Chapter 2
LITERATURE STUDY

2.1 INTRODUCTION

Machine learning is a subset of artificial intelligence that allows machines to make predictions or to take decisions based on data. Machine learning algorithms make autonomous predictions by leveraging known properties of training data without requiring direct human intervention. By leveraging known properties of training data, machine learning algorithms can generalize from past observations to make informed predictions on new, unseen instances. This ability to autonomously learn and predict without explicit human intervention has made machine learning an indispensable tool in various domains.

Deep learning is a subset of machine learning. 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. Deep Learning has become famous for its impressive results in food image classification. 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.

The most used image classification method includes traditional machine learning methods or deep learning methods. Traditional machine learning methods depend on manual feature extraction and classification. There are a variety of factors that make manual feature extraction difficult. For instance, it is usually difficult to accurately predict the real meaning of an image which results in low classification accuracy. Food images have high intraclass variance and low inter-class variance due to which traditional machine learning approaches do not recognize complex features [2-3].

Deep learning methods can automatically learn food features through a deep neural network. This section describes all the recent articles on preprocessing, food image classification considering the type of food, different methods of classification, different types of networks and different frameworks of deep learning. It is followed by detailed study of the Convolutional neural network, time complexity of CNN model and Different food image datasets studied.

2.2 PREPROCESSING

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. In food images, it is very necessary to retain edges after pre-processing an image, as it is the factor that detects the shape of an image. Preprocessing is a technique that improves image quality by removing noise and unwanted objects from the background, making the image ready for further processing.

Images captured by sensors, spy satellites, cameras or downloaded from the internet or pictures taken in routine life contain a lot of noise. It is necessary to remove the image noise so the interesting content can be highlighted. 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. Filtering in image preprocessing is a method that removes noise from images.

There are many filtering techniques like median filter, mean filter, maximum filter and minimum filter. The following Table 2.1 describes the comparison of different filtering techniques.

Table 2.1 Comparison of various Filtering Techniques

Filter Name	Advantages	Disadvantages	When to Use
Mean Filter [6]	Smooths images by averaging neighboring pixel values.	May cause blurring of fine details and edges.	Use when the main objective is general noise reduction.
	Simple and computationally efficient.Can be effective for reducing	Less effective for preserving sharp features and textures.	Suitable for smoothing images with uniform noise.
Median Filter [7]	Excellent at removing impulse noise (salt-and-pepper noise).	May cause blurring of textures and gradual transitions.	Use when the primary goal is to remove salt-and-pepper noise.
	Preserves edges and fine details better than linear filters.	Less effective for reducing other types of noise.	
	Simple and computationally efficient.	Could not preserve edges for higher noise images.	
Maximum Filter [8]	Emphasizes bright regions and highlights image features.	May introduce halos or artifacts around strong edges.	Use when the goal is to enhance or detect bright regions/features.
	Useful for edge detection and enhancing image structures.	Less effective for noise reduction compared to other filters.	Suitable for applications such as edge detection or object
	Helps identify objects or regions with high intensity values.		
Minimum Filter[7-8]	Emphasizes dark regions and highlights image structures.	May introduce halos or artifacts around strong edges.	Use when the objective is to enhance or detect dark
	Useful for edge detection and enhancing low-intensity features.	Less effective for noise reduction compared to other filters.	Suitable for applications such as edge detection or object
	Helps identify objects or regions with low intensity values.		

The effectiveness and suitability of these filters depend on the specific characteristics of the image, the nature of the noise or features and the desired outcome. Many authors have attempted to implement various pre-processing methods for various types of datasets to remove noise and irregularities from the image so the main objects can be highlighted.

K. Ramat et al. proposed techniques for image denoising called hybrid lifting directional. The prime motive of the paper is to show how pre-processing techniques can increase classification accuracy for satellite images [47]. A discrete wavelet interpolation denoising technique has been developed which classifies the image using a support vector machine. The proposed method has been tested on medical images and satellite images to show its efficacy over conventional denoising techniques.

Maider et al. [48] have discussed various pre-processing techniques for hyperspectral images. The authors have discussed the main steps of pre-processing like dead pixels,

spikes, compression and spectral pre-processing, along with the types of images for which they have been implemented, with the benefits and drawbacks of each technique. The effect of preprocessing methods on CNN for recognizing facial expressions has been shown by Diah et al. [49]. The comparison has been done on pre-processing methods, namely cropping, face detection, resizing and adding noise, to decide which method gives the best classification accuracy. The performance of CNN has been improved by combining two techniques: cropping and adding noise. The classification accuracy has improved to 97.06% from 86.08%.

The importance of pre-processing for Magnetic Resonance classification and segmentation has been highlighted by J. Anitha et al. [50]. They have also worked with texture-based techniques. The proposed technique worked in three steps: mask generation, mask logical conversion and lastly, masking. The dataset has been collected from the Devaki scan center in Madurai and contains 540 images. The author has also discussed the convergence rate for the proposed approach. The experiment results witness that the accuracy is high and the processing time is low in the proposed approach.

A tutorial on various preprocessing methods for MNIST handwritten digit classification problems has been given by Siham et al. [51]. The importance of the various techniques has been analyzed on CNN, LeNet and Drop Connect together. The experiment results show that the combination of rotation and elastic improves accuracy. A novel preprocessing method has been proposed by Haozheng et al. [52] to improve automatic modulation classification. The RadioML2016.10a dataset has been used. The experiment results show a 10% improvement in accuracy for CNN using the proposed method. The experiments proved that combining the proposed method with the fine-tuned CNN gives the best accuracy.

An overview of various image preprocessing techniques for a wide range of medical imagery has been presented by P. Vasuki et al. [53]. The paper discusses preprocessing techniques for X-rays, fundus images and mammograms. According to the survey, every image is different in contrast and quality. The paper clearly states that preprocessing is the mandatory step before processing any image.

A comparative study on various techniques of preprocessing for image fusion based on CNN has been done by Jyoti et al. [54]. Image fusion is a technique that gives a different focus to a single image. Three filters have been used on three different datasets: medical, colour multi-focus dataset and infrared visual. The results show that median filters give the best classification accuracy on any dataset, while other filters give the second-best accuracy.

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. [8]. A comparison has been done on filters based on computational complexity and performance. The conventional filter is good for low noise but fails to preserve edges. The Database algorithm is good for images with low noise density.

A hybrid median filter for removing impulse noise from an image has been proposed by M. Narsimha et al. [56]. The proposed filter is a nonlinear filter, an improved version of the median filter, which helps to remove noise and preserve main features. The

experiment has been implemented in MATLAB. According to the results, the hybrid median filter is simple to understand and performs better than the median filter. According to the author, the disadvantage of the proposed filter is that it has a high computational cost, so to avoid that, new filters should be developed.

A new filter which is a hybrid combination of min and max filter has been implemented by Prity et al. [57] which is an extension to the median filter. The filter is useful for removing impulse noise from the image. The proposed method is implemented on various color images. The algorithm uses variable window sizes. The corrupted pixels are recognized by their local extrema intensity. The authors claim that the presented method can remove up to 90% of the noise from the image.

The effects of linear and nonlinear filters on the preprocessing of MRI images have been studied by Suhas et al. [58]. Different filters have been applied to MRI brain and spinal cord images to compare the results. It has been concluded from the experimental results that the proposed method will increase the accuracy of classification more than other existing filters. A similar filter was also developed by A. Jalalian et al. [59] for MRI images.

A new preprocessing technique for improving skin lesion classification accuracy has been implemented by Behnam et al. [60]. The performance has been compared with two datasets: raw and ROI-extracted images. The empirical study shows that the training of the CNN model with the proposed method can improve accuracy and reduce the training time. The reason is that the unwanted background has been removed and only the necessary details of an image have been passed to the classifier.

An image-processing framework for the diagnosis of retinal disease using three different CNN models has been proposed by Akash et al. [61]. The SD-OCT dataset, built from 10 different categories of retinal images, has been used. The work successfully detected four diseases from OCT images. The authors have suggested that the limitation of the work is that biological variations in the eye cannot be detected and future models can be proposed for the same.

A dataset for breast cancer has been created by Sami et al using pre-processing the images [62]. The main idea of this work is to create a dataset so the operational time for the used network can be saved and accuracy can be improved. The method has mainly three parts: 1) the background, 2) removal of pectoral muscle and 3) image enhancements. The proposed method can remove 100% of the image background.

Youlian Zhu et al. have proposed an improvement in the existing median filter which add a mask over the image [63]. The proposed algorithm reduces time complexity to O(N) and increases the performance. The author made a comparison of Lena's images on the existing filter, fast median filter algorithm and proposed method.

A new filter with three stages for removing impulse noise has been developed by Varatharajan et al. [64]. The mammogram images have been used for the experiment. The tri-state value is replaced by the decision tree. The tri-state value can be modified by mean or midpoint. The noise pixels are replaced by nonlinear asymmetrically trimmed values. The proposed method performs better for noise elimination at a high density of images.

It has been found from studies that the best filter for removing impulse noise is a median filter [48-53]. Many researchers have used a median filter and its variant for

efficient noise removal. But most techniques have difficulty in removing impulse noise while preserving edges and contours. From the above study the following conclusions can be made.

Research Findings

- Median filter gives false decisions in high noise density images hence unsuitable for images with noise densities of 80% and above.
- Due to the automatic modulation approach median filter miss classified pixels and visually unpleasant filtered images with 50% noise density.
- Median filter is good for low noise density but fails to preserve edges.

The Median filter operates by replacing each pixel in an image with the median value of its neighboring pixels. The main problem with the traditional median filters is that each pixel is filtered without regard to whether it is a noisy pixel or a noise-free pixel [8]. The median filter is effective in reducing impulse noise but it may introduce some blurring or loss of texture and gradual transitions. The median filter may not effectively preserve

edges when the noise level exceeds 50%. In fact, the median filter can cause blurring of edges and fine details when the noise is substantial. The reason is that the filter considers the median value within the window and replaces the pixel with that value, irrespective of its relationship to neighboring pixels. As a result, it can blur edges and reduce their sharpness, especially when the noise level is high.

To overcome this limitation of the median filter, a novel median-based method has been proposed which is able to differentiate between noisy and healthy pixels. A new algorithm for noise removal is proposed that adds features to the median filter and merges both mean and median filters to calculate a more accurate pixel value from noisy images. It identifies corrupted pixels first and then removes impulse noise to improve the quality of food images.

2.3 FOOD SEGMENTATION

In computer vision, image segmentation is the process of partitioning a digital image into multiple segments (sets of pixels, also known as super pixels) [20] [65-66][77-82]. The goal of segmentation is used to simplify and/or change the representation of an image to make it more meaningful and easier to analyze. Image segmentation is the process of assigning a label to every pixel in an image such that pixels with the same label share certain characteristics. Segmentation is the last step before food classification where various food images are identified. There are different types of segmentation techniques and they can be divided mainly into two categories: layer-based segmentation methods and block-based segmentation methods. Block-based segmentation can be partitioned into region-based segmentation and edge-based segmentation. Other types are segmentation based on weakly supervised learning in CNN, threshold segmentation methods, segmentation based on clustering, etc.

Segmentation makes the boundary detection of irregular food portions easy and hence gives better detection of the food portion. Matsuda et al. [67] discussed separating the food images from the background regardless of the lighting conditions or if the food is mixed or not. According to Yang et al. [68], it is very difficult to separate those food images which does not contain any specific attribute. Ramadevi et al. [69] showed synergy between segmentation and object recognition is done using the EM algorithm, OSTU and genetic algorithm. They have also discussed the difference between region-based and edge-based segmentation.

Yan Hao [70] has evaluated and recap different image segmentation algorithms and compared them with pros and cons. A similar approach has been used by Shakuntala and Surendra [71] and Vairaprakash and Subbu [72], After studying different types of image segmentation, they have concluded that “all the works done in the field of image segmentation are needed to be monitored manually”. There is no such method which can detect the objects with precision and without any database, which obviously takes time to get build.”

W. Liming et al. [73] had developed a method that is a combination of top-down recognition with bottom-up image segmentation. On the other hand, Wataru and Keiji [74] have proposed neither a traditional approach nor the fully convolutional approach. The approach is a combination of fully convolutional networks and a back-propagation-based approach.

Bryan et al. [75] computed multiple segmentations of images and then learned the object classes to choose the correct segmentation. A similar approach can also be found by Zhu et al. [76], They have combined two concepts: A set of segmented objects can be partitioned into classes based on global and local features; and perceptually, similar object classes can be used to assess the accuracy of image segmentation. They have shown improvement in the accuracy of segmenting food images using a segmentation compared to normalized cut without classifier feedback when there is no prior information about the scene. This idea has improved the overall accuracy of classification.

It is concluded from the literature survey that it is difficult to find a segmentation way to adapt with all images. At present, the research of image segmentation theory is not perfect and there are still many practical problems in applied research. New efficient solutions are still required particularly in annotation and data augmentation to improve the performance of image segmentation. Segmentation techniques are not suitable for food due to complex food structure and variations in the appearance of food. Achieving perfect food segmentation in all scenarios is a challenging task and the performance of segmentation techniques can vary depending on the specific image characteristics and application requirements.

2.4 FOOD CLASSIFICATION

Image recognition can be achieved with a machine learning-based approach or deep learning-based approach [45]. Recently, deep learning has become very famous as it gives impressive results. It is inspired by the structure and function of the human brain's neural networks [83]. It is probably the best approach in cases when we do not have enough pre-defined features. Many research works have been done on food image

recognition that has used convolution neural networks with different combined approaches and different datasets. Table 2.2 describes different methods for food classifications proposed and used by different researchers. In the Table, Top1 accuracy is the convolutional accuracy of model prediction that is exactly as the expected answer. Top5 accuracy is any one of model's highest five probabilities answer that match with the expected answer.

Table 2.2 Different Methods of Food Classification

Sr. No.	Reference	Approach	Dataset (Type of Food)	Top1 Accuracy (%)	Top5 Accuracy (%)	Remarks
1	A food recognition system for diabetic patients based on an optimized bag-of-features model [90]	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 (Selected food such as Bread, Cheese, Egg products, Legumes, Meat, Pizza, Potatoes, Rice)	75.0	-	Some of Examples of misclassification of images has been presented. The proposed model gives very less classification accuracy.
2	Food-101-mining discriminative components with random forest in computer vision [91]	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 (Popular food in USA)	56.4	-	A novel large-scale benchmark dataset has been introduced for the recognition of food.
3	Food Image Recognition with Deep Convolutional Features [45]	Used deep CNN with Fisher Vector with HOG and color patches.	UEC-Food-100 (Popular Japanese Food)	72.3	92.0	DCNN features can boost classification performance by integrating it with conventional features. The pre trained DCNN has 60 million floating values which needs to be reduced to make the application suitable for mobile devices.
4	Food Image Recognition Using Deep	Used deep CNN for food photo recognition task in	UEC-Food-100 (Popular	78.7	-	In addition to high classification accuracy, DCNN

Sr. No.	Reference	Approach	Dataset (Type of Food)	Top1 Accuracy (%)	Top5 Accuracy (%)	Remarks
	Convolutional Network with Pretraining and Fine-Tuning [92]	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.	Japanese Food)			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 (Famous foods in Japan & other Countries)	67.57	89.0	
5	Food recognition for dietary assessment using deep convolutional neural networks [93]	Used a deep CNN 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 [94]	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(derived from FOOD101)	76.0	-	
			Menu-Match(Food from three restaurants (Asian, Italian, Soup of 10 types))	81.4	-	
7	Food image recognition using very deep convolutional networks [95]	Used deep learning approach for the classification and fine-tuned the image recognition architecture Inception.	UEC-Food-100	81.5	97.3	One important result of the study is showing that fine-tuning a pre-trained network can achieve good results in a reasonable time. The achieved results are still in the preliminary stage derived from google pre-trained network. The proposed approach
			UEC-Food-256	76.2	92.6	
			Food-101	88.3	96.9	

Sr. No.	Reference	Approach	Dataset (Type of Food)	Top1 Accuracy (%)	Top5 Accuracy (%)	Remarks
						needs to be modified in order to achieve better result.
8	Deep-Based Ingredient Recognition for Cooking Recipe Retrieval [96]	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, proposed 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 (Popular Chinese dishes from "go cooking" and web)	82.1	95.9	
9	Deep food: deep learning-based food image recognition for computer-aided dietary assessment. [97]	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	The limitation is that to improve the performance of 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 (Famous foods in Japan & other Countries)	54.7	81.5	
10	Food Calorie Measurement Using Deep Learning Neural Network [98]	Used the Graph cut method and uses Deep convolution Neural Network to classify food images and have achieved	Own database with 10000 high resolution images	99.0	-	Mixed food portion images have not been considered.

Sr. No.	Reference	Approach	Dataset (Type of Food)	Top1 Accuracy (%)	Top5 Accuracy (%)	Remarks
		remarkable accuracy for a single food image.				
11	Food recognition: a new dataset, experiments, and results [99]	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 (Pictures captured by a digital camera in dining hall)	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. The limitation of the system is that if the region of interest is overlapping the system will fail to achieve the expected results.
12	FoodNet: recognizing foods using an ensemble of deep networks [100]	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(Mix of eastern and western meals)	72.12	91.61	
13	Food recognition using a fusion of classifiers based on CNNs [101]	Proposed a CNNs Fusion approach based on the concepts of decision templates and decision profiles and their similarity that improves the classification performance with respect to using CNN models separately.	Food-101 Food-11 (selected food such as Bread, Dairy product, Dessert, Egg, Fried food, Meat, Noodles/Pasta)	-	-	combination of multiple classifiers based on different convolutional models that complement each other hence improving performance.
14	Exploring food detection	Proposed a model that uses	FCD (Italian food)	98.81		The proposed method has been

Sr. No.	Reference	Approach	Dataset (Type of Food)	Top1 Accuracy (%)	Top5 Accuracy (%)	Remarks
	using CNNs [102]	GoogleNet for feature extraction, PCA for feature selection, and SVM for classification.	Ragusa DS	95.78		implemented on a very small dataset. For future work the performance can be evaluated on larger dataset containing a much wider range of dishes.
15	Classification of food images through interactive image segmentation [103]	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 [104]	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

From the above Table, it is observed that many authors have tried to classify different types of food from available food datasets using various deep learning techniques.

In previous years, several approaches have been proposed to classify food from images. Research shows that deep learning gives very effective results while dealing with a large dataset of images [26-27] [84-87][89-104]. In the past few years, many researchers have also focused on the problem of classifying the different types of food images.

Sawal et al. has proposed a deep CNN for a new dataset containing Malaysian food items and Food-101 [105]. There are 3300 food items belonging to 11 food classes. A 24-layer

model has been proposed and compared with VGG19. It was noticed that the proposed model performed better compared to the pre-trained model VGG19.

A small CNN has been constructed by Jianing Teng et al. for the recognition of Chinese food [106]. For this purpose, a Chinese food dataset has been created for 25 different food items, which has a total of 8734 Chinese food images. An effort was made to construct a five-layer CNN that performs similar tasks as a bag of features. The results were compared to the original Bag-of-Feature model to examine the similarities between the two models and to identify the factors influencing the model's performance. The proposed model is able to achieve 97.12% of top-1 accuracy and 99.86% of top-5 accuracy.

A detailed review analysis has been presented by Janmenjoy Nayak et al. on the journey of food processing since it evolved [107]. Starting with the real-world problems faced by researchers in food processing in the early days, how advanced techniques like machine learning and deep learning play a key role in all the problems have been discussed. Types of artificial neural networks used in food processing with the application of deep learning in food processing has been discussed. A year-wise critical survey has been conducted on the published articles for both Artificial Neural Networks (ANN) and Deep Neural Networks (DNN). Also, the analysis has been done on the number of conferences and the number of journal papers published for both ANN and DNN. A database-wise (like Elsevier, Springer, Wiley, etc.) unique analysis has been done on papers published on ANN and DNN on food processing.

A critical investigation has been done on how the regularization scheme plays an important role in image classification by Xuhong Li et al. Some of the regularization techniques have been proposed and tested along with transfer learning in convolutional neural networks [108]. Also, the standard L2-SP scheme for inductive transfer learning has been defined, which needs future improvements.

An empirical study on how convolutional neural networks can be set up for food image recognition has been carried out by Yi Sen et al. [109] Many issues like how to prepare a

dataset, which CNN architecture can be chosen, which optimizer to use, how many images should be there in a dataset for best performance and which image augmentation techniques should be used have been addressed. A unique conclusion has been given that a minimum of 300 images per class is required for optimal performance of the model. They have concluded that data augmentation techniques are more effective and deeper networks like Xception and Inceptionv3 give better results.

David et al. tried to classify food images in the FOOD-101 dataset using CNN and were able to achieve 86.97% testing accuracy [110]. They have used a 2D convolutional layer along with the Max-pooling function for classifying the food items. A comparison of the proposed model was made with all the work done by different authors on the same dataset, FOOD-101. It has been quite evident that the proposed model outperforms in terms of accuracy. They have concluded that CNN gives better results when the dataset is large.

Narit et al. [111] have designed a prediction model for classifying Thai fast food images. A dataset has been created with 3960 Thai food images of 11 food items. By fine-tuning the Inception V3 model, which was already trained on the ImageNet dataset [112], the researchers were able to achieve an 88.33% classification average accuracy.

A transfer learning-based approach has been implemented by Jianing et al. [113] for food detection tasks that can achieve high accuracy even on small networks. They combined image classification with object detection. They have used the datasets Food-5K, containing food and non-food images, Japanese food image datasets UECFood100 and UECFood256. It has been found that generalization performance can be greatly improved by initializing with transferred features.

The cross-modal alignment and transfer network known as ATNet was presented by Lei et al. [114]. A Chinese food dataset vireoFood-172 and western food item dataset ingredients-101 were used. AtNet achieved 86.2%, 87.3% top 1 accuracy and 96.6%, 96.8% top 5 accuracy for VireoFood-172 and Ingredients-101, respectively. The author

has suggested that stronger transfer learning techniques can be applied, which can lead to an increase in the performance of the model.

Masud et al. have introduced a new 22-class database based on Australian dietary guidelines named Food-22 [115]. A pretrained DCNN has been used in two ways, namely transfer learning to retrain DCNN and extracting features to train conventional classifiers. Transfer learning has been implemented on ResNet-50. A similar level of accuracy has been achieved by both methods, but the training time for the second method is lower. Also, a comparison has been made with the newly created dataset by using different classification techniques. As the proposed framework eliminates the need for extracting hand-crafted features, it gives better accuracy as compared to similar existing methods in the literature.

A traditional Bengali food image dataset consisting of 7 items has been created by Asif et al. [116]. There are a total of 1089 images belonging to seven different classes of Bengali food. To increase the size of the dataset, Gaussian noise and rotations have been applied, which increase the images of the training set to 2619. To avoid overfitting, various real-time augmentation techniques have been applied. A model from scratch has been developed to classify Bengali food images, that was able to achieve 86% testing accuracy, which does not give prominent accuracy on the newly made dataset. Hence Transfer learning, along with fine-tuning, has been applied to the pre-trained model VGG16. A remarkable accuracy of 98% has been achieved on the traditional Bengali food image dataset by implementing the concept of transfer learning along with fine-tuning.

Different concepts for classifying food images on the ETH-101 dataset have been used by Sirawan et al. [117]. The authors have made certain changes like replacing the average pooling layer with the global average pooling layer to avoid overfitting; batch normalization; and drop-out layers have been added in the prebuilt MobileNet architecture. They name the new model the "modified MobileNet architecture." According to the results, the proposed modified MobileNet architecture performs better as compared to the original MobileNet architecture.

Sapna et al. [118] have investigated food image classification accuracy using pre-trained SqueezeNet and VGG-16 CNN on the FOOD-101 dataset. The concept of transfer learning, along with data augmentation and fine-tuning of the hyperparameters, has been implemented to obtain better performance. Even with fewer parameters, SqueezeNet was able to achieve 77.20% testing accuracy. VGG16 achieved an 85.07% testing accuracy while using a deeper network.

An experiment was done to classify Javanese traditional food items by Puteri et al. [119]. The dataset has been created for 794 images of 17 different traditional food items from Indonesia. The dataset was preprocessed and classified with different classifiers, namely KNN, SVM, LDA, Random Forest, Naïve Bayes and deep learning using the Resnet50 model. It has been proved from the experiment results that the Random Forest classifier achieved the best accuracy compared to the other classifiers.

An attempt was made by Pengcheng et al. [120] to fine-tune the Faster R-CNN network on the Dish-233 food dataset. The dataset is a subset of the dish dataset, including 233 dishes and 49,168 images. The experiment showed that fine-tuning the R-CNN method improved performance by 5% as compared to other existing methods and the average accuracy achieved was 75.40%.

An effort is made to classify Indian food images by Shobha et al. [121]. The dataset contains twenty different food classes having 500 food items in each class. Instead of developing a model from scratch, the concept of transfer learning has been implemented on several models, namely VGG16, ResNet, InceptionV3 and VGG19. The concept of transfer learning saves computational cost and time. The result showed that Inception v3 has the highest accuracy of 87.9%.

A multi-layered deep convolutional neural network has been developed by Paritosh et al. [122], which improves efficiency by taking advantage of existing deep network features. An Indian food image dataset has been created consisting of 50 different food items, each of which contains 100 food images. The experiment was conducted on Food-101 and the Indian Food Image Database. They have concluded that the proposed

Ensemble net model gives better accuracy as compared to fine-tuned models AlexNet, GoogleNet and ResNet on both ETH-101 and the Indian food dataset.

The combination of support vector machine and bag-of-feature has been first evaluated by Yuzhen Lu, with an accuracy of 56% [123]. A small dataset having ten food classes and a total of 5822 images has been created. A five-layer CNN has been constructed and achieved 74% classification accuracy, which is better than the combined approach of the Bag-of-feature and support vector machine. However, the constructed CNN model suffered from an overfitting issue due to the small number of images in the dataset.

It has been concluded from literature survey that a strong collection of images, Dataset, is the key element to achieve best classification accuracy [45]. A dataset is created considering the number of food classes and the type of food. Table 5 describes the number of available datasets that has been used by different researchers with the food content including the number of food classes and number of food images. Since deep learning is data hungry, a large collection of food images is required to train a food-classification model. Food image datasets vary in many aspects such as, a single food image, mixed food image, non-mixed food, several food groups, liquid food image, type of cuisine and total images per food class. Table 2.3 summarizes different food image databases with their respective features.

Table 2.3 Summary of food image databases

Name of dataset	Year	#images	#Food classes	Food content
Food85[164]	2010	8500	85	Japanese Food
Chen [165]	2012	5000	50	Chinese Food
UNIMIB2016[166]	2016	1027	73	Pictures captured by a digital camera in the dining hall
Food524DB [167]	2017	247636	524	Merging food classes from existing database vireo, food-101, food50 & modified version of UECFOOD256
FFOcat [168]	2018	58962	156	Selected Food images Downloaded from Web
Foodx-251[169]	2019	158000	251	Selected food items like cakes, sandwiches, puddings, pasta, soups, etc.

Name of dataset	Year	#images	#Food classes	Food content
FooDD [170]	2015	3000	30	Single and mixed food images including fruits
PFID [171]	2009	1098	61	Fast food items from USA
TADA [172]	2009	256+ 50replica	-	Common food in USA
Food50[173]	2009	5000	50	Japanese Food
UEC-Food100[174]	2012	9060	100	Popular Japanese Food
Food101[157]	2014	101000	101	Popular Food in USA
UECFood256[175]	2014	31397	256	Famous foods in Japan and other Countries
UNICT-FD889[176]	2014	3583	899	Different Nationality dishes like Italian, Thai, Japanese, etc.
Diabetes2[177]	2014	4868	11	Selected Food
New Dataset [12]	2014	5000	11	Central European Food
Menu-match [15]	2015	646	41	Food from three restaurants (Asian, Italian, Soup of 10 types)
VIREOFood-172[114]	2016	110241	172	Popular Chinese dishes from “go cooking” and web. Eight major groups are vegetable, soup,egg,meat,seafood,fish,Bean product etc.
ChineseFoodNet [21]	2017	185000	280	food images either taken from real dishes or recipe pictures or selfies.
Food20	2020	2000	20	Indian Food
Indian-100	2019	5000	50	Indian Food
FFML	2020	1281	424	Romanian Food

Other than these, Anthimopoulos et al. [2] used one visual database created with 5000 food images and organized into 11 classes reflecting the nutritional habits in central Europe in 2014. Also, in 2017, Paritosh et al. [32] proposed work on the Indian food dataset containing 100 Indian food images of 50 different classes. It is the first Indian food dataset that is available to download.

It has been found that much work has been done on Chinese, Japanese and American fast-food items, but the essence of Gujarati food items is missing. It has been observed from the literature review that there is no dataset available for Gujarati food items.

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 research work is mainly focusing on the accurate classification of Gujarati food with high efficiency.

2.5 Deep Learning Framework

A deep learning framework is an interface or a tool used to combine deep learning algorithms, pre-built models and optimized components. Instead of writing hundreds of lines of code, deep learning frameworks build a model quickly. The framework provides good community support and parallelizes the process to reduce computations [11-25].

Some of the famous deep learning Frameworks are Torch, Theano, Tensorflow, Chainer, Keras, Apache Singa, MXNet, Caffe, Microsoft Cognitive Toolkit CNTK, Deep Learning 4j and Neon. The detailed comparison of these tools is given in Table 2.4.

The frameworks are studied from the date when the first and stable version was released, language supported, operating system supported, type of library support and the support for GPU (Graphics Processing Unit), CPU (Central Processing Unit) or TPU (Tensor Processing Unit).

The list of deep learning frameworks is very exhaustive. Every few months new deep learning frameworks are introduced. The frameworks which support both the CNN and RNN models are listed in Table 2.4. This research work has used Tensorflow and Keras frameworks.

Table 2.4 Deep Learning Frameworks

Name	Initial Release	Stable Release	Language Supported	Operating System Supported	Type of Library Support	Support for GPU/CPU/TPU
Torch [11]	2002	2017	Lua,LuaJIT,C, CUDA,C++	Linux,Android,MacOS X,iOS	Deep Learning & Machine Learning	CPU & GPU
Theano[13]	2007	2019	Python,CUDA	Linux,MacOS,Windows	Machine Learning	CPU & GPU
Tensorflow[14]	2015	2020	Python, C++, CUDA	Linux,MacOS,Windows,Android	Machine Learning	CPU & GPU and optimize for TPU
Chainer [16]	2015	2019	Python	-	Deep Learning	Best for GPU
Keras [17]	2015	2019	Python	Ios,Android	Deep Learning & Machine Learning	GPU & TPU
Apache Singa[18]	2015	2020	C++, Python,Java	Linux,macOS,Windows	Machine Learning	CPU & GPU
MXNet[19]	-	2020	C++, Python,R,Java ,Julia,Javascript,Scala,Go,Perl	Windows,macOS,Linux	Deep Learning & Machine Learning	GPU
Caffe[22]	-	2017	C++	Linux,macOS,Windows	Deep Learning	CPU & GPU
Microsoft Cognitive Toolkit CNTK[1]	2016	2019	C++	Windows,Linux	Deep Learning & Machine Learning	GPU
Deeplearning4j [24]	-	2019	Java,CUDA,C, C++	Linux,macOS,windows,Android,iOS	Deep Learning & Machine Learning	CPU & GPU
Neon[25]	-	2018	Python	-	Deep Learning	-

2.6 Convolutional Neural Network

A neural network is a set of neural nodes. A CNN is divided into many layers where each node is connected to all the nodes of the previous layer. Convolutional Neural Network (CNN) is the main model of Deep Learning networks to do image recognition and image classification. CNN takes input, processes and classifies it under predefined categories. The main advantages of CNN are : parameter sharing, sparse interactions and equivalent

[1] https://www.microsoft.com/en-us/research/product/cognitive-toolkit/

representations [2]. 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 [26-32].

A general CNN architecture comprises alternate arrangements of convolution and pooling layers followed by one or more fully connected layers at last as shown in Fig.2.1

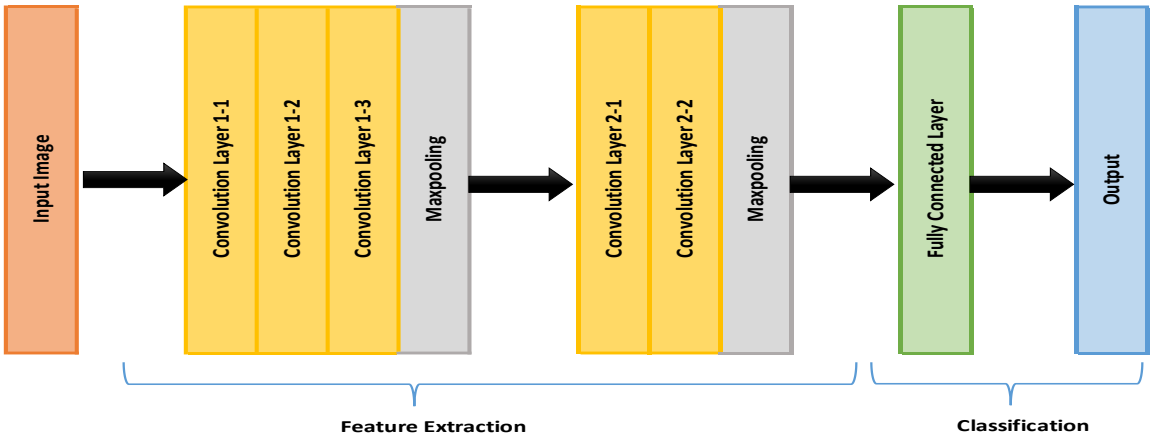

Fig. 2.1 CNN Architecture

In addition to these layers, different activation functions, optimizers, loss functions and various regulatory functions such as dropout and batch normalization are also useful to optimize the CNN performance. The arrangements of these components play a major role in designing new CNN architecture. This section briefly discusses the role of these components in CNN architecture.

2.6.1 Input Layer

Input layer contains the input given to the model which generally be an image or sequence of images. The input given in the layer can be either grayscale image or RGB image and is made up of pixels. 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 input layer can be added with input shape. The input layer can be added by using the following command.

Input_layer = Input_shape(height, weight, channels) (2.1)

In this command input_shape is the size of the input image in terms of height, width and channels. Channels represent the number of color channels eg. 1 for Grayscale image, 2 for custom image and 3 for RGB images.

2.6.2 Convolutional Layer

Convolutional layer applies filter to input image to extract features and hence also called as a feature extractor layer. In this layer, a filter will pass over the image, pixels are scanned and then a feature map is created which helps to classify the input image. Each feature belongs to some class. The filter essentially goes over the input image and is called convolving. We have to include activation functions in this layer too. The operation of the convolutional layer can be defined as follows.

$$\boldsymbol{f_l^n(u,v) = \sum_x \sum_{p,q} i_x\ (p,q)\ .e_l^n\ (a,b)} \tag{2.2}$$

Where $\boldsymbol{i_x\ (p,q)}$ is an element of input image $\boldsymbol{i_x}$. It is element wise multiplied by $\boldsymbol{e_l^n\ (a,b)}$ which is index of $\boldsymbol{n^{th}}$convolution Kernal $\boldsymbol{n_l}$of the $\boldsymbol{l^{th}}$ layer. Where output feature map of $\boldsymbol{l^{th}}$ convolutional operation can be defined as follows.

$$\boldsymbol{F_l^n = \left[f_l^n\ (1,1), \ldots \ldots f_l^n\ (u,v); \ldots, f_l^k\ (u,v)\right]} \tag{2.3}$$

$\boldsymbol{Where\ f_l^n\ (u,v)}$ is $\boldsymbol{(u,v)}$ element of feature matrix

$\boldsymbol{i_x\ \ (p,q)}$ is $\boldsymbol{(p,q)}$ Element of $\boldsymbol{x^{th}}$ channel of image

$\boldsymbol{f_l^k\ (u,v) =\ (u,v)}$ element of $\boldsymbol{k^{th}}$ kernal of $\boldsymbol{l^{th}}$ layer.

Convolutional operation is able to share weight and hence, an image with a different set of features can be extracted by sliding kernel. This makes convolutional operation efficient as compared to fully connected layers [33]. The convolutional operation can be categorized further by the type and size of filters, type of padding, kernel size and type of activation function used in the layer. The typical structure of the pooling layer can be defined as follows:

model.add (Conv2D(num_filters,kernel_size, padding, activation_function) (2.4)

In this command, filters are matrices by which we can extract the features of an image. It is a learnable parameter. Kernel size represents height and width of kernel. The size of the kernel influences the region of the input evaluated at each step of the convolution. If Padding = “same”, the output of the image has the same size as the input image. If there is a difference in size, padding should be set to “valid” instead of “same”. The activation function of a node defines the output of that node given an input or set of inputs.

2.6.3 Pooling Layer

The next layer, the pooling layer helps to reduce the spatial dimensions (width and height) of the input volume, while retaining the most important features, and hence, also known as down sampling or sub sampling. It reduces the number of pixels while not using important information. It is applied after the convolutional operation and reduces the dimensionality of each feature map but retains the most important information. There are three types of pooling as define below.

1. Maximum Pooling 2. Average Pooling 3. Minimum Pooling

1. Maximum Pooling: It computes the maximum value of the feature in the feature map covered by kernel/filter. It is retaining the strongest of the pixels while ignoring the weaker ones.

2. Average Pooling: It computes the average value of the feature in the feature map covered by kernel/filter.

3. Minimum Pooling: It computes the minimum value of the feature in the feature map covered by kernel/filter.

Any pooling operation can be defined as

$$\boldsymbol{U_n^l = P_g\left(Z_n^l\right)} \tag{2.5}$$

Where, U_n^l = pooled feature map of n^{th} layer for l^{th} input feature map Z_n^l

$P_g\ (.)$ = pooling operation type.

The typical structure of the pooling layer can be defined as follows:

Model.add (MaxPool2D(pool_size, strides) (2.6)

Here, pool_size defines the size of pooling window. Stride means a number of dimensions in pooling or step size for moving the pooling window [1].

2.6.4 Fully Connected Layer

This layer also known as the dense layer. There could be more than one dense layer. The first fully connected layer takes the input from the feature analysis and applies weights to predict the correct label. In this layer, every neuron in the previous layer is connected to every neuron in the next layer. It is the final layer where the classification actually happens. The fully connected layer can be added by using the following command.

model.add (Dense(units, activation)) (2.7)

In this command, units specifies the number of neurons in the fully connected layer. The activation defines which activation function to be applied to the fully connected layer.

2.6.5 Output Layer

The output layer of a neural network computes the final probabilities for each label in a classification task by performing a linear transformation. The output layer is one where we get the predicted classes. The output layer can be added by using the following command.

output_layer = Dense (units, activation) (2.8)

In this command units represent the number of output units or classes in specific problem. The activation defines which activation function to be applied to the output layer.

2.6.6 Dropout

Intentionally dropping data from a neural network is a technique used for improving processing speed and reducing the time to obtain results known as dropout. Dropout is

a regularization technique that improves generalization by skipping connections with a certain probability. It is used to avoid overfitting problem. Randomly dropping some of the connections produces several thinned networks and lastly, one final network is selected with small weights [33]. Dropout can be added before any fully connected layer including output layer. The dropout can be added by using the following command.

Dropout_layer = Dropout(dropout_rate) (2.9)

In this command the dropout rate defines the fraction of input units that randomly sets to 0 for training. The rate typically vary between 0 and 1. Here 0 means no units are dropped out and 1 means all units are dropped out.

2.6.7 Classification of CNN

There are various pre-Built models that exists for CNN like LeNet, AlexNet, GoogleNet, VGGNet, ResNet50 as discussed below.

- **LeNet:** This architecture was proposed in 1998 by LeCun et al. [34] and was primarily meant for OCR. It is the first popular CNN architecture originally trained to classify handwritten digits. The image input size is 28 X 28. A number of layers, Feature map, size of the image, kernel size stride and activation function used at each layer has been shown in following table 6. It is a deep neural network which has 60 million Parameters. The summary of the Lenet architecture is as shown in Table 2.5.

Table 2.5 Lenet Architecture

Layer	Layer Type	Feature Maps	Size	Kernel Size	Stride	Activation
Input	Image	1	32X32	-	-	-
1	Convolution	6	28X28	5X5	1	Tanh
2	Average Pooling	6	14X14	2X2	2	-
3	Convolution	16	10x10	5X5	1	Tanh
4	Average Pooling	16	5x5	2X2	2	-
5	Convolution	120	1x1	5X5	1	Tanh
6	Fully Connected	-	84	-	-	Tanh
Output	Fully Connected	-	10	-	-	Softmax

- **Alexnet:** This architecture was proposed by Alex Krizhevsky et al. [35] in 2012. Alexnet won the ILSVRC-2012 competition and achieved a winning top-5 test error rate of 15.3%, compared to 26.2 % achieved by the second-best entry. The architecture has five convolutional Layers and three fully connected layers.

  Alexnet is a deeper architecture having more convolutional layers and more filters per layer. Dropout has been added to avoid overfitting. The summary of the Alexnet architecture has been shown in Table 2.6.

Table 2.6 Alexnet Architecture

Layer	Feature Map	Size	Kernel Size	Stride	Activation
Image	1	227X227X3	-	-	-
Convolution	96	55X55X96	11X11	4	ReLU
Max Pooling	96	27X27X96	3X3	2	-
Convolution	256	27X27X256	5X5	1	ReLU
Max Pooling	256	13X13X256	3X3	2	-
Convolution	384	13X13X384	3X3	1	ReLU
Convolution	384	13X13X384	3X3	1	ReLU

- **GoogleNet:** This model introduced a new module known as inception modules and hence, also known as Inception. GoogleNet is one of the models which grab attention and winner of ImageNet contest in 2014. Inception modules used image distortion and batch normalization.

  Batch normalization is an appreciable technique, which is deployed in GoogleNet for improving the speed, performance and stability. This architecture is deep with a total of 22 layers and having 4 million parameters [36]. The summary of the GoogleNet architecture has been shown in Table 2.7.

Table 2.7 GoogleNet Architecture

Layer	Feature Map Size	Kernel Size	Stride	Activation
Image	224X224X3	-	-	-
Convolution	112X112X64	7X7	2	ReLU
Max Pooling	56X56X64	3X3	2	
Convolution	56X56X64	1X1	1	ReLU
Convolution	56X56X192	3X3	1	ReLU

Layer	Feature Map Size	Kernel Size	Stride	Activation
Max Pooling	28X28X192	3X3	2	-
Inception Module	28X28X256	-	-	-
Inception Module	28X28X840	-	-	-
Max Pooling	14X14X480	3X3	2	
Inception Module	14X14X512	-	-	-
Inception Module	14X14X512	-	-	-
Inception Module	14X14X512	-	-	-
Inception Module	14X14X528	-	-	-
Inception Module	14X14X832	-	-	-
Max Pooling	7X7X832	3X3	2	-
Inception Module	7X7X832	-	-	-
Inception Module	7X7X1024	-	-	-
Average Pooling	1X1X1024	7X7	1	-
Dropout	1X1X1024	7X7	1	-
Fully Connected	1024	-	-	ReLU
Fully Connected	1000	-	-	ReLU
Fully Connected	1000	-	-	Softmax

- **VGGNet:** This is one of the most preferred CNN architectures in the recent past developed by Simonyan and Zisserman in 2014. There are two models: VGG16 and VGG19 [37]. VGG16 has 16 convolutional layers with 138 million of parameters. VGG19 has 19 convolutional layers with 143 millions of parameters. The architecture of VGGNet has been shown in Table 2.8.

Table 2.8 VGGNet Architecture

No	Convolution	Output Dimension	Pooling	Output Dimension
Layer 1 & 2	convolution layer of 64 channel of 3X3 kernel with padding 1, stride 1	224X224X64	Max pool stride=2,Size 2X2	112X112X64
Layer 3 & 4	Convolution layer of 128 channel of 3X3 kernel	112X112X128	Max pool stride=2,Size 2X2	56X56X128
Layer 5,6,7	Convolution layer of 256 channel of 3X3 kernel	56X56X256	Max pool stride=2,Size 2X2	28X28X256
Layer 8,9,10	Convolution layer of 512 channel of 3X3 kernel	28X28X512	Max pool stride=2,Size 2X2	14X14X512
Layer 11,12,13	Convolution layer of 512 channel of 3X3 kernel	14X14X512	Max pool stride=2,Size 2X2	7X7X512

- **ResNet:** It is proposed by Kaimin et al. in 2015. When building deep neural networks, the team of Microsoft is the first to introduce skipping connection while not compromising on quality.

  The skipping actually enables skipping one or more layers. It is certainly seen innovative to design such a deep network with up to 152 layers without any quality compromise. The key idea behind ResNet is the use of residual blocks, which allow information to bypass several layers. The advantage of residual block is that it avoids the vanishing gradient problem [38]. The detail architecture of ResNet is shown in Table 2.9.

Table 2.9 ResNet50 Architecture

Layer Type	Feature Maps	Size	Kernel Size	Stride	Activation
Convolutional	64	7X7	3X3	2	ReLU
Max Pooling	64	3X3	-	2	-
Residual Block	64	1X1	1X1	1	ReLU
Residual Block	64	3X3	3X3	1	ReLU
Residual Block	64	1X1	1X1	1	ReLU
Residual Block	128	3X3	3X3	2	ReLU
Residual Block	128	1X1	1X1	1	ReLU
Residual Block	128	3X3	3X3	1	ReLU
Residual Block	128	1X1	1X1	1	ReLU
Residual Block	256	3X3	3X3	2	ReLU
Residual Block	256	1X1	1X1	1	ReLU
Residual Block	256	3X3	3X3	1	ReLU

2.7 TIME COMPLEXITY IN CNN

A CNN 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. tried to find how fully connected layers and the dataset are related through a number of experiments with different types of datasets [124]. The datasets can be divided into deeper datasets and wider datasets. A deeper dataset has more images per class in the training set then other dataset. 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 [125]. 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.

Matheus Gutoski et al. studied how data augmentation affects CNN performance. The author has implemented data augmentation on both balanced and unbalanced datasets on two different CNN models to see its effect on the performance of CNN [126]. They have concluded that smaller augmentation does not have a great impact on the performance of CNN for both types of datasets. A complex network needs higher augmentation in order to improve CNN model performance for both types of datasets.

The effect of filter size on image classification has been shown by Owais Mujtaba et.al. [127]. A CNN model has been developed which differs by only filter size and is implemented on two datasets, namely CIFAR10 and Fashion MNIST. Through experiments, the authors concluded that the loss increases as the filter size increases and accuracy decreases as the filer size increases. The problem associated with small filter size is the computational cost that is very important when dealing with large datasets.

For image classification, how the batch size and learning rate affect the generalizability of CNN has been studied on a histopathology dataset by Ibrahem Kandel et al. [128], A comparison has been made between different CNNs, varying batch size and learning rates on the Patch-Camelyon dataset. They have concluded that learning rate and batch size have a correlation and have a notable effect on the performance of CNN [128].

Higher learning rates with large batch sizes perform better as compared to lower learning rates. On the other hand for fine tuning, a low learning rate and a smaller batch size perform better. For a real-life scenario, to determine the optimal batch size, the authors recommended starting with a smaller batch size like 16 or 32.

Ercan Avşar has discovered the effect of pre-processing of an image on CNN performance for pneumonia detection. Two networks namely MobileNetV2 and EfficientNetB0 are used to determine the effect of preprocessing techniques. [129]. The dataset contains images of chest X-rays of patients with pneumonia disease. The highest classification accuracy was achieved by implementing Wiener Filtering on MobileNetV2.

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 [130]. 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 [131]. 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.

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 [132]. 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.

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 [133]. 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.

Yee Liang Thian et al. have shown how training data volume can affect the CNN model's performance in the radiology domain. Two datasets named ChestX-ray14 and CheXpert have been used for the experiments. The CNN model has been trained by increasing the training dataset size to see its effect. They concluded that an imbalanced dataset can significantly affect the performance of CNN [134]. From the learning curves, they found that the performance of the model increases as the training size increases.

Zhao DD et al. have proposed a space-efficient quantization scheme for deep CNN and introduced a model compression approach. The method takes eight or fewer bits to represent the 32-bit weights [135]. The method reduces the storage space requirement of a deep convolution neural network. The method could achieve 14X compression with the same accuracy as the model. The proposed method could be one of the solutions to CNN's space storage problem. One more effort has been made by Yu Cheng et al., by replacing the linear projection with the circular projection in fully connected layers, which results in redundancy of parameters of CNN [136]. With the proposed method, they could achieve space complexity from O (d2) to O (d). The experiments conducted on three datasets, namely MNIST, CIFAR-10 and ImageNet, show the proposed method can significantly reduce the storage requirement of the model and minimize the error rate.

An experiment was done by Saad Naeem et al. to reduce the large computational requirements of CNN to make them run on mobile devices. Initially, the authors survey various techniques that can be combined to reduce the training time of neural networks [136]. They study deep compression techniques that reduce the time required for training the network. They have worked on different parameters and concluded that there is no direct way to reduce space and time complexity and increase the accuracy of CNN.

Tanya Makkar et al. have studied the time complexity of KNN and CNN for the recognition of handwritten digits on the MNIST dataset [137]. It has been shown that CNN performs with high accuracy with a loss of 1.9% as compared to KNN with a loss of 3.8%. An investigation into the accuracy of CNN constraints on time and cost has been done by Kaiming He et al. The authors took depth, width, filter size and stride of the architecture into consideration [156]. They initially replace some layers with others and see their effect and modify the architecture of the model accordingly. They have concluded that if the depth of the model is increased, then the width and filter size should be reduced to increase model performance.

Nu Wen et al. proposed block-sparse CNN architecture that converts a dense convolution kernel to sparse. The proposed network solves the problem of overfitting, which occurs in traditional CNN when the number of parameters increases [138]. They have proved that block-sparse CNN reduces the space and time complexity and improves performance.

The purpose of this study is to find the the factors that affect the model's performance, the time each layer takes to run, how it affects the model's overall performance and factors that directly effect time complexity of the model. From the literature review mentioned above, the following conclusions can be made:

Research Findings

- Deeper architecture works best with deeper datasets and shallow architecture with wider datasets.
- There is a relationship between the number of dense layers and the number of neurons with respect to the dataset.
- The number of convolutional and dense layers directly affects the runtime of the model.
- The lower filter size and higher batch size can increase the model's performance but increase the computational cost of the model [9].

- When the learning rate is low, a lower batch size gives a better result. When the number of layers is greater, keeping the lower learning rate gives a better result.

- The max-pooling layer reduces the parameter count, which decreases computational complexity.

- For sensitive results, models with a less number of layers are more successful.

- The accuracy of the model not only depends on the number of convolutional layers or the depth of the network but also on the number of convolutional filters in one layer and the size of the convolution kernel.

- A higher augmentation size does not seem to introduce overfitting.

Concluding Remarks:

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 CNN has improved the result accuracy of food image recognition. Different food image datasets have been studied and found 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. It has been found that different preprocessing techniques have been used for efficient noise removal, but most techniques have difficulty in removing salt and pepper noise while preserving edges and contours. Earlier studies have used CNN for different perspectives and given the depth of each layer of CNN but estimating the time taken by these layers is missing.

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. To start with the proposed work, the first step is to create a dataset. The next chapter will discuss the dataset.