

## 영상 인식 알고리즘을 이용한 안전 보호구(안전모) 탐지에 관한 연구

노천명<sup>1</sup> · 김기관<sup>1</sup> · 이수봉<sup>2</sup> · 강동훈<sup>1</sup> · 이재철<sup>1†</sup><sup>1</sup>경상대학교 해양시스템공학과, <sup>2</sup>아디아랩

## A Study on Safety Helmet Detection Using Image Recognition Algorithm

Chun-myoung Noh<sup>1</sup>, Ki-Kwan Kim<sup>1</sup>, Su-bong Lee<sup>2</sup>, Dong-hoon Kang<sup>1</sup>, and Jae-chul Lee<sup>1†</sup><sup>1</sup>Dept. of Ocean System Engineering, Gyeongsang Nat'l Univ., Republic of Korea<sup>2</sup>ADIALab, Pusan, Republic of Korea

Received 30 March 2020; received in revised form 1 June 2020; accepted 30 November 2020

## ABSTRACT

Safety accidents at work sites are directly related to workers' lives, and the manufacturing industry's interest in safety accidents is increasing every year. Safety accidents at work sites are caused by a variety of factors, and it is difficult to predict when and why they occur. In this research, an intelligent image recognition-based worker safety protection device wearing algorithm that can determine suitability of wearing safety protective devices is developed and the proposed algorithm is sought to be applied to the site. In this study, the You only look once (YOLO) algorithm is applied to analyze the presence of workers wearing safety protection equipment in real time. Accuracy of object detection for safety protection equipment is very important. Thus, this study compared/analyzed the algorithms of two YOLO systems (YOLOv2, YOLOV3) and improved the performance of the model by changing Hyperparameters, Fine-tuning and Dataset of the selected algorithms. In the future, studies will be conducted on how to improve the accuracy of object detection and complement the accuracy of object detection in the proposed YOLO series algorithm.

**Key Words:** Object detection, Real-time detection, Safety protection

## 1. 서 론

제조 산업의 작업현장에서는 매년 안전사고가 발생하고 있으며, 2015년 이후로 안전사고로 인한 사망자도 800명 이상이 된다(Fig. 1). 작업장에서 안전사고는 작업자의 생명과 직접적인 연관이 있기 때문에 작업장내 안전사고를 예방할 수 있는 방법에 대한 연구는 제조산업에서 중요한 과제 중

하나이다.

안전사고는 다양한 요인(환경적 · 인적 · 기계적 요인)에 의해 발생한다. 환경적 요인으로는 작업 현장 내에 안전성이 확보되지 못하거나 혼잡한 경우 발생하고, 인적 요인은 부적합한 도구 사용이나 안전 보호구를 착용하지 않았을 경우, 마지막으로 기계적 요인으로는 기계가 노후가 되었거나 목적에 맞지 않게 사용되었을 경우에 안전사고가 발생하게 된다. 본 연구에서는 여러 사고 요인들 중 인적 요인에 중점을 두고 연구를 진행했다.

인적 요인으로 인한 안전 사고 예방은 작업현장

<sup>†</sup>Corresponding Author, [j.c.lee@gnu.ac.kr](mailto:j.c.lee@gnu.ac.kr)

©2020 Society for Computational Design and Engineering

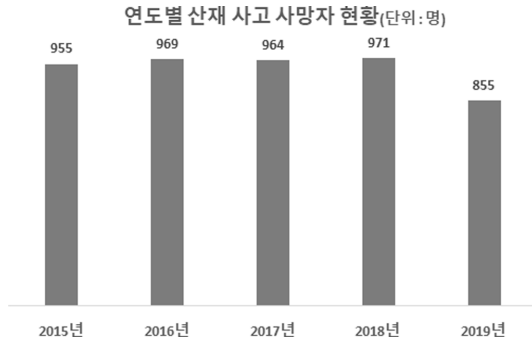


Fig. 1 Status of industrial accidents and deaths by year<sup>[1]</sup>

에서 작업자들이 안전 사고 예방을 위해 작업자의 신체를 보호할 안전 보호구 착용이 우선이다. 안전 보호구는 여러 종류가 존재하고 상황 별 착용하는 보호구가 다르지만 본 연구에서는 작업현장에서 공통적으로 착용하는 헬멧을 안전 보호구로 한정하여 진행한다.

본 연구에서는 객체 탐지를 위한 알고리즘들 중 실시간으로 영상을 처리하여 목표로 하는 객체를 찾을 수 있도록 하는 알고리즘들에 대해 중점을 두고 분석했다. 이후에는 안전 보호구 이미지 데이터를 이용하여 알고리즘 모델에 대해 학습을 진행하고 성능을 비교한다. 그리고 촬영된 동영상에 적용하여 학습한 모델의 객체 인식 성능을 영상으로 파악한다.

본 논문의 연구는 Fig. 2의 순서로 진행했으며, 1장은 서론 연구에 대한 전반적인 내용을 서술한다. 2장은 본론으로 알고리즘과 학습에 이용된 Dataset 그리고 모델 학습 결과를 비교하고, 3장은 결론 및 향후 연구 계획에 대하여 기술하였다.

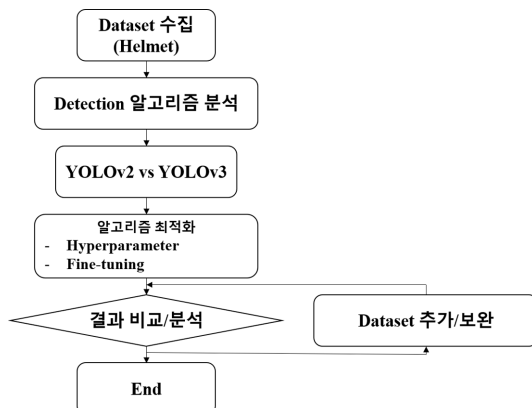


Fig. 2 Research process

## 2. 본 론

### 2.1 Object Detection 알고리즘

이전에도 안전모 탐지에 대해 지속적으로 연구되어 왔다. 객체 탐지를 위한 딥러닝 알고리즘 중 Faster-RCNN<sup>[2]</sup>을 기반으로 하여 안전모를 탐지하거나<sup>[3]</sup>, R-FCN<sup>[4]</sup>을 기반으로 하여 거리, 밝기, 포즈 변화 등에 대한 안전모 탐지<sup>[5]</sup>를 실시한 선행 연구들이 있다. 선행되어진 연구들은 데이터 구축, 여러 상황에 따른 알고리즘 성능에 대해 분석 및 연구를 진행했다.

본 논문에서는 실시간으로 안전 보호구를 탐지하기 위해 객체 탐지에 대한 정확성 보다는 영상 처리 속도에 초점을 두었다.

다양한 알고리즘 중에서 영상 처리 속도에 중점을 둔 대표적인 알고리즘에는 You Only Look Once(YOLO)<sup>[6]</sup>(Fig. 2)와 Single-Shot Detection(SSD)<sup>[7]</sup>(Fig. 3) 두 가지가 존재한다. 두 알고리즘에 대한 간단한 설명은 2.1.1과 2.1.2에 기술한다.

#### 2.1.1 You Only Look Once(YOLO)

YOLO<sup>[6]</sup>는 빠른 영상 처리 속도를 구현함으로써 실시간에 가장 가까운 영상 처리 속도를 보여주고 있는 알고리즘이다.

이미지가 입력되면 이미지를  $S \times S$ 의 사각형으로 나누어 각 사각형 마다 미리 설정된 B개의 영역을 지정하고, 각 영역마다 클래스의 확률을 예측한다.

각각의 영역은 식 (1)로 Confidence Score를 계산하고, 검출하고자 하는 클래스에 대해 식 (2)와 같은 조건부 확률을 가진다. 최종적으로 학습된 모델은 클래스 분류 시, 지정한 임계치(Threshold) 보다 높은 영역들 중 가장 높은 확률을 가진 영역을 최종적으로 클래스로 분류하게 된다.

$$\Pr(Object) \times IOU^{truth}_{pred} \quad (1)$$

$$\Pr(Class_i|Object) \quad (2)$$

그리고 특징추출기로 사용된 Darknet-53<sup>[6]</sup>(Fig. 3)은 YOLOv2버전에서 사용된 Darknet-19<sup>[8]</sup>를 확장한 모델로 총 53개의 층으로 이루어져 있으며, 기존의 Darknet-19에서 Residual Block을 추가했다.

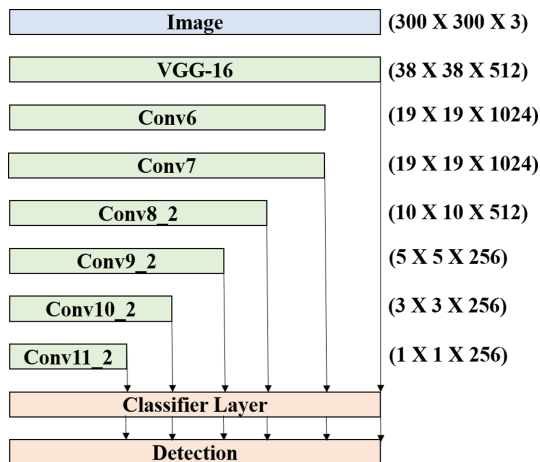
	Type	Filters	Size	Output
1x	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3 / 2	128 × 128
	Convolutional	32	1 × 1	
	Convolutional	64	3 × 3	
2x	Residual			128 × 128
	Convolutional	128	3 × 3 / 2	64 × 64
	Convolutional	64	1 × 1	
	Convolutional	128	3 × 3	
8x	Residual			64 × 64
	Convolutional	256	3 × 3 / 2	32 × 32
	Convolutional	128	1 × 1	
	Convolutional	256	3 × 3	
8x	Residual			32 × 32
	Convolutional	512	3 × 3 / 2	16 × 16
	Convolutional	256	1 × 1	
	Convolutional	512	3 × 3	
4x	Residual			16 × 16
	Convolutional	1024	3 × 3 / 2	8 × 8
	Convolutional	512	1 × 1	
	Convolutional	1024	3 × 3	
4x	Residual			8 × 8
	Avgpool		Global	
	Connected		1000	
	Softmax			

Fig. 3 Darknet-53 Architecture<sup>[6]</sup>

### 2.1.2 Single-Shot Multibox Detector(SSD)

SSD는 YOLO 알고리즘과 함께 탐지 속도가 뛰어난 알고리즘으로, 특징 추출기는 VGG-16<sup>[7]</sup>을 사용한다.

YOLO 알고리즘과 가장 큰 차이점은 여러 개의 Feature map을 각각의 Feature map에 Convolutional layer를 사용하여 Output size를 줄여 여러 스케일의 객체를 탐지하고자 했습니다(Fig. 4).

Fig. 4 SSD Network Architecture<sup>[7]</sup>

본 논문에서는 두 알고리즘 중 Real-time에 가장 근접한 속도를 보여주는 YOLO 알고리즘<sup>[6]</sup>을 이용한다.

## 2.2 Dataset 수집

Dataset은 안전모탐지를 위해 선행 연구된 사진들을 3,884개의 이미지를 수집했다. Fig. 5의 유형과 같은 이미지들을 수집했다. 그리고 이미지들을 Training과 Validation을 위해 각각 70%:30%로 나누어 Training을 진행하였다.

Class는 ‘Person with Helmet, Person without Helmet’ 2가지로 나누어 학습을 진행하였고, 학습을 진행하기 이전에 YOLO에서 Labelling을 위해 제공하는 Yolo-Mark<sup>[9]</sup>라는 프로그램을 사용하였다.



Fig. 5 Initial Dataset Type

## 2.3 Mean Average Precision (mAP)

mAP는 객체 탐지 알고리즘에서 가장 널리 사용되는 모델 성능 척도이다.

AP는 Precision과 Recall 그래프에서 그려진 면적을 계산한 값이다. Precision은 다음 식 (3)으로 표현할 수 있는데,

$$Precision = \frac{TP}{All\ detection} \quad (3)$$

여기서 TP는 검출 결과가 옳게 된 것을 의미한다. 즉 Precision은 검출된 결과 중 옳게 검출된 것을 의미한다.

그리고 Recall은 다음의 식 (4)으로 표현할 수 있다.

$$Recall = \frac{TP}{All\ groundtruths} \quad (4)$$

Recall은 검출이 되어야할 모든 객체들에 대해 옳게 검출한 것을 의미한다.

위 식들을 이용하여 Precision-Recall 그래프가 그려지고, 구해진 AP값들의 평균값을 구한 값이 mAP이다. mAP 이외에도 다양한 모델 성능의 척도가 있지만, YOLO에서 제시한 성능과 비교를 위해 mAP를 이용하여 성능을 분석하였다.

### 3. 사례연구

#### 3.1 YOLO 알고리즘 비교/분석

본 연구에 적합한 모델을 선정하기 위하여 YOLOv2와 YOLOv3 버전의 결과를 비교한다. 두 알고리즘의 성능 판단 기준은 앞서 서술한 Mean Average Precision(mAP)을 사용한다.

본 연구에 더 적합한 알고리즘을 선택하기 위해 YOLO에서 제시한 두 알고리즘의 성능 차이(mAP)를 검증하기 위해 두 알고리즘의 결과를 비교/분석하였다. 두 알고리즘에 대해 동일한 Hyperparameter를 적용하여 Training을 진행하였다. Table 1은 YOLOv2와 YOLOv3의 Training 결과를 보여주고 있다.

Table 1 YOLOv2 vs YOLOv3

	Batch Size	Width/Height	mAP	Loss
YOLOv2	8	608	19.71%	0.8774
YOLOv3	8	608	47.03%	0.5284

YOLOv2와 YOLOv3의 결과는 각각 19.71%와 47.03%로 결과 차이가 매우 큰 것을 볼 수 있다.

#### 3.2 YOLOv3 알고리즘 최적화

Table 1에서 비교한 YOLOv3의 성능은 YOLOv2 보다는 높은 결과를 보여주고 있지만, 좋은 결과를 보여주지 않았다. 이후 Hyperparameter 조정과 Fine-tuning, Dataset을 변경을 통해 성능을 향상시키고자 하였다.

##### 3.2.1 Hyperparameter 수정

수정을 실시한 Hyperparameter는 Table 2를 보면 알 수 있다. Training을 1번 진행할 때 사용되는 이미지 수를 나타내는 Batch Size를 8, 64로 두었다. 이는 컴퓨터의 성능에 따라 차이를 둘 수 있

다. 컴퓨터의 성능이 우수할수록 큰 숫자의 Batch Size를 설정하여 매 학습마다 많은 수의 데이터를 처리할 수 있다.

그리고 Training시에 입력되는 이미지의 크기를 조정하는 Image Size는 '416×416'과 '608×608'으로 나누었는데, YOLO에서 제시한 성능표에서 두 가지 Image Size에 대해서 성능차이를 보이고 있기에 두가지 조건으로 설정하였고 학습 시에 Padding(입력 데이터의 주변을 특정 값으로 채워 늘리는 것)을 진행해야 하기에 학습과정이 번거로워지는 단점이 존재하여 이미지의 가로와 세로 값을 동일하게 설정하였다. 최종적으로 4가지의 경우를 두고 Training을 진행하였다.

Hyperparameter의 경우 마다 Training 결과는 Table에서 볼 수 있다. 먼저 Batch Size를 '8'로 두었을 때 Image Size를 달리 두었다. 그리고 각각의 경우에 대한 결과는 이미지 사이즈가 '608×608'일 때 mAP가 더 높게 나타났다. 다음으로 Batch Size를 '64'로 두었을 때 이미지 사이즈 별 Training을 진행하였고, 여기서도 마찬가지로 이미지사이즈가 '608×608' 일 때 더 좋은 성능을 나타내고 있다.

Table 3의 결과들을 토대로 Batch Size와 Image Size가 클수록 더 좋은 결과를 나타내고 있다. 하지만 가장 좋은 성능을 나타낸 경우에도 mAP가 약 57%로 낮은 성능을 나타내고 있다.

Table 2 Hyperparameter

Batch size	8
	64
Image size	416×416
	608×608

Table 3 Results by Hyperparameter

Batch Size	Width/Height	mAP	Loss
8	416	38.52%	0.5749
8	608	47.03%	0.5284
64	416	47.05%	0.2613
64	608	56.68%	0.2656

##### 3.2.2 Fine-tuning 적용

mAP를 향상 더 향상시키기 위해 Fine-tuning 기법을 적용하고자 한다. Fine-tuning은 Transfer learning에 사용될 Dataset의 크기와 Pre-trained된

모델에 사용된 Dataset과의 유사도에 따라 Fine-tuning할 모델의 범위를 설정된다. 범위 설정 시에는 4가지 경우가 있다.

- 데이터의 양이 많고 유사도가 클 때
- 데이터의 양이 많고 유사도가 적을 때
- 데이터의 양이 적고 유사도가 클 때
- 데이터의 양이 적고 유사도가 적을 때

첫번째 상황에서는 데이터의 양과 유사도에서 기존의 모델과 차이가 없기에 범위를 설정하는데 어려움이 없다. 두번째 상황은 기존의 데이터와 유사도가 적기에 모델 전체에 대해 학습을 진행하는 것이 좋다. 세번째는 Dataset의 유사도가 크기 때문에 데이터의 양이 적어도 모델의 학습 범위를 작게 설정해서 학습을 하는 것이 좋다. 네번째 상황에서는 기존 Dataset과의 유사도가 떨어지는데 학습할 Dataset의 양도 적기에 범위 설정이 매우 어렵고, 모델의 범위 설정 학습 후 성능이 떨어질 수 있다.

본 연구에서는 학습에 사용된 데이터의 양이 충분하지 못하기 때문에 세번째 상황과 유사하다고 판단하고, Pre-trained 한 YOLOv3의 가중치를 사용하여 YOLO 네트워크 중 81번 레이어 이후의 레이어들에 Fine-tuning을 적용하여 training을 진행하였다.

Fine-tuning한 모델의 학습에 Hyperparameter는 앞선 결과들 중에서 가장 결과가 좋았던 Batch Size가 64이고 이미지의 사이즈가 608일 때의 Hyperparameter를 적용하여 iteration 10000번 학습을 진행한다. Fine-tuning 한 모델의 성능 결과는 Table. 4에 나타나 있는데, 모델 학습의 결과는 mAP가 55.03%이다.

Table 4 Result by Fine-tuning

Batch Size	Width/Height	mAP	Loss
64	608	55.03	0.2713

### 3.2.3 Dataset 보완

Hyperparameter 조정과 Fine-tuning을 통한 모델의 성능을 향상시키고자 했지만, 약 57%라는 낮은 성능이 나타났다.

이후 성능 향상을 위해 다양한 방면으로 모델 성능의 저하에 대한 이유를 분석하였고, 모델 학

습의 기초가 되는 Dataset 선택과 Labelling에 대해 초점을 두고 분석을 진행하였다.

앞서 진행한 Dataset은 Fig.5에서 볼 수 있듯이 여러 사람들이 안전보호구를 착용하고 있는 모습에 대해 Training을 진행하였다. 그래서 안전 보호구에 대해 이미지 내에 찾고자 하는 안전모만 존재하는 Fig. 6와 같은 이미지들을 수집하여 새롭게 Training을 진행하였고, 결과는 Table. 5과 같이 나타났다. Training은 클래스를 'Helmet'으로 단일 클래스로 진행하고 이전에 사용한 Hyperparameter를 사용하여 Darknet-53에 진행하였다.



Fig. 6 Changed Dataset Type

Table 5 Result of Changed Dataset

Batch Size	Width/Height	mAP	Loss
64	608	97.2	0.0517

결과는 이전에 진행되었던 Training들 보다 훨씬 뛰어난 97.2%라는 결과를 보여주었다.

### 3.3 결과 비교 분석

5가지(Hyperparameter: 4, Fine-tuning: 1)의 학습 파라미터를 기반으로 YOLOv3모델의 결과를 비교했다(Table 3-4). 먼저 Hyperparameter를 조정하여 얻어낸 4가지 경우의 결과를 보면, Batch Size가 크고 이미지 크기가 클 수록 좋은 결과를 나타낸다는 것을 확인할 수 있었다(Table 3). 그리고 성능 향상을 위해 Fine-tuning을 진행했지만, 기대한 성능 향상의 효과는 나타나지 않았다(Table 5). 5가지의 결과 중 가장 좋은 성능을 나타낸 56.68%의 mAP를 보여준 모델이 YOLOv3에서 제시한 COCO Dataset<sup>[10]</sup>에 대하여 모델을 학습하고 나타난 mAP가 57.9%인 경우와 성능이 비슷하다<sup>[6]</sup>.



이후 Dataset을 이미지내에 찾고자 하는 객체만 존재하는 이미지들로 변경하였고, 클래스를 단일 클래스로 변경하여 새롭게 Training을 진행하였다. 최종적인 결과는 97.2%로 이전의 결과들 보다 큰 폭으로 향상되었다.

Fig. 7은 앞서 Training한 모델을 가져와 이미지에 대해 적용하여 결과를 추출하였다. 사용된 이미지는 Fig. 7을 보면 이미지 내에서 안전모를 잘 찾아내고 있지만, Fig. 8을 보면 안전모의 각도가 달라졌을 때 찾지 못한 결과가 나타났다.

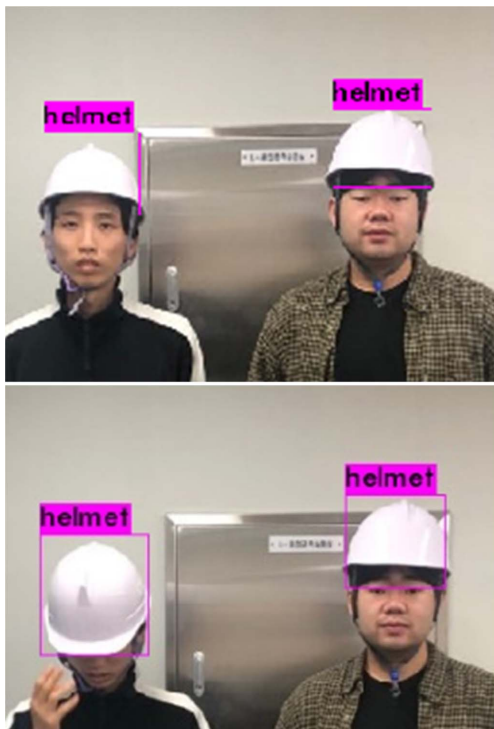


Fig. 7 Results of Safety Helmet Detection



Fig. 8 Error of Safety Helmet Detection

#### 4. 결론 및 향후 연구

작업 현장에서 안전사고는 여러 요인에 의해 발생한다. 그리고 안전사고 예방을 위해 많은 방법들이 연구되고 지속적으로 노력을 기울이고 있다.

본 연구에서는 작업현장에서 작업자들의 안전사고를 예방하기 위해 작업자들의 안전 보호구의 착용 여부를 영상 인식 알고리즘을 적용하여 실시간으로 판단하고자 했다. 영상 인식과 관련된 알고리즘들을 찾아보고, 알고리즘의 특성을 분석했다. 그리고 연구의 목적과 부합하는 YOLO 알고리즘을 찾아 연구에 접목시켰다.

YOLO 알고리즘은 Real-time에 가장 근접하여 Object Detection 문제를 해결할 수 있는 알고리즘으로, YOLO 알고리즘의 3가지 버전 중 2개의 버전에 대해서 모델 학습을 진행하고 성능을 파악하고 비교했다.

YOLOv2와 YOLOv3를 비교하여 YOLOv2의 성능이 YOLOv3와 비교하여 모델을 검증하고, 본 연구에 적합한 모델을 찾아보았다. 이후 성능이 더 뛰어난 YOLOv3의 Hyperparameter를 조정하여 이미지의 크기가 크고 Batch Size가 클수록 좋은 성능이 나타난다는 결과를 얻었다. 그리고 Fine-tuning을 통해 성능 향상을 기대했지만 원하는 결과를 얻지 못했다.

이후 다양한 방면으로 YOLO 알고리즘에 대해 분석을 진행하였고, Dataset을 찾고자 하는 객체만 존재하는 이미지들로 구성하여, 단일 클래스로 Training을 진행하였다. 그리고 결과는 이전에 진행되었던 Training의 결과들보다 큰 폭으로 향상된 97.2%라는 결과를 보였고 Dataset이 모델을 학습할 때 큰 영향을 끼친다는 결과를 얻어냈다. 하지만 Dataset의 다양성 부족으로 안전모를 탐지해 내지 못한 경우도 있었다.

향후 연구에서 모델의 학습에 사용된 Dataset에 대한 다양성을 증가시켜 여러 조건에서도 객체 탐지를 할 수 있도록 하고, 여러 안전 보호구들에 대해서도 모델 학습을 진행하여 실제 현장에서의 적용 가능성에 대해 연구를 진행할 것이다.

#### 감사의 글

이 연구는 ‘산업전문인력역량강화사업’ (2020년 친환경·스마트선박R&D전문인력 양성사업, 과제

번호: P0001968), ‘광역협력권산업육성사업’(P0004736)의 재원으로 한국산업기술진흥원(KIAT)과 한국연구재단 ‘지역대학우수과학자지원사업’(2017R1D1A3B03028250)의 지원과 과학기술정보통신부와 정보통신사업진흥원이 지원하는 경남 SW융합클러스터 2.0 특화산업 강화사업(S0312-20-1002) 받아 수행된 연구임.

## References

1. Kosha, <http://www.kosha.or.kr/kosha>
2. Ren, S., He, K., Girshick, R. and Sun, J., 2015, Faster R-CNN, IEEE Int. Conf. Comput. Vision, pp.1440-1448.
3. Kim, D., Kong, J., Lim, J. and Sho, B., 2020, A study on Data Collection and Object Detection using Faster R-CNN for Application to Construction Site Safety, *KOSHAM*, 20(1), pp.119-126.
4. Dai, J., Li, Y., He, K. and Sun, J., 2016, R-fcn: Object Detection via Region-based Fully Convolutional Networks, *Advances in Neural Information Processing System*, pp.379-387.
5. Park, S., Yoon, S. and Heo, J., 2019, Image-Based Automatic Detection of Construction Helmets using R-FCN and Transfer Learning, *Journal of the Korean Society of Civil Engineers*, 39(3), pp.399-407.
6. Redmon, J., Divvala, S., Girshick, R. and Farhadi, A., 2015, You only Look Once: Unified, Real-time Object Detection, Proc. the IEEE Conf. on Comp. Vision and Pattern Recognition, pp.779-788.
7. Liu, W., *et al.*, 2016, SSD: Single Shot Multibox Detector, in Proc. ECCV, pp.21-37.
8. Redmon, J. and Farhadi, A., 2017, YOLO9000: Better, Faster, Stronger. In CVPR, pp.7263-7271.
9. Yolo-Mark Manual, [http://github.com/AlexeyAB/Yolo\\_mark](http://github.com/AlexeyAB/Yolo_mark)
10. Lin, T.Y., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., Perona, P., Ramanan, D., Zitnick, C.L. and Dollar, P., 2014, Microsoft COCO: Common Objects in Context, arXiv: 1405.0312.



**노 천 명**

2013년~2019년 경상대학교 조선해양공학과 학사  
2019년~현재 경상대학교 해양시스템공학과 석사과정  
관심분야: Artificial Intelligence, Computer Vision  
이메일: n941114@gmail.com



**김 기 관**

2014년~2020년 경상대학교 조선해양공학과 학사  
2020년~ 경상대학교 해양시스템공과 석사  
관심분야: Deep Learning, Object Detection  
이메일: kikwan00@gmail.com



**이 수 봉**

2013년~2020년 동의대학교 컴퓨터과학과 학사  
2020년~현재 (주)아디아랩 연구원  
관심분야: Deep Learnig, Computer vision  
이메일: subong.lee@adialab.com



**강 동 훈**

1992년~1999년 부경대학교 조선해양시스템공학과 학사  
2001년~2003년 부산대학교 조선해양공학과 석사  
2004년~2007년 일본 오사카 대학 선박해양공학과 박사  
2009년~현재 경상대학교 조선해양공학과 교수  
관심분야: System performance, Hydrodynamic  
이메일: kangdonghoon@gnu.ac.kr



**이 재 철**

2002년~2009년 울산대학교 조선  
공학과 학사

2009년~2011년 부산대학교 조선  
해양공학과 석사

2011년~2015년 부산대학교 조선  
해양공학과 박사

2016년~현재 경상대학교 조선해양  
공학과 부교수

관심분야: Artificial Intelligence,  
Green & Smart Ship

이메일: j.c.lee@gnu.ac.kr

---