PROJECT PROPOSAL FOR INGRAIDIENTS: AN AI TOOL FOR INGREDIENT IDENTIFICATION AND RECIPE GENERATION.

Felicia Liu
Student# 1006950042
lfelicial.liu@mail.utoronto.ca

Anipreet Chowdhury
Student# 1006914396
anipreet.chowdhury@mail.utoronto.ca

Siddharth Khanna
Student# 1006773341
Sid.khanna@mail.utoronto.ca

Arthur Zhuang
Student# 1006233997
arthur.zhuang@mail.utoronto.ca

ABSTRACT

The "IngrAIdients" project aims to develop a deep learning model capable of identifying ingredients from images of prepared dishes. Leveraging convolutional neural networks (CNNs) and recurrent neural networks (RNNs), the project will extract visual features and predict corresponding ingredients, enabling users to monitor caloric intake, avoid allergens, and simplify dietary management. This model will be trained on large-scale datasets like Recipe1M+, with data augmentation techniques applied for better generalization. The system's architecture will be evaluated against baseline models, including Random Forest-based methods, ensuring a comprehensive approach to ingredient identification. Ethical considerations around data bias and privacy will be addressed through diverse dataset collection and user testing. The final product is envisioned as a mobile application capable of real-time ingredient detection, offering practical and health-conscious benefits for everyday users. —-Total Pages: 8

1 INTRODUCTION

In today's increasingly health-conscious society, the ability to swiftly identify ingredients in food can revolutionize meal preparation, dietary tracking, and nutrition management. The goal of this project is to create a deep learning algorithm that can identify ingredients from a basic picture of a prepared dish. We will assess various model architectures using existing and self-collected datasets to determine the most effective approach. This tool will report the ingredients required to recreate the dish, allowing users to gain nutritional insights by providing a simpler and more effective method for monitoring caloric intake and avoiding allergens. Deep learning is a highly suitable approach for this task due to its ability to process large datasets and analyze specific visual components required to identify key ingredients accurately. This project not only offers useful benefits but also demonstrates the increasing potential of deep learning in routine tasks like cooking and meal planning.

2 ILLUSTRATION

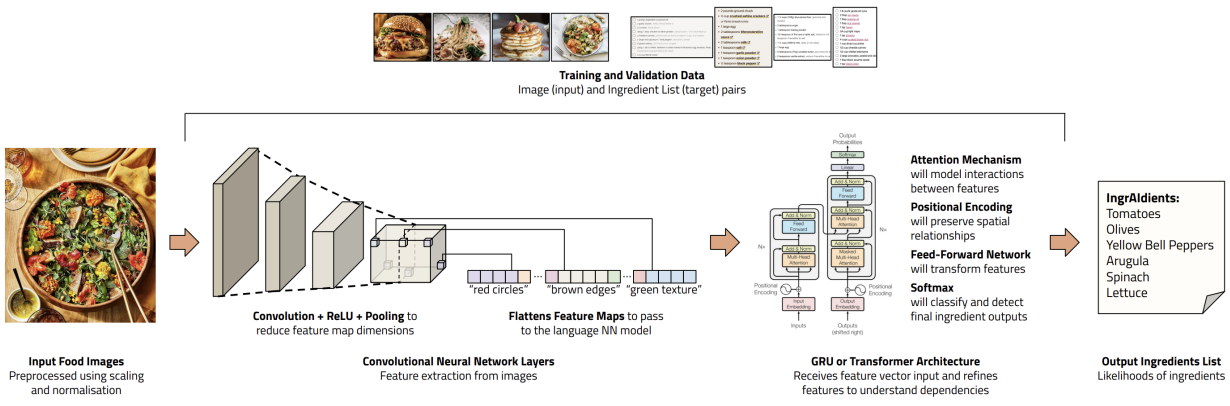

Figure 1: System Context Diagram depicting deep-learning integration

IngrAIdients will feature an app interface where users can take a photo and receive a list of ingredients on their phone. At its core, the system will use a supervised deep learning algorithm trained on pairs of dish images and ingredient lists. It will potentially incorporate a CNN for input image feature extraction and a transformer model for generating natural language text output.

3 Background and Related Work

We referenced five papers to explore architectures for addressing our project proposal. Each article employed different methods for dataset creation and utilized various architectures and machine learning tools, summarized below:

3.1 Ingredient Detection, Title and Recipe Retrieval from Food Images (Chopra and et al., 2023)

This paper implemented a model for ingredient detection, and title and recipe retrieval from food images, with an emphasis on Indian cuisine. Notable features of this project were the custom dataset created for Indian food, the use of transfer learning, and utilization of a transformer for ingredient prediction. Dataset creation involved web scraping well known Indian recipe websites, data processing in the form of standardizing, and lemmatizing ingredient names. The DenseNet model with the Adadelta optimizer was used for image extraction, while the transformer was used to conduct the NLP heavy task of ingredient prediction. This paper also developed a website that accepts an image as input, identifies the dish's name and ingredients, and scrapes the web for matching recipes.

3.2 Sous-chef.ai (Hall, 2021)

This paper developed an AI-powered app for ingredient identification and recipe suggestions allowing users to upload food images, specify preferences, and receive customized recipe recommendations using computer vision and natural language processing. A notable feature of this paper was the implementation of a sliding window approach for multi-object detection, employed to enable the detection of multiple ingredients in a single image. The approach crops smaller sections from a larger image, and filters out noise significantly improving the tools ability to detect ingredients in more complex images with a higher accuracy.

3.3 Recipe Detection of Food Image using Deep learning (CNN) (Tank, 2023)

The key feature of this article was the extensive data collection used to train and optimize the model. This project leveraged widely available datasets like Food-101, UEC-FOOD100, Food-5K, Chinese Food Material dataset, and Food20 dataset. They then used image resizing and normalization to standardize the data, followed by image augmentation techniques like translation, rotation and flipping to diversify available data. Regarding architecture, this article employed a straightforward CNN architecture for ingredient detection, relying on the robustness provided by the comprehensive training data.

3.4 Food Classification from Images Using Convolutional Neural Networks (Attokaren and et al., 2017)

This paper emphasized the effectiveness of CNNs in classifying food from images, particularly their method for estimating calories directly from images. The authors proposed web scraping to map food names to average calorific values based on portion size. This approach can be adapted to estimate food size and identification using our database.

3.5 Food-101 – Mining Discriminative Components with Random Forests (Bossard and et al., 2014)

This paper proposed an alternative approach not involving deep learning, instead using a multi-class Support Vector Machine and a random forest algorithm to classify food from images. While it did not outperform CNNs, it achieved competitive results, making it a valuable baseline for comparison. The paper also introduced an alternative to the computationally expensive sliding window technique by using superpixels - clusters of similar pixels used in image segmentation to simplify recognition tasks and capture meaningful regions. This approach could be combined with our deep learning model to reduce computational complexity while maintaining accuracy.

3.6 Overview

There were several key ideas and techniques that we came across within these papers. The web scraping mechanism described in Ingredient Detection, Title and Recipe Retrieval (Chopra and et al., 2023) will help in creating our dataset, while the sliding window mechanism described in Sous-chef.ai (Hall, 2021) suggests a method of breaking down an image into smaller parts to help increase prediction accuracy. The use of CNNs (specifically resNet) in both Recipe Detection of Food Images (Tank, 2023) and Food Classification from Images using CNNs (Attokaren and et al., 2017) further validates our design choice to use a CNN for image extraction tasks. Finally, the alternative method of identifying ingredients using the random forest algorithm outlined in Food-101 – Mining Discriminative Components with Random Forests (Bossard and et al., 2014) provides a great baseline model to compare our performance to.

4 DATA PROCESSING

We will utilize existing datasets such as Recipe1M+ (Rec) and the Food Ingredients and Recipe Dataset with Images (Foo) as our primary input for training. Recipe1M+ features 1 million recipes and 13 million food images, designed for image-to-recipe and recipe-to-image retrieval, while the Food Ingredients and Recipe Dataset with Images provides diverse recipes and images, structured for ingredient recognition and image-based analysis.

3-Ingredient Charred Green Beans with Ricotta and Lemon	['1 1/2 pounds green beans, trimmed', '2 cups whole-milk ricotta', '3 tablespoons extra-virgin olive oil, plus more for drizzling', '3/4 teaspoon kosher salt, plus more', '1 teaspoon finely grated lemon zest', 'Freshly ground black pepper', 'Lemon wedges (for serving)']	3-ingredient-charred-green-beans-with-whipped-ricotta

Figure 2: Images and text mapping from the Food Ingredients and Recipe Dataset with Images (Foo)

{"ingredients": [{"text": "1 c. elbow macaroni"}, {"text": "1 c. cubed American cheese (4 ounce.)"}, {"text": "1/2 c. sliced celery"}, {"text": "1/2 c. minced green pepper"}, {"text": "3 tbsp. minced pimento"}, {"text": "1/2 c. mayonnaise or possibly salad dressing"}, {"text": "1 tbsp. vinegar"}, {"text": "3/4 teaspoon salt"}, {"text": "1/2 teaspoon dry dill weed"}], "url": "http://cookeatshare.com/recipes/dilly-macaroni-salad-49166", "partition": "train", "title": "Dilly Macaroni Salad Recipe", "id": "808033e39b", "instructions": [{"text": "Cook macaroni according to package directions; drain well."}, {"text": "Cold."}, {"text": "Combine macaroni, cheese cubes, celery, green pepper and pimento."}, {"text": "Blend together mayonnaise or possibly salad dressing, vinegar, salt and dill weed; add in to macaroni mix."}, {"text": "Toss lightly."}, {"text": "Cover and refrigeratewell."}, {"text": "Serve salad in lettuce lined bowl if you like."}, {"text": "Makes 6 servings."}]},

Figure 3: Image and text mapping from the Recipe1M+ (Rec)

Data Cleaning and Preprocessing: All images will be uniformly resized to ensure consistent input for the model, eliminating variability caused by differing image dimensions. Ingredient names will be standardized (e.g., "tomatoes" vs. "tomatos," "leaves" vs. "leaf") to maintain consistency throughout the dataset and ensure accurate model recognition. Incomplete, missing, or duplicate ingredient entries will be corrected or removed to guarantee each recipe is complete and error-free. Additionally, any unnecessary components, such as cooking instructions or irrelevant text, will be eliminated. This helps streamline the data for model training. Quantities and measurement units (e.g., "2 cups of sugar") will also be stripped from ingredient lists, focusing the dataset solely on the presence of ingredients rather than their specific amounts. Furthermore, watermarks, branding, and other extraneous visual elements will be removed from images, ensuring the model is trained only on relevant visual features for ingredient prediction.

Ingredient Tokenization: We will tokenize multi-word ingredients (e.g., "olive oil" or "black pepper") to break them into individual tokens while preserving their meaning. Ingredient lists will be structured into a standardized, machine-readable format, ensuring consistency and improving the model's ability to interpret variations in ingredient names.

Data Augmentation: To diversify the image dataset and improve model robustness, we will apply several data augmentation techniques. These include random cropping, rotations, and color adjustments (e.g., brightness, contrast, and saturation changes). Additionally, horizontal and vertical flipping will introduce further variations, and slight scaling will simulate different zoom levels. Gaussian noise and slight blurring will be applied to mimic real-world imperfections, improving the model's ability to generalize to noisy or lower-quality images.

Validation and Testing: We will build a web scraper to collect similarly structured data from sources such as BBC Good Food (BBC). The scraper will extract essential elements like the dish name, an image of the dish, and its corresponding ingredients. The gathered images will be cropped and resized to match the dimensions of the training data, ensuring consistency for evaluation. This external dataset will help validate the model's performance on new, unseen data, offering a reliable measure of generalization and real-world applicability.

5 ARCHITECTURE

Our network consists of several components. The process begins with feeding input images into a CNN (such as ResNet-50 (He and et al., 2016) or VGG-16 (Simonyan and Zisserman, 2015)) to obtain a vector representation (image embedding). This embedding is then passed through an RNN (GRU (Cho and et al., 2014) or bi-directional LSTM (Schuster and Paliwal, 1997)) to predict ingredients. We will learn a joint embedding space between images and ingredient lists to enable better alignment and comparison. Currently from research, we think this will be a good approach but we aim to explore different architectures such as transformer-based approaches, which may perform better than current solutions.At this stage, our architecture consists of:

5.1 CNN (RESNET-50 (HE AND ET AL., 2016) OR VGG-16 (SIMONYAN AND ZISSERMAN, 2015))

The CNN extracts key image features and converts them into high-dimensional embeddings. ResNet-50, with its residual connections, helps capture complex visual patterns, while VGG-16 provides simpler yet effective feature extraction. Both models are pre-trained on large datasets for improved performance.

5.2 RNN (GRU (CHO AND ET AL., 2014) OR BI-DIRECTIONAL LSTM (SCHUSTER AND PALIWAL, 1997)

An RNN (either GRU or bi-directional LSTM) handles sequence modeling, predicting ingredients from the image embeddings. GRU is efficient with fewer parameters, while bi-directional LSTM improves the model's ability to capture contextual information from both directions in a sequence.

5.3 JOINT EMBEDDING SPACE

We create a shared embedding space where both images and ingredient lists are mapped. This allows the model to measure similarity and align visual and textual data, improving ingredient matching and retrieval tasks.

5.4 OPTIONAL TRANSFORMER MODEL

For a more sophisticated ingredient prediction approach, we may explore transformer architectures to enhance learning across both image and text modalities.

Details such as the number of layers and specific hyperparameters will be fine-tuned during experimentation, utilizing tools like Ray Tune for efficient optimization. We will implement various normalization techniques, including batch and layer normalization, and explore regularization methods such as dropout and weight decay. Additionally, we will apply other tuning strategies learned in class to further optimize model performance.

6 BASELINE MODEL

Our baseline model will use the Random Forest-based discriminative component mining approach as described in the Food-101 paper (Bossard and et al., 2014). This model identifies discriminative parts (or components) of food images by clustering superpixels using Random Forests. Superpixels are groups of pixels with similar visual characteristics, which help reduce the complexity of image recognition tasks by focusing on meaningful regions (Sup). The Random Forest classifier learns to distinguish food categories by mining these components and using them to classify the images. We will follow the same approach of training a Random Forest on superpixel features, then applying a multi-class SVM (Support Vector Machine) for final classification (Bossard and et al., 2014). This method provides a straightforward, non-neural network comparison, as it efficiently mines visual features without relying on deep learning architectures. This would be a useful baseline model to compare with, since this is relatively simpler to make, and had shown comparable results to a CNN model which was trained using traditional deep learning techniques. The paper was able to achieve an average accuracy of 50.76 percent using the random forest model (Bossard and et al., 2014). For a more holistic baseline evaluation, we will also calculate the F1, recall, and precision score of this model.

7 ETHICAL CONSIDERATIONS

A key ethical concern that could arise from this project is the potential for cultural bias if the training data is biased, heavily favoring certain cuisines, which could lead to inaccuracies in recognizing ingredients from underrepresented cultures, marginalizing users from diverse backgrounds, and limiting the model's usefulness for them. Another ethical issue is privacy, as uploading food images could unintentionally expose personal information such as dietary preferences, habits, and even location. Additionally, the model's widespread use could impact food businesses by providing insights into popular ingredients or recipes, which may inadvertently pressure smaller or local businesses to conform to broader food trends, affecting cultural diversity in cuisine. Addressing these concerns requires using diverse, representative datasets, ensuring strong privacy protections, and consideration for the broader social impacts on the food industry and cultural preservation.

8 PROJECT PLAN

A project plan is vital for coordinating the team on objectives, tasks, and timelines, enabling open effective communication and increasing the likelihood of a successful project.

Our team alignment charter describes how our team will work together, by establishing shared values, expectations, and methods to guide us toward collaborative success.

Table 1: Team Charter

Team Goals and Values	• Academic Excellence, Collaborative Teamwork, Learning Growth
Team Expectations and Guideline Methods	• Complete tasks to the best of your ability within team internal deadlines. • Notify the team 2 days in advance if issues arise. • Decisions will be made using a Multi Voting system after discussion. • Resolve conflicts through respectful, open communication. • Hold in-person meetings for better communication and team cohesion. • Share meeting agendas in advance; document minutes for reference.
Communication and Working Platforms	• **Discord** : For general discussion, questions, work updates, announcements, and document sharing. • **Google Drive** : Central hub for all documents, files, and deliverables. • **GitHub** : Group repository for code. • **Cell Phone Numbers** : For urgent contact.

The following table outlines each team member's responsibilities, progress, and deadlines. Tasks are evenly distributed to ensure balance and allow any member to step in if needed. We follow a weekly structure with buffer time, assigning tasks for independent work before syncing up to provide updates, feedback, and plan next steps.

Table 2: Task Distribution: Completed

Team Formation (Sep 17, 2024) COMPLETED	• **Internal Deadline** - Sept 17th (9PM) • **Everyone**: Find teammates and join the group on Quercus to get started. • **Anipreet**: Set up the GitHub and Colab workspace for streamlined coding and collaboration. • **Arthur**: Collect contact info to ensure easy communication among team members. • **Felicia**: Create the Google Drive and Discord to centralize files and facilitate communication. • **Sid**: Organize schedules and coordinate a meeting time to ensure everyone's availability aligns.
Project Proposal (Oct 4, 2024) COMPLETED	• **Internal Deadline** - Oct 2nd (4PM) • **Everyone**: Brainstorm personal interest project ideas, create a shortlist from instructor-provided ideas (Sept 16th), collaborate in meetings to define scope, and research feasibility (Sept 23rd). • **Anipreet**: Deep dive into shortlisted ideas for feasibility, research background and baseline NN models, and convert the final document into LaTeX. • **Arthur**: Search for potential datasets, research data processing and architecture options, and complete those sections of the document. • **Felicia**: Pre-read instructions, set up the document framework (Sept 23rd), create a PM plan and project outlook image, proofread final submission, and complete related sections. • **Sid**: Consult TAs on idea feasibility (Sept 17th), write the project description and introduction.

Table 3: Task Distribution: Future

Progress Report (Nov 1, 2024) TO-DO	• **Internal Deadline** - Oct 28th • **Everyone**: Collaborate on refining the model, develop all technical aspects of the project, resolve any implementation issues, and regularly track project progress. • **Anipreet**: Test various NN configurations starting from baseline model, document the outcomes, and recommend improvements for optimization (Oct 22nd). • **Arthur**: Clean and preprocess the data for training (Oct 15th), explore methods of web-scraping for personalized test data, recommend improvements to enhance the dataset. • **Felicia**: Implement baseline model (Oct 15th), adjust model parameters based on initial results, update the project documentation, and manage LaTeX formatting. • **Sid**: Pre-read instructions (Oct 8th), integrate the user interface with the model, test user interactions, and ensure smooth input/output functionality.
Final Deliverable (Nov 29, 2024) TODO	• **Internal Deadline** - Nov 27th • Everyone: Agree on the content and structure of the final document, continue with technical development tasks, finalize our deep learning model (Nov 18th), and collaborate on revising drafts. • **Anipreet**: Pre-read instructions (Nov 4th), set up the document framework, and write sections on the introduction, baseline model, and ethical considerations. • **Arthur**: Write sections on background & related work, evaluate the model on new data, and handle LaTeX conversion. • **Felicia**: Write sections on data processing, architecture, and discussion. • **Sid**: Write sections on quantitative and qualitative results, project difficulty, and manage LaTeX conversion.
Project Presentation (Nov 29, 2024) TODO	• **Internal Deadline** - Nov 25th • **Everyone**: Collaborate to brainstorm the flow of the demonstration video, create drafts iteratively, provide personal feedback for improvement, and finalize video submission (Nov 25th). • **Anipreet**: Record the primary voiceover and edit the video for clarity and engagement (Nov 25th). • **Arthur**: Pre-read instructions and set up the framework for the video storyboard (Nov 4th). • **Felicia**: Develop the content for the storyboard and script to ensure coherence (Nov 11th). • **Sid**: Handle the recording of the video components, ensuring quality visuals (Nov 18th).

Our plan accommodates our different programs and schedules while maintaining clear communication and progress tracking.

9 Risk Register

As with any complex project, it is important to assess the most impactful risks to ensure both timely delivery and successful outcomes. The following is a comprehensive risk register, outlining the potential risks, their likelihood and impact, and the mitigation strategies we will employ to manage them effectively.

9.1 Biased Training Data

Likelihood: Medium. Data bias is common, especially when datasets lack diversity. While we'll work to collect diverse data, there's still a moderate chance of bias.

Impact: High. A biased model could alienate users from certain cultures, severely limiting its fairness and effectiveness.

Mitigation We will collect diverse datasets that represent various cuisines, and use data augmentation to increase diversity and perform user testing across different cultural backgrounds to ensure fairness.

9.2 Difficulty in Achieving High Model Accuracy

Likelihood: Low. Using techniques like hyperparameter tuning, cross-validation, and reviewing research papers will likely reduce accuracy issues.

Impact: Medium. Accuracy issues could impact the model's effectiveness, though the model would still function with room for improvement.

Mitigation We will improve accuracy by experimenting with different architecture, and use hyperparameter tuning, cross-validation and data augmentation. Additionally, we will reference existing research papers to guide our approach.

9.3 Computational Resource Limitations

Likelihood: Medium. Training machine learning models is resource-intensive, and despite optimizations, resource constraints could still arise.

Impact: High. Limited computational resources could cause delays or force a reduction in model complexity, impacting its performance.

Mitigation We will optimize code to reduce computational load, use smaller data subsets for initial testing, and implement parallel processing where possible. If necessary, we'll schedule resource-intensive tasks during off-peak hours or utilize university-provided computing resources.

9.4 Scope Creep Leading to Project Delays

Likelihood: Low. Clear scope definition and monitoring reduce the risk of scope creep, though some additional features may be introduced.

Impact: High. If scope creep occurs, it could delay the project and shift focus from core tasks, affecting the project's timeline and outcomes.

Mitigation We will define the project scope clearly, regularly review progress, prioritize core functionality, and use project management tools to track timelines.

9.5 Unanticipated Software Bugs and Crashes

Likelihood: Low. Practices like version control and continuous testing make serious bugs unlikely, but minor bugs may still appear.

Impact: Medium or High. Minor bugs may cause small delays, but severe bugs in critical areas could significantly impact project timelines and functionality.

Mitigation We will mitigate software bugs by using version control, implementing continuous testing and integration, and maintaining a structured debugging plan to quickly identify and resolve issues during development.

9.6 Summary

There are other potential risks, such as a team member dropping the course or model training taking longer than expected, but we believe these are not major concerns due to our well-defined team structure, task distribution, and access to necessary resources. The primary risks we have identified, like biased training data and computational limitations, are well-managed with the mitigation strategies we have in place. Overall, we are confident that the risks are addressed effectively, and the project is on track for successful completion.

10 Links to Collab Notebook and Github page

Link to collab notebook
Link to GitHub

References

Welcome to good food. https://www.bbcgoodfood.com/. Accessed: 2024-10-03.

Food ingredients and recipes dataset with images. https://www.kaggle.com/datasets/pes12017000148/food-ingredients-and-recipe-dataset-with-images. Accessed: 2024-10-02.

Recipe1m+: A dataset for learning cross-modal embeddings for cooking recipes and food images - mit. https://im2recipe.csail.mit.edu/. Accessed: 2024-10-02.

Superpixel - an overview — sciencedirect topics. https://www.sciencedirect.com/topics/computer-science/superpixel. Accessed: 2024-10-02.

D J Attokaren and et al. Food classification from images using convolutional neural networks. In *TENCON 2017 - IEEE Region 10 Conference*, pages 2801–2806, Nov 2017.

L Bossard and et al. Food-101 – mining discriminative components with random forests. In *Computer Vision – ECCV 2014*, pages 446–461, Cham, 2014.

K Cho and et al. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar, Oct 2014.

B Chopra and et al. Ingredient detection, title and recipe retrieval from food images. In *2023 Fourteenth International Conference on Ubiquitous and Future Networks (ICUFN)*, pages 289–293, Jul 2023.

B Hall. Using ai to identify ingredients and suggest recipes. https://medium.com/@brh373/using-ai-to-identify-ingredients-and-suggest-recipes-95482e2aca7d, 2021. Accessed: 2024-10-02.

K He and et al. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, Jun 2016.

M Schuster and K K Paliwal. Bidirectional recurrent neural networks. *IEEE Trans. Signal Process.*, 45(11):2673–2681, Nov 1997. doi: 10.1109/78.650093.

K Simonyan and A Zisserman. Very deep convolutional networks for large-scale image recognition. https://arxiv.org/abs/1409.1556, 2015.

D Tank. Recipe detection of food image using deep learning (cnn). https://medium.com/@imdhawaltank/recipe-detection-of-food-image-using-deep-learning-65eb382aeb38, 2023. Accessed: 2024-10-02.