Integrating Computer Vision and Deep Learning in Nutrition: A Novel Model for Multi-Label Ingredient Classification

Aayan Behura, Nikhil Venkateswaran, Aanya Shetty, Kiran Sapkota
behaayan@gmail.com, nikhilvenkateswaran4@gmail.com, aanyashetty1@gmail.com, kxs133@shsu.edu

ABSTRACT

Each year, poor diets contribute to more deaths in the United States than any other risk factor. Food analysis through image classification has risen up the ranks as a revolutionary technology for diet assessment and health monitoring. Yet, many existing food-analysis systems focus on dish-level or single-label classification, while relatively fewer approaches address ingredient-level multi-label recognition. In this work, we present a multi-label classification model powered by a ResNet 50 backbone. A central contribution of our study is our self-curated ingredient dataset comprising 183 ingredient classes, specifically designed with variety for real-world applications. We trained a custom head using this dataset, using focal loss and threshold optimization techniques to optimize classification performance. The model reached a F1 score of 63.82% at an optimal threshold of 0.70. Our dataset and model provide a benchmark for further research in automated visual assessments of food items. This work can legitimize a new paradigm for AI-driven ingredient recognition as a foundation for data-driven dietary assessment.

INTRODUCTION

Poor diet is the leading cause of mortality in the United States [15]. 11 million deaths and 255 million disability-adjusted life-years were attributable to dietary risk factors, including a high sodium intake, a low intake of whole grains, and a low intake of fruits [1]. The lack of a nutritional diet contributes to the development of many chronic diseases [7]. The impact is especially detrimental when it comes to children, where chronic undernourishment often leads to stunted growth and death [6].

A major challenge preventing progress on combating poor dietary choices is a lack of inaccurate food identification. Manual food labeling is known to be prone to error, especially with visually similar food [9]. Foods that look the same can have vastly different nutritional information, and such uncertainty could undermine efforts to build balanced diets. Even labels produced by companies are oftentimes unreliable, with US seafood alone having a mislabeling rate of 39.1 % [2]. Beyond that, nutritional research is often

hindered by a lack of access to precise ingredient identification as a result of outdated, inconsistent, and inaccessible food consumption databases (FCDBs) [5].

In the past, Artificial Intelligence (AI) has found a wide range of applications within image analysis. More specifically, deep learning, a sub-branch of AI, has shown great promise in areas such as computer vision, natural language processing, and speech recognition. Deep learning is also beginning to aid in the analysis of food for agricultural and food processing purposes [4]. By training models on curated datasets of thousands of images, they can learn to recognize visual features, helping revolutionize human interaction with food for the better.

In this paper, we present how deep learning algorithms can perform automatic ingredient recognition and identification. The specific objectives of this work are: 1) collecting a reliable, accurate, and up-to-date dataset of common foods; 2) using transfer learning with a ResNet-50 backbone and advanced optimization techniques to train a model for accurate multi-label ingredient classification; and 3) comprehensively evaluating model performance using multiple metrics.

RELATED WORK

Empirically, computer vision has been applied to food analysis due to its potential to improve public health outcomes [19]. Visual analysis techniques have been used to assess food quality by examining a variety of attributes (freshness, texture, appearance, etc.). These techniques were executed with minimal sample preparation and a high processing speed, making it a practical alternative to a manual food analysis [13]. Such systems sparked an initial interest in computer vision by highlighting the advantages of its usability and scalability. Once the spark was lit, it set the stage for further exploration of nutrition-related applications. More recently, image classification has been pointed towards dietary monitoring and medical applications. A systematic review of AI food analysis noted its potential to support chronic disease prevention through more consistent objective food intake analysis while acknowledging challenges in current models, including difficulty distinguishing food from non-food objects [3]. More importantly, most food analysis systems operate at the dish level, assigning a single label per image rather than identifying individual ingredients. This zoomed-out focus limits the computer vision's true potential for fine-grained nutritional assessment. Rather than staying at the surface level, multiple ingredients should be identified individually for complete understanding.

The evolution of food recognition systems has taken place alongside the creation of high-quality benchmark datasets. UECFOOD is considered to be one of the largest datasets used for food recognition. It provides food classification based on images of food that have been captured in well-controlled environments [10]. These datasets allowed researchers to achieve significant advancements in food classification problems and established benchmark evaluation methods. However, the majority of this dataset had only one bounding box per image, limiting its applicability to real-world classification tasks. The development of advanced food recognition systems would require more sophisticated algorithms and,

therefore, multi-labeled annotations of ingredients. These limitations call for custom datasets specifically tailored for accurate ingredient recognition. Unlike existing benchmarks, this work is dedicated to building a dataset designed to reflect the presence of several ingredients in one dish. In addition, when training on these datasets, class imbalance causes models to perform poorly on rare but equally important classes. To remedy this, the authors in [12] introduced focal loss. This is a variant of loss that down-weights easy background examples and focuses on hard foreground examples.

Advancements in deep convolutional neural networks (CNNs) have significantly improved image recognition performance across many domains, especially food. For instance, the authors in [17] achieved 99% accuracy when classifying single food portions from high-resolution food images. Additionally, Zhang et al. [18] developed a multi-task deep convolutional neural network that simultaneously performed dish identification, cooking-method recognition, and multi-label ingredient detection. Using a dataset of 360 food classes, with at least 500 images per class, their system shared lower-level network layers across these tasks. Broadly, the introduction of large-scale CNNs demonstrated how deep learning architectures trained upon thorough datasets can be effective. By learning hierarchical visual representations of data, they have proven to perform well on large-scale classification tasks. Increasing network depth introduces optimization challenges such as vanishing gradients and the degradation problem. Residual Networks (ResNet) address this issue by introducing skip connections that enable more effective gradient propagation through deep architecture [8]. Consequently, substantially deeper networks can be trained reliably with a strong performance on visual recognition tasks. Additionally, transfer learning has become highly efficient for visual recognition, especially when labeled data in the target domain is limited [11]. Thus, in this work, we use a transfer learning approach to fine-tune a model for ingredient classification using a ResNet-50 backbone.

METHODOLOGY

Dataset Collection

We collected our own dataset of publicly available food images from the internet. We utilized a novel Python image scraper that collects images from Google Images using Selenium through the Firefox webdriver. Algorithm 1 shows the pipeline of this scraper.

Algorithm 1 Image Scraper Pipeline

```
class ImageScraper
procedure init
    Initialize headless browser
    Define search URL and blacklist
end procedure
procedure await_load
    Loop until image loads or timeout
```

```
end procedure
procedure gather(query, count)
    Open Google Images page
    while fewer than count URLs collected do
        Click thumbnail and load image
        Filter invalid URLs
        Append valid URL to list
        Scroll to load more images
    end while
end procedure
procedure download_urls(dest)
    for each stored URL do
        Download and save PNG
    end for
    Write URL list to CSV
end procedure
```

We conducted 213 individual search queries using the previously described scraper, requesting 100 images each time. Ultimately, this gave us a total of 21,300 images to utilize. However, a substantial number of these images were duplicates, not of good quality, or simply did not meet our search queries. To ensure accuracy during training, we manually deleted 9,566 images using objective criteria. Some classes were also deleted entirely, leaving us with 183 classes. Figure 1 details the distribution of images among classes post-deletion. With the remaining 11,734 images we had left, we manually assigned them to either the training set (10,039 images) and the validation set (1,695 images), creating an approximate 85%-15% split between the images. The split was performed by hand for each of the 183 classes to ensure closeness to the 85%-15% ratio while ensuring that near-duplicate or highly similar images were not split across the two sets. To ensure images from the same source were not shared, we cross-checked each image to the web source identified by the scraper, thus preventing any overlap between images from the same source.

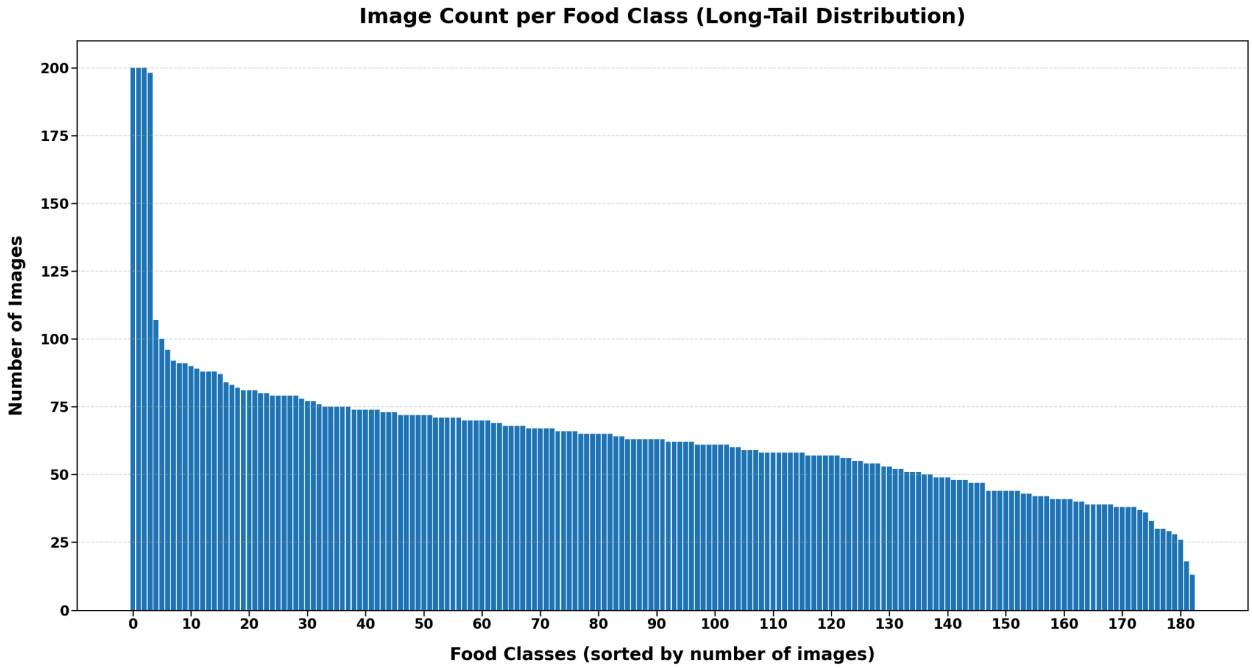

Figure 1: Long-Tail Distribution of data. The x-axis represents the classes of data (sorted by size from largest to smallest), while the y-axis represents the number of images in each class

Dataset Annotation and Label Encoding

Each image in the dataset was manually assigned one ground-truth ingredient label based on its ingredient class. As part of the dataset-cleaning process, each image was manually reviewed, and images that did not correspond to their assigned ingredient label were removed. We used a MultiLabelBinarizer to convert these ingredient labels into a 183-dimensional binary vector, with each dimension corresponding to one of the 183 ingredient classes represented in the dataset. The model produced an independent confidence score for each of the 183 ingredient classes, allowing the output to be compared directly with the corresponding binary target vector. During evaluation, these confidence scores were converted into binary predictions using a classification threshold. This allowed the model to produce predictions for multiple ingredient classes when their confidence scores exceeded the selected threshold, despite the ground-truth dataset containing only one annotated ingredient per image.

Pre-Processing of Image Data

We built a systematic pipeline for pre-processing images using PyTorch to allow for optimal training conditions. All images were resized to 224×224 pixels to match the input size expected by ResNet-50 and were converted from PIL format to PyTorch tensors. Additionally, all images were normalized using the mean ([0.485, 0.456, 0.406]) and standard deviation ([0.229, 0.224, 0.225]) of the ImageNet dataset, which ResNet 50 was pre-trained on. For the training set, we applied data augmentations including random horizontal flipping (p=0.5), random rotation (±20°), random resized cropping (scale 0.8-1.0),

color jittering (brightness, contrast, saturation, and hue variations), and random affine transformations (translation up to 10%). For the validation set, only resizing and normalization were applied.

Model Architecture

Transfer learning was utilized by fine-tuning a ResNet-50 backbone and developing a custom head with three layers. The ResNet-50 backbone extracts 2048 features from a 224x224x3 RGB input image. These features are then passed onto the custom head. The first layer reduces dimensionality from 2048 layers to 1024 features, the second layer further reduces it to 512 features, and the final layer maps these features to confidence scores for each ingredient class. Between these layers, we applied dropout layers with progressively decreasing rates of 0.5, 0.4, and 0.3. Batch normalization and ReLU activation also occurred after each layer. Figure 2 shows this pipeline.

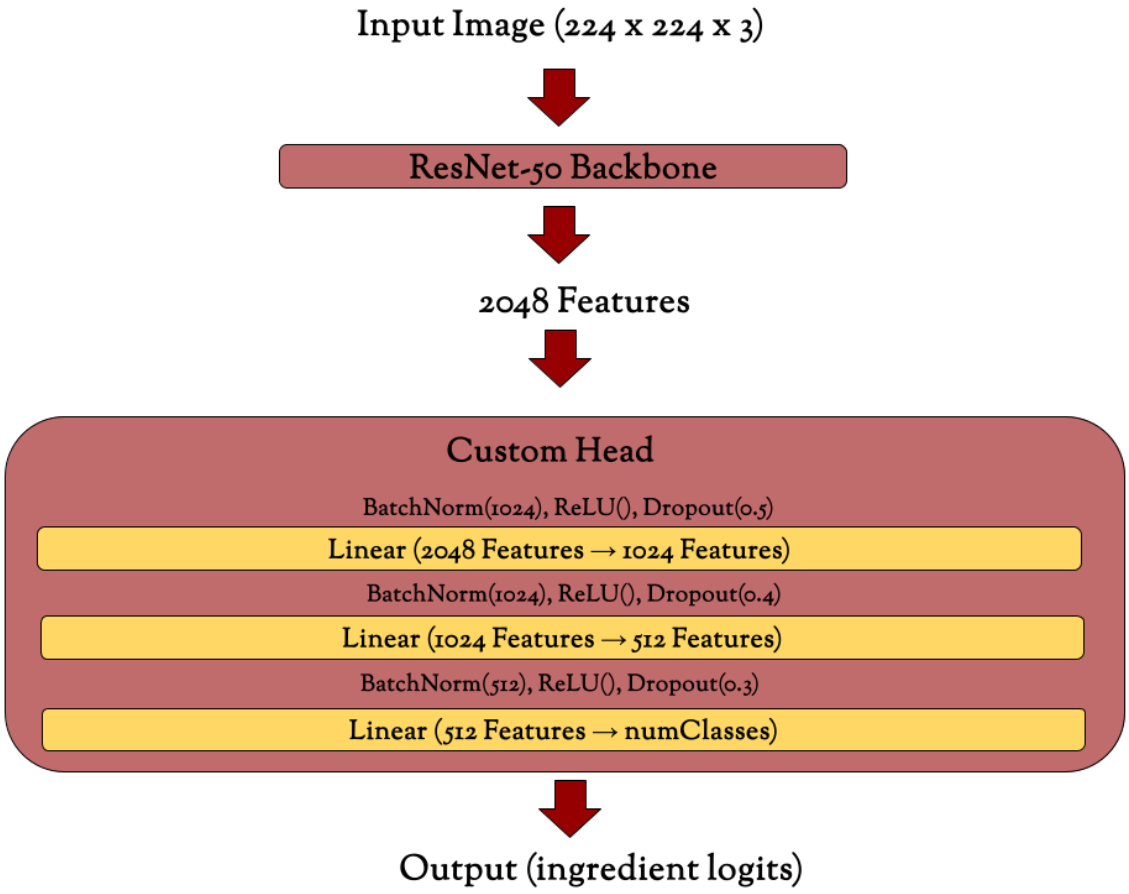

Figure 2: We used a ResNet-50 backbone with a custom head to train a model capable of classifying or labeling multiple ingredients from an image.

Loss Function

To make it work for multi-label classification, we modified the standard Binary Cross Entropy loss formula in order to account for dataset imbalance and the difficulty in classifying some ingredients. First, we applied positive class weights to classes representing rare ingredients. This was done by first finding the positive count, or the number of training images where an ingredient *j* is present in an image *i*, by using the formula:

$$P_j = \sum_{i=1}^{T} y_{i,j},$$

where P_j = the positive count for an ingredient *j*, T = the total number of training images, and $y_{i,j}$ = the label for image *i* and ingredient *j* (either 1 or 0 depending on if the ingredient is present or absent, respectively)

P_j was then subtracted from the total number of training images T to find the negative count N_j, which represented the total training images where an ingredient *j* was not present in an image *i* through the formula:

$$N_j = T - P_j,$$

where N_j = the negative count for an ingredient *j*.

After calculating the positive and negative count, we computed a raw weight for each ingredient *j* using the formula:

$$w_{raw,j} = \sqrt{\frac{N_j}{P_j+1}},$$

where $w_{raw,j}$ = the raw class weight for ingredient *j*.

This value represented the imbalance between classes, with rarer ingredients receiving larger weights while more common ingredients received smaller weights. The square root was used to soften the imbalance correction, as the full ratio would produce exceedingly large weights for rarer classes. A term of 1 was added to the denominator to avoid division by zero.

Following the calculation of raw weights, we clipped them to a range of 1 to 50 to ensure that common classes would not receive a weight below 1, very rare classes could not receive disproportionately high weights, and that the maximum possible weight was capped at 50. To do so, we used the formula:

$$w_{clipped,j} = min(max(w_{raw,j}, 1), 50),$$

where $w_{clipped,j}$ = the clipped weight for ingredient *j*.

The final clipped weights were then applied to the positive term of the Binary Cross Entropy loss equation, which penalizes failure to detect rarer ingredients more heavily than failure to detect more common ingredients. This would have no impact if an ingredient was absent. For image *i* and ingredient *j*, the formula became:

$$BCE_{i,j} = \; - [w_{clipped,j} y_{i,j} log(p_{i,j}) + (1 - y_{i,j}) log(1 - p_{i,j})],$$

where $p_{i,j}$ = the model's predicted probability that ingredient *j* is present in image *i*.

Additionally, Focal Loss was used to down-weight easy-to-classify examples during training. The modified formulation for the Focal Loss used in this paper is given by:

$$FL_{i,j} = \alpha(1 - p_t)^{\gamma} \cdot BCE_{i,j}, \; where \; p_t = p_{i,j} y_{i,j} + (1 - p_{i,j})(1 - y_{i,j}),$$

where p_t = the probability the model assigned to the correct label.

This formulation introduces two hyperparameters: the focusing parameter $\gamma \geq 0$ (we use $\gamma = 2$) and the balancing factor $\alpha \in [0, 1]$ (we use $\alpha = 0.25$).

The final training loss was then calculated to measure how far the model's predicted probabilities were from the ground-truth labels while accounting for class imbalances and example difficulty. We calculated this using the formula:

$$L = \frac{1}{183B} \sum_{i=1}^{B} \sum_{j=1}^{183} FL_{i,j},$$

where *B* = the number of images in a batch

This loss was minimized during training using backpropagation with the AdamW optimizer, so that errors on rarer ingredients, which carry larger class weights, produced larger gradients and had a greater influence on the model's updates.

Model Training

Model training was conducted using the T4 GPU hardware accelerator through Google Colab. The model was trained over 30 epochs with a batch size of 64. During each epoch, the model performs a training phase and a validation phase. During the training phase, an AdamW optimizer updates the model's weights using the calculated loss. We used the OneCycleLR Policy with cosine annealing to determine the learning rate throughout training, where the maximum learning rate was ten times the base learning rate. If validation accuracy is higher than seen before, the current model's weights are saved. Algorithm 2 showcases the training loop:

Algorithm 2 Training Loop

Require: Training dataset D_{train}, validation dataset D_{val}, number of epochs $E = 30$ **Ensure:** Trained model with best validation accuracy

```
Initialize model parameters
Initialize optimizer, learning rate scheduler, and gradient scaler
best_acc ← 0
for epoch = 1 to E do
    for all batch in D_train do
        Perform forward pass with mixed precision
        Compute focal loss
        Perform backward pass with gradient scaling
        Clip gradients
        Update model weights
        Step learning rate scheduler
    end for
    Evaluate model on D_val
    if validation accuracy > best_acc then
        best_acc ← validation accuracy
        Save model checkpoint
    end if
end for
```

After the training loop was completed, we aimed to optimize the threshold by which to judge the probabilities outputted by the sigmoid function. To do this, we tested the model's performance at different thresholds and used the threshold that yielded the highest F1 score for model evaluation.

RESULTS

Training Results

Our model was trained on the training set (see Section 3.1 Dataset Collection) for 30 epochs, after which it was determined that the model demonstrated rapid early learning before plateauing. Training accuracy saw the largest initial growth, going from 81.40% to 99.00% within the first 12 epochs. However, subsequent epochs faced minimal growth, with training accuracy reaching a maximum of 99.61%. Validation accuracy increased to 98.44% in epoch 2 before decreasing to 97.79% at epoch 6. Validation accuracy finished at 99.14% by epoch 30, demonstrating strong generalization capabilities with a minor 0.47% difference in training and validation accuracy. We also saw minimal overfitting while the model was training, suggesting that the model can perform well on unseen datasets.

Training loss and validation loss saw a similar pattern. Training loss decreased rapidly from 0.0335 in epoch 1 to 0.0059 in epoch 10, but converged to 0.0015 by epoch 30. Likewise, validation loss decreased from 0.0258 to 0.0064 from epoch 1 to epoch 10, respectively, but converged to 0.0039 by epoch 30. This demonstrates effective model convergence with the validation loss remaining within 2.6x of training loss. Furthermore, the use of the OneCycleLR Policy and Focal Loss function (see Methodology) helped optimize model training, with both losses decreasing without divergence. Further training beyond 30 epochs could yield positive results, increasing the accuracy and decreasing the loss further.

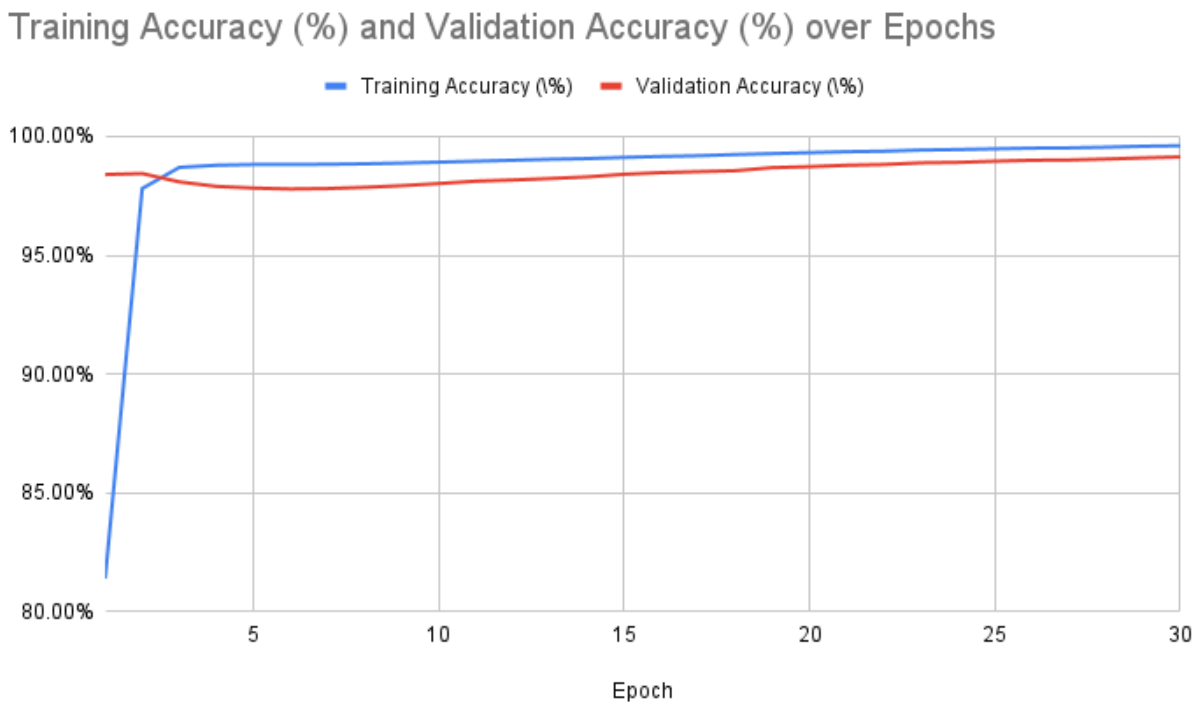

Figure 3: Training and Validation Accuracy over Epochs. The x-axis represents the epoch, while the corresponding y-values of the blue and red lines represent training and validation accuracy, respectively

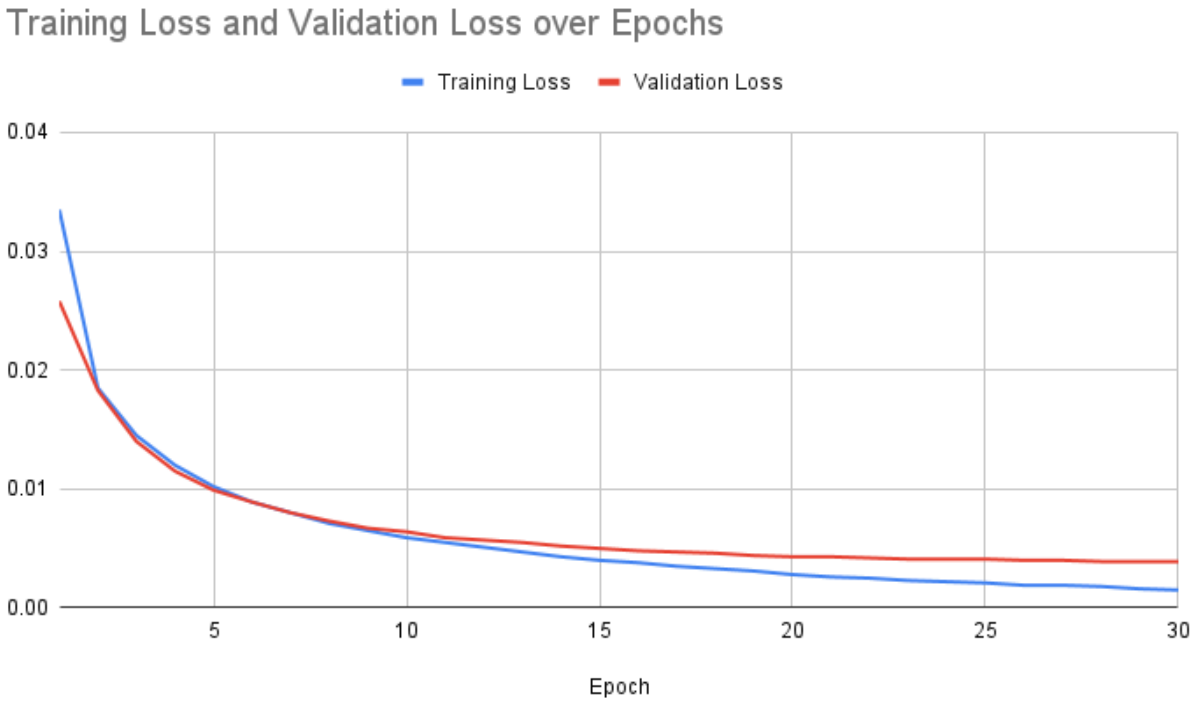

Figure 4: Training and Validation Loss over Epochs. The x-axis represents the epoch, while the corresponding y-values of the blue and red lines represent training and validation loss, respectively.

Error Analysis

Performance analysis across the 183 ingredients revealed substantial variation among classes, largely driven by visual distinction and dataset representation, resulting in many high- and low-performing classes.

Visual similarity represented a classification challenge, particularly for ingredients lacking distinct visual features. White powders, dairy products, and oils specifically formed dense confusion clusters due to similar resemblances and textures. Oils were a good example of this, with canola_oil (F1 = 0.32), olive_oil (F1 = 0.38), vegetable_oil (F1 = 0.46), and sesame_oil (F1 = 0.55) proving to be nearly indistinguishable. Analysis of false positives revealed that olive_oil was incorrectly predicted in 14 images, while vegetable_oil generated 13 false positives, often being confused with alternative oils. Conversely, ingredient classes with distinctive visual characteristics, such as color pat terns, unique textures, and characteristic shapes, achieved near-perfect performances, with banana, kiwi(fruit),

pineapple, jackfruit, and artichoke achieving F1 scores of 1.0 with zero false positives, indicating that the backbone is successful when visual features are sufficiently distinctive, even with minimal training.

Dataset classes were imbalanced in order to mimic real-world ingredient distributions, with class sizes ranging from 7 samples to 100 samples in the dataset, creating a disparity in representation. Ingredient sets with high support achieved substantially better performances, with clams (F1 = 0.97), feta (F1 = 0.92), ricotta (F1 = 0.94), and salmon (F1 = 0.97) all having close to 100 samples. Meanwhile, ingredients with only 7 samples exhibited highly variable performance, with F1 scores ranging from 0.00, as seen through cream_(dairy), to 0.46. This behavior confirmed bias towards common classes, as the weighted-average F1-score (70.13%) exceeded the macro-average F1-score (62.08%), leading us to conclude that high-frequency ingredients disproportionately affected performance.

F1 Score

While the model achieved 99.14% in validation accuracy, our F1-score was 63.82% at the optimal threshold of 0.70. This took into account the precision and recall of true positives, while overall accuracy also considers true negatives. With 183 ingredient classes and class imbalances within the dataset, predicting exact ingredient combinations becomes substantially more challenging despite focal loss algorithms evening out the imbalance.

Optimization

Systematic threshold optimization revealed trade-offs between precision and recall that substantially impacted model performance. At the default classification rate threshold of 0.50, the model exhibited strong recall (87.43%) but poor precision (46.97%), resulting in an F1 score of 0.58 and a 29.6% over-prediction rate. This result came from excessive false positives (see Error Analysis). Ultimately, 0.70 was the optimal threshold, improving the F1 score to 0.64 (increasing by approximately 0.06), increasing precision to 58.29% (+11.32 percentage points), while recall decreased to 75.46% (-11.97 percentage points). This optimization effectively reduced over-prediction while maintaining a high level of ingredient detection.

However, threshold adjustment introduced a new failure, with 190 images (11.2% of the validation set) receiving zero predictions above the 0.70 threshold, meaning the model lacked confidence in all ingredients. The precision-recall curve (see Figure 5) demonstrated a sharp peak at 0.70 before declining rapidly at higher thresholds, resulting in limited room for threshold-based optimization. Notably, the optimal threshold substantially exceeded the 0.50 default, suggesting that multi-label food classification inherently requires higher confidence thresholds to accurately recognize ingredients without overprediction.

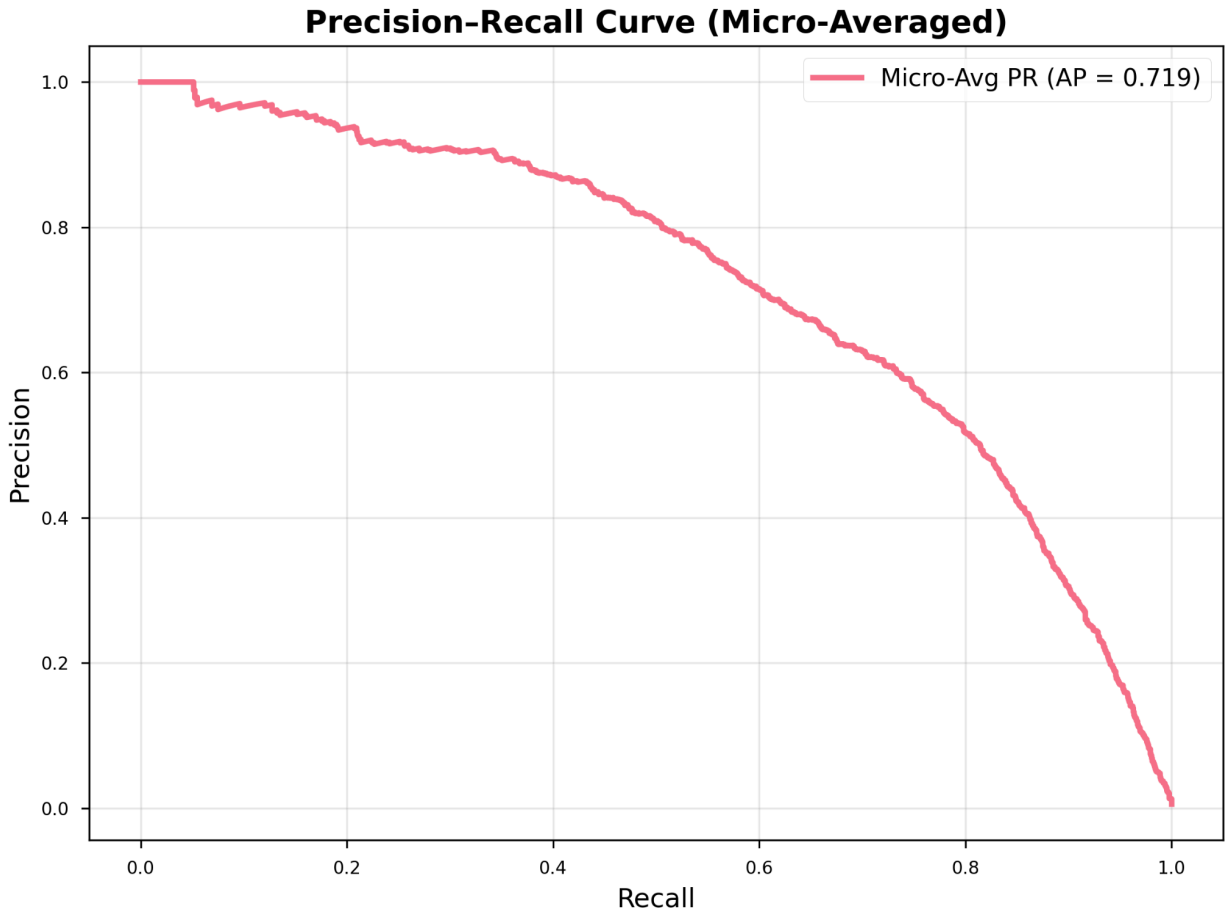

Figure 5: Precision-Recall Curve. The x-axis represents recall, while the y-axis represents precision.

DISCUSSION

This study addressed the critical challenge of inaccurate food identification through a machine learning model that recognizes ingredients in images. We developed a multi label classification system with a ResNet-50 backbone to identify individual ingredients in images, moving beyond single-label dish classification, which has dominated previous works. The model was trained on 183 ingredient classes with 11,734 images collected using a Python image scraper, divided into an 85%/15% split among training and validation datasets, respectively. Following this, a PyTorch pipeline was developed to pre-process images for optimal training conditions, and transfer learning was utilized through a custom head with three layers built off the ResNet-50 backbone. The Focal Loss function was included to weight classes appropriately to address class imbalance. After 30 training epochs with OneCycleLR scheduling, our model achieved a training accuracy of 99.61%, validation accuracy of 99.14%, training loss of 0.0015, and validation loss of 0.0039. Our model functioned at an F1 score of 63.82% at the optimal

threshold of 0.70. Notably, 42.24% of validation images received exact predictions, meaning 716 out of 1695 cases featured complete ingredient combinations. These results suggest that deep learning can perform ingredient recognition with high accuracy and strong generalization when visual features are distinctive, while also showing that visual ambiguity and class imbalance require targeted solutions for maximum efficiency.

This work advances beyond current food recognition systems in several ways. As opposed to previous approaches that achieve single-label classification, our multi-label framework identifies individual ingredients within complex meals, enabling precise nutritional tracking that allows for real-world applications beyond simple recognition. With our custom head, we specified dimensionality up to 512 features, allowing the model to be specified for food recognition. Using a threshold of 0.5 resulted in poor precision, high levels of over-prediction, and excessive false positives. Thus, we concluded that 0.70 was the optimal threshold and solved these issues at the expense of recall, implying that multi-label food classification requires higher confidence thresholds to accurately recognize ingredients. Finally, most applications of the Focal Loss function tend to primarily focus on dense object detection. We expanded this application to multi-label classification, allowing us to utilize the function for semantic imbalance rather than for spatial imbalance.

Overall, the model's performance demonstrated several key strengths that advance food image analysis. Notably, the model achieved a 99.14% validation accuracy, suggesting that the ResNet-50 backbone can reliably distinguish presence and absence in multi-ingredient contexts when properly adapted. Furthermore, the model was especially successful when distinct visual features were present, with five ingredient classes (banana, artichoke, jackfruit, kiwi, and pineapple) achieving perfect F1 scores of 1.0 with zero false positives. Classes with high support (99-100 samples) averaged a 94.31% F1 score, confirming the model's high capacity when ingredients have strong representation within the dataset. Ultimately, this work contributes three significant contributions to the field. First, it shows that under the proper architecture and optimization levels, it is possible to perform multi-ingredient level classification, moving past single-label classification. Second, it provides quantitative evidence that classification is largely affected by dataset sizes, although visual distinctiveness is another key factor. Finally, optimizing thresholds was associated with a 5.76% increase in F1 score, implying that optimization can significantly boost performance in multi-label food classification.

Several critical design decisions proved essential to the model's success. One key adoption was the Focal Loss function ($\alpha = 0.25$, $\gamma = 2.0$) with dynamic class weight ing (pos_weight clamped between 1.0-50.0×) to address class imbalance. After 9,926 images were eliminated from the original dataset, some classes were severely underrepresented, which would have been difficult to solve under standard binary cross-entropy loss. Despite the weighted-average F1-score (70.13%) substantially exceeding the macro average (62.08%) by 8.05 percentage points, indicating that high-frequency ingredients disproportionately affect performance, the model saw significant improvement after implementation of the Focal Loss function. Additionally, the use of a transfer learning approach provided significant and immediate benefits, achieving 98.40% validation accuracy by epoch 1 with minimal fine-tuning, though task-specific

adaptation over the subsequent 29 epochs improved this to 99.14%. The ResNet-50 backbone's architecture with 50 layers allowed the model to learn visual features from low-level textures to high-level ingredient specificities. Our custom classification head provided the capacity required to model complex ingredient occurrence patterns while maintaining regularization, as seen through a 0.47% gap in accuracy between training and validation after 30 epochs. These choices proved to be vital to the model's performance.

Despite the model's overarching success, there were several weaknesses that limited the model's applicability to real-world scenarios. First, visually ambiguous ingredients, such as white powders (flour, sugar), liquid fats (oils), and dairy products achieved poor performances, with F1 scores ranging from 0.00% to 46.15%. This indicates fundamental limitations in distinguishing ingredients with similar visual appearances. Second, class imbalance that was a means to recreate real-world data variation continued to have an impact on training despite Focal Loss and class weighting, with 11.2% of images receiving 0 predictions. This resulted in classes with higher frequencies having higher F1 scores, while lower-frequency classes tended to have variable accuracy. Future work should aim to make several improvements. First, addressing visual ambiguity through architectural improvements could be beneficial in improving performance in challenging categories like oils. Ideas include context-aware prediction mechanisms, which use context to create accurate predictions [16]. Targeted data collection specific to underrepresented ingredient classes could narrow the gap between weighted and macro-average F1 scores, allowing for more balance within the dataset. Furthermore, implementing per-class threshold optimization could better account for precision-recall tradeoffs for individual ingredients. Despite these barriers, our model performs better than similar architectures. Table 1 compares our model with methods proposed by [19] using a baseline ResNet-50, AFM, and AFCM architectures, providing context for our model's performance.

Table 1

Model	Recall	Macro F1	Weighed F1	F1
Morales et al. ResNet-50	46.94%	N/R	N/R	57.24%
Morales et al. AFM	51.84%	N/R	N/R	60.04%
Morales et al. AFCM	50.06%	N/R	N/R	60.04%
Our model: ResNet-50 + Focal Loss	75.46%	62.08%	70.13%	63.82%

By using a pipeline that focuses strictly on feature reduction and direct probability mapping as well as a dataset that contains nearly double the number of classes (183 vs. 93 classes) with more diverse images, our work yields consistently stronger results than the current baseline.

The demonstrated feasibility of automated ingredient recognition offers promising real-world applications. Most immediately, this technology can address the critical food mislabeling problem that affects consumer decision-making. Integration of ingredient recognition technologies similar to this model into point-of-sale systems, mobile applications (especially those pertaining to dietary tracking), or restaurant menu verification tools could empower customers to make informed, real-time dietary choices and reduce manual identification errors that occur with visually similar food. Regarding disease prevention, where dietary risk factors contribute to 11 million annual deaths, automated ingredient recognition allows for more consistent and accurate food analysis, improving the efficacy of interventions for diet-related conditions. In research contexts, precise ingredient identification can strengthen the ability for breakthroughs and innovation to be created by reducing measurement error in dietary assessment. Finally, this technology could be especially important to vulnerable populations, as automated ingredient recognition in household and institutional settings could create a meaningful difference, ensuring adequate nutritional diversity and preventing dietary patterns. As deep learning continues to advance, ingredient recognition technology will prove to be a necessity in various applications, and initiatives to spread this technology represent a promising path towards removing roadblocks that prevent individuals from reaching good nutrition.

CONCLUSION

This paper presents a novel deep learning model capable of accurately classifying multiple ingredients from a single food image. The model uses the ResNet-50 backbone with a custom head that specifies down to 512 features, and is trained on a tailored dataset of food images containing 183 ingredient classes. Beyond that, numerous state-of-the-art techniques were used to ensure model quality, such as focal loss and threshold optimization. The model achieved a F1-score of 63.82% at a threshold of 0.70. Future research should emphasize addressing visual ambiguity and further reducing the impact of class imbalance on training. Ultimately, our results demonstrate the promise of intelligent and automated food labeling not just as an innovation, but as a life-saving technology.

REFERENCES

[1] Ashkan Afshin et al. “Health effects of dietary risks in 195 countries, 1990–2017: a systematic analysis for the Global Burden of Disease Study 2017”. In: *The Lancet* 393.10184 (May 2019), pp. 1958–1972. issn:0140-6736.doi:10.1016/s0140-6736(19)30041-8.url:http://dx.doi.org/10.1016/S0140-6736(19)3004

1- 8.

[2] Sarah Ahles, Christina A. Mireles DeWitt, and Rosalee S. Hellberg. "A meta analysis of seafood species mislabeling in the United States". In: *Food Control* 171 (May 2025), p. 111110. issn: 0956-7135. doi: 10.1016/j.foodcont.2024.111110. url: http://dx.doi.org/10.1016/j.foodcont.2024.111110.

[3] Lameck Mbangula Amugongo et al. "Mobile computer vision-based applications for food recognition and volume and calorific estimation: A systematic review". en. In: *Healthcare (Basel)* 11.1 (Dec. 2022), p. 59.

[4] Premkumar Borugadda and Hemantha Kumar Kalluri. "A Comprehensive Analysis of Artificial Intelligence, Machine Learning, Deep Learning and Computer Vision in Food Science". In: *Journal of Future Foods* (July 2025). issn: 2772-5669. doi: 10.1016/j.jfutfo.2025.07.002. url: http://dx.doi.org/10.1016/j. jfutfo.2025.07.002.

[5] Sarah Brinkley et al. "The state of food composition databases: data attributes and FAIR data harmonization in the era of digital innovation". en. In: *Front. Nutr.* 12 (Mar. 2025), p. 1552367.

[6] Indiran Govender et al. "Malnutrition in children under the age of 5 years in a primary health care setting". In: *South African Family Practice* 63.1 (Sept. 2021). issn: 2078-6204. doi: 10.4102/safp.v63i1.5337. url: http://dx.doi.org/10. 4102/safp.v63i1.5337.

[7] Sareen S. Gropper. "The Role of Nutrition in Chronic Disease". In: *Nutrients* 15.3 (Jan. 2023), p. 664. issn: 2072-6643. doi: 10.3390/nu15030664. url: http: //dx.doi.org/10.3390/nu15030664.

[8] Kaiming He et al. "Deep residual learning for image recognition". In: (2015). eprint: 1512.03385 (cs.CV).

[9] Erica Howes et al. "Image-Based Dietary Assessment Ability of Dietetics Students and Interns". In: *Nutrients* 9.2 (Feb. 2017), p. 114. issn: 2072-6643. doi: 10.3390/ nu9020114. url: http://dx.doi.org/10.3390/nu9020114.

[10] Yoshiyuki Kawano and Keiji Yanai. "Food image recognition with deep convolutional features". In: *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*. Seattle Washington: ACM, Sept. 2014.

[11] Xuhong Li et al. "Transfer learning in computer vision tasks: Remember where you come from". en. In: *Image Vis. Comput.* 93.103853 (Jan. 2020), p. 103853.

[12] Tsung-Yi Lin et al. "Focal Loss for dense object detection". In: (2017). eprint: 1708.02002 (cs.CV).

[13] Ji Ma et al. "Applications of computer vision for assessing quality of agri-food products: A review of recent research advances". en. In: *Crit. Rev. Food Sci. Nutr.* 56.1 (2016), pp. 113–127.

[14] Roberto Morales, Angela Martinez-Arroyo, and Eduardo Aguilar. "Robust Deep Neural Network for Learning in Noisy Multi-Label Food Images". In: Sensors 24.7 (Mar. 2024), p. 2034. issn: 1424-8220. doi: 10.3390/s24072034. url: http://dx.doi.org/10.3390/s24072034.

[15] Emily D. Matthews and Emma L. Kurnat-Thoma. "U.S. food policy to address diet-related chronic disease". In: *Frontiers in Public Health* 12 (May 2024). issn: 2296-2565. doi:

10.3389/fpubh.2024.1339859. url: htp://dx.doi.org/10. 3389/fpubh.2024.1339859t.

[16] Heinrich Peters et al. “Context-aware prediction of active and passive user engagement: Evidence from a large online social platform”. In: *Journal of Big Data* 11.1 (Aug. 2024). issn: 2196-1115. doi: 10.1186/s40537- 024- 00955- 0. url: http://dx.doi.org/10.1186/s40537-024-00955-0.

[17] Parisa Pouladzadeh et al. “Food calorie measurement using deep learning neural network”. In: *2016 IEEE International Instrumentation and Measurement Technology Conference Proceedings*. Taipei, Taiwan: IEEE, May 2016.

[18]Xiaodi Zhang, Yuqing Lu, and Shiliang Zhang. “Multi-Task Learning for Food Identification and Analysis with Deep Convolutional Neural Networks”. In: *Journal of Computer Science and Technology* 31.3 (May 2016), pp. 489–500. issn: 1860-4749. doi: 10.1007/s11390-016-1642-6.

[19] Zhiyao Zhao et al. “Application of machine vision in food computing: A review”. en. In: *Food Chem.* 463.Pt 4 (Jan. 2025), p. 141238.