

# 딥러닝을 적용한 영상기반 군함 식별정보 획득에 관한 연구

강지영<sup>1)</sup> · 김우주<sup>\*,1)</sup>

<sup>1)</sup> 연세대학교 산업공학과

## A Study on the Acquisition of Identification Information from Warship Image with Deep Learning

Jiyoung Kang<sup>1)</sup> · Wooju Kim<sup>\*,1)</sup>

<sup>1)</sup> Industrial Engineering, Yonsei University College of Engineering, Korea

(Received 27 July 2021 / Revised 1 November 2021 / Accepted 13 December 2021)

### Abstract

Identifying warships contacted at sea is important to prepare for threats. It is necessary to obtain a basis to identify warships. In this study, we propose a 2-step model that acquires the warship's type and hullnumber with identification information from the warship images. The model classifies the warship's type and detects its hullnumber area by applying object detection, then recognizes hullnumber through text recognition algorithms. Proposed model achieved high performance by using state-of-the-art deep learning algorithms.

Key Words : Object Detection(객체탐지), Text Recognition(문자인식), Deep Learning(딥러닝), Warship Identification(군함 식별)

### 1. 서론

최근 딥러닝 기술 중 컴퓨터 비전의 발전은 다양한 산업분야에서 널리 활용되고 있다. 자율주행, 방사선 촬영영상 분석, 제조 공정의 이상 탐지 등 산업 전반에 걸쳐 컴퓨터 비전을 접목하고 있다. 이러한 기술 성장에 맞추어 국방 분야에서도 이를 적용한 연구가 활발해 지고 있으며, 해상활동에서 획득되는 군함 영상에 대해서도 딥러닝을 적용한 몇몇 연구가 있다.

해당 연구에서는 해상에서 군함, 상선, 여객선 등을 탐지<sup>[1]</sup>하거나 군함의 국적을 식별<sup>[2]</sup>하기 위해 객체탐지(Object Detection)를 적용하였고, 군함의 함급/함종을 식별하기 위해 영상분할(Segmentation)을 적용하는 것을 제안<sup>[3]</sup>하고 있다. 해상에서 접촉한 군함을 식별하는 것은 위협을 대비하기 위해 중요하다. 위의 연구에서는 해상에서 접촉물과 군함에 대한 식별의 중요성을 기반으로 딥러닝을 연구하였으나, 군함에 대해 식별을 근거할 수 있는 구체적인 정보 획득은 제한적이다.

본 연구에서는 최신 딥러닝 기술을 적용하여 해상에서 군함에 대한 식별정보를 획득하기 위한 모델을 제

\* Corresponding author, E-mail: wkim@yonsei.ac.kr  
Copyright © The Korea Institute of Military Science and Technology

안하고자 한다. 본 연구에서 획득하고자 하는 식별정보는 군함의 함형과 함정번호이다. 이때 함형은 함정이 수행하는 임무에 따라 분류하는 기준으로 구축함(DDH), 초계함(PCC), 구조함(ATS), 군수지원함(AOE) 등을 말하며, 함정번호는 해당 함정에 하나씩 부여되는 특정한 번호를 말한다. 함형과 함정번호는 나라마다 부여하는 기준에 근거하고 있다. 이에 함형과 함정번호는 특정 군함을 지정할 수 있는 정보이므로 표적 여부를 결정할 수 있는 근거가 될 수 있다. 또한, 이 정보를 활용하여 온라인에 축적된 탑재 무장, 수행 임무 등 기초 정보에 접근할 수 있다. 그러므로 함형과 함정번호를 획득하는 것은 영상에서 단순히 군함의 존재 여부를 판단하는 것이 아닌 군함을 식별할 수 있는 중요한 정보라고 할 수 있다.

본 연구에서는 우선으로 우리나라 군함을 중심으로 하여 연구하였으며, 해상에서 군함 영상을 활용해 함형과 함정번호를 획득할 수 있는 모델을 연구하였다. 제안하는 객체탐지와 문자인식의 딥러닝 알고리즘을 적용하여 2단계로 구성하였고, 함형에 따른 함정번호 부여 기준을 활용할 수 있도록 정보를 추가하였다.

## 2. 선행연구

### 2.1 객체탐지

객체탐지(Object Detection)란 이미지 내에 포함된 객체를 분류하고 위치를 추정하는 것을 말한다. 영상정보로부터 특징을 추출하는 합성곱 신경망(CNN)을 활용한 객체탐지는 크게 2-stage 방식과 1-stage 방식으로 나눌 수 있다. 2-stage 방식은 객체가 존재할 것으로 예측되는 영역을 찾는 객체영역 후보 추정(Region Proposal)과 해당 객체를 인식하는 분류(Classification)를 순차적으로 실행하는 방식으로 대표적인 모델로는 Faster R-CNN<sup>[4]</sup> 등의 R-CNN 계열이다. 특히 Faster R-CNN은 합성곱 신경망을 적용하여 특징맵(feature map)을 추출한 다음 객체가 있을 것으로 추정되는 관심영역(RoI, Region of Interest)을 선정한다. RPN(Region Proposal Network)를 통해 관심영역(RoI)를 선정하고, 각각의 영역에 대해 분류를 진행한다. 이 방식은 정확도가 높지만 2개의 심층 신경망을 연산하기 때문에 속도가 느리다는 단점이 있다. 이러한 단점을 보완하기 위해 1-stage 방식이 제안되었다. 1-stage 방식은 객체영역 후보 추정과 객체분류를 동시에 진행함

으로써 속도를 향상하였다. 대표적인 모델은 YOLO<sup>[5]</sup>, SSD<sup>[6]</sup>, EfficientDet<sup>[7]</sup> 등이 있다. 객체탐지는 자율주행, CCTV 기반 영상보안 분야 등에서는 활용되는 알고리즘으로 정확도만 아니라 신속성의 중요도 또한 높아지고 있는 추세이다. 특히, 1-stage 모델 중 속도가 빠른 모델이 바로 YOLO계열의 모델이다. 이와 같은 이점으로 YOLO 계열은 많은 객체탐지와 관련된 연구에서 활용되고 있다. YOLO은 합성곱 네트워크를 통해 특징맵(feature map)을 추출하여 경계상자 정보, 객체를 포함하고 있을 확률, 객체의 클래스를 생성, 객체를 탐지하는 1-stage 구조이다. 특히 YOLOv4<sup>[8]</sup> 모델의 속도는 65FPS 이상을 달성한 실시간 객체탐지가 가능한 모델이다. 이는 1-stage 모델 중 가장 빠른 모델로서 최신 딥러닝을 기법을 활용하여 정확도와 속도를 향상하였다.

본 연구에서 제안하고자 하는 해상에서 접촉한 군함 정보를 획득하는 것은 정확성뿐만 아니라 신속성 역시 중요한 요소이다. 이에 따라 실시간 정보를 획득할 수 있도록 객체탐지 알고리즘으로 YOLOv4를 Baseline으로 선정하였고, 객체탐지 알고리즘을 적용하여 함형을 분류하고 함정번호 영역을 인식할 수 있도록 하였다.

### 2.2 문자인식

문자인식 기술은 문자 영역을 검출하는 Text Detection과 검출된 영역 내 문자가 무엇인지 판별하는 Text Recognition으로 구분할 수 있다. 문자인식의 기술은 실생활에서 자동차 번호판 검출, 가스계량기 검침 등 적극적으로 활용되고 있다. 본 연구에서는 함정번호, 즉 숫자를 인식하기 위한 Text Recognition 영역이다. Text Recognition에 관한 대표적인 모델로는 이미지 기반 시퀀스(sequence) 인식방법인 CRNN(Convolutional Recurrent Neural Network)<sup>[9]</sup>이 있다. 이미지 내에서 문자는 연속된 형태(Sequence)로 표현되고 있으며 이를 다루는 순환 신경망(RNN, Recurrent Neural Network)을 적용하는 것이 필요하다고 설명하고 있다. 합성곱 신경망 계층을 통해 추출한 특징맵(feature map)을 특징시퀀스(feature sequence)로 변환하여 순환 신경망 계층으로 입력함으로써 텍스트 연속 벡터를 예측한다. 특히 순환 신경망으로부터 만들어진 프레임은 CTC(Connetionist Temporal Classification)<sup>[10]</sup> 알고리즘을 적용하여 가장 높은 확률을 가진 레이블을 찾도록 한다.

CRNN 모델에서는 VGG 구조를 기반으로 하여 합성곱 신경망 계층에 적용하였고, 순환 신경망 계층에는 Bidirectional-LSTM구조<sup>[11]</sup>를 적용하였다. 특히 Bidirectional LSTM의 양방향 연결은 이미지 기반 시퀀스(sequence)에서 유용하고 상호 보완적 역할을 수행한다. 합성곱 신경망과 순환신경망 계층이 연결된 CRNN 기본 알고리즘은 문자열의 연속적 형태를 고려할 때 각 신경망의 이점을 잘 활용할 수 있는 구조이다. 본 연구에서는 변형된 CRNN을 구성하여 함정번호를 인식할 수 있도록 하였다.

### 3. 데이터 구축

현재 선박 관련 구축된 공개데이터 위성에서 촬영한 이미지 데이터셋이 있으며, CIFAR-10과 같은 대규모 이미지 데이터셋에는 선박(ship class)과 관련한 이미지를 제공하고 있다. 하지만 이는 모두 군함이라는 특정한 분류기준으로 구축되어 있지 않기 때문에 군함과 관련한 딥러닝 연구를 위해서는 별도의 데이터셋 구축이 필요하다. 이에 본 연구에서는 군함의 식별정보를 획득하기 위해 온라인상에 공개된 우리나라 군함 이미지 1080장을 구축하였다. 군함의 함형은 해상에서 주요 작전을 수행하는 전투함 및 지원함으로 DDG, DDH, FFG, FF, PCC, PKG, LST, MLS, AOE, ATS 등 총 10개의 함형 선정하였고, 각 함형에 속한 함정번호는 『2020-2021 한국군 무기연감』<sup>[12]</sup>을 기준으로 992(DDG), 811(FFG), 57(AOE) 등 총 10개 함형, 68척의 함정에 관한 이미지 데이터셋을 Fig. 1과 같이 수집하였다. 구축된 데이터셋의 세부내용은 Table 1과 같다. 함형별로 운영하고 있는 함정의 척수가 상이함에 따라 구축된 함형별 데이터 수가 다소 차이가 있다.



Fig. 1. Sample image of warship

Table 1. Dataset table

Ship type (standard displacement)	Data	Ship type (standard displacement)	Data
DDG (7,600 t)	114	PKG (440 t)	121
DDH (4,400 t)	141	LST (2,600t~4950 t)	109
FFG (2,300 t)	122	MLS (2,400t~3,300 t)	80
FF (2,000 t)	105	AOE (4,200t~10,600 t)	105
PCC (1,300 t)	119	ATS (3,747 t)	64

#### 3.1 객체탐지를 적용한 식별 정보 획득

합성곱 신경망 네트워크가 깊어짐에 따라 객체탐지의 성능이 크게 향상되었다. 그러나 합성곱 신경망을 통해 특징맵(feature map)을 추출하는 과정에서 소형 객체의 특징(feature) 정보를 소실할 수 있는 어려움이 있다<sup>[13]</sup>. 본 연구에서 인식하고자 하는 함정번호는 영상정보 내에서 함정에 비해 매우 작은 소형 객체로 간주할 수 있다. 이로 인해 객체탐지 알고리즘에서 함정번호가 정확하게 인식되지 않을 가능성이 있다. 이에 수집된 데이터 중 500장을 대상으로 YOLOv4 알고리즘만을 적용하여 함정정보와 함정번호 획득 가능성에 대한 실험을 우선으로 수행하였다. 군함의 영상에서 함형과 함정번호를 동시에 인식할 수 있도록 하기 위해 실험에서는 함형 10개, 함정번호 숫자 10개, 총 20개 class로 지정하여 실험을 진행하였다. 함형의 AP (Average Precision)는 90 % 이상으로 높은 성능을 보여주었으나, 함정번호 숫자에 대한 AP는 40~50 %로 인식률이 현저히 낮은 것을 확인할 수 있다.(Fig. 2) 따라서 본 연구에서는 함정번호의 인식률을 향상하기 위하여 함정번호 영역을 별도의 class로 지정하여 객체탐지를 진행하고 한다.



(a) 모델 예측 : PKG 768      (b) 모델 예측 : ATS 3  
 실제 정보 : PKG 728      실제 정보 : ATS 31

Fig. 2. Result of object detection(ship type, hullnumber)

다시 말해서, 함형의 분류 및 함정번호 영역의 탐지를 위해 객체탐지를 적용하였고, 함정번호 인식을 위해 문자인식을 적용하였다. 제안하는 모델은 단계화하여 구현할 수 있도록 구성하였으며, 각 단계에 적합하도록 데이터셋을 구축하였다.

### 3.2 객체탐지 학습용 데이터 생성

객체탐지의 학습을 위해서는 이미지 내에 포함된 객체(object)의 속성(class)과 이미지 내 실제위치(ground-truth)를 지정하기 위한 라벨링(labeling) 작업이 필요하다. 함형 10개와 함정번호 영역 1개, 총 11개로 라벨링 작업을 수행하였고, 라벨링 작업을 지원해 주는 오픈소스인 LabelImg tool<sup>1)</sup>을 사용하여 주석(annotation)을 생성하였다.

### 3.3 함정번호 인식을 위한 학습용 데이터 생성

함정번호는 문자인식 학습을 위하여 객체탐지의 실제위치(ground truth)와 같도록 별도 분할(crop)하여 Fig. 3과 같이 학습 데이터셋을 구축하였다. 문자인식에서는 함정번호 이미지와 정답(ex. 32, 560, 681, 973)을 쌍으로 하여 주석을 생성하였다. 또한, 함정번호의 수가 68개인 것을 고려하여, 회전, 좌우 이동, 밝기변화를 주어 데이터를 증강하였다.



Fig. 3. Sample image of hullnumber

## 4. 제안하는 모델

본 연구에서 군함 이미지에서 함형과 함정번호를 동시에 인식할 방법을 제시한다. 함형과 함정번호의 각각의 정보만 있다면 군함에 대한 정확한 정보 사항을 획득하는데 제한이 있다. 군함은 동일한 함형이라도 건조연도에 따라 외형이나 보유 무장 등이 조금씩 상이할 수 있다. 또한, 함정번호의 숫자를 정보만으로는 군함 정보를 왜곡할 가능성이 있다. 함정번호가 군함의 선수(bow)에 위치해 있는데, 이때 군함의 움직임

이나 해수로 함정번호 정보가 소실될 수 있기 때문이다. 예를 들면, 함정번호가 “570”의 함형은 “MLS”이나, 만약 “0”의 정보가 소실되면 함정번호 “57”로 인식이 되며 이때의 함형은 “AOE”이며, 이는 왜곡된 정보를 생성할 수 있다. 이에 따라 본 연구에서 제안하는 모델은 군함의 함형과 함정번호를 인식함으로써 특정 군함 정보를 정확하게 생성할 수 있도록 하는 특징을 가지고 있다.

앞서 3.1 실험결과 토대로 객체탐지 알고리즘에서 함정번호를 개별숫자로 인식하는 것은 특징맵 추출 과정에서 정보소실로 성능 저하를 확인할 수 있었다. 또한, 군함과 관련된 공개이미지는 대규모 데이터셋 구축에 제한점이 있어 함정번호를 이미지 분류(Classification)의 문제로 접근하는 것은 높은 성능을 달성하는 데 제한점이 많다. 이에 본 연구에서는 분류 문제로 접근하는 것이 아닌, 함정번호를 구성하는 숫자를 인식하는 문자인식(Text Recognition)으로 단계화하는 모델을 제안한다.

제안하는 군함의 함형과 함정번호 획득 모델은 크게 2단계 과정으로 구성하였다. 1단계에서는 객체탐지 알고리즘을 적용하여 군함의 함형(ship type)을 분류하고 군함 선체에 표시된 함정번호(Hull number) 영역을 탐지한다. 2단계에서는 1단계에서 탐지된 함정번호 영역을 별도 분리하여 해당 함정의 숫자 인식에 대한 단계를 진행한다. 문자인식 진행 후 함형에 따른 함정번호 정보를 입력하여 함정번호를 출력한다.(예를 들면 우리나라의 경우 함형 DDG는 900번대 함정번호가 부여되고, FFG는 800번대 함정번호를 부여된다.) 모델은 객체탐지 알고리즘에 의해 인식된 군함의 영역(Bounding box)과 함형(class)를 저장하고, 군함의 영역 내부에 포함된 함정번호영역(이미지)을 Crop하여 문자인식 알고리즘에 입력한다. 문자인식의 결과값인 함정번호에 함형에 따른 함정번호 정보를 추가하고, 앞서 저장된 함형과 최종 군함 식별정보(함형, 함정번호)를 출력한다. 모델 구조는 Fig. 4와 같다.

### 4.1 함형과 함정번호 영역의 객체탐지

함형의 분류와 함정번호 영역을 탐지하기 위한 1단계에는 속도와 정확도가 높은 YOLOv4 모델을 기본 구조로 하였다. YOLOv4 모델은 CSPDarknet53, SPP, PAN, YOLOv3 방식으로 구성되어 있다. CSPDarknet53은 Darknet53구조에 CSP(Cross-Stage-Partial)를 적용하였다. CSP는 입력 계층을 나누어 일부분은 합성곱

1) Tzutalin. LabelImg. Git code (2015).  
<https://github.com/tzutalin/labelImg>

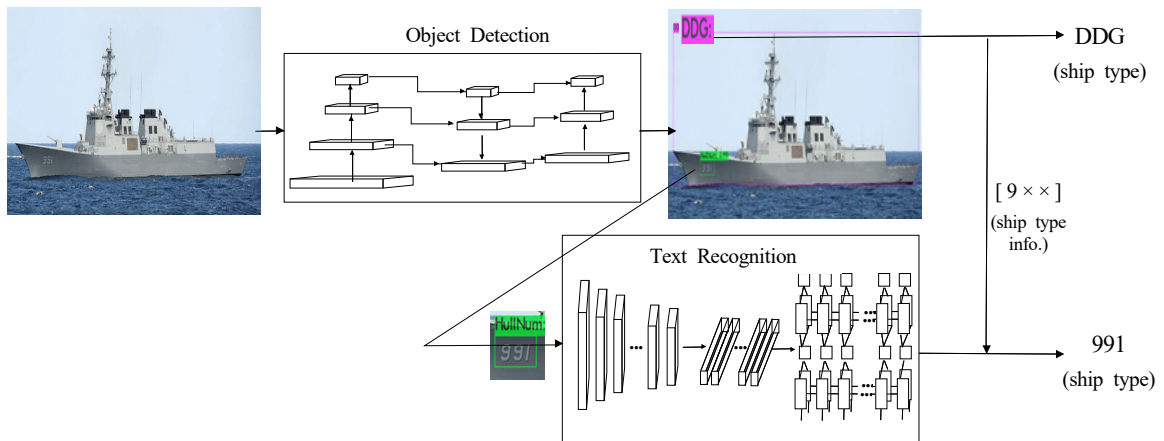


Fig. 4. Proposed model structure

연산을 진행하고, 합성곱 연산 결과를 나머지 일부분과 결합(Concatenation)하는 구조로서, 이를 이용하여 CSPDarknet53는 정확도를 유지하면서 연산량을 줄이는 효과를 달성할 수 있다. CSPDarknet53의 뒤에 배치되는 SPP(Spatial Pyramid Pooling)<sup>[14]</sup>의 기능은 고정된 크기에 의해 제한되지 않는 합성곱 신경망의 입력을 만드는 것이다. SPP는 서로 다른 4개의 척도를 가지는 최대풀링을 사용하여 특징맵을 생성한다. SPP는 네트워크의 실행 속도를 줄이지 않으면서 수용 필드를 늘리고 중요한 컨텍스트 특징을 효과적으로 분리하는 역할을 수행한다. SPP 이후 PAN(Path Aggregation Network)<sup>[15]</sup>을 연결하였다. PAN은 하향식 피쳐 피라미드 뒤에 상향식 경로 확대 구조를 통해 특징맵을 재구성하여 객체를 예측하는 부분으로 전달하는 역할을 수행한다. 객체를 예측하기 위해 세 가지 다른 척도(Scale)로 객체를 예측하는 YOLOv3의 방식을 사용하였다. 또한 저자<sup>[8]</sup>는 Attention 모듈을 결합시 연산량 추가보다 성능향상의 효과가 더 크다고 설명하고 있다. Attention 모듈은 특징맵에서 강조해야 할 특징을 찾아 특징의 중요도를 조정하는 역할을 수행하며, YOLOv4에 적용한 Attention 모듈의 방식은 SAM(Spatial attention)<sup>[16]</sup>을 변형한 방식이다. 본 연구에서는 저자가 제안하는 YOLOv4와 Attention 모듈을 결합한 SAM-YOLOv4을 객체탐지 단계의 알고리즘으로 선정하여 실험을 진행하였다.

#### 4.2 합성번호 인식

문자인식을 위한 CRNN의 기본적인 구조는 Table 2

의 #1과 같으며 합성곱 신경망 계층, 순환 신경망 계층, Transcription 계층으로 구성되어 있다. 합성곱 신경망 계층은 입력 이미지에서 합성곱 연산을 통해 특징맵을 추출한다. 특징맵은 연속된 정보로 변환하는 과정을 거쳐 순환신경망 계층에 입력되며, Transcription 계층에서는 순환신경망 계층에 의해 예측된 값을 label sequence로 변환한다. 이때 예측된 값은 해당문자 또는 문자가 존재하지 않는 공백(blank)으로 매핑된다. CTC 알고리즘에 의해 반복되는 레이블은 합쳐지고, 공백(blank)을 삭제하여 최종 문자열을 출력한다.

CRNN의 합성곱계층의 특징은 3×3 커널(kernel)을 사용하는 VGG<sup>[17]</sup> 형태를 기반으로 하고 있으며, 3번째와 4번째 최대풀링 계층에서 1×2 보폭(strides)를 적용하여 너비의 크게 하는 특징맵을 생성할 수 있도록 하였다. 또한 순환 신경망에는 순방향의 정보와 역방향의 정보가 호 교환할 수 있는 Bidirectional LSTM를 사용하였다. 본 연구에서는 기존 VGG형태의 합성곱 신경망 계층을 Table 2와 같이 변경하였다. 합성곱 계층에 이전의 특징맵 정보를 연결하는 ResNet<sup>[18]</sup> 형태와 DenseNet<sup>[19]</sup> 형태로 각각 변환하여 구성하였다. Resnet의 Residual Block의 특징은 합성곱 층의 입력과 출력을 합하는 구조이다. DenseNet은 합성곱 층을 통해 이전에 생성한 특징맵을 결합(Concatenation)하는 DenseBlock과 DenseBlock 사이에 1x1 합성곱과 풀링 Transition 층에 두어 특징맵의 크기를 줄이는 구조이다. 본 연구에서는 합성곱 계층 변형을 통해 보다 정확한 특징맵을 추출할 수 있도록 실험을 진행하였다. 특히 DenseNet은 구조의 특성상 입력 이전의 모든 레

Table 2. Structure of text recognition

	#1	#2	#3	#4	#5
Convolution layer 구조	<i>(VGG Type)</i> Convolution(64, k:3×3) MaxPooling(2×2) Convolution(128, k:3×3) MaxPooling(2×2) Convolution(256, k:3×3) Convolution(256, k:3×3) MaxPooling(1×2) Convolution(512, k:3×3) BatchNorm(512) Convolution(512, k:3×3) BatchNorm(512) MaxPooling(1×2) Convolution(512, k:2×2)		<i>(ResNet Type)</i> Convolution(64, k:3×3) BatchNorm(64) Residual Block(64, k:3×3) x2 Residual Block(128, k:3×3) x2 Residual Block(256, k:3×3) x2 Residual Block(512, k:3×3) x2 MaxPooling(4×4)	<i>(DenseNet Type)</i> Convolution(64, k:7×7) BatchNorm(64) MaxPooling(1×2) DenseBlock(conv(1×1), conv(3×3))x6 Transition Layer(conv(1×1), AvgPool(2×2)) DenseBlock(conv(1×1), conv(3×3))x12 Transition Layer(conv(1×1), AvgPool(1×2)) DenseBlock(conv(1×1), conv(3×3))x24 Transition Layer(conv(1×1), AvgPool(2×2)) DenseBlock(conv(1×1), conv(3×3))x16 BatchNorm(1024)	
Recurrent layer	Bi-LSTM × 2		Bi-LSTM × 2	Bi-LSTM × 2	
Input image size	32 × 32	64 × 32	256 × 32	64 × 32	128 × 32
예측 프레임수	7	15	8	7	15

이어를 결합하여 특징맵을 생성하므로 초기 정보를 네트워크 후반부까지 효과적으로 전달할 수 있다. 이에 DenseNet 네트워크는 이미지와 관련된 여러 네트워크에서 적용되고 있다. 따라서 본 연구에서도 CRNN의 합성곱 계층에 DenseNet을 적용하는 것을 제안하고 있으며, 합성곱 계층을 변경하여 비교 실험을 진행하였다.

기존 CRNN에서는 길이가 정해져 있지 않은 실제 이미지 내 문자열의 특성을 고려하여 입력 이미지의 가로(w)의 크기를 정하지 않고, 높이(h)만을 32로 정하여 학습하였다. 본 연구에서는 합정번호 숫자는 2-3개로 정해져 있으며, 해당 영역의 평균의 픽셀크기는 63(w) × 36(h)인 점을 고려하여, 입력이미지 크기를 정하여 학습을 진행하였다. 특히 입력이미지의 크기와 합성곱 계층의 형태에 따라 예측프레임의 크기는 달라진다. CTC알고리즘은 높은 확률을 가지고 있는 레이블을 예측하는데, 예측프레임 수를 조정하여 성능을 확인할 수 있도록 입력값을 조정하였다.

마지막으로 합정번호 영역에 대한 문자인식 후 함형에 따른 정보, 즉 함형에 따라 특정되는 시작 숫자를 입력하여 최종 출력될 수 있도록 하였다.

## 5. 실험결과

구축된 데이터 1080장 중 836장은 학습, 244장은 평가에 사용하였다. 제안하는 모델의 학습은 단계별로 입력데이터의 크기와 속성(class) 수가 상이하고, 각 단계의 목표가 다르므로 1단계 YOLOv4 구조와 2단계 CRNN 구조를 각각 진행하였다.

### 5.1 객체탐지 실험결과

본 연구에서 구축된 군함 관련 데이터셋은 대규모로 구축할 수 있는데 한계점이 있다. 이에 성능향상을 위해 MS COCO 데이터셋에서 사전 학습된 가중치를 적용하여 전이학습을 진행하였고, 학습률 0.001, Epoch 40,000을 적용하여 학습하였다.

객체탐지에서 사용하는 평가지표인 IoU, mAP, Precision, Recall을 활용한 평가결과이다. Table 3의 학습결과에서 본 연구의 1단계 객체탐지에 적용한 YOLOv4, SAM-YOLOv4 모두 mAP50에서 96 % 이상 높은 성능을 확인할 수 있다. 특히, Attention 모듈을 결합한 SAM-YOLOv4 알고리즘에서 IoU 지표의 향상된 성능을 확인할 수 있으며, 본 연구에서는 합정번호

Table 3. Result of object detection

구분	IoU	mAP50	Precision	Recall
YOLOv4	84.18 %	96.12 %	97.00 %	90.05 %
SAM-YOLOv4	86.73 %	96.19 %	98.00 %	95.00 %

Table 4. Result of each class in SAM-YOLOv4

class	AP	class	AP
DDG	96.43 %	PKG	95.33 %
DDH	95.00 %	LST	100.00 %
FFG	96.45 %	MLS	97.14 %
FF	95.83 %	AOE	96.24 %
PCC	85.33 %	ATS	100.00 %
Hull number Area			96.63 %

호영역의 정보손실을 최소화할 수 있는 높은 IoU 지표를 보인 SAM-YOLOv4 구조가 적합하다.

Table 4에서 각 함형별로 성능(AP)을 살펴보면 LST, MLS, AOE, ATS 등 군함의 외형 특징이 명확하게 구분되는 지원함 함형의 경우에는 높은 성능을 확인할 수 있다. 반면 DDG, DDH, FFG, FF, PKG 등과 같은 전투함의 경우에는 선체 형태, 보유 무장이 비슷한 형



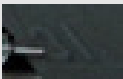

Fig. 5. Result of object detection

태로 보일 수 있어 지원함에 비해 다소 정확도가 떨어진 결과를 알 수 있다. 또한, 함정번호 영역에 대한 AP는 96.63 %로 여러 요소를 포함하고 있는 함정 측면 영상에서 함정번호 숫자만을 포함하고 있는 영역을 높은 성능으로 탐지하고 있다. 1단계 객체탐지의 함형 분류 및 함정번호 영역 탐지 결과 예시는 Fig. 5와 같다.

### 5.2 문자인식 실험결과

문자인식 실험 시에는 제3장에서 언급한 데이터 어그멘테이션을 통해 학습 데이터를 증강하여 2,565장으로 학습하였다. 학습 시 Epoch 2000, 학습률 0.0005를 적용하였다. 문자인식 성능평가 지표로는 예측된 문자

Table 5. Result of text recognition

구분	Conv.	Input image size	예측 문자수	정확도		[예측프레임] 결과 (ex1)		[예측프레임] 결과 (ex2)
#1	VGG Type	32×32	7	77.7 %		[ _ _ _ _ 3 _ 1 ] output 31		[ 9 _ _ _ 9 _ 2 ] output 992
#2		64×32	15	75.5 %		[ _ _ _ _ _ _ _ _ _ _ 3 _ 1 ] output 31		[ 9 _ _ _ _ _ _ _ _ _ _ 9 _ 2 ] output 992
#3	ResNet Type	256×32	8	85.3 %		[ 3 3 _ _ _ _ 1 1 ] output 31		[ 9 _ _ _ _ 9 2 ] output 992
#4	DensenNet Type	64×32	7	86.6 %		[ _ _ _ 3 _ 1 _ ] output 31		[ _ 9 _ 9 _ 2 _ ] output 992
#5		128×32	15	81.7 %		[ 3 _ _ _ _ _ _ _ _ _ 1 ] output 31		[ 9 _ _ _ _ _ _ 9 9 9 2 ] output 992

열과 정답 문자열의 일치 여부를 나타내는 정확도이다. 실험결과는 Table 5와 같다.

문자인식 실험결과는 #4 DenseNet 형태의 문자인식 알고리즘에서 입력이미지값 64(w)×32(h) 일 때 86.6 %의 높은 성능을 달성했음을 확인할 수 있다. 특히 예측 프레임의 각 숫자와 공백(blank)의 배열을 살펴보면, #4의 형태, DenseNet의 특징맵 추출 방식이 본 연구의 문자인식에 가장 적합한 구조를 나타내고 있음을 확인할 수 있다. 또한, 합정번호의 숫자 수를 고려했을 때 예측 프레임의 수가 15개 일 때 보다 7개 일 때 훨씬 성능이 높음을 확인할 수 있다. 이에 본 연구에서 제안하는 모델의 2단계 문자인식 구조에는 #4 DensetNet 형태의 CRNN 구조가 가장 적합하다.

알고리즘의 성능을 고려하여 예측 문자 수 7~8개인 #1, #3, #4의 알고리즘 구조에서 Precision과 Recall 지표를 활용하여 합정번호를 구성하고 있는 숫자별 성능을 분석하였다.(Table 6)

Table 6. Evaluation of each digit

No	#1		#3		#4	
	Precision	Recall	Precision	Recall	Precision	Recall
9	91.13 %	91.87 %	93.54 %	94.30 %	92.25 %	96.75 %
8	87.65 %	85.54 %	93.50 %	84.74 %	95.89 %	86.74 %
7	94.59 %	94.59 %	93.91 %	97.29 %	92.24 %	96.39 %
6	88.88 %	84.21 %	90.16 %	92.98 %	90.00 %	94.73 %
5	95.89 %	94.59 %	94.59 %	94.59 %	95.89 %	94.59 %
3	68.09 %	82.05 %	94.59 %	89.74 %	91.42 %	84.61 %
2	83.78 %	75.60 %	86.84 %	80.48 %	94.87 %	90.24 %
1	88.29 %	89.24 %	93.61 %	94.64 %	91.75 %	95.70 %
0	94.11 %	94.12 %	88.88 %	94.12 %	94.11 %	94.12 %

기존 CRNN 알고리즘인 #1 실험에서는 “8”, “6”, “3”의 숫자에 대한 성능이 다른 숫자에 비해 낮은 것을 확인할 수 있으며, 합성곱 신경망을 변경한 #3, #4 실험에서 해당 숫자에 대한 향상된 성능을 확인할 수 있다. 특히 제안하는 알고리즘인 #4 실험에서 성능 평가 지표는 대체로 높은 성능을 보여주고 있으나 여전히 숫자 “8”, “3”의 경우에는 Recall이 다른 숫자에 비해 상대적으로 낮았다.

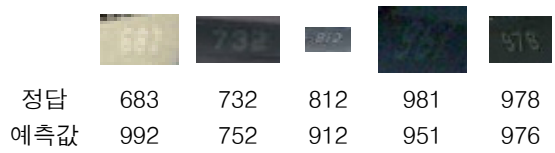


Fig. 6. Sample of incorrect in text recognition

구체적으로 문자인식 알고리즘(#4)이 오답을 예측한 예시를 살펴보면 Fig. 6과 같다. 본 연구에서 학습된 알고리즘을 통해 오답을 예측하는 경우를 분석해보면 합정번호 영역 이미지의 크기가 소형이거나, 선체색상과 합정번호 글자색상의 차이가 크지 않는 경우이다. 특히 숫자 ‘8’, ‘3’에 대해 오류를 다수 확인할 수 있으며, CRNN 구조의 특성상 문자를 연속적으로 학습하기 때문에 다른 합정번호로 예측하는 경우를 확인할 수 있다.

마지막으로 제안하는 모델에서는 문자인식의 알고리즘의 출력 이후 합형에 따른 합정번호 정보를 입력하여 최종 출력할 수 있도록 하였다. 이 결과 문자인식의 정확도는 #4 구조에서 87.5 %로 향상할 수 있었다. 예를 들면 Fig 6의 합정번호 ‘812’는 FFG 합형으로 합정번호는 8로 시작한다. 이때 문자인식의 예측값은 ‘912’로 합형정보 ‘8××’를 추가하여 정확한 합정번호인 ‘812’를 예측할 수 있다. 이는 합정번호 부여 기준에 근거한 것으로 앞서 객체탐지의 높은 신뢰성을 바탕으로 합형정보를 추가함으로써 문자인식의 신뢰도를 향상할 수 있다.

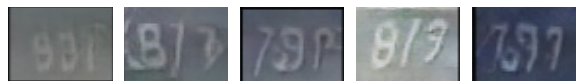


Fig. 7. Result of DCGAN

본 연구에서 제안하는 문자인식의 방법에서 87.5 %의 비교적 높은 성능을 달성하였으나, 보다 향상된 군함 식별 정보를 획득하기 위해 성능향상을 위한 연구가 요구된다. 특히 해상에서는 영상정보를 획득하는 환경과 장비 등 상황에 따라 군함의 선체색상과 합정번호의 색상 차가 변할 수 있다. 그러므로 인식을 향상을 위해서는 다양한 상황에서의 군함 영상정보가 필요한 반면 본 연구가 구축한 데이터셋은 소규모라는 제한점이 있다. 이에 데이터를 증강하기 위해 생성적 적대 신경망 모델 중 DCGAN(Deep Convolution GAN)<sup>[20]</sup> 네트워크를 활용하여 데이터증강 방안을 강구 하였다.



그러나 원천(raw) 데이터의 한계로 DCGAN을 통해 생성된 이미지(Fig. 7)를 학습 데이터로 활용하는 것은 충분하지 못하였다. 향후 데이터셋을 일정 수준 보강함으로써 생성적 적대 신경망을 활용한 데이터 증강이 가능할 것으로 예상되며, 이를 통해 함정번호 인식 성능을 개선할 수 있을 것으로 기대된다.

## 6. 결론 및 발전방향

본 연구는 딥러닝을 적용하여 식별정보를 획득할 수 있는 모델을 처음으로 제안하였다. 해상에서 군함의 영상정보 특성을 고려하여 객체탐지와 문자인식의 알고리즘 2단계 과정을 수행하는 모델을 구현함으로써 각 과정의 특성을 고려하여 정확도 높은 군함정보를 획득할 수 있었다. 본 연구에서는 실시간 객체탐지를 구현할 수 있는 YOLOv4 알고리즘을 적용하여 통해 군함의 함형 분류과 함정번호 영역 탐지에서 높은 성능을 달성하였다. 또한 객체탐지 알고리즘의 단점인 소형 객체탐지의 단점을 해결하고 숫자인식의 정확도를 높이기 위해 합성곱 계층을 변형한 CRNN 알고리즘을 구성하고 함형에 따른 함정번호 정보를 입력하여 함정번호 인식에서 보다 향상된 성능을 달성할 수 있었다.

본 연구의 기여점은 해상에서 획득할 수 있는 이미지를 기반으로 군함의 식별정보를 획득하는 모델을 구현한 최초의 시도이다. 이전에는 군함과 관련해 객체탐지 연구가 이루어졌으나, 본 연구에서 획득한 군함정보의 역할을 고려할 때 실제 해상에서 활용도는 증대될 것이다.

또한, 국방분야에 활용할 수 있는 군함 이미지 관련 공개 데이터셋을 구축하였다. 구축한 데이터는 단순히 상위 범주에서 “군함”이라는 분류하는 것이 아닌 함형별로 데이터셋을 구축하였다. 특히 본 데이터셋은 식별을 위한 군함정보를 획득하기 위해 주로 전방 및 측면 군함 영상으로 구성되어 있다. 본 연구에서 구축된 데이터셋의 특징을 바탕으로 보유 무장에 관한 Segmentation 연구 등 다양한 군함 관련 딥러닝 연구분야에 적극적으로 활용 될 수 있을 것으로 생각된다.

비록 대규모 데이터셋 구축이 제한적인 국방분야에 관한 연구이지만, 전이학습과 데이터증강을 통해 최신 딥러닝 알고리즘을 적용하여 높은 성능을 달성 할 수

있다는 것을 확인하였다. 이는 공개된 자료를 활용하여 향후에도 발전하는 딥러닝을 기술을 빠르게 국방 분야에 적용하여 의미 있는 연구성과를 도출할 수 있을 것으로 기대된다.

본 연구에서 제안하는 모델은 향후 시스템으로 구축하는 것이 요구된다. 제안하는 모델을 통해 함형과 함정번호를 획득할 수 있으나 군함의 자세(방향)로 인해 함형만 획득한 경우, 또는 다중 함정이 영상정보에 포함되어 있어 여러 척의 군함 정보를 획득할 경우 등이 발생한다. 본 모델을 활용하여 시스템을 구축하기 위해서는 객체탐지와 문자인식 간 매커니즘의 추가가 필요하다. 예를 들면 함형 정보만 획득한 경우에는 시스템적으로 해당 군함의 영상정보를 재획득을 지시하는 매커니즘이 필요하며, 여러 척의 군함 정보를 획득한 경우에는 근거리에서 획득된 군함부터 순차적으로 문자인식이 진행될 수 있도록 하는 매커니즘이 필요하다. 본 연구에서는 먼저 우리나라 군함을 대상으로 연구를 진행하였다. 제안하는 모델을 토대로 향후에는 주변국의 군함에 관한 데이터셋을 구축하여 보다 많은 군함의 유형에 대해 정보를 획득할 수 모델로 개선할 수 있을 것으로 기대된다.

## References

- [1] K. Kim, S. Hong, B. Choi, E. Kim, “Probabilistic Ship Detection and Classification Using Deep Learning,” Appl. Sci., Vol. 8, No. 6, p. 936, 2018.
- [2] J. Kim, J. Park, H. Moon, “The Study on the Model for Detection and Classification of Multinational Warship Using Deep Learning,” Journal of the Military Operations Research Society of Korea, Vol. 46, No. 2, pp. 73-83, 2020.
- [3] S. Kim, B. Yoon, “A Study on Composition of Warship Type/Class Identification System,” The Korea Institute of Information and Communication Engineering 2019 Proceeding, Vol. 23, No. 2, pp. 607-610, 2019.
- [4] S. Ren, K. He, R. Girshick, J. Sun, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks,” IEEE transactions on pattern analysis and machine intelligence, Vol. 39, No. 6, pp. 1137-1149, 2016.

- [5] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, Real-Time Object Detection," In Proceedings of the IEEE Conference pp. 779-788, 2016.
- [6] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.Y. Fu, and A. C Berg. "SSD: Single Shot Multibox Detector," In Proceedings of the European Conference on Computer Vision, pp. 21-37, 2016.
- [7] M. Tan, R. Pang, & Q. V. Le, "Efficientdet: Scalable and Efficient Object Detection," In Proceedings of the IEEE/CVF Conference, pp. 10781-10790, 2020.
- [8] A. Bochkovskiy, C. Y. Wang, and H. Y. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," arXiv:2004.10934, 2020.
- [9] B. Shi, X. Bai & C. Yao, "An End-to-End Trainable Neural Network for Image-based Sequence Recognition and its Application to Scene Text Recognition," IEEE Vol. 39, No. 11, pp. 2298-2304, 2016.
- [10] A. Graves, S. Fernandez, F. J. Gomez, and J. Schmidhuber, "Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks," Proc. Int. Conf. Mach. Learn, pp. 369-376, 2006.
- [11] Z. Huang, W. Xu, K. Yu, "Bidirectional LSTM-CRF Models for Sequence Tagging," arXiv:1508.01991, 2015.
- [12] S. Ahn, D. Oh, "2020-2021 ROK Military Weapon Systems," DEFENSE TIMES, pp. 260~263, 2020.
- [13] Choi J. H., Tae Young Han, Seung Hyun Lee, Byung Cheol Song, "Deep Learning-based Small Object Detection," Journal of the Institute of Electronics and Information Engineers, Vol. 55, No. 7, pp. 57~66, 2018.
- [14] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial Pyramid Pooling in Deep Convolutional Networks Forvisual Recognition," IEEE/TPAMI, Vol. 37, No. 9, pp. 1904-1916, 2015.
- [15] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path Aggregation Network for Instance Segmentation," InProceedings of the IEEE/CVPR, pp. 8759-8768, 2018.
- [16] S. Woo, J. Park, Y. Lee and I. Kweon, "CBAM: Convolutional Block Attention Module," In Proceedings of the European Conference on Computer Vision, pp. 3~19, 2018.
- [17] K. Simonyan & A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," arXiv preprint arXiv:1409.1556, 2014.
- [18] K. He, X. Zhang, S. Ren, J. Sun, "Deep Residual Learning for Image Recognition," IEEE/CVPR, pp. 770~778, 2016.
- [19] G. Huang, Z. Liu, K. Q. Weinberger, van der Maaten L., "Densely Connected Convolutional Networks," arXiv:1608.06993, 2016.
- [20] A. Radford, L. Metz, and S. Chintala, "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks," arXiv preprint arXiv:1511.06434, 2015