IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received March 20, 2020, accepted April 5, 2020, date of publication May 6, 2020, date of current version May 15, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2991810

An Open-Ended Continual Learning for Food Recognition Using Class Incremental Extreme Learning Machines

GHALIB AHMED TAHIR AND CHU KIONG LOO, (Senior Member, IEEE)
Faculty of Computer Science and Information Technology, University of Malaya, Kuala Lumpur 50603, Malaysia

Corresponding author: Chu Kiong Loo (ckloo.um@um.edu.my)

This work was supported in part by the Grand Challenge Grant - HTM (Wellness) under Grant GC003A-14HTM, in part by the University of Malaya, IIRG under Grant under Grant IIRG002C-19HWB, in part by the University of Malaya, International Collaboration Fund for project Developmental Cognitive Robot with Continual Lifelong Learning under Grant IF0318M1006, in part by the MESTECC, Malaysia and ONRG Grant under Project ONRG-NICOP- N62909-18-1-2086, and in part by the Office of Naval and Research Global, U.K., under Grant IF017-2018.

ABSTRACT State-of-the-art deep learning models for food recognition do not allow data incremental learning and often suffer from catastrophic interference problems during the class incremental learning. This is an important issue in food recognition since real-world food datasets are open-ended and dynamic, involving a continuous increase in food samples and food classes. Model retraining is often carried out to cope with the dynamic nature of the data, but this demands high-end computational resources and significant time. This paper proposes a new open-ended continual learning framework by employing transfer learning on deep models for feature extraction, Relief F for feature selection, and a novel adaptive reduced class incremental kernel extreme learning machine (ARCIKELM) for classification. Transfer learning is beneficial due to the high generalization ability of deep learning features. Relief F reduces computational complexity by ranking and selecting the extracted features. The novel ARCIKELM classifier dynamically adjusts network architecture to reduce catastrophic forgetting. It addresses domain adaptation problems when new samples of the existing class arrive. To conduct comprehensive experiments, we evaluated the model against four standard food benchmarks and a recently collected Pakistani food dataset. Experimental results show that the proposed framework learns new classes incrementally with less catastrophic inference and adapts domain changes while having competitive classification performance.

INDEX TERMS Food recognition, deep learning, open-ended continual learning, class incremental extreme learning machine, adaptive class incremental extreme learning machine, adaptive reduced class incremental kernel extreme learning machine.

I. INTRODUCTION

In the open-ended continual learning, the new images of existing classes arrive continuously, and novel classes always appear. Two types of incremental learning are an important component in open-ended continual learning: 1) Data incremental learning 2) Class incremental learning. The data incremental learning improves the recognition performance of existing classes and adapt domain changes using newly available images. In contrast, class incremental learning continuously gains knowledge from novel classes. Similarly, like the two aspects of learning in humans, these two types of incremental learning are necessary for acquiring new concepts and improving classification performance of existing classes.

This demonstrates the importance of open-ended continual learning in many real-world recognition problems, including food recognition as the dataset is dynamic, and new concepts of interest occur over time. Concerning the food recognition problem, the users upload food images from a smartphone belonging to existing classes and new classes each day. For existing classes, the class definitions vary between users, thus it is important to assume that users have different concepts of the same class. Analysis of the food datasets used for

The associate editor coordinating the review of this manuscript and approving it for publication was Noor Zaman.

evaluating the proposed framework also exhibits that there are high inter-class similarity, and intra-class variations among images (See Section 5). For new tags, there is no pre-defined limit for the food dataset due to "large vocabulary". The recent rise in popularity of food on social media, TV shows, food blogs, etc. has led to the production of large amounts of new food tags also known as food porn [1]. All of this makes the food datasets open-ended and dynamic. Despite this, most of the conventional methods of food recognition assume fixed datasets and high diversity in the classes at the beginning. However, food recognition in real-world scenarios can be catered by the use of open-end continual learning, but there are challenges associated with open-ended learning, that need to be discussed.

The paper is organized as follows: In section 1.1, we described the problems of open-ended continual learning in context with food recognition. In section 1.2, we have discussed the main contribution of this study. Section 2 discusses state-of-the-art networks for food recognition. Section 3 presents the architecture of the framework in context with open-ended continual learning. It discusses the methodology to extract and select features and the novel method for classification. Section 4 presents the experimental results. The results demonstrate the superiority of the model as compared to existing approaches while satisfying the criteria of open-ended continual learning. Section 5 presents the discussion, which is followed by the conclusion in Section 6.

A. PROBLEM DEFINITION

Deep learning networks for food image classification have achieved state-of-the-art performance on various food datasets. Despite this, these models have assumed fixed datasets which have increased the gap between laboratory and real-world environment. Most of the image datasets in real life are open-ended and dynamic [2], [3]. However, a well-trained deep learning model tends to forget the previous information while learning new information — a concept known as catastrophic forgetting [4]. This means a model learned on fixed food datasets cannot be trained easily on additional samples of current food classes or new food classes without significantly affecting its previous performance. These two problems hinder the usage of deep learning models for open-ended continual learning.

To understand the catastrophic forgetting during incremental learning, two theories have been proposed. The first theory suggests that human plasticity decreases on neurons that have learned previous information. This decreased plasticity on neurons helps to retain previous information [5]. The second theory explains that human extracts high-level information and stores that information in different brain areas while retaining episodic memories [6]. By keeping in view these two theories, researchers have proposed techniques for deep learning methods to learn incrementally. They have used regularization parameters, images of previously learned classes, and support vectors. However, these methods have certain assumptions and trade-offs. At first, they cannot take the benefit of new samples of existing learned classes. Secondly, newly trained classes requires equal number of samples as initially prepared classes.

B. CONTRIBUTION OF STUDY

This paper aims to deal with the challenges faced in an actual environment. It has used transfer learning due to high generalization ability of pre-trained deep learning networks and determined the best network for extracting features from food images. To cater, for lack of datasets from various orientations, it uses online data augmentation during transfer learning. Experimental analysis has shown that online data augmentation has significantly enhanced the model ability to classify food images from different possible orientations. Secondly, this research work has used a Relief F method for ranking features, based on which it reduces the dimensions of extracted features. The redundant features increase the computational complexity of the model.

Finally, and importantly, to deal with the challenge of class incremental learning and domain adaptation, this paper has proposed a novel method using extreme learning machines. Among all the various neural based learning systems. This study has chosen extreme learning machines for classification due to their fast training and testing time while retaining competitive classification accuracy as compared to other learning methods [7], [8]. However, the experimental result shows that existing extreme learning machine methods are unable to continuously learn a large number of food classes. For this reason, this research work has proposed a novel adaptive reduced class incremental kernel extreme learning machine (ARCIKELM). It reduces the complexity of kernel matrix, has good generalization ability and can learn continuously new food classes with less catastrophic forgetting.

The contribution of this study are summarized as follows.

1) This work has determined a more accurate feature extraction method from ResNet-50, DenseNet-201, and Inception-Resnet-V2 for food recognition. It has evaluated the effect of online data augmentation during transfer learning for food recognition and determined that online data augmentation significantly improves model performance for various orientations. We have significantly reduced the feature dimension and training time by using the Relief F method.

2) This study has proposed a novel classifier for class incremental and data incremental learning by using extreme learning machine due to its fast training and testing time. It adjusts its architecture dynamically to reduce catastrophic forgetting when new food classes are added. For existing classes, it adapts to domain changes by learning sequentially and adding new neurons based on our proposed strategy.

3) A comprehensive experiment has been conducted to evaluate performance, catastrophic forgetting, and found that the proposed approach outperforms existing methods.

II. RELATED WORK

Most of the existing approaches for food recognition assume that the training dataset has all the classes and variations

within them. However, in a real-world setting, the limited number of classes and images of existing food classes are initially available. Therefore, the food recognition system must satisfy the criteria of open-ended continual learning. It should be able to learn new classes incrementally in real-time and does not need to store exemplary images dataset as this will increase the complexity of the model and data security concerns. The online learning should be considered to adapt variation in the existing class images. This section briefly discussed the present literature on food recognition in context with mentioned challenges. It includes a wide range of methods, from simple Euclidean distances to spatial pyramid convolution neural networks for food recognition. Finally, this section summarizes the research gaps of current approaches and the contribution of the proposed framework to existing literature.

A. DISTANCE BASED METHOD

Jackman *et al.* [9] used "color and wavelet texture features" and Mahalanobis distances for classification on the small "Pork and Ham" dataset. They achieved an accuracy of 100 percent. Ciocca *et al.* [10] extracted visual descriptors from patches and used a pre-trained k nearest neighbor for classification. They achieved an accuracy of 85% on the datasets gathered by them. Duan *et al.* [11] applied SIFT, Gabor BOF for extracting features from an image dataset and then the Euclidean distance is used for classification. Phetphoung *et al.* [12] used histograms of color in RGB, HSV color spaces and shape properties for extracting features. Finally, k nearest neighbor is used to classify sushi dataset with an accuracy of 93.7%. Kong *et al.* [13] developed dietcam and applied to sift feature extractor on 5 food classes and achieved an accuracy of 92% using the nearest neighbor classifier. All of these approaches use different types of feature extractor to improve performance for fixed datasets and are not employed for open-ended continual learning.

B. PROBABILISTIC MODEL AND DECISION TREE

There are several methods which have explored decision trees for food classifications. Bossard *et al.* [14], have used random forest for classification and have achieved an accuracy of 50.71% on the food 101 dataset. Wang *et al.* [15] developed a mobile application by using the recursive Bayesian model. Herranz *et al.* [16] focused on a probabilistic model for the classification of restaurant images and dishes dataset. All of the existing probabilistic model and decision tree approaches uses fixed dataset and does not take into account open-ended continual learning.

C. SUPPORT VECTOR MACHINE

Support vector machine for food classification has been studied extensively. There are many variants of support vectors with different feature extraction methods. Few of these methods have also been integrated with mobile applications. These methods do not, however, satisfy the criteria of open-ended continual learning proposed in this work. Puri *et al.* [17], Zhu *et al.* [18], Anthimopoulos *et al.* [19], Sasano *et al.* [20] and Joutou and Yanai [21] studied support vector and multiple vector kernel support vector using color and texture features. Luo *et al.* [22] explored nutrition and texture features on food dataset with a support vector machine for classification. Kawano and Yanai [23] developed a mobile application that uses a fisher vector to encode HoG patches and color patches. Almaghrabi *et al.* used support vector machine for classification on 80 classes and achieved an accuracy of 80%. Tammachat *et al.* [24] focused on Bag-of-Feature (BOF) of Segmentation based Fractal Texture Analysis (SFfA) and color histogram for feature extraction. They have used a support vector machine for classification on 40 different types of Thai food with an accuracy of 70%. Velvizhy *et al.* studied surf bag of words for feature extraction and support vector machine for classification of 11 food classes. Zheng *et al.* [25] concatenated codebook of color and D-Sift features and achieved an accuracy of 70.84% on the UECFOOD100 data set using a linear support vector machine. Bosch *et al.* [26] built an ensemble classifier by using global and local feature extraction methods. They have used a support vector machine with the voting-based method for late decision fusion classification. They achieved an accuracy of 86.31% on 39 food classes. Miyano *et al.* [27] exploited the bag of the feature using local and color features from images and support vector machine for classification. Nguyen *et al.* [28] experimented using SIFT and shape context descriptors on PFID dataset. They used a support vector machine for classification and have achieved an accuracy of 68%. Farinella *et al.* [29] used consensus vocabulary with a bag of visual words and support vector machines for classification on the PFID dataset. They achieved an accuracy of 79.20%. Yang *et al.* [30] classified pairwise local features of PFID dataset using support vector machine and have achieved an accuracy of 80% on the PFID dataset. Kusumoto *et al.* [31] used hybrid aggregation of sparse color and edge histogram features and support vector machine for classification. They have achieved an accuracy of 30% on the PFID dataset and 85% on 300 images of 10 classes of dataset gathered by them. Pouladzadeh *et al.* [32] extracted features by using the edge detection method and counting the pixel in the region of interest. Finally, they used a support vector machine for classification and achieved an accuracy of 92.21% on 20 food classes of dataset gathered by them. Matsuda and Yanai [33] have used Gabor texture features from the selected region, SIFT-BOF, and Color-SIFT with the spatial pyramid. For final classification, they explored multiple kernel support vector machines and achieved an accuracy of 65.2% on the Japanese food data set. Their work is important in the area of food recognition as authors studied the classification of multiple foods in an image. Ruz *et al.* [34] used color, hue and multiple types of 'sift' features for classifying restaurant food images using a support vector machine. Their work has also explored GPS position for identification of the restaurant where the photo was taken and automatically retrieving the available menu from the internet.

Kusumoto *et al.* [31] applied discriminant feature extraction based on sparse encoding on the RFID dataset and performed classification using support vector machine. Overall, existing SVM based methods and its variant for food recognition have mainly focused on feature engineering. There is no method which addresses the open-ended continual learning challenges.

D. CONVOLUTIONAL NEURAL NETWORK

Wang *et al.* [15] explored the partially asymmetric multi-task convolutional neural network. They experimented on 24,690 food images of 100 classes of restaurants and dishes. Heravi *et al.* [35] applied Spatial Pyramid Convolutional Neural Network which they have evaluated on their gathered dataset and achieved an accuracy of 94%. Kawano and Yanai [23] focused on developing mobile applications and implemented a deep convolution method for classification. They achieved an accuracy of 72.26% on UECFOOD100 dataset. Tanno *et al.* [36] worked on a convolutional neural network for classification of their food image dataset. Liu *et al.* [37] explored deep learning for food classification and achieved a competitive accuracy. Myers *et al.* [38], Zhang *et al.* [39], Subhi and Ali [40], Yanai and Kawano [41], Christodoulidis *et al.* [42] Wu *et al.* [43] focused on different variations of deep learning for classification of food data sets. Some of these studies generated their own food datasets while other studies used already available food benchmarks for evaluation of their proposed approaches. However, the convolutional network softmax function used for classification has fixed architecture, and cannot address the challenges of open-end learning.

E. EXTREME LEARNING MACHINE

Recent studies have explored extreme learning machine for food classification due to fast training time and good performance [44]. Supervised Extreme Learning Committee, uses multiple kernel extreme learning as committee members. It takes features of colors, shape, texture, and predict multiple results. The Structured SVM, which acts as a supervisor, gives the final output. Their method uses full kernel matrix and batch learning and does not address issues of open-ended continual learning.

An extensive literature review has established the fact that most of the existing methods for food recognition used fixed class datasets, cannot incrementally learn new classes, and unable to adapt to domain variations. The only method which partially addresses this issue is proposed by [45]. However, their framework has limitations as they have to extract features from all the previous data and retrain the model. The calculation cost and computational resources associated with retraining are high which makes their approach less favorable. The framework proposed in this study contributes to the existing literature by addressing these challenges. It takes into account the high generalization ability of deep features and replaced fixed class softmax classifier with the classifier based on extreme learning machine. It uses extreme learning machine due to its fast training time. However, existing methods of extreme learning machine cannot learn continuously. For this reason, we have proposed a novel classifier that can learn continuously and adapt domain changes. We have briefly discussed our approach in the following sections.

III. THE APPROACH

A. SYSTEM OVERVIEW

The proposed framework takes into account two attributes of open-ended learning 1) Class incremental learning 2) Data incremental learning. It consists of three modules A. Feature extraction module B. Feature selection and C. Classification. Fig. 1 show the flow chart of the proposed architecture. During a training period, each incoming image is given to the selected deep feature extractor module that extracts the features. They are then ranked using the Relief F method, and the best features are chosen based on our proposed strategy. The selected features from the Relief F method are considered as the final representation of the image. Although, the extracted features from the deep model have good generalization ability. The fixed class architecture softmax for final classification does not take advantage of this ability. This study uses novel ARCIKELM which learns from these representations while satisfying both criteria of open-ended continual learning. It adds new output and hidden neurons when image representations belong to the novel class. In case they are from existing classes, it updates the model sequentially by our proposed strategy and only adds new hidden neurons when required.

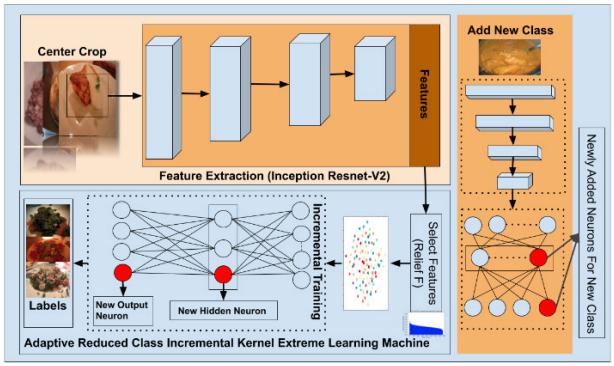

FIGURE 1. Flow chart of proposed framework. The red circle shows neurons which are incrementally added to the network.

During the classification stage, the features are extracted from the test image by using the same deep feature extractor. The best representations are chosen based on the ranking of the features from the Relief F method. Finally, ARCIKELM makes the final decision.

B. IMAGE FEATURE REPRESENTATION

Deep learning models for food recognition can learn the best image representations and eliminate the need for handcraft feature extraction process which is based on prior knowledge. Our proposed framework applies transfer learning with online

data augmentation on pre-trained image-net deep model which is then used to extract features. In transfer learning a model created for one task is used again as the base point for a second task. When modeling the second task, it enables for fast advancement and enhanced results. It is prominent in deep learning for feature extraction because the training of the model from scratch requires specialized computational resources such as GPU and long training time. This is the major hindrance for real-time incremental learning. However, various studies have shown a high generalization ability of deep features from pre-trained models on generic datasets [46] such as ImageNet. It is also beneficial for learning a novel classes. To determine the best architecture of a model for food feature extraction, we have explored three state-of-the-art deep learning networks, ResNet-50 [47], DenseNet-201 [48], and Inception ResNet-V2 [49]. The ImageNet weights initialize the models. They then are fine-tuned using food datasets and the resultant model is used for feature extraction. The experiment section has discussed the results.

In the following sub-section, we present types of online data-augmentation used during transfer learning.

1) DATA AUGMENTATION

Data augmentation generates transformed versions of pictures that are in the same class as the initial picture in the training. Transformations include a variety of image processing activities like zoom, horizontal shift, rotation, etc. The objective is to add fresh, credible instances to the training. This implies, creating different variants of training that the model is highly likely to detect. For example, the horizontal flip of food is a plausible scenario because the picture could be taken in either direction. However, the vertical flip of food would produce a useless result as a picture of food turned upside down does not make any sense. The study has applied 'shifting', flip, zoom range, Channel Shift Range and fill mode. They are explained in the following paragraph.

In shifting, image pixels are moved in the horizontal or vertical direction. The group of pixels which are deleted in one region is copied into another region of the image. This study uses both horizontal and vertical shifts. Flipping includes inverting the rows or columns of pixels. The vertical flip reverses the columns of pixels and the horizontal flip reverses the rows of the pixel. The experiments have investigated the horizontal flip as a vertically flipped image of food does not make any sense. In zoom augmentation, the picture is zoomed in or out, either by adding fresh pixel values around the picture or by interpolating pixel values respectively. It is important for food images as various users take images at different zoom levels. Channel shifting is the method of capturing the pixel's red, green or blue values in a picture and adding those values to pixels in distinct positions in the picture. The experiments have shifted channel by 30 degrees using online data augmentation. Finally, in the fill augmentation, fresh pixels are introduced in the picture that is not included in the initial picture boundaries. For example, after rotation black area in the corners would be added in the picture that was not included in the original picture. The Fill mode specifies how we manage these regions. This is handled by either using a constant values method or default method. The constant value is applied to all the new pixels and sub-pixels. In the method, all the new pixels are either given the black value or white. To fill the area with black, 0 is set as the default value and to fill the area with a white, default value is set as 1. Fig.2 illustrates the different modes of online data augmentation applied in this study on a randomly selected food image.

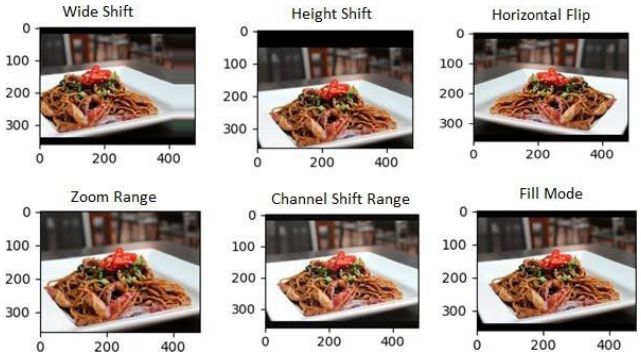

FIGURE 2. Different modes of online data augmentation used in this study on randomly selected image.

The results in the experimental section discuss in detail the impact of data augmentation. Based on the results in the experimental section, the pro-posed framework has used Inception-Resnet-V2 for feature extraction after applying transfer learning with online data augmentation. It is the first step in the food recognition process. The next section explains Feature selection.

C. IMAGE FEATURE SELECTION

The extracted features from the deep learning model have a very high dimension and there are only subsets of features that are relevant in determining the results. These low ranked features increase the computational complexity of classifiers. This study uses Relief F to rank features and selects the best features. It is the only algorithm for detecting feature dependencies that do not search through feature combinations but uses the concept of the nearest neighbor to detect feature dependencies [50]. This study has evaluated three configurations for selection of features and determined which configuration has higher accuracy and less training time for food. Finally, for classification purposes, it has used the best configuration. The experimental section presents the results of the feature selection.

D. CLASSIFICATION METHODS

The final step in our proposed model is incremental learning and recognition. Most of the existing studies use batch learning approaches for the classification of food, which requires fixed classes, making them unsuitable for real-time settings. This paper evolves the knowledge in the area of extreme

learning machines for continuous recognition of food dishes. For this purpose, this study has proposed a novel method that learns incrementally with less training time, higher accuracy, and better generalization ability. This section has discussed the basics of class incremental extreme learning machine, followed by our proposed method. The notations for symbols and acronym definition in this paper are summarized in Table 1 and Table 2.

TABLE 1. Notation definition.

Symbol	*Description*
$X_b; X_n$	Observed samples of based and new classes
$k_b; k_n$	Initial kernel and updated kernel matrix
$\beta^{(b)}\ \beta^{(n)}$	Initial output weights and updated output weights
M	Transformation matrix
N	Neuron eligibility matrix
μ_{ih}	Membership value

TABLE 2. Acronyms definition.

Symbol	*Description*
ELM	Extreme learning machine
KELM	Kernel extreme learning machine
CIELM	Class incremental extreme learning machine
ACIELM	Adaptive class incremental extreme learning machine
ARCIKELM	Adaptive reduced class incremental kernel extreme learning machine

1) CLASS INCREMENTAL ONLINE SEQUENTIAL EXTREME LEARNING MACHINE

CIOSELM addresses the problem of continuous learning in extreme learning machine [51]. At first, an ELM [52] base classifier is trained on initially available classes known as ELM_1. When the new class arrives, ELM_1 is adapted to a new classifier ELM_2 by using an incremental algorithm and labeled data of new and old classes. The structure of ELM changes and the network will add output neurons.

The mathematical derivation is as follows. At first, in correspondence with the ELM algorithm, after the training phase of a dataset of d is completed, the output of weight is calculated by Equation (1).

$$\beta_0 = k_0^{-1} H_0^T T_0 \tag{1}$$

where,

$$H_0 = \begin{bmatrix} G(a_1, b_1, x_d) & \cdots & G(a_1, b_1, x_1) \\ \vdots & \cdots & \vdots \\ G(a_1, b_1, x_n) & \cdots & G(a_n, b_n, x_n) \end{bmatrix}$$

$$and\ T_0 = \begin{bmatrix} t_1^T \\ \vdots \\ t_n^T \end{bmatrix} \ and\ K_0 = H_0^T H_0 \tag{2}$$

From new class dataset, H_1 is computed by use of the Equation (3)

$$\boldsymbol{H_1} = \begin{bmatrix} \boldsymbol{G\left(a_1, b_1, x_s^1\right)} & \cdots & \boldsymbol{G\left(a_1, b_1, x_s^1\right)} \\ \vdots & \cdots & \vdots \\ \boldsymbol{G\left(a_1, b_1, x_s^n\right)} & \cdots & \boldsymbol{G\left(a_n, b_n, x_s^n\right)} \end{bmatrix} \tag{3}$$

Now, the number of samples in T_1 is represented by m + 1-dimension Equation (4), where m is the number of columns in T_0

$$\mathbf{T_1} = \begin{bmatrix} \mathbf{0} & \cdots & \mathbf{0} & \mathbf{1} \\ \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{1} \end{bmatrix}_{\mathbf{N_1 * (m+1)}} \tag{4}$$

After combining d and s datasets, β_1 is calculated by use of Equation (5)

$$\boldsymbol{\beta_1} = \boldsymbol{K_1^{-1}} \begin{bmatrix} \boldsymbol{H_0} \\ \boldsymbol{H_1} \end{bmatrix}^{\boldsymbol{T}} \begin{bmatrix} \boldsymbol{T_0 \cdot M} \\ \boldsymbol{T_1} \end{bmatrix} \tag{5}$$

where M is a transformation matrix and is denoted by Equation (6).

$$\boldsymbol{M} = \begin{bmatrix} \mathbf{1} & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \ddots & \vdots & \mathbf{0} \\ \mathbf{0} & \cdots & \mathbf{1} & \mathbf{0} \end{bmatrix} \tag{6}$$

It can be seen that β_1 is calculated through incremental learning without providing previous dataset.

The method has enabled extreme learning machines to incrementally learn new classes. However, it has numerous drawbacks [51]. The significant drawback is a fixed number of hidden neurons. This increases the plasticity of neurons and results in catastrophic forgetting. When new classes arrive, there are more neurons required to handle new incoming data. However, a fixed number of neurons in it results in under fitting. Similarly, when a large number of hidden neurons are given initially, it results in overfitting of data. Adaptive CIELM [53] has addressed this problem. However, extensive experimentation carried out by this paper show that their method is not stable. We suspect that this is because of the poor generalization ability, due to random input weights.

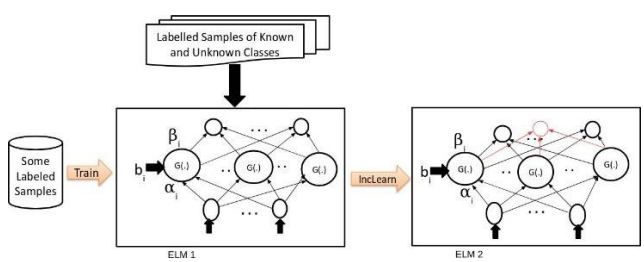

FIGURE 3. Class incremental extreme learning machine.

2) ADAPTIVE REDUCED CLASS INCREMENTAL KERNEL EXTREME LEARNING MACHINE

By keeping in view the existing problems, this study has proposed a novel method that is based on reduced online

sequential kernel extreme learning machine [54]. It can learn incrementally new classes and can adjust hidden neurons by itself, which makes classifier suitable for continual food recognition. This experimental section consists of a comprehensive evaluation. Experimental results have established the superiority of the method over existing approaches of the extreme learning machine. This section explains the mathematical working of the proposed approach.

a: INITIALIZATION PHASE

At first, the data from base classes is used to train the batch RKELM algorithm. With the 10 percent B mapping samples randomly selected from the data of base classes, it calculates the kernel matrix by use of equation (7).

$$\boldsymbol{k_b} = \boldsymbol{K}\left(\boldsymbol{x_b}, \boldsymbol{x_b}\right) \tag{7}$$

and then the initial output weight on training classes is computed by Equation (8).

$$\boldsymbol{\beta}^{(b)} = \boldsymbol{G_b}\boldsymbol{K_b^T}\boldsymbol{T_b} \tag{8}$$

where,

$$\boldsymbol{G_b} = \left(\boldsymbol{I}/\boldsymbol{C} + \boldsymbol{K_b^T}\boldsymbol{K_b}\right)^{-1}$$
$$\boldsymbol{and\ T_b} = [\boldsymbol{t_1}, \ldots\ldots\ldots \boldsymbol{t_n}] \tag{9}$$

The algorithm also initiates the dynamic matrix 'Neuron Eligibility Matrix' which is denoted by N. It contains the last time instance at which the neuron is activated.

b: ADDING NEW CLASS

Suppose, another new class data arrives, $\mathbb{N}_n = \{(x_i, t_i)\}_{i=N_0+1}^{N_0+N_n}$ It randomly selects the 10 percent of data from new class and updates the existing output weights by multiplying it with transformation matrix. As a result, total number of output neurons in the network is increased by 1. The output neurons in the previous beta is updated by using Equation (10).

$$\boldsymbol{\beta}^{(n)} = \boldsymbol{\beta}^{(b)} \cdot \boldsymbol{M} \tag{10}$$

where, M is the transformation matrix,

$$M = \begin{bmatrix} 1 & \cdots & 0 & 0 \\ \vdots & \ddots & \vdots & 0 \\ 0 & \cdots & 1 & 0 \end{bmatrix} \tag{11}$$

After that, hidden neurons for newly added class will be added to the network by use of 'Growing hidden Neuron' methodology which is explained in section below. The number of hidden nodes which is added to the network cannot be more than the incoming data. Finally, the 'Neuron Eligibility Matrix' is updated and its size should become equivalent to the total number of neurons in the kernel matrix.

c: SEQUENTIAL LEARNING

In case, when the data of existing classes in the network arrives $N_e = \{(x_i, t_i)\}_{i=N_0+1}^{N_e+1}$. At first, the multidimensional Gaussian function is used to measure the similarity by using equation (12), due to its simplicity and wide usage in a range of applications.

$$\mu_{ih} = exp\left(-\frac{|x_i - c_h|^2}{2\sigma^2}\right); \quad i = 1, \ldots\ldots, N; \quad h = 1, \ldots\ldots, L \tag{12}$$

where μ_{ih} is the membership value, $0 \leq \mu_{ih} \leq 1$. In it σ is the spread of radius and c_h is the hth vector of kernel matrix. The membership function is used to measure the similarity between sample x_i and the hth vector of kernel matrix. When the incoming vector is close to the neurons in the kernel matrix, its membership degree is high. On the other hand, when it is away from existing neurons, then its membership value will be low. Based on the membership value it will analyze whether the network needs to add new neuron or update the existing network. The brief description of these two processes is described below. After this, if there are any redundant neurons, they are removed from the network and the 'Neuron Activation Matrix' will be updated.

i) ANALYZE IF NEW HIDDEN NODE REQUIRED

To check weather new hidden neuron is required during sequential learning. It determines the vector h0 in kernel matrix which has maximum fuzzy member ship value with input vector by use of the Equation (13) and Equation (14).

$$\mu^{h_0}\left(x\left(k\right)\right) = \max_{1 \leq h \leq H} \left|\mu^h\left(x\left(k\right)\right)\right| \tag{13}$$

where x(k) represents the data at kth sequence.

$$h_0 = \arg \max_{1 \leq h \leq H} \left|\mu^h\left(x\left(k\right)\right)\right| \tag{14}$$

When the maximum fuzzy membership μ^{h_0} is equal to zero or less than predefined threshold, then the existing kernel matrix is not enough. As a result, kernel matrix is updated, and the new neuron is added. The new hidden node will be added to the network by using growing neuron methodology. In case the fuzzy membership is greater than zero or predefined threshold, it will not add new neuron to kernel matrix and updates the network and 'Neuron Activation Matrix'.

ii) UPDATING OF NETWORK

When no new hidden neurons are added, the network needs to be updated. The output weight β(n) then becomes

$$\beta^{(n)} = Z_1^{-1}\begin{bmatrix} K_e \\ K_n \end{bmatrix}^T \begin{bmatrix} T_e \\ T_n \end{bmatrix} \tag{15}$$

where,

$$Z_n = \frac{I}{C} + \begin{bmatrix} K_e \\ K_n \end{bmatrix}^T \begin{bmatrix} K_e \\ K_n \end{bmatrix}, \quad K_e = K\left(X_n, X_L\right) \tag{16}$$

For sequential learning of data chunk $\boldsymbol{\beta^n}$ is expressed as a function of $\boldsymbol{\beta^e}, \boldsymbol{z_n}, \boldsymbol{k_n}$ ***and*** $\boldsymbol{T_n}$. Now $\boldsymbol{Z_n}$ can be written as,

$$Z_n = \frac{I}{C} + \left[K_e^T K_n^T\right]\begin{bmatrix} K_e \\ K_n \end{bmatrix} = Z_e + K_n^T K_n \tag{17}$$

and

$$\begin{aligned}\begin{bmatrix} K_e \\ K_n \end{bmatrix}^T \begin{bmatrix} T_e \\ T_n \end{bmatrix} &= K_e^T T_e + K_n^T T_n \\ &= Z_e Z_e^{-1} K_e^T K_e + K_n^T T_n \\ &= Z_e \beta^{(e)} + K_n^T T_n \\ &= (Z_n - K_n^T K_n)\beta^{(e)} + K_n^T T_n \\ &= Z_n \beta^{(e)} - K_n^T K_n \beta^{(e)} + K_n^T T_n \end{aligned} \tag{18}$$

After, substituting $\boldsymbol{\beta^{(n)}}$ is denoted by,

$$\begin{aligned}\beta^{(n)} &= Z_n^{-1}(Z_n \beta^{(e)} - K_n^T K_n \beta^{(e)} + K_n^T T_n) \\ &= \beta^{(e)} + Z_n^{-1} K_n^T (T_n - K_n \beta^{(e)})\end{aligned} \tag{19}$$

Similarly, when the (K + 1) chunk of data arrive $N_{k+1} = \{(x_i, t_i)\}_{i=\left(\sum_{j=0}^{k} N_j\right)+1}^{\sum_{j=0}^{k+1} N_j}$ In it $k \geq 0$ and N_{k+1} represents total observations in the (k+1)-th chunk, assuming this we have.

$$Z_{k+1} = Z_k + K_{k+1}^T K_{k+1} \tag{20}$$

$$\beta^{k+1} = \beta^{(k)} + Z_{k+1}^{-1} K_{k+1}^T \left(T_{k+1} - K_{k+1}\beta^{(k)}\right) \tag{21}$$

where,

$$K_{k+1} = K(X_{k+1}, X_L), X_{k+1} = \{X_i\}_{(\sum_{j=0}^{k} N_j+1)}^{\sum_{j=0}^{k+1} N_j} \textit{ and}$$

$$T_{k+1} = \begin{bmatrix} t^T_{\sum_{j=0}^{k} N_j+1} \\ \vdots \\ t^T_{\sum_{j=0}^{k+1} N_j+1} \end{bmatrix} \tag{22}$$

Z_{k+1}^{-1} relative of Z_{k+1} will be used for computing β^{k+1}. It is then derived by use of woodburry equation as follows.

$$\begin{aligned}Z_{k+1}^{-1} &= (Z_k + K_{k+1}^T K_{k+1})^{-1} \\ &= Z_k^{-1} - Z_k^{-1} K_{k+1}^T (+K_{k+1} Z_k^{-1} K_{k+1}^T)^{-1} K_{k+1} Z_k^{-1}\end{aligned} \tag{23}$$

Let $\boldsymbol{G_{k+1} = K_{k+1}^{-1}}$ then $\boldsymbol{\beta^{k+1}}$ can be computed as follows:

$$G_{k+1} = G_k - G_k K_{k+1}^T (I + K_{k+1} G_k K_{k+1}^T)^{-1} K_{k+1} G_k \tag{24}$$

$$\beta^{k+1} = \beta^{(k)} + G_{k+1} K_{k+1}^T \left(T_{k+1} - K_{k+1}\beta^{(k)}\right) \tag{25}$$

iii) ANALYZE IF EXISTING HIDDEN NODES SHOULD BE REMOVED

In this step, algorithm determines the elements in 'Neuron Activation Matrix' which has minimum value. This will determine the neuron which are not activated in the near past for last k new incoming vectors. It will check based on class imbalance ratio weather these neurons belong to minority class or majority class. If these neurons belong to majority class, the algorithm considers these neurons as redundant neurons and deletes it from the network as described in the following section. The algorithm prevents the neurons of minority class to be deleted as it can result in complete loss of information related to that class.

iv) DELETION OF REDUNDANT NEURONS

When the algorithm has decided to delete the redundant neuron $\boldsymbol{h_r}$ from the network. The corresponding row in the 'Neuron Activation Matrix' $\boldsymbol{N^{h_r}}$, Output Weights $\boldsymbol{\beta^{h_r}}$ will be deleted. Table 3 has described the methodology to delete neurons from G matrix

TABLE 3. Pruning of G matrix.

Input
G matrix, p: index of neuron to be pruned
Output
Pruned G Matrix
Steps
1. Remove the p'th element from G Matrix and place it in front
2. N = size of G Matrix,
3. G = G(2:N, 2:N)
4. F= G(1:N, 1)
5. e = G(1,1)
6. $G = G - f * f'/e$

v) UPDATING OF NEURON ACTIVATION MATRIX

If the algorithm has not added any neuron to the network, the 'Neuron Activation Matrix' is updated by using Equation (26).

$$\boldsymbol{N^{k_0} = k} \tag{26}$$

k, represents the last incoming data vector and $\boldsymbol{h_0}$ is the neuron which has maximum membership value and is calculated by Equation (14).

d: GROWING HIDDEN NEURONS

The incremental growth of hidden neurons is proposed by [53]. This study has used the kernel matrix instead of using random weights. The experiment demonstrated that the kernel matrix for growing hidden neurons can increase network stability and generalization ability. According to it, when additional hidden neurons are added then it has to minimize.

$$\begin{bmatrix} K_e & Z_e \\ K_n & Z_n \end{bmatrix} \beta - \begin{bmatrix} T_e \\ T_n \end{bmatrix} \tag{27}$$

where,

$$\boldsymbol{K_n = K(X_n, X_L) \textit{ and } Z_n = K(X_n, X_n)} \tag{28}$$

$\boldsymbol{Z_e}$ is the zero block matrix and $\boldsymbol{Z_n}$ is the output matrix of newly added hidden nodes. Thus, we have:

$$\beta^{(n)} = G_n^{-1} \begin{bmatrix} K_e & 0 \\ K_n & Z_n \end{bmatrix}^T \begin{bmatrix} T_e \\ T_n \end{bmatrix} \tag{29}$$

And

$$\begin{aligned}G_n &= \begin{bmatrix} K_e & 0 \\ K_n & Z_n \end{bmatrix}^T \begin{bmatrix} K_e & 0 \\ K_n & Z_n \end{bmatrix} \\ &= \begin{bmatrix} G_n & K_n^T Z_n \\ Z_n^T K_n & Z_n^T Z_n \end{bmatrix}\end{aligned} \tag{30}$$

where $G_e = K_e^T K_e = P_e^{-1}$, $\beta^{(e)} = (K_e^T K_e)^{-1} K_e^T T_e$ and $G_n = G_e + K_n^T K_n$

By reformulating Equation (27) to:

$$\begin{bmatrix} K_e & 0 \\ K_n & Z_n \end{bmatrix}^T \begin{bmatrix} T_e \\ T_n \end{bmatrix} = \begin{bmatrix} K_e^T T_e + K_n^T T_n \\ Z_n^T T_n \end{bmatrix} = \begin{bmatrix} G_n \beta^{(e)} - K_n^T K_n \beta^{(e)} + K_n^T T_n \\ Z_n^T \end{bmatrix} = \begin{bmatrix} G_1 \\ G_2 \end{bmatrix} \tag{31}$$

After, applying block inversion matrix the equation becomes

$$P_n = \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix} = G_n^{-1} = \begin{bmatrix} G_n & K_n^T Z_n \\ Z_n^T K_n & Z_n^T Z_n \end{bmatrix}^{-1} \tag{32}$$

The equation can be revised to:

$$\beta^{(n)} = \begin{bmatrix} U \\ D \end{bmatrix} = \begin{bmatrix} P_{11} G_1 + P_{12} G_2 \\ P_{21} G_1 + P_{22} G_2 \end{bmatrix} \tag{33}$$

The schulur component of G_n is $S = Z_n^T Z_n - Z_n^T K_n P_n K_n^T Z_n$ and the matrix $P_n = G_n^{-1}$ is invertible when S is invertible.

Based on the above derivation, the incremental growth of hidden neurons is calculated by Equation (34) and Equation (35).

$$\begin{aligned} P_{11} &= P_n + P_n K_n^T Z_n S^{-1} Z_n^T K_n P_n \\ P_{12} &= -P_n K_n^T Z_n S^{-1} \\ P_{21} &= -S^{-1} Z_n^T K_n P_n \\ P_{22} &= S^{-1} \end{aligned} \tag{34}$$

$$\begin{aligned} U &= \beta^{(e)} + P_n K_n^T \left(I + Z_n S^{-1} Z_n^T \left(K_n P_n K_n^T - I \right) \right) \\ D &= -S^{-1} Z_n^T K_n \left(\beta^{(e)} + P_n K_n^T \left(T_n - K_n \beta^{(e)} \right) \right) \\ &\quad + S^{-1} Z_n^T T_n \end{aligned} \tag{35}$$

3) RELATION TO ACIELM

The ARCIKELM algorithm inherits core features from Adaptive CIELM. Like Adaptive CIELM, it can learn sequentially, add new neurons, handle new classes. However, important variations exist between them. Table 4 represents the proposed algorithm.

4) HANDLING OF CATASTROPHIC FORGETTING

The theory of human plasticity states that decreasing plasticity on neurons helps to retain prior knowledge [5]. By keeping this in view, the proposed classifier adapted continuously based on incoming data. It creates hidden neurons Equation (34), output neurons and adjusts existing neurons to handle novel classes and images Equation (24) Equation (25). The formation of new neurons decreases plasticity for old neurons and reduces the catastrophic forgetting of previously learned information. During sequential learning, when novel images of existing class arrives it computes a fuzzy relationship with the existing neurons and adds new neuron when the fuzzy relationship is less than a specific threshold. This helps in retaining the previous concepts and learning new variations in existing classes. Experimental results in this study also showed that fixed hidden neurons in CIELM results in increased catastrophic forgetting during open-ended learning.

TABLE 4. Proposed adaptive RCIKELM algorithm.

Initialization phase
Initialize the model using small chunk $\mathbb{N}_b = \{\{x_i, t_i\}\}_{i=1}^{N_0}$ of training data from the given base classes data $\mathbb{N} = \{(x_i, t_i)\ x_i \in R^m, t_i \in R^l, i = 1,2,3,\ldots\ldots\}$
1. Randomly selects 10 percent of data of base classes X_b .
2. Compute the initial kernel matrix by using Equation (7).
3. Estimate the initial output weight by using Equation (8).
4. set k = 0
5. Initiates a dynamic matrix 'Neuron Eligibility Matrix' denoted by N.
Adding new class
6. Present the Initial data of new class
7. Randomly select 10 percent of data of new class X_n
8. Update existing β by use of Equation (10) to add new output neuron.
9. Add hidden neurons for new class data X_n using Equation (34) and Equation(35)
10. Update G and kernel Matrix K_n
11. Expand 'Neuron Eligibility Matrix' *N*
Sequential learning phase
12. Present the (K+1) th chunk of new observations. $N_{k+1} = \{(x_i, t_i)\}_{i=(\sum_{j=0}^{k} N_j)+1}^{\sum_{j=0}^{k+1} N_j}$ where N_{k+1} denotes the number of observation in k+1 chunk of data
13. Compute the fuzzy membership between input vector and kernel matrix by using Equation (12).
14. Compute maximum membership value by use of the Equation (14). If the maximum membership is less than threshold, add new hidden neuron by use of Equation (34) and Equation (35).
15. Otherwise compute the kernel matrix for the (k+1)-th chunk of data N_{k+1} and set T_{k+1} by use of Equation (22).
16. Update G and calculate the output weights β_{k+1} by use of Equation (24) and Equation (25).
17. Analyze redundant neurons in the network and remove the corresponding row in N^{h_r} and β^{h_r} . Update the G matrix by using the methodology described in Table 3.
18. set k = k+1
19. If new class data go to step 6 else go to step 12

IV. EXPERIMENTS AND RESULTS

In this section, this study discusses food datasets, implementation, along with the details analysis of each phase of the proposed framework on these datasets. The proposed framework has three modules. A) Feature extraction B) Feature Selection C) Classification. Based upon, that the experiments are categorized into three major parts. In the first experiment, the study determines the best feature extractor for food datasets from state-of-the-art deep learning networks and determine the impact of online data augmentation during transfer learning. The second experiment is to study the reduction in classification time by using Relief F for feature selection.

The third experiment investigates the performance of novel ARCIKELM classifier against existing extreme learning machines based approaches. In the first part of this

experiment, the paper evaluated classification measures, and in the second part, the study evaluated catastrophic forgetting measures. After that, the paper discusses the results of the stability analysis of the novel classifier as compared to ACIELM. Finally, we have compared the proposed framework with other approaches for food recognition.

A. DATASETS

To evaluate the proposed approach this research has used four existing food databases that are widely used by researchers. Besides that, this study has collected the Pakistani Food dataset. This section briefly describe important components of the datasets used for experimentation.

1) UECFOOD100

The dataset "UECFOOD100" contains 100 different sorts of food photographs. Each food photo has a bounding box, which indicates the location of the food item in the photo. Food categories in this dataset are mainly famous foods in Japan. Therefore, some categories might not be common to people from other countries than Japanese [55]. Fig. 4 displays inter-class similarity and intra-class variations of the UECFOOD100 dataset.

FIGURE 4. Row 1 displays inter-class similarity in UECFOOD100. Row 2 shows intra-class variations.

2) FOOD101

It contains 101,000 real-world images divided into 101 food classes. Fig. 5 show that there are visually and semantically similar food classes such as Apple pie, bread pudding, baklava, carrot cake Discriminative Components with Mussels, Onion rings, Paella, Edamame, Risotto, Bibimbap, Omelets, Lobster bisque, Eggs benedict, Macarons and much, etc. Besides domain adaptation, it poses a challenge for incremental learning due to a large number of food categories.

FIGURE 5. Row 1 displays inter-class similarity in Food101. (a) apple pie, (b) bread pudding, (c) baklava, (d) carrot cake. Row 2 shows intra-class dissimilarity of clan-chowder.

3) UECFOOD256

"UECFOOD256" has 256 classes of food photos. The food portion in the picture has bounded boxes. Most of the classes in the dataset are famous foods in Japan and other countries. Some food classes are only known to Japanese [56].

UECFOOD256 has a large number of categories that poses the challenge for the incremental learning algorithm. Fig. 6 show the domain adaption challenge in UECFOOD256 due to inter-class similarity and intra-class variations.

FIGURE 6. Row 1 displays inter-class similarity in UECFOOD256 and row 2 shows intra-class dissimilarity.

4) PFID

This dataset has 1388 food images from 15 different categories. It is gathered from various chains and includes pizza, salads, and burgers. The dataset has three instances and every instance in the restaurant setting has four still images with wrappers and without wrappers. Besides that, it includes six still images in a lab environment, stereo images, and the 360-degree video of food on turntable. The Collection is open source and available free for researchers. [57]. Fig. 7 show inter-class similarity and intra-class dissimilarity of PFID dataset.

FIGURE 7. Row 1 displays inter-class similarity in PFID and row 2 shows intra-class dissimilarity.

5) PAKISTANI FOOD

The data is composed of 100 food classes of Pakistani food and is collected for this research study. It was crawled from the Internet and was verified by the expert dietitian. It consists of 4928 images. Due to the complex variety of food dishes in Pakistan. This data set poses a challenge for food recognition methods in real-world settings.

Fig. 8 displays high inter-class similarity and intra-class variations of food classes, which poses a challenge to existing classifiers.

FIGURE 8. Row 1 displays inter-class similarity in Pakistani Food and row 2 shows intra-class dissimilarity.

TABLE 5. Attribute of datasets.

Dataset	*Total Classes*	*Total Instances*	*Train/Test Instances*
Food 101	101	101330	91050/10280
UECFOOD100	100	16184	12864/3320
UECFOOD256	256	31148	28128/3020
PFID	15	1388	1248/140
Pakistani Food	100	4928	4448/480

B. IMPLEMENTATION

To determine the best feature extractor, we carried out the experiments on Google Colab (12 GB GPU, 12 GB Ram) for all the datasets on three state-of-the-art deep learning networks. For measuring classification performance and catastrophic forgetting, the paper has also carried out the experiments by using Google Colab (12 GB). The final version of the proposed framework uses the python Django framework and the integrated framework is deployed on the Amazon ec2 cloud. It is accessed by smartphone devices through Web service.

C. IMAGE FEATURE REPRESENTATION

This experiment has determined the best feature extractor for food recognition from three state-of-the-art networks: ResNet-50, DenseNet-201 and Inception-ResNet-V2. Table 6 shows the comparison of classification accuracy after applying transfer learning with online data augmentation. Inception-ResNet-V2 has better performance as compared to the remaining networks for all the dataset. This is because it combines the benefits of ResNet and Inception model. For this reason, this study has selected Inception-Resnet-V2 as a feature extractor in our proposed framework. Moreover, empirical evidence shows that training with residual connection accelerates the training [49].

TABLE 6. Comparison of classification accuracy on deep learning models.

Dataset	*Classes*	*ResNet-50*	*DenseNet-201*	*Inception Renet-V2*
UECFOOD100	100	80.28%	81.12 %	81.54%
UECFOOD256	256	66.84%	69.23 %	69.293%
FOOD101	101	80.48%	80.63 %	83.73%
PFID	15	100%	100%	100%
Pakistani Food	98	63.13%	69.38%	70.42%

1) IMPACT OF ONLINE DATA AUGMENTATION

This section has presented the results about the effect of online data augmentation during transfer learning for food datasets. Table 7 shows the experimental settings of online data augmentation for this research study. Table 8 exhibits the classification accuracy on the non-augmented dataset. In all of the cases, the augmented model has better performance as compared to the non-augmented model. The maximum improvement of 2.71% is observed for the Pakistani Food dataset. The difference in classification accuracy is not very large for augmented and non-augmented datasets. However, the test has only non-augmented images. In the real world scenario when the user provides images from different angles, the augmented model supposed to perform much better as compared to the non-augmented model. To evaluate this hypothesis, we have evaluated the test data from different orientations on model fine-tuned with augmented and non-augmented images. Fig. 9 exhibits that a model fine-tuned with augmented datasets has significant resilience. There is a significant reduction in classification accuracy for a model fine-tuned with a non-augmented dataset.

TABLE 7. Experimental settings of data augmentation.

Augmentation Type	*Value*
Wide Shift	0.2
Height Shift	0.2
Horizontal Flip	Randomly flip inputs horizontally
Zoom range	0.8 to 1
Channel Shift Range	30
Full mode	Reflect

TABLE 8. Classification accuracy on model trained with non augmented dataset.

UECFOOD100	*UECFOOD256*	*FOOD101*	*PFID*	*Pakistani Food*
79.98%	69.32%	81.81%	100%	67.71%

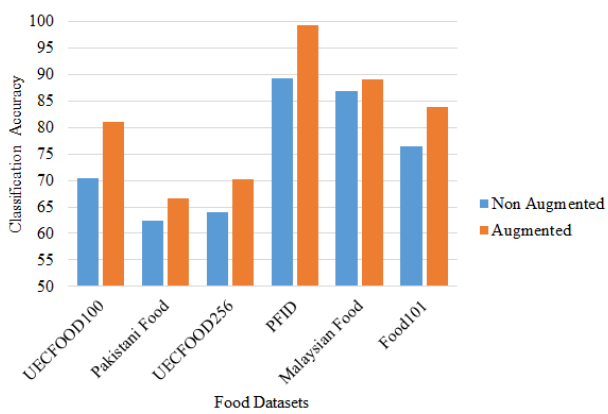

FIGURE 9. Comparison of classification performance on model fine-tuned with augmented dataset and non-augmented dataset.

TABLE 9. Comparison of average training time and classification accuracy on different subsets of features models.

Datasets	Training Time (All Features)	Accuracy (All Features)	Training Time (Subset B)	Accuracy (Subset B)	Training Time (Subset A)	Accuracy (Subset A)
Food101	1.12e+03	87.72	882.31	87.76	754.40	87.84
UECFOOD100	616.40	89.53	132.79	89.25	7.95	89.63
UECFOOD256	626.93	77.91	121.47	78.25	21.64	77.01
PFID	0.8147	100	0.79	100	0.10	100
Pakistani Food	48.36	75.93	2.3809	75.56	0.54	75.97

D. FEATURE SELECTION

To choose a reasonable number of features the analysis was performed using the Relief F method. This study has selected two subsets of features which have the highest score. The subset A consists of top 500 features, and subset B consists of top 1000 features. By using the proposed approach, this work has evaluated learning time and classification performance for each subset of the feature. Fig. 10 demonstrates that not all features significantly contribute to classification. Table 9 show the comparison of training time and classification accuracy. The experimental result shows that subset A has the best accuracy and training time is significantly reduced as compared to the other two combinations. Based on the feature selection results, this research work has used subset A for classification. Fig. 11 show the significant reduction in training time by appropriate feature selection.

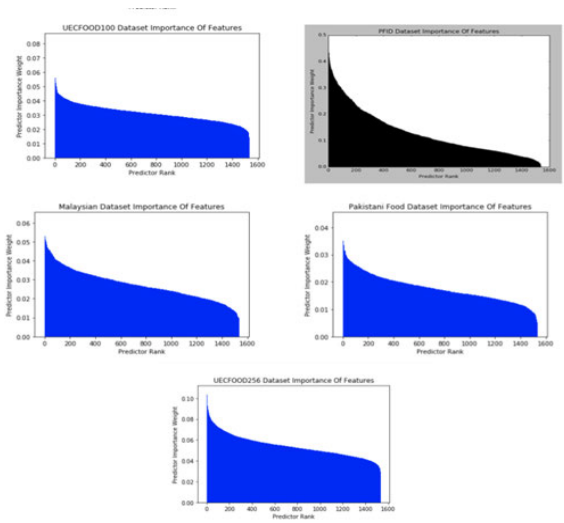

FIGURE 10. Weights of features using Relief-F method exhibits that many features do not contribute significantly towards classification.

E. CLASSIFICATION

This section discusses the experimental results of the proposed ARCIKELM classifier compared to the existing extreme learning machine based classifier. In the section 4.5.1, the paper evaluated the measure of classification performance, and the section 4.5.2 presented the results of catastrophic forgetting measures during incremental learning.

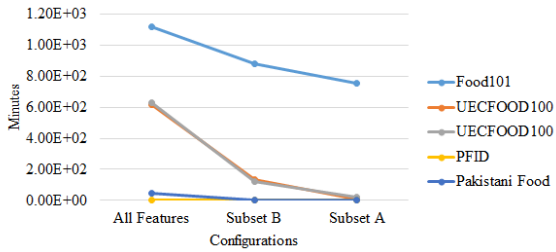

FIGURE 11. Comparison of training time for different configurations.

1) CLASSIFICATION PERFORMANCE

The research work uses the following metrics to measure the costs associated with predictions made by the model.

a: ACCURACY

Accuracy is the ability of classifier to accurately predict the food class. It is denoted by following equation

$$accuracy = \frac{(TP + FN)}{(TP + FP + FN + TN)} \times 100 \tag{36}$$

b: F1 SCORE

It is a weighted average of precision and recall. It takes into account both false positives and false negatives predicted by the model. The following is the mathematical equation to compute F1 Score.

$$F1Score = \frac{(2 * (Precision * Recall))}{Precision + Recall} \tag{37}$$

c: RECALL SCORE

It is the ability of a model to correctly classify among all actual food classes.

$$I_k = accuracy_k^* - accuracy_{k,k} \tag{38}$$

d: PRECISION SCORE

It is the ability of a model to correctly classify positive values.

$$Precision\ Score = \frac{TP}{(TP + FP)} \tag{39}$$

Table 12 and Table 13 show the comparison of classification accuracy, F1 score, Recall Score, Precision Score for all the datasets. The experimental results show that ARCIKELM has better performance as compared to CIELM and ACIELM. The average classification accuracy difference of the proposed model as compared to other incremental models for UECFOOD256, Pakistani Food, Food 101, and

UECFOOD100 is 7.03%, 5.6%, 2.67%, and 1% respectively. The PFID dataset has only 15 classes and has the same classification accuracy for ARCIKELM and CIELM. The proposed method even performs better than the batch learning ELM method for UECFOOD256 and is at par for PFID datasets. On all the other measures F1 accuracy, Precision, and Recall, ARCIELM performs better as compared to existing extreme learning machine based class incremental classifiers.

The Adaptive CIELM has the worst performance as compared to all the other methods and is highly unstable. Although the ACIELM does not have fixed architecture, it is suspected that this unstable nature is due to random weights. Moreover, F1 and Precision measures across the datasets except PFID for several sessions are undefined, as there are no false negatives and true positives. The true negatives are irrelevant for the calculation of the F1 measure. In the experiments, if the F1 and precision values in the session are undefined, we have replaced it with 1 and denoted all these computations by $*$ with the value.

2) CATASTROPHIC FORGETTING

The Kemker *et al.* [58] and Chaudry *et al.* [59] proposed catastrophic forgetting measures for incremental learning. The proposed network is trained by using the training data of the first session. After training, it will be evaluated by the same training data. The resultant accuracy was considered as the ideal performance of network which is denoted by $accuracy_{ideal}$.

The following training session trained the network with new food class and evaluated it by using testing data of all the previous sessions represented by $ccuracy_{k,j}$. The experiments were conducted for all the prior test's setup up to current session and labeled as $curacy_{k,all}$.

a: INTRANSIGENCE

Intransigence was measured relative to a standard classification model which had access to all the datasets at all times. The reference classification model was tested with the testing data of the kth session. Intransigence for the kth session was then calculated as:

$$I_k = accuracy_k^* - accuracy_{k,k} \tag{40}$$

As intransigence was defined as the difference between the accuracy of an incremental-learned network and a reference model, negative intransigence (i.e. I $<$ 0) implies that incremental learning up to session k positively impacts the model's knowledge about it.

b: FORGETTING

The forgetting factor of a training session is the difference between the maximum knowledge in the previous training sessions and the current session. [59]. Equation (41) quantifies forgetting of the jth session after incrementally training the network up to session k:

$$f_j^k = \max_{l \in \{1,\ldots,K-1\}} accuracy_{l,j} - accuracy_{k,j}, \quad j < k \tag{41}$$

The average forgetting at the kth training session is then written as:

$$F_k = \frac{1}{K-1} \sum_{j=1}^{k-1} f_j^k \tag{42}$$

c: BASE SESSION

It is the ability of the network to recall the knowledge from the first training session and is represented by equation.

$$\Omega_{base} = \frac{1}{K-1} \sum_{j=2}^{k} \frac{accuracy_{j,1}}{accuracy_{ideal}} \tag{43}$$

d: NEW SESSION

It is the ability of the network to recall newly learned knowledge and is calculated by following equation.

$$\Omega_{new} = \frac{1}{K-1} \sum_{j=2}^{k} accuracy_{j,j} \tag{44}$$

e: ALL SESSION

It is the ability of the network to retain the previous knowledge and learn new knowledge

$$\Omega_{all} = \frac{1}{K-1} \sum_{j=2}^{k} \frac{accuracy_{j,all}}{accuracy_{ideal}} \tag{45}$$

The Table 10 presents the configuration of the dataset for carrying out this experiment. The initial number of classes in the base session is 2. At first, all the algorithms (CIELM, ACIELM, and ARCIKELM) are trained on the base session. Algorithms learn the new classes in remaining sessions in an incremental manner. In ARCIKELM, the kernel parameter of all the experiments is set to 1, and the fuzzy membership threshold during sequential learning is set to 0.

For ACIELM, it inserts new hidden nodes when the hidden error becomes more than the threshold. The threshold of hidden error is set 0.8 for all the experiments. For CIELM the

TABLE 10. General settings of datasets.

Dataset	Base Classes	Total Sessions	Total Classes	Total Instances	Training/Testing
Food101	2	99	101	101330	91050/10280
UECFOOD100	2	98	100	16184	12864/3320
UECFOOD256	2	254	256	31148	28128/3020
PFID	2	13	15	1388	1248/140
Pakistani Food	2	98	100	4928	4448/480

TABLE 11. Average catastrophic measures of all datasets. Bold indicates the best measures.

	CIELM					ACIELM					ARCKIELM				
Dataset	Ω_{base}	Ω_{all}	Ω_{new}	I_k	F_k	Ω_{base}	Ω_{all}	Ω_{new}	I_k	F_k	Ω_{base}	Ω_{all}	Ω_{new}	I_k	F_k
Food101	0.9284	0.8790	0.8857	0.0360	0.0500	0.4677	0.5688	0.9897	0.4642	0.5861	**0.9426**	**0.9029**	**0.9059**	**0.0110**	**0.0397**
UECFOOD100	0.9575	0.9165	0.8484	0.0290	0.0670	0.3714	0.5450	0.9759	0.5240	0.6520	**0.9731**	**0.9195**	**0.8417**	**0.0180**	**0.0380**
UECFOOD256	0.4315	0.6793	0.7656	0.0646	0.1058	0.1006	0.4722	0.9482	0.5481	0.5399	**0.5116**	**0.8019**	**0.8602**	**-0.0048**	**0.1040**
PFID	1.0	1.0	1.0	0	0	0.9877	0.9948	1.0	0.0279	0.0359	**1.0**	**1.0**	**1.0**	**0**	**0**
Pakistani Food	0.8831	0.7326	0.7349	0.0681	0.0948	0.4026	0.4605	0.9482	0.4570	0.4570	**0.9769**	**0.8213**	**0.8167**	**0.0244**	**0.0924**

TABLE 12. Comparison of Average classification accuracy denoted by A, F1 score (Micro) denoted by F1, Precision score(Micro) denoted by P, Recall score(Micro) denoted by R best measures.

	ELM(Batch)				CIELM				ACIELM				ARCIKELM			
Dataset	A	F1	P	R	A	F1	P	R	A	F1	P	R	A	F1	P	R
Food101	**88.26**	88.26	88.26	88.26	84.60	84.60	84.60	84.60	41.84	41.84	41.84	41.84	87.27	87.16	87.16	87.16
UECFOOD100	**90.69**	90.69	90.69	90.69	87.74	87.74	87.74	87.74	38.28	38.28	38.28	38.28	88.74	88.64	88.64	88.64
UECFOOD256	75.93	75.94	75.94	75.94	69.48	69.48	69.48	69.48	39.90	39.90	39.90	39.90	**76.51**	76.42	76.42	76.42
PFID	100	100	100	100	100	100	100	100	97	97	97	97	**100**	100	100	100
Pakistani Food	**76.04**	76.05	76.05	76.05	69.24	69.24	69.24	69.24	26.42	26.42	26.42	26.42	74.84	74.84	74.84	74.84

TABLE 13. Comparison of Average classification accuracy denoted by A, F1 score (Macro) denoted by F1, Precision score (Macro) denoted by P, Recall score (Macro) denoted by R. If the F1 and precision values in any session are undefined, we have replaced it with 1 and denoted all these computations by * with the value.

	ELM(Batch)				CIELM				ACIELM				ARCIKELM			
Dataset	A	F1	P	R	A	F1	P	R	A	F1	P	R	A	F1	P	R
Food101	**88.26**	88.18	88.37	88.17	84.60	84.45	84.89	84.48	41.84	61.23	92.73*	41.64	87.27	87.04	87.27	87.06
UECFOOD100	**90.69**	84.81	86.98	84.04	87.74	81.90	83.73	80.27	38.28	88.08*	91.51*	34.13	88.74	82.11	84.91	81.22
UECFOOD256	75.93	76.62	79.30	75.65	69.48	69.83	73.29	68.96	39.90	69.76*	88.11*	39.83	**76.51**	75.14	78.31	76.16
PFID	100	100	100	100	100	100	100	100	97	98.38	99.17	96.67	**100**	100	100	100
Pakistani Food	**76.04**	76.06	78.33	76.07	69.24	70.31	71.96	68.85	26.42	60.29*	78.75*	26.58	74.84	74.10	77.40	74.65

hidden neurons remain constant throughout the experiment. Table 11 presents catastrophic measures for food datasets which is averaged across all sessions. The experimental results show the superiority of ARCIKELM for each of the datasets. In all of the cases, it reduces catastrophic forgetting as compared to ACIELM and CIELM. It is worth noting that the UECFOOD256 dataset has the most significant difference of 12.26% in the "All Session" accuracy as compared to the second-best model. It is due to the reason that this dataset has a large number of classes and this increases catastrophic forgetting for the networks with fixed hidden neurons. Figure 12 and Figure 13 shows the graphs of catastrophic measures for each session of UECFOOD256 dataset and Pakistani Food dataset. For all the sessions ARCIKELM has less catastrophic forgetting as compared to CIELM and ACIELM.

3) STABILITY ANALYSIS

The experiment was designed to analyze the stability for increasing hidden nodes in the Adaptive class incremental extreme learning machine and the proposed Adaptive reduced class incremental kernel extreme learning machine. This experiment has given full access to all the classes for minimizing the impact of catastrophic forgetting during incremental learning. It has measured the stability of the growth of hidden neurons in sequential settings. In ACIELM new hidden neurons are added based on hidden error. Fig. 14 and Fig. 15 exhibits experimental results for analyzing stability of ACIELM and ARCIKELM.

The experimental results show the highly unstable nature of the Adaptive class incremental extreme learning machine. However, ARCIKELM is stable when new hidden nodes are added incrementally. To further investigate the instability of ACIELM, the paper has reevaluated the results of the original theory of adaptive ELM proposed by [53]. This study has assumed the original configuration for re- evaluation. The initial number of neurons is 300. It learns incrementally one by one and adds single hidden neurons when the hidden error exceeds a threshold during training. We observed similar instability with the same experimental settings settings and

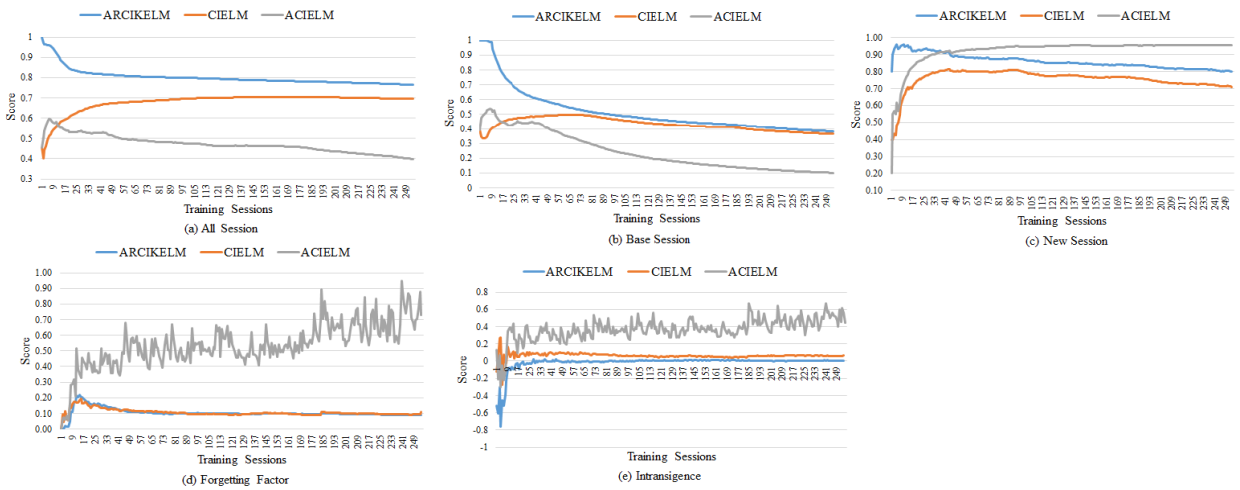

FIGURE 12. Comparison of catastrophic forgetting measures of each session of UECFOOD 256 dataset.

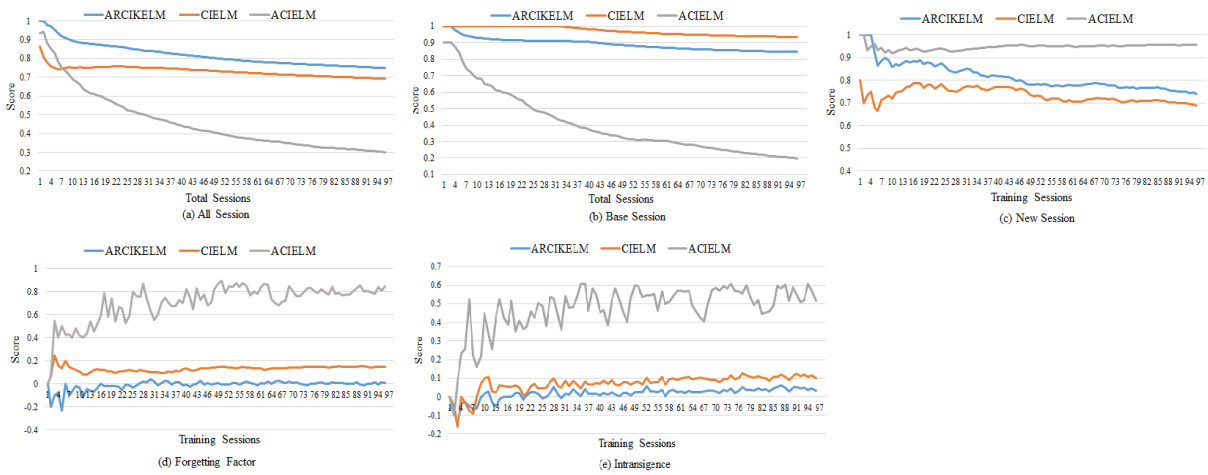

FIGURE 13. Comparison of catastrophic forgetting measures of each session of Pakistani Food dataset.

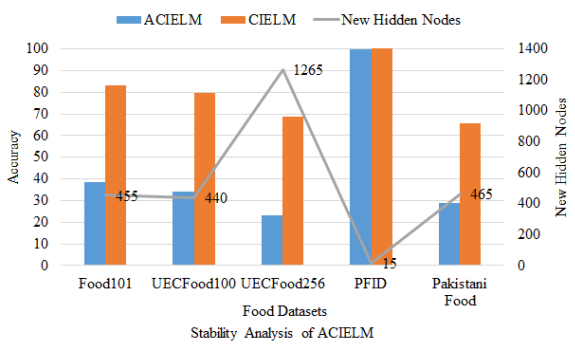

FIGURE 14. Stability Analysis of Adaptive CIELM. The grey line in the graph exhibits the incrementally added neurons when full access to all classes are given.

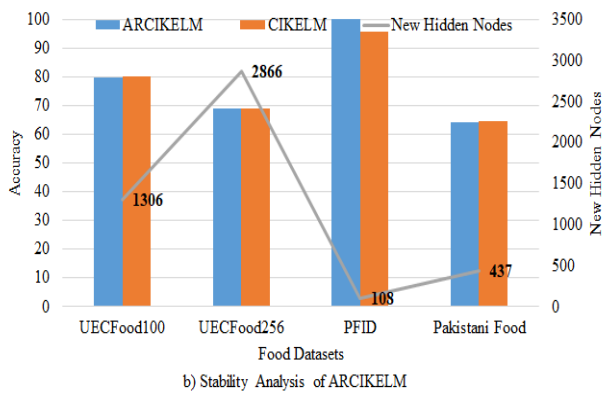

FIGURE 15. Stability Analysis of ARCIEKLM. The grey line in the graph exhibits the incrementally added hidden neurons when full access to all classes are given.

increasing hidden nodes at different hidden error threshold. The revaluation results are shown in Fig. 16.

The instability in adaptive class incremental extreme learning machine can be due to a lack of generalization ability and random input weights. The kernel matrix with good generalization ability has resolved this problem in ARCIKELM.

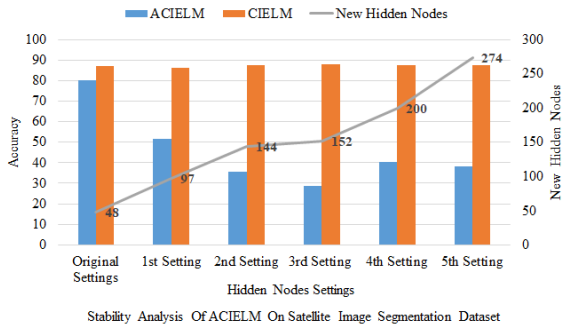

FIGURE 16. Stability Analysis by using configurations of previous study [53]. The grey line in the graph exhibits the incrementally added hidden neurons.

F. COMPARISON WITH OTHER FRAMEWORKS

Fig. 17 have compared the performance of proposed approach with the state-of-the-art methods. Result shows that the algorithm improves performance for PFID dataset against recent approaches and past approaches. The best accuracy for PFID was reported by SELC which has been improved by 9.1%. The proposed approach has less training and testing time as it takes only 10 percent of initially available class data as compared to SELC which uses all data for computing kernel matrix [44]. The most important part is that proposed method adds new classes Fig. 18 in an incremental fashion shows the comparison of Food101 against the state of the art. The method significantly improves accuracy while satisfying criteria of open-ended continual learning. Fig. 19

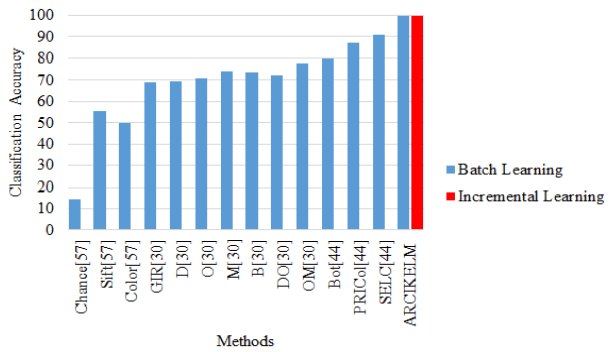

FIGURE 17. Comparison with other networks for PFID dataset. The red bar shows the average accuracy for incrementally trained network while blue bar shows final accuracy. The network with no red bar does not have ability to learn incrementally.

The method significantly improves accuracy while satisfying criteria of open-ended continual learning. Fig. 19 show the comparison of the proposed approach against other methods for UECFOOD100 dataset. The proposed framework has slightly less final classification accuracy as compared to SELC which uses full kernel matrix and multiple feature channels. However, the average accuracy is better and proposed model learned all classes in incremental manner.

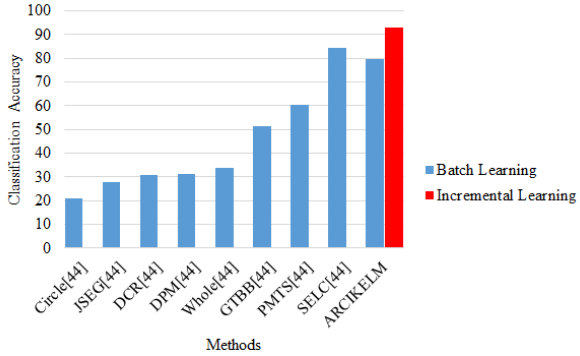

FIGURE 18. Comparison with other networks for UECFOOD100 dataset.

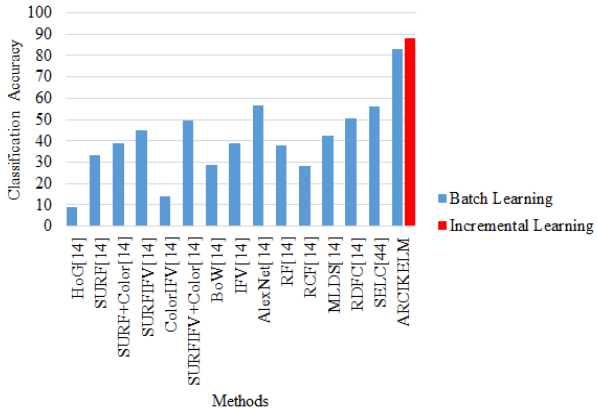

FIGURE 19. Comparison with other networks for Food101 dataset comparison with other networks for UECFOOD100 dataset.

V. DISCUSSION

Based on the extensive experiments carried out on four standard food benchmarks and newly gathered Pakistani food dataset for this study, we can state the following considerations. The modern deep learning networks have given good discriminatory features. In the experiments, Inception-Resnet-V2 has better performance as compared to others. It has combined the ability of ResNet and Inception network to outperform others. This work has considered transfer learning instead of learning from scratch and has not explored those models which require retraining. The training of the model from scratch was not in line with this study's research objective. It requires large training time and specialized computational resources such as GPU. All of this made these feature extractors unsuitable for real-time settings. However, transfer learning requires less training time, gives good discriminatory features and by combining it with incremental classifiers, the objective of open-ended continual learning was achieved.

In the second step, the proposed framework has selected important features. The features extracted from deep learning model has a very large dimension and were unsuitable for real-time settings. For this purpose, the Relief F method is applied to rank the features, and based on the ranking

best features are selected. This study has experimented with various configurations and selected the best configuration of 500 features. The experimental results in this paper have proved that the Relief F method has reduced the accumulative learning time of the proposed classifier for all the datasets by 52.14%.

Finally, the last step is class incremental and data incremental learning and are major challenges in open-ended continual learning. This research has evolved an extreme learning machine. The proposed method grows its hidden and output neurons dynamically and incrementally learns new classes. During sequential learning, it adapts to domain changes and reduces catastrophic forgetting by decreasing plasticity on previous neurons. The paper has investigated catastrophic forgetting for food recognition by using five measures: Base Session, All Session, New Session, Forgetting Factor, and Intransigence. The experimental results showed reduced catastrophic forgetting of the classifier as compared to CIELM, ACIELM. It is worthy to note that for UECFOOD-256, there is a maximum reduction in catastrophic forgetting for all measures as compared to ACIELM and CIELM. This exhibits that as the number of classes, the proposed model performs better. The four classification measures: Accuracy, F1 measure, Precision, and Recall exhibit that classification performance is better than ACIELM, CIELM for all datasets and is better with the batch learning model for UECFOOD256 and PFID datasets. The comparison with other frameworks SELC, PMTS, GTB etc. also showed that the proposed approach has competitive classification performance besides satisfying the criteria of open-end continual learning. Finally, the experiment section analyzed the stability of the model when new neurons are added. The results showed significant improvement in the stability of the method as compared to its predecessor ACIELM.

VI. CONCLUSION

The Food recognition dataset is open-ended and dynamic. There is a continuous increase in food samples and classes. Existing deep learning models for food recognition assumes that all the food classes and variations within the food classes exist initially. They suffer from catastrophic forgetting during class-incremental learning. This research study addresses these challenges by proposing a new framework of open-ended continual learning for food recognition.

It uses state-of-the-art deep learning networks to extract features, Relief F for feature ranking and selection, and ARCIKELM for classification. For feature extraction, this paper takes into account the excellent generalization ability of deep model features. It has evaluated three state-of-the-art deep networks and founded that Inception-Resnet-V2 has superior performance as compared to others. However, features extracted from deep learning models have a very high dimension and increases classification time. The framework has used the Relief F method to determine the optimal length and found that the Relief F method has reduced the accumulative learning time of the proposed classifier for all the datasets by 52.14%. For addressing the challenge of data incremental learning and class incremental learning, the framework used the novel Adaptive reduced class incremental kernel extreme learning machine. It dynamically increases hidden neurons and output neurons. The decreased plasticity of previous neurons reduces catastrophic forgetting. Experimental results on five catastrophic forgetting measures and four classification performance measures demonstrated that the proposed classifier has superior performance as compared to existing ACIELM and CIELM and is at par with batch classifier. The comparison of the proposed framework with other architectures for food recognition like supervised extreme learning committee, PMTS, GTBB etc. show competitive performance while satisfying the criteria of open-ended continual learning.

VII. FUTURE WORK

In future work, we aim to reduce catastrophic forgetting by using a hybrid scheme for open-ended continual learning. The online clustering method like self-organizing incremental neural network can be used to select the mapping nodes which best represent the class for ARCIKELM, and help to select the nearest nodes during classification. This can reduce catastrophic forgetting and is noise invariant when the input for classification is far away from existing neurons. The other direction is the auto-scaling of computational resources in a cloud environment. As the novel classes and new images arrive, the computational resources required increases. Similarly, during classification, the user requests vary at different time intervals. This requires that the open- ended continual learning system must be able to auto-scale its computational resources in a cloud environment. Finally, future studies will further enhance the proposed framework towards explainable AI. It will be able to explain its classification prediction.

REFERENCES

[1] S. Rousseau, *Food 'Porn' in Media*. Dordrecht, The Netherlands: Springer, Nov. 2014, doi: 10.1007/978-94-007-0929-4_395.

[2] C. Chen, W. Min, X. Li, and S. Jiang, "Hybrid incremental learning of new data and new classes for hand-held object recognition," *J. Vis. Commun. Image Represent.*, vol. 58, pp. 138–148, Jan. 2019, doi: 10.1016/j.jvcir.2018.11.009.

[3] T. Mensink, J. Verbeek, F. Perronnin, and G. Csurka, "Distance-based image classification: Generalizing to new classes at near-zero cost," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 11, pp. 2624–2637, Nov. 2013, doi: 10.1109/TPAMI.2013.83.

[4] D. Maltoni and V. Lomonaco, "Continuous learning in single-incremental-task scenarios," *CoRR*, vol. abs/1806.08568, pp. 56–73, Apr. 2018.

[5] G. Ditzler, M. Roveri, C. Alippi, and R. Polikar, "Learning in nonstationary environments: A survey," *IEEE Comput. Intell. Mag.*, vol. 10, no. 4, pp. 12–25, Nov. 2015, doi: 10.1109/MCI.2015.2471196.

[6] J. D. Power and B. L. Schlaggar, "Neural plasticity across the lifespan," *Wiley Interdiscipl. Rev., Develop. Biol.*, vol. 6, no. 1, p. e216, Jan. 2017, doi: 10.1002/wdev.216.

[7] G.-B. Huang, L. Chen, and C.-K. Siew, "Universal approximation using incremental constructive feedforward networks with random hidden nodes," *IEEE Trans. Neural Netw.*, vol. 17, no. 4, pp. 879–892, Jul. 2006, doi: 10.1109/TNN.2006.875977.

[8] X. Liu, C. Gao, and P. Li, "A comparative analysis of support vector machines and extreme learning machines," *Neural Netw.*, vol. 33, no. 9, pp. 58–66, Sep. 2012, doi: 10.1016/j.neunet.2012.04.002.

[9] P. Jackman, D.-W. Sun, P. Allen, N. A. Valous, F. Mendoza, and P. Ward, "Identification of important image features for pork and turkey ham classification using colour and wavelet texture features and genetic selection," *Meat Sci.*, vol. 84, no. 4, pp. 711–717, Apr. 2010, doi: 10.1016/j.meatsci.2009.10.030.

[10] G. Ciocca, P. Napoletano, and R. Schettini, "Food recognition and leftover estimation for daily diet monitoring," in *New Trends in Image Analysis and Processing—ICIAP*, V. Murino, E. Puppo, D. Sona, M. Cristani, and C. Sansone, Eds. Cham, Switzerland: Springer, 2015, pp. 334–341.

[11] P. Duan, W. Wang, W. Zhang, F. Gong, P. Zhang, and Y. Rao, "Food image recognition using pervasive cloud computing," in *Proc. IEEE Int. Conf. Green Comput. Commun. IEEE Internet Things IEEE Cyber, Phys. Social Comput.*, Aug. 2013, pp. 1631–1637, doi: 10.1109/GreenCom-iThings-CPSCom.2013.296.

[12] W. Phetphoung, N. Kittimeteeworakul, and R. Waranusast, "Automatic sushi classification from images using color histograms and shape properties," in *Proc. 3rd ICT Int. Student Project Conf. (ICT-ISPC)*, Mar. 2014, pp. 83–86, doi: 10.1109/ICTISPC.2014.692322.

[13] F. Kong and J. Tan, "DietCam: Automatic dietary assessment with mobile camera phones," *Pervas. Mob. Comput.*, vol. 8, no. 1, pp. 147–163, Feb. 2012, doi: 10.1109/ICT-ISPC.2014.6923223.

[14] L. Bossard, M. Guillaumin, and L. Van Gool, "Food-101—Mining discriminative components with random forests," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 446–461.

[15] H. Wang, W. Min, X. Li, and S. Jiang, "Where and what to eat: Simultaneous restaurant and dish recognition from food image," in *Advances in Multimedia Information Processing—PCM*, E. Chen, Y. Gong, and Y. Tie, Eds. Cham, Switzerland: Springer, 2016, pp. 520–528.

[16] L. Herranz, R. Xu, and S. Jiang, "A probabilistic model for food image recognition in restaurants," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jun./Jul. 2015, pp. 1–6, doi: 10.1109/ICME.2015.7177464.

[17] M. Puri, Z. Zhu, Q. Yu, A. Divakaran, and H. Sawhney, "Recognition and volume estimation of food intake using a mobile device," in *Proc. Workshop Appl. Comput. Vis. (WACV)*, Dec. 2009, pp. 1–8, doi: 10.1109/WACV.2009.5403087.

[18] F. Zhu, M. Bosch, T. Schap, N. Khanna, D. Ebert, C. Boushey, and E. Delp, "Segmentation assisted food classification for dietary assessment," *Proc. SPIE*, vol. 7873, Jan. 2011, Art. no. 78730, doi: 10.1117/12.877036.

[19] M. M. Anthimopoulos, L. Gianola, L. Scarnato, P. Diem, and S. G. Mougiakakou, "A food recognition system for diabetic patients based on an optimized bag-of-features model," *IEEE J. Biomed. Health Informat.*, vol. 18, no. 4, pp. 1261–1271, Jul. 2014, doi: 10.1109/JBHI.2014.2308928.

[20] S. Sasano, X.-H. Han, and Y.-W. Chen, "Food recognition by combined bags of color features and texture features," in *Proc. 9th Int. Congr. Image Signal Process., BioMed. Eng. Inform. (CISP-BMEI)*, Oct. 2016, pp. 815–819, doi: 10.1109/CISP-BMEI.2016.7852822.

[21] T. Joutou and K. Yanai, "A food image recognition system with multiple kernel learning," in *Proc. 16th IEEE Int. Conf. Image Process. (ICIP)*, vols. 285–288, Nov. 2009, pp. 285–288, doi: 10.1109/ICIP.2009.5413400.

[22] Y. Luo, C. Ling, and S. Ao, "Mobile-based food classification for type-2 diabetes using nutrient and textual features," in *Proc. Int. Conf. Data Sci. Adv. Anal. (DSAA)*, Oct./Nov. 2014, pp. 563–569, doi: 10.1109/DSAA.2014.7058127.

[23] Y. Kawano and K. Yanai, "FoodCam: A real-time mobile food recognition system employing Fisher vector," in *MultiMedia Modeling*, C. Gurrin, F. Hopfgartner, W. Hurst, H. Johansen, H. Lee, N. O'Connor, Eds. Cham, Switzerland: Springer, 2014, pp. 369–373.

[24] N. Tammachat and N. Pantuwong, "Calories analysis of food intake using image recognition," in *Proc. 6th Int. Conf. Inf. Technol. Electr. Eng. (ICITEE)*, Oct. 2014, pp. 1–4, doi: 10.1109/ICITEED.2014.7007901.

[25] J. Zheng, Z. Wang, and X. Ji, "Superpixel-based image recognition for food images," in *Proc. IEEE Can. Conf. Elect. Comput. Eng. (CCECE)*, May 2016, pp. 1–4, doi: 10.1109/CCECE.2016.7726860.

[26] M. Bosch, F. Zhu, N. Khanna, C. J. Boushey, and E. J. Delp, "Combining global and local features for food identification in dietary assessment," in *Proc. 18th IEEE Int. Conf. Image Process.*, Sep. 2011, pp. 1789–1792, doi: 10.1109/ICIP.2011.6115809.

[27] R. Miyano, Y. Uematsu, and H. Saito, "Food region detection using bag-of-features representation and color feature," in *Proc. VISAPP*, 2012, pp. 709–713.

[28] D. T. Nguyen, Z. Zong, P. O. Ogunbona, Y. Probst, and W. Li, "Food image classification using local appearance and global structural information," *Neurocomputing*, vol. 140, pp. 242–251, Sep. 2014, doi: 10.1016/j.neucom.2014.03.017.

[29] G. M. Farinella, D. Allegra, M. Moltisanti, F. Stanco, and S. Battiato, "Retrieval and classification of food images," *Comput. Biol. Med.*, vol. 77, pp. 23–39, Oct. 2016, doi: 10.1016/j.compbiomed.2016.07.006.

[30] S. Yang, M. Chen, D. Pomerleau, and R. Sukthankar, "Food recognition using statistics of pairwise local features," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 2249–2256, doi: 10.1109/CVPR.2010.5539907.

[31] R. Kusumoto, X. Han, and Y.-W. Chen, "Hybrid aggregation of sparse coded descriptors for food recognition," in *Proc. 22nd Int. Conf. Pattern Recognit.*, Aug. 2014, pp. 1490–1495, doi: 10.1109/ICPR.2014.265.

[32] P. Pouladzadeh, S. Shirmohammadi, and T. Arici, "Intelligent SVM based food intake measurement system," in *Proc. IEEE Int. Conf. Comput. Intell. Virtual Environ. Meas. Syst. Appl. (CIVEMSA)*, Jul. 2013, pp. 87–92, doi: 10.1109/CIVEMSA.2013.6617401.

[33] Y. Matsuda and K. Yanai, "Multiple-food recognition considering co-occurrence employing manifold ranking," in *Proc. 21st Int. Conf. Pattern Recognit. (ICPR)*, Nov. 2012, pp. 2017–2020.

[34] C. Ruz, C. Pieringer, B. Peralta, I. Lillo, P. Espinace, R. Gonzalez, B. Wendt, D. Mery, and A. Soto, "Visual recognition to access and analyze people density and flow patterns in indoor environments," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, Jan. 2015, pp. 1–8, doi: 10.1109/wacv.2015.8.

[35] E. J. Heravi, D. Puig, and H. H. Aghdam, "Classification of foods using spatial pyramid convolutional neural network," in *Frontiers in Artificial Intelligence and Applications*. Amsterdam, The Netherlands: IOS Press, 2016.

[36] R. Tanno, K. Okamoto, and K. Yanai, "DeepFoodCam: A DCNN-based real-time mobile food recognition system," in *Proc. 2nd Int. Workshop Multimedia Assist. Dietary Manage.*, Oct. 2016, p. 89, doi: 10.1145/2986035.2986044.

[37] C. Liu, Y. Cao, Y. Luo, G. Chen, V. Vokkarane, M. Yunsheng, S. Chen, and P. Hou, "A new deep learning-based food recognition system for dietary assessment on an edge computing service infrastructure," *IEEE Trans. Services Comput.*, vol. 11, no. 2, pp. 249–261, Mar. 2018, doi: 10.1109/TSC.2017.2662008.

[38] A. Myers, N. Johnston, V. Rathod, A. Korattikara, A. Gorban, N. Silberman, S. Guadarrama, G. Papandreou, J. Huang, and K. Murphy, "Im2Calories: Towards an automated mobile vision food diary," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1233–1241, doi: 10.1109/ICCV.2015.146.

[39] W. Zhang, D. Zhao, W. Gong, Z. Li, Q. Lu, and S. Yang, "Food image recognition with convolutional neural networks," in *Proc. IEEE 12th Int. Conf. Ubiquitous Intell. Comput. IEEE 12th Int. Conf. Autonomic Trusted Comput. IEEE 15th Int. Conf. Scalable Comput. Commun. Associated Workshops (UIC-ATC-ScalCom)*, Aug. 2015, pp. 690–693, doi: 10.1109/UIC-ATC-ScalCom-CBDCom-IoP.2015.139.

[40] M. A. Subhi and S. Md. Ali, "A deep convolutional neural network for food detection and recognition," in *Proc. IEEE-EMBS Conf. Biomed. Eng. Sci. (IECBES)*, Dec. 2018, pp. 284–287, doi: 10.1109/IECBES.2018.8626720.

[41] K. Yanai and Y. Kawano, "Food image recognition using deep convolutional network with pre-training and fine-tuning," in *Proc. IEEE Int. Conf. Multimedia Expo Workshops (ICMEW)*, Jun. 2015, pp. 1–6, doi: 10.1109/ICMEW.2015.7169816.

[42] S. Christodoulidis, M. Anthimopoulos, and S. Mougiakakou, "Food recognition for dietary assessment using deep convolutional neural networks," in *New Trends in Image Analysis and Processing—ICIAP*, V. Murino, E. Puppo, D. Sona, M. Cristani, and C. Sansone, Eds. Cham, Switzerland: Springer, 2015, pp. 458–465.

[43] H. Wu, M. Merler, R. Uceda-Sosa, and J. R. Smith, "Learning to make better mistakes: Semantics-aware visual food recognition," in *Proc. ACM Multimedia Conf. (MM)*, Oct. 2016, pp. 172–176, doi: 10.1145/2964284.2967205.

[44] N. Martinel, C. Piciarelli, and C. Micheloni, "A supervised extreme learning committee for food recognition," *Comput. Vis. Image Understand.*, vol. 148, pp. 67–86, Jul. 2016.

[45] S. Horiguchi, S. Amano, M. Ogawa, and K. Aizawa, "Personalized classifier for food image recognition," *IEEE Trans. Multimedia*, vol. 20, no. 10, pp. 2836–2848, Oct. 2018, doi: 10.1109/TMM.2018.2814339.

[46] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, "DeCAF: A deep convolutional activation feature for generic visual recognition," *CoRR*, vol. abs/1310.1531, pp. 647–655, Jun. 2013.

[47] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *CoRR*, vol. abs/1512.03385, pp. 770–778, Jun. 2015.
[48] G. Huang, Z. Liu, and K. Q. Weinberger, "Densely connected convolutional networks," *CoRR*, vol. abs/1608.06993, pp. 2261–2269, Jul. 2016.
[49] C. Szegedy, S. Ioffe, and V. Vanhoucke, "Inception-v4, inception-ResNet and the impact of residual connections on learning," *CoRR*, vol. abs/1602.07261, pp. 4278–4284, Feb. 2016.
[50] R. J. Urbanowicz, M. Meeker, W. G. L. Cava, R. S. Olson, and J. H. Moore, "Relief-based feature selection: Introduction and review," *CoRR*, vol. abs/1711.08421, pp. 189–203, Jul. 2017.
[51] Z. Zhao, Z. Chen, Y. Chen, S. Wang, and H. Wang, "A class incremental extreme learning machine for activity recognition," *Cognit. Comput.*, vol. 6, no. 3, pp. 423–431, Sep. 2014, doi: 10.1007/s12559-014-9259-y.
[52] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: Theory and applications," *Neurocomputing*, vol. 70, nos. 1–3, pp. 489–501, Dec. 2006, doi: 10.1016/j.neucom.2005.12.126.
[53] Y. Lan, Y. C. Soh, and G.-B. Huang, "A constructive enhancement for online sequential extreme learning machine," in *Proc. Int. Joint Conf. Neural Netw.*, Jun. 2009, pp. 1708–1713, doi: 10.1109/IJCNN.2009.5178608.
[54] W.-Y. Deng, Y.-S. Ong, P. S. Tan, and Q.-H. Zheng, "Online sequential reduced kernel extreme learning machine," *Neurocomputing*, vol. 174, pp. 72–84, Jan. 2016, doi: 10.1016/j.neucom.2015.06.087.
[55] H. Hassannejad, G. Matrella, P. Ciampolini, I. De Munari, M. Mordonini, and S. Cagnoni, "Food image recognition using very deep convolutional networks," in *Proc. 2nd Int. Workshop Multimedia Assist. Dietary Manage. (MADiMa)*. New York, NY, USA: ACM, 2016, pp. 41–49, doi: 10.1145/2986035.2986042.
[56] Y. Kawano and K. Yanai, "Automatic expansion of a food image dataset leveraging existing categories with domain adaptation," in *Proc. ECCV Workshop Transferring Adapting Source Knowl. Comput. Vis. (TASK-CV)*, 2014, pp. 3–17.
[57] M. Chen, K. Dhingra, W. Wu, L. Yang, R. Sukthankar, and J. Yang, "PFID: Pittsburgh fast-food image dataset," in *Proc. 16th IEEE Int. Conf. Image Process. (ICIP)*, Nov. 2009, pp. 289–292, doi: 10.1109/ICIP.2009.5413511.
[58] R. Kemker, A. Abitino, M. McClure, and C. Kanan, "Measuring catastrophic forgetting in neural networks," *CoRR*, vol. abs/1708.02072, Aug. 2017.
[59] A. Chaudhry, P. K. Dokania, T. Ajanthan, and P. H. S. Torr, "Riemannian walk for incremental learning: Understanding forgetting and intransigence," *CoRR*, vol. abs/1801.10112, pp. 556–572, Aug. 2018.

GHALIB AHMED TAHIR received the bachelor's degree in information technology and the master's degree in computer science from NUST, in 2012 and 2015, respectively. He is currently pursuing the Ph.D. degree with University of Malaya. He received the Masters in Project Management from Comsats 2016. His research interests include image processing and auto scaling in cloud.

CHU KIONG LOO (Senior Member, IEEE) received the B.Eng. degree (Hons.) in mechanical engineering from the University of Malaya, in 1996, and the Ph.D. degree from University Sains Malaysia, in 2004, specializing in neurorobotics. He is currently a Full Professor with the Department of Artificial Intelligence, Faculty of Computer Science and Information Technology, University of Malaya. His main research interest includes neuroscience inspired machine intelligence. He was the IEEE Systems, Man and Cybernetics SMC Society Vice-Chairman for Malaysia Chapter, from 2013 to 2014. He was also the President of Asia Pacific Neural Network Assembly (APNNA), in 2014.