

# 유도무기 영상의 회전 객체 탐지를 위한 분류 및 회귀방법의 통합적 접근

## A Combined Approach of Classification and Regression for Oriented Object Detection of Missiles

장 석 원 · 백 재 민\* · 강 민 석\*\*

Sukwon Jang · Jae-Min Baek\* · Min-Seok Kang\*\*

### 요 약

본 논문은 회전 바운딩 박스를 통해 딥러닝 기반의 유도무기를 검출하는 방법을 제안한다. 회전 바운딩 박스에 포함된 각도 정보는 클러스터를 필터링하거나 유도무기의 단 분리 시 표적탄을 추적하는데 유용하게 활용될 수 있다. 각도를 예측하기 위해 기존의 높은 탐지 성능과 빠른 실행속도를 보이는 YOLOv5를 기반으로 각도 파라미터를 추가하여 네트워크를 구성하였다. 각도 파라미터 추가로 인한 학습 성능 저하를 줄이기 위해 회귀방법(regression)과 분류방법(classification)이 통합된 방식을 제안하였다. 기존의 회귀방법(regression)만 사용한 방법에 비해 네트워크 헤드(head) 컨볼루션(convolution) 레이어의 파라미터 수를 줄일 수 있어 유도무기 탐지 성능과 각도 예측 성능을 향상시킬 수 있었다.

### Abstract

In this paper, a novel missile detection method based on a rotated bounding box is proposed. The angle information included in the rotated bounding box is employed to reduce the effect of clutter noise or to track the missile's forehead when the separation of the missile proceeds. For angle prediction, YOLOv5 used with additional angle parameters considerably degrades the estimation accuracy and improves the computational complexity. To prevent learning performance degradation owing to additional parameters, an angle prediction approach based on a combination of regression and classification is proposed in this paper. As the number parameters of the last layer of the network head are lower than that when using only regression, the proposed method provides excellent detection performance and leads to a significant reduction of error in terms of angle prediction.

Key words: Convolutional Neural Network, Deep Learning, Supervised Learning, Oriented Object Detection, Artificial Intelligence

### I. 서 론

최근 딥러닝 기반의 객체 탐지 및 추적 알고리즘이 등장하여 좋은 성능을 보임에 따라 국방분야에서도 딥러닝

「이 논문은 2021년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(No. 2021R111A3043152).」

국방과학연구소(Agency for Defense Development)

\*강릉원주대학교 기계공학과(Department of Mechanical Engineering, Gangneung-Wonju National University)

\*\*공주대학교 전기전자제어공학부(Division of Electrical, Electronic & Control Engineering, Kongju National University)

· Manuscript received October 13, 2021, 2021 ; Revised November 1, 2021, 2021 ; Accepted December 3, 2021, 2021. (ID No. 20211013-092)

· Corresponding Author: Min-Seok Kang (e-mail: mskang@kongju.ac.kr)

기반의 감시 및 정찰 시스템의 개발 및 관심이 증대되고 있다. 국방과학연구소에서는 수행하는 유도무기의 비행 시험 시, 유도무기의 비행 영상을 계측하여 여러 이벤트를 육안으로 확인하고, 정성적으로 평가하기 위해 광학 추적 장비를 사용하는데, 최근 이 분야에서도 딥러닝 기반의 객체 탐지 및 추적 알고리즘이 적용된 광학 추적 장비가 개발되고 있다.

하지만 딥러닝 알고리즘의 높은 성능에도 불구하고, 여러 한계가 존재한다. 첫째, 데이터 부족으로 인해 학습된 알고리즘의 성능이 제한적이어서 발사장 주변의 유도무기와 유사한 구조물들에 대한 오 탐지 확률이 높다. 따라서 딥러닝 객체 탐지 알고리즘을 활용하여 추적 알고리즘을 개발할 경우, 원하는 표적인 유도무기를 추적하지 못하고, 발사장의 구조물을 추적하여 추적을 실패할 수 있다. 두 번째로, 단 분리 상황에서 부스터와 모탄의 명확한 구분이 어렵기 때문에 사용자가 원하는 추적 목표물을 추적하는데 제한이 발생한다.

이러한 문제점들은 각도 파라미터를 도입하여 탐지된 표적들 중에서 추적장비 운용자가 원하지 않는 표적을 필터링함으로써 해결될 수 있다. 바운딩 박스와 함께 각도를 예측하는 여러 방법론들이 제안되었다<sup>[1][2]</sup>. 제안된 방법들은 각도 파라미터를 추가한 후 이를 회귀방법으로 해결할 수 있다. 하지만 이러한 접근 방식은 네트워크 헤드 컨볼루션 레이어의 파라미터 수를 크게 증가시켜 네트워크가 학습되기 어렵게 만든다. 유도무기의 경우, 보안상의 이유로 유도무기 영상 데이터 확보에 한계가 있어 데이터가 적기 때문에 네트워크 파라미터의 수가 증가할수록 과적합(overfitting)이 발생되어 성능이 저하될 가능성이 높다<sup>[3][4]</sup>.

따라서 본 논문에서는 각도를 예측하기 위해 각도 파라미터를 회귀방법 문제로만 푸는 것이 아니라, 회귀방법과 분류방법으로 분리하여 푸는 방법을 제안한다. 이 방법은 기존 방식에 비해 네트워크 헤드 컨볼루션 레이어의 파라미터 수를 매우 감소시킬 수 있기 때문에 연산량 측면에서 성능을 개선할 수 있다.

## II. 본 론

유도 무기의 비행속도는 매우 빠르기 때문에 이를 정

확히 추적하기 위해서 효율적인 연산량을 지닌 알고리즘을 사용해야 한다. 따라서 본 논문에서는 추후 추적 알고리즘에서 활용하기 위하여 실시간으로 객체 탐지가 가능하면서 매우 우수한 객체 탐지 성능을 보이는 YOLOv5<sup>[5]</sup>를 기반 알고리즘으로 활용하였다.

### 2-1 YOLOv5

YOLOv5의 구조는 그림 1과 같다. 피쳐(feature)를 추출하는 백본(backbone), 추출된 피쳐를 융합(fusion)하여 성능을 높이는 넥(neck), 피쳐를 바운딩 박스 파라미터로 변환하는 헤드와 같이 3종류 부분으로 이루어져 있다.

백본은 여러 층의 컨볼루션 레이어와 풀링(pooling) 레이어를 통해 입력 이미지로부터 다양한 크기의 피쳐맵을 추출하는 부분이다. Convolution with Batch normalization and Leaky Relu(CBL), Cross Stage Partial(CSP), Spatial Pyramid Pooling(SPP)의 기법이 사용되었다. CBL은 컨볼루션 레이어, 배치 정규화, Leaky-relu의 활성 함수로 이루어진 블록으로 피쳐를 추출하는데 기본적으로 사용되는 블록이다. CSP는 피쳐맵의 일부에만 컨볼루션 연산을 수행하고, 나머지 부분과 통합시키는 방법이다. 일부의 피쳐맵만 컨볼루션 레이어를 통과시키기 때문에 연산량을 줄일 수 있고, 역전과 과정에서 그라디언트(gradient)의 흐름을 효율적으로 수행하게 되어 성능이 개선될 수

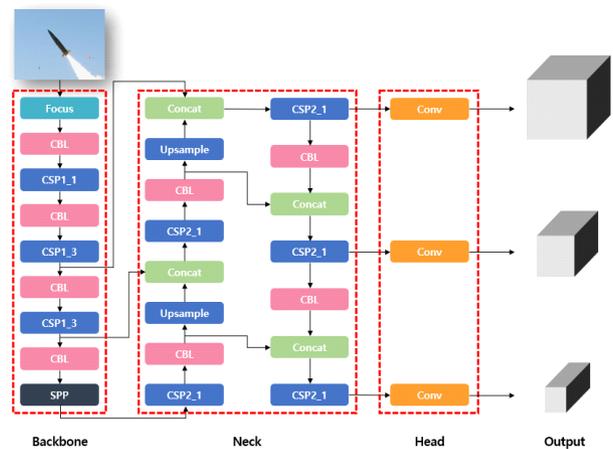


그림 1. YOLOv5 구조  
Fig. 1. The architecture of the YOLOv5.

있다. SSP는 피쳐맵을 다양한 크기의 필터로 풀링한 후 다시 합쳐줌으로써 성능 향상을 가져온다.

넥 부분에서는 다양한 크기의 피쳐맵을 융합하는데 path aggregation network(PAN)을 사용하여 낮은 레벨의 피쳐와 높은 레벨의 피쳐를 섞어주어 성능을 향상시킨다.

헤드는 컨볼루션 레이어를 이용해 넥으로부터 출력된 피쳐를 네트워크의 최종 출력으로 변환해주는 부분으로 바운딩박스 파라미터( $x, y, w, h$ ), 물체가 존재할 확률, 클래스가 존재할 확률로 변환해준다.

## 2-2 제안하는 방법

### 2-2-1 각도 파라미터 설정

지향 객체 탐지(oriented object detection)에서 사용되는 회전 바운딩 박스는 그림 2와 같이  $x, y, w, h, \theta$ 의 5개의 파라미터로 표현되며, 각각 중심 좌표( $x, y$ ), 폭( $w$ ), 높이( $h$ ), 각도( $\theta$ )를 의미한다. 각도는 유도무기의 머리 부분이  $x$ 축과 이루는 각도를 의미하며,  $[-\pi, \pi]$ 의 범위를 갖는다.

### 2-2-2 앵커박스(Anchor Box)

앵커박스는 Faster R-CNN<sup>[6]</sup>서 처음 제안된 후 현재 객체 탐지 분야에서 널리 사용되고 있다. 그림 3과 같이 이미지를 균일한 그리드 셀(grid cell)로 나눈 후에 각 그리드마다 종횡비가 다른 앵커박스를 배치하고, 이 중 정답 바운딩 박스와 가장 겹치는 앵커박스만 선별하여 학습에

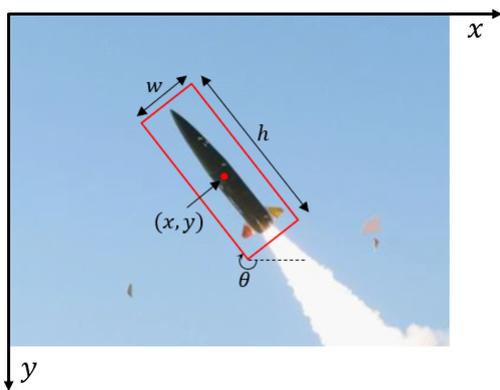


그림 2. 회전 바운딩 박스  
Fig. 2. Rotated bounding box representation.

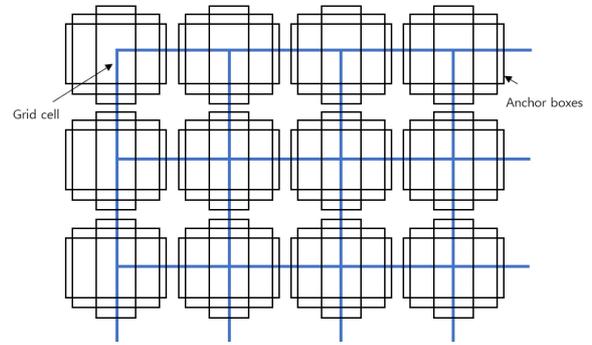


그림 3. 앵커박스의 개념  
Fig. 3. Concept of anchor box.

사용한다. 각 앵커박스는 미리 정해진 바운딩 박스 파라미터( $x, y, w, h$ )를 갖고 있기 때문에 네트워크는 이 앵커박스의 중심점  $x, y$ 를 얼마나 이동시킬지,  $w, h$ 를 얼마나 스케일링(scaling)할지를 학습하게 된다.

### 2-2-3 경계 불연속 문제

박스의 각도를 표현할 때 앵커박스의 각도가 하나로 고정되어 있으면 각도의 주기성 때문에  $0^\circ$  부근에서 경계 불연속 문제가 발생한다. 예를 들어 그림 4와 같이 네트워크가 예측한 각도가  $330^\circ$ 이고, 참값(ground truth)이  $30^\circ$ 일 경우, 평균 제곱 오차(mean square error)를 손실함수로 사용하게 되면 학습이 각도 예측 값을 반대방향으로 회전시켜 실측 자료와 가까워지도록 진행되어  $300^\circ$ 를 회전

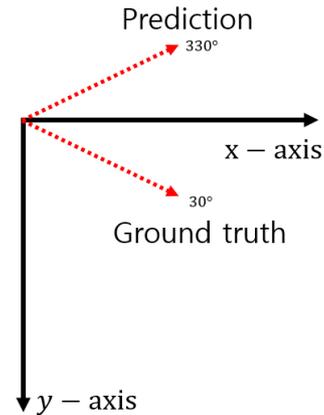


그림 4. 경계 불연속 문제  
Fig. 4. Border discontinuity problem.

하여야 한다. 하지만 각도 예측을 시계방향으로 회전시키면 60°만 회전시켜도 참값에 일치시킬 수 있다.

이처럼 손실함수가 각도 예측을 참값과 일치시키기 위해 필요한 값을 제대로 반영하지 못하는 경계 불연속 문제가 발생하기 때문에 각도에 대한 예측 성능 저하를 불러일으킨다. 이를 해결하기 위해 이전에 제안된 방식들은 그림 5와 같이 다양한 여러 각도를 갖는 앵커박스들을 배치하였다[14].

앵커박스의 각도들은 x축에 대칭이 되도록 정해진다. 앵커박스 각도의 개수가 2개, 4개, 8개일 때 각각의 앵커박스의 각도는 그림 6과 같다.

이와 같이 여러 각도의 앵커박스를 사용하게 되면 경계 불연속 문제를 해결할 수는 있지만, 고정된 각도의 앵커박스만 사용하는 일반적인 객체 탐지(object detection)에 비해 사용되는 앵커박스의 개수가 크게 증가한다. 그로 인해 네트워크 헤드 컨볼루션 레이어의 파라미터 수가 크게 증가하게 되어 네트워크가 학습하기 어려워지고 성능의 저하를 불러일으킨다. 위와 같은 문제는 학습 데이터의 양이 부족할수록 심화된다.

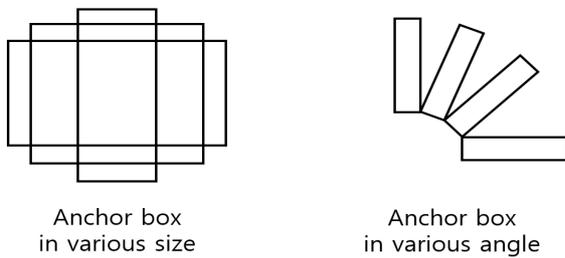


그림 5. 다양한 앵커박스  
Fig. 5. Various anchor box.

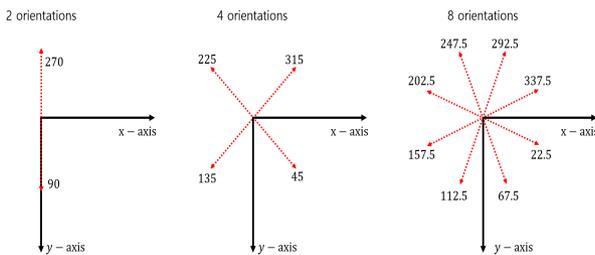


그림 6. 각도분류 개수에 따른 앵커박스의 각도  
Fig. 6. Anchor box depending on the number of orientation.

## 2-2-4 회귀 및 분류방법의 융합을 통한 접근

앞서 언급한 해결하기 위해 본 논문에서는 각도 예측 문제를 회귀방법 문제로만 푸는 것이 아니라, 회귀방법과 분류방법이 결합된 문제로 접근하였다. 그림 7과 같이 분류방법을 통해 각도분류(angle class)를 예측하고, 회귀방법을 통해 세부 각도( $\theta_{reg}$ )를 예측한다. 최종 각도는 식 (1)과 같이 나타낼 수 있다.

$$\hat{j} = \operatorname{argmax}_j (t_{\theta_{cls}}^*(j)), \text{ where } j = 0, 1, \dots, n_0 - 1$$

$$\theta_{cls} = \frac{180}{n_0} + \frac{360}{n_0} \hat{j}, \theta_{reg} = \frac{180}{n_0} (2t_{\theta_{reg}}^* - 1)$$

$$\theta = \theta_{cls} + \theta_{reg} \tag{1}$$

\*이 붙은 파라미터는 네트워크가 이미지를 입력으로 받아 예측한 값들이다.  $t_{\theta_{cls}}^*(j) \in [0, 1]$ ,  $t_{\theta_{reg}}^* \in [0, 1]$ 는 시그모이드(sigmoid)<sup>[7]</sup> 함수에 의해 계산된 네트워크가 예측한 최종 출력이다.  $t_{\theta_{cls}}^*(j)$ 는 바운딩 박스가 각도분류 ( $j = 0, 1, \dots, n_0 - 1$ )에 속할 확률을 의미한다.  $t_{\theta_{reg}}^*$ 는 세부 각도와 관련된 네트워크 출력으로 식(1)의 변환 과정을 거쳐  $\theta_{reg}$ 로 계산된다.  $n_0$ 는 각도분류의 개수를 의미한다.  $\theta_{reg}$ 의 범위는  $\theta_{reg} \in \left[-\frac{180}{n_0}, \frac{180}{n_0}\right]$ 와 같이  $n_0$ 에 의해 정해진다.

YOLOv5는 3개의 헤드를 갖고 있고, 각각은 3개의 다른 크기와 중첩비를 갖는 앵커박스에 대한 예측을 출력으로 내놓고 256, 512, 1,024개의 입력채널 수를 갖는다. 그림 8은 기존방식과 제안한 방식의 최종 출력 파라미터의 구성을 나타낸다.  $n_a (= 3)$ 는 헤드가 갖고 있는 앵커박

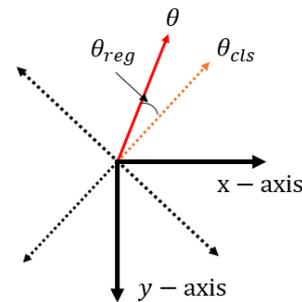


그림 7. 각도 표현 방법  
Fig. 7. Angle representation.

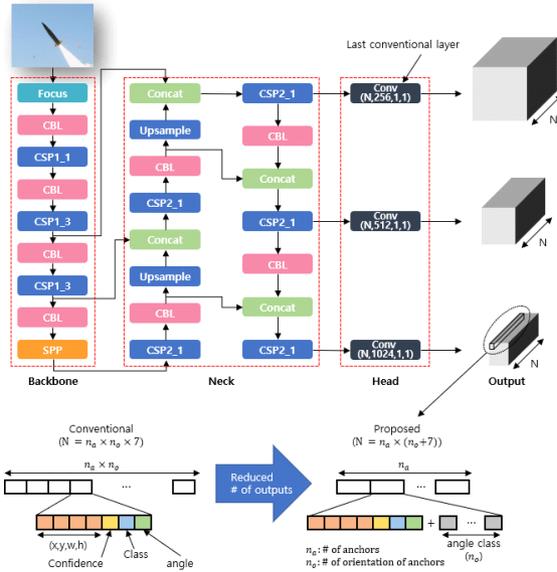


그림 8. 최종 출력 파라미터 비교  
Fig. 8. Comparison between output parameter.

스의 수를 의미한다. 각도에 대해 회귀방법만 사용하는 기존 방식의 경우, 이 각각의 앵커박스에 각도의 개수만큼 곱한 값이 총 앵커박스의 개수가 되어 출력 파라미터의 수( $N=n_a \times n_o \times 7$ )가 크게 증가한다. 하지만 제안한 방식의 경우, 각각의 크기 및 중첩비를 갖는 앵커박스가 한 개씩만 필요하므로 증가되는 출력 파라미터의 수 ( $N=n_a \times (n_o+7)$ )를 줄일 수 있다.  $n_o$ 에 따른 네트워크 출력의 수는 표 1과 같다.

최종 출력의 수가 달라졌기 때문에 넥의 피쳐맵을 최종 출력으로 변환해 주는 헤드의 컨볼루션 레이어의 수 또한 달라져야 한다. 컨볼루션 레이어는 출력 채널 수, 입력 채널 수, 가로 커널 크기, 세로 커널 크기와 같이 4개의

표 1.  $n_o$ 에 따른 네트워크 최종 출력의 파라미터 수  
Table. 1. The number of output parameters.

$n_o$	Conventional	Proposed
2	42	27
4	84	33
8	168	45
16	336	69
32	672	117

표 2.  $n_o$ 에 따른 네트워크 헤드의 파라미터 수  
Table. 2. The number of parameters of network head.

$n_o$	Conventional	Proposed	Ratio
2	75,390	48,460	1.6
4	150,780	59,235	2.5
8	301,560	80,776	3.73
16	603,120	123,855	4.87
32	1,206,240	210,015	5.74

파라미터로 정의될 수 있고, 이 컨볼루션 레이어 파라미터의 수는 ‘출력 채널 수 × 입력 채널 수 × 가로 커널 크기 × 세로 커널 크기 + 출력 채널 수’와 같이 나타낼 수 있다. 각도 파라미터의 개수에 따른 헤드의 파라미터 수는 표 2와 같다.  $n_o$ 가 커질수록 파라미터 수의 차이는 더욱 커진다.

### 2-2-5 네트워크의 손실함수 설정

바운딩 박스와 각도 파라미터들을 예측하기 위해 여러 항들로 이루어진 손실 함수(loss function)를 사용하였다. 네트워크는 그림 9와 같이 앵커박스를 이동시킬 양과 폭과 높이를 조절할 양을 학습한다. 네트워크의 출력을 바운딩 박스로 변환하는 과정은 식 (2)와 같이 표현할 수 있다.  $C_x, C_y$ 는 그리드 셀의 좌표,  $t^*$ 는 네트워크의 출력,  $w_a, h_a$ 는 앵커박스의 폭과 높이,  $b^*$ 는  $t^*$ 와  $w_a, h_a$ 를 이용해 구한 최종 바운딩 박스 파라미터를 의미한다.

$$\begin{aligned} b_x^* &= (t_x^* + C_x) \times w_a, b_y^* = (t_y^* + C_y) \times h_a \\ b_w^* &= t_w^* \times w_a, b_h^* = t_h^* \times h_a \end{aligned} \quad (2)$$

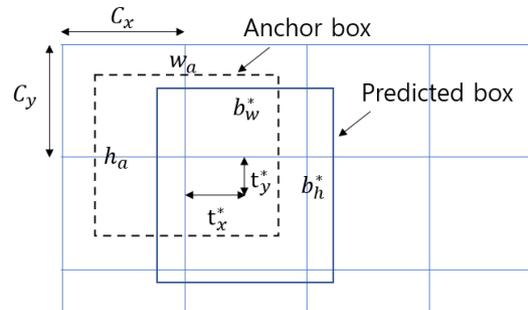


그림 9. 바운딩 박스 표현  
Fig. 9. Bounding box representation.

학습데이터 셋의 참값을 학습하기 위한 파라미터들로 변환하는 과정은 식 (3)과 같다. 참 각도 값인  $\theta_i$ 를 앵커 박스의 각도들과의 제곱오차 비교를 통해 가장 최소가 되는 각도분류( $\hat{i}$ )를 구한다. 이를 원-핫 인코딩(one-hot encoding)을 통해 각도분류 라벨로 변환한다. 세부 각도는 앞서 구한 최소 오차에  $\frac{n_0}{180}$ 를 곱하여 변환하여 손실함수의 입력으로 사용한다.

$$\begin{aligned} \hat{i} &= \underset{i}{\operatorname{argmin}} \left( \left( \theta_i - \frac{180}{n_0} - \frac{360}{n_0} i \right)^2 \right) \\ &, \text{ where } i = 0, 1, \dots, n_0 \\ \theta_{a_i} &= \frac{180}{n_0} + \frac{360}{n_0} \hat{i} \\ t_{\theta_{reg}} &= (\theta_i - \theta_{a_i}) \frac{n_0}{180}, l_{\theta_{cls}} = \text{one-hot}(\hat{i}) \end{aligned} \quad (3)$$

물체가 존재할 확률(objectness score), 클래스 확률(class score), 각도 분류에 대해서는 초점손실(focal loss)을 사용하였고, 세부 각도는 스무딩된 평균절대오차 손실(smooth-L1 loss)를 바운딩 박스 파라미터에 대해서는 IOU 손실(IOU-loss)를 사용하였다. 전체 손실(total loss)은 아래 식과 같이 나타낼 수 있다.  $p$ 는 소프트맥스 함수(softmax function)에 의해 계산된 각 클래스에 속할 확률,  $l$ 은 클래스 라벨,  $b, b^*$ 는 각 네트워크가 예측한 바운딩 박스, 참 바운딩 박스를 나타내는 파라미터( $x, y, w, h$ )를,  $p_{\theta_{cls}}$ 는 소프트맥스 함수에 의해 계산된 각 각도분류에 속할 확률,  $l_{\theta_{cls}}$ 는 각도분류 라벨(angle class label),  $t_{\theta_{cls}}^*$ 는 네트워크가 예측한 세부 각도,  $t_{\theta_{cls}}$ 는 참 세부 각도,  $N_p$ 는 양의 앵커박스(positive anchor box)의 수,  $N_t$ 는 총 앵커박스의 수를 나타낸다.  $k_1, k_2, k_3, k_4$  각 손실함수의 비율을 조절해주는 계수로 반복적인 실험을 통해 얻어지며, 각각 0.5, 0.05, 0.05, 0.05의 값이 사용되었다.

$$\begin{aligned} L_t &= \frac{k_1}{N_t} \sum_i L_{FL}(p(i), l(i)) \\ &+ \frac{k_2}{N_p} \sum_i L_{IOU}(b(i), b^*(i)) \\ &+ \frac{k_3}{N_p} \sum_i L_{FL}(p_{\theta_{cls}}(i), l_{\theta_{cls}}(i)) \\ &+ \frac{k_4}{N_p} \sum_i L_{smoothL1}(t_{\theta_{reg}}(i), t_{\theta_{reg}}^*(i)) \end{aligned} \quad (4)$$

## 2-3 실험

### 2-3-1 학습 데이터셋

제안하는 방법과 기존 방식의 학습 및 평가를 위하여 국방과학연구소 안홍시험장에서 측정한 이미지, Google open image dataset, 인터넷상에서 크롤링을 통해 얻은 이미지를 활용하여 유도무기 학습 데이터 셋 1,800장 과 테스트 데이터 셋 350장을 구성하였다.

### 2-3-2 실험 세부사항

실험의 목적은 기존 방식과 제안한 방식에 대해 각도 분류의 수에 따른 바운딩 박스의 검출 정확도와 각도 파라미터의 예측 정확도를 평가함에 있다. 실험은 i7-7700k, GeForce TITAN RTX, 32GB RAM 사양의 환경에서 이미지 크기 604×604, 배치 사이즈(batch size) 32로 진행하였다. 각 방식에 대해 ImageNet 데이터로 사전 학습된 네트워크를 유도무기 학습데이터 셋으로 전이학습을 진행하였다.

회귀방법만 이용한 기존 방식과 논문에서 제안한 회귀 방법과 분류방법을 이용하는 방식에 대해  $n_o$ 를 1, 2, 4, 8, 16, 32 개로 변경하면서 학습 데이터 셋으로 학습을 수행한 후, 테스트 셋에 대한 검출 성능과 예측한 각도의 정확도가 어떻게 변하는지를 확인하였다.

검출성능을 평가하기 위해선 일반적으로 객체 탐지 분야에서 많이 사용되는 mAP@50(mean average precision)을 활용하였고, 예측한 각도에 대한 정확도는 옳은 예측(true positive)으로 인정된 바운딩 박스들에 대해서만 식 (5)을 통해 각도 에러를 구하고, 이 에러들의 평균절대오차를 활용하여 측정하였다.

$$\text{Average angle error} = \sum_{i \in \text{true positive}} \frac{|\theta_{prediction}^i - \theta_{gt} - 2k\pi|}{N} \quad (5)$$

$k$ 는 에러의 범위가  $[-\pi, \pi]$ 을 만족하도록 해주는 정수이다. 식 (5)와 같은 에러식을 사용하는 이유는 그림 10과 같은 경우 단순히 뺄셈을 이용해 오차를 계산할 경우,  $330 - 30 = 300$ 으로 실제오차 값을 제대로 반영할 수 없는 반면, 식 (5)를 사용할 경우,  $330 - 30 - 360 = -60$ 으로

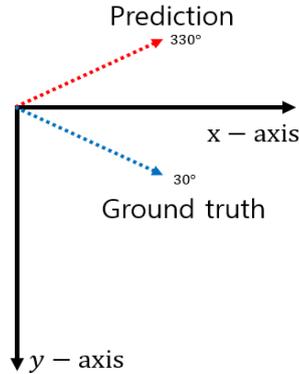


그림 10. 바운딩 박스 표현  
Fig. 10. Bounding box representation.

실제오차를 제대로 반영할 수 있기 때문이다.

### 2-3-3 실험 결과

성능을 분석한 결과는 표 3과 같다.

먼저  $n_o = 1$ 인 경우를 보게 되면 기존 방식과 제안한 방식의 출력 파라미터는 같아진다. 또한 이 경우 2-2-2에서 설명한 경계 불연속 문제가 발생하기 때문에  $n_o > 1$ 인 대부분의 경우들에 비해 검출성능이 낮고 각도 오차가

표 3. 정확도 및 각도 추정 오차 결과  
Table. 3. Result of mAP(mean average precision) and angle error.

$n_o$	mAP@0.5	Angleerror
Conventional		
1	0.76	6.61°
2	0.89	3.58°
4	0.79	2.76°
8	0.80	2.49°
16	0.53	2.23°
32	0.22	2.08°
Proposed		
1	0.76	6.61°
2	0.89	2.88°
4	0.89	2.46°
8	0.91	2.32°
16	0.87	2.41°
32	0.86	2.53°

큰 것을 확인할 수 있다.

기존 방식의 경우, 각도 분류의 수가 증가할수록 검출 성능이 감소하고, 각도 오차가 증가하는 것을 확인할 수 있었다.  $n_o$ 가 증가할수록 여러 각도를 갖는 앵커박스들이 많아지면서 각도 예측 성능이 좋아졌다. 그러나 검출 성능의 경우에는 네트워크 헤드 컨볼루션 레이어 파라미터 수 증가로 인해 하락하는 양상을 보였는데, 특히  $n_o=16,32$ 인 경우에는 과적합을 넘어서 거의 학습이 제대로 수행되지 않아 검출성능이 급격하게 저하되었다.

제안한 방식의 경우에는  $n_o$ 가 증가하면서 검출성능이 향상되고, 각도 에러는 감소하여  $n_o = 8$ 에서 가장 좋은 성능을 보였고, 그 이후로는 검출 성능이 하락하고 각도 오차 또한 커졌다. 이는  $n_o$ 가 너무 커지게 되면 헤드 컨볼루션 레이어 파라미터 수가 증가하여 학습이 효율적으로 이루어지지 않기 때문이다<sup>3),4)</sup>.

전체적으로 제안한 방식이 검출성능과 각도 예측에 있어서 기존방식에 비해 좋은 성능을 보였다. 하지만 기본 방식의  $n_o=16,32$ 인 경우에는 매우 낮은 각도 오차를 보였는데, 이는 학습이 제대로 이루어지지 않아 비교적 바운딩 박스와 각도를 예측하기 쉬운 이미지에 대해서만 옳은 예측을 수행하였고, 그로 인해 매우 적은 수의 예측에 대해서만 오차를 계산하였기 때문이다. 실제  $n_o=16,32$ 인 경우, 전체 정답 중에 춘 비율이 40 %와 0.2 %로 적어도 70 % 이상인 다른 경우들에 비해 매우 적었다.

객체 탐지 알고리즘의 성능을 평가하기 위해서 보편적으로 활용되는 mAP에는 IoU(intersection over union) 문턱값이 존재하고, 주로 0.5 값이 사용된다. 정답과 네트워크가 예측한 바운딩 박스의 IoU가 IoU 문턱 값보다 클 경우 옳은 예측으로 인정한다. 그림 11과 같은 경우에 IoU 문

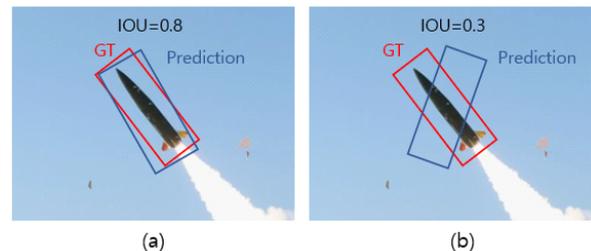


그림 11. 정답과 네트워크가 예측한 바운딩 박스의 IOU  
Fig. 11. IoU between ground truth and prediction.

터 값이 0.5이면 그림 11(a)의 예측 바운딩 박스는 옳은 예측으로 인정되지만, 그림 11(b)의 예측 바운딩 박스는 틀린 예측이다. IoU 문턱 값이 높아질수록 매우 정답과 유사한 바운딩 박스만이 옳은 예측으로 인정된다. 따라서 IoU 문턱 값이 높은 구간에서 mAP가 높다면 더 정밀한 예측을 했다고 볼 수 있다. 일반적으로 IoU 문턱 값이 0.5 일 때와 그 이상일 때의 성능을 많이 보기 때문에 0.5부터 0.9까지의 성능을 비교하였다<sup>[3]</sup>.

표 4의 결과를 보면 제안한 방식의  $n_o = 4, 8$ 인 경우가 가장 좋은 결과들을 보여줬다. 이를 통해 제안한 방식이 기존 방식에 비해 더 정밀한 검출을 수행했다고 볼 수 있다. 또한 앞서 확인한 것처럼 정밀한 예측 또한  $n_o$ 가 커진다고 항상 증가하는 것이 아니라, 너무 커지게 되면 학습이 효과적으로 되지 않아 성능이 저하됨을 알 수 있다.

각도 예측 성능을 조금 더 자세히 분석하기 위해 전체 에러 중 에러 절대 값이  $0^\circ \sim 5^\circ$ 인 경우의 비율을 비교 분석하였다. 전체적으로 제안한 방식이 기존 방식에 비해 좋은 비율을 보였으며,  $n_o$ 가 증가할수록 비율이 좋아지는 양상을 보였다. 하지만  $n_o = 16, 32$ 인 경우 기존 방식에서는 비율이 향상된 반면, 제안한 방식에서는 비율이 하락하였다. 이는 기존 방식에서는 학습이 제대로 수행되지

표 4. IoU 문턱값에 따른 mAP 비교  
Table. 4. Result of mAP on different IOU threshold.

$n_o$	IoU threshold				
	0.5	0.6	0.7	0.8	0.9
Conventional					
2	0.89	0.80	0.60	0.24	0.010
4	0.79	0.69	0.56	0.23	0.019
8	0.80	0.73	0.55	0.23	0.014
16	0.53	0.49	0.38	0.20	0.010
32	0.22	0.23	0.15	0.06	0.007
Proposed					
2	0.89	0.79	0.60	0.26	0.014
4	0.89	<b>0.83</b>	<b>0.62</b>	0.25	0.020
8	<b>0.91</b>	0.80	0.60	<b>0.28</b>	<b>0.021</b>
16	0.87	0.80	0.58	0.26	0.014
32	0.86	0.76	0.57	0.22	0.001

표 5. 각도 에러 절대값의 분포  
Table. 5. Distribution of angle error absolute value.

$n_o$	$0^\circ \sim 5^\circ$	$5^\circ \sim 180^\circ$	Total	$0^\circ \sim 5^\circ / \text{total}$
Conventional				
2	304	87	391	0.78
4	337	51	388	0.87
8	343	50	393	0.87
16	318	39	357	0.89
32	245	31	276	0.89
Proposed				
2	347	65	412	0.84
4	363	49	412	0.88
8	366	45	411	0.89
16	353	49	402	0.88
32	341	56	397	0.86

않아 정답으로 인정되는 수가 확연히 줄어들면서 비교적 예측하기 쉬운 이미지들에 대해서만 오차가 계산되어 비율이 향상되었기 때문이다. 반면 제안한 방식은 네트워크 헤드 컨볼루션 레이어 파라미터 수의 증가로 인해 각도 예측 성능이 저하되면서 비율이 감소하였다(표 5).

### III. 결 론

본 논문은 각도 파라미터를 예측하기 위해 회귀방법과 분류방법을 융합한 방법을 제안했다. 제안한 방법은 네트워크 헤드 컨볼루션 레이어의 파라미터 수를 기존 방식에 비해 줄일 수 있다. 여러 개의 각도 분류에 대해 검출 성능과 각도 예측성능을 분석하였다. 전반적으로 제안된 방식이 기존의 방식에 비해 좋은 성능을 보였고, 각도 분류의 수가 증가하면서 성능이 향상되다가 일정한 값 이후에는 증가된 네트워크 헤드 컨볼루션 레이어 파라미터 수의 영향으로 인해 성능이 하락하였다. 결과적으로 데이터셋의 크기가 작은 유도무기 영상의 회전객체 탐지를 위해 본 논문에서 제안한 방법이 더 적합함을 확인하였다.

### References

[1] T. Tian, Z. Pan, X. Tan, and Z. Chu, "Arbitrary-oriented

inshore ship detection based on multi-scale feature fusion and contextual pooling on rotation region proposals," *Remote Sensing*, vol. 12, no. 2, p. 339, Jan. 2020.

[2] L. Li, Z. Zhou, B. Wang, L. Miao, and H. Zong, "A novel CNN-based method for accurate ship detection in HR optical remote sensing images via rotated bounding box," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 1, pp. 686-699, May 2020.

[3] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, NV, Jun. 2016, pp. 770-778.

[4] J. Kim, S. Shin, C. Jung, and C. Kim, "A deep-learning-based method for recognizing existence of

power-lines in infrared images," *The Journal of Korean Institute of Communications and Information Sciences*, vol. 45, no. 1, pp. 159-162, Jan. 2020.

[5] GitHub, "YOLOV5-master," 2021. Available: <https://github.com/ultralytics/yolov5.git/>

[6] S. Ren, K. He, G. Ross, and S. Jian, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in Neural Information Processing Systems*, vol. 28, pp. 91-99, 2015.

[7] C. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall, "Activation functions: Comparison of trends in practice and research for deep learning," 2018. Available: <https://arxiv.org/abs/1811.03378>

장 석 원 [국방과학연구소/연구원]

<https://orcid.org/0000-0001-7581-7240>



2013년 2월: 경북대학교 전자공학과 (공학사)  
 2016년 2월: 서울대학교 전기정보공학부 (공학석사)  
 2016년 6월~2018년 5월: 한국항공우주산업연구원  
 2018년 5월~현재: 국방과학연구소 연구원

[주 관심분야] 컴퓨터 비전, 인공지능, 객체 탐지 및 추적

강 민 석 [공주대학교/조교수]

<https://orcid.org/0000-0002-7021-6873>



2013년 2월: 아주대학교 전자공학과 (공학사)  
 2015년 2월: 포항공과대학교 전자전기공학과 (공학석사)  
 2019년 2월: 포항공과대학교 전자전기공학과 (공학박사)  
 2019년 2월~2020년 8월: 국방과학연구소

선임연구원  
 2020년 9월~현재: 공주대학교 전기전자제어공학부 조교수  
 [주 관심분야] 레이더 신호 처리 및 영상, 레이더 간섭계, 빔 패턴 합성, 압축센싱 응용 등

백 재 민 [강릉원주대학교/조교수]

<https://orcid.org/0000-0001-9115-9452>



2012년 2월: 고려대학교 기계공학과 (공학사)  
 2018년 2월: 포항공과대학교 IT융합공학과 (공학박사)  
 2018년 5월~2020년 8월: 국방과학연구소 선임연구원  
 2020년 9월~현재: 강릉원주대학교 기계

공학과 조교수  
 [주 관심분야] 비선형 제어, 적응/강인 제어, 정밀 제어, 로봇 제어, 위성/드론 제어, 스펙트럼 추정, 인공지능 등