Vietnam Journal of Science and Technology ***60*** *(5) (2022) 837-852*

doi:10.15625/2525-2518/16299

AN EMPIRICAL EVALUATION OF FEATURE EXTRACTION FOR VIETNAMESE FRUIT CLASSIFICATION

Thuan Trong Nguyen[1, 2, *], Hai Le[1, 2], Truong Nguyen[1, 2], Thinh Le[1, 2], Khanh Duong[1, 2], Quyen Tran[1, 2], Vu Bui[1, 2], Hop Nguyen[1, 2], Nguyen D. Vo[1, 2], Khang Nguyen[1, 2]

[1]*University of Information Technology, Ho Chi Minh City, Viet Nam, Quarter 6, Linh Trung Ward, Thu Duc City, Ho Chi Minh City, Viet Nam*

[2]*Vietnam National University, Quarter 6, Linh Trung Ward, Thu Duc City, Ho Chi Minh City, Viet Nam*

[*]Email: *18521471@gm.uit.edu.vn*

Received: 18 July 2021; Accepted for publication: 9 August 2021

Abstract. In recent years, Vietnamese fruit production has marked significant progress in terms of scale and product structure. Viet Nam enjoys suitable climates for tropic, subtropical fruits, and some temperate fruits. Thanks to the diversified ecology, implementing an automatic classification system has received a significant concern. In this paper, we address the problem of Vietnamese fruit classification. However, the first requirement to explore this problem is the qualified dataset. To this end, we first introduced the UIT-VinaFruit20 dataset, a novel Vietnamese fruit image dataset that includes 63,541 images from 20 types of fruits from three regions of Viet Nam. The diversity in different fruits in distinct areas poses many new challenges. In addition, we further leverage the feature extraction from hand-crafted and deep learning features along with the SVM classification model to effectively classify Vietnamese fruits. The extensive experiments conducted on the UIT-VinaFruit20dataset provide a comprehensive evaluation and insightful analysis. An encouraging empirical success was obtained as the EfficientNetB0 feature achieved the best results of 85.465 % and 86.919 % in terms of Accuracy and Macro F1-score, respectively.

Keywords: fruit classification, image classification, feature extraction, UIT-VinaFuit20.

Classification numbers: 4.7.3, 4.7.4.

1. INTRODUCTION

Image classification is a core task in computer vision (CV), which attempts to understand an entire image comprehensively as a whole. The goal is to classify an image by assigning it to a corresponding label. In image classification, Convolutional Neural Network (CNN) has become the state-of-the-art (SOTA) method. CNN showed its effectiveness in improving accuracy in benchmark datasets such as ImageNet. CNN revolutionized this field by learning basic shapes in the first layers and evolving to learn the features of images in the deeper layers, resulting in more accurate image classification. However, CNN may take a long time to train on large datasets. Another approach based on this process is to reuse model weights from pre-trained models developed for standard CV benchmark datasets. Specifically, the images through the feature extractor transform to feature vectors that describe specific features. These features can then be used as input during the development of a new model [1, 2].

Viet Nam is one of the world's tropical fruit meccas. Fruit is sold from a plethora of street vendors to supermarkets and is consumed in large quantities in daily life in Viet Nam. Therefore, an automatic Vietnamese fruit classification system has become a promising domain for researchers in the CV community. Fruit classification allows automatic identification of fruit items from each image captured by a camera in a smartphone. It means people could easily access fruits, especially foreign tourists who enjoy exploring Vietnamese food. However, the classification of fruits still has many challenges because of the similarities in shape and color between different types of fruits in three regions of Viet Nam, which is easy to cause confusion in prediction, even if humans cannot recognize them exactly. Specifically, the fruits captured may vary in camera's conditions, such as scale, lighting, and background. In addition, to the best of our knowledge, a qualified dataset on Vietnamese fruit is still lacking. Therefore, this paper focuses on introducing a novel Vietnamese fruit dataset. In addition, we perform empirical evaluations in our dataset.

The main contributions of our paper can be summarized as follows:

- We introduced a novel Vietnamese fruit image dataset named UIT-VinaFruit20[1], containing 63,541 images of 20 popular Vietnamese fruits.
- We provided a comprehensive evaluation based on extensive experiments on the UIT-VinaFruit20 dataset. Specifically, we conduct experiments with 2 hand-crafted features (HOG, LPB) and 19 deep learning features (VGG16, VGG19, ResNet50, ResNet50V2, ResNet101, ResNet101V2, ResNet152, ResNet152V2, InceptionV3, InceptionResNetV2, Xception, MobileNet, MobileNetV2, DenseNet121, DenseNet169, DenseNet201, NASNetMobile, NASNetLarge, and EfficientNetB0) to evaluate the impact of feature vectors on the SVM classification model.

This paper is organized as follows: we review the related work in Section 2 and introduce the UIT-VinaFruit20 dataset in the next section, then focus on our approach methods in Section 4. Section 5 provides experimental results as well as analysis. Finally, we present the conclusion and future work in Section 6.

2. RELATED WORK

2.1. Food Dataset

In recent years, food classification in general or fruit classification in particular has received more and more attention in the research community. This work required a qualified dataset. As a result, datasets have increased rapidly in both quality and quantity. Table 1 provides a statistical summary of the publicly available datasets for food classification. In general, almost all datasets focus on foods which have many different ingredients. For example, ChineseFoodNet [3], a large Chinese cuisine dataset, contains 185,628 images from 208 food categories that the author collected from a top favorite Chinese food survey. FoodX-251 [4]

[1]UIT-VinaFruit20 published athttps://uit-together.github.io/datasets/

includes 158,000 images for 251 classes, which are fine-grained and visually similar. A large-scale and highly diverse food image dataset with 500 categories and about 400,000 images called ISIA Food-500 [5], which cover various countries and regions, includes Eastern and Western cuisines, and the existing typical ones mainly belong to 11 categories as classified in the dataset The Vietnamese food dataset is VinaFood21 [1], which contains 13,950 images from 21 popular dishes from street food and daily meals in Viet Nam.

Table 1. Statistics of publicly available datasets for foods classification as well as fruit and vegetable classification.

Datasets	Images	Categories	Coverage	Year
ChineseFoodNet [3]	185,628	208	Chinese Foods	2007
FoodX-251 [4]	158,846	251	Foods	2019
ISIA Food-500 [5]	399,726	500	Foods	2020
VinaFood21 [1]	16,726	21	Vietnamese Foods	2021
VegFru [8]	160,000	317	Fruits and Vegetables	2017
Fruit 360[2]	90,483	131	Fruits and Vegetables	2020

The vegetable and fruit datasets in Kaggle are numerous, but the published datasets are still scarce. Fruits 360 is a dataset published in Kaggle, which is used as a benchmark dataset for fruit and vegetable recognition with 90,380 images and 131 classes. VegFru [6] dataset includes 160,000 images and covers vegetables and fruits of 25 upper-level categories and 292 subordinate classes, taking all species in common.

2.2. Fruit Classification

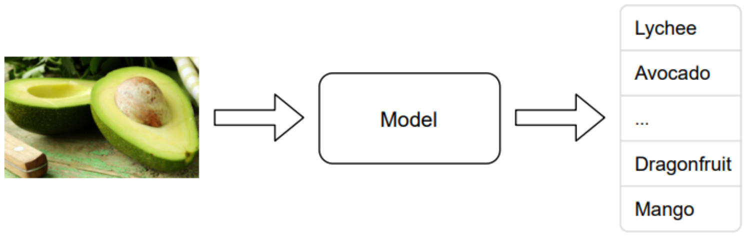

Figure 1. The fruit classification problem: It receives a fruit image and returns the fruit name in the list of the fruits.

We can describe the fruit classification problem as it receives a fruit image as input and returns the fruit name that belongs to the fruit list as defined (the problem of fruit classification

[2]https://www.kaggle.com/moltean/fruits

as described in Figure 1). There have been several approaches to fruit classification in recent years. Hung *et al*. [7] presented a solution to the classification of fruits by performing pixel classification. The algorithm learns the most important features of a fruit. W. C. Seng and S. H. Mirisaee [8] proposed a method for fruit recognition system, which combines three approaches of feature analysis: color-based, shape-based, and size-based then using kNN classification. Lei Hou [9] proposed a fruit recognition algorithm based on CNN. The regions train the CNN to extract from original images, and the original image type is based on the fusion of classification results of each region. Duong *et al*. [10] applied EfficientNet and MixNet, two families of deep neural networks, to build an efficient expert system to identify fruits. Recently, SpinalNet [11] achieved an accuracy of 99.90 % in the Fruit 360 dataset. It allows each hidden layer to receive a part of the inputs and outputs of the previous layer. Therefore, the number of incoming weights in a hidden layer is significantly lower than the traditional model.

2.3. Transfer Learning

Transfer Learning is a Machine Learning technique whereby a model is trained and developed for one task and then reused for a second related task. Transfer learning benefits from reducing the training time for a neural network model, resulting in lower generalization errors. In practice, it is hard to train a CNN from scratch because of the lack of a sufficiently large dataset. Instead, it is common to train a CNN on a large dataset (e.g., ImageNet dataset, which contains 1.2 million images with 1000 categories) and then use the pre-trained model either as an initialization or a fixed feature extractor to perfrom the task of interest. There are two approaches to implementing transfer learning. The first is the feature extractor from the pre-trained model and then training a classifier on top of it. And the second is fine-tuning the pre-trained model keeping learned weights as initial parameters.

Feature Extraction has proven to be useful for training traditional machine learning models (e.g., kNN, Logistic Regression, SVM.). In this work, the classification model receives the vector features extracted from the deep learning model as input. It greatly reduces the training time while retaining the accuracy. In 2021, Nguyen *et al*. [1] applied the feature extractor technique of Transfer Learning using pre-trained weights on the ImageNet dataset combined with Linear SVM and achieved an accuracy of 73.99 % on the VinaFood21 dataset. Compared to the training CNN model from scratch, they demonstrate that Transfer Learning performs better than training a small dataset on a deep learning model with millions of parameters. Vo *et al*. [2] also used this technique for classifying X-ray images in a COVID dataset and achieved an accuracy of 97 % with the VGG19 features and SVM.

3. UIT-VINAFRUIT20

3.1. Category Selection

Located in the tropical zone, Viet Nam is truly a heaven when it comes to fruits. There are even tours arranged exclusively for tourists who enjoy visiting orchards to witness how fruits are grown and try fresh fruits in the garden. Many Fruit Festivals are held every year in many localities, attracting millions of locals and foreign visitors. Therefore, we examined many websites about travels with several blogs written for Vietnamese cuisine then summarized the most well-known Vietnamese fruits, which received high recommendations from locals and foreigners. After that, we found 20 popular fruits (see Figure 2) to start the first step in building the Vietnamese fruit dataset, which is data collection.

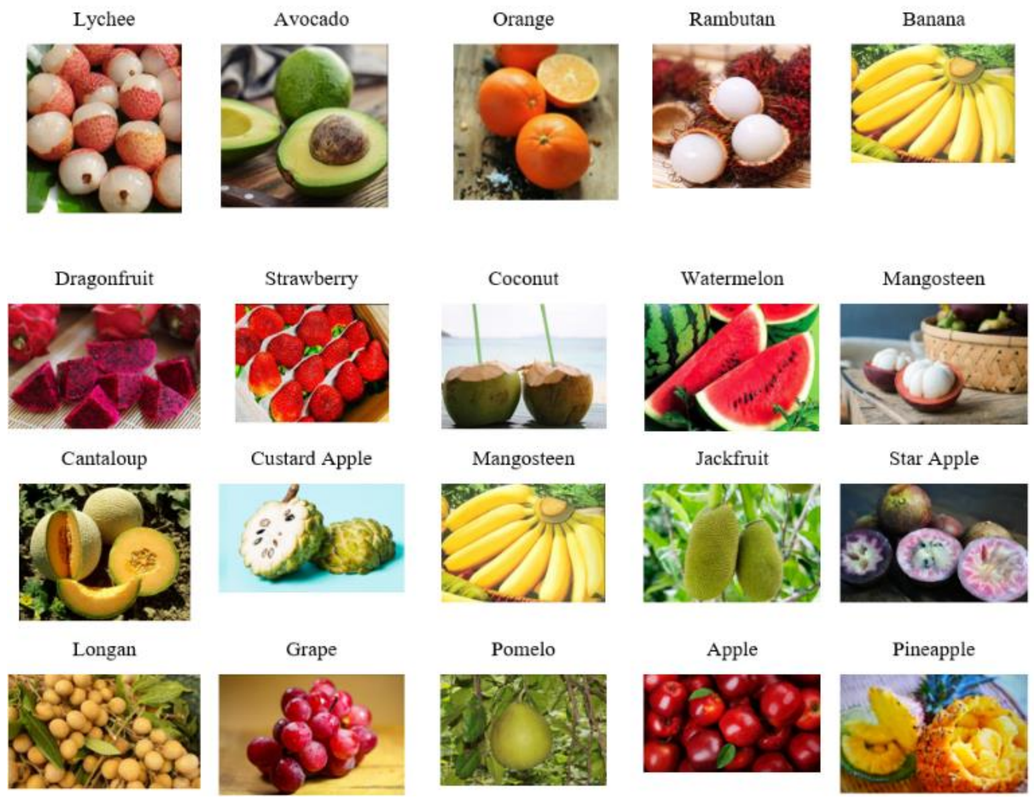

Figure 2. Illustration of 20 fruits in UIT-VinaFruit20.

3.2. Data Annotation

We mainly collected images from Google imagesand expanded search terms by adding some keywords. In addition, we also used smart phones to take photos of the fruits that we met every day. We collected approximately 100,000 images, which were saved in 20 different folders named after each fruit. However, these images might be kept in an incorrectly labeled folder. Furthermore, some images were of low quality and were duplicated. Hence, we conducted data cleaning by removing duplicated and low-quality images as well as moving the images to the correct folder.

In the next step, to ensure no mistakes in each class as well as to improve the quality of our dataset, we performed two sub-stages to verify the labels. The first sub-stage is for each people to double-check their label. The second is to cross-check each other's labels. If we found any error in the dataset, we would discuss it together.

3.3. Dataset Description

UIT-VinaFruit20 contains 63,541 images corresponding to 20 Vietnam popular fruits, with an average of about 3,700 images per class. UIT-VinaFruit20 is the first Vietnamese fruit dataset with three main features promising more challenges.

(1) Diversity in the color of one fruit. We collected both images from unripe fruit to the fruit we can eat, which have different colors in each period of fruit. Furthermore, the similarity in color between fruits is also a factor of interest.

(2) Similar in shape. Fruit categories from our dataset cover various high intra-class distances (different look but similar type) and low inter-class distances (similar look but another type).

(3) Diversity in the appearance of fruit in the image. We were not limited toimages containing only one type of fruit in them. Therefore, fruit can appear in various locations of an image with many sub-objects. For example, fruit can occur in a meal that includes other foods, or fruit can be arranged in a gift basket containing another item.

4. METHOD

This section presents the experimental process on the UIT-VinaFruit20 dataset and the metrics for evaluating performance. The whole process is shown in Figure 3.

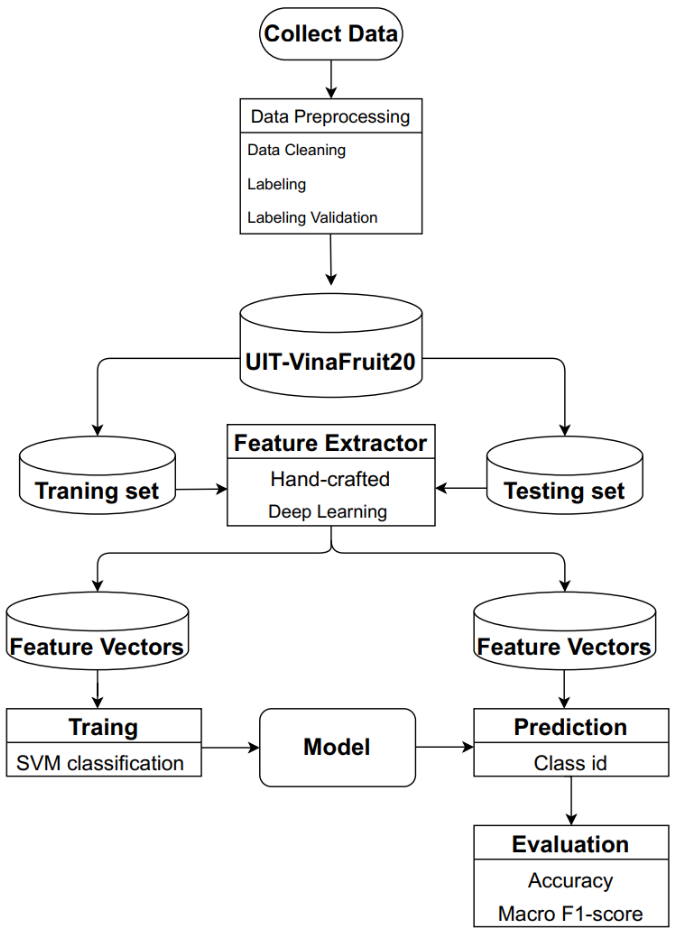

Figure 3. Diagram of our workflow. The overall experiment process for the fruit classification on the UIT-VinaFruit20.

4.1. Implement Process

UIT-VinaFruit20 was divided into training and testing sets with a ratio of 8 : 2, specifically we used 53,868 images for training and 9,673 images for testing (the experimental dataset is shown in Figure 2).The classification model was built based on the Transfer Learning technique. However, we intended to examine the features from hand-crafted and deep learning. Therefore, we only implemented feature extractors, starting with a feature extractor and then feeding feature vectors into the SVM classification model. We used Keras Applications[3], which are deep learning models available with pre-trained weights, to implement the feature extractor. We used pre-trained weights in the ImageNet dataset for all feature extractors from the deep learning model. In addition, we used the SVM model with a default kernel of RBF from the scikit-learn[4] library for Pythonto perform classification.

4.2. Data Pre-processing

All input images were resized as well as converted to fit the default standard of the feature extractors. The images for the HOG and LBP feature were resized to 64x128 and 16x16, respectively. For the deep learning model, VGG, ResNet, MobileNet, DenseNet, EfficientNet, and NASNetMobile features require 224x224 input image. Inception family and Xception require input image with a size of 299x299. Only NasNetLarge has input image of 331x331 in size. More detailed information is given in Table 2.

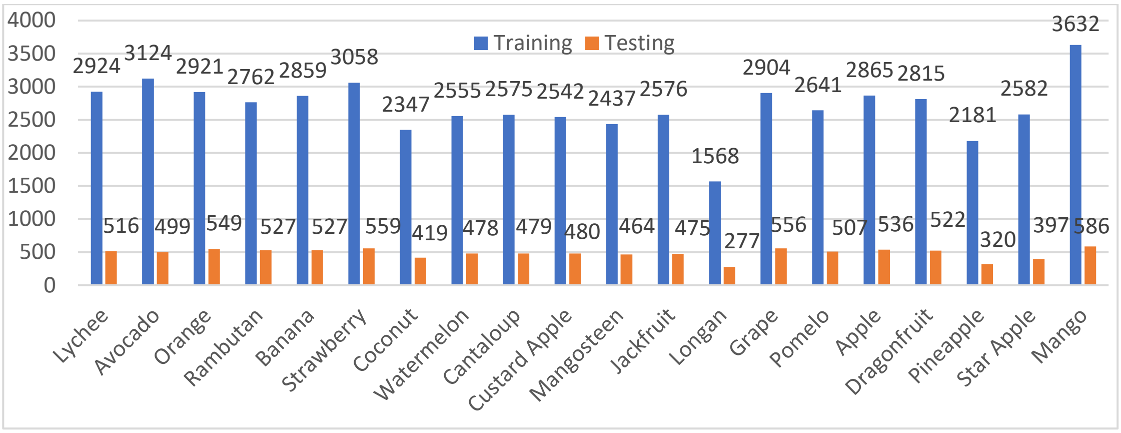

Figure 4. Organization of experimental data.

4.3. Feature Extractor

4.3.1. Hand-crafted Features

Histogram of Oriented Gradients (HOG): Dalal N. and Triggs B. introduced the HOG [12] feature extraction, in which an image is divided into small square cells and then the histogram of oriented gradients in each of them is computed. When dividing and calculating

[3]https://keras.io/api/applications/

[4]https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html

each cell one by one, the histogram’s value is only local, so the next step is to normalize the larger blocks, including these small cells. The feature vector is obtained from normalized blocks.

Local Binary Pattern (LBP): Ojala introduced LBP [13] in 1996 to the human facial recognition problem. LPB is a method of accessing image information by measuring local contrast based on looking at each pixel. Each pixel in the image (center pixel) has n-neighbor pixels. Then n-bit binary sequences are initialized corresponding to n-neighbor pixels. If the center pixel intensity is greater than or equal to its neighbor, the value returns 1. Otherwise, it returns 0.

Table 2. The default input image size and output vector shape from feature extraction.

Feature	Input	Output	Feature	Input	Output
HOG	(64,128,3)	(1,3786)	InceptionResNetV2	(299,299,3)	(1,1536)
LPB	(16,16,1)	(1,256)	Xception	(299,299,3)	(1,2048)
VGG16	(224,224,3)	(1,4096)	MobileNet	(224,224,3)	(1,1000)
VGG19	(224,224,3)	(1,4096)	MobileNetV2	(224,224,3)	(1,1280)
ResNet50	(224,224,3)	(1,2048)	DenseNet121	(224,224,3)	(1,1024)
ResNet50V2	(224,224,3)	(1,2048)	DenseNet169	(224,224,3)	(1,1664)
ResNet101	(224,224,3)	(1,2048)	DenseNet201	(224,224,3)	(1,1920)
ResNet101V2	(224,224,3)	(1,2048)	NASNetMobile	(224,224,3)	(1,1056)
ResNet152	(224,224,3)	(1,2048)	NASNetLarge	(331,331,3)	(1,4032)
ResNet152V2	(224,224,3)	(1,2048)	EfficientNetB0	(224,224,3)	(1,1280)
InceptionV3	(299,299,3)	(1,2048)			

4.3.2. Deep learning features

VGG: Although VGGNet [14] didn't win the ILSVRC 2014 competition, it is still one of the top deep learning networks that attract considerable attention because its efficiency has been significantly improved compared to previous networks. In particular, instead of using an extensive filter like 11×11 in AlexNet or 7×7 in ZFNet, VGGNet uses a small 3×3 filter. Therefore, the number of parameters is less, the computation is less complicated, the weights converge faster, and the model can prevent overfitting. VGGNet has many variations whose difference comes from the number of layers in the network architecture. For example, VGG16 has 1st and 2nd blocks consisting of 2 convolutional layers and a MaxPool layer. Then the 3rd, 4th, and 5th blocks include three convolutional layers and a MaxPool layer. Finally, a fully connected SoftMax layer is trained. On the other hand, instead of having three convolutional layers in the 3rd, 4th, and 5th blocks like VGG16, VGG19 has four convolutional layers and a MaxPool layer. This change makes an improvement in this architecture which helps the VGG19 network learn more deeply.

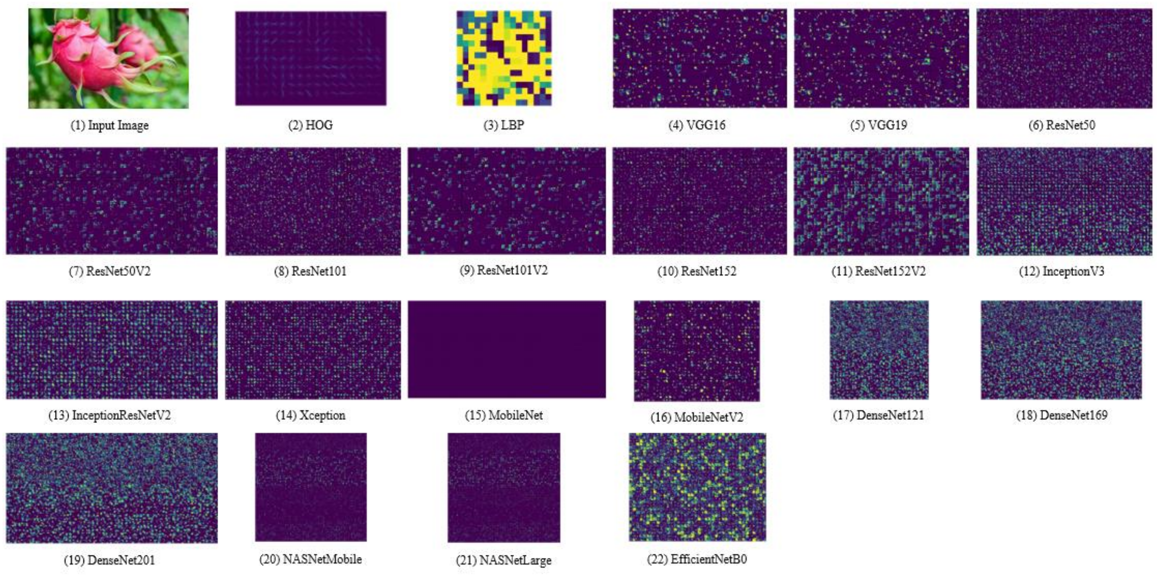

Figure 5. Visualization of the feature from hand-crafted and deep learning extraction. Features are displayed as a feature map for the deep learning feature.

ResNet: Based on VGGNet, many CNN models have been developed to become more insightful. However, many experiments have shown that the model's accuracy will be saturated to a certain threshold or even make it less accurate. ResNet [15] solved this by introducing a new concept, which is the skip connection. The skip connection connects the previous layer to the next layer and skips some intermediate layers to reduce information loss. ResNet is also the earliest architecture to adopt Batch Normalization. The architecture of ResNet consists of 2 characteristic blocks: Convolution block and Identity block. The convolution block consists of a convolution branch and a skip connection branch which passes through the 1x1 convolutional layer before adding to the convolution branch. On the other hand, the Identity block also has a convolution branch and a skip connection branch that does not apply a 1x1 convolution but adds the previous layers' values directly to the convolution branch. These two blocks are called residual blocks. ResNet also has many variations, such as ResNet-18, ResNet-34, and ResNet-50, with a suffix specifying the number of layers in the network architecture. For example, ResNet 50 has 50 layers with one convolutional layer, 1 MaxPool layer, and 46 layers in residual blocks. The last one is AvgPool and a SoftMax layer. There are many variants of ResNet architecture with the same concept but with a different number of layers: ResNet-18, ResNet-34, ResNet-50, ResNet-110, ResNet-164, ResNet-1202, etc. Besides the ResNet version 1, the ResNet family also contains version 2, all about using the pre-activation of weight layers instead of post-activation.

DenseNet: DenseNet [16] is quite similar to ResNets. However, DenseNet can solve the problem of vanishing gradients better than ResNet. DenseNet has a simple connectivity pattern to ensure the maximum flow of information between layers in forwarding computation and backward gradient computation. This network connects all layers, so each layer obtains additional inputs from all preceding layers and passes its feature maps to all subsequent layers. DenseNet-121 has only 8 million parameters, but it is more accurate than ResNet-50 with nearly 26 million parameters on the ImageNet dataset. DenseNet also has many variations such as DenseNet-121, DenseNet-169, DenseNet-201, and DenseNet-264.

Inception: Inception has many popular versions such as Inception-V1, Inception-V2, Inception-V3, Inception-V4, and Inception-ResNet. Inception-V3 [17] is the successor to the network architecture of Inception-V1. A Batch Normalization layer and a ReLu activation are added after the convolution layers. Inception-V3 solved representational bottlenecks. For example, the output’s size dropped dramatically compared to the input and efficient computation factorization method. Inception-ResNet-v2 [18] is the latest version in the Inception family with the difference in incorporates residual connections (replacing the filter concatenation stage of the Inception architecture).

Xception: Xception [19] is proposed by the Google research team, which improved from Inception-V3 and replaced standard modules with convolutional masses that can be separated in depth. Depthwise convolution is to perform a convolution of 3x3 on each input channel and combine the results. Then pointwise convolution operates a convolution of 1x1 on the convolution results in depthwise convolution. The order of the two operations is inconsistent: Inception performs a convolution of 1x1, then a convolution of 3x3. Meanwhile, depthwise convolution does the opposite. In short, Xception improves the efficiency of the model without increasing the network’s complexity.

MobileNet: MobileNet [20] used a new convolutional calculation called Separable Convolution, reducing the model size and computational complexity. Thanks to this improvement, the model can be run on mobile applications, embedded devices, or handle real-time tasks. In 2017, a group of researchers from Google announced MobileNetV2, which yielded impressive results. In particular, the network architecture uses an inverted residual structure. The input and output are thin bottleneck layers instead of expansion one like traditional residual block because the authors believe that intermediate layers must do the nonlinear activation functions. Hence, they need to be thicker to make more transformations. On the other hand, experiments show that using nonlinear transformations at the input and output of residual blocks will cause information loss. The authors also replaced the nonlinearity function at the input and output layers with linear projections. These above changes help MobileNetV2 reduce the input parameters by 30 %, 30-40 %, and the accuracy is significantly improved compared to MobileNetV1.

NASNet: NAS stands for Neural Architecture Search. Equipped with an abundance of computing power and engineering genius, Google introduced NASNet [21], which framed finding the best CNN architecture as a Reinforcement Learning problem. The idea was to search the best combination of the given search space parameters of filter sizes, output channels, strides, layers, etc. This work designs a new search space (which we call the "NASNet search space") that enables transferability. The algorithm will search for the best convolutional layer (or "cell") on the CIFAR-10 dataset and then apply this cell to the ImageNet dataset by stacking together more copies of this cell. In addition, Scheduled DropPath is a new regularization technique that significantly improves generalization in the NASNet models. On CIFAR-10 itself, a NASNet found by our method achieves an error rate of 2.4 %, which is a state-of-the-art result. Although the cell is not directly searched on ImageNet, a NASNet constructed from the best cell achieves, among the published works, a state-of-the-art accuracy of 82.7 % top-1 and 96.2 % top-5 on ImageNet. When evaluated at different levels of computational cost, the accuracies of NASNets exceed those of the state-of-the-art human-designed models. For instance, a small version of NASNet is NASNetMobile, which also achieves 74.8 % top-1 accuracy. In comparison, the most extensive version is NASNetLarge, which reaches 82.7 % top-1 accuracy.

EfficientNet: Mingxing Tan and Quoc V. Le introduced EfficientNet [22] as a new approach for model scaling. Previous researcheshave just focused on scaling depth or scaling width or scaling resolution separately. Still, experiments have demonstrated that scaling does not improve performance after reaching a certain threshold. So that, they came up with the idea that scaling all three depth, width, and image resolution attributes strategically to improve the current performances. The EfficientNet research searches to efficiently scale CNN architectures using the calculation of compound scaling parameters. As a result, they have gained a family of deep learning models with eight variations (B0 to B7). The smallest version of EfficientNet is EfficientNetB0. This architecture is similar to NASNetMobile, but it mainly utilizes the bottlenecks we have seen in MobileNetV2. Furthermore, this research adds squeeze-and-excite optimization and the powerful Swish activation function, which has returned an impressive result with an accuracy of 77 % when testing on the ImageNet dataset.

4.4. Classification Model

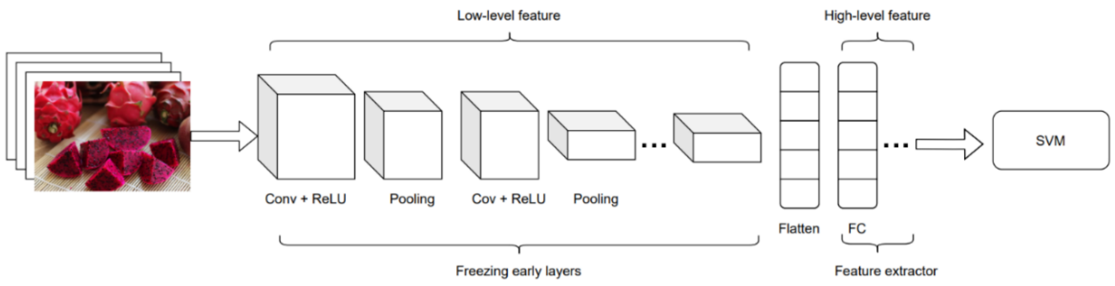

Figure 5. Overall of the training phase. We applied transfer learning to the deep learning model as a feature extractor, and SVM performs classification.

Support Vector Machine (SVM) is a supervised machine learning algorithm used for both classification or regression challenges. However, it is mainly used in classification problems. In the SVM algorithm, we perform classification by using a kernel trick technique to transform our data. SVM finds a hyperplane so that the distance between the dividing line and the data points between classes is maximum (known as the maximum-margin hyperplane).

In our training phase, we used the transfer learning technique. We performed both hand-drafted and deep learning as a feature extractor to achieve vector features for more detail. In the deep learning feature extractor, we only used high-level features to ensure fairness between each model (it means we removed the SoftMax layer and reused the last FC layer as a vector feature) (see Figure 5). We used 19 deep learning models to extract features, such as: VGG16, VGG19, ResNet50, ResNet50V2, ResNet101, ResNet101V2, ResNet152, ResNet152V2, InceptionV3, InceptionResNetV2, Xception, MobileNet, MobileNetV2, DenseNet121, DenseNet169, DenseNet201, NASNetMobile, NASNetLarge, and EfficientNetB0. In addition, we found that the fruits have a significantly similar color and texture, so we also used two hand-crafted feature extractors. Therefore, HOG and LPB have performed feature extraction from the input images. The requirement about the default of an input image's size and the output of a feature vector is reported in Table 2, and we visualized the features in Figure 5. After that, the feature vectors are fed into the SVM model to conduct classification.

4.5. Evaluation Metrics

We have a confusion matrix for every class $c_i \in C = \{1, \dots, n\}$ such that the $i - th$ confusion matrix considers class g_i as the positive class and all other classes c_j with $j \neq i$ as the negative class. The formula below will denote TP_i, TN_i, FP_i, and FN_i to indicate true positives, true negatives, false positives, and false negatives in the confusion matrix associated with the $i-th$ class.

Accuracy: indicates quantitatively the number of correct predictions made.

$$Accuracy = \frac{1}{|C|}\sum_{i=1}^{|C|}\frac{\mathrm{TP_i}}{\mathrm{TP_i}+TN_i+FP_i+FN_i} \tag{1}$$

Macro Average Precision: indicates quantitatively the number of correct positive predictions made.

$$Precision_{macro} = \frac{1}{|C|}\sum_{i=1}^{|C|}\frac{\mathrm{TP_i}}{\mathrm{TP_i}+FP_i} \tag{2}$$

Macro Average Recall: indicates quantitatively the number of correct positive predictions made out of all positive predictions made.

$$Recall_{macro} = \frac{1}{|C|}\sum_{i=1}^{|C|}\frac{\mathrm{TP_i}}{\mathrm{TP_i}+FN_i} \tag{3}$$

Macro Average F1-score: provides a way to combine both precision and recall into a single measure that captures both properties.

$$F1_{macro} = 2\frac{Precision_{macro}Recall_{macro}}{Precision_{macro}+Recall_{macro}} \tag{4}$$

If F1-score has a large value (very close to 1), this indicates that a classifier performs well for each class. The Macro Average F1-score is given more importance for the dataset with an imbalanced class distribution because it averages over larger groups, namely over the performance for individual classes rather than observations.

5. RESULTS AND DISCUSSION

5.1. Experimental Results

The experimental results are summarized in Table 3 for Accuracy and Macro F1-score. Macro F1-score is more helpful than Weighted F1-score to reflect the class imbalance distribution in UIT-VinaFruit20 because it considers performance for specific classes rather than observations. In general, the EfficientNetB0 feature has the best results, achieving 85.465 % and 86.919 % for the Accuracy and Macro F1-score, respectively, which far surpasses any other features. This dominance has several challenges with ResNet family version 1. Specifically, the figures for Macro F1-score are 81.075 %, 80.929 %, and 80.059 %, respectively for ResNet101, ResNet152, and ResNet50. In contrast, InceptionResNetV2 features reach the lowest results, with the figures of only 5.159 % and 0.508 % for Accuracy and Macro F1-score, respectively. In addition, the experiments have shown that hand-crafted features, such as LBP, have had a significant improvement over other deep learning features such as NASNetLarge, RestNet152V2, and NASNetMobile. Furthermore, not only is HOG higher than the features just listed, it is even slightly better than ResNet101V2 and InceptionV3, and very close to Xception.

Table 3. The experiment results on four datasets in Accuracy (%) and Macro F1-score (%).

Feature	Accuracy	F1-score	Feature	Accuracy	F1-score
InceptionResNetV2	5.159	0.508	MobileNet	30.797	33.758
NASNetLarge	5.169	0.526	DenseNet169	37.444	42.431
ResNet152V2	5.417	0.928	DenseNet121	41.445	45.985
NASNetMobile	5.655	1.236	DenseNet201	41.487	46.017
LPB	6.244	2.229	VGG16	68.510	71.447
ResNet101V2	6.296	2.115	VGG19	68.665	71.690
InceptionV3	7.836	5.330	ResNet50	77.566	80.059
HOG	8.632	7.085	ResNet152	78.466	80.929
Xception	9.139	7.078	ResNet101	78.621	81.075
ResNet50V2	11.144	9.634	**EfficientNetB0**	**85.465**	**86.919**
MobileNetV2	25.525	28.668			

We specifically analyze the best and the worst classification model, starting with the model using the features from EfficientNetB0 with SVM classification. The fruits achieving the best results in recognition are "Watermelon" and "Avocado" with the F1-score of 75.495 % and 51.793 %, respectively. In the next section, we will have a more detailed analysis of the results of this model. In contrast, based on the classification report, it is found that the model with the InceptionResNetV2 feature performs very poorly. The model can only implement correct predictionsfor "Avocado" and "Orange", but their F1 scores are not significant, only 9.805 % and 0.364 %, respectively. Therefore, the figure for Macro F1-score is insignificant, just only 0.508 % for 20 classes.

5.2. Analysis

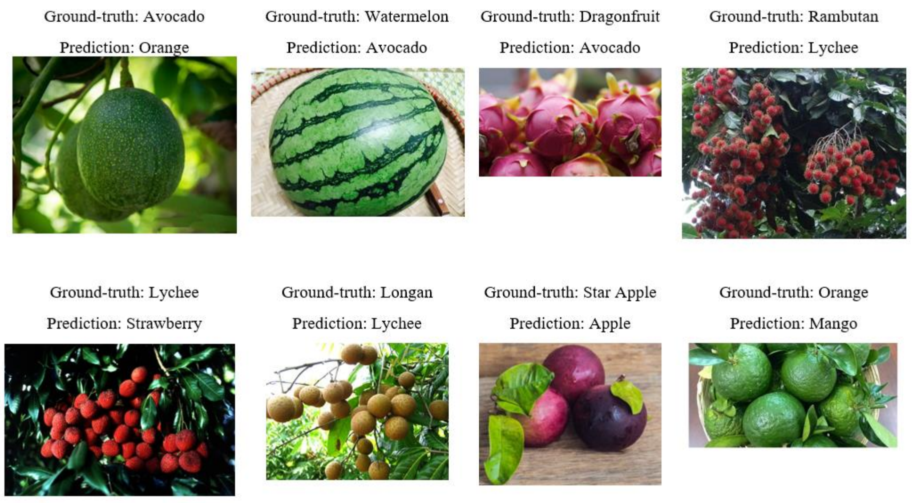

Figure 6. Images of the fruits that are incorrectly predicted.

We first explain how reusing pre-trained weights in the ImageNet dataset leads to high performance. We found 1000 classes from the ImageNet dataset which also included fruits (e.g., Kiwi, Orange, Apple, Watermelon, Peach, Banana, Grape, Lemon, etc.). It had several classes such as Orange, Apple, Watermelon, Banana, and Grape, which are very similar to those in the UIT-VinaFruit20 dataset. So our model has inherited useful knowledge from the knowledge learned on ImageNet. However, it only accounts for an insignificant number. Thus, the confusion is still widespread because some fruits have the same common features. The confusion matrix is reviewed and the most confusing classes are summarized in Figure 6.

The confusion often occurs in the class with high intra-class distances and low inter-class distances. Due to the differences in each region of Viet Nam, the same fruit grown in different regions also has differences in color and shape. Almost all classes take into account most of the confusion which is usually the same shape or color. For example, in Figure 6, "Dragon fruit" and "Avocado" have a considerable color difference, but their bodies are oval in shape. Furthermore, "Lychee" and "Strawberry" have the same color and shape, and both of them are very close to "Rambutan". In practice, it is challenging for people to recognize precisely with just one look. To explain the model's bad case, we found that the deep learning features give bad results for the classification model. The model only has correct predictions in the class that appears in ImageNet. This means that the SVM model just only worked with the feature vector that had been linearly separated before. As the two problems have been mentioned, we can conclude that the model only learned the fruits' concept for both issues. Therefore, unfreezing more layers or adding some FC layers for more details on high-level features could be a solution to this problem which we will experiment with in the future.

6. CONCLUSION AND FUTURE WORK

In this paper, we introduce a novel dataset of Vietnamese fruits with 20 popular fruits in Viet Nam, UIT-VinaFruit20. The dataset contains 63,541 fruit images taken in daily life. We have conducted extensive experiments to evaluate the impact of both hand-crated features and deep learning features on the SVM model. The experimental results showed that the EfficientNet-B0 features achieved outperforming results with an Accuracy of 85.465 % and a Macro F1-score of 86.919 %. For future work, we plan to collect more samples to extend the current dataset as well as reduce the classes of imbalanced distributionsand continue to improve the fruit recognition models. We hope to publish the UIT-VinaFruit20 dataset and our methods to contribute to the research community on Vietnamese food classification in general and fruit classification in particular.

Acknowledgment. The research team would like to express our sincere thanks to the Multimedia Communications Laboratory (MMLab) and Faculty of Information Science and Engineering - the University of Information Technology, VNU-HCM for supporting our team in this research process.

CRediT authorship contribution statement. Thuan Trong Nguyen: Methodology, Experimental, Formal analysis. Author 2-8: Dataset. Nguyen D. Vo and Khang Nguyen: Supervision, etc.

Declaration of competing interest. The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

REFERENCES

1. Thuan Trong Nguyen, Thuan Q. Nguyen, Dung Vo, Vi Nguyen, Ngoc Ho, Nguyen D. Vo, Kiet Van Nguyen, and Khang Nguyen. - VinaFood21: A Novel Dataset for Evaluating Vietnamese Food Recognition, Proceedings of the International Conference on Computing and Communication Technologies (RIVF), 2021, pp. 1-6.
2. Vo Thi Mot, Vo Duy Nguyen, Nguyen Tan Tran Minh Khang - Extract Features and Classification of Chest X-Ray Images, TNU Journal of Science and Technology. **226** (07) (2021). https://doi.org/10.34238/tnu-jst.
3. Chen X., Zhu Y., Zhou H., Diao L., Wang D. - Chinesefoodnet: A large-scale image dataset for chinese food recognition. *arXiv preprint arXiv:1705.02743*, 2017.
4. Kaur P., Sikka K., Wang W., Belongie S., Divakaran. - Foodx-251: A dataset for fine-grained food classification, *arXiv preprint arXiv*:1907.06167, 2019
5. Min W., Liu L., Wang Z., Luo Z., Wei X., Wei X., Jiang S. - Isia food-500: A dataset for large-scale food recognition via stacked global-local attention network, Proceedings of the 28th ACM International Conference on Multimedia, 2020, pp. 393-401.
6. Hou S., Feng Y., Wang Z. - Vegfru: A domain-specific dataset for fine-grained visual categorization, Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 541-549.
7. Hung C., Underwood J., Nieto J., Sukkarieh S. - A feature learning based approach for automated fruit yield estimation, Proceedings of the Field and service robotics, 2015, pp. 485-498.
8. Sen, Woo Chaw and Mirisaee S. H. - A new method for fruits recognition system, Proceedings of the International Conference on Electrical Engineering and Informatics, 2009, pp. 130-134.
9. Hou L., Wu Q., Sun Q., Yang H., Li P. - Fruit recognition based on convolution neural network, Proceedings of the International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD), 2016, pp. 18-22.
10. Duong L. T., Nguyen P. T., Di Sipio C., Di Ruscio D. - Automated fruit recognition using EfficientNet and MixNet, Computers and Electronics in Agriculture **9** (5) (2020). https://doi.org/10.1016/ j.jmrt.2020.06.079.
11. Kabir H. M., Abdar M., Jalali S. M., Khosravi A., Atiya A. F., Nahavandi S., Srinivasan D. - Spinalnet: Deep neural network with gradual input, *arXiv preprint* arXiv:2007.03347, 2020
12. Dalal N., Triggs B. - Histograms of oriented gradients for human detection, Proceedings of the IEEE computer society conference on computer vision and pattern recognition, 2005, pp.886-893.
13. Timo Ojala, Matti Pietikainen, and David Harwood - A comparative study of texture measures with classification based on featured distributions. Proceedings of the Pattern recognition, 1996, pp. 51-59.
14. Simonyan K., Zisserman A. - Very deep convolutional networks for large-scale image recognition, *arXiv preprint* arXiv:1409.1556, 2014

15. He K., Zhang X., Ren S., Sun J. - Deep residual learning for image recognition, Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770-778.
16. Huang G., Liu Z., Van Der Maaten L., Weinberger K. Q. - Densely connected convolutional networks. Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 4700-4708.
17. Szegedy C., Liu W., Jia Y., Sermanet P., Reed S., Anguelov D., Erhan D., Vanhoucke V., Rabinovich A. - Going deeper with convolutions, Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 1-9.
18. Szegedy C., Ioffe S., Vanhoucke V., and Alemi A. - Inception-v4, inception-resnet and the impact of residual connections on learning, Proceedings of the AAAI conference on artificial intelligence, 2017, pp. 4278-4284.
19. Chollet F. - Xception: Deep learning with depthwise separable convolutions, Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 1251-1258.
20. Howard A. 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* arXiv:1704.04861, 2017.
21. Zoph B., Vasudevan V., Shlens J., Le Q. V. - Learning transferable architectures for scalable image recognition, Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 8697-8710.
22. Tan M., Le Q. - Efficientnet: Rethinking model scaling for convolutional neural networks,Proceedings of the International Conference on Machine Learning, 2019, pp. 6105-6114.