DEEP-LEARNING STRATEGIES FOR FOOD IMAGE CLASSIFICATION INVOLVING FINE-TUNING AND DEEP FEATURE EXTRACTION

A MAJOR PROJECT-II REPORT

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE AWARD OF THE DEGREE
OF

MASTER OF TECHNOLOGY
IN
INFORMATION SYSTEMS

Submitted by:
PRANJAL KUMAR SINGH
2K22/ISY/12

Under the supervision of
PROF. SEBA SUSAN

DEPARTMENT OF INFORMATION TECHNOLOGY
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

May, 2024

DELHI TECHNOLOGICAL UNIVERSITY
(FORMERLY DELHI COLLEGE OF ENGINEERING)
Bawana Road, Delhi-110042

CANDIDATE'S DECLARATION

I, Pranjal Kumar Singh, 2K22/ISY/12 student of M.Tech in Information Systems, hereby declare that the Major Project-II dissertation titled "**DEEP-LEARNING STRATEGIES FOR FOOD IMAGE CLASSIFICATION INVOLVING FINE-TUNING AND DEEP FEATURE EXTRACTION**" which is submitted by me to the Department of Information Technology, Delhi Technological University, Delhi in partial fulfillment of the requirement for the award of the degree of Master of Technology, is original and not copied from any source without proper citation. This work has not previously formed the basis for the award of any degree, Diploma Associateship, Fellowship, or other similar title or recognition.

Place: Delhi
Date: 21/05/2024

PRANJAL KUMAR SINGH
2K22/ISY/12

DEPARTMENT OF INFORMATION TECHNOLOGY
DELHI TECHNOLOGICAL UNIVERSITY
(FORMERLY DELHI COLLEGE OF ENGINEERING)
Bawana Road, Delhi-10042

CERTIFICATE

I hereby certify that the Major Project-II dissertation titled "**DEEP-LEARNING STRATEGIES FOR FOOD IMAGE CLASSIFICATION INVOLVING FINE-TUNING AND DEEP FEATURE EXTRACTION**" which is submitted by Pranjal Kumar Singh, 2K22/ISY/12, Department of Information Technology, Delhi Technological University, Delhi in partial fulfillment of the requirement for the award of the degree of Master of Technology, is a record of the project work carried out by the student under my supervision. To the best of my knowledge, this work has not been submitted inpart or full for any degree or diploma to this University or elsewhere.

Place: Delhi
Date: 21/05/2023

Prof. SEBA SUSAN
SUPERVISOR
Professor
Department of Information Technology
DELHI TECHNOLOGICAL UNIVERSITY

ABSTRACT

With the exponential proliferation of food-related material on digital platforms, automatic food picture categorization has emerged as a critical study field. Deep learning models such as EfficientNetB0, Xception, and Inception-v3, which are known for their ability to use transfer learning, have become crucial tools in this sector. In this thesis, we critically evaluate the performance of such models on the complex Food-101 dataset that encompasses 101 various types of food. Our study found out that Xception is leading in its performance, with an awe-inspiring accuracy rate of 84.54%, which surpasses other models. Based on this breakthrough, we explore how deep feature extraction techniques and powerful classification algorithms like SVM, Random Forest, and CatBoost can be integrated. Our findings prove how effective it is when we combine linear SVM with Xception attributes, which achieve a top accuracy of 93% for food image categorization. We have also analyzed the possibility of using features acquired from the pooling layer of EfficientNetB0 showing its superiority compared to others when linked to a Catboost classifier. This revolutionary study not only demonstrates the technological impact of these deep learning architectures but also shows their combined effects with machine learning classifiers, thereby advancing the frontier of accurate food image classification to new heights.

ACKNOWLEDGEMENT

I would like to express my sincere gratitude towards my supervisor Prof. Seba Susan for providing her invaluable guidance, comments, and suggestions throughout the course of the project.

The results of this thesis would not have been possible without support from all who directly or indirectly, have lent their hand throughout the course of the project. I would like to thank my parents and faculties of the Department of Information Technology, Delhi Technological University, for their kind cooperation and encouragement which helped me complete this thesis. I hope that this project will serve its purpose to the fullest extent possible.

PRANJAL KUMAR SINGH
2K22/ISY/12

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Need for Food Classification

In the current world, people are suffering from more health diseases than they ever used to. Today's generation is getting more worried about health and consuming a healthy diet, but due to hectic schedules, it becomes impossible for us to check our diet all the time, which is why automatic food monitoring has become necessary. Nutrition science, over the time, has discovered vitamins, minerals, and other components that make up our foods. Foods are classified into different groups in order to simplify dietary recommendations. For example, it’s easier to eat two orange fruits instead of consuming 50 milligrams of vitamin C.

Adding to the problem of consuming a balanced number of vitamins and minerals, a dish with a lot of different foods complicates the issue even further. Therefore, it would be better to recognize the generic type of a particular food item and we can use it to determine its nutritional value, e.g., calories. Calories are mainly used in order to know the energy content of food items. Calorie counting can provide users with a rudimentary understanding of their daily caloric consumption.

1.2 Integrated Machine and Deep Learning for Image Classification

Advancements in machine learning and deep learning algorithms has advanced the field of computer vision in terms of the human-like classification accuracies achieved. One such application profoundly impacted is the classification of different classes of food images [31], a critical task in today's data-driven world where visual content is prolifically shared on digital platforms. Machine learning makes it possible to uncover complex patterns from huge datasets [32]. These algorithms can recognize minute variations in textures, forms, and colors when used to classify food images, making it possible to distinguish between various gourmet products. On the other hand, deep learning has shown to be unmatched in its ability to extract useful information from high-dimensional image data [33]. Deep learning has outperformed feature engineering in various applications in computer vision [34]. Convolutional neural networks (CNNs) have proved to be efficient at extracting hierarchical representations of images [35], which is crucial in differentiating subtle features of food products. The performance of computer vision algorithms has been greatly accelerated and improved in terms of accuracy and efficiency due to the integration of deep neural networks and machine learning algorithms

[36]. Additionally, because machine learning algorithms are adaptable, they may continuously become better with new data, guaranteeing that categorization models develop and get better over time [37].

1.3 CNNs for Image Classification

Computer vision techniques may be used to create systems that can detect and categorise various food products. CNNs are the most widely utilised architecture for image identification and detection. Image classification using CNNs has shown to be an optimal method since convolutional neural networks can generate the scoring function directly from the pixels in an image and its built-in convolutional layer decreases an image's high dimensionality without losing its information.

Over the past few decades, lots of research has been made in the field of deep learning in order to design an optimal Convolutional Neural Network that can recognize and classify different images. The five CNNs (Inception-v3, EfficientNetB0, Xception, DenseNet121, and MobileNet) that we shall explore in this thesis are among the best-performing CNNs in today's globe.

Inception-v3 is the third iteration of Google's Inception Convolutional Neural Network, which was first unveiled for the ImageNet Recognition Challenge [2]. The main purpose of Inception-v3 is to utilise less computational resources by changing the Inception designs from previous versions. This idea was proposed by Szegedy et al., 2015[3].

By using neural architecture search to create a new baseline network and scaling it up, EfficientNets can be created [4]. In contrast to the best available Convolutional Neural Networks, EfficientNet-B7 is 8.4 times smaller and 6.1 times faster on inference but still achieves top-of-the-line 84.3% accuracy on ImageNet.

Convolutions make the Inception network computationally inefficient. These convolutions occur not only spatially, but also across depth. As a result, for each additional filter, we must do the convolution over the input depth to determine only a single output map, and as a result, the depth becomes a big hindrance in the DNN. Researchers attempted to reduce the depth, which is where Xception came into play. Xception, introduced by Chollet François [1], is an extreme version of inception and it takes the principles of Inception to their logical conclusion. In Inception, 1x1 convolutions were used to compress the original input, and we used different types of filters on each depth space from each of those input

spaces. This step is simply reversed by Xception. It employs 1x1 convolution across the depth range in order to reduce depth.

In order to address the declining accuracy brought on by high-level neural networks' disappearing gradient, DenseNet was created. DenseNet joins (.) the output of the subsequent layer with the output of the preceding layer. DenseNet121 is a 121 layers deep CNN proposed by Iandola et al., 2014 [16]. MobileNet is a computer vision model that is compact, low-latency, and low-power. As compared to other networks with conventional convolutions and the same depth in the nets, it greatly minimizes the number of parameters by using depthwise separable convolutions which is the reason why it is a lightweight deep CNN.

1.4 Machine Learning Classifiers

1.4.1 SVM

Support Vector Machines (SVMs) belong to the family of supervised classifiers; they perform well for both linear and non-linear data, and the selection of the kernel function and tuning parameters like C and γ is generally what determines how well they perform. The most basic type of SVM, suited for data that can be separated linearly, is the linear kernel. When the link between features and classes is linear, it functions well. By utilizing polynomials to translate features into a higher-dimensional space, the polynomial kernel enhances SVM's ability to handle non-linear data. The polynomial's degree is controlled by the parameter d (degree). For non-linear data, the RBF kernel, commonly referred to as the Gaussian kernel, is frequently utilized. It has infinite-dimensional spatial mapping capabilities.

The RBF kernel has two important parameters, one is the regularization parameter (C) and the other is gamma (γ). We have set C as 1 and gamma as 0.1. Maximizing the margin while minimizing the training error are trade-offs that are balanced by the regularization parameter (C). Gamma describes the range of an individual training example's impact. High levels indicate close influence, while low values indicate distance. Low gamma can result in underfitting, whereas high gamma can result in overfitting.

1.4.2 Random Forest

One of the ensemble learning techniques, Random Forest is a flexible and popular machine learning algorithm. To build a more robust, accurate model, ensemble approaches integrate several different independent models. Particularly, Random Forest is a collection of decision trees. Decision trees are

unique models that rely their choices on the features provided as input. They split the data into subsets depending on the values of the features, resulting in a decision-tree-like structure. Nodes in decision trees indicate features, and branches on those nodes reflect decisions depending on those features. The leaves, or the last nodes in a tree, stand for the forecasts or class labels.

A random subset of the dataset is used to train each tree. Additionally, only a random subset of traits is taken into account for splitting at each node. Each tree "votes" for a class in classification tasks. The Random Forest makes a prediction for the class that receives the most votes. For regression tasks, the predictions of all trees are averaged to obtain the final prediction. We have set n_estimators to be 100 and max_depth to be 20.

1.4.3 XGBoost

Extreme Gradient Boosting, or XGBoost, is a kind of gradient boosting technique used in machine learning. XGBoost is a technique for group learning. During the training phase, it creates a number of decision trees and combines their predictions to provide precise and reliable forecasts. Instead of using a single decision tree, XGBoost builds a strong learner from a series of weak learners (shallow trees). We use a learning rate of 0.1 and have set objective to multi:softmax for multiple classification. We use max_depth of 6.

1.4.4 CATBOOST

Yandex created the robust gradient boosting library CatBoost [41] to effectively handle categorical information in machine learning applications. It is notable for being scalable, resilient, and capable of achieving excellent results on a variety of datasets. CatBoost is a flexible and effective gradient-boosting toolkit that performs well with categorical data, reduces overfitting, and generates reliable and comprehensible models.

In this thesis, an effort has been made to analyze the performances of five CNNs mainly known as Inception-v3, EfficientNetB0, Xception, DenseNet121, and MobileNet in terms of their accuracies. We also explore the synergistic combination of deep learning with machine learning algorithms, specifically Support Vector Machines (SVMs) with linear, polynomial ad Gaussian/RBF kernels, Random Forest, and XGBoost, to raise the accuracy of food image classification.

CHAPTER 2

RELATED WORK

Earlier when CNNs were not used for food classification and recognition, it was done with the help of traditional algorithms using feature extraction techniques. But traditional algorithms had some drawbacks such as not making appropriate use of texture features of food. In [5], Tao et al proposed a feature extraction algorithm called color completed local binary pattern (CCLBP). The model uses CCLBP for extracting texture feature of image and HSV color histogram and Border pixel classification (BPC) color histogram for extracting color features of image. This model has obtained recognition rate that is higher by only 5% than those achieved through conventional methods of feature extraction.

Zhou et al. [6] were among the first to apply deep learning to food categorization. They examined research that employed deep learning as a data analysis tool to address difficulties in the food sector, such as calorie estimation and food recognition. The survey findings revealed that deep learning beat manual feature extractors and typical machine learning algorithms.

Pan et al. [7] introduced a DeepFood system that comprises of a deep learning-based features extractor, feature selection, and the Sequential minimal optimisation (SMO) classifier. For feature extraction, they used 3 CNNs that are AlexNet [8], CaffeNet [12], and ResNet [15]. They observed that the DeepFood framework, which combines ResNet deep feature sets, Information Gain (IG) feature selection, and the SMO classifier, outperforms previous food classification systems, with an average accuracy of 87.78%. AlexNet and CaffeNet fared equally in feature extraction, with average accuracies of 80.415% and 80.756%, respectively.

Shaha and Pawar [9] have used the pre-trained model VGG19 and fine-tuned the network parameters of it. Later on, they went to compare the results of fine-tuned VGG19 model with the results of fine-tuned VGG16 and AlexNet in terms of average recall, precision, and F-score. They have used two state-of-the-art databases GHIM10K and CalTech256 for the performance evaluation of VGG19. From the results, it was concluded that fine-tuned VGG19 performed better than the rest of the 4 two models AlexNet and VGG16 on both databases. VGG19 achieved a precision of 99.23 and an F1-score of 99.30 on the GHIM10K database and for CalTech256, VGG19 achieved a precision of 88.88 and an F1-score of 88.65.

Yanai and Kawano [10] have examined deep convolutional neural networks (DCNN) for food recognition. They have pre-trained the DCNN by picking up 1000 food categories from 21,000 categories of ImageNet and added them with the ILSVRC 1000 ImageNet categories. They have fine-tuned the pre-trained deep convolutional neural network in Caffe [12] with the help of a total of 2000 image categories. For trials, they employed the UEC-FOOD100 and UEC-FOOD256 datasets, where the DCNN attained accuracies of 78.77% and 67.57%, respectively.

VijayaKumari et. al. [11] have used pre-trained EfficientNetB0 [14] and trained it on the Food101 dataset which consists of 101,000 real-world images of food divided into 101 different categories. They trained the EfficientNetB0 in four distinct techniques by altering the supplemented data and the model's learning rate. The report concluded that EfficientNetB0 beat GoogleLeNet and Inception-v3 with 80% accuracy.

Yadav et. al. [14] have used pre-trained SqueezeNet and VGG16 convolutional neural networks. They utilized 10 classes out of 101 classes in the Food101 dataset and trained both models on those specific 10 classes. They concluded that VGG19 outperformed SqueezeNet, with training accuracy of 94.02% and a validation accuracy of 85.07%.

Chen et al. [17] have proposed an auto-clean CNN model for online food prediction image cleaning. They have used the Mealcome dataset (MLC dataset) which contains two parts, one is the clean part (MLC-CP) and another one is the dirty part (MLC-DP). They have divided the task of auto-cleaning the images into two parts, firstly they used three pre-trained CNN models that are VGG16, AlexNet, and CaffeNet for single-task comparison in order to identify which model is most suitable for food recognition and classification. They have used the MLC-CP dataset for single-task comparison. It was found that CaffeNet was the best among all three models therefore they proposed an auto-clean CNN model which was inspired by CaffeNet and trained it on MLC-CP and MLC-DP datasets.

Singla et al. [18] have used a pre-trained GoogleLeNet CNN model and evaluated the model based on food/non-food classification and food categorization. They have created their datasets by collecting images from the real world and social platforms. They have created two datasets named Food-5K and Food-11 and used them for food/non-food classification and food categorization respectively. They achieved an accuracy of 98.3% and 99.2% after fine-tuning the last 2 layers and the last six layers respectively on the Food-5K dataset. For food categorization, they achieved an accuracy of 83.6% on the Food-11 dataset.

Özsert Yiğit, Gözde, and B. Melis Özyildirim [19] have developed three deep convolutional network (DCNN) structures and compared their performances with pretrained models AlexNet and CaffeNet. Three DCNN structures have been trained on Food11 and Food101 datasets with the learning method being changed for all three models. They have used stochastic gradient descent, Nesterov's accelerated gradient, and Adaptive Moment Estimation as the learning methods for the three different structures. The first of the three structures proposed is similar to AlexNet but it has different number of layers and different kernel size. The second structure does not have the fifth convolutional layer and the third structure is the same as the first one but it does not use a local response normalization. In the paper, it is concluded that AlexNet and CaffeNet have higher accuracies compared to these three structures. Structure one was the best among all three, it gave an accuracy of 73.80% when trained with adam learning technique whereas AlexNet had an accuracy of 86.92%.

Kagaya et al. [20] have analyzed the performance of CNN for food detection and food recognition and compared it with existing techniques that are spatial pyramid matching (SPM), colour histogram and SVM, and GIST features and SVM. For normalization, they have used local response normalization (LRN). It was concluded that existing techniques had accuracies between 50% to 60% whereas CNN with a kernel size of 5x5 and 6-fold cross-validation had an accuracy of 73.70%.

Subhi, Mohammed A., and Sawal Md Ali [21] have proposed 24 layers deep convolutional neural network out of which 21 layers are convolutional and 3 layers are fully connected. They have fixed the stride to 1 pixel for every convolutional layer and dimensions are preserved after convolution. They evaluated the food recognition capability of the model on 5800 food images distributed over 11 food categories.

Yu et. al. [22] have analyzed the performances of Inception-ResNet and Inception-V3 models on the ETHZ-FOOD-101 dataset which has 101 classes of food and around 1000 images in each class. Before training and testing the models, they pre-processed the images in order to remove background variations of the images. They have done white balancing of the images with help of the grey world method and then applied histogram equalization for getting better contrast and luminance. On full-layer training, Inception-ResNet achieved top-1 accuracy of 72.55% and top-5 accuracy of 91.31%.

Attokaren et al. [23] have retrained the Inception-v3 model on the Food-101 dataset which contains 101 different classes of food items and images in this dataset are filled with noise and intense colour. Some

of the images have the wrong label. They have properly labeled the images and rescaled them to the dimensions 299x299. Their proposed approach achieved an accuracy of 86.97%.

Alex M. Goh and Xiaoyu L. Yann [24] have used 4 classes out of 101 classes of food items of the Food-101 dataset and trained the pre-trained Inception-v3 model on it with the help of transfer learning. They have cut out the last three layers of the model and taken results from the bottleneck layer and considered them to be the feature results. They have intentionally not cleaned the images to check the accuracy of the model and rescaled the images to a maximum side length of 512 pixels.

Haddadi et al. [25] have developed an intrusion detection system (IDS) with the help of a 2-layer feed-forward neural network and backpropagation algorithm. They evaluated the model on the DARPA dataset and divided the data into 80% and 20% for training and testing respectively. The model is trained for 599 epochs and stopped because of early stopping. Two studies have been conducted using various numbers 7 of relationships between the training and test datasets. The findings suggested that the proposed IDS performed almost as well in both experiments and that the detection rates were extremely close.

Zhang et al. [26] have taken a pre-trained DCNN which is trained on the ILSVRC 1000-class dataset and retrained it on a custom dataset of 360 different classes of food items. For data cleaning, they trained a one-class SVM with deep convolutional features. They changed the output number of the last fully connected layer of the model to 360 as there are 360 classes of food items. Their framework consists of three major tasks that are food identification, cooking method recognition, and food ingredient detection task. Their network achieved 57.25% and 82.29% in the top-1 and top-5 accuracies respectively in the case of food identification task. For cooking method recognition, their model achieved an accuracy of 69.50%, and for the food ingredient detection task they achieved a precision of 60.74%.

Ragusa et al. [27] have used pre-trained AlexNet and VGG models and considered two basic transfer learning techniques which are, using the models as feature extractors and fine-tuning the models. An SVM classifier is trained on top of the features that were taken from the training set. At the time of fine-tuning, a new layer with only two nodes is substituted for the network's final layer, which originally contained 1000 units. They evaluated the models on the Flickr-NonFood dataset by splitting the dataset asymmetrically into two halves, each comprising 3583 and 4422 samples. From the results, it was concluded that AlexNet with binary SVM achieved an accuracy of 94.86% and VGG with binary SVM achieved an accuracy of 91.99%.

Rajayogi et al. [28] have analyzed and compared the performances of four CNNs pre-trained on the ImageNet dataset that are VGG16, VGG19, Inception-v3, and ResNet. They have considered the Indian Food Dataset which contains 20 classes of food items. Since there are images with multiple food items, the dataset by its very nature contains a lot of noise. The image samples also have a lot of colors, and some of them have labels that are incorrect. Each of the 20 classes of dataset contains 500 images out of which 400 are taken for training and 100 for testing the models. They have used a dropout of 0.2 to avoid overfitting of the models. They have observed that Inception-v3 outperformed all three models with an accuracy of 87.9% and a loss rate of 0.5893.

Hassannejad et al. [29] have taken three different datasets ETH Food-101, Food-101, and UEC Food-256, and augmented the images by cropping them out and resizing them to 299x299 and distorting the contrast, hue, brightness, and saturation of the images. ETH dataset has already been divided in 75% training and 25% testing but for Food-101 and UEC Food-256, they have randomly split the data into 80% for training and 20% for testing. They used pre-trained Google Inception-v3 model for evaluation which achieved top-1 accuracies of 88.28%, 81.45%, and 76.17% for datasets ETH Food-101, Food-101, and UEC Food-256 respectively.

ŞENGÜR et al. [30] have considered two pre-trained models VGG16 and AlexNet for feature extraction and used an SVM classifier to determine the class label of input images. They have used three available datasets named Food-5k, Food-11, and Food-101 that contain 2, 11, and 101 classes of food respectively. VGG16's and AlexNet's fc6 and fc7 layers features are extracted and concatenated in various combinations to get the best accuracy on three datasets. Results of the paper depict that VGG16's fc6 feature sets concatenated with AlexNet's fc6 feature sets gives the best accuracies of 99.00%, 89.33%, and 62.44% on the Food-5k, Food-11, and Food-101 datasets respectively.

Jogin et al. [16] investigated several classification techniques, such as SoftMax, Fully Connected Neural Network (FCN), SVM, Nearest Neighbor classifier and Convolutional Neural Network (CNN). Although not the best method for classifying images, the Nearest Neighbor classifier performs better than random guessing with an accuracy of 28.2%. SoftMax reaches 34.1% accuracy, whereas SVM reaches 37.4%. The study shows that while CNNs outperformed with an accuracy of 85.97%, FCN only provided 46.4% accuracy. The study came to the conclusion that CNN, in particular, shows considerable potential for a wide range of tasks linked to computer vision, voice recognition, and security, obtaining astounding accuracy rates.

Farooq et al. [17] made use of the 1098 images from 61 categories of a fast-food image dataset. For feature extraction, AlexNet was employed. For classification tasks, independent features extracted from the FC6, FC7 and FC8 fully connected layers of the network were used. These feature representations were used to train the SVM classifier. Using features from the FC6, FC7, and FC8 layers, average accuracies of 70.13%, 66.39%, and 57.2%, respectively, were obtained for the 61 categories of fast food images.

Islam et al. [42] explored efficient techniques for classifying food images using deep CNNs that have already been trained. Two approaches were looked into: retraining DCNNs on images of food and using pre-trained DCNN features to train conventional classifiers. A novel food image database, Food-22, aligned with Australian dietary guidelines, was introduced for evaluation purposes. Comparative evaluations of Food-22 and current databases indicated similar accuracy for both approaches, with the latter showing much shorter training periods.

Phiphitphatphaisit and Surinta [43] proposed a novel approach utilizing the ResNet50-LSTM network. Robust spatial features were extracted using advanced VGG, ResNet and DenseNet architectures. Temporal features were then extracted through a Conv1D-LSTM network, combining convolutional and long short-term memory networks. The resulting ResNet50+Conv1D-LSTM network achieved the highest accuracy on the challenging Food-101 dataset as compared to the state of the art.

Zhang et al. [44] proposed a food mage recognition system utilizing convolutional neural networks (CNN) and tested on two food image datasets. The results indicated that adding RGB color features significantly improved accuracy for fruit images but unexpectedly reduced accuracy for multi-food images. The study concluded that CNN's effectiveness is influenced by the dataset size, with larger datasets leading to better performance.

Other advances in the state of the art for food image recognition include ensembles of fine-tuned CNNs [45], integration of deep features with hand-crafted features like Histogram of Gradients (HoG) [46], and mobile-based food image recognition [47] involving identification of regions of interest in the food image. The review in [48] provides a comprehensive collection of deep learning models applied to the task of food image recognition.

CHAPTER 3

METHODOLOGY

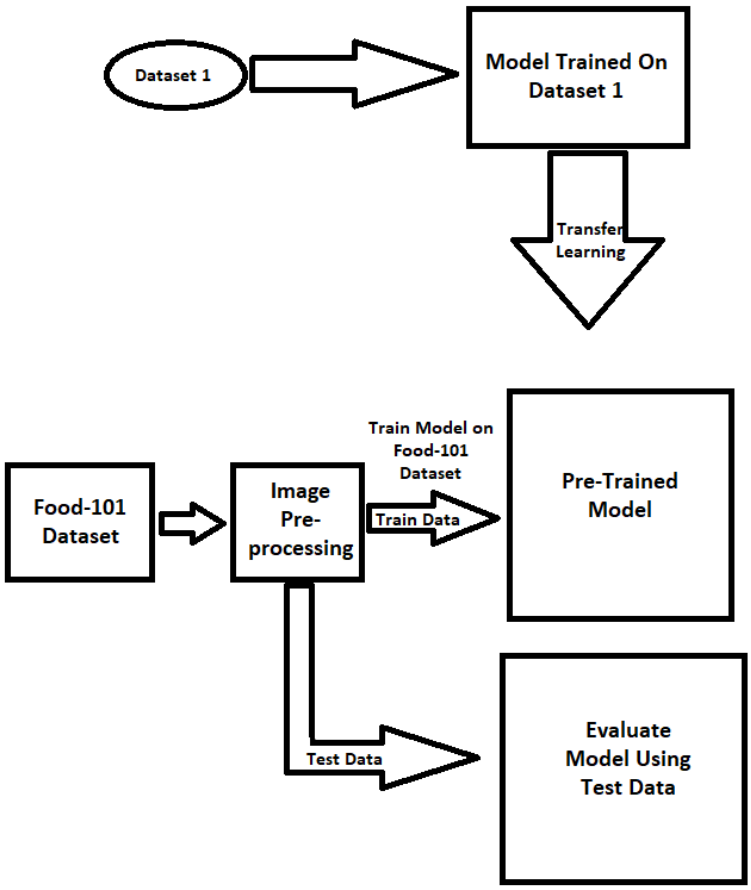

Fig. 3.1. Deep learning architecture for food classification.

Figure 3.1 shows the suggested deep learning architecture. Here dataset 1 is nothing but ImageNet dataset on which all the five models Inception-v3, EfficientNetB0, Xception, DenseNet121, and MobileNet are pre-trained on. We are using these five pre-trained models and with the help of the concept of transfer learning [9][11], we have trained our pre-trained models on Food101 dataset. Food101 images are preprocessed first in which image augmentation occurs, this implies that images are transformed and the dataset is extended. These adjustments offer a fresh perspective for capturing the object in real life rather than altering the target class of photos. Because of these modifications, the dataset now comprises a range of pictures, making our model robust and generalizable when trained on slightly different images.

3.1 Transfer learning using deep pre-trained networks

First, we'll look at the pre-trained Inception-v3 model. Inception-v3 includes convolutional layers, average pooling layers for calculating the average for each patch of the feature map, max-pooling layers, a concat layer that joins all of its input blobs together to form a single output blob, and fully connected layers that connect all of the neurons in one layer to all of the neurons in another layer. We have unfrozen all of the layers in the pre-trained Inception-v3 model. Then, to reduce training loss, we flattened the output layer to one dimension and added a fully connected layer with hidden units equal to the number of classes. To minimise overfitting, we added a 0.5 dropout rate and a final SoftMax layer for classification. To minimise overfitting and provide quicker weight updates during training, we employed stochastic gradient descent (SGD) as the optimizer and golorot_uniform as the kernel regularizer. There are a total of 22,630,277 parameters, with 22,595,845 being trainable.

Second, we have EfficientNetB0. We have made all of the layers trainable. To the pre-trained EfficientNetB0 model, we used Average Pooling to compute the average of all the values from the section of pictures covered by the kernel. We've included a thick layer with hidden units equal to the number of classes and a dropout rate of 0.5. Finally, we implemented a SoftMax layer for classifying multiple classes.. To optimise, we employed stochastic gradient descent with a learning rate of 0.1 and momentum of 0.9. There are a total of 5,213,192 parameters, of which 5,171,169 are trainable.

Third, we have Xception. We have made all of the layers trainable. We used Global Average Pooling to the pre-trained Xception model to compute the average of all values from the region of pictures covered by the kernel, and then added a flatten layer. We've included a thick layer with hidden units equal to the number of classes and a dropout rate of 0.5. We utilised the Nadam optimizer, with the lr initialised at 0.0001. Finally, we implemented a SoftMax layer to classify several classes. There are a total of 21,068,429 parameters, among which 21,013,901 are trainable.

Then we took DenseNet121. All of DenseNet121's layers are trainable. We utilised the Adam optimizer with a learning rate of 0.1 and a dropout of 0.5. We applied the AveragePooling2D layer with a pool size of (4,4) and then a flattened layer at the end. There are a total of 7,141,029 parameters, of which 7,057,381 are trainable.

Finally, we have MobileNet, Keras' first mobile computer vision model. We configured all the layers as trainable and used Adam optimiser with a learning rate of 0.1. We set the dropout to 0.5 before adding the AveragePooling2D and flatten layers, as well as a thick layer with hidden inputs equal to the number of classes.

3.2 Data Augmentation

The following parameters are considered for image augmentation:

- Rotation range = 90: This method of augmentation allows us to rotate the picture by 0 to 360 degrees clockwise. The image's pixels spin this way. To use this argument, we must pass the rotation range parameter to the ImageDataGenerator class's constructor.
- Brightness range = [0.1, 0.7]: Here, the range begins at zero, which denotes that the image is not bright. Additionally, the top range is 1, denoting the widest range of brightness. The range is defined to be between 0.1 and 0.7.
- Width shift range = 0.5: The image is really shifted to the left or right (horizontal). A positive value selected at random will move the picture to the right, while a negative value will move it to the left.
- Height shift range = 0.5: It produces a vertical image shift. If the value is a float number, it specifies how much the image's width or height will change. If it's an integer number, however, the width or height will simply be modified by that many pixel values.

3.3 Training Procedure

All five models are trained for 30 epochs on 101 distinct classes of the Food101 dataset, with a batch size of 32.

We preserved the learning rate as a variable for each model. It varies according to the epochs.

Epoch	Learning Rate
1-5	0.001
6-10	0.0002
11-15	0.00002
16-30	0.0000005

Table 3.1 Learning Rate

Table 3.1 shows the learning rate that has been set for a particular epoch.

For Inception-v3 we have taken SGD optimizer. We have set a checkpoint to save the best-only training accuracy.

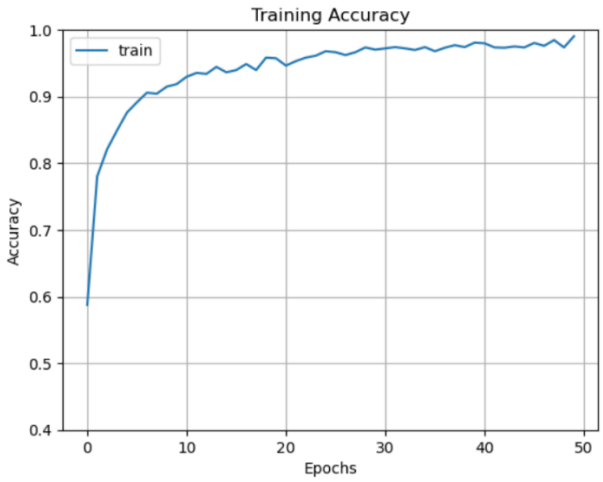

Fig. 3.2. Training accuracy of Inception-v3.

Figure. 3.2 Shows the training accuracy graph for the fine-tuned Inception-v3 model which achieved the highest training accuracy of 80.45%.

For EfficientNetB0, we have used SGD optimizer with a learning rate of 0.1 and momentum of 0.9. We achieved a training accuracy of 72.94% on the 101 training classes of the Food101 dataset.

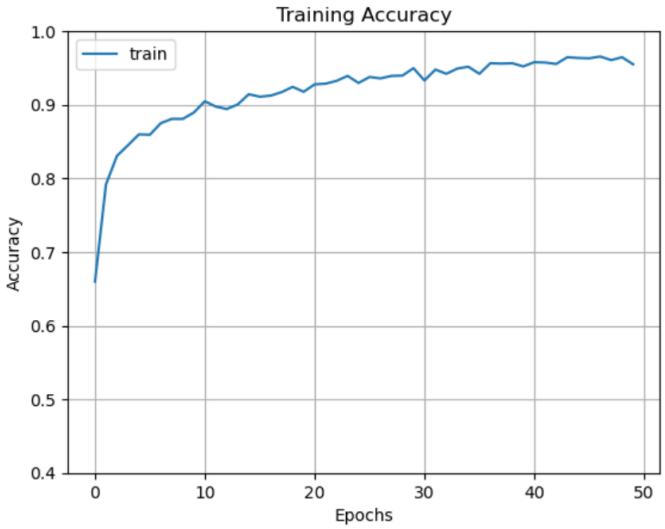

Fig. 3.3. Training accuracy of EfficientNetB0.

Figure 3.3 Shows the training accuracy graph for the fine-tuned EfficientNetB0 model.

For Xception, we have used Nadam optimizer with a learning rate of 0.0001. The highest training accuracy that the Xception model achieved is 91.11%.

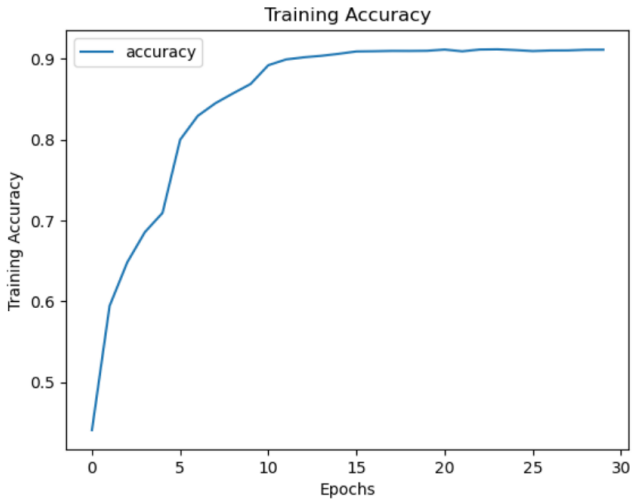

Fig. 3.4. Training accuracy of Xception.

Figure 3.4. Shows the training accuracy graph for the fine-tuned Xception model.

For DenseNet121, we have used Adam optimizer with a learning rate of 0.1. DenseNet121 achieved the highest training accuracy of 73.02% after 30 epochs.

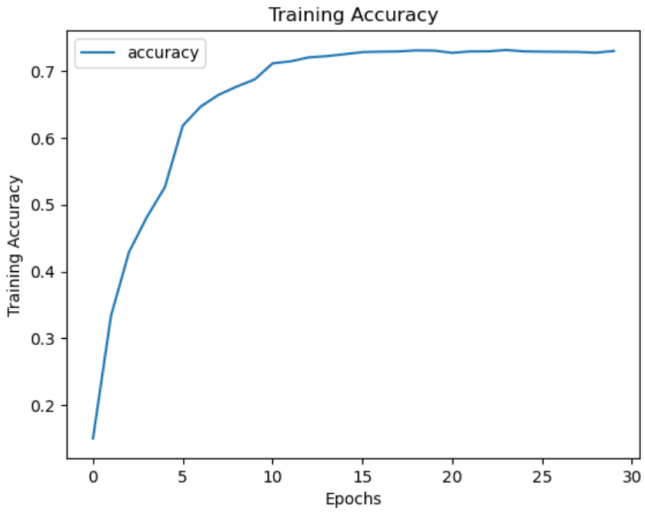

Fig. 3.5. Training accuracy of DenseNet121.

For MobileNet also we have used Adam optimizer with its learning rate initialized to 0.1. It was the fastest model when it comes to the training time comparison because of its compact size and low latency. The highest training this model achieved on the Food-101 dataset was 75.60% for 30 epochs.

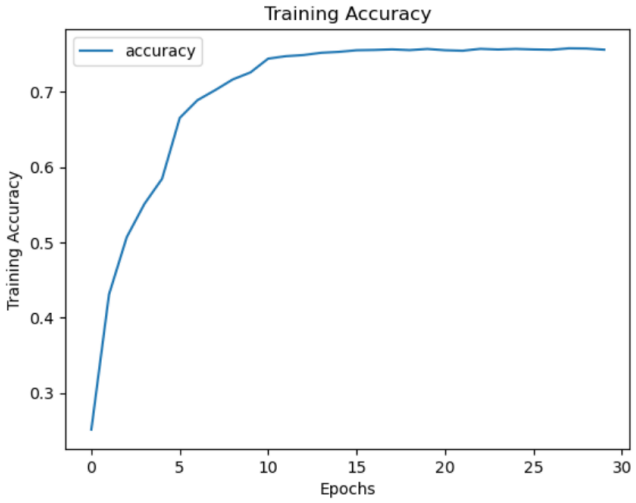

Fig. 3.6. Training accuracy of MobileNet.

3.4 Integrated SVM-Xception model

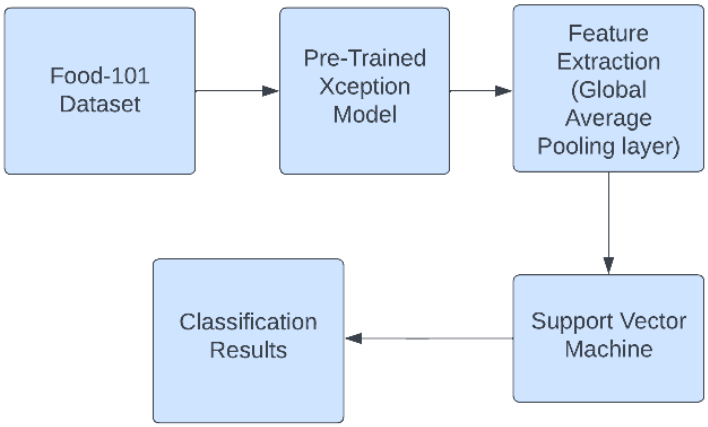

Fig. 3.7. Feature extraction and classification pipeline.

After the pre-trained Xception model is trained on the Food-101 dataset, learning complex features from the images. Features are extracted from the last layer of the Xception model, which is the Global Average Pooling layer, resulting in a feature vector of 2048 dimensions. Each image in the dataset is transformed into a set of 2048 features. The extracted features are fed into various machine learning classifiers, such as SVM (with linear, polynomial and Gaussian/RBF kernels), Random Forest, and XGBoost. These classifiers are trained on the extracted features to learn the patterns and relationships within the data. The classifiers output the final classification results, indicating the predicted classes for the given food images. This process allows us to leverage the pre-trained Xception model's feature extraction capabilities and then use traditional machine learning classifiers for accurate food image classification

Each image is transformed into a 2048-dimensional feature vector by the Xception pre-trained model. We have 18750 features in the training dataset and 6250 features in the testing dataset.

Features extracted from Xception are fed into SVM (linear, polynomial and Gaussian/RBF kernels), Random Forest, and XGBoost classifiers, and their accuracies are compared for evaluation. We applied Support Vector Machines (SVM) with three distinct kernels, training each on the extracted features from Xception**.**

Support Vector Machines (SVMs) belong to the family of supervised classifiers; they perform well for both linear and non-linear data, and the selection of the kernel function and tuning parameters like C and γ is generally what determines how well they perform. The most basic type of SVM, suited for data that can be separated linearly, is the linear kernel. When the link between features and classes is linear, it functions well. By utilizing polynomials to translate features into a higher-dimensional space, the polynomial kernel enhances SVM's ability to handle non-linear data. The polynomial's degree is controlled by the parameter d (degree). For non-linear data, the RBF kernel, commonly referred to as the Gaussian kernel, is frequently utilized. It has infinite-dimensional spatial mapping capabilities.

The RBF kernel has two important parameters, one is the regularization parameter (C) and the other is gamma (γ). We have set C as 1 and gamma as 0.1. Maximizing the margin while minimizing the training error are trade-offs that are balanced by the regularization parameter (C). Gamma describes the range of an individual training example's impact. High levels suggest intimate influence, and low amounts indicate distance. A low gamma can lead to underfitting, whereas a high gamma can cause overfitting.

3.5 Fusion of Deep Features Using CatBoost

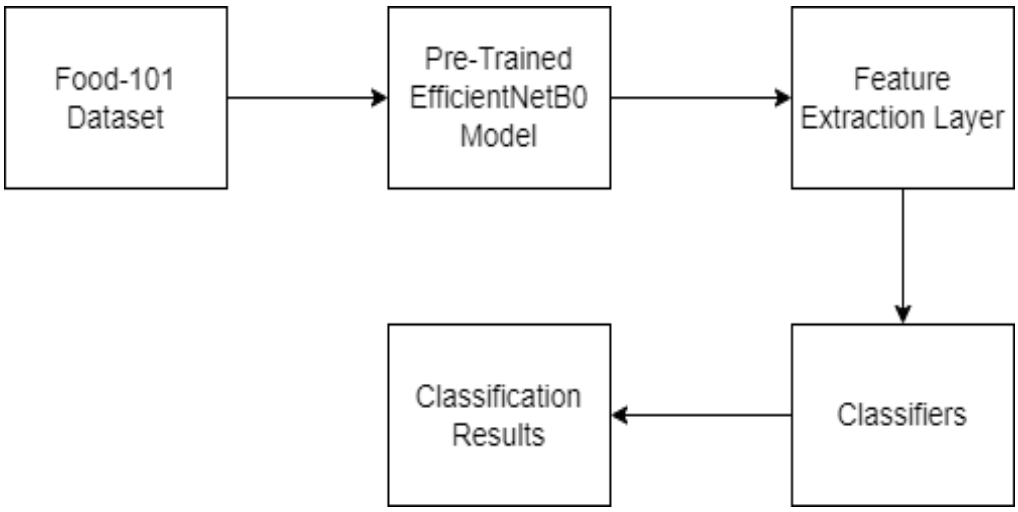

Fig. 3.8. Hybrid Feature Extraction and Classification Pipeline.

We employ a pre-trained EfficientNetB0 model to extract complicated characteristics from the Food101 dataset's pictures. These are collected by the Average Pooling layer, which yields a feature vector of shape (None, 3, 3, 1280). This means that these characteristics should be flattened to obtain a tensor having the shape 11520. Thus, each image in this dataset is modified according to this principle generating as many as 11520 features. Further on these characteristics are pooled together for further computing using several machine learning classifiers that include CatBoost, SVM and Random Forest. These classifiers make use of the collected features to learn about patterns and correlations within data. During post training phase, classifiers make final classification results indicating the expected classes for food pictures under consideration. Therefore, it functions by first using extraction abilities of a pre-trained EfficientNetB0 model followed by traditional machine learning frameworks that will bring about accurate categorization of food images.

EfficientNetB0 presents an intricate design for fetching compound layouts from input snapshots. Convolutional layers form part of its input flow while middle flow includes repeated residual blocks.; and the exit flow consists of further convolutional layers. The model introduces depth-separable convolutions and shortcut connections to enhance its feature

extraction performance of images having a size of 224x224 pixels and three RGB color channels. Unlike the exit flow which contains 8 residual blocks, intermediate flow consists of 16 repeated residual blocks that make it more robust in differentiating complicated image features.

For optimization we use Stochastic Gradient Descent (SGD) optimizer having a learning rate of 0.1 and momentum of 0.9 together with dropout rate 0.5 to prevent overfitting as well as facilitate learning.

During training on the Food 101 data set, EfficientNetB0 model works with a batch size equal to 32 across several epochs thus allowing for iterative refining its parameters. The learning rate remains flexible throughout the training process enabling adaptive tweaks for optimized model efficiency in every epoch.

CatBoost is a kind of gradient boosting such that it is good at dealing with machine learning problems where categorical data is involved for instance classification and regression challenge. It is outstanding in performance and provides cutting-edge outcomes particularly for regression and classification tasks. CatBoost excels when it comes to working with raw categorical features. This column is essential because the model is trained with an efficient approach that converts categorical attributes to numeric values so simplifying data preparation for improved performance can happen through relevance learning. Some hyperparameters are available in CatBoost that can be adjusted to enhance performance. The key aspects to take into account are the number of trees, the depth of the tree, and the learning rate.

Learning rate defines the magnitude of each step in gradient descent, establishing the model's convergence speed and performance as well.The depth of each decision tree is a major determinant of the difficulty while the total complexity of the ensemble is affected by the number of trees.

The CatBoost classifier was fine tuned for a balance between complexity and generalization by constraining the tree depth to 5 and setting the learning rate to 0.01. To prevent overfitting and ensure robustness enhancing the model’s accuracy and stability in categorizing food photos the following were used: 1000 trees, early stopping, and L2 regularization..

CHAPTER 4

RESULTS AND DISCUSSION

4.1 Dataset

In this section, we will be discussing the dataset that we have used and the results that we got after training and testing all three models on the particular dataset.

We utilised the Food101 dataset, which consists of 101 distinct food groups or categories, each with 1000 real-world photos. Food101 is a tough multimodal dataset, with noise in the training pictures. [13]. Out of these 1000 photographs for each class, 750 are allocated to training and 250 are for testing. We used transfer learning to train all 101 food classes' pre-trained models on their training photos.

.

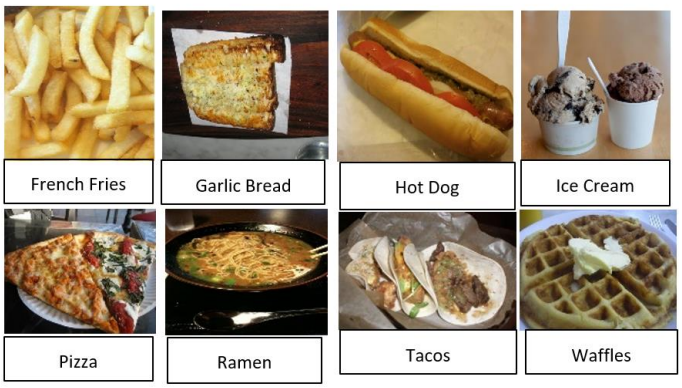

Fig. 4.1. Eight classes of Food.

Figure 4.1 represents three classes out of 101 classes the of Food-101 dataset that we have used for the evaluation of models. Each of these classes has two different folders, one contains training images and the other contains testing images.

Classes	Train Images	Test Images
101	750	250
Total images	75750	25250

Table 4.1 Training and Testing Images

Table 4.1 shows the training and testing images. We have a total of 75750 images for training and 25250 images for testing.

4.2 Results

4.2.1 Results of Transfer Learning

We will compare the outcomes of all five models based on testing accuracy by assessing them on the test dataset.

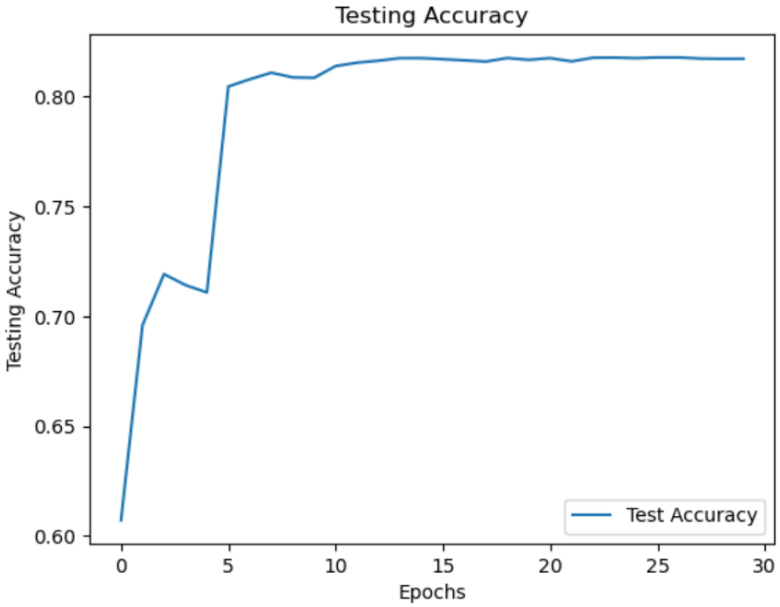

Fig. 4.2. Testing accuracy of Inception-V3.

Figure 4.2 represents the testing accuracy of Inception-v3 for the 101 classes of food. Inception-v3 attained the greatest testing accuracy of 81.73% across 30 epochs.

.

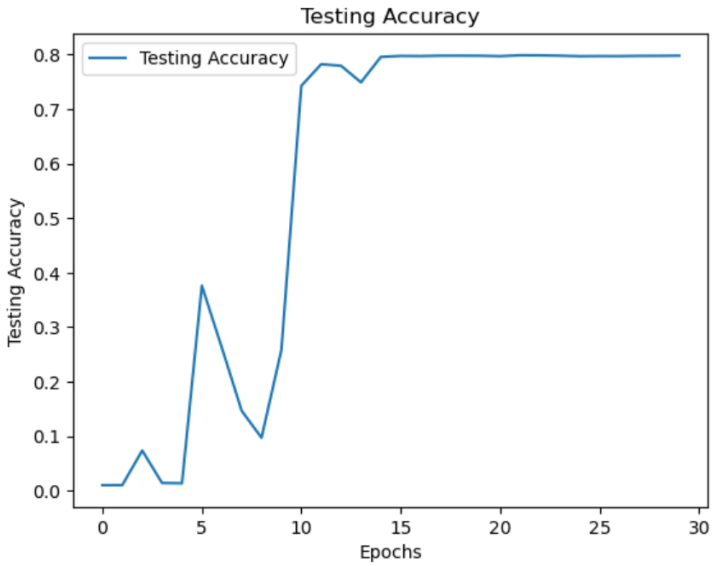

Fig. 4.3. Testing accuracy of EfficientNetB0.

Figure 4.3 represents the testing accuracy graph of EfficientNetB0. The testing accuracy recorded for the model after 30 epochs was 79.81%.

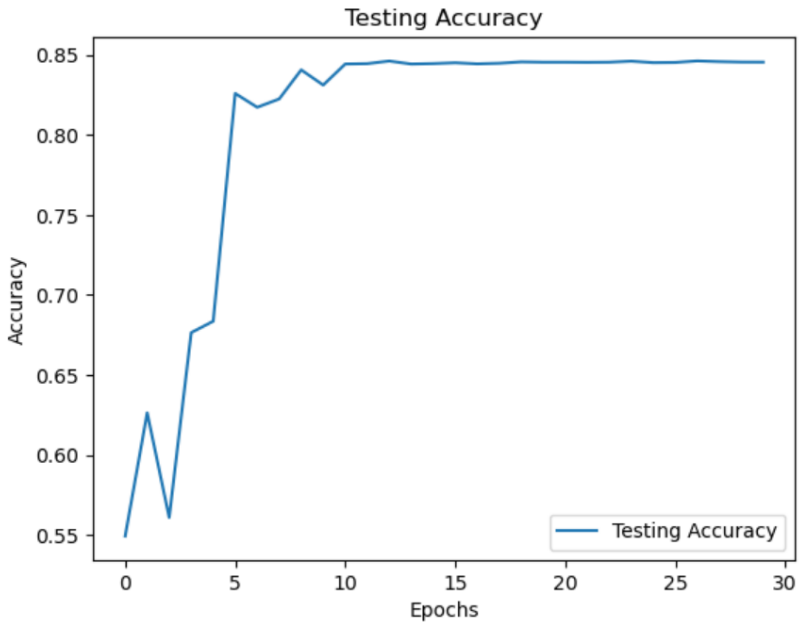

Fig. 4.4. Testing accuracy of Xception.

Figure 4.4 shows the testing accuracy of the Xception model. Xception obtained a testing accuracy of 84.54% after 30 epochs. This model achieved the highest testing accuracy among all the five models.

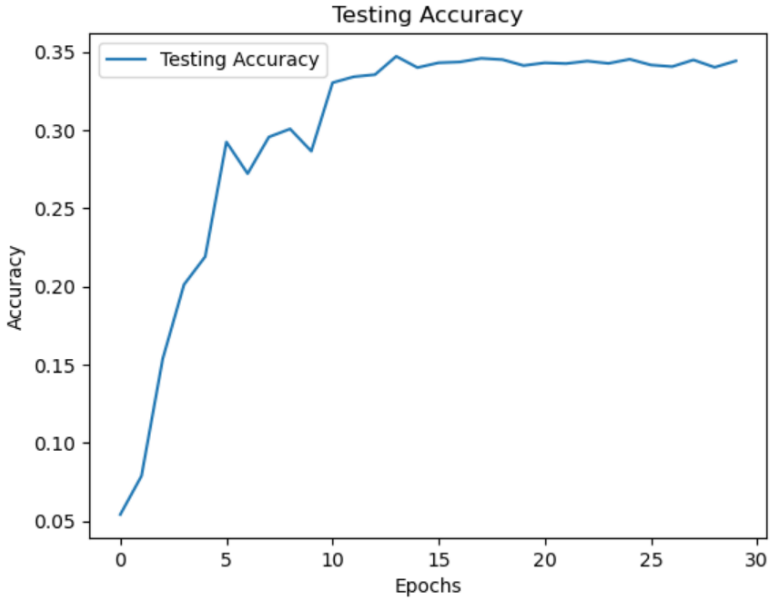

Fig. 4.5. Testing accuracy of DenseNet121.

DenseNet121 achieved the lowest testing accuracy among all five models. It achieved a testing accuracy of 34.41% after the completion of 30 epochs.

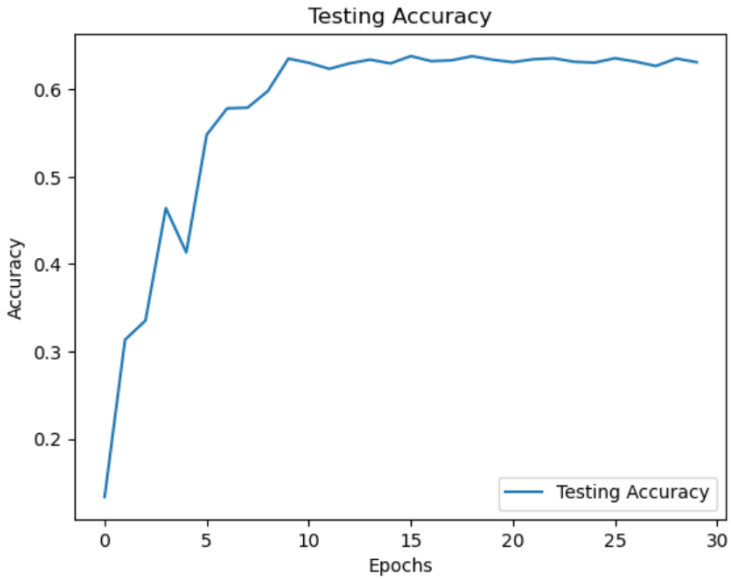

Fig. 4.6. Testing accuracy of MobileNet.

Figure 4.6 depicts the training accuracy graph for MobileNet. MobileNet obtained a testing accuracy of 63.09% after 30 epochs.

CNN	Testing Accuracy
Inception-v3	81.73%
EfficientNetB0	79.81%
Xception	**84.54%**
DenseNet121	34.41%
MobileNet	63.09%

Table 4.2 Accuracy comparison of three pre-trained models

Table 4.2 contains the list of all the convolutional neural networks along with their testing accuracies. As shown, Xception outperformed all other models in the classification challenge, with an accuracy of 84.54%, while Inception-v3 came in second with 81.73%. The DenseNet-121 model fared the poorest with 34.41% for the food picture classification assignment.

4.2.2 Results of Integrated Xception-SVM model

Fig. 4.7 illustrates the testing accuracy curve of the Xception model across 30 epochs. As observed from this graph, the highest test accuracy achieved by Xception is 88.94%.

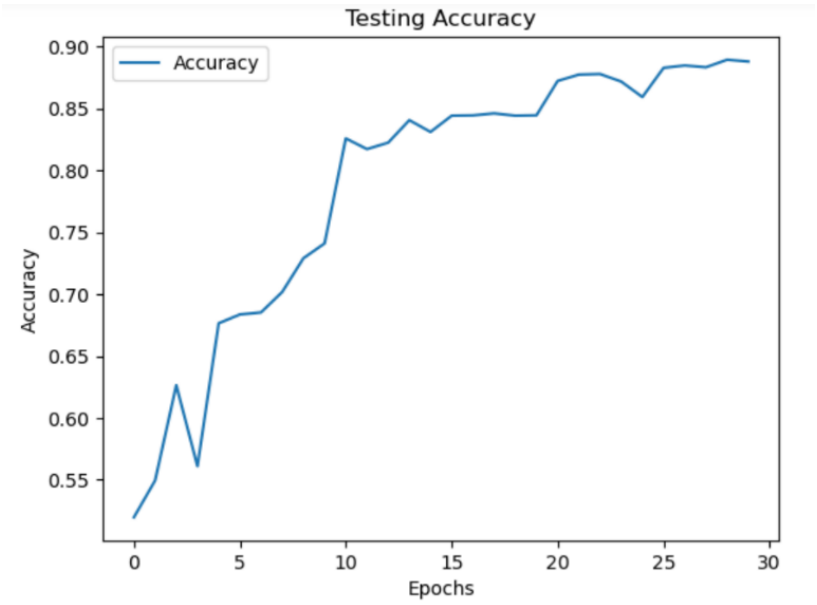

Fig. 4.7. Testing Accuracy of Integrated SVM-Xception

Each image is transformed into a 2048-dimensional feature vector by the the Xception pre-trained model is utilized with a training dataset containing 18,750 features and a testing dataset containing 6,250 features. All classifiers are trained using the complete set of 18,750 features for the purpose of classification.

The features obtained from the Xception model are inputted into various classifiers, including Support Vector Machines (SVM) with linear, polynomial, and Gaussian/RBF kernels, as well as Random Forest, CatBoost and XGBoost. The accuracies of these classifiers are then compared for evaluation. Specifically, we employed Support Vector Machines (SVM) with three different kernels, training each on the features extracted from Xception.

Table 4.3 presents the classification accuracy of different models on the test features extracted from Xception for the three different kernels. Amongst these, the linear kernel SVM exhibited superior performance, achieving the highest accuracy of **93%**. The random forest implementation, utilizing 100 decision trees led to an accuracy of

78.32%. For XGBoost, we fine-tuned the learning process with a rate of 0.1, employing a multi:softmax objective for multiple classifications. We set the maximum depth of the trees to 6, resulting in an accuracy of 68.92%. For CatBoost we have used 1000 decision trees which led to an accuracy of 91.36%.

Model	**Accuracy**
Xception-SVM (Linear)	**93%**
Xception-SVM (Polynomial)	86.20%
Xception-SVM (RBF)	89.63%
Xception-CatBoost	91.36%
Xception-Random Forest	78.32%
Xception-XGBoost	68.92%
Xception base model	84.54%

Table 4.3 Performance comparison of different classifiers and Xception

4.2.3 Results of Integrated EfficientNetB0-CatBoost model

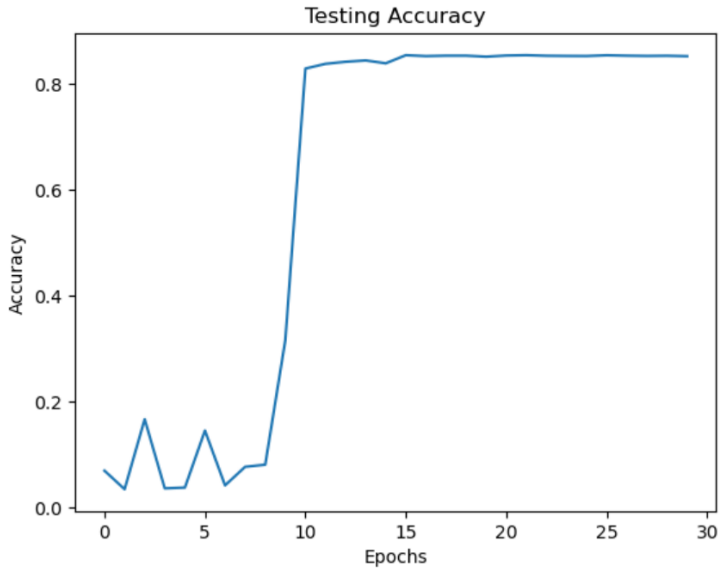

Fig. 4.8 Testing Accuracy of Integrated CatBoost-EfficientNetB0

Figure 4.8 presents the testing accuracy curve of the EfficientNetB0 model over 30 epochs, showcasing its performance trajectory. Notably, the model attains a peak test accuracy of 81.49%, highlighting its efficacy in accurately classifying food images.

Then, CatBoost, SVM, and Random Forest classifiers use the retrieved features from EfficientNetB0 as input, allowing for a thorough comparison of their individual accuracies.

Each food image is transformed into a high-dimensional 11520-feature vector, with the training dataset comprising 18,750 features and the testing dataset containing 6,250 features. Then, all classifiers are trained on the 18,750 characteristics to assist classification tasks, ensuring a full evaluation of their performance across varied food categories.

We then tested 11520-dimensional feature vectors from EfficientNetB0 model using the CatBoost classifier and we had good results on the testing dataset, achieving an accuracy of **87.53%.** This illustrates how well features taken from EfficientNetB0 work with the CatBoost classifier to provide precise food image recognition.

The characteristics were applied to Support Vector Machines (SVMs) with a linear kernel. The EfficientNetB0 model produced extracted features with a classification accuracy of **84.61** percent on the testing dataset.

Our random forest implementation produced an accuracy of **74.15%** by using 200 decision trees and a maximum depth of 30.

Model	Accuracy
EfficientNetB0-CatBoost	**87.53%**
EfficientNetB0-SVM(Linear)	84.61%
EfficientNetB0-Random Forest	74.15%
EfficientNetB0 base model	81.49%

Table 4.4 Performance comparison of different classifiers combined with EfficientNetB0.

Table 4.4 shows how accurate some of these methods are that have been explored in our research project, where CatBoost reached the highest point as it had more application on our experiment which dealt with food image classification using features obtained from EfficientNetB0.

CHAPTER 5
CONCLUSION

We evaluated the effectiveness of five well-known Convolutional Neural Networks (CNNs) and several machine learning methods in identifying deep features taken from the Xception model for food image recognition. We used pre-trained models such as Inception-v3, EfficientNetB0, Xception, DenseNet121, and MobileNet, which were first trained on the ImageNet dataset and fine-tuned to categorise 101 food categories from the Food101 dataset. When it comes to test accuracies, our assessment was based on demonstrating that Xception has the greatest classification rates (84.54%) despite its modest size compared to Inception V3.

Our investigation found that linear SVMs outperformed other methods when analysing Xception-derived test characteristics. The linear kernel SVM is the optimal classifier for deep feature classification compared to other alternatives. We emphasised the importance of transfer learning by using a pre-trained Xception model for feature extraction, which contributed considerably to improved classification outcomes. We conclude from our study that Xception-SVM (93%) is the best hybrid model followed by EfficientNetB0-CatBoost (87.53%), proving the efficiency of the proposed hybrid framework involving deep features and machine learning classifiers for classification of food images.

Our research concluded that deep learning models, are highly efficient for categorising food images. We acquired CatBoost as the best option among conventional machine learning classifiers in terms of efficiency compared to SVM and Random Forest because it can properly cope with gathered food picture information. Combining classical machine learning classifiers with deep learning feature extraction has potential for food image classification research and applications. This study emphasises the necessity of selecting appropriate classifiers and optimising hyperparameters to improve classification accuracy and overall system performance.

CHAPTER 6

REFERENCES

[1] Chollet, François. "Xception: Deep learning with depthwise separable convolutions." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1251-1258. 2017.

[2] Szegedy, Christian, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. "Going deeper with convolutions." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1-9. 2015.

[3] Szegedy, Christian, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. "Rethinking the inception architecture for computer vision." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818-2826. 2016.

[4] Tan, Mingxing, and Quoc Le. "Efficientnet: Rethinking model scaling for convolutional neural networks." In *International conference on machine learning*, pp. 6105-6114. PMLR, 2019.

[5] Tao, Huawei, Li Zhao, Ji Xi, Ling Yu, and Tong Wang. "Fruits and vegetables recognition based on color and texture features." *Transactions of the Chinese Society of Agricultural Engineering* 30, no. 16 (2014): 305-311.

[6] Zhou, Lei, Chu Zhang, Fei Liu, Zhengjun Qiu, and Yong He. "Application of deep learning in food: a review." *Comprehensive reviews in food science and food safety* 18, no. 6 (2019): 1793-1811.

[7] Pan, Lili, Samira Pouyanfar, Hao Chen, Jiaohua Qin, and Shu-Ching Chen. "Deepfood: Automatic multi-class classification of food ingredients using deep learning." In *2017 IEEE 3rd international conference on collaboration and internet computing (CIC)*, pp. 181-189. IEEE, 2017.

[8] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." *Communications of the ACM* 60, no. 6 (2017): 84-90.

[9] Shaha, Manali, and Meenakshi Pawar. "Transfer learning for image classification." In *2018 second international conference on electronics, communication and aerospace technology (ICECA)*, pp. 656-660. IEEE, 2018.

[10] Yanai, Keiji, and Yoshiyuki Kawano. "Food image recognition using deep convolutional network with pre-training and fine-tuning." In *2015 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, pp. 1-6. IEEE, 2015.

[11] VijayaKumari, G., Priyanka Vutkur, and P. Vishwanath. "Food Classification using Transfer Learning Technique." *Global Transitions Proceedings* (2022).

[12] Jia, Yangqing, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. "Caffe: Convolutional architecture for fast feature embedding." In *Proceedings of the 22nd ACM international conference on Multimedia*, pp. 675-678. 2014.

[13] Bossard, Lukas, Matthieu Guillaumin, and Luc Van Gool. "Food-101–mining discriminative components with random forests." In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part VI 13*, pp. 446-461. Springer International Publishing, 2014.

[14] Yadav, Sapna, and Satish Chand. "Automated food image classification using deep learning approach." In *2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS)*, vol. 1, pp. 542-545. IEEE, 2021.

[15] He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep residual learning for image recognition." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770-778. 2016.

[16] Iandola, Forrest, Matt Moskewicz, Sergey Karayev, Ross Girshick, Trevor Darrell, and Kurt Keutzer. "Densenet: Implementing efficient convnet descriptor pyramids." *arXiv preprint arXiv:1404.1869* (2014).

[17] Chen, Hao, Jianglong Xu, Guangyi Xiao, Qi Wu, and Shiqin Zhang. "Fast auto-clean CNN model for online prediction of food materials." Journal of Parallel and Distributed Computing 117 (2018): 218-227.

[18] Singla, Ashutosh, Lin Yuan, and Touradj Ebrahimi. "Food/non-food image classification and food categorization using pre-trained googlenet model." In *Proceedings of the 2nd International Workshop on Multimedia Assisted Dietary Management*, pp. 3-11. 2016.

[19] Özsert Yiğit, Gözde, and B. Melis Özyildirim. "Comparison of convolutional neural network models for food image classification." *Journal of Information and Telecommunication* 2, no. 3 (2018): 347-357.

[20] Kagaya, Hokuto, Kiyoharu Aizawa, and Makoto Ogawa. "Food detection and recognition using convolutional neural network." In *Proceedings of the 22nd ACM international conference on Multimedia*, pp. 1085-1088. 2014.

[21] Subhi, Mohammed A., and Sawal Md Ali. "A deep convolutional neural network for food detection and recognition." In 2018 IEEE-EMBS conference on biomedical engineering and sciences (IECBES), pp. 284- 287. IEEE, 2018.

[22] Yu, Qian, Dongyuan Mao, and Jingfan Wang. "Deep learning based food recognition." *Technical report, Stanford University* (2016).

[23] Attokaren, David J., Ian G. Fernandes, A. Sriram, YV Srinivasa Murthy, and

Shashidhar G. Koolagudi. "Food classification from images using convolutional

neural networks." In TENCON 2017-2017 IEEE Region 10 Conference, pp. 2801-

2806. IEEE, 2017.

[24] Goh, Alex M., and Xiaoyu L. Yann. "Food-image Classification Using

Neural Network Model." Int. J. of Electronics Engineering and Applications 9,

no. 3 (2021): 12-22.

[25] Haddadi, Fariba, Sara Khanchi, Mehran Shetabi, and Vali Derhami.

"Intrusion detection and attack classification using feed-forward neural network."

In 2010 Second international conference on computer and network technology,

pp. 262-266. IEEE, 2010.

[26] Zhang, Xi-Jin, Yi-Fan Lu, and Song-Hai Zhang. "Multi-task learning for food

identification and analysis with deep convolutional neural networks." Journal of Computer Science and Technology 31, no. 3 (2016): 489-500.

[27] Ragusa, Francesco, Valeria Tomaselli, Antonino Furnari, Sebastiano Battiato, and Giovanni M. Farinella. "Food vs non-food classification." In Proceedings of the 2nd International workshop on multimedia assisted dietary management, pp. 77-81. 2016.

[28] Rajayogi, J. R., G. Manjunath, and G. Shobha. "Indian food image classification with transfer learning." In 2019 4th International Conference on Computational Systems and Information Technology for Sustainable Solution (CSITSS), vol. 4, pp. 1-4. IEEE, 2019.

[29] Hassannejad, Hamid, Guido Matrella, Paolo Ciampolini, Ilaria De Munari, Monica Mordonini, and Stefano Cagnoni. "Food image recognition using very deep convolutional networks." In Proceedings of the 2nd International workshop on multimedia assisted dietary management, pp. 41-49. 2016.

[30] Şengür, Abdulkadir, Yaman Akbulut, and Ümit Budak. "Food image classification with deep features." In 2019 international artificial intelligence and data processing symposium (IDAP), pp. 1-6. Ieee, 2019.

[31] Joutou, Taichi, and Keiji Yanai. "A food image recognition system with multiple kernel learning." In *2009 16th IEEE International Conference on Image Processing (ICIP)*, pp. 285-288. IEEE, 2009.

[32] Susan, Seba, and Ashu Kaushik. "Weakly supervised metric learning with majority classes for large imbalanced image dataset." In *Proceedings of the 2020 4th International Conference on Big Data and Internet of Things*, pp. 16-19. 2020.

[33] Gheisari, Mehdi, Fereshteh Ebrahimzadeh, Mohamadtaghi Rahimi, Mahdieh Moazzamigodarzi, Yang Liu, Pijush Kanti Dutta Pramanik, Mohammad Ali Heravi et al. "Deep learning: Applications, architectures, models, tools, and frameworks: A comprehensive survey." *CAAI Transactions on Intelligence Technology* (2023).

[34] Saini, Manisha, and Seba Susan. "Comparison of deep learning, data augmentation and bag of-visual-words for classification of imbalanced image datasets." In *Recent Trends in Image Processing and Pattern Recognition: Second International Conference, RTIP2R 2018, Solapur, India, December 21–22, 2018, Revised Selected Papers, Part I 2*, pp. 561-571. Springer Singapore, 2019.

[35] Susan, Seba, and Jatin Malhotra. "CNN Pre-initialization by minimalistic part-learning for handwritten numeral recognition." In *Mining Intelligence and Knowledge Exploration: 7th International Conference, MIKE 2019, Goa, India, December 19–22, 2019, Proceedings 7*, pp. 320-329. Springer International Publishing, 2020.

[36] Saini, Manisha, and Seba Susan. "Bag-of-Visual-Words codebook generation using deep features for effective classification of imbalanced multi-class image datasets." *Multimedia Tools and Applications* 80 (2021): 20821-20847.

[37] Susan, Seba, and Jatin Malhotra. "Learning image by-parts using early and late fusion of auto-encoder features." *Multimedia Tools and Applications* 80, no. 19 (2021): 29601-29615.

[38] Jogin, Manjunath, M. S. Madhulika, G. D. Divya, R. K. Meghana, and S. Apoorva. "Feature extraction using convolution neural networks (CNN) and deep learning." In *2018 3rd IEEE international conference on recent trends in electronics, information & communication technology (RTEICT)*, pp. 2319-2323. IEEE, 2018.

[39] Farooq, Muhammad, and Edward Sazonov. "Feature extraction using deep learning for food type recognition." In *Bioinformatics and Biomedical Engineering: 5th International Work-Conference, IWBBIO 2017, Granada, Spain, April 26–28, 2017, Proceedings, Part I 5*, pp. 464-472. Springer International Publishing, 2017.

[40] Breiman, Leo. "Random forests." *Machine learning* 45 (2001): 5-32.

[41] Prokhorenkova, Liudmila, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. "CatBoost: unbiased boosting with categorical features." *Advances in neural information processing systems* 31 (2018).

[42] Islam, Kh Tohidul, Sudanthi Wijewickrema, Masud Pervez, and Stephen O'Leary. "An exploration of deep transfer learning for food image classification." In *2018 Digital Image Computing: Techniques and Applications (DICTA)*, pp. 1-5. IEEE, 2018..

[43] Phiphitphatphaisit, Sirawan, and Olarik Surinta. "Deep feature extraction technique based on Conv1D and LSTM network for food image

recognition." *Engineering and Applied Science Research* 48, no. 5 (2021): 581-592.

[44] Zhang, Weishan, Dehai Zhao, Wenjuan Gong, Zhongwei Li, Qinghua Lu, and Su Yang. "Food image recognition with convolutional neural networks." In *2015 IEEE 12th Intl Conf on Ubiquitous Intelligence and Computing and 2015 IEEE 12th Intl Conf on Autonomic and Trusted Computing and 2015 IEEE 15th Intl Conf on Scalable Computing and Communications and Its Associated Workshops (UIC-ATC-ScalCom)*, pp. 690-693. IEEE, 2015.

[45] Tasci, Erdal. "Voting combinations-based ensemble of fine-tuned convolutional neural networks for food image recognition." *Multimedia Tools and Applications* 79, no. 41-42 (2020): 30397-30418.

[46] Pouladzadeh, Parisa, and Shervin Shirmohammadi. "Mobile multi-food recognition using deep learning." *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 13, no. 3s (2017): 1-21.

[47] Kawano, Yoshiyuki, and Keiji Yanai. "Food image recognition with deep convolutional features." In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*, pp. 589-593. 2014.

[48] Zhou, Lei, Chu Zhang, Fei Liu, Zhengjun Qiu, and Yong He. "Application of deep learning in food: a review." *Comprehensive reviews in food science and food safety* 18, no. 6 (2019): 1793-1811.

[49] Singh, Pranjal Kumar, and Seba Susan. "Transfer Learning using Very Deep Pre-Trained Models for Food Image Classification." In *2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, pp. 1-6. IEEE, 2023.

[50] P. K. Singh and S. Susan, "Integrated Xception-Linear SVM Model for Food Image Recognition," 2023 International Conference on Intelligent Computing, Simulation and Optimization (ICICSO), Goa, India, 2023.

PUBLICATIONS

[1] P. K. Singh and S. Susan, "Transfer Learning using Very Deep Pre-Trained Models for Food Image Classification," 2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT), Delhi, India, 2023, pp. 1-6, doi: 10.1109/ICCCNT56998.2023.10307479.

[2] P. K. Singh and S. Susan, "Integrated Xception-Linear SVM Model for Food Image Recognition," 2023 International Conference on Intelligent Computing, Simulation and Optimization (ICICSO), Goa, India, 2023.