




DSO 569: FINAL PROJECT REPORT

INDIAN FOOD CLASSIFICATION SYSTEM

GROUP 8: LAVANYA DESHMUKH, NITHYA ANANTHA PADMANABHA GORUR, OMKAR VYAS, PARAMJEET SINGH SAHRAWAT, TARUNI SUNDER

Indian cuisine offers a diverse range of traditional dishes, each with unique flavors and regional influences. By leveraging Convolutional Neural Networks (CNNs) and transfer learning, we have developed a classifier capable of accurately identifying and categorizing these dishes from images. The significance of this project lies in its potential to enhance various applications, including travelers applications and nutrition applications, and can be extended to include recipe recommendation systems. Automating the identification of Indian dishes can provide users with nutritional information, culinary inspiration, and cultural insights, promoting a deeper appreciation of Indian cuisine. In the following sections, we will discuss the methodology, dataset, model architecture, training procedure, and evaluation metrics of the food classifier. We will also address potential future directions for improvement.

1. Prerequisite Knowledge - CNNs & Transfer Learning

1. 1 Convolutional Neural Networks (CNNs)

CNNs are a class of deep neural networks specifically designed for analyzing visual data, making them particularly powerful in image classification. CNNs utilize convolutional layers to extract spatial patterns and features from input images through the application of filters known as kernels. These layers detect edges, textures, and other visual patterns, enhancing the model's expressive power via activation functions. Pooling layers complement convolutional layers by reducing feature map dimensions while retaining crucial information. This combination enables CNNs to learn hierarchical representations, capturing both low-level features like edges and high-level features such as shapes and objects.

1.2 Transfer Learning

Transfer Learning leverages pre-trained CNN models trained on large datasets, such as MobileNet, ResNet, and InceptionV3, and adapts them to new tasks or datasets with limited labeled data. By fine-tuning pre-trained models on new data, transfer learning allows for faster convergence and better generalization performance, making it particularly useful in scenarios where collecting large datasets is impractical or costly.

2. Dataset and Processing

2.1 Data

The dataset consists of 5855 images obtained from publicly available online datasets. The data is partitioned into 4020 training images, 1250 validation images, and 585 test images. Each image represents an ethnic Indian food dish across 20 different categories.

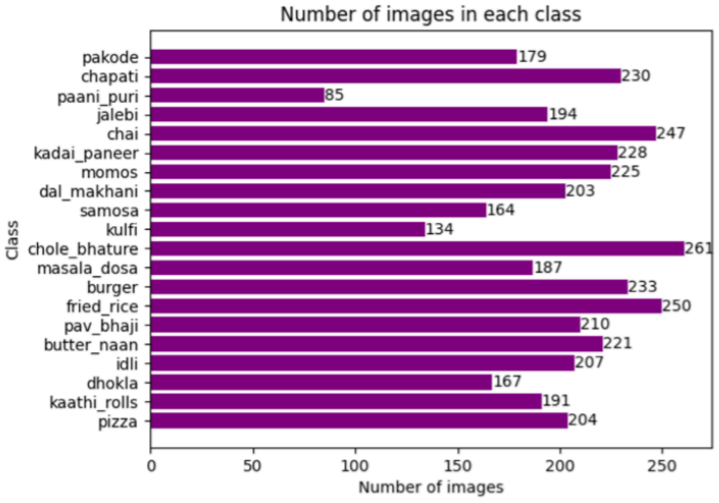


Image 2: Dataset

2.2 Data Augmentation

Several data augmentation techniques to handle diversity, scarcity and invariant features of data have been used as food images can be taken in different sets of environments and settings which enhance the robustness and generalization capability of models. The below methods were implemented :

- Rescaling: Image pixels were normalized to a range [0,1] to facilitate faster train convergence.
- Brightness Range: Random adjustments to the brightness of images have been included within the specified range of [0.8, 1.2], allowing the model to learn under various lighting conditions.
- Zoom Range: Images were randomly zoomed in or out by a factor of 0.2, introducing variability in the scale of objects to enable the model to learn to recognize food items at varying distances.
- Horizontal Flipping: Food presentations can vary, with dishes arranged differently. This augments the dataset with mirrored images, allowing the model to learn different orientations.
- Rotation: Rotation with a range of ±40 degrees was implemented to simulate the diverse viewpoints and angles as different people can capture food images from different orientations.

3. Model Architecture

The following explains the deep learning models that were used for the food image classification system. A base Convolutional Neural Network (CNN) model was designed to set standards for building and improving further models on classification accuracy. Subsequently, pre-trained transfer learning models ResNet50 and Inception V3 were used for their powerful architecture and performance in image classification problems.

3.1 Base Model

The model serves as a benchmark for the models used and is a standard CNN image classification architecture. The architecture contains 3 convolutional layers with 32, 64, and 128 filters respectively and each convolution layer is followed by a max-pooling layer with a 2x2 filter. Each convolutional layer incorporates Rectified Linear Unit (ReLU) activation functions, allowing the model to capture complex patterns present in the data. The max-pooling layers simplify the data representation. The model architecture further contains a flattening layer that connects to dense (fully connected) layers with 128 neurons for the model to learn higher-level features and make predictions. The activation function for the softmax activation generates class probabilities, determining the most likely class out of the 20 classes for each input image. An enhanced version of the base model architecture including a 20% dropout rate and L2 regularization was also built but underperformed as compared to the base model, possibly due to over-regularization.

For training the model, Adam optimizer, categorical cross-entropy loss function, and accuracy metric are employed for parameter optimization and model evaluation across 10 epochs. Model Checkpoint callback has been utilized to ensure that the best-performing model weights are preserved and this benchmark model serves as a reference point for evaluating the efficacy of alternative architectures.

3.2 Transfer Learning Models

Two pre-trained model architectures have been utilized for the classification of the food images – ResNet50 and Inception V3. The pre-trained layers on ImageNet are frozen to prevent them from being updated during training. A custom classification on top of the model's feature extraction layers is added including a flatten layer to organize the features, followed by a dense (fully connected) layer with 128 neurons that helps the model learn more complex patterns, and finally, a dense layer (output) with 20 (number of categories) neurons is added.
For training of the model, Adam optimizer, categorical cross-entropy loss function, and accuracy metric are employed for parameter optimization and model evaluation across 10 epochs. Model Checkpoint callback has been utilized to ensure that the best-performing model weights are preserved.

3.2.1 ResNet50

ResNet50 is a 50-layer CNN (48 convolutional layers, 1 MaxPool layer, and 1 average pool layer) primarily used in image classification. A bottleneck residual block uses 1x1 convolutions, which reduces the number of parameters and matrix multiplications. It employs residual connections to mitigate the vanishing gradient problem, facilitating deeper networks. This enables faster training of each layer.

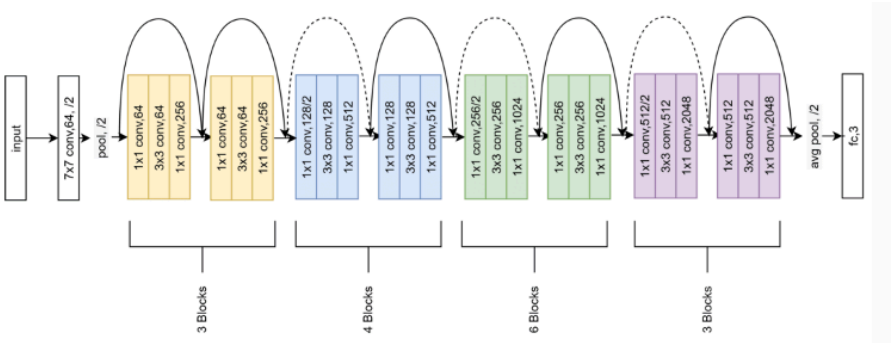


Image 3: ResNet50 Architecture

Reasons to Select	**Limitations**
Architectures capture intricate features, perform well on complex image classification	Might suffer from overfitting, especially when training data is limited
Common in image-related tasks; reliable	Can be computationally expensive

3.2.2 Inception V3

Inception V3 is a 48-layer CNN designed for image classification. Inception V3 utilizes "inception modules" composed of multiple parallel convolutional layers with different filter sizes, allowing the network to capture features at different scales simultaneously. It also incorporates factorized convolutions, batch normalization, and global average pooling to improve training efficiency and performance.

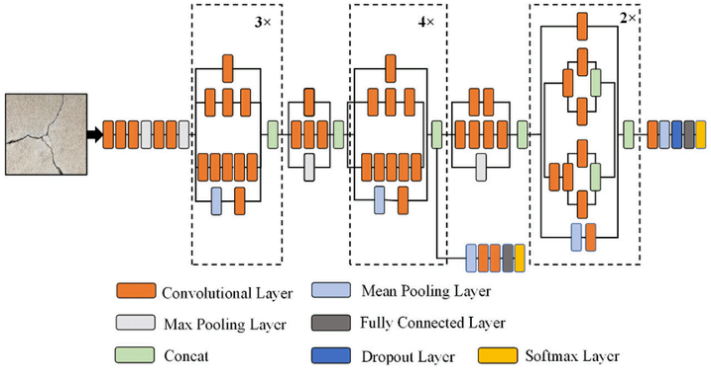


Image 4: Inception V3 Architecture

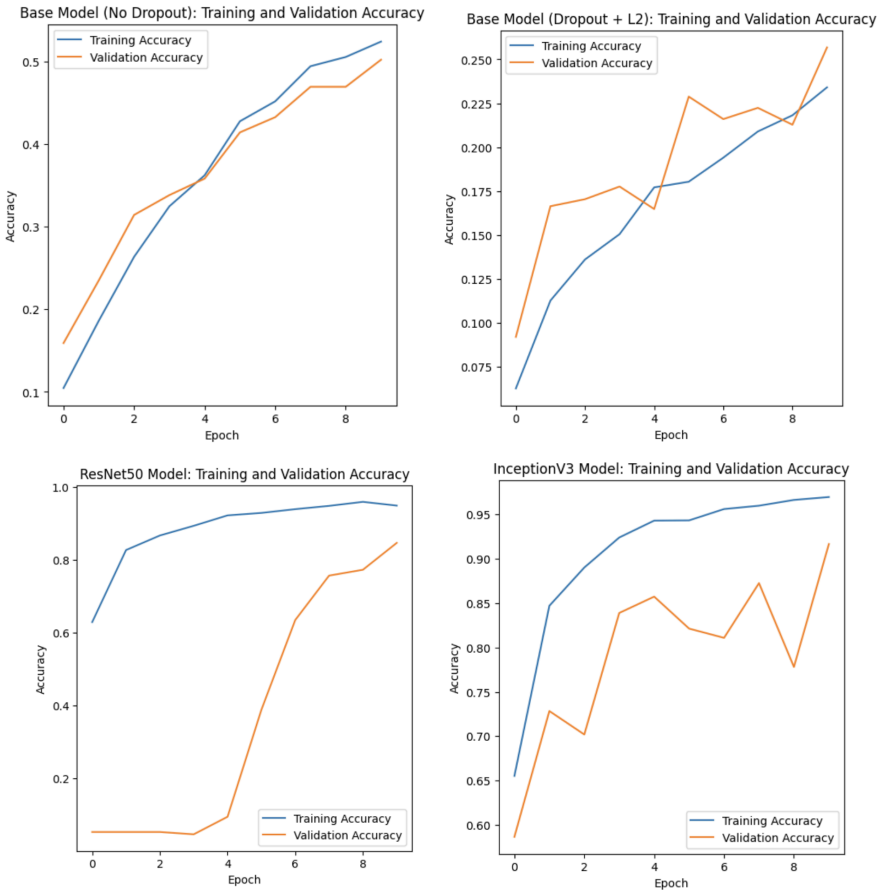
Reasons to Select	**Limitations**
High accuracy; outperforms other architecture	Can be computationally expensive
Fine balance between speed and accuracy	Sensitivity to hyperparameters; might require careful tuning for optimal performance

4. Results and Future Enhancements

4.1 Model Evaluations

In evaluating our models' performance, we compared a base CNN model (with/without dropout and regularization), ResNet50, and Inception V3. While the basic CNN was a starting point, ResNet50 showed improvement in accuracy due to its deeper architecture. However, Inception V3 outperformed both models, highlighting the significance of its detailed design in achieving better results. Multiple parallel convolutional layers with different filter sizes in the inception modules and a large number of parameters in Inception V3 allow for capturing features at various scales well. Overall comparisons among the models are provided below.

Model	Train Accuracy	Validation Accuracy	Test Accuracy
Base Model (without dropout)	0.52	0.50	0.48
Base Model (with dropout and L2)	0.23	0.26	0.23
ResNet50 Model	0.95	0.85	0.85
Inception V3 Model	0.97	0.92	0.87



4.2 Novelty Factor: Real-World Food Images

In our pursuit of creating a robust and practical classification model, we went beyond traditional testing methodologies by incorporating a personal gallery of real-life food images. These images, generously contributed by friends and classmates, represent diverse culinary experiences captured in everyday settings. The results from this prediction with the Inception V3 models are included in the last section of the code notebook.

4.3 Future Enhancements

4.3.1 Expansion of Dataset and Multicultural Classification

Expand the dataset to include global cuisines, enabling users to identify dishes from diverse culinary traditions. Additionally, recognize regional variations within specific cuisines for more nuanced insights.

4.3.2 Continuous Model Refinement and Performance Optimization

Continuously refine and optimize the image classification models through ongoing training and performance evaluation. Regularly update the dataset, refine data augmentation, and experiment with advanced model architectures to enhance accuracy and robustness.

5. Possible Applications

5.1 Integration Into a Travel App:

This enhancement will allow users to identify unfamiliar dishes while traveling to different countries. Users can discover the names of dishes from various cuisines worldwide, enhancing their culinary experiences.

5.2 Integration into Health & Fitness Apps

Users may opt for dishes with lower calorie counts or adjust their portion sizes to achieve a more balanced and nutritious meal, promoting overall health and well-being.

5.3 Improved Recipe Recommendations

By incorporating this image classification functionality into the recipe app, users can take a photo of the dish they want to cook or have on hand and get personalized recipe recommendations based on the identified dish.

6. Group Contribution

Our project was a team effort. Lavanya took charge of the initial steps, collecting the data, and ensuring its diversity through various data augmentation techniques. She added a unique touch by gathering and using the model to classify real-life food images from friends and family, enhancing the novelty factor of our project. Omkar and Nithya then built the architecture of the two base CNN models, trained and evaluated them, which served as the reference point for other models. Taruni and Paramjeet explored transfer learning models - ResNet50 and Inception V3 architectures, training and evaluating them.

However, it was our brainstorming sessions that truly set our project apart. Together, we explored future enhancements and potential applications, drawing on unique insights. This collaborative effort ensured that our project was not only comprehensive but also had strong applications in the real world.