

Using AI Computer Vision to Predict Allergens in Food Images

Luke Foley T00224345

Computing with Software Development

# Abstract

…………………………..

Table of Contents

[Abstract 2](#_Toc184603603)

[Introduction 5](#_Toc184603604)

[Chapter 1: Literature Review 6](#_Toc184603605)

[1.1 Computer Vision 6](#_Toc184603606)

[1.1.1 What is Computer Vision? 6](#_Toc184603607)

[1.1.2 History of Computer Vision 6](#_Toc184603608)

[1.1.2 How Computer Vision Works 7](#_Toc184603609)

[1.1.3 Computer Vision Tasks 8](#_Toc184603610)

[1.1.4 Computer Vision Models 10](#_Toc184603611)

[1.1.5 Applications of Computer Vision 10](#_Toc184603612)

[1.2 Machine Learning 11](#_Toc184603613)

[1.2.1 Machine Learning Methods 12](#_Toc184603614)

[1.3 Deep Learning AI Frameworks 13](#_Toc184603615)

[1.3.1 Pytorch 13](#_Toc184603616)

[1.3.2 TensorFlow 13](#_Toc184603617)

[1.4 Allergens 14](#_Toc184603618)

[Chapter 2: Methodology 15](#_Toc184603619)

[2.1 Overview 15](#_Toc184603620)

[2.2 Research Question 15](#_Toc184603621)

[2.3 Technologies 16](#_Toc184603622)

[2.3.1 Python 16](#_Toc184603623)

[2.3.2 PyCharm IDE 16](#_Toc184603624)

[2.3.3 Selenium Library 16](#_Toc184603625)

[2.3.4 Beautiful Soup Library 17](#_Toc184603626)

[2.4 Design 17](#_Toc184603627)

[Chapter 3: Implementation 18](#_Toc184603628)

[4.1 Prototype 18](#_Toc184603629)

[4.1.1 Python Environment 18](#_Toc184603630)

[4.1.2 Scraping Data from an Individual Page 18](#_Toc184603631)

[4.1.3 Scraping Pipeline 20](#_Toc184603632)

[4.1.3 Initial Data Cleaning 21](#_Toc184603633)

[References 22](#_Toc184603634)

[Figure 1. A Diagram CNN Layers 8](#_Toc184603635)

[Figure 2: Multiclass vs Multilabel Classification 9](#_Toc184603636)

[Figure 3: Classification vs Object Detection 10](#_Toc184603637)

[Figure 4 Link Scraping Script 20](#_Toc184603638)

[Figure 5 Automated Scraping Script 20](#_Toc184603639)

# Introduction

Allergic reactions to food have become a growing concern for individuals worldwide, with the prevalence of food allergies rising in recent years. These reactions can range from mild discomfort to life-threatening anaphylaxis, making allergen identification vital for those affected. This be challenging for individuals with visual impairments or when encountering unfamiliar foods.

In recent years, computer vision has become a powerful tool for analysing images and recognising objects. This research investigates how deep-learning computer vision models can be utilised in the identification of potential allergens in foods.

# Chapter 1: Literature Review

## Computer Vision

### What is Computer Vision?

Computer vision is a branch of artificial intelligence (AI) that aims to enable machines to interpret visual information from the world in a manner similar to humans. It involves developing algorithms and systems to process, analyse, and extract meaningful information from images, videos, and other visual inputs. While AI allows computers to think, computer vision enables them to see, observe, and understand (IBM, 2021a).

Computer vision utilises various techniques, including image processing, machine learning, and deep learning. Traditional computer vision methods used manual feature crafting and algorithms, but systems have become much more capable with advancements in deep learning, especially convolutional neural networks (CNNs). These AI-driven techniques have resulted in breakthroughs in areas such as medical imaging, autonomous driving, and augmented reality (Szeliski, 2022).

### 1.1.2 History of Computer Vision

The evolution of computer vision has been closely linked to advances in computational power and algorithm development. According to Szeliski, the origins of the computer vision field was in the 1970s with research aimed at mimicking human vision by interpreting visual inputs through geometric models and image-processing techniques. Early computer vision focused on tasks such as edge detection and simple pattern recognition, but these were limited by the computational constraints at the time. For example, Marr’s theory of vision in the 1970s proposed a multi-stage process for understanding visual scenes, featuring both edge detection and 3D scene reconstruction. However, progress was slowed by the lack of sufficient hardware capabilities to implement these theories at scale (Szeliski, 2022).

During the 1980s and 1990s, computer vision research focused on understanding more complex visuals, such as motion analysis and 3D object recognition. This was supported by introducing new mathematical models and optimisation techniques. Algorithms like the Hough transform for detecting shapes and early approaches to stereo vision laid the foundation for more advanced object detection and scene understanding systems (Szeliski, 2022). However, these early methods often depended on hand-crafted features and models, which required extensive domain knowledge and manual tuning.

A major change occurred in the early 2000s with the introduction of machine-learning-based methods for image recognition. Feature-based techniques such as Scale-Invariant Feature Transform (SIFT), allowed for more accurate detection of objects under varying conditions (Szeliski, 2022).

The rise of deep learning in the 2010s significantly transformed the computer vision field. Convolutional neural networks (CNNs), which were first applied to computer vision tasks in the 1980s, gained widespread adoption after the success of models like AlexNet in the ImageNet competition in 2012. These deep learning models significantly outperformed traditional methods by automatically learning hierarchical feature representations from data, eliminating the need for manual feature extraction (Szeliski, 2022).

Today, computer vision continues to evolve rapidly, driven by advances in deep learning architectures, the availability of large datasets, and improvements in computational hardware such as GPUs and TPUs. These advancements have enabled breakthroughs in applications ranging from autonomous driving to medical image analysis (Szeliski, 2022).

### How Computer Vision Works

Computer vision requires a large amount of data. It analyses this data repeatedly to identify differences between images, and ultimately recognise them. Two key technologies are used in computer vision to accomplish this – deep learning, a type of machine learning, and convolutional neural networks (CNN) (IBM, 2021a).

#### 1.1.2.1 Deep Learning

Deep learning is a subset of machine learning that employs multilayered neural networks called deep neural networks. It aims to replicate the intricate decision-making capabilities of the human brain (IBM, 2024).

The main difference between traditional machine learning models and deep learning models is the structure of the neural network architecture. Whereas traditional models use simple neural nets with only one or two layers, deep learning models use three or more – but typically hundreds or thousands of layers (IBM, 2024).

#### 1.1.2.2 Convolutional Neural Networks (CNNs)

Convolutional neural networks (CNNs) are a specialised type of deep learning architecture designed specifically for analysing visual data. Unlike traditional neural networks, CNNs are particularly effective at processing images by treating them as collections of pixels and using a system of labels to categorise each pixel (Amazon, 2024a). The labelled pixels are then processed using convolution, a mathematical operation that enables the network to extract meaningful patterns from the image. CNNs take a multi-layered approach to extract these patterns progressively, where each layer of the network plays a different role. At a high level, a CNN comprises three main types of layers: convolutional layers, pooling layers, and fully connected layers. The convolutional layers determine the output of neurons that are connected to local regions of the input by calculating the scalar product between their weights and the region connected to the input volume (O’Shea and Nash, 2015). This mechanism enables CNNs to detect features like edges, textures, or more complex shapes in images. CNNs are particularly effective for tasks such as object detection, recognition, and image classification (Voulodimos et al., 2024).

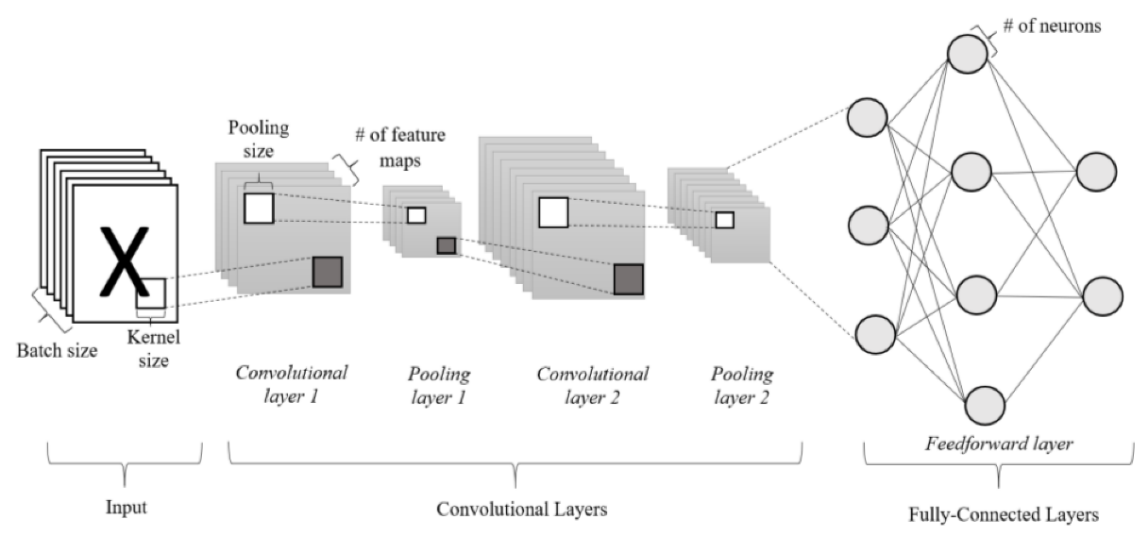


Figure 1. A Diagram CNN Layers

### Computer Vision Tasks

Computer vision is capable of performing various tasks. The following are some examples of established computer vision tasks:

#### Image classification (Multi-class classification)

Image classification is one of the most fundamental tasks in computer vision. It can be defined as the task of categorising an image into one of several predefined classes. For example, a classification model might take an image and label it as “dog” or “cat” based on its learned patterns. It forms the basis for other more complex tasks such as object detection, localisation, and segmentation (Rawat and Zenghui Wang, 2017).

#### Multilabel Image Classification

Multi-label image classification is a machine-learning task where a single image can be associated with multiple labels simultaneously. This is opposed to the traditional task of single-label classification (i.e., multi-class), where each instance is only associated with a single class label (Read and Perez-Cruz, 2014).

A dog and a plant

Description automatically generated

Figure 2: Multiclass vs Multilabel Classification

#### Object Detection

Object detection is a more advanced technique that not only classifies objects within an image but also localises them with bounding boxes. “Object detection is the process of detecting instances of semantic objects of a certain class (such as humans, airplanes, or birds) in digital images and video” (Voulodimos et al., 2024). Unlike image classification, which assigns labels to the entire image, object detection identifies and classifies multiple objects within the image while also pinpointing their locations.

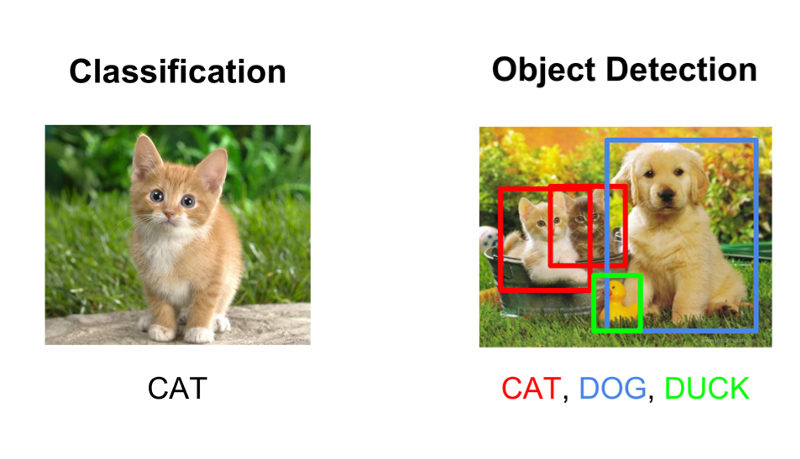


Figure 3: Classification vs Object Detection

### Computer Vision Models

……..

### Applications of Computer Vision

Computer vision has become an integral part of various industries today, from entertainment to healthcare, and everyday life. The following are some real-world use cases of computer vision outlined by Amazon (Amazon, 2024a).

* Security and Safety: Computer vision is used to improve the safety of public spaces, industrial sites, and high-security areas to detect unusual activities like unauthorised access. Similarly, computer vision can improve personal safety. For example, computer vision ensures compliance with safety protocols, such as workers wearing personal protective equipment in workplaces.
* Operational Efficiency: Computer vision can be used in business operations, such as automatically identifying product defects before shipment, detecting maintenance and safety issues in machinery, and facial recognition for employee identification.
* Healthcare: Healthcare is one of the leading industries that apply computer vision technology. Computer vision is used to improve diagnosis and treatment with tools like automatic X-ray analysis, tumour detection, and MRI-based symptom discovery.
* Autonomous Vehicles: Autonomous vehicles rely heavily on computer vision to recognise real-time images and create 3D maps from camera inputs. This enables them to detect other road users, signs, pedestrians, and obstacles to ensure safe navigation.  
  In semi-autonomous vehicles, computer vision uses machine learning to monitor driver behaviour. For example, signs of distraction, fatigue and drowsiness based on the driver's head position, eye tracking, and upper body movement. Alerts are issued to the driver to reduce the likelihood of accidents.
* Agriculture: In agriculture, computer vision is used to boost productivity and reduce costs through automation. Satellite imaging and drone footage are used to analyse large areas of land to predict weather and crop yields. Animal monitoring with computer vision is another strategy in smart farming.

## Machine Learning

Machine learning is a branch of artificial intelligence that allows systems to learn and improve on their own without the need for explicit programming. This is achieved by feeding the system large amounts of data. As these systems gain more “experience,” they can continually adjust and enhance their performance. Providing larger and more diverse datasets can further improve the effectiveness of these systems (Google Cloud, 2024).

Machine learning employs various algorithms that learn from data iteratively to improve, describe, and predict outcomes. As these algorithms process training data, they can develop increasingly accurate models based on that information. A machine learning model is the result produced when a machine learning algorithm is trained with data. After the training process, when data is entered into the model, it generates an output. For instance, a predictive algorithm will create a predictive model. When you provide this model with data, it will deliver a prediction based on the patterns identified in the training data (Hurwitz, 2018).

### Machine Learning Methods

IBM categorises machine learning models into three main types: supervised, unsupervised, and semi-supervised learning. (IBM, 2021b).

#### Supervised

Supervised learning uses labelled datasets to train algorithms for accurate data classification or outcome prediction. As data is input into the model, it adjusts its weights iteratively until it becomes well-tuned. This process includes cross-validation to avoid issues of overfitting or underfitting. Supervised learning allows organisations to address various large-scale, real-world problems, such as categorising spam emails into separate folders. Common techniques in supervised learning include neural networks, naïve Bayes, linear regression, logistic regression, random forests, and support vector machines (SVM).

#### Unsupervised

#### Unsupervised learning uses algorithms to examine and group unlabeled data into clusters to find hidden patterns without needing human help. This allows it to spot similarities and differences in data, making it great for exploring data, segmenting customers, cross-selling products, and recognising images and patterns. It also helps make models simpler by reducing the number of features through techniques like principal component analysis (PCA) and singular value decomposition (SVD). Other common algorithms in unsupervised learning include neural networks, k-means clustering, and probabilistic clustering methods.

#### Semi-supervised

Semi-supervised learning strikes a balance between supervised and unsupervised learning. It utilises a small set of labelled data to guide the classification and feature extraction process for a much larger, unlabelled dataset. This approach helps address the challenge of having limited labelled data in supervised learning and is particularly beneficial when labelling data is too expensive.

*Reinforcement learning*

Reinforcement learning is an additional type of learning similar to supervised learning but doesn’t rely on sample data for training. Instead, the model learns through trial and error, with successful outcomes reinforced over time.

## Deep Learning AI Frameworks

## Pytorch

PyTorch is a versatile, open-source deep-learning library developed by the AI Research team at Facebook. Since its release in 2016, PyTorch has rapidly gained popularity due to its exceptional flexibility and user-friendly interface (GeeksforGeeks, 2024; Nvidia, 2024). One of PyTorch's standout features is its dynamic computational graph, also known as "define-by-run." This feature allows developers to modify network architecture on the fly, enabling real-time debugging and experimentation. As a result, the model development process is significantly faster (Nvidia, 2024).

In addition, PyTorch has a rich offering of tools and libraries that extend its core functionality, including torchvision for computer vision tasks (TorchVision, 2024). PyTorch has compatibility with GPU acceleration ensures efficient training of large-scale models, making it an excellent choice for developing convolutional neural networks (CNNs) and other deep learning architectures (GeeksforGeeks, 2024).

## TensorFlow

TensorFlow is another deep-learning library developed by the Google Brain engineering team. It was licensed as open source in 2015 and has since gained significant popularity for its flexibility and scalability (GeeksforGeeks, 2024). TensorFlow features a range of libraries and toolkits such as TensorBoard for visualisation. One significant benefit of TensorFlow is its ability to deploy models across diverse platforms, including CPUs, GPUs, Tensor Processing Units (TPUs), and mobile devices (GeeksforGeeks, 2024). However TensorFlow, one drawback is that it has a steep learning curve compared to other frameworks like PyTorch.

## Allergens

An allergy happens when the immune system reacts excessively to a harmless substance, wrongly identifying it as a danger and initiating an immune response. These substances are known as allergens and can cause symptoms when ingested, inhaled, or come into contact with the skin (Allergy Ireland, 2024).

Allergens can lead to various health issues, such as food allergies, anaphylaxis, allergic rhinitis, and contact dermatitis. Furthermore, allergens that are inhaled can worsen atopic conditions, including asthma, atopic dermatitis (eczema), and hives (Allergy Ireland, 2024).

Food allergies represent a growing health problem worldwide, and it is estimated that 1 in 10 adults are affected (Tanno and Demoly, 2022).

#### 1.3.1 Food Allergens Allergen Categories

The Food Safety Authority of Ireland defines 14 major allergens that businesses must declare if used in their food (Food Safety Authority, 2024). These Allergens are as follows:

* Cereals containing gluten
* Crustations
* Eggs
* Fish
* Peanuts
* Soybeans
* Milk
* Nuts
* Celery
* Mustard
* Sesame seeds
* Sulphur dioxide and sulphites
* Lupin
* Molluscs

# Chapter 2: Methodology

## 2.1 Overview

The primary objective of this project is to develop a computer vision system that can accurately predict the presence of the 14 major food allergens in food images. To achieve this, as researched in the literature review, a Convolutional Neural Network (CNN) will be utilised. The approach taken to train the model will be multilabel classification training, as each image can be associated with any number of the 14 allergens.

To train a CNN model effectively, a large and well-annotated dataset is essential. Due to the specific nature of the data required, this dataset will be gathered by scraping a large quantity of food images and their corresponding ingredient lists from reputable recipe websites.

Once the data is gathered, the data will be pre-processed and mapped to the predefined list of the 14 major allergens. The final dataset should consist of food images, with a vector annotation indicating the presence of each allergen.

The PyTorch framework will be used to train the CNN due to its intuitive interface and strong support for CNN architectures. PyTorch also supports GPU hardware acceleration, allowing large datasets and complex models to be handled efficiently, significantly accelerating the training process. Access to an Nvidia RTX 4080 GPU should prove beneficial in training.

Transfer learning will be investigated in the next semester, leveraging pre-trained models for allergen prediction.

## 2.2 Research Question

How can AI computer vision be used to predict the presence of allergens in food images?

## 2.3 Technologies

### Data Wrangling with Python - Programming Training | Insoft Services2.3.1 Python

Python is a versatile and high-level programming language that is widely used across various domains such as web development, data science, software engineering, and artificial intelligence. Its simple English-like syntax makes it easy to learn. Python comes with an extensive standard library and thousands of additional libraries designed for various applications. For instance, NumPy and Pandas for data manipulation, Matplotlib for visualisation, and TensorFlow and PyTorch for AI and machine learning projects (Amazon, 2024b). These tools make Python ideal for AI projects, allowing data scientists and developers to clean data, train models, and implement algorithms with minimal effort.

### A green and black background with white text Description automatically generated2.3.2 PyCharm IDE

PyCharm is an integrated development environment (IDE) created by JetBrains specifically for Python development. It offers a wide range of features, including code completion, real-time error detection, support for virtual environments, and integration with GitHub. Additionally, the professional edition of the software is available for free to students. Its compatibility with various Python libraries, which can be easily installed from the terminal, makes PyCharm an excellent choice for this project.

### Advanced Selenium | RTTS2.3.3 Selenium Library

The Selenium Python library is a tool designed for automating web browsers. It is a tool that allows developers to control web browsers through code, making it useful for interacting with dynamic web pages. Selenium can perform actions such as clicking buttons and navigating through menus, which is helpful for scraping data that loads content dynamically. Selenium supports multiple browsers, including Chrome, Firefox, and Edge (Selenium, 2024).

### Beautiful Soup: Introduction to web scraping with Python2.3.4 Beautiful Soup Library

Beautiful Soup is a Python library designed for web scraping. It is lightweight, user-friendly, and simplifies the process of navigating the structure of web pages. This library allows developers to locate and extract specific elements on a page using tags, attributes, and CSS selectors. One of its strengths is its ability to process raw HTML. However, Beautiful Soup does not handle JavaScript or dynamic content on its own. To enhance its web scraping capabilities, Beautiful Soup can be used in combination with Selenium. Selenium can interact with and render dynamic web pages, capturing the fully loaded HTML content, which can then be passed to Beautiful Soup for parsing and extracting data (Beautiful Soup, 2024).

## 2.4 Design

**Functional Requirements**

A functional requirements analysis was conducted using the MoSCoW Method, which categorises project requirements based on their priority and necessity:

**Must-Have:**

* Python environment with essential libraries
* Data scraping pipeline
* Allergen Mapping and dataset organisation
* Training of CNN with dataset

**Should-Have:**

* Predict the presence of allergens in a food image
* Evaluate model performance

**Could-Have:**

* Transfer learning leveraging pre-trained models
* Develop UI

# Chapter 3: Implementation

## 4.1 Prototype

The initial prototype focused on creating a web scraping pipeline for food images and ingredients, as this is the foundational component required for creating an allergen prediction model. The website chosen for this prototype for gathering this information was Food.com. This decision was influenced by several factors. Food.com offers a comprehensive and extensive collection of recipes of approximately 500k, making it a good choice for this prototype. Each recipe page is structured consistently, which allowed for reusable scripts for each recipe. In exploring the page structure with the browser inspect tool, it was observed that each recipe page has its images and ingredient text embedded directly within the HTML. This made the scraping process simpler as the handling of dynamically loaded JavaScript content was not needed for the recipe page.

### 4.1.1 Python Environment

Firstly, a python environment was set up in PyCharm, and the Beautiful Soup library was installed via pip.

A screenshot of a computer

Description automatically generated  


### 4.1.2 Scraping Data from an Individual Page

After establishing the project setup, tests of the Beautiful Soup library to extract elements from web pages were performed to become familiar with the web scraping process. As the page for each recipe contained the food images and ingredients directly in the HTML structure without any dynamic JavaScript loading, capturing this data was relatively straightforward. The required HTML elements were inspected via browser developer tools. Once the relevant tags and elements were identified, a script was written to extract these elements. The scraped data is organised and stored in a “dataset” folder. Each recipe is assigned its own subdirectory, named with the scraped title of the recipe. All images are saved in an “images” subdirectory. The scraped ingredients are written to an “ingredients.json” file and stored in the recipe subdirectory.

The script was effectively able to gather the necessary data from an individual recipe page, storing the images and ingredient annotations in a structured format.

A screenshot of a computer program

Description automatically generatedA screenshot of a computer program

Description automatically generated

A screen shot of a computer program

Description automatically generatedA black background with white text

Description automatically generated

### 4.1.3 Scraping Pipeline

With a script to scrape the data from a single recipe page working, the next task was to set up a function that will navigate the website and automate the scraping process across multiple recipes. To achieve this, the website search function is used. When the site’s search URL is loaded, ten cards with links to recipes are provided. Each search page can be appended to the URL “https://www.food.com/search/**?pn=2**”, providing ten recipe cards with each page. A script can be written to iterate through each page and pass each recipe URL to the existing recipe scraping function. However, unlike the individual recipe pages, the search results page initially loads a limited set of elements. The recipe cards are loaded dynamically via JavaScript after the initial load. Beautiful Soup is not able to handle this dynamic loading itself. To overcome this, Selenium was used. Selenium interacts with the page in a way that emulates how a real user would. By taking this approach, it ensures that all elements of the page are loaded, before extraction.

A computer screen shot of a program code

Description automatically generatedA computer screen shot of a program code

Description automatically generatedA function was written to extract the URLs from a search page, ensuring that all elements are loaded first with selenium, as shown in figure 4. The link scraping function, and the recipe scraping function, could now be used together to automate the scraping process, shown in figure 5.

Figure Link Scraping Script

Figure Automated Scraping Script

This script was then run on 200 search pages, resulting in approximately 5K recipes scraped.

### 4.1.4 Initial Allergen Mapping

The first sprint also attempted mapping the scraped ingredients to allergens. Firstly,

a dictionary of allergen sources was created based on health websites such as the Celiac Disease Foundation site, which had a comprehensive list of gluten sources.

A screenshot of a computer program

Description automatically generated

Figure Preliminary Allergen Map

With a preliminary mapping defined, a script was written to compare the allergen dictionary against the scraped ingredients dataset. However, this proved ineffective as the scraped ingredients dataset contained many naming variations, contained unnecessary characters, and sometime included measurements. This variability made direct comparisons between the allergen dictionary and the ingredients list virtually impossible.

To address this, the RapidFuzz library was used. This library facilitates fuzzy string matching. This allows a similarity threshold to be set to determine acceptable matches between strings. For example, “eggs”, “egg” and “fresh eggs” would all be matched to the allergen “Eggs”, in the dictionary with an appropriate threshold.

Subsequently, the script was altered to incorporate fuzzy string matching into the allergen mapping process. The script generates a binary allergen vector for each recipe. This vector represents the presence (1) or the absence (0) of each of the 14 predefined allergens. When the script was applied to the dataset with a string similarity threshold set at 80, it produced significantly improved results.

A screen shot of a computer program

Description automatically generatedA screenshot of a computer program

Description automatically generated

Figure Initial Allergen Map Result

Figure Allergen Mapping Script

# References

Allergy Ireland, 2024. *Allergy Ireland: Allergy Statistics and Facts*. [online] Available at: <https://www.allergy-ireland.ie/allergy> [Accessed 8 December 2024].

Amazon, 2024a. *What is Computer Vision? - Image recognition AI/ML Explained - AWS*. [online] Amazon Web Services, Inc. Available at: <https://aws.amazon.com/what-is/computer-vision/> [Accessed 24 October 2024].

Amazon, 2024b. *What is Python? - Python Programming Language Explained - AWS*. [online] Amazon Web Services, Inc. Available at: <https://aws.amazon.com/what-is/python/> [Accessed 19 November 2024].

Beautiful Soup, 2024. *Beautiful Soup Documentation — Beautiful Soup 4.12.0 documentation*. [online] Available at: <https://www.crummy.com/software/BeautifulSoup/bs4/doc/> [Accessed 9 December 2024].

Food Safety Authority, 2024. *The 14 Allergens*. [online] Food Safety Authority of Ireland. Available at: <https://www.fsai.ie/Business-Advice/Starting-a-Food-Business/Allergens> [Accessed 14 November 2024].

GeeksforGeeks, 2024. *Keras vs Tensorflow vs Pytorch*. [online] GeeksforGeeks. Available at: <https://www.geeksforgeeks.org/keras-vs-tensorflow-vs-pytorch/> [Accessed 8 December 2024].

Google Cloud, 2024. *What is Machine Learning? Types & Uses*. [online] Google Cloud. Available at: <https://cloud.google.com/learn/what-is-machine-learning> [Accessed 1 November 2024].

Hurwitz, J., 2018. Machine Learning For Dummies®, IBM Limited Edition.

IBM, 2021a. *What is Computer Vision? | IBM*. [online] Available at: <https://www.ibm.com/topics/computer-vision> [Accessed 24 October 2024].

IBM, 2021b. *What Is Machine Learning (ML)? | IBM*. [online] Available at: <https://www.ibm.com/topics/machine-learning> [Accessed 31 October 2024].

IBM, 2024. *What Is Deep Learning? | IBM*. [online] Available at: <https://www.ibm.com/topics/deep-learning> [Accessed 1 November 2024].

Nvidia, 2024. *What is PyTorch?* [online] NVIDIA Data Science Glossary. Available at: <https://www.nvidia.com/en-us/glossary/pytorch/> [Accessed 8 December 2024].

O’Shea, K. and Nash, R., 2015. *An Introduction to Convolutional Neural Networks*. Available at: <http://arxiv.org/abs/1511.08458> [Accessed 23 October 2024].

Rawat, W. and Zenghui Wang, 2017. Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review. *Neural Computation*, 29(9), pp.2352–2449. https://doi.org/10.1162/NECO\_a\_00990.

Read, J. and Perez-Cruz, F., 2014. *Deep Learning for Multi-label Classification*. Available at: <http://arxiv.org/abs/1502.05988> [Accessed 14 November 2024].

Selenium, 2024. *Getting started*. [online] Selenium. Available at: <https://www.selenium.dev/documentation/webdriver/getting\_started/> [Accessed 9 December 2024].

Szeliski, R., 2022. *Computer Vision: Algorithms and Applications*. Springer Nature.

Tanno, L.K. and Demoly, P., 2022. Food allergy in the World Health Organization’s International Classification of Diseases (ICD)‐11. *Pediatric Allergy and Immunology*, 33(11), p.e13882. https://doi.org/10.1111/pai.13882.

TorchVision, 2024. *torchvision — Torchvision 0.20 documentation*. [online] Available at: <https://pytorch.org/vision/stable/index.html> [Accessed 8 December 2024].

Voulodimos, A., Doulamis, N., Doulamis, A. and Protopapadakis, E., 2024. *Deep Learning for Computer Vision: A Brief Review*. [online] Available at: <https://onlinelibrary.wiley.com/doi/epdf/10.1155/2018/7068349> [Accessed 17 October 2024].