Engineering Journal issn: 0125-8281

Article

Bread Browning Stage Classification Model using VGG-16 Transfer Learning and Fine-tuning with Small Training Dataset

Prapassorn Tantiphanwadi[1,a,*] **and Kritsanun Malithong**[2,b]

1 Department of Industrial Engineering, Faculty of Engineering at Khamphaeng Saen, Kasetsart University, Thailand
2 Department of Food Engineering, Faculty of Engineering at Khamphaeng Saen, Kasetsart University, Thailand
E-mail: [a,*]prapassorn.tant@ku.th (Corresponding author), [b]kritsanun@ku.th

Abstract. Convolutional neural network (CNN) is a popular tool to recognize image features even though its weakness requirement of massive training dataset. However, the implementation of the network in production process needs to worry about, that is, the deal with at least two constraints, small training dataset and the less diversity of browning stages among the bread production batches. This paper is aimed to achieve a high predictive accuracy model to classify the bread browning stage that is capable to deal with these constraints. With small training dataset of 900 original images from a production batch, the research performs five steps, starting with a few convolutional layers, adding image augmentation technique and transfer learning with pre-trained CNN model to enhance feature extraction with fine-tuning in final step. The final model of VGG-16 transfer learning and fine-tuning, trained with 18,000 artificial images, successfully achieves very high training accuracy of 98.89% and a very low loss of 2.86% at a small number of epochs 30 with its predictive accuracy of 100%.

Keywords: Bread browning stage, VGG-16, transfer learning, fine-tuning, image augmentation.

ENGINEERING JOURNAL Volume 26 Issue 11
Received 5 May 2022
Accepted 21 November 2022
Published 30 November 2022
Online at https://engj.org/
DOI:10.4186/ej.2022.26.11.1

DOI:10.4186/ej.2022.26.11.1

1. Introduction

Most bread customers are very sensitive to the bread color appearance. Although other criteria such as smell and softness is very important but color is the most consideration the customer will give the first consideration. Many bread manufacturers are looking for automated inspection system to improve their processes and bread appearance to be more appetite, efficient and profitable with customer satisfaction in-mind.

Today, there are many innovative companies [1]-[2] that manufacture machine vision system for bread inspection based on image processing techniques. The system can inspect both varieties of bread types and bread defects. Examples of bread types are bread, bun, cookie, biscuit, croissant, muffin, etc. Some bread defective examples are the values of surface color, the area of topping coverage, crack length and width, etc. Small pieces of metals can be detected by x-rays. An innovative company, Sesotec [3], manufactures an artificial intelligence (AI) that utilizes x-rays beam with machine learning algorithm to detect different types of metals. Recently, contaminants within bread are studied with near infrared (NIR) spectroscopy for detection [4] and fully convolutional neural network for classification.

Machine learning tools are recently utilized together with digital image processing techniques, to detect bread browning stages and other defects A color-based machine vision system is developed to evaluate muffin surface stages utilizing the discriminant analysis classification algorithm [5] with accuracy better than 80%. The Support Vector Machine (SVM) algorithm is utilized to classify biscuit into eight different groups [6] with accuracy ranged between 86.75% and 87.25%. Furthermore, SVM algorithm is capable to classify moving biscuits that are on conveyor belt at high speed of 9 meters per minute [7] with accuracy above 96%.

Some advanced digital image processing techniques are applied in the purpose of more bread feature extraction. For example, a wavelet function is applied on biscuit image to extract four features of color, size, shape and texture [8] or the six threshold methods [9]-[10] are used to extract features on baked cookie and biscuit.

Today, deep learning is an interesting algorithm to be applied on bread surface. A research [11] utilizes a simple neural network to classify surface color of baked bread with accuracy of 93%. Current method uses a few convolutional layers and Inception-v3 module [12], trained with augmented images of small square pieces of bread crust, it is able to obtain training accuracy 98.8% after 200 training epochs.

Utilizing CNN with small training dataset is a very active area of research and it can be implemented in the real-time manufacturing situation. Many works have been investigated and studied in wide area, such as defect detection in production line, etc. [13]-[17]. In this paper, some constraints that affect the training accuracy and loss of CNN model for bread browning stage classification are as follows:

- Small training dataset of each browning stage.
- Diversities of bread browning stages from different production batches.
- Long training time consumption will affect uncomfortable working environment to the production workers, especially on the event of new product introduction.

In order to get a very high accuracy model that is capable to deal with these constraints, the research proposes a model with the techniques of enhancing the numbers of training image with image augmentation, efficient feature extraction with the pretrained VGG-16 model and weight improvement with fine-tuning technique. The expected training classification model should achieve production productivity acceptance level of 98% or above within 30 epochs with small training dataset. The detailed works are presented in the following sections.

2. Related Deep Learning knowledge

2.1. Image Augmentation

Currently, image augmentation is a very popular simulation method to provide plenty of expected artificial images. Keras utilities provide the augmentation thru ImageDataGenerator function or Generative adversarial networks (GANs) algorithm that composes of generator and discriminator. The augmented data has been proven among pretrained CNN models in that training accuracy results are better than those without augmented data [18]-[19].

Our chosen bread sample is the circular bun, then the selected geometric techniques are limited to rotation and horizontal flip only. Color shading are the crucial augmentation, thus changing in hue values, saturation and brightness are recommended for the experimentation. With the strategy, five artificial images out of an original image are generated.

Rotation is the movement on a plane (x, y) in which its radius is constant. Its angle, θ, starting with zero at a point at the end of radius. The transformed matrix, T, is defined as in Eq. (1) [20]:

$$T = \begin{bmatrix} cos\theta & sin\theta & 0 \\ -sin\theta & cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{1}$$

Image is treated as two dimensional array, then a pixel coordinate is (x, y, 1) in which its transformation is obtained from the dot product as shown in Eq. (2) and Eq. (3):

$$C_{transformed} = [x \;\; y \;\; 1] \begin{bmatrix} cos\theta & sin\theta & 0 \\ -sin\theta & cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2}$$

$$C_{transformed} = [xcos\theta - ysin\theta \;\; xcos\theta + ysin\theta \;\; 1] \tag{3}$$

The rotation with the angle θ is equal to 180°, it is called as horizontal flipping.

Our research uses Keras ImageDataGenerator [21] to perform the rotation transformation and color shading by changing in brightness, saturation and hue of an original bread image.

2.2. VGG-16 CNN-based Model

K. Simonyan and A. Zisserman [22] introduce VGG-16 as a large-scale image recognition model. It composes of five convolutional blocks with sixteen convolutional layers, as shown in Fig. 1, and has been trained with ImageNet dataset, composed more than fourteen million images belonging to thousand classes. Its uniform architecture consists of the following:

- Input layer that accepts image size of 224×224.
- Filters of size of 3×3 and stride, fixed to 1.
- ReLU function that operate after convolution layer.
- Max pooling layers that operate over a 2×2 pixel area with stride 2.
- Fully-Connected (FC) layers.
- softmax layer.

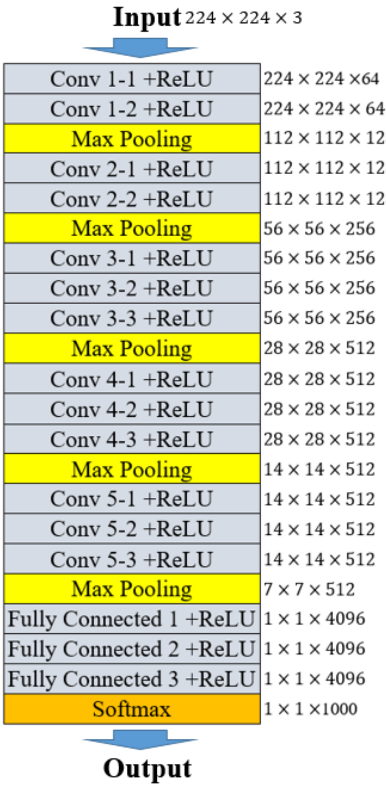

Input 224 × 224 × 3
Conv 1-1 +ReLU 224 × 224 ×64
Conv 1-2 +ReLU 224 × 224 × 64
Max Pooling 112 × 112 × 128
Conv 2-1 +ReLU 112 × 112 × 128
Conv 2-2 +ReLU 112 × 112 × 128
Max Pooling 56 × 56 × 256
Conv 3-1 +ReLU 56 × 56 × 256
Conv 3-2 +ReLU 56 × 56 × 256
Conv 3-3 +ReLU 56 × 56 × 256
Max Pooling 28 × 28 × 512
Conv 4-1 +ReLU 28 × 28 × 512
Conv 4-2 +ReLU 28 × 28 × 512
Conv 4-3 +ReLU 28 × 28 × 512
Max Pooling 14 × 14 × 512
Conv 5-1 +ReLU 14 × 14 × 512
Conv 5-2 +ReLU 14 × 14 × 512
Conv 5-3 +ReLU 14 × 14 × 512
Max Pooling 7 × 7 × 512
Fully Connected 1 +ReLU 1 × 1 × 4096
Fully Connected 2 +ReLU 1 × 1 × 4096
Fully Connected 3 +ReLU 1 × 1 × 4096
Softmax 1 × 1 ×1000
Output

Fig. 1. VGG-16 Architecture

2.3. ReLU Function

The rectifier linear unit activation (ReLU) function is an activation function that it maintains its positive value, $x \geq 0$, whereas all negative values are declared to be zero [23]. Its mathematical form is given by

$$ReLU(x) = max\{x, 0\} = \begin{cases} 0, & if\ x < 0 \\ x, & if\ x \geq 0 \end{cases} \tag{4}$$

2.4. Softmax Function

The outputs from fully connected layers are flatten into a vector of numbers. In classification task, Softmax function convert the numbers to probabilities with total is equal to one. For a vector $x \in R^n$, with n components of real number, the n-probability values (z) is defined as [23].

$$z_j = \frac{e^{x_j}}{\sum_{i=1}^{n} e^{x_j}}\ ;\ j = 1, \dots, n \tag{5}$$

2.5. Transfer Learning

Transfer learning is a technique that a model utilizes what other models have already learned. The other models can be pretrained CNN models, such as VGG, ResNet, Inception, MobileNet, etc., that have been trained with large dataset. The techniques has been widely used [24]-[25] providing model with high accuracy percentage.

There are many ways to use the pretrained models for transfer learning [26] as follows:

- Use only classified block of chosen pretrained model.
- Use some convolutional blocks for feature extraction of chosen pretrained model.
- Use weight initialization from chosen pretrained model.

2.6. Fine-tuning

Fine-tuning is a technique that a model utilizes in order to make improvement in its weight by training. Thus, the technique normally is a subsequent step to transfer learning. For example, one can unfreezes the last two layers of a pretrained CNN model for training in that its weight can get update to new dataset. The technique has been widely use as in [27]-[29].

2.7. Dropout

Dropout is a common technique to reduce overfitting phenomena that is usually occurred because of small training dataset. It randomly masks out some data points during forward processing. Typically, dropout fraction is ranged between 0.2 for input layer and 0.5 for hidden layers.

2.8. Stochastic Gradient Descent (SGD)

Stochastic Gradient Descent (SGD) is an algorithm that uses gradient, of both magnitude and direction, to adjust parameters that make the loss value converging to minima value. Its loss is the multi-class cross-entropy loss function which is used together with softmax function. Its mathematical combined form is as in Eq. (6) [30].

$$L(\hat{y}, y) = -\sum_{k}^{K} y^{(k)} log\left(\frac{e^{\hat{y}^{(k)}}}{\sum_{j-1}^{K} e^{\hat{y}^{(j)}}}\right) \tag{6}$$

where:

L is loss function.

$y^{(k)}$ is the true value of class k, ranged between 0 and 1.

$\hat{y}^{(k)}$ is the predicted value of class k, ranged between 0 and 1.

SGD is used with a large training dataset because, in each epoch, it is trained from random data in minibatch and this makes variation in SGD performance but get faster training time than that of gradient descent. Momentum is the method that makes SGD move to minimum point in faster time.

2.9. Confusion Matrix

It is the method to measure performance between predicted value and test image data. The measured performance consists of:

- Terms are defined as shown in Fig. 2.
  - True Positive (TP)
  - True Negative (TN)
  - False Positive (FP)
  - False Negative (FN)

		True class	
		Positive	Negative
Predicted class	Positive	True Positive (TP)	False Positive (FP)
	Negative	False Negative (FN)	True Negative (TN)

Fig. 2. Confusion matrix

- Accuracy is the fraction of all predicted correctly.

$$accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{7}$$

- Precision is the fraction of predicted correctly and all predicted positively.

$$precision = \frac{TP}{TP+FP} \tag{8}$$

- Recall is the fraction of predicted correctly and all true positively.

$$recall = \frac{TP}{TP+FN} \tag{9}$$

- $F1$ score is the measure of recall and precision at the same time.

$$F1 = \frac{2*recall*precision}{recall+precision} \tag{10}$$

3. Materials and Methods

Our objective is to generate a model that can be implemented in production process under the constraints of small training dataset and less diversity. With small numbers of bread production lots, five steps are performs starting with the simple CNN model through the more sophisticated ones.

3.1. Breads

Bread quality criteria are classified into three classes as overbaked, underbaked and perfect or baked status, as shown in Fig. 3. Image dataset is chosen from a production batch which consists of 1,500 pieces with 500 pieces in each criteria.

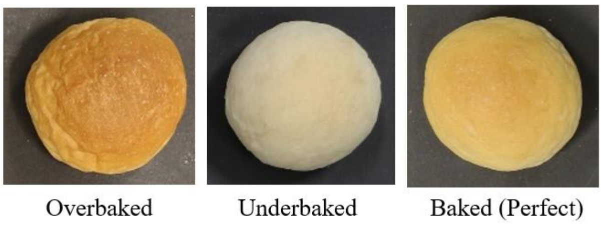


Fig. 3. Three bread browning stage status.

3.2. Image Acquisition

With our self-developed experimental equipment, as shown in Fig. 4, we can protect the environmental light penetrated into the image area and with its black color background, the total of 1,500 images of all classes can be acquired with a good condition. Then, the images are cropped and resized into 150×150 pixels as the inputs for any classification model.

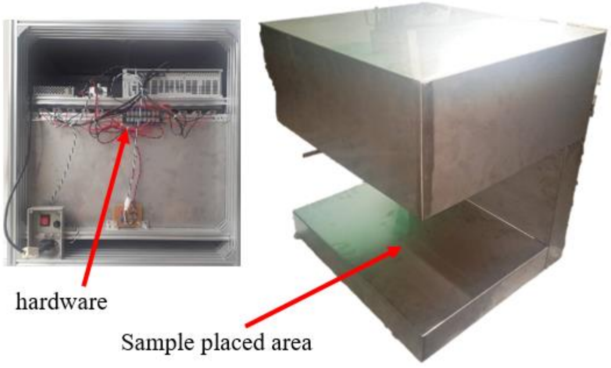


Fig. 4. Self-developed image taken equipment.

3.3. Simple CNN Model

Our research started with the randomized dataset of 900 training images with 300 of each class and 300 validation images with 100 of each class. The input image dimension is (150, 150).

As shown in Table 1, our simple CNN model composes of three convolutional layers, each with ReLU function and max pooling operation. The ReLU function will let input of positive and zero values pass their values as outputs and otherwise are passed as zero. Then the feature maps will be pulled by max pooling operation with size reduction.

The output of the 17×17 feature maps from the 3rd convolution layer are fed into a flatten layer which is used to flatten out the 128 feature maps to feed to the next two dense layers each getting along with dropout layers of 0.01 fraction. After the last dense layer, softmax function is to predict probability distribution for the three browning classes.

Categorical cross-entropy is used for loss function due to three classes of bread browning stage. Their true labels are one-hot encoded as overbaked: [1,0,0], under-baked: [0,1,0] and baked: [0,0,1]. Along with the loss function, stochastic gradient descent with momentum (SGDM) algorithm is chosen with parameters of momentum value of 0.9 and learning rate value of 0.0001. SGDM requires minibach in each training epoch which its size is 30 and the desired number of epoch is ranged between 30 and 50.

Table 1. Simple CNN model summary.

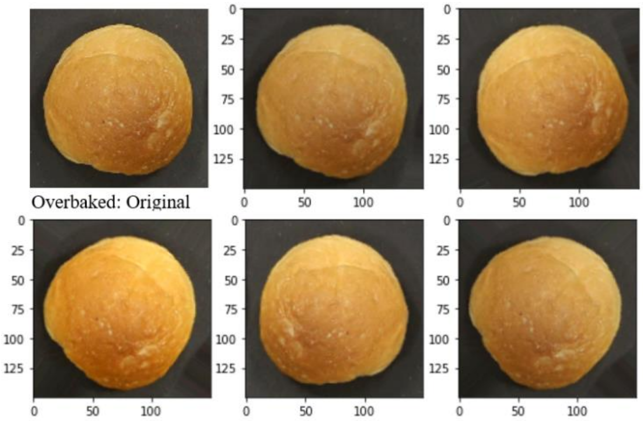
Layer (Type)	Output Shape	Para-meters	Activa-tion
Conv2D	$(None, 148, 148, 16)$	448	Relu
Maxpool2D	$(None, 74, 74, 16)$	0	-
Conv2D	$(None, 72, 72, 64)$	9280	Relu
Maxpool2D	$(None, 36, 36, 64)$	0	-
Conv2D	$(None, 34, 34, 128)$	73856	Relu
Maxpool2D	$(None, 17, 17, 128)$	0	-
Flatten	$(None, 36992)$	0	-
Dense	$(None, 512)$	18940416	Relu
Dropout	$(None, 512)$	0	-
Dense	$(None, 512)$	262656	Relu
Dropout	$(None, 512)$	0	-
Dense	$(None, 3)$	1539	Softmax

3.4. Simple CNN Model with Image Augmentation

With the production constraints, small training dataset and less diversity among batches, more artificial images are synthesized to experiment with previous simple CNN model.

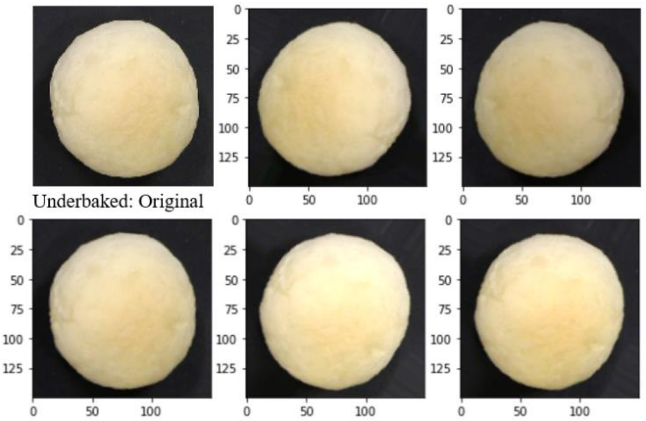
3.4.1. Image Augmentation

Keras utility function called ImageDataGenerator is an image augmented function. With our experimental bread properties of browning surface and rounded shape, the chosen augmented functions are rotation and horizontal flip for geometry and changing in hue, saturation and brightness for color shading. With many experimental works, the found suitable criteria are rotation with value of 50, horizontal flip, hue changing value of 0.5, saturation ranged values from 0.9 to 1.15 and brightness ranged values from 0.85 to 1.2. An original bread image of each class is demonstrated with the resulted artificial images are shown in Fig. 5, 6 and 7.



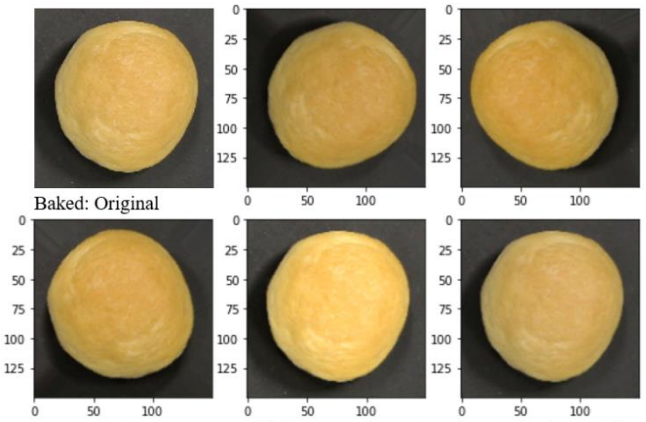
Keras ImageDataGenerator with five operations of rotation=50, horizontal flip, hue=0.5, saturation=(0.9-1.15) and brightness=(0.85-1.2).

Fig. 5. Overbaked image augmentation.



Keras ImageDataGenerator with five operations of rotation=50, horizontal flip, hue=0.5, saturation=(0.9-1.15) and brightness=(0.85-1.2).

Fig. 6. Underbaked image augmentation.



Keras ImageDataGenerator with five operations of rotation=50, horizontal flip, hue=0.5, saturation=(0.9-1.15) and brightness=(0.85-1.2).

Fig. 7. Baked image augmentation.

.

3.4.2. Simple CNN Model with Image Augmentation

In the scenario, the simple CNN model architecture, as shown in Table 1, and training parameters are still used with artificial images. For each epoch, the Keras generator randomly generates 30 artificial images. Our minibatch size is set at 30, thus 900 randomly artificial images are trained in each epoch. Hence, for total epochs of 30, total number of artificial images is 18,000 utilized for training.

Within the same epoch, the validation generator retrieves 20 original images. With validation minibatch at 15, all 300 original images are utilized in each epoch.

3.5. VGG-16 Transfer Learning

Instead of using a few convolutional layers, VGG-16 model is used for transfer learning as feature extractor. Thus all convolutional layers in Table 1 are replaced with more sophisticated VGG-16 convolutional blocks.

In the scenario, all five convolutional blocks of VGG-16 are frozen from training. Thus their weights are not updated after receiving inputs of original images. The VGG-16 architecture summary is shown in Table 2. An example of the feature map of bread browning stages of the first layer is shown in Fig. 8.

Table 2. Pretrained VGG-16 model summary.

Layer (Type)	Layer Name	Layer Trainable
Inputlayer	Input_1	False
Conv2D	Block1_conv1	False
Conv2D	Block1_conv2	False
Maxpool2D	Block1_pool	False
Conv2D	Block2_conv1	False
Conv2D	Block2_conv2	False
Maxpool2D	Block2_pool	False
Conv2D	Block3_conv1	False
Conv2D	Block3_conv2	False
Conv2D	Block3_conv3	False
Maxpool2D	Block3_pool	False
Conv2D	Block4_conv1	False
Conv2D	Block4_conv2	False
Conv2D	Block4_conv3	False
Maxpool2D	Block4_pool	False
Conv2D	Block5_conv1	False
Conv2D	Block5_conv2	False
Conv2D	Block5_conv3	False
Maxpool2D	Block5_pool	False
Flatten	Flatten	False

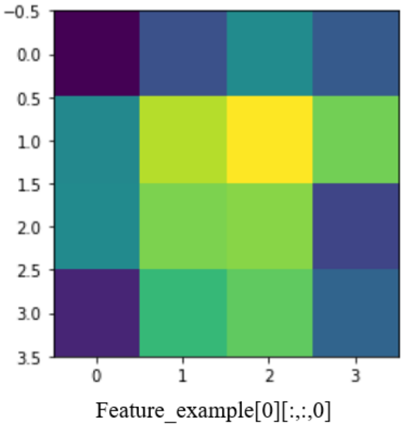


Fig. 8. Example of feature map from pretrained VGG-16.

The extracted features from VGG-16 are passed as inputs into the classification layers in that are still the same as those of simple CNN model. The classification layers compose of three dense layers and two dropout layers as shown in Table 1, written below dash line.

3.6. VGG-16 Transfer Learning with Image Augmentation

Instead of utilize frozen VGG-16 as one time feature extractor, in the scenario, frozen VGG-16 is included into training process with artificial images. Then each epoch of training, weights are updated through frozen VGG-16 too. The model summary is as shown in Table 3.

Table 3. VGG-16 transfer learning with image augmentation model summary.

Layer (Type)	Output Shape	Parameters
Model(VGG)	$(None, 8192)$	14714688
Dense	$(None, 512)$	4194816
Dropout	$(None, 512)$	0
Dense	$(None, 512)$	262656
Dropout	$(None, 512)$	0
Dense	$(None, 3)$	1539

In the training, ImageDataGenerator function is used to generate training artificial images as the same ways as of previous scenario, simple CNN model with image augmentation and other training parameters are utilized as the same as of simple CNN model.

3.7. VGG-16 Transfer Learning and Fine-tuning with Image Augmentation

As known, the upper convolutional layers of VGG-16 learns small local patterns whereas the deeper convolutional layers will depend on the more complex and larger patterns. Then, the way to fine-tuning is to unfreezing some deeper blocks, block-4 and block-5 as shown in Table 4. Thus, their weights will get updated in each epoch during training.

Table 4. Pretrained VGG-16 with last two trainable layers model summary.

Layer (Type)	Layer Name	Layer Trainable
Inputlayer	Input_1	False
Conv2D	Block1_conv1	False
⋮	⋮	⋮
Maxpool2D	Block3_pool	False
Conv2D	Block4_conv1	True
⋮	⋮	⋮
Conv2D	Block5_conv3	True
Maxpool2D	Block5_pool	True
Flatten	Flatten	True

Instead of utilizing frozen VGG-16, the scenario includes VGG-16 with fine-tuning of block-4 and block-5 into training process with artificial images. Then each epoch of training, weights are updated through deeper layers of VGG-16 too. The model summary is the same as in Table 3. As in previous scenario, ImageDataGenerator function is used to generate training artificial images as in the same way as well as other training parameters do.

4. Experimental Results

As mentioned earlier, the experiment uses only one production batch of bread that contains small dataset of 900 training images with 300 each class of overbaked, underbaked and perfected or baked and 300 validation images with 100 each class.

4.1. Simple CNN Model

As shown in Fig. 9, both training and validation accuracies approximately reach 90%, 96% at epoch 10 and 20, respectively and move steadily to 96% at epoch 30. Both training and validation curves are closed together in which reflect no overfitting phenomena. This is because the images in a production batch have steady color shading and clear criteria for each class. However, production output performance level should be higher than 98% or above.

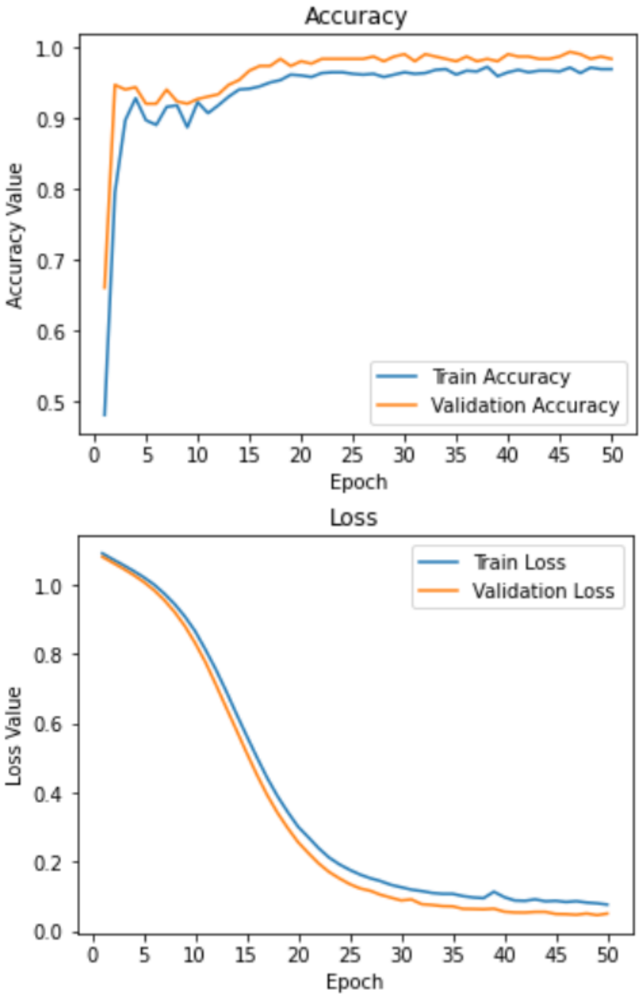


Fig. 9. Simple CNN model measured performance.

The simple CNN model loss in both training and validation are high approximately 86%, 30% at epoch 10 and 20, respectively, and reduces sharply to 13% at epoch 30. During the initial stage of training, loss percentage is still high and take several training epochs before going down. That is reflected to small amount of training dataset.

4.2. Simple CNN model with Image Augmentation

As shown in Fig. 10, the result of training accuracy is poorer than those of the simple CNN model with the original images whereas the loss performance are still remain the same. Accuracy percentages of both training and validation approximately reach 77% and 92% at the epoch of 10 and 20, respectively, and approximately stable 95% at the epoch 30. This is because the training artificial images contain both the various physical and the color shading properties and much more amount of training images. Both training and validation curves are closed together that reflect no overfitting phenomena.

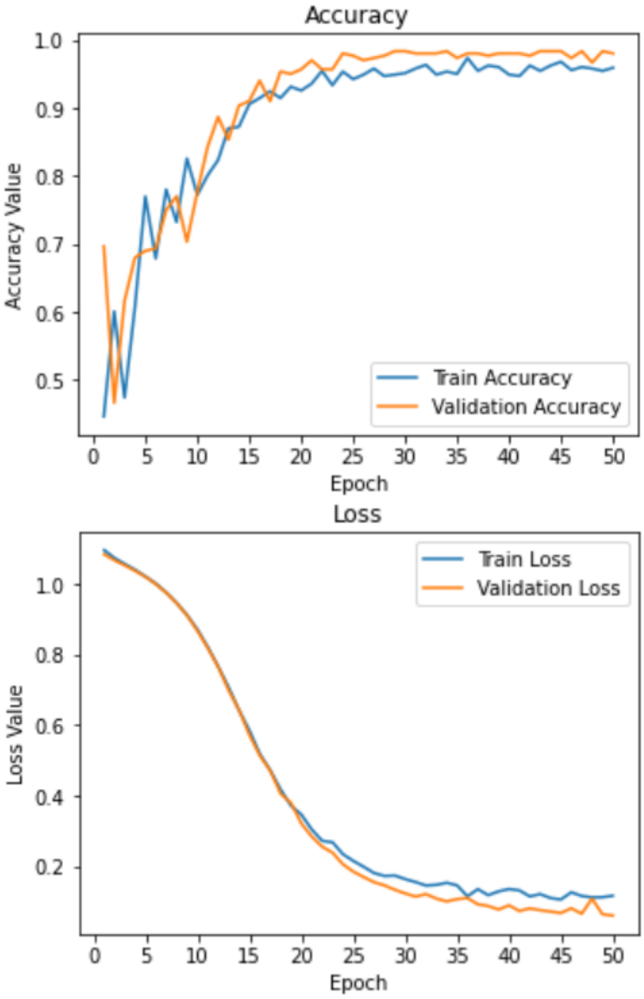


Fig. 10. Simple CNN with image augmentation model measured performance.

The loss performance of both the training and validation are still high approximately 87%, 35% at epoch of 10 and 20, respectively, and still approximately unstable

16% at epoch 30. The loss patterns are no different than those of simple CNN model.

4.3. VGG-16 Transfer Learning

As shown in Fig. 11, the result of training accuracy approximately reach 84%, 94% at epoch 10 and 20, respectively, and is approximately stable 95% at epoch 30. Compare the result to the training accuracy of simple CNN model, the VGG-16 transfer learning model provide similar training pattern in which rising sharply at beginning period. This is because of good quality of original images and clear quality criteria. However, the VGG-16 transfer learning curve provide more fluctuation along the training path, this is because of the feature extractor VGG-16 are capable to extract much more features than those of simple CNN model.

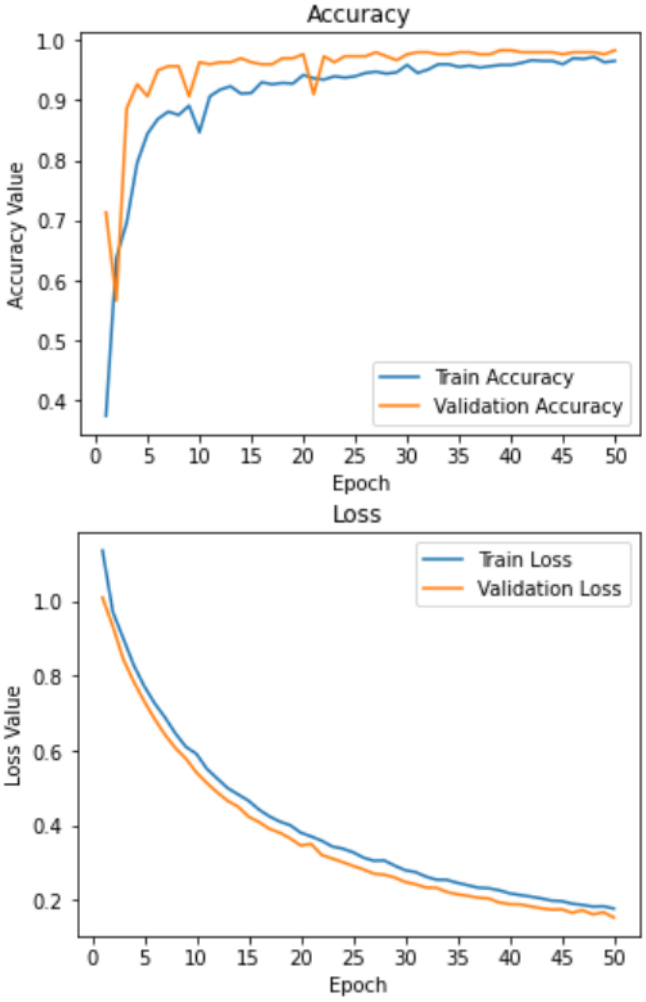


Fig. 11. VGG-16 transfer learning model measured performance.

However, there is an obvious improvement in the training loss. The VGG-16 transfer learning model loss both training and validation are approximately 59%, 38% at the epoch 10 and 20, respectively, and still approximately unstable 21% at epoch 30. During the initial stage of training, the loss is much more reduced than that of simple CNN model and it tends to continuously reducing with the increasing epochs. The obvious loss reduction during the initial stage of training comes from more extracted features contributed to the training model.

4.4. VGG-16 Transfer Learning with Image Augmentation

As shown in Fig. 12, the training accuracy curve is in the same pattern as of VGG-16 transfer learning model but its curve containing more fluctuation along the training path. This is because VGG-16 with all frozen layers is in training model in which training with artificial image. Accuracy percentage of both training and validation reach 85%, 89% at the epoch 10 and 20, respectively, and stable 92% at epoch 30.

However, the model loss patterns are the same as those of VGG-16 transfer learning model.

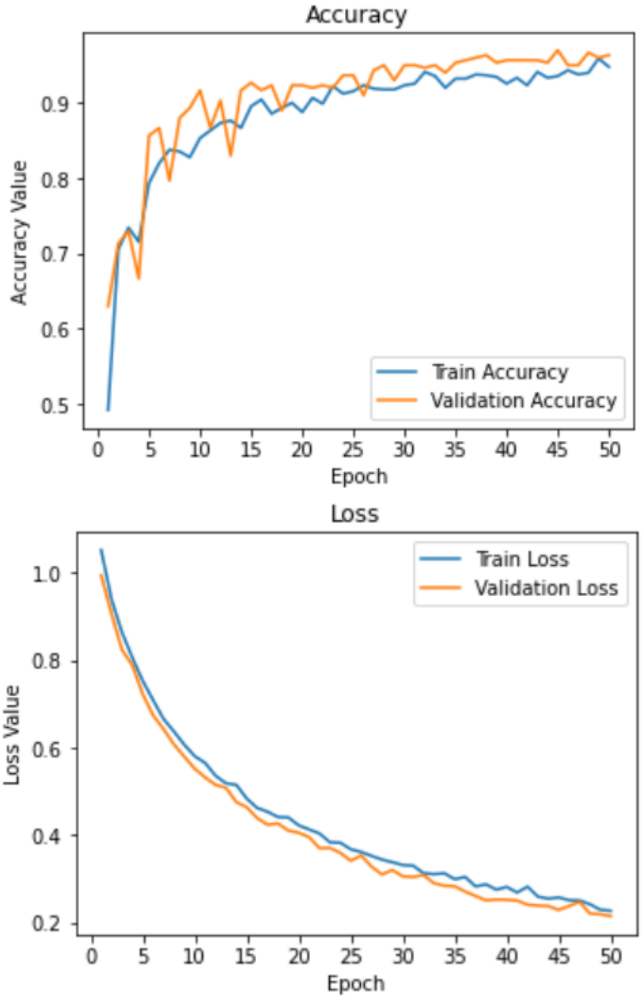


Fig. 12. VGG-16 transfer learning with image augmentation model accuracy and loss performance.

4.5. VGG-16 Transfer Learning and Fine-tuning with Image Augmentation

As shown in Fig. 13, the training accuracy curve is in the same pattern as of VGG-16 transfer learning model which is risen sharply at initial epochs. But its curve containing small fluctuation along the training path. This is because VGG-16 with fine-tuning of unfrozen block-4 and block-5 is capable to more updating on weights during training with artificial image. Accuracy percentage of both training and validation reach sharply about 95% and 97% at the epoch 5 and 10, respectively, and approximately stable 99% at epoch 30.

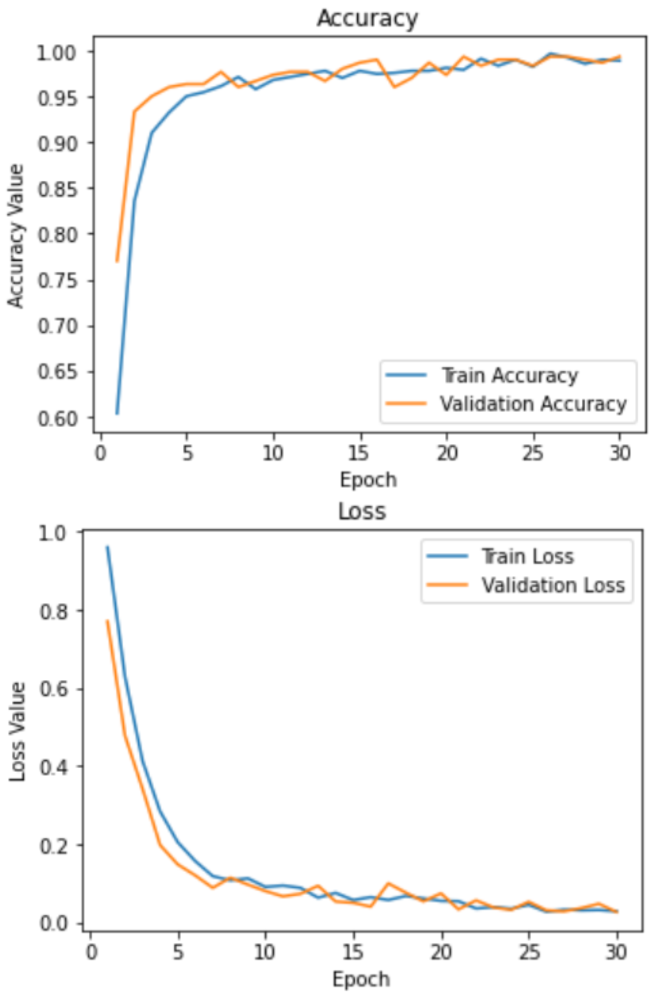


Fig. 13. VGG-16 transfer learning and fine-tuning with image augmentation model accuracy and loss performance.

There is also obvious improvement in the training loss during initial training period. The loss of both training and validation are down sharply 21% and 9% at epoch 5 and 10, respectively, and still approximately stable 3% at epoch 30.

Among the training accuracy and loss performance of all five models, one can obviously conclude that the final model of VGG-16 transfer learning and fine-tuning with image augmentation can provide the best training accuracy and loss performance in small number of epochs of approximately 98.89% and 2.86%, respectively, at epoch 30.

4.6. Model Training Time

The training time to achieve expected model accuracy is important in order to implement a model in real production process. With a standard personal computer should be able to execute the algorithm in production operation. As shown in Table 5, all five models provide training times ranging between 0.19 minutes and 1.40 minutes for average time per epoch and ranged between 5.60 minutes and 42.13 minutes for 30 epochs. In the case of implementation in a production process, in which accuracy acceptance level should be above 98%, the final model, VGG-16 transfer learning and fine-tuning with image augmentation should be selected. This is because its training time is on the average of 1.4 minutes per epoch and 42.13 minutes per 30 epochs.

Table 5. Comparison of model training times.

Model	Time per epoch (minutes)	Time per 30 epoch (minutes)
Model-1	0.19	5.60
Model-2	0.23	6.78
Model-3	0.02	0.50
Model-4	0.69	20.75
Model-5	1.40	42.13

4.7. Model Predicted Performance

300 test images with 100 of each class are fed into each predictive model. Confusion matrices are used to measure classification performaces as summary in Table 6.

Table 6. Measured performance report.

Model	Criteria	Stages		
		Over baked	Under baked	Baked
Model-1	Samples	100	100	100
	Precision	0.90	1.00	0.98
	Recall	0.98	1.00	0.89
	F1 score	0.94	1.00	0.93
	Accuracy			0.96
Model-2	Samples	100	100	100
	Precision	0.94	0.99	0.96
	Recall	0.96	1.00	0.93
	F1 score	0.95	1.00	0.94
	Accuracy			0.96
Model-3	Samples	100	100	100
	Precision	0.99	0.98	0.96
	Recall	0.98	0.98	0.97
	F1 score	0.98	0.98	0.97
	Accuracy			0.98
Model-4	Samples	100	100	100
	Precision	0.96	1.00	0.86
	Recall	0.97	0.88	0.96
	F1 score	0.97	0.94	0.91
	Accuracy			0.94
Model-5	Samples	100	100	100
	Precision	1.00	1.00	1.00
	Recall	1.00	1.00	1.00
	F1 score	1.00	1.00	1.00
	Accuracy			1.00

All models provide high classification accuracies above 94%. Model-5 or VGG-16 transfer learning and fine-tuning with image augmentation model can achieve both 100% classification accuracy and also other measured criteria. Its confusion matrix is shown in Fig. 14.

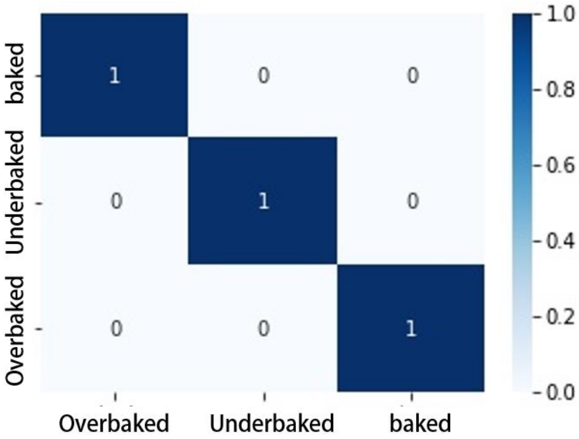


Fig. 14. Confusion Matrix of final model: VGG-16 transfer learning and fine-tuning with image augmentation.

4.8. Consideration of Training Sample Size

The final model of VGG-16 transfer learning and fine-tuning with image augmentation can provide the best training accuracy 98.89% and loss 2.86% at epoch 30 with its training sample size of 900 images. However, there is a curious question in that sample size can be less than 900 images and what the size should be. Thus, with the final model, the training processes are experimented with various training sample sizes ranged from 360 to 900 images, their training accuracies are compared as shown in Fig. 15. The more training sample sizes are, the more training accuracies increase and the more training loss decrease with less fluctuation. Both fluctuation in training accuracy and loss cause from small size of training images. At epoch 30, their training accuracies are 98.89%, 98.77%, 98.96%, 99.32%, 98.39%, 98.22% and 95.83% with training sample sizes of 900, 810, 720, 630, 540, 450 and 360 images, respectively.

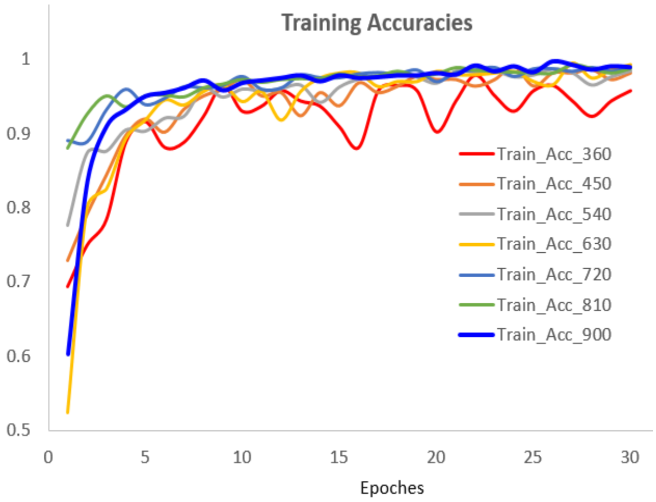


Fig. 15. Training accuracies with various sample sizes.

As the final model can provide various best accuracies above 98%, two samples t-tests are performed with the best accuracy of 900 training sample size with 95% confidence for both data of epochs 21-30 and epochs 26-30. Their p-values are shown in Table 7. There are two training sample sizes of 810 and 720 images that their p-values are above 0.05. Thus, both sample sizes are able to provide average training accuracies as the same as of 900 sample size with 95% confidence.

Table 7. Two samples t-tests with the best accuracy of 900 sample size with 95% confidence.

Training sample size	Data: epoch 21-30		Data: epoch 26-30	
	Average accuracy	p-value	Average accuracy	p-value
900	0.98789	-	0.99068	-
810	0.98533	0.117	0.98620	0.077
720	0.98593	0.181	0.98716	0.079
630	0.98119	0.034	0.98320	0.137
540	0.97982	0.003	0.97866	0.018
450	0.97731	0.003	0.98132	0.007
360	0.95000	0.000	0.94720	0.002

5. Conclusion

The paper presents the recognition and classification methods for bread browning stages with small dataset and less diversities of surface appearance. These two constraints and time consumption are always the obstacles to implement any network model in real operation on the production line. For the experiment, 1,500 pieces of breads are withdrawn from only one production batch with three classes of bread quality criteria as overbaked, underbaked and perfected or baked.

In order to achieve a recognition and classification model of bread browning stage with high accuracy, the research breaks down tasks into five steps starting model with a few simple convolutional layers, adding image augmentation and utilizing pretrained VGG-16 model for transfer learning and fine-tuning.

Image augmentation, thru the Keras ImageDataGenerator function, helps to generate a number from 900 original to 18,000 artificial images. The suitable parameters to simulate bread browning stages are rotation and horizontal flip to propagate surface texture in all angles and the color shading diversities by changing hue, saturation and brightness. With more training images, the training accuracy curve obtain fluctuation along its path.

The pretrained VGG-16, frozen all convolutional layers, is used as feature extractor both external extraction, fed as inputs, and internal extraction during training iteration loops. Because of the frozen VGG-16 architecture, more features are extracted providing higher accuracies and much more loss reduction especially in the initial epochs.

Adding fine-tuning by unfrozen the last two convolutional blocks, block-4 and block-5, of VGG-16, the weights can be updated with artificial images in each training epochs. Its accuracy pattern is much more smooth and rises sharply during initial epochs as well as loss pattern declines sharply too.

VGG-16 transfer learning and fine-tuning with image augmentation is the best model that can achieve very high training accuracy approximately 98.89% and very low training loss 2.86% at a small number of epochs 30 of training bread size at 900 images. Its training time is average of 1.40 minutes per epoch and 42.13 minutes per 30 epochs. Furthermore, its test or predicted classification accuracy is at 100%. Based on its very high accuracy performance and small model training time, it is possible to implement the model on the production line. However, with two sample t-test, there are two more training sample sizes of 720 and 810 images that are able to provide the same training accuracy at 95% confidence.

The type of bread, bun, is the basic shape and color that one can transfer the recognition and classification knowledge to other similar bread types, such as biscuit, cookies, etc., that utilize the bread browning stage for automated inspection.

Acknowledgement

I would like to grateful thanks of supports from the Research and Academic Services Administration Division, the Faculty of Engineering at Khamphaeng Saen and all the project team members at the Kasetsart Univerisity, Khamphaeng Saen Campus for their research funds and valuable recommendation.

References

[1] Montrose Inc. “Vision Inspection of Buns, Breads and Baked Goods.” https://montrose-tech.com/solutions/buns-reads-and-baked-goods/ (accessed 15 January 2022).

[2] Iris Solution Incorporated. “Bread Inspection System.” https://www.irissi.com/bread.html (accessed 15 January 2022).

[3] Sesotec. “Detecting Contaminants in bread and baked goods.” https://www.sesotec.com/apac/en/industries/sub/bread-and-baked-goods (accessed 15 January 2022).

[4] J. Yin, S. Hameed, L. Xie, and Y. Ying, “Non-destructive detection of foreign contaminants in toast bread with near infrared spectroscopy and computer vision techniques,” *J. Food Meas. Charact.*, vol. 15, no. 1, pp. 189-198, Feb. 2021.

[5] M. Abdullah, S. A. Aziz, and A. M. D. Mohamed, “Quality inspection of bakery products using a color-based machine vision system,” *J. Food Qual.*, vol. 23, no. 1, pp. 39-50, Mar. 2000.

[6] S. Nashat and M. Z. Abdullah, “Multi-class color inspection of baked foods featuring support vector machine and Wilk’s λ analysis,” *J. Food Eng.*, vol. 101, no. 4, pp. 370-380, Dec. 2010.

[7] S. Nashat, A. Abdullah, S. Aramvith, and M. Z. Abdullah, “Support vector machine approach to real-time inspection of biscuits on moving conveyor belt,” *Compute. Electron. Agric.*, vol. 75, no. 1, pp. 147-158, Jan. 2011.

[8] N. S. Eilani bt. and A. R. Sulaiman, “Texture analysis for biscuit using wavelet,” in *International Conference on Electrical Engineering and Informatics*, Selangor, MY, 2009, pp. 52-55.

[9] P. Parikh, P. Mehta, and C. K. Modi, “Non-destructive quality evaluation of chocolate chip cookies,” in *International Conference on Communication Systems and Network Technologies*, Jamu, IN, 2011, pp. 694-698.

[10] Y. Wang, C. Shi, C. Zhang, and Q. Liao, “A real-time computer vision system for biscuit defect inspection,” in *International Conference on Computer Vision Theory and Applications*, Berlin, DE, 2015, pp. 531-536.

[11] O. Paquet-Durand, D. Solle, M. Schirmer, T. Becker, and B. Hitzmann, “Monitoring baking processes of bread rolls by digital image analysis,” *J. Food Eng.*, vol. 111, no. 2, pp. 425-431, Jul. 2012.

[12] W. D. S. Cotrim, V. P. R. Minim, L. B. Felix, and L. A. Minim, “Short convolutional neural networks applied to the recognition of the browning stages of bread crust,” *J. Food Eng.*, vol. 277, pp. 1-8, Jul. 2020.

[13] M. Olson, A. J. Wyner, and R. Berk, “Modern neural networks generalize on small data sets,” in *Conference on Neural Information Processing Systems*, Montreal, CA, 2018, pp. 1-10.

[14] S. Feng, H. Zhou, and H. Dong, “Using deep neural network with small dataset to predict material defects,” *Material and Design*, vol. 162, pp. 300-310, Jan. 2019.

[15] L. Chen, Z. Jiang, and Z. Wang, “Image recognition based on convolutional neural network with small data set,” in *PhotonIcs & Electromagnetics Research Symposium,* Rome, IT, 2019, pp. 816-820.

[16] S. Wallelign, M. Polceanu, T. Jemal, and C. Buche, “Coffee grading with convolutional neural networks using small datasets with high variance,” in *Conference of Computer Graphics, Visualization and Computer Vision*, Slovany. CZ, 2019, pp. 113-120.

[17] L. Sun, K. Liang, Y. Song, and Y. Wang, “An improved CNN-based apple appearance quality classification method with small samples,” *IEEE Access*, vol. 9, pp. 68054-68065, 2021.

[18] D. Li, W. Xie, B. Wang, W. Zhong, and H. Wang, “Data augmentation and layered deformable mask R-CNN-based detection of wood defect,” *IEEE Access*, vol. 9, pp. 108162-108174, 2021.

[19] R. Poojary, R. Raina, and A. K. Mondal, “Effect of data-augmentation on fine-tuned CNN model performance,” *IAES International Journal of Artificial Intelligence*, vol. 10, no. 1, pp. 84-92, Marใ 2021.

[20] R. Chityala and S. Pudipeddi, “Affine transformation,” in *Image Processing and Acquisition using Python*, 2nd ed. Boca Raton, FL: CRC Press, 2021, ch. 6, sec. 6.2.2, pp. 123-136.

[21] Tensorflow. "tf.keras.preprocessing.imageDataGenerator" https://www.tensorflow.org/api_docs/python/tf/keras/preprocessing/image/ImageDataGenerator (accessed 15 January 2022).

[22] K. Simonyan and A. Zisserman, "Very deep convolutional neural networks for large-scale image recognition," in *International Conference on Learning Representations*, Sandiago, CA, 2015, pp. 1-14.

[23] O. Calin, "Activation functions," in *Deep Learning Architectures*ใ Cham, CH: Springer, 2020, ch. 2, sec. 2.1, pp. 21-40.

[24] M. Alencastre-Miranda, R. M. Johnson, and H. I. Krebs, "Convolutional neural networks and transfer learning for quality inspection of different sugarcane varieties," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 2, pp. 787-794, 2021.

[25] M. Ahmad, M. Abdullah, H. Moon, and D. Han, "Plant disease detection in imbalanced datasets using efficient convolutional neural networks with stepwise transfer learning," *IEEE Access*, vol. 9, pp. 140565-140580, 2021.

[26] J. Brownlee, "How to use pre-trained models and transfer learning," in *Deep Learning for Computer Vision*. 2019, ch. 18, sec. 18.4, pp. 192-196.

[27] X. Zhang, F. Yan, Y. Zhuang, H. Hu, and C. Bu, "Using an ensemble of incrementally fine-tuned CNNs for cross-domain object category recognition," *IEEE Access*, vol. 7, pp. 33822-33833, 2019.

[28] Z. Situ, S. Teng, H. Liu, J. Luo, and Q. Zhou, "Automated sewer defects detection using style-based generative adversarial networks and fine-tuned well-known CNN classifier," *IEEE Access*, vol. 9, pp. 59498-59507, 2021.

[29] G. Vrbancic and A. V. Podgorelec, "Transfer learning with adaptive fine-tuning," *IEEE Access*, vol. 8, pp. 196197-196211, 2020.

[30] I. Zafar, G. Tzanidou, R. Burton, N. Patel, and L. Araujo, "CNN model architecture," in *Hands-On Convolutional Neural Networks with TensorFlow*, Birmingham, UK: Packt, 2018.

Prapassorn Tantiphanwadi holds a Bachelor of Science degree in Physics from Chulalongkorn University, Bangkok, Thailand and a Master of Science degree in Physics from Utah State University, Utah, U.S.A in 1998. She also earns two degrees from Kasetsart University, Bangkok, Thailand: Master of Engineering in Industrial Production Technology and Doctor of Engineering in Industrial Engineering in 2018.

Her professional career is Quality and Six Sigma Management with over 20 years of experience working with famous factories of electronics and automotive fields. She has been certified as Six Sigma Master Blackbelt and Sony Six Sigma trainer and currently extend knowledge to lean automation with IoT environment. Currently, she is ranked a lecturer and serves as a faculty member in the Department of Industrial Engineering, Faculty of Engineering at Khamphaen Saen, Kasetsart University. Her interested researches are on automated inspection system utilizing image processing, machine learning and deep learning in all industrial fields of food, electronics, automotive, etc. Other professional fields are industrial statistics, such as statistical quality control, design of experiment and data science for industry and logistics.

She received best track paper award in Statistics and Optimization in the Sixth International Conference on Industrial Engineering and Operations Management, Kuala Lumpur, Malaysia March 8-10, 2016.

Krisanun Malithong obtained his first degree in mechanical engineering (B.Eng) from King Mongkut's University of Technology North Bangkok, Bangkok, Thailand in 2001. He received a master of engineering in mechanical engineering (M.Eng.) in 2005 and a doctor of philosophy (Ph.D.) in mechanical engineering from Chulalongkorn University, Bangkok, Thailand in 2010.

He had been employed in position of research and development engineer at The Regional Center of Robotics Technology as a part of the department of mechanical engineering, faculty of engineering, Chulalongkorn University in 2010. The research directions cover the research of the control of mechanical system, including the field serveying device, manufacturing, automation, and CAD/CAM/CAE technology. He is currently a lecturer in the department of food engineering, faculty of engineering at Khamphaeng Saeng, Kasetsart University. His research interests include food processing machinery, automatics control and robotics in food industry, and Community development by BCG economy and Sustainable development.