



합성곱 신경망을 이용한 프로펠러 캐비테이션 침식 위험도 연구

김지혜¹·이형석²·허재욱^{2,†}
창원대학교 조선해양공학과¹
현대중공업 선박해양연구소 선박성능연구실²

A Study on the Risk of Propeller Cavitation Erosion Using Convolutional Neural Network

Ji-Hye Kim¹·Hyoungseok Lee²·Jea-Wook Hur^{2,†}
Department of Naval Architecture and Marine Engineering, Changwon National University¹
Ship Performance Research Department, Hyundai Maritime Research Institute, Hyundai Heavy Industries²

This is an Open-Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License(<http://creativecommons.org/licenses/by-nc/3.0>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

Cavitation erosion is one of the major factors causing damage by lowering the structural strength of the marine propeller and the risk of it has been qualitatively evaluated by each institution with their own criteria based on the experiences. In this study, in order to quantitatively evaluate the risk of cavitation erosion on the propeller, we implement a deep learning algorithm based on a convolutional neural network. We train and verify it using the model tests results, including cavitation characteristics of various ship types. Here, we adopt the validated well-known networks such as VGG, GoogLeNet, and ResNet, and the results are compared with the expert's qualitative prediction results to confirm the feasibility of the prediction algorithm using a convolutional neural network.

Keywords : Convolutional Neural Network(CNN, 합성곱 신경망), Deep learning(딥러닝), Propeller(프로펠러), Cavitation(캐비테이션), Erosion(침식)

1. 서론

선박용 프로펠러에서 발생하는 캐비테이션은 생성 위치 및 거동 특성에 따라 얇은층캐비테이션(sheet cavitation), 버블캐비테이션(bubble cavitation), 날개끝 보텍스캐비테이션(tip vortex cavitation), 구름형 캐비테이션(cloud cavitation) 등으로 구분된다. 이 중 특정 조건에서 불안정한 거동을 보이는 캐비테이션의 경우, 기포가 붕괴되면서 프로펠러 표면에 직접적인 충격을 주어 침식의 주요 원인이 될 수 있다. 프로펠러 날개에서 발생하는 침식은 구조 강도를 저하시켜 프로펠러의 파손을 야기할 수 있기 때문에 설계 단계에서 주의가 필요하며, 이러한 캐비테이션의 침식 위험도는 실험 및 실적선 자료를 기반으로 각 기관마다 정성적으로 판단하고 있다. 본 연구에서는 프로펠러에서 발생하는 캐비테이션 침식 위험도를 정량적으로 지표화 할 수 있는 방법의 일환으로 합성곱 신경망(convolutional

neural network)을 기반으로 딥러닝(deep learning) 알고리즘을 구현하였다. 이미지의 시각적 특성을 감지하는 것을 주 목적으로 하는 합성곱 신경망 기반의 딥러닝 알고리즘 구현을 위해 파이썬(Python)으로 작성된 오픈 소스 신경망 라이브러리인 케라스(Keras)를 사용하였다. 최근 딥러닝을 이용하여 이미지를 통해 프로펠러에서 발생하는 캐비테이션을 탐지하고, 수치 해석 결과를 학습시킴으로써 프로펠러 형상에 따른 캐비테이션의 크기를 예측하는 등의 연구들이 활발히 수행되고 있다(Shora et al., 2017; Chao et al., 2020). 본 연구를 통해 구현된 합성곱 신경망 기반의 딥러닝 알고리즘은 모형시험 결과 획득한 다양한 조건과 선종에 대한 캐비테이션 침식 패턴을 학습하고, 학습 결과를 바탕으로 침식 위험도가 높은 캐비테이션 패턴의 예측 정확도를 높이고자 하였다. 이 때, 기 개발되어 성능이 검증된 네트워크 모델인 VGG(Simonyan & Zisserman, 2014), GoogLeNet (Szegedy et al., 2015),

ResNet (He et al., 2016) 등을 적용하여 침식 위험도가 높은 캐비테이션 패턴을 학습하고, 학습된 각 모델의 예측 정확도를 비교하여 합성곱 신경망을 이용한 프로펠러 캐비테이션 침식 위험도 예측 알고리즘의 타당성을 확인하였다.

2. 딥러닝 알고리즘 구현

딥러닝은 데이터 표현을 학습하기 위해 층(layer)을 겹겹이 쌓아 올려 구성된 신경망(neural network)이라는 모델을 사용하는 기계 학습 알고리즘의 집합으로 정의된다. 여기서 층은 하나 이상의 텐서를 입력으로 받아 하나 이상의 텐서를 출력하는 데이터 처리 모듈이며, 어떤 층에서 입력 데이터가 처리되는 변환은 그 층의 가중치(weight)를 파라미터(parameter)로 가지는 함수로 표현된다. 따라서, 딥러닝을 통한 학습은 주어진 입력을 정확한 목표 값에 매핑하기 위해 신경망의 모든 층에 있는 가중치 값을 찾는 과정을 의미한다. 신경망의 예측 정도를 높이기 위해서는 신경망을 통한 예측 값과 목표 값의 차이를 파악할 필요가 있으며, 이는 신경망의 손실 함수(loss function) 또는 목적함수(objective function)를 통해 수행한다. 또한 이 차이를 이용하여 가중치를 수정하는 과정은 딥러닝의 핵심 알고리즘인 역전파(backpropagation) 알고리즘을 구현한 옵티마이저(optimizer)가 담당한다. 다시 말해 딥러닝 학습이란 위의 훈련 과정을 반복하며 손실 함수를 최소화하는 가중치 값을 산출하는 과정(Fig. 1)을 말한다 (Chollet, 2017).

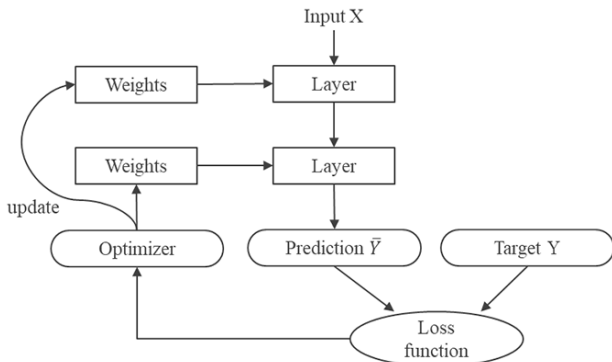


Fig. 1 Basic deep learning structure

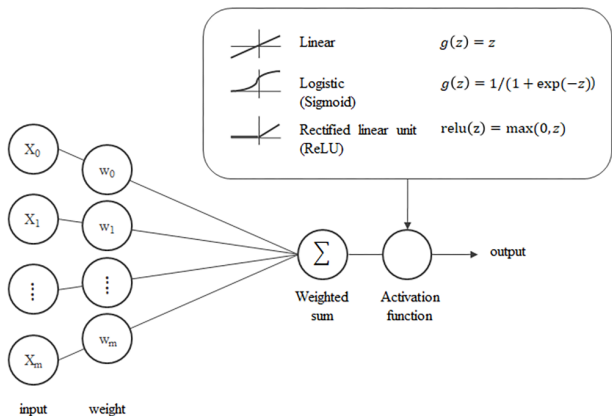


Fig. 2 Activation functions in Perceptron model

또한, 딥러닝 알고리즘에서는 입력 값과 가중치에 의해 계산된 예측 값을 다음 층으로 바로 전달하지 않고 주로 비선형 함수를 통과시킨 후 전달하는데, 이때 사용하는 함수를 활성화 함수(activation function, Fig. 2)라고 한다. 초기 개발된 함수는 로그함수 형태의 Sigmoid 함수로 함수 값을 0~1로 제한하였다. 하지만 역전파 과정에서 그래디언트(gradient) 값이 소실되는 치명적인 단점을 보완하고자 ReLU 함수가 제안 되었고, 최근까지 ReLU 계열의 활성화 함수가 가장 많이 사용되고 있다.

2.1 딥러닝 프레임워크

본 연구에서는 파이썬(Python v3.6)으로 작성된 오픈 소스 신경망 라이브러리인 케라스(Keras v2.3.1)를 딥러닝 프레임워크로 사용하였다. 백엔드 엔진(backend engine)으로 GPU 기반의 텐서플로(Tensorflow_gpu v1.13.1)를 사용하였고, 해석 속도를 높이기 위해 GPU에서 학습을 진행시켜 NVIDIA CUDA 심층 신경망 라이브러리(cuDNN v10.0)라고 불리는 최적화된 딥러닝 연산 라이브러리를 이용하였다. 해석 장치는 NVIDIA Quadro P4000로 메모리 사양은 8GB(VRAM)이다.

2.2 합성곱 신경망

합성곱 연산을 이용하여 이미지의 공간적인 정보를 추출하는 합성곱 신경망은 완전 연결(fully connected) 신경망 대비 연산량을 획기적으로 줄이고 이미지 분류에 탁월한 성능을 보이는 것으로 알려져 있다 (LeCun et al., 1995). 합성곱 신경망은 합성곱 연산을 통해서 이미지의 특징을 추출하는 합성곱 층과 추출한 특성 맵(feature map)의 크기를 줄이는 풀링(pooling) 연산을 수행하는 풀링 층의 조합으로 구성된다. 여기서, 효과적인 합성곱 연산을 위해서는 이미지 특성을 추출하는 필터(filter)의 개수, 필터가 움직이는 이동 간격(stride), 데이터의 크기를 유지하기 위해 입력 데이터 주변에 특정 값을 채우는 패딩(padding) 옵션을 적절히 사용해야 한다. 이렇게 합성곱 층과 풀링 층을 거쳐 처리된 입력 데이터를 이용하여 최종적으로 완전 연결층에서는 가중치의 행렬 곱 연산과 이를 최소화하기 위한 학습 과정을 수행하여 연산 결과를 출력한다(Fig. 3).

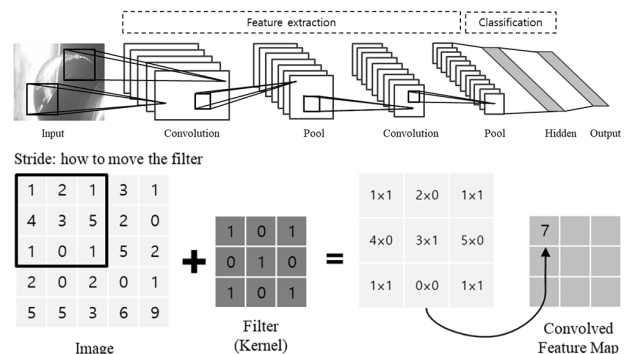


Fig. 3 CNN structure

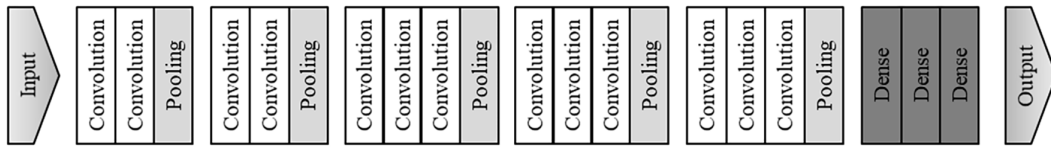


Fig. 4 VGG16 model structure

정리하면, 합성곱 신경망은 합성곱 층(convolution layer)과 풀링 층(pooling layer)을 반복적으로 사용하면서 이미지의 불변하는 특징을 찾고, 그 특징을 입력 값으로 완전 연결 신경망(fully connected neural network)으로 보내 이미지 추출(classification)을 수행한다.

3. 신경망 학습 과정 및 결과

3.1 데이터 전처리

일반적으로 학습 데이터의 개수가 충분하지 않은 상황에서 신경망의 학습을 지속할 경우 과대 적합, 즉 훈련시킨 데이터에 대한 정확도가 높아지는 반면 훈련 시 사용되지 않은 검증용 데이터에 대한 정확도는 떨어지는 현상이 발생할 수 있다. 때문에 학습시키는 이미지에 여러 종류의 랜덤 변환을 적용하여 학습 데이터를 증식시키는 방법을 사용하여 이와 같은 현상을 예방할 수 있다. 본 연구에서는 케라스에서 제공하는 ImageData Generator 클래스를 사용하여 설정한 범위 내에서 랜덤하게 증식된 이미지가 학습에 사용되도록 하였다.

3.2 네트워크 모델을 적용한 합성곱 신경망

딥러닝을 적용하는 일반적이고 매우 효과적인 방법은 잘 알려진 딥러닝 네트워크를 사용하는 것이다. 본 연구에서는 ILSVRC(ImageNet Large Scale Visual Recognition Challenge)에서 우수한 성능이 입증된 아래 3가지 네트워크 모델을 적용하여 예측 결과를 비교해보고자 한다.

3.2.1 VGG

2014년 옥스포드 VGG 그룹에서 개발된 딥러닝 모델로, 알렉스 넷(Alex et al., 2012)의 커널 크기를 11에서 3x3으로 줄였다. 모든 합성곱 층과 풀링 층에 다수의 3x3 필터를 적용하였으며, 2~4개의 연속되는 합성곱 층과 풀링 층으로 구성되었다. 5x5 이상 크기의 필터를 1개 사용하는 것보다 모델의 연산량이 감소하며 정확도가 상승함을 확인하였으며, 신경망 구조가 단순하고 모델의 수정 및 확장이 간편하다는 장점이 있다. 하지만 층이 깊어 역전파 시 그라디언트가 작아져 해가 수렴되지 않는 문제(gradient vanishing)가 발생할 가능성이 높다. 본 연구에서는 합성곱 층 13개, 완전 연결 층 3개로 이루어진 VGG16(Fig. 4)을 사용하였다.

3.2.2 GoogLeNet

해당 모델은 VGG 모델의 연산량 증가로 인한 처리 속도 저하 문제를 해결하기 위해 인셉션 모듈(inception module, Fig. 5)을 도입하였으며, 인셉션 모듈은 다양한 크기의 필터와 1x1 풀링 층을 여러 개 결합하여 하나의 블록으로 사용한다.

3.2.3 ResNet

기존의 신경망 학습의 목적이 손실 함수(H(x)-y)를 최소화하는 가중치 함수(H(x))를 찾는 것이라면, ResNet은 잔차(Residual, F(x)=H(x)-x)를 최소화하도록 학습한다. 이를 위해 ResNet은 2개의 합성곱 연산 과정을 거친 특성 맵에 합성곱 층을 거치지 않은 특성 맵을 결합한 블록(Fig. 6)을 반복적으로 쌓아서 구성되었다.

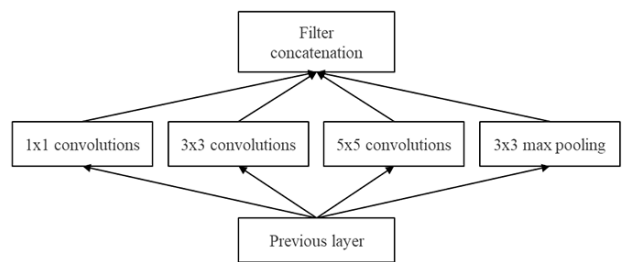


Fig. 5 GoogLeNet Inception model structure

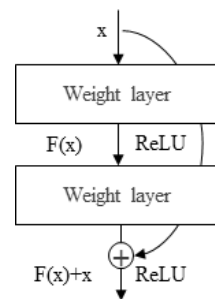


Fig. 6 ResNet building block structure

즉, 입력 데이터를 합성곱 층을 건너뛰어 출력에 바로 더하는 스킵 연결(skip connection, shortcut connection) 구조를 적용하여 역전파 과정에서의 신호 감쇠를 방지하여 학습 정확도를 올렸다.

3.3 신경망 학습 및 예측 결과

3.3.1 학습 데이터 구성 및 절차

학습을 위한 전체 데이터 구성은 Fig. 7과 같다. 먼저 전체 데이터셋(dataset)을 훈련(training), 검증(validation), 시험(test)용 데이터셋으로 나누고, 훈련용 데이터셋을 이용하여 모델을 학습시킨다. 이어 검증용 데이터셋을 이용하여 모델의 정확도(accuracy) 및 손실 함수 값(loss)을 계산한다. 이때, 검증용 데이터셋은 훈련용 데이터셋에 포함되지 않은 이미지를 사용하였다.

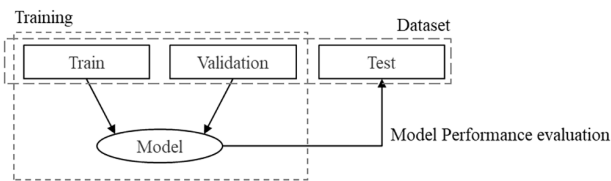


Fig. 7 Training and test dataset

시험용 데이터셋을 이용하여 최종적으로 모델의 정확도를 평가한다. 이때, 모든 데이터셋을 한 번 학습하는 단위를 1 에폭(epoch)이라고 한다. 해석 장비의 메모리의 한계와 해석 속도 저하로 인하여 한 번의 에폭에서 모든 데이터셋을 포함하기 어렵기 때문에, 전체 데이터 셋은 배치 크기(batch size)로 나누어 학습한다(minibatch). 프로펠러에서 발생하는 캐비테이션 침식 위험도 학습을 위해 캐비테이션의 패턴을 Fig. 8과 같이 구분하

여 직접 학습용 데이터셋을 구성하였다. 우선, 안정적으로 발생하는 얇은층캐비테이션의 경우 캐비테이션으로 인한 침식의 위험성이 낮다고 판단하였다(normal). 침식 위험도가 높은 캐비테이션 패턴(erosion)은 얇은층캐비테이션의 크기(volume)가 커짐에 따라 캐비테이션의 경계면에서 불규칙적으로 발생하는 캐비테이션(S.C., Separated Cavitation from sheet cavitation)과 얇은층 캐비테이션과 독립되어 프로펠러면에서 발생하는 캐비테이션(I.C., Isolated Cavitation)으로 분류하였다. 이 때, 데이터셋을 구성하는 각 이미지는 모형시험 시 촬영한 동영상에서 추출하였으며, 학습에 충분한 분량(이미지 개수: 훈련용 2,400개, 검증용 1,200개, 시험용 300개)의 다양한 크기 및 선종과 캐비테이션수 조건의 결과를 사용하였다. 또한, 앞서 3.1.절에서 언급하였듯이 학습 정확도를 높이기 위하여 회전, 이동, 크기 변환 등을 통하여 입력 데이터의 개수를 증폭시켰다.

3.3.2 학습 결과

네트워크 모델을 적용한 신경망 학습 결과의 성능 비교를 위해 4개의 합성곱과 풀링 층을 갖는 간단한 합성곱 신경망(simple CNN)을 Table 1과 같이 구성하였다. Table 1에서 확인할 수 있듯이 해당 모델은 합성곱과 풀링 층을 번갈아가며 4번 쌓아 이미지의 특성을 추출하였다. 이 과정에서 각 합성곱과 풀링 층이 출력하는 특성 맵을 시각화하여 학습된 필터들이 어떻게 입력을 분해하는지 확인하였다. Fig. 9는 각각 합성곱과 풀링 층을 지날 때 변환된 이미지를 보여준다. 각 층에서 추출한 특성은 층의 깊이가 깊어짐에 따라 입력된 이미지의 시각적 정보가 줄어들고

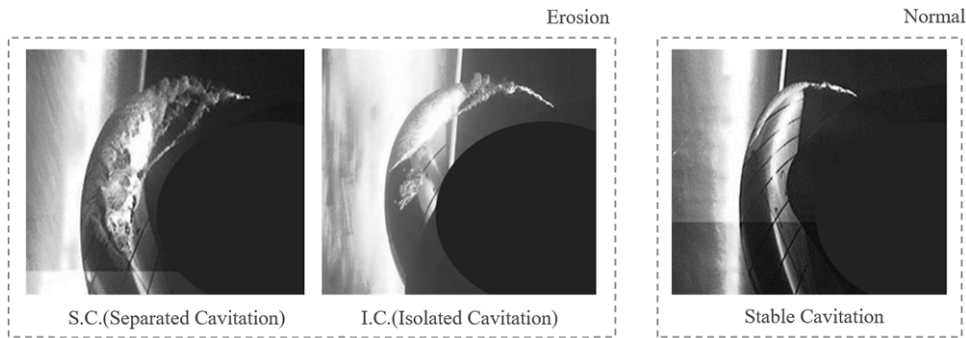


Fig. 8 Classification of cavitation pattern

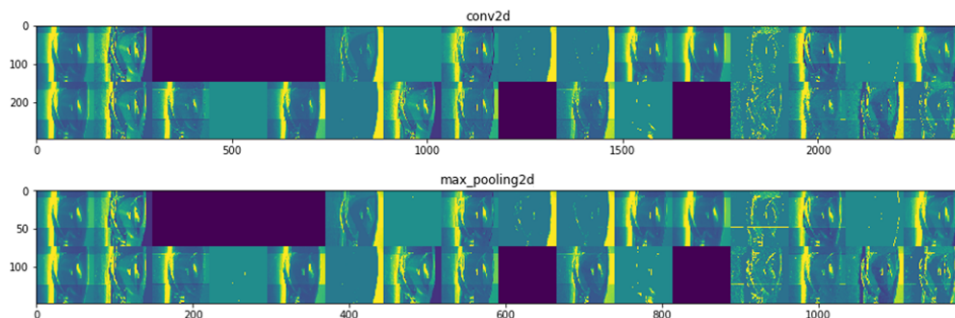


Fig. 9 Activation channels passing the first convolution and pooling layer

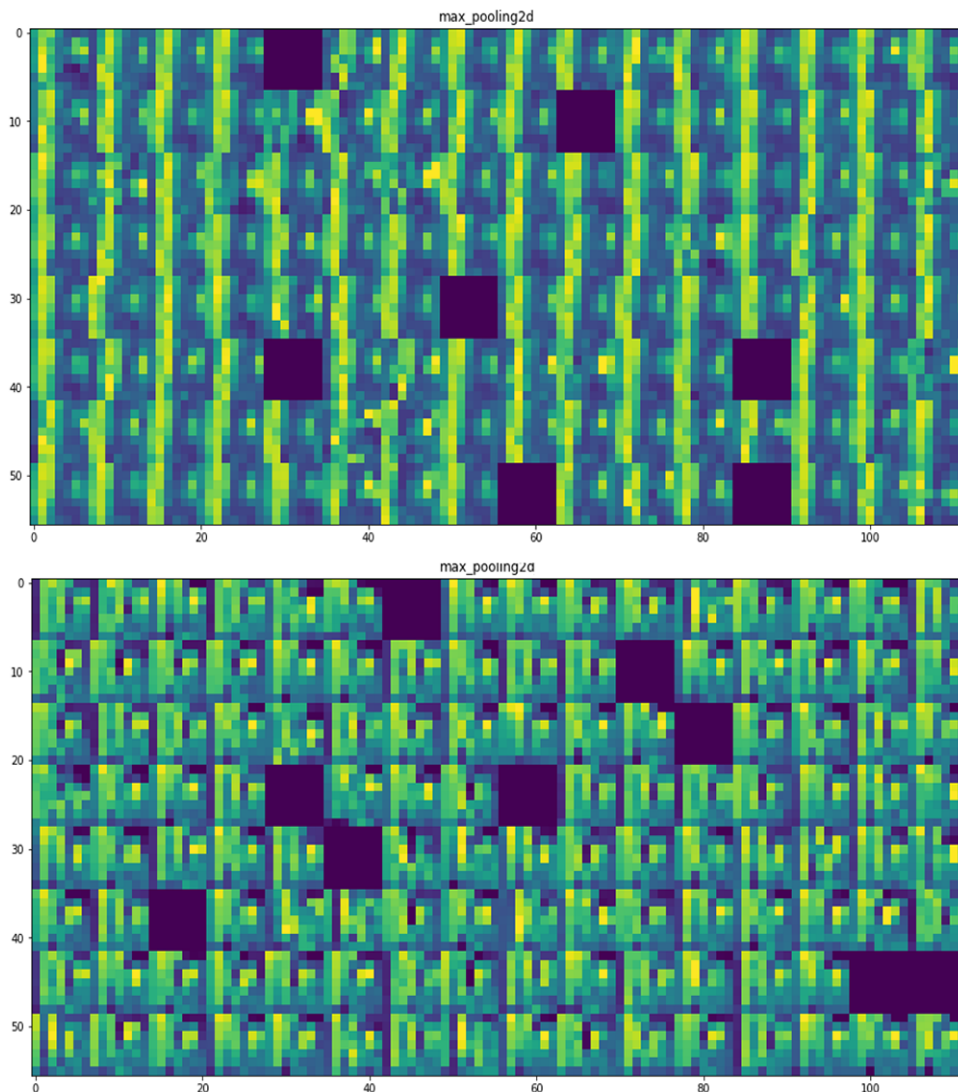


Fig. 10 Activation channels passing the last convolution and pooling layer(Top: Erosion, Bottom: Normal)

Table 1 Model summary of simple CNN

| Layer (type) | Output shape | Param # |
|---------------------|----------------------|---------|
| Conv2d_1 (Conv2D) | (None, 148, 148, 32) | 896 |
| Max_pooling2d_1 | (None, 74, 74, 32) | 0 |
| Conv2d_2 (Conv2D) | (None, 72, 72, 64) | 18496 |
| Max_pooling2d_2 | (None, 36, 36, 64) | 0 |
| Conv2d_3 (Conv2D) | (None, 34, 34, 128) | 73856 |
| Max_pooling2d_3 | (None, 17, 17, 128) | 0 |
| Conv2d_4 (Conv2D) | (None, 15, 15, 128) | 147584 |
| Max_pooling2d_4 | (None, 7, 7, 128) | 0 |
| flatten_1 (Flatten) | (None, 6272) | 0 |
| dense_1 (Dense) | (None, 128512) | 802944 |
| dense_2 (Dense) | (None, 1) | 387 |

Total params: 1,044,163
 Trainable params: 1,044,163
 Non-trainable params: 0

특성을 강조한 추상적 데이터로 변환된다(Fig. 10). Table 1에 output shape은 적용된 필터의 크기와 갯수를 나타내며, 가중치 파라미터의 개수는 층을 거듭하며 필터의 크기와 개수에 따라 증가한다. 최종적으로 분류단계에 거쳐 1개의 예측 값을 출력하게 된다. 해당 모델을 이용해 총 50 에폭에 걸쳐 학습되는 모델의 학습 및 검증 정확도와 손실 함수 값을 Fig. 11에 나타내었다. 학습용과 검증용 데이터셋의 모델에 대한 정확도는 1로 수렴되어 가는 것을 확인할 수 있으며, 학습용 데이터셋의 손실 함수 또한 0 에 가깝게 수렴되었다. 하지만 검증용 데이터셋의 손실함수는 수렴성을 가지지 못함을 확인할 수 있는데 이처럼 학습용 데이터셋에 한정하여 높은 정확도를 갖는 현상을 과대 적합(overfitting)이라고 정의한다. 과대 적합은 모델이 훈련용 데이터셋에 대해서는 필요 이상으로 높은 정확도를 보이면서 훈련용 데이터셋 이외의 다양한 변수에 대한 예측도는 낮은 현상을 말한다. 이를 해결하는 방법으로는 더 많은 훈련용 데이터셋을 확보하거나, 정규화(regulation), 제약 조건, 드롭아웃(drop-out) 등의 방법을 적용하여 원하는 정도의 복잡도를 갖는 모델

을 찾아야 한다. Table 2는 시험용 데이터셋에 대한 해당 모델의 정확도를 보여준다. 각 캐비테이션 패턴에 해당하는 학습에 사용되지 않은 이미지를 모델에 입력 값으로 넣어주었을 때, 모델이 이를 잘 구별할수록 정확도가 1에 가까운 값을 갖는다. 해당 모델의 경우 S.C.(Separated Cavitation)을 제외한 다른 캐비테이션 패턴의 정확도가 떨어짐을 확인할 수 있다($10e-5$ 이하의 정확도는 0으로 표기함). 본 연구에서는 훈련용 데이터셋이 제한되어 있기 때문에, 기존 연구를 통해 잘 구성된 네트워크 모델을 적용하여 각 모델의 특성을 파악하고 충분한 캐비테이션 침식 예측 정확도를 확보하고자 하였다.

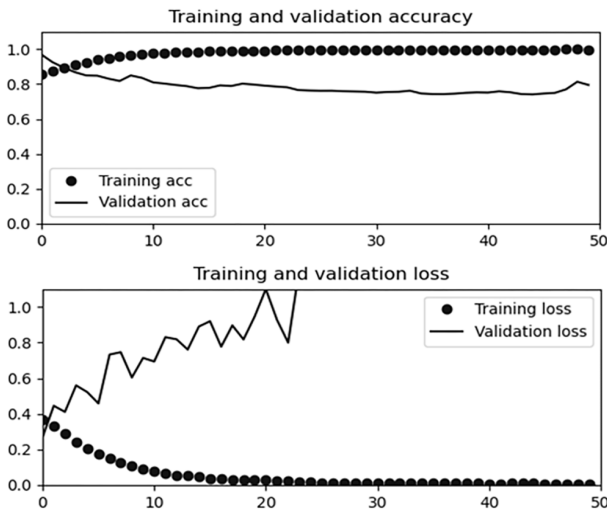


Fig. 11 Training and validation accuracy and loss using simple CNN

Table 2 Accuracy of test dataset using simple CNN

| | S.C. | I.C. | Normal |
|--------|-------|-------|--------|
| S.C | 1.000 | 0.000 | 0.000 |
| I.C. | 0.000 | 0.597 | 0.403 |
| Normal | 0.141 | 0.000 | 0.859 |

Table 3 Model summary using VGG16

| Layer (type) | Output shape | Param # |
|------------------------------|-------------------|------------|
| vgg16 (Model) | (None, 4, 4, 512) | 14,714,688 |
| flatten_1 (Flatten) | (None, 8,192) | 0 |
| dense_1 (Dense) | (None, 256) | 2,097,408 |
| dense_2 (Dense) | (None, 3) | 771 |
| Total params: 16,812,867 | | |
| Trainable params: 16,812,867 | | |
| Non-trainable params: 0 | | |

첫번째로 적용한 모델은 VGG 모델(Table 3)로, 앞서 4개의 합성곱과 풀링층을 갖는 간단한 합성곱 신경망(simple CNN)과 비교하였을 때, 층의 깊이가 4배 이상 깊어지며 전체 파라미터

수가 10배 이상 증가하였다. 이에 따라 훈련용 데이터셋 뿐만 아니라 검증용 데이터셋에 대한 정확도 및 손실함수 값이 충분히 수렴함을 확인할 수 있다(Fig. 12). 또한, 프로펠러에 침식을 유발할 가능성이 높은 캐비테이션 패턴(S.C., I.C.)에 대한 예측 정확도를 100% 확보하였으나, 침식 가능성이 낮은(normal) 이미지에 대한 예측 정확도는 86.8%에 그쳤다(Table 4). 학습 속도 및 예측 정확도 개선을 위해 두번째로 적용한 모델은 GoogLeNet(Inception V3) 모델로(Table 5) VGG 모델과 비교하여 층의 깊이가 대략 4배 정도 깊어진 반면, 병렬 구조의 인셉션 모듈을 적용하여 전체 파라미터 개수는 약 1.5 배 증가하였다.

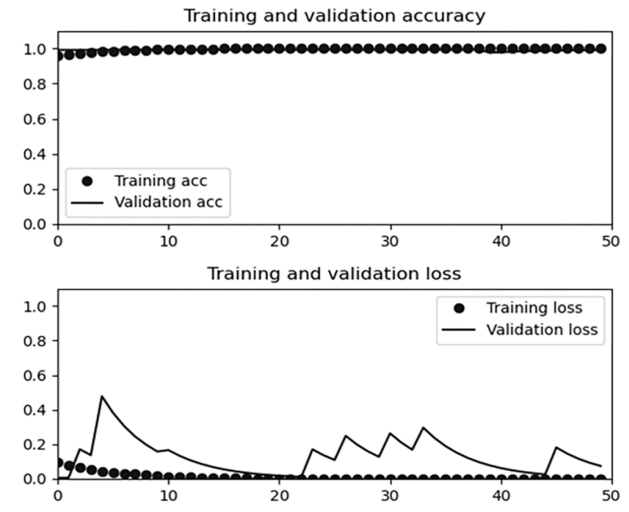


Fig. 12 Training and validation accuracy and loss using VGG16

Table 4 Accuracy of test dataset using VGG16

| | S.C. | I.C. | Normal |
|--------|-------|-------|--------|
| S.C | 1.000 | 0.000 | 0.000 |
| I.C. | 0.000 | 1.000 | 0.000 |
| Normal | 0.132 | 0.000 | 0.868 |

Table 5 Model summary using GoogLeNet

| Layer (type) | Output shape | Param # |
|------------------------------|---------------------|------------|
| Inception_v3 (Model) | (None, 3, 3, 2,048) | 21,802,784 |
| flatten_1 (Flatten) | (None, 18,432) | 0 |
| dense_1 (Dense) | (None, 256) | 4,718,848 |
| dense_2 (Dense) | (None, 3) | 771 |
| Total params: 26,522,403 | | |
| Trainable params: 26,487,971 | | |
| Non-trainable params: 0 | | |

해당 모델의 경우 50 에폭에 걸쳐 충분한 학습 정확도와 손실함수 값을 갖는 것을 확인할 수 있으며(Fig. 13), 침식 위험도가 낮은 캐비테이션 패턴의 예측 정확도(Table 6)가 92.4%

로 VGG 모델에 비해 정확도가 대략 5% 증가함을 확인하였다. 마지막으로 ResNet 모델은(Table 7) GoogLeNet 모델과 비교하여 층의 깊이가 대략 3배 정도 깊어진 반면, 스킵 연결 방식을 적용하여 전체 파라미터 개수는 대략 1.5배 증가하였다. 또한, 스킵 연결 방식을 적용함으로써 층이 깊어질수록 학습이 어렵고 성능이 저하되던 기존 모델들의 문제점을 해결하였다. 해당 모델을 적용하여 학습한 결과 모든 패턴에 대해 92% 이상의 충분한 학습 정확도와 손실함수 값을 확보하였으며(Fig. 14), 캐비테이션 침식 위험 예측 정확도는 앞선 GoogLeNet 모델의 결과와 유사함을 확인하였다(Table 8).

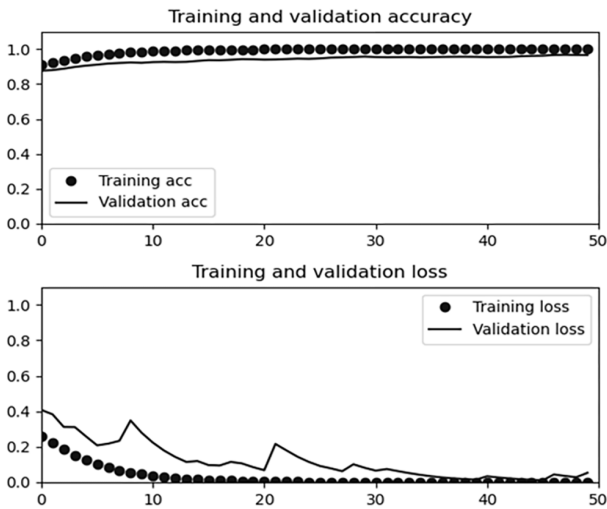


Fig. 13 Training and validation accuracy and loss using Inception V3

Table 6 Accuracy of test dataset using Inception V3

| | S.C. | I.C. | Normal |
|--------|-------|-------|--------|
| S.C | 1.000 | 0.000 | 0.000 |
| I.C. | 0.000 | 1.000 | 0.000 |
| Normal | 0.076 | 0.000 | 0.924 |

Table 7 Model summary using ResNet

| Layer (type) | Output Shape | Param # |
|----------------------|---------------------|------------|
| Inception_v3 (Model) | (None, 3, 3, 2,048) | 21,802,784 |
| flatten_1 (Flatten) | (None, 18,432) | 0 |
| dense_1 (Dense) | (None, 256) | 4,718,848 |
| dense_2 (Dense) | (None, 3) | 771 |

Total params: 26,522,403
 Trainable params: 26,487,971
 Non-trainable params: 0

최종적으로 각 모델에 따라 학습을 위해 소요된 시간을 Table 9에 비교하여 나타내었다. 파라미터 개수 대비 GoogLeNet과 ResNet 모델을 적용한 결과가 높은 예측 정확도와 속도를 보였

으며, VGG 모델의 경우 앞선 두 모델에 비해 예측 정확도가 낮고 학습에 필요한 시간이 더 요구됨을 확인하였다.

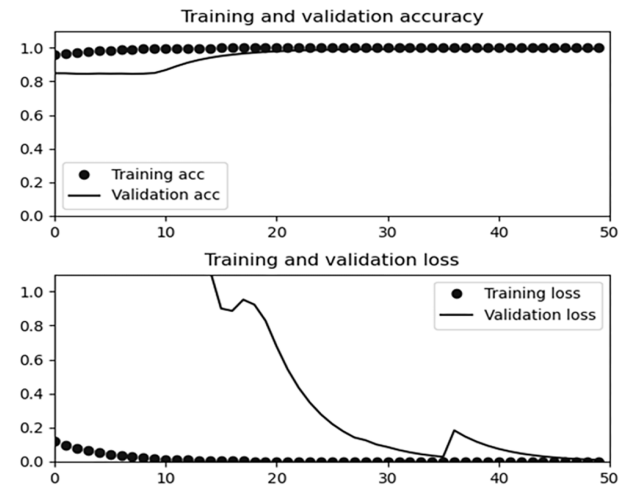


Fig. 14 Training and validation accuracy and loss using ResNet

Table 8 Accuracy of test dataset using ResNet

| | S.C. | I.C. | Normal |
|--------|-------|-------|--------|
| S.C | 1.000 | 0.000 | 0.000 |
| I.C. | 0.000 | 1.000 | 0.000 |
| Normal | 0.076 | 0.000 | 0.924 |

Table 9 Training time according to network models

| Model | Training time | No. of parameters |
|------------|-----------------------|-------------------|
| Simple CNN | 1,133.205s (18.9 min) | 1,044,163 |
| VGG | 1,264.249s (21.1 min) | 16,812,867 |
| GoogLeNet | 1,293.572s (21.6 min) | 26,522,408 |
| ResNet | 1,307.147s (21.8 min) | 36,695,939 |

4. 결론

본 연구에서는 프로펠러에서 발생하는 캐비테이션 침식 위험도를 정량적으로 지표화 할 수 있는 방법의 일환으로 합성곱 신경망을 기반으로 딥러닝 알고리즘을 구현하였다. 단순한 합성곱 신경망 구조를 구축하고 기존 연구를 통해 잘 구성된 네트워크 모델을 적용한 학습 결과와 비교하였다. 이를 통해 각 모델의 특성을 파악하고 충분한 캐비테이션 침식 예측 정확도를 확보하고자 하였다. 잘 알려진 바와 같이 학습을 위한 층의 깊이가 깊어질수록 학습의 정확도가 높아짐을 확인하였으며, 층이 깊어질수록 학습이 어렵고 성능이 저하되는 현상은 필터의 개수 조절, 층의 구성 형식의 변경, 가중치를 전달하는 방식의 변환 등으로 해결할 수 있음을 확인하였다. 또한, 기 개발된 네트워크 모델이 침식 위험도가 높은 캐비테이션을 충분한 정도로 구분

해 낼 수 있음을 확인하였다. 해당 기법을 통해 모형시험 시 육안으로 놓칠 수 있는 부분을 포함하여 실시간으로 이미지를 처리하고, 모델에 지속적으로 데이터를 학습시킴으로써 프로펠러에서 발생하는 캐비테이션의 거동 특성 및 침식 가능성에 대한 수치화된 자료를 제시할 수 있을 것으로 기대된다. 또한 캐비테이션 침식 위험도에 대한 threshold 값을 얼마로 할 것인가에 대한 사용자별 기준을 결정하는 과정이 필요하겠지만, 결론적으로 현 단계에서 기존에 개발된 모델을 적용한 딥러닝 알고리즘의 예측 수준이 침식 위험도를 정량적으로 평가하기 위한 초기 틀로서 충분한 예측 정확도를 보이고 있는 것으로 판단된다. 추후 본 연구 방식을 적용하여 캐비테이션 발생 시 계측된 변동압력 데이터를 바탕으로 침식 위험도가 높은 캐비테이션의 분류를 시도하고자 하며, 이를 통해 캐비테이션 발생 패턴에 따른 침식 위험도를 시각적(visual), 그리고 음향학적(acoustic) 데이터를 기반으로 지표화 할 수 있을 것으로 기대한다. 최종적으로 한 가지 패턴의 인식뿐만 아니라 동시다발적으로 발생하는 캐비테이션 패턴을 구분할 수 있도록 예측 기법을 확장하고자 한다.

References

Alex, K., Ilya, S. & Geoffrey, E. H., 2012. ImageNet classification with deep convolutional neural networks, *Advances in Neural Information Processing System*, pp.1097–1105.

Chao, Q., Tao, J., Wei, X., Wang, Y., Meng, L. & Liu, C., 2020. Cavitation Intensity recognition for high-speed axial piston pumps using 1-D convolutional neural networks with multi-channel inputs of vibration signals. *Alexandria Engineering Journal*, pp. 4463–4473.

Chollet, F., 2017. *Deep learning with python translated by park, H.* S. Manning Publications: New York.

He, K., Zhang, X., Ren, S. & Sun, J., 2016. Deep residual learning for image recognition, *IEEE conference on computer vision and pattern recognition*, Las Vegas, USA, 27–30 2016.

LeCun, Y. et al., 1995. Learning algorithms for classification: A comparison on handwritten digit recognition. *Neural networks: the statistical mechanics perspective*, pp. 261–276.

Simonyan, K. & Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 1556, pp.1–14.

Shora, M., Ghassemi, H. & Nowruzi, H., 2017. Using computational fluid dynamic and artificial neural networks to predict the performance and cavitation volume of a propeller under different geometrical and physical characteristics, *Journal of Marine Engineering & Technology*, 17(2), pp. 59–84.

Szegedy, C. et al., 2015. Going deeper with convolutions. *IEEE conference on computer vision and pattern recognition*, Boston, USA, 7–12 June 2015.



김 지 혜

이 형 석

허 재 욱