



Received: 2025/08/21
Revised: 2025/09/02
Accepted: 2025/09/16
Published: 2025/09/30

***Corresponding Author:**

Dooyoung Kim

Dept. of Artificial Intelligence, Republic of Korea
Naval Academy
1, Jungwon-ro, Jinhae-gu, Changwon-si,
Gyungsangnam-do, 51704, Republic of Korea
Tel: +82-2-907-5246
E-mail: dykim07@navy.ac.kr

CNN 기반의 선저 표면 손상 유형 분류 모델

CNN-based Classification Model of Ship Hull Surface Damage

양민우¹, 김두영^{2*}

¹해군사관학교 인공지능학과 사관생도

²해군 중령(진)/해군사관학교 인공지능학과 부교수

Minwoo Yang¹, Dooyoung Kim^{2*}

¹Midshipman, Dept. of Artificial Intelligence, Republic of Korea Naval Academy

²CDR(Sel.), ROK Navy/Associate professor, Dept. of Artificial Intelligence,
Republic of Korea Naval Academy

Abstract

본 논문에서는 GRN 기반의 CNN 모델을 통해 함정 표면의 손상을 분류하고, 그 분류 근거를 시각화하여 제시하였다. 먼저, 자체 CNN 모델을 설계하여 선박의 손상 6개의 클래스를 분류하였다. 실험 결과, 정확도 97.6%로 6개의 클래스를 분류하였다. 다음으로 자체 CNN 모델의 분류 근거를 Grad-CAM을 활용하여 시각화하였다.

This study presented a GRN-based customized CNN model to classify naval/maritime ship damage and to visualize the decision basis of the classification. The custom CNN classified ship damage into six categories, achieving a classification accuracy of 97.6%. Furthermore, the decision-making process of the model was visualized using Grad-CAM, which highlighted the key regions contributing to the classification.

Keywords

합성곱 신경망(Convolutional Neural Network),
설명 가능한 AI(Explainable AI),
선저 검사(Hull Inspection),
분류 문제(Classification Problem),
경량 딥러닝(Lightweight Deep Learning)

1. 서론

선박 검사란, 선박 시설 결함으로 인한 해양사고를 방지하는 한편 해양 사고로 인한 인명·선박·화물의 안전을 확보하기 위하여 선박의 주요시설에 대한 기준적합 여부를 확인하는 절차이다. 검사 유형에는 건조검사, 정기검사, 중간검사, 임시검사, 임시항해검사, 국제협약검사가 있으며, 이 중 중간검사는 정기검사와 정기검사 사이에 수행되는 보완된 검사[1]로서 선박의 이상 유무를 보다 짧은 주기로 식별할 수 있다는 장점이 있다. 이러한 검사 중 3년에 한 번씩 의무적으로 수행되는 선저 검사[2]는 선박 하부의 도장 상태, 부착 생물, 부식, 균열 등을 육안 혹은 장비를 사용하여 종합적으로 점검하는 검사이다. 선저검사는 선박의 해상 안전, 환경 보호, 그리고 선박의 구조 안정성 등을 보장하기 위한 필수적 절차이다. 민간 선박의 선저 검사는 민간 선급기관 및 대행업체에서 수행되고 있으며[3], 대한민국 해군의 경우 출동 임무를 완료한 함정에 대해 SSU 요원을 활용하거나, 함정을 정기적으로 견선거에 입거하는 등의 방법을 통해 선저 검사를 실시하고 있다.

최근에는 선저 검사의 효율성과 정밀도의 향상을 위해 인공지능 기반 자동화 기술의 도입이 활발히 이루어지고 있다[4]. ROV(remotely operated vehicle)를 활용한 부식 탐지, 영상 분류, 다중 손상 인식 알고리즘 등 다양한 연구가 진행되고 있으며, 이는 검사자의 주관적 판단에 의존하던 전통적 방식에서 정량적이고 반복 가능한 자동 평가 시스템으로 전환되고 있음을 보여준다.

본 연구는 이러한 흐름을 반영해, 선저 표면 손상 유형 분류 정확도를 향상시킨 CNN 모델을 제안하였다.

2. 관련 연구

Widiarti et al.[5]은 ROV(드론)으로 촬영한 선체의 외판 이미지를 활용해 선체의 부식 정도를 low, medium, severe의 3단계로 분류하였다. LeNet 기반의 CNN 모델을 제안하여 실제 수상 및 수중 환경에서 선체 외판의 부식 정도를 평가한 것이 특징이다. 수상 환경에서의 테스트 데이터셋에 대한 정확도는 91.1%, 수중 환경에서는 86.6%의 정확도를 보였으나 severe 단계의 부식이 수상·수중 환경의 노이즈로 인해 클래스 오분류 문제가 발생하였다.

Li et al.[6]은 선체의 도장에 발생한 불량을 분류하는 연구를 통해 sagging, orange skin 등 6가지의 클래스로 분류하였다. IDATLWGAN(Improved Deep Adversarial Transfer Learning with Wasserstein GAN)이라는 모델을 구축하여 CNN 기반의 분류 및 GAN 기반 데이터 증강을 결합하였으며, zero-shot 방식을 도입해 학습 시 새로운 클래스에 대한 분류가 가능하도록 설계된 것이 특징이다. 전체 분류 정확도는 91.7% - 94.7%로, 높은 정확도뿐만 아니라 zero-shot을 통한 새 손상 유형에 대한 확장성을 보이고 있지만, 선체 도장 부분에만 한정되어 있어 다른 환경 및 도메인에서의 일반화가 검증되지 않았다[6].

Chin et al.[7]은 선체 및 해양 구조물 표면에 붙은 biofouling 정도를 자동으로 평가하도록 한 연구를 수행하여 biofouling 정도를 정상, 따개비 등 10개의 클래스로 분류하였다. GoogLeNet Inception V3 기반의 아키텍처로 ImageNet의 사전 가중치를 전이 학습하여 실제 선체 및 해양 구조물의 biofouling 정도를 평가한 것이 특징이다. 10개 클래스에 대한 정확도는 67% - 89%으로, 같은 데이터셋에서 SVM, random forest 등의 머신러닝 모델 대비 약 10% 정도의 높은 정확도를 기록하였다[7].

3. 모델 구축

본 연구에서는 다양한 환경에서 촬영한 선박 손상 이미지에 대한 정량적 성능이 향상된 선박 손상 분류

모델을 제시한다. 선박 손상 이미지 데이터셋을 구축하여 수중, 건선거 등 다양한 환경에서의 선저 검사를 가능하게 하였으며, 기존 선행 연구 대비 매개변수 수와 연산량을 줄여 경량화하였다.

3.1 학습 데이터 구축

본 연구에서는 선박 관련 국제기관에서 언급한 주요 손상 중 대표적인 5가지[8-12]와 정상 상태를 선정하여 총 6가지 선박 손상 클래스의 데이터셋을 구축하였다. Biofouling(생체 부착물)[13], corrosion(부식)[14-16], crack(균열), deformation(부풀음), normal(정상)[17], paint peel(페인트 벗겨짐)[14,17]과 관련된 공개 이미지 데이터셋을 이용하여 통합 선박 손상 이미지 데이터셋을 구성하였다. 이때, 원활한 학습을 위해 앞서 설명한 6개의 클래스와 관련된 이미지를 224×224 픽셀로 다운샘플링한 후 정규화를 거쳐 저장하였다.

3.2 모델 구조 및 학습

연구에 사용되는 데이터셋이 이미지임을 고려해 이미지 분류 연구의 대표적 범주 중 하나인 CNN을 기반으로 분류 모델을 구축하였다. Table 1은 본 연구에서 구축한 CNN 모델의 구조를 나타낸 것이다.

본 연구에서 제안한 모델은 크게 특징 추출기와 분류기로 구성하였다. 특징 추출기는 입력 이미지로부터 유용한 시각적 특징을 추출하는 역할을 수행하며, 총 5개의 계층(block)으로 구성하였다. 이 중 첫 번째 블록은 3×3 커널 크기의 합성곱(Conv) 연산을 수행한 후, batch normalization[18] 및 ReLU6[19] 활성화 함수를 거쳐 2×2 커널, stride값이 2인의 max pooling[20] 계층을 통과한다. 해당 블록에서는 이미지 크기를 절반으로 줄이고 중요한 특징만을 유지할 수 있도록 한다. 두 번째 블록에서는 연산량을 줄이기 위하여 depthwise separable convolution[21] 구조를 적용하였다. 또한 depthwise와 pointwise 연산을 분리하여 계산 효율을 높이는 동시에 표현력을 유지하도록 하였다. 이후 세 번째 블록에서는 다양한 스케일의 정보를 포착하기 위하여 multi-scale convolution[22] 구조를 도입하였다. 이는 입력 feature map을 3×3, 5×5, 그리고 dilation 값으로 2가 적용

Table 1. Structure of proposed CNN model

| Layer | Hyper-parameters |
|---------------------------|-----------------------------------------------------------------------------------------------------------|
| Conv1 | in_channels = 3, out_channels = 32, kernel = 3×3, stride = 1, padding = 1, ReLU6 |
| MaxPool1 | kernel = 2×2, stride = 2 |
| DW(ConvBlock1 (Layer 2)) | Depthwise: in = 32, groups = 32, Pointwise: out = 64, BatchNorm, ReLU6 |
| MultiScaleBlock(Layer 3) | Parallel branches: 3×3 conv, 5×5 conv → Dilated 3×3 convs → BatchNorm + ReLU6 + GRN, Concat: out = 126 |
| DW(ConvBlock2(Layer 4)) | Depthwise + Pointwise conv: in = 126, out = 192, BatchNorm, ReLU6 |
| Classifier Depthwise Conv | in = 192, groups = 192, kernel = 3×3, padding = 1 |
| Classifier Pointwise Conv | in = 192, out = 128, BatchNorm, ReLU6 |
| AdaptiveAvgPool2d | Output: [batch, 128, 1, 1] |

된 3×3 커널로 각각 병렬 처리한 후 출력을 결합하는 방식이다. 각 branch에서는 batch normalization, ReLU6, GRN(global response normalization)[23]을 적용해 각 채널의 전역 반응을 정규화하고 중요도를 동적으로 조정한다. 네 번째 블록은 다시 depthwise separable convolution 구조를 적용한 계층으로, 앞선 멀티 스케일 블록의 출력을 받아 추가적인 특징을 추출한다. 다섯 번째 특징 추출기 블록은 3×3 depthwise separable convolution으로 구성되며, 이후 batch normalization 및 ReLU6를 거쳐 adaptive average pooling[24]를 수행한다. 이로써 feature map의 공간 정보를 1×1으로 압축하며, 출력은 128차원의 채널 벡터로 나타낸다.

분류기는 단일 fully connected(이하 FC) 신경망으로 구성된다. 먼저 flatten layer를 통해 (128, 1, 1)의 feature map을 1차원 벡터 (128)로 변환한다. 이후 최종 FC layer를 통과해 6개의 클래스에 대한 출력확률을 가지는 벡터로 매핑된다. 이 출력 벡터는 입력 이미지가 6개의 손상 클래스 중 어느 하나에 속할 확률 분포를 나타내며, softmax 함수를 통해 최종 분류 확률을 계산한다.

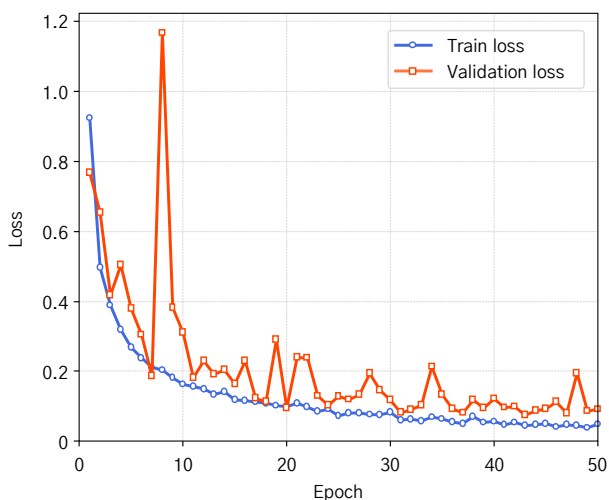
모델 학습 과정에서는 Google Colab 환경에서 Tesla T4-GPU를 활용하였고, Pytorch 프레임워크를 사용하였다. corrosion 클래스의 데이터가 다른 클래스에 비해 현저히 적은 클래스 간 불균형 문제를 고려하여 데이터 수에 역비례하는 weighted cross entropy[25]를 사용해 해결하고자 하였다. 최적화 기법은 adam optimizer[26]을 사용하였으며, 학습률

은 0.0005, weight decay는 0.00001로 설정하였다. 각 batch는 32개의 데이터로 고정하였고 학습은 총 50 epoch 진행하였다. 학습 후 도출된 결과를 바탕으로 hyper-parameter를 조정하였다.

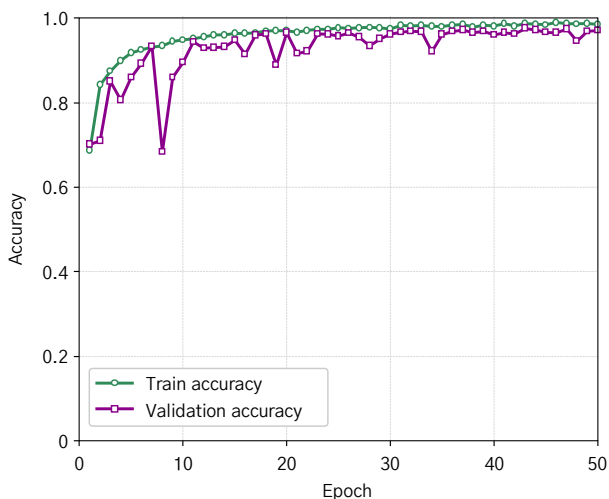
4. 실험 결과

Fig. 1은 각 epoch마다 제안된 모델의 학습 및 검증 데이터셋에 대한 정확도와 손실률을 나타낸 곡선이다. 학습 초기에는 검증 정확도가 일정하지 않은 변동성을 보였으며, 특히 8 epoch에서는 학습률 초기 설정의 영향으로 일시적인 성능 저하가 나타났다. 그러나 8 epoch 이후 검증 정확도가 점차 상승하며, 안정적인 학습 흐름을 나타냈고, 학습 정확도와 비슷한 추세를 보이며 전반적으로 수렴하는 양상을 보였다. 최종적으로 50 epoch 학습 이후 모델의 검증 정확도는 97.49 %를 기록하였다. 가장 높은 검증 정확도의 해당 epoch의 가중치 파일을 로드하여 별도의 테스트 데이터셋에 적용한 결과, accuracy 97.6 %, f1-score 0.975, precision 0.976을 기록하였다. 이는 제안한 모델이 과적합을 억제하면서도 실제 테스트 데이터에 대해서도 강건한 예측이 가능하다는 것을 의미한다. 특히, 다중 클래스 간 분류 상황에서 precision과 f1-score가 모두 0.97 이상이라는 점에서 클래스 간 균형이나 분류 오류가 최소화되었음을 의미한다.

손상 클래스별 모델의 예측 정확도를 확인하기 위하여 이미지의 실제 라벨(true)과 예측된 라벨(predicted)을 Fig. 2와 같이 혼동 행렬(confusion mat-



(a) Train vs. validation loss



(b) Train vs. validation accuracy

Fig. 1. Learning curve of the proposed model

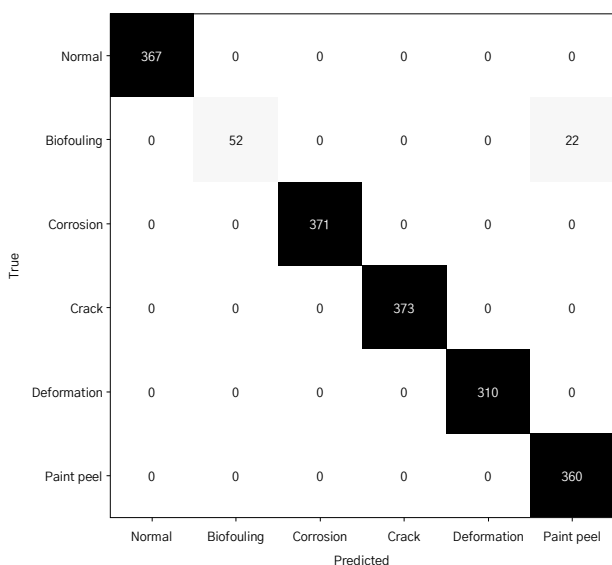


Fig. 2. Confusion matrix of the test result

rix)로 도식화하였다. 앞서 제시했듯, 테스트 데이터셋에 대해 분류 정확도는 97.6 %로 대부분의 함정 손상 클래스를 높은 정확도로 판별하였다. 클래스별로 살펴보면, biofouling, crack, deformation, normal 클래스는 각각 100.0 %, 98.93 %, 99.47 %, 99.36 %의 정확도를 기록하며 매우 우수한 분류 성능을 보였다. 또한, paint peel 클래스는 전체 375개 중 360개를 정확히 분류하여 96.00 %의 정확도를 기록하였다. 반면 corrosion 클래스의 정확도는 70.27 %로 비교적 낮았으며 총 74개의 corrosion 이미지 중 52개만 정확하게 분류하였다. 이는 전체 데이터셋 중 corrosion class의 데이터 수가 다른 클래스에 비해 현저히 부족했기 때문으로 분석된다. 또한, corrosion 클래스를 paint peel로 오분류한 경우가 총 22회 발생하였다. 이는 해당 클래스가 유사한 시각적 특성을 공유하고 있을 가능성이 크며, 제안된 CNN 모델의 성능이 이러한 차이를 명확히 구분하기에는 제한적이라는 것을 의미한다.

Fig. 3(d)는 올바르게 crack 클래스로 분류한 것에 대한 근거를 Grad-CAM으로 시각화한 것이다. 해당 그림에서 실제 crack이 Grad-CAM에서 초록색 계열의 colormap으로 표시되어 있으며, 이를 통해 제안된 모델이 crack이 발생한 부분을 정확한 근거로 분류했음을 알 수 있다. 반면 Fig. 3(e)는 corrosion 클래스를 paint peel 클래스로 잘못 분류한 경우를 나타낸 것이다. 앞서 설명한 바와 같이, corrosion 클래스가 paint peel 클래스와 유사한 시각적 성질을 띠고 있어 분류를 정확히 하지 못한 것으로 보인다. 또한, corrosion 클래스의 데이터 수가 다른 클래스에 비해 현저히 적었기 때문에 오분류했을 가능성이 높다. Fig. 3(f)의 경우 분류 자체는 올바르게 수행되었으나 Grad-CAM 시각화에서는 색상 패턴이 반전되어 나타났다. deformation 클래스는 원형의 손상을 특징으로 하며, 일반적으로 손상 부위가 초록색으로 시각화되어야 한다. 그러나 Fig. 3(f)에서는 원형 손상 주위로 초록색 계열의 colormap으로 시각화되었으며, 이는 모델이 손상 영역과 정상 영역을 반대로 인식하여 이를 판단 근거로 활용했음을 보여준다. 즉 deformation 클래스가 normal 클래스와 경계가 유사해 모델이 실제 손상 부위 대신 주변을 근거로 인식했을 가능성이 높으며, 모델이 손상 내부보다는 경계를 주요 특징으로 학습했음을 의미한다.

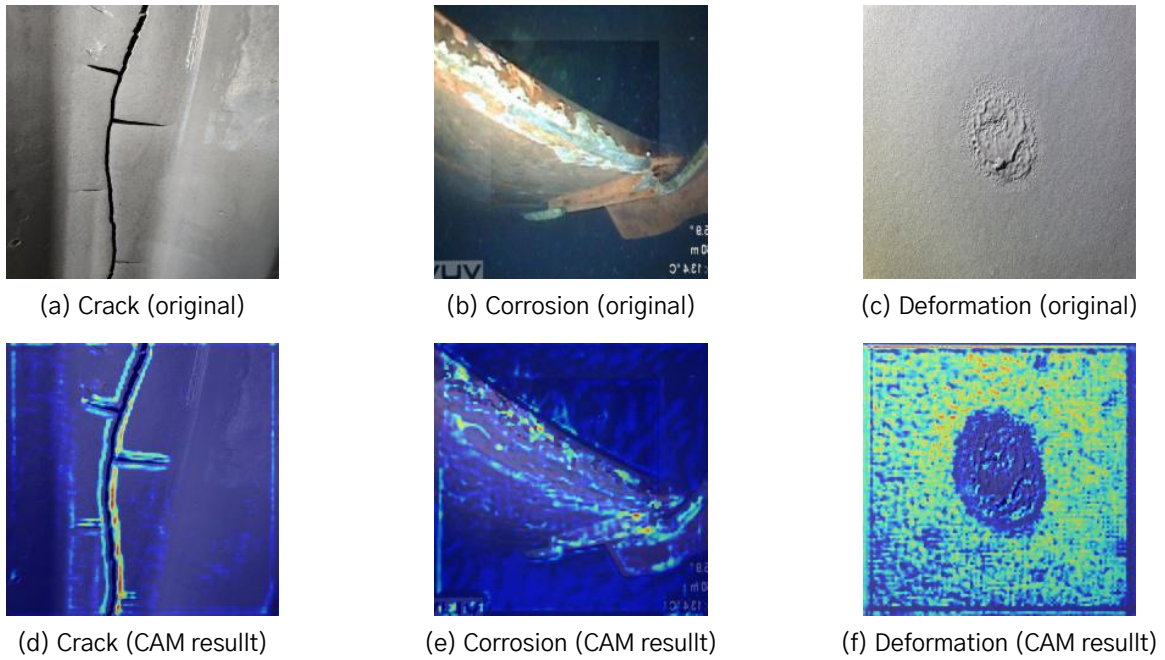


Fig. 3. CAM results of the test dataset

Table 2. Comparison of current research and prior research

| Research | Proposed model | Accuracy |
|-----------------------------------------------------------------|--------------------------------|-----------------------------------------|
| Corrosion Detection on Ship Hull Using ROV (Widiatri et al.[5]) | Customized CNN based on VGGNet | 86.6 % (underwater) 91.1 % (surface) |
| A Zero-Shot Image Classification Method (Li et al.[6]) | IDATLWGAN based on CNN and GAN | 91.7 % - 94.3 % |
| Intelligent Image Recognition System (Chin et al.[7]) | Customized CNN based on LeNet | 67 % - 89 % |
| This research | Customized CNN based on GRN | 97.6 % |

본 연구는 앞서 언급한 기존 선행 연구의 연장선상에서 진행되었다. Table 2는 이들 선행 연구에 대한 제안 모델과 정확도를 표로 정리한 것이다. Widiarti et al.[5]의 경우, 단순 부식에 한해 수중에서 86.6 %, 수상에서 91.1 %에 그쳤다. Li et al.[6]도 선박 도장 부분이라는 단일 도메인에 한해 91.7 % - 94.3 %의 정확도를 보였다. Chin et al.[7] 또한 단일 biofouling에 한해 67 % - 89 %의 정확도를 나타냈다. 하지만 본 연구에서는 수중, 수상, 선박 도장 부분의 3가지 도메인에서 normal을 포함한 6가지 선저 및 선체 표면 손상 유형을 97.6 %의 정확도로 분류하였으며, 단순 CNN 구조를 적용하는 것이 아닌 GRN 기반의 경량 네트워크를 새롭게 구축하였음에도 불구하고 선행 연구 대비 높은 정확도를 보였다.

5. 결론

앞서 설명한 바와 같이, 선박의 손상 파악은 항해 안전에 있어 필수적이며, 이를 위해서는 철저한 객관성과 정확성이 요구된다.

본 연구에서는 일반적으로 사용하는 이미지 분류 기술 중 CNN을 채택하여 선박 및 함정의 선체-선저 손상 파악을 수행함으로써 선저 검사의 자동화에 기여하는 방식을 제안하였으며, 일반 선박의 이미지 데이터를 수집하여 다운샘플링 및 정규화로 전처리된 이미지 데이터셋을 구축하였다. 이후, CNN 기반의 분류 모델을 제안하여 위의 데이터셋을 통해 모델의 학습, 검증 및 테스트를 수행한 결과 최종 테스트 정확도 97.6 %의 뛰어난 결과를 얻을 수 있었다. 또한,

자동 분류에만 국한되지 않고 Grad-CAM을 활용해 제안한 모델이 선체 손상을 분류 근거를 시각화하여 확인할 수 있었으며 누구나 제안된 모델을 쉽게 이용할 수 있도록 하였다. 현재 군에서 다양한 무인자산의 활용이 증가하고 있는 추세에서 제안된 모델의 활용 가능성은 매우 높을 것으로 기대된다.

본 연구는 딥러닝 기술을 활용해 다양한 선박 손상 사례 중 기존 연구의 성능을 개량하는 데 그 의미가 있다. 또한 같은 주제를 다루는 segmentation 모델 대비 경량이며, 노이즈에 덜 민감하다는 이점이 있다. 이 점을 바탕으로, 단순히 선박 손상의 종류를 분류하는 것에 그치지 않고 선박의 손상을 탐지 및 실시간으로 근거를 시각화하는 연구로 확장하고자 한다.

참고문헌

- [1] 국토교통부, 선박검사제도 개요. (2012). https://www.molit.go.kr/USR/policyData/m_34681/dtl?id=143&dtl?id=143&
- [2] 선박안전법 시행규칙 제19조 2항. (2025).
- [3] 선박안전법 제60조. (2023).
- [4] 조원진, “선박 검사 프로세스에 AI 도입...디지털 전환 전략 속대 낸다,” 서울경제 (2023. 12. 14). <https://sedaily.com/NewsView/29YIDEXCC>
- [5] Widiarti, E., Fitriyani, D., Rachmawati, R. (2024). “Corrosion Detection on Ship Hull Using ROV Based on Convolutional Neural Network,” *International Journal of Marine Engineering Innovation and Research*, Vol. 9, No. 1, pp. 1-7.
- [6] Li, X., Zhang, Y., Wang, H., Chen, J. (2024). “A Zero-Shot Image Classification Method of Ship Coating Defects Based on IDATLWGAN,” *Journal of Marine Science and Engineering*, Vol. 12, No. 3, 456.
- [7] Chin, C. S., Si, J., Clare, A. S., Ma, M. (2017). “Intelligent Image Recognition System for Marine Biofouling Using Softmax Transfer Learning and Deep Convolutional Neural Networks,” *Complexity*, 2017, Article ID 5730419.
- [8] International Maritime Organization. (2023). “Guidelines for the Control and Management of Ships’s Biofouling to Minimize the Transfer of Invasive Aquatic Species (MEPC.378(80)).”
- [9] American Bureau of Shipping. (2017). “Guidance Notes on the Inspection, Maintenance and Application of Marine Coating Systems. American Bureau of Shipping.”
- [10] Hull integrity and Ship Maintenance Software – ShipManager Hull. <https://dnv.com/services/hull-integrity-and-ship-maintenance-software-shipmanager-hull-1531/>
- [11] International Association of Classification Societies. 2014. “Recommendation No. 42 Rev.1 – Guidelines for Surveys, Assessment and Repair of Hull Structure.”
- [12] DNV. (2016). “Rules for Classification: Ships, Part 3 Chapter 1 – General Principles.”
- [13] Automating the Assessment of Biofouling in Images and Video Footage [Dataset]. (2024). https://figshare.com/articles/dataset/Automating_the_assessment_of_biofouling_in_images/26537158
- [14] LIACi Semantic Segmentation Dataset for Underwater Ship Inspections. (2024). <https://data.sintef.no/product/dp-9e112cec-3a59-4b58-86b3-ecb1f2878c60>
- [15] MaVeCoDD Dataset: Marine Vessel Hull Corrosion in Dry-Dock Images. (2021). <https://data.mendeley.com/datasets/ry392rp8cj/1>
- [16] Fushi Computer Vision Project. (2025). <https://universe.roboflow.com/liangs/fushi-045qu>
- [17] 선박 도장 품질 데이터. (2022). <https://www.aihub.or.kr/aihubdata/data/view.do?currMenu=115&topMenu=100&dataSetSn=71447>
- [18] Ioffe, S., & Szegedy, C. (2015). “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift,” In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*. arXiv: 1502.03167
- [19] Krizhevsky, A. (2009). “Convolutional Deep Belief Networks on CIFAR-10.” Technical Report.
- [20] Yamaguchi, T., Takahashi, S., Minematsu, N., & Hirose, K. (1990). “Neural Network for Speaker-Independent Isolated Word Recognition,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. 38, No. 11, pp. 1842-1845.
- [21] Chollet, F. (2017). “Xception: Deep Learning with Depthwise Separable Convolutions,” In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp.1251-1258.
- [22] Xu, Y., et al. (2014). “Scale-Invariant Convolutional Neural Networks.” arXiv:1411.6369.
- [23] Woo, S., et al. (2024). “ConvNeXt V2: Co-designing and Scaling ConvNets with Masked Autoencoders.” arXiv preprint:2405.03301v1.
- [24] Lee, C.-Y., Gallagher P.W., & Tu, Z. (2015). “Generalizing Pooling Functions in Convolutional Neural Networks: Mixed, Gated, and Tree,” In *Proceedings of the 18th International Conference on Artificial Intelligence and Statistics (AISTATS)*.
- [25] Yue, S., et al. (2022). “Imbalanced Malware Images Classification: a CNN Based Approach,” arXiv:1708.08042.
- [26] Kingma D. P., & Ba, J. (2014). Adam: A method for Stochastic Optimization.