IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 26 October 2024, accepted 18 November 2024, date of publication 20 November 2024, date of current version 2 December 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3503672

RESEARCH ARTICLE

Thai Food Recognition Using Deep Learning With Cyclical Learning Rates

NAWANOL THEERA-AMPORNPUNT AND PANISA TREEPONG
College of Computing, Prince of Songkla University, Phuket 83120, Thailand

Corresponding author: Panisa Treepong (panisa.t@phuket.psu.ac.th)

This work was supported by the College of Computing, Prince of Songkla University, under Grant COC6604007S and Grant COC6604032S.

ABSTRACT Automated food recognition is essential in order to streamline dietary monitoring. To build and evaluate complex food recognition models, large datasets of annotated food images are crucial. In this paper, we introduce a new dataset called THFOOD-100, which is specifically designed for this purpose. This dataset consists of 53,459 high-quality images of popular Thai dishes categorized into 100 classes. We conducted a comprehensive comparison of 23 deep convolutional neural network and vision transformer architectures to establish a strong baseline for classification performance on the THFOOD-100 dataset. Additionally, we proposed training the models using cyclical learning rates, which has been shown to improve model generalization and significantly reduce training time. We demonstrated the effectiveness of cyclical learning rates with three standard optimizers on THFOOD-100, ETHZ Food-101, and UEC-Food256. The top-performing model achieved a 96% classification accuracy on THFOOD-100, showing great promise for real-world applications. Our new dataset is specifically aimed at better representing Thai cuisine in food recognition research, and our analyses offer valuable insights into the shortcomings of current models.

INDEX TERMS Deep learning, food dataset, food recognition, image classification, learning rate.

I. INTRODUCTION

Dietary monitoring is essential for the management and treatment of many medical conditions, such as diabetes, obesity, Alzheimer's disease, and vitamin and mineral deficiency. It is also valuable to pregnant women, breastfeeding mothers, and health-conscious people, as an awareness of nutritional intake allows them to adjust their eating habits to match their goals. Food recognition systems offer a promising future for effortless dietary monitoring through automatic food labeling from images taken using devices such as smartphones, smart glasses, and wearable cameras. While different use cases may require tracking different quantities absorbed from food, such as energy, macronutrients, vitamins, and minerals, identifying the type of food consumed is a vital initial step.

Although food recognition can be considered a special case of fine-grained object recognition, where the model needs to distinguish between subcategories within a category, food images have unique characteristics that pose additional challenges compared to object recognition. For example, food ingredients have deformable shapes, and partial occlusion of ingredients in the dish is often unavoidable. Furthermore, there is a significant intra-class variation and inter-class pattern overlap. Additionally, cuisines are inherently specific to regions and cultures, unlike regular objects.

Training accurate classifiers requires large amounts of high-quality task-specific labeled data. While techniques such as transfer learning [1], weakly supervised pretraining [2], self-supervised learning [3], and synthetic data generation [4] have been developed to ease this requirement, models pretrained in the traditional supervised fashion still often perform the best [5]. Food image datasets have grown steadily in recent years, yet they predominantly feature a limited range of cuisines. Western, Chinese, and Japanese cuisines are well represented in multiple datasets. In contrast, other cuisines are significantly underrepresented or not included at all, even in datasets that cover a mix of cuisines. This lack of diversity in visual food patterns across

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

various cuisines hinders the development of methods that can accurately recognize a wide variety of cuisines. The bias also skews our perception of progress in this field. In response to this issue, we have developed THFOOD-100, a new dataset of Thai food images for the food recognition task. This dataset comprises 53,459 images categorized into 100 classes. Our aim is for this dataset to allow Thai cuisine to be better represented in research within this area.

Notable qualities of THFOOD-100 include high-quality images, accurate labels, high distinguishability, and dishes exclusively prepared by restaurants. We carefully screened the images to ensure that only high-quality ones were included in the dataset. Each image was manually preprocessed to remove irrelevant parts and enhance its quality. The image labels were specified by domain experts and had to match the user-provided labels if available. Classifying food images can be challenging due to label overlap (e.g., fried cheese and mozzarella sticks), obstruction of crucial ingredients, or visual similarity between dishes. To address this, we carefully categorized the dishes and excluded images that could not be confidently identified. We only included restaurant-prepared dishes, which were more consistent and standardized than user-prepared ones.

Convolutional neural networks (CNNs) and vision transformers (ViTs) have become prevalent in recent food recognition research due to their superior image classification accuracy compared to traditional visual feature extractors [6], [7]. However, training these models requires significant computational power, given the model's complexity, the need for large datasets, and the optimization of numerous hyperparameters. Smith [8] introduced cyclical learning rates (CLR), a type of learning rate schedule characterized by incremental and decremental steps repeated in cycles. Experimentation has shown that CLR allows for the use of higher learning rates, resulting in fewer training steps required. Throughout our study, we utilized CLR and showcased the method's effectiveness in producing models with higher accuracy with short training process. A quicker turnaround time could stimulate faster progress in the advancement of deep learning for food recognition and other related tasks.

The study's key contributions are as follows:

- We create the THFOOD-100 dataset, which consists of 53,459 high-quality images of Thai food categorized into 100 classes. This dataset is publicly accessible for research purposes.
- We propose the use of cyclical learning rate schedule for training deep neural networks for food recognition. Our evaluation across three food datasets demonstrates that it outperforms other learning rate schedules while keeping training time low.
- We establish a strong baseline for prediction performance on THFOOD-100 by thoroughly evaluating CNN and ViT architectures at two different image resolutions.

Our contributions and findings provide valuable resources and advance the development of automated food recognition methods, ultimately benefitting dietary monitoring users.

TABLE 1. List of public datasets for food recognition.

Dataset Name	Images	Classes	Cuisine(s)	Ref.
Food2K	1,036,564	2000	Chinese	[10]
ISIA Food-500	399,726	500	Mixed	[11]
ISIA Food-200	197,323	200	Mixed	[12]
ChineseFoodNet	185,628	208	Chinese	[13]
VegFru	160,731	292	Fruit & Veg.	[14]
FoodX-251	158,000	251	Mixed	[15]
ETHZ Food-101	101,000	101	Western	[16]
Fruits-360	90,483	131	Fruit & Veg.	[17]
THFOOD-100 (Ours)	**53,459**	**100**	**Thai**	
UEC-Food256	31,395	256	Mixed	[18]
MAFood-121	21,175	121	Mixed	[19]
FruitNet	19,526	6	Fruit	[20]
FruitVeg-81	15,737	81	Fruit & Veg.	[21]
THFOOD-50	15,688	50	Thai	[9]
UEC-Food100	14,361	100	Japanese	[22]
UNICT-FD1200	4,754	1200	Mixed	[23]
PFID	4,545	101	American	[24]
Sushi-50	3,963	50	Japanese	[25]
UNICT-FD889	3,583	889	Mixed	[26]
FooDD	3,000	23	Generic	[27]
ECUSTFD	2,978	19	Generic	[28]

II. RELATED WORK

A. FOOD RECOGNITION DATASETS

Large datasets of labeled food images are necessary to properly train and evaluate food recognition methods. Although the proliferation of smartphones and Internet access has led to an exponential increase in the availability of food images, the process of labeling, filtering, and preprocessing these images still require significant human efforts. As a result, currently available public datasets for food recognition are still lacking in size, quality, and variety of cuisines covered. Table 1 lists existing public datasets for the food recognition task. While the number of images in these datasets as a whole may already be sufficient to build accurate models, the diversity of cuisines covered is still inadequate. To this end, we constructed THFOOD-100, a new dataset of Thai food images focusing on high image quality and accurate labels, which allows Thai food to be better represented in research in this area.

Compared to the only other Thai food image dataset, THFOOD-50 [9], THFOOD-100 doubles the number of classes and more than triples the number of images. Images in THFOOD-50 were obtained using search engines, and images of uncooked food, images containing multiple dishes, as well as non-food images were included. To eliminate these issues, we emphasized image quality, accurate labels, and non-ambiguity through strict filtering standards and meticulous manual effort when constructing THFOOD-100. As Thai food is underrepresented in most datasets with mixed cuisines, THFOOD-100 can be readily merged with other datasets with minimal extra processing to form a larger dataset, which can be used to pretrain or benchmark models.

B. FOOD RECOGNITION

The ubiquity of mobile devices, advances in computer vision algorithms, and faster hardware have resulted in the emergence of a field of food computing. While there are many research tasks in this area, food recognition from images is one of the most fundamental tasks, with many use cases such as food logging, calorie estimation, and self-checkout. Earlier works employed color features, texture features, and traditional visual descriptors paired with standard machine-learning classifiers. Reference [29] used color histograms, speeded-up robust features (SURF), and linear support vector machines to build a real-time mobile food recognition system. In [30], ingredients were first detected at individual pixel granularity based on local textures using semantic texton forests. The frequencies and locations of detected ingredients were further classified using multi-kernel support vector machines to determine the food type.

Later, deep neural networks such as CNNs and ViTs were made possible thanks to graphic processing unit (GPU)-accelerated training and started to gain traction. Unlike other methods, deep learning models can perform both feature extraction from raw images and classification based on these features by themselves. Although these models required large amounts of training data, their classification performance was unmatched by prior methods, thanks to their ability to represent complex visual patterns. As a result, CNNs and ViTs have since become one of the most widely used image classifiers. Reference [31] compared CNN-based features to several traditional visual descriptors, including Gabor features, local color contrast, color and edge directivity descriptor (CEDD), and local binary patterns (LBP). The features were analyzed using the k-nearest neighbor algorithm and support vector machines to identify food types. The CNN features significantly outperformed all other visual descriptors. Reference [32] utilized an ensemble of two models based on Inception and ResNet architectures and an output fusion scheme for food recognition, taking the best of both models and improving the accuracy. Reference [33] used a cyclical learning rate schedule together with snapshot ensemble to produce a set of CNNs whose outputs were averaged to produce the final prediction. The idea of using cyclical learning rates to train CNNs is similar to ours. However, we use only the final model instead of multiple models from different stages of training. Reference [34] compared the recognition accuracy of ResNet, ResNeXt, and SENet architectures and used them in their food logging system, FoodAI. Reference [35] extended the Wide Residual Network to enhance recognition of vertical patterns present in many food types. Reference [36] developed a CNN with a self-attention mechanism and utilized ensemble learning to take advantage of each model's strong points, ultimately improving the classification accuracy. For our work, we compared a wide array of CNN and ViT architectures to provide a strong baseline of classification performance on THFOOD-100.

C. PORTION ESTIMATION

Many dietary monitoring applications, such as calorie counting and nutrient tracking, benefit from portion estimation or volume estimation as it allows the nutrients being tracked to be more precisely quantified. However, it is challenging to infer scale from regular images taken in an uncontrolled manner. Consequently, portion estimation methods typically use one of the following approaches: including a reference object with a known size (e.g., a spoon) in the image, using depth information from 3-dimensional (3D) depth sensors, and controlling the distance and angle between the food and the camera. To accurately estimate food volume, image-based approaches often involve reconstruction of the food's 3D model. The number of images required varies depending on the method, ranging from one image to two images to multiple images. By necessity, single-image methods make strong assumptions about the relationship between shapes and textures, leading to lower overall accuracy. Requiring multiple images imposes a significant burden on the user but yields a more accurate estimate.

Dehais et al. used two images to estimate food volume in a 3-stage system [37]. First, salient points in both images are matched, and the objects' scale is inferred from a reference card placed near the food plate. The system then performs a dense reconstruction to create a 3D model of the food. Lastly, different food items on the plate are segmented and separated from the background, and their volumes are estimated. Gao et al. also used two images of the food, one top-down and one side view [38]. Echo ranging through a smartphone's speaker and microphone was used to measure the distance to the food. Yang et al. proposed a method that infers scale from a smartphone's motion sensor, the length or the width of the smartphone, and setting the bottom of the smartphone on the tabletop (which must be flat), eliminating the need to use a reference object or a specialized depth sensor [39]. Vinod et al. used a single image of food placed on a reference board [40]. This image was matched to the reference 3D model of the food, and the object pose was estimated. The calorie was then calculated from the volume of the 3D model.

These methods, while innovative, only provide estimates of the food volume or weight. Therefore, a separate food recognition and/or food segmentation step remains crucial to estimating the nutrients in the food. Any portion estimation method can be used with any food recognition method, so that advances in either task will improve the overall accuracy of nutrient tracking.

D. CYCLICAL LEARNING RATES (CLR)

Learning rate (LR) is the central hyperparameter of the training process. At each training iteration, the magnitude of model parameter updates is the product of LR and the gradient. The most straightforward LR policy is fixed LR, where LR is kept constant throughout the training process.

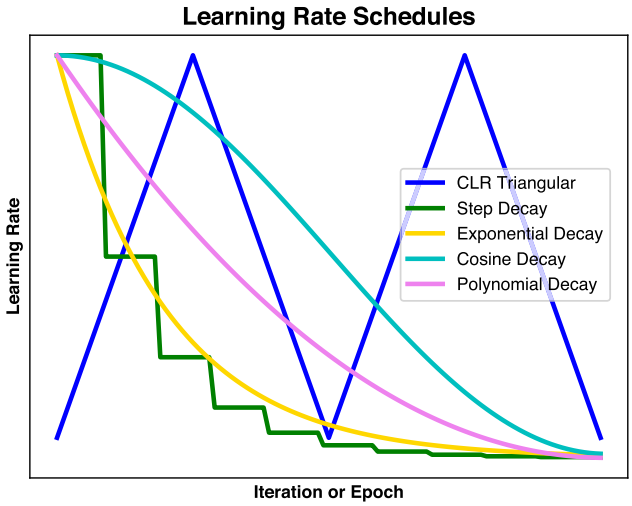

FIGURE 1. Commonly used learning rate schedules. This figure is only meant to show the overall patterns. Relative scale of learning rates and number of iterations should not be inferred from the figure.

Even then, the LR value must be appropriately tuned. Small LR results in slow training, while large LR results in the model parameters bouncing around, not converging to a local optimum. Beyond fixed LR, methods for accelerating convergence while ensuring a good optimum is reached have been developed. These methods fall into two classes: pre-defined LR schedules and adaptive learning rate methods. With pre-defined LR schedules, LR is specified as a fixed function of the training step. Commonly used LR schedules include step functions, exponential decay, cosine decay, polynomial decay, and CLR [8], as shown in Fig. 1.

With CLR, LR increases linearly from a fixed minimum to a fixed maximum, then decreases linearly back to the minimum. This cycle may be repeated multiple times. This policy is called the *triangular* policy. The author also proposed two additional CLR policies: *triangular2* and *exp_range*, where LR decays by half in the former, and by an exponential factor in the latter after each cycle. CLR has been shown to provide similar accuracy with faster convergence, made possible through a higher maximum LR [41].

In the adaptive learning rate approach, a different LR is used for each parameter instead of using the same LR for all parameters. Momentum [42] weighs each update by an average of the recent gradients, improving convergence by effectively dampening oscillation and reinforcing consistent updates. Nesterov momentum [43] additionally looks ahead and adjusts the LR accordingly based on the approximate future gradient. While both momentum-based approaches are not generally referred to as adaptive learning rate methods, they also effectively utilize a different LR for each parameter.

Other adaptive learning rate methods build upon one another. In AdaGrad [44], the LR is weighted by the cumulative gradients squared, normalizing the updates of sparse and dense parameters. RMSprop [45] and Adadelta [46] improved upon this by replacing cumulative gradients with the exponential moving average of the square of past gradients, fixing AdaGrad's problem of large cumulative gradients stagnating later parameter updates. Adadelta additionally replaced the LR with an average of recent parameter updates, eliminating the need for specifying the LR. Adam [47] built upon RMSprop by incorporating momentum into its update rule and adding bias correction. These methods are also referred to as optimization algorithms or optimizers.

Although both classes of approaches were designed toward the same goal, they can be applied simultaneously by using the product of the global LR and a parameter-specific LR as the effective LR. As CLR has been shown to provide a significant speedup without sacrificing accuracy, we pair it with adaptive learning rate methods and evaluate its effectiveness for our task.

III. THFOOD-100 DATASET

In this work, we constructed a new food image dataset called THFOOD-100 and used it to evaluate a variety of models and methods. This section describes the dataset characteristics, data collection procedure, and data preprocessing methodology. THFOOD-100[1] is a dataset of Thai food images taken by users and manually labeled by domain experts for the food recognition task. Each image has a single label corresponding to the dish's name. Both popular Thai dishes and local dishes from all regions are incorporated. Only images of restaurant-prepared food are included, and stringent criteria are imposed to ensure high image quality and accurate labels. We manually inspected all images and performed image adjustments when necessary. We imposed these requirements so that every image provides sufficient information to designate the class, at least for trained humans. This high degree of distinguishability gives us a reference point, which provides a better sense of progress when developing and evaluating classifiers. The dataset construction method is detailed below.

A. IMAGE COLLECTION

We collected food images from Wongnai, Thailand's leading restaurant review platform. Users may give ratings, write reviews, or upload photos for each restaurant. In this first step, we aimed to maximize the number of images and types of food collected. The detailed image collection steps are as follows:

1) We collected as many food images as practically possible from restaurants in all regions. Wongnai allows the uploading user to optionally provide the dish name. In order to minimize the human effort involved, only dish images labeled by the uploader were collected in this step.
2) We manually filtered out low-quality, irrelevant, and incorrectly labeled images. These included non-food images, blurry images, images with bad lighting, and images with prominent compression artifacts. We set

[1] Publicly available at https://github.com/NawanolT/THFOOD-100

FIGURE 2. Examples of excluded images. Top row shows typical images from six classes. Bottom row shows excluded images in the corresponding classes along with the reason for exclusion.

the minimum image size as 400×400 pixels. Examples of excluded images are depicted in Fig. 2.

3) We determined the classes by first grouping dish names that refer to the same dish. Dishes that were too difficult even for humans to distinguish were excluded. For example, grilled pork was often highly similar to deep-fried pork, so we included only grilled pork as it contained more images. Only dishes typically consumed as part of main meals were included. Raw fruit and vegetables, drinks, snacks, and desserts were also excluded. The top 100 classes with the most images were retained in this step.
4) At this step, some classes still had fewer than 200 images, which was the target we had set as the minimum number of images per class. Therefore, we obtained more images by manually collecting images not labeled by the user while simultaneously filtering out low-quality images. As the user-provided labels were unavailable, we only included images we could confidently identify. This step ensured an adequate number of images for both training and evaluation for every class.

B. IMAGE PREPROCESSING

As the images were taken by the users in various conditions in an uncontrolled manner, many images required further adjustments to improve the quality and remove irrelevant parts of the images. The image preprocessing steps were as follows:

1) The image's lighting was adjusted if necessary. If the resulting image was still too dark or looked unnatural, we excluded it from the dataset.
2) If an image contained multiple dishes, we divided it into multiple images, each with a single label. The divided images must still meet the earlier requirements, including image resolution.
3) The image was cropped so that the non-relevant parts were minimized, preventing confusion about which dish the label referred to and allowing the model to focus on the relevant portion.

Each image was preserved at its original resolution upon collection to accommodate models that can utilize images of any size. The median image resolution was 1920×1311.

C. IMAGE LABELING

Two domain experts independently labeled the images. Only images they could confidently label were retained. Despite this, the recognition of the remaining images remains challenging because some dishes have multiple unique ingredients or unique combinations of ingredients. The model needs to discover such information from images, each of which may visibly contain only a subset of those ingredients. Images of food cooked or prepared in non-standard styles were excluded.

Duplicated images and near-duplicates were removed from the dataset using duplicate image detection software and human verification. Two images of the same plate of food were considered non-duplicated if the camera perspectives differed.

To evaluate the labels' accuracy and provide a reference level for human classification performance, we tested the two domain experts six weeks after the labeling process using a subset of the test set. We sampled three to four images from each class to form a mini test set of 350 images. Their classification accuracy was 0.9857 and 0.9829, respectively.

D. DATASET STATISTICS

The final dataset contains 53,459 images in 100 classes. Each image contains a single label. Fig. 3 illustrates the distribution of the number of images in each class. The number of images per class ranges from 201 to 3,619, signifying a significant class imbalance. The top 20 classes account for roughly

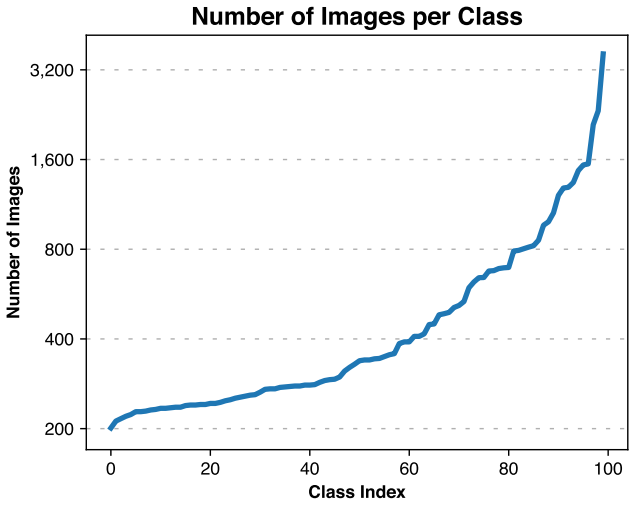

FIGURE 3. Distribution of the number of images in each class.

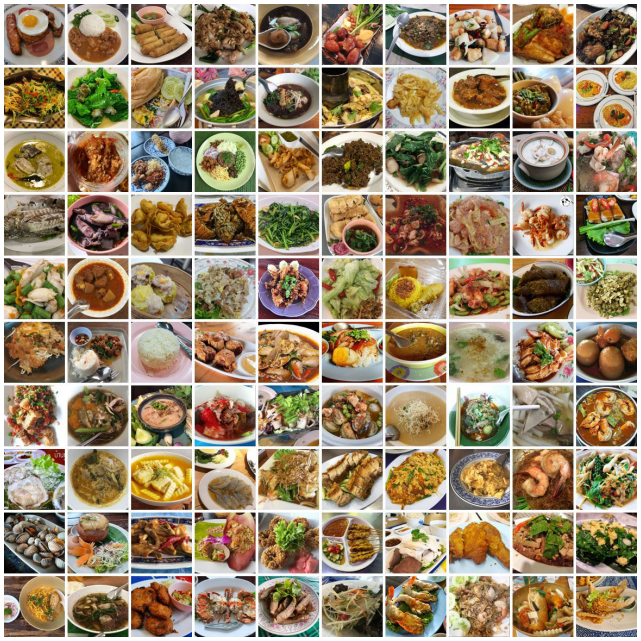

FIGURE 4. Sample images from all classes in the THFOOD-100 dataset.

half of all images. This distribution does not necessarily correspond to the users' actual consumption preference or frequency, as users tend not to upload photos of regular, everyday meals to the restaurant review platform because they seem uninteresting. We therefore suggest giving equal importance to all classes when evaluating a classifier using THFOOD-100, e.g., by measuring balanced accuracy.

Fig. 4 depicts sample images from all classes in THFOOD-100. Note that irrelevant backgrounds were removed from the images so that there is no ambiguity and attention is directed toward the actual food. Fig. 5 illustrates the strong intra-class variation of 27 randomly chosen classes. Differences in restaurants' preparation styles, lighting conditions, and

FIGURE 5. Sample images of the THFOOD-100 dataset in groups of three images per class showing strong intra-class variation.

camera angles contributed to the diversity of the images in the dataset.

Fig. 6 provides examples of visually similar classes. Images in these classes can be challenging even for trained humans to label confidently. Food ingredients have deformable shapes, and their appearance can be affected by other ingredients and the cooking method, making them harder to recognize than rigid objects. For this reason, this dataset and other food recognition datasets in general would serve as excellent benchmarks of the generalization power of image classifiers.

The classifier training, hyperparameter optimization, and evaluation processes typically require three non-overlapping subsets: the training, validation, and test sets. We divided the dataset into the training, validation, and test subsets using ratio of 55:20:25. However, instead of performing the division at the granularity of images, we divided the dataset at the granularity of restaurants. This process was performed independently for each class. For example, for class x, if restaurant y was assigned to the validation set, all images of class x from restaurant y would be assigned to the validation set. This division method allows the evaluation of a classifier's ability to generalize across *restaurants* instead of across images. Restaurants often have their own styles of preparing and garnishing a dish, making generalizations across restaurants more challenging. However, this evaluation method reflects actual use cases. We also ensured that, for each class, the validation and test sets have at least a proportionate number (20% and 25%, respectively) of unique restaurants, which helped reduce the variance of measured accuracy metrics.

FIGURE 6. Visually similar classes in THFOOD-100.

THFOOD-100 prioritizes high image quality and accurate labels, accomplished through manual verification and preprocessing. Our dataset only includes images of food prepared by restaurants, not food made in households, which is prevalent on search engines. Restaurant-prepared food is generally more standardized and of higher quality than user-prepared food. We ensured that each image contained enough information to identify the class. We also provided human experts' accuracy level of 0.9843 as a reference point and a potentially attainable goal for researchers to aim for.

IV. METHODOLOGY

The goal of this work is to establish a strong baseline of classification performance on THFOOD-100 and compare CLR with other standard LR schedules in various configurations. This section describes the training and testing processes, the CNN and ViT models used as the classifiers, the CLR method used in model training, the data augmentation technique, and the software library used in model training.

A. OVERVIEW

Figure 7 depicts the overview diagram showing the classification model training process using CLR and the testing phase. The food image dataset is divided into non-overlapping subsets for training and testing. Images in the training subset are grouped into batches which are fed into the CNN or ViT model to compute the predictions. The predictions are compared with the ground truth labels to calculate the prediction errors. The errors are used to compute the gradient. The CLR module specifies the learning rate based on the current batch number. The gradient and the learning rate are then multiplied, and the results are used to update the model's weights. The same training process repeats with another batch of data, until the specified number of passes (epoch) over the training data is reached. The resulting model can be evaluated using the test images to determine its classification accuracy or used in a food logging application.

B. CNN AND ViT MODELS

Convolutional neural networks (CNNs) are a class of artificial neural networks with convolutional layers. These layers allow detection of objects in an image regardless of where they are in the frame. As a result, CNNs perform very well in image classification tasks compared to earlier methods, such as relying on visual descriptors as feature extractors. Vision transformers (ViT) are also a class of artificial neural networks created by adapting transformer designed for natural language processing to the computer vision tasks. An image is divided into patches, which are converted to vectors. These vectors are then processed by a transformer encoder to make predictions. As CNNs and ViT gain popularity, many architectures have been developed over the years. Which architecture performs the best depends on the patterns in the images, dataset size, image resolution, among other factors. Smaller architectures tend to require smaller training sets but are not as good at capturing complex patterns compared to larger architectures. To provide a strong baseline for THFOOD-100, we compared as many standard CNN and ViT architectures as possible. They are listed along with their sizes in terms of the number of parameters in Table 2. Note that some families of architectures provide similar architectures that only differ in size (number of layers and number of nodes in each layer).

Training a CNN/ViT from scratch requires a large amount of training data, in the order of millions of images. With smaller training datasets, it is likely to suffer from overfitting. Fortunately, transfer learning is a technique that allows training CNNs/ViT using smaller datasets. The model can be further fine-tuned to adapt to the target task. A recent study showed that using transfer learning for a dataset of our size would provide faster convergence and better generalization performance than training from scratch [1]. Therefore, we used transfer learning with fine-tuning to train all models in this study.

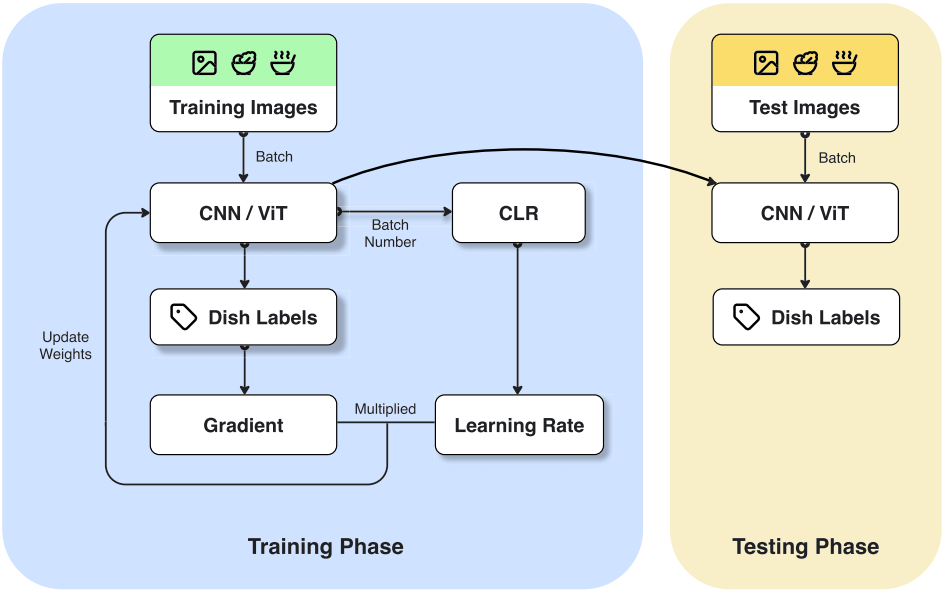

FIGURE 7. Overview diagram showing the classification model training process using CLR and the testing phase.

TABLE 2. CNN and ViT architectures included in the comparison.

CNN/ViT Architecture	Number of Parameters
DenseNet121 [48]	7.0M
DenseNet169 [48]	12.6M
EfficientNetV2B0 [49]	5.9M
EfficientNetV2B1 [49]	6.9M
EfficientNetV2B2 [49]	8.8M
EfficientNetV2B3 [49]	12.9M
EfficientNetV2S [49]	20.3M
InceptionResNetV2 [50]	51.8M
InceptionV3 [51]	21.8M
MobileNet [52]	3.2M
MobileNetV2 [53]	2.3M
MobileNetV3Large [54]	3.0M
MobileNetV3Small [54]	0.9M
MobileNetV4-Hybrid-M [55]	11.1M
NASNetMobile [56]	4.3M
RegNetX016 [57]	7.9M
RegNetX032 [57]	14.4M
RegNetY016 [57]	9.9M
RegNetY032 [57]	18.0M
ResNet50 [58]	23.6M
ResNet50V2 [59]	23.6M
SwinV2-T [60]	28.4M
Xception [61]	20.9M

The model training process with transfer learning and fine-tuning is as follows: First, the network is pretrained using the large ImageNet dataset [62]. These pretrained models are provided by the software library. In order to adapt the model for our task, the original output layer of the network is replaced by a two-dimensional global average pooling layer, followed by a dropout layer and the output layer in the one-hot encoding representation. In the transfer learning step, the weights of the original network are fixed, and only the weights of the output layer are trained. In the subsequent fine-tuning step, all weights of the network are no longer fixed and are trained at the same time. The dataset of the target task (i.e., the food recognition dataset) is used for training in both steps.

The first step is necessary because training the completely random weights of the new output layer requires a larger LR compared to fine-tuning the weights of the feature extractors of the network. Using a large LR for the whole network would destroy the feature extractors learned in the source task. After the first set of weights of the output layer is obtained, we further fine-tune the visual feature extractors represented by the network weights to adapt from the source task to the target task using a lower LR. We have found from our experiments that this second step significantly boosts the classification accuracy, indicating that some distinguishing visual patterns of food are distinct from those of everyday objects.

C. CYCLICAL LEARNING RATES

A LR schedule specifies how LR changes throughout the course of the training process. The most straightforward LR schedule is fixed LR, where LR is kept constant. Beyond fixed LR, examples of LR schedules include step functions, exponential decay, cosine decay, polynomial decay, and CLR.

Smith [8] introduced three CLR policies: *triangular*, *triangular2*, and *exp_range*. We have found that the *triangular* policy with only one cycle and 20 epochs works best for our models and dataset. CLR can be used with all standard optimizers. However, because of the tight interaction between the LR schedule and optimizer, CLR's effectiveness may vary and must be separately evaluated for each optimizer.

Even with CLR, however, the minimum and maximum LRs are hyperparameters that still need to be specified. Small LRs result in slow training, while large LRs result in the model parameters bouncing around, not converging to a local optimum. The author suggested performing an ''LR range test,'' which involves training the model for only several epochs while increasing the LR and then plotting LR versus validation accuracy. The largest LR where the accuracy still has an increasing trend would be the optimal LR. Based on our experimentation, however, the LR range test does not produce the optimal maximum LR. Due to this and the fact that the LR range test requires human input, we decided to use greedy hyperparameter optimization using the validation set to optimize maximum LR. We used $LR = 0.01 \times 1.5^k$, with k being an integer, starting with any user-defined value and increasing and decreasing until a local optimum is reached. We have found from our experiments that this optimization problem is convex, so the local optimum is also the global optimum. The minimum LR was set as 0.05 of the maximum LR, as suggested in [63]. We used the built-in CLR implementation for PyTorch and [64] for TensorFlow.

D. DATA AUGMENTATION

Data augmentation is a technique for artificially increasing the training dataset size. Images randomly undergo various transformations and adjustments to produce slightly modified images, which are then used as additional training images. Although not as helpful as additional real data, these transformed images can improve the model's generalization by reducing overfitting. There are two approaches to performing data augmentation: offline and online. In the offline approach, images are transformed before model training to produce a static augmented training dataset. Each image can be transformed in multiple ways, producing multiple augmented images. In the online approach, image transformation operations are implemented as the first layers of the network. Right before the model processes each image, the image is randomly transformed before it is used for training in place of the original image. We used the online approach to train our models because it allowed us to reap the benefits of data augmentation without increasing computational requirements due to a larger training set.

V. RESULTS

We first compare various CNN and ViT architectures to provide a strong baseline on THFOOD-100. We then compare CLR to other LR schedules using commonly used optimizers on three food datasets. We assess the effectiveness of data augmentation in improving classification performance. We examine the best model's misclassifications to gain insights into improving the results further. Finally, we analyze the relationship between the size of the dataset and the prediction accuracy to determine how much improvement can be gained with more data.

A. EVALUATION MEASURES

We use balanced accuracy (BA) as the primary performance metric. The conventional accuracy metric is affected by the class distribution of test data, which is highly unbalanced on THFOOD-100. Conventional accuracy would give a larger emphasis on the classes with more images. A classifier's performance can be more appropriately measured by giving equal importance to each class. Balanced accuracy is defined as

$$\mathrm{BA} = \frac{1}{m}\sum_{i=1}^{m}\frac{\mathrm{TP}_i}{\mathrm{TP}_i + \mathrm{FN}_i}, \tag{1}$$

where m is the number of classes and TP_i and FN_i are the numbers of true positives and false negatives for class i, respectively. It can be viewed as either the average of recall of each class or the weighted accuracy, where each sample is weighted inversely proportional to the number of samples in that class. BA ranges from 0 to 1, with a higher value corresponding to a lower classification error rate, similarly to conventional accuracy. In addition to the described top-1 BA, we also use top-3 BA as a secondary metric. An image's classification is considered correct if any of the top three predicted labels matches the true label.

B. TRAINING PROCESS

We used transfer learning with fine-tuning to train all CNN and ViT models. We used CLR with 20 epochs as the LR schedule to accelerate training. We used the Adam optimizer by default. LRs for the transfer learning and fine-tuning steps were separately optimized using the validation set. We used the greedy approach to optimize all hyperparameters. Class imbalance was handled by assigning class weights inversely proportional to the number of training samples. Non-square images were maximally center-cropped to a square shape before other operations. Two image resolutions were used: 256×256 and 512×512 pixels. Random zooming and random cropping data augmentation methods were implemented as the first layers of all models. Random zoom ratios were in the range of [1, 1.25]. 256×256 images were randomly cropped to 224×224, and 512×512 images were randomly cropped to 448×448. Random zooming was not applied during testing, and images were center-cropped to produce the same image resolution as training images. The image's aspect ratio was always kept identical to the original image's. The batch size was 64, while the dropout ratio was 0.3. We repeated the training and evaluation of each configuration at least three times and reported the average value.

We implemented MobileNetV4-Hybrid-M and SwinV2-T using the PyTorch library and the remaining CNN models using the TensorFlow library. All implementation details were matched to ensure a fair comparison. We conducted all experiments on a machine with an Intel Core i5-12400F processor, 64 gigabytes of memory, and an Nvidia GeForce RTX 3090 graphic processing unit (GPU) running the Ubuntu 23.10 operating system.

TABLE 3. CNN and ViT architecture comparison on THFOOD-100.

Architecture	Image Resolution			
	256 × 256		512 × 512	
	Top-1	Top-3	Top-1	Top-3
DenseNet121 [48]	0.8815	0.9654	0.9244	0.9823
DenseNet169 [48]	0.8916	0.9732	0.9318	0.9849
EfficientNetV2B0 [49]	0.9049	0.9777	0.9463	0.9880
EfficientNetV2B1 [49]	0.9111	0.9764	0.9493	0.9902
EfficientNetV2B2 [49]	0.9088	0.9786	0.9490	0.9880
EfficientNetV2B3 [49]	0.9131	0.9791	0.9525	0.9906
EfficientNetV2S [49]	0.9195	0.9809	0.9476	0.9887
InceptionResNetV2 [50]	0.8787	0.9657	0.9339	0.9857
InceptionV3 [51]	0.8426	0.9489	0.9168	0.9795
MobileNet [52]	0.8435	0.9522	0.9015	0.9765
MobileNetV2 [53]	0.8663	0.9602	0.9093	0.9765
MobileNetV3Large [54]	0.8919	0.9704	0.9334	0.9831
MobileNetV3Small [54]	0.8428	0.9507	0.9151	0.9797
MobileNetV4-Hybrid-M [55]	0.9116	0.9775	0.9416	0.9832
NASNetMobile [56]	0.8563	0.9546	0.9103	0.9762
RegNetX016 [57]	0.8838	0.9689	0.9234	0.9825
RegNetX032 [57]	0.8802	0.9659	0.9210	0.9802
RegNetY016 [57]	0.8938	0.9725	0.9230	0.9806
RegNetY032 [57]	0.8973	0.9719	0.9404	0.9869
ResNet50 [58]	0.8608	0.9554	0.9127	0.9770
ResNet50V2 [59]	0.8182	0.9370	0.8916	0.9704
SwinV2-T [60]	**0.9522**	**0.9913**	**0.9638**	**0.9936**
Xception [61]	0.8674	0.9592	0.9225	0.9808

C. CNN AND ViT ARCHITECTURE COMPARISON

Table 3 presents the comparison of CNN and ViT architectures on THFOOD-100. There is a considerable difference in classification accuracy across different architectures. SwinV2-T performed best overall, followed by EfficientNetV2 family of models and MobileNetV4-Hybrid-M. Interestingly, models that performed well included both small and large models. The rankings of architectures at both image resolutions are similar. At image resolution of 256 × 256, SwinV2-T outperformed other models by a large margin, indicating that the model can effectively recognize fine details even at low resolution. At 512 × 512, the gaps narrowed down but SwinV2-T still produced highest classification accuracy of 0.9638. Comparing this to human experts' accuracy of 0.9843, SwinV2-T still made 131% more errors than humans, which suggests that there is still room for improvements. However, such level of accuracy is already adequate for food logging, where the user can correct the occasional errors, especially since top-3 accuracy is over 0.99.

Increasing image resolution from 256 × 256 to 512 × 512 lowered the error rate by 39% on average, except for SwinV2-T where the error rate decreased by 24%. Such a significant improvement despite no other changes to the models implies that the level of detail present in 256 × 256 images is insufficient for accurate classification for most models. On the other hand, the smaller accuracy gap for SwinV2-T suggests that it is more effective at classifying lower-resolutioned images. Increasing image resolution raises both memory and computational requirements, which are proportional to the number of pixels. 512 × 512 is the image resolution limit on our single-GPU testbed.

D. CYCLICAL LEARNING RATES

In this experiment, we compared CLR with other LR schedules using common optimizers. The top two models for the 512 × 512 image resolution (SwinV2-T and EfficientNetV2B3) and their best configurations for THFOOD-100 from the previous section were employed. We chose the higher resolution as the basis of the comparison because the classification accuracy at the lower resolution was inadequate for real-world usage. Adam, Stochastic Gradient Descent (SGD), and SGD with Nesterov momentum (with momentum = 0.99) were included as the optimizers. The learning rate was optimized separately for each configuration. Four LR schedules were included in the comparison: CLR with 20 epochs, fixed LR with 100 epochs, fixed LR with 20 epochs, and cosine decay with 20 epochs. With cosine decay, a linear warmup period of 1 epoch was used, increasing the learning rate from 0.05 of the target LR to the target LR. A cosine decay function was then applied for the remaining 19 epochs. These four LR schedules were evaluated using THFOOD-100 and two standard benchmark datasets: ETHZ Food-101 and UEC-Food256. 55%, 20%, and 25% of the images in each dataset were assigned as the training set, validation set, and test set, respectively. Balanced accuracy was the primary performance metric for THFOOD-100, but conventional accuracy was used for ETHZ Food-101 and UEC-Food256 to allow direct comparison with existing works. Statistical significance tests were performed using one-tailed unpaired t-tests. p-values lower than 0.05 indicate that the difference is statistically significant.

ETHZ Food-101 is a commonly used benchmark dataset for food recognition. It contains 101,000 images in 101 classes and was developed by Bossard et al. [16]. The images were taken by users and collected from the Internet. The dataset includes food from various cuisines, but the majority are Western. The classes are perfectly balanced, with 1,000 images each. The authors provided an official test set split with manually cleaned images and labels. The remaining training images were intentionally not cleaned, so they contained some noises, such as unnatural colors and incorrect labels. We randomly split the remaining images into the training set and the validation set.

UEC-Food256 is a publicly available food image dataset with 31,395 images in 256 classes, developed by Kawano and Yanai [18]. It was expanded from UEC-Food100, which has 100 classes of primarily Japanese food, to include various cuisines such as American, Italian, French, Chinese, Korean, Vietnamese, Thai, and Indonesian. The 256 classes are unbalanced, with 100 to 728 images in each class. The authors did not provide an official split, so we randomly divided the dataset into the training set, validation set, and test set. The dataset also provides the bounding box indicating the

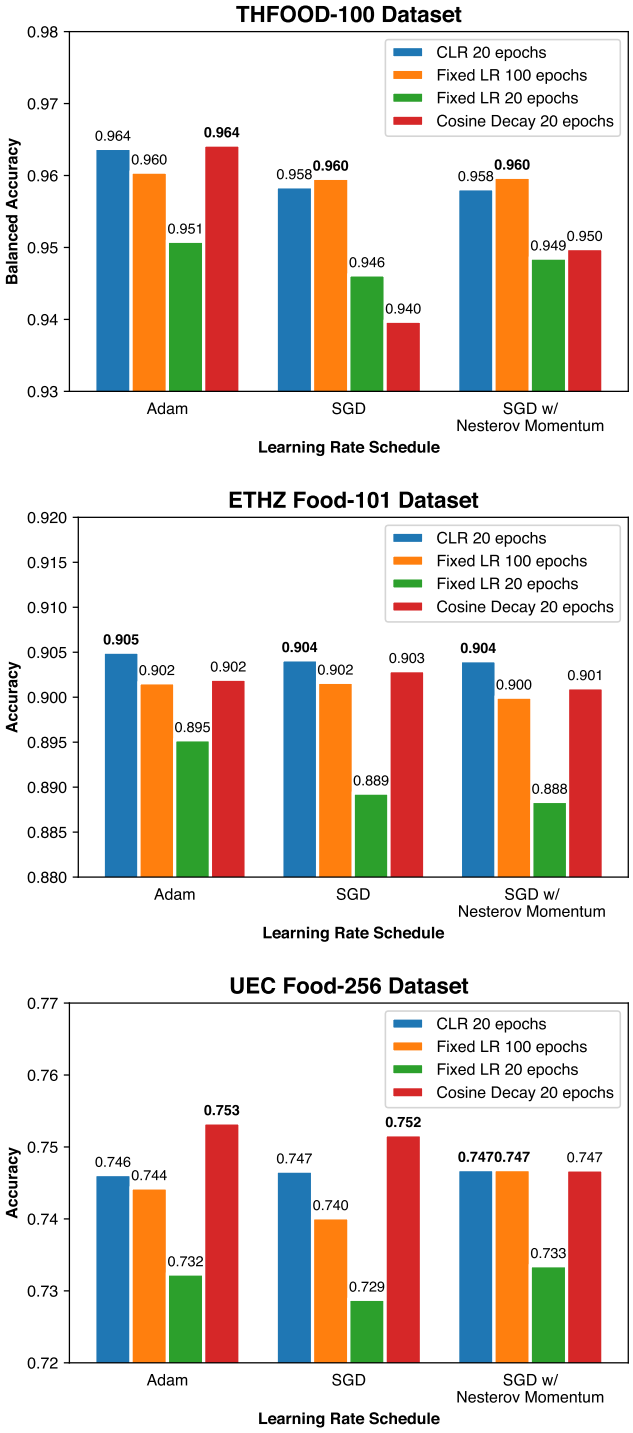

FIGURE 8. Comparison of learning rates schedules on three datasets using SwinV2-T architecture.

location of the food item in each image, but we did not use this information in our training scheme.

Fig. 8 presents the results of comparison using the SwinV2-T ViT architecture. On THFOOD-100, CLR produced similar levels of accuracy (p-values > 0.23) to cosine decay with Adam optimizer, and fixed LR 100 epochs with SGD and SGD w/ Nesterov momentum optimizer. In the remaining cases, CLR produced higher accuracy than other LR schedules with statistical significance (p-values < 0.016). The choice of optimizer did have a small impact on the accuracy in most cases. Overall, the Adam optimizer tended to produce highest accuracy.

On ETHZ Food-101, CLR outperformed all other LR schedules with statistical significance (p-values < 0.035). Cosine decay and fixed LR 100 epochs produced similar results, while fixed LR 20 epochs produced lowest accuracy. The choice of optimizer did not significantly affect the results.

On UEC Food-256, with Adam optimizer, CLR produced similar accuracy to fixed LR 100 epochs, and lower accuracy than cosine decay. When SGD was used, CLR outperformed fixed LR 100 epochs (p-value $= 0.037$) and produced similar accuracy to cosine decay (p-value $= 0.070$). With SGD w/ Nesterov momentum, CLR, fixed LR 100 epochs, and cosine decay produced similar results. Fixed LR 20 epochs produced lowest accuracy regardless of optimizer used. The choice of optimizer affected the results slightly but there were no systematic trends that held across all LR schedules.

To summarize the results for the SwinV2-T model, the number of cases where CLR performed better, equal, and worse than cosine decay are 5, 3, 1. The number of cases where CLR performed better, equal, and worse than fixed LR 100 epochs are 5, 4, 0. Cosine decay performed well on the UEC Food-256 dataset, while CLR performed well on the THFOOD-100 and ETHZ Food-101 datasets. We inspected these datasets to find key differences that might help explain the contrasting results. We found that UEC Food-256 contains many images of 3–5 Japanese dishes served together in small plates and bowls. Each dish often has a valid label of their own, so there can be confusion about which dish the label refers to. The authors of the dataset provided a bounding box information which would eliminate the problem, but we did not use it because we framed our task as image classification rather than object detection. Furthermore, the number of images per class for UEC Food-256 is lower at 123, compared to THFOOD-100's 535 and ETHZ Food-101's 1,000. UEC Food-256 includes a more varied types of cuisines than the other datasets. These differences in the characteristics of the datasets could have led to the dissimilar results.

Fig. 9 depicts the results of the comparison using EfficientNetV2B3 CNN architecture. On THFOOD-100, CLR produced similar levels of accuracy to fixed LR 100 epochs when the SGD or SGD w/ Nesterov momentum optimizer was used (p-values ≥ 0.45), surpassing fixed LR 20 epochs and cosine decay with statistical significance (p-values ≤ 0.03) in all cases except between CLR and cosine decay when SGD w/ Nesterov momentum optimizer was used (p-value $= 0.12$). With Adam optimizer, however, CLR performed significantly better than the remaining LR schedules (p-values ≤ 0.02). If we turn our focus to the optimizers, we found no significant differences between them, except in one case: when CLR were used with Adam, the accuracy was higher than with other

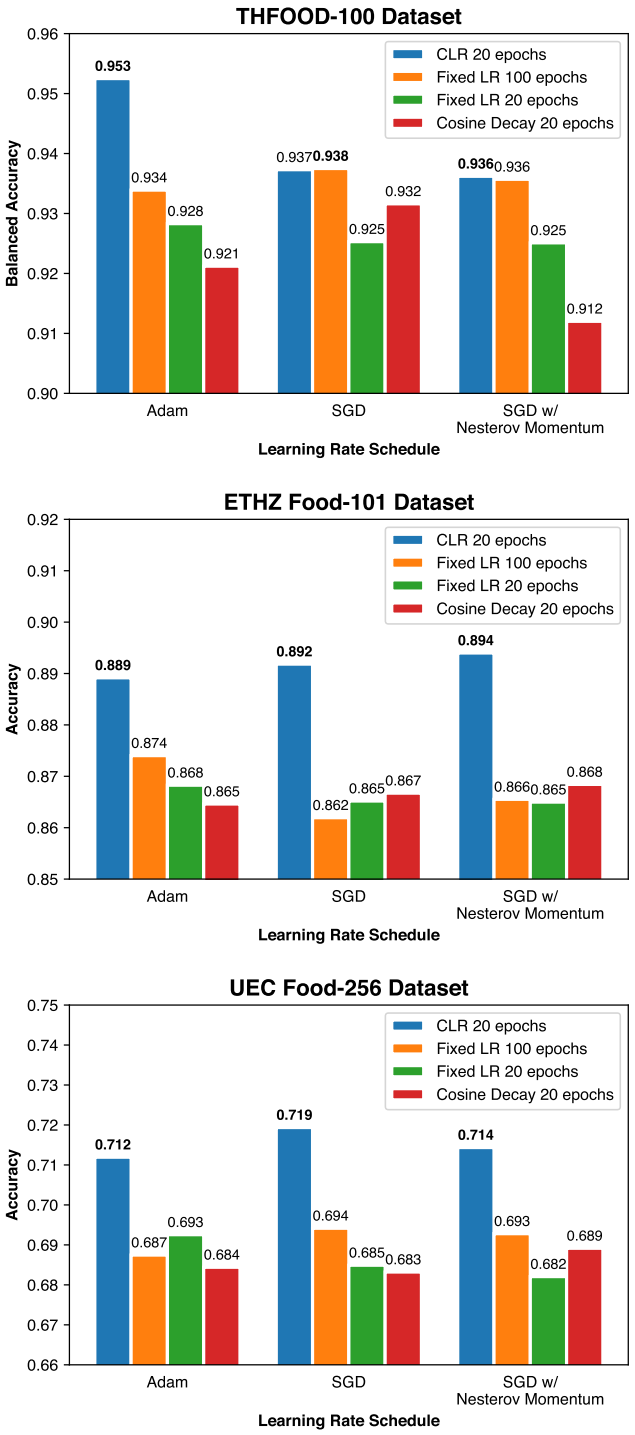

FIGURE 9. Comparison of learning rates schedules on three datasets using EfficientNetV2B3 architecture.

optimizers (p-values $\leq$ 0.009). As training time is proportional to the number of epochs but not affected by the LR schedule or optimizer, CLR 20 epochs requires only one-fifth of the training time required by fixed LR 100 epochs. Although CLR produced similar accuracy to fixed LR 100 epochs in some cases, the 5x training speedup it provides is still substantial advantage.

TABLE 4. Existing results on food recognition on the ETHZ Food-101 dataset.

Authors	Year	Method/Model	Accuracy
Bossard et al. [16]	2014	Random Forest	50.76%
Bossard et al. [16]	2014	CNN – AlexNet	56.40%
Yanai and Kawano [65]	2014	CNN – AlexNet	70.41%
Meyers et al. [66]	2015	CNN – GoogLeNet	79.00%
Liu et al. [67]	2015	CNN – GoogLeNet	77.40%
Cui et al. [68]	2017	CNN – ResNet-50	85.50%
Ofli et al. [69]	2017	CNN – ResNet-50	80.90%
Cui et al. [70]	2018	CNN – Inception V3	88.80%
Lee et al. [71]	2018	CNN – ResNet-50	81.67%
Martinel et al. [35]	2018	CNN – WISeR	90.27%
Tan and Le [72]	2019	CNN – EfficientNet-B4	**91.50%**
Kornblith et al. [1]	2019	CNN – Inception V4	90.00%
Tan and Le [73]	2019	CNN – MixNet-M	87.20%
Grill et al. [74]	2020	CNN – ResNet-50	88.50%
Chen et al. [75]	2020	CNN – ResNet-50	89.40%
Dwibedi et al. [76]	2021	CNN – ResNet-50	76.70%
Zhang et al. [77]	2021	CNN – ResNet-50	85.28%
Li et al. [78]	2023	CNN – PresNet	89.02%
Sheng et al. [79]	2024	CNN – GSNet-2.0	88.40%

TABLE 5. Existing results on food recognition on the UEC-Food256 dataset.

Authors	Year	Method/Model	Accuracy
Kawano and Yanai [80]	2014	Fisher Vector	50.10%
Yanai and Kawano [65]	2015	CNN – AlexNet	67.57%
Liu et al. [67]	2016	CNN – GoogLeNet	54.70%
Hassannejad et al. [81]	2016	CNN – Inception V3	76.17%
Bolanos and Radeva [82]	2016	CNN – GoogLeNet	63.16%
Martinel et al. [35]	2018	CNN – WISeR	**83.15%**
Ciocca et al. [83]	2018	CNN – ResNet-50	71.70%
Tahir and Loo [84]	2020	CNN – InceptionResNetV2	69.29%
Mao et al. [85]	2021	CNN – DenseNet-121	72.36%
Xiao et al. [86]	2024	Vision Transformer – Swin-DR	82.77%

On ETHZ Food-101 and UEC Food-256, CLR's results stood out from the rest by a significant margin, regardless of the optimizer used (p-values $\leq$ 0.006 for ETHZ Food-101 and p-values $\leq$ 0.015 for UEC Food-256). The results for fixed LR 100 epochs, fixed LR 20 epochs, and cosine decay were similar, as there were no systematic trends across configurations. There were also no significant differences between the three optimizers.

Comparing between fixed LR 100 epochs and fixed LR 20 epochs, the results on THFOOD-100 were better with 100 epochs, just as we had expected the gradual optimization process to yield a better model. However, on ETHZ Food-101 and UEC Food-256, the results with 100 epochs were not always better. Twenty training epochs could be enough for the models to capture the patterns on these datasets, or the variation across runs is simply larger than the effect of increasing the number of training epochs.

TABLE 6. Comparison of various data augmentation methods.

Data Augmentation Method	Top-1	Top-3	p-value
None	0.9607	0.9938	(baseline)
Brightness	0.9594	0.9928	0.9441
Contrast	0.9597	0.9930	0.8658
Flipping	0.9628	0.9933	0.0502
Rotation	0.9614	0.9931	0.2511
Zooming	0.9644	0.9936	**0.0036**
Cropping	**0.9654**	**0.9947**	**0.0105**
Zooming and cropping	0.9638	0.9934	**0.0062**

Baseline of comparison is the no data augmentation setup (denoted "None"). p-values are based on comparing top-1 balanced accuracy, and values lower than 0.05 are bolded.

To summarize the results on both models, CLR outperformed cosine decay and fixed LR 100 epochs in most cases (13 out of 18 and 12 out of 18 cases, respectively). In the remaining cases, cosine decay and fixed LR 100 epochs produced similar results to CLR, except for only one case where cosine decay performed better than CLR. Fixed LR 20 epochs performed worst in most cases. It is important to note that fixed LR 100 epochs require 5x training time compared to other setups which required only 20 epochs. To put it in perspective, training the SwinV2-T on the training and validation subsets for 20 epochs takes roughly 3.5 hours on our testbed. With 100 epochs, the training time becomes 17.4 hours. High training time makes it more costly to perform hyperparameter optimization, ultimately resulting in suboptimal model performance. For reference, previous results on the ETHZ Food-101 and UEC-Food256 datasets are presented in Tables 4 and 5. Our results did not set a new state of the art, as this was not the intention of our work. We simply used the neural network architectures that worked best for THFOOD-100 for these datasets. However, the fact that CLR produced better accuracy than other LR schedules in most configurations suggests that the state-of-the-art models could produce even better results if combined with CLR.

E. DATA AUGMENTATION

Data augmentation is a commonly employed technique to reduce overfitting and improve the accuracy of a classifier. This experiment aims to evaluate whether data augmentation is beneficial and which transformation or combination thereof produces the significant improvements on THFOOD-100.

We employed the same best configuration from the previous experiments. The SwinV2-T architecture was used. The image resolution was 512 × 512 when cropping data augmentation was used and 448 × 448 otherwise, keeping the resolution of the transformed image uniform. Each type of data augmentation involves performing the alteration to a random degree within the specified range. The range for brightness adjustment was [−0.2, 0.1] relative to the range of pixel values (255). The rotation angle's range was not limited, and the missing pixels of the rotated image were filled with reflections of the image. Cropping was performed by choosing a random 448 × 448 part of the image to retain. The range for contrast adjustment was [−0.2, 0.2]. As we wished to restrict zooming to zooming in only, each image's zoom ratio was randomized between 1 and 1.43x (corresponding to 0.7 height and width factors). The image's aspect ratio was kept identical to the original image's. Data augmentation was only performed during training, except for cropping, which became center cropping during testing. Each method was compared to the no data augmentation setup using one-tailed unpaired t-tests, with the expectation that data augmentation would increase the accuracy.

The results are presented in Table 6. As the top-3 balanced accuracy is similar and close to 1, we focus our attention on top-1 balanced accuracy. We first started with a comparison of single image transformations. Only zooming and cropping improved the accuracy with statistical significance. We subsequently evaluated the combination of zooming and cropping. However, zooming alone, cropping alone, and their combination all resulted in similar accuracy improvements. Cropping, which produced highest accuracy, decreased error rate by 12%.

Overall, the results are similar to those reported in [87], where zooming in improved the accuracy. The edges of food images typically only contain noninformative parts, such as the food plate, sauces, drinks, and background. Zooming in magnifies the patterns of actual food, thereby effectively increasing image resolution. However, compared to fixed zoom ratios, random zoom ratios provide the same benefits while allowing the model to learn about patterns near the edges of the images. These edges are critical in some images as the main part of the food is not always in the center. The fact that combining zooming and cropping did not yield further improvements suggests that cropping could be providing similar effects to zooming. Favoring simplicity, we conclude that either zooming or cropping alone is the optimal data augmentation method, providing a modest yet consistent improvement.

F. MISCLASSIFICATION ANALYSIS

In this section, we examine the top model's misclassifications to gain insights into its shortcomings and potential approaches to further reducing the errors. We sampled a uniform number of images for each class from the training set to remove the confounding effect of class imbalance. After sampling, all classes contained 151 training images. The top errors, defined as top three pairs of classes (x, y) where $x \neq y$ with the highest fraction of images in class x misclassified as class y, are depicted in Fig. 10. Among the top errors, dumplings misclassified as crispy pork belly, deep-fried squid misclassified as deep-fried soft-shell crab, and fried wontons misclassified as fried spring rolls were the most prevalent, accounting for 4.65%, 3.49%, and 2.67% of all errors, respectively.

FIGURE 10. Top misclassifications of the best model. On each row, images on the left are misclassified images with the actual labels while images on the right are training images of the predicted class.

Dumplings and crispy pork belly are dishes humans can easily recognize and distinguish. The primary source of the model's confusion was likely the shrimp roe on top of the dumplings, as all misclassified images contained it. In our dataset, several restaurants top the dumplings with shrimp roe, but most of their images were assigned to the test set. The few remaining images in the training set also had different appearances. Thus, the model, encountering the unfamiliar dumpling styles, mistakenly associated the shrimp roe with the crispy pork belly's skin, which was its unique characteristic. The presence of soy sauce, more frequently served with crispy pork belly, might have also contributed to the confusion. This shortcoming of the model does not apply to humans, who view objects as a whole and would realize that the objects' overall shapes were closer to dumplings than crispy pork belly. Generative models mimic human perception by considering the overall object, unlike discriminative models such as CNNs that focus solely on distinguishing features. However, developing generative models that can classify images as well as discriminative models with limited data remains a challenge. Increasing the dataset size remains a reliable and straightforward solution that could prevent this misclassification.

Deep-fried squid being misclassified as deep-fried soft-shell crab was the next biggest source of errors. While squid and soft-shell crab have distinctive shapes and textures, both dishes are similarly coated in batters and often topped with fried garlic, which covers the meat and hinders recognition. However, after examining all misclassified images, we can confidently say that humans would not make such mistakes, thanks to better comprehension of object structures in three dimensions and perception of minute details such as squid skin and tentacles. The model likely could not group the similarities among the diverse images and generalize from them due to the lack of structural understanding. Algorithmic improvements are necessary to overcome this limitation, although more data would incrementally reduce these errors.

Fried wontons and fried spring rolls are both deep-fried dishes with similar golden brown color tones. Both are often served with plum sauce or chili sauce. While the textures of both dishes are often similar, in most cases their shapes are unmistakably different. Fried spring rolls have cylindrical shapes while fried wontons generally have triangular shapes. Indeed, the misclassified fried wontons did not have triangular shapes. They were folded in an uncommon way that the middle part resembled the ends of fried spring rolls. While there were a wide variation of fried wonton shapes in the training images, none looked like the misclassified images. Therefore, it is understandable that the model made such mistakes. Humans, however, are less likely to have the same confusion, because the fried wontons did not have cylindrical shapes. This suggests that the model relies on textures more than objects' shapes when classifying images that are different from what it has seen before. Ensuring that the training dataset is

comprehensive enough to cover most or all patterns the user may encounter is one solution. Nevertheless, a more reliable and permanent solution would be to develop new models with improved shape comprehension that rely on shapes more than textures.

In summary, the model had good texture recognition, potentially comparable to humans'. Dishes that the model was confused with usually had an overall composition and textures similar to the actual dish. However, it faced a challenge in recognizing the shapes of objects, an area where it is noticeably less proficient than humans. Some misclassifications were due to insufficient patterns covered by the training set. More data and improvements to the models are two approaches that would allow the models will close the gap and come closer to human experts' level of classification accuracy.

G. ACCURACY SCALING WITH DATASET SIZE

Even with transfer learning, neural networks still benefit from more training data for the target task. At some point, however, there will likely be diminishing returns. With this experiment, we aimed to explore the trends of how much the size of the training dataset impacts prediction performance. This knowledge would guide future efforts in data collection toward improving the models.

We performed the experiments in two scenarios. In the first scenario, we sampled an equal number of images from each class to form a balanced training set. In the second scenario, we maintained the original unbalanced class distribution while varying the fraction of the training data used. In both cases, the entire test set was used for evaluation.

The results are illustrated in Fig. 11. We plot the accuracy on the logit scale, defined with the logit function

$$\begin{aligned}\text{logit}(p) &= \log\left(\frac{p}{1-p}\right) \\ &= \log\left(\frac{n_{\text{correct}}}{n_{\text{incorrect}}}\right), \end{aligned} \tag{2}$$

where p is the balanced accuracy, and n_{correct} and $n_{\text{incorrect}}$ are the weighted numbers of correctly and incorrectly classified samples, respectively. An additive increase in accuracy in the logit scale corresponds to a multiplicative increase in the odds of a correct classification. The logit function can capture the fact that as accuracy approaches 1, it becomes increasingly difficult to increase the accuracy by the same absolute amount. The training set sizes are plotted on a logarithmic scale.

In the case of balanced classes, each doubling of the training set size produces a linear improvement in top-1 and top-3 accuracy for the entire range of 1–151 images per class. When the classes are unbalanced, each doubling of the training set size produces a linear improvement in top-1 and top-3 accuracy from 4–267 images per class. Doubling the number of images per class to 534 provides diminishing returns. These results indicate that the model can still benefit from more training data, although the rate

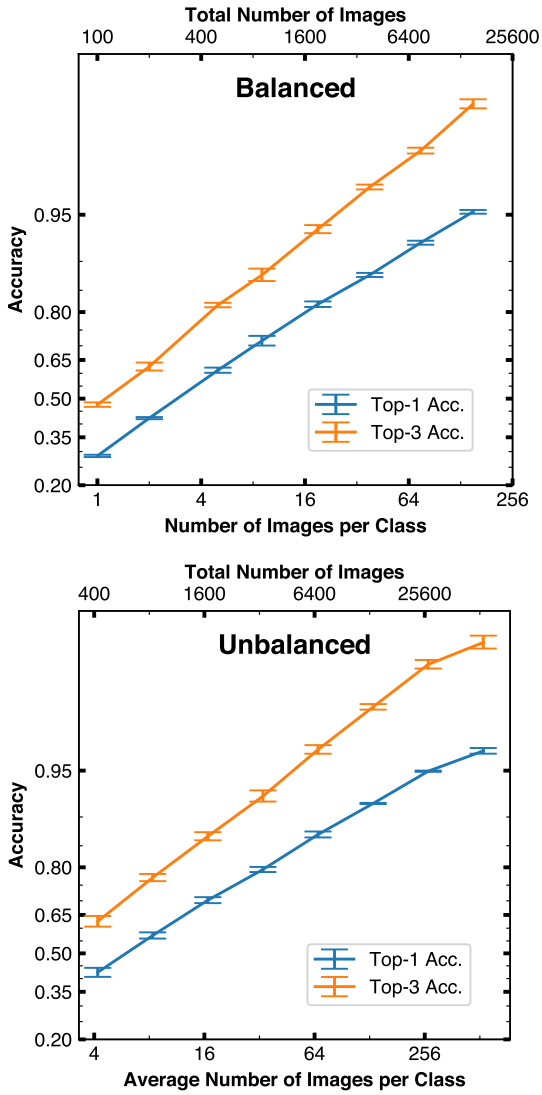

FIGURE 11. How prediction accuracy scales with training dataset size. Top: balanced classes. Bottom: unbalanced classes. Y-axes are in logit scale, while X-axes are in logarithmic scale.

of improvements starts to decrease beyond 267 images per class. Therefore, collecting more data remains a straightforward approach to improving classification accuracy. Other potential approaches include creating new neural network architectures targeted to food recognition, more extensive or better pretrain datasets than ImageNet, and better transfer learning techniques. One of the ultimate goals of computer vision is creating classification models that perform well (e.g., at the human level) without a large amount of training data. However, for now, it remains a challenge.

VI. CONCLUSION

This paper introduces THFOOD-100, a new dataset for Thai food recognition comprising 53,459 images categorized into 100 classes, made publicly available for research purposes.

We established a strong baseline for classification performance by extensively comparing CNN and ViT architectures. Our experiments with 256 × 256 and 512 × 512 image resolutions demonstrated that SwinV2-T, unlike other models, can achieve satisfactory accuracy even at a lower resolution. Comparison between CLR and other LR schedules on three food datasets demonstrated CLR's superior performance. The resulting models generally exhibited better generalization despite the low training time. Analyses of the top model's misclassifications and how accuracy scales with dataset size suggest that more data would further reduce the error rate. However, further advances in classification models or training methods are necessary to attain human-level performance, particularly with limited data.

REFERENCES

[1] S. Kornblith, J. Shlens, and Q. V. Le, "Do better ImageNet models transfer better?" in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 2661–2671.

[2] M. Singh, L. Gustafson, A. Adcock, V. De Freitas Reis, B. Gedik, R. P. Kosaraju, D. Mahajan, R. Girshick, P. Dollár, and L. Van Der Maaten, "Revisiting weakly supervised pre-training of visual perception models," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 794–804.

[3] Y. Tian, W. Zhang, P. Su, Y. Xu, P. Zhuang, X. Xie, and W. Zhao, "S4: Self-supervised learning with sparse-dense sampling," *Knowl.-Based Syst.*, vol. 299, Sep. 2024, Art. no. 112040.

[4] M. B. Sariyildiz, K. Alahari, D. Larlus, and Y. Kalantidis, "Fake it till you make it: Learning transferable representations from synthetic ImageNet clones," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 8011–8021.

[5] M. Goldblum, H. Souri, R. Ni, M. Shu, V. Prabhu, G. Somepalli, P. Chattopadhyay, M. Ibrahim, A. Bardes, and J. Hoffman, "Battle of the backbones: A large-scale comparison of pretrained models across computer vision tasks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 36, 2024, pp. 1–12.

[6] W. Min, S. Jiang, L. Liu, Y. Rui, and R. Jain, "A survey on food computing," *ACM Comput. Surveys*, vol. 52, no. 5, pp. 1–36, Sep. 2020.

[7] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16×16 words: Transformers for image recognition at scale," 2020, *arXiv:2010.11929*.

[8] L. N. Smith, "Cyclical learning rates for training neural networks," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2017, pp. 464–472.

[9] C. Termritthikun, P. Muneesawang, and S. Kanprachar, "NU-InNet: Thai food image recognition using convolutional neural networks on smartphone," *J. Telecommun., Electron. Comput. Eng. (JTEC)*, vol. 9, nos. 2–6, pp. 63–67, 2017.

[10] W. Min, Z. Wang, Y. Liu, M. Luo, L. Kang, X. Wei, X. Wei, and S. Jiang, "Large scale visual food recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 8, pp. 1–18, Aug. 2023.

[11] W. Min, L. Liu, Z. Wang, Z. Luo, X. Wei, X. Wei, and S. Jiang, "ISIA food-500: A dataset for large-scale food recognition via stacked global-local attention network," in *Proc. 28th ACM Int. Conf. Multimedia*, Oct. 2020, pp. 393–401.

[12] W. Min, L. Liu, Z. Luo, and S. Jiang, "Ingredient-guided cascaded multi-attention network for food recognition," in *Proc. 27th ACM Int. Conf. Multimedia*, Oct. 2019, pp. 1331–1339.

[13] X. Chen, Y. Zhu, H. Zhou, L. Diao, and D. Wang, "ChineseFoodNet: A large-scale image dataset for Chinese food recognition," 2017, *arXiv:1705.02743*.

[14] S. Hou, Y. Feng, and Z. Wang, "VegFru: A domain-specific dataset for fine-grained visual categorization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 541–549.

[15] P. Kaur, K. Sikka, W. Wang, S. Belongie, and A. Divakaran, "FoodX-251: A dataset for fine-grained food classification," 2019, *arXiv:1907.06167*.

[16] L. Bossard, M. Guillaumin, and L. Van Gool, "Food-101–mining discriminative components with random forests," in *Proc. Comput. Vis. ECCV 13th Eur. Conf.*, Zurich, Switzerland. Cham, Switzerland: Springer, Sep. 2014, pp. 446–461.

[17] H. Muresan and M. Oltean, "Fruit recognition from images using deep learning," *Acta Universitatis Sapientiae, Inf.*, vol. 10, no. 1, pp. 26–42, Aug. 2018.

[18] Y. Kawano and K. Yanai, "Automatic expansion of a food image dataset leveraging existing categories with domain adaptation," in *Proc. Comput. Vis. (ECCV) Workshops*, Zurich, Switzerland. Cham, Switzerland: Springer, Sep. 2015, pp. 3–17.

[19] E. Aguilar, M. Bolaños, and P. Radeva, "Regularized uncertainty-based multi-task learning model for food analysis," *J. Vis. Commun. Image Represent.*, vol. 60, pp. 360–370, Apr. 2019.

[20] V. Meshram and K. Patil, "FruitNet: Indian fruits image dataset with quality for machine learning applications," *Data Brief*, vol. 40, Feb. 2022, Art. no. 107686.

[21] G. Waltner, M. Schwarz, S. Ladstätter, A. Weber, P. Luley, M. Lindschinger, I. Schmid, W. Scheitz, H. Bischof, and L. Paletta, "Personalized dietary self-management using mobile vision-based assistance," in *New Trends in Image Analysis and Processing—ICIAP 2017*. Cham, Switzerland: Springer, 2017, pp. 385–393.

[22] Y. Matsuda, H. Hoashi, and K. Yanai, "Recognition of multiple-food images by detecting candidate regions," in *Proc. IEEE Int. Conf. Multimedia Expo*, Jul. 2012, pp. 25–30.

[23] 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.

[24] 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)*, 2009, pp. 289–292.

[25] J. Qiu, F. P.-W. Lo, Y. Sun, S. Wang, and B. Lo, "Mining discriminative food regions for accurate food recognition," 2022, *arXiv:2207.03692*.

[26] G. M. Farinella, D. Allegra, and F. Stanco, "A benchmark dataset to study the representation of food images," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2014, pp. 584–599.

[27] P. Pouladzadeh, A. Yassine, and S. Shirmohammadi, "FooDD: Food detection dataset for calorie measurement using food images," in *Proc. New Trends Image Anal. Process. ICIAP Workshops*. Cham, Switzerland: Springer, 2015, pp. 441–448.

[28] Y. Liang and J. Li, "Computer vision-based food calorie estimation: Dataset, method, and experiment," 2017, *arXiv:1705.07632*.

[29] Y. Kawano and K. Yanai, "Real-time mobile food recognition system," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2013, pp. 1–7.

[30] H. He, F. Kong, and J. Tan, "DietCam: Multiview food recognition using a multikernel SVM," *IEEE J. Biomed. Health Informat.*, vol. 20, no. 3, pp. 848–855, May 2016.

[31] G. Ciocca, P. Napoletano, and R. Schettini, "Food recognition: A new dataset, experiments, and results," *IEEE J. Biomed. Health Informat.*, vol. 21, no. 3, pp. 588–598, May 2017.

[32] E. Aguilar, M. Bolaños, and P. Radeva, "Food recognition using fusion of classifiers based on CNNs," in *Proc. 19th Int. Conf. Image Anal. Process. (ICIAP)*, Catania, Italy. Cham, Switzerland: Springer, Sep. 2017, pp. 213–224.

[33] G. A. Tahir and C. K. Loo, "Explainable deep learning ensemble for food image analysis on edge devices," *Comput. Biol. Med.*, vol. 139, Dec. 2021, Art. no. 104972.

[34] D. Sahoo, W. Hao, S. Ke, W. Xiongwei, H. Le, P. Achananuparp, E.-P. Lim, and S. C. H. Hoi, "FoodAI: Food image recognition via deep learning for smart food logging," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2019, pp. 2260–2268.

[35] N. Martinel, G. L. Foresti, and C. Micheloni, "Wide-slice residual networks for food recognition," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2018, pp. 567–576.

[36] R. Abiyev and J. Adepoju, "Automatic food recognition using deep convolutional neural networks with self-attention mechanism," *Human-Centric Intell. Syst.*, vol. 4, no. 1, pp. 171–186, Jan. 2024.

[37] J. Dehais, M. Anthimopoulos, S. Shevchik, and S. Mougiakakou, "Two-view 3D reconstruction for food volume estimation," *IEEE Trans. Multimedia*, vol. 19, no. 5, pp. 1090–1099, May 2017.

[38] J. Gao, W. Tan, L. Ma, Y. Wang, and W. Tang, "MUSEFood: Multi-sensor-based food volume estimation on smartphones," in *Proc. IEEE SmartWorld, Ubiquitous Intell. Comput., Adv. Trusted Comput., Scalable Comput. Commun., Cloud Big Data Comput., Internet People Smart City Innov. (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*, Aug. 2019, pp. 899–906.

[39] Y. Yang, W. Jia, T. Bucher, H. Zhang, and M. Sun, "Image-based food portion size estimation using a smartphone without a fiducial marker," *Public Health Nutrition*, vol. 22, no. 7, pp. 1180–1192, 2019.

[40] G. Vinod, J. He, Z. Shao, and F. Zhu, "Food portion estimation via 3D object scaling," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2024, pp. 3741–3749.

[41] L. N. Smith and N. Topin, "Super-convergence: Very fast training of neural networks using large learning rates," in *Proc. SPIE*, vol. 11006, 2019, pp. 369–386.

[42] N. Qian, "On the momentum term in gradient descent learning algorithms," *Neural Netw.*, vol. 12, no. 1, pp. 145–151, Jan. 1999.

[43] Y. Nesterov, "A method for unconstrained convex minimization problem with the rate of convergence O (1/k2)," *Dokl. Akad. Nauk. SSSR*, vol. 269, no. 3, 1983, p. 543.

[44] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *J. Mach. Learn. Res.*, vol. 12, no. 7, pp. 2121–2159, 2011.

[45] G. Hinton, N. Srivastava, and K. Swersky, "Neural networks for machine learning lecture 6a overview of mini-batch gradient descent," 2012. [Online]. Available: https://www.cs.toronto.edu/ hinton/coursera/lecture6/lec6.pdf

[46] M. D. Zeiler, "ADADELTA: An adaptive learning rate method," 2012, *arXiv:1212.5701*.

[47] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

[48] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2261–2269.

[49] M. Tan and Q. V. Le, "EfficientNetv2: Smaller models and faster training," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 10096–10106.

[50] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-v4, Inception-ResNet and the impact of residual connections on learning," in *Proc. AAAI Conf. Artif. Intell.*, 2017, vol. 31, no. 1, pp. 1–16.

[51] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the Inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2818–2826.

[52] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient convolutional neural networks for mobile vision applications," 2017, *arXiv:1704.04861*.

[53] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4510–4520.

[54] A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, and V. Vasudevan, "Searching for MobileNetV3," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 1314–1324.

[55] D. Qin, C. Leichner, M. Delakis, M. Fornoni, S. Luo, F. Yang, W. Wang, C. Banbury, C. Ye, B. Akin, V. Aggarwal, T. Zhu, D. Moro, and A. Howard, "MobileNetV4-universal models for the mobile ecosystem," 2024, *arXiv:2404.10518*.

[56] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, "Learning transferable architectures for scalable image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 8697–8710.

[57] I. Radosavovic, R. P. Kosaraju, R. Girshick, K. He, and P. Dollár, "Designing network design spaces," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 10425–10433.

[58] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.

[59] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *Proc. Eur. Conf. Comput. Vis.*, Amsterdam, The Netherlands. Cham, Switzerland: Springer, 2016, pp. 630–645.

[60] Z. Liu, H. Hu, Y. Lin, Z. Yao, Z. Xie, Y. Wei, J. Ning, Y. Cao, Z. Zhang, and L. Dong, "Swin transformer v2: Scaling up capacity and resolution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 12009–12019.

[61] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1800–1807.

[62] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.

[63] L. N. Smith, "A disciplined approach to neural network hyper-parameters: Part 1–learning rate, batch size, momentum, and weight decay," 2018, *arXiv:1803.09820*.

[64] B. Kenstler. (2024). *Cyclical Learning Rate (CLR)*. Accessed: May 4, 2024. [Online]. Available: https://github.com/bckenstler/CLR

[65] 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.

[66] A. Meyers, N. Johnston, V. Rathod, A. Korattikara, A. Gorban, N. Silberman, S. Guadarrama, G. Papandreou, J. Huang, and K. P. Murphy, "Im2Calories: Towards an automated mobile vision food diary," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 1233–1241.

[67] C. Liu, Y. Cao, Y. Luo, G. Chen, V. Vokkarane, and Y. Ma, "DeepFood: Deep learning-based food image recognition for computer-aided dietary assessment," in *Proc. Int. Conf. Smart Homes Health Telematics*, Wuhan, China. Cham, Switzerland: Springer, 2016, pp. 37–48.

[68] Y. Cui, F. Zhou, J. Wang, X. Liu, Y. Lin, and S. Belongie, "Kernel pooling for convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2921–2930.

[69] F. Ofli, Y. Aytar, I. Weber, R. A. Hammouri, and A. Torralba, "Is saki #delicious: The food perception gap on Instagram and its relation to health," in *Proc. 26th Int. Conf. World Wide Web*, Apr. 2017, pp. 509–518.

[70] Y. Cui, Y. Song, C. Sun, A. Howard, and S. Belongie, "Large scale fine-grained categorization and domain-specific transfer learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4109–4118.

[71] K.-H. Lee, X. He, L. Zhang, and L. Yang, "CleanNet: Transfer learning for scalable image classifier training with label noise," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 5447–5456.

[72] M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 6105–6114.

[73] M. Tan and Q. V. Le, "MixConv: Mixed depthwise convolutional kernels," 2019, *arXiv:1907.09595*.

[74] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. H. Richemond, E. Buchatskaya, C. Doersch, B. A. Pires, Z. D. Guo, M. G. Azar, B. Piot, K. Kavukcuoglu, R. Munos, and M. Valko, "Bootstrap your own latent-a new approach to self-supervised learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 21271–21284.

[75] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 1597–1607.

[76] D. Dwibedi, Y. Aytar, J. Tompson, P. Sermanet, and A. Zisserman, "With a little help from my friends: Nearest-neighbor contrastive learning of visual representations," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 9588–9597.

[77] Y. Zhang, S. Zheng, P. Wu, M. Goswami, and C. Chen, "Learning with feature-dependent label noise: A progressive approach," 2021, *arXiv:2103.07756*.

[78] Y. Li, H. Han, S. Shan, and X. Chen, "DISC: Learning from noisy labels via dynamic instance-specific selection and correction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 24070–24079.

[79] G. Sheng, W. Min, T. Yao, J. Song, Y. Yang, L. Wang, and S. Jiang, "Lightweight food image recognition with global shuffle convolution," *IEEE Trans. AgriFood Electron.*, vol. 2, no. 2, pp. 392–402, Sep. 2024.

[80] Y. Kawano and K. Yanai, "FoodCam-256: A large-scale real-time mobile food RecognitionSystem employing high-dimensional features and compression of classifier weights," in *Proc. 22nd ACM Int. Conf. Multimedia*, Nov. 2014, pp. 761–762.

[81] 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.*, Oct. 2016, pp. 41–49.

[82] M. Bolaños and P. Radeva, "Simultaneous food localization and recognition," in *Proc. 23rd Int. Conf. Pattern Recognit. (ICPR)*, Dec. 2016, pp. 3140–3145.
[83] G. Ciocca, P. Napoletano, and R. Schettini, "CNN-based features for retrieval and classification of food images," *Comput. Vis. Image Understand.*, vols. 176–177, pp. 70–77, Nov. 2018.
[84] G. A. Tahir and C. K. Loo, "An open-ended continual learning for food recognition using class incremental extreme learning machines," *IEEE Access*, vol. 8, pp. 82328–82346, 2020.
[85] R. Mao, J. He, Z. Shao, S. K. Yarlagadda, and F. Zhu, "Visual aware hierarchy based food recognition," in *Proc. Int. Conf. Pattern Recognit.* Cham, Switzerland: Springer, 2021, pp. 571–598.
[86] Z. Xiao, G. Diao, and Z. Deng, "Fine grained food image recognition based on Swin transformer," *J. Food Eng.*, vol. 380, Nov. 2024, Art. no. 112134.
[87] N. Theera-Ampornpunt and P. Treepong, "Optimizing hyperparameters for Thai cuisine recognition via convolutional neural networks," *Traitement du Signal*, vol. 40, no. 3, pp. 1187–1193, Jun. 2023.

NAWANOL THEERA-AMPORNPUNT received the B.Sc. degree in computer science from Carnegie Mellon University, USA, in 2009, and the Ph.D. degree in computer science from Purdue University, USA, in 2017. He is currently an Assistant Professor with the College of Computing, Prince of Songkla University, Phuket, Thailand. His research interests include object recognition, image classification, and artificial neural network optimization.

PANISA TREEPONG received the B.Sc. degree in information technology from the Prince of Songkla University, Phuket, Thailand, in 2007, the M.Sc. degree in bioinformatics from the King Mongkut's University of Technology Thonburi, Thailand, in 2010, and the Ph.D. degree in computer science from Université Bourgogne Franche-Comté, France, in 2017. She is currently an Assistant Professor with the College of Computing, Prince of Songkla University. Her research interests include bioinformatics and machine learning.