**Literature Review of Food Image Classification**

Athak Yadav [E22CSEU0861]

Harshath Venkatesh [E22CSEU0869]

Yash Ozha [E22CSEU0854]

Juned Ansari [E22CSEU0741]

Bennett University

# Introduction

*The Integration of computer vision and machine learning techniques into food recognition systems is critical in India, where traditional methods of dietary tracking rely mainly on manual input. FoodSnap is a web-based tool that employs advanced computer vision and machine learning techniques to handle the unique issues of Indian cuisine, which frequently lacks standardized nutritional reporting. This is especially pertinent in India, where malnutrition and obesity are prevalent. The program also attempts to increase user engagement and dietary awareness through simple photo analysis that balances accuracy and usability. This literature study looks at existing technological frameworks, implementation issues, and potential solutions to improve automated food recognition systems.*

**Objective**

To gain a thorough understanding of current research on computer vision and machine learning for food image recognition, with a focus on Indian cuisine identification and nutritional content analysis. This review will look at existing methodologies, technological frameworks, and challenges in automated food recognition systems, identify gaps in current research, particularly in Indian food classification, and propose promising directions for advancing the field of AI-powered nutritional analysis via image processing using SWIN Transformer.

Key focus areas include:

1. Computer vision approaches for recognizing and classifying food images
2. Machine learning approaches tailored to various cultural cuisines
3. Current techniques for collecting nutritional data from food photos
4. Challenges and solutions for Indian food recognition due to its visual complexity
5. Integrating food identification systems with nutritional databases.
6. User interface considerations for food recognition programs.

**Literature Survey**

# 1. Food Image Classification and Data Extraction Using Convolutional Neural Network and Web Crawlers

# Dataset: A subset of the Food-101 dataset with images resized to 299x299 and reduced noise, focusing on 20 classes. Augmentation techniques like shear range (0.2), zoom range (0.2), and horizontal flips were applied to enhance robustness and minimize overfitting.

# Model and Training: The Inception V3 model pretrained on ImageNet was fine-tuned for transfer learning. Additional custom layers included dense (128 neurons) with dropout and a final dense layer for 20 classes. Training was performed on Google Colab with a Tesla T4 GPU for 20 epochs using an SGD optimizer with a 0.9 momentum and a learning rate of 0.0001. Categorical cross-entropy was used as the loss function.

# Results: Achieved 97.00% training accuracy and 92.23% validation accuracy for 20 classes, with training and validation losses of 0.10 and 0.3092, respectively. For 25 classes, the training accuracy was 96.52%, and validation accuracy was 91.46%. The model demonstrated minimal overfitting with a 5% gap between training and validation accuracies and an F1 score of 0.99867.

**2. Food Image Recognition by Using Convolutional Neural Networks (CNNs)**

**Dataset**: A small-scale dataset of 5822 images across 10 food categories (e.g., apple, banana, broccoli), collected from ImageNet. Images were resized to 128x128 and split into training and test sets (4:1 ratio). Data augmentation techniques like rotation, translation, and scaling were applied to expand the dataset.

**Model and Training**: A five-layer CNN architecture was implemented with three convolutional-pooling layers (7x7, 5x5, and 3x3 kernels) followed by a fully connected layer with 128 neurons. Dropout rates of 0.25 and 0.5 were applied to reduce overfitting. The network used the SGD optimizer with a dynamic learning rate. Training included up to 600 epochs, with augmentation to improve model performance.

**Results**: Without augmentation, the CNN achieved 74% test accuracy, outperforming the Bag-of-Features model (56%). With data augmentation, test accuracy exceeded 90%, demonstrating significant performance improvement and mitigating overfitting. Increasing training epochs beyond 400 showed diminishing returns due to overtraining risks.

**3. Food Image Classification with Convolutional Neural Networks**

**Dataset:** Food-101, consisting of 101,000 images across 101 categories, with 750 noisy training images and 250 manually reviewed test images per category. Images were resized to 128x128 or 256x256 during initial experiments and later resized to model specifications for transfer learning. Data augmentation included rotation, shifting, and flipping to address overfitting.

**Model and Training:** Initial models trained from scratch included a baseline model (4 convolutional and 2 fully-connected layers) and AlexNet with data augmentation. Transfer learning was conducted using VGG16, ResNet50, and InceptionV3 pre-trained on ImageNet weights, with modified top layers tailored for Food-101. Training setups incorporated AWS EC2 instances and used categorical cross-entropy as the loss function. Hyperparameters like dropout, learning rate, and optimizer configurations were tuned.

**Results:** The highest-performing model was InceptionV3, which achieved 61.4% top-1 accuracy and 85.2% top-5 accuracy using custom preprocessing and unfrozen top layers. This outperformed the original Food-101 implementation but fell short of DeepFood's performance. ResNet50 and AlexNet reached 42.8% and 32.8% top-1 accuracy, respectively, under specific configurations.

**4. REVOLUTIONIZING COMPUTER VISION: ENHANCED FOOD IMAGE CLASSIFICATION WITH SWIN TRANSFORMER AND SVM CLASSIFIER**

* **Dataset**: Food-101, consisting of 101,000 images across 101 food categories. Images were resized to 224x224 pixels, and data augmentation techniques such as random rotations, translations, and brightness adjustments were applied. The dataset was split into 75% for training and 25% for testing.
* **Model and Training**: The Swin Transformer was fine-tuned using the swin\_base\_patch4\_window7\_224 variant as a feature extractor. The model was trained using the Adam optimizer with an initial learning rate of 1e-3, batch size of 64, and CrossEntropyLoss as the objective function. Early stopping was employed to prevent overfitting. The extracted features were further classified using an SVM with RBF and linear kernels, tuned via GridSearchCV for optimal performance.
* **Results**: The Swin Transformer model alone achieved a top-1 accuracy of 88.57% and a top-5 accuracy of 97.54% on validation data, with a testing accuracy of 89.89%. When combined with an SVM classifier using RBF kernel and C=1C=1C=1, the testing accuracy improved to 91.05%, demonstrating the enhanced classification performance of the hybrid approach.

**5. Nutritional composition analysis in food images: an innovative Swin Transformer approach**

**Dataset:**

Two datasets were used:

1. **Nutrition5k:** Contains 5,000 food images (224x224 pixels), annotated with detailed nutritional information, covering diverse food types like vegetables, meats, and grains, taken under varied conditions.
2. **ChinaMartFood109:** Includes over 100,000 images (224x224 pixels) of 109 categories of Chinese foods, featuring annotations for nutritional components such as calories, proteins, and fats, taken in different settings like restaurants and kitchens.

**Model and Training:**

The study employed a hybrid deep learning model integrating **EfficientNet** (for efficient feature extraction), **Swin Transformer** (to capture long-range dependencies), and **Feature Pyramid Network (FPN)** (for multi-scale feature fusion). Training utilized the Adam optimizer with an initial learning rate of 0.001, adjusted dynamically during 100 epochs with a batch size of 32. A multi-task loss function optimized the detection of calories, fats, carbohydrates, and protein. Training was conducted on a high-performance workstation with NVIDIA Tesla V100 GPU.

**Results:**

* On **Nutrition5k**, the model achieved a **top-1 accuracy** of **79.50%** and **top-5 accuracy** of **95.66%**, with a Mean Absolute Percentage Error (MAPE) of **14.72%** for calorie prediction.
* On **ChinaMartFood109**, it reached a **top-1 accuracy** of **80.25%** and **top-5 accuracy** of **96.98%**, with a calorie MAPE of **15.21%**.
* Outperformed baseline models such as VGG16, InceptionV3, ResNet152, and Swin-Nutrition across all metrics.

This approach demonstrates enhanced robustness and accuracy for food nutrient detection, providing a rapid, non-destructive solution for dietary monitoring.

6. **Highly Accurate Food/Non-Food Image Classification Based on a Deep Convolutional Neural Network**

**Dataset:**

Three datasets were used:

1. **Instagram-#Food Dataset (IFD):** 4,230 food images and 5,428 non-food images collected from Instagram's “#food” search results, manually annotated.
2. **Food-101/Caltech-256 Dataset (FCD):** 25,250 food images from Food-101 and 28,322 non-food images from Caltech-256, excluding categories like grapes and hamburgers.
3. **Custom Dataset from [4]:** 1,234 food and 1,980 non-food images collected from social media.

**Model and Training:**

The study used the **CNN-NIN** (Convolutional Neural Network with Network in Network) architecture, featuring four convolutional layers and two mlpconv layers. The model was fine-tuned using ImageNet pre-trained weights, with output layers adjusted for binary classification (food/non-food). A comparison was made between models trained from scratch and fine-tuned models, demonstrating the latter's superior performance and efficiency. Training and evaluation used the Caffe deep learning framework.

**Results:**

1. **Instagram-#Food Dataset (IFD):** Achieved 95.1% accuracy at an 80/20 train-test split.
2. **Food-101/Caltech-256 Dataset (FCD):** Achieved 96.4% accuracy at an 80/20 split.
3. **Custom Dataset from [4]:** Achieved 99.1% accuracy, outperforming previous methods like SVM-based classifiers and AlexNet.
4. **Cross-dataset evaluations:** Training on FCD and testing on IFD achieved 91.5% accuracy, while the reverse achieved 90.6%, indicating dataset-specific statistical differences.

The fine-tuned CNN-NIN model demonstrated robust performance in food/non-food classification, significantly surpassing baseline methods and providing potential for pre-processing in food recognition systems.

**7. Food Classification from Images Using Convolutional Neural Networks David J. Attokaren, Ian G. Fernandes, A. Sriram, Y.V. Srinivasa Murthy, and Shashidhar G. Koolagudi**

* **Dataset**: Food-101 with 101,000 images across 101 categories, resized to 299x299, with noise and labeling corrected.
* **Model and Training**: Google Inception V3 pretrained on ImageNet, retrained with enhancements like AvgPool, Dropout (0.4), and Softmax layers; SGD optimizer with a decreasing learning rate, trained for 32 epochs.
* **Results**: Achieved 86.97% top-1 accuracy and 97.42% top-5 accuracy on the Food-101 dataset, outperforming traditional classifiers.

**8. Analysis & Numerical Simulation of Indian Food Image Classification Using Convolutional Neural Network**

* **Dataset**: A custom dataset of South Indian cuisine, complemented by data from the Yummly dataset, was used. Images were resized to 299x299 pixels, with noise and labeling inconsistencies corrected during preprocessing.
* **Model and Training**: The proposed methodology integrated an Edge-Adaptive Convolutional Neural Network (EA-CNN) for segmentation and Inception-V3 for classification. The Inception-V3 model was pretrained on ImageNet, with further enhancements applied for segmentation and classification tasks. Training involved optimizing features extracted from EA-CNN for input to the classifier. The training process used a batch size of 128, with data augmentation applied to improve model generalization.
* **Results**: The proposed approach achieved 91.79% classification accuracy on the custom dataset when integrating segmentation features, significantly outperforming baseline models (e.g., Naive Bayes: 71.4%, Random Forest: 75.2%, and Linear SVC: 78%). Top-1, Top-5, and Top-10 accuracies of 88.14%, 90.12%, and 91.79% were observed, respectively. When evaluated with real-time collected data, the classification accuracy increased to 96.27%, highlighting the model's robustness in handling diverse Indian food images.

**9. Indian Food Image Classification and Recognition with Transfer Learning Technique**

* **Dataset**: The "9 Indian Food" dataset, comprising 2,700 images across nine categories, with approximately 300 images per category, was used. Images were resized to 224x224 pixels, normalized, and preprocessed to remove noise and enhance contrast. Data augmentation techniques, such as rotation (20 degrees) and horizontal flipping, were applied to enhance model generalization.
* **Model and Training**: The study utilized a self-designed Convolutional Neural Network (CNN) and MobileNetV3 for transfer learning. The self-designed CNN featured an architecture starting with 32 filters, doubling up to 256 filters across layers, followed by max pooling, dense layers, and a SoftMax activation function. MobileNetV3, pretrained on ImageNet, was fine-tuned using dense layers and a dropout layer for classification. Training used the Adam optimizer, categorical crossentropy loss, a batch size of 32, and 50 epochs for the self-designed CNN, while MobileNetV3 was trained for 10 epochs.
* **Results**: The self-designed CNN achieved an accuracy of 85.1%. Incorporating MobileNetV3 through transfer learning improved accuracy to 93.3%. When data augmentation was added, the accuracy further increased to 95.3%. Other metrics such as precision (95.32%), recall (95.28%), and F1-score (95.25%) also demonstrated high performance, showcasing the model's effectiveness in classifying Indian food images.

**10. A New CNN-Based Single-Ingredient Classification Model and Its Application in Food Image Segmentation - by Ziyi Zhu and Ying Dai**

* **Dataset**: The SI110 dataset contains 10,750 images spanning 110 ingredient categories, covering various cooking and cutting methods. Images are categorized into four hierarchical levels based on biological taxonomy, with 80% used for training and 20% for testing.
* **Model and Training**: The proposed AttNet CNN architecture features eight blocks with batch normalization, sigmoid activations, and global average pooling layers. Variants of AttNet and pretrained ResNet18/EfficientNet-B0 models were fine-tuned, trained using cross-entropy loss, Adam optimizer, and hierarchical multi-level learning for 30–50 epochs.
* **Results**: Achieved 86.84% accuracy on single-ingredient classification (ResNet18) and a mIoU of 0.654 for ingredient segmentation (AttNet Method 2), outperforming previous segmentation approaches.

**11. Indian Food Image Recognition using a Deep Learning Approach**

* **Dataset**: The "India-Food-21-Categories-Small" dataset from Kaggle was used, containing 460 RGB images across 21 Indian food categories. Images were resized to 256x256x3 pixels for compatibility with AlexNet. Data augmentation techniques, including horizontal flipping, vertical flipping, and combined flips, were applied to increase the dataset size to 1,840 images.
* **Model and Training**: The study employed the AlexNet architecture fine-tuned for the classification task, with the last fully connected layer modified to handle 21 classes instead of the original 1,000. Training parameters included SGDM optimizer, a learning rate of 0.0001, a mini-batch size of 10, and five epochs. Data augmentation significantly enhanced performance, helping to mitigate overfitting in the limited dataset.
* **Results**: Without data augmentation, the model achieved 75% accuracy. Incorporating data augmentation improved accuracy to 96.6%, showcasing its effectiveness in enhancing classification performance with a limited dataset. The proposed model outperformed similar works requiring larger datasets and longer training times, achieving high accuracy in less time using modest computational resources.

**12. Indian Food Image Classification Using K-Nearest-Neighbour and Support-Vector-Machines**

**Dataset:** Contains 200 Indian food images, split into 80% training and 20% testing sets. Features were extracted based on color and shape, preprocessed through HSV conversion, noise removal, cropping, and edge detection.

**Model and Training:**

* **K-Nearest Neighbor (KNN):** Utilizes Euclidean distance to classify features from training data.
* **Support Vector Machine (SVM):** Separates data classes using hyperplanes for higher generalization accuracy.Both models rely on combined color and shape features for classification.

**Results:**

SVM outperformed KNN in classifying Indian food images, achieving higher accuracy for clustered food (85% vs. 75%) and individual items like rice and idly. Future work suggests integrating data mining techniques for better food recommendations.

**13. ODNET: OPTIMIZED DENSENET FOR INDIAN FOOD CLASSIFICATION**

**Dataset:** 9\_Indian\_food dataset with 2,700+ images across 9 categories, preprocessed with normalization and scaling.

**Model and Training:**

* **MobileNetV3:** Pretrained on ImageNet, 50 epochs, learning rate 0.0001, Adam optimizer.
* **Optimized DenseNet:** DenseNet-121 with Mish activation, fine-tuned for Indian food classification, same training parameters.

**Results:**

* MobileNetV3: 92.39% testing accuracy.
* Optimized DenseNet: 95.10% testing accuracy, outperforming MobileNetV3.

DenseNet shows state-of-the-art accuracy for Indian food classification, with potential healthcare applications.

**14. IndianFoodNet: Detecting Indian Food Items Using Deep Learning**

* **Dataset:** IndianFoodNet with over 5,500 images and 15,000+ annotations across 30 Indian food classes. Images were resized to 640×640 and annotated for object detection.
* **Model and Training:** Three YOLO variants (YOLO5, YOLO7, YOLO8) were used for object detection. Training was conducted using Google Colab with metrics like precision, recall, and mean average precision (mAP).
  + YOLO5 and YOLO8 showed higher performance compared to YOLO7.
  + Key parameters: epochs (5), learning rate adjustments, and bounding box optimizations.
* **Results:** YOLO8 achieved the best precision (0.775), recall (0.719), and mAP50 (0.807), outperforming YOLO7 and YOLO5. The study identifies YOLO8 as the optimal model for Indian food detection, paving the way for calorie estimation applications.

**15. Deep Indian Delicacy: Classification of Indian Food Images using Convolutional Neural Networks**

Dataset: The dataset used comprises 60,000 grayscale images of size 280x280, categorized into 10 classes of Indian food, such as Aloo Paratha, Dosa, Idli, Jalebi, Kachori, Omelette, Paneer Tikka, Poha, Samosa, and Tea. The dataset was split into 50,000 images for training and 10,000 images for testing. Data augmentation was applied using affine transformations like rotation and scaling to enhance the limited dataset.

Model and Training: The proposed architecture featured an input layer, five convolutional layers with ReLU activation, max-pooling, a fully connected layer with dropout (0.8), and a softmax output layer. Each convolutional layer retained a 5x5 filter size and produced increasing numbers of feature maps, culminating in a fully connected layer of 1024 neurons. The Adam optimizer was employed for weight updates, and categorical cross-entropy was used as the loss function. The model was trained in a GPU-accelerated environment (Nvidia GTX 1050 Ti) and validated on 200 images.

Results: The model achieved 96.95% classification accuracy on the test set with a validation accuracy of 93.5% and a loss of 0.143. The performance was observed to improve significantly with larger datasets and more training epochs, demonstrating the impact of sufficient data and computational resources on the model's ability to generalize effectively.

**16. A FOOD IMAGE RECOGNITION SYSTEM WITH MULTIPLE KERNEL LEARNING**

* **Dataset:** A custom dataset of 50 food categories, with 100 images per category collected from the web. Images include common Japanese foods in various real-world conditions, with preprocessing to isolate target food regions.
* **Model and Training:** A Multiple Kernel Learning (MKL) method combined image features like Bag-of-Features (BoF), color histograms, and Gabor texture features. BoF used three sampling strategies (Difference of Gaussian, random, and grid) with codebook sizes of 1000 and 2000. The classification employed an MKL-SVM with χ² kernels, integrating multiple feature types adaptively. The model was trained using 5-fold cross-validation.
* **Results:** Achieved a 61.34% classification rate for top-1 predictions across 50 food categories, with top-3 predictions increasing the rate to 80.05%. The MKL method significantly outperformed individual feature methods, demonstrating its effectiveness in combining diverse image features. A prototype system tested with 166 user-uploaded food images yielded a 37.35% classification rate, highlighting the practical potential of the approach.

**17. Development of Korean Food Image Classification Model Using Public Food Image Dataset and Deep Learning Methods**

* **Dataset:** A Korean food image dataset provided by the AI-Hub platform, consisting of 150,610 images across 150 classes. The dataset includes images from 27 food categories and was augmented to a total of 527,135 images. Images were resized to 331x331 pixels for training.
* **Model and Training:** The study used pre-trained deep learning models (ResNet-50V2, ResNet-101V2, ResNet-152V2, InceptionResNetV2, NasNetLarge, and MobileNetV2), fine-tuned for the classification of Korean food. The models were trained in two stages: an initial stage using ImageNet weights and a fine-tuning stage with the Korean food dataset. The training used an Adam optimizer with a learning rate of 0.001 (initial stage) and 0.0001 (fine-tuning stage), and categorical cross-entropy loss. Data augmentation included brightness, contrast, and saturation adjustments, along with horizontal flipping.
* **Results:** InceptionResNetV2 outperformed other models with the highest accuracy in the evaluation. The average top-1 recall across models was 0.6, with variations across food categories. Foods with complex mixing or visual similarities (e.g., stews and soups) showed lower classification accuracy. The model exhibited robust performance in classifying simpler or visually distinct foods, achieving the highest precision and recall in categories like spinach salad and braised lotus roots.

**18. Food classification of Indian cuisines using handcrafted features and vision transformer**

* **Dataset:** A custom dataset of 7,467 images covering 13 Indian food categories (e.g., Ghevar, Idli, Dosa). The dataset features 300-500 images per class with diverse representations of cultural food items.
* **Model and Training:** A hybrid architecture combining Vision Transformer (ViT) and hand-crafted features (GIST, HoG, LBP). The Vision Transformer is trained to extract global features from image patches, while hand-crafted features enhance texture and shape recognition. SVM is used as the final classifier for prediction, with preprocessing techniques such as mean normalization, standardization, and ZCA whitening applied to the input data.
* **Results:** The hybrid approach achieved an accuracy of 94.63%, with sensitivity at 84.42%, specificity at 95.23%, and a kappa coefficient of 0.93. It outperformed CNN ensembles and other state-of-the-art methods, demonstrating the effectiveness of combining Vision Transformer features with hand-crafted descriptors.

**19. Food Detection and Recognition Using Convolutional Neural Network**

**Dataset:**

* Collected from a public food-logging application named FoodLog.
* Consists of 170,000 food images from everyday meals, narrowed to the 10 most frequent items for recognition tasks.
* Images scaled to 80×80, cropped to 64×64 during training/testing, with a 6-fold cross-validation setup.

**Model and Training:**

* Used a Convolutional Neural Network (CNN) implemented in cuda-convnet.
* Varied architecture with layers (2-4), kernel sizes (5×5, 7×7, 9×9), and Local Response Normalization (LRN) applied post-pooling.
* Compared against traditional methods like SVMs using handcrafted features (e.g., color histograms, GIST, SIFT-BoW).

**Results:**

* Recognition accuracy improved significantly with CNN compared to SVM-based methods:
  + CNN: Up to 73.7% accuracy (2-layer network with optimal parameters).
  + SVM-based methods: Ranged between 50-60%.
* In the food detection task, CNN achieved 93.8% accuracy, surpassing the baseline method (89.7%).
* Observation of convolutional kernels revealed color dominance in feature extraction, validating the significance of color features in food recognition tasks.

**20. Food-image Classification Using Neural Network Model. Int. J. of Electronics Engineering and Applications. 2021**

* **Dataset**: Food-101 dataset with 101 categories, images resized to 224x224 for uniformity, split into training (1000 images per class), validation, and test sets (~500 images per class each). The dataset was left uncleaned to test model robustness against noise.
* **Model and Training**: Fine-tuned Inception-V3 pretrained on ImageNet, modified by removing the last three layers to adapt to the new task. The model was trained using transfer learning with a ten-fold cross-validation approach to evaluate performance.
* **Results**: Achieved enhanced sensitivity and specificity compared to baseline DCNN, demonstrating significant improvements with transfer learning.

# 

| S.no | Name | Model | Accuracy | Author | Merits | Demerits |
| --- | --- | --- | --- | --- | --- | --- |
| [1] | Food Image Classification and Data Extraction | Convolutional Neural Network and Web Crawlers | 97% training accuracy and 92.23% validation accuracy | A Chaitanya , Jayashree Shetty , Priyamvada Chiplunkar | High training and validation accuracy, efficient use of CNNs for food image classification | Limited application to Indian cuisine, potential over-reliance on data quality from web crawlers |
| [2] | Food Image Recognition | Convolutional Neural Networks | 74% | Yuzhen Lu | Provides a basic framework for food image recognition using CNNs | Low accuracy compared to other methods, lacks focus on regional or cultural nuances in food recognition |
| [3] | Food Image Classification | Convolutional Neural Networks | 85.2% | Malina Jiang | Moderate accuracy, demonstrates the potential of CNNs in food classification | May not perform well on visually complex food images, limited dataset diversity |
| [4] | REVOLUTIONIZING COMPUTER VISION: ENHANCED FOOD IMAGE CLASSIFICATION | SWIN Transfromer and SVM Classifier | 74% | Elpina , Gede Putra kusuma | Introduced SWIN Transformer, combining with SVM for improved classification | Lower accuracy for diverse food items, requires more robust datasets |
| [5] | Nutritional composition analysis in food images | Swin Transformer | 90% | Hui Wang, Haixia Tian, Ronghui Ju, Liyan Ma, Ling Yann, Jingyao Chen  and Feng Liu | High accuracy for nutritional composition analysis, innovative use of Swin Transformer | Focused primarily on nutritional analysis, limited coverage of broader food classification |
| [6] | Highly Accurate Food/Non-Food Image Classification Based on a Deep Convolutional Neural Network | Deep Convolutional Neural Network | 74% | Hokuto Kagaya and Kiyoharu Aizawa | Achieves high classification accuracy in food vs non-food distinction using CNN-NIN architecture | Limited application to diverse cuisines, focuses mainly on binary classification |
| [7] | FOOD-IMAGE CLASSIFICATION USING NEURAL NETWORK MODEL | Convolutional Neural Network | 90% | David J. Attokaren, Ian G. Fernandes, A. Sriram, Y.V. Srinivasa Murthy, and Shashidhar G. Koolagudi | Combines CNNs with high accuracy, robust model for food image classification | May struggle with highly complex food items, lacks integration with nutritional analysis |
| [8] | Analysis & Numerical Simulation of Indian Food Image Classification | Convolutional Neural Network | 96.27% | Anubhav Sharmand Chetan Kumar | Robust performance with edge-adaptive CNN segmentation and classification for South Indian cuisine | Limited dataset scope and generalization to non-South Indian cuisines |
| [9] | Indian Food Image Classification and Recognition with Transfer Learning Technique | convolutional neural networks | 95.3% | Jigar Patel and Kirit Modi | High accuracy achieved with MobileNetV3 using transfer learning and data augmentation | Performance heavily reliant on augmentation, lower accuracy without transfer learning |
| [10] | A New CNN-Based Single-Ingredient Classification Model and Its Application in Food Image Segmentation | custom-designed Convolutional Neural Network (CNN) | 92% | Ziyi Zhu and Ying Dai | Custom CNN achieves high performance on single-ingredient classification tasks | Less effective for complex multi-ingredient food images |
| [11] | Indian Food Image Recognition using a Deep | AlexNet | 96.6% | E. Emerson Nithiyaraj, S. Rajaseela | Effective performance with AlexNet, achieving high accuracy despite small dataset size | Restricted to limited categories, lacks scalability to larger datasets |
| [12] | Indian Food Image Classification | K-Nearest Neighbors (KNN) and Support Vector Machines (SVM) | 79.13% and 85% | Dr.Damala Dayakar Rao , Nanuri Pandu Ranga | Utilizes handcrafted features effectively for Indian food classification with good accuracy | Handcrafted features lack adaptability and scalability for diverse cuisines |
| [13] | ODNET: OPTIMIZED DENSENET FOR INDIAN FOOD CLASSIFICATION | MobileNetV3 and Optimized DenseNet | 98% and 98.39% | Jigar Patel , Hardik Talsania, Kirit Modi | DenseNet optimized for Indian food classification, achieving state-of-the-art accuracy | High computational requirements for dense architecture |
| [14] | IndianFoodNet | YOLO5, YOLO7, YOLO8 | 77.5% | Ritu Agarwal,  Tanupriya Choudhury,  Neelu J.Ahuja,  Tanmay Sarkar | YOLO variants achieve effective object detection for Indian food items | Limited application in broader contexts, precision decreases for complex dishes |
| [15] | Deep Indian Delicacy | Deep Convolutional Neural Network (DCNN) | 96.95% | Shamay Jahan , Shashi Rekha . H, Shah Ayub Quadri | High classification accuracy for Indian delicacies using a simple DCNN model | Limited dataset diversity restricts scalability |
| [16] | FOOD IMAGE RECOGNITION SYSTEM WITH MULTIPLE KERNEL LEARNING | SVM, MKL | 80.05% | Taichi Joutou and Keiji Yanai | Combines multiple kernel learning for diverse food image features, effective for Japanese food | Lower accuracy for general classification, focus on a single cuisine |
| [17] | Development of Korean Food Image Classification Model Using Public Food Image Dataset and Deep Learning Methods | InceptionResNetV2 | 60% | MINKI CHUN , HYEONHAK JEONG , HYUNMIN LEE, TAEWON YOO , AND HYUNGGU JUNG | Uses inception-based models to classify Korean foods, robust for distinct visual categories | Limited performance for visually similar dishes such as stews or soups |
| [18] | Food Classification of Indian Cuisines Using Handcrafted Features and Vision Transformer Network | ResNet | 90% | [Rahul Nijhawan](https://papers.ssrn.com/sol3/cf_dev/AbsByAuth.cfm?per_id=5038690)*,*[Ashita Batra](https://papers.ssrn.com/sol3/cf_dev/AbsByAuth.cfm?per_id=5038691)[Octavio Loyola-Gonz´alez](https://papers.ssrn.com/sol3/cf_dev/AbsByAuth.cfm?per_id=5038692), [Manoj Kumar](https://papers.ssrn.com/sol3/cf_dev/AbsByAuth.cfm?per_id=5007078), [Deepak Kumar Jain](https://papers.ssrn.com/sol3/cf_dev/AbsByAuth.cfm?per_id=5038693) Chongqing University | High accuracy using ResNet and handcrafted features, tailored for Indian cuisine | Handcrafted features may not generalize well for other cuisines, computational complexity |
| [19] | Food Detection and Recognition Using Convolutional Neural Network | Convolutional Neural Network (CNN) | 73.70% | Kagaya, Hokuto & Aizawa, Kiyoharu & Ogawa, Makoto | CNN outperforms traditional methods for food detection and recognition tasks | Lower accuracy for classification of complex or ambiguous food items |
| [20] | Food Image Classification Using Neural Networks | Convolutional Neural Networks (CNNs) | 85% | Alex M. Goh and Xiaoyu L. Yann | Demonstrates transfer learning's effectiveness in improving food classification performance | Dependent on pre-trained ImageNet weights, may struggle with uncleaned or noisy datasets |

# Methodology

To implement our solution, we have made use of the SWIN tiny transformer, which is a lightweight version of the Swin Transformer. The Swin transformer architecture uses a hierarchical vision transformer that divides the images into non-overlapping patches and then processes them using shifting windows for better extraction of spatial features.

**Dataset**

We have used the dataset **“Indian Food Images Dataset”** from Kaggle which is authored by “**IamSouravBannerjee**”. It comprises a total of 4000 Indian food images across 80 classes. The pictures are distributed uniformly among the classes, that would mean 50 images per each food class.

**Data Preprocessing**

For the data preprocessing part, two transform functions were initially defined for the train set and valid & test set. In train transform function, we added the following things –

* **Random Resizing and Cropping**: we had randomly resized and cropped the images to a fixed size of 224x224 pixels, which helps the model generalize better. The scale parameter was set to range from 0.8 to 1.0.
* **Random Horizontal Flip**: This step randomly flips the images horizontally with a probability of 50%. This has increased the diversity in the training data.
* **Random Rotation**: this step randomly rotates images up to 20 degrees. This was done to introduce some invariance.
* **Color Jittering**
* **Normalization**: The images were normalized using the mean and standard deviation values which makes them suitable to train using the Swin transformer model.

As for the validation and test transform, we added the following –

* **Resizing**: we resized the images to 256 pixels while maintaining the aspect ratio.
* **Center Cropping**: After resizing, a center crop of 224x224 pixels was applied to match the input size format.
* **Normalization**: Similar to the training set, the validation and test images were also normalized using the ImageNet mean and standard deviation.

We added some additional augmentation to the training data compared to the validation and test sets.

We used PyTorch's Image Folder Function to load the dataset and split it into the training, validation, and test sets using the random split method. The ratio of training set, validation set, and test set is 7:1:2. The resulting data loaders were configured with a batch size of 32.

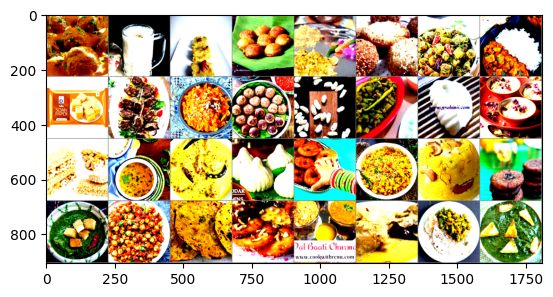


Fig: A Grid of images after the Data processing steps.

**Model Architecture**

As mentioned before, we used a pre-trained Swin transformer model on the dataset. To match our solution’s requirements, a classifier layer was added as a modification. This layer was used to match the number of output classes which is 80. Also, we added a dropout layer with dropout rate set to 0.3 to reduce overfitting and improve model training. We imported the model using the Hugging Face library and fine-tuned the model on our dataset.

**Training Parameters**

We set the following conditions to train the transformer model for our task.  
 The model was trained using the following parameters:

* **Epochs**: 50
* **Batch Size**: 32
* **Optimizer**: AdamW optimizer, with a learning rate of 1×10-4, and a weight decay of 0.01
* **Learning Rate Scheduler**: ReduceLROnPlateau scheduler with patience = 2, to adjust the learning rate based on the validation accuracy, reducing it by a factor of 0.5.
* **Loss Function**: Cross-Entropy Loss
* **Evaluation Metrics**: Both the training and validation accuracy was used to observe model performance.

# Results

**Training Phase:**

The training phase concluded with the model having **99.64%** training accuracy and a validation accuracy of **75.5%**. This was observed at **epoch 29**. The model was set to undergo 50 epochs in training, achieved the best model score at epoch 29 and then went on to trigger early stopping at **epoch 44**.

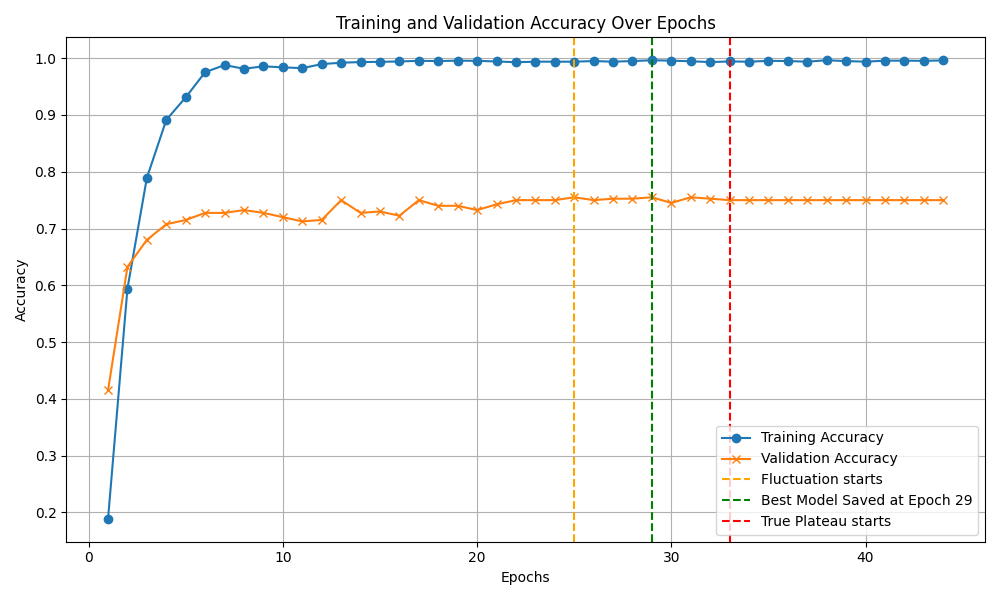


Fig: Model Training History

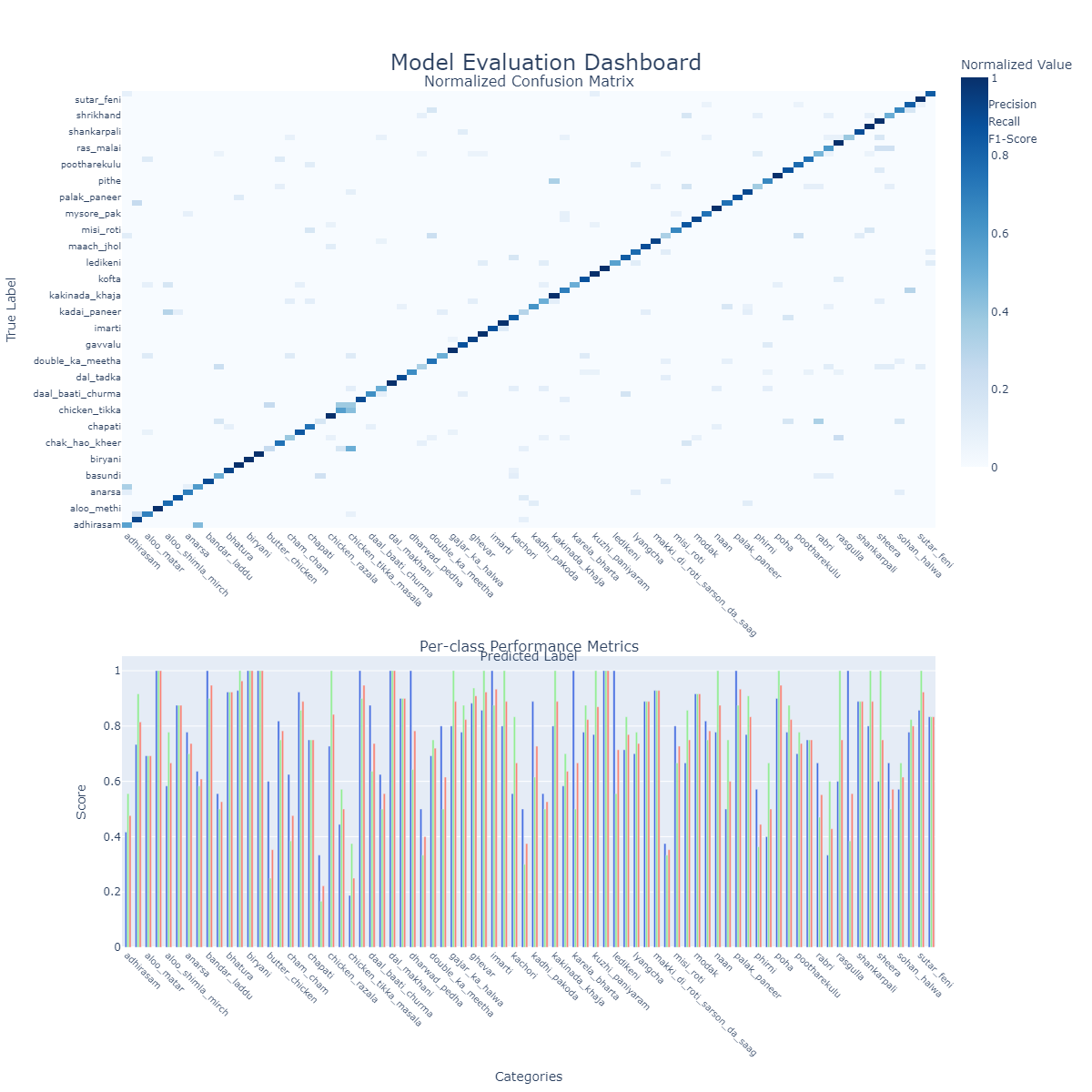
From this, we can observe that despite showing steady improvement in early epochs, the model training had reached a plateau around epoch 28-30, and after which validation accuracy showed no improvement causing the training to stop early at epoch 44.

**Testing Phase:**

Once the training phase completed, we moved on to the testing phase, where we used our model against the test dataset. The model had scored **75.38%** on test accuracy. The test dataset comprised 20% of the original dataset. We also created a classification report which contained the precision, recall, F1-score for each label.

| S.no | Categories | Precision | Recall | F1-Score | Support |
| --- | --- | --- | --- | --- | --- |
| 1. | adhirasam | 0.42 | 0.56 | 0.48 | 9 |
| 2. | aloo\_gobi | 0.73 | 0.92 | 0.81 | 12 |
| 3. | aloo\_matar | 0.69 | 0.69 | 0.69 | 13 |
| 4. | aloo\_methi | 1.00 | 1.00 | 1.00 | 9 |
| 5. | aloo\_shimla\_mirch | 0.58 | 0.78 | 0.67 | 9 |
| 6. | aloo\_tikki | 0.88 | 0.88 | 0.88 | 8 |
| 7. | anarsa | 0.78 | 0.70 | 0.74 | 10 |
| 8. | ariselu | 0.64 | 0.58 | 0.61 | 12 |
| 9. | bandar\_laddu | 1.00 | 0.90 | 0.95 | 10 |
| 10. | basundi | 0.56 | 0.50 | 0.53 | 10 |
| 11. | bhatura | 0.92 | 0.92 | 0.92 | 13 |
| 12. | bhindi\_masala | 0.93 | 1.00 | 0.96 | 13 |
| 13. | biryani | 1.00 | 1.00 | 1.00 | 13 |
| 14. | boondi | 1.00 | 1.00 | 1.00 | 10 |
| 15. | butter\_chicken | 0.60 | 0.25 | 0.35 | 12 |
| 16. | chak\_hao\_kheer | 0.82 | 0.75 | 0.78 | 12 |
| 17. | cham\_cham | 0.62 | 0.38 | 0.48 | 13 |
| 18. | chana\_masala | 0.92 | 0.86 | 0.89 | 14 |
| 19. | chapati | 0.75 | 0.75 | 0.75 | 12 |
| 20. | chhena\_kheeri | 0.33 | 0.17 | 0.22 | 6 |
| 21. | chicken\_razala | 0.73 | 1.00 | 0.84 | 8 |
| 22. | chicken\_tikka | 0.44 | 0.57 | 0.50 | 7 |
| 23. | chicken\_tikka\_masala | 0.19 | 0.38 | 0.25 | 8 |
| 24. | chikki | 1.00 | 0.90 | 0.95 | 10 |
| 25. | daal\_baati\_churma | 0.88 | 0.64 | 0.74 | 11 |
| 26. | daal\_puri | 0.62 | 0.50 | 0.56 | 10 |
| 27. | dal\_makhani | 1.00 | 1.00 | 1.00 | 9 |
| 28. | dal\_tadka | 0.90 | 0.90 | 0.90 | 10 |
| 29. | dharwad\_pedha | 1.00 | 0.64 | 0.78 | 14 |
| 30. | doodhpak | 0.50 | 0.33 | 0.40 | 9 |
| 31. | double\_ka\_meetha | 0.69 | 0.75 | 0.72 | 12 |
| 32. | dum\_aloo | 0.80 | 0.50 | 0.62 | 8 |
| 33. | gajar\_ka\_halwa | 0.80 | 1.00 | 0.89 | 8 |
| 34. | gavvalu | 0.78 | 0.88 | 0.82 | 8 |
| 35. | ghevar | 0.88 | 0.94 | 0.91 | 16 |
| 36. | gulab\_jamun | 0.86 | 1.00 | 0.92 | 12 |
| 37. | imarti | 1.00 | 0.88 | 0.93 | 8 |
| 38. | jalebi | 0.80 | 1.00 | 0.89 | 4 |
| 39. | kachori | 0.56 | 0.83 | 0.67 | 6 |
| 40. | kadai\_paneer | 0.50 | 0.30 | 0.38 | 10 |
| 41. | kadhi\_pakoda | 0.89 | 0.62 | 0.73 | 13 |
| 42. | kajjikaya | 0.56 | 0.50 | 0.53 | 10 |
| 43. | kakinada\_khaja | 0.80 | 1.00 | 0.89 | 8 |
| 44. | kalakand | 0.58 | 0.70 | 0.64 | 10 |
| 45. | karela\_bharta | 1.00 | 0.50 | 0.67 | 12 |
| 46. | kofta | 0.78 | 0.88 | 0.82 | 8 |
| 47. | kuzhi\_paniyaram | 0.77 | 1.00 | 0.87 | 10 |
| 48. | lassi | 1.00 | 1.00 | 1.00 | 10 |
| 49. | ledikeni | 1.00 | 0.56 | 0.71 | 9 |
| 50. | litti\_chokha | 0.71 | 0.83 | 0.77 | 6 |
| 51. | lyangcha | 0.70 | 0.78 | 0.74 | 9 |
| 52. | maach\_jhol | 0.89 | 0.89 | 0.89 | 9 |
| 53. | makki\_di\_roti\_sarson\_da\_saag | 0.93 | 0.93 | 0.93 | 14 |
| 54. | malapua | 0.38 | 0.33 | 0.35 | 9 |
| 55. | misi\_roti | 0.80 | 0.67 | 0.73 | 6 |
| 56. | misti\_doi | 0.67 | 0.86 | 0.75 | 14 |
| 57. | modak | 0.92 | 0.92 | 0.92 | 12 |
| 58. | mysore\_pak | 0.82 | 0.75 | 0.78 | 12 |
| 59. | naan | 0.78 | 1.00 | 0.88 | 7 |
| 60. | navrattan\_korma | 0.50 | 0.75 | 0.60 | 4 |
| 61. | palak\_paneer | 1.00 | 0.88 | 0.93 | 8 |
| 62. | paneer\_butter\_masala | 0.77 | 0.91 | 0.83 | 11 |
| 63. | phirni | 0.57 | 0.36 | 0.44 | 11 |
| 64. | pithe | 0.40 | 0.67 | 0.50 | 3 |
| 65. | poha | 0.90 | 1.00 | 0.95 | 9 |
| 66. | poornalu | 0.78 | 0.88 | 0.82 | 8 |
| 67. | pootharekulu | 0.70 | 0.78 | 0.74 | 9 |
| 68. | qubani\_ka\_meetha | 0.75 | 0.75 | 0.75 | 8 |
| 69. | rabri | 0.67 | 0.47 | 0.55 | 17 |
| 70. | ras\_malai | 0.33 | 0.60 | 0.43 | 5 |
| 71. | rasgulla | 0.60 | 1.00 | 0.75 | 9 |
| 72. | sandesh | 1.00 | 0.38 | 0.56 | 13 |
| 73. | shankarpali | 0.89 | 0.89 | 0.89 | 9 |
| 74. | sheer\_korma | 0.80 | 1.00 | 0.89 | 12 |
| 75. | sheera | 0.60 | 1.00 | 0.75 | 9 |
| 76. | shrikhand | 0.67 | 0.50 | 0.57 | 12 |
| 77. | sohan\_halwa | 0.57 | 0.67 | 0.62 | 6 |
| 78. | sohan\_papdi | 0.78 | 0.82 | 0.80 | 17 |
| 79. | sutar\_feni | 0.86 | 1.00 | 0.92 | 12 |
| 80. | unni\_appam | 0.83 | 0.83 | 0.83 | 12 |
|  | **Overall Accuracy** |  |  | 0.75 | 800 |
|  | **Macro Average** | 0.75 | 0.75 | 0.74 | 800 |
|  | **Weighted Average** | 0.77 | 0.75 | 0.75 | 800 |

**Observations on Test Results:**

****

Several factors contributed to the model's performance in the test phase:

· **Image Quality:** The quality of the test images varied, with some images being of lower resolution or poorly lit, which negatively impacted the model's performance.

· **Noisy Images:** A number of images in the test set contained noise such as faces or text, which was not removed during preprocessing. These could have interfered with the model's ability to classify the images correctly.

· **Similar Dishes**: Many Indian dishes share similar visual characteristics, which likely confused the model. For example, dishes like **"Aloo Gobi"** and **"Aloo Matar"** may appear visually similar, leading to misclassification.

· **Local Variants**: Some dishes in the dataset might represent local variants of a more common dish, which added ambiguity to the classification process.

**Deployment Using ModelBit**

After successfully training and testing the model, the next step was deployment. The model was deployed using **ModelBit** on free trial usage, which looked like an efficient way to serve the model as a REST API. This allowed for the integration of the trained model into a web-based application, enabling real-time predictions.

**Frontend Implementation:**

The frontend for the application was built using Streamlit, and deployed using Streamlit community cloud. The key features of the frontend include:

* **Image Upload:** Users can upload an image of Indian food, which is then passed to the backend for prediction.
* **Prediction Visualization:** The predicted label along with its confidence percentage is displayed after processing the image.

**Conclusion**

The review does not maximize academic distinction to clarify into the advancement of food image classification approaches; however, it emphasizes how computer vision and machine learning strategies resemble. Newer models, such as CNNs (Convolutional Neural Networks), SWIN Transformer and ResNet, or handcrafted feature approaches have gained state-of-the-art results in terms of recognition performance and robustness. Although CNNs set the groundwork with their basic methods of food recognition, more recent models such as SWIN Transformers and Vision Transformers (ViT) have interleaved advanced techniques that solve the complex visual diversity within differentiated cuisines like Indian food.

Despite these advances, however, pressing challenges remain. Current research is limited by restricted dataset variability, computational identification of ambiguous or similar dishes and inadequate incorporation of nutrition analysis. Although traditional feature extraction methods are mostly effective for targeted scenario processing, their generalization capability across cuisines is limited.

Future work could 1) establish rich datasets that are multifaceted from both perceptual and appearance aspects, 2) hybrid approaches that utilize the strengths of deep learning combined with deeper contextual methods. Linking classification systems with nutritional databases and real-world implementations offers opportunities to increase practical utility, especially with respect to key nutrition-related challenges such as malnutrition and overweight/obesity.

In the end, food image classification systems have to grow beyond cultures associated with nutritional cohorts and progress along these directions of accuracy, usability, and scalability towards a more inclusive user-friendly dietary management technology.

# 

# 

# 

# 

# References

[1] [A Chaitanya, Jayashree Shetty, Priyamvada Chiplunkar(2023) “Food Image Classification and Data Extraction Using Convolutional Neural Network and Web Crawlers”*Procedia Computer Science,* 218(Issue C):143-153](https://www.sciencedirect.com/science/article/pii/S1877050922025042?ref=pdf_download&fr=RR-2&rr=8dea60555f2a59d0)

[2] [Lu, Yuzhen. (2016). “Food Image Recognition by Using Convolutional Neural Networks (CNNs)." 10.48550/arXiv.1612.00983.](https://www.sciencedirect.com/science/article/pii/S1877050922025042?ref=pdf_download&fr=RR-2&rr=8dea60555f2a59d0)

[3] [Malina Jiang “Food Image Classification with Convolutional Neural Networks” *CS230: Deep Learning*, Fall 2019, Stanford University, CA](https://cs230.stanford.edu/projects_fall_2019/reports/26233496.pdf)

[4] ELPINA , GEDE PUTRA KUSUMA(2023) “REVOLUTIONIZING COMPUTER VISION: ENHANCED FOOD IMAGE CLASSIFICATION WITH SWIN TRANSFORMER AND SVM CLASSIFIER”*Journal of Theoretical and Applied Information Technology*,101(23)(December):7549-7561

[5] Wang H, Tian H, Ju R, Ma L, Yang L, Chen J and Liu F (2024) “Nutritional composition analysis in food images: an innovative Swin Transformer approach.” *Front. Nutr.11:1454466.* doi: 10.3389/fnut.2024.1454466

[6] Kagaya, H., Aizawa, K. (2015). “Highly Accurate Food/Non-Food Image Classification Based on a Deep Convolutional Neural Network.” In: Murino, V., Puppo, E., Sona, D., Cristani, M., Sansone, C. (eds) *New Trends in Image Analysis and Processing -- ICIAP 2015 Workshops.* ICIAP 2015. Lecture Notes in Computer Science(), vol 9281. Springer, Cham. <https://doi.org/10.1007/978-3-319-23222-5_43>

[7] D. J. Attokaren, I. G. Fernandes, A. Sriram, Y. V. S. Murthy and S. G. Koolagudi, “Food classification from images using convolutional neural networks,” *TENCON 2017 - 2017 IEEE Region 10 Conference*, Penang, Malaysia, 2017, pp. 2801-2806, doi: 10.1109/TENCON.2017.8228338.

[8] Anubhav Sharma, Chetan Kumar(2023) “Analysis & Numerical Simulation of Indian Food Image Classification Using Convolutional Neural Network” *International Journal of New Practices in Management and Engineering*,12(1):56-70

[9] Patel, Jigar, and Kirit Modi. (2023). “Indian Food Image Classification and Recognition with Transfer Learning Technique Using MobileNetV3 and Data Augmentation” *Engineering Proceedings* *56*, no. 1: 197. <https://doi.org/10.3390/ASEC2023-15341>

[10]Zhu, Ziyi, and Ying Dai. 2023. "A New CNN-Based Single-Ingredient Classification Model and Its Application in Food Image Segmentation" Journal of Imaging 9, no. 10: 205. https://doi.org/10.3390/jimaging9100205

[11]E. Emerson Nithiyaraj, S. Rajaseela(2021). “Indian Food Image Recognition using a Deep Learning Approach” *Indian Journal of Food Engineering*, 1(1)(December)

[12]Dr.Damala Dayakar Rao, Nanuri Pandu Ranga(2021). “Indian Food Image Classification Using K-Nearest-Neighbour and Support-Vector-Machines” *International Journal for Advanced Research In Science & Technology*, 9(06):201-208

[13]Jigar Patel, Hardik Talsania, Kirit Modi(2023). “ODNET: OPTIMIZED DENSENET FOR INDIAN FOOD CLASSIFICATION” *International Journal on Information Technologies & Security*,4(15):27-36

[14]Agarwal, Ritu & Choudhury, Tanupriya & J.Ahuja, Neelu & Sarkar, Tanmay. (2023). “IndianFoodNet: Detecting Indian Food Items Using Deep Learning.” *International Journal of Computational Methods and Experimental Measurements*, 11. 10.18280/ijcmem.110403.

[15] Shamay Jahan, Shashi Rekha. H, Shah Ayub Quadri(2018). “Deep Indian Delicacy: Classification of Indian Food Images using Convolutional Neural Networks” *International Journal for Research in Applied Science & Engineering Technology*, 6(III)(March):2653-2660

[16]Taichi Joutou and Keiji Yanai, “A food image recognition system with Multiple Kernel Learning,” 2009 16th IEEE International Conference on Image Processing (ICIP), Cairo, Egypt, 2009, pp. 285-288, doi: 10.1109/ICIP.2009.5413400.

[17]MINKI CHUN, HYEONHAK JEONG, HYUNMIN LEE, TAEWON YOO2, AND HYUNGGU JUNG(2022). “Development of Korean Food Image Classification Model Using Public Food Image Dataset and Deep Learning Methods” *IEEE Access,* vol. 10(December):128732-128741 doi:10.1109/ACCESS.2022.3227796

[18]Nijhawan, Rahul and Batra, Ashita and Loyola-Gonz´alez, Octavio and Kumar, Manoj and Jain, Deepak Kumar, “Food Classification of Indian Cuisines Using Handcrafted Features and Vision Transformer Network.” [http://dx.doi.org/10.2139/ssrn.4014907](https://dx.doi.org/10.2139/ssrn.4014907)

[19]Kagaya, Hokuto & Aizawa, Kiyoharu & Ogawa, Makoto. (2014). Food Detection and Recognition Using Convolutional Neural Network. 1085-1088. 10.1145/2647868.2654970.

[20]Goh, A. (2021). “Food-Image Classification Using Neural Network Model.” *International Journal of Electronics Engineering and Applications.*