대한조선학회논문집
Journal of the Society of Naval Architects of Korea
eISSN:2287-7355, Vol. 59, No. 2, pp. 125-133, April 2022
https://doi.org/10.3744/SNAK.2022.59.2.125

Check for updates

합성곱신경망을 활용한 과구동기 시스템을 가지는 소형 무인선의 추진기 고장 감지

백승대[1]·우주현[2,†]
한국해양대학교 조선해양시스템공학과[1]
한국해양대학교 조선 · 해양개발공학부[2]

Fault Detection of Propeller of an Overactuated Unmanned Surface Vehicle based on Convolutional Neural Network

Seung-dae Baek[1]·Joo-hyun Woo[2,†]
Department of Naval Architecture and Ocean System Engineering, Korea Maritime and Ocean University[1]
Major of Naval Architecture and Ocean System Engineering, Korea Maritime and Ocean University, [2]

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.

This paper proposes a fault detection method for a Unmanned Surface Vehicle (USV) with overactuated system. Current status information for fault detection is expressed as a scalogram image. The scalogram image is obtained by wavelet-transforming the USV's control input and sensor information. The fault detection scheme is based on Convolutional Neural Network (CNN) algorithm. The previously generated scalogram data was transferred learning to GoogLeNet algorithm. The data are generated as scalogram images in real time, and fault is detected through a learning model. The result of fault detection is very robust and highly accurate.

Keywords : Unmanned surface vehicle(무인 수상 선박), Fault detection(고장 감지), Convolutional neural network(합성곱신경망), ROS, Overacturated(과구동기), Wavelet transform(웨이블렛 변환)

1. 서 론

선박이 자율화되고 그에 따른 연구와 실험의 목적으로 많은 무인선이 개발되어지고 있다. 개발되어진 다수의 무인선의 추진 시스템을 비교해보면, 무인선은 조종성과 조작성 향상을 위해 하나의 추진기가 아닌 다수의 추진기를 사용하거나 Azimuth 추진기를 사용하는 것이 일반적이다. Fig. 1은 국내외에서 개발된 2개 이상의 추진기를 사용하는 소형무인선의 예시를 나타내고 있다.

소형무인선은 일반적으로 연안이나 내수면에서 운용되는데, 이런 운용환경에서는 어망이나 로프 등의 해양부유물이 존재할 수 있다. Fig. 2-(Left)는 연안환경에서 발견될 수 있는 다양한 해양부유물들을 나타내고 있다. 이러한 해양부유물들은 소형무인선의 추진기에 유입되거나 얽혀 추진시스템의 고장을 야기시킬 수 있다. Fig. 2-(Right)는 실제 실험 중 추진기에 이물질이 걸린 상황을 나타내고 있다.

무인시스템의 추력을 발생시키는 프로펠러가 고장 나게 되면, 경로계획기술이나 충돌회피기술과 같은 유도명령이 생성되어도 실제로 무인시스템을 제어할 수 있는 제어력을 발생시키지 못하기 때문에 제어불능이나 충돌 등의 위험 상황이 발생할 수 있다. 부유물에 의한 추진기의 고장상황은 프로펠러 이외의 시스템에 추가적인 고장을 발생시키기도 한다. 예를 들어, 전기추진 무인선의 경우 전기모터구동을 위한 모터드라이버 등의 속도제어시스템(Electronic Speed Controller, ESC)을 포함하는

Fig. 1 (Left) ME120 USV of OceanAlpha (Right) Heron USBV of Clearpath Robotics

Received : 3 January 2022 | Revised : 28 January 2022 | Accepted : 21 February 2022
† Corresponding author : Joohyun Woo, jhwoo@kmou.ac.kr

Fig. 2 (Left) Marine debris, (Right) Propeller failure due to the Marine debris

것이 일반적인데, 부유물이 추진기에 얽히는 등의 고장상황에 노출될 경우, ESC에 과도한 전기부하가 가해질 수 있으며, 이 때문에 과열 등의 추가적인 고장이 발생할 수 있다. 무인선의 운용 관점에서도 추진기의 고장이 발생할 시 즉시 임무 수행을 중단하고 정비를 해야 하므로, 추진기의 고장은 임무수행의 편의성과 효율성을 저해시키는 요인이 될 수 있다.

하지만, 앞서 언급하였듯 소형무인선의 경우 일반적으로 다수의 추진기를 사용하므로 하나의 추진기를 사용하는 경우에 비해 추진기의 고장확률이 높아진다. 더불어, 추진기의 고장진단은 조류, 파랑, 풍하중 등의 환경외란이 섞여 있는 무인선의 운동응답을 바탕으로 고장여부를 판단해야 하므로 관측성이 낮고 외란에 민감하여 운용자가 직관적으로 판단하는데 어려움이 따른다. 본 연구에서는 추진기가 고장환경에 쉽게 노출되고 고장 유무의 관측성이 낮으며 운용자의 개입으로 판단하여야 하는 한계를 극복하기 위한 방안으로 다수의 추진기를 갖는 소형 무인선의 추진기 고장진단 기법을 제안한다. 제안하는 기법은 합성곱신경망(Convolutional Neural Network, CNN)을 바탕으로 무인선 스스로 운동응답의 경향을 바탕으로 추진기의 이상여부를 판단하는 기법이다. 제안 기법은 다양한 운항 상황을 포함하는 시뮬레이션 결과를 바탕으로 학습되었으며, 학습된 모델을 바탕으로 추진기 고장여부를 실시간으로 판단할 수 있는 기법이다.

고장 진단 및 검출 문제는 보통 FDD(Fault Detection and Diagnosis)라고 표현하는 데 FDD는 산업분야에서 다양하게 연구되어지고 있다. Cho et al. (2021)의 연구에서는 Pitch sensor를 토대로 Kalman filters를 활용하여 풍력발전기의 결함을 감지하고 ANN 학습 모델을 통해 고장을 예측하였다. Taheri et al. (2021)의 연구 에서는 복잡하고 불확실한 시계열 정보의 특성을 학습할 수 있는 DRNN (Deep Recurrent Neural Network)을 기반으로 하여 HVAC(Heating, Ventilation & Air Conditioning) 공기조화장치의 고장을 진단하는 기법과 그에 최적화된 Hyperparameters를 구성하는 방법에 대해 제안 하였다. 로보틱스 분야에도 고장진단에 대한 연구는 활발히 이루어지고 있다. Amoozgar et al. (2013)의 연구에서는 TSKF(Two-Stage Kalman filters)를 활용하여 로터의 제어 효율성 손실로 모델링된 결함 모델을 토대로 각 4개의 구동기의 결함을 추정하였고 테스트 베드 적용을 통해 실험적 평가도 진행하였다. Zhang et al. (2015)의 연구에서 고장상황에 대해 Grey Relational Analysis를 통해 부족한 데이터를 확보하여 시스템 식별을 통해 AUV(Autonomous Underwater Vehicle)의 추진기 고장을 식별하였다.

본 논문은 다음과 같이 구성되어 있다. 2장에서는 선박의 정상상황과 고장상황에서의 제어 입력값과 선박의 상태정보를 토대로 스캘로그램(Scalogram)을 생성하고 생성된 이미지를 학습시키기 위한 제안기법에 대한 내용이 설명되어 있다. 3장에서는 제안 기법을 적용시키기 위한 시뮬레이션 환경과 그 환경 내에서 데이터 획득 과정을 나타내고 전이학습을 통한 학습 모델 생성을 하며 앞선 제안 기법들의 적용을 서술한다. 4장에서는 새로운 데이터를 토대로 제안 기법의 성능 검증을 진행하고 5장에서는 결론 및 고찰에 대한 내용을 다루고 있다.

2. CNN 기반 무인선 추진기 고장진단

본 연구가 진행되는 워크플로우(Work flow)는 Fig. 3과 같다. 우선 무인선의 운항 시뮬레이션을 통해 고장상황과 정상상황에 대한 운항데이터를 획득한다. 무인선의 시뮬레이터는 ROS (Robot Operating System)의 Gazebo 환경에서 구현된 무인선 시뮬레이터인 VRX 시뮬레이터를 사용하였다. 시뮬레이션에서 획득하는 정보는 제어기에서 선박에 전달되는 제어입력 정보와 무인선의 상태정보의 시계열 데이터로 구성된다. 이렇게 얻어진 시계열 데이터는 웨이블렛 변환(wavelet transform)을 통해 스캘로그램 이미지로 변환되는데, 이 과정에서 각 데이터의 시계열 거동패턴이 시각화되어 합성곱신경망의 입력으로 사용될 수 있는 이미지 형태로 변환된다. 이후에는 전처리로 생성된 이미지를 CNN계열의 이미지 분류 네트워크인 GoogLeNet에 전이학습(transfer learning)시켜 학습 모델을 생성한다. 고장진단을 위한 모델이 학습된 이후에는 무인선의 운항 상태 및 제어입력 데이터를 스캘로그램으로 변환 후 학습모델에 입력하여 추진기의 고장여부를 실시간으로 판단한다. 각 단계에 대한 상세한 설명은 다음과 같다.

먼저, 데이터에 대한 획득 및 전처리(pre-processing)하는 작업은 다음과 같다. 본 연구에서는 총 3가지 종류의 시계열 데이터를 이용하여 무인선 추진기의 고장여부를 파악하는데, 이는

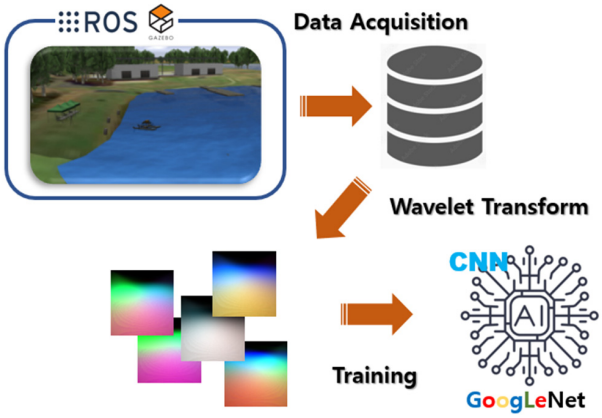

Fig. 3 Work flow of the proposed method in this paper

시스템의 제어입력으로 사용되는 1개의 시계열 정보와, 운동응답으로 정의되는 2개의 시계열 정보로 구성된다. 본 연구에서는 Fig. 8과 같은 4개의 추진기가 고정된 홀로노믹(Holonomic)한 운동을 할 수 있는 무인선을 대상선박으로 선정하였는데, 해당 선박이 입력받을 수 있는 전후동요(Surge), 좌우동요(Sway), 선수동요(Yaw) 자유도에 대한 제어명령(식 8 참조) 중 선수동요에 대한 제어입력 M_z만을 고장진단을 위한 시계열 데이터로 활용하였다. 본 연구에서는 전후동요에 대한 제어입력 F_x는 고정되어있고, 좌우동요에 대한 제어입력 F_y는 0이라고 가정하였다. 제어입력과 더불어, 무인선의 상태정보로는 선박의 전속도 V와 선수동요 각속도(Yawrate) r정보를 활용하였다. 이렇게 선수동요 제어 입력 M_z, 전속도 V, 선수동요 각속도 r로 구성되는 3개의 시계열 데이터를 고장 판단의 기준으로 사용하는 이유는 추진기 고장 시 선박이 제어입력을 주었을 때 기대되는 운동응답을 잘 추종하는지 여부를 통해 추진기의 고장여부를 유추할 수 있기 때문이다.

시계열 정보획득 이후에는 웨이블렛 변환과정을 통해 스캘로그램 이미지를 만들어 내는 과정을 거친다. 스캘로그램은 스펙트로그램(spectrogram)의 일종으로, 소리나 파동을 시각화 하여 파악하기 위한 도구로 활용되며, 파형(waveform)과 스펙트럼(spectrum)의 특징이 조합되어 있다. 스캘로그램의 파형은 시간축의 변화에 따른 진폭 축의 변화를 볼 수 있고 스펙트로그램의 파형은 주파수 축의 변화에 따른 진폭축의 변화를 볼 수 있다. 스펙트로그램은 시간 축과 주파수 축의 변화에 따라 진폭의 차이를 이미지의 밝기차이로 표현한다. 스펙트로그램은 대역필터(band-pass filters), 푸리에변환(Fourier transform), 웨이블렛 변환 등의 과정을 통해 생성되어질 수 있는데, 이때 웨이블렛 변환을 이용하여 변환되어지는 것을 스캘로그램이라 한다. 신호처리에 일반적으로 사용되는 푸리에변환 같은 경우는 신호가 시간적으로 변하지 않는다는 가정에서 주파수 성분을 표시한다. 그 이유는 푸리에변환은 시간에 국한되지 않는 사인(sine)파의 합으로 데이터를 나타내기 때문이다. 따라서 갑작스러운 변화를 효율적으로 나타내지 못한다. 이에 반해 웨이블렛 변환은 신호가 시간적으로 주파수성분이 변하는 신호에 대하여 시간과 주파수성분을 표현할 수 있다. 무한대로 확장되는 사인파와는 대조적으로 웨이블렛(Wavelet)은 평균이 0이고 빠르게 크기가 줄어드는 파동이다. 웨이블렛은 굉장히 다양하게 존재한다. 웨이블렛 변환에서 가장 중요한 개념으로 스케일링(scaling), 이동(shifting) 이 있다. 스케일링은 표현할 수 있는 시간에 따라 신호를 늘리거나 줄이는 과정을 말한다. 식 (1)에서 s는 스케일링 계수이며 t는 시간을 나타내며, 식 (2)는 수학적으로 등가 주파수를 의미한다. 이때, F_{eq}는 웨이블렛의 중심주파수이고 δ_t는 샘플링시간을 나타낸다. 따라서 웨이블렛 변환의 식은 스케일이 증가하면 주파수가 감소하고, 스케일이 감소하면 주파수가 증가하는 반비례의 관계를 가지고 있다. 이동은

$$\Psi(\frac{t}{s}) \tag{1}$$

$$F_{eq} = \frac{C_f}{s\delta_t} \tag{2}$$

시간 축을 따라 웨이블렛을 움직여주는 과정을 말한다. 이러한 개념들을 활용한 웨이블렛 변환의 절차를 정리하면 우선 작은 스케일의 웨이블렛을 신호의 시작부터 끝까지 이동시키면서 비교한다. 신호와 웨이블렛의 유사도를 C_f라는 계수에 담고 신호의 끝까지 비교한 후, 다시 스케일을 확장시켜 이 작업을 반복한다. 사전에 설정한 스케일까지 이 과정을 반복한다. 스캘로그램은 시간적으로 변화하는 신호를 효율적으로 표현할 수 있으므로, 순간적으로 발생하는 추진기의 고장신호 감지에 적합한 기법이다.

본 연구에서는 추진기 고장신호의 감지에 활용되는 선수동요 제어 입력 M_z, 전속도 V, 선수동요 각속도 r 시계열 데이터 각각을 스캘로그램으로 변환 한 후, 그 결과를 이미지의 R, G, B 각 채널의 밝기이미지(intensity image)로 표현하는 과정(Jayalakshmy & Sudha, 2020)을 통해 전처리를 수행하였다. Fig. 4는 고장감지를 위해 사용되는 시계열데이터를 웨이블렛 변환을 통해 이미지로 변환시키는 과정을 나타내고 있다.

이렇게 전처리가 완료된 데이터는 CNN의 학습데이터로 활용된다. 학습은 기존의 알고리즘에 전이 학습을 통해 이루어지는데, 전이 학습이란 특정 분야에서 학습된 신경망의 일부 능력을 유사하거나 전혀 새로운 분야에서 사용되는 신경망 학습에 이용하는 과정이다. 전이학습은 보통 새로운 모델을 만드는 것보다 적은 양의 학습 데이터를 사용하며 학습 속도도 빠르고 우수한 성능을 발휘한다. 고장 진단의 경우에도 일반적인 이미지를 분류하는 문제와 매우 유사하기 때문에 기존의 학습모델을 이용하는 전이 학습을 통해 학습을 효과적으로 진행하였다.

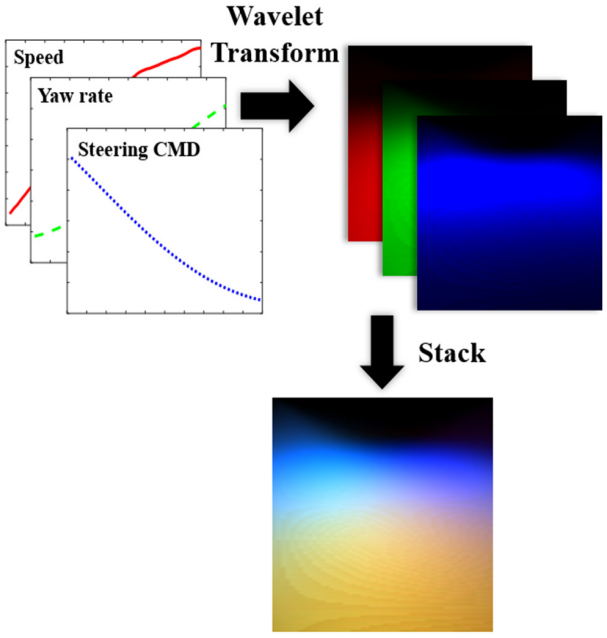

Fig. 4 Generation of the training image using wavelet transform

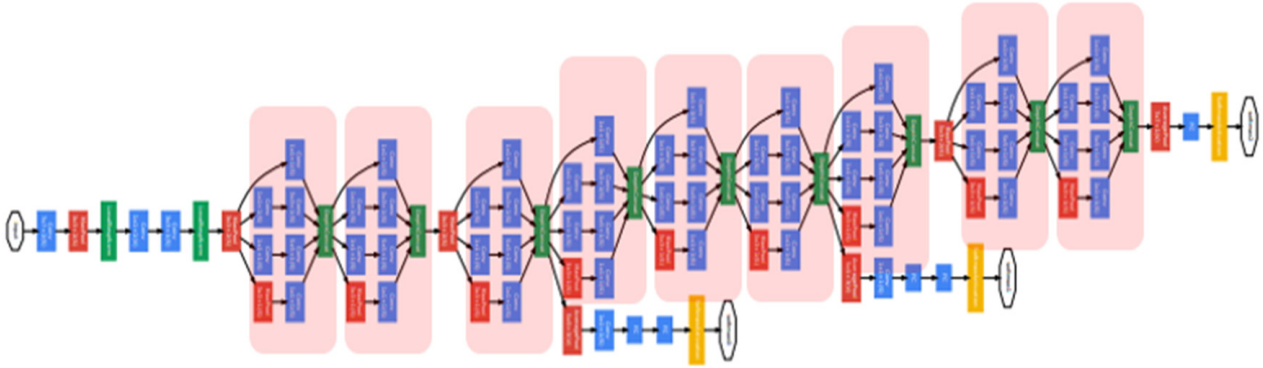

Fig. 5 GoogLeNet architecture proposed by (Szegedy et al., 2015)

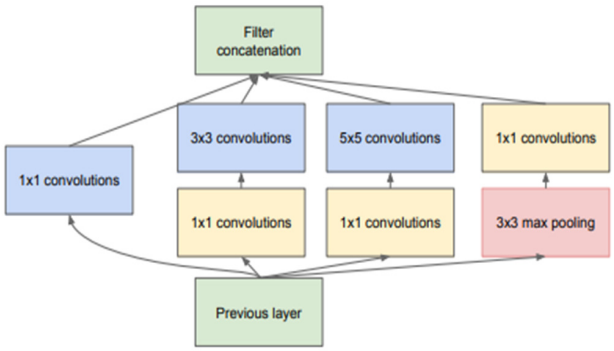

Fig. 6 Inception module with dimensionally reduction in (Szegedy et al., 2015)

지금 널리 알려진 CNN 기술 중 이미지 분류 알고리즘은 여러 종류가 있다. (Kim, 2018)의 연구를 참조해 보았을 때 대표적으로 깊은 합성곱신경망을 처음 사용하여 이미지 분류문제의 성능을 향상시킨 AlexNet이 있고, AlexNet보다 깊은 구조를 가지며 ReLU를 사용한 VGGNet, 보다 깊은 레이어(layer)를 쌓아 오류율을 획기적으로 낮춘 ResNet 등 이외 DenseNet, MobileNet등 각각의 특성과 성격이 다른 네트워크들이 존재한다. 이중에서 본 논문은 GoogLeNet이라는 CNN 네트워크를 활용한다. GoogLeNet은 2014년 ILSVRC(ImageNet Large Scale Visual Recognition Challenge)에서 1등을 차지한 모델로 Inception이라고 더 많이 알려진 네트워크이다. 일반적으로 딥러닝(deep learning) 네트워크의 성능은 구조가 깊어질수록 레이어가 넓을수록 성능이 좋아지는 것으로 알려져 있다. 하지만 실제로 학습을 할 때는 파라미터가 많아지면 과적합(over-fitting)이나 기울기 소실(Gradient vanishing)과 같은 문제가 생겨 학습이 어렵게 된다. 이때 GoogLeNet의 핵심 아이디어는 제한된 계산 자원을 이용해 최적의 성능을 내게 네트워크를 구성하려 한 것이다. 보통의 다른 모델은 하나의 합성곱 레이어(convolution layer)를 사용하는데 GoogLeNet은 작은 합성곱 레이어를 여러 개 사용하여 네트워크를 만들어 연산하고 결과값을 합치는 방식으로 구성하였다. 이러한 모듈을 Inception 모듈이라 한다. 전체 신경망의 구조는 Fig. 5 와 같이 도식화 할 수 있다. (Szegedy et al., 2015) 이때 빨간색 네모와 같은 마치 작은 신경망을 연상하는 모듈들이 연결되어 있는데 이것을 전체 신경망 안에 다른 신경망이 있는 것 같다 하여 Inception이라는 이름이 붙여졌다. 이 빨간색 상자를 확대해서 보면 Fig. 6과 같다. 일반적인 CNN의 구조는 이전 층의 출력을 입력으로 받아 합성곱(Convolution)과 풀링(pooling)을 진행한다. 이때 Inception 모듈에서는 (1,1), (3,3), (5,5)의 크기가 다른 세 가지의 필터를 병렬로 사용하여 합하는 식의 구조를 가진다. 이 구조는 결국 여러 필터를 사용함으로써 하나의 필터를 사용할 때 보다 특징을 더 잘 잡아낼 수 있게 된다. 하지만 계산 또한 늘어나게 되는데 이때 그림과 같이 (1,1) 크기의 필터를 추가적으로 적용하게 되고 이를 통해 특징 지도(Feature map)의 크기를 줄여 연산량을 감소시켰다.

3. 제안기법 적용

3.1 시뮬레이션 환경

본 연구의 제안기법은 ROS의 Gazebo환경 내에서 적용되어진다. Gazebo는 로봇에 대한 연구와 테스트를 위한 시뮬레이션 환경으로 빠르고 손쉽게 사용할 수 있다는 장점이 있다. 본 연구에서는 무인선박에 대한 시뮬레이션을 위해 VRX 시뮬레이터를 활용하였다. Virtual Robot X (VRX) 시뮬레이터는 RoboNation과 미해군 연구소(ONR)가 미국 해군 대학원(NPS) 및 Open Robotics(OR)과 협력하여 개발한 오픈 소스 시뮬레이션 환경이다. VRX의 동역학 모델 및 환경 모델은 Gazebo의 독립적인 모델 플러그인으로 구현된다. 선박의 조종 모델은 식은 (Bingham et al., 2019)의 연구를 참조하여 식 (3)으로 표현되며 파랑(Wave)에 의한 하중은 우변에 외력항으로 더해진다. 이때 M_{RB}는 강체의 가속도항의 질량 및 부가질량 계수행렬이고, C_{RB}는 강체의 속도항의 감쇠계수이다. M_A는 유체력에 의한 가속도항의 부가질량계수, C_A와 D는 유체력에 의한 속도항의 감쇠계수이다. 그리고 τ는 선체에 가해지는 외력을 의미하는데, τ_{wind}와 τ_{waves}는 각각 바람에 의해 선박에 가해지는 외력과 파도에 의한 선박에 가해지는 외력을 뜻한다. 무인선 동역학의 상태변수 ν는 전후동요 속도 u, 좌우동요 속도 v, 선수동요 각속도 r로 구성되는 벡터로, v = [u, v, r]과 같이 정의된다. v_r은 조류의

속도를 고려한 상대속도 벡터를 의미한다.

$$M_{RB}\dot{\nu}+C_{RB}(\nu)\nu+M_A\dot{\nu}_r+C_A(\nu_r)\nu_r+D(\nu_r)\nu_r = \tau+\tau_{wind}+\tau_{waves} \quad (3)$$

VRX GAZEBO상의 Wave model은 (Tessendorf, 2001)의 연구. (Fréchot, 2006)의 연구을 참조하는데 이 model은 Gerstner WaveModel에 의해 생성되었다.

$$X = X_0 - \sum_{i=1}^{N} q_i A_i \sin(k_i \bullet X_0 - w_i t + \phi_i) \quad (4)$$

$$z = \sum_{i=1}^{N} A_i \cos(k_i \bullet X_0 - w_i t + \phi_i) \quad (5)$$

Wave model은 식 (4), 식 (5)와 같이 표현한다. 초기위치는 $X_0 = \{x_0, y_0\}$이고, Wave의 높이 z가 생긴 곳에서 $X = \{x, y\}$이다. q_i는 선명도(Steepness)를 뜻하며 A_i는 진폭(Amplitude)을 뜻한다. k_i는 파수벡터(Wavevector)를 뜻하고, w_i는 각주파수, ϕ_i는 위상이다. 파수벡터는 파의 진행방향과 수평인 벡터(Vector)인데 규모는 파수(Wavenumber) $k = 2\pi/\Lambda$와 같다. 이때 Λ는 파의 길이이다. 파수와 각주파수는 deep water dispersion relation 식 (6)과 관련되어 있다.

$$w^2 = gk \quad (6)$$

$$z = (1 - e^{-\frac{t}{\tau}}) \sum_{i=1}^{N} A_i \cos(k_i \bullet X_0 - w_i t + \phi_i) \quad (7)$$

시뮬레이션의 시작부에서 생기는 과도상태의 파도 높이 z는 식 (7)와 같다. Wave field model은 Pierson-Moskowitz wave spectrum과 Constant wavelength-amplitude ratio 두 개를 사용한다.

시뮬레이션의 대상선박으로 WAM-V(Wave Adaptive Modular Vessel)플랫폼을 사용하였다. 선박의 실제 외형과 시뮬레이터에 적용되어진 외형은 Fig. 7에서 확인할 수 있고 주요 제원은 Table 1과 같다. 본 연구에서는 대상 무인선이 추력 배분(Thrust allocation)과정을 통해 모든 수평면 자유도 방향으로 홀로노믹한 기동을 할 수 있도록 각도가 고정된 4개의 추진기를 갖는 형태로 추진시스템을 구성하였다. 추진기 추력 배분은

Table 1 Specification of WAM-V

Length (m)	5
Width (m)	2.5
Height (m)	1
Draft (m)	0.5
Weight (kg)	80
Hull type	Catamaran
Thruster (EA)	4

Fig. 7 Research target vessel (WAM-V platform)

(Yoon et al., 2018)의 연구를 참조하였다. Fig. 8는 WAM-V에 설치된 각 추진기의 배치를 표현하고 있다. (Yoon et al., 2018)의 논문에서는 선박의 수평운동 τ_h을 추력 벡터 f와 추진기의 배치 행렬의 곱으로 표현하였다. 식으로 표현하면 다음 식 (8)과 같다.

$$\tau_h = [F_x F_y M_z]^T = T_h f_h \quad (8)$$

이때 F_x, F_y, M_z는 각각 x방향 힘, y방향 힘, z축 기준 회전 모멘트를 나타내고 T_h는 배치행렬이다. 배치행렬 T_h는 다음 식 (9)와 같다.

$$T_h = \begin{bmatrix} \cos(\alpha_1) & \cdots & \cos(\alpha_4) \\ \sin(\alpha_1) & \ddots & \sin(\alpha_4) \\ d_1\sin(\beta_1) & \cdots & d_4\sin(\beta_4) \end{bmatrix} \quad (9)$$

이때 각각의 추진기에 대한 번호는 Fig. 8에서 확인할 수 있고, α_n와 β_n $(n = 1,...,4)$은 각각의 추진기 번호에 맞는 α, β 각도이다. 이때 α는 x축에 대한 추진방향 각도이고 β는 x-y 평면에서 추진기의 추진방향에 대한 무게중심과 추진기를 이은 선분의 각도이다. 그리고 γ는 x축과 무게중심과 추진기를 이은 선분과의 각도이고 $\beta = \pi - (\alpha - \gamma)$의 식으로 표현할 수 있다. d_n $(n = 1,...,4)$는 선박의 무게중심에서 각각의 추진기까지의 거리를 나타낸다. 이 때 식 (8)에서 추력 f_h를 알기 위해서 배치행렬 T_h의 역행렬을 양변에 곱해주어야 하지만 T_h는 정방행렬이 아니기 때문에 T_h의 의사역행렬을 구해야 한다. 의사역행렬을 구하여 f_h를 구하면 최종적으로 식 (10)이 나오게 된다.

$$f_h = T_h(T_h T_h^T)^{-1}\tau_h = T_h^+ \tau_h \quad (10)$$

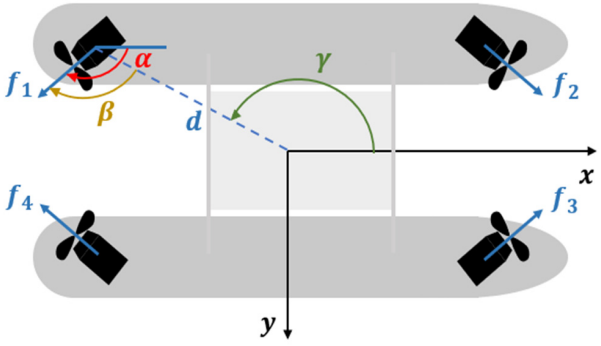

Fig. 8 Geometry arrangement of thrust allocation

3.2 데이터 획득 과정

본 연구에서 사용되는 데이터는 선수각 추종을 수행하는 무인선의 제어시뮬레이션을 VRX시뮬레이터 상에서 수행하고, 이때 획득한 선박의 제어 입력값과 상태정보를 바탕으로 생성된 데이터를 활용한다. 선수각 추종은 무작위로 정의된 임의의 목표선수각을 추종하는 시나리오로 반복적으로 목표선수각이 전환되도록 시뮬레이션 환경을 구성하였으며, 선수각 제어는 사전에 튜닝이 완료된 PD제어기를 사용하였다.

목표선수각을 추종하는 각각의 시나리오는 일정한 확률로 추진기에서 고장이 발생하도록 시뮬레이션 환경을 구성했는데, 고장상황에서는 무인선에 설치된 4개의 추진기 중 1개의 추진기에서 추력이 발생하지 않는다고 가정하였다.

반복적인 선수각 추종 시뮬레이션이 수행되는 중 제어입력 및 상태정보 변수 3종류에 대한 웨이블렛 변환을 통해 정상(Normal) 및 비정상(Abnormal)상황에 대한 학습데이터를 실시간으로 생성한다. 하나의 학습 이미지는 10Hz로 획득된 5초에 대한 시계열데이터 정보들을 담고 있는 스캘로그램 이미지로 정의되는데, 본 연구에서는 시뮬레이션 과정에서 획득된 10,000장의 정상/비정상 이미지를 활용하여 학습을 진행하였다. 정상상태와 비정상상태의 데이터를 토대로 만든 스캘로그램 이미지의 예시는 Fig. 9에서 확인할 수 있다.

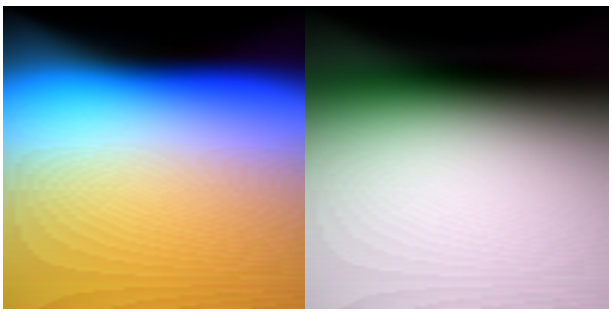

Fig. 9 (Left) Normal image (Right) Abnormal image

3.3 전이 학습

본 연구에서는 GoogLeNet 모델에 전이 학습을 진행한다. 전이 학습을 진행할 때 모델의 Hyperparameter는 Table 2에서 확인할 수 있다.

Table 2에서 확인할 수 있듯이 Optimizer는 sthochastic gradient descent method를 바탕으로 15개의 샘플이미지로 구성된 배치(batch)에 대한 최적화를 통해 네트워크의 학습을 진행하였다. 이때 초기 학습률(Initial learning rate)은 0.0001로 설정하였으며, Learning rate drop factor와 Learning rate drop period는 각각 0.1과 10으로 설정하였다. 학습은 총 20 Epoch에 대해 수행하였다.

이와 같은 Hyperparameter의 설정 값으로 학습을 진행하였을 때 정확도(accuracy)와 손실(loss)은 Fig. 10을 통해 확인할 수 있다. 앞서 3.2장에서 언급한 것과 같이 총 10,000장의

Table 2 Hyperparameters setting for trained GoogLeNet

Hyperparameter	Value
Momentum	0.9
Initial learning rate	0.0001
Learning rate drop factor	0.1
Learning rate drop period	10
Max epochs	20
Batch size	15
Optimizer	Stochastic gradient descent method

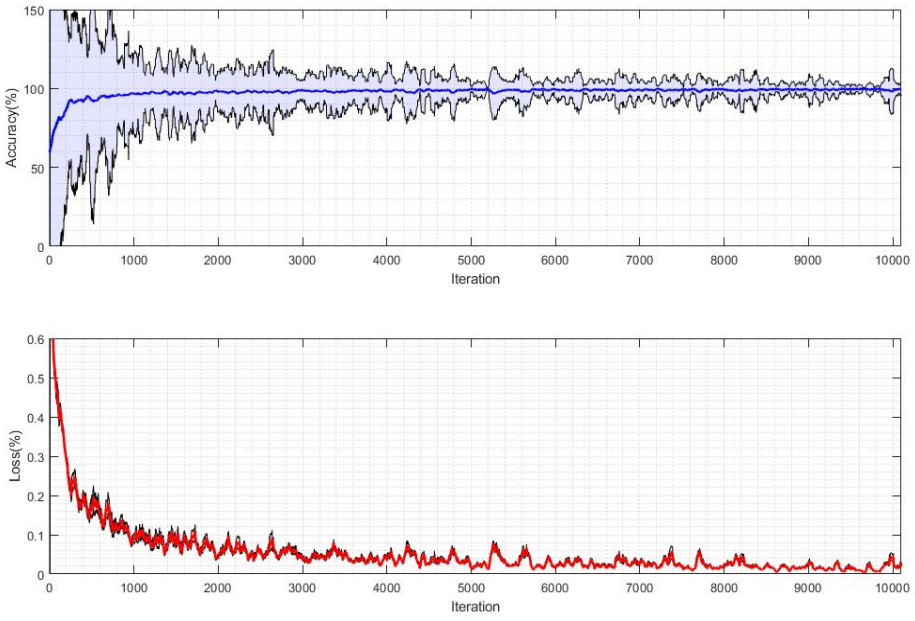

Fig. 10 Accuracy and loss trained by GoogLeNet

데이터가 학습에 사용되었고 학습 데이터(training data)와 검증 데이터(validation data)의 비율은 8:2로 설정하였다.

Fig. 10에서 파란 실선은 학습과정의 진행에 따른 정확도의 그래프를 나타내는데, 정확도 그래프의 윈도우 크기를 50으로 정의한 이동평균(moving average)값을 나타낸다. 불투명하게 표현된 파란색 음영영역은 이동평균을 계산하는데 사용한 윈도우 내에서 계산한 이동분산(moving variance)값의 영역을 나타내는데, 1-σ에 해당하는 영역을 도식화 하여 표현한 그래프이다. 그래프를 통해 보았을 때, 학습의 정확도는 시간이 흐름에 따라 임의의 값으로 수렴하는 경향을 확인할 수 있는데, 이때의 정확도 값은 약 99.3%로 나타난다. 학습이 진행됨에 따라, 이동분산값이 점진적으로 감소는 경향 또한 확인할 수 있다.

Fig. 10 그래프에서 빨간 실선은 학습데이터의 라벨값과 예측값의 차이로 정의되는 손실에 대한 그래프를 나타내는데, 정확도 그래프와 동일하게 윈도우 크기를 50으로 정의한 이동평균(moving average)값을 실선으로 표현하고 있다. 손실값은 학습이 진행됨에 따라 0에 수렴하는 경향을 확인할 수 있다.

4. 성능 검증

본 논문에서 제안하는 방법의 성능을 검증하기 위해, 학습에 사용되지 않은 데이터를 바탕으로 성능 검증 시뮬레이션을 진행하였다. 성능 검증은 2개의 시나리오로 진행된다. Fig. 11은 시나리오 1의 결과 값으로 정상 상태에서 튜닝된 PD제어기를 토대로 150° 의 목표선수각을 추종하며 진행한다. 시나리오 1은 선박의 선수좌현의 추진기에 고장이 발생하는 상황으로 정의하였다. Fig. 12는 시나리오 2의 결과 값으로 정상 상태에서 튜닝된 PD제어기를 토대로 150° 의 목표선수각을 추종하며 진행한다. 시나리오 2는 선박의 선미우현의 추진기에서 고장이 발생하는 상황으로 정의하였다. 이때 ψ는 선박의 선수각을 나타내며 ψ_d는 목표 선수각을 나타낸다. 그림에서 선박의 선수각 그래프를 보았을 때 고장 발생 이전인 30초 전까지 목표선수각을 따라가며 선수각 오차를 줄이는 것을 볼 수 있다. 그리고 선수동요 제어입력 M_z와 선수동요 각속도 r의 그래프를 보면 선박의 선수각 오차가 줄어듦에 따라 0에 수렴하며 입력된 제어입력을 토대로 선박이 목표로 하는 거동을 보이는 것을 알 수 있다. 이렇게 정상 상태에서는 선박이 선수각의 오차를 줄이며 목표 선수각을 추종하는 거동을 보인다. 그런데 30초 지점에서 선박의 추진기에 고장이 발생한다.각각의 시나리오의 궤적에서 고장 발생 이후의 상황을 보면 선박의 목표 경로(path)를 추종하지 못하고 일정 선수각으로 오차가 발생하는 것을 볼 수 있다. 그래프에서 선수각과 목표 선수각을 비교해 보면 30초 이후에 2개의 시나리오 모두 목표선수각인 150° 를 추종하지 못하고 일정 선수각으로 정상상태 오차를 가지며 진행한다. 선수동요 제어입력 M_z와 선수동요 각속도 r의 그래프를 비교해보면, 30초 이후에 선수동요 제어입력 M_z의 값은 계속해서 증가하며 일정 값으로 정상 상태가 유지되는데 선박의 선수동요 각속도 r는 0에 수렴하는 형태를 보이며 선수각이 입력된 값에 맞게 제어되지 못하여 선수각 오차가 계속해서 발생하게 된다. 제어기는 정상상태에 맞게 튜닝 되었기 때문에 고장 상황이 발생하면 목표 선수각을 정확히 추종하지 못하고 정상상태 오차를 가지게 된다.

이러한 특성을 가지는 데이터를 사전에 학습된 고장 감지 모델로 고장유무를 판단한 결과를 그림에서 Normal/Abnormal로

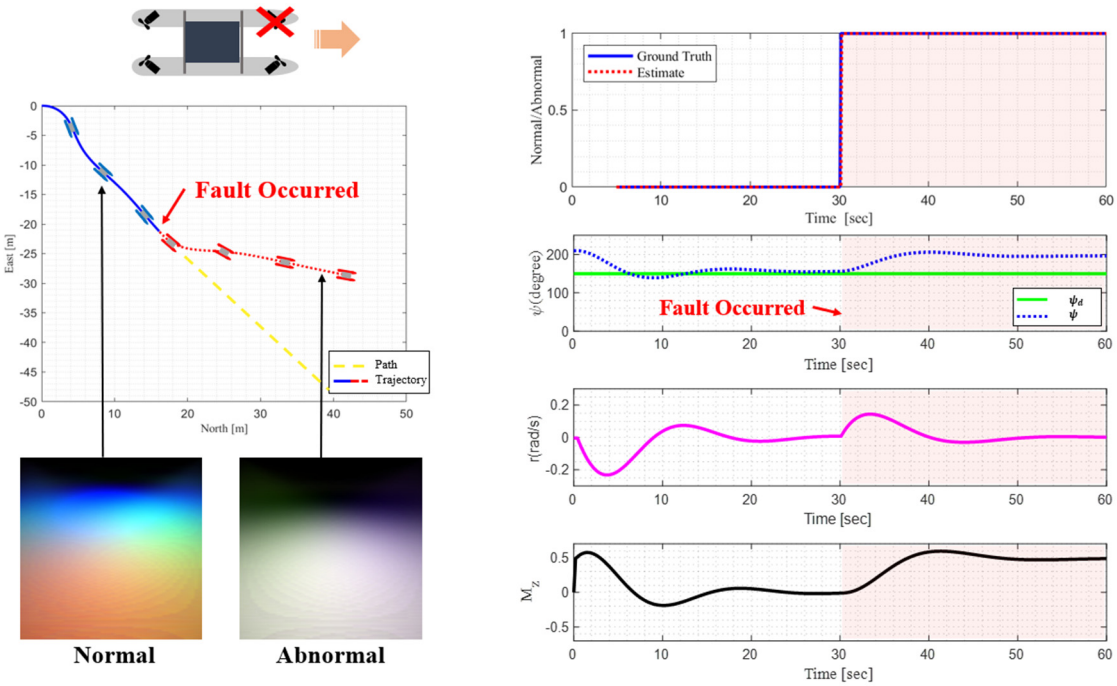

Fig. 11 Results for the failure scenario 1 (bow-port thruster failure at 30 seconds)

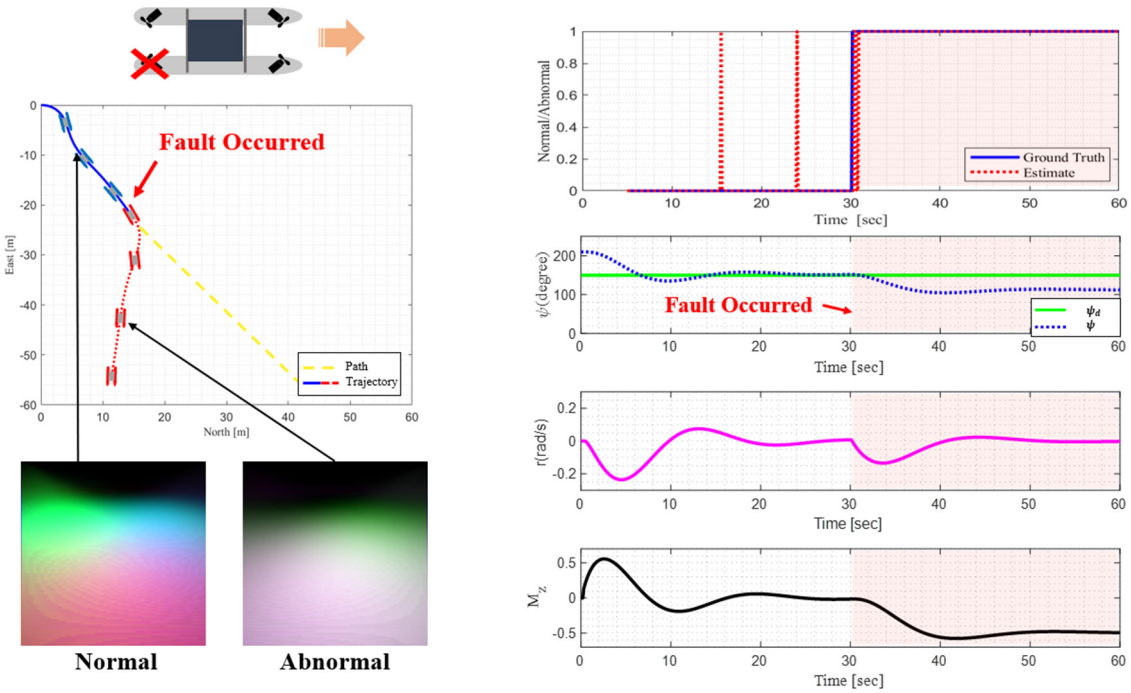

Fig. 12 Results for the failure scenario 2 (stern-starboard thruster failure at 30 seconds)

표시된 그래프로 확인할 수 있다. 우선 시나리오 1에서는 정상 상태와 30초 이후 고장이 발생한 비정상상태를 오차 없이 예측하는 것을 볼 수 있다. 시나리오 2에서는 시나리오 1과 마찬가지로 정상 상태와 비정상상태를 비교적 높은 성능으로 예측하지만 약 16초와 24초의 구간에서 튀는 현상이 발생하는 것을 볼 수 있다. 이러한 오류가 생겨 고장 유무가 정확하게 판단되지 않는 현상들이 발생하는 것을 방지하기 위해 본 연구에서는 1초 이상 현재상태가 지속될 경우에 현재 상태를 신뢰성이 있다고 판단하였다.

다양한 상황에서의 보다 신뢰성 있는 시뮬레이션 검증을 위해 몬테 카를로 시뮬레이션(Monte-Carlo simulation)을 진행하였다. 불확실한 조건하에서 발생하는 결과의 범위를 얻을 수 있는 수학적 기법인 몬테 카를로 시뮬레이션을 진행하기 위해 데이터는 다음과 같은 방법으로 수집하였다. 선박은 앞선 시나리오와 마찬가지로 목표선수각을 추종하고 고장상황을 포함하는 40초의 데이터로 수집하였다. 이때 직진 또는 선회하는 등 다양한 상황의 조건을 만족시키기 위해 목표선수각을 0° ~ 360°를 60° 간격으로 나눠 후보 목표선수각을 설정하였고, 하나의 시나리오가 실행될 때 마다 후보 목표선수각중 임의로 하나의 목표선수각을 추종하는 형태로 구성하였다. 그리고 4개의 추진기 중 임의로 하나의 추진기를 고장 추진기로 설정하였다. 추가적으로 선박이 정상적인 상태로 갈 때 까지 출발 후 10초 동안은 목표선수각 추종이나 고장상태 없이 직진만 수행하였다. 선박이 출발하고 10초 이후부터 목표선수각을 추종함과 동시에 고장 감지를 시작하였고 고장의 발생 시점도 15초에서 30초 사이에 균등분포의 확률로 결정되게 설정하였다. 이와 같이 선박의 다양한 상황에서의 고장 발생 감지 성능을 검증하기 위해 500개의 테스트 데이터를 생성하였다. Fig. 13에서 500개의

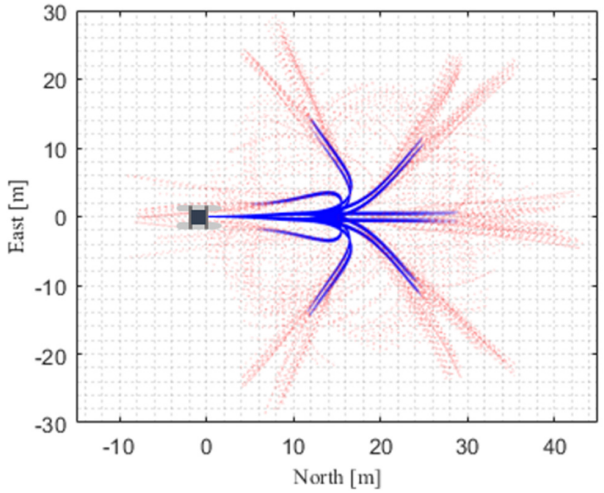

Fig. 13 Monte-carlo simulation result for validation (USV trajectory)

Table 3 Failure detection performance

Performance index	Value
Detection accuracy (%)	100
Detection time mean (sec)	3.46
Detection time standard deviation (sec)	1.96

테스트 데이터의 궤적을 확인할 수 있고 앞선 시나리오 1,2와 마찬가지로 파란실선은 정상상태의 궤적이고 빨간 점선은 고장 발생시 궤적을 나타낸다. 그리고 Table 3을 통해 500개 테스트 데이터의 감지 정확성을 확인할 수 있다. 결과는 500개의 테스

트 데이터 모두 고장 상황을 감지하였다. 추가적으로 고장의 실제 발생 시점과 고장 감지 시점의 차이를 계산하여 탐지 시간을 500개의 각 테스트 데이터에서 계산하였고, 그 탐지 시간의 평균, 표준편차 또한 Table 3에서 볼 수 있다. 고장은 평균 약 3.46초의 시간이 지났을 때 감지되었고 표준편차는 1.96초 이다.

5. 결 론

본 연구는 ROS Gazebo 시뮬레이션 환경에서 수집한 데이터를 토대로 선박의 제어입력과 상태정보를 활용해 고장 유무를 감지하는 기법을 제안하였다. 이때 제어입력과 상태정보를 스캘로그램 이미지로 변환하여 이미지 분류 네트워크인 GoogLeNet에 전이학습을 진행하여 고장 감지를 위한 합성곱신경망 네트워크를 구성하였다.

제안 방법은 웨이블렛 변환을 통한 스캘로그램 이미지의 특성을 갖기 때문에 갑작스러운 기동변화에 따른 고장여부 판단을 수행할 수 있다는 장점이 있으며, 학습에 사용되지 않은 검증데이터에 대하여 약 99.3%의 높은 정확도를 보이며 고장여부를 판단할 수 있음을 확인하였다.

추진기 고장이 발생하는 무인선 운항 시나리오에 대하여 실시간으로 추진기의 고장을 검진하는 시뮬레이션을 통해 성능검증을 수행하였다. 검증결과 제안기법은 높은 정확도와 강건성을 가지며 실시간으로 고장상황에 대한 검진이 가능함을 확인할 수 있었다.

본 연구의 한계는 실제 해상환경이 아닌 시뮬레이션 환경에서 개발과 검증이 진행되었다는 점을 들 수 있다. 실제 해상환경은 환경외란의 영향이 많은 영향을 미치고, 센서(Sensor)의 계측잡음 등에 대한 신호특성을 고려해야 한다. 더불어, 고장의 발생여부를 판단하는데 그치지 않고, 고장의 유형이나 고장장비를 특정하는 고장진단 문제로 연구를 확장할 예정이다.

후 기

이 논문은 2021년도 정부(미래창조과학부)의 재원으로 한국연구재단 생애 첫 연구 사업의 지원을 받아 수행된 연구임 (NRF-2021R1G1A1095671)

References

Amoozgar, M.H., Abbas, C., & Youmin Z., 2013. Experimental test of a two-stage Kalman filter for actuator fault detection and diagnosis of an unmanned quadrotor helicopter. *Journal of Intelligent & Robotic Systems*, 70.1, pp.107-117.

Bingham, B., Agüero, C., McCarrin, M., Klamo, J., Malia, J., Allen, K., Lum, T., Rawson, M. & Waqar, R., 2019. Toward maritime robotic simulation in gazebo. *OCEANS 2019 MTS/IEEE SEATTLE* , 1-10 October 2019.

Cho, S., Choi, M., Gao, Z., & Moan, T., 2021. Fault detection and diagnosis of a blade pitch system in a floating wind turbine based on Kalman filters and artificial neural networks. *Renewable Energy*, 169, pp.1-13.

Fréchot, J., 2006. Realistic simulation of ocean surface using wave spectra. Institute for Systems and Technologies of Information, Control and Communication. *Proceedings of the first international conference on computer graphics theory and applications (GRAPP 2006)*, Portugal, February 2006, pp.76-83.

Jayalakshmy, S., & Sudha, G.F., 2020. Scalogram based prediction model for respiratory disorders using optimized convolutional neural networks. *Artificial intelligence in medicine*, 103, 101809.

Kim, B.M., 2018. Trend of image classification technology based on deep learning. *Korea Institute Of Communication Sciences*, 35(12), pp.7-14.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A., 2015. Going deeper with convolutions. *Proceedings of the IEEE conference on computer vision and pattern recognition,* USA, June 2015, pp.1-9

Taheri, S., Ahmadi, A., Mohammadi-Ivatloo, B., & Asadi, S., 2021. Fault detection diagnostic for HVAC systems via deep learning algorithms. *Energy and Buildings*, 250, 111275.

Tessendorf, J., 2001. Simulating ocean water. *Simulating nature: realistic and interactive techniques*. SIGGRAPH, 1(2), pp.5.

Yoon, S.M., Lee, C.M., & Kim, K., 2018. Implementation of heading angle and depth keeping control of ROV with multiple thrusters by thrust allocation. *Journal of Ocean Engineering and Technology*, 32(1), pp.68-75.

Zhang, M.J., Wang, Y.J., Xu, J.A., & Liu, Z.C., 2015. Thruster fault diagnosis in autonomous underwater vehicle based on grey qualitative simulation. Ocean Engineering, 105, pp.247-255.

백 승 대

우 주 현