information

Article

Indigenous Food Recognition Model Based on Various Convolutional Neural Network Architectures for Gastronomic Tourism Business Analytics

Mohd Norhisham Razali [1], Ervin Gubin Moung [2,*], Farashazillah Yahya [2], Chong Joon Hou [2], Rozita Hanapi [1], Raihani Mohamed [3] and Ibrahim Abakr Targio Hashem [4]

1 Faculty of Business Management, Universiti Teknologi Mara Cawangan Sarawak, Kota Samarahan 94350, Sarawak, Malaysia; hishamrazali@uitm.edu.my (M.N.R.); rozita8282@uitm.edu.my (R.H.)
2 Faculty of Computing and Informatics, Universiti Malaysia Sabah, Kota Kinabalu 88400, Sabah, Malaysia; fara.yahya@ums.edu.my (F.Y.); joonhou1995@gmail.com (C.J.H.)
3 Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, Serdang 43400, Selangor Darul Ehsan, Malaysia; raihanimohamed@upm.edu.my
4 Department of Computer Science, College of Computing and Informatics, University of Sharjah, Sharjah 27272, United Arab Emirates; ihashem@sharjah.ac.ae
* Correspondence: ervin@ums.edu.my

Abstract: In gastronomic tourism, food is viewed as the central tourist attraction. Specifically, indigenous food is known to represent the expression of local culture and identity. To promote gastronomic tourism, it is critical to have a model for the food business analytics system. This research undertakes an empirical evaluation of recent transfer learning models for deep learning feature extraction for a food recognition model. The VIREO-Food172 Dataset and a newly established Sabah Food Dataset are used to evaluate the food recognition model. Afterwards, the model is implemented into a web application system as an attempt to automate food recognition. In this model, a fully connected layer with 11 and 10 Softmax neurons is used as the classifier for food categories in both datasets. Six pre-trained Convolutional Neural Network (CNN) models are evaluated as the feature extractors to extract essential features from food images. From the evaluation, the research found that the EfficientNet feature extractor-based and CNN classifier achieved the highest classification accuracy of 94.01% on the Sabah Food Dataset and 86.57% on VIREO-Food172 Dataset. EFFNet as a feature representation outperformed Xception in terms of overall performance. However, Xception can be considered despite some accuracy performance drawback if computational speed and memory space usage are more important than performance.

Keywords: food recognition; deep learning; transfer learning; CNN; food sentiment; food features; gastronomic tourism

check for updates

Citation: Razali, M.N.; Moung, E.G.; Yahya, F.; Hou, C.J.; Hanapi, R.; Mohamed, R.; Hashem, I.A.T. Indigenous Food Recognition Model Based on Various Convolutional Neural Network Architectures for Gastronomic Tourism Business Analytics. *Information* **2021**, *12*, 322. https://doi.org/10.3390/info12080322

Academic Editor: Gholamreza Anbarjafari (Shahab)

Received: 23 June 2021
Accepted: 8 August 2021
Published: 11 August 2021

1. Introduction

Food and beverage expenditures are estimated to account for roughly a quarter of total tourism spending worldwide. As food and tourism are inextricably linked, gastronomic tourism, in which the local cuisine serves as the primary attraction for travelers, has gained popularity in recent years [1]. Local foods can contribute to the development of a local brand, which encourages tourism growth in several countries [2]. Sabah, one of Malaysia's states, is a well-known tourist destination for its magnificent scenery, contributing significantly to its economy. Sabah's diversity of indigenous groups and subgroups is notable for its unique traditions, cultures, practices, and traditional local foods. According to [3], it is highly likely that acceptance of local food brands among tourists and Sabah residents is critical to preserving the culinary heritage and providing visitors with a sense of uniqueness, and special, memorable experiences. Besides the preservation and appreciation, local

foods also require more innovative strategies to make them more appealing, particularly to younger generations, and survive in a competitive business environment [4,5]. Thus, this research proposes a deep learning-based food recognition model for the Sabah food business analytics system to promote gastronomic tourism in Sabah. Subsequently, the trained model is deployed to a web application system to demonstrate the end-user functionality to automatically recognize the specific name of foods based on real-time food images.

The recent emergence of deep-learning techniques, particularly in food recognition studies [6–9], has motivated the investigation of deep-learning approaches, particularly in dealing with the local context of this research. Despite extensive research in deep learning to support food recognition tasks [10], the analysis can be extended further, particularly when dealing with the novel or local context encountered in this research. Currently, there is a lack of research investigating the effect of recent deep-learning models on food recognition performance, particularly on the feature extraction aspect of food recognition studies. Additionally, food recognition remains a difficult task, due to the foods' complex appearance, which include a range of shapes, sizes, and colors, as well as their reliance on the foods' local context or origins [6]. The following summarizes this paper's contributions:

- An empirical analysis was conducted to investigate the effect of deep-learning techniques on food recognition performance, using transfer-learning approaches as feature extractors on the Sabah Food Dataset and the VIREO-Food172 Dataset.
- A Sabah Food Dataset was created, which contains 11 different categories of popular Sabah foods. It was used to train the machine-learning model for the classification of Sabah foods.
- A preliminary prototype of a web-based application for a food recognition model is presented.

The following sections outline the structure of this paper. Section 2 discusses the related works of deep learning in food recognition, and Section 3 discusses the theoretical background of transfer learning through the use of a pre-trained deep-learning architecture. Subsequently, Section 4 explains the details of the experiment's procedure that was conducted. Then, in Section 5, the results of the experiments and the deployment of the food recognition model are discussed. Finally, Section 6 discusses the overall conclusion of the work and future works.

2. Related Works

Machine learning is used as a data processing technique to solve a wide range of problems in a variety of fields, including smart homes [11], human identification in healthcare [12], face recognition [13–15], water quality research [16], and many more. In traditional machine learning, tedious and exhaustive feature extraction is a very common practice in order to produce a highly discriminative feature. However, due to computational and storage capability advancements, a more profound representation of features based on deep learning has become a common practice for better performance for classification and regression. A deep Artificial Neural Network (ANN) composed of various layers with multilevel feature learning defines the general concept of deep learning. Specifically, a set of components comprising pooling, convolutional, and fully connected layers dubbed as the Convolutional Neural Network (CNN) has gained popularity as a pattern-recognition technique, including in studies involving food recognition. This is due to the fact that the recognition capability is exceptional, even with simple CNN configurations. For instance, Lu [17] demonstrated four layers of hidden neurons to classify ten categories of a small-scale food images dataset. The RGB component of the image was reshaped into a two-dimensional form as input data. First, a convolutional layer with a 7 by 7 dimension and a stride value of one was used to extract 32 feature maps. Secondly, a 5 by 5 size of convolutional layers was used to extract 64 feature maps. Lastly, a total of 128 feature maps were generated from 3 by 3 convolutional layers. The best accuracy on the test set reported was 74%. However, over-fitting is suspected as a result of the limited size of the training data, which limits the accuracy of the testing dataset at a higher epoch.

A study conducted by [18] implemented CNN to recognize 11 categories of self-collected Malaysian foods. The architecture of VGG19-CNN was modified by adding more layers consisting of 21 convolutional layers and three fully connected layers as compared to 16 convolutional layers in VGG19. However, the performance results were not reported. Islam et al. [19] evaluated their proposed CNN configuration and Inception V3 model on the Food-11 dataset for their food recognition module. The images were reshaped into 224 by 224 by 3 dimensions, and ZCA whitening was applied to eliminate unnecessary noise within the images. The accuracy reported for the proposed CNN configuration and pre-trained Inception V3 model was 74.7% and 92.86%, respectively.

The hyper-parameter configurations in conventional CNN are complicated and time-consuming. Jeny et al. [20] proposed another method for managing the massive number of layers by implementing a FoNet-based Deep Residual Neural Network and testing it on six categories of Bangladesh foods. The model comprises 47 layers that contained pooling layers, activation functions, flattened layers, dropout and normalization. The reported accuracy of 98.16% on their testing dataset outperformed the Inception V3 and MobileNet models, which reported an accuracy of 95.8% and 94.5%, respectively.

In summary, previous research has demonstrated that CNN and transfer learning-based techniques are effective at food image recognition. However, there is a lack of analysis and evaluation of recent CNN architecture models, particularly in terms of feature extraction. Furthermore, CNNs have hyperparameters that must be tuned to the newly created dataset. Table 1 summarizes the related works on CNN models.

Table 1. A summary of related works on CNN models for food recognition.

Authors	Dataset	Number of Categories	Techniques	Results
Lu (2016) [17]	Small-scale dataset	10	A proposed CNN configuration model with 3 convolution-pooling layers and 1 fully connected layer.	Test set accuracy of 74%
Subhi and Ali (2018) [18]	Self-collected Malaysian foods dataset	11	Modified VGG19-CNN model with 21 convolutional layers and 3 fully connected layers.	Not reported
Islam et al. (2018) [19]	Food-11 dataset	11	(i) A proposed CNN configuration model with 5 convolution layers, 3 max-pooling layers and 1 fully connected layer. (ii) Inception V3 pre-trained model with 2 fully connected layers.	(i) Proposed approach achieved 74.7% accuracy. (ii) Pre-trained Inception V3 achieved 92.86% accuracy.
Jeny et al. (2019) [20]	Self-collected Bangladesh foods dataset	6	FoNet-based Deep Residual Neural Network with 47 layers comprises of pooling layers, activation functions, flattened layers, and dropout and normalization.	Testing set accuracy of 98.16%.

3. A Transfer Learning Approach Using Pre-Trained Deep Learning Architecture

This section discusses the theoretical background of the approaches that have been considered for feature extraction. The approaches to feature extraction include ResNet50, VGG16, MobileNet, Xception, Inception, and EfficientNet. Additionally, the RGB component of an image is used to represent the features.

3.1. ResNet50

The ResNet50 approach was introduced in the 2015 ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [21]. This model is a residual learning framework that can alleviate the vanishing gradient problem of Deep Convolutional Neural Networks during deeper networks' training. The ResNet50 model was pre-trained on over a million high-resolution images from the ImageNet database. Zahisham et al. (2016) [22] proposed a ResNet50-based Deep Convolutional Neural Network (DCNN) for the food recognition task. The ResNet50 model architecture is imitated; pre-trained weights are imported; and classification layers are trained on three different food datasets (UECFOOD100, ETHZ-FOOD101, and UECFOOD256). The rank one accuracy achieved for the proposed DCNN-ResNet50 model was 39.75%, 41.08%, and 35.32% on the UECFOOD100, ETHZ-FOOD10, and UECFOOD256 datasets, respectively. This proposed model outperformed the accuracy of CNN-3CV (25%, 24.3% and 22%), CNN + Support Vector Machine (SVM) (33.1%, 31.9% and 30%) and CNN-5CV (20%, 17.9% and 15.5%).

3.2. VGG16

The VGG-16 approach was introduced by Simonyan and Zisserman [23] at the 2014 ILSVRC conference and was developed by the University of Oxford's Visual Graphics Group. This model is widely used in image classification tasks, as it can outperform the AlexNet-based model. The VGG-16 is trained on the ImageNet dataset with over fifteen million high-resolution images and 22,000 image classes. A comparison of CNN tolerance to the intraclass variety in food recognition was conducted by [24]. The feature extraction process was carried out using a variety of pre-trained CNN models, including ResNet, VGG16, VGG19, MobileNet, and InceptionV3. Additionally, the Food101 dataset was used to evaluate their performance. It was reported that InceptionV3 obtained the highest Top-1 accuracy of 87.16%, followed by VGG16 with a Top-1 accuracy of 84.48, when using 70% as the training set and 30% as the testing set.

3.3. MobileNet

Howard et al. [25] proposed MobileNet, a low-latency, low-computation model for on-device and embedded applications. Its architecture is based on depthwise separable convolution that significantly reduces computation and model size while maintaining classification performance similar to that of large-scale models, such as Inception. The ImageNet database was used in their experiment, and it was reported that the MobileNet achieved an accuracy of 70.6%, which is comparable to GoogLeNet (69.8%) and VGG-16 (71.5%) while requiring approximately ten times the computational resources required by GoogLeNet and VGG-16. Additionally, on the Stanford Dogs dataset, the MobileNet model achieved an accuracy of 83.3% for fine-grained recognition, which is nearly identical to the 84% accuracy of a large-scale Inception model, with ten times the computation and a twentyfold reduction in the parameter count. Following that, the paper in [7] implemented FD-MobileNet-TF-YOLO as an embedded food recognizer. FD-MobileNet was used as a food categorizer, while TF-YOLO was used as an ingredient locator and classifier. The FD-MobileNet approach achieved higher downsampling efficiency by conducting 32 downsamples within 12 levels on an image of 224 by 224 dimensions, resulting in reduced computational complexity and costs. The TF-YOLO approach identified smaller objects in images, using the YOLOv3-tiny procedure based on the K-means technique. The recognition accuracy of FD-MobileNet was 94.67%, which is higher than MobileNet's recognition accuracy of 92.83%.

3.4. Xception

Chollet [26] introduced the Xception model, a modified depthwise separable convolution model based on the Inception model. The Xception model outperforms the Inception model because it reduces the number of model parameters and makes more efficient use of them, allowing for the learning of richer representations with fewer parameters. On

the ImageNet dataset, the Xception model achieved the highest Rank-1 accuracy of 79%, followed by Inception V3 at 78.2%, ResNet-152 at 77%, and VGG-16 at 71.5%. Additionally, the Xception model outperforms Inception V3 in terms of mean accuracy prediction (mAP) when evaluated against the FastEval14k dataset, containing 14,000 images classified into 6000 classes. In another report, Yao et al. [27] conducted a study on the classification of peach disease using the traditional Xception and the proposed improved Xception model. The proposed improved Xception network was based on ensembles of regularization terms of the L2-norm and mean. An experiment was conducted using a peach disease image dataset comprised of seven different disease categories and seven commonly used deep-learning models. It was reported that the validation accuracy for Xception and the improved Xception was 92.23% and 93.85%, respectively.

3.5. Inception

Inception is a deep neural network architecture for computer vision, introduced by [28] at the 2014 ILSVRC conference. The Inception architecture uses a sparse structure of a convolutional network with one-by-one convolution dimensions to reduce dimensionality. GoogLeNet is a deep-learning model that uses the Inception architecture, which comprises nine modules. Inception modules employ a total of 22 layers and five pooling layers. Singla et al. [29] demonstrated the feasibility of the Inception network—GoogLenet—for food category recognition. They reported that the food identification module achieved an accuracy of 83.6% when tested against the Food-11 dataset.

3.6. EfficientNet

Tan and Le [30] proposed the EfficientNet (EFFNet) model, which utilizes a simple and effective compound coefficient to scale up CNN structurally. In comparison to conventional neural network approaches, EFFNet uses a fixed set of scaling coefficients to scale each dimension of depth, width, and resolution uniformly. The EFFNet baseline network was built with the AutoML Mobile Neural Architecture Search (MNAS) framework to optimize accuracy and efficiency, while the remaining architecture was built with mobile inverted bottleneck convolution (MBConv). The performance of EFFNet on ImageNet is compared to that of conventional CNNs, and the findings shows that EFFNet models outperform conventional CNN models in both accuracy and efficiency. For instance, the EfficientNet-B0 model achieved a Rank-1 and Rank-5 accuracy of 77.1% and 93.3%, higher than ResNet-50's Rank-1 (76%) and Rank-5 (93.3%) accuracy. Liu et al. [31] implemented a transfer learning-based EFFNet model to recognize and classify maize leaf disease images. For their experiments, a larger leaf dataset containing 9279 images classified into eight disease categories was divided into a 7:3 training to testing set ratio. The reported recognition accuracy of their proposed model (98.52%) outperformed VGG-16's accuracy of 93.9%, Inception V3's accuracy of 96.35%, and ResNet-50's accuracy of 96.76%.

4. Experiments

4.1. Food Dataset Preparation

For the classification performance evaluation, two types of datasets are used: (i) the Sabah Food Dataset and (ii) the VIREO-Food172 Dataset. Figures 1 and 2 illustrate food image samples from the Sabah Food Dataset and VIREO-Food172 Dataset. These images are real-world food images that are diverse in terms of quality and image background. Most of the images have a cluttered and non-uniform background.

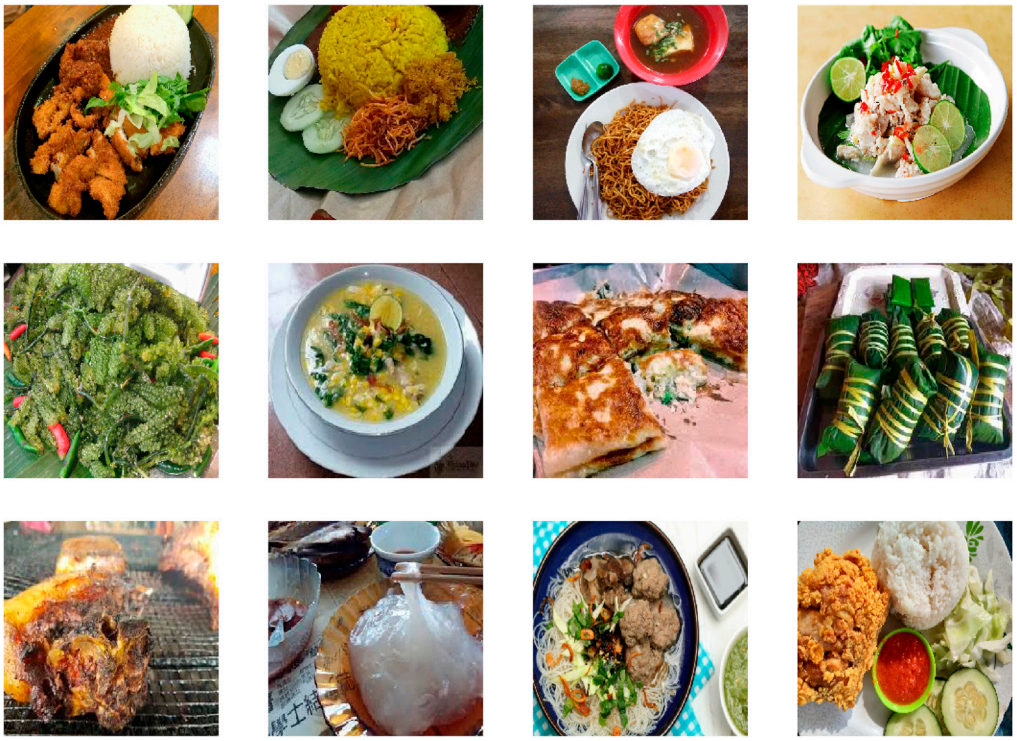


Figure 1. Sabah Food Dataset samples.

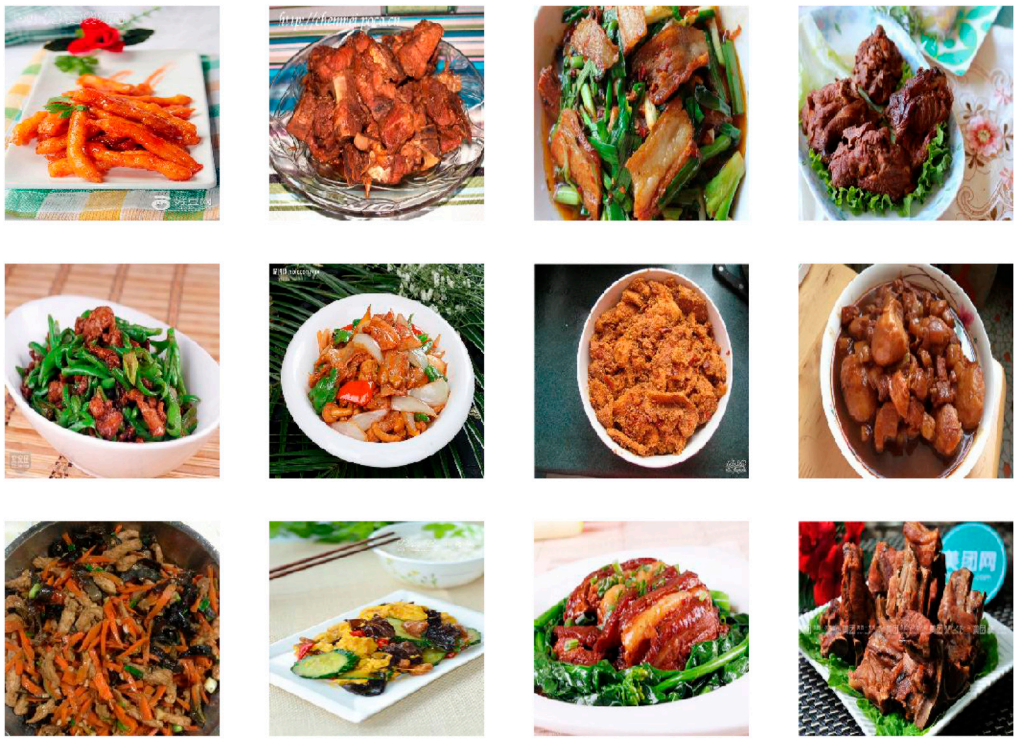


Figure 2. VIREO-Food172 Dataset samples.

The Sabah Food Dataset is a newly created food dataset that was used in this study. The images in Sabah Food Dataset were gathered via Google image search and include a range of image resolutions and compression formats. A total of 1926 food images were collected for the Sabah Food Dataset, which includes 11 different famous food categories. The details for each food category of the Sabah Food Dataset are presented in Table 2.

The purpose of this dataset is to train a machine-learning classifier for the purpose of developing a Sabah food recognition model.

Table 2. Image number for each category of food items for Sabah Food Dataset.

Category Label	Category Name	Number of Images
1	Bakso	85
2	Sinalau Bakas	219
3	Ambuyat	242
4	Barobbo	83
5	Buras	198
6	Martabak Jawa	92
7	Nasi Kuning	245
8	Mee Tauhu	145
9	Hinava	164
10	Latok	236
11	Nasi lalap	217

The VIREO-Food172 Dataset [32] samples, as shown in Figure 2, are popular Chinese dishes retrieved from Google and Baidu image searches. Based on the recipes, the images were labeled with category names as well as over 300 ingredients. This dataset comprises 172 food categories from eight major groups, including (i) soup, (ii) vegetables, (iii) bean products, (iv) egg, (v) meat, (vi) fish, (vii) seafood, and (viii) staple. However, only ten categories (categories 1 to 10) of the food images were used in this experiment. The details for each food category of the VIREO-Food172 Dataset are presented in Table 3. For performance evaluation, a total of 9015 food images were selected from the VIREO-Food172 Dataset's ten categories. The test will be more challenging, due to the low interclass differences among those ten categories, most of which are pork-based. This will serve to further validate the system's capability for accurate classification.

Table 3. Image number for each category of food items for VIREO-Food-172 dataset.

Category Label	Category Name	Number of Images
1	Braised pork	1023
2	Sautéed spicy pork	987
3	Crispy sweet and sour pork slices	991
4	Steamed pork with rice powder	803
5	Pork with salted vegetable	997
6	Shredded pork with pepper	708
7	Yu-Shiang shredded pork	1010
8	Eggs, black fungus, and sautéed sliced pork	830
9	Braised spare ribs in brown sauce	712
10	Fried sweet and sour tenderloin	954

As for data training and testing preparation, 80% and 20% of the datasets (Sabah Food Dataset and VIREO-Food172 Dataset) are prepared for the training and testing dataset, respectively. For the Sabah Food Dataset, the distribution of the training and testing datasets is selected randomly, using the Python random sampling function. Additionally, the images in the training and testing datasets are not identical. The datasets are available upon request from the corresponding author for reproducibility purposes. For the VIREO-Food172 Dataset, the 80% (training dataset) and 20% (testing dataset) distribution was provided by the original source of the database.

4.2. Feature Representations and Classifiers

In order to conduct a more thorough evaluation, the efficiency of the feature representation based on the transfer learning approaches described in Section 3 is compared. The six pre-trained CNN models selected as the feature extractor are (i) ResNet50, (ii) VGG16,

(iii) MobileNet, (iv) Xception, (v) Inception V3, and (vi) EfficientNet. In addition to the list, (vii) the RGB component of an image is also used as the feature representation.

Furthermore, ten different classifiers are considered for this paper. They are (i) non-linear SVM (one-versus-one mode), denoted as SVM (OVO), (ii) non-linear SVM (one-versus-all mode), denoted as SVM (OVA), (iii) Linear SVM (one-versus-one mode), denoted as LSVM (OVO), (iv) LSVM (one-versus-all mode), denoted as LSVM (OVA), (v) Decision Tree (DT), (vi) Naïve Bayes (NB), (vii) Artificial Neural Network (ANN), (viii) Random Forest (RF), (ix) k-Nearest Neighbor (kNN), and (x) CNN.

The proposed approach in this paper is labeled as "Feature Representation + Classifier". For instance, an approach labeled ResNet50 + SVM (OVO) implies the use of ResNet50 as a feature representation and SVM (OVO) as a classifier. Table 4 shows the CNN feature extractor's configuration details. The following are the definitions of the parameters shown in Table 4:

1. The Model denotes a convolutional base of existing pre-trained CNN models as a feature extractor.
2. The No.of.param denotes the total number of model parameters from the input layer to the final convolutional layer.
3. The Input Shape (x, y, z) denotes input image data with a three-dimensional shape. The x represents the height of an image; the y represents the image's width; and the z represents the depth of an image.
4. The *Output Shape* (x, y, z) denotes the output data shape produced from the last convolutional layer. The x represents the height of an image; the y represents the image's width; and the z represents the depth of an image.
5. The *Vector size* denotes an output shape that is flattened into a one-dimensional linear vector.

Table 4. The architecture of pre-trained CNN-based models.

Model	No. of. Param	Input Shape (x, y, z)	Output Shape (Conv2D) (x, y, z)	Vector Size (Conv1D)
ResNet50	25,636,712	(224, 224, 3)	(32, 32, 2)	(1, 2048)
VGG16	138,357,544	(224, 224, 3)	(64, 64, 1)	(1, 4096)
MobileNet	3,228,864	(64, 64, 3)	(128, 128, 2)	(1, 32,768)
Xception	22,910,480	(299, 299, 3)	(32, 32, 2)	(1, 2048)
Inception V3	21,802,784	(299, 299, 3)	(128, 128, 3)	(1, 49,152)
EFFNet	5,330,564	(224, 224, 3)	(16, 16, 245)	(1, 62,720)

The images are resized to fit the fixed input form of the pre-trained CNN model. Numerous hyperparameters are included in pre-trained CNN models, and as shown in the second column of Table 4, EFFNet and VGG16 generate the most and fewest parameters, respectively. The Output Shape (Conv2D) and Vector Size (Conv1D) of the final CNN layer, which serves as the feature representation, are manually reshaped into a one-dimensional vector before being fed into a machine-learning classifier. The Conv2D generates the spatial features necessary for the detection of edges and colors. Both Input Shape and Output Shape represent the height, width, depth of the image. The Conv2D features are fed into the sequential model for classification.

The summary of the CNN architecture used for the data training phase is shown in Table 5. The following are the definitions of the parameters shown in Table 5:

1. The Layer denotes the layer name.
2. The Type denotes the type of layer.
3. The Output denotes feature maps generated from the layer.
4. The number of parameters of a layer is denoted as No.of.param.
5. The conv2d_1, conv2d_2, and conv2d_3 denotes the convolutional layer of 1, 2, 3.
6. The max_pooling2d_1 and max_pooling2d_2 denotes the max-pooling layer of 1 and 2.

7. The dropout_1, dropout_2, dropout_3, and dropout_4 denotes the dropout layer of 1, 2, 3, 4.
8. The flatten_1 denotes the flatten layer.
9. The dense_1, dense_2, and dense_3 denotes the dense layer 1, 2, 3.

Table 5. The layers configuration of the CNN classifier.

Layer	Type	Output	No.of.Param
conv2d_1	(Conv2D)	(None, 64, 64)	500
conv2d_2	(Conv2D)	(None, 64, 64)	33,825
max_pooling2d_1	(MaxPooling2	(None, 32, 32)	0
dropout_1	(Dropout)	(None, 32, 32)	0
conv2d_3	(Conv2D)	(None, 32, 32)	84,500
max_pooling2d_2	(MaxPooling2	(None, 16, 16)	0
dropout_2	(Dropout)	(None, 16, 16)	0
flatten_1	(Flatten)	(None, 32,000)	0
dense_1	(Dense)	(None, 500)	16,000,500
dropout_3	(Dropout)	(None, 500)	0
dense_2	(Dense)	(None, 250)	125,250
dropout_4	(Dropout)	(None, 250)	0
dense_3	(Dense)	(None, 12)	3012
		Total parameters:	16,247,587
		Trainable parameters:	16,247,587
		Non-trainable parameters	0

The layers of the CNN classifier shown in Table 5 is a network that comprises three layers of neurons: two convolutional-pooling layers and one fully connected layer. The input is based on two parameters: (i) the *output shape* of the features generated by the pre-trained CNN model, referred from Table 4, and (ii) the color features of a two-dimensional, reshaped 64 by 64 image, where the color features are composed of an image's RGB component. The first convolutional-pooling layer has the kernel dimensions of 3 by 3 to extract 32 feature maps. Subsequently, a max-pooling layer is added in a 2 by 2 dimension region. The fully connected layer has 512 rectified linear unit neurons with 11 and 10 Softmax neurons that indicate the 11 Sabah Food Dataset categories and the 10 VIREO-Food172 Dataset categories. In this paper, the Keras deep learning packages are used to train the CNN model [2,33].

On the other hand, the *Conv1D* features are represented in a vector and feed to ten machine-learning classifiers, including (i) non-linear SVM (OVO), (ii) non-linear SVM (OVA), (iii) LSVM (OVO), (iv) LSVM (OVA), (v) DT, (vi) NB, (vii) ANN, (viii) RF, (ix) kNN, and (x) CNN. The parameters for each classifier used in this work are presented in Tables 6–15.

Table 6. SVM (OVO) parameters from scikit-learn library.

Parameters	Value	Description
C	1.0	Regularization parameter.
kernel	rbf	Specifies the kernel type to be used in the algorithm.
degree	3	Degree of the polynomial kernel function ('poly').
gamma	scale	Kernel coefficient for 'rbf', 'poly' and 'sigmoid'.
coef0	0.0	Independent term in kernel function.
decision_function_shape	ovo	Multi-class strategy.

Table 7. SVM (OVA) parameters from scikit-learn library.

Parameters	Value	Description
C	1.0	Regularization parameter.
kernel	rbf	Specifies the kernel type to be used in the algorithm.
degree	3	Degree of the polynomial kernel function ('poly').
gamma	scale	Kernel coefficient for 'rbf', 'poly' and 'sigmoid'.
coef0	0.0	Independent term in kernel function.
decision_function_shape	ovr	Multi-class strategy.

Table 8. LSVM (OVO) parameters from scikit-learn library.

Parameters	Value	Description
penalty	l2	Specifies the norm used in the penalization. The 'l1' leads to coef_ vectors that are sparse.
loss	square_hinge	Specifies the loss function. 'hinge' is the standard SVM loss (used e.g., by the SVC class) while 'squared_hinge' is the square of the hinge loss.
dual	True	Select the algorithm to either solve the dual or primal optimization problem.
tol	0.0001	Tolerance for stopping criteria.
C	1.0	Regularization parameter. The strength of the regularization is inversely proportional to C. Must be strictly positive.
multi_class	ovo	Whether to calculate the intercept for this model. If set to false, no intercept will be used in calculations (i.e., data is expected to be already centered).
intercept_scaling	1	When self.fit_intercept is True, instance vector x becomes [x, self.intercept_scaling], i.e., a "synthetic" feature with constant value equals to intercept_scaling is appended to the instance vector.
class_weight	None	Set the parameter C of class i to class_weight[i]*C for SVC. If not given, all classes are supposed to have weight one.
verbose	0	Enable verbose output. Note that this setting takes advantage of a per-process runtime setting in liblinear that, if enabled, may not work properly in a multithreaded context.
random_state	None	Controls the pseudo-random number generation for shuffling the data for the dual coordinate descent (if dual = True). When dual = False the underlying implementation of LinearSVC is not random and random_state has no effect on the results.
max_iter	1000	The maximum number of iterations to be run.

Table 9. LSVM (OVA) parameters from scikit-learn library.

Parameters	Value	Description
penalty	l2	Specifies the norm used in the penalization. The 'l1' leads to coef_ vectors that are sparse.
loss	square_hinge	Specifies the loss function. 'hinge' is the standard SVM loss (used e.g., by the SVC class) while 'squared_hinge' is the square of the hinge loss.
dual	True	Select the algorithm to either solve the dual or primal optimization problem.
tol	1e-4	Tolerance for stopping criteria.
C	1.0	Regularization parameter. The strength of the regularization is inversely proportional to C. Must be strictly positive.

Table 9. *Cont.*

Parameters	Value	Description
multi_class	ovr	Whether to calculate the intercept for this model. If set to false, no intercept is used in calculations (i.e., data are expected to be already centered).
intercept_scaling	1	When self.fit_intercept is True, instance vector x becomes [x, self.intercept_scaling], i.e., a "synthetic" feature with constant value equals to intercept_scaling is appended to the instance vector.
class_weight	None	Set the parameter C of class i to class_weight[i]*C for SVC. If not given, all classes are supposed to have weight one.
verbose	0	Enable verbose output. Note that this setting takes advantage of a per-process runtime setting in liblinear that, if enabled, may not work properly in a multithreaded context.
random_state	None	Controls the pseudo-random number generation for shuffling the data for the dual coordinate descent (if dual = True). When dual = False, the underlying implementation of LinearSVC is not random and random_state has no effect on the results.
max_iter	1000	The maximum number of iterations to be run.

Table 10. Decision tree parameters from scikit-learn library.

Parameters	Value	Description
criterion	gini	This function is used to measure the quality of a split.
splitter	best	The strategy used to choose the split at each node.
max_depth	None	The maximum depth of the tree
min_samples_split	2	The minimum number of samples required to split an internal node.
min_samples_leaf	1	The minimum number of samples required to be at a leaf node.

Table 11. Naïve Bayes parameters from scikit-learn library.

Parameters	Value	Description
var_smoothing	1e-9	Portion of the largest variance of all features that is added to variances for calculation stability.
sample_weight	None	Weights applied to individual samples.
Deep	True	Return the parameters for this estimator and contained sub-objects that are estimators if the value is true.

Table 12. ANN parameters from scikit-learn library.

Parameters	Value	Description
hidden_layer_sizes	(100,)	The ith element represents the number of neurons in the ith hidden layer.
activation	relu	Activation function for the hidden layer.
solver	adam	The solver for weight optimization.
alpha	0.0001	L2 penalty (regularization term) parameter.
batch_size	auto	Size of minibatches for stochastic optimizers.
learning_rate	constant	Learning rate schedule for weight updates.
max_iter	200	The maximum number of iterations.

Table 13. Random Forest parameters from scikit-learn library.

Parameters	Value	Description
n_estimators	100	The number of trees in the forest.
criterion	gini	The function to measure the quality of a split.
max_depth	None	The maximum depth of the tree.
min_samples_split	2	The minimum number of samples required to split an internal node.
min_samples_leaf	1	The minimum number of samples required to be at a leaf node.
max_features	auto	The number of features to consider when looking for the best split.

Table 14. kNN parameters from scikit-learn library.

Parameters	Value	Description
n_neighbors	5	Number of neighbors to use by default for kneighbors queries.
weights	uniform	Weight function used in prediction.
algorithm	auto	Algorithm used to compute the nearest neighbors.

Table 15. CNN parameters from TensorFlow tf.keras.layers.Conv2D function.

Parameters	Value	Description
kernel_size	32 (3,3)	This parameter determines the dimensions of the kernel.
strides	(1,1)	This parameter is an integer or tuple/list of 2 integers, specifying the step of the convolution along with the height and width of the input volume.
padding	valid	The padding parameter of the Keras Conv2D class can take one of two values: 'valid' or 'same'.
activation	relu	The activation parameter to the Conv2D class, allowing us to supply a string specifying the name of the activation function you want to apply after performing the convolution.

4.3. Performance Metrics

The *accuracy* metric is used as the performance metric to measure the model's overall performance on the testing set, supposing that CM is a confusion matrix of n by n dimensions, where n is the total number of different food categories. Furthermore, the row of CM indicates the actual category, while the column of CM indicates the predicted category. Finally, let $C_{i,j}$ indicates the CM cell's value at index row i and column j, where $i, j = 1, 2, \ldots, n$. The *accuracy* metrics is defined as in (1):

$$accuracy = \frac{\sum_{i,j=1}^{n} C_{i,j}}{\sum_{i=1}^{n} \sum_{j=1}^{n} C_{i,j}} \quad (1)$$

5. Results and Discussions

This section is divided into four main sections. Section 5.1 describes the experiment results of the trained model on the Sabah Food Dataset and VIREO-Food172 Dataset, followed by Section 5.2, which describes the comparison of feature dimensions, using CNN as the classifier. Finally, Section 5.3 demonstrates the deployment of the food recognition model through a prototype web application.

5.1. Experiments Results

Figures 3 and 4 shows the classification accuracy of six CNN-based features derived from the transfer-learning process and one color feature over seven different traditional machine learning classifiers and one CNN-based classifier, tested on the Sabah Food Dataset and VIREO-Food172 Dataset, respectively. As seen in Figures 3 and 4, this paper evaluates a total of 56 combinations of machine-learning approaches.

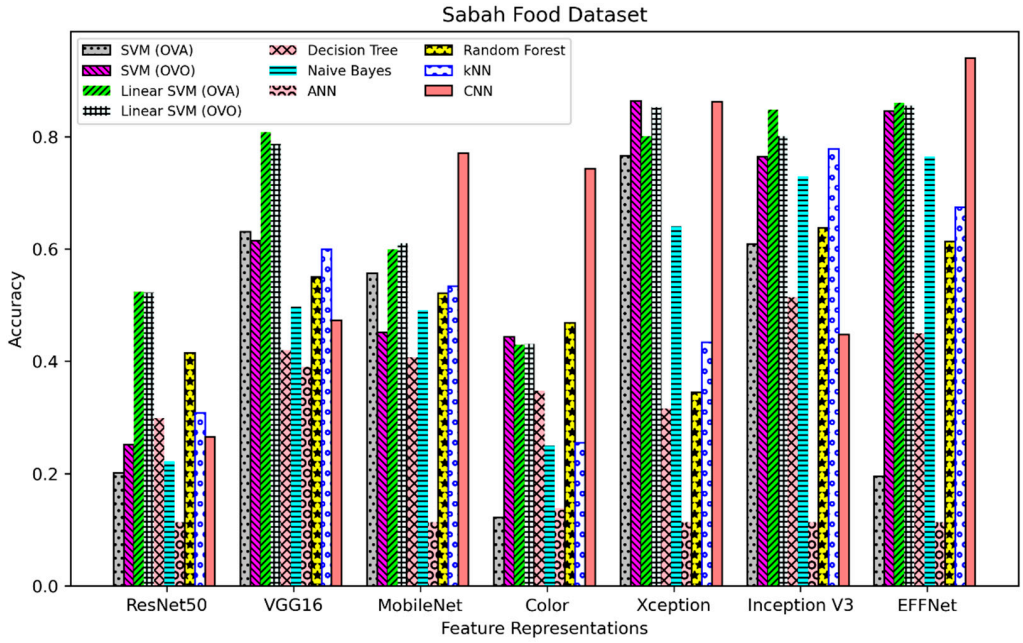


Figure 3. A comparison of the performance of six feature representations and ten classifiers on the Sabah Food Dataset.

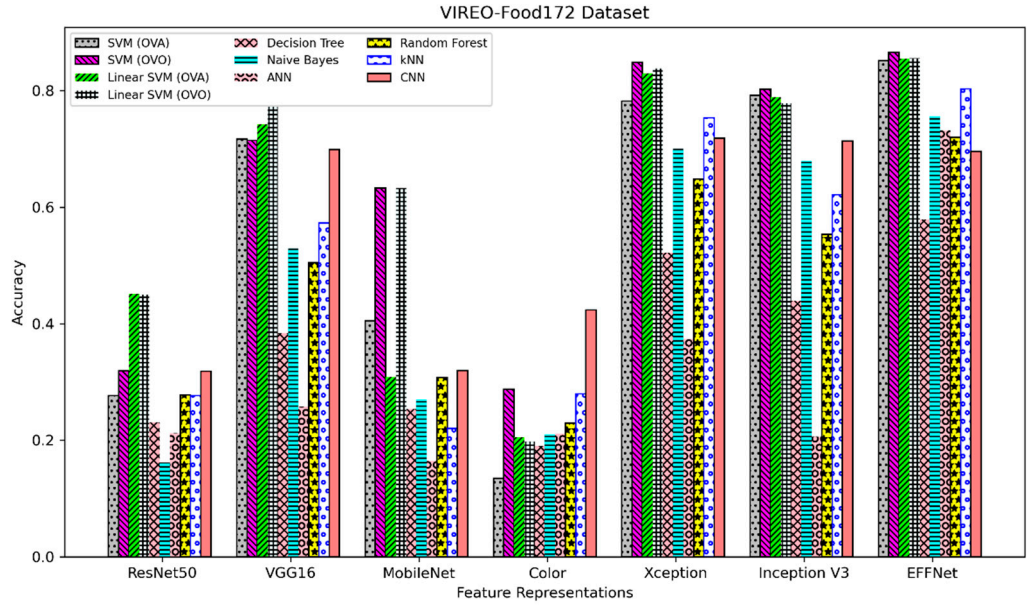


Figure 4. A comparison of the performance of six feature representations and ten classifiers on the VIREO-Food172 Dataset.

The ten highest accuracies of those machine-learning approaches for the Sabah Food Dataset and VIREO-Food172 Dataset shown in Figures 3 and 4 are presented in Tables 16 and 17. The bold formatted machine learning approaches and accuracy in Tables 16 and 17 indicate the best machine learning approaches in that table. Additionally,

Figures 5 and 6 present the confusion matrix for the CNN configuration that performs the best on the Sabah Food and VIREO-Food172 testing sets, respectively.

Table 16. The ten highest performing machine-learning approaches for Sabah Food Dataset.

Machine Learning Approaches	Accuracy
EFFNet + CNN	**0.9401**
Xception + SVM (OVO)	0.8632
Xception + CNN	0.8620
EFFNet + LSVM (OVA)	0.8601
EFFNet + LSVM (OVO)	0.8553
Xception + LSVM(OVO)	0.8522
InceptionV3 + LSVM (OVA)	0.8475
EFFNet + SVM (OVO)	0.8459
VGG16 + LSVM(OVA)	0.8082
Xception + LSVM (OVA)	0.8003

Table 17. The ten highest performing machine-learning approaches for VIREO-Food172 Dataset.

Machine Learning Approaches	Accuracy
EFFNet + SVM (OVO)	**0.8657**
EFFNet + LSVM (OVO)	0.8560
EFFNet + LSVM (OVA)	0.8553
EFFNet + SVM (OVA)	0.8516
Xception + SVM (OVO)	0.8489
Xception + LSVM (OVO)	0.8382
Xception + LSVM (OVA)	0.8304
EFFNet + KNN	0.8035
InceptionV3 + SVM (OVO)	0.8025
InceptionV3 + SVM (OVA)	0.7917

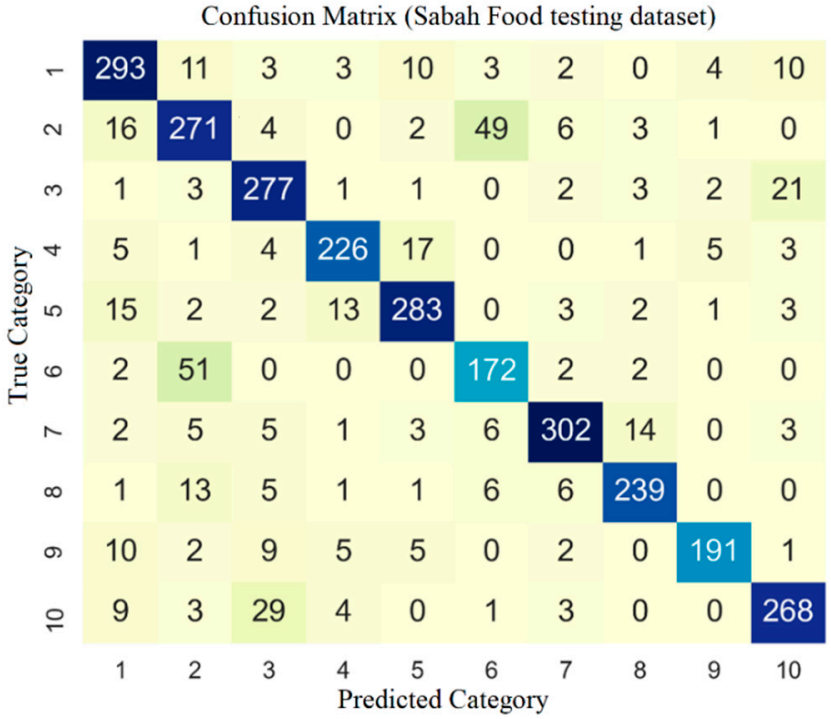


Figure 5. Confusion matrix of EFFNet + CNN applied on Sabah food testing set.

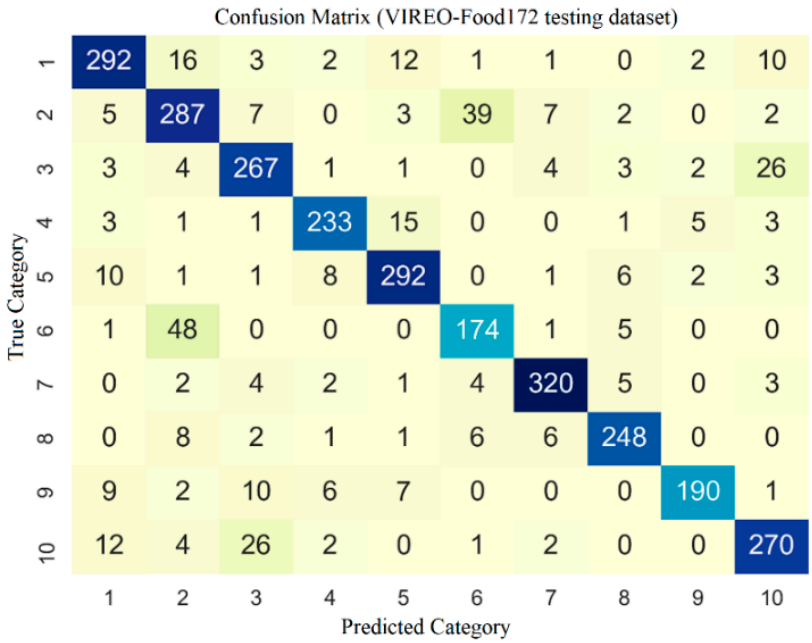


Figure 6. Confusion matrix of EFFNet + SVM (OVO) applied on VIREO-Food172 testing set.

From Table 16, it can be seen that the EFFNet + CNN approach gives the best performance, yielding 0.9401 accuracy for the Sabah Food Dataset. This is followed by Xception + SVM (OVO) (0.8632) and Xception + CNN (0.8620). Additionally, as shown in Table 16, performance decreases significantly from EFFNet + CNN to Xception + SVM (OVO) (the accuracy drops with 0.0769 difference) before gradually decreasing from Xception + SVM (OVO) and the rest of the top 10 highest performing approaches (with differences ranging from 0.0012 to 0.0377). The results suggest that the EFFNet + CNN may only work well on a specific training and testing dataset of the Sabah Food Dataset rather than representing the overall best approach. Nevertheless, EFFNet + CNN is the best performing approach on the Sabah Food Dataset.

On the other hand, for the VIREO-Food172 Dataset, it is observed that the EFFNet + SVM (OVO) provides the best performance (0.8657), as shown in Table 17. However, compared to the top ten performing machine-learning approaches in the Sabah Food Dataset, the differences between each machine-learning approach on the VIREO-Food172 Dataset are more stable (with differences ranging from 0.0007 to 0.0269). In contrast to the best performing approach on the Sabah Food Dataset (Table 16), there is no significant drop in accuracy from the highest to the second-highest accuracy on the VIREO-Food172 Dataset. Additionally, both the Sabah Food Dataset and the VIREO-Food172 Dataset demonstrate that EFFNet provides the best performance when used as a feature representation.

As previously stated, there are seven different feature representations. Therefore, Tables 18 and 19 present seven machine-learning approaches for the Sabah Food Dataset and VIREO-Food172 Dataset, with the best one selected from each group of feature representations and ranked from best to worst accuracy. In Tables 18 and 19, the bold formatted machine learning approaches and accuracy denote the best machine learning approaches in that table. Tables 18 and 19 are similar in that EFFNet is the best feature representation, followed by Xception, Inception V3, and VGG16. Further examination of Table 18 reveals that the accuracy falls precipitously between Color + CNN (0.7422) and ResNet50 + LSVM (OVA) (0.5236), yielding 0.2186 differences. On the other hand, examining Table 19 reveals a gradual decline in accuracy within the first four machine-learning approaches before a significant decrease from VGG16 + LSVM (OVO) (0.7725) to MobileNet + LSVM (OVO) (0.6332), yielding a 0.1393 difference. This drop in accuracy is significant because it tells us which machine-learning approaches should be considered for any future work or

subsequent experiments if accuracy is the most important factor in the food recognition model development.

Table 18. The highest accuracy based on feature representation for Sabah Food Dataset.

Machine Learning Approaches	Accuracy
EFFNet + CNN	**0.9401**
Xception + SVM (OVO)	0.8632
Inception V3 + LSVM (OVA)	0.8475
VGG16 + LSVM (OVA)	0.8082
MobileNet + CNN	0.7708
Color + CNN	0.7422
ResNet50 + LSVM (OVA)	0.5236

Table 19. The highest accuracy based on feature representation for VIREO-Food172 Dataset.

Machine Learning Approaches	Accuracy
EFFNet + SVM (OVO)	**0.8657**
Xception + SVM (OVO)	0.8489
Inception V3 + SVM (OVO)	0.8025
VGG16 + LSVM (OVO)	0.7725
MobileNet + LSVM (OVO)	0.6332
ResNet50 + LSVM (OVA)	0.4519
Color + CNN	0.4237

When the similarities between Tables 18 and 19 are compared, it is seen that EFFNet, Xception, Inception V3, and VGG16 provide more stable performance, with EFFNet feature representation being the best. As a result, an ensemble-based approach based on these four feature representation methods can be considered for future work.

Additionally, Tables 20 and 21 present ten machine-learning approaches for the Sabah Food Dataset and VIREO-Food172 Dataset. The best one was selected from each classifier group and ranked from best to worst accuracy. In Tables 20 and 21, the bold formatted machine learning approaches and accuracy represent the best machine learning approaches in that table. Tables 20 and 21 are then subjected to a similar analysis. From Tables 20 and 21, it can be seen that the EFFNet-based feature representation appears most frequently. Table 20 shows four occurrences of EFFNet, whereas Table 21 shows eight occurrences. Although this is a minor point, it is worth noting that the SVM (OVO) classifier (Xception + SVM (OVO) in the Sabah Food Dataset and EFFNet + SVM (OVO) in the VIREO-Food172 Dataset) appears in the top two of Tables 20 and 21.

Table 20. The highest accuracy based on classifier for Sabah Food Dataset.

Machine Learning Approaches	Accuracy
EFFNet + CNN	**0.9401**
Xception + SVM (OVO)	0.8632
EFFNet + LSVM (OVA)	0.8601
EFFNet + LSVM (OVO)	0.8553
Inception V3 + KNN	0.7783
Xception + SVM (OVA)	0.7657
EFFNet + Naïve Bayes	0.7642
Inception V3 + Random Forest	0.6368
Inception V3 + Decision Tree	0.5142
VGG16 + ANN	0.3899

Table 21. The highest accuracy based on classifier for VIREO-Food172 Dataset.

Machine Learning Approaches	Accuracy
EFFNet + SVM (OVO)	**0.8657**
EFFNet + LSVM (OVO)	0.8560
EFFNet + LSVM (OVA)	0.8553
EFFNet + SVM (OVA)	0.8516
EFFNet + KNN	0.8035
EFFNet + Naïve Bayes	0.7561
EFFNet + ANN	0.7315
EFFNet + Random Forest	0.7201
Xception + CNN	0.7182
EFFNet + Decision Tree	0.5791

In a subsequent analysis of the Sabah Food Dataset and the VIREO-Food172 Dataset, the overall performance of each feature representation is compared in Table 22. The value in the second row and second column in Table 22 (EFFNet) is produced by calculating the average of all machine-learning approaches that use EFFNet as a feature representation technique for the Sabah Food Dataset. This calculation is repeated for all feature representations and both datasets to fill in the second and third columns in Table 22. The value in the fourth column of Table 22 is filled with a value produced by the *Overall Score* defined in (2). The *Overall Score* is calculated by averaging the Mean Accuracy of the Sabah Food Dataset and the Mean Accuracy of the VIREO-Food172 Dataset from the second and third columns of Table 22. Equation (2) is applied to all of the feature representations listed in Table 22 to complete the fourth column.

$$Overall\ Score = \frac{MASFD + MAVFD}{2} \tag{2}$$

where

MASFD = Mean Accuracy of Sabah Food Dataset, and
MAVFD = Mean Accuracy of VIREO-Food172 Dataset.

The *Overall Score* in (2) indicates the performance of a feature representation on both proposed datasets. Following that, the *Overall Score* calculated in (2) is used to facilitate the comparison of all feature representations. The bold formatted Feature Representation and Overall Score in Table 22 represent the best Feature Representation.

Table 22. The overall performance of all feature representations.

Feature Representation	Mean Accuracy of Sabah Food Dataset	Mean Accuracy of VIREO-Food172 Dataset	Overall Score
EFFNet	0.6311	0.7714	**0.7013**
Xception	0.5991	0.7017	0.6504
Inception V3	0.6240	0.6375	0.6308
VGG16	0.5770	0.5896	0.5833
MobileNet	0.5053	0.3516	0.4285
ResNet50	0.3121	0.2977	0.3049
Color	0.3626	0.2370	0.2998

From Table 22, it can be seen that the EFFNet has the best overall performance, followed by Xception, Inception V3, and VGG16 before the *Overall Score* drops significantly for MobileNet, ResNet50, and Color. Therefore, a combination of the EFFNet, Xception, Inception V3, and VGG16 approaches can be considered as components of an ensemble-based approach.

Table 23 shows the overall performance of each classifier. The *Overall Score* in the fourth column of Table 23 is calculated based on (2), which is obtained by averaging the Mean Accuracy of the Sabah Food Dataset and the Mean Accuracy of the VIREO-Food172

Dataset from the second and third columns of Table 23. Similar to the analysis conducted in Table 22, the *Overall Score* calculated in (2) is used to facilitate the comparison of all classifiers. The bold formatted Classifier and Overall Score in Table 23 represent the best Classifier.

Table 23. The overall performance of classifiers.

Classifier	Mean Accuracy of Sabah Food Dataset	Mean Accuracy of VIREO-Food172 Dataset	Overall Score
LSVM (OVO)	0.6941	0.6466	**0.6704**
LSVM (OVA)	0.6954	0.5976	0.6465
SVM (OVO)	0.6049	0.6389	0.6219
CNN	0.6431	0.5555	0.5993
kNN	0.5117	0.5041	0.5079
SVM (OVA)	0.4398	0.5656	0.5027
Naïve Bayes	0.5133	0.4725	0.4929
Random Forest	0.5071	0.4633	0.4852
Decision Tree	0.3933	0.3714	0.3824
ANN	0.1563	0.3082	0.2323

From Table 23, it can be seen that the LSVM (OVO) classifier gives the best overall performance (0.6704), followed by LSVM (OVA) (0.6465), SVM (OVO) (0.6219), and CNN (0.5993) as the classifier. After the CNN classifier, there is a significant drop of *Overall Score* from CNN to kNN, yielding 0.914 difference. As a result, if one is considering a classifier, LSVM (OVO) and LSVM (OVA) are the best options. Additionally, for future work, LSVM (OVO), LSVM (OVA), and SVM (OVO) can be considered as components of an ensemble-based approach.

Finally, Table 24 compares the accuracy of the other methods in Table 1 as well as the accuracy of the food recognition reported in [32] to our work. However, a direct comparison between our model and their model is not possible, due to the differences in the training and testing conditions. Nonetheless, our best performance of 94.01% is comparable to that of [20], which has a 98.16% accuracy. Additionally, our EFFNet + CNN model outperformed the CNN and InceptionV3+CNN models in terms of overall accuracy.

Table 24. A comparison of several food recognition models.

Work	Dataset	Number of Categories	Model	Accuracy (%)
Our Proposed method	Sabah Food Dataset	11	EFFNet + CNN	94.01
Our Proposed method	VIREO-Food172 Dataset	The first ten categories in VIREO-Food172 Dataset	EFFNet + SVM (OVO)	85.57
Jeny et al. (2019) [20]	Self-collected Bangladesh foods dataset	6	FoNet	98.16
Islam et al. (2018) [19]	Food-11 dataset	11	InceptionV3 + CNN	92.86
Chen and Ngo [32]	VIREO-Food172 Dataset	20	MultiTaskCNN	82.12 (Top-1)
Lu (2016) [17]	Small-scale dataset	10	CNN	74.00

5.2. A comparison of Feature Dimension Using CNN as the Classifier

In this work, pre-processing and feature extraction is performed, using a transfer learning strategy based on a pre-trained CNN. As the pre-trained CNN is built up with several layers, there is an option to use all the layers or to pick only a few layers in order to extract the relevant features. The relevance of features is determined by the type and volume of datasets used to train CNNs. For instance, the ImageNet dataset is used to train the CNN model, as it is one of the benchmark datasets in Computer Vision. However, the types of datasets and volume of data used to train the pre-trained CNN models vary, and the effectiveness of the transfer-learning strategy is dependent on the degree to which

the trained models are related to the applied problem domains. Hence, the experiments conducted in this work have revealed the compatibility between the pre-trained CNN model as the feature extractor with the food recognition domain, especially the local food dataset, based on their classification performance.

The selection of layers in the pre-trained CNN model determines not only the relevancy of features but also their feature dimension. The size of the generated features determines the efficiency of running the algorithm. A large number of features entails additional computational effort but likely result in more discriminative features. As shown in Figure 7, the size of the generated features varies according to the final layer or the layer selection on the CNN architecture. It can be seen that the EFFnet has generated the largest feature dimensions (62,720), followed by Inception V3 (49,152), MobileNet (32,768), VGG16 (4096), Color (4096), ResNet50 (2048), and Xception (2048).

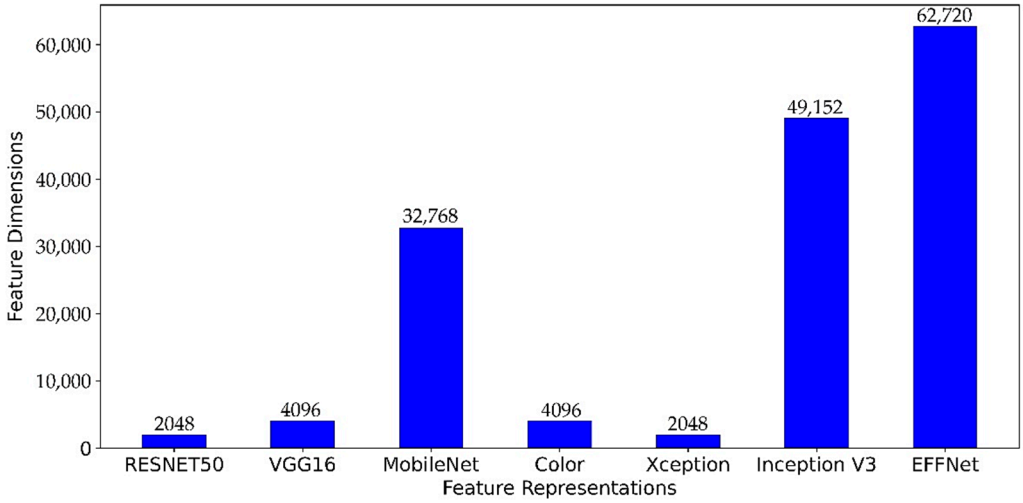


Figure 7. The vector size of the feature presentation (Conv1D).

Table 25 compares the feature dimension and the Overall Score of feature representation. The bold formatted Feature Representation and Overall Score in Table 25 represent the best Feature Representation. While EFFNet as feature representations outperforms Xception in terms of overall performance, Table 25 shows that if computational speed is more important than performance, Xception as feature representation can be considered at the cost of some accuracy performance. The results in Table 25 also indicate that the data used in EFFNet training potentially contain the most relevant and consistent data for extracting meaningful features from food images when compared to other CNN models.

Table 25. A comparison of feature dimension and overall accuracy performance.

Feature Representation	Feature Dimension	Overall Score
EFFNet	**62,720**	**0.7013**
Xception	2048	0.6504
Inception V3	49,152	0.6308
VGG16	4096	0.5833
MobileNet	32,768	0.4285
ResNet50	2048	0.3049
Color	4096	0.2998

Additionally, Figure 8 presents the length and width of features of a pre-trained CNN model for training with a CNN classifier. Each bar in Figure 8 has a label that represents the length and width values. In this case, the length and width are equal. As seen in Figure 8, the Conv2D features generated by EFFNet are minimal (16, 16, 245), compared to the Conv1D features. Despite the high depth of the feature dimension (245), the experiment

revealed no noticeable effect of time efficiency during the training phase. Based on this finding, the model trained with EFFNet features is the best model, as it achieves the highest overall accuracy and generates highly distinctive, yet compact features. In this context, the depth level (z) of the feature's representation determines the efficacy of the classification performance, as more insight of spatial information can be generated. Furthermore, the level of depth of features (z) are more likely have less effect on the overall classification efficiency, compared to the value of x and y axis of the features. As depicted in Figure 8, the MobileNet- and Inception V3-based feature representations produce the highest values of x and y but cost more in terms of execution time than ResNet50, VGG16, Xception, and EFFNet based feature representations. However, in addition to the compatibility of the pre-trained CNN models with the newly developed classification model, the shape of the feature representations is another factor that must be taken into account in the experiment settings.

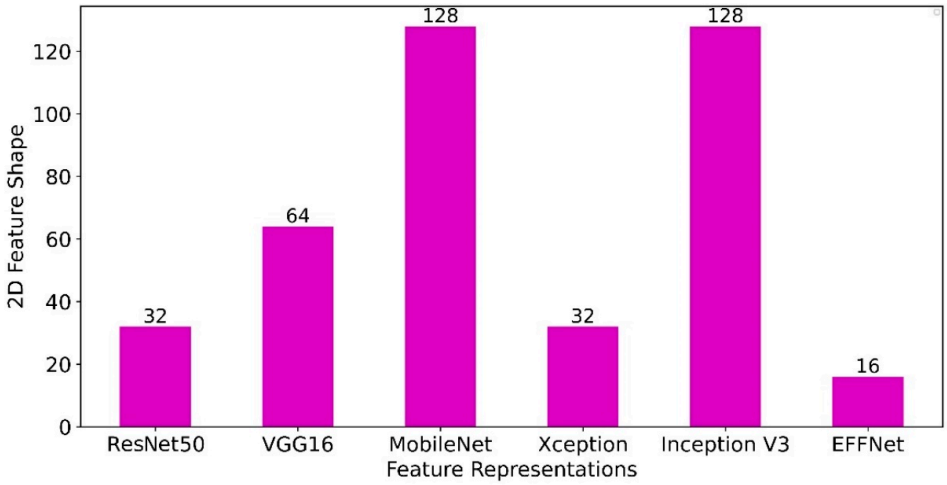


Figure 8. The shape of the feature representation's features (Conv2D).

5.3. Food Recognition Model Deployment

As described previously, a web application system is deployed with the best recognition model (EFFNet-LSVM). The trained model is prepared as a NumPy data structure file using the Joblib library. At the same time, the back-end algorithm for food recognition is integrated with HTML using the Flask framework. Figure 9 shows the main homepage of the preliminary outcome of the prototype web application. Two modules are developed: food recognition and customer feedback module, as shown in Figures 10–12.

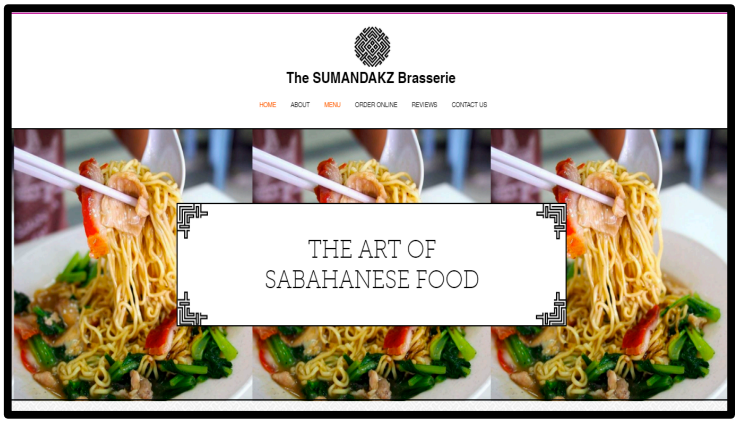


Figure 9. Homepage of the prototype web application.

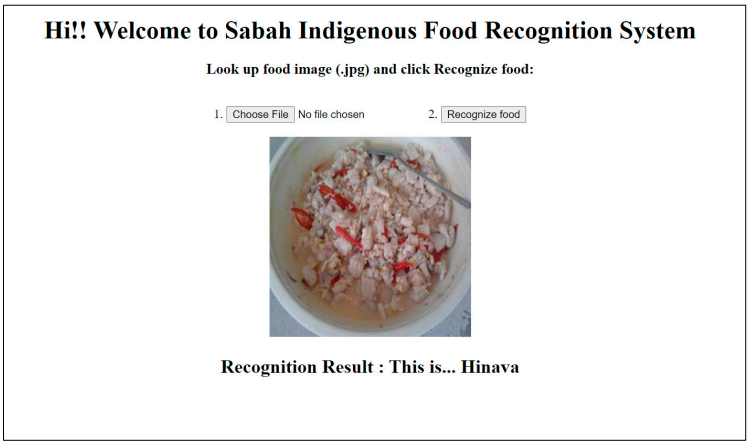


Figure 10. The food recognition module.

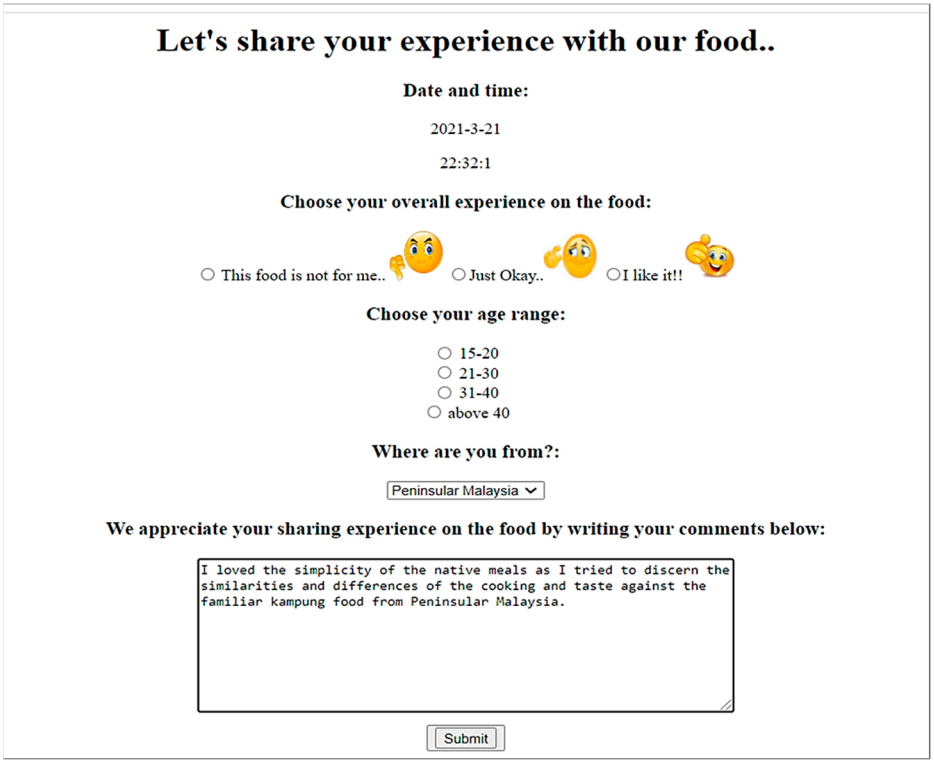
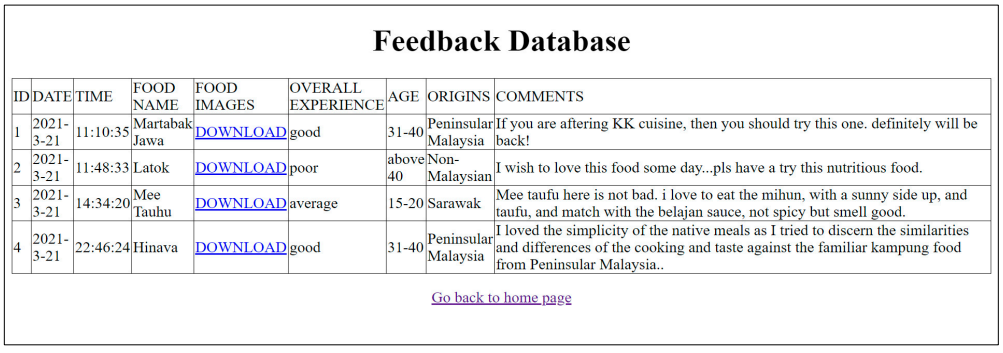


Figure 11. Customer feedback form module.



Feedback Database

ID	DATE	TIME	FOOD NAME	FOOD IMAGES	OVERALL EXPERIENCE	AGE	ORIGINS	COMMENTS
1	2021-3-21	11:10:35	Martabak Jawa	DOWNLOAD	good	31-40	Peninsular Malaysia	If you are aftering KK cuisine, then you should try this one. definitely will be back!
2	2021-3-21	11:48:33	Latok	DOWNLOAD	poor	above 40	Non-Malaysian	I wish to love this food some day...pls have a try this nutritious food.
3	2021-3-21	14:34:20	Mee Tauhu	DOWNLOAD	average	15-20	Sarawak	Mee taufu here is not bad. i love to eat the mihun, with a sunny side up, and taufu, and match with the belajan sauce, not spicy but smell good.
4	2021-3-21	22:46:24	Hinava	DOWNLOAD	good	31-40	Peninsular Malaysia	I loved the simplicity of the native meals as I tried to discern the similarities and differences of the cooking and taste against the familiar kampung food from Peninsular Malaysia..

Go back to home page

Figure 12. List of customer feedback.

As shown in Figure 10, the user must upload a JPG image of the food and click the Recognize food button to invoke the back-end of the food recognition algorithm. The food's name will then appear beneath the image. Finally, another feature included in this system is the ability to collect user feedback on foods via a form, as shown in Figure 11. The administrator can then view all of the customer feedback, as shown in Figure 12.

To summarize, the prototype web application is designed to accomplish three purposes: (i) to provide a food recognition feature for users who are unfamiliar with the food's name, (ii) to enable users to share their food-related experiences via a feedback feature, and (iii) to enable the administrator of this web application system to collect image and feedback data for use in food sentiment analyses and food business analytics. Furthermore, the user's new images can be added to the current food dataset to update the training database, which in turn updates the training model.

6. Conclusions

This paper compared the performance of 70 combinations of food recognition approaches, which consist of six different pre-trained CNN-based models used as feature extractors, one feature representation based on the RGB component of an image, and ten commonly used machine-learning classifiers. Additionally, two types of datasets were used for performance evaluation: (i) the Sabah Food Dataset and (ii) the VIREO-Food172 Dataset. From the comparison, on the Sabah Food Dataset, it was found that the EFFNet + CNN (94.01% accuracy) approach gives the best performance, followed by Xception + SVM (OVO) (86.32% accuracy). However, the significant drop of accuracy from 94.01% to 86.32% suggests that the EFFNet + CNN may be an outlier and only works well on a specific training and testing dataset of the Sabah Food Dataset, rather than representing the overall best approach. On the VIREO-Food172 Dataset, it was found that the EFFNet + SVM (OVO) (86.57% accuracy) provides the best performance, followed by EFFNet + LSVM (OVO) (85.60% accuracy). In comparison to the Sabah Food Dataset, the difference between the best and second-best performing approaches in VIREO-Food172 Dataset is insignificant (0.97% difference). It should be noted that the best performing feature representation for both the Sabah Food Dataset and the VIREO-Food172 Dataset is the EFFNet-based feature representation. This is supported by the paper's discussion of the *Overall Score* of feature representation, which demonstrates that EFFNet has the highest *Overall Score* of feature representation. A similar comparison was made for the classifiers, and it was found that the LSVM (OVO) classifier gives the best overall performance for food recognition, followed by LSVM (OVA) as the classifier. In terms of computational complexity and memory space usage, while EFFNet (with 62,720 feature dimension) as feature representations outperformed Xception in terms of overall performance, if computational speed and memory space usage are more important than performance, then Xception (with 2048 feature dimension) can be considered at the expense of a small accuracy performance reduction. As part of the implication of this work, this paper also presented a food recognition model for indigenous foods in Sabah, Malaysia, by utilizing a pre-trained CNN model as a feature representation and a classifier. The classification accuracy (94.01%) achieved by EFFNet + CNN in the performance evaluation results for the Sabah Food Dataset is very promising for real-time use. As a result, a prototype web-based application for the Sabah food business analytics system was developed and implemented using the EFFNet + CNN approach for a fully automated food recognition using real-time food images.

Future Work

For future work, this research should conduct more experiments to obtain a more rigorous analysis of the CNN hyper-parameters and the CNN layers to achieve more solid and concrete findings. The types and number of implemented CNN layers and the feature shape can be further analyzed. Additionally, the feature selection algorithm can be studied further to reduce the dimensionality of the features, as this has a significant effect on the computational time. Furthermore, the criteria for selecting the training database

for a food recognition system can be explored further. It was found in [34] that using the database's mean class in the training database can potentially improve the system's performance. Finally, to further improve the accuracy, a study on an ensemble-based approach, using a combination of EFFNet, Xception, Inception V3, VGG16, LSVM, and CNN, can be considered. Another interesting area to consider is food sentiment analysis. The user feedback data can be incorporated into a food sentiment analysis module, with the aim that it will assist business owners in remaining informed about the market acceptance of their food products. The customer feedback data can be analyzed further to improve the quality and innovation of indigenous foods, allowing them to be more commercialized and ultimately contribute to Sabah's gastronomic tourism industry. Finally, another area that can be investigated is the food business prediction module, which allows for the analysis of food market trends and provides additional data to industry practitioners in order to strategize their food business direction.

Author Contributions: M.N.R., writing—original draft preparation, data curation; E.G.M., writing—review and editing, supervision, funding acquisition, visualization; F.Y., writing—review, conceptualization; C.J.H., writing—review and editing; R.H., writing—review; R.M., writing—review; I.A.T.H., writing—review. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Center for Research (PPP), Universiti Malaysia Sabah, under grant number SGA0006-2019 and GA19095. The APC was funded by the Center for Research (PPP), Universiti Malaysia Sabah, under grant number SGA0006-2019 and GA19095.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Conflicts of Interest: The authors declare that they have no conflict of interest.

References

1. Fam, K.S.; Syed Annuar, S.N.; Tan, K.L.; Lai, F.H.; Ingko, I.A. Touring destination and intention to consume indigenous food: A case of Kadazan-Dusun food in Sabah. *Br. Food J.* **2019**, *122*, 1883–1896. [CrossRef]
2. Mnguni, E.; Giampiccoli, A. Proposing a model on the recognition of indigenous food in tourism attraction and beyond. *Afr. J. Hosp. Tour. Leis.* **2019**, *8*, 1–13.
3. Noor, A.M.; Remeli, M.R.B.; Hanafiah, M.H.M. International tourist acceptance of Sabah's gastronomy product. *Curr. Issues Hosp. Tour. Res. Innov.* **2012**, *57*, 377.
4. Danting, Z.; Quoquab, F.; Mahadi, N. Enhancing the Tourism Operation Success in Sabah Malaysia: A Conceptual Framework. *Int. J. Eng. Technol.* **2018**, *7*, 147–151. [CrossRef]
5. Nasrudin, N.H.; Harun, A.F. A preliminary study on digital image performance to stimulate food taste experience. *Bull. Electr. Eng. Inform.* **2020**, *9*, 2154–2161. [CrossRef]
6. Kiourt, C.; Pavlidis, G.; Markantonatou, S. Deep Learning Approaches in Food Recognition. In *Machine Learning Paradigms*; Springer: Cham, Switzerland, 2020; pp. 83–108.
7. Prasanna, N.; Mouli, D.C.; Sireesha, G.; Priyanka, K.; Radha, D.; Manmadha, B. Classification of Food categories and Ingredients approximation using an FD-Mobilenet and TF-YOLO. *Int. J. Adv. Sci. Technol.* **2020**, *29*, 3101–3114.
8. Upreti, A.; Malathy, D.C. Food Item Recognition, Calorie Count and Recommendation using Deep Learning. *Int. J. Adv. Sci. Technol.* **2020**, *29*, 2216–2222.
9. Yang, H.; Kang, S.; Park, C.; Lee, J.; Yu, K.; Min, K. A Hierarchical deep model for food classification from photographs. *KSII Trans. Internet Inf. Syst.* **2020**, *14*, 1704–1720. [CrossRef]
10. Razali, M.N.; Manshor, N. A Review of Handcrafted Computer Vision and Deep Learning Approaches for Food Recognition. *Int. J. Adv. Sci. Technol.* **2020**, *29*, 13734–13751.
11. Mohamed, R.; Perumal, T.; Sulaiman, M.; Mustapha, N. Multi-resident activity recognition using label combination approach in smart home environment. In Proceedings of the 2017 IEEE International Symposium on Consumer Electronics (ISCE), Kuala Lumpur, Malaysia, 14–15 November 2017; pp. 69–71. [CrossRef]
12. Zainudin, M.; Sulaiman, M.; Mustapha, N.; Perumal, T.; Mohamed, R. Two-stage feature selection using ranking self-adaptive differential evolution algorithm for recognition of acceleration activity. *Turk. J. Electr. Eng. Comput. Sci.* **2018**, *26*, 1378–1389.
13. Moung, E.G.; Dargham, J.A.; Chekima, A.; Omatu, S. Face recognition state-of-the-art, enablers, challenges and solutions: A review. *Int. J. Adv. Trends Comput. Sci. Eng.* **2020**, *9*, 96–105. [CrossRef]

14. Dargham, J.A.; Chekima, A.; Moung, E.G. Fusing facial features for face recognition. In *Distributed Computing and Artificial Intelligence*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 565–572.
15. Dargham, J.A.; Chekima, A.; Moung, E.; Omatu, S. Data fusion for face recognition. In *Distributed Computing and Artificial Intelligence*; Springer: Berlin/Heidelberg, Germany, 2010; pp. 681–688.
16. Yahya, F.; Fazli, B.; Sallehudin, H.; Jaya, M. Machine Learning in Dam Water Research: An Overview of Applications and Approaches. *Int. J. Adv. Trends Comput. Sci. Eng.* **2020**, *9*, 1268–1274. [CrossRef]
17. Lu, Y. Food Image Recognition by Using Convolutional Neural Networks (CNNs). *arXiv* **2016**, arXiv:1612.00983.
18. Subhi, M.A.; Ali, S.M. A Deep Convolutional Neural Network for Food Detection and Recognition. In Proceedings of the 2018 IEEE-EMBS Conference on Biomedical Engineering and Sciences (IECBES), Kuching, Sarawak, Malaysia, 3–6 December 2018; pp. 284–287.
19. Islam, M.T.; Karim Siddique, B.M.N.; Rahman, S.; Jabid, T. Food Image Classification with Convolutional Neural Network. In Proceedings of the 2018 International Conference on Intelligent Informatics and Biomedical Sciences (ICIIBMS), Bangkok, Thailand, 21–24 October 2018.
20. Jeny, A.A.; Junayed, M.S.; Ahmed, T.; Habib, M.T.; Rahman, M.R. FoNet-Local food recognition using deep residual neural networks. In Proceedings of the 2019 International Conference on Information Technology, ICIT 2019, Bhubaneswar, Odisha, India, 20–22 December 2019.
21. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778. [CrossRef]
22. Zahisham, Z.; Lee, C.P.; Lim, K.M. Food Recognition with ResNet-50. In Proceedings of the 2020 IEEE 2nd International Conference on Artificial Intelligence in Engineering and Technology (IICAIET), Kota Kinabalu, Malaysia, 26–27 September 2020; pp. 1–5.
23. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. In Proceedings of the 3rd International Conference on Learning Representations (ICLR), San Diego, CA, USA, 7–9 May 2015.
24. Taşkıran, M.; Kahraman, N. Comparison of CNN Tolerances to Intra Class Variety in Food Recognition. In Proceedings of the 2019 IEEE International Symposium on INnovations in Intelligent SysTems and Applications (INISTA), Sofia, Bulgaria, 3–5 July 2019; pp. 1–5.
25. Howard, G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv Preprint* **2017**, arXiv:1704.04861.
26. Chollet, F. Xception: Deep Learning with Depthwise Separable Convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 1251–1258.
27. Yao, N.; Ni, F.; Wang, Z.; Luo, J.; Sung, W.-K.; Luo, C.; Li, G. L2MXception: An improved Xception network for classification of peach diseases. *Plant. Methods* **2021**, *17*, 1–13. [CrossRef] [PubMed]
28. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deep-er with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1–9. [CrossRef]
29. Singla, A.; Yuan, L.; Ebrahimi, T. Food/Non-food Image Classification and Food Categorization using Pre-Trained GoogLeNet Model. In Proceedings of the 2nd International Workshop on Multimedia Assisted Dietary Management, Amsterdam, The Netherlands, 16 October 2016.
30. Tan, M.; Le, Q.V. Efficientnet: Rethinking model scaling for convolutional neural networks. *arXiv* **2019**, arXiv:1905.11946.
31. Liu, J.; Wang, M.; Bao, L.; Li, X. EfficientNet based recognition of maize diseases by leaf image classification. *J. Physics: Conf. Ser.* **2020**, *1693*, 012148. [CrossRef]
32. Chen, J.; Ngo, C.-W. Deep-based Ingredient Recognition for Cooking Recipe Retrieval. In Proceedings of the 24th ACM international conference on Multimedia, Amsterdam, The Netherlands, 15–19 October 2016; pp. 32–41.
33. Hatcher, W.G.; Yu, W. A Survey of Deep Learning: Platforms, Applications and Emerging Research Trends. *IEEE Access* **2018**, *6*, 24411–24432. [CrossRef]
34. Dargham, J.A.; Chekima, A.; Moung, E.G.; Omatu, S. The Effect of Training Data Selection on Face Recognition in Surveillance Application. *Adv. Intell. Syst. Comput.* **2015**, *373*, 227–234. [CrossRef]