

Article

Inspection of Underwater Hull Surface Condition Using the Soft Voting Ensemble of the Transfer-Learned Models

Byung Chul Kim ¹, Hoe Chang Kim ¹, Sungho Han ² and Dong Kyou Park ^{3,*}

- ¹ School of Mechanical Engineering, Korea University of Technology and Education, Cheonan 31253, Korea; mir7942@koreatech.ac.kr (B.C.K.); gh1ckd12@koreatech.ac.kr (H.C.K.)
- ² SLM Global Co., Ltd., Daejeon 34037, Korea; sh.han@slm-global.co.kr
- ³ Department of Electromechanical Convergence Engineering, Korea University of Technology and Education, Cheonan 31253, Korea
- * Correspondence: pdongkyou@koreatech.ac.kr; Tel.: +82-41-640-8614

Abstract: In this study, we propose a method for inspecting the condition of hull surfaces using underwater images acquired from the camera of a remotely controlled underwater vehicle (ROUV). To this end, a soft voting ensemble classifier comprising six well-known convolutional neural network models was used. Using the transfer learning technique, the images of the hull surfaces were used to retrain the six models. The proposed method exhibited an accuracy of 98.13%, a precision of 98.73%, a recall of 97.50%, and an F₁-score of 98.11% for the classification of the test set. Furthermore, the time taken for the classification of one image was verified to be approximately 56.25 ms, which is applicable to ROUVs that require real-time inspection.

Keywords: hull cleaning condition; underwater inspection image; soft voting ensemble classification; transfer learning



Citation: Kim, B.C.; Kim, H.C.; Han, S.; Park, D.K. Inspection of Underwater Hull Surface Condition Using the Soft Voting Ensemble of the Transfer-Learned Models. *Sensors* **2022**, *22*, 4392. <https://doi.org/10.3390/s22124392>

Academic Editors: KWONG Tak Wu Sam, Yun Zhang, Xu Long and Tiesong Zhao

Received: 6 May 2022
Accepted: 7 June 2022
Published: 10 June 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The submerged part of a ship's hull is susceptible to biofouling in the form of pollutants or organisms such as water mosses and seagrass that attach themselves to the bottom and sides of the submerged surfaces. This phenomenon not only damages the surface of ships but also provides unwanted resistance during normal operation, resulting in inferior performance [1–3]. In addition, when a ship enters a port, the various pollutants attached to the hull surface can contaminate the seawater in the port. In the case of ships traveling abroad, aquatic alien creatures that are transported to a different region can disrupt local marine ecosystems [4]. Thus, hull surfaces must be cleaned periodically while ships are anchored in a port.

Conventional hull cleaning is performed by divers. Recently, however, studies have been conducted on cleaning hull surfaces using a remotely operated underwater vehicle (ROUV) [5–9]. A human operator observes the submerged hull surface through a camera mounted on the ROUV and checks its condition. The ROUV is subsequently remotely controlled to clean the affected parts of the hull. However, autonomous hull cleaning without human intervention requires ROUVs capable of recognizing the hull condition. In addition, since the hull condition should be immediately fed back to the ROUV, the process of recognition must occur in real-time.

However, owing to underwater conditions, the images observed by an ROUV through its camera are not clear. In addition, the images may differ depending on the depth of operation, underwater conditions, and lighting; consequently, existing image-processing methods are insufficient for the accurate recognition of hull conditions. Therefore, this study proposes a classification method to recognize the hull condition using convolutional neural networks (CNNs) [10] with images of the hull surface acquired through the ROUV camera. Based on the image, the hull condition is categorized into two classes: (1) positive

class: the hull surface is contaminated enough to require cleaning; and (2) negative class: the hull surface is clean enough to not require cleaning.

When using CNN models, both models and training datasets are important. The models used in this study were trained using a publicly available dataset called ImageNet [11]. Therefore, they cannot be used to classify hull surface conditions. Instead, we collected hull surface images under various underwater conditions using our ROUV. However, collecting underwater hull images is difficult because it requires permission from ship owners. In addition, less-than-optimal underwater conditions complicate the process of collecting the images of submerged hull surfaces. Therefore, in this study, we used a transfer learning [12,13] technique to retrain pretrained models for image classification to enable high accuracy with a small number of images. Furthermore, the required training time is short. However, several well-known pretrained models exist for image classification. These can present different classifications for the same image. To overcome this problem and increase accuracy, we used a soft voting ensemble [14] technique comprising transfer-learned models. Finally, to the best of our knowledge, this is the first study on hull surface inspection using machine learning techniques.

The major contributions of this study are as follows:

1. The condition of the hull surface was classified with high accuracy by using CNN models with hull surface images.
2. Our own training dataset was obtained under various underwater conditions using the developed ROUV.
3. Transfer learning of the pretrained models was used to adapt the pretrained models to classification of the hull surface.
4. A higher accuracy was obtained using a soft voting ensemble technique comprising several transfer-learned models.

The remainder of this paper is organized as follows. Section 2 describes previous related studies. Section 3 describes the soft voting ensemble classifier used in this study, and Section 4 describes the generation of the dataset used for training the CNNs. Section 5 discusses the proposed method and experimental results. Finally, Section 6 concludes the paper.

2. Related Works

2.1. Inspection of Products and Underwater Objects

Several studies have used images from cameras to inspect product defects in various fields. Chang et al. [15] proposed a method for the defect inspection of the color filter, which is a component of the TFT-LCD module, using fuzzy inference from the inspection images. Jiang et al. [16] proposed a method for inspecting printed circuit boards (PCBs) for defects using logistic regression from inspection images. Zhang et al. [17] used a genetic algorithm, artificial neural network, and expert system to inspect copper strip images for defects. Zhao et al. [18] studied the image-based defect inspection of concrete surfaces. Siegel et al. [19] and Mumtaz et al. [20] studied aircraft defect inspection. Amosov et al. [21] studied the defect inspection of rivet joints in aircrafts. Raouf et al. [22] proposed a machine-learning-based fault classification system for the fault detection of rotating vector reducers.

For hull surface inspection, methods using ultrasound [23,24] or sonar [25] have previously been used to inspect coating breakdown, corrosion, and cracks. However, with the improvement in underwater camera performance and image-processing technology, studies on the automatic inspection of hull surfaces have received significant attention. Neghdaripour and Firoozfam [26] proposed a stereo vision system for underwater hull inspections. Navarro et al. [27] proposed a sensor system and a method for detecting defects on a hull surface using thresholds from the images obtained. Fernández-Isla et al. [28] proposed a method for detecting defects from images of a hull surface using wavelet transform. Masi et al. [29] and Ortiz et al. [30] used artificial neural networks to detect corrosion in seabed pipelines and hulls.

Chin et al. [31] classified biofouling images using transfer learning of the Inception V3 model. Gormley et al. [32] classified images of aquatic creatures attached to marine structures using CoralNet [33]. Bloomfield et al. [34] classified images of aquatic creatures using a CNN. Liniger et al. [35] reviewed classification methods using deep learning to categorize marine growth on offshore structures.

In this study, similar to the studies by Chin et al. [31], Gormley et al. [32], and Bloomfield et al. [34], CNNs were used to classify underwater images. However, the classification target in this study was the condition of hull surfaces and not aquatic creatures. This study utilized transfer learning and a soft voting ensemble.

2.2. Datasets for Training

Many publicly available datasets exist for training the CNN models. The MNIST dataset [36] contains a set of handwritten digits from zero to nine. It contains 60,000 training images and 10,000 test images. The images are 28×28 grayscale images. They are often used to train simple models. ImageNet [11] is the largest image dataset used in computer vision. It contains more than 14 million images and more than 20,000 categories with a typical category, such as “balloon” or “strawberry”. Most image classification studies have used it as a benchmark dataset. The COCO dataset [37] is a large-scale object detection, segmentation, and captioning dataset published by Microsoft. It contains image annotations across 80 categories with over 1.5 million object instances. It is often used as a benchmark algorithm to compare object detection performance. In addition, image datasets for indoor scenes [38], celebrities [39], dog breeds [40], and flowers [41] exist.

A small number of publicly available datasets exist for underwater images. Moreover, these are mainly datasets for aquatic creatures living underwater. Chin et al. [42] shared a dataset with 1326 labeled images divided into 10 classes, such as algae and balanus. Shihavuddin [43] published a dataset for the identification of coral reef species. CoralNet [33] is a dataset used for benthic image analysis. It also functions as a data repository and collaboration platform. This platform for sharing training data can help overcome the lack of available data. O’Byrne et al. [44] presented a method for overcoming the lack of underwater images. They generated a photorealistic synthetic scene of underwater inspection sites using an encoder–decoder model trained with 2500 images.

In this study, images of underwater hull surfaces were required, but there is no publicly available dataset for them. The existing datasets do not consider underwater objects or focus only on aquatic creatures, such as coral reefs. In this study, we collected images using the ROUV by SLM Global [45].

3. Architecture for the Classification of Underwater Hull Surface Condition

3.1. Problem Definition

The problem to be solved in this study is defined as follows:

Given a two-dimensional image x of hull surfaces that is input through the ROUV’s camera and labeled as clean (negative class) or unclean (positive class), define a binary classifier $h(x)$ that can classify the hull condition via image x ,

where the output of $h(x)$ indicates the probability $P(\text{unclean}|x)$ that the input image x is unclean. For a given threshold ϵ , if $P(\text{unclean}|x) > \epsilon$, x is classified as unclean, and if $P(\text{unclean}|x) \leq \epsilon$, x is classified as clean.

3.2. Soft Voting Ensemble Architecture

In this study, we defined a classifier using a soft voting ensemble of the well-known CNN models DenseNets [46], EfficientNets [47], Inceptions [48–50], MobileNets [51–53], ResNets [54,55], and VGGs [56], as shown in Figure 1. The soft voting ensemble classifier is a combination of multiple models. In these models, decisions are made by combining individual decisions based on probability values to specify that the data belong to a particular class. [14] In the soft voting ensemble, predictions are weighted based on the classifier’s importance and merged to obtain the sum of weighted probabilities.

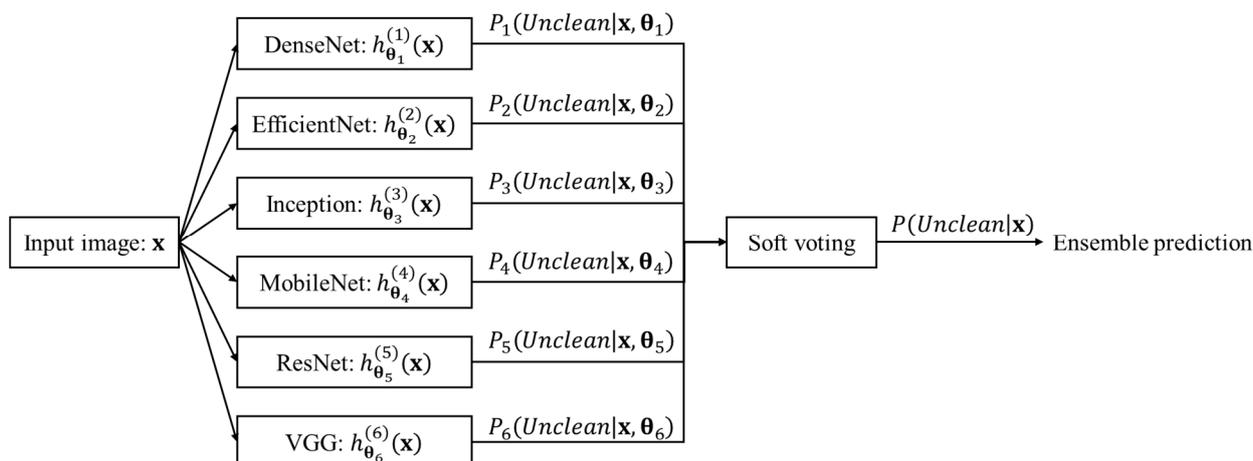


Figure 1. Soft voting ensemble classifier consisting of six models.

The classification method using the soft voting ensemble is as follows:

Step 1. Each of the six models is represented by Equation (1):

$$P_k(\text{unclean}|\mathbf{x}, \theta_k) = h_{\theta_k}^{(k)}(\mathbf{x}), \quad (1)$$

where k is the number representing each model participating in the soft voting, $h_{\theta_k}^{(k)}(\mathbf{x})$ represents the k -th model for classification, θ_k is the weights of the k -th model, \mathbf{x} is the input image, and $P_k(\text{unclean}|\mathbf{x}, \theta_k) \in [0, 1]$ —the output value of the k -th model—is the probability that the input image \mathbf{x} is clean. In this study, $k = 1 \sim 6$ represent DenseNet, EfficientNet, Inception, MobileNet, ResNet, and VGG, respectively.

Step 2. Each model is retrained with our dataset using transfer learning to determine the weights θ_k . The dataset creation and transfer learning method are described in Sections 3.3 and 4, respectively.

Step 3. $P_k(\text{unclean}|\mathbf{x}, \theta_k)$ is evaluated for each model. $P_k(\text{unclean}|\mathbf{x}, \theta_k)$ is the probability that input image \mathbf{x} is clean by the k -th classification model.

Step 4. By averaging all $P_k(\text{unclean}|\mathbf{x}, \theta_k)$ s, the final prediction value $P(\text{unclean}|\mathbf{x}) \in [0, 1]$ is evaluated using Equation (2):

$$P(\text{unclean}|\mathbf{x}) = \frac{1}{6} \sum_{k=1}^6 P_k(\text{unclean}|\mathbf{x}) = \frac{1}{6} \sum_{k=1}^6 h_{\theta_k}^{(k)}(\mathbf{x}). \quad (2)$$

Step 5. Finally, for a given threshold ε , image \mathbf{x} is classified as clean if $P(\text{unclean}|\mathbf{x}) > \varepsilon$.

Even if the number of models participating in soft voting changes, the overall process does not change. Only the number six in Equation (2) changes to the number of models.

3.3. Transfer Learning of the Pretrained Models

The optimal weights of the six models comprising the soft voting ensemble were selected using transfer learning of the pretrained models. Transfer learning [12,13] is a machine learning technique in which a model developed for a task is reused as the starting point for a model for a second task. The six models used in this study comprised 26 sub-models, as shown in Table 1, and they were pretrained for the ImageNet dataset [11]. For each model, optimal hyperparameters and weights were selected through transfer learning and hyperparameter tuning.

Table 1. Pretrained models used for the transfer learning.

Model	Sub-Models
DenseNets [46]	DenseNet121, DenseNet169, DenseNet201
EfficientNets [47]	EfficientNetB0~EfficientNetB7
Inceptions	InceptionV3 [48], InceptionResNetV2 [49], Xception [50]
MobileNets	MobileNet [51], MobileNetV2 [52], MobileNetV3-Large [53], MobileNetV3-Small [40]
ResNets	ResNet101 [54], ResNet152 [54], ResNet50 [54], ResNet101V2 [55], ResNet152V2 [55], ResNet50V2 [55]
VGGs [56]	VGG16, VGG19

Transfer learning is applied as follows. First, the input size of the pre-learned models is redefined to the size of the input image. Subsequently, the pixel values of the input image are normalized to ensure that each pixel value is between 0 and 1. Second, the layers for multiclass classification used in the pretrained models are replaced with layers for binary classification. For this purpose, a global average pooling [57] layer and dropout layer [58] are appended to the last convolution layer of the pretrained model. Finally, a fully connected layer with one node is appended using a sigmoid function as an activation function, as defined in Equation (3):

$$f(x) = \frac{1}{1 + e^{-x}} \quad (3)$$

The redefined models are trained as follows. First, only the weights of the newly appended layers among the layers of the redefined models are tuned by training. Training lasts for 20 epochs with a given learning rate α_1 for the training dataset, for which the mini-batch gradient descent method is used. The Adam optimizer [59] (with momentum parameters, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\varepsilon = 10^{-7}$) is used as an optimizer. For the loss function, the average of the binary cross-entropy values between the actual label values, $y^{(i)}$, and predicted values, $h_{\theta_k}^{(k)}(x^{(i)})$, of the images is evaluated, as defined in Equation (4):

$$J(\theta) = \frac{1}{m} \sum_{i=1}^m \left[-y^{(i)} \log(h_{\theta_k}^{(k)}(x^{(i)})) - (1 - y^{(i)}) \log(1 - h_{\theta_k}^{(k)}(x^{(i)})) \right], \quad (4)$$

where m is the number of the images used for training.

Following this, the weights of all the layers are fine-tuned for 10 epochs at a new learning rate $\alpha_2 (= \lambda \cdot \alpha_1)$ that is obtained by reducing the learning rate α_1 by a factor of $\lambda (< 1)$. Finally, the weights corresponding to the highest validation accuracy among all the epochs are selected.

Optimal hyperparameters such as dropout rate, learning rates, batch size, and sub-models are selected using hyperparameter tuning. First, the hyperparameters are tuned for each sub-model in Table 1. Subsequently, the optimal hyperparameters are selected using a random search method [60]. In this method, the value of each hyperparameter is randomly sampled from the search space comprising them, and the validation accuracy is measured. This is repeated dozens of times for each sub-model. Finally, among the sub-models of each model, the model with the highest validation accuracy was selected.

4. Collection and Creation of the Dataset

4.1. Description of the ROUV

In this study, images of the hull surfaces were collected using the ROUV developed by SLM Global [45] to clean underwater hull surfaces, which is illustrated in Figure 2. The ROUV attaches itself to the hull surface and crawls along it using electrically driven magnetic wheels. It is remotely controlled and monitored by an operator through a tether cable. While moving along the hull surface, the ROUV brushes off the pollutants on the hull

surface with two brushes installed at the bottom of the ROUV. The ROUV possesses one camera and two lights in the front and one camera at the rear. The front and rear cameras are used to check the condition of the hull surface before and after cleaning, respectively. The videos are recorded at 10 frames per second (FPS). Table 2 lists the main specifications of the ROUV.



Figure 2. ROUV for underwater hull surface cleaning.

Table 2. Main specifications of the ROUV.

Dimensions	Weight	Power Consumption	Crawling Speed	Cleaning Area Capability
1.5 m × 1.0 m × 0.6 m	270 kg	Less than 15 kW	0~60 cm/s	Maximum 1440 m ²

4.2. Dataset Creation

To retrain the pretrained models, images of the hull surfaces and their labels are required. The size and number of channels of the images were 512×512 and 3, respectively. The labels are represented as either clean or unclean. Images were extracted at intervals of 1 s from the video that was recorded by the ROUV. Subsequently, the images were manually labeled.

First, a rectangular area of 512×512 pixels containing the camera image was cut from the dashboard image of the ROUV to obtain an image of the camera region, as shown in Figure 3a. As the camera lens has a circular shape, its pure image also has a circular area. To use only the pure camera image, the circular portion is extracted using the Boolean intersection of the front camera image (Figure 3a) and mask (Figure 3b). The resulting image, shown in Figure 3c, was used for training.

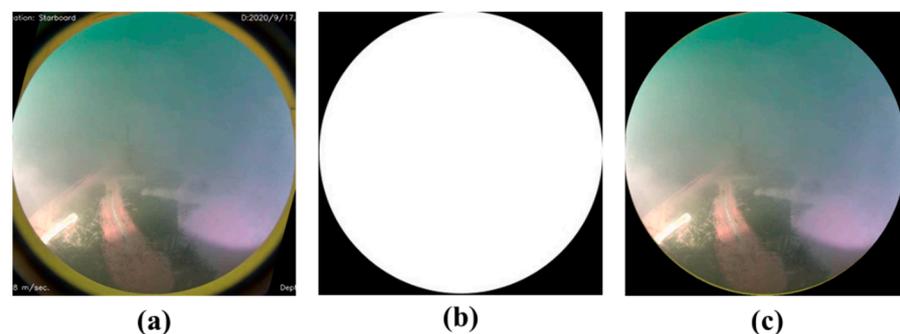


Figure 3. (a) Camera image of size 512×512 , (b) mask image, and (c) final image used for training.

Figure 4 shows several images from the dataset. Figure 4a,b are clean images, and Figure 4c–h are unclean images. White draft marks are observed in the image in Figure 4b, barnacles are observed in the images in Figure 4c,d, and green mosses are seen in the images in Figure 4e–h. In Figure 4d, the tail fin of a fish can be seen, and in Figure 4g, the coating on the hull surface is shown to have been peeled off. In Figure 4e, the image

is obscured by floating matter, and in Figure 4h, the lighting is too strong. As shown in Figure 4, underwater images include various objects, such as draft marks, fish, and floating matter. Furthermore, the underwater conditions such as lighting and the physical state of the hull surface vary. Thus, identifying the hull condition is difficult. Faulty classification due to similarity in colors of different objects is also a concern. For instance, draft marks, peeled sections of the hull surface, and barnacles are all generally white; however, only the images with barnacles should be classified as unclean.

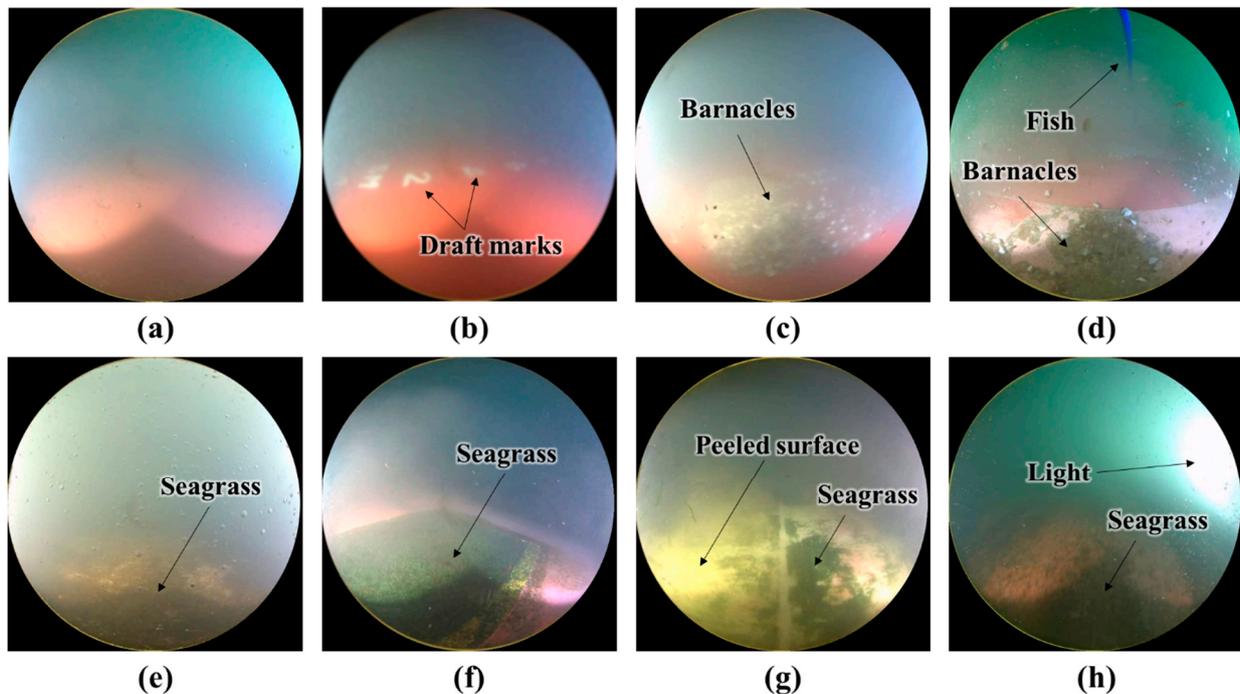


Figure 4. Example images: (a,b) clean and (c–h) unclean images.

In this study, 5683 images were extracted from videos of 20 hull surfaces at different dates and locations. These were split into two image sets: 2035 clean and 3648 unclean images. To obtain an equal number of images from the two image sets, for training, validation, and testing, 2000 images were randomly selected from each image set. Finally, each image set was split into a training, validation, and testing set in a 60:20:20 ratio. Consequently, for each class, the image set was split into 1200, 400, and 400 images, respectively.

To increase accuracy, the images of the training set were augmented by randomly applying one or more of the following four methods:

1. Brightness adjustment to randomly adjust the brightness of an image;
2. Contrast adjustment to randomly adjust the contrast of an image;
3. Saturation adjustment to randomly adjust the saturation of an image;
4. Cropping to randomly remove a particular region from an image.

Considering that the images acquired at the same position on the same hull surface may vary according to the depth and ambient brightness of the seawater, the adjustment of the brightness, contrast, and saturation can improve the accuracy. Cropping can also improve accuracy. However, the commonly used augmentation techniques of translation, rotation, flipping, and scaling were avoided because we experimentally verified that they did not improve the accuracy. We assume that such transformations do not significantly alter the images. Using the aforementioned methods, four augmented images were generated from per image. Figure 5 shows examples of augmented images.

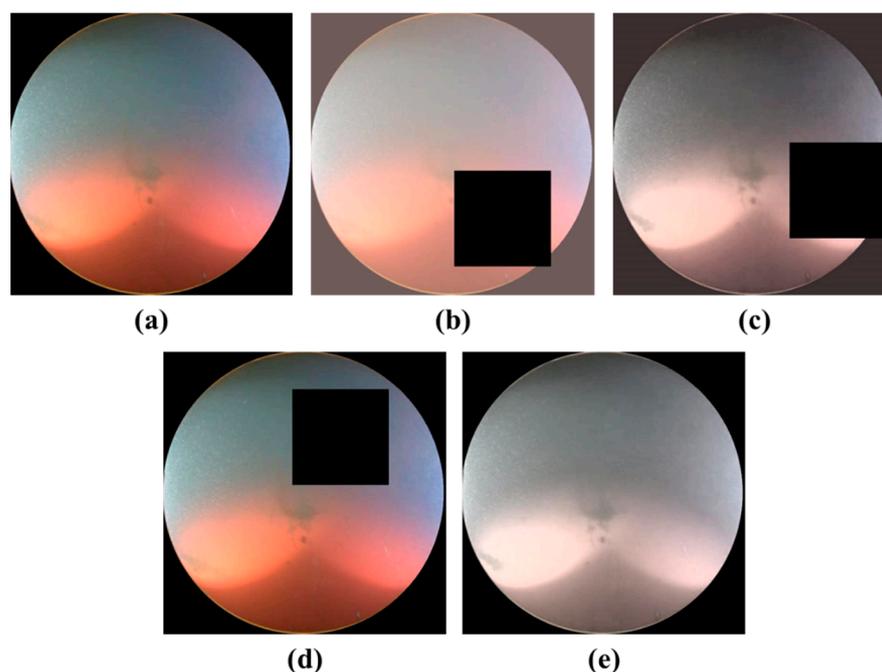


Figure 5. Examples of augmented images: (a) original image; (b) brightness, contrast, and cropping; (c) brightness, contrast, saturation, and cropping; (d) cropping; (e) saturation.

The configuration of the final dataset is listed in Table 3. The test set is used to retrain the pretrained models. The validation set is used to optimize the models via hyperparameter tuning. The test set is used for testing the models and soft voting ensemble classifier.

Table 3. Configuration of the training, validating, and test datasets.

	Training Set	Validation Set	Testing Set	Total
Clean (Negative)	6000	400	400	6800
Unclean (Positive)	6000	400	400	6800
Total	12,000	800	800	13,600

5. Implementation and Experiments

In this study, the proposed soft voting ensemble classifier was implemented using Python and Google's TensorFlow 2 and was run on computers with an Intel Xeon 3.00 GHz CPU, 128 GB RAM, and two NVIDIA TITAN RTX graphic cards. The pretrained models and weights provided by TensorFlow 2 were used. Retraining and hyperparameter tuning were performed according to the methods described in Section 4. The average time for retraining each model is shown in Table 4.

Table 4. Average elapsed time for retraining each model.

Models	Average Time for Retraining (Hour)
DenseNet	1.41
EfficientNet	2.75
Inception	0.66
MobileNet	0.58
ResNet	0.69
VGG	0.93

Table 5 lists the search space for hyperparameter tuning. Some of the values of the batch size in Table 5 may have been selected owing to the memory limitations of the graphic cards.

Table 5. Search space for the hyperparameter tuning.

Hyperparameters	Values
Sub-models	Sub-models of each model in Table 1
Initial learning rate (α_1)	0.3, 0.01, 0.03, 0.01, 0.003, 0.001, 0.0003, 0.0001
Learning rate multiplier (δ)	0.1, 0.01, 0.001, 0.0001
Batch size	4, 8, 16, 32, 64, 96, 128

To determine the optimal values of the hyperparameters for each sub-model in Table 1, 50 samples per sub-model were randomly selected from the values in Table 5. Subsequently, the sub-model was trained with the selected hyperparameter values. For the validation set, the sub-model with the highest accuracy was selected as the optimal model. The accuracy is defined as:

$$Accuracy = \frac{True\ positive + True\ negative}{Total}, \quad (5)$$

where the threshold for classification, ε , is set to 0.5. Table 6 lists the optimal hyperparameter values for each model. Table 7 shows that the training and validation accuracies are greater than 98% and 97%, respectively.

Table 6. Optimal hyperparameter values for each model.

Models	Sub-Models	Initial Learning Rate, α_1	Learning Rate Multiplier, λ	Dropout Rate	Batch Size
DenseNet	DenseNet201	0.003	0.01	0.1	32
EfficientNet	EfficientNetB4	0.01	0.01	0.2	8
Inception	InceptionV3	0.03	0.001	0.1	128
MobileNet	MobileNetV3-Large	0.01	0.01	0.2	32
ResNet	ResNet50V2	0.03	0.001	0.3	32
VGG	VGG16	0.0001	0.1	0.3	16

Table 7. Classification results for training and validation sets.

	DenseNet	EfficientNet	Inception	MobileNet	ResNet	VGG
Training accuracy (%)	99.38	98.85	99.58	98.66	99.58	99.45
Validation accuracy (%)	97.88	97.75	97.84	97.00	97.60	97.60

Table 8 presents the classification results of the test set using the soft voting ensemble classifier that comprises six optimal models. The precision, recall, and F_1 -score were calculated as:

$$Precision = \frac{True\ positive}{True\ positive + False\ positive}, \quad (6)$$

$$Recall = \frac{True\ positive}{True\ positive + False\ negative}, \quad \text{and} \quad (7)$$

$$F_1\ score = 2 \frac{Precision \times Recall}{Precision + Recall}. \quad (8)$$

Table 8 shows that both the test accuracies and F_1 -scores of the six models are higher than 96%. Therefore, even if used independently for classification, the six models can achieve an accuracy of 96% or higher. The soft voting ensemble classifier has a higher accuracy, precision, and F_1 -score than the six models. Only the recall value of the soft voting ensemble classifier comes behind that of one of the models, i.e., EfficientNet. Therefore, we verified that the images of hull surfaces can be classified with higher accuracy when using the soft voting ensemble classifier.

Figure 6 shows examples of the classification of the images of the underwater hull surfaces using the soft voting ensemble classifier. The soft voting ensemble classifier correctly classifies the images with seagrass and barnacles, shown in Figure 6a–d, as unclean. The images with draft marks, floating matter, and peeled surfaces, shown in

Figure 6e–g, were correctly classified as clean. Furthermore, in Figure 6h, the dark surface color due to lighting is not recognized as seagrass but as a clean surface.

Table 8. Classification results using the soft voting ensemble classifier for test set.

	DenseNet	EfficientNet	Inception	MobileNet	ResNet	VGG	Voting
True positive	380	392	384	386	382	387	390
False negative	20	8	16	14	18	13	10
False positive	11	9	5	9	8	9	5
True negative	389	391	395	391	392	391	395
Total	800	800	800	800	800	800	800
Accuracy (%)	96.13	97.88	97.38	97.13	96.75	97.25	98.13
Precision (%)	97.19	97.76	98.71	97.72	97.95	97.73	98.73
Recall (%)	95.00	98.00	96.00	96.50	95.50	96.75	97.50
F ₁ -score (%)	96.08	97.88	97.34	97.11	96.71	97.24	98.11

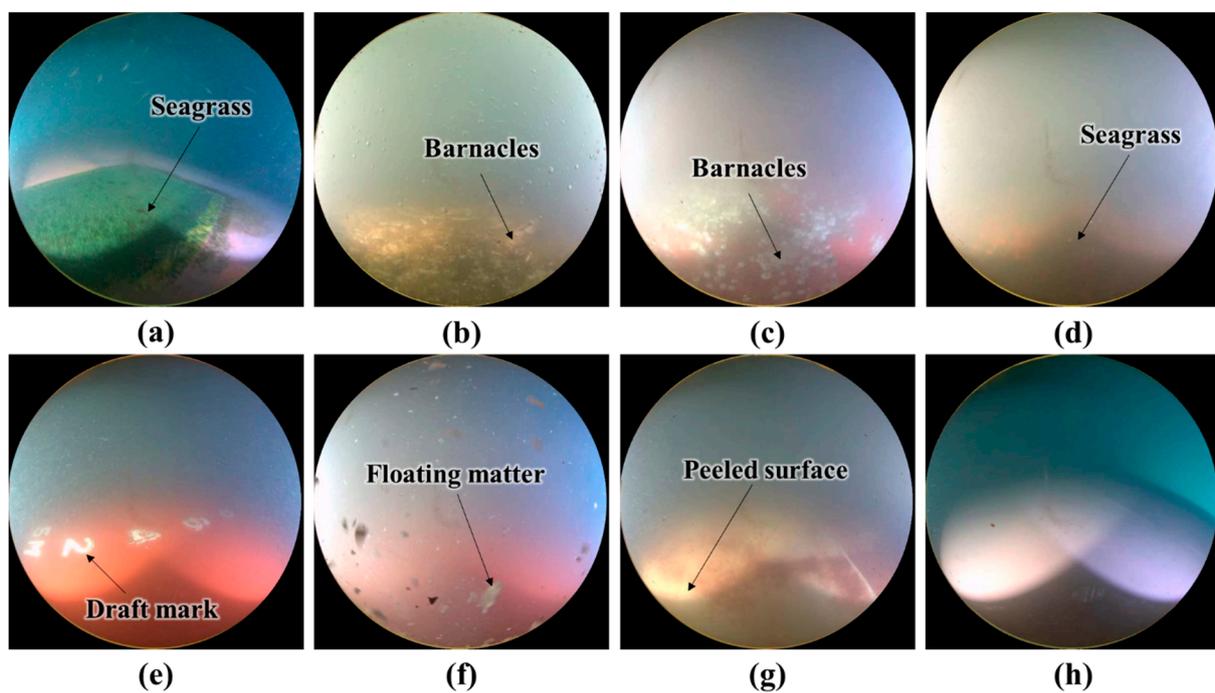


Figure 6. Examples of classified images with the soft voting ensemble: (a–d) images classified as unclean and (e–h) images classified as clean.

Figure 7 shows examples of the mis-classified images. Since Figure 7a–c only contain a small area of seagrass, the images were labeled as clean. However, the soft voting classifier seems to classify these images as unclean because of the seagrass. In Figure 7d, the dark colored seagrass and seawater overlap; consequently, the hull surface was erroneously identified as seawater. In Figure 7e, the seagrass was not correctly recognized owing to the disturbance caused by the floating matter. The image in Figure 7f has draft marks and seagrass; however, only the seagrass was recognized.

Figures 8–14 show the receiver operating characteristic (ROC) and precision call (PR) curves for varying classification threshold. For the soft voting ensemble classifier in Figure 8, the area under the curve (AUC) was almost one, and the highest over the other six models. This indicates that the soft voting ensemble classifier has the best ability to classify the conditions of the hull surface among the other six models.

Figure 15 shows the results of the sensitivity analysis, in which one of the models was eliminated. Compared with the results in Table 7, the cases using the five models among the six models were also superior to using only one model. Compared with the case using the six models, except for precision, using the six models (depicted as None in Figure 15)

is superior in terms of accuracy, recall, and F_1 -score. The case eliminating EfficientNet is superior in terms of precision but inferior in accuracy, recall, and F_1 -score. Specifically, the F_1 -score, which is the harmonic mean of the precision and recall, of the six models was higher than that of the five models. In conclusion, the case using the six models was superior to that using the five models.

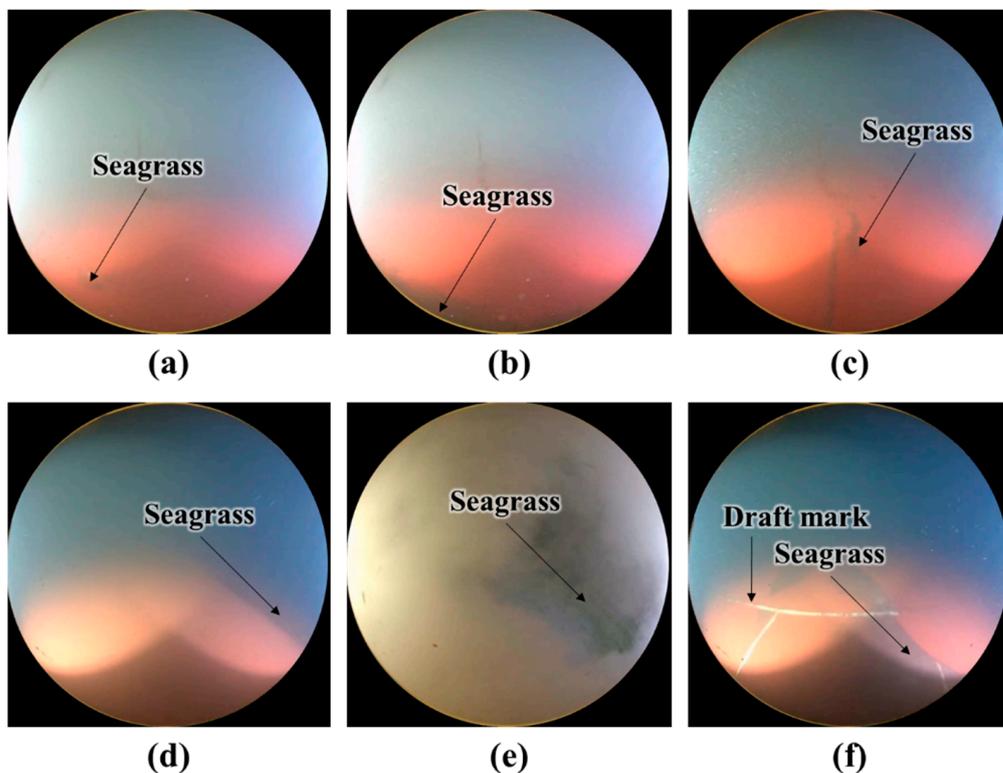


Figure 7. Examples of mis-classified images with the soft voting ensemble classifier: (a–c) False positive and (d–f) false negative.

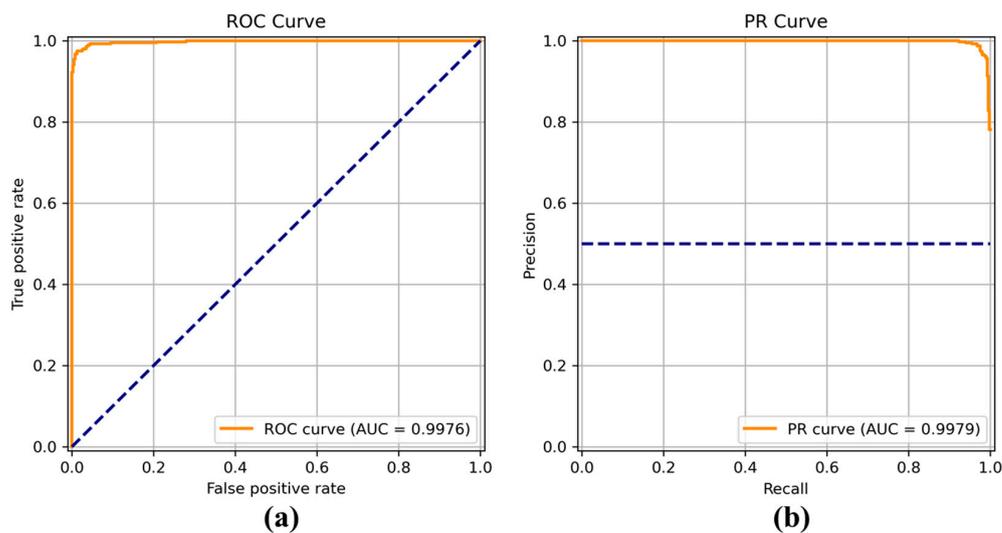


Figure 8. (a) Receiver operating characteristic (ROC) curve and (b) precision recall (PR) curve of the soft voting ensemble classifier.

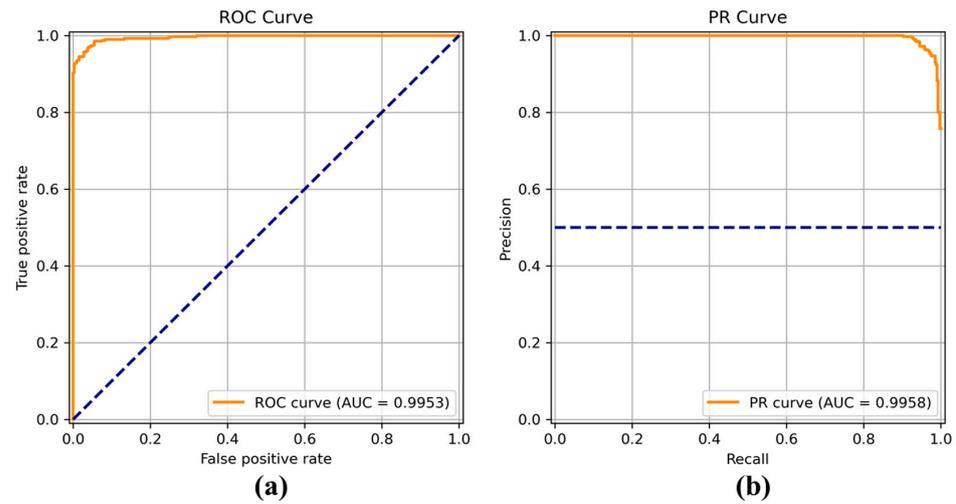


Figure 9. (a) Receiver operating characteristic (ROC) curve and (b) precision recall (PR) curve of DenseNet.

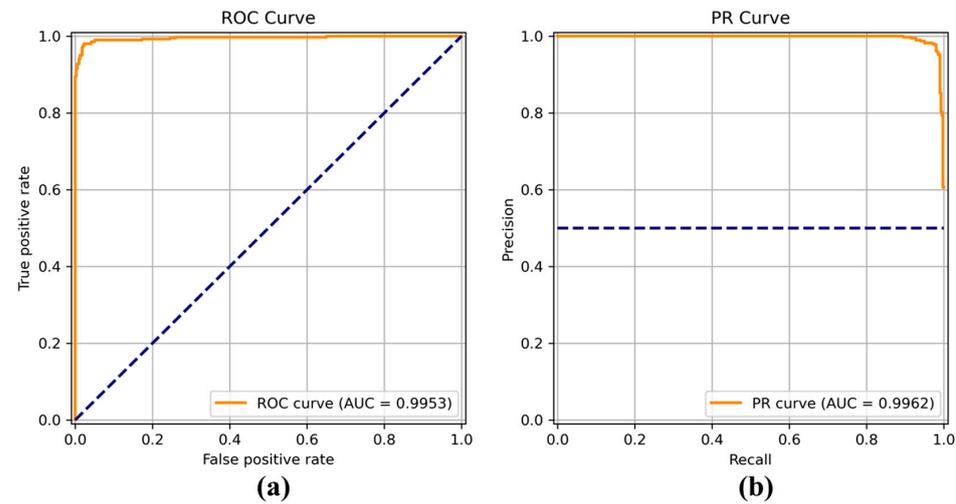


Figure 10. (a) Receiver operating characteristic (ROC) curve and (b) precision recall (PR) curve of EfficientNet.

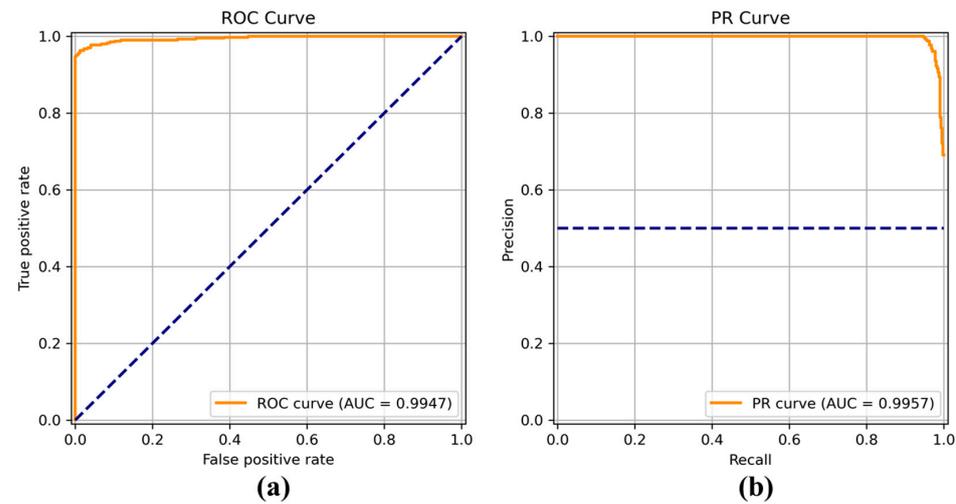


Figure 11. (a) Receiver operating characteristic (ROC) curve and (b) precision recall (PR) curve of Inception.

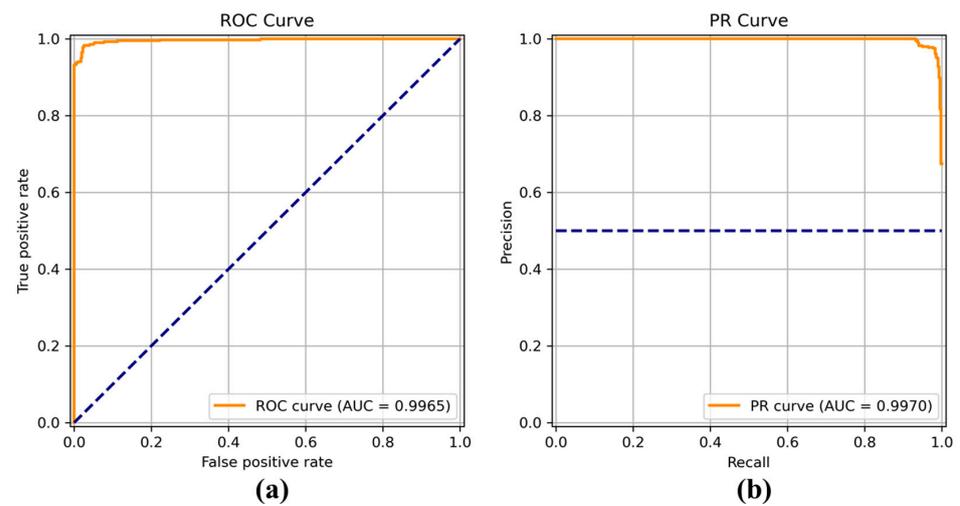


Figure 12. (a) Receiver operating characteristic (ROC) curve and (b) precision recall (PR) curve of MobileNet.

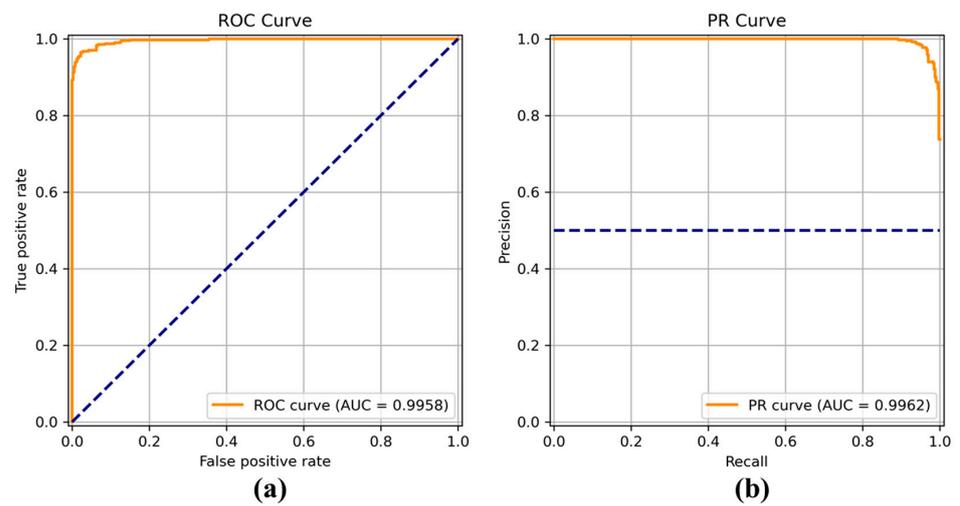


Figure 13. (a) Receiver operating characteristic (ROC) curve and (b) precision recall (PR) curve of ResNet.

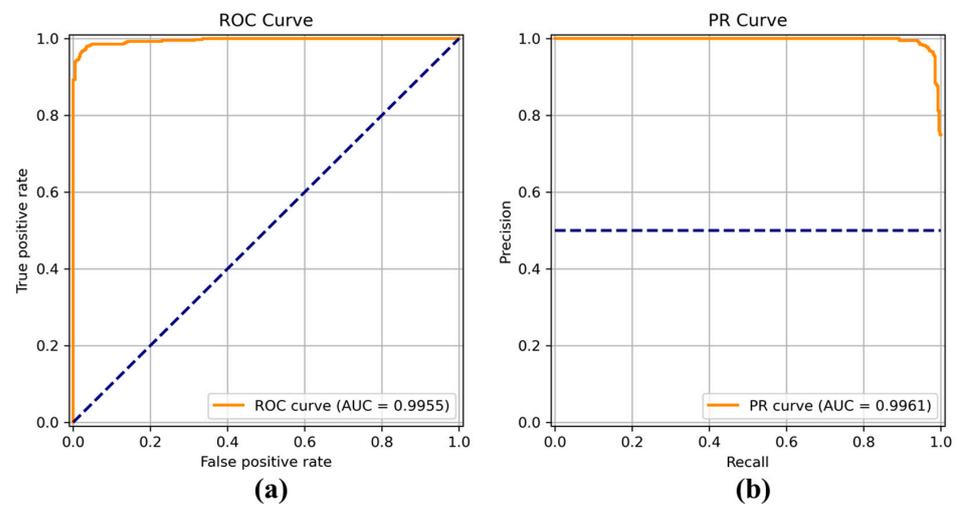


Figure 14. (a) Receiver operating characteristic (ROC) curve and (b) precision recall (PR) curve of VGG.

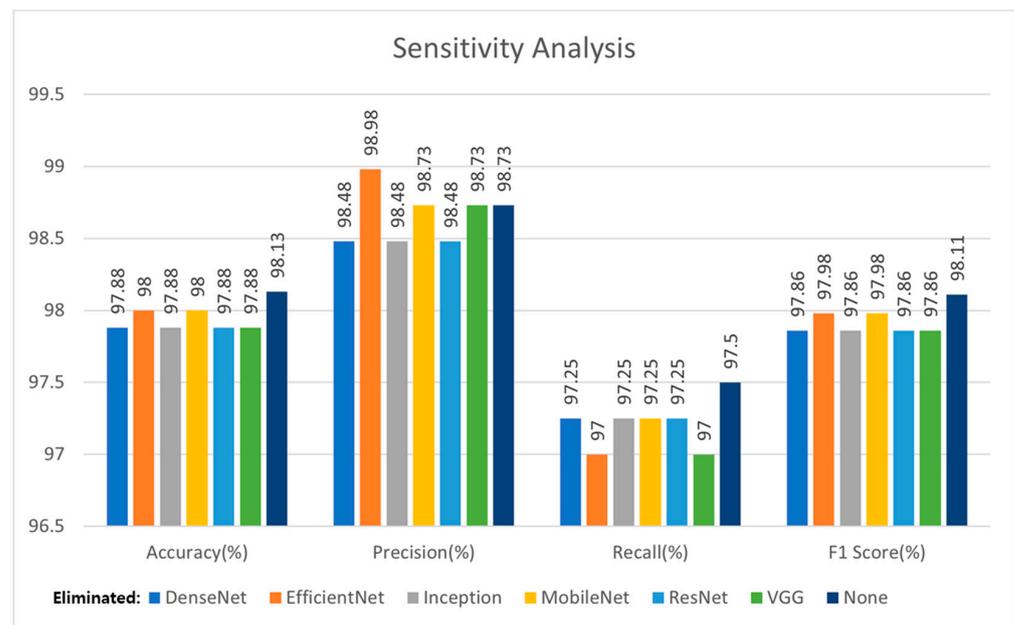


Figure 15. Sensitivity analysis results by eliminating one of the models.

The processing speed for the classification is a key factor for real-time applications. To verify this, the time required to classify an image was measured. Table 9 presents the results of this study. Classifying one image in the test set requires an average of 56.25 ms using only the CPU, which equates to approximately 17 FPS. Based on the speed of the ROUV, images can be sufficiently processed in real-time, even with ROUVs possessing a relatively low CPU performance.

Table 9. Elapsed time for classifying one image using the soft voting ensemble (CPU only).

	DenseNet	EfficientNet	Inception	MobileNet	ResNet	VGG	Voting
Time (ms)	14.25	14.23	6.90	4.95	6.72	9.20	56.25
FPS	70	70	144	202	148	108	17

6. Conclusions

In this study, a method for inspecting the condition of hull surfaces using images from an ROUV camera was proposed. The classification of images was achieved using a soft voting ensemble classifier comprising six well-known CNN models. To tune the models, they were retrained with images of the hull surfaces. The results of the implementation and experiments showed that the classification accuracy and F₁-score of the test set were approximately 98.13% and 98.11%, respectively. Furthermore, the proposed method was found to be highly applicable to ROUVs, which require real-time inspection performance.

However, the proposed method requires further improvement. As the dataset used in this study was collected from only a small number of inspection videos, the scope of the results of this study is limited. Therefore, many images that include various types of ships, underwater conditions, and lighting are needed. However, because ship owners are reluctant to provide hull images of their ships, collecting images is difficult. Therefore, in future studies, we plan to apply a data augmentation method using generative models to generate artificial images of the hull surfaces.

Author Contributions: Conceptualization, B.C.K.; data curation, B.C.K. and H.C.K.; formal analysis, B.C.K. and D.K.P.; funding acquisition, B.C.K.; investigation, B.C.K.; methodology, B.C.K.; project administration, B.C.K.; resources, S.H.; software, B.C.K. and H.C.K.; supervision, B.C.K.; validation, S.H. and D.K.P.; visualization, B.C.K. and H.C.K.; writing—original draft, B.C.K. and D.K.P.; writing—review and editing, B.C.K. and D.K.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea Government (MOE) (Project IDs: NRF-2020R1I1A3066259 and 2021RIS-004), and the Korea Evaluation Institute of Industrial Technology (KEIT) grant funded by the Korea Government (MOTIE) (Project IDs: 20012218 and RS-2022-00143813).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

1. Tribou, M.; Swain, G. The use of proactive in-water grooming to improve the performance of ship hull antifouling coatings. *Biofouling* **2010**, *26*, 47–56. [[CrossRef](#)] [[PubMed](#)]
2. Adland, R.; Cariou, P.; Jia, H.; Wolff, F.C. The energy efficiency effects of periodic ship hull cleaning. *J. Clean. Prod.* **2018**, *178*, 1–13. [[CrossRef](#)]
3. Hua, J.; Chiu, Y.-S.; Tsai, C.-Y. En-route operated hydroblasting system for counteracting biofouling on ship hull. *Ocean Eng.* **2018**, *152*, 249–256. [[CrossRef](#)]
4. Hewitt, C.; Campbell, M.L. *The Relative Contribution of Vectors to the Introduction and Translocation of Invasive Marine Species*; Marine Pest Sectoral Committee: Canberra, Australia, 2010.
5. Lee, M.H.; Park, Y.D.; Park, H.G.; Park, W.C.; Hong, S.; Lee, K.S.; Chun, H.H. Hydrodynamic design of an underwater hull cleaning robot and its evaluation. *Int. J. Nav. Arch. Ocean* **2012**, *4*, 335–352. [[CrossRef](#)]
6. Albitar, H.; Dandan, K.; Ananiev, A.; Kalaykov, I. Underwater robotics: Surface cleaning technics, adhesion and locomotion systems. *Int. J. Adv. Robot. Syst.* **2016**, *13*, 7. [[CrossRef](#)]
7. Yan, H.; Yin, Q.; Peng, J.; Bai, B. Multi-functional tugboat for monitoring and cleaning bottom fouling. *IOP Conf. Ser. Earth Environ. Sci.* **2019**, *237*, 022045. [[CrossRef](#)]
8. Kostenko, V.V.; Bykanova, A.Y.; Tolstonogov, A.Y. Underwater robotics complex for inspection and laser cleaning of ships from biofouling. *IOP Conf. Ser. Earth Environ. Sci.* **2019**, *272*, 022103. [[CrossRef](#)]
9. Song, C.; Cui, W. Review of underwater ship hull cleaning technologies. *J. Mar. Sci. Appl.* **2020**, *19*, 415–429. [[CrossRef](#)]
10. Lecun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [[CrossRef](#)]
11. Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; Fei-Fei, L. ImageNet: A large-scale hierarchical image database. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2009), Miami, FL, USA, 20–25 June 2009; pp. 248–255. [[CrossRef](#)]
12. Weiss, K.; Khoshgoftaar, T.M.; Wang, D. A survey of transfer learning. *J. Big Data* **2016**, *3*, 9. [[CrossRef](#)]
13. Ribani, R.; Marengoni, M. A Survey of transfer learning for convolutional neural networks. In Proceedings of the 2019 32nd SIBGRAPI Conference on Graphics, Patterns and Images Tutorials (SIBGRAPI-T), Rio de Janeiro, Brazil, 28–31 October 2019; pp. 47–57. [[CrossRef](#)]
14. Sherazi, S.W.A.; Bae, J.-W.; Lee, J.Y. A soft voting ensemble classifier for early prediction and diagnosis of occurrences of major adverse cardiovascular events for STEMI and NSTEMI during 2-year follow-up in patients with acute coronary syndrome. *PLoS ONE* **2021**, *16*, e0249338. [[CrossRef](#)] [[PubMed](#)]
15. Chang, C.L.; Chang, H.H.; Hsu, C.P. An intelligent defect inspection technique for color filter. In Proceedings of the 2005 IEEE International Conference on Mechatronics (ICM '05), Taipei, Taiwan, 10–12 July 2005; pp. 933–936. [[CrossRef](#)]
16. Jiang, B.C.; Wang, C.C.; Chen, P.L. Logistic regression tree applied to classify PCB golden finger defects. *Int. J. Adv. Manuf. Tech.* **2004**, *24*, 496–502. [[CrossRef](#)]
17. Zhang, X.; Liang, R.; Ding, Y.; Chen, J.; Duan, D.; Zong, G. The system of copper strips surface defects inspection based on intelligent fusion. In Proceedings of the 2008 IEEE International Conference on Automation and Logistics, Qingdao, China, 1–3 September 2008; pp. 476–480. [[CrossRef](#)]
18. Zhao, G.; Wang, T.; Ye, J. Anisotropic clustering on surfaces for crack extraction. *Mach. Vision Appl.* **2015**, *26*, 675–688. [[CrossRef](#)]
19. Siegel, M.; Gunatilake, P.; Podnar, G. Robotic assistants for aircraft inspectors. *IEEE Instrum. Meas. Mag.* **1998**, *1*, 16–30. [[CrossRef](#)]

20. Mumtaz, M.; Masoor, A.B.; Masood, H. A new approach to aircraft surface inspection based on directional energies of texture. In Proceedings of the 2020 20th International Conference on Pattern Recognition, Istanbul, Turkey, 23–26 August 2010; pp. 4404–4407. [[CrossRef](#)]
21. Amosov, O.S.; Amosova, S.G.; Iochkov, I.O. Deep neural network recognition of rivet joint defects in aircraft products. *Sensors* **2022**, *22*, 3417. [[CrossRef](#)]
22. Raouf, I.; Lee, H.; Kim, H.S. Mechanical fault detection based on machine learning for robotic RV reducer using electrical current signature analysis: A data-driven approach. *J. Comput. Des. Eng.* **2022**, *9*, 417–433. [[CrossRef](#)]
23. Carvalho, A.; Sagrilo, L.; Silva, I.; Rebello, J.; Carneval, R. On the reliability of an automated ultrasonic system for hull inspection in ship-based oil production units. *Appl. Ocean Res.* **2003**, *25*, 235–241. [[CrossRef](#)]
24. Akinfiyev, T.S.; Armada, M.A.; Fernandez, R. Nondestructive testing of the state of a ship's hull with an underwater robot. *Russ. J. Nondestruct.* **2008**, *44*, 626–633. [[CrossRef](#)]
25. Vaganay, J.; Elkins, M.; Willcox, S.; Hover, F.; Damus, R.; Desset, S.; Morash, J.; Pollidoro, V. Ship Hull Inspection by Hull-Relative Navigation and Control. In Proceedings of the OCEANS 2005 MTS/IEEE, Washington DC, USA, 17–23 September 2005; Volume 1, pp. 761–766. [[CrossRef](#)]
26. Negahdaripour, S.; Firoozfam, P. An ROV stereovision system for ship hull inspection. *Int. J. Ocean. Eng.* **2006**, *31*, 551–564. [[CrossRef](#)]
27. Navarro, P.; Iborra, A.; Fernández, C.; Sánchez, P.; Suardiá, J. A sensor system for detection of hull surface defects. *Sensors* **2010**, *10*, 7067–7081. [[CrossRef](#)]
28. Fernández-Isla, C.; Navarro, P.J.; Alcover, P.M. Automated visual inspection of ship hull surfaces using the wavelet transform. *Math. Probl. Eng.* **2013**, *2013*, 101837. [[CrossRef](#)]
29. Masi, G.D.; Gentile, M.; Vichi, R.; Bruschi, R.; Gabetta, G. Machine Learning approach to corrosion assessment in Subsea Pipelines. In Proceedings of the OCEANS 2015, Genova, Italy, 18–21 May 2015; pp. 1–6. [[CrossRef](#)]
30. Ortiz, A.; Bonnis-Pascual, F.; Garcia-Fidalgo, E.; Company-Corcoles, J.P. Vision-based corrosion detection assisted by a micro-aerial vehicle in a vessel inspection application. *Sensors* **2016**, *16*, 2118. [[CrossRef](#)] [[PubMed](#)]
31. Chin, C.S.; Si, J.T.; Clare, A.S.; Ma, M. Intelligent image recognition system for marine fouling using softmax transfer learning and deep convolutional neural networks. *Complexity* **2017**, *2017*, 5730419. [[CrossRef](#)]
32. Gormley, K.; McLellan, F.; McCabe, C.; Hinton, C.; Ferris, J.; David, I.K.; Scott, B.E. Automated image analysis of offshore infrastructure marine biofouling. *J. Mar. Sci. Eng.* **2018**, *6*, 2. [[CrossRef](#)]
33. CoralNet. Available online: <https://coralnet.ucsd.edu/> (accessed on 6 May 2022).
34. Bloomfield, N.J.; Wei, S.; Woodham, B.A.; Wilkinson, P.; Robinson, A.P. Automating the assessment of biofouling in images using expert agreement as a gold standard. *Sci. Rep.* **2021**, *11*, 2739. [[CrossRef](#)] [[PubMed](#)]
35. Liniger, J.; Jensen, A.L.; Pedersen, S.; Sørensen, H.; Mai, C. On the autonomous inspection and classification of marine growth on subsea structures. In Proceedings of the OCEANS 2022, Chennai, India, 21–24 February 2022; pp. 1–7. [[CrossRef](#)]
36. MNIST Database. Available online: <http://yann.lecun.com/exdb/mnist/> (accessed on 29 May 2022).
37. Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft COCO: Common objects in context. In Proceedings of the 2014 European Conference on Computer Vision (ECCV 2014), Zurich, Switzerland, 6–12 September 2014; pp. 740–755. [[CrossRef](#)]
38. Quattoni, A.; Torralba, A. Recognizing indoor scenes. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2009), Miami, FL, USA, 20–25 June 2009; pp. 413–420. [[CrossRef](#)]
39. Liu, Z.; Luo, P.; Wang, X.; Tang, X. Deep learning face attributes in the wild. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV 2015), Santiago, Chile, 7–13 December 2015; pp. 3730–3738. [[CrossRef](#)]
40. Khosla, A.; Jayadevaprakash, N.; Yao, B.; Fei-Fei, L. Novel dataset for fine-grained image categorization. In Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2011), Colorado Springs, CO, USA, 21–23 June 2011.
41. Nilsback, M.; Zisserman, A. Automated flower classification over a large number of classes. In Proceedings of the 2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing, Bhubaneswar, India, 16–19 December 2008; pp. 722–729. [[CrossRef](#)]
42. Chin, C. Marine Fouling Images. IEEE Dataport. Available online: <https://ieee-dataport.org/documents/marine-fouling-images> (accessed on 6 May 2022).
43. Shihavuddin, A. Coral Reef Dataset. Mendeley Data. Available online: <https://data.mendeley.com/datasets/86y667257h/2> (accessed on 6 May 2022).
44. O'Byrne, M.; Pakrashi, V.; Schoefs, F.; Ghosh, B. Semantic segmentation of underwater imagery using deep networks trained on synthetic imagery. *J. Mar. Sci. Eng.* **2018**, *6*, 93. [[CrossRef](#)]
45. SLM Global. Available online: <http://www.slm-global.com/> (accessed on 6 May 2022).
46. Huang, G.; Liu, Z.; van der Maaten, L.; Weinberger, K.Q. Densely Connected Convolutional Networks. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2017), Honolulu, HI, USA, 21–26 July 2017; pp. 2261–2269. [[CrossRef](#)]
47. Tan, M.; Le, Q. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. In Proceedings of the 36th International Conference on Machine Learning (ICML), Long Beach, CA, USA, 9–15 June 2019; pp. 6105–6114.

48. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the inception architecture for computer vision. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2016), Las Vegas, NV, USA, 27–30 June 2016; pp. 2818–2826. [[CrossRef](#)]
49. Szegedy, C.; Ioffe, S.; Vanhoucke, V.; Alemi, A.A. Inception-v4, Inception-ResNet and the Impact of residual connections on learning. In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI 2017), San Francisco, CA, USA, 4–9 February 2017; pp. 4278–4284.
50. Chollet, F. Xception: Deep Learning with depthwise separable convolutions. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2017), Honolulu, HI, USA, 21–26 July 2017; pp. 1251–1258. [[CrossRef](#)]
51. 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* **2017**, arXiv:1704.04861. [[CrossRef](#)]
52. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.-C. MobileNetV2: Inverted residuals and linear bottlenecks. In Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2018), Salt Lake City, UT, USA, 18–22 June 2018; pp. 4510–4520.
53. Howard, A.; Sandler, M.; Chu, G.; Chen, L.-C.; Chen, B.; Tan, M.; Wang, W.; Zhu, Y.; Pang, R.; Vasudevan, V.; et al. Searching for MobileNetV3. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV 2019), Seoul, Korea, 27 October–2 November 2019; pp. 1314–1324. [[CrossRef](#)]
54. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2016), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778. [[CrossRef](#)]
55. He, K.; Zhang, X.; Ren, S.; Sun, J. Identity mappings in deep residual networks. In Proceedings of the 2016 European Conference on Computer Vision (ECCV 2016), Amsterdam, The Netherlands, 11–14 October 2016; pp. 630–645. [[CrossRef](#)]
56. Simonyan, K.; Zisserman, A. Very Deep convolutional networks for large-scale image recognition. In Proceedings of the 3rd International Conference on Learning Representations (ICLR 2015), San Diego, CA, USA, 7–9 May 2015.
57. Lin, M.; Chen, Q.; Yan, S. Network in network. In Proceedings of the 2nd International Conference on Learning Representations (ICLR 2014), Banff, Canada, 14–16 April 2014.
58. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.
59. Kingma, D.P.; Ba, L.J. Adam: A Method for stochastic optimization. In Proceedings of the 3rd International Conference on Learning Representations (ICLR 2015), San Diego, CA, USA, 7–9 May 2015.
60. Bergstra, J.; Bengio, Y. Random search for hyper-parameter optimization. *J. Mach. Learn. Res.* **2012**, *13*, 281–305.