A Transfer Learning-based Deep Convolutional Neural Network Approach for White Shrimp Abnormality Classification

Korawit Orkphol[1], Kathawach Satianpakiranakorn[2*], Jidapa Chaihuadjaroen[3], and Tamnuwat Valeeprakhon[4]

[1,2,3,4]Department of Computer Engineering, Faculty of Engineering at Sriracha, Kasetsart University Sriracha Campus, Chonburi, Thailand

Email: korawit@eng.src.ku.ac.th, kathawach@eng.src.ku.ac.th*, jidapa@eng.src.ku.ac.th, tamnuwat@eng.src.ku.ac.th

Received: August 28, 2024 / Revised: October 10, 2024 / Accepted: October 20, 2024

***Abstract*—Shrimp transportation frequently leads to product damage, necessitating a sorting system to identify and remove compromised shrimp prior to processing. This research aims to develop a transfer learning-based deep convolutional neural network system capable of accurately categorizing shrimp into seven classes: Complete body, crunched head, head loss, head loss with remaining chin, cut tail, torn in half, and total crunched. A dataset comprising 405 color shrimp images, each with dimensions of 1,920 x 1,080 pixels, was augmented using geometric transformations to expand the dataset to 6,480 images. These augmented images were then employed to train four state-of-the-art transfer learning-based models (NasNetLarge, InceptionResNetV2, EfficientNetV2L, ConvNeXtXLarge) from Keras Applications. These models also were subsequently compared to a baseline CNN.**

Results demonstrate that the ConvNeXtXLarge model outperformed the others, achieving the highest accuracy (95%), precision (0.96%), recall (0.95%), and F1-score (0.95%), underscoring its superiority in shrimp damage classification. An analysis of misclassifications revealed potential confusion between certain damage classes, suggesting areas for future refinement to enhance the model's ability to differentiate between similar types of damage.

***Index Terms*—Anomalies Who Le White Shrimp, Deep Learning, Image Classification, Transfer Learning**

I. Introduction

Vannamei shrimp is one of the important export products of Thailand and a major economic commodity. Thailand has the potential to process shrimp into various products to meet market demand. Thailand's main export markets included the United States, Japan, the People's Republic of China, and the Republic of Korea. According to the Thailand Foreign Agricultural Trade Statistics 2023 by the Agricultural Information Center, the Office of Agricultural Economics, Ministry of Agriculture and Cooperatives. In 2023, the income from shrimp products accounted for 2.52% of the value of agricultural export products, bringing in the eighth-highest income of agricultural products and overall products. For fresh, chilled, or frozen shrimp, the export volume was 15,273 metric tons, with an export value of 3,794 million THB, accounting for 58.52% of all shrimp exports [1].

To maintain the freshness of white shrimp before entering the processing process, all steps from catching shrimp to transporting shrimp from farmers to the production line must be done rapidly. The process will start with farmers catching shrimp at nighttime to prevent damage from heat, using a net to scoop shrimp out of the pond. Then, the size sorting staff from the shrimp farm or the factory will sort the sizes. After sorting, the shrimp will be placed in baskets, as shown in Fig. 1, and transported to the factory by a refrigerated truck. However, during transportation, shrimp will be packed in baskets and stacked in layers to allow for large quantities of goods on each transportation trip. In this step, some shrimp on top of the baskets that are stacked on top of each other may be damaged. In addition, shrimp at the bottom of the basket may be crushed by the weight of the shrimp above, which may cause damage. When it arrives at the factory, the shrimp will be sorted by taking the undamaged shrimp to the production line. Damaged shrimp will be sorted into grades for processed snacks and animal feed grades. However, even though the shrimp in the baskets are fresh, some of the shrimp that were damaged during transportation, such as shrimp heads that have fallen off or bodies that have broken, will be damaged. In this step, factory workers will have to take the shrimp out of the baskets and use manual labor to select the damaged shrimps and send the undamaged shrimps to the processing line. This step will require all manual labor.

Fig. 1. Shrimp containers for transportation

The visual characteristics of shrimp, such as color, texture, size, and shape, are key factors that influence consumer purchase decisions. They are also important indicators for classifying products according to different quality levels. Visual inspection is the most common method for quality control of shrimp before entering the production line. It is a common practice in the seafood industry. However, visual inspection is prone to errors [2], [3]. Repetitive work can intensely decrease human efficiency. In addition, there are problems of rising wages and labor shortages. Therefore, there is a demand to use automated systems instead of human labor.

The most effective technique for solving these issues in the food business is operational automation. To automate all processes; Artificial Intelligence (AI), Machine Learning (ML), and Deep Learning (DL) technologies are applied [4]. Machine learning and Artificial Intelligence (AI) can be used for the transformation of food safety and quality data management [5]. Machine learning and deep learning are applied to automate detection in many shrimp products. For peeled shrimp products, Valeeprakhon [6] proposed a Deep Convolutional Neural Network based on VGG-16 for the classification abnormalities of peeled shrimp. Yu [7] proposed YOLO-4 for classifying peeled and shell-on shrimp detection. For de-heading, back cutting or butterfly cutting and removing the vein, Thanasarn [3] proposed automated recognition of deveined shrimps with linear discriminant analysis and Support Vector Machine (SVM) based on grayscale image parameters. To measure the freshness of shrimp, Zhang [8] proposed deep learning detection of shrimp freshness via smartphone camera picture which is an easy and low-cost method to detect shrimp freshness. Wang [9] proposed a deep learning-based freshness assessment method for chilled red shrimp using a shrimp net, which is a lightweight module. The method was based on a Convolutional Neural Network (CNN) architecture that was trained on a data set of images of frozen red shrimp with different levels of freshness. Prema and Visumathi [10] proposed shrimp Freshness detection based on CNN compared with Hybrid CNN and SVM with DCGAN which is proposed by Radford et al. and Yeh et al for data augmentation and found that Hybrid CNN and SVM with GAN is better than classic CNN model. Most researches on fresh shrimp are related to the shrimp products industry by using image classification with Machine Learning (ML) which is focused on size, freshness, and quality inspection of shrimp products after passing the production line. No research has been found to distinguish shrimp with incomplete conditions before entering the production line.

The use of ML is to select disqualified shrimps before entering the production line. If we can distinguish the types of shrimp that are rejected by shape, for example, shrimp that is severely damaged, such as broken bodies or mushy bodies, should be classified as feed grade. Shrimp with head loss, head loss with only the shrimp chin, or crunch head shrimp can be sorted as shrimp grades higher than feed grade. This will increase the value of shrimp more than all rejected shrimp that were sorted as feed grade. In this research, four top-1 accuracy models with over 80% accuracy from Keras Applications [11], namely NasNetLarge [12], InceptionResNetV2 [13], EfficientNetV2L [14], and ConvNeXtXLarge [15], were selected for testing on a dataset with CNN as the baseline. The objective was to classify shrimps into seven categories: complete body shrimp, crunch head shrimp, shrimp with head loss, head loss shrimp with the remaining chin, cut tail shrimp, torn in half shrimp (upper half and lower half), and total crunched shrimp, as shown in Table I. The experimental results revealed that ConvNeXtXLarge achieved the highest accuracy of 95%.

TABLE I
CLASSES OF SHRIMP DATASETS

Class Label	Type	Status	Example
Complete	Complete body shrimp	Normal	
Busted Head	Crunch head shrimp	Abnormal	
HeadOff	Shrimp with head loss	Abnormal	
Chin	Head loss shrimp with remaining chin	Abnormal	
TailOff	Cut tail shrimp	Abnormal	
Half	Torn-in-a-half shrimp	Abnormal	 
Crushed	Total crunched shrimp	Abnormal	

II. METHODOLOGY

In this research, 405 images of normal and disqualified shrimps, with dimensions of 1,920 x 1,080 pixels, were used. The images were divided into 7 categories:

1. Complete body shrimp (80 images)
2. Crunched head shrimp (56 images)
3. Shrimp with head loss (54 images)
4. Head loss shrimp with remaining chin (56 images)
5. Cut tail shrimp (43 images)
6. Torn in half shrimp (45 images)
7. Total crunched shrimp (71 images)

The images were then subjected to data pre-processing before being used to train a transfer learning model. The pre-processing steps were as follows:

A. Data Preprocessing and Geometric Transformation Augmentation

1) Geometric Transformation Augmentation

Due to the limited number of input images, geometric transformation augmentation was employed. This research explores two geometric transformation techniques, image reflection, and image rotation, to generate a new set of images. In the first step, the original image is rotated by predefined angles: 0 degrees, 45 degrees, 90 degrees, 135 degrees, 180 degrees, 225 degrees, 270 degrees, and 315 degrees. Rotation is a common type of geometric transformation used for image data augmentation [16].

However, the input image is rectangular as shown in Fig. 2 (I_1), which makes it impossible to rotate the image. Therefore, image preprocessing is required before the augmentation process. The preprocessing steps are as follows:

1. Convert the input image to a binary image (I_2).
2. Remove noise from the image using opening and closing operations (I_3).
3. Find the center of the image by contouring the objects in the image, selecting the largest object, cropping the largest object to a square, and finding the center of that square (I_4).
4. Once the center of the image is found, expand the green background to make the image size 1500 x1500 pixels.
5. Flip the image (I_5).
6. The original image and the flipped image are fed into the geometric transformation augmentation.

Due to the limited number of input images, geometric transformation augmentation was employed. This research explores two geometric transformation techniques, image reflection and image rotation, to generate a new set of images. In the first step, the original image is rotated by predefined angles: 0 degrees, 45 degrees, 90 degrees, 135 degrees, 180 degrees, 225 degrees, 270 degrees, and 315 degrees. Rotation is a common type of geometric transformation used for image data augmentation [16]. The rotation equation is given in (1). The and represent the new positions of each pixel after the rotation process, the x and y pairs of coordinates denote the raw image, and represents the rotation angle.

$$\begin{bmatrix} f_x \\ f_y \end{bmatrix} = \begin{bmatrix} \cos\varphi & -\sin\varphi \\ \sin\varphi & \cos\varphi \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix} \qquad (1)$$

This step produces a total of eight images. Subsequently, the original image is reflected across the y-axis and rotated by the same set of predefined angles as shown in Fig. 2 (I_6-I_8). This second step generates an additional eight images, resulting in a total of sixteen images, as shown in Fig. 3.

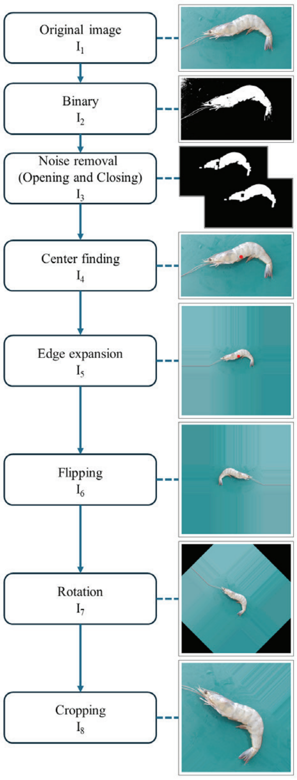

Fig. 2. Overview of the preprocessing method

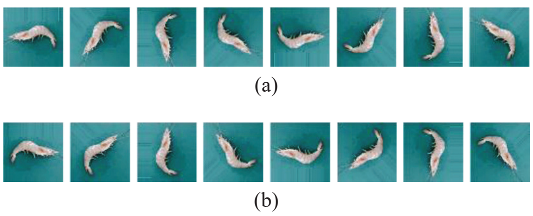

Fig. 3. (a) Shows rotations of the original image at 0°, 45°, 90°, 135°, 180°, 225°, 270°, and 315°. (b) Shows the reflected image with rotations at 0°, 45°, 90°, 135°, 180°, 225°, 270°, and 315°.

2) Dataset

This dataset consists of 6,480 color images of shrimp, categorized into seven classes:

1. Complete body shrimp 1,280 images.
2. Crunch head shrimp 896 images.
3. Shrimp with head loss 864 images.
4. Head loss shrimp with remaining chin 896 images.
5. Cut tail shrimp 688 images.
6. Torn in half shrimp 720 images.
7. Total crunched shrimp 1,136 images.

All images' dimensions of 1,920 x 1,080 pixels were resized to 224 x 224 pixels to match the input requirements of the model and normalized to a range of 0-1. The dataset was randomly shuffled and divided into training (70%) and testing (30%) sets.

B. Classical CNNs and Transfer Learning Techniques

Convolutional Neural Networks (CNNs) are a category of deep neural networks specifically engineered for the processing of structured grid data, such as images[17], [18]. These networks are comprised of multiple layers, including convolutional layers, pooling layers, and fully connected layers [19].

CNNs have established themselves as a foundational framework in the domains of deep learning and computer vision due to several compelling factors: their computational efficiency, capability to effectively process image data, inherent translation invariance, hierarchical feature learning, demonstrated effectiveness, scalability, and robust community support [20], [21].

Transfer learning is a machine learning technique whereby a model developed for a specific task is repurposed as the initial framework for a model addressing a subsequent task [22], [23]. This methodology is particularly advantageous in scenarios where the subsequent task is constrained by limited data. By leveraging transfer learning, the development of high-performing models can be achieved with significantly reduced time and computational resources, thus underscoring its utility across various domains of machine learning [24].

1) Choose the family of Transfer Learning Techniques

In this research, models were selected for comparison from the Keras Applications library [11], which provides performance benchmarks for 38 models. The Top-1 Accuracy metric, which indicates a model's performance on the ImageNet validation dataset, was focused on. Four models with Top-1 Accuracy exceeding 80% and representing the best-performing model within their respective families were identified NasNetLarge, InceptionResNetV2, EfficientNetV2L, and ConvNeXtXLarge.

2) Model of Transfer Learning Techniques

The four Convolutional Neural Network (CNN) models, NasNetLarge, InceptionResNetV2, EfficientNetV2L, and ConvNeXtXLarge, were selected for comparison in this research. All models utilized transfer learning and exhibited Top-1 Accuracy exceeding 80%. Their key strengths and limitations are as follows:

NasNetLarge, developed by Google AI, employs Neural Architecture Search (NAS) to automatically discover optimal model architectures [12]. This results in high image classification accuracy but also requires significant computational resources.

InceptionResNetV2, developed by Google Research, combines Inception and Residual Network (ResNet) architectures [13]. It utilizes Inception Modules with multi-scale filters and Residual Connections to enhance learning and mitigate the vanishing gradient problem. This model is known for its accuracy and speed, but its complex architecture poses challenges for modification.

EfficientNetV2L, also developed by Google Research, is an advancement upon EfficientNet, designed for high computational efficiency and accurate image classification [14]. It employs Progressive Learning, beginning with smaller image sizes and resolutions and gradually increasing them for faster and more efficient training. Fused-MBConv modules are used to reduce the model's parameter count and complexity, while scaling techniques allow for adaptation to available resources.

ConvNeXtXLarge, developed by Meta AI Research, is inspired by the Swin Transformer but utilizes a CNN structure for simplicity and high processing efficiency [15]. It employs a Macro Design, dividing images into patches for individual processing, and Inverted Bottlenecks, similar to ResNet but with reversed Convolutional and Depthwise Convolution layers for reduced parameter count and complexity. Large Kernel Sizes are used to enhance feature extraction, and Layer Normalization replaces Batch Normalization for improved training stability.

C. ConvNeXtXLarge Architecture Definition

The ConvNeXtXLarge architecture is shown in Fig. 4. The architecture begins by taking a 224x224 pixel input image with three color channels (RGB). This input image is then processed by a stem layer, which performs initial feature extraction and downsampling. In the stem layer, a convolutional layer with a stride of 4 is typically used, resulting in a significant reduction in spatial resolution while increasing the depth of the extracted features. The output of the stem layer is a feature map with a spatial dimension of 56x56 and 256 channels.

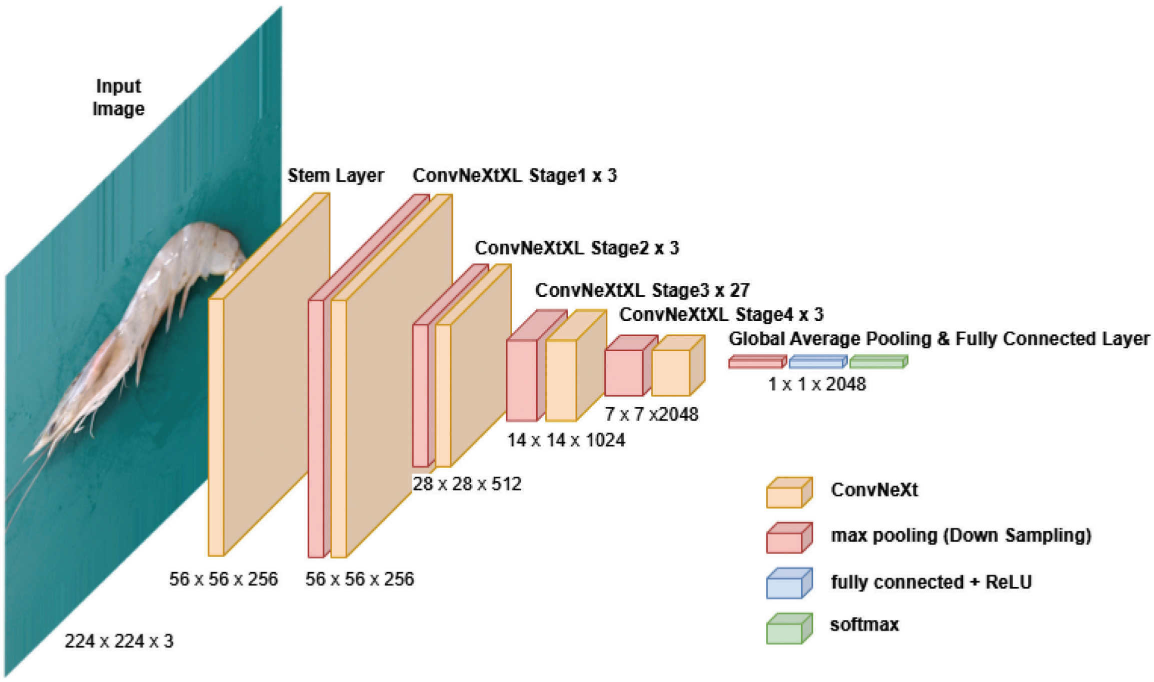

Fig. 4. ConvNeXtXLarge Architecture Diagram

Subsequent to the stem layer, the architecture features three consecutive ConvNeXtXL stages. Each stage is composed of multiple ConvNeXtXL blocks, which serve as the fundamental building blocks of the network. These blocks consist of three key components: Depthwise convolutions, layer normalizations, and inverted bottleneck layers. The Depthwise convolutions are applied to each input channel individually, offering computational efficiency. Following the Depthwise convolutions, layer normalizations are used to normalize the activations, improving the stability and convergence of the model during training. The inverted bottleneck layers then expand the number of channels before the Depthwise convolutions, allowing the model to learn more complex and expressive features.

As the feature map progresses through the ConvNeXtXL stages, its spatial dimensions are progressively downsampled while the number of channels increases. Specifically, the output of Stage 1 has a spatial dimension of 56x56 and 256 channels, Stage 2 produces a 28x28 feature map with 512 channels, and Stage 3 yields a 14x14 feature map with 1024 channels. Stage 4 further downsamples the spatial dimension to 7x7, while maintaining the 2048 channels.

The final component of the architecture is the classification head, which processes the 7x7x2048 output from Stage 4 to generate class probabilities. Global average pooling is applied to reduce each channel to a single value, resulting in a 1x1x2048 vector. This vector is then fed into fully connected layers, which learn to map the extracted features to the desired output classes. A Rectified Linear Unit (ReLU) activation function is applied to introduce non-linearity and improve the model's ability to learn complex relationships. Finally, a softmax layer is used to convert the raw scores into probabilities for each class, indicating the model's confidence in its predictions for the input image.

D. Experiments

This study investigated the application of five image classification models for shrimp image classification with two different scenarios to investigate the impact of data augmentation on a dataset. The study uses an original dataset of 405 images and compares it to an augmented dataset of 6480 images. Four pre-trained models were employed for transfer learning: InceptionResNetV2, NASNetLarge, EfficientNetV2L, and ConvNeXtXLarge were chosen from the Keras Applications library based on their high Top-1 accuracy, exceeding 80%. These models were pre-trained on the ImageNet dataset [25]. To preserve the learned features, all layers within these models were frozen. New trainable layers were added atop the pre-trained models (details in Table II) to adapt them to the shrimp classification task. The final layer was a fully connected dense layer with seven neurons and a softmax activation function, which generated a seven-class probability distribution for the shrimp images.

TABLE II
ConvNeXtXLarge LAYER DETAILS

Layer no.	Layer Type	Output Shape	Param #
0-296	ConVNeXtXLarge Layers		
297	Dense_5 (Dense)	(None,512)	1049088
298	Dense_6 (Dense)	(None,256)	131328
299	Dense_7 (Dense)	(None,128)	32896
300	Dense_8 (Dense)	(None,64)	8256
301	Dense_9 (Dense)	(None,7)	455

Total Params: 349,384,327
Trainable Params: 1,226,119
Non-trainable Params: 348,158,208

Each pre-trained model required specific input preprocessing. InceptionResNetV2, NASNetLarge, and EfficientNetV2L necessitated scaling input pixels to a range between -1 and 1. ConvNeXtXLarge, however, accepted the original input pixel range of 0 to 255. The classical CNN model, on the other hand, required scaling between 0 and 1. A batch size of 32 was used to train all models. The default training epoch was set to 30, with early stopping implemented using patience of 3. The learning rate was set to 0.001 and the optimizer was set to Adam. This signifies that training would halt and restore the best parameter weights as the final model if the validation loss failed to improve for three consecutive epochs. Each model was trained ten times, with the results reported as average precision, recall, F1-score, and accuracy. Moreover, average training time, average inference time, and memory usage were reported to compare the computational complexity of each model.

The experiments were conducted on a Windows 11 Home system equipped with an AMD Ryzen 7 5800 8-Core Processor at 3.40 GHz, 32.0 GB of installed RAM, and an Nvidia GeForce RTX 3070 graphics card. Python was the programming language utilized, and TensorFlow version 2.10.1 facilitated the training on the Nvidia GPU.

III. Result and Discussion

The table presents a comparison of the transfer learning models and the classical CNN model on the shrimp image classification task with and without augmentation. The results reported are the average training time, inference time, memory usage, precision, recall, F1-score, and accuracy across ten training repetitions for the seven classes. The overall results show that the accuracy of the augmented dataset is higher than the non-augmented dataset. The augmentation technique plays a crucial role in increasing model performance.

For the performance comparison, the transfer learning model, ConvNeXtXLarge, achieved superior performance compared to the other transfer learning models and the classical CNN model, particularly in F1-score and accuracy, reaching a value of 95%. This observation could be attributed to the size and complexity of ConvNeXtXLarge. As the largest model among those available in Keras applications, it has 350.1 million trainable parameters and a size of 1310 MB. This increased capacity might have contributed to its superior performance not only on the ImageNet validation dataset during pre-training but also on our specific shrimp abnormality classification validation dataset.

ConvNeXtXLarge demonstrated the highest accuracy among the evaluated models and exhibited a relatively efficient training time, ranking second only to NASNetLarge. Notably, ConvNeXtXLarge achieved a 15% accuracy improvement over NASNetLarge. However, it also presented the highest inference time and memory consumption among all the transfer learning models considered. This is attributed to its complex architecture, comprising 297 layers.

The training curves for loss and accuracy, as illustrated in Fig. 5 and Fig. 6 respectively, provide insights into the training dynamics of the ConvNeXtXLarge model. Early stopping was triggered at epoch 7 due to the inability of the model to improve validation loss for three consecutive epochs. Consequently, the training process reverted to the weights associated with epoch 4, which achieved a validation accuracy of approximately 0.96.

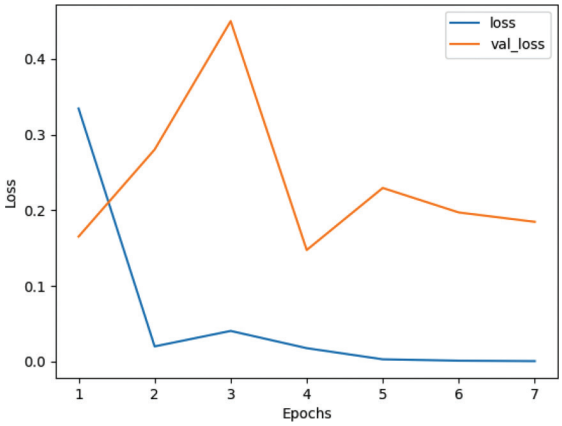

Fig. 5. Loss and validation loss when training ConvNeXtXLarge Model

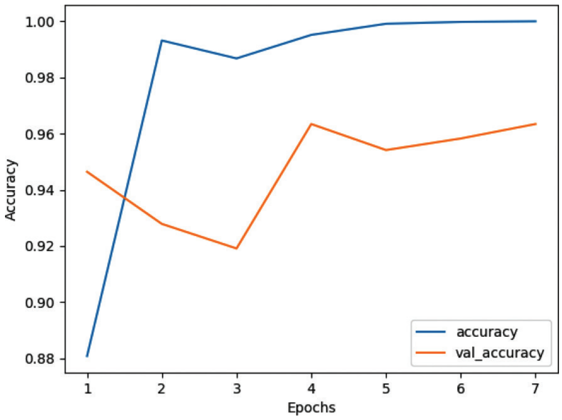

Fig. 6. Accuracy and validation accuracy when training ConvNeXtXLarge

Fig. 7 depicts the confusion matrix as a heatmap, visualizing the relationship between true labels and predicted labels. The dominant presence of values along the diagonal signifies a high degree of accurate predictions. However, a minor number of misclassifications are observed, with the BustedHead class exhibiting the lowest F1-score (0.93) as detailed in Table IV. This class appears to be visually similar to the Chin and Complete classes, potentially leading to confusion during the model's learning process. Similar misclassifications are observed for the TailOff and Half classes, both achieving an F1-score of 0.94 (second lowest).

The superior performance observed in certain models, notably ConvNeXtXLarge, can be attributed to their advanced architectures. ConvNeXtXLarge benefits from the ConvNeXt architecture, incorporating Depthwise convolution, inverted bottlenecks, and attention mechanisms, which facilitate efficient feature extraction. As depicted in Table III, ConvNeXtXLarge yields the highest accuracy, but also exhibits the highest inference time and memory consumption.

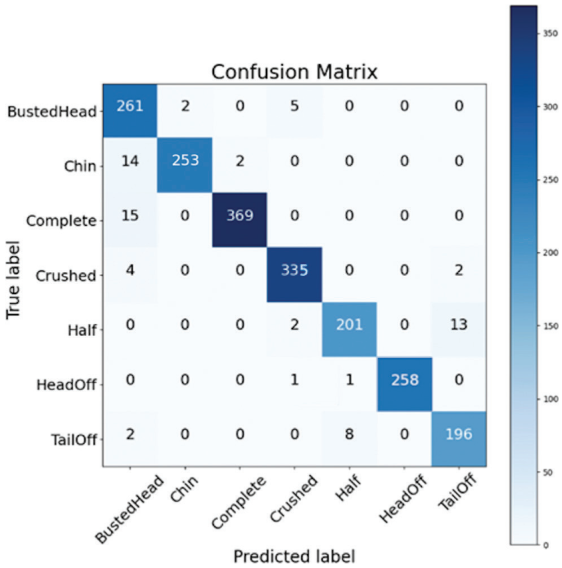

Fig. 7. Confusion matrix of the ConvNeXtXLarge Model

EfficientNetV2L leverages the EfficientNet architecture, prioritizing a balance of model size, processing speed, and accuracy through techniques such as compound scaling and progressive learning. Table III reveals that this model achieves the second-highest accuracy while maintaining lower inference time and memory usage compared to the top-performing model.

In contrast, InceptionResNetV2 and NASNetLarge, while employing complex architectures with inception modules and residual connections, exhibit comparatively lower performance. In Table III shows these two models demonstrate lower inference time and memory usage; however, exhibit comparatively lower performance.

TABLE III
THE COMPARISON AVERAGE OF TRAINING TIME, INFERENCE TIME, MEMORY USAGE, PRECISION, RECALL, F1- SCORE AND ACCURACY OF AUGMENTATION (AUG) AND NO AUGMENTATION (No AUG)

Model	Avg. Training Time (mins)	Avg. Inference Time (ms per Sample)	Memory Usage (GiB)	Avg. Precision	Avg. Recall	Avg. F1-score	Avg. Accuracy
Classical CNN (AUG)	21.42	26.79	3.44	0.79	0.77	0.77	0.77
Classical CNN (No AUG)	1.27	28.73		0.75	0.72	0.72	0.72
InceptionResNetV2 (AUG)	14.6	22.15	3.75	0.84	0.84	0.84	0.84
InceptionResNetV2 (No AUG)	2.09	31.58		0.818	0.81	0.807	0.81
NASNetLarge (AUG)	11.47	22.48	3.99	0.81	0.80	0.80	0.80
NASNetLarge (No AUG)	1.83	32.77		0.79	0.78	0.77	0.78
EfficientNetV2L (AUG)	19.10	23.49	4.28	0.88	0.88	0.88	0.88
EfficientNetV2L (No AUG)	2.53	35.48		0.83	0.82	0.82	0.82
ConvNeXtXLarge (AUG)	12.83	24.16	4.91	0.96	0.95	0.95	0.95
ConvNeXtXLarge (NoAUG)	2.35	37.29		0.94	0.93	0.93	0.93

TABLE IV
PRECISION, RECALL, F1-SCORE OF SEVEN CLASSES

Class Label	Precision	Recall	F1-score	Supports
BustedHead	0.88	0.97	0.93	268
Chin	0.99	0.94	0.97	269
Complete	0.99	0.96	0.98	384
Crushed	0.98	0.98	0.98	341
Half	0.96	0.93	0.94	216
HeadOff	1.00	0.99	1.00	260
TailOff	0.93	0.95	0.94	206
Accuracy	-	-	0.96	1944
Macro avg	0.96	0.96	0.96	1944
Weighted avg	0.97	0.96	0.96	1944

The goal of this model selection process is to identify the optimal model for deployment on an embedded system connected to a camera for shrimp sorting on a conveyor belt. Resource constraints of the system play a crucial role in this selection.

For systems with ample resources, ConvNeXtXLarge is recommended due to its superior performance. However, when resources such as processing power and memory are limited, a smaller model with acceptable accuracy and inference time, like EfficientNetV2L, becomes a more suitable choice.

In extremely resource-constrained scenarios, InceptionResNetV2 or NASNetLarge may be considered. While these models offer reduced inference time and memory usage, this efficiency comes at the cost of lower accuracy. Therefore, the final model selection requires a careful trade-off between performance and resource utilization based on the specific limitations of the embedded system.

Future improvements could be achieved by fine-tuning the transfer learning models. This involves freezing certain layers of the pre-trained models and retraining them with an adjusted learning rate. Additionally, the shrimp dataset could be further refined to ensure class balance, mitigating potential bias and enhancing model performance.

IV. CONCLUSION

Shrimp transportation often results in damage to the product. Consequently, it is crucial to implement a shrimp sorting system prior to processing to remove damaged shrimp from the production line. Damaged shrimp are categorized into six grades and sold at varying prices, thereby enhancing the value of these compromised products.

This research employed two scenarios: a shrimp dataset without augmentation and an augmented shrimp dataset. Geometric transformation augmentation was used to increase the dataset size and serve as input to four top-1 accuracy models with over 80% accuracy from Keras Applications: NasNetLarge, InceptionResNetV2, EfficientNetV2L, and ConvNeXtXLarge. These models were evaluated against a baseline CNN model on a dataset with seven shrimp classes: complete body shrimp, crushed head shrimp, shrimp with head loss, head loss shrimp with remaining chin, cut tail shrimp, torn-in-half shrimp (upper half and lower half), and totally crushed shrimp.

Experimental results revealed two primary findings. Firstly, the augmented shrimp dataset achieved superior performance compared to the non-augmented dataset. Secondly, in the performance comparison, the transfer learning model ConvNeXtXLarge emerged as the superior model achieving the highest accuracy (95%), precision (0.96%), recall (0.95%), and F1-score (0.95%). Analysis of misclassifications indicated that certain classes share similar characteristics, potentially causing model confusion. However, further accuracy improvement is possible through fine-tuning. Future research directions include developing an automated shrimp sorting machine on a conveyor belt utilizing this model.

Acknowledgment

During the preparation of this work, the authors used ChatGPT, Google Gemini, and Microsoft Copilot to check and correct grammatical errors and paraphrase passages during the manuscript writing process. After using these tools/services, the authors reviewed and edited the content as needed and took full responsibility for the content of the publication.

To thank Mr. Patcharapol Boonsin, owner of a shrimp pond in Chachoengsao Province, who provided information about shrimp farming and about shipping shrimp to the processing factory.

References

[1] Centre for Agricultural Information, Office of Agricultural Economics. (2024, Aug. 22). *Thailand Foreign Agricultural Trade Statistics 2023*. [Online]. Available: https://www.oae.go.th/assets/portals/1/files/jounal/2567/tradestat2566.pdf

[2] N. Masunee, S. Chaiprapat, and K. Waiyagan, "Development of an Image Processing System in Splendid Squid Quality Classification," in *Proc. ICDIP*, 2013, pp. 252-259.

[3] N. Thanasarn, S. Chaiprapat, K. Waiyakan et al., "Automated Discrimination of Deveined Shrimps Based on Grayscale Image Parameters," *J. Food Process Eng.*, vol. 42, no. 4, p. e13041, Mar. 2019.

[4] P. H. Andersen, *Sustainable Operations Management (SOM) Strategy and Management: An Introduction to Part I.* Melbourne, AU: Palgrave Macmillan, 2019, pp. 15-25.

[5] R. Kler, G. Elkady, K. Rane et al., "Machine Learning and Artificial Intelligence in the Food Industry: A Sustainable Approach," *Journal of Food Quality*, vol. 2022, no. 1, pp. 1-9, May 2022.

[6] T. Valeeprakhon, K. Orkphol, and P. Chaihuadjaroen, "Deep Convolutional Neural Networks Based on VGG-16 Transfer Learning for Abnormalities Peeled Shrimp Classification," *International Scientific Journal of Engineering and Technology (ISJET)*, vol. 6, no. 2, pp. 13-23, Dec. 2022.

[7] H. Yu, X. Liu, H. Qin et al., "Automatic Detection of Peeled Shrimp Based on Image Enhancement and Convolutional Neural Networks," in *Proc. The 8th International Conference on Computing and Artificial Intelligence*, 2022, pp. 439-450.

[8] Y. Zhang, C. Wei, Y. Zhong et al., "Deep Learning Detection of Shrimp Freshness Via Smartphone Pictures," *Journal of Food Measurement and Characterization* , vol. 16, no. 5, pp. 3868-3876, Jun. 2022.

[9] K. Wang, C. Zhang, R. Wang et al., "Quality Non-Destructive Diagnosis of Red Shrimp Based on Image Processing," *Journal of Food Engineering*, vol. 357, p. 111648, Jul. 2023.

[10] K. Prema and J. Visumathi, "An Improved Non-Destructive Shrimp Freshness Detection Method Based on Hybrid CNN and SVM with GAN Augmentation," in *Proc. 2022 International Conference on Advances in Computing, Communication and Applied Informatics (ACCAI)*, 2022, pp. 1-7.

[11] Keras Team. (2024, August 22). *Keras Applications*. [Online]. Available: https://keras.io/api/applications/

[12] B. Zoph, V. Vasudevan, J. Shlens et al., "Learning Transferable Architectures for Scalable Image Recognition," in *Proc. The IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8697-8710.

[13] C. Szegedy, S. Ioffe, V. Vanhoucke et al., "Inception-v4, Inception-Resnet and the Impact of Residual Connections on Learning," in *Proc. The AAAI Conference on Artificial Intelligence*, 2017. pp. 4278-4284.

[14] M. Tan and Q. Le, "Efficientnetv2: Smaller Models and Faster Training," in *Proc. International Conference on Machine Learning, PMLR, 2021*, pp. 10096-10106.

[15] Z. Liu, H. Mao, C. Y. Wu et al., "A Convnet for the 2020s," in *Proc. The IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 11976-11986.

[16] N. E. Khalifa, M. Loey, and S. Mirjalili, "A Comprehensive Survey of Recent Trends in Deep Learning for Digital Images Augmentation," *Artif. Intell. Rev.*, vol. 55, no. 3, pp. 2351-2377, Sep. 2022.

[17] Y. LeCun, Y. Bengio, and G. Hinton, "Deep Learning," *Nature*, vol. 521, no. 7553, pp. 436-444, May. 2015.

[18] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet Classification with Deep Convolutional Neural Networks," *Adv. Neural Inf. Process. Syst.*, vol. 25, 2012, pp. 84-90.

[19] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, Massachusetts: MIT Press, 2016, pp. 1-800.

[20] W. Rawat and Z. Wang, "Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review," *Neural Comput.*, vol. 29, no. 9, pp. 2352-2449, Jun. 2017.

[21] J. Gu, Z Wang, J. Kuen et al., "Recent Advances in Convolutional Neural Networks," *Pattern Recognition*, vol. 77, pp. 354-377, May. 2018, doi: https://doi.org/10.1016/j.patcog.2017.10.013

[22] S. J. Pan and Q. Yang, "A Survey on Transfer Learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345-1359, Oct. 2009.

[23] K. Weiss, T. M. Khoshgoftaar, and D. Wang, "A Survey of Transfer Learning," *Journal of Big Data*, vol. 3, no. 1, p. 9, Dec. 2016, doi: 10.1186/s40537-016-0043-6

[24] C. Tan, F. Sun, T. Kong et al., "A Survey on Deep Transfer Learning," in *Proc. 27th International Conference on Artificial Neural Networks, Rhodes, Greece*, 2018, pp. 270-279.

[25] J. Deng, W. Dong, R. Socher et al., "Imagenet: A Large-Scale Hierarchical Image Database," in *Proc. 2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248-255.

Korawit Orkphol is a lecturer at the Department of Computer Engineering, Faculty of Engineering at Sriracha, Kasetsart University, Sriracha campus, Thailand. He graduated Ph.D. in Computer Science and Technology from Harbin Engineering University, P.R. Harbin, China. His research field including of Artificial Intelligence, Machine Learning, and Natural Language Processing.

Kathawach Satianpakiranakorn is a lecturer at the Department of Computer Engineering, Faculty of Engineering at Sriracha, Kasetsart University, Sriracha campus, Thailand. He graduated M.Eng. in Computer Applied Technology from Harbin Engineering University, P.R. Harbin, China. His research field including of Artificial Intelligence, Computer Visions, and Digital Image Processing.

Jidapa Chaihuadjaroen is an Assistant Professor at the Department of Computer Engineering, Faculty of Engineering at Sriracha, Kasetsart University, Sriracha campus, Thailand. She graduated M.S in Computer Science from King Mongkut's Institute of Technology Ladkrabang, Thailand. Her research field including of Computer Vision and Image Processing.

Tamnuwat Valeeprakhon is an Assistant Professor at the Department of Computer Engineering, Faculty of Engineering at Sriracha, Kasetsart University, Sriracha campus, Thailand. He graduated M.Eng in Computer Engineering from Khon Kaen University, Thailand. He was granted a scholarship from the National Science and Technology Development Agency (NSTDA) for studying at the Kochi University of Technology, Japan. His research interests including of Computer Vision, Image Processing, and Artificial Intelligence.