Preprints.org

Article

Not peer-reviewed version

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

Aayan Behura , Nikhil Venkateswaran , Aanya Shetty , Kiran Spakota *

Posted Date: 21 April 2026

doi: 10.20944/preprints202604.1480.v1

Keywords: deep learning; computer vision; food recognition; multi-label classification; ResNet-50; ingredient detection; dietary assessment

Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a Creative Commons CC BY 4.0 license, which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 21 April 2026 doi:10.20944/preprints202604.1480.v1

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

Article

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

Aayan Behura [1], Nikhil Venkateswaran [2], Aanya Shetty [1] and Kiran Sapkota [3,*]

[1] Rock Bridge High School, 4303 S Providence Rd, Columbia, MO 65203, USA
[2] Millard North High School, 1010 S 144th St, Omaha, NE 68154, USA
[3] Department of Public Health, Sam Houston State University, 1905 University Ave, Huntsville, TX 77340, USA
* Correspondence: kxs133@shsu.edu

Abstract

Each year, poor diets contribute to more deaths in the United States than any other risk factor. Image classification has emerged as a promising opportunity to enhance food analysis capabilities for diet assessment and health monitoring. However, existing models are often limited to single-label classification due to a lack of ingredient-level data, hindering their applicability to food analysis tasks. In this work, we present a novel multi-label classification model powered by a ResNet-50 backbone. We trained a custom head on our self-curated dataset comprising 183 ingredient classes, using focal loss and threshold optimization to enhance classification performance. The model achieved 99.14% validation accuracy and reached a macro 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.

Keywords: deep learning; computer vision; food recognition; multi-label classification; ResNet-50; ingredient detection; dietary assessment

1. Introduction

Poor diet is the leading cause of mortality in the United States [1]. In 2017, 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 [2]. The lack of a nutritional diet contributes to the development of many chronic diseases [3]. The impact is especially detrimental when it comes to children, where chronic undernourishment often leads to stunted growth and death [4].

A major challenge preventing progress on combating poor dietary choices is a lack of accurate food identification. Manual food labeling is known to be prone to error, especially with visually similar food [5]. 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 % [6]. 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) [7].

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 [8]. Trained on curated datasets of thousands of images, models can learn to recognize visual features, helping revolutionize human interaction with food for the better.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 21 April 2026 doi:10.20944/preprints202604.1480.v1

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.

2. Related Work

Empirically, computer vision has been applied to food analysis due to its potential to improve public health outcomes [9]. 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 [10]. 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. Recent developments in image classification have simplified its implementation in fields related to nutrition and medicine. As observed by a systematic study of AI food analysis [11] , AI food analysis has the potential to play a role in preventing chronic diseases. This is primarily through the ability to provide consistent and objective food intake analysis. Nevertheless, current frameworks face challenges such as the inability to differentiate between food items and non-food items. Currently, existing frameworks mainly function on a dish-to-dish basis, where a single label is assigned per image rather than identifying individual ingredients. Effective nutritional assessment requires a transition from dish-level classification to multi-label identification.

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 [12]. 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 [13] 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 led to great improvements in image recognition, particularly in foods. For instance, the authors in [14] achieved 99% accuracy when classifying single food portions from high-resolution food images. 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 [15]. 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 [16]. Thus, in this work, we use a transfer learning approach to fine-tune a model for ingredient classification using a ResNet-50 backbone.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 21 April 2026 doi:10.20944/preprints202604.1480.v1

3. Materials and Methods

3.1. 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 use. However, a substantial number of these images were not of good quality or simply did not meet our search queries. To ensure accuracy during training, we manually deleted 9,926 images that were of low quality or did not match the class they belonged to. Some classes were also deleted entirely, leaving us with 183 classes. Figure 1 details the distribution of images among classes after deletion. With the 11,374 images we had left, we used approximately 85% of the images for our training set and 15% for the validation set.

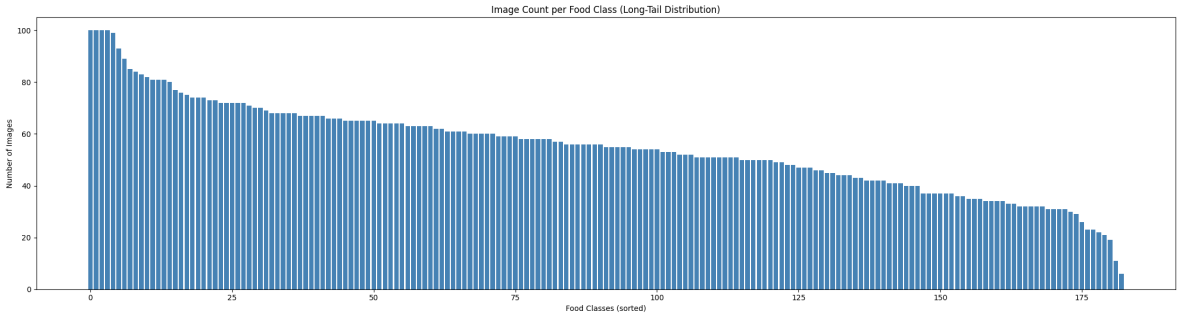

Figure 1. Long-Tail Distribution of data. The x-axis represents the classes of data (sorted by size from largest to smallest), while the y-axis represents the number of images in each class.

3.2. 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

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 21 April 2026 doi:10.20944/preprints202604.1480.v1

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 ($\pm 20°$), 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.

3.3. 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 features 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.

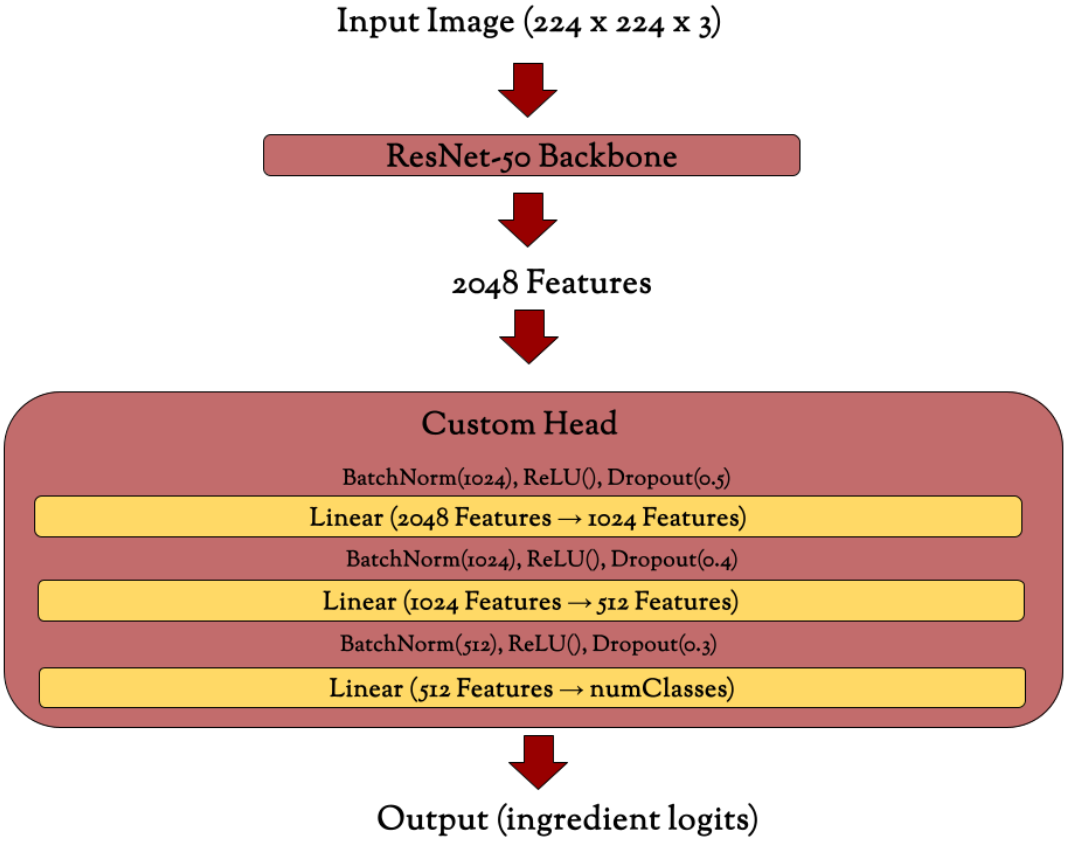

Figure 2. We used a ResNet-50 backbone with a custom head to train a model capable of labeling multiple ingredients from an image.

3.4. Loss Function

To adapt it 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 rarer ingredients. Additionally, Focal Loss was used to down-weight easy-to-classify examples during training. The formulation for the Focal Loss is given by:

$$\mathrm{FL}(p_t) = -\alpha_t (1 - p_t)^{\gamma} \log(p_t)$$

where p_t is defined as

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 21 April 2026 doi:10.20944/preprints202604.1480.v1

$$p_t = \begin{cases} p, & \text{if } y = 1, \\ 1 - p, & \text{if } y = 0. \end{cases}$$

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 $\alpha = 0.25$).

3.5. 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 $\mathcal{D}_{train}$, validation dataset $\mathcal{D}_{val}$, 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 $\leftarrow 0$
  for epoch = 1 to E **do**
    for all batch in $\mathcal{D}_{train}$ **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 $\mathcal{D}_{val}$
    if validation accuracy $>$ best_acc **then**
      best_acc $\leftarrow$ 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.

4. Results

4.1. 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.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 21 April 2026 doi:10.20944/preprints202604.1480.v1

Table 1. Summary of training and validation accuracy and loss across 30 epochs.

Epoch	Training Accuracy (%)	Validation Accuracy (%)	Training Loss	Validation Loss
1	81.40%	98.40%	0.0335	0.0258
2	97.81%	98.44%	0.0185	0.0183
3	98.70%	98.09%	0.0145	0.0140
4	98.79%	97.90%	0.0120	0.0115
5	98.82%	97.83%	0.0102	0.0099
6	98.82%	97.79%	0.0089	0.0089
7	98.83%	97.81%	0.0080	0.0080
8	98.85%	97.86%	0.0071	0.0073
9	98.88%	97.93%	0.0065	0.0067
10	98.92%	98.02%	0.0059	0.0064
11	98.96%	98.12%	0.0055	0.0059
12	99.00%	98.17%	0.0051	0.0057
13	99.04%	98.23%	0.0047	0.0055
14	99.07%	98.30%	0.0043	0.0052
15	99.12%	98.41%	0.0040	0.0050
16	99.16%	98.48%	0.0038	0.0048
17	99.19%	98.52%	0.0035	0.0047
18	99.24%	98.56%	0.0033	0.0046
19	99.28%	98.69%	0.0031	0.0044
20	99.32%	98.73%	0.0028	0.0043
21	99.36%	98.79%	0.0026	0.0043
22	99.38%	98.82%	0.0025	0.0042
23	99.43%	98.89%	0.0023	0.0041
24	99.45%	98.91%	0.0022	0.0041
25	99.48%	98.96%	0.0021	0.0041
26	99.51%	99.00%	0.0019	0.0040
27	99.52%	99.01%	0.0019	0.0040
28	99.55%	99.05%	0.0018	0.0039
29	99.59%	99.10%	0.0016	0.0039
30	99.61%	99.14%	0.0015	0.0039

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 never exceeding 2.6 times the 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.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 21 April 2026 doi:10.20944/preprints202604.1480.v1

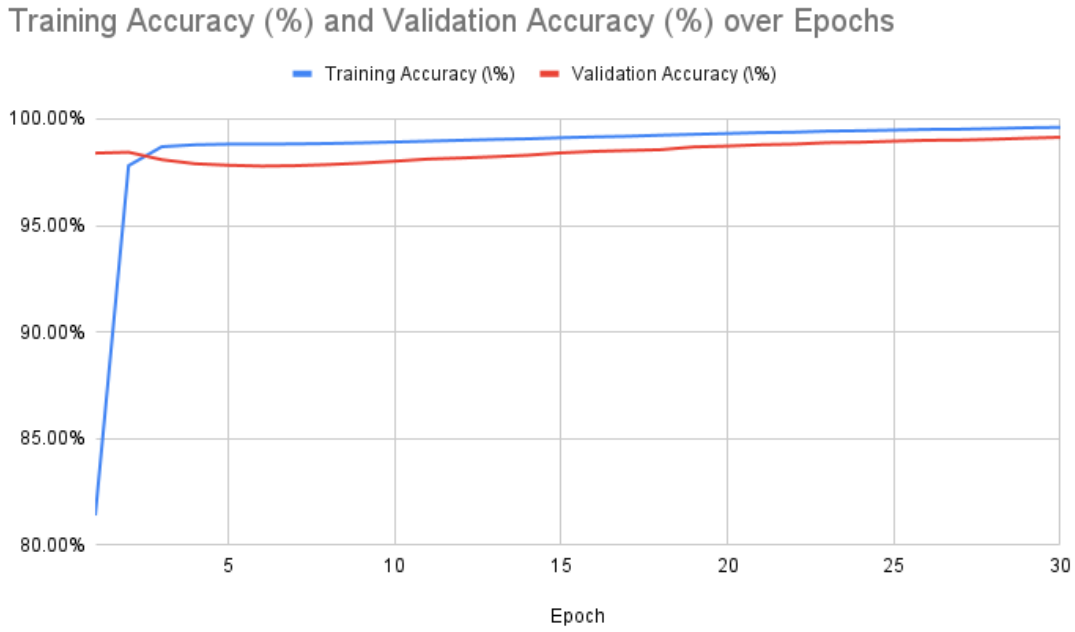

Figure 3. Training and Validation Accuracy over Epochs. The x-axis represents the epoch, while the corresponding y-values of the blue and red lines represent training and validation accuracy, respectively

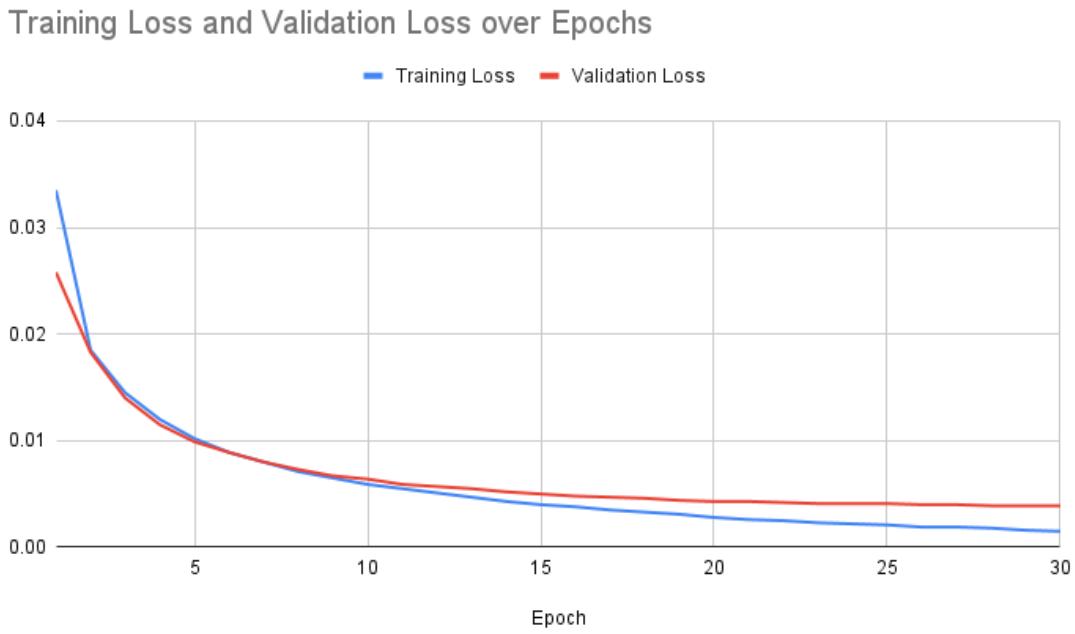

Figure 4. Training and Validation Loss over Epochs. The x-axis represents the epoch, while the corresponding y-values of the blue and red lines represent training and validation loss, respectively

4.2. 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

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 21 April 2026 doi:10.20944/preprints202604.1480.v1

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 patterns, 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. 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%) at the default threshold of 0.50. This confirmed that high-frequency ingredients disproportionately affected performance.

4.3. F1 Score

While the model achieved 99.14% in validation accuracy, our F1-score was 63.82% at the optimal threshold of 0.70. The F1 score accounts for precision and recall among 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.

4.4. 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 issue, with 190 images (11.2% of the validation set) receiving zero predictions above the 0.70 threshold, meaning the model did not predict any ingredient with sufficient confidence for those images. 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.

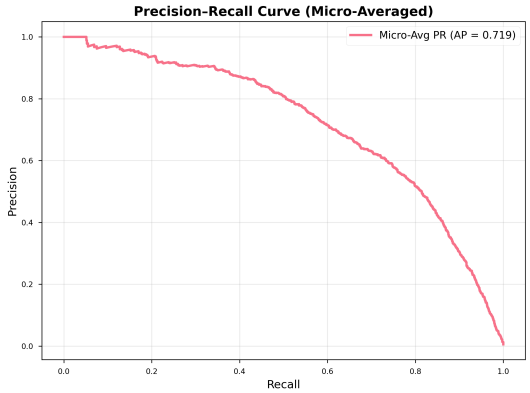

Figure 5. Precision-Recall Curve. The x-axis represents recall, while the y-axis represents precision.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 21 April 2026 doi:10.20944/preprints202604.1480.v1

5. Discussion

This study addressed the problem of inaccurate food identification through a machine learning approach that recognized ingredients in images. A multi-label classification model with a ResNet-50 backbone was created to identify individual ingredients in images, moving beyond single-label dish classification, which has been prevalent across prior studies. The model was trained on 183 ingredient classes, using 11,374 images collected with a Python image scraper. These images were then divided into an 85%/15% split among training and validation datasets, respectively. A PyTorch pipeline was developed to preprocess images for optimal training conditions. Transfer learning was also utilized through a custom head with three layers built off the ResNet-50 backbone. The Focal Loss function was implemented to address class imbalance by weighing the classes appropriately. 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. The model was able to operate at an F1 score of 63.82% at an optimal threshold of 0.70. Notably, 42.24% of validation images received exact predictions, suggesting that distinct visual features can lead to high accuracy and strong generalization in terms of ingredient recognition.

This work improves upon current food recognition systems in several ways. Unlike single-label classification models, our ingredient-level approach allows for real-world applications. Unlike 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 identified 0.70 as the optimal threshold. This improved precision and reduced false positives, though at the expense of recall. 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.

The performance of the model revealed many important strengths. First, the ResNet-50 backbone can effectively detect ingredients in a multi-ingredient context, as seen through the 99.14% validation accuracy. Furthermore, the model performed best when visual features were distinct. For instance, the model had perfect F1 scores of 1.0 with zero false positives for five ingredient classes (banana, artichoke, jackfruit, kiwi_(fruit), and pineapple): each of which contained features unique to the group. Additionally, classes with high support (99-100 samples) averaged an F1 score of 94.31%, indicating that the model performs well on ingredient classes with strong representation. Ultimately, this work makes 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 design decisions proved essential to the model's success. The deletion of thousands of images from the original dataset created an imbalance between some ingredient classes. However, after the implementation of the Focal Loss function ($\alpha = 0.25$, $\gamma = 2.0$) with dynamic class weighting (pos_weight clamped between 1.0–50.0×), the model saw significant improvement. Additionally, our transfer learning approach with the ResNet-50 backbone provided significant benefits, as seen through a mere 0.47% gap in accuracy between training and validation after 30 epochs.

Although the model saw overarching success, several drawbacks limited its applicability to real-world scenarios. First, F1 scores for visually ambiguous ingredients, such as white powders (flour, sugar), liquid fats (oils), and dairy products, ranged from 0.00% to 46.15%, suggesting that the model had difficulty differentiating between them. Second, class imbalance that was meant to recreate real-world data variation continued to have an impact on training despite Focal Loss and class weighting,

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 21 April 2026 doi:10.20944/preprints202604.1480.v1

with 11.2% of images receiving 0 predictions. As a result, classes with higher frequencies maintained higher F1 scores, while classes with lower frequencies tended to be more variable. Future work should aim to make several improvements. First, addressing visual ambiguity through architectural refinements could boost performance in these categories. Secondly, future work should target data collection for underrepresented ingredient classes. This could narrow the gap between weighted and macro-average F1 scores, allowing for more balance within the dataset. Finally, implementing per-class threshold optimization could better account for precision-recall tradeoffs for individual ingredients.

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 reduce measurement error in dietary studies, enabling more reliable research outcomes. 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 concerning 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 maintaining good nutritional habits.

6. Conclusions

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 reduces dimensionality 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 99.14% validation accuracy with an 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.

Author Contributions: Conceptualization, A.B. and N.V.; methodology, A.B.; software, A.B. and N.V.; validation, A.B., N.V. and A.S.; investigation, N.V.; data curation, A.B.; writing—original draft preparation, A.B., N.V. and A.S.; writing—review and editing, K.S.; visualization, N.V. and A.B.; supervision, K.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The original data presented in the study are openly available in FigShare at https://doi.org/10.6084/m9.figshare.31866427.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Matthews, E.D.; Kurnat-Thoma, E.L. U.S. Food Policy to Address Diet-Related Chronic Disease. *Frontiers in Public Health* **2024**, *12*. https://doi.org/10.3389/fpubh.2024.1339859.
2. Afshin, A.; Sur, P.J.; Fay, K.A.; Cornaby, L.; Ferrara, G.; Salama, J.S.; Mullany, E.C.; Abate, K.H.; Abbafati, C.; Abebe, Z.; et al. Health Effects of Dietary Risks in 195 Countries, 1990–2017: A Systematic Analysis for the Global Burden of Disease Study 2017. *The Lancet* **2019**, *393*, 1958–1972.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 21 April 2026 doi:10.20944/preprints202604.1480.v1

3. Gropper, S.S. The Role of Nutrition in Chronic Disease. *Nutrients* **2023**, *15*, 664. https://doi.org/10.3390/nu15030664.
4. Govender, I.; Rangiah, S.; Kaswa, R.; Nzaumvila, D. Malnutrition in Children Under the Age of 5 Years in a Primary Health Care Setting. *South Afr. Fam. Pract.* **2021**, *63*, e1–e6. https://doi.org/10.4102/safp.v63i1.5337.
5. Howes, E.; Boushey, C.; Kerr, D.; Tomayko, E.; Cluskey, M. Image-Based Dietary Assessment Ability of Dietetics Students and Interns. *Nutrients* **2017**, *9*, 114. https://doi.org/10.3390/nu9020114.
6. Ahles, S.; DeWitt, C.A.M.; Hellberg, R.S. A Meta-Analysis of Seafood Species Mislabeling in the United States. *Food Control* **2025**, *171*, 111110. https://doi.org/10.1016/j.foodcont.2024.111110.
7. Brinkley, S.; Gallo-Franco, J.J.; Vázquez-Manjarrez, N.; Chaura, J.; Quartey, N.K.A.; Toulabi, S.B.; Odenkirk, M.T.; Jermendi, E.; Laporte, M.A.; Lutterodt, H.E.; et al. The State of Food Composition Databases: Data Attributes and FAIR Data Harmonization in the Era of Digital Innovation. *Front. Nutr.* **2025**, *12*, 1552367.
8. Borugadda, P.; Kalluri, H.K. A Comprehensive Analysis of Artificial Intelligence, Machine Learning, Deep Learning and Computer Vision in Food Science. *Journal of Future Foods* **2025**, *1*. https://doi.org/10.1016/j.jfutfo.2025.07.002.
9. Zhao, Z.; Wang, R.; Liu, M.; Bai, L.; Sun, Y. Application of Machine Vision in Food Computing: A Review. *Food Chemistry* **2025**, *463*, 141238. https://doi.org/10.1016/j.foodchem.2025.141238.
10. Ma, J.; Sun, D.W.; Qu, J.H.; Liu, D.; Pu, H.; Gao, W.H.; Zeng, X.A. Applications of Computer Vision for Assessing Quality of Agri-Food Products: A Review of Recent Research Advances. *Critical Reviews in Food Science and Nutrition* **2016**, *56*, 113–127. https://doi.org/10.1080/10408398.2012.715234.
11. Amugongo, L.M.; Kriebitz, A.; Boch, A.; Lütge, C. Mobile Computer Vision-Based Applications for Food Recognition and Volume and Calorific Estimation: A Systematic Review. *Healthcare* **2022**, *11*, 59. https://doi.org/10.3390/healthcare11010059.
12. Kawano, Y.; Yanai, K. Food Image Recognition with Deep Convolutional Features. In Proceedings of the Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication, New York, NY, USA, 2014. https://doi.org/10.1145/2638728.2638760.
13. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal Loss for Dense Object Detection. In Proceedings of the IEEE International Conference on Computer Vision (ICCV). IEEE, Oct 2017, pp. 2980–2988.
14. Pouladzadeh, P.; Kuhad, P.; Peddi, S.V.B.; Yassine, A.; Shirmohammadi, S. Food Calorie Measurement Using Deep Learning Neural Network. In Proceedings of the 2016 IEEE International Instrumentation and Measurement Technology Conference (I2MTC), Taipei, Taiwan, May 2016.
15. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, Jun 2016, pp. 770–778.
16. Li, X.; Grandvalet, Y.; Davoine, F.; Cheng, J.; Cui, Y.; Zhang, H.; Belongie, S.; Tsai, Y.H.; Yang, M.H. Transfer Learning in Computer Vision Tasks: Remember Where You Come From. *Image and Vision Computing* **2020**, *93*, 103853. https://doi.org/10.1016/j.imavis.2019.103853.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.