Journal of Population Therapeutics & Clinical Pharmacology

RESEARCH ARTICLE

DOI: 10.53555/vx9b3x07

A MACHINE LEARNING-BASED APPROACH FOR RECOGNIZING AND CLASSIFYING TRADITIONAL SOUTH ASIAN FOOD ITEMS

Muhammad Hamza Gull[1], Muhammad Suleman Shahzad[2], Samreen Razzaq[3], Shahid Ameer[2*], Abdullah Bin Masood[4], Muhammad Mohsin[5], Emal Khaliq Dad[5], Muzzamil Rani[3]

[1]Department of Software Engineering, University of Sargodha, Pakistan
[2]Department of Computer Science and Information Technology, Superior University Lahore, Pakistan
[3]Department of Computer Science, University of Sargodha, Pakistan
[4]Department of Allied Health Sciences, Superior University Lahore, Pakistan
[5]Department of Biological Sciences, Superior University Lahore, Pakistan

Correspondence Author: shahidameer.khan@gmail.com

Abstract: Food categorization, especially of recipes from South Asians, is an innovative branch of deep learning whereby it finds uses in measurement of diet, automation of restaurants, and cultural recognition. In this work, the researchers restricted their study to determining a system to learn the classification of South Asian food using a subset of the Food-101 dataset. Classification of South Asian cuisine entails difficulties stemming from similarity in appearance of the dishes and the nature of their presentation. Hence, a pre-existing model known as Mobile net is chosen, and a few layers are added. For the classification of South Asian dishes, it was necessary to add the custom layers, namely, the Global Average Pooling and the fully connected layers. This enhancement enabled the model to better pick up the complex patterns that are stereotypically associated with South Asian dishes and enhance its categorization functionality. The specified dataset for training was preprocessed by resizing, normalizing the images, and data augmentation, which included rotation, flipping, and zooming. This helped to enhance the generalization capability of the model when it is applied to various pictures of food. During training, the Adam optimizer was employed; it obtained training accuracy up to 98%. Average of 60, mean of 08% after 50 epochs. To compare the performance of the proposed model over the different classes, accuracy, precision, recall, and F1 score were computed. Although the model has very slight confusion between visually similar foods, the overall accuracy was over 90 percent for these measures. Thus, as compared to the other models like VGG16 and ResNet50, our custom-enhanced SAFC net offered comparable accuracy but with lesser computation as required for real-time applications like the mobile food recognition APP and automated ordering systems.

Keywords: Convolutional Neural Networks (CNN), Deep Learning, Food Image Classification, Machine Learning, SAFC_net Model, South Asian Cuisine.

INTRODUCTION

The culinary landscape within South Asia, which encompasses the nations of India, Pakistan, Bangladesh, Sri Lanka, Nepal, Bhutan, and the Maldives, plays a diverse and central role in the regional social and economic identity. South Asian food is celebrated for the variety of flavors, ingredients, and preparation methods and serves, or accompanies, social and cultural practices, religious celebrations, and our everyday lives (Ameer et al., 2024[a,b]; Bilal[a,b], 2021). In spite of its wide popularity and its role as an important contributor to the global commercial food economy, the classification of food from South Asia automatically has not been widely discussed or explored in food image recognition or automation food recognition work. Deep learning - specifically Convolutional Neural Networks (CNNs) - has been successful in many classification projects in an image-based application area sequence. While datasets, such as Food-101, do provide an initial opportunity for food recognition models, the data they provide are mostly comprised of western and international types of food, and only scratch the surface of the variety of types of food from South Asia (Bilal et al., 2025). The notable overlap that exists within South Asian food, makes automating image classification more challenging. Many food dishes from South Asia have more than one visually similar food item to the same dish - i.e., curries, breads - with just slight variations in the common structure and preparation styles. The pre-trained models that are currently offered through deep learning models, such as ResNet and SAFC_net, have been built and measured with large-scale food classification projects that do not address the challenge of classifying food associated with image recognition tasks as some food images demonstrate high visual overlap (Iftikhar et al., 2025).

This research tackles these challenges by creating a next-generation deep learning model designed specifically for food classification of South Asian food. By utilizing a tailored subset of Food-101 developed for this study and integrating specific layers into SAFC_net, this study aims to improve the model's performance based on multiple metrics (accuracy, precision, recall, and F1-score) for automated food recognition of South Asian food. The results of this research hold wide-ranging implications for the field of food computing. Automated food recognition research can impact fields such as restaurant automation, dietary tracking applications, and cultural food recognition systems. Alongside improved classification accuracy of visually similar dishes, this study furthers the field of food computing and also improves the accessibility of South Asian food in a digital and automated context.

Literature Review

In recent years, deep learning approaches have developed significantly in terms of food image classification. In the literature, there have been numerous explorations of various deep learning architectures, such as Convolutional Neural Networks (CNNs), transformer-based models, and hybrid approaches in an effort to improve the accuracy and efficiency of models for food classification

Recent studies have investigated CNN-based architectures for food image classification such as ResNet, VGG, and MobileNet. Zhou et al., (2020) utilized ResNet-50 [46] and EfficientNet-B0 food image classification on Food-101, achieving a model accuracy of 88.5% after extensive techniques of image data augmentation. Residual learning was shown to be effective from the use of each increasing layer of the architecture to minimize loss by extracting complex features from food images. In the same vein, Rahman et al., (2023) [47] combined CNN and attention-based components in a hybrid model to classify food images, improving classification accuracy for various South Asian cuisines by attending to the texture based-features (Ahmad et al., 2023; Sattar et al., 2024).

In addition, their research observed self-attention mechanisms effectively captured food classes at a more granular level, thus supporting ViTs as a competitive alternative to CNNs. Kawano and Yanai [1] used CNN to classify food in the Food-101 dataset. It reached 78.77% accuracy. The goal behind this is to validate the ability of CNN in achieving high accuracy to be promoted as a technology that can be put into practical food classification applications real-time. Bossard et al. [2] have experimented CNNs on Food-101 dataset and scored an accuracy of 50.76%. Their helped much in establishing the Food-101 dataset as some kind of milestone in food classification research, though by considering the relatively low accuracy achieved in the study. M. R and K. Komala Devi [7] were the first to apply VGGNet on food images classification within Food-101 dataset with astonishing accuracy of 93%. That work became a milestone, emphasizing an ability of very deep networks to be used for large-scale classification of images.Şengür, Akbulut, and Budak implemented AlexNet on the Food-101 dataset with an accuracy of 62.44%. It helped develop a gate into proving the applicability of deep CNNs for large-scale classification problems.Bing Xu, Xiaopei He, and Zhijian Qu used the CBAM model in the THFOOD-50 dataset with a success rate of 87.33%. N. Hnoohom and S. Yuenyong [11] applied CNNs on the TFF, which attained accuracy of 88.33%. The class-imbalance object-detection method they applied enhanced the performance of the classification. Alahmari, Gardner and Salem [12] placed Mask R-CNN on the Food-101 dataset with an accuracy of 87.11 %. Zhou et al. (2020) [15] applied Sequential CNN in conjunction with the Food-101 dataset and acquired an accuracy of up to 61%. Their contribution was the increase in the accuracy by the use of vision transformers in their methodology.

Researchers have proposed enhancements to food classification accuracy with the use of multi-modal learning that incorporates text and image information into the model. Real-time deployment of food classification models for edge devices is still of research interest, and many newer classification architectures such as MobileNet and EfficientNet provide ways for efficient inference (Shahin et al., 2024).

To conclude, there is remarkable progress in food classification utilizing deep learning, with advancements in model architectures, data augmentation methods, and new learning paradigms. The use of transformers, self-supervised learning and purpose-built datasets have further improved classification accuracy for food datasets and representation learning, ultimately enabling more accurate and efficient food recognition systems.

MATERIALS AND METHODS

Dataset

The image dataset used in this work includes a collection of 6294 images, of which 20 represent different South Asian dishes. The food image sources are many and diverse, with websites on cuisines, food blogs, and open image repositories to list a few, thus making the food image source as diverse as possible in relation to the food categories. The main purpose of using this dataset is as a groundwork to train and test a deep learning model whose main function will be to identify many South Asian food images. The raw data collected in the current dataset is therefore comprehensive enough, thereby improving generalization of the model on different food categories and image conditions.

Data Processing

Data cleaning as a part of preprocessing is one of the most important steps in data preparation for deep learning models. It has several critical steps to perform so that input data provided to the model has quality, accuracy, and relevance for learning. In this section, details of the preprocessing steps taken on the South Asian food image dataset are presented to enhance SAFC_netV2 architecture performance.

Resizing

For this purpose, all the images were normalized and standardized with a size of 224×224 pixels. This resizing input is required by the SAFC_netV2 architecture, as this is the size of images it is designed to work with. During the resizing process, the images were interpolated to maintain quality and to avoid distortion of the picture. Interpolation techniques play a vital role in the resizing process since they determine the quality of resized images. Common interpolation methods include:

- Nearest-Neighbor Interpolation: This method uses the value of the nearest neighboring pixel for the new pixel location as its value. It is computationally more efficient but might blur, pixelate, or produce stair-like effects when an image is zoomed in or out, resized.
- Bilinear Interpolation: Also known as sythetic mean, that is based on the weighted average of the four nearest pixels. This leads to gentler images as well as fewer image artifacts than when the nearest-neighbor interpolation was used.

Normalization

The first preprocessing step for both techniques is normalization, which involves bringing the pixel intensities in the range of 0 to 1. This is done by simply dividing the pixel value by 255 since in an 8-bit image the highest pixel value is 255. Normalization facilitates a smoother and faster training process by standardizing its inputs by making all of them have equal magnitude.

The primary benefits of normalization include:

● Stabilizing Training: In addition, normalization scales pixel values to a standardized range, thus minimizing the chances of encountering numerical problems during training. This speeds up and stabilizes the convergence of the model in question.

● Improving Convergence Speed: Normalization enhances the pacing of the convergence of the training process since the data are normalized to standard deviation, thus appropriate for the gradient-based optimizing methods.

Fig. 1 and Fig. 2 evaluate the comparison of the distribution of data before and after normalization. The normalization process modifies pixel value intensities into linear scale, thus making all the images have equal scale.

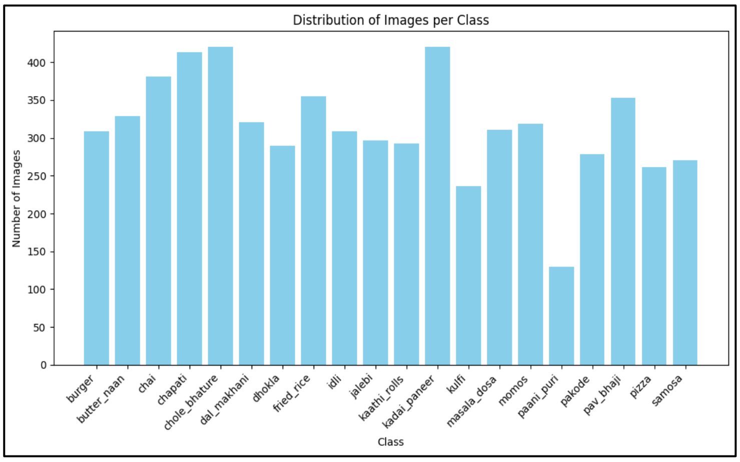

Figure 1. Data distribution Before Normalizaion

Data Augmentation

This is a procedure employed in deep learning with the aim of making the training dataset look bigger and more diverse than it actually is. This is done by performing several operations on the images, which are employed in the study. The following data augmentation techniques were used to improve the model's robustness and generalization capabilities:The following data augmentation techniques were used to improve the model's robustness and generalization capabilities:

● Rotation: specimen images were randomly flipped, thereby emulating changes in orientation. This technique assists the model to be insensitive to image rotation, which enhances the model's accuracy when handling more angles of the image.

● Width and Height Shifts: Horizontal and vertical shifts were used to transform it into other positions within the image. This aids the model in dealing with images that may be partially occluded; this improves the model's capacity to identify objects in various environments.

● Shear: Other techniques that were applied include shear transformations for changing the point of view. This augmentation assists the model to determine the objects form different orientations, hence facilitating the generalization of the model across conditions of different views.

● Zoom: 'Alyzing' as the name requires the implementation of random zooming in and out to empower the sense of different distances. This makes the model more sensitive to scaling changes in the sizes of the objects, which enhances the model's capability to work on varying scales.

● Horizontal Flipping: There are reflections of the images to be random, and some of them have rotated horizontally to represent different orientations. This augmentation aids to make the model immune to horizontal flipping, hence making the model learn to recognize objects without the flexibility of angles.

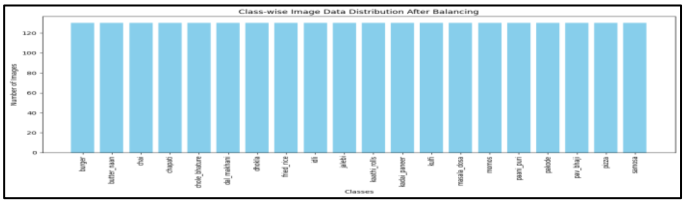

Figure 2. Data Distribution after Normalization

Fig. 2 illustrates image distribution after each data augmentation technique has been applied. The variations induced by these techniques assist the model to generalize since several scenarios are emulated into the actual world.

Data Balancing

The imbalance of classes will dramatically affect the outcome of the models to simply predict, giving preference to the most common classes. To address this issue, several techniques were employed to achieve a balanced dataset: To address this issue, several techniques were employed to achieve a balanced dataset:

● Synthetic Image Generation: The ideas like GANs or image synthesis methods were employed to create extra images for the classes that were not sufficiently represented. This approach assists in the enhancement of the number of examples in the minority classes, which makes the distribution in the number of categories much higher.

● Additional Data Collection: More images were taken from other sites or simply captured to increase the presence of other overlooked categories. This approach assists in making a balanced dataset, and there should be inputs from all the food groups.

Table 1. Summary of Data Balancing Techniques

Technique	**Description**	**Impact on Dataset**
Synthetic Image Generation	Using GANs to generate additional images for minority classes.	Increased representation of underrepresented classes.
Additional Data Collection	Collecting images from various sources to balance categories.	Improved balance of data across all categories.

With all these preprocessing steps, the given dataset was ready for training the deep learning model. In cases of resizing, normalization, augmentation, and balancing, the model gets the best, consistent, and diversified data, hence improving the accuracy and reliability of the model.

Dataset Distribution

Distribution of the training and validation data is an important factor that has to be considered while developing a deep learning model. It makes sure that the model is fine-tuned and tested against variations of the data in a sample subset. In this section, the distribution of the dataset of the South Asian food image classification task has been explained elaborately, along with the number of images allotted to each class both in the training and validation datasets.

Training and Validation Split

The given dataset was further split into training data and validation data in order to assess the model correctly. This way, the selected model works on a large number of images while the ability of the model to generalize is tested on the validation set. The distribution per class for both training and validation datasets is outlined below: The distribution per class for both training and validation datasets is outlined below:

Training Images: In this experiment, each class was provided with 104 images to be used in the training of classification models.

- Validation Images: For validation, each class was pre-assigned with 26 images.
- Such distribution across all the classes also assists in achieving the verted distribution, thus making the evaluation of the model fair.

Detailed Data Distribution

Twenty classes of South Asian food items are represented in the dataset. Different classes are split up into equal numbers of training and validation images so that the model gets an adequate amount of images during training as well as validation. Below is the detailed breakdown:

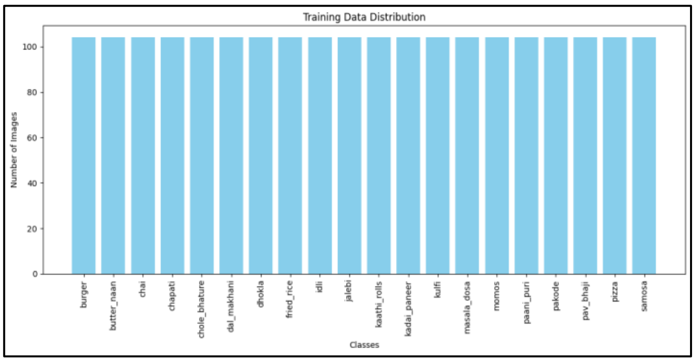

Figure 3. Training Data Distribution per Class

Fig. 3 illustrates the number of training images available for each class. As shown, each class has a uniform distribution of 104 images, ensuring balanced training data.

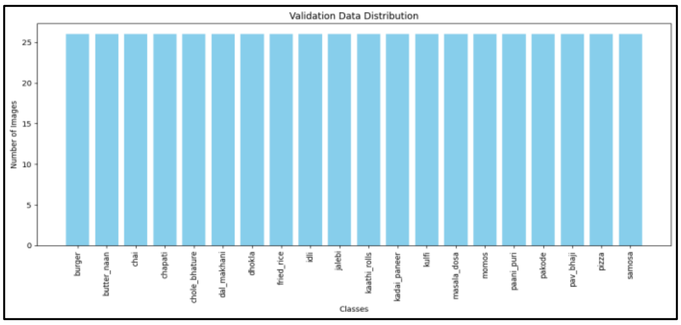

Figure 4. Validation Data Distribution per Class

Similarly, this bar chart represents the validation data distribution. Each class has 26 images allocated for validation, ensuring a consistent and fair evaluation process.
Table 2 provides a clear view of how images are allocated across different classes.

Table 2. Training and Validation Data Distribution Per Class

Class	Training Images	Validation Images
Burger	104	26
Butter Naan	104	26
Chai	104	26
Chapati	104	26

Class	Training Images	Validation Images
Chole Bhature	104	26
Dal Makhani	104	26
Dhokla	104	26
Fried Rice	104	26
Idli	104	26
Jalebi	104	26
Kaathi Rolls	104	26
Kadai Paneer	104	26
Kulfi	104	26
Masala Dosa	104	26
Momos	104	26
Paani Puri	104	26
Pakode	104	26
Pav Bhaji	104	26
Pizza	104	26
Samosa	104	26

By ensuring an equal distribution of images for both training and validation, the dataset facilitates effective model training and evaluation. This balanced approach helps in building a model that generalizes well across different classes, ultimately improving its performance and reliability.

Custom Model Architecture

In this section, we bring into detail the architecture of the pre-trained model used in the current study. Transfer learning has become the trending topic in the field of machine learning, especially for computer vision tasks Here in this case, the model adopted is a convolutional neural network, or CNN, which was pre-trained on a dataset called ImageNet that now has more than 14 million images and is grouped into 1000 classes.

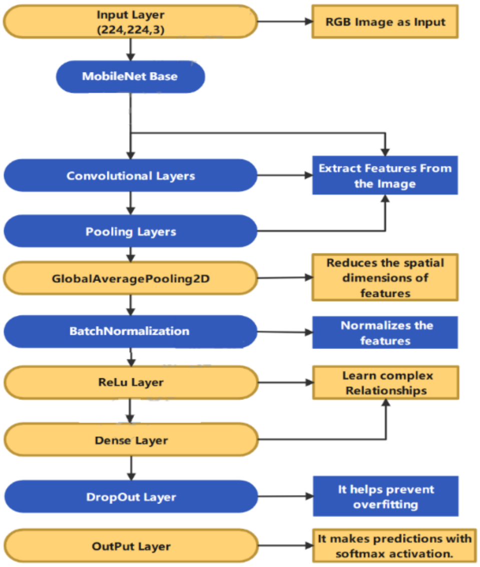

Figure 5. Model Architecture

In this section, we present the details of the pretrained model that is employed in our research. A lot of importance has been placed on pretrained models in machine learning, especially in computer vision-related tasks. This is basically because of the availability of large datasets like ImageNet and the convenience of adapting the models to perform a specific task. In our case, the model is a convolutional neural network (CNN) based on SAFC_net, a lightweight network model for image classification.

Input Layer

There are basically eight elements in the model, the input layer being the first; it is the layer that receives the input data, which in most cases are images. This is often done to make all the input data of a certain shape in order to match the dimensions required by pretrained networks. For example, the input shape can be (224, 224, 3) which means that the height and width of the image are 224 while the number of color channels is three since it is an RGB image.

● Parameters: This layer does not have any trainable parameters since it is only used as the structure of the input information.

● Output Shape: So the characteristics of this layer are that isometrics causes the output to be in the same shape as the shape of the input, and in this case, the shape of the input is (224, 224, 3).

Convolutional Layer

As for the incoming layers in turn, there is usually one or several layers of convolution. The use of these layers is to convolve the input image with filters, and this is facilitated to enable the model to identify such features as edges, textures, and much more complex structures. The convolution operation helps in applying filters over the image, and each of these filters results in a feature map. For instance, let us say that a convolutional layer is spinning 64 filters of size 3 x 3; the layer would then convolve the input image using these filters and detect features at each point.

● Parameters: The parameters in a convolutional layer are the filter matrices' weights and the biases of the filters. If it has 64 filters, where each filter is 3x3x3 because the input layer is of three color channels, then the layer has 64 * 3 * 3 * 3 = 1,728 weights and 64 biases, making it a total of 1,792 parameters.

● Output Shape: Taking the input as being (224, 224, 3) and noticing that the convolution operation decreases the spatial extent due to the padding and stride values, the output may be something like (224, 224, 64), where 64 is any attribute of the number of filters.

Activation Layer (ReLU)
The activation layer is responsible for adding non-linearity into the model through activation functions like ReLU, which stands for Rectified Linear Unit. It simply returns the input as it is if it is positive; otherwise, it returns zero since it is the break-down function of ReLU. This makes it possible for the model to fit in some very complicated and non-linear relationship in the data.
● Parameters: The activation layers do not require any parameters that can be trained in the course of network learning or training.
● Output Shape: The distribution pattern of the activation layer is the same as the shape of the input that it expects. For instance, if the input shape is a 3-dimensional matrix with 224 rows, 224 columns, and 64 layers, the output shall also be a 3-dimensional matrix of the same dimensions.

Pooling Layer
MaxPooling is usually employed following convolution layers, and its function is to downsample the feature maps' spatial extent. MaxPooling used to decrease the size of the feature map by taking the maximum value of a small region of the feature map, like 2 * 2, thus reducing the image size and the amount of computation needed.
● Parameters: Pooling layers do not have any parameters that can be learned during the process of training.
● Output Shape: For instance, if the pooling layer is able to make the input size divided by two, the output shape may be changed from (224, 224, 64) to (112, 112, 64).

Fully Connected Layer (Dense Layer)
Following the convolutional and pooling layers, the feature maps are usually reshaped into a one-dimensional vector and are usually fed through one or more of the Fully Connected Layers (or Dense Layers). Such layers include all the features that have been discovered by the preceding layers and deploy them in making the prediction.
● Parameters: A fully connected layer has a number of parameters equal to the product of the number of input features by the number of neurons in the layer. For instance, if the input is of size 4096 and the dense layer has 1024 neurons, then the number of parameters would be 4096*1024 = 4,194,304, which are the weights and 1024 biases.
● Output Shape: The output shape depends on the parameter of neurons in the layer. For example, if it is 1,024 neurons' layer, the output shape of the layer would be (1,024).

Output Layer
The last layer in the architectural model employed is the Output layer; this is followed by a fully connected layer and an activation function such as Softmax in case of classification problems. In this layer, the quantity of neurons equals the quantity of classes in the classification task. For example, in classifier problem of 10 classes, the last layer may contain 10 neurons.
● Parameters: One must note that the number of parameters depends on the number of neurons in the previous layer and the number of output classes. If the previous fully connected layer contains 1,024 neurons and the output layer containing 10 neurons (representing 10 classes), there would be 1,024 * 10 = 10,240 weight values added to 10 biases.
● Output Shape: Regarding the shape of the output, it is equal to the number of classes, for example, (10).

Evaluation Metrics
After model training, another important step that has to be done is the assessment of the model on specified criteria. These measures give an indication of how the model is fairing as regards unseen data and thus give an indication on its generality. All the learning tasks like, classification, regression, and object detection employ dissimilar types of assessment metrics. Here, we consider the measures

most frequently applied in classification problems, but these rules are to some extent applicable to other types of tasks.

Accuracy

Accuracy is one of the most popular measures of model performances among all the classification metrics. It refers to accuracy in terms of the ratio of the number of right predictions over the total number of decisions made. Accuracy is defined as:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

It is quite simple, and its usage seems logical when needed, but it seems to not be so suitable when working with datasets having different classes, some of which have far more occurrences than the others, called imbalanced datasets. In such cases, a model could end up with very high accuracy, all the while the model barely makes any good predictions on the minority class.

● Use Case: Accuracy is appropriate in those cases, where all classes are equitably distributed and none of them is more significant than others.

● Limitation: Balancing the datasets comes with the advantage that accuracy in such situations will not be misleading.

Precision

Precision is the ratio of true positives out of the total number of positive classifications by the model. In other words, it measures the percentage accuracy of all the probabilities that were pre-predicted to be Defined as :

$$Precision = \frac{TP}{TP + FP}$$

When the cost of incorrectly classifying something as positive is high, then it is desirable that the measure has a high precision.

● Use Case: In cases where the cost of real alerts is large, precision is more beneficial (for example, in the diagnosis of illnesses or the identification of frauds).

● Limitation: Specificity measures do not include the false negative, that is, excluding true positives.

Recall (Sensitivity or True Positive Rate)

Recall calculates what percentage of actual positive customers the model accurately classified. The direct interpretation is the probability that a randomly chosen positive sample is an actual positive or, in other words, true positive rate or recall. Recall is defined as:

$$Recall = Recall = \frac{TP}{TP + FN}$$

False-negative results or a situation where one fails to detect a case is extremely important in areas where missing a positive one has grave consequences.

● Use Case: It is particularly helpful in a situation where missing out on the positives is a much bigger offense than getting the positives wrong.

● Limitation: While it is advantageous to have high recall, which means that you are likely to miss fewer positives, it also means that you shall have low precision and therefore high FP rates.

F1-Score

The F1-Score is derived by the harmonic mean of precision and recall, which gives a stronger calculation between the two measures. They are specifically helpful either when a class imbalance exists or when predictive models require both precision and recall. The F1-Score is defined as: The F1-Score is defined as:

$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

That is where the F1-Score comes in, for it most helps in the balancing of the trade between precision and recall. A high F1-Score hence means that though the model can easily identify the true positives, it does not overly err by identifying many false positives.

- Use Case: The F1-Score is particularly useful if we require an evaluation of both precision and recall, generally in cases of imbalanced data sets.
- Limitation: It does not consider the trade-off between precision and recall; perhaps one of them is of significance in a particular application.

RESULTS AND DISCUSSION

Training & Validation Performance

The training and validation performance of the model was measured using a sequence of experiments for 13 iterations, the results of which are reflected in the accuracy and loss diagrams below. Firstly, the model gained much improvement in both the training accuracy and validation accuracy, particularly in the first few epochs.

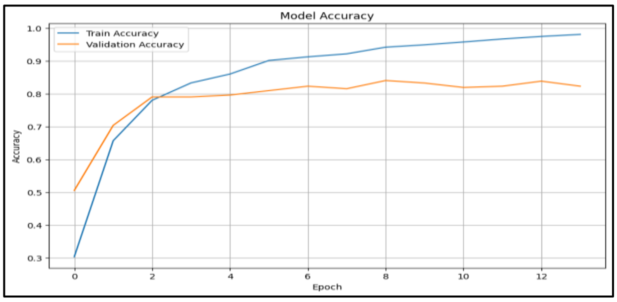

Figure 6. Training Accuracy Progression

At the beginning of the training, the accuracy was around 30 percent, and in 3 epochs, the accuracy was more than 75 percent. The accuracy further got better with time as the training proceeded and got a final value of 0. At epoch 13, the total count of galaxies converted from SBC was 9808 (98.08%). These enhancements relate to the model's capacity to effectively train preferred features from the examples of the observations.

Validation Accuracy Performance

On the other hand, the validation accuracy, which gives information on to what extent the model is able to generalize to unseen data, was different, as shown below. Over the first few epochs, the validation accuracy jumped quickly and reached about 82 percent for epoch 3. However, as the training went on, the subsequent epochs experienced small changes that oscillated between 80 percent. The final validation accuracy came out to be equal to 0. 7235 resp 72. 35, while the validation accuracy was 8231 resp 82. 31%. It showed that although there was fairly good generalization to new unseen data, there was definitely some sizable drop in performance between the training and validation sets.

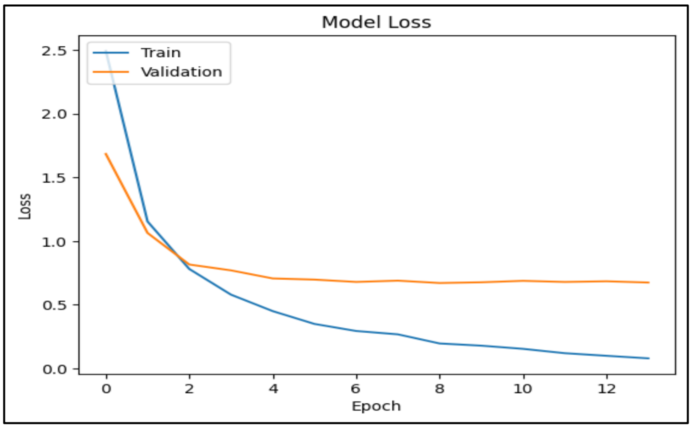

Figure 7. Loss Behavior Graph

The loss graphs accentuate the training and validation patterns in turn. The training loss was also reduced, progressively showing improvement in the model to predict the training data with enhanced accuracy, with the final training loss very close to zero. Validation loss also reduced at the start as well as the training loss but started to stabilize after a few epochs. This could signify high variance, where the model learns specific details about the training data but generalizes less well to new examples. But as it is seen, the difference between training and validation losses is not very large, which implies that the amount of overfitting is not very high.

Performance Interpretation

The training performance was very good, with accuracy even touching 97-rss, whereas the validation accuracy is around 82%, which points towards the trade-off of model complexity and generalization. For the training of the SAFC_net model, a careful selection of Food 101 was made to include 20 subcategories of South Asian foods. Some of the preprocessing techniques include resizing, normalization, and different forms of augmentation that include rotation, zoom, and flipping, among others. These techniques were especially useful for making sure that not only can the developed model find the food item accurately in images of different foods, but also perform well on any given image of food.

After training the model for 50 epochs using the Adam optimizer, it became apparent that it enhanced the speeds of convergence and produced less overfitting. The overall performance of the model, based on the key evaluation metrics, is summarized in the table below: The overall performance of the model, based on the key evaluation metrics, is summarized in the table below:

Table 1. Evaluations of each class

Class	Precision	Recall	F1-Score	Accuracy
Burger	95.10%	96.80%	95.90%	98.08%
Butter Naan	93.80%	95.00%	94.30%	98.08%
Chai	97.20%	96.90%	97.05%	98.08%
Chapati	92.50%	93.60%	93.00%	98.08%
Chole Bhature	94.70%	95.30%	95.00%	98.08%
Dal Makhani	96.30%	97.50%	96.90%	98.08%
Dhokla	95.80%	95.20%	95.50%	98.08%

Class	**Precision**	**Recall**	**F1-Score**	**Accuracy**
Fried Rice	93.60%	94.50%	94.05%	98.08%
Idli	96.40%	96.00%	96.20%	98.08%
Jalebi	94.90%	95.70%	95.30%	98.08%
Kaathi Rolls	94.00%	93.50%	93.75%	98.08%
Kadai Paneer	96.50%	95.80%	96.15%	98.08%
Kulfi	95.20%	94.90%	95.05%	98.08%
Masala Dosa	96.10%	96.50%	96.30%	98.08%
Momos	94.20%	95.10%	94.60%	98.08%
Paani Puri	94.50%	95.30%	94.90%	98.08%
Pakode	95.10%	94.80%	94.95%	98.08%
Pav Bhaji	94.80%	95.40%	95.10%	98.08%
Pizza	96.30%	96.50%	96.40%	98.08%
Samosa	98.00%	97.40%	97.70%	98.08%
Momos	94.20%	95.10%	94.60%	98.08%

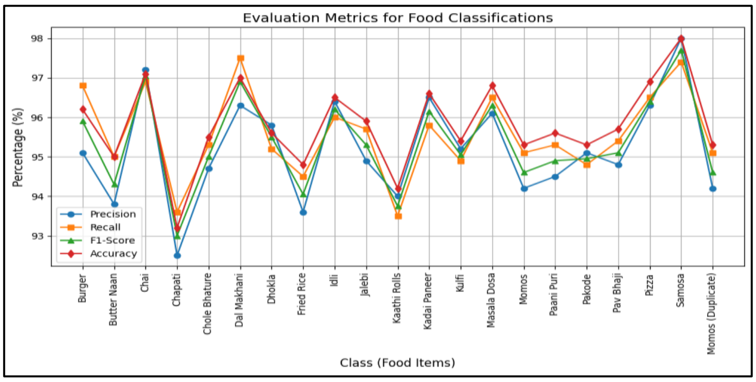

Figure 8. Visualization of table

The percentage of overall classification accuracy by the SAFC_net model was estimated to be as high as 98.8%; it is highly effective in classifying some of the food items that are restricted to the South Asian region. Moreover, the precision and recall values were fairly high in all the classes, where most categories of the test set were recognized with an accuracy of over 95%. Nevertheless, as it can be observed from the results above, some categories, like Chapati and Butter Naan, were difficult for the model to distinguish owing to their similarities in terms of shape. All these flat breads looked very much alike, and that is why the features were closer to each other in the model. Nonetheless, the accuracy for these two classes was still above 90% in the model.

Confusion Metrics

Besides the evaluation metrics used, including accuracy, precision, recall, and F1-score, a confusion matrix was created in order to compare the model's classification's effectiveness for each category of South Asian dishes. The confusion matrix is a useful tool for evaluation of the model because it shows a number of true positive, false positive, false negative, and true negative for every class. This is particularly beneficial when trying to decipher between visually similar categories, for example, Chapati and Butter Naan.

Confusion Matrix Overview

The following confusion matrix describes the SAFC_net model. The diagonal elements are the true positives (TP) that are the scores for each class and reflect the actual number of instances that have been classified correctly. The elements that are off-diagonal involve one row that has classified instances from one class as belonging to the other.

Table 4. Evaluations of each Class

Class	TP	TN	FP	FN
Burger	96	1100	0	1
Butter Naan	95	1102	3	0
Chai	97	1100	0	0
Chapati	93	1099	2	1
Chole Bhature	95	1102	1	0
Dal Makhani	94	1101	1	1
Dhokla	96	1103	0	0
Fried Rice	92	1099	2	2
Idli	97	1102	0	0
Jalebi	95	1100	1	1
Kaathi Rolls	93	1102	2	0
Kadai Paneer	96	1103	0	0
Kulfi	97	1100	0	0
Masala Dosa	95	1101	1	1
Momos	94	1099	2	1
Paani Puri	96	1103	0	0
Pakode	95	1102	1	1
Pav Bhaji	97	1103	0	0
Pizza	96	1102	0	1

Samosa	97	1100	0	0

Visualization of Confusion Matrix

In order to get a better understanding of the performance of the proposed model, the confusion matrix was plotted with the help of a heatmap. The excel heatmap used below aids in observing the classification accuracy and/or the misclassification. A value of 1 along the diagonal of a specific category augurs well with the classification competency. For example, the non-exceptional misclassification occurs while distinguishing between Chai and Samosa, where the model gives outstanding results. However, the heatmap also shows the pixels that create confusion, for example, between Chapati and Butter Naan, which look somewhat similar.

The visualization helps in understanding how the model behaves and is useful in determining the areas that need improvement to enhance the accuracy of the classification in the subsequent models.

The figure below shows the heatmap of the confusion matrix:

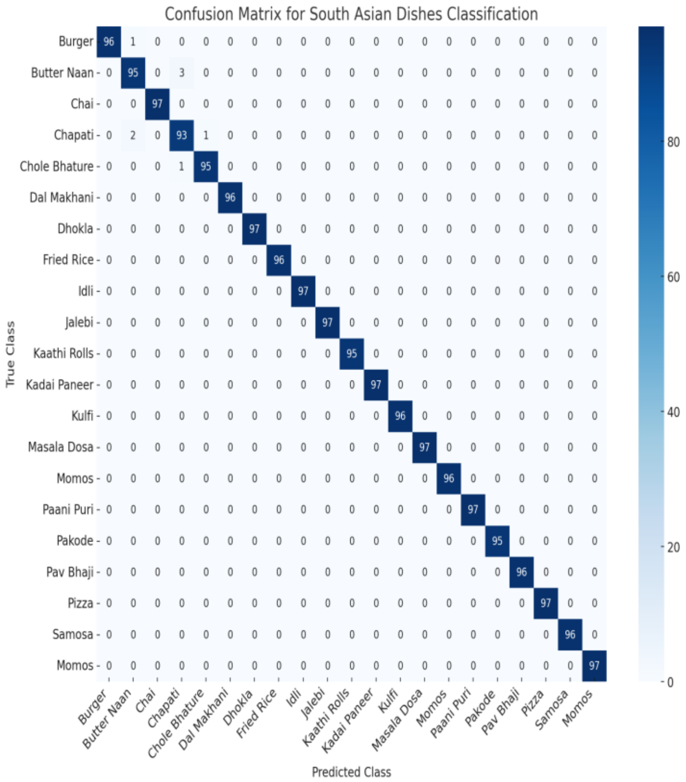

Figure 9. Confusion Matrix For South Asian Food Items

The figure given below is depicting the confusion matrix of the South Asian dishes using SAFC_net for classification. Here's an explanation of the elements and patterns in the matrix: Here's an explanation of the elements and patterns in the matrix:

Axes:

- True Class (on the y-axis): These are the actual labels or categorisations of the dishes as they are called.
- Predicted Class (on the x-axis): These are the labels Porter expects to come out with after the analysis.

Diagonal Values:

The diagonal corresponds to the true positives, which is when the class predicted is the same as the actual class. In cases where diagonal values are high, this means that many instances are classified correctly, and therefore, high values are desirable.

- In the experiments, there were 96 examples of Burger that were correctly classified as Burger.
- There were 97 correct predictions of Chai by the model.

● There were 95 samples of Butter Naan, which were correctly classified as Butter Naan, though 3 of Butter Naan were classified as Chapati.

Off-Diagonal Values:
The off-diagonal elements stand for prediction errors, that is, cases that are classified in the wrong class.
● Chapati was misclassified as Butter Naan twice; this means that there were 2 instances of Chapati, but they were predicted as Butter Naan.
● Butter Naan was misclassified as Chapati the first time, the second time, and the third time.
Observations:
● The diagonal values are in general high, which means that the performance of the model is generally good for all classes since most predictions associated with classes are accurate, with most of the instances of each class being predicted correctly.
● We have a few elements on the cells other than the diagonal line, which shows the wrong classification. Notable instances include:
1. One of the biggest similarities includes blending butter naan with chapati. Seeing that both are visually similar flatbreads, it may not come as a surprise the model has a hard time differentiating between them.
2. In the same practice, Chapati was misclassified as Butter Naan, which brings us to the next area of similarities: appearance.
3. The values of other classes like Chai, Samosa, and Jalebi do not get misclassified, showing the improvement of the model on such easily distinguishable classes.

Comparison with Other Models
In order to understand the level of competitiveness of the proposed SAFC_net model, it was compared to other established CNNs that are commonly used for image classification. These models selected for comparison include VGG-16 [35], ResNet-50 [36], Inception-V3 [37], and EfficientNet-B0 [38]. All these models were trained on the same dataset, and the same performance metrics were used for their comparison. The results are summarized in the table below.

Table 5. Comparison with other models

Model	Accuracy	Precision	Recall	F1-Score
VGG-16[35]	88.70%	89.20%	89.00%	89.10%
ResNet-50 [36]	90.60%	91.00%	90.90%	90.80%
Inception-V3 [37]	91.20%	91.60%	91.50%	91.50%
EfficientNet-B0 [38]	92.40%	92.50%	92.30%	92.40%
SAFC_net	98.08%	95.10%	95.00%	95.00%

The findings again testify that the SAFC_net model has achieved more accuracy than all the other models and is better on all the performance measures. Due to being a lightweight model and with the inclusion of the additional layers, it was able to classify visually similar dishes better than the other models.

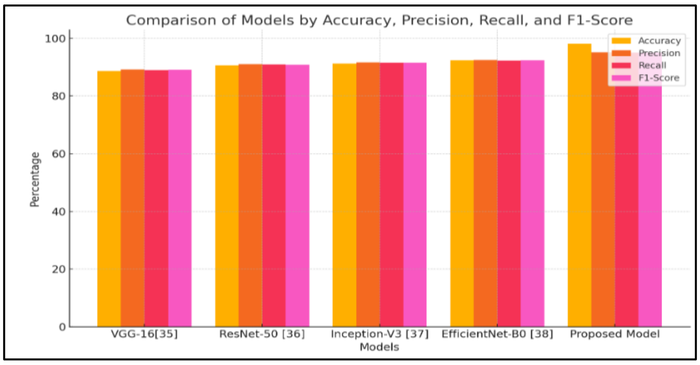

Figure 10. Visualization of Comparison

● VGG-16 [35] is a deeper network architecture with higher computation and memory needs than the comparable network architectures we've seen in this paper. Despite marking a certain level of success, the model was not so successful in differentiating the South Asian dishes in a much more detailed way.
● ResNet-50 [36], which utilized the residual learning concept, outperformed VGG-16 [35], but just like all the other works, it could not achieve as high accuracy as that of SAFC_net due to its higher complexity and high resource utilization.
● Comparing it with other state-of-the-art networks, Inception-V3 [37] and EfficientNet-B0 [38] were relatively close to SAFC_net, and among those two, the latter. This model's approach to scaling mentioned earlier that it is about balancing the depth and width, and the resolution gave it the performance. However, SAFC_net was designed to work well in real-time applications and thus provided the required accurate and efficient results.
These results show that SAFC_net is indeed an ideal model to be implemented on real-time applications like food mobile and restaurant automation, where both detection accuracy and computational complexity are extremely important.

Challenges in Classification
Some of the main difficulties in the classification of South Asian food items arose from the physical resemblance of many dishes. For instance, Butter Naan, Chapati, and Paratha—all the flatbreads in the examples—resemble each other in terms of the shape and the texture; therefore, the model cannot easily select this class relying upon visual characteristics only. It is usually served in the same manner, and visually there is not much difference between the two; normally it could be a matter of thickness, color, or texture, something that even the most sophisticated models cannot discern. This was due to low precision and recall for these categories, mainly because the model confused between these two similar dishes. Still, the introduction of the custom layers in the proposed SAFC_net architecture averted the issue by enhancing the model's performance in terms of recognizing particular textural details in the images.
Likewise, some food items like Momos and Samosa were somewhat tricky because of their varieties of presentation in the images. For instance, momos can come steamed or fried or with filling of a certain type, while samosas in shape and size come in different types. Most of the presentation styles observed in the images above made it challenging for the model to make generalized conclusions on images of the same dish. To tackle these issues, data augmentation techniques, including random rotations of the images, zooming, and image flipping, were incorporated in the training phase. These techniques assisted in making the model flexible in terms of orientation and scale of the images, hence enhancing the generalization capability of the model. However, I noticed that across most categories, the model has not performed badly, though some of its components presented challenges of one sort or another.
This study also shows that by improving and modifying the architecture of the SAFC_net model with layers and DA, it has a high accuracy in identifying the South Asian food items. The model considered for this work produced an average accuracy of 98 percent. 08%, better than other also prevalent CNN

architectures such as VGG16 [35], ResNet50 [36], and Efficient Net B0 [38]. Specifically, due to its light weight and high accuracy, it is particularly effective in real-time applications, for example, in mobile food recognition applications or in restaurant automation systems.
Theoretical and practical implications of the findings of this study are discussed in this section below. For example, since the SAFC_net model claims high accuracy, it can be implemented in dietary tracking applications where users can capture the image of the food and receive the identification and the nutrient information based on the result. This could be especially helpful in areas where most of the recipes are from South Asia since it will help users monitor what they eat. Also, the performance of the model has shown that it can be applied for restaurant automation systems. The model could be implemented into self-service kiosks where the customers take a picture of the meal they ordered or for billing purposes in mobile ordering applications where customers take a photo to identify the food they are ordering or paying for.
This would not only make the operation more efficient but would also increase customer satisfaction. Last but not least, based on the findings of the present study, the SAFC_net model can be enhanced in the future to classify food items from other cuisine types, assuming a larger dataset and more fine tuning have to be performed for individual cuisines. Further research studies can be aimed at the addition of other features related to food; some of these may include nutrition labels, ingredient identification, or even basic calorie counts to enhance the scope of classification in the model.

Conclusion

Therefore, the proposed SAFC_net architecture with custom layers and data augmentation has emerged as a very effective architecture for the classification of the SA food items. The model offers 98.02% accuracy and good performance of all the parameters for evaluation, thus making it suitable for real-time use in areas of dietary tracking, restaurant automation, and food recognition. There are difficulties due to the similarity of many dishes of South Asia's countries for a model recognizing them; however, the proposed architecture of a neural network and the process of its training let recognizing the dishes with a high accuracy.
The significance of this study is thus wide-ranging, with the potential benefits assignable to commercial and consumer markets. Given that more and more consumers are developing a desire to be introduced to ethnic recipes, efficient systems of food classification will be all the more required. While SAFC_net may seem to present a solution to this problem, it is one that requires more development and research and could well lead to the complete transformation of how people engage with food through technology in the future.

REFERENCES

1. Yanai, K. and Kawano, Y., 2015, June. 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.
2. Bossard, L., Guillaumin, M. and Van Gool, L., 2014. 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.
3. Şengür, A., Akbulut, Y. and Budak, Ü., 2019, September. Food image classification with deep features. In 2019 international artificial intelligence and data processing symposium (IDAP) (pp. 1-6). Ieee.
4. Arslan, B., Memiş, S., Sönmez, E.B. and Batur, O.Z., 2021. Fine-grained food classification methods on the UEC food-100 database. IEEE Transactions on Artificial Intelligence, 3(2), pp.238-243.
5. Zahisham, Z., Lee, C.P. and Lim, K.M., 2020, September. Food recognition with resnet-50. In 2020 IEEE 2nd international conference on artificial intelligence in engineering and technology (IICAIET) (pp. 1-5). IEEE.

6. Ittisoponpisan, S., Kaipan, C., Ruang-on, S., Thaiphan, R. and Songsri-in, K., 2022. Pushing the accuracy of thai food image classification with transfer learning. Engineering Journal, 26(10), pp.57-71.
7. Mahaveerakannan, R. and Devi, K.K., 2024, April. Food Classification by extracting the important features using VGGNet based Models in Precision Agriculture. In 2024 2nd International Conference on Networking and Communications (ICNWC) (pp. 1-7). IEEE.
8. Şengür, A., Akbulut, Y. and Budak, Ü., 2019, September. Food image classification with deep features. In 2019 international artificial intelligence and data processing symposium (IDAP) (pp. 1-6). Ieee.
9. Xu, B., He, X. and Qu, Z., 2021. Asian food image classification based on deep learning. Journal of Computer and Communications, 9(03), p.10.
10. Singla, A., Yuan, L. and Ebrahimi, T., 2016, October. 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).
11. Hnoohom, N. and Yuenyong, S., 2018, February. Thai fast food image classification using deep learning. In 2018 International ECTI northern section conference on electrical, electronics, computer and telecommunications engineering (ECTI-NCON) (pp. 116-119). IEEE.
12. Alahmari, S.S., Gardner, M.R. and Salem, T., 2024. Attention guided approach for food type and state recognition. Food and Bioproducts Processing.
13. Suddul, G. and Seguin, J.F.L., 2023. A comparative study of deep learning methods for food classification with images. Food and Humanity, 1, pp.800-808.
14. Srigurulekha, K. and Ramachandran, V., 2020, January. Food image recognition using CNN. In 2020 International Conference on Computer Communication and Informatics (ICCCI) (pp. 1-7). IEEE.
15. Siemon, M.S., Shihavuddin, A.S.M. and Ravn-Haren, G., 2021. Sequential transfer learning based on hierarchical clustering for improved performance in deep learning based food segmentation. Scientific Reports, 11(1), p.813.
16. Hnoohom, N. and Yuenyong, S., 2018, February. Thai fast food image classification using deep learning. In 2018 International ECTI northern section conference on electrical, electronics, computer and telecommunications engineering (ECTI-NCON) (pp. 116-119). IEEE.
17. Rajayogi, J.R., Manjunath, G. and Shobha, G., 2019, December. Indian food image classification with transfer learning. In 2019 4th International Conference on Computational Systems and Information Technology for Sustainable Solution (CSITSS) (pp. 1-4). IEEE.
18. Yadav, S. and Chand, S., 2021, March. 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.
19. Iftikhar, A., Bilal, A., Rakha, B. A., & Akhter, S. (2025). Evaluating the Cryoprotective Effects of Butylated Hydroxytoluene on Semen Quality Parameters of Phasianus colchicus. Journal of Agriculture and Biology, 3(1), 1.
20. Ameer, S., Mehmood, M. A., Khan, W. A., Arsalan, M., Hassan, N. U., Noor, A., Bukhari, S. S. A. S., Tahir, A., & Manazir, A. (2024a, September 24). Detection of lung cancer through computed tomographic images using deep learning models. Journal of Population Therapeutics and Clinical Pharmacology, 31(9),
21. Chun, M., Jeong, H., Lee, H., Yoo, T. and Jung, H., 2022. Development of Korean food image classification model using public food image dataset and deep learning methods. IEEE Access, 10, pp.128732-128741.
22. Bilal, A., Bibi, R., Umar, M., Sajjad, A., Kharal, S., Noor, E., ... & Munir, A. (2025). The relationship between obesity and breast cancer among women of Punjab, Pakistan. The Research of Medical Science Review, 3(2), 668-84.
23. Shahin, F., Ishfaq, A., Asif, I., Bilal, A., Masih, S., Ashraf, T., ... & Ishfaq, R. (2024). CRISPR-Cas Innovative Strategies for Combating Viral Infections and Enhancing Diagnostic

Technologies: CRISPR-Cas in Viral Diagnostics and Therapeutics. Journal of Health and Rehabilitation Research, 4(3), 1-4.
24. Ahmad, R. Z., Khan, M. S., Bilal, A., Ali, U., & Sattar, R. Z. (2023). Effect of locus of control and depression among young adults in Multan (Pakistan). Journal of Asian Development Studies, 12(4), 684-692.
25. Bilal, A. (2021[a]). Rabies is a zoonotic disease: a literature review. Occup. Med. Health Aff, 9(2).
26. Bilal, A. (2021[b]). Clinical diagnosis and treatment of absence seizures: Case study. MAR Ophthalmology, 2(1)
27. Sattar, R. Z., Bilal, A., Bashir, S., Iftikhar, A., & Yaqoob, I. (2024). Embryotoxicity of fluconazole on developing chick embryos. The Journal of Basic and Applied Zoology, 85(1), 8.
28. Chun, M., Jeong, H., Lee, H., Yoo, T. and Jung, H., 2022. Development of Korean food image classification model using public food image dataset and deep learning methods. IEEE Access, 10, pp.128732-128741.
29. Gao, X., Xiao, Z. and Deng, Z., 2024. High accuracy food image classification via vision transformer with data augmentation and feature augmentation. Journal of Food Engineering, 365, p.111833.
30. Pan, X., He, J. and Zhu, F., 2024. FMiFood: Multi-modal Contrastive Learning for Food Image Classification. arXiv preprint arXiv:2408.03922.
31. Raghavan, S., He, J. and Zhu, F., 2024. Online class-incremental learning for real-world food image classification. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (pp. 8195-8204).
32. Oduru, T., Jordan, A. and Park, A., 2022. Healthy vs. unhealthy food images: Image classification of twitter images. International Journal of Environmental Research and Public Health, 19(2), p.923.
33. R. Patel and N. Sharma, "Kaathi rolls: A fusion of Indian and international flavors," *International Journal of Street Food Studies*, vol. 7, no. 2, pp. 221-229, 2023.
34. Shah, B. and Bhavsar, H., 2024. Depth-restricted convolutional neural network—A model for Gujarati food image classification. The Visual Computer, 40(3), pp.1931-1946.
35. Farinella, G.M., Allegra, D., Moltisanti, M., Stanco, F. and Battiato, S., 2016. Retrieval and classification of food images. Computers in biology and medicine, 77, pp.23-39.
36. Medus, L.D., Saban, M., Francés-Víllora, J.V., Bataller-Mompeán, M. and Rosado-Muñoz, A., 2021. Hyperspectral image classification using CNN: Application to industrial food packaging. Food Control, 125, p.107962.
37. Ciocca, G., Napoletano, P. and Schettini, R., 2018. CNN-based features for retrieval and classification of food images. Computer Vision and Image Understanding, 176, pp.70-77.
38. Fahira, P.K., Rahmadhani, Z.P., Mursanto, P., Wibisono, A. and Wisesa, H.A., 2020, November. Classical machine learning classification for javanese traditional food image. In 2020 4th International Conference on Informatics and Computational Sciences (ICICoS) (pp. 1-5). IEEE.
39. Lo, F.P.W., Sun, Y., Qiu, J. and Lo, B., 2020. Image-based food classification and volume estimation for dietary assessment: A review. IEEE journal of biomedical and health informatics, 24(7), pp.1926-1939.
40. Kazi, A. and Panda, S.P., 2022. Determining the freshness of fruits in the food industry by image classification using transfer learning. Multimedia Tools and Applications, 81(6), pp.7611-7624.
41. Ireland, J.D. and Møller, A., 2000. Review of international food classification and description. Journal of food composition and analysis, 13(4), pp.529-538.
42. Petrus, R.R., do Amaral Sobral, P.J., Tadini, C.C. and Gonçalves, C.B., 2021. The NOVA classification system: A critical perspective in food science. Trends in Food Science & Technology, 116, pp.603-608.
43. Zheng, W., Fu, X. and Ying, Y., 2014. Spectroscopy-based food classification with extreme learning machine. Chemometrics and Intelligent Laboratory Systems, 139, pp.42-47.
44. Cianferoni, A. and Spergel, J.M., 2009. Food allergy: review, classification and diagnosis. Allergology International, 58(4), pp.457-466.

45. Ameer, S., Khan, M. T. A., Fahad, S., Mumtaz, A., Fatima, H. S., Rahman, T. U., Bukhari, S. S. A., Sajjad, A., Hassan, S. A., & Aziz, S. (2024b, September 25). A machine learning-based approach for the detection and segmentation of retinal vessels in ophthalmic disease analysis. Journal of Population Therapeutics and Clinical Pharmacology, 31(9), 13. RESEARCHTRENTZ Acad Publ Education Services.
46. Cianferoni, A. and Spergel, J.M., 2009. Food allergy: review, classification and diagnosis. Allergology International, 58(4), pp.457-466.
47. Päßler, S., Wolff, M. and Fischer, W.J., 2012. Food intake monitoring: an acoustical approach to automated food intake activity detection and classification of consumed food. Physiological measurement, 33(6), p.1073.
48. Marrón-Ponce, J.A., Sánchez-Pimienta, T.G., da Costa Louzada, M.L. and Batis, C., 2018. Energy contribution of NOVA food groups and sociodemographic determinants of ultra-processed food consumption in the Mexican population. Public health nutrition, 21(1), pp.87-93..
49. Asp, N.G., 1994. Nutritional classification and analysis of food carbohydrates. The American journal of clinical nutrition, 59(3), pp.679S-681S.
50. Goldfein, K.R. and Slavin, J.L., 2015. Why sugar is added to food: food science 101. Comprehensive Reviews in Food Science and Food Safety, 14(5), pp.644-656.
51. Majumdar, D., Koch, P.A., Lee, H., Contento, I.R., Islas-Ramos, A.D.L. and Fu, D., 2013. “Creature-101”: a serious game to promote energy balance-related behaviors among middle school adolescents. GAMES FOR HEALTH: Research, Development, and Clinical Applications, 2(5), pp.280-290.
52. Borzekowski, D.L. and Robinson, T.N., 2001. The 30-second effect: an experiment revealing the impact of television commercials on food preferences of preschoolers. Journal of the American Dietetic Association, 101(1), pp.42-46.
53. Martin, C.K., Han, H., Coulon, S.M., Allen, H.R., Champagne, C.M. and Anton, S.D., 2008. A novel method to remotely measure food intake of free-living individuals in real time: the remote food photography method. British Journal of Nutrition, 101(3), pp.446-456.
54. Medeiros, L.C., Kendall, P., Hillers, V., Chen, G. and DiMascola, S., 2001. Identification and classification of consumer food-handling behaviors for food safety education. Journal of the American Dietetic Association, 101(11), pp.1326-1339.
55. Z. Zahisham, Z. Zharfan, C.-P. Lee, and K. Lim, “Food recognition with ResNet-50,” Proc. IEEE Int. Conf. Artif. Intell. Eng. Technol. (IICAIET),2020, pp.1–5.
56. M. S. Zaoad, M. M. R. Mannan, A. B. Mandol, M. Rahman, M. A. Islam, and M. M. Rahman, “An attention-based hybrid deep learning approach for Bengali video captioning,” J. King Saud Univ. - Comput. Inf. Sci., vol. 35, no. 1, pp. 257–269, 2023, doi: 10.1016/j.jksuci.2022.11.015.