Modeling And Optimizing Gujarati Food Classification Through Deep Learning

A SYNOPSIS

Submitted in partial fulfillment of the

requirements for the award of the

degree of

DOCTOR OF PHILOSOPHY

in

COMPUTER SCIENCE & ENGINEERING

By

BHOOMI SANKET SHAH

Under Guidance of

DR. HETAL BHAVSAR

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

FACULTY OF TECHNOLOGY & ENGINEERING

THE MAHARAJA SAYAJIRAO UNIVERSITY OF BARODA

VADODARA-390002 (INDIA)

MAY 2023

ABSTRACT

People today pay close attention to their diet and overall wellness. Consuming foods with a lot of calories can be unhealthy and can contribute to major medical disorders like heart disease, chronic diseases, and hypertension. Keeping track of food intake is essential for a successful dietary assessment system. Accurate food identification and calorie counting techniques can aid individuals in their fight against obesity, which is the root of all weight problems. To avoid obesity, it is very necessary to control the amount of food taken. For that, the first step is to identify the food and the second is to count the calories.

Accurate Identification of food is very important as based on which calories can be counted and that helps to reduce the risk of serious health conditions. This research work is mainly concentrated on food identification.

People in Gujarat are very fond of eating different varieties of food. Gujarati food is very vivid in taste and texture due to its rich heritage and history. As most Gujarati food is oily and sweet, consuming this food daily in larger portions results in obesity. In order to remain healthy, it is very necessary to keep track of food intake for which correct food identification is the first step to control the diet. A lot of work has been done on other kinds of food but no previous efforts have been made to classify Gujarati food. Therefore, being Gujarati, this study focuses on classification of Gujarati food images.

This research work created a new dataset called "Traditional Gujarati Food Images Dataset (TGFD)". To start with initially dataset has been created with five popular food items in Gujarat namely Dhokla, Handvo, Khakhra, Khandvi, and Patra consisting 1764 images. The dataset is created by the images collected from the internet, pictures taken using mobile phones, and real images captured by visiting different restaurants, it contains a lot of noise. The food image quality parameters are primarily resolution and impulse noise. It is very necessary to remove the noise and unwanted objects from images so the food can be correctly identified. Image pre-processing techniques are helpful to achieve this goal.

In order to remove noise and improve the quality of the images. An algorithm called ISMF (Improved Selective Median Filter) has been developed to pre-process the images from TGFD. The output of this algorithm can be used further to process food images in order to classify them correctly.

To see the effect of Gujarati food images on existing pre-built models Simulation has been done on five models namely VGG16, VGG19, Resnet50, Inceptionv3, and Alexnet. For TGFD the highest accuracy of 81.58 % is obtained with Inceptionv3. To further improve the classification accuracy, Transfer Learning has been implemented on all five models. For more precision, fine-

tuning has been applied to all five models. The results of the simulation, Transfer learning, and Fine-tuning on all models have been compared in terms of accuracy.

By doing simulation and implementing Transfer learning and Fine tuning it has been observed that factors that directly impact the model's accuracy include the number of convolutional layers, the number of neurons in fully connected layers, the number of filters, and the size of the filters. Taking these parameters into consideration this research developed a model for Gujarati food image classification called "Depth Restricted Convolutional Neural Network (DRCNN)" in order to increase accuracy and efficiency. The performance of the proposed model is compared with existing models in terms of Recall, Precision, and F1-score. With DRCNN the classification accuracy has been improved from 81.58% to 95.48%. The proposed model size is 48 times smaller than the Inception v3 model and hence takes half the time to run compared to other pre-built models with millions of parameters.

The performance of DRCNN has been tested in two ways to check its effectiveness and versatility. In the first test case, the DRCNN is tested by increasing Gujarati food image classes. TGFD has been expanded from 5 to 10,12,15 and up to 20 Gujarati food classes. The performance of DRCNN is improving with a large number of images showing the versatility of the model. In the second case, DRCNN is tested against different food datasets. DRCNN gives outstanding accuracy for any type of food item.

In order to calculate the computational complexity of the model, computational studies have been conducted to determine the factors that affect the CNN model's performance, the time it takes for each layer to run, and how this affects the model's overall performance.

To prove it practically the time complexity of eight different models has been discovered, by varying the size of the filters, number of convolutional layers, number of filters, number of fully connected layers, and kernel size. The results show that factors such as an optimizer, batch size, filter, and neurons significantly impact the time required by the model. The model's convolutional layers, max pool, and fully connected layers affect the model's performance. It has been found that 90% of computational time is taken by convolutional layers and 5-10% of the time is taken by fully connected layers.

Table of Contents

Sr. No.	Topic	Page No.
i	Abstract	2
ii	Table of Contents	5
iii	List of Figures	6
iv	List of Tables	6
1	**INTRODUCTION**	7
	1.1 Overview of Deep Learning	7
	1.2 Introduction to Image Classification	8
	1.2.1 Introduction to Food Image Classification	8
	1.2.2 CNN	9
	1.2.3 Transfer Learning	9
	1.2.4 Fine Tuning	10
	1.3 Motivation For This Work	10
	1.4 Problem Statement, Objectives, Research Contributions	10
	1.4.1 Problem Statement	11
	1.4.2 Objectives	11
	1.4.3 Research Contributions	11
2	**LITERATURE STUDY**	15
	2.1 Study of Various Preprocess Algorithms	15
	2.2 Food Classification	16
	2.3 Time Complexity	21

3	**TGFD – Traditional Gujarati Food Dataset**	24
	3.1 Creating TGFD	24
	3.2 Data Augmentation	24
4	***ISMF* – An Improved Selective Median Filter for Preprocessing**	26
	4.1 Median Filter	26
	4.2 Improved Selective Median Filter	26
5	**Optimization Through Transfer Learning**	30
	5.1 Simulation	30
	5.2 Transfer Learning	30
6	**Optimization Through Hyperparameter Tuning**	32
7	**DRCNN – Depth Restricted Convolution Neural Network**	34
	7.1 Hyperparameter Selection for *DRCNN* Using Empirical Analysis	34
	7.2 Experiments and Results	37
	7.3 Testing *DRCNN*	39
8	**Time Complexity of CNN models**	41
9	**Conclusions**	45
10	**Publications**	47
11	**References**	48

List of Figures

Figure Number	Figure Caption	Page No.
1.	Research Flow	12
2.	Sample Food Items from Traditional Gujarati Food Dataset	20
3.	Algorithm ISMF	22
4.	Comparison Graph of PSNR And MSE For Different Noise for Food Image	23
5.	Applying Filters to Noise Image	24
6.	Transfer Learning	26
7.	Fine-Tuning	27
8.	The Architecture of The Proposed Model DRCNN	29
9.	Algorithm Of DRCNN	31
10.	Accuracy Curve For DRCNN	33
11.	Loss Curve For DRCNN	33
12.	Time Complexity of Each Model Based on Chosen Parameters	39

List of Tables

Table Number	Table Caption	Page No.
1.	Different Method of Food Classification	14
2.	Comparison Of Median Filter and ISMF on TGFD	23
3.	Classification Accuracy of Models in Simulation	25
4.	Classification Accuracy of Models in Transfer Learning	26
5.	Classification Accuracy of Models in Fine-Tuning	28
6.	Comparison of DRCNN With Fine-Tuned Existing Models	32
7.	Performance of Models on TGFD Based on Evaluation Parameters	32
8.	Comparison of Fine-Tune Inceptionv3 And DRCNN	34
9.	Result of DRCNN For Different Datasets	34
10.	CNN Architecture with Parameters	37

1 INTRODUCTION

This chapter briefly introduces deep learning and food image classification. It also tells the motive behind this work, the problem statement, and the research contributions of this work.

Food is an essential element of everyone's life. Obesity is increasing at a higher rate day by day risking the lives of many people. According to World Health Organization in 2020, an estimated 39 million children under the age of 5 years were overweight or obese. In 2016, 39% of adults aged 18 years and over (39% of men and 40% of women) were overweight. Most of the world's population lives in countries where overweight and obese kill more people than underweight. Obesity can be preventable by controlling the amount of food intake. It is possible by first identifying the consumed food and then measuring the calories in order to prevent obesity and remain healthy. For the entire process, the first important step is food identification. This research work mainly focuses on the correct identification of food.

1.1 INTRODUCTION TO FOOD IMAGE CLASSIFICATION

Food image identification plays a very important role in today's era. The food domain can be divided into two parts. The first is to recognize food items, and the second is to estimate the calorie. Accurate methods for food identification and calorie estimation can help people to fight against obesity which is the cause of being overweight.

With the use of food image classification, people can identify the food and with calorie estimation amount of food consumed. It helps to reduce the risk of serious health conditions like hypertension, chronic diseases, and heart disease [1]. Classification of food images is a very challenging task as the dataset of food images is not linear. The nature of food is very diverse, which makes the food image classification task more challenging [2].

According to research Deep learning methods for image classification gives more accurate and efficient results as compared to traditional methods. Many works have been done to classify different types of food like Western food, Japanese food, Fast food items, Chinese food, and south Indian food but no work has been done to classify Gujarati food [3][5] [7-9]. There are so many varieties of Gujarati food. To evaluate the dietary aptitudes of people from various ethnicities, the classification of their traditional foods makes a huge impact. Being Gujarati, it steered us to do a detailed study in the field of Gujarati food domain through deep learning.

This work uses deep learning as a tool for Gujarati food classification to achieve impressive results.

1.2 OVERVIEW OF DEEP LEARNING

Deep learning is an advanced technology for image processing, speech recognition, object detection, and food science and engineering [1]. It works with artificial neural networks, which are designed to imitate how humans think and learn. Unlike Machine Learning, in deep learning, basic details about the data need to be given, that process through many layers, and the computer trains to recognize the patterns on its own. The availability of a large number of datasets and high Processing GPU makes deep learning techniques very successful [3]. Many recent articles have been surveyed in the food domain including food recognition for this work [11-26].

Deep learning techniques can be divided into mainly three categories. 1. Supervised learning 2. Unsupervised learning 3. Reinforcement learning. Food identification with deep learning belongs to supervised learning [2].

Deep learning networks are mathematical models that work like human brains. This mathematical model is created in form of a neural network that consists of neurons. The neural network is divided into three major layers input layer (the first layer of the neural network), the hidden layer (all middle layers of the neural network), and the output layer (the last layer of the neural network.). The most popular deep learning networks for supervised learning can be described as follows.

1.2.1 CNN: Convolutional Neural Network

Convolutional Neural Network (CNN) is the main category of Deep Learning networks to do the image recognition and image classification. CNN takes input, processes it and classifies it under predefined categories. The main advantages of CNN are, parameter sharing, sparse interactions, and equivalent representations [2]. A layer in the neural network is nothing but a collection of neurons that takes an input and provides an output. The input of each of these neurons is processed through the activation function assigned to the neurons. Many successful research works have been done on food object recognition through CNN proving that CNN gives the best result in terms of accuracy and error rate for object recognition [16-22].

1.2.2 RNN: Recurrent Neural Network

It is a very popular deep-learning model that uses recursion techniques to build models. RNN saves the output of the current layer which will be input to the next layer. It can memorize previous inputs due to its internal memory, and hence it is especially used as a language model. It is mostly used in natural language processing and speech processing [18, 23], text analysis, and machine translation [21-22].

1.2.3 RvNN: Recursive Neural Network

RvNN can handle the inputs of different modalities [18]. RvNN has been especially successful in NLP. RvNN separates the image into different segments and forms a syntactic tree [24].

For image classification, most of the research work has used CNN and RNN [14–22].

According to studies It has been found that among all the deep learning networks CNN is most suited for image classification and gives the best classification accuracy by reducing the error rate for object recognition. So, this research work uses Convolution Neural Networks to classify Gujarati Food items.

1.2.4 Transfer Learning

In transfer learning, a pre-trained model that has already been trained on some dataset can be used for another related task [4]. The benefit of transfer learning is that instead of developing and training everything from scratch, the weight of the pre-trained model can be freezed and only the custom layers need to be retrained [5]. This costs, ultimately, the time taken by the model to train, and it is a very good solution to the problem of having smaller datasets [40-41].

1.2.5 Fine Tuning

Fine-tuning is more flexible as compared to transfer learning as the feature extraction part of the model along with the classification part can be changed [10]. It is possible to fine-tune all the layers, but it is always good practice to fix the upper layers as they contain features that are more generic in nature [5]. In this technique, some of the final layers of the freezed model are initially unfreezed, by adding some additional layers, followed by training of both the unfreezed layers and additional custom layers at once.

1.3 MOTIVATION FOR THIS WORK

The traditional food of a country resembles some part of its culture, too. One can get an idea of a country's food habits and food culture by virtue of its traditional food processing and preservation method. The people of Gujarat are known for their hospitality and most of this fame is earned by the unique tests of the traditional foods here. Until now, more research on western-style foods has been conducted through deep learning and food classification. This concept will open doors of acceptance for Gujarati Food or cuisine at a global level. As none of the work has been done till now for Gujarati Food Classification this research work focuses on classifying Gujarati Food Images through Deep Learning. With this work, the spread of true information about the Gujarati food image globally will be enhanced through the internet and social media. It will be helpful to food bloggers and foodies to understand Gujarati cuisine.

As most Gujarati Food is naturally oily and sweet, consuming this food daily in larger portions results in obesity. There are so many varieties of the same Gujarati food item that can be varied by different cooking methods, camera quality, illumination, and presentation. Also, Gujarati Food items do not have regular shapes in general. Hence for Gujarati Food, the crucial aspect is to identify the food correctly. This motivated to design a new model that can classify Gujarati Food Accurately and reduce time compared to all existing models.

1.4 PROBLEM STATEMENT, OBJECTIVES, SCOPE, AND RESEARCH CONTRIBUTIONS

1.4.1 Problem Statement

The prime idea of this research is to design and develop a model for Gujarati Food image classification with improved accuracy and performance by making the network lightweight.

1.4.2 Objectives

- To design and, develop a model(s) that can recognize Gujarati Food.
- To make the algorithm lighter in terms of memory and time both so as to be used for handheld devices.
- To give better performance with a greater number of food classes or a dataset of different food items.
- To evaluate the model performance based on parameters such as optimizers, activation function, epochs, batch size, and learning rate.

1.4.3 Scope

- The scope of research is to study theoretical and empirical studies of existing models of Deep Learning.

- Detailed study of Deep Learning methods and compare the proposed method with existing algorithms.
- To preserve or increase accuracy as compared to other existing models.
- Understand and empirical study on image classification with different parameters such as the texture of food, color, and size and with different types of food items.
- Understanding research issues with possible improvements.
- Classify Gujarati Food Accurately for 5 to 20 Food Classes.

1.4.4 Research Contributions

- A new dataset named "Traditional Gujarati Food Images Dataset (TGFD)" has been created. The dataset contains images belonging to five food classes namely Dhokla, Handvo, Khakhra, Khandvi, and Patra which are famous food items in Gujarat. There are a total of 1764 images with at least 300 images per class in the dataset, which are divided into training, validation, and testing with 70%, 20%, and 10% respectively.
- Pre-processing is essential for improving the quality of an image. An algorithm named ISMF (Improved selective median filter) has been developed for the removal of impulse noise in food images to improve classification accuracy. Using ISMF the mean square error has been decreased, and the performance of automatic food classification has been improved. ISMF performs better than the median filter in terms of detail preservation and image denoising.
- Implemented Transfer Learning on models, namely VGG16, VGG19, Resnet50, Inceptionv3, and Alexnet. By changing the classification layer of the model and freezing the rest of the layers. In addition to that, a flattened layer followed by a fully connected layer with a Softmax activation function has been added to the model. By implementing transfer learning the validation accuracy is increased by 5% from simulation. The highest accuracy achieved is 86.22% by the Inceptionv3 model. The model contains 256,005 trainable parameters.
- To further increase the classification accuracy Fine-tuning has been implemented on all models, namely VGG16, VGG19, Resnet50, Inceptionv3, and Alexnet. A new dense layer with a dropout of 0.5 has been added. In the last layer, the Softmax activation function has been used. By implementing Fine-tuning, the validation accuracy is increased by 8% from simulation. The Inceptionv3 model achieved the best classification accuracy of 89.36%. The model contains 22,024,357 trainable parameters.
- A model named "Depth Restricted Convolutional Neural Network (DRCNN)" has been developed for Gujarati Food Image Classification. The parameters considered for the model are the number of convolutional layers, the number of neurons in fully connected layers, the number of filters, and the filter size, which directly affect the model's accuracy. The DRCNN model achieves a remarkable classification accuracy of 95.48%,

which is more than all the fine-tuned models. The DRCNN model contains 482069 parameters, which is 48 times less than the Inceptionv3 model, which gives the highest classification accuracy after fine-tuning among all the pre-trained models. The performance is also measured in terms of F1 Score, Precision, and Sensitivity. The validation loss of DRCNN is only 0.8041 which is remarkable as compared to Transfer Learning and Fine tuning. Also, the DRCNN model takes only 30 minutes to run on the NVIDIA GPU GeForce GTX 1650, which is very low as compared to other pre-built models, which take 50 to 60 minutes to run as they have millions of parameters.

- DRCNN is tested in two ways to check its effectiveness and performance. In the first test case, the DRCNN is run for a higher number of Gujarati food image classes starting from 10 to 15 to 20 Gujarati food items. In the second case, the DRCNN is run for different food datasets to see its performance. The performance of DRCNN is improving with a large number of image classes and gives outstanding performance for any type of food item proving the versatility of the model.
- Analysed and Derived the Time Complexity of the Proposed Model and CNN Model. Eight different models were tested varying by the size of filters, number of convolutional layers, number of filters, number of fully connected layers, and kernel size. The result shows that factors like an optimizer, batch size, filter, and neurons greatly impact the time taken by the model. From this, it has been derived that the convolutional layers, max pool, and fully connected layers directly affect the performance of the model. And since DRCNN has a minimum number of convolutions and fully connected layers the model's computational complexity is less than most of the existing models.

As a concluding remark, the overall research work is shown in below Fig. 1.

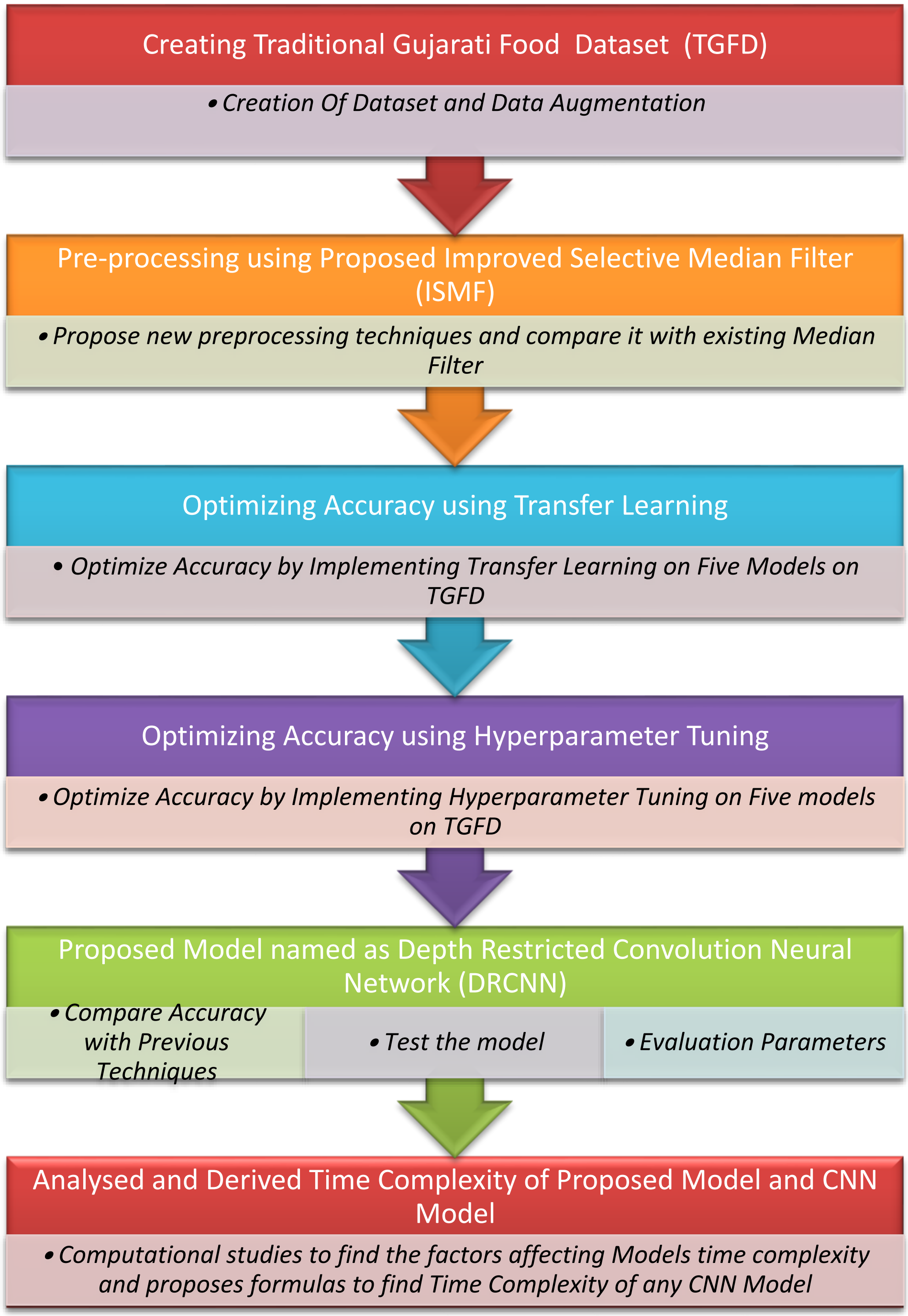


Fig 1. Research Flow

2. LITERATURE STUDY

For Food Image Classification Deep Learning has become famous for its impressive results. It can learn features automatically just like a human does. It is probably the best approach in cases when we don't have enough pre-defined features. This section studied all the recent articles on preprocessing, food image classification considering the type of food, different methods of classification, different types of networks, and time complexity of CNN model.

2.1 STUDY OF VARIOUS PREPROCESSING ALGORITHM

Image processing is an approach to strengthen the images received from a camera or satellite. The process of removing noise from an image seems to be easy, but in reality, it is complex in nature, as it involves considerable time, technology, and resources by the editor.
An overview of various image pre-processing techniques for a wide range of medical imagery has been presented by P. Vasuki et al. [6]. The pre-processing techniques for X-rays, fundus images, and mammograms have been discussed and clearly state that preprocessing is the mandatory step before processing any image.
A comparative study on various techniques of pre-processing for image fusion based on CNN has been done by Jyoti et al. Image fusion is a technique that gives a different focus to a single image [7]. Three filters have been used on three different datasets: medical, color multi-focus dataset, and infrared visual. The results showed that median filters give the best classification accuracy on any dataset compared to other filters.
A study on the median filter with various variants to discard the salt and pepper noise from grayscale images has been presented by Anwar et al. A comparison has been done on filters based on computational complexity and performance [8]. The conventional median filter is good for low noise but fails to preserve edges.
A dataset for breast cancer has been created by Sami et al using pre-processing the images [9]. The main idea of this work is to create a dataset so the operational time for the used network can be saved and the accuracy can be improved. The method has mainly three parts: the first is removing the background, the second is the removal of the pectoral muscle and the third is image enhancements. The proposed method can remove 100% of the image background.

2.2 FOOD CLASSIFICATION

Many research works have been done on food image recognition that has used convolution neural networks with different combined approaches and different datasets. Table 1 describes different methods for food classifications.

Table 1: Different Methods of Food Classification

Sr. No.	Reference	Approach	Dataset	Top1 Accuracy	Top5 Accuracy	Remarks
1	A food recognition system for diabetic patients based on an optimized bag-of-features model [11]	Two ANN models were used. one is without hidden layer and another is with one hidden layer. Classification was done using three supervised methods SVM, ANN, and Random Forests (RF)	Diabetes	75.0%	-	Some of Examples of misclassification of images has been presented.
2	Food-101-mining discriminative components with random forest in computer vision [12]	Used the approach called Random Forest Discriminant Components (rfdc) and compare it with various other methods, also have introduced Food-101 Dataset having 101000 images	Food-101	56.4%	-	A novel large-scale benchmark dataset has been introduced for the recognition of food.
3	Food Image Recognition with Deep Convolution	Used deep convolutional neural network with Fisher	UEC-Food-100	72.3	92.0	DCNN features can boost classification performance by integrating it with

	al Features [13]	Vector with HOG and color patches.				conventional features.
4	Food Image Recognition Using Deep Convolutional Network with Pretraining and Fine-Tuning [14]	Used deep convolutional neural network for food photo recognition task in the ImageNet, and have implemented this combination on Twitter photo data. Achieved high level of accuracy proving DCNN gives best result on large scale image data.	UECFood-100	78.7%	-	In addition to high classification accuracy, DCNN was very suitable for large-scale image data, since it takes only 0.03 seconds to classify one food photo with GPU.
			UEC-Food-256	67.57	89.0	
5	Food recognition for dietary assessment using deep convolutional neural networks [15]	Used a deep convolution neural network with 6 layers, used to classify food image Patches. Experiments have achieved attractive result.	Own database with 573 food items.	84.90	-	The presented results are preliminary. future work should include a more thorough investigation on the optimal architecture as well as the training parameters of the network.
6	Im2calories: towards an automated mobile vision food diary [16]	Used CNN based classifier to estimate the food size and labels and apply this method to a dataset of images from 23 different restaurant.	Food-101	79.0%	-	The proposed approach is able to tackle some of the problems in estimating calories from food but there is lot of scope for more work in future. The approach does not accurately measure the calories.
			Food201 segmented	76.0	-	
			Menu-Match	81.4	-	
7	Food image recognition using very deep convolution	Used deep learning approach for the classification and fine-tuned the	UEC-Food-100	81.5	97.3	One important result of our study is showing that fine-tuning a pre-trained network can achieve good results in
			UEC-Food-256	76.2	92.6	
			Food-101	88.3	96.9	

	al networks [17]	image recognition architecture Inception.				a reasonable time
8	Deep-Based Ingredient Recognition for Cooking Recipe Retrieval [18]	Proposed deep architecture namely Arch-D that defines relationship between food and ingredients label through multitask learning.	UEC-Food-100	82.1	97.3	The current approach basically cannot distinguish recipes for dishes that have the same ingredients but appear visually different mainly due to different cooking methods. In addition, our multi-task model could not deal with ingredients (e.g., honey, soy40 bean oil) that are not observable or visible from dishes. Secondly, while this paper considers the zero-shot problem of unknown food categories, how to couple this problem together with unseen ingredients remains unclear.
			VIREO	82.1	95.9	
9	Deep food: deep learning-based food image recognition for computer-aided dietary assessment. [19]	Proposed new algorithm based on CNN and achieved impressive result on two datasets namely Food-101 and UEC-FOOD-256	Food-101	77.4%	93.7%	To improve performance of the algorithms a real word mobile devices and cloud computing-based system is needed to enhance the accuracy of current measurements of dietary caloric intake.
			UEC-Food-256	54.7	81.5	
10	Food Calorie Measureme	Used the Graph cut method and uses Deep	Own database with	99.0	-	Mixed food portion images have not been considered.

	nt Using Deep Learning Neural Network [20]	convolution Neural Network to classify food images and have achieved remarkable accuracy for a single food image.	10000 high resolution images			
11	Food recognition: a new dataset, experiments, and results [21]	Proposed a new dataset that contains 1,027 canteen trays for a total of 3,616 food instances belonging to 73 food classes. The food on the tray images have been manually segmented using carefully drawn polygonal boundaries.	UNIMINB2016	78.3	-	The dataset designing by an automatic tray analysis pipeline that takes a tray image as input, finds the regions of interest and predicts for each region the corresponding food class.
12	FoodNet: recognizing foods using an ensemble of deep networks [22]	Proposed a multilayered ensemble of networks that take advantage of three deep CNN fine-tined subnetworks. also proposed a new Indian Food dataset.	Indian Food database	73.50	94.40	The experimental results show that our proposed ensemble net approach outperforms consistently all other current state-of-the-art methodologies for all the ranks in both the databases.
			ETH Food-101	72.12	91.61	
13	Food recognition using a fusion of classifiers based on CNNs [23]	Proposed a CNNs Fusion approach based on the concepts of decision templates and decision profiles and their similarity that improves the	Food-101 Food-11	-	-	combination of multiple classifiers based on different convolutional models that complement each other hence improving performance

		classification performance with respect to using CNN models separately.				
14	Exploring food detection using CNNs [24]	Proposed a model that uses GoogleNet for feature extraction, PCA for feature selection, and SVM for classification.	FCD	98.81 %		-
			Ragusa DS	95.78 %		
15	Classification of food images through interactive image segmentation [25]	Proposed a segmentation algorithm based on random forest and has used Boundary Detection & Filling methods. Also compared the proposed algorithm with three existing methods.	Food-101	90.5	-	The proposed method is validated by a four-fold cross-validation technique on a publicly available Food 101 dataset.
16	Food image recognition by using convolutional neural networks [26]	Developed a model with Five-layer CNN, and the first ever combining bag-of-features model with support vector machine to achieve a high level of accuracy.	ImageNet	74	-	Due to limited training data, the CNN model suffered from overfitting. The issue was addressed by expanding the training data through various affine transformations

Apart from these [4-5][10] [40-41], many researchers have tried to classify different types of food items with different techniques like transfer learning and hyperparameter tuning. The following section discusses the same.

Rajayogi et al. [27] al have implemented different transfer learning techniques on the Indian food dataset. Unlike the traditional methods of building a model from the scratch, pre-trained models are used in this project which saves computation time and cost and also has given better results. The Indian food dataset of 20 classes with 500 images in each class is used for training and validation. The models used are IncceptionV3, VGG16, VGG19, and ResNet. After experimentation, it was found that Google InceptionV3 outperformed other models with an accuracy of 87.9% and a loss rate of 0.5893.

Ashutosh Singla et al. [28] have implemented transfer learning on the pre-Trained GoogLeNet Model for food/nonfood classification and food recognition on the Food-5K dataset. The experimental results show an overall accuracy of 99.2% on food/non-food image classification and 83.6% on food categorization.

Raheel Siddiqi [29] has shown the effectiveness of transfer learning and fine-tuning on Inception v3 and VGG16 for automated fruit image classification. Using Transfer learning on VGG16 best classification accuracy of 99.27% is achieved. Fine-tuning using VGG16 has produced 98.01% classification accuracy while transfer learning using Inception v3 has produced 98.1% classification accuracy.

2.3 TIME COMPLEXITY IN CNN

A convolutional neural network is a combination of convolutional layers, fully connected layers, and pooling layers. The number of parameters at convolutional layers and fully connected layers are known as learnable parameters, and layers with such parameters are known as learnable layers, which significantly affect the overall performance of the network. Till now, many researchers have tried to find the relationship between different hyperparameters of CNN through empirical research.

Shiv Ram Dubey et al. try to find how fully connected layers and the dataset are related through a number of experiments with different types of datasets [30]. The datasets can be divided into deeper datasets and wider datasets. A deeper dataset has more images per class in the training set than another. A wider dataset has less number of images per class in the training set. A shallow neural network has one fully connected layer, whereas a deeper network is a combination of convolutional layers, fully connected layers, and pooling layers [31]. The researchers have concluded that deeper architecture performs better with deeper datasets while shallow architecture achieves better results with wider datasets. The shallow neural network requires more dense layers for wider datasets and the deeper neural network requires more dense layers for deeper datasets.

The effect of filter size on image classification has been shown by Owais Mujtaba et.al. CNN model, which differs by only filter size and is implemented on two datasets, namely CIFAR10 and Fashion MNIST [32]. Through experiments, the authors concluded that the loss increases as the filter size increases, while accuracy decreases as the filer size increases. The problem associated with less filter size is the computational cost, which is very important when dealing with large datasets.

Kamil Dimililer et al. have studied how the number of layers affects the success of the model for the Brain Tumor Progression dataset. Several different CNN models which vary by the number of convolution and dense layers have been tried and tested to see their performance on the dataset [33]. They have observed that for sensitive results, a model which has a very less number of layers performs better. They have concluded that for binary classification, the result could be reduced by 7% by using deeper architecture.

Sanjit Maitra et. al. has shown the input parameters' effect on the accuracy of CNN for diabetic retinopathy. The input parameters such as the number of filters in one layer, number of convolutional layers, activation function, and size of the convolution kernel are considered [34]. They have concluded that the model results in higher accuracy and lower runtime when convolutional layers have fewer filters. Two important factors that significantly affect classification performance are the number of filters in convolution layers and the size of filters.

James Mou et al. studied how a number of filters affect the accuracy of the f CNN model of the speech recognition model on the Libri Speech dataset [35]. They have concluded that for the speech recognition model, the word accuracy for the LVCSR model gets better with an increase in the number of filters of the convolutional embedding layer.

Somenath Bera et al. discussed the effect of the pooling strategy on CNN. Five different pooling techniques have been applied to three hyperspectral types of datasets to make a comparison of hyperspectral remote sensing image classification [36]. The comparison has been done on the 2D CNN model, which extracts only spatial features. They have concluded from the experiments that for all three datasets, max-pooling gives better accuracy as compared to another pooling strategy for CNN.

From the literature survey, it is concluded that Every model has its own limitation due to the nonlinear nature of food. Food image recognition is a hot topic in computer vision, and the use of convolution neural networks has improved the result accuracy of food image recognition. It is found from the literature survey that no work has been done till now to classify Gujarati Food Images and most importantly there is no dataset available for Gujarati Food items.

As a resident of Gujarat and considering Gujarati Food this research work proposes a model which can classify Gujarati Food Images accurately with less amount of time.

3. Traditional Gujarati Food Dataset (TGFD)

3.1 Creating TGFD

A new dataset has been created for Traditional Gujarati Food items (TGFD). There are many Gujarati Food items, from which the famous Gujarati Food items have been selected in such a way that they all have almost the same color with minor variations. This makes it more challenging to develop a model that classifies them correctly. The dataset contains five food items, namely Dhokla, Handvo, Khakhra, Khandvi, and Patra, as shown in Fig. 2. The dataset consists of unique images collected from the internet, pictures taken using mobile phones, and real images captured by visiting different restaurants. To consider images from different aspects various poses, rotations, lighting, and shapes were considered while collecting images. There are a total of 1764 images with at least 300 images per class in the dataset, which are divided into training, validation, and testing with 70%, 20%, and 10% respectively. A validation dataset has been introduced to fine-tune the model hyperparameters and to avoid overfitting [37,38]. This means the model occasionally sees the data but never learns from it. The number of images has been increased by implementing different augmentation techniques as discussed in the following section for better performance of the model.

Dhokla Handvo Khakhra Khandvi Patra

Fig. 2 Sample Food Items from Traditional Gujarati Food Dataset

3.2 Data Augmentation

An effort is made with extraordinary augmentation techniques to expand the dataset artificially in order to achieve high accuracy and avoid overfitting. The images are resized to 224x224 before processing. The images are rotated at 45,90,135 and 180 degrees randomly to give correct predictions from any angle. The weight_shift_range and height_shift_range are set to 0.3 after multiple experiments, which helps to view incomplete images more clearly and predict correctly. The shear_range and zoom_range are set to 0.3, which is used to transform the orientation of the image. The horizontal flip parameter is set to true to flip the images horizontally, which helps to predict different patterns in order to increase accuracy.

4. Improved Selective Median Filter (ISMF)

Image pre-processing is a technique that improves image quality by removing noise and unwanted objects from the background. Noise means variation in the brightness of an image or a blurred image. There are different noises, such as salt and pepper noise, Gaussian noise, and Rician Noise. Natural images are affected by these, but food images are affected by impulse noise due to poor illumination quality of the camera, poor lighting, or blur image. The food image quality parameters are primarily resolution and impulse noise. Images captured by cameras or downloaded from the internet contain a lot of noise. It is necessary to remove the image noise so the interesting content can be highlighted. This chapter highlights the significance of such work in the context of food images.

4.1 MEDIAN FILTER

A median filter is the famous filtering technique used for removing impulse noise. It is a non-linear filter technique where the value of each reconstructed pixel is equal to the median of its nearest neighbors in the input image [39].

The drawback of a median filter is, it performs uniformly throughout the image irrespective of whether the image pixel is corrupted or not. [40] The median filter is effective in the case of low noise density only. While the noise density is higher than 50%, the size of the filtering window should to be enlarged to suppress serious noise which leads to edges and details being blurred. [41]

To limit this deficiency, first, noisy pixels should be identified and then replaced by the filter operator, while other pixels remain unchanged [42]. Based on this fact, to improve noise suppression and detail preservation simultaneously a new filter has been proposed.

4.2 IMPROVED SELECTIVE MEDIAN FILTER

In food images, it is very necessary to retain edges after pre-processing an image, as detects the shape of an image. To overcome the limitation of the median filter, a novel median-based method “Improved Selective median filter (ISMF)" has been proposed which is able to differentiate between noisy and healthy pixels. The Proposed algorithm sorts all the elements of the extracted window while the median filter only checks for the fifth pixel of the window. The pixels that lie between 0 and 255 is considered to be noisy otherwise the pixel is termed healthy. All the operations are performed in a fixed 2D Kernel of size 3X3. Any operation performed is based on the pixel values in the current processing window.

This approach only treats the noisy pixels, which reduces the time complexity and search space and improve the performance at the same time. The proposed method is simple, easy to implement, and requires very little computation. In this work, the emphasis is given to removing impulse noise from food images. The algorithm for ISMF has been shown in Fig 3.

Step 1: To begin, select a 3x3 2D window.

Step 2: The process begins from the center pixel (Pij).

Step 3: If the value of Pij is between 0 and 255, then it is not the corrupted pixel and its value remains unchanged.

Step 4: If Pij = 0 or Pij = 255, then the pixel is noisy and its value will be calculated as below.

4.1 If all the neighbors are noisy, then take the mean of all the elements and replace the Pij value with the same. Even if we take the median of the elements, it will be noisy.

4.2 If some of the pixels are noisy, calculate the number of noisy pixels and store the value in the variable noisy pixel (NP). The value of Pij will be decided according to the following conditions:

4.2.1 If NP =2, then convert the array into 1D and arrange the elements in ascending order. Remove the corrupted pixel from the array. Find the median of the remaining elements of the array. Replace the result value with the corrupted pixel Pij.

4.2.2 If 3<=NP<=5, then convert the array into 1D and arrange the elements in increasing order. Eliminate the noisy pixel from the array, find the mean of the remaining elements, and replace the resultant value with the noisy pixel.

4.2.3 If NP >=6, then find the window's min, max, and med values, where min is the window's minimum element, max is the window's maximum element, and the med is the window's median. If max>med>min, then replace the median with Pij, else keep the original value and move to the next pixel by increasing the window size from 3x3 to 5x5.

Step 5: Move to the next pixel and repeat steps 1 to 4 for all the remaining pixels of the image.

Fig. 3 ALGORITHM ISMF

To check the performance of the proposed model, Image noise density has increased from 10% to

90%. The window size of 3X3 for all filters has been considered. The performance comparison of ISMF has been made with the original median filter using the matrices PSNR (peak signal-to-noise ratio) and MSE (Mean Square Error). Table 2 shows the PSNR and MSE for the ISMF and original median filter by increasing noise from 10% to 90% for different images. The higher PSNR result indicates better image denoising.

The lower MSE resultant pixel indicates better image quality. PSNR and MSE are calculated using Eq1.and Eq2.

$$\text{PSNR} = 10\log_{10} X\left(\frac{255^2}{MSE}\right) \qquad \text{Eq. 1}$$

$$\text{MSE} = \frac{1}{a*b}\sum_{i=0}^{a}\sum_{j=0}^{b}(Ni,j - Ci,j)^2 \qquad \text{Eq. 2}$$

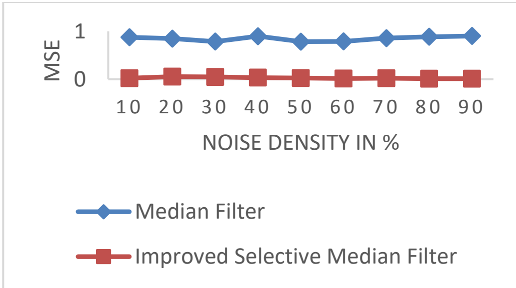
Here Ni,j and Ci,j denotes the original image and restored image, a and b are the width and height of an image.

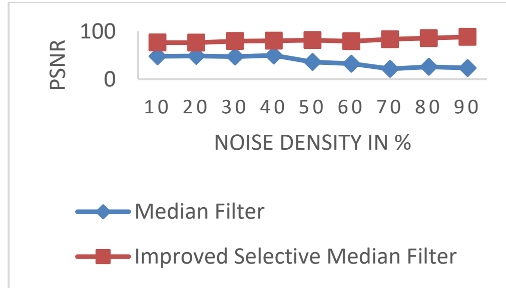
Table 2: Comparison of Median filter and ISMF on TGFD

Noise Density in %	Median Filter		Improved Selective Median Filter	
	PSNR	**MSE**	**PSNR**	**MSE**
10	47.86	0.8755	76.48	0.0212
20	48.56	0.8466	76.25	0.0546
30	47.25	0.7855	79.54	0.0475
40	49.78	0.8965	80.25	0.0325
50	35.89	0.7845	81.48	0.0254
60	32.45	0.7895	79.25	0.0154
70	21.69	0.8566	83.45	0.0212
80	25.89	0.8854	85.78	0.0114
90	23.45	0.9014	88.23	0.0110

For performance, TGFD has been considered. The performance comparison of PSNR and MSE of the Standard Median filter and ISMF has been shown in Table 2.

It is proven from the above table that PSNR increases as the noise increases in images from 10% to 90 % and MSE decreases accordingly. It proves ISMF performs better than the standard median filter in each of the cases. As compared to the median filter the MSE for ISMF has decreased by 3 to 5 % which is remarkable. The graphical representation of the same has been shown in Fig.4.





MSE performance for Different Noise PSNR performance for Different Noise

Fig.4 Comparison graph of PSNR and MSE for different noise for Food image

Fig.5 shows the image output after applying ISMF and Standard Median filter. It is clear from Fig.4 and Fig.5 that ISMF performs better than the standard median filter and also preserved the edges of food images and gives better visualization of images.

(a) Noisy Image (b) Standard Median Filter (c) ISMF

Fig. 5: Applying filters to noise images: (a) Noisy image (b) Standard median filter (c) ISMF

5. Optimization Through Transfer Learning

This chapter briefly explains the implementation of simulation and transfer learning on pre-built models to improve performance.

5.1 SIMULATION

Simulation is an important research method to understand the operational behavior of the model [4]. There are an ample number of pre-built models that exist, like VGG16, Alexnet, and Inceptionv3, which are already trained on the ImageNet dataset and give remarkable performance [43]. In order to check the performance of the pre-built model on the newly created TGFD, this research work has considered VGG16, VGG19, Resnet50, Inceptionv3, and Alexnet to check its training and testing accuracy, and their results are as shown in Table 3.

Table 3: Classification accuracy of models in Simulation

Model Name	Simulation Test Accuracy (%)	Test Loss (%)
VGG16	78.24	12.15
VGG19	80.45	11.77
ResNet50	45.56	0.69
Inceptionv3	81.58	20.31
Alexnet	58.87	6.23

The highest accuracy achieved was 81.58% by the Inceptionv3 model, but the classification loss was 20.31%, which is very high. This loss is further increased with the number of epochs, resulting in overfitting the model. None of the models provide satisfactory classification accuracy on TGFD. To increase the classification accuracy of TGFD with more efficiency and at same time reducing the loss, an effort is made to implement the strong transfer learning techniques.

5.2 TRANSFER LEARNING

In transfer learning, the upper layers of the Convolutional neural network are more generic in nature while the lower layers are more task-specific. It is a good idea not to change the upper layers as it has generic features that can be the same for other models and might be trained on different datasets [4-5]. The classification part of the model can be changed, by freezing the rest of the layers to load the trained weights of the model [40-41]. Now, the remaining CNN acts as a fixed feature extractor for the new dataset. Fig.6 shows the architecture of Transfer Learning.

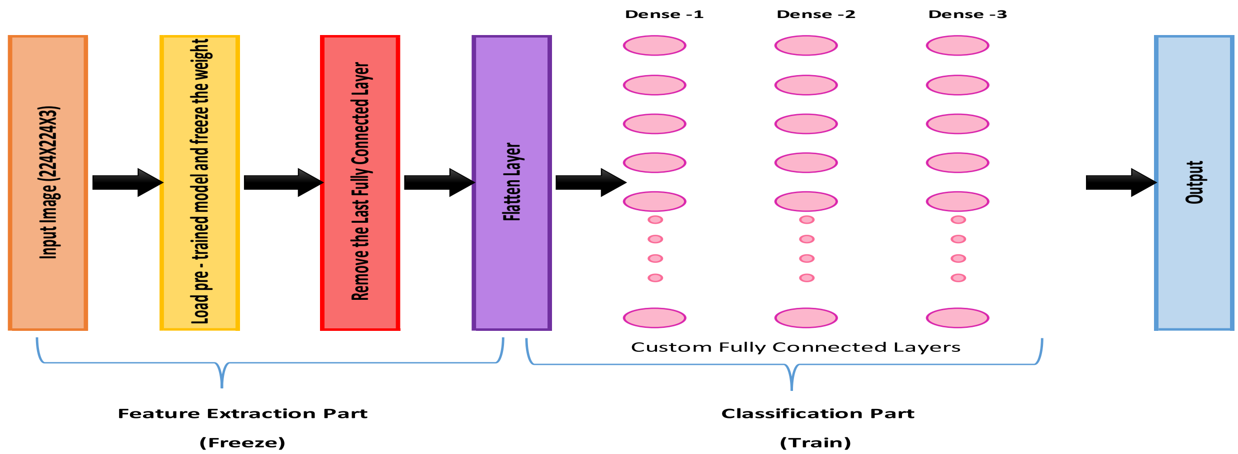


Fig. 6 Transfer Learning

Transfer Learning has been implemented by changing the classification layer of the model and freezing the rest of the layers of the model. In addition to that, a flatten layer followed by a fully connected layer with Softmax activation function has been added to the model. The model is compiled using Adam optimizer. Initially, all the models run for 500 epochs, but it has been observed that after 100 epochs, accuracy is not improving. The result of classification accuracy after implementing Transfer Learning on TGFD for 100 epochs is shown in Table 4.

Table 4: Classification accuracy of models in Transfer Learning

Model Name	Simulation Test Accuracy (%)	Transfer Learning Test Accuracy (%)	Trainable Parameters
VGG16	78.24	83.91	125,445
VGG19	80.45	85.06	125,445
ResNet50	45.56	52.3	501,765
Inceptionv3	81.58	86.22	256,005
Alexnet	58.87	62.93	21,136

As seen in Table 4, ResNet50 does not give good accuracy as the model starts overfitting after 70 epochs. A TGFD is a deeper dataset and deeper architecture gives better results with it [31]. Alexnet fails to achieve good accuracy as it is a shallow architecture. The highest accuracy achieved is 86.22% by the Inceptionv3 model. The model contains 256,005 trainable parameters. The accuracy is better than the result of the simulation but not optimal and it is taking large amount of time and has 256005 Trainable parameters. To further improve the classification results on TGFD dataset, fine-tuning has been done on the models.

6. Optimization Through Hyperparameter Tuning

As hyperparameter tuning improves classification accuracy, an effort was made to implement transfer learning along with fine-tuning [5]. Fine-tuning is more flexible as the feature extraction part of the model along with the classification part can be changed. It is possible to Fine-tune all the layers, but it is always a good practice to fix the upper layers as it contains more generic features [10].

Fig. 7 shows the architecture of Fine-tuning.

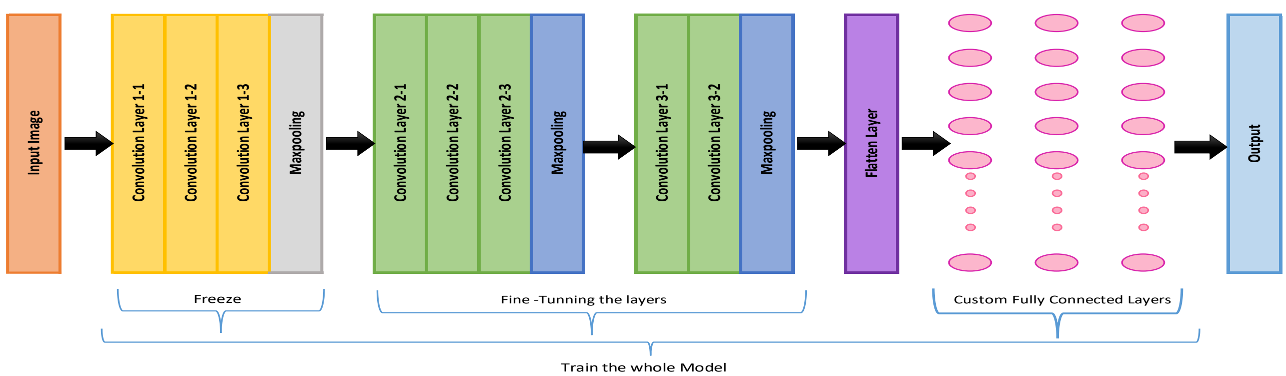


Fig. 7. Fine-tuning

In this research work the following changes have been made. The last Convolutional layer and the pooling layer have been changed, keeping the other layers frozen. A new dense layer along with a dropout has been added [44]. The parameters of the model are as below:

- The dropout rate is set to 0.5 as this gives the best results with hidden layers and dense layers [44].
- When there are more than two classes, Softmax is preferable as it returns probabilities of each class so in the last layer Softmax activation function has been used [50].
- After doing experiments with other optimizer an Adam optimizer has been used to compile the model as it gives the best classification accuracy for TGFD.
- A learning rate of 0.002 has been chosen for Adam optimizer with the use of a learning rate range test [45]
- The categorical cross-entropy has been used as a loss function as it is a multi-class classification model [51].
- The models run for 100 epochs.

The five models VGG16, Resnet50, Inceptionv3, VGG19, and Alexnet have been fine-tuned with these parameters and their corresponding accuracy for TGFD along with trainable parameters are as shown in Table 5.

Table 5: Classification accuracy of models in Fine-Tuning

Model Name	Fine-tuning Test Accuracy (%)	Trainable Parameters
VGG16	85.23	14,840,133
VGG19	87.3	20,149,829
ResNet50	62.32	23,587,712
Inceptionv3	89.36	22,024,357
Alexnet	68.73	28,063,621

As shown in Table 5, the Inceptionv3 model achieved the best classification accuracy of 89.36%. with 22,024,357 trainable parameters. The training loss is increasing with the number of 80 epochs in the ResNet50, resulting in model overfitting.

Simulation, transfer learning, and fine-tuning have been implemented for five models, namely VGG16, VGG19, Inceptionv3, Alexnet, and Resnet50. The best classification accuracy achieved by Inceptionv3 after transfer learning is 86.22% and after fine-tuning it is 89.36%. By implementing fine-tuning, the classification accuracy of Inceptionv3 has been increased by 8% compared to simulation.

The results achieved by the experiments conducted in sections 4 and 5 show that by implementing transfer learning and fine-tuning, the testing accuracy has been increased by at least 5% and 8%, respectively, proving that transfer learning along with fine-tuning significantly improves classification accuracy.

As TGFD is a deeper dataset and according to the research findings, a deeper network gives better results with deeper dataset [31]. As Alexnet is shallow architecture it doesn’t give a good result with TGFD.

To resolve the issue of overfitting and to further improve the classification accuracy, a model from scratch has been developed .

7. DRCNN-Depth Restricted Convolution Neural Network

All the pre-built Deep Convolutional Neural Networks like VGG16, Inceptionv3, VGG19 have a large no. of layers and, hence contain a huge number of parameters [46]. Such networks need hours, days, or even weeks to train. Also, the classification accuracy achieved by Inceptionv3 after transfer learning is 86.22% and after
fine-tuning it is 89.36% which can be further improve. So, in order to reduce amount of time required to train a model with improved classification accuracy lightweight network model having less number of parameters and a small Convolutional kernel size is required. This research work has tried to developed a more accurate and efficient model for classifying Gujarati Food dataset with less number of parameters.

To make the model lightweight it has been constructed with 11 Convolutional layers, 4 Max-pooling layers followed by one fully connected layer, and a SoftMax layer. The proposed model named "The Depth restricted Convolutional neural network (DRCNN)". Fig.8 shows the architecture of the proposed model DRCNN.

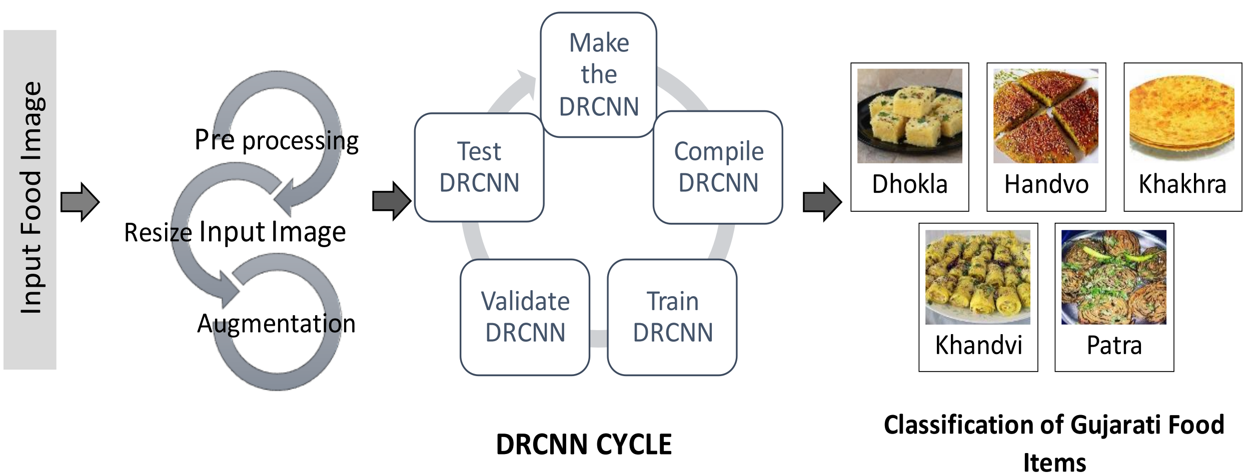


Fig. 8 Architecture of the proposed model DRCNN

7.1 Hyperparameter Selection for DRCNN Using Empirical Analysis

For any CNN model, it is very necessary to decide the hyperparameters like the number of convolutional and the number of fully connected layers, batch size, kernel size, number of filters in a layer, optimizer, and learning rate, etc. From the literature survey, the following research observations have been found and based on this study The DRCNN model has been built.

- There are two types of datasets: Deeper and Wider. A deeper dataset has more images per class than a wider one. A deeper architecture works best with deeper datasets and a shallow architecture with wider datasets [31]. the proposed dataset, TGFD, is a deeper dataset as it has more number of images per class. Hence deeper network gives better results with it.
- The number of convolutional and fully connected layers directly affects the runtime of the model. So, less number of layers should be selected in order to reduce computational complexity [33].
- The lower filter size can increase the performance of the model. The lower filter size and low learning rate were chosen for DRCNN in order to increase the accuracy of the model [47].
- When the learning rate is low, a lower batch size gives a better result. A lower batch size of 32 has been chosen for DRCNN as it helps to improve accuracy with a low learning rate [48].
- When the number of layers is greater, keeping the lower learning rate gives a better result [48].
- The model results in higher accuracy and lower runtime when convolutional layers have fewer filters [35].
- The number of filters in convolution layers and the size of filters have a significant effect on the accuracy of the system [47].
- The max-pooling layer reduces the parameter count, which decreases computational complexity [36]. So, in DRCNN max-pooling has been used in order to reduce model complexity.
- The accuracy of the model depends more on the number of convolution filters in one layer and the size of the convolution kernel and less on the number of convolutional layers or the depth of the network [34].

The proposed model has a lower number of convolutional and dense layers as compared to all the pre-
trained models. Since it restricts the depth of the model the name given as DRCNN.

Based on the selected hyperparameters, a model from scratch has been developed and is able to achieve better accuracy compared to previous implementation with minimum loss.

The input images are converted to 224x224x3 and given as input to DRCNN. Batch Normalization (BN) has been added after each Convolutional layer to make the training and learning of DRCNN faster and easier. A higher learning rate (LR) can be obtained by adding BN layers without compromising convergence [49]. Each Convolutional layer is followed by a pair of BN and Relu activation functions to increase classification accuracy and handle nonlinearity. Relu activation function has been used except for the last layer as it is easy to train and often achieves better performance [50].

The model compiles using the following parameters.

- The trainable parameter in the model is 481,557.
- Adam optimizer is a combination of AdaGrad that can deal with sparse gradients and RMSProp, which is able to deal with non-stationary objectives. It requires less memory, easy to implement, and also computationally efficient. Hence, for the proposed model Adam optimizer is used [55].
- The learning rate is set to 0.0001 for the Adam optimizer on TGFD using Cyclic Learning Rate [45].
- The categorical cross-entropy loss function has been used for error calculation it is a multi-class classification model [51].

The algorithm to train TGFD on DRCNN has been shown in Fig.9

<u>**Algorithm Steps:**</u>

Input:

Training, Validation and Testing instance set 'T', an Image set and a label value

Image Set I(i) = {I1(i), I2(i),…..,In(i)}

Label Set L(i) = {Class1, Class2, ------, Class n}

Initialization:

Step 1: Collect the images from mobile, Internet and Real images clicked by visiting restaurants to prepare the dataset.

Pre-processing Phase:

Step 2: For each instance of input data,

Remove Background Noise

Resize the image into the specified range

Step 3: Apply augmentation techniques to increase the size of data artificially

Define the model:

Step 4: Construct the model Depth Restricted convolutional neural network

Step 5: Extract the input, output, and intermediate properties of layers in the model

Step 6: *Configure the model and* load the data in the model DRCNN

Feature Extraction Phase:

Step 7: Apply activation functions to get features dataset from selected or configure layer of the model

Step 8: Prepare feature dataset and its equivalent target label for hyper-parameter tuning, training, and testing of model

Parameter Hyper tuning Phase:

Step 9: Select the optimizer and learning rate for the model

Step 10: Compile the model

Training Phase:

Step 11: Initialize the parameter tuned for a model of DRCNN
Step 12: Initialize the feature data and label data for the training dataset.
Step 13: Train the model for DRCNN algorithms.
Validation Phase:
Step 14: Initialize the feature data for the validation dataset.
Step 15: Validate the model DRCNN
Testing Phase:
Step 16: Initialize the feature data for the testing dataset.
Step 17: Load the trained model of DRCNN algorithms.
Step 18: Check the Testing accuracy of the model to check the model's efficiency.

Fig. 9 Algorithm to train TGFD of DRCNN

7.2 Experiments and Results

To see the effect of TGFD on DRCNN the experiments were implemented on an Intel i7-9750H Lenovo Legion Y540 CPU @ 2.60GHz processor, which supports a multicore processor equipped with a GeForce GTX 1650 NVIDIA GPU with 8GB of memory. Python 3.8.8 was used in the Deep Learning Framework, and Python 3.8.8 was used in Keras 2.7 and TensorFlow 2.7. The performance evaluation of the model is measured using the parameters like accuracy, F1 Score, precision, and recall.
Table 6 shows the results for the DRCNN and fine-tuned models with classification accuracy along with the number of parameters, number of convolutional layers and fully connected layers used.

Table 6: Comparison of the DRCNN with Fine-tuned existing Models

Model Name	Classification Accuracy (%)	No. of Parameters	Convolutional Layer/Fully Connected Layer
VGG16	85.23	138,357,544	13/3
VGG19	87.3	143,667,240	16/3
ResNet50	62.32	25,636,712	>17/1
Inceptionv3	89.36	23,851,784	>60/3
Alexnet	68.73	62378344	5/3
DRCNN	**95.48**	**482069**	**11/1**

It can be seen from Table 6 that the DRCNN model achieves a remarkable classification accuracy of 95.48%, which is more than all the fine-tuned models. The DRCNN model contains 482069 parameters, which is 48 times less than the Inceptionv3 model, which gives the highest classification accuracy after fine-tuning among all the pre-trained models. The validation loss of DRCNN is only 0.8041. Since DRCNN has less number of parameters it takes 30 minutes to run as compared to another pre-built model which has millions of parameters and hence takes 50 to 60 minutes to run. Though TGFD is a balanced dataset along with accuracy the other evaluation parameters, namely precision, F1 score, and sensitivity has been considered as accuracy sometimes bias to classes and misleading. The results of same models have been shown in Table 7.

Table 7: Performance of models on TGFD dataset based on evaluation parameters

Model	Accuracy (%)	Precision	Sensitivity	F1 Score
VGG16	85.23	0.82	0.93	0.87
VGG19	87.3	0.75	0.86	0.80
Alexnet	68.73	0.81	0.76	0.78
GoogleNet	89.36	0.82	0.88	0.83
Resnet50	62.32	0.92	0.67	0.77
DRCNN	95.48	0.95	0.90	0.92

The training and testing accuracy is shown in Fig. 10, and the training and testing loss is shown in Fig 11. The DRCNN runs for 500 epochs but the accuracy and loss graph has been shown for 100 epochs.

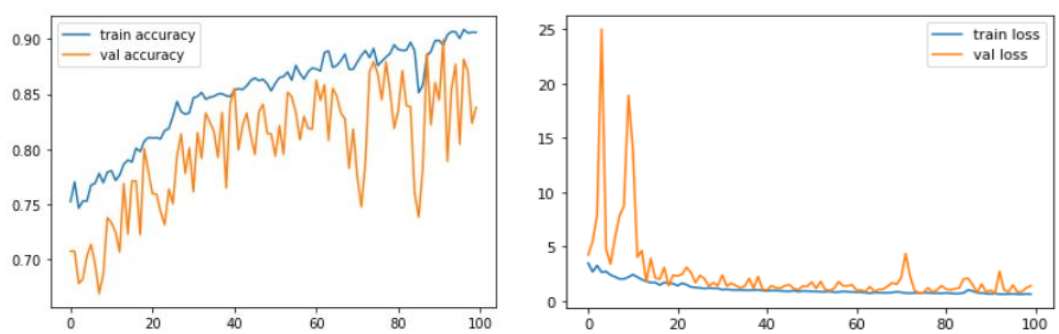


Fig. 10 & 11 Accuracy and Loss curves for DRCNN with Training epoch set to 100

From this figure, we can say that the constructed models testing accuracy is more as compared to training.

7.3 Testing DRCNN

The effectiveness and performance of DRCNN is tested in two ways. In the first case, the DRCNN is tested by increasing the types (class) of Gujarati food images and in the second case The DRCNN is run with different food datasets to see its generalizability.

Test Case 1

TGFD is extended and the DRCNN is tested on a greater number of Gujarati food classes. TGFD has been Extended from 5 food classes to 20, by including food items like muthiya(99), khichu(81), poha(110), thepla(104), chapati(400), puri(400), white rice(400), biryani(358), Gulab Jamun(170), salad(400), Gujarati dal(400), dabeli(108), samosa(400), upma(400). The total food items in Extended TGFD contain a total of 6080 images. The DRCNN is run in the same environment, compiled with Adam optimizer at a learning rate of 0.0001 for 100 epochs. The same experiments were conducted on all fine-tuned models to compare the results. The results obtained by the experiments are shown in Table 8.

It is observed from Table 8 that when food classes are increased the accuracy of DRCNN increases while in other pre-built model, the accuracy decreases. The performance of DRCNN is improving with a large number of images.

Table 8: Comparison of Fine-tuned Inceptionv3 and DRCNN for a greater number of food classes

No. of Gujarati food class	**No. of Gujarati food images**	**VGG16**	**VGG19**	**Inceptionv3 (%)**	**Alexnet**	**Resnet50**	**DRCNN (%)**
5	1747	85.23	87.3	89.36	68.73	62.32	**95.48**
7	2016	76.21	73.79	81.65	70.23	58.54	**91.12**
10	2311	70.59	73.95	75.76	71.24	55.54	**93.36**
12	3110	70.44	72.01	77.61	72.45	54.85	**92.98**
15	4310	72.37	66.64	85.7	65.54	52.21	**93.76**
20	6080	75.49	70.23	67.5	69.32	50.25	**96.10**

Test Case 2

In this test case, all the models are tested for different existing datasets having different types of food items to check their versatility. The datasets considered are Food20, Indian-100, Food-101, FFML Dataset, and UECFOOD100 and the details are shown in Table 9 [56-59] The models run for two datasets of Indian food.

Table 9: Result of DRCNN for Different Datasets

Name of Dataset	Type of Food item in the dataset	No. of food class	No. of food images	Accuracy of VGG16 (%)	Accuracy of VGG19 (%)	Accuracy of Inceptionv3 (%)	Accuracy of Alexnet (%)	Accuracy of Resnet50 (%)	Accuracy of DRCNN (%)
TGFD	Gujarati	5	1747	85.23	87.3	89.36	68.73	62.32	**95.48**
Food20	Indian	20	2000	56.62	70.78	70.56	19.32	45.58	**95.5**
Indian-100	Indian	50	5000	57.23	58.12	65.47	27.4	17.12	**97.7**
Food-101	All Mix types of Food	101	101000	67.34	69.34	53.38	35.25	35.89	**98.98**
FFML Dataset	Romanian food dishes	424	1281	47.12	54.23	48.01	23.45	17.26	**99.79**
UECFOOD100	Japanese food	100	14461	54.67	57.89	60.21	50.12	57.12	**99.10**

Following are the observations after the experiments:

Observation:

- It has been observed that other pre-built models do not give good results on all datasets but DRCNN gives remarkable accuracy on all types of datasets, especially on FFML and UECFOOD100 which is almost 100%.
- Alexnet and Resnet50 give a poor performance on all types of datasets.
- VGG16 and VGG19 give good performance for FOOD20 but for the rest of the dataset, the performance is not good.
- Inceptionv3 fails to give good accuracy in almost all other types of datasets.

It can be concluded from above observations that the pre-built models are not sufficiently generalized. It is also difficult to adapt them practically because of poor speed and contains more number of layers which requires more time in execution. Most of the pre-built models are complex in nature and suffers from overfitting problem with different food classes.

8. Time Complexity of CNN Models

The convolution neural network is gaining a lot of popularity in image classification problems nowadays. It has been used in many different classification problems, like medical imaging, handwritten digits, image classification, etc. It is very critical to estimate the time required by the model to achieve the desired task [52]. The proposed work involves computational studies to find the factors that affect the model's performance, the time each layer takes to run, and how it affects the model's overall performance. As it is not feasible computationally to try every possible combination of all input parameters, the chosen hyperparameters for the work are the number of convolution layers, number of dense layers, pool size, size of filters, size of neurons, number of filters and size of the convolution kernel [32].

Among all the operations, convolutional layers, pooling layers, and dense layers are the important ones. The number of parameters at any layer is the count of "learnable" elements. The input layer provides the shape, but it has no learnable parameters. The pooling layer does not have learnable parameters but it helps in reducing the dimension of the feature map and the parameter count result in reduced computational complexity. According to Kaiming He et al., fully connected layers and pooling layers take only 5 to 10% of the computational time and 90% of the time is taken by Convolutional layers [53]. We can reduce the time complexity by wisely choosing the number of convolutions and fully connected layers. Kaiming et al. have proposed a formula for only the convolution layer which does not consider many factors like batch size and Learning rate. After Extensive research and scientific methods, this research work proposed a formula to find the time complexity of a whole CNN model. Though dense layers affect only 5–10% of the complexity of the model [53], this research work has considered convolution layers and fully connected layers in order to find the computational complexity of the model accurately.

Each convolutional layer contains filters that have a depth, number of kernels and filter size, which varies for each convolutional layer. The computational complexity of a convolution layer is a multiplication of these parameters. The total computational complexity of convolution layers is obtained by doing a summation of the complexity of the individual convolution layer.

By considering the learning rate and batch size, the time complexity of the convolution layer can be calculated as per Eq.1

$$\left(\sum_{n=1}^{d} k_{n-1} \cdot s_n^2 \cdot f_n \cdot l_n^2\right) \cdot r_1 \cdot b_1 \qquad (1)$$

Here d is the depth of the convolutional layer, l_n is the length of the output feature map, f_n is the number of filters in the n^{th} layer, S_n is the length of the filter, k_{n-1} defines the number of input channels in the l^{th} layer, *r1* is the learning rate, *b1* is the batch size.

Considering a Fully connected layer, each layer consists dimension of the input/output channel, the width of the input, the height of the input, and the number of outputs.These parameters are linked to one another. It is a layer that connects higher layers with the output layer. This layer contains a number of neurons which will vary for each layer and the output size depends on these neurons. To calculate the time complexity of each fully connected layer it is require to multiply the parameters of each fully connected layer and finally add all the layer's complexity in order to find the total complexity of all fully connected layers of the model

The time complexity of the fully connected layer can be calculated as per Eq.2.

$$\left(\Sigma_{l=1}^{f} D \cdot W \cdot H \cdot N\right) \qquad (2)$$

Here *f* is the Depth of the fully connected layer; D, W, H, and N define the Dimension of the input/output channel, the width of the input, the height of the input, and the number of outputs respectively.

The total time complexity of a CNN model can be calculated as per Eq. 3.

$$\left(\sum_{n=1}^{d} k_{n-1} \cdot s_n^2 \cdot f_n \cdot l_n^2\right) \cdot r_1 \cdot b_1 \quad + \quad \left(\Sigma_{l=1}^{f} D \cdot W \cdot H \cdot N\right) \qquad (3)$$

To prove it practically, eight different architectures have been created as shown in Table 10. The architecture named from A to H varying by the size of filters, number of convolutional layers, number of filters, number of fully connected layers, and kernel size [54] to examine the effect of each layer according to Eq.1 and Eq.2 and the overall time required to run the model on TGFD.

Table 10: CNN Architectures with parameters

Model	Number of Convolutional Layers	Number of Filters	Pooling size	Filter size	Number of Dense layers	Neurons in each Dense layer	Time (In Seconds)
A	2	64,32	2X2, 1X1	*5X5, 3X3*	1	20	1208
B	2	32,16	2X2, 1X1	*7X7,5X5*	1	20	1189
C	2	16,8	2X2, 2X2	3X3, 3X3	1	20	1216
D	2	16,8	2X2, 2X2	5X5, 3X3	2	64,20	1270
E	3	64,32, 16	2X2,2X2, 1X1	3X3,3X3,3X3	3	128,64,20	1645
F	2	64,32	2X2, 2X2	5X5, 5X5	3	128,64,20	1119
G	2	64,32	2X2, 1X1	3X3, 3X3	2	64,20	1290

H	3	64,32, 16	2X2,2X2, 1X1	3X3,3X3,3X3	2	128,20	1600

After a detailed investigation of the experiments following are the research observation which will be helpful to a new researcher to design a CNN model.

Observation

- Models A, B, and C have the same number of convolutional and fully connected layers, but they take different amount of time to run as the parameters of the model vary with the number of filters and size of filters. Hence, it proved that the time complexity of the model also depends on the number of filters and size of filters along with the number of convolution layers.
- Model A and B has the same number of convolutional and fully connected layers, the same pooling size but they vary by the number of filters and filter size. The time taken by A is greater than B proving that the number of filters and size of filters has a significant effect on the time complexity of the model.
- Model E has 3 convolution and 3 fully connected layers, so it took the highest time as compared to all other models.
- Model E and F has same number of dense layers but E has one more convolutional layer than F. as the convolutional layer takes 90% of the computational time there is a huge difference in the time taken by both models to run.
- Model C and D has same number of convolutional layers but D has one more dense layer than A.as the dense layer only takes 5 to 10% of the computational time there is not much difference in the time taken by the model D.
- Models A and F have the same number of convolution layers, but F has more fully connected layers, but model F has more filter size as compared to model A, so model F took less time to run compared to model A. This shows that keeping the filter size higher can reduce the accuracy of the model but decrease the computational cost of the model. The filter size is inversely proportional to the accuracy of the model [13]
- Models E and H have the same number of convolution layers and all other parameters except fully connected layers.
- Model H has 2 fully connected layers hence it takes less time than model E, which has 3 fully connected layers.
- The number of operations for a convolution layer is much larger than the number of operations for a dense layer.
- The per epoch time for a dense layer is greater than the per epoch time of the convolution layer.
- It is also not necessary that a greater number of parameters require higher operations.
- It is not necessary that if the model has a higher number of layers, it also has a higher Computational complexity.

- An optimizer, batch size, filter, and neurons greatly impact the time taken by the model.
- The convolutional layers, max pool, and fully connected layers directly affect the performance of the model [52].

Fig.12 shows the time complexity of each of the model.

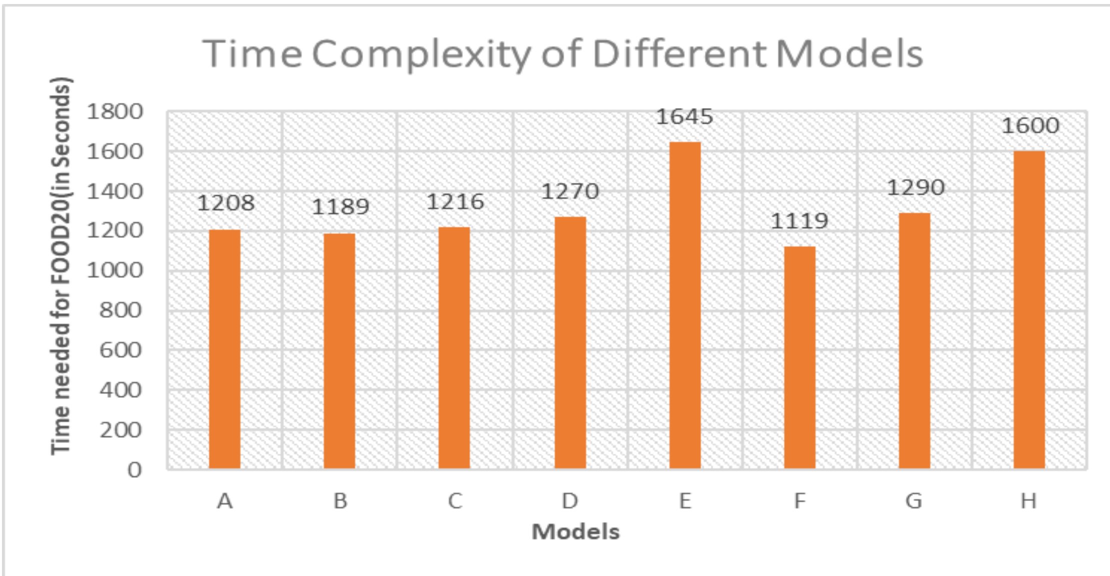


Fig. 12: Time Complexity of each model based on chosen parameters

9. CONCLUSION

Food identification is the first step for proper dietary assessment. Proper identification of food can be done with good-quality images. For that pre-processing is required. Image pre-processing is helpful to increase the quality of an image before further processing. This research work proposes a new algorithm ISMF that works in two phases. In the first phase, the corrupted pixel is identified and secondly, the algorithm is applied only to noisy pixels which saves computational time. The experiment results reveal that the ISMF performs better than the median filter in removing the impulse noise from the images by preserving edges and fine details so processing the result images can give better accuracy. ISMF sorts all the pixels of the window while the median filter only sorts the value of center pixel of the window. The PSNR and MSE value is also better than the median filter proving ISMF gives better performance and reduces error.

❖ The first attempt to classify Gujarati food items is made in this research work. A new dataset named TGFD has been developed which consists of five items, namely Dhokla, Handvo, Khakhra, Khandvi, and Patra. Multiple empirical research have been done by using different learning rates, to see their effect on the accuracy of the model. Ways of improvising the accuracy have also been implemented, like data augmentation, hyperparameter tuning, and using batch normalization and dropout.

❖ Transfer learning has been implemented on pre-trained models, namely Alexnet, VGG19, Resnet50, VGG16, and Inceptionv3 in order to see the performance on TGFD.

❖ Fine-tuning has also been applied in order to increase the accuracy by changing the feature extraction part along with the classification part, and the results have been compared.

❖ To further improve classification accuracy and to design a lightweight model a Depth Restricted Convolutional Neural Network has been proposed. The model has been built by tuning several hyperparameters. The TGFD dataset is the trained and tested on the DRCNN and achieves 95.48% of remarkable classification accuracy with a loss rate of 0.8041. The DRCNN model size is 48 times smaller than the Inception v3 model. The model's outstanding results on extended TGFD and different types of food datasets proves its versatility.

❖ The time complexity of any CNN model is a practical issue that all researchers find nowadays. Finding time complexity helps the researchers decide the impact of each hyperparameter they choose for building a model. This research work tried to find the time complexity of the model and that which are the crucial parameters for determining the time complexity of the model.

10. PUBLICATIONS

➢ Research Paper Presented/Published

[1] Bhoomi Shah and Hetal Bhavsar (2023), "Depth Restricted Convolutional Neural Network - A Model for Gujarati Food Image Classification", Accepted in The Visual Computer – International Journal of Computer Graphics [**SCI, Scopus Indexed, and UGC Care Journal**]

[2] Bhoomi Shah and Hetal Bhavsar (2022), "Time Complexity in Deep Learning Models", Presented in 4th International Conference on Innovative Data Communication Technology and Application", Published in Procedia Computer Science Journal, Elsevier, Volume 215, Issue C, pp. 202-210.

DOI: 10.1016/j.procs.2022.12.023 **[Scopus Indexed]**

[3] Bhoomi Shah and Hetal Bhavsar (2020), "Overview of Deep Learning in Food Image Classification for Dietary Assessment System", Presented in Sixth conference on Intelligent Systems, Technologies and Applications (ISTA), Published in Advances in Intelligent Systems and Computing Series, Springer, Volume 1353, pp. 265-285. DOI: 10.1007/978-981-16-0730-1_18 **[Scopus Indexed]**

[4] Bhoomi Shah and Hetal Bhavsar, "An Improved Selective Median Filter for noise reduction on Food Images" **[In a process of Publication in SCI Journal]**

➢ Copyright

[1] **Indian Copyright:** Bhoomi Shah and Hetal Bhavsar "CLASSIFICATION OF GUJARATI FOOD IMAGES USING DEPTH RESTRICTED CNN MODEL" (May 2022) **[Copyright Granted]**

[2] **Indian Copyright:** Bhoomi Shah and Hetal Bhavsar "Mathematical Estimation of Computational Complexities of Deep Learning Models" (Dec 2022) **[Copyright Granted]**

11. REFERENCES

[1] G. William, Y. Wei, “A survey of deep learning: platforms, applications and emerging research trends”, Special section on human-centered smart systems and technologies. IEEE Access 6 (2018)

[2] B. Shah · H. Bhavsar, “Overview of Deep Learning in Food Image Classification for Dietary Assessment System”, Advances in Intelligent Systems and Computing (2020)

[3] Z. Lei, Z. Chu, L. Fei, Q. Zhengjun, H. Yong, “Application of deep learning in food: a review”, Comprehensive Rev. Food Sci. Food Safety (2019)

[4] J. J. Bird, D. R. Faria, “From Simulation to Reality: CNN Transfer Learning for Scene Classification”,10th International Conference on Intelligent Systems, pp. 619-625, (2019)

[5] Z. Shen, A. Shehzad, S. Chen, H. Sun, J. Liu, “Machine Learning Based Approach on Food Recognition and Nutrition Estimation”, Procedia Computer Science,174 pp. 448–453, (2020)

[6] Vasuki, P., Kanimozhi, J., Devi, M.B.,” Proceedings - 2017 IEEE International Conference on Electrical, Instrumentation and Communication Engineering”, ICEICE, pp. 1–6, (2017)

[7] Rani, M., Yadav, J., Rathee, N., Goyal, S.,” Comparative Study of Various Preprocessing Technique for CNN-based Image Fusion” IEEE Delhi Section Conference, DEFCON 2022

[8] M Anwar Shah, Javed Iqbal Bangash Arshad Khan, “Comparative analysis of median filter and its variants for removal of impulse noise from grayscale images”, Journal of King Saud University, Elsevier (2020)

[9] Abhijith Reddy Beeravolu, Sami Azam, “Preprocessing of Breast Cancer Images to Create Datasets for Deep-CNN”, IEEE Access, vol. 9, pp. 33438-33463 (2021)

[10] R. Siddiqi, “Effectiveness of transfer learning and fine tuning in automated fruit image classification”, Proceedings of the 2019 3rd International Conference on Deep Learning Technologies, pp. 91-100, (2019)

[11] M. Anthimopoulos, L. Gianola, L. Scarnato, P. Diem, S. Mougiakakou, “A food recognition system for diabetic patients based on an optimized bag-of-features model”, J. Biomed. Health Inform. 1261–1271 (2014)

[12] L. Bossard, L., M. Guillaumin, Lv. Gool, “Food-101-mining discriminative components with random forests in computer vision”, ECCV, pp. 446–461 (2014)

[13] Y. Kawano, K. Yanai, “Food Image Recognition with Deep Convolutional Features” (ACM), pp. 589–593 (2014)

[14] K. Yanai, Y. Kawano, “Food Image Recognition Using Deep Convolutional Network with Pretraining and Fine-Tuning”, IEEE, pp. 1–6 (2015)

[15] S. Christodoulidis, M. Anthimopoulos, S. Mougiakakou, “Food recognition for dietary assessment using deep convolutional neural networks”, ICIAP, pp. 458–465, 2015

[16] A. Meyers, “Im2calories: towards an automated mobile vision food diary”, IEEE International Conference Computer Vision, pp.1233–1241, 2015

[17] H. Hassannejad, G. Matrella, P. Ciampolini, I. Munari, M. Mordonini, “Food image recognition using very deep convolutional networks”, Proceedings of the 2nd International Workshop on Multimedia Assisted Dietary Management, pp. 41–49, 2016

[18] J. Chen, C. Ngo, “Deep-Based Ingredient Recognition for Cooking Recipe Retrieval”, ACM, pp. 32–41, 2016

[19] Yan, L., Guanling, C., Vinod, V., Yunsheng, M., Chang, L.,Yu, C., “Deep food: deep learning based food image recognition for computer-aided dietary assessment”, pp.37–48, 2016

[20] P. Pouladzadeh, P. Kuhad, S. Peddi, A. Yassine, S. Shirmohammadi, “Food Calorie Measurement Using Deep Learning Neural Network”, IEEE, pp. 1–6, 2016

[21] G. Ciocca,P. Napoletano,R. Schettini,”Food recognition:a new dataset, experiments and results”, IEEE, J.Niomed Health Inform , 2016

[22] P. Paritosh, D. Akella, T. Bappadi, N. Puhan, “FoodNet: recognizing foods using ensemble of deep networks”, IEEE Signal Process,2017

[23] A. Eduardo, B. Marc, R. Petia, “Food recognition using fusion of classifiers based on CNNs”, Image Analysis and Processing, ICIAP, pp.213–224, 2017

[24] A. Eduardo, A., B. Marc, R. Petia, “Exploring food detection using CNNs”, Computer Aided Systems Theory,EUROCAST, pp. 339–347, 2017

[25] I. Sanasam, S. Ayan, K. Pritee, “Classification of food images through interactive image segmentation”, Asian Conference on Intelligent Information and Database Systems, pp. 519–528, 2018

[26] L. Yuzhen, “Food image recognition by using convolutional neural networks”, Vision and Pattern Recognition, 2019

[27] G. Manjunath,G. Shobha,J R Rajayogi, “Indian Food Image Classification with Transfer Learning”, International Conference on Computation System and Information Technology for Sustainable Solutions, IEEE, 2019

[28] S. Ashutosh, L. Yuan, T. Ebrahimi, “Food/Non-food Image Classification and Food Categorization using Pre-Trained GoogLeNet Model”, MADiMa '16: Proceedings of the 2nd International Workshop on Multimedia Assisted Dietary Management ,2016

[29] J. Nayak, K. Vakala, P. Dinesh, B. Naik, D. Pelusi, “Intelligent food processing:

Journey from artificial neural network to deep learning", Computer Science Review, 2020

[30] Shabbeer S., S. Ram, V. Pulabaigari, S. Mukherjee, "Impact of Fully Connected Layers on Performance of Convolutional Neural Networks for Image Classification", Neurocomputing, 2019

[31] Bansal A., C. Castillo, R. Ranjan, R. Chellappa, "The dos and donts for cnn-based face verification", IEEE, pp. 2545–2554, 2017

[32] Khanday O., S. Dadvandipour, M. Lone "Effect of filter sizes on image classification in CNN: a case study on CFIR10 and Fashion_MNIST datasets", IAES International Journal of Artificial Intelligence, 2021

[33] Azat H., B. Sekeroglu, K. Dimililer,"A Pre-study on the Layer Number Effect of Convolutional Neural Networks in Brain Tumor Classification", International Conference on Innovations in Intelligent Systems and Applications , 2021

[34] Maitra S., R. Ojha and K. Ghosh, "Impact of Convolutional Neural Network Input Parameters on Classification Performance", SDMIT , Mangalore, IEEE, 2018

[35] Mou J., J. Li, "Effects of Number of Filters of Convolutional Layers on Speech Recognition Model Accuracy" International Conference on Machine Learning and Applications , 971-978, IEEE, 2020

[36] Bera S., V. Shrivastava,"Effect of pooling strategy on convolutional neural network for classification of hyperspectral remote sensing images", IET Image Processing, pp.480-486, 2020

[37] T. Bylander, L. Tate, "Using validation sets to avoid overfilling in AdaBoost", Proceedings of the Nineteenth International Florida Artificial Intelligence Research Society Conference, pp. 11-13, 2006

[38] X. Ying, "An Overview of Overfitting and its Solutions",Journal of Physics Conference Series, 2019

[39] Nodes, T.A., Gallagher, N.C," Median filters: some modifications and their properties", IEEE Transaction on Speech Signal Process,pp. 739–746, 1982

[40] Satyabrata Biswal, Nilamani Bhoi, "A new filter für removal of salt and pepper noise", International Conference on Signal Processing, Image Processing and Pattern Recognition, 2013

[41] Hongjin Ma, Yufeng Nie, "A two-stage filter for removing salt-and-pepper noise using noise detector based on characteristic difference parameter and adaptive directional mean filter", Plos One, 2018

[42] S. Arastehfar, A. A. Pouyan, "A. Jalalian, An enhanced median filter for removing noise from MR images, Journal of AI and Data Mining", 2013

[43] C. Yin, A. McKay, "Introduction to modeling and simulation techniques",8th International Symposium on Computational Intelligence and Industrial Applications and 12th China-Japan International Workshop on Information

Technology and Control Applications, 2018

[44] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Alex Krizhevsky, “Dropout: A Simple Way to Prevent Neural Networks from Overfitting”, Journal of Machine Learning Research, 2014

[45] Leslie N. Smith,” Cyclical Learning Rates for Training Neural Networks”, IEEE Winter Conference on Applications of Computer Vision (WACV), 2017

[46] F. Lei, X. Liu, Q. Dai, B. Ling, “Shallow Convolutional neural network for image classification”, SN Applied Sciences, Springer Berlin Heidelberg, 2020

[47] Khanday O., S. Dadvandipour, M. Lone, “Effect of filter sizes on image classification in CNN: a case study on CFIR10 and Fashion MNIST datasets”, IAES International Journal of Artificial Intelligence, pp. 872~87, 2021

[48] Kandel I., M. Castelli, “The effect of batch size on the generalizability of the convolutional neural networks on a histopathology dataset”, ICT Express, pp. 312–315, 2020

[49] S. Ioffe, C. Szegedy, “Batch normalization: accelerating deep network training by reducing internal covariate shift”, Proceedings of the 32nd International Conference on International Conference on Machine Learning, pp.448–456, 2015

[50] Shiv Ram Dubey, Satish Kumar Singh, Bidyut Baran Chaudhuri, “Activation Functions in Deep Learning: A Comprehensive Survey and Benchmark”, Neurocomputing, 2022

[51] Zhilu Zhang, Mert R. Sabuncu, “Generalized Cross Entropy Loss for Training Deep Neural Networks with Noisy Labels”, 32nd Conference on Neural Information Processing Systems, 2018

[52] Dargan S., M. Kumar, M. Ayyagari, G. Kumar, “A Survey of Deep Learning and Its Applications: A New Paradigm to Machine Learning”, Archives of Computational Methods in Engineering, pp. 1071–1092, 2020

[53] He K., J. Sun, “Convolutional Neural Networks at Constrained Time Cost”, computer vision and pattern recognition, IEEE,2015

[54] Shekhar H., Seal S., S. Kedia, Amartya Guha, “Survey on applications of machine learning in the field of computer vision”, Emerging Technology in Modelling and Graphics, 2020

[55] D.Kingma, J. Ba, “Adam: A Method for Stochastic Optimization”, International Conference on Learning Representations ICLR, 2015

[56] Food20 dataset “food20 dataset (indian food) | Kaggle”

[57] L.Bossard, M. Guillaumin, and L. Van Gool, “Food-101 – Mining Discriminative Components with Random Forests”, European Conference on Computer Vision, ECCV 2014 pp. 446–461, 2014

[58] F. Bettina-Elena, C. Ioan “Food For Machine Learning Dataset”, ResearchGate, April 2020

[59] UEC FOOD 100,“http://foodcam.mobi/dataset100.html”