Machine learning-based recognition on Crowdsourced Food Images

Aditya kulkarni
aditya.kulkarni@uconn.edu

Follow this and additional works at: https://opencommons.uconn.edu/srhonors_theses

Part of the Computer and Systems Architecture Commons, Data Storage Systems Commons, and the Other Computer Engineering Commons

Recommended Citation
https://opencommons.uconn.edu/srhonors_theses/803
Machine learning-based recognition on Crowdsourced Food Images

Author: Aditya KULKARNI

Supervisors: Dr. Caiwen Ding, Dr. Xiang Chen

A thesis submitted in fulfillment of the requirements for the degree of Bachelor of Science in Engineering in the Intelligent & Efficient Systems Laboratory Computer Science and Engineering Department

May 25, 2021
Declaration of Authorship

I, Aditya KULKARNI, declare that this thesis titled, “Machine learning-based recognition on Crowdsourced Food Images” and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: Aditya Kulkarni

Date: May 25 2021
“Thanks to my solid academic training, today I can write hundreds of words on virtually any topic without possessing a shred of information, which is how I got a good job in journalism.”

Dave Barry
UNIVERSITY OF CONNECTICUT

Abstract

Caiwen Ding
Computer Science and Engineering Department

Bachelor of Science in Engineering

Machine learning-based recognition on Crowdsourced Food Images

by Aditya Kulkarni

With nearly a third of the world’s population suffering from food-induced chronic diseases such as obesity, the role of food in community health is required now more than ever. While current research underscores food proximity and density, there is a dearth in regard to its nutrition and quality. However, recent research in geospatial data collection and analysis as well as intelligent deep learning will help us study this further.

Employing the efficiency and interconnection of computer vision and geospatial technology, we want to study whether healthy food in the community is attainable. Specifically, with the help of deep learning in the field of health geography, we aim to utilize image recognition to gather and model the role of the community food environment in shaping obesity and related chronic diseases.
# Contents

Declaration of Authorship  iii

Abstract  vii

1 Introduction  1
  1.1 Introduction  1
  1.2 Thesis Description  2
  1.3 The Problem  3
  1.4 Current Solutions  4
  1.5 Our Solution  4
  1.6 Motivation  5

2 Background  7
  2.1 Image Classification  7
    2.1.1 Binary Classification  8
    2.1.2 Multi-Class Classification  8
      One-vs-Rest for Multi-Class Classification  8
      One-vs-One for Multi-Class Classification  8
  2.2 Food Image Classification  9
    2.2.1 Object Detection  9
    2.2.2 Image Segmentation  10
  2.3 Convolutional Neural Networks (CNN)  11
    2.3.1 Neural Networks  11
    2.3.2 Convolutional Neural Network  11
      Layers of a CNN  12
    2.3.3 Mask R-CNN  15

3 Materials and Method  17
  3.1 Overview  17
  3.2 Data  17
    3.2.1 Data Generation  18
      Restaurant Data  18
      Food Image Dataset (FID)  19
    3.2.2 Data Pre-processing  19
      Test-Train-Validation Data Split  19
      Data Augmentation  20
  3.3 Nutritional Information of Food Images  21
  3.4 Architecture  21
    3.4.1 Learning Architecture A: Calorie Mama API  21
    3.4.2 Learning Architecture B: Mask R-CNN Model  22
  3.5 Full End-to-End Food Image Pipeline  23
# List of Figures

1.1 Image Recognition Steps ................................................. 2

2.1 Image Recognition Steps ................................................. 7
2.2 Overview of Object Detection Tasks ................................. 9
2.3 An example of Image Segmentation .................................. 9
2.4 Types of Image Segmentation ......................................... 10
2.5 A Simple Neural Network ............................................. 11
2.6 Illustration of a biological neuron (left) and its mathematical model (right) .................................................. 11
2.7 CNN architecture ....................................................... 12
2.8 Pattern of 3D convolution ............................................. 13
2.9 Pooling Layers in a CNN ................................................. 14
2.10 Mask R-CNN framework for instance segmentation ............ 15
2.11 Mask R-CNN recognition example (Source) ....................... 15

3.1 Mask R-CNN Segmentation Techniques ............................. 17
3.2 Restaurant database provides restaurants that each have food images used to generate nutritional data .................. 18
3.3 An Image Recognition Pipeline ..................................... 20
3.4 Command used to convert PNG images to JPEG format ...... 21
3.5 Mask R-CNN Architecture ............................................ 22
3.6 Mask R-CNN Segmentation Techniques ............................ 23
3.7 End-to-End Pipeline .................................................. 23
3.8 Snippet of a Nutritional Information Generated Table ......... 24

4.1 Image Recognition Steps ................................................. 25
4.2 Approximate Hartford Area Selected displayed using Google Maps .................................................. 26
4.3 Snippet of Database created by Google Place API Calls .... 27
4.4 Snippet of Database created by Tripadvisor Content API Calls ..................................................... 28
4.5 Image Recognition Steps ............................................. 28
4.6 Image Recognition Steps ............................................. 29
4.7 AIcrowd’s Food Recognition Challenge Datasets ............... 29
4.8 Accurate Recognition of (a) Coffee (b) Salad (c) White Bread .................................................. 30
4.9 No Recognition of (a) Sushi (b) Bread (c) Meat and Rice ... 30
4.10 Results by Calorie Mama API [Simplified] ....................... 31
4.11 Image Recognition Steps ............................................. 31
4.12 Image Segmentation model performed on Gelato image ...... 32

5.1 Results by Calorie Mama API [Simplified] ....................... 34
5.2 The Nugget Calorie Conundrum ..................................... 37
## List of Abbreviations

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Full Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>AI</td>
<td>Artificial Intelligence</td>
</tr>
<tr>
<td>API</td>
<td>Application Programming Interface</td>
</tr>
<tr>
<td>CNN</td>
<td>Convolutional Neural Network</td>
</tr>
<tr>
<td>FIC</td>
<td>Food Image Classification</td>
</tr>
<tr>
<td>FIC</td>
<td>Food Image Dataset</td>
</tr>
<tr>
<td>FCN</td>
<td>Fully Convolutional Network</td>
</tr>
<tr>
<td>GIS</td>
<td>Geographic Information System</td>
</tr>
<tr>
<td>GPS</td>
<td>Global Positioning System</td>
</tr>
<tr>
<td>JSON</td>
<td>JavaScript Object Notation</td>
</tr>
<tr>
<td>ML</td>
<td>Machine Learning</td>
</tr>
<tr>
<td>NN</td>
<td>Neural Network</td>
</tr>
<tr>
<td>FIC</td>
<td>Food Image Classification</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

1.1 Introduction

Recent advances in deep learning and related areas has acted as a catalyst in computer vision. However, even now, computer vision and pattern recognition is still no where close to matching human capabilities, suffering the most with intra-category differences. Using computer vision, a field of artificial intelligence (AI) that enables computers to generate information from visual data, we plan on instantaneously providing nutritional data of food images directly from photos and videos. An system exhibiting this technology would have several vital practical applications. For instance, customers at a restaurant could use an application on their smart device (i.e., phone or tablets) to see nutritional data on what they are about to consume simply by taking a picture. Another implementation could be to expedite the process of counting macros- the process of tracking how many grams of each macro nutrient you consume per day. This is extremely useful for the active individual that leads a "wellness-oriented" lifestyle and can now generate macros in an instant [1]. This is done by taking a picture of their meals to instantly calculate the macros rather than typing the information in. Also, it can be used as a measure of obesity and other diseases in a community food environment. Generally, images provide a vast amount of information that can creatively be captured and mined for further analysis.

Image Recognition, especially with food, is not easy. Recognition is extremely different due to difference in textures, lighting, background, and variation. While current "Food Image Recognizers" in the form of applications and models exist today, they suffer when it comes to intra-class variation due to either imperfect recognition models or inadequate food labels. This does not apply that these applications are insufficient but instead displays the need of a full end-to-end recognition system that is catered to these pitfalls.

Not surprisingly, models that focus on eliminating intra-class variance in classification neural networks and capture the requested data have been developed [2]. Yet, there is a dearth in classification neural networks such as these when it comes to food images being the priority.

This thesis provides our method to gather proprietary food and restaurant data in a specified location based on location size. In addition, it revolves around the implementation of a food classification pipeline that is centered on the issues reported above based upon the advancements in deep learning and computer vision, and more precisely convolutional neural networks. These recognition algorithms gather label and nutritional information from underlying data and thus presents an innovative approach to current visual recognition systems. Furthermore, these models were trained with the help of a self generated dataset as well as open-source benchmarks. Due to increased effectiveness of deep learning and increased smartphone
penetration and image-based social media [3], present-day image recognition has achieved substantial results in the field without underscore on certain principal issues.

By highlighting these very issues, we were able to generate a food classification pipeline that is particularly fine-tuned for food image recognition applications, including the custom trained convolutional neural networks (CNNs). Deep CNNs have made breakthroughs in processing image, video, speech and text [4] and have immense representational capacity for the future. By using excelled existing neural network architecture as a base, we were able to build highly accurate food image recognition and classification models. In addition, to encapsulate all of our findings, we created our food image classification system using the proprietary data generated as input to the neural networks. Although fairly simple, our trained models achieved very high accuracy on unknown data and standard benchmarks. Our findings thus demonstrate the viability of a more detailed implementation of a complete end-to-end food recognition system that solves previous limitations without relying on hard-coded prior knowledge.

1.2 Thesis Description

The following honors thesis begins with a survey of the current solutions and relevant literature, followed with a description of our implemented solution as well as a detail analysis of our end-to-end food image classification pipeline. Specifically, Chapter 2 discusses present-day computer science related solutions to the stated problem and further delves into a holistic idea to our machine learning model solutions. Chapter 3 describes the background and related study on machine learning, food image classification, unsupervised feature learning, and convolutional neural network architecture. In addition, this chapter sequentially provides a careful description of our thought process, data generation method- including pre-processing of our two major datasets, and a careful illustration of neural models. Chapter 4 reports some of our experimental analysis as well as some results of our food image classification and segmentation models. Finally, Chapter 5 summarizes the take-away results, elucidates future extensions and the impact of the project and provides concluding remarks.

The goal of this project is fourfold:

- First, to gather labelled restaurant data and food images in a specified area
• Second, to give a description of machine learning models that can successfully describe food images.

• Third, to provide methods of generating nutritional information based on described food labels.

• Fourth, to encapsulate the two goals above and illustrate an real-world end-to-end pipeline that generates nutritional location of food when solely given food images and geospatial information.

The overall project stated as "our work", "our project", or "our model" refers to the joint work done under supervision of Dr. Caiwen Ding with the help of Dr. Peter Chen and Dr. Ran Xu.

1.3 The Problem

After using nutrition-tracking applications and software tools such as MyFitnessPal and MyMarcos+ [1, 5], we found data entry to be unnecessarily tedious for the user. We have an interest in the field of computer vision, so we wanted to see how effectively a computer-vision approach can address an existing problem. We decided on image data extraction as a project due to its inherent difficulty and potentially widespread applications. Domain-restricted computer vision is currently only used in a few specific cases, such as depositing a check by taking a photo with your phone, but we think it could be applied to numerous significant use cases, starting with nutritional data. In general, this system can provide food-related data instantaneously solely based on the food images provided.

American cuisine over the last two decades has had an inconspicuous yet dramatic transformation [6]. Caused by the fast-paced lifestyle, catered interactive lifestyle and budget-friendly food choices, Americans are habituated to eating out rather than having a home cooked meal. In fact, the average American eats an average of 4.2 commercially prepared meals per week. In other words, as a nation, America eats out between four and five times a week, on average [7]. Although extremely affordable and delicious, this excess "away-food" consumption has been causing an overall increase in customer obesity and other related chronic diseases such as hypercholesterolemia and heart disease. Simply put, American citizens are becoming obese and having health issues due to the repercussions of increased "away-food" consumption. As stated by the American Psychological Association, it's an abundance of unhealthy, heavily advertised, low-cost food that underlies the nation's obesity crisis [8].

As restaurants routinely serve food with more calories than people require, dining out represents a risk factor for overweight, obesity, and other diet-related chronic diseases [9]. This lifestyle induced shift in food culture has Americans eating considerably more calorie-rich yet nutrient-poor restaurant food [10]. Over the past 50 years, the health of Americans has gotten worse, and now 71% of Americans are overweight or obese. That means a staggering 100 million people in America are obese. Today, eating processed foods and fast foods may kill more people prematurely than cigarette smoking [11]. To make matters worse, this current cuisine culture has been affecting communities particularly in socioeconomically deprived and food-insecure populations, where they don't have easy access to whole, fresh foods [11].
1.4 Current Solutions

In view of the obesity epidemic, it is vital for consumers to be well informed on the nutrition data of foods served by locations at the community level to have a healthier diet and make better food choices. While the Food and Drug Administration’s (FDA) implementation of menu labeling provides consumers with consistent nutrition information for standard menu items, it does not give them the list of food choices and recommend dietary decisions. Selection of food choices while ordering at restaurants is influenced by multiple tiers of contextual uncertainties, such as food cues, cultural appropriateness, and nutrition education; and these factors significantly differ amongst individuals and largely dictate their food choices [12].

Along with weight tracking, the precise and accurate estimation of dietary caloric intake is imperative to assess the effectiveness of weight loss interventions. Today, there are several solutions provided by both legal entities and co-operations alike. Self-reported dietary intake is not only assessed by methods of manual real-time recording with the help of food diaries and the duplicate portion method—where subjects weigh and put aside a duplicate portion of all the foods they have eaten—but also by methods of recall through dietary histories, food frequency questionnaires, and 24-hour dietary recalls [13]. Although the 24 h dietary recall is currently the gold standard for reporting, this method still experiences bias as the participant estimates their own dietary intake both in short and long term. Most consumers lack the capacity to judge the caloric content of food. Assessing their own intake often causes the participant to both underreport and underestimate [14]. Clearly, a widespread solution is required to instill an effective behaviour-based method of measuring nutrition information that also discloses the nutritional culture of the community. A system that can automatically generate nutritional information directly from food images in real-time can help solve this underlying problem.

A study similar to our method is by Jerry Shannon [15]. Shannon’s method involves in a deep-learning nutrition information tracking approach that includes location-based daily mobility and food sources tracking. Shannon further analyses by providing qualitative inquiries of the community on the participant level. Yet, the approach of combining nutritional information and deep learning seems to be the most accurate in generating a reliable nutritional foodscape of a community to further provide indicators of obesogenity and other illnesses.

1.5 Our Solution

A solution to curb participant bias and provide an overall increase in self-reported survey accuracy is a mobile cloud computing system. This system employs mobile computing devices, such as smartphones and tablets, to capture the dietary information in natural living environment. Additionally, we utilize the computing capacity in the cloud to analyze the dietary information automatically for objective dietary assessment [16]. Among the plethora of ‘health-based’ cloud computing applications that have been released in the past decade such as MyFitnessPal and MyNetDiary, many have proposed to improve dietary estimates. However, even with the availability of features such as food intake tracking, cloud storage, and exercise logging, users are still forced to manually enter all their information to generate results. To overcome these barriers, our research and development efforts have been based on creating a visual-based dietary information analysis system. Specifically, our project is based on creating a food image classifier that can generate nutritional
information and other data from the image. While models such as this have been made before, generating food information on multi-labelled data from food images effectively and efficiently remains a challenging and open research problem. Additionally, our project involved gathering restaurant and food data in certain locations through which we can further analyse the community food culture. We wish to expand on the available models to stimulate progress tackling this issue.

Our food recognition model has uses on multiple levels. For instance, it can be used by macro-tracking applications such as Lose It! ("Lose It!") to expedite the process of adding in logged food items. In addition, with the help of location-based services and geo-spatial technology, our system can be implemented to advance public nutrition knowledge by analyzing the community food environment, restaurant data, and access to healthy food. This crowdsourcing approach has been shown to adequately assess food choices influenced by local food environments by web-scraping food-related tweets [17].

As the pilot study, we decided to generate restaurant and food data in the Hartford, Connecticut area. Through data mining of public restaurant and food images, we were able to use this gather training data for our nutritional information generating classifier. We conclude our study with working classification models and a deep analysis on the Hartford food environment.

1.6 Motivation

Obesity is a common, serious, and costly disease that affects over 42% of Americans. Along with increased weight, it induces obesity-related conditions that include heart disease, stroke, type-2 diabetes and certain types of cancer. Furthermore, these diseases are most prevalent in lower-income minority groups such as Non-Hispanic Black adults at 49.6% in 2017-18 [18]. In addition, well-being is now becoming a topic of great interest and essential factor to the improvement in quality of life [19]. It is a proven fact that paying more attention to our nutrition can make America healthier [20] especially by minimizing the affects of obesity-related diseases.

The field of computer vision seemed to provide a viable solution to the underscored concern. There are several health-oriented software tools and applications similar to our proposed model with the aim of lowering obesogenic-related diseases. In fact, 60% of Americans want to feel healthier and 51% want to lose weight in 2021 [21]. However, less than three percent of Americans have a healthy lifestyle [22]. A probable reason for this conundrum would be caused by the plethora of applications that cater to the health conscious minority who will put in extra effort relative to the rest of the local population. Our interest in machine learning pushed us to finding an answer to link consumer nutritional information and push for a healthier community culture. The generalized goal of our project is to help improve the health of American citizens both at an individual level and on a large scale- be it community or national. Not only can our deep learning based food image recognition approach directly provide the required nutritional information that consumers seek but also create a map of the community food environment by considering the food environment to diet relationship of its residents. With the addition of geo-spatial tracking to the recognition model, the nutritional information of the users can be linked to their location to create community environment data. The deep learning model tackles these obesogenic diseases directly by being an instrument that is capable of ingesting previously unused data.
Chapter 2

Background

This chapter gives a brief introduction to the topics that have been specified in the thesis.

2.1 Image Classification

Image Classification refers to the computer vision technique of identifying what an image represents. This task allows machines to interpret and categorize objects from visual input such as images or videos. Often referred to as “image classification” or “image labeling”, it is a vital component in solving several computer vision-based machine learning problems. Furthermore, it is capable of identifying these objects within images by analyzing and then drawing conclusions from them. Our project of identifying food images is performed using an image classification model that is trained to recognize the various classes of food images.

During the training state, an image classification model is given images with their associated labels. Each label is the name of a distinct concept, or class, that the model will learn to recognize (“Image Classification: Tensorflow Lite”). With sufficient training data, a recognition model can easily predict whether new images belong to any of the classes it has been trained on. When provided a new image, the model returns the label as well as the probabilities it believes the image representing it are. An example output for an image of a dog vs. cat classifier could be as shown below:

\[
\begin{array}{c|c}
\text{Animal Type} & \text{Probability} \\
\hline
\text{Dog} & 0.84 \\
\text{Cat} & 0.16 \\
\end{array}
\]

From this table, we can see that the model accurately believes the input image to be a dog with 84% confidence. In addition, it also anticipates that the image given is a cat with 16% probability.
Broadly, image classification is split up into two main sub-groups:

1. Single Class Classification
2. Multi-Class Classification

### 2.1.1 Binary Classification

Single class recognition, also known as binary classification, refers to machine learning models that put one label on an image. This is very useful for any binary predictions such as labeling an image as a cat or dog, or checking if something is present in the image or not. Although there are only two classes involved (True; Not True), classes may overlap where the instance can be in both groups or even in none. A few well-known binary classification learning algorithms are Support vector machines (SVM), logistic, and perceptron.

Given a specific dataset, tested data can be labelled into four different assignments as shown in the table below.

<table>
<thead>
<tr>
<th>Test Outcome Positive</th>
<th>Condition positive</th>
<th>Condition negative</th>
</tr>
</thead>
<tbody>
<tr>
<td>True Positive</td>
<td>True Positive</td>
<td>False Positive</td>
</tr>
<tr>
<td>False Negative</td>
<td>False Negative</td>
<td>True Negative</td>
</tr>
</tbody>
</table>

**Table 2.2: Tested instances arranged in 2x2 contingency table**

The columns correspond to the actual value whereas the rows correspond to the tested classification value. Often, this table is described as a "confusion matrix" with confidence scores as the values to describe the performance of a classification model.

### 2.1.2 Multi-Class Classification

On the other hand, multi-class classification models solve the problem of classifying instances from an image into one of three or more labels. Heuristic methods are often used to split a multi-class classification problem into multiple binary classification problems to implement binary models. Two examples of these methods are: 1) One-vs-Rest (OvR) 2) One-vs-One (OvO)

**One-vs-Rest for Multi-Class Classification**

In the One-vs-Rest (OvR) or One-vs-All (OvA) classification scheme, multiple binary classifiers, such as SVM and perceptron, are trained to distinguish examples from one class from all other examples. When given a N-class instances dataset, the method requires N-binary classifier models as the number of class labels present in the dataset should be equal to the number of generated binary classifiers.

**One-vs-One for Multi-Class Classification**

In this method, each binary classifier is trained to discriminate between examples of one class and examples belonging to one other class. Therefore, if there are K
classes, the OvO scheme requires training and storing \( K(K - 1)/2 \) different binary classifiers, which can be seen as a disadvantage when \( K \) is large (Pawara et al.).

### 2.2 Food Image Classification

In this thesis, we implement the classification techniques described above to one image category- food images. For machine learning models, classifying food is not an easy feat. With extreme difference in textures, portion size, consistencies, and shapes, food is a very versatile group. What makes identification worse is when certain food objects can be molded into other ones. For instance, a pizza can be folded into half can be incorrectly identified as a calzone. To accurately classify food images, machine learning models use image detection and image segmentation techniques. These models almost always require large amounts of well-rounded data to be good recognizers.

**Figure 2.2: Overview of Object Detection Tasks (Source)**

### 2.2.1 Object Detection

**Figure 2.3: An example of Image Segmentation (Source)**
Generally, object detection is used to describe a set of related computer vision tasks that involve in detecting or identifying objects in digital data such as images and videos. Models are trained to recognize certain boxes in images and link them to certain labels. In contrast to image classification, this task does not predict the type or class of an object in given data. Instead of providing the class label, if gives the bounding box defined by a point, height, and width, and the class label given for each box.

2.2.2 Image Segmentation

Image segmentation is an extension of Object Detection/ Object Recognition. It refers to the more detailed granular information about the shape of an image instead of simple detecting the object. Segmentation plays a central role in a broad range of applications, including medical image analysis (e.g., tumor boundary extraction and measurement of tissue volumes), autonomous vehicles (e.g., navigable surface and pedestrian detection), video surveillance, and augmented reality to count a few [23]. There are two types of image segmentation:

1. **Semantic Segmentation** -
   This type concerns the process of segmenting pixels of an images into their respective classes. In Figure 2.4, the cat is segmented with a different color from the background. In addition, multiple background objects such as the grass and sky are different colors as well. All objects in the same class are considered one entity and are hence denoted with the same color (such as the trees in the same image).

2. **Instance Segmentation** -
   While semantic segmentation achieves fine-grained inference by making dense predictions inferring labels for every pixel [24], instance segmentation is more thorough; All detected objects are masked their own distinct colors. This way, only pixels related to a specific object have the same color. Unlike semantic segmentation, in this type, multiple objects of the same class are treated as distinct units and are hence colored differently.
2.3 Convolutional Neural Networks (CNN)

2.3.1 Neural Networks

Neural Networks are among the most powerful (and popular) algorithms used for classification. This network is a set of neurons organized in layers of many shapes and sizes. Each neuron is a mathematical function that multiplies its one or more vector inputs with certain weights. These input products are then summed up and passed to a non-linear function, known as an activation function, to become the neuron’s output. An activation function is a non-linear function applied by a neuron to introduce non-linear properties in the network [25].

![Figure 2.5: A Simple Neural Network](image)

By comparing this output vector with the data of the other inputs and outputs (stored as ground truth labels), the computation process is tweaked and re-trained to generate better results.

2.3.2 Convolutional Neural Network

Convolutional neural networks, also called ConvNets and CNNs, are a type of neural network that have at least one layer comprised of convolutional units. CNNs are among the most successful and widely used architectures in the deep learning
community, especially for computer vision tasks like image classification, object detection, image recognition [23].

The design of a CNN was inspired by the visual cortex of the human brain: each of the layers convolve the input and pass its result to the next layer. A convolution unit receives its input from multiple units from the previous layer which together create a proximity that share their weights. This is vital in image, speech and text processing as all the units in the neighborhood carry related information. In addition, the convolution units reduce the number of units in the mapping. In turn, this lowers the number of parameters that the model has to learn which decreases the probability of overfitting. They consider the context/shared information in the small neighborhoods. Typical CNNs use from five to twenty-five distinct layers of pattern recognition.

**Figure 2.7: Convolutional Neural Network Architecture (Source)**

**Layers of a CNN**

1. **Input layer**-

   This layer contains input image data and is represented by a three-dimensional matrix. Before we feed it into the model, we need to reshape it to a single column. For instance a $5 \times 5 = 25$ dimensional image is converted to $25 \times 1$. If we have $m$ training examples, then the dimension of the input will be $(25, m)$.

2. **Convolution layers**-

   These layers have a very important task: they extract different features of the input. The initial layer extracts all the lower level features such as the lines and corners of an image. Similarly, the higher-level layers extract layers higher-level features.

   Figure 2.8 describes the process of convolutional layers in a CNN. The $D \times N \times N$ sized input image is convolved to $k \times k \times D$. A $e$ kernel input segmented image produces $e$ features Convolution of an input segmented image.
3. **Subsampling (or) Pooling layers**-

The pooling layer helps reduce the resolution of the features in the input image. This decreased the impact of distortion and noise.

Pooling is usually done in one of two methods-

- **Max Pooling**: This process selects the brighter pixels from the image.
- **Average Pooling**: This method smooths out the image occasionally causing levelfness.

4. **Non-linear layers**-

- **Relu**
  The Rectified Linear Unit (ReLU) is an activation function that is defined by the positive part of its argument. It implements the function $y = \max(x, 0)$, keeping the sizes of input and output layer the same. Compared to other non-linear layers, it increases the non-linear properties of the overall network as well as the decision function without affecting the convolution layer’s receptive fields. It also trains faster than other non-linear layers.

- **Continuous trigger (non-linear) function**
  This non-linear layer passes each element, element by element, in each
feature through a continuous trigger function. A continuous trigger function can be absolute of hyperbolic tangent, hyperbolic, sigmoid, or tangent.

5. Fully connected layers

Often used as the final layers of the convoluted neural network, the fully connected layers sum a weighting of the earlier feature layers, to find a specific target output from the inputs. All the elements in each of the features of the earlier layers are a part of the calculation of each element of every output feature.

In this thesis, we focus on creating a specific type of convolutional neural network for our food images: The Mask R-CNN. The project uses this architecture, to perform instance segmentation on food images.
2.3.3 Mask R-CNN

Developed by Facebook AI Research, the Mask R-CNN is a deep neural network that is aimed to solve instance segmentation problems in machine learning and computer vision. Simply put, it can separate different objects in an image or a video. You give it an image, it gives you the object bounding boxes, classes and masks [26].

To begin, the model presents regions where it believes an object resides based on the provided input image. Then, it predicts the class of the object. First, it generates proposals about the regions where there might be an object based on the input image. Second, it predicts the class of the object and refines the box bounding this object to generate a mask in pixel level of the object based on the first stage proposal. As this architecture returns the binary object mask, class label, and object bounding box, the Mash R-CNN is great at pixel-level segmentation.

We perform image segmentation on our generated food images in this thesis. By doing so, we are able to separate food images from their backgrounds. This object that is segmented is then inspected to generate its nutritional information. I will get into much more detail about this in Chapter 3.
Chapter 3

Materials and Method

3.1 Overview

In this chapter, we first describe the process undergone for Data Selection, Data Collection, and Data Processing. Further, this part is extended by delineating the learning architecture used to train our classification and recognition models. These were the essential building blocks of our full end-to-end system. Finally, the process of integrating these two individual components into a complete end-to-end system is shown.

Overall, we created two trained models using our data. One using Calorie Mama, an instant food recognition API and the other using Detectron2, Facebook AI Research’s next generation library that provides state-of-the-art detection and segmentation algorithms.

Of the overall goal, our empirical study has been conducted in the Hartford area which consists of West Hartford, East Hartford, and Hartford itself, the capital city of Connecticut. With a population at 123,088 according to the 2019 United States Census and a diverse makeup of 12.7% non-Hispanic white, 36.1% Black or African American, 2.3% Asian, and 45.4% Hispanic or Latino according the 2018 American Community Survey, we believed that the Hartford area would be a great “test field” for our project as we required a diverse and thriving food culture with a close proximity towards us.

![Figure 3.1: An image of Hartford, CT on Google Maps](image)

3.2 Data

The study relied on Data in the form of two datasets: Restaurant Locations and Food Images that were generated from all open and functioning restaurants in the Hartford, Connecticut area during February 2020 - May 2021. Our dataset was implemented primarily using Python. This data is then put into classifiers to create models that perform instance segmentation and classification to generate nutritional information of given food images.
Chapter 3. Materials and Method

3.2.1 Data Generation

Data generation was an integral part of this project. Both the Restaurant Data and Food Images Dataset (FID) are taken from multiple sources. The primal Restaurant Data was a premise to creating the FID.

Restaurant Data

The restaurant dataset was instantiated through Python scripts using the Google Place API and Yelp API to automate a GET request from their server as shown below:

https://maps.googleapis.com/maps/api/place/details/json?place_id=ChIJN1t_tDeuEmsRUsoyG83frY4&fields=name,rating,formatted_phone_number &key=YOUR_API_KEY

The above example requests the details of a place by place_id, and includes the name, rating, and formatted_phone_number fields. Here, YOUR_API_KEY stands for your own API key in order for the request to work in your application [27].

The restaurant data derived from the scripts were further validated by other sources including but not limited to Google Maps, phone calls, and Yelp! to ensure the highest degree of data currency and consistency. Restaurants that were permanently closed or unmatched were dropped from the list, and those with uncertainties were cross-validated using further phone inquiries or search engine results. All of the restaurant data was acquired in the February - April 2020 period.

In our project, all restaurants in a radius of approximately five miles or eight kilometers from the center of Hartford were a part of our dataset. All locations of the type Restaurants of these types (according to the Place Types) were included.

bakery, cafe, convenience_store, grocery_or_supermarket, meal_takeaway, restaurant, and supermarket

Restaurant data generated included this information: name, photo, place_id (filter), type (filter), opening_hours, price_level, rating, review, user_ratings_total. Furthermore, we set the optional parameter permanently_closed to False to ensure that only open restaurants were selected. This was essential during this stage as many restaurant locations shut down due to the COVID-19 Pandemic.

A JSON file encompassing all above tags was generated as the output.
We were able to gather a total of 532 restaurants in Hartford with food images available online. 487 of those restaurants were from Google Place and 239 restaurants from Yelp API. The sizable overlap benefited us as we had increased amount of images for specific restaurants. Restaurant data was stored in the form of comma-separated values files on Google Drive for efficient collaboration and storage.

Food Image Dataset (FID)

The primary data required after gathering restaurant data was unfiltered genuine food images from those restaurants in the Hartford, CT area. Food images for each restaurant were collected using a proprietary web image collector. Initially, we decided to reach the restaurant owners and have them send us images that their had been posting on social media and on their menus. However, we found that many restaurants sent us beautified images that looked nothing like the food they served through their websites. We also thought of going to the 500+ restaurants ourselves to collect unadulterated images but quickly realized that this was too time consuming. In the end, we solidified our data generation approach with the help of a Python-based scraping program. For each of the five-hundred and thirty two restaurants, images of the twenty top-listed reviews were collected.

Using them, we were able to collect the highest amount authentic food images captured by unbiased consumers without too much difficulty. The images gathered were scraped from food reviews that the customers had put online. In particular, we used Python’s Google Place API, Trip Advisor API, and Yelp API. The FID include images of various food items all captured by customers who post ratings and reviews online. As a result, images from this dataset tend to exhibit much higher variability and oftentimes have erroneous data in the form of multiple food items in one images and pictures of the restaurant and not food. We had to eliminate this excess inessential data before feeding them to our models. The goal on this dataset is to have focused food images from each restaurant to train the classifier on what the food is.

It should be noted that after copious trial and error with other data collection methods, this proprietary custom method was the most accurate and complete to generating the required food images from each restaurant. As all the images either came from Google Review, the most famous business review site or Yelp!, the most popular restaurant review website these food images definitely represent food items that the customers paid for and consumed compared to menu food images that restaurant owners beautify and not serve.

A total of 23,163 images were used from the 532 restaurants as primary training and validation data.

3.2.2 Data Pre-processing

There are a few image pre-processing techniques used in this thesis. This is done to the data to normalize the data stream to ensure maximum efficiency from the proposed model.

Test-Train-Validation Data Split

The FID is split 15% for validation 15% for testing and the remaining for training. As this is a smaller dataset, the model trained will not generalize well for data from
the validation and test set and will cause overfitting. To curb this, we perform Data augmentation.

**Data Augmentation**

Data Augmentation a deep learning algorithm strategy that increases the breadth or diversity of the dataset. As this data is used to train models, data augmentation enlarges the set without actually collected new data from different sources. These techniques ensure that any image taken from any angle will be able to get classified. Although many CNN architectures are created with great depth intent, not much focus is put on their data augmentation policy. In our project, we use random and directed transforms to rotate, flip, warm and change the lighting of images randomly.

Firstly, we used `ImageDataGenerator`, a Keras deep learning library that provides us with the ability to use data augmentation automatically when training a model. Using arguments like `rotation_range` and `zoom_range` from this library, we were above to rotate and shift images in the data stream respectively.

Images were randomly rotated at an angle up to 45 to ensure training and validation from any degree. Further, images selected randomly were horizontally and vertically shifted using the same library. This moves all the pixels of the image in a direction, either vertically or horizontally while keeping the dimensions of the image the same.

Images were flipped horizontally and vertically by setting the `horizontal_flip` and `vertical_flip` argument to True to allow for “incomplete” or “half” images to be predicted.

The brightness of certain images were randomly darkened and brightened to allow or the model to generalize training for images of different lighting levels. This was achieved with the `brightness_range` argument to the `ImageDataGenerator` constructor. Values in the range \([0.5, 1.0]\) darkened the images and values in the range \([1.0, 1.5]\) brightened the images.
The `zoom_range` argument randomly zoomed the image by adding new pixels around the given images. Upon our observations, the range for the zoom was be $[1 - x, 1 + x]$, where $x$ is the value provided. When $x = 0.4$, the random zoom is between $[0.6, 1.4]$.

All the images were also converted to JPEG format to ensure consistency for the model. This was done with a Command line script using ImageMagick’s `mogrify`, an inline image modification tool. Figure 3.4 shows the command we gave to convert all our PNG images into JPEG format.

```bash
>for /r /d %a in (*) do mogrify -format jpg "%~a/*.png"
```

**Figure 3.4:** Command used to convert PNG images to JPEG format

To delete the extra PNG files, we used the command `del /S *.png` while removes all files with .PNG file extension in a specific directory.

Finally, all food images were standardized and reshaped to 544*544 pixels by targeting the center of the image while cropping the long side to ensure compatibility as inputs for the proposed neural networks. Each image was linked to the restaurant using a ‘PLACE_ID’, which referred to the Restaurant ID.

This is the formula used to resize the images. For images denoted by $H \times W$, where $H$ is its height and $W$ is its width:
- If $H > 544$ and $H \geq W$, then $H_{\text{new}} = 544, W_{\text{new}} = 544 \times W / H$;
- If $W > 544$ and $W \geq H$, then $W_{\text{new}} = 544, H_{\text{new}} = 544 \times H / W$;
- If $H \leq 544$ and $W \leq 544$, then no change was done.

For example, if the image was 1000 * 500, we simply resized it to 544 * 272 pixels.

### 3.3 Nutritional Information of Food Images

### 3.4 Architecture

There are two models we gather results from in this project: 1) **Model A:** Using Calorie Mama API 2) **Model B:** Using Mask R-CNN Model.

#### 3.4.1 Learning Architecture A: Calorie Mama API

The first model implemented uses Calorie Mama, a deep learning-based food image recognition API. The API is used to generate nutritional facts and information of the supplied food images. Calorie Mama requires a POST request, a request method supported by HTTP used by the World Wide Web, with the API user key and the local file path of the image as a parameter. The request format is shown below:

```bash
curl -H -i -F media=@image.jpeg https://api-2445582032290.production.gw.apicast.io/v1/foodrecognition?user_key=YOUR KEY HERE
```

Once a food image is uploaded and the request is pulled, the Calorie Mama API returns predicted nutritional information of the food based on the image. These nutrition facts include:
Chapter 3. Materials and Method

• Calories
• Macronutrients (i.e., carbohydrates, proteins, and fats)
• Transfat
• Serving Size
• Micronutrients (i.e., vitamins and minerals)
• etc.

Although we have a variety of nutritional facts to choose from, we focus only on Calorie Information and Portion Size in this project.

For each image, first data collection is done, then data processing, and finally they are passed into the model architecture.

The nutritional facts of all food images is then aggregated to collect consolidated information for each restaurant i.e. every geographic location. This information can be compared to the average calorie information of all restaurants by type, name, and most importantly, location- acting as a display and/or factor of obesogenity in a community’s food environment.

3.4.2 Learning Architecture B: Mask R-CNN Model

The other model we use to recognize food from restaurant food images is implemented as a Mask R-CNN in Python. A Mask R-CNN is a deep neural network aimed to solve instance segmentation problem in machine learning or computer vision. In other words, it can separate different objects in an image or a video. You give it a image, it gives you the object bounding boxes, classes and masks. (Zhang)

This model extends the Faster R-CNN by adding a branch for predicting an object mask in parallel with the existing branch for bounding box recognition. A great pre-trained model we found initially was Facebook AI Research’s Detectron2. The most popular readily available dataset for image classification is the ImageNet database, which has been used to train the Google Inception CNN. However, this dataset does not focus on food images. In fact, the only valid dataset that we could find that was truly fixated on food images was the AIcrowd Food Recognition Challenge. As stated on their website, this dataset contains:

• train-v0.4.tar.gz: This is the Training Set of 24120 (as RGB images) food images, along with their corresponding 39328 annotations in MS-COCO format.
3.5 Full End-to-End Food Image Pipeline

The main purpose of this research project was with the aim of creating a complete nutritional information generating model using only the food image as the required input (geographic location is optional but recommended). We have achieved this with either of the two classification models given above. The process of the pipeline is as follows:

1. Collect Restaurant Location data within a specified radius
2. Generate Food Images using Customer Reviews and other online resources for each restaurant
3. Use these images as input data to either of the two models listed above
4. Gather labelled data (or) Nutritional Information from the model
5. Send Food Label and Serving Size to generate proportionate nutritional information.

We store all the images labeled by their restaurant index, image index, Image source and food id to ensure thorough results. A snippet of a database is shown below in figure 3.8

<table>
<thead>
<tr>
<th>Restaurant_index</th>
<th>Image_index</th>
<th>Image_source</th>
<th>name</th>
<th>food_id</th>
<th>group</th>
<th>score</th>
<th>totalCarbs</th>
<th>totalFat</th>
<th>protein</th>
<th>calories</th>
<th>fiber</th>
<th>sodium</th>
<th>servings_size</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>Google Image Combo Pizza</td>
<td>5e835e82980995318 Pizza</td>
<td>90</td>
<td>0.25</td>
<td>0.1</td>
<td>0.1</td>
<td>2405</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>10</td>
<td>Google Image Combo Pizza</td>
<td>5e835e82980995318 Pizza</td>
<td>107</td>
<td>0.26</td>
<td>0.1</td>
<td>0.1</td>
<td>2420</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>11</td>
<td>Google Image Chicken Nugget</td>
<td>763402856483210</td>
<td>Fried Chicken</td>
<td>69</td>
<td>0.003</td>
<td>0.002</td>
<td>0.002</td>
<td>40</td>
<td>NA</td>
<td>NA</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Figure 3.8: A Snippet of a Nutritional Information Generated Table**
Chapter 4

Experiments

As stated earlier, the work done for this thesis was a pilot study created as a test study for a much larger scale project. The two datasets we use to train our models (Model A and Model B) are:-

1. Our Hartford dataset: Comprised of top most reviewed items crowd-sourced from websites online.

2. AlCrowd’s Food Recognition Challenge- Starter Kit dataset: A challenge dataset available here.

In this chapter, I describe the result of our implemented restaurant data generator, food image collection system, food recognition, and image classification models as well as our end-to-end pipeline. The same dataset- our Hartford data- was used to evaluate all models.
4.1 Data Collection

4.1.1 Restaurant Data Accumulation

In our project, we aimed to append all open restaurants in Hartford, East Hartford, and West Hartford into our dataset. Upon further analysis, we realized that this simply meant an eight kilometers, a five mile radius centered around the middle of Hartford County, Connecticut.

Further, by using Google Place API, we were able to find all the restaurants in our required radius which was a five mile space containing all OPEN restaurants. Although the API did not have a specific ‘food locations’ field, we included all places that made or sold food items. This included bakeries, cafes, convenience stores, grocery stores, supermarket stores, meal takeaways and finally restaurants.

there are 10 photos per result at the moment. 5 reviews for each location. Although this way not ideal, it was the best we were able to do.

This was not enough data for us. We wanted to have a holistic review of the food landscape around Hartford and only five food images per restaurant does not create for one. This led us to searching for something similar to Google Place API.

The solution we found was the Tripadvisor Content API. Through its Content API, Tripadvisor provides free, up-to-date rating content to select travel websites and apps [28].

Using the Tripadvisor Content API, we could collect the three top-reviewed food images for each restaurant. Overall, this allowed for eight total food images per restaurant for our pipeline.

For a certain amount of time, we thought of using the Yelp API as well for more training data. However, upon future analysis we decided not to go further with the Yelp API due to time constraints. Yet, we believe that our trained models have very promising results. A future extension of the project will contain images from various other crowd-sourced sites such as Facebook and Yelp.

All the restaurant data collected using our Python scripts were stored in a refined database. In the database, each restaurant was referenced by their restaurant_id primary key. In addition, every restaurant had a name field, address column, url, phone number entry, rating score, etc. Table 4.1 shows a minute snippet of the whole table. In total, the restaurant information database had the size 955 * 55, with 955 restaurant entries (rows) and 39 features (columns) for each restaurant. The surplus amount of data from simple API calls shows how uncomplicated it is to utilize current technology.
4.1. Data Collection

for information gather. This restaurant data excel file is not stored on the shared Github given in Appendix A as this data is to be published in the future.

4.1.2 Food Image Generation

The next clear step was to generate crowd-sourced images for each of the restaurants gathered. In this project, the word restaurant is used as a collective for all types of food eateries such as bakeries, stores, wineries, etc.

The reason for crowd-sourcing images was twofold:

1. Crowd-sourced images provided us with image data directly by the consumer which preserves authenticity
2. Going to the restaurant locations would have been time consuming, less effective, and not probable due to COVID-19.

Utilizing APIs in Python, we observed a clear solution of gathering food images from restaurants. In particular, we narrowed it down to two APIs: Google Place API and Tripadvisor Content API.

The Google Place API allows users to access information stored as part of the Google Maps Platform. This API service returns information about places using HTTP requests where each of these places is defined within this API as establishments, geographic locations, or prominent points of interest [29]. We used the Place Details place request on the platform to return a more detailed information about a specific place, including 5 user reviews per restaurant call. These Google reviews provided at least 5 food images- at least one per review- for each restaurant API call.

Similar to the Google Place API, the Tripadvisor Content API supplied us with free, up-to-date rating content to websites [28]. By calling the API and parsing the response, we were able to save the data from the response into our food image database. In particular, each API call (restaurant call) provided us with 3 user reviews.

In total, with the above two APIs, we had 8 reviews for each restaurant. That meant at least 8 food images that was already labeled for the model. As both APIs we labeled by restaurant_id as well as restaurant_name, we were able to merge the data flawlessly. Although we wanted to use more APIs such as the Yelp Developers [30], we could not due to time constraints. However, our models show great success with the two used APIs itself.
Chapter 4. Experiments

As each image from both APIs was provided in the form of an image url, we had to manually enter each url to save them into our cloud database. A special thanks to Dr. Xu, Dr. Chen and their teams for gathering these images for us based on the information provided by our Python-API scripts. The Tripadvisor Content API food mapping had a total of 3256 images from 239 restaurants and the Google Place API supplied a collective of 19907 images from 487 restaurants. All of these images gathered were labeled based on 1) Review they were taken from 2) Restaurant there were provided from 3) Location (Hartford / East Hartford / West Hartford). This way, we were able to have a large amount of structured data to feed into the classifiers. Figure 4.11 shows a images from the ninth restaurant from Tripadvisor. All of the titles of these images in the figure have been appended with _resized as they have been pre-processed for the models from Learning Architecture A and Learning Architecture B. We will describe the results of these models in the Section Full End-To-End Food Image Classification.

4.2 Full End-To-End Food Image Classification

Our end-to-end pipeline involves the earlier data collection steps as input data to our Learning Architectures as described in Chapter 3. Further, the nutritional
4.2. Full End-To-End Food Image Classification

Figure 4.6: An example of a food classifying application (Source)

information generated as the output is analyzed to observe the community food landscape.

All of the images collected in the previous steps are used as input data to our models. These images are first pre-processed, resized, and augmented to 544 * 544 JPEG files to ensure consistency in data and reduce overfitting. The augmentation helps in adding additional input to the models without actually providing new true images from the restaurants.

In addition, we used the AIcrowd Food Recognition Challenge Dataset [31] containing 25,389 labeled images to further aid the training process of our models. As a result, we have a much larger dataset for the models to train on. Figure 4.7 displays the dataset used for further neural network improvement.

Figure 4.7: AIcrowd’s Food Recognition Challenge Datasets

4.2.1 Food Recognition

This section was of paramount importance in our project as the recognition of food is what leads us to the nutritional information generation. Fortunately, both of our Learning Architectures successfully and accurately classified most images provided with very few mistakes. While Learning Architecture A, the Calorie Mama API, almost always recognized the food correctly, with a 97% accuracy of our 200 images food label test, our Learning Architecture B: occasionally had confusions regarding labeling due to the smaller training dataset- using only our Hartford food
datasets and the AIcrowd data. As Architecture A utilizes the training data provided by Calorie Mama [32], it had the better accuracy of the two. Both architectures took in the same pre-processed images as the input ensuring no bias to either method.

Through specific case studies of each model, we were able to observe the type of food images for which our models did not perform the best. As stated earlier, our Calorie Mama API model almost always correctly identified the food item. However, our Mask R-CNN model occasionally was not able to judge the food object in the picture and returned no recognized food label. For instance, in figure 4.8, we observe the accurate recognition from the Mask R-CNN model. Although these food objects are pretty intricate, the model identifies them with fairly high accuracies (68%+). However, as displayed in figure 4.9, this model is not perfect. The architecture was not able to put a label on simple food items like Sushi (a), and Rice with Meat (c). Foods too similar to each other such as various types of tea and coffee or too rare such as a chef’s signature dish struggled in this model. This occasionally occurred in my architecture as my dataset only contained a few images for specific food items. Due to this, certain images that were not similar looking to their standard (such as half-eaten food images) we not labeled.
4.2. Full End-To-End Food Image Classification

Figure 4.10: Results by Calorie Mama API [Simplified]

Figure 4.11: FoodAI results for the same image

Figure 4.10 displays the correct identification and nutritional information generation by Calorie Mama API. The image contains a Beef burger with a side of fries. Please note that the image (B) does not contain all the information gathered by the API. When compared to the FoodAI model displayed by 4.11, we can clearly see how much more precise our Learning Architecture results are when compared to other available models.

While nutritional information was generated directly for the Calorie Mama model, our other learning architecture required one extra step—linking the identified labels derived from the images to a nutritional database. For this important process, we used the FoodData Central, an integrated data system that provides expanded nutrient profile data and links to related agricultural and experimental research [33]. By doing so, we were able to gather nutritional data of each given image and complete the whole pipeline. Table 4.2 displays a very small part of the final nutritional data:
gathering by Pipeline B. A special thanks to Dr. Xu and his team for curating this database. Both databases, each generated from one of the two pipelines, have been further analyzed in our project to observe the patterns between the role of food in a community and obesogenity and related diseases.

### Chapter 4. Experiments

#### 4.2 Experimental: Image Segmentation

To increase accuracy for the models, I also tried Image Segmentation for the images before feeding them to the model architecture. This would in turn remove the background and unwanted objects from the picture so that the model would train by focusing on the food rather than other subjects. Although, this did drastically increase the accuracy for certain images, my created segmentation model was not very clever and would often have a hard time figuring out what the object (food item) was. This is common as segmentation takes a great deal of training data to be effective and accurate. The code that I used to do this is provided in Appendix A. Figure 4.12 shows a result of our working image segmentation model.

![Image with subject and background](image1.png)

![Masked object with deleted background](image2.png)

**Figure 4.12: Image Segmentation model performed on Gelato image**

<table>
<thead>
<tr>
<th>Food Image</th>
<th>Food ID</th>
<th>Food name</th>
<th>Food portion</th>
<th>Calories</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pizza with meat and cheese, from restaurant or fast food, medium crust</td>
<td>58106725</td>
<td>Pizza</td>
<td>5 small slices = 340 g</td>
<td>274-135</td>
</tr>
<tr>
<td>Tuna, fresh, baked or broiled, small fillet = 340 g</td>
<td>567.8</td>
<td>Tuna</td>
<td>2 small fillets = 340 g</td>
<td>26153120</td>
</tr>
</tbody>
</table>
Chapter 5

Conclusion

To conclude, I first summarize the work presented in this honors thesis that was done as part of the research project. The proposed system for learning the nutritional information of crowd-sourced food images has been substantiated with validation trials given in this section. Additionally, I provide a remarks section that gives a holistic perspective on our work helping the public health of the community. Furthermore, I describe certain limitations of the current constructed architecture and also provide detail on our future work direction.

5.1 Discussion

The work done as part of the FIC project has proven to be an innovative approach to the food image classification problem. State-of-the-art machine learning techniques, including neural networks and deep learning are utilized. The main objective of our work for this thesis specifically was to develop a deep neural network system that could 1) accurately classify food images and 2) generate their nutritional information. For the same, two models using Convolutional Neural Networks- Mask R-CNN specifically- and were trained on our collected 23,163 Hartford database of food images and 24120 training images from AIcrowd’s Food Recognition Challenge [31]. This thesis serves as a pilot study for a large project. We mostly focused on the construction of the machine learning architecture of the final project.

In particular, our end-to-end system integrates diverse datasets, convolutional layered networks, and large databases on nutritional information. This integration serves as the pipeline that generates the required information from a direct visual stream.

In the test area of Hartford county, we gathered information about all open restaurants using Google Place API. Furthermore, we crowd-sourced food images from the top eight reviews for each respective restaurant using the Google Maps Platform and Tripadvisor. The importance of using both Google Maps and Tripadvisor was to gather a good mix of menu items from each available eatery. This data was securely labeled with restaurant and image IDs, which was then stored in the cloud as input to our CNNs.

With two convolutional network architectures, we were able to create several robust food recognition modules that were used to store food data as their input. In addition, using the output of the food detection systems, the nutritional information of the food was also generated through a food label nutritional database. By weaving all of these components together, we completed our end-to-end system that generated nutritional information of food images from restaurant food images in the Hartford area.
34

Chapter 5. Conclusion

Figure 5.1: Results by Calorie Mama API [Simplified]

This pipeline created is unconventional when compared to previous food recognition systems. While those pipelines usually ingest larger datasets over larger geographical areas, this project focuses on a cluster of restaurants all in the same area. This was done to construct a food image classifier and nutritional information generator, so as to analyse the food nutrition landscape. This approach ensures that we are capable of modeling the role of food sources available in the community in shaping its resident’s health and well-being.

Since I contributed to the food and restaurant information generation, and classification R-CNN models parts of the project the most, I ensured to provide evidence of our success in Chapter 4 of this thesis. Our detection pipelines are of the highest capacity by augmenting state-of-the-art image recognition and information generation. If there was no time constraint, it would have been possible for me to further extend the project by creating another model from scratch or even adding another test area. However, even without them, I believe that the project so far shows the capability of our pipeline on a larger scale in the field of deep learning- with the recognition systems and in the field of health geography- with the analysis of the shaped community food environment.

5.2 Remarks

Our work involved constructing an innovate deep learning food image detection model that generated nutritional information of crowdsourced images provided. The images were all taken from restaurants in Hartford, East Hartford, and West Hartford. The most effective way of gathering the images was using online customer and business review websites such as Google and Tripadvisor. The geo-referenced food images collected were labeled to the restaurants they belonged to such that an array of images were embedded in each restaurant in the list of restaurants. This array of arrays was constructed as a large spread database holding detailed information of each image like its calorie count and portion size. We validated both the models’ findings with the restaurants to ensure reliability of deep learning network technology in accurately generating nutritional information from food data.
Our pioneering research project links convolutional neural networks with georeferencing technology to enhance the ability of food identification and nutritional data generation. In addition, we were able to analyze the community food culture and environment through the information returned by our models. Choosing CNNs was the best choice based on their promising results in the field of image recognition and other computer vision tasks. Particularly, the Mask R-CNN is what we settled on. The Mask R-CNN extends the Faster R-CNN by adding a fully convolutional network (FCN) to the top of the CNN features, generating a mass segmentation output.

Our dataset contained:

1. 3256 Tripadvisor Crowdsourced images from 239 restaurants
2. 19907 Google Images Crowdsourced from 487 restaurants
3. 26658 Images from AICrowd Food Recognition Challenge

Together, the merged dataset together was fairly large with a skewed variety of drinks, fast foods, and cultural foods. All of the data clustered together was used to further analyse the community food environment—especially the role of the community food landscape in shaping an obesogenic environment. It is important to understand that the AICrowd images were simply used to train the models while the Crowdsourced images were also used in creating the community foodscape. Both of our learning architectures A and B were highly accurate in identifying the right food images. In addition, they were able to correctly generate the nutritional information for almost every image given. While A used the Calorie Mama API, B involved in our local pre-trained Detectron Mask R-CNN model. Architecture A usually outperformed our local model in B due to the difference in training data provided—Calorie Mama API is trained with hundreds of thousands of different images while our Mask R-CNN had only a little less than fifty-thousand images. In addition, we were not able to perform further data augmentation on the Mask R-CNN model due to hardware limitations.

This project helped contribute to creating an effective fault-less tool that helps in visualizing the community, their nutrition landscape, and the proximity to nutrition-rich food. Using our deep learning CNNs and a mapping of the community geography via GIS, we are able to delineate the spatial food environment of specific neighbourhoods for further analysis in nutrition and illness. These metrics are usually overlooked by current accommodated spatial metrics, making our research crucial in the field of community health. For instance, relationships between consumer income, food quality, food location proximity, job, restaurant pricing, and the Supplemental Nutrition Assistance Program (SNAP) is usually not the main concern of similar studies. By adding in the nutrition of food in each community, we can have a considerably more accurate metric to compare the food intake with links to community illness such as obesity that we can use to contribute to related studies even further.

In addition, the data generated from crowdsourced food images can be used as another measure of community environment health. Currently menu labeling, which refers to a requirements by the FDA to provide consumers with consistent nutrition information for standard menu items [34], is the indirect tool used to analyze the food landscape. However, by blending public health, health psychology, and urban planning perspectives, our model caters for a holistic analysis of the community nutrition environment including the type and location of Food Outlets.
(stores, restaurants) as well as their Accessibility (operation hours, distance, pricing, etc.). This further helps understand the factors in molding an obesogenic community food landscape in the long-term on both a short-scale (per neighbourhood), and large-scale (state-wide or nation-wide). Further, an accurate instantaneous food nutritional information generating application in the hands of each community resident would help them gauge their personal health individually as a community. More importantly, this project creates a solid platform for future research, providing us with the community’s behaviour such as eating patterns & ethnic food intake as well as sociodemographics like psychosocial factors & perceived nutrition environment.

5.2.1 Limitations

Although our project has been an extreme success, a few caveats have lowered the accuracy of our models and caused hindrances in our analysis. These pitfalls and their solutions should be noted before proposing this geo-spatial deep learning method for future consumer nutrition environment research. Firstly, when comparing both our models, Model A and Model B, we see that our Calorie Mama model outperforms our Detectron Mask R-CNN model. The main reason for this is the insufficient amount of training data when compared to Calorie Mama’s database. As our local Mask R-CNN was only trained on 1) our Hartford restaurant food data and 2) AICrowd Food Recognition Challenge Database [31], the Calorie Mama model does often surpass the local Mask R-CNN model. By adding in more data through various other crowdsourced sources such as Yelp, Zomato, and Facebook, our detectron model can be easily trained to perform better.

In addition, due to us using the Calorie Mama API, we are held by their restrictions. For instance, While analyzing the data from our model, we found out that the API does not really recognize the calorie and portion size relation very well. A simple example of this is shown below in Figure 5.2. Image (A) shows a crowdsourced image from our restaurant and food data containing six chicken nuggets. Although the API correctly recognized them as chicken nuggets, when providing the calories, it returned 40 calories (calories of a single nugget)- all when the serving size is 1 unit. This caused us issues when analyzing as usually, 1 unit implies everything in the image and on the plate, not just providing the calorie count of a single nugget when the image has six of them. For further analysis, we decided to put in a zoomed in picture of the nuggets in the Calorie Mama model to observe any discrepancies. Surprisingly, we the zoomed in Image (B), the model wrongly identifies the nuggets as Tater Tots with accurate calorie count- causing even more discrepancy in our data. Even though the Calorie Mama model was highly accurate in most cases, certain cases with several similar food items such as nuggets, pieces, etc. was hard for the model to generate nutritional information to. We emailed Calorie Mama to standardize their calories across food items so that we can utilize the modified API in the next stages.

Furthermore, we did not take into account the social factors of an obesogenic environment during our study. Although this aspect of the community directly factors into the nutritional landscape, despite worthwhile attempts, we did not find a compatible solution with our proposed method. This is due to the lack of social food data available online per community as most community members do not utilize social media to share neighbourhood food on a regular basis. Finally, as stated earlier, it must also be noted that all our finding was a part of the pilot study and relied on data in the form of selected crowd-sourced menu food items from all restaurants in
5.3. Future Work

Our work of classifying crowd-sourced food images was a daunting task. From generating images online to implementing our nutritional databases, the whole project was no easy task; Especially, due to recognition issues in intra-class variation of food images such as drinks and seeds. Although the results we gathered from this pilot study show a show a definitive answer of putting a dent in community food culture, we have much more to explore. Our project was done only in the Hartford county with a proprietary rather small dataset. Even though this dataset was meant to be small and focused as it was meant to be in a specified area, our work can be further extended in many ways. Yet, an addition of area or specific other test studies could be a great direction to head in the future.

From our study we found convolutional neural networks to work the best for our data. Yet, there are still ways we can increase our training and test accuracies. A more exhaustive hyperparameter search can be a solution for this. In addition, more models in the area of semantic segmentation and instance segmentation such as other MASK R-CNN systems as well as more architectures like Yolact and Poly-YOLO were not explored during our work. As these Neural Networks (NNs) would shed more light in our extended project, a possible extension involving in trying different architectures would be valid future work.

In addition, given our study in Deep Neural Networks such as the Mask R-CNN, our project can be extended by creating an individualized unique Neural Network that utilizes similar components such as convolution layers and also focuses on our data type- Food Images.

Furthermore, we believe that the models trained using our Hartford food and AICrowd datasets can be migrated to various other food datasets as well We draw this conclusion because of the accuracy increase we observed after adding in the Tripadvisor API images. As we saw more precise identifications after three extra images per restaurant, adding in more distinct and different datasets such as images from the Yelp API would make our results more reliable.
Chapter 5. Conclusion

Just like most machine learning architectures, our system performance was restricted due to hardware limitations. Enhanced hardware in the future would make it possible to have our models training with more epochs.

An extension to the project would be to re-train the models specifically to recognize images within a specific food subset. For instance, training the model to detect a particular type of food such as vegetables, seeds, juices or noodles would get rid of the intra-class variance and similar food confusion errors.

5.4 Project Impact

We believe that our project has a very large impact potential. Given that we already had a substantially high accuracy score on a subset of data in a small location, it will be possible to create a national-wide or world-wide instantaneous food image classifier. This implementation can be done as an application that uses captured food images and the Global Positioning System to provide an accurate Nutritional Fact Label for restaurant and store bought food.

In addition, this application can utilize the users data to construct a community food landscape to observe each neighbourhood’s consumed food for further analysis. This, in turn, can help gauge the health of a specific community and observe links between bad nutrition, poverty, unequal distribution of wealth, etc.

Also all the information returned from images that are screened by our system could be continuously compared to a reputed nutritional food database such as the FoodData Central [33] to ensure accuracy of data before being provided to the user. As we have already analyzed the generated data with FoodData Central for a variety of our testing data, we believe that our gathering information is highly accurate, especially when our training data size is considered.

Further, the system that our project has created can be extended by using the information generated in several ways. For instance, an application created that replicates our end-to-end food image pipeline can be continued by implementing an allergen detection system as well. As our models generate the nutritional information of the food image captured, a user could input their allergies in the system to which the model can provide warnings if it detects food allergic to them. This would be extremely helpful in many situations especially if the user is not able to communicate with the cook or does not understand the nutrition fact label language.
Appendix A

Code & Github Link

A.1 Where is the project code available?

All of the open-source parts of the Food Image Classification (FIC) Research Project working with Dr. Caiwen Ding, Dr. Peter Chen, Dr. Ran Xu at the University of Connecticut is stored on my Github.

The Github link: https://github.com/adikulkarni11/FoodImageClassification
Acknowledgements

First and foremost, I would like to express my sincere gratitude to my project supervisors Dr. Caiwen Ding and Dr. Xiang Chen for allowing me to work on this project for the past two years. I have learned an immense amount from this research opportunity that has refined my knowledge in the field of Computer Science and Artificial Intelligence. I would also like to thank them for their continued guidance and support throughout my final years at the University of Connecticut that in turn sparked an interest in machine learning research and convinced me to pursue further graduate studies.

Also, I would like to convey special thanks to Dr. Ran Xu for the extended advice and support during the term of the project. In addition, I wish to acknowledge Evelyn Johnson and the rest of Dr. Xu's for manual coding and helping shape the Hartford food environment.

In addition, I would like to thank Apoorva Kulkarni and Eric Wang for proof reading this honors thesis.
Bibliography


[28] Content API. URL: https://developer-tripadvisor.com/content-api/.


