.1001/jama.282.16.1530
3. Hokuto Kagaya, Kiyoharu Aizawa, and Makoto Ogawa. 2014. Food detection and recognition using convolutional neural network. *Proceedings of the 22nd ACM international conference on Multimedia* (2014). DOI:http://dx.doi.org/10.1145/2647868.2654970
4. Jun Zhou, Dane Bell, Sabrina Nusrat, Melanie Hingle, Mihai Surdeanu, and Stephen Kobourov. 2018. Calorie estimation from pictures of Food: Crowdsourcing Study. *Interactive Journal of Medical Research* 7, 2 (2018). DOI:http://dx.doi.org/10.2196/ijmr.9359

**Make a separate repository for Spring 2024**

* Snap Calorie deep dive (my improvements tracking macros and micros AND allergies associates with the detected food product)

Learning process of Neural Networks:

1. Initialize the inter-neuron connection weights to random values.
2. Feed the neural network one exemplar at a time, each time using the error between desired output and actual output to change correction weights between neurons.
3. Repeat (2) until the output error is in reasonable proximity of desired output for every exemplar. Each time you do this with the entire training set it’s called an “EPOCH”. This learning phase can take thousands of EPOCHs to reach the desired output.
4. After training is done, the neural network will react instantly to not only the desired inputs, but variations of those inputs.