Smart Ingredient Identifier: An AI-Powered Food Analysis System Using Deep Learning

Arnav Kumar
Chandigarh University
Mohali, Punjab, India.
arnavkkumar606@gmail.com

Ansh Singh
Chandigarh University
Mohali, Punjab, India.
anshmahajan2004@gmail.com

Dayal Chandra Sati
Chandigarh University
Mohali, Punjab, India.
dayal.e13263@cumail.in

***Abstract*—The perception of food images is not an easy task because of the differences in description, cooking styles and cultural diversity. In this paper, a Smart Ingredient Identifier is discussed, which runs a deep learning algorithm to identify dishes based on the food image and deduce information about the ingredients. The suggested system uses an EfficientNet-B0 system with a transfer learning model and is trained on a large scale dataset of 181 Indian and Western cuisine food categories. Two phases of training approach are used to enhance convergence and generalization. Experimental testing indicates that the model obtains a validation rate of 84.8 and a small generalization gap indicating that the model is efficient when employed in a large variety of foods. The real-time interaction is also facilitated by creation of a web-based application. The planned solution offers a viable basis on smart food analysis system.**

***Keywords*—Food image analysis, ingredient identification, deep learning, EfficientNet, transfer learning**

I. Introduction

Computer vision Food analysis has become a significant research field because it has a large number of applications in health care, nutrition management, and smart lifestyle systems. The exponential expansion of food order apps, online food journals, and health consumers has heightened the necessity of smart systems that can be able to interpret food images autonomously. Automated food recognition eliminates manual entry, decreases the effects of human error, and allows scalability to large masses of users.

Most of the food recognition systems that exist are highly based on dish-level recognition, and this level is not useful to a big extent in real life practical application. On the other hand, ingredient-level knowledge plays a significant role in practice in such applications as allergy prevention, calorie estimation, and personal diet recommendations. Being able to recognize food stuffs by use of pictures is not an easy task as the food products metamorphose so much as regards the visual image when preparing food. The mixing, frying, and garnishing are all linked to the concept of occlusion, the change of the texture and the color, and, therefore, ingredient identification happens to be much more complex than the normal object recognition.

Recent developments in deep learning and convolutional neural networks (CNNs) especially have enhanced image classification and have allowed automated food recognition systems. Nevertheless, most of the existing models are characterized by a low generalization per cuisine when trained using small or culturally biased data sets that are dominated by Western foods. This is a weakness that decreases their performance in areas that vary in culinary practices. To overcome these problems, this paper presents a Smart Ingredient Identifier that uses transfer learning and large-scale diverse datasets to obtain a robust food recognition both of Indian and Western cuisines. Through an integrated approach of effective neural architectures and trained strategies, the proposed system will provide effective, scalable, and realistic food image detection that can be used in the real world.

II. Literature Review

Food image analysis has also been widely studied in computer vision because it is significant to the field of nutrition, health, and intelligent food systems [1]. Visual representations made by visual representations such as color histograms, texture representations and shape representations were the first food recognition techniques. Although these techniques were quite successful with the conditions controlled, they were highly sensitive to variations in lighting conditions, perspectives and presentation of food and could not be effectively applied to real-life conditions [2]. The development of the deep learning system was a major transformation in the field and the convolutional neural networks (CNNs) flooded the market as the most orchestrated way of detecting food in images.

Deep CNN-based models trained on massive data sets such as Food-101 resulted in better accuracy of classification of dishes, especially when the architecture of an image such as AlexNet, VGGNet, and ResNet were used. In these developments, dish-level recognition was given much attention by most CNN-based systems and does not deliver much information on ingredient composition [3]. To overcome this limitation, some of the studies employed multi-label classification models and attention mechanisms in order to add information about ingredients or recipes [4]. Others leveraged on visual information together with textual recipe information to make ingredient prediction and the others attempted to localize discriminatory points in food image. However, they are often linked to the great complexity of computation, low scalability and real-time implementation problems [5].

More recent research has explored transformer-based architectures and attention-based architectures, including Vision Transformers and hybrid CNN transformer models, this is proven to be more successful because it can encode long-range information in images [4].

Nevertheless, such models typically require rather significant data, and massive computational power, hence they are not very feasible in resource-constrained contexts [5]. Also, bias in the data sets is one of the greatest flaws of the existing literature as most of the publicly available food datasets are very biased towards western food. The trained models based on these datasets hence have low levels of generalization to culturally dissimilar foods, such as Indian food [6]. Even though lightweight and mobile friendly models have been proposed in order to overcome the limitation of deployment, they would be of lower accuracy. In its turn, the offered system addresses these concerns by training on a large and diverse collection of Indian and Western types of food and applying EfficientNet-B0 with a systematic use of transfer learning, which guarantees a sensible trade-off between accuracy, efficiency, and feasibility[7].

III. PROPOSED SYSTEM ARCHITECTURE

It is a prototype end to end smart system of food analysis that tried to combine deep learning based image recognition with a user-friendly web interface. It is architecture that is worried about precision, real time scalability and usability. The entire system is further subdivided into a series of inter-related modules with each having the mandate to execute a certain task within the food analysis path [8].

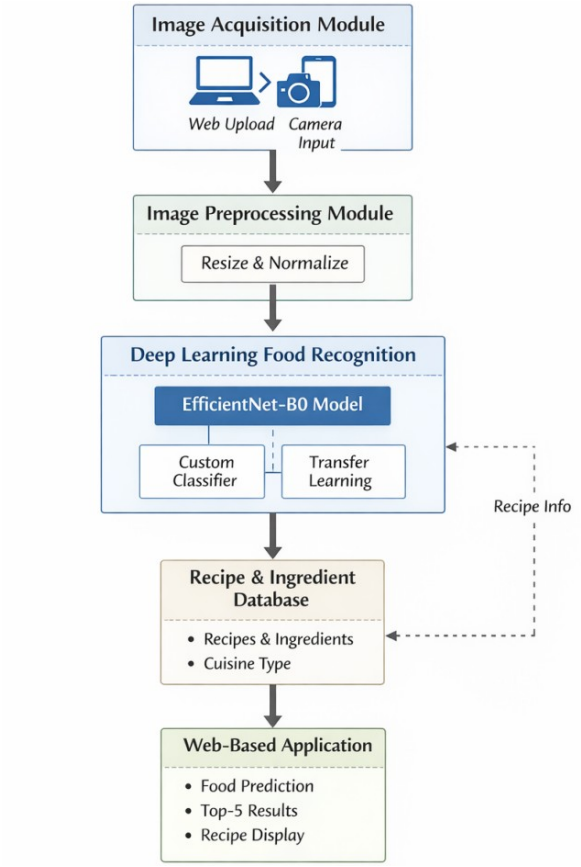

Fig. 1. Flowchart of the proposed Smart Ingredient Identifier system.

A. System Overview

In this system the parts that make up the system follows the modular architecture where parts are .

- Image acquisition module
- Image preprocessing module
- Deep learning food recognition module.
- Mapping recipe and ingredient of modules.
- Visualization and interaction module in web based.

An input of the user, represented by an image of a food, and the calculation of the predicted food, typically triggers the workflow in the form of ingredients and cooking instructions [9].

B. Image Acquisition Module

Image acquisition module is an access point of the system. The images of the foods are posted using web based interface that was done under Gradio framework. The system can contain a conventional photograph like JPEG and PNG. The picture uploaded is validated to know that it is a compatible picture and sent to preprocessing module [10].

The module is easily customizable with real-time interaction with the users, and the camera-based image capture can be facilitated with references to the mobile or Internet-of-Things user devices [11].

C. Image Preprocessing Module.

- The raw food image is then subjected to a series of preprocessings in order to get a regular input into the deep learning model:
- Downsampling the image to 224 x 224 pixels resolution.
- ImageNet standard deviation ImageNet mean and standard deviation.
- Image data to PyTorch models conversion.

During training, other data augmentation methods are used which include random cropping, flipping and rotating among others. Inference makes use of deterministic preprocessing; in this case, the predictions must remain unchanged [12].

D. Deep Learning Food Recognition Module.

This is the main element in the suggested system. It is charged with the detection of the visual part of the picture that is fed into it and the identification of the food [13].

In this case, EfficientNet-B0 was used to extract features. EfficientNet-B0 has created the backbone network because it has a balanced trade-off between the accuracy and computational efficiency. The scaling in this model is the compound scaling that is to optimize the network depth, width and resolution in a combination. ImageNet weights are already trained into the model, and it allows the model to converge much quicker and gain superior features of learning [14].

1) Custom Classification Head

- Full connection layer comes down to the dimension of features

- ReLU activation introduces non-linearity
- The training is stabilized by the usage of the batch normalization
- The dropout will remove overfitting
- The last linear layer generates 181 food classes scores of the classes

2) Transfer Learning Strategy

- The system can transfer learning strategy is two phase:
- Warmup Phase: Warmup does not train backbone layers, it just trains classifier layers.
- Fine-Tuning Phase: The EfficientNet blocks that were left unfrozen are now trained together with the classifier.
- This makes the pattern of the model to suit the general visual peculiarities and adapt to the food-specific ones.

E. Recipe and Ingredient Mapping Module.

When the food type has been predicted, the type of food is compared with the dish that has been identified after which a structured database of recipes is compared to the system. There are all types of food associated with [15]:

- Ingredient list
- Cooking recipes step-by-step.
- Type of food (Indian or western)

Using this mapping, the opportunity to find out the indirect ingredients in the context of learned food patterns is available, which can be useful both in the practice and classification [16].

F. Web Based Application Module.

The web application is created to show the way in which the proposed system can be used in practice. The application provides:

- Image upload functionality
- Showing the expected name of food.
- Bars of best-5 scores.
- The recipe and its ingredients and preparation.
- Classification of output cuisine.

It has a simple interface that is easy to use and responsive hence compatible with both desktop and mobile applications [17].

G. Workflow Description This workflow description provides the workflow of G. System. Workflow description This workflow description incorporates the workflow of G. System [18].

The entire system workflow process is as follows:

- The web interface gives the user an option to post an image of a food.
- Image made normal and processed.
- It is modeled on the basis of the EfficientNet and takes in processed image as input.
- There is a model that forecasts the type of food.
- Similar to ingredients and information on the recipe are memorized.
- The user is presented with the results in real time.

H. Design Considerations

The following considerations are made by the system architecture:

- Scalability: Unlike other classifiers, the retraining of the classifier is necessary to add new categories of food.
- Engineering: EfficientNet-B0 can be used to make fast inferences on very little hardware.
- Generality: Imprints on various food enhance nutritional value.
- Deployability: The design is a web based design, and thus, easy to deploy.

IV. METHODOLOGY

This part provides a description of the entire methodology that was involved in the design, training and implementation of the proposed Smart Ingredient Identifier. It has a methodology that is founded on diversity of data, effective features learning, and effective generalization to other cuisines [19].

A. Data Preparation and Data Collection.

The small dataset and the weakness of cuisine bias were overridden by a large and heterogeneous dataset on the access to a multitude of food pictures. The dataset includes [20]:

- Indian Food Images Data 80 most popular dishes that are eaten in India.
- Food-101 List of 101 popular western foods.

The whole dataset (including integration) consists of 181 categories of foods. The classes were held through stratification of the images into training, validation and testing groups.

- Training set: 113,900 images
- Validation set: 20,760 images
- Test set: Withdrawn to final test.

The size of the classes is about 629 images on average and this assists in representing data to a great extent compared to the small scale experiments that have been utilized in the past.

B. Data Preprocessing

Raw foods pictures differ in regards to the level of resolutions, the background, and the light. The following preprocessing steps were used to maintain and ensure the effectiveness of learning [21]:

- Image resizing to 224 × 224 pixels
- Conversion to tensor format
- ImageNet normalization values in mean and standard deviation of pixel values.

In the model training, data augmentation was used to improve generalization and minimize overfitting:

- Random resized cropping
- Horizontal flipping
- Rotation within ±15 degrees
- The effect of jittering of color (change of brightness and contrast).

The sole use of the deterministic preprocessing was done to determine stability of prediction so that it could be possible to carry out inference and validation[22].

C. Model Selection

It was decided that EfficientNet-B0 would be used as the backbone architecture based on the fact that it was possible to obtain a high accuracy rate with fewer parameters. The scaling scheme applied to the EfficientNet is in the fact that the depth, the width and the resolution of the input are balanced and scaled in balance (this can be appropriate in the resource-constrained environment) [23].

To allow the model to use the learnt low-level and middle-level visual representations including edges, textures, and shapes, the backbone network had been trained to apply pre-trained ImageNet weights [25].

D. Network Architecture Design.

EfficientNet-B0 model takes the base model and produces the high-level feature vector which is forwarded through a classification head specially designed. The classifier consists of:

- The layer to have a deep-connecting layer to have smaller features.
- Non-linearity induced by the activation of the Rectified Linear Unit (ReLU).
- Normalization of learning by stabilization.
- A dropout is employed in order to reduce overfitting.
- The output neuron with 181 neurons that represent the type of food.

Through the architecture, the system is bought to achieve discriminatory food-specific representations without sacrificing the computational efficiency of the system [26].

E. Transfer Learning Strategy.

Two phase transfer learning strategy was also taken to achieve the best training performance.

1) Phase I – Warmup Training

The entire EfficientNet architecture was frozen in the preliminary stage and the head of the classifier was trained only. The stage enables the newly added layers to become acquainted with the task of food classification although it does not give rise to any form of perturbation to the already trained feature representations [27].

- Duration: 3 epochs
- Learning rate: 0.001

2) Phase II – Fine-Tuning

The third stage involved the unfreezing of the remaining three EfficientNet-B0 blocks and to a certain degree fine-tuning could occur. This enables the network to obtain higher level properties that are associated with food image properties.

- Duration: 22 epochs
- Learning rate: 0.0001
- Cosine annealing Cosine annealing learning rate.

Convergence is greatly enhanced and catastrophic forgetting is not care-taken of by this stage methodology [28].

TRAINING HYPERPARAMETER	CONFIGURATION
Optimizer	AdamW
Loss Function	cross-entrpy loss
Batch Size	32
Weight Decay	0.01
Dropout Rate	0.25
Total Epouchs	25

TABLE I. TRAINING HYPERPARAMETERS:

G. Metrics of Performance Evaluation

Standard classification metrics: Standard classification metrics were used to measure model performance:

- Precision: Overview accuracy of predictions.
- Separation between training and validation: Generalization measure.
- Top-5 Acuracy: Making prediction as a priority..

These would provide a clue of predictive performance, and also the strength of the model.

H. Implementation and Deployment

The whole system was done in PyTorch. The training was performed on an NVIDIA RTX 3050 that can run CUDA, which implies and explains that the training efficiency is much quicker compared to its execution on the CPU.

The most successful weights of the model were saved following the training and they were included in a web based application developed using Gradio. The level of this implementation can be used to interact with users with the model in real time and receive food predictions and ingredient and recipe information [29].

V. IMPLEMENTATION, EXPERIMENTAL RESULTS AND ANALYSIS

A. Implementation

1. Model Training Setup

The model was developed and trained using the following configuration:

PARAMETERS	VALUE
Framework	Pytorch
Optimizer	AdamW
Batch Size	32
Epochs	25
Loss Function	Cross-Entropy
Hardware	NVIDIA RTX3050 (4GB)

TABLE II. *IMPLEMENTATION AND TRAINING CONFIGURATION*

Training was performed using GPU acceleration, reducing training time to approximately 8.5 hours for the final model.

B. Model Performance

The final trained model achieved:

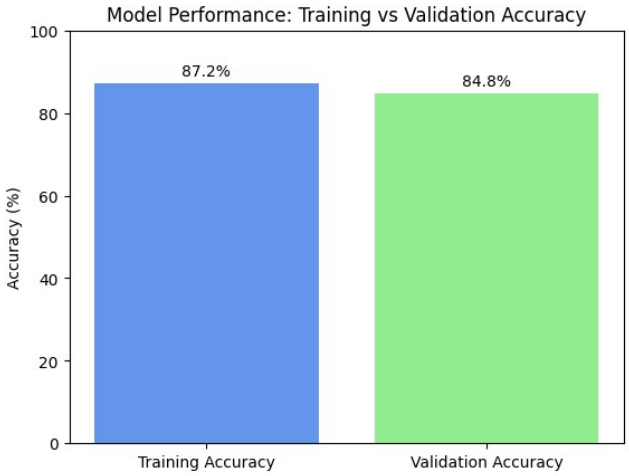

Fig. 2. Training vs validation accuracy.

The small gap indicates excellent generalization and minimal overfitting.

C. Model Comparison

Model Version	**Dataset Size**	**Accuracy**	**Overfitting Gap**
V1	3,150 images	65.5%	+32%
V2	3,150 images	45.6%	−13%
V3	3,150 images	63.6%	+15.5%
Final Model	113,900 images	84.8%	+2.5%

TABLE III. *MODEL COMPARISION RESULTS*

The results clearly show that dataset size plays a crucial role in reducing overfitting and improving accuracy.

D. Application Interface

A real-time web-based application was developed using Gradio to demonstrate practical usability of the trained model.

The system allows users to:

- Upload food images
- View top-5 predicted dish classes with confidence scores
- Retrieve ingredient details
- Access cooking instructions
- Identify cuisine type (Indian or Western)

This deployment validates the model's effectiveness in real-world food analysis beyond offline evaluation.

VI. CONCLUSION & FUTURE WORK

In this paper, the author will introduce Smart Ingredient Identifier, the AI-based food analysis system that can identify dishes based on their pictures and provide information about ingredients based on deep learning. The system has overcome major shortcomings of current food recognition techniques especially low cuisine-to-cuisine generalization and overfitting with small dataset. The model was trained on a varied dataset of 181 food categories including Indian and Western foods that were trained using an EfficientNet-B0 architecture and a two-phase transfer learning strategy.

The experimental findings indicate the strong generalization of 84.8 validation accuracy with a small 2.5% training -validation gap. Investigations involving comparative analysis ensured that bigger and more heterogeneous datasets make a huge impact on overfitting and performance.

To illustrate an actual application, a Gradio web application in real time was created, which enables users to post images of food and obtain the prediction of dishes, their ingredients, and cooking instructions. The future work entails ingredient level segmentation, nutritional estimations, mobile/edge implementation, and implementation with smart kitchen and IoT systems to have intelligent food management.

VII. REFERENCES

[1]. R. L. Bossard, M. Guillaumin, and L. Van Gool, "Food-101: Mining Discriminative Components with Random Forests," in Proc. European Conf. Computer Vision (ECCV), Zurich, Switzerland, 2014, pp. 446–461.

[2]. ETH Zurich, "Food-101 Dataset," 2014.

[3]. M. Tan and Q. V. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," in Proc. Int. Conf. Machine Learning (ICML), 2019, pp. 6105–6114.

[4]. A. Dosovitskiy et al., “An Image Is Worth 16×16 Words: Transformers for Image Recognition at Scale,” in Proc. Int. Conf. Learning Representations (ICLR), 2021.
[5]. A. Salvador et al., “Inverse Cooking: Recipe Generation From Food Images,” in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2019, pp. 10453–10462.
[6]. A. Salvador, M. Hynes, Y. Aytar, J. Marin, F. Ofli, and A. Torralba, “Learning Cross-Modal Embeddings for Cooking Recipes and Food Images,” IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 43, no. 1, pp. 291–307, Jan. 2021.
[7]. J. Marin et al., “Recipe1M+: A Dataset for Learning Cross-Modal Embeddings,” arXiv preprint, 2018.
[8]. J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You Only Look Once: Unified, Real-Time Object Detection,” in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2016, pp. 779–788.
[9]. I. Loshchilov and F. Hutter, “Decoupled Weight Decay Regularization,” in Proc. Int. Conf. Learning Representations (ICLR), 2019.
[10]. I. Loshchilov and F. Hutter, “SGDR: Stochastic Gradient Descent with Warm Restarts,” arXiv preprint, 2016.
[11]. M. Lata et al., “AI and machine learning in cybersecurity: Practices, opportunities and challenges,” EDPACS, pp. 1–13, 2025.
[12]. J. Hu, L. Shen, and G. Sun, “Squeeze-and-Excitation Networks,” in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2018, pp. 7132–7141.
[13]. M. Sandler et al., “MobileNetV2: Inverted Residuals and Linear Bottlenecks,” in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2018, pp. 4510–4520.
[14]. J. Deng et al., “ImageNet: A Large-Scale Hierarchical Image Database,” in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2009, pp. 248–255.
[15]. W. Min, S. Jiang, L. Liu, Y. Rui, and R. Jain, “A Survey on Food Computing,” ACM Computing Surveys, vol. 52, no. 5, pp. 1–36, 2019.
[16]. M. Bolaños, A. García-García, and P. Radeva, “Food Ingredients Recognition Through Multi-Label Learning,” arXiv preprint, 2017.
[17]. Z. Zhu, “A CNN-Based Single-Ingredient Food Image Classification Model,” Sensors, vol. 23, no. 18, 2023.
[18]. J. Li, J. Sun, Z. Li, and Y. Rui, “Predicting Relative Ingredient Amounts from Food Images,” arXiv preprint, 2020.
[19]. PyTorch Team, “PyTorch: An Imperative Style, High-Performance Deep Learning Library,” 2020.
[20]. PyTorch Documentation, “CUDA Semantics,” 2024.
[21]. NVIDIA Corporation, “CUDA Toolkit Documentation,” Version 12.4, 2024.
[22]. Gradio Team, “Gradio: Build Machine Learning Web Apps,” 2024.
[23]. OpenCV Team, “Open Source Computer Vision Library,” 2024.
[24]. Kaggle, “Indian Food Images Dataset,” 2020.
[25]. K. Simonyan and A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition,” arXiv preprint, 2014.
[26]. K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778.
[27]. S. Minaee et al., “Image Classification Using Deep Learning: A Survey,” IEEE Access, vol. 9, pp. 793–806, 2021.
[28]. C. Shorten and T. M. Khoshgoftaar, “A Survey on Image Data Augmentation for Deep Learning,” Journal of Big Data, vol. 6, no. 60, 2019.
[29]. W. Rawat and Z. Wang, “Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review,” Neural Computation, vol. 29, no. 9, pp. 2352–2449, 2017.