FINAL PROJECT REPORT FOR INGRAIDIENT: AN AI TOOL FOR INGREDIENT IDENTIFICATION

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

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

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

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

—-Total Pages: 9

1 INTRODUCTION

This project is motivated by the growing trend toward health-conscious living, where the ability to quickly identify ingredients in food can enhance meal preparation, dietary tracking, and nutrition management. As more individuals aim to improve their well-being, there is a demand for tools that facilitate nutrition insights. Our objective is to develop a deep learning algorithm that can accurately detect a short list of ingredients (output) from a single image of a prepared dish (input). This tool can assist users in monitoring food intake, understanding nutritional content, and managing dietary needs, making it valuable for a wide range of individuals, from those with specific dietary restrictions to those seeking healthier lifestyles.

Deep learning models, particularly Vision Transformers (ViTs), are ideal for ingredient identification from food images due to their ability to capture global context by processing images as sequences of patches. This enables ViTs to learn complex patterns, differentiate similar ingredients, and scale effectively with large, diverse datasets, ensuring accurate and reliable ingredient detection, addressing our application's requirements.

2 ILLUSTRATION

Our model uses a Vision Transformer (ViT) for feature extraction from preprocessed food images. The ViT processes images by dividing them into patches and capturing intricate visual patterns and spatial relationships. These patterns, such as "red circles" or "green textures," are refined through the transformer's self-attention mechanism to capture contextual dependencies. The model outputs a likelihood-based ingredient list for each image. A diagram of our model is shown in Figure 1.

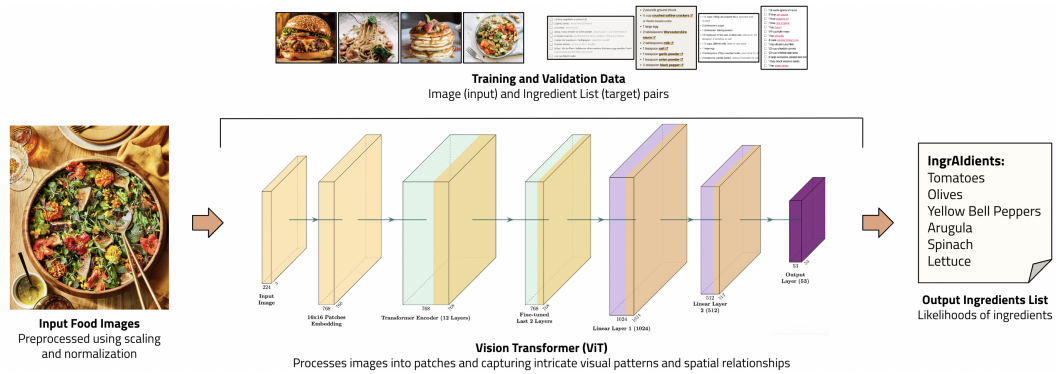

Figure 1: System Context Diagram depicting deep-learning integration.

3 Background and Related Works

Several studies have provided valuable insights into ingredient detection from food images, influencing the development of both our baseline model and deep learning architecture as well as our approach to dataset selection and data processing.

Baseline and Model: Initially, we considered using a CNN (e.g., ResNet) for ingredient detection through multiclass classification, influenced by the work in Attokaren et al. (2017) which demonstrated the effectiveness of CNNs for food classification and calorie estimation, successfully linking food names to calorific values. However, as we delved deeper into the research, we found a more promising alternative in Vision Transformers (ViTs). As discussed in Maurício et al. (2023), ViTs significantly outperform CNNs for this task, achieving 96.45% accuracy, 86.38% validation accuracy, 10.8% loss, and 18.25% validation loss. Based on these results, we decided to use the ViT as the foundation for our deep learning model. For our baseline, we referenced the approach by Bossard et al. (2014), who proposed a non-deep learning method using superpixel segmentation for food classification. While this method is less effective than ViT, it provides competitive results and offers better computational efficiency, making it a useful comparison for our baseline SVM model.

Data Processing and Training: Hall (2021) introduced an AI app for ingredient identification and recipe suggestions, utilizing a sliding window technique for detecting ingredients in complex food images. This approach enhanced multi-object detection accuracy, and hence, influenced our training strategy. Tank (2023) emphasized using large, diverse datasets like Food-101 and Food20, applying extensive preprocessing (e.g., resizing, normalization, augmentation) to improve feature-based ingredient detection. This informed our preprocessing pipeline and the use of data augmentation to enhance model robustness.

These studies collectively informed our choices in dataset creation, data processing, model architecture, and baseline comparison, helping shape our understanding of robust approaches to ingredient detection from food images.

4 Data Processing

Collected Data: Our baseline and deep learning models are trained on Recipe1M+ (Rec), a large and diverse dataset with over 51,000 valid recipes and 700,000 images, which provides a robust foundation of data variety, helping the model to generalize across different cuisines and food types. The Recipe1M+ dataset is well-structured in JSON format, with ingredients listed in a consistent noun-first format (e.g., ”wheat flour, white, all-purpose”), simplifying ingredient extraction. Each recipe is paired with at least one image, often multiple, introducing natural variations in lighting and cropping that enhance model robustness.

Text-Based Ingredient Processing: Each recipe in Recipe1M+ is identified by a unique ID, with ingredients stored as text strings. Irrelevant metadata was removed by eliminating brackets, splitting strings by commas, trimming whitespace, and organizing ingredients into a list. Heuristic rules extracted base ingredient names, focusing on the main ingredient and its primary descriptor while excluding non-informative terms like ”raw” and generic names like ”spices.” Inconsistencies were resolved by treating plurals as identical, removing brand names (e.g., ”Delallo”), and remapping terms like ”catsup” to ”ketchup.” Cleaned ingredient lists, with duplicates removed, were saved in text files named by recipe IDs.

Ingredient List Rebalance Processing: Statistical analysis revealed significant class imbalance in the initial dataset, with ingredient frequencies ranging from 1 recipe (e.g., ”anchovy”) to 20,000 recipes (e.g., ”salt”). Over 175 ingredients appeared fewer than 100 times, while fewer than 10 ingredients had counts exceeding 5,000. To address this, we consolidated 375 ingredients into 32 broader categories (e.g., meats, fruits, nuts, flours, root vegetables) and further divided large categories like ”flour” into subcategories (”flour1,” ”flour2,” ”flour3”) to balance counts. This restructuring yielded 53 unique classes, with ingredient counts ranging from 3,500 to 6,000, significantly improving balance. Additionally, general ingredients like ”salt” and ”sugar” were removed as they provided minimal predictive value. Figure 2 shows the transformation of the data before and after text-based processing and dataset rebalancing. It’s evident the generalized and rebalanced ingredients still retain the essence of the original items. Figure 3 compares the unbalanced and balanced

distributions of ingredient statistics across recipes, highlighting ingredient counts and frequency disparity becoming balanced.

Image-Based Photo Processing: Recipe1M+ provides a second JSON layer that associates each recipe ID with multiple images. Downloaded from the MIT server, images were resized to 224x224 pixels with center cropping to preserve aspect ratio, and converted to RGB, discarding other formats. Original 0-255 RGB values were retained to optimize storage. We chose not to apply any additional augmentation, as natural variations in lighting, angles, and resolutions for each recipe provided sufficient diversity. Figure 4 shows the processed images.

Final Processed Dataset and Statistics: The dataset is organized into ”images” and ”labels” folders, each containing training (70%), validation (15%), and test (15%) subfolders, created using the split-folder library. Images for each recipe are stored in subfolders named by their recipe IDs, with corresponding ingredient lists saved as .txt files in the labels folder, ensuring a clear link between images and ingredient data. The dataset, rebalanced into 53 classes, covers 78,000 websites and 50,768 recipes with over 700,000 images and 32 processed unique ingredients, averaging 14.14 images and 4.49 ingredients per recipe. The broad ingredient grouping and balanced class distribution prevent ”unknown” data from appearing in the dataset and enable the model to handle all variations in the dataset. Figure 3, again, visually confirms the balanced ingredient counts across the recipe.

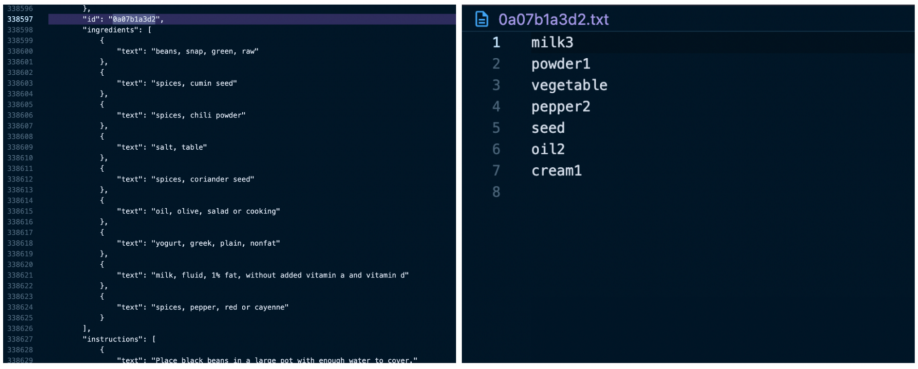

Figure 2: Raw Ingredient List label information in JSON format (left). Cleaned and Balanced Ingredient List label information in TEXT format (right).

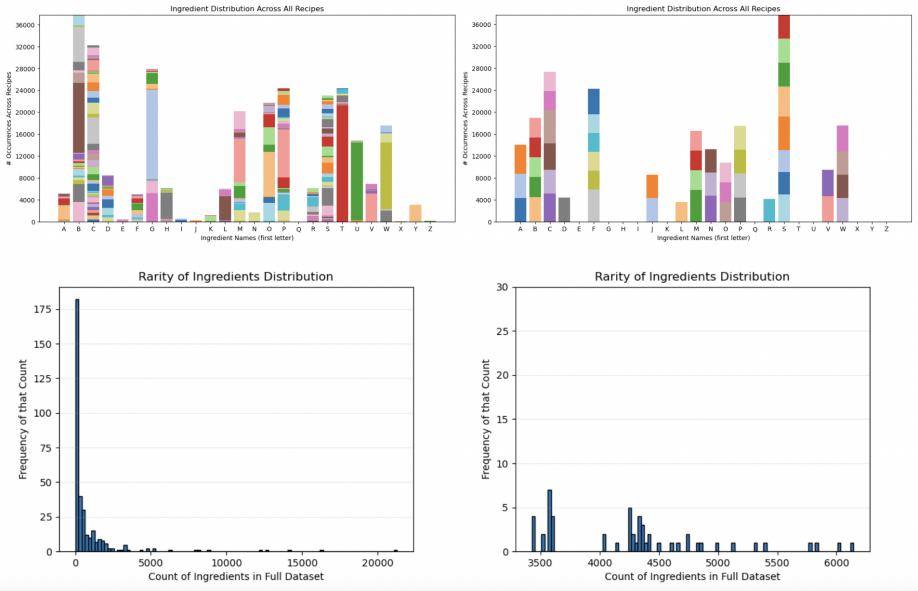

Figure 3: Unbalanced (left set) VS Balanced (right set): Statistics of Ingredient Distribution Across Recipes and Rarity of Ingredient Count Distributions.

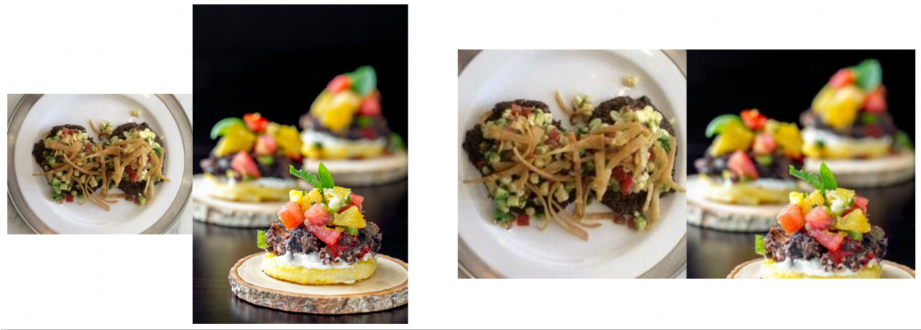

Figure 4: Comparison of Recipe ID: 0a07b1a3d2 before and after cropping.

5 ARCHITECTURE

The final model is based on the Vision Transformer (ViT) architecture, specifically the vit_b_16 variant pre-trained on ImageNet, chosen for its ability to capture both global and local dependencies through self-attention mechanisms. This pre-trained backbone reduces training time and enhances feature extraction. We replaced the ViT's pre-trained classification head with a custom feedforward network tailored for multi-label ingredient detection consisting of three fully connected layers with dimensions $768 \rightarrow 1024 \rightarrow 512 \rightarrow$ output size (53). Each intermediate layer includes Layer Normalization and ReLU activation to ensure stable training and add non-linearity. Dropout layers with rates of 0.3 and 0.2 are included to prevent overfitting, especially since Vision Transformers tend to overfit on smaller datasets. Initially, the parameters of the pre-trained ViT backbone were frozen to preserve the general features learned during pre-training, with only the custom classification head being updated. Later, the last two transformer layers of the ViT encoder were unfrozen for fine-tuning. This gradual unfreezing approach helped the model adapt to ingredient detection while retaining the generalizable features from the pre-training phase.

In terms of the processing pipeline, input images are tokenized into patches and processed through the ViT backbone to extract high-level features, which are then passed through the custom classification head for final multi-label predictions. This architecture's robustness, flexibility, and ability to adapt to specific tasks make it well-suited for ingredient detection, ensuring efficient and high-performance results. The model can be seen in Figure 1 given above.

6 BASELINE MODEL

For our baseline model, we created a classifier that uses a Support Vector Machine (SVM) to solve the same issue of multi-label classification. The baseline model consists of two main components: dividing the image into superpixels and extracting features, followed by classification based on these extracted features. To implement this, we implemented a One-vs-Rest Classifier wrapped around a simple Support Vector Classifier (SVC) from the SciKit-Learn library. The One-vs-Rest Classifier transforms the multi-class problem into several binary classification tasks, training a smaller SVC model for each class. We configure the smaller SVC models with Radial Basis Function (RBF) kernels to enable non-linear classification. Additionally, we ensure that the model generates probabilities as outputs and adjust the class weights to account for imbalances in class frequencies. The model can be seen in Figure 5.

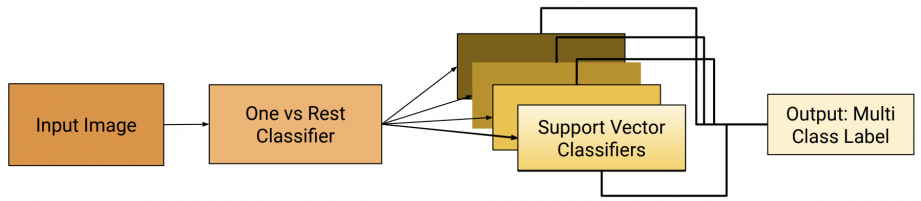

Figure 5: Baseline Model SVM Implementation.

The large computational demands of creating individual small SVC models and the additional data processing required to form superpixels resulted in long training times. To address this, we reduced the dataset size by sampling only 10% of the original data, which made training more manageable and reduced computational costs. To evaluate model performance, we measure the F1 score, precision, and recall, allowing for an effective comparison with the primary model.

7 Quantitative Results

This section presents the performance results of the Vision Transformer (ViT) model and the Support Vector Machine (SVM) baseline using both the training and test datasets. To evaluate the models, we used Precision, Recall, and F1 Score metrics. However, for a fair evaluation in the multi-label classification task, we implemented custom evaluation metrics that grouped labels with numerical suffixes (e.g., nut1, nut2) under their base label (e.g., nut). This ensured that predictions for any variant of a grouped label were treated as correct if the corresponding ground truth included any label from the group and this adjustment was particularly crucial to address label splits introduced for balancing purposes.

The quantitative results are given in Table 1 and Figure 6 below, and show that the ViT model outperforms the SVM baseline on the test set, with higher F1 score, precision, and recall, indicating better generalization and making it more suitable for the multi-label classification task. During training, the ViT achieved a Precision of 0.3111, Recall of 0.4336, and F1 Score of 0.3343, reflecting its focus on balancing precision and recall. In contrast, the Support Vector Machine (SVM) baseline achieved a Precision of 0.3661, Recall of 0.9527, and F1 Score of 0.5290, indicating an overemphasis on recall at the expense of precision. The test set results further highlighted these differences: the ViT attained a Precision of 0.3153, Recall of 0.4447, and F1 Score of 0.3401, while the SVM dropped sharply to 0.1253, 0.3619, and 0.1861 respectively. These results underscore the SVM's overfitting tendencies and inability to generalize effectively, while the ViT maintained consistency and adaptability across datasets.

Table 1: Performance Data of SVM (Baseline) and ViT (Model) on Training and Testing Data.

Metric	SVM (Training)	ViT (Training)	SVM (Test)	ViT (Test)
Precision	0.3661	0.3111	0.1253	0.3153
Recall	0.9527	0.4336	0.3619	0.4447
F1 Score	0.5290	0.3343	0.1861	0.3401

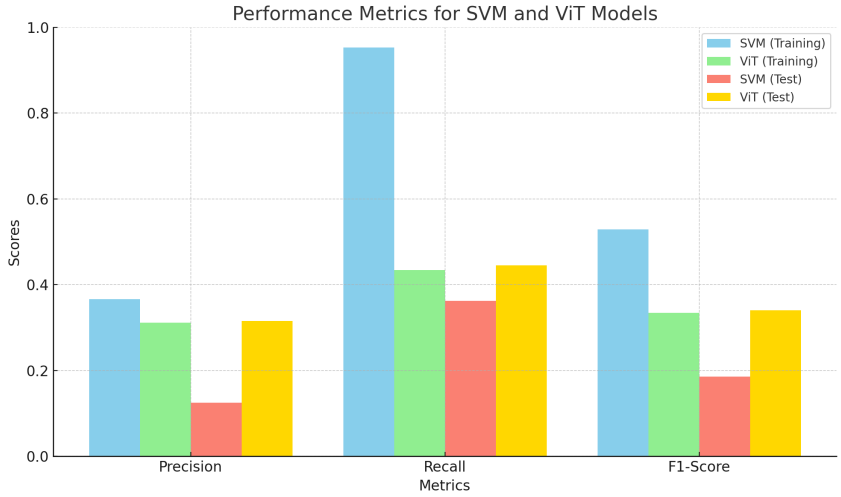

Figure 6: Performance Evaluation of SVM (Baseline) and ViT (Model) on Training and Testing Data.

8 Qualitative Results

Figure 7 below provides sample outputs from our model to illustrate its capabilities and provide context for its quantitative performance. These examples demonstrate the model's strengths in accurately predicting ingredients and highlight areas where it encounters challenges, offering a deeper understanding of its behavior across various input scenarios.

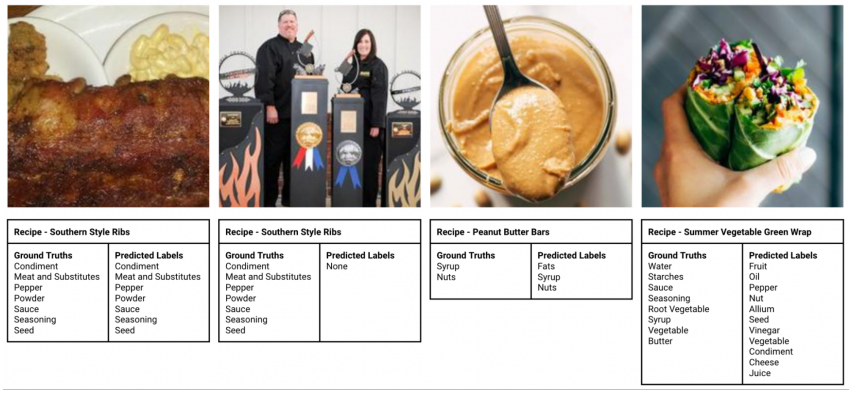

Figure 7: Predicted Labels from the Test Set Compared to Ground Truths. Images from left to right show: A) Ribs, B) People at a Rib Competition, C) Peanut Butter, D) Green Wrap.

Resilient to Poor-Quality Images: The model demonstrated resilience to poor-quality images, learning meaningful features rather than overfitting. For example, in Figure 7A (ribs) and Figure 7B (people), both images came from the same recipe with identical ground truth, but the model accurately predicted ingredients for Figure 7A but none for Figure 7B. This highlights its ability to extract relevant features like color and texture while recognizing when an image does not represent ingredients accurately. These observations help explain why precision, recall, and F1 scores are not higher: contextually accurate predictions can still be penalized by mismatches in poorly representative dataset images, reflecting limitations in the data rather than the model.

Exceeding Ground Truth Predictions: An example of the model exceeding the ground truth is seen in Figure 7C, which shows peanut butter. While the ground truth labels are syrup and nuts, the model also correctly predicted fats. This additional prediction likely stems from the model's exposure to similar food items during training, allowing it to infer related ingredients. This result helps explain why precision, recall, and F1 scores are not higher: the model's ability to infer contextually accurate ingredients sometimes leads to predictions that don't align perfectly with the ground truth, highlighting limitations in the dataset labels rather than a failure of the model.

Similar Characteristics cause Confusion: Some limitations in our model can be seen in Figure 7D, where it correctly identifies only one ingredient despite many ingredients being present. This misclassification is primarily due to the ingredients having visually similar characteristics, such as color and texture (e.g. bright fruits and veggies), which causes the model to confuse one for another. To address this issue, further training on examples with subtle differences between similar ingredients would be beneficial. By doing so, the model would be better equipped to extract and differentiate higher-level features, improving its ability to make accurate predictions in cases where ingredient variations are less pronounced.

Inputs that our model does not do well on include non-representative images or those with visually similar ingredients to other classes, where subtle differences in color and texture cause model confusion and misclassifications. However, the qualitative results show our model's strengths, its ability to learn meaningful features such as color and texture enabling it to make contextually accurate predictions and even generalise more contextually accurate ingredients.

9 Evaluation of the Model on New Data

Web Scraping Test Labels: We developed a web scraper to extract 484 unique recipes, ingredients, and images from the Pinch of Yum websites (Pin), which were not seen in the Recipes 1M+ dataset

used in model training. Using the requests and BeautifulSoup libraries, we scraped the "All Recipes" page, which displays 12 recipes per page, and extracted all URLs. After filtering out non-recipe links, we retrieved 1,116 recipes, with 530 recipes featuring a bolded ingredient format, ideal for extraction, shown in Figure 8 (Left) below. We efficiently isolated ingredient data using the spaCy NLP library by targeting HTML patterns such as checkbox inputs and "strong" tags. While most of the processed data was reasonable, some entries required manual refinement due to inconsistencies in noun usage (e.g., uncommon terms like "ears" as measurements) and unnecessary adjectives in ingredient descriptions. After removing duplicate entries, we curated a dataset of 484 unique recipes and wrote the labels to a JSON file containing unique generated recipe IDs, website links, and complete ingredient lists shown in Figure 8, (middle). This format mirrors that of our training data, allowing us to directly apply our preprocessing steps such as ingredient grouping and class rebalancing. The final labels are in the standardized format shown in Figure 8 (right), where each recipe is stored as a text file named by its ID, with each line representing a single ingredient.

Web Scraping Test Images: The images were extracted by identifying the first image from the HTML source of each URL, which typically depicted the completed dish, capturing all the ingredients, whereas later images often showed the preparation process and didn't show all ingredients in the recipes. The images were then processed using the same techniques applied during training and validation: center cropping, ensuring RGB channels only, and normalizing pixel values to a 0-255 range. All recipe images were also named using the same generated IDs, enabling precise alignment with their corresponding labels. This structured approach ensures that our dataset is clean, accessible, and ready for use in testing evaluation and comparison.

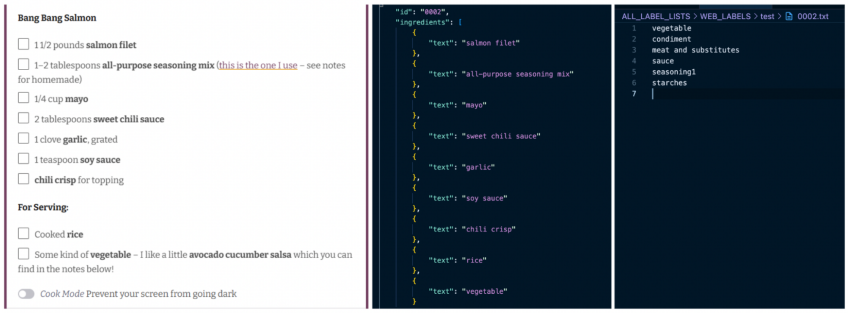

Figure 8: Left - Pinch of Yum Recipe Formatting. Middle - Recipe ingredients extracted using NLP. Right - Fully processed data with category grouping and rebalancing to our 53 classes.

Evaluation: Using our custom web-scraped and processed test set of 484 unique recipes, we evaluated the Vision Transformer (ViT) model's performance on entirely unseen data. Surprisingly, our model performed better on this new test set than on the original, achieving a Precision of 0.4098, Recall of 0.5389, and F1 Score of 0.4479, as shown in Table 2 and Figure 9.

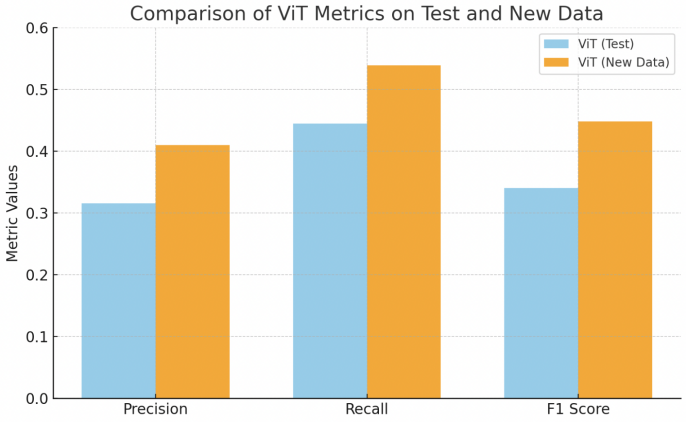

Figure 9: Performance Data of ViT (Model) on Original Testing Dataset and Web-scraped Test Data.

Table 2: Performance Data of ViT (Model) on Original Test Data and New Test Data.

Metric	ViT (Original Test Data)	ViT (New Test Data)
Precision	0.3153	0.4098
Recall	0.4447	0.5389
F1 Score	0.3401	0.4479

10 DISCUSSION

This section will discuss and further interpret our quantitative and qualitative results, highlighting our model's strength and limitations, limitations of the dataset, as well as surprising observations and key takeaways.

General and Strengths: Overall, our model has exceeded our initial expectations demonstrating robust performance in multi-label ingredient detection, handling dataset inconsistencies and achieving balanced precision and recall and outperforming the SVM baseline on the test set. One notable strength is how the ViT model demonstrated resilience in identifying relevant ingredients even with poor-quality or non-representative images, suggesting it effectively learned meaningful features such as texture, color, and shape, rather than simply memorizing training examples. This resilience was evident in cases where the ViT correctly predicted relevant ingredients even when the provided ground truths were incomplete or incorrect, seen in Figure 7B. The model's ability to generalize and predict additional contextually correct ingredients was another key strength and suggests the model did learn some contextual relationships between ingredients. The improvement in metrics on the cleaner web-scraped dataset further indicates that the ViT effectively leveraged its learned features when provided with structured and consistent data. While our quantitative results for precision, recall, and F1 score fall in the range of 0.3-0.5, this performance is reasonable and good given the complexity of the multi-label classification task, the large number of possible classes, and the inherent challenges within the dataset, such as class imbalance, visually similar categories, and inconsistencies in ground truth labels. Overall, the ViT model has demonstrated solid performance, showing its robustness in handling data inconsistencies and its ability to generalize across diverse datasets, which reinforces its potential for real-world applications in ingredient detection and multi-label classification tasks.

Model Limitations: The ViT model still has limitations that highlight areas for improvement. Firstly, we noticed it struggled to differentiate visually similar ingredients, like some "fruit" and "veggies" due to overlapping features (e.g. apple and tomato), which impacted precision in its classifications. The model also faces some challenges with rare ingredients due to some class imbalance in the training data, leading to ingredient bias and limiting generalization. While our ViT excelled at feature extraction, occasional overfitting to irrelevant image components, like backgrounds, pointed to the need for further refinement in feature extraction. These limitations suggest that improvements in dataset quality and optimization techniques are still needed for broader applicability and better performance of our model. Another limitation of our model is the significant computational resources required by the transformer architecture.

Dataset Limitations: One limitation of the dataset is that it remains imbalanced, with ingredient counts still ranging from 3,500 to 6,000 despite our rebalancing. Additionally, grouping ingredients into broader categories reduced the model's ability to learn the unique characteristics of individual ingredients, limiting its capacity to make more specific predictions. While this approach helped achieve better generalization across ingredients and cuisines, it sacrificed ingredient nuance and detail. Another challenge was the quality of the original training dataset, which contained inconsistencies in label parsing (e.g., plural forms, long names with multiple adjectives) and sometimes lacked ingredients, as shown in the ground truth labels in Figure 7C. The images were also inconsistent, with many not representing the dish accurately, such as the example in Figure 7B. Future improvements could focus on refining data quality and selecting a high quality dataset, addressing label parsing inconsistencies, and ensuring better image representation for more accurate training.

Surprising Observations: Initially, we were quite surprised that the model performed better on the new web-scraped data compared to the original test set, even though we would have expected

the opposite. The original test set was more similar to the training and validation data, making it reasonable to assume that the model would perform better on it. This enhanced performance on our web-scraped data likely stems from the higher quality of the new dataset: our web-scraped labels had standardized bolded ingredient formats which would minimise parsing inconsistencies observed in the original test set, our web-scraped images also featured less poor-representation photos, such as the example in Figure 7B mentioned in our Qualitative Results. These data quality improvements enabled our model's predictions to align better, resulting in better metric performance.

Key Takeaways: This project highlights our ViT model's strong performance in multi-label ingredient detection, showcasing its ability to handle dataset inconsistencies and generalize well across diverse data sources. While challenges remain with visually similar ingredients, class imbalances, and occasional overfitting, the model's resilience suggests its potential for practical applications in ingredient detection. Notably, the improved performance on the cleaner web-scraped dataset underscores the importance of high-quality data in enhancing the model's effectiveness. Moving forward, refining dataset quality, addressing class imbalances, and optimizing feature extraction will be key to improving the model's accuracy and robustness.

11 Ethical Considerations

A key ethical concern in this project is cultural bias. If the training dataset over-represents certain cuisines, the model may fail to accurately recognize ingredients from underrepresented cultures, leading to marginalized users and skewed recommendations, especially for individuals following specific regional diets. This could limit the model's usefulness and accessibility for a diverse audience. Privacy is another significant issue, as uploading food images might inadvertently expose personal information, such as dietary preferences, habits, or location. Protecting user privacy is essential to ensure that sensitive data is not compromised during model training or usage. The model's ingredient remapping strategies help to address these ethical concerns by consolidating specific ingredients into broader groups 1) ensuring that underrepresented ingredients are still recognized within their respective categories, while mitigating overrepresentation of more common items, and 2) reducing the risk of inadvertently revealing sensitive information about individual ingredients, which could be tied to user preferences, dietary habits, or regional identities.

12 Project Difficulty and Quality

While the project initially seemed straightforward, it turned out to be more challenging than anticipated for several reasons. First, the multi-label classification task was inherently complex. With a large number of ingredients, many of which shared overlapping features like color, texture, or shape, it was difficult for us to create meaningful groupings for the classes that were balanced and for the model to distinguish between them. This challenge was further exacerbated by class imbalance, where as much as we tried to create balanced classes, more common ingredients still dominated parts of the dataset, making it harder for the model to accurately identify the rarer ingredients. Additionally, the quality of the selected dataset introduced several complications. While the dataset was initially chosen for its large and diverse collection of pre-parsed recipes paired with images, we encountered issues such as label inconsistencies, missing ingredients, and poorly representative images. These problems added significant noise to the training process, hindering the model's ability to learn meaningful patterns and forcing it to train on data that was incorrect or misleading. Another challenge came from using Vision Transformers (ViT) for the project, which required substantial computational resources. These models, though powerful and suitable for our project application, are very resource-intensive, and training on our large dataset resulted in extremely long training times and slow iterations of optimization, which was a source of difficulty for our team. We learned that ViTs are also prone to overfitting, which required careful and time-consuming hyperparameter tuning. For these reasons, we believe our project was sufficiently challenging and despite these challenges, the model exceeded our expectations, showing resilience and robustness in handling complex, real-world data.

13 Links to Collab Notebook and Github Page

Our group utilized version control through Git and GitHub and used Colab to access GPU resources. Here are links to our Deep Learning Model and Baseline Model.

REFERENCES

Pinch of Yum. URL https://pinchofyum.com/.

Recipe1M+: A Dataset for Learning Cross-Modal Embeddings for Cooking Recipes and Food Images - MIT. URL https://im2recipe.csail.mit.edu/.

David J. Attokaren, Ian G. Fernandes, A. Sriram, Y. V. Srinivasa Murthy, and Shashidhar G. Koolagudi. Food classification from images using convolutional neural networks. In *TENCON 2017 - 2017 IEEE Region 10 Conference*, pp. 2801–2806, November 2017. doi: 10.1109/TENCON.2017.8228338. URL https://ieeexplore.ieee.org/document/8228338/?arnumber=8228338. ISSN: 2159-3450.

Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101 – Mining Discriminative Components with Random Forests. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars (eds.), *Computer Vision – ECCV 2014*, pp. 446–461, Cham, 2014. Springer International Publishing. ISBN 978-3-319-10599-4. doi: 10.1007/978-3-319-10599-4_29.

Brian Hall. Using AI to Identify Ingredients and Suggest Recipes, December 2021. URL https://medium.com/@brh373/using-ai-to-identify-ingredients-and-suggest-recipes-95482e2aca7d.

José Maurício, Inês Domingues, and Jorge Bernardino. Comparing Vision Transformers and Convolutional Neural Networks for Image Classification: A Literature Review. *Applied Sciences*, 13(9):5521, January 2023. ISSN 2076-3417. doi: 10.3390/app13095521. URL https://www.mdpi.com/2076-3417/13/9/5521. Number: 9 Publisher: Multidisciplinary Digital Publishing Institute.

Dhawal Tank. Recipe Detection of Food Image using Deep learning (CNN), July 2023. URL https://medium.com/@imdhawaltank/recipe-detection-of-food-image-using-deep-learning-65eb382aeb38.