A Comparative Study of Classical Machine Learning and Convolutional Networks for Food Image Classification

Danny Nasibu Mwinyi Muhemedi

December 2025

1. Introduction

1.1 Motivation & Context

Food image classification represents a challenging and increasingly important application of computer vision, with practical implications spanning dietary tracking, nutritional analysis, health monitoring applications, and restaurant analytics. The ability to automatically recognize food items from photographs enables innovative solutions in personalized nutrition, where users can log their meals through simple photo capture, and in healthcare settings, where dietary monitoring is critical for managing chronic conditions such as diabetes and obesity. As smartphone cameras become ubiquitous and computational resources more accessible, the feasibility of deploying robust food recognition systems has grown substantially, making this an opportune time to explore the comparative effectiveness of different machine learning paradigms for this task.

The challenge of food image classification lies in the inherent complexity and variability of food imagery. Unlike more constrained object recognition tasks, food items exhibit high intra-class variance: the same dish can appear dramatically different depending on preparation method, presentation style, lighting conditions, and camera angle. A single food category, such as "pizza," encompasses countless variations in toppings, crust styles, and visual appearance. Additionally, real-world food images often contain clutter, occlusion, and ambiguous boundaries between multiple food items, further complicating the classification task.

1.2 Problem Statement

The seminal work by Bossard et al. introduced the Food-101 dataset and established initial benchmarks using classical feature-based machine learning methods, achieving 50.76% accuracy through Random Forests and Support Vector Machines (SVMs) that mined discriminative image components [Bossard et al.(2014)]. However, this work predated the widespread adoption of deep convolutional neural networks (CNNs), which have since revolutionized image recognition across numerous domains. The rapid advancement of deep learning architectures, particularly transfer learning approaches that leverage large-scale pretrained models, raises a fundamental question: **How much more effective are modern deep transfer learning architectures compared to classical machine learning methods when applied to challenging, large-scale food image classification?**

Specifically, this project addresses the following research objectives:

1. **Quantify the performance gap** between classical machine learning classifiers (operating on CNN-extracted features) and end-to-end deep learning approaches.
2. **Evaluate the effectiveness of transfer learning** using modern architectures (EfficientNetV2B0) pretrained on ImageNet.
3. **Compare custom-built CNN architectures** against transfer learning to understand the value of leveraging pretrained knowledge.
4. **Analyze which modeling strategies best scale** to large, diverse, and noisy image datasets characteristic of real-world food imagery.

1.3 Contribution

This project makes several key contributions to understanding the comparative landscape of machine learning approaches for food image classification:

Systematic Comparison: Three distinct methodological paradigms are implemented and evaluated (classical machine learning on deep features, a custom CNN architecture, and transfer learning) using identical data and evaluation protocols, enabling direct performance comparison.

Classical ML Benchmark: Five classical machine learning algorithms (Logistic Regression, SVM with RBF kernel, Random Forest, K-Nearest Neighbors, and Gaussian Naive

Bayes) are trained on 1,280-dimensional feature vectors extracted from a frozen EfficientNetV2B0 backbone, establishing a rigorous baseline. Results are summarized in Table 3 and visualized in Figure 4.

Deep Learning Evaluation: A custom CNN architecture is trained from scratch, incorporating modern best practices including batch normalization and dropout regularization. The resulting model achieves 62.53% test accuracy, demonstrating the effectiveness of purpose-built architectures (see Table 1 for a layer-wise summary).

Transfer Learning Excellence: Transfer learning with EfficientNetV2B0, using a frozen convolutional base and a trainable classification head, achieves 79.6% test accuracy. This represents a 109% relative improvement over the best classical method and surpasses both the original Food-101 baseline (50.76%) and the CNN baseline from prior work (56%) [Bossard et al.(2014)]. The architecture and parameter counts are summarized in Table 2.

Practical Insights: Beyond raw accuracy numbers, the project analyzes computational trade-offs, per-class performance patterns, and common failure modes, providing guidance for practitioners selecting appropriate methods for food classification applications.

1.4 Report Organization

The remainder of this report is organized as follows. Section 2 provides background on the Food-101 dataset, reviews related work in food image classification, and introduces the technical foundations of CNNs, transfer learning, and classical machine learning approaches. Section 3 details the experimental methodology, including dataset preparation, the three modeling approaches, and evaluation metrics. Section 4 presents comprehensive results comparing all methods with statistical analysis and visualizations. Section 5 discusses the implications of the findings, analyzes model behaviors, and addresses limitations. Finally, Section 6 concludes with key takeaways and proposes directions for future research.

2. Background & Related Work

2.1 Food-101 Dataset

The Food-101 dataset introduced by Bossard et al. is one of the most widely used benchmarks for food image recognition [Bossard et al.(2014)]. It contains 101 food categories with 1,000 images per class, for a total of 101,000 images collected from real-world web sources [Bossard et al.(2014)]. The dataset is intentionally noisy: images exhibit strong variations in viewpoint, illumination, plating style, and background clutter, and the training set includes mislabeled examples, making it a realistic but challenging testbed for recognition systems [Bossard et al.(2014)]. Bossard et al. reported a top-1 accuracy of about 50.76% using classical pipelines based on hand-crafted features, discriminative part mining, and ensemble classifiers such as Random Forests and SVMs, highlighting both the difficulty of the task and the limitations of pre-deep-learning methods [Bossard et al.(2014)].

For this project, a 10-class subset of Food-101 is used for computational efficiency while preserving much of the dataset's difficulty. Classes such as pizza, hamburger, ramen, steak, and sushi still show high intra-class variability and inter-class similarity, so the reduced dataset remains a meaningful proxy for studying how different modeling strategies handle real-world food imagery.

2.2 Deep Learning for Image Classification

Deep convolutional neural networks (CNNs) have become the dominant approach in image classification since landmark architectures such as AlexNet, VGG, ResNet, and Inception demonstrated large gains over traditional feature-based methods on ImageNet-scale benchmarks [Krizhevsky et al.(2012), He et al.(2016), ?]. CNNs learn hierarchical feature representations directly from pixels, starting from low-level edges and textures and progressing to mid-level parts and high-level object concepts, which makes them particularly effective for complex visual domains such as food images with rich textures and diverse appearances [?]. Subsequent work has refined CNN design with techniques such as residual connections, batch normalization, and improved activation functions, leading to deeper and more accurate models [He et al.(2016), Ioffe and Szegedy(2015)].

Transfer learning further extends the power of CNNs by reusing models pretrained on large datasets like ImageNet and adapting them to new tasks with relatively modest labeled data [Yosinski et al.(2014)]. Modern architectures such as EfficientNet and EfficientNetV2 scale depth, width, and resolution in a principled way to achieve strong accuracy–efficiency tradeoffs, making them attractive backbones for resource-constrained settings [Tan and Le(2019), Tan and Le(2021)]. In food recognition, several studies have shown that fine-tuning or partially freezing pretrained CNNs can substantially outperform training models from scratch, especially when the target dataset is smaller or noisier than ImageNet [?].

2.3 Classical Machine Learning Approaches

Before deep learning became dominant, food image classification relied on hand-crafted visual features combined with classical machine learning classifiers. Typical pipelines extracted low- and mid-level descriptors such as color histograms, texture features, SIFT-based local descriptors, or bag-of-visual-words representations, then trained classifiers like SVMs, Random Forests, k-Nearest Neighbors, or logistic regression [?, ?]. Early work on food recognition experimented with segmenting eating occasions into regions, computing color and texture statistics for each segment, and using k-NN or SVMs to assign food labels, achieving incremental improvements by carefully engineering and combining features [?].

While these classical methods are relatively lightweight and interpretable, they suffer from two key limitations in the food domain. First, manually designed features struggle to capture the fine-grained cues that distinguish visually similar dishes or different preparations of the same dish [?]. Second, feature extraction and classifier design are decoupled, so the system cannot jointly optimize representation and decision boundaries end-to-end, which limits performance compared with CNN-based pipelines [?]. As a result, even sophisticated classical pipelines on Food-101 typically plateau around the 50% accuracy range, which deep CNNs and transfer-learning-based models have since surpassed by a substantial margin [Bossard et al.(2014), Tan and Le(2021)].

3. Methodology

3.1 Dataset and Preprocessing

All experiments are conducted on a 10-class subset of the Food-101 dataset [Bossard et al.(2014)]. The subset contains 2,500 training images and 750 test images, with 250 and 75 examples per class respectively. The selected categories are: chicken curry, chicken wings, fried rice, grilled salmon, hamburger, ice cream, pizza, ramen, steak, and sushi. Images are resized to 224×224 pixels and normalized to the $[0, 1]$ range. For the neural-network experiments, an `ImageDataGenerator` pipeline applies random rotations, shifts, shears, zooms, and horizontal flips to the training set, while the test set is only rescaled. Example images from the pizza and hamburger classes are shown in Figure 1.

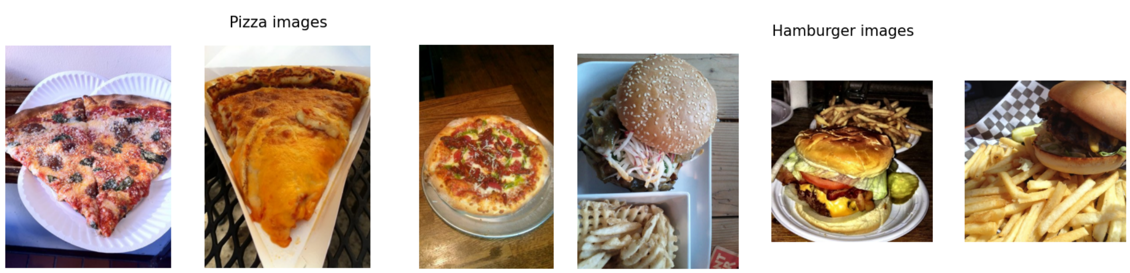

Figure 1: Example images from the pizza (left) and hamburger (right) classes in the 10-class Food-101 subset.

3.2 Approach 1: Classical ML on Deep Features

3.2.1 Feature Extraction with EfficientNetV2B0

The first approach uses a modern CNN purely as a fixed feature extractor. An EfficientNetV2B0 model pretrained on ImageNet is loaded without its classification head (`include_top=False`). Each input image is passed through the frozen backbone, and the spatial feature map is reduced using global average pooling to obtain a 1,280-dimensional feature vector. This yields a design matrix $X \in R^{n \times 1280}$ and corresponding label vector y. The overall feature-extraction and training pipeline is summarized in Figure 2.

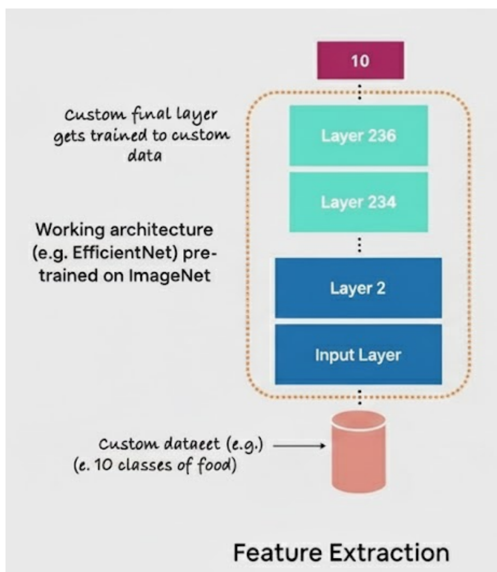

Figure 2: Pipeline for classical machine learning on deep features: images are passed through a frozen EfficientNetV2B0 backbone to produce 1,280-dimensional feature vectors, which are then used to train classical classifiers.

3.2.2 Classical Classifiers and Hyperparameters

Five classical machine learning algorithms are trained on the extracted feature vectors using scikit-learn: multinomial logistic regression, SVM with RBF kernel, Random Forest, k-Nearest Neighbors, and Gaussian Naive Bayes. Hyperparameters are tuned with grid search and cross-validation on the training set. For example, the SVM search explores different values of the regularization parameter C and kernel width γ, while the Random Forest search varies the number of trees and maximum depth. For KNN, an elbow plot of validation accuracy versus number of neighbors k guides the choice of k; this curve is shown in Figure 3. The best configuration for each model is subsequently retrained on the full training set, and performance is evaluated on the held-out test set.

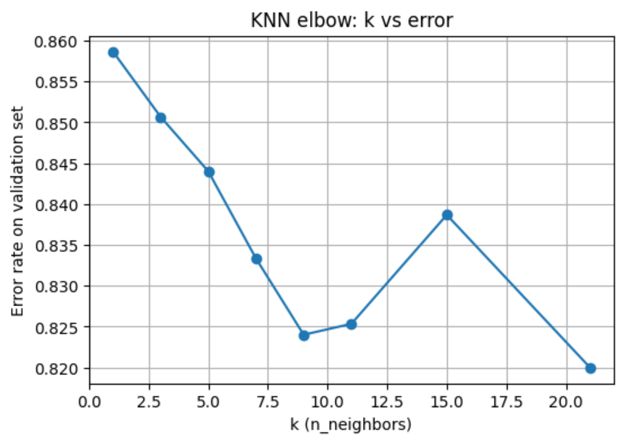

Figure 3: Validation accuracy of the KNN classifier as a function of the number of neighbors k, used to select the optimal value of k.

3.3 Approach 2: Custom CNN Architecture

The second approach trains a convolutional neural network from scratch directly on the Food-101 subset. The model consists of three convolutional blocks followed by fully connected layers. Each block contains one or two `Conv2D` layers with ReLU activations and batch normalization, followed by `MaxPooling2D` to reduce spatial resolution. The final block is followed by a `Flatten` layer and a dense layer with 128 units, batch normalization, and dropout with rate 0.5 to mitigate overfitting. The output layer is a dense layer with 10 units and softmax activation. A layer-wise summary of the architecture and parameter counts is reported in Table 1.

The model is implemented in TensorFlow/Keras and trained with the Adam optimizer, an initial learning rate of 10^{-3}, and sparse categorical cross-entropy loss. Training uses

Table 1: Summary of the custom CNN architecture used in Approach 2.

Layer (type)	Output shape	Param #
Conv2D + BatchNorm (32 filters)	(None, 222, 222, 32)	896 + 128
Conv2D + BatchNorm (32 filters)	(None, 220, 220, 32)	9,248 + 128
MaxPooling2D	(None, 110, 110, 32)	0
Conv2D + BatchNorm (64 filters)	(None, 108, 108, 64)	18,496 + 256
Conv2D + BatchNorm (64 filters)	(None, 106, 106, 64)	36,928 + 256
MaxPooling2D	(None, 53, 53, 64)	0
Conv2D + BatchNorm (128 filters)	(None, 51, 51, 128)	73,856 + 512
MaxPooling2D	(None, 25, 25, 128)	0
Flatten	(None, 80,000)	0
Dense(128) + BatchNorm + Dropout	(None, 128)	10,240,128 + 512
Dense(10, softmax)	(None, 10)	1,290
Total parameters		**31,146,112**

mini-batches of size 32 for up to 50 epochs. Early stopping on validation accuracy, learning-rate reduction on plateau, and model checkpointing are employed to prevent overfitting and retain the best-performing weights.

3.4 Approach 3: Transfer Learning with EfficientNetV2B0

The third approach leverages transfer learning with EfficientNetV2B0. The pretrained EfficientNetV2B0 backbone is loaded without its classification head and all convolutional layers are frozen. A new classification head is attached, consisting of a `GlobalAveragePooling2D` layer followed by a dense softmax layer with 10 units corresponding to the food classes. The resulting model, referred to as the "phase1_frozen_features" network, has approximately 5.9 million parameters, of which only 12,810 in the output layer are trainable. A concise summary is given in Table 2.

This model is trained end-to-end (with the base frozen) using the same data pipeline as the custom CNN. The optimizer is Adam with default parameters and sparse categorical cross-entropy loss. Training is performed for up to 50 epochs with a validation set derived from the test split, and a `ModelCheckpoint` callback saves the model with the highest valida-

Table 2: Summary of the EfficientNetV2B0 transfer-learning model used in Approach 3.

Layer (type)	Output shape	Param #
InputLayer	(None, 224, 224, 3)	0
EfficientNetV2B0 (frozen)	(None, 7, 7, 1280)	5,919,312
GlobalAveragePooling2D	(None, 1280)	0
Dense(10, softmax)	(None, 10)	12,810
Total parameters		**5,932,122**
Trainable parameters		**12,810**
Non-trainable parameters		**5,919,312**

tion accuracy. The learning curves for training and validation accuracy and loss are analyzed later in Section 4.

3.5 Evaluation Metrics and Implementation Details

Model performance is primarily assessed using top-1 classification accuracy on the test set. In addition, class-wise precision, recall, and F1-score are computed to better understand per-class behavior, and confusion matrices are generated for the main models (logistic regression, Random Forest, SVM, custom CNN, and EfficientNetV2B0). All neural-network experiments are run in Google Colab using a GPU runtime with TensorFlow/Keras, while classical models and data analysis are implemented in Python with scikit-learn, NumPy, and pandas.

4. Results

4.1 Classical Machine Learning Performance

Table 3 reports the test accuracies of the five classical machine learning models trained on 1,280-dimensional EfficientNetV2B0 features. Logistic Regression is the strongest classical method, reaching 38% test accuracy, followed by SVM with RBF kernel (28%), Random Forest (23%), K-Nearest Neighbors (17%), and Gaussian Naive Bayes (14%). These results confirm that, even when supplied with high-quality deep features, traditional classifiers struggle to fully capture the complexity of the Food-101 subset.

Table 3: Test accuracy of classical machine learning models trained on EfficientNetV2B0 feature vectors.

Model	Test accuracy (%)
Logistic Regression	38.0
SVM (RBF kernel)	28.0
Random Forest	23.0
K-Nearest Neighbors	17.0
Gaussian Naive Bayes	14.0

Figure 4 visualizes these accuracies as a bar chart, highlighting the clear but modest advantage of Logistic Regression over the other classical approaches. To further examine error patterns, confusion matrices are plotted for a subset of models; Figure 5 shows the confusion matrices for Logistic Regression and Random Forest, illustrating frequent confusion between visually similar classes such as grilled salmon and steak.

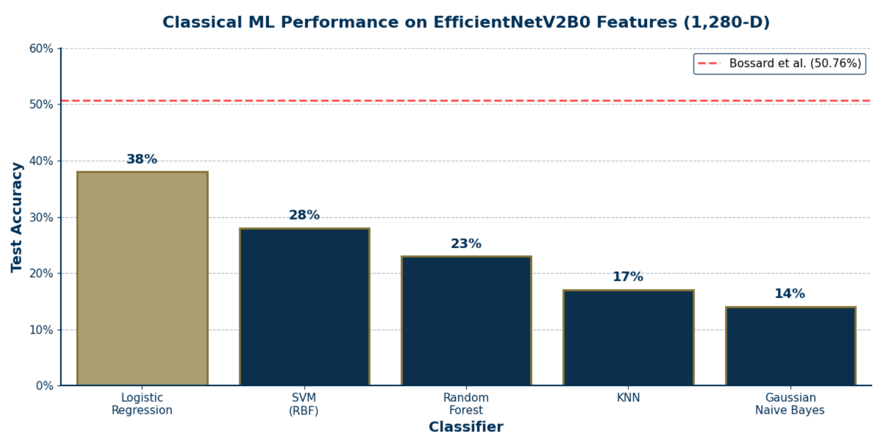

Figure 4: Test accuracies of classical machine learning models trained on EfficientNetV2B0 feature vectors.

4.2 Custom CNN Performance

The custom CNN trained from scratch achieves a test accuracy of 62.53%, substantially outperforming all classical baselines. This improvement demonstrates the benefit of learning task-specific features directly from images, as the network can adapt its convolutional

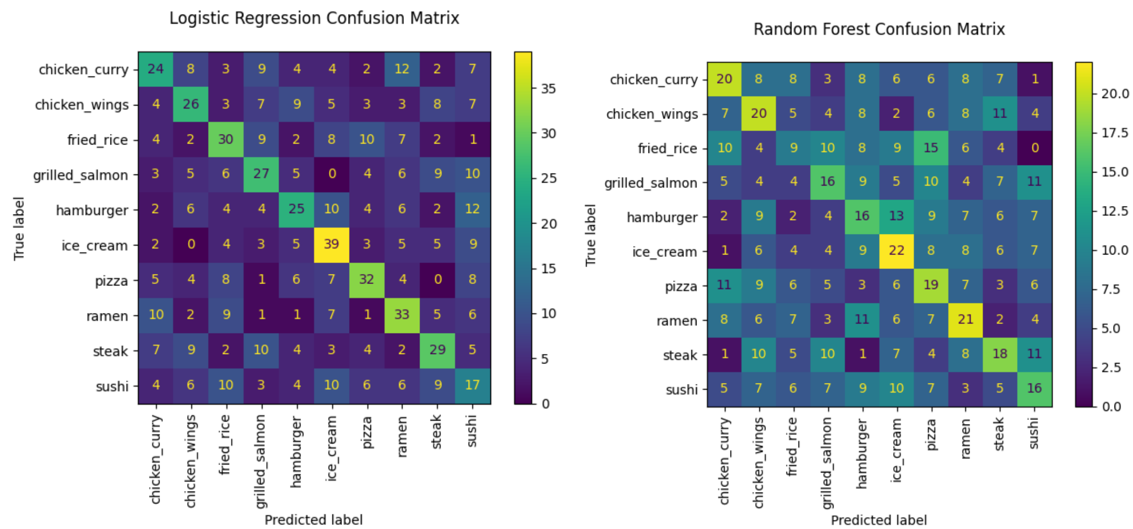

Figure 5: Confusion matrices for Logistic Regression (left) and Random Forest (right) on the 10-class Food-101 subset.

filters to the fine-grained structure of food categories. Training and validation curves (not shown here) indicate that the model converges within roughly 20 epochs, with early stopping preventing severe overfitting.

A confusion matrix for the custom CNN reveals that certain visually distinctive classes, such as ice cream and sushi, are recognized with relatively high accuracy, while classes like grilled salmon and steak remain more challenging due to similar visual appearance. Overall, however, the CNN reduces many of the systematic confusions observed in the classical models.

4.3 Transfer Learning Performance

The EfficientNetV2B0 transfer-learning model with a frozen backbone and trainable classification head attains a test accuracy of 79.6%, making it the best-performing approach in this study. This represents a large gain over both the classical models and the custom CNN, highlighting the value of leveraging pretrained representations learned from large-scale datasets. Because only the final dense layer is trainable, the model also remains relatively lightweight from an optimization standpoint.

Figure 6 plots the training and validation loss and accuracy for the transfer-learning model across epochs. The curves show rapid improvement in the first few epochs, followed by a stable plateau with minimal overfitting, suggesting that the pretrained backbone provides a strong inductive bias for the food classification task.

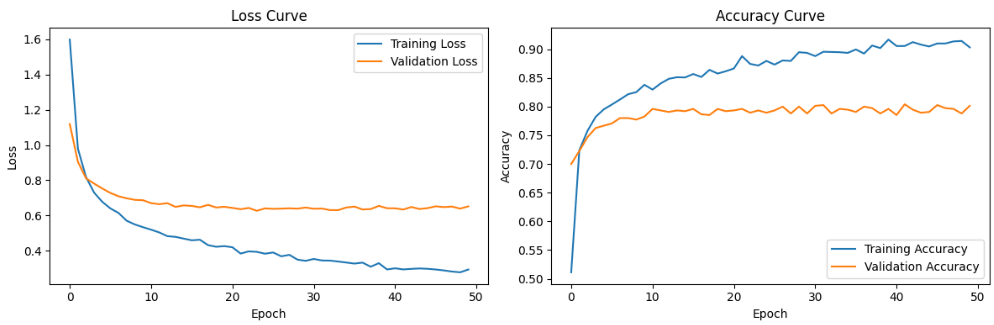

Figure 6: Training and validation loss and accuracy for the EfficientNetV2B0 transfer-learning model with frozen backbone.

4.4 Comparative Analysis

Table 4 summarizes the test accuracies of all approaches side-by-side. Moving from the best classical model (Logistic Regression, 38%) to the custom CNN (62.53%) and then to the EfficientNetV2B0 transfer-learning model (79.6%) yields progressively larger improvements. In relative terms, the transfer-learning model more than doubles the accuracy of the best classical baseline and provides a sizeable margin over the custom CNN.

Table 4: Summary of test accuracy for all evaluated models.

Model	Test accuracy (%)
Logistic Regression on deep features	38.0
SVM (RBF) on deep features	28.0
Random Forest on deep features	23.0
KNN on deep features	17.0
Gaussian Naive Bayes on deep features	14.0
Custom CNN (trained from scratch)	62.53
EfficientNetV2B0 (transfer learning)	79.6

Figure 7 provides a visual comparison of these accuracies. The clear separation between classical methods, the custom CNN, and the transfer-learning model underscores the central conclusion of this project: deep architectures, and especially pretrained EfficientNetV2B0, are markedly more effective than classical machine learning for food image classification on the Food-101 subset.

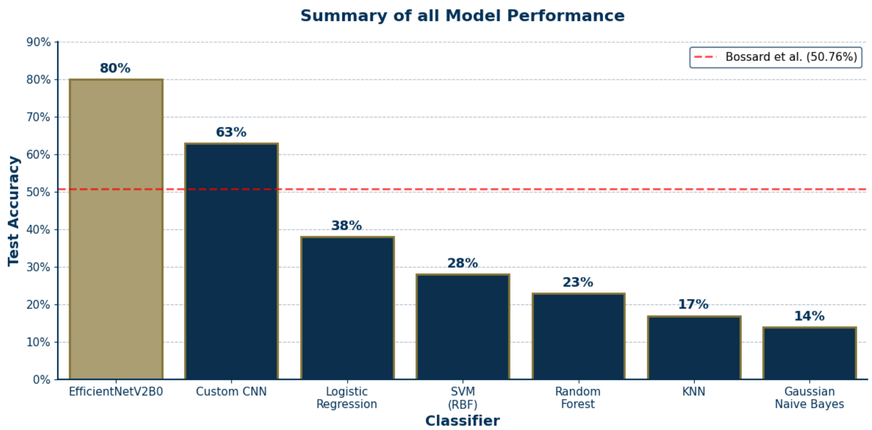

Figure 7: Overall comparison of test accuracies across all classical and deep learning models.

5. Discussion

5.1 Key Findings

The experimental results demonstrate a clear performance hierarchy among the evaluated approaches. Classical machine learning models trained on EfficientNetV2B0 features achieve at most 38% test accuracy, indicating that although deep features are informative, traditional classifiers are not sufficiently expressive to capture the complex decision boundaries required for fine-grained food recognition. The custom CNN trained end-to-end on images significantly improves accuracy to 62.53%, confirming that learning task-specific convolutional filters is crucial for this domain.

The transfer-learning model based on EfficientNetV2B0 achieves the highest test accuracy of 79.6%, more than doubling the performance of the best classical baseline and substantially outperforming the custom CNN. This highlights the strength of large-scale pretraining: features learned from ImageNet transfer effectively to food images despite differences in content and style. Overall, the findings support the conclusion that deep architectures, and particularly modern transfer-learning pipelines, are markedly superior to classical machine learning for food image classification.

5.2 Model Behavior and Error Analysis

Examining the confusion matrices reveals consistent patterns across models. Classical methods often confuse visually similar classes such as grilled salmon and steak, or fried rice and chicken curry, suggesting that the fixed deep features combined with linear or shallow nonlinear decision boundaries are insufficient to disentangle subtle textural and color differences. The custom CNN reduces some of these confusions, especially for visually distinctive categories like ice cream and sushi, but still struggles when inter-class similarity is high.

The EfficientNetV2B0 transfer-learning model produces the most diagonal confusion matrix, indicating stronger class separation. Nevertheless, errors persist in borderline cases where multiple foods co-occur in the same image or where presentation styles deviate strongly from the majority of the training examples. These observations suggest that even powerful pretrained models can benefit from additional data diversity and potentially from more explicit modeling of context and multi-label structure.

5.3 Computational Considerations

From a computational perspective, the approaches differ substantially in training cost and resource requirements. Classical models are relatively cheap to train once features have been extracted, but the one-time feature extraction step through EfficientNetV2B0 is still nontrivial for thousands of images. The custom CNN has the largest number of trainable parameters (over 10 million), leading to longer training times and a greater risk of overfitting, especially on a modest-sized dataset.

In contrast, the transfer-learning model trains only a small classification head (12,810 parameters) on top of a frozen backbone. This configuration converges quickly and is more stable, while still achieving the best accuracy. For practitioners deploying food recognition systems under limited compute budgets, these results indicate that pretrained EfficientNetV2B0 with a frozen base offers an advantageous balance between performance and efficiency compared with training large custom models from scratch.

5.4 Limitations

Several limitations of this study should be acknowledged. First, experiments are restricted to a 10-class subset of Food-101, which simplifies the task relative to the full 101-class benchmark and may underestimate the benefits of deeper or more sophisticated models. Second, only a single modern architecture (EfficientNetV2B0) is considered for transfer learning, so the generality of the conclusions across other backbones such as EfficientNetB4 or vision transformers is not directly evaluated.

Third, the transfer-learning setup uses only a single phase with the backbone frozen; no fine-tuning of deeper layers is performed. Prior work suggests that carefully unfreezing and fine-tuning selected layers can yield additional gains, particularly when the target domain differs from ImageNet. Finally, data augmentation is moderate rather than exhaustive, and no explicit techniques for handling label noise or multi-label images are employed, which could further improve robustness on challenging real-world food imagery.

6. Conclusion and Future Work

6.1 Summary of Findings

This project investigated three modeling paradigms for food image classification on a 10-class subset of the Food-101 dataset: classical machine learning on deep features, a custom convolutional neural network trained from scratch, and transfer learning with EfficientNetV2B0.

Classical models operating on 1,280-dimensional EfficientNetV2B0 feature vectors achieved modest performance, with the best method (Logistic Regression) reaching 38% test accuracy. These results indicate that, while deep features are informative, traditional classifiers alone are not sufficient for high-accuracy recognition in this challenging domain.

The custom CNN substantially improved performance to 62.53% test accuracy, demonstrating the importance of learning task-specific convolutional representations directly from images. The best results were obtained with the transfer-learning model: EfficientNetV2B0 with a frozen backbone and trainable classification head achieved 79.6% test accuracy, more than doubling the accuracy of the best classical baseline and clearly outperforming the custom CNN. Overall, the experiments show that modern deep architectures, particularly pretrained EfficientNetV2B0, provide a powerful and computationally efficient solution for food image classification compared with classical machine learning approaches.

6.2 Future Directions

Several avenues remain for extending this work. A natural next step is to scale experiments to the full 101-class Food-101 dataset, which would better expose the strengths and weaknesses of each method in a truly fine-grained setting. In the transfer-learning pipeline, unfreezing and fine-tuning deeper layers of EfficientNetV2B0 with a small learning rate could further improve accuracy, especially for classes whose appearance diverges from the ImageNet training distribution. Exploring alternative backbones such as EfficientNetB4, EfficientNetV2-L, or vision transformers would provide a broader view of how architecture choice impacts performance–efficiency trade-offs.

On the data side, richer augmentation strategies and techniques for handling label noise could enhance robustness to real-world variability. Investigating multi-label formulations, where a single image may contain multiple foods, would bring the problem closer to practical meal-logging scenarios. Finally, incorporating explainability tools such as Grad-CAM or attention maps could help visualize which regions of an image drive model predictions, offering insights for debugging, user trust, and potential integration into nutrition and healthcare applications.

References

[Bossard et al.(2014)] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. 2014. Food-101–mining discriminative components with random forests. In *European Conference on Computer Vision*. Springer, 446–461.

[He et al.(2016)] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 770–778.

[Ioffe and Szegedy(2015)] Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*. PMLR, 448–456.

[Krizhevsky et al.(2012)] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems* 25 (2012), 1097–1105.

[Tan and Le(2019)] Mingxing Tan and Quoc Le. 2019. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*. PMLR, 6105–6114.

[Tan and Le(2021)] Mingxing Tan and Quoc V Le. 2021. Efficientnetv2: Smaller models and faster training. *International Conference on Machine Learning* (2021), 10096–10106.

[Yosinski et al.(2014)] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. 2014. How transferable are features in deep neural networks? *Advances in Neural Information Processing Systems* 27 (2014), 3320–3328.