

결함검출 적용을 위한 YOLO 딥러닝 알고리즘 비교

김익수^a, 이문구^a, 전용호^{a*}

Comparative Analysis of Defect Detection Using YOLO of Deep Learning

Ik-Su Kim^a, Moon Gu Lee^a, Yongho Jeon^{a*}^a Department of Mechanical Engineering, Ajou University

ARTICLE INFO

Article history:

Received	27	November	2021
Revised	1	December	2021
Accepted	4	December	2021

Keywords:

YOLO
Deep learning
Object detection
Defect detection
CNN

ABSTRACT

Recently, metal 3D printing technology has developed and has been widely applied in fields such as mechanical parts and construction sites. However, the problem of output defects must be resolved. These defects appear as pores and microcracks in the output, which can be confirmed through microscopic analysis of the output. In addition, if the understanding of pores or cracks is unclear or many images need to be checked in a short time, an error might occur. Therefore, this study aims to develop a precision object detection algorithm using deep learning. The purpose is to automatically detect defects using deep learning-based You Only Look Once (YOLO). Through comparison using YOLO v3 and v5 algorithms, the accuracy and speed were compared to analyze which YOLO model was efficient in the defect detection process.

1. 서론

금속 3D 프린팅, 용사 공정, 용접 공정, 레이저 클래딩 공정 등의 과정에서 내부의 결함에 대한 문제는 지속적으로 논의되어 왔다^[1]. 또한 결함의 유무, 결함의 정도를 파악하는 것은 결과물에 대한 신뢰도, 제품의 품질 및 물성 저하, 공정변수 등 많은 문제점에 대해 예방 및 해결 할 수 있으며 결함의 유무와 정도를 파악하지 못 할 경우 제품의 품질, 내구, 성능에 직접적인 영향을 미치기 때문에 매우 중요하다. 그래서 이러한 결함의 검출 및 검수를 사람이 직접 수행한다면 상대적으로 높은 인건비의 지출이 예상되며, 단시간에 다수의 결함을 확인하여야 하거나 방대한 양의 데이터를 확인하여야 할 경우 오류가 발생할 수 있는 문제점을 안고 있다. 또한 생산성 향상을 위한 공장자동화 시스템이 각광받고 있고^[2] 스마트 팩토리가 확산됨에 따라서 딥러닝을 활용한 정밀 객체 검출 알고리즘의 선택 및 적용에 관한 연구가 제안되고 있다.

특히, 현재 다수의 객체 검출 알고리즘이 개발되어 사용되고 있지만, 데이터 처리 속도가 빠르고 정확도가 높으며 널리 사용되고 있기 때문에 신뢰도가 높은 딥러닝 기반의 YOLO (you only look once)^[3]를 이용하여 결함을 자동적으로 검출하는 연구들이 진행되고 있다^[4].

자동 객체 검출을 이용한다면 공정 이후의 현미경 및 CT 분석의 진행 시 출력물 내부의 기공 또는 미세 균열을 별다른 조작없이 자동으로 검출할 수 있기 때문에 기존에 사람이 수행했던 결함 검출보다 빠른 속도로 많은 양의 작업을 수행할 수 있으며 비용적 측면에 대해서도 긍정적 결과를 기대할 수 있다^[5]. 결과적으로 전체적인 공정 과정을 보았을 때 필요한 시간과 비용이 절감되며 효율적인 작업으로의 발전을 기대할 수 있다.

본 연구에서는 RTX 3080의 PC 환경에서는 더 이상 릴리즈가 되고 있지 않는 YOLOv4^[6]를 제외한 비교적 최신 객체 검출 알고리즘인 Pytorch 기반 YOLOv3, v5의 알고리즘 별 비교를 통해

* Corresponding author. Tel.: +82-31-219-3652

E-mail address: princaps@ajou.ac.kr (Yongho Jeon).

어떠한 알고리즘이 더 많은 양의 이미지를 처리할 수 있는지, 기공이나 미세 균열의 크기에 따라 결함의 검출 능력, 정확도, 속도를 비교하여 어떠한 알고리즘이 효율적인지를 분석하였다. 비교 분석에 필요한 데이터 전처리를 하기 위해 200개의 이미지를 준비했고 이미지 사이즈는 416×416 으로 동일한 뒤 data augmentation을 통해 5,982장으로 데이터를 증강시켰다. 그중 70%는 트레이닝 이미지로 20%는 검증 이미지, 마지막으로 10%는 테스트 이미지로 검증 이미지를 사용했다. 이를 통해, 각각의 알고리즘을 적용하여 분석하고 비교하였다.

2. Object detection

2.1 연구 방법

본 연구에서는 공정간 객체 결함을 자동으로 인식 및 분류하는 방법을 제안하는 알고리즘을 버전 별로 비교 분석하고 이에 따른 결과를 제시하고자 한다. 결함 검출 알고리즘으로는 convolution neural network (CNN)^[7] 기반 최신 알고리즘인 YOLOv3와 YOLOv5를 사용했고 데이터 셋은 레이저 클래딩 공정, 용접 공정, 용사 공정 중 현미경 촬영을 이용해 발견된 미세 균열과 기공의 이미지를 준비했다. Over fitting의 방지와 정확도의 증가를 위해 데이터 증강 기법인 data augmentation을 이용하였고, 5,982개의 이미지로 증강시켜 결함 발생 여부의 기본 데이터로 준비했다. 각 알고리즘의 비교 분석은 precision, mAP, recall, 검출 속도를 기준으로 비교하고 특히, 주요 결과인 precision과 검출 속도를 비교하는데 초점을 맞춰 비교하고자 한다.

이미지를 분류하기 위해서 사용되는 일반적인 합성곱 신경망(convolution neural network, CNN)의 구조는 convolution layer와 pooling layer의 다양한 조합으로 구성된다. 합성곱 신경망(CNN)은 필터링 기법을 인공신경망에 적용하여 시각적 이미지 혹은 동영상보다 효과적으로 처리하기 위해 고안되었고 이후 현재의 딥러닝에서 사용되고 있는 형태의 CNN이 제안되었다. 기존의 필터링 기법은 고정되어 있는 필터를 이용하여 이미지를 처리했지만 CNN의 기본 개념은 행렬로 표현된 필터의 각 요소가 데이터 처리에 적합하도록 학습을 자동화하는 것이다. 기존의 인공신경망은 affine으로 명시된 fully-connected 연산과 ReLU와 같은 비선형 활성화 함수의 합성으로 정의된 계층을 쌓는 구조이지만 CNN은 합성곱 계층과 풀링 계층이라는 새로운 층을 fully-connected 계층 이전에 추가함으로써 원본 이미지에 필터링 기법을 적용한 뒤에 필터링 된 이미지에 대해 분류 연산이 수행되도록 구성된다. 본 연구에 적용된 알고리즘은 CNN을 기반으로 한 YOLO로, 2016년에 버전 1을 공개한 비교적 최신 딥러닝 기반 object detection 알고리즘이며, one-stage detector의 대표 모델

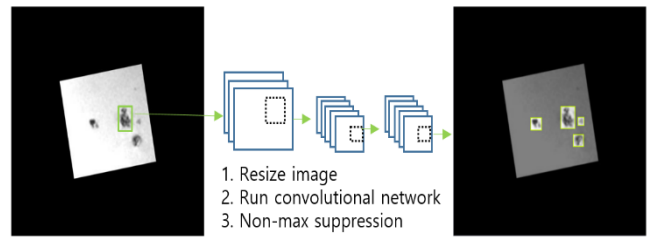


Fig. 1 The YOLO detection system

이다. One-stage detector란 이름과 같이 이미지를 한 번만 보고 convolution 층의 grid cell 별로 찾고자 하는 물체가 있을 확률과 해당 grid cell이 어떤 클래스인지 확률을 계산하며 classification과 localization을 동시에 수행해 객체를 탐지하는 방법이다. 이러한 방법을 통해 정확도를 보다 중요하게 생각하는 two-stage detector 대비 속도를 향상시킬 수 있다^[8].

이러한 원리를 활용한 YOLO의 큰 장점은 하나의 회귀 문제(single regression problem)의 계산이 한 번에 가능하다는 것이다. YOLO는 사람의 눈으로 특정 환경을 봤을 때 찾고자 하는 물체를 한 번에 파악할 수 있는 것과 동일한 형태로 구성되어 있다. 예를 들어 사람은 Fig. 1의 이미지를 보았을 때 일련의 사고를 거치지 않고도 어디에 어떤 결함이 있는지를 쉽게 인지할 수 있다. 이와 같이 YOLO의 구조는 다른 객체 검출이 하고 있는 슬라이딩 윈도우 방식이나 region proposal의 복잡한 방법 대신 하나의 파이프라인을 활용하여 결과를 빠르게 검출하는 특징이 있다.

YOLO의 구조는 하나의 컨볼루션 네트워크가 이미지 내에서 찾고자 하는 객체의 bounding box의 위치 정보인 x, y, w, h의 값과 bounding box의 class 확률을 동시에 계산해준다. 이러한 방식의 장점으로는 첫째, 기존의 검출과정을 하나의 회귀 문제로 바꾸었기 때문에 검출 속도가 빠르다. 실 예로, YOLO의 경우 1초에 45프레임, Fast-YOLO의 경우 1초에 155 프레임을 처리할 수 있으며, 이는 실시간 객체 탐지가 가능함을 의미한다. 둘째, YOLO는 객체 검출 시 이미지 전체를 보기 때문에 클래스의 모양 및 주변 정보까지 학습한다. 그러나, YOLO 이전의 객체 검출 모델 중 가장 유명한 모델인 Fast R-CNN^[9]은 주변 정보까지 처리하지 못하기 때문에 background error가 생긴다. 그에 반해 YOLO는 이미지 전체를 학습하게 되며 background error가 생길 확률이 적어진다. 마지막으로 YOLO는 찾고자 하는 물체의 일반적인 특징을 학습하기 때문에 이미지를 학습하여 테스트할 때 불특정한 객체의 오류가 있다고 하더라도 뛰어난 성능을 보여준다.

2.2 Data set

데이터셋은 용접, 레이저 클래딩, 용사 공정 과정에서 나온 미세

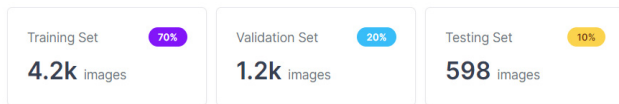


Fig. 2 Data set composition

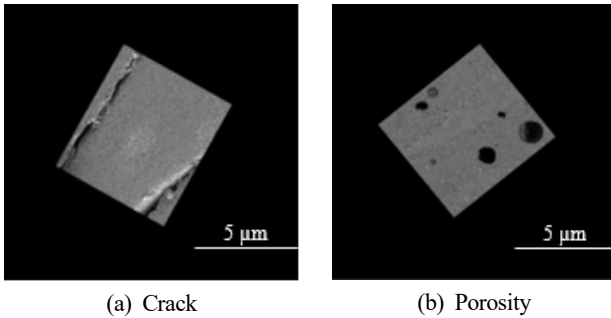


Fig. 3 Data set

균열, 기공을 현미경으로 촬영한 200개의 데이터 셋으로, Fig. 2와 같다. 결함에 대해서는 labeling 툴을 사용하여 준비하였으며, 단일 이미지의 크기는 416×416 이다. 또한 data augmentation을 이용하여 총 데이터를 5,982개로 증강시켰다. data augmentation에 적용된 방법은 flip과 5° 씩 회전하는 rotation 기법을 이용했다. 결과 도출 과정에서 객체 검출을 제대로 했는지에 대해 검사하며 정밀도와 직접적인 관계가 있는 data split을 위해 Roboflow 사이트에서 제공하고 있는 data separate를 이용하여 Fig. 2와 같이 training set 4,200개 validation set 1,200개 testing set 598개로 분류하였다. 분석을 위한 classes는 Fig. 3과 같이 주요한 품질인자인 미세균열과 기공으로 설정하였다.

2.3 YOLOv3

본 연구에서 중점적으로 비교하고자 하는 YOLOv3^[10] 알고리즘은 YOLOv2와 비교했을 때 성능면에서 정확도와 GPU 사용성이 향상되었지만 그만큼 가중치가 커졌기 때문에 속도 즉 FPS 관점에서는 성능이 저감되었다고 판단할 수 있다. 그러나 여전히 다른 객체 검출 알고리즘보다는 상대적으로 빠른 속도로 객체 검출을 할 수 있기 때문에 향상된 버전이라고 볼 수 있다. Table 1에 여러 object detection 알고리즘에 대한 선행 연구 결과들을 정리하였다. mAP의 경우 YOLOv3-608이 FPN FRCN에 비해 1.2% 가량 낮지만 time이 약 3배 빠르기 때문에 실시간성을 고려하는 본 연구의 취지에 YOLOv3 알고리즘이 Table 1의 알고리즘 가운데 가장 강력한 모델이라고 할 수 있다. YOLOv3는 darknet-53으로 53개의 convolution layer를 사용하고 있으며 Table 1의 성능 지표를 봤을 때 정확도는 비슷하지만 속도측면에서 월등한 성능을 확인할 수 있다.

Table 1 YOLOv3 improvement^[10]

Method	mAP-50	Time
SSD321	45.4	61
DSSD321	46.1	85
R-FCN	51.9	85
SSD513	50.4	125
DSS513	53.3	156
FPN FRCN	59.1	172
RetinaNet-50-500	50.9	73
RetinaNet-101-500	53.1	90
RetinaNet-101-800	57.5	198
YOLOv3-320	51.5	22
YOLOv3-416	55.3	29
YOLOv3-608	57.9	51

2.4 YOLOv5

본 연구에서 중점적으로 비교하고자 하는 YOLOv5^[11]는 이전의 YOLO 시리즈와는 다른 backbone을 사용하여 FPS와 mAP면에서 뛰어난 성능을 보인다. 가중치별로 지금까지 나온 YOLOv5의 모델들은 YOLOv5 s, m, l, x, s6, m6, l6, x6로 총 8개이다. 모델별로 s부터 x6까지 점진적으로 정확도가 향상되는 반면 속도는 느려진다. 이들은 각각 depth multiple과 width multiple의 차이로 발생하게 된다. 또한, 이전의 시리즈와 다른 backbone의 아키텍처를 사용하는데 CSPNet^[12]을 이용하여 학습능력을 강화시키는 것과 동시에 정확도를 유지하며 경량화를 가능하게 구성되었다. 기본적으로 본래 parameter 수가 감소하면 정확도도 감소하게 되지만 bottle-neck 현상을 방지하여 shortcut connection으로 덧셈 연산이 필요 없게 되며 결과적으로 연산량이 작아짐에도 정확도는 향상되는 효과가 있다.

Table 2에는 YOLOv3와 YOLOv5의 가중치 별 mAP, speed, parameter를 비교하여 정리하였다. 특히, YOLOv5의 parameter가 현저히 감소한 것을 확인할 수 있는데 이는 앞서 언급한 CSPNet backbone의 결과값을 정량적으로 살펴본 수치이다. YOLOv5를 기준으로 아직까지 학술 논문이 발표되지 않았기 때문에 성능에 대해 정확히 알 수는 없지만 Github에 언급된 성능 지표를 확인하면 이전의 YOLO버전 보다 parameter는 작아 졌음에도 정확도는 비슷하거나 높으며 추론 속도는 더 빨라진 것을 확인할 수 있다. YOLOv3를 제외한 YOLOv5s부터 YOLOv5x6까지 비교해 본 결과 mAP의 수치는 이미지 사이즈가 크고 가중치가 높은 YOLOv5x6의 값이 가장 높았지만 속도는 이와 반비례하는 것을 확인할 수 있고 speed와 parameter는 이미지 사이즈가 같을 때 비례하게 되는 것을 확인할 수 있다.

Table 2 YOLOv5 comparison^[10,11]

Model	Size (pixels)	mAP	Speed (ms)	Parameters (M)
YOLOv3	416	55.3	29	61.9
YOLOv5s	640	56.0	6.4	7.2
YOLOv5m	640	63.9	8.2	21.2
YOLOv5l	640	67.2	10.1	46.5
YOLOv5x	640	68.9	12.1	86.7
YOLOv5s6	1280	63.0	8.2	16.8
YOLOv5m6	1280	69.0	11.1	35.7
YOLOv5l6	1280	71.6	15.8	76.8
YOLOv5x6	1280	72.4	26.2	140.7

3. 성능 비교 실험

용사, 용접, 레이저 클래딩 공정 중 발견된 결함에 대해 현미경을 이용하여 이미지 셋을 만들었고 200개의 데이터로는 신뢰성 있는 결과값을 만들 수 없기 때문에 data augmentation을 통해 5,982개의 데이터로 증폭시켰다. Image K 알고리즘을 사용하여 이미지 사이즈를 416 × 416 사이즈로 규격화 하였고 data set의 class는 porosity와 crack으로 세팅했으며 aatch 사이즈는 16, 반복횟수는 500으로 지정했다. 실험결과 Fig. 4와 같이 YOLOv3의 precision은 95.2%, recall은 91.4%. 그리고 mAP는 94.7%로 높은 정확도를 확인할 수 있었다. 또한, FPS는 35로 사람이 느끼는 real-time detection의 기준인 30 FPS보다 높아 충분히 실시간 감지가 가능한 것으로 확인할 수 있었다.

YOLOv5의 성능을 비교하기 위해 여러 모델 중 실시간성이 좋은 YOLOv5s를 이용하여 비교 실험을 진행하였다. 실험 결과 Fig. 5와 같이 precision은 94.63%, recall은 92.06%, mAP는 95.77%가 도출되었다. YOLOv5s의 FPS는 140으로 알려져 있

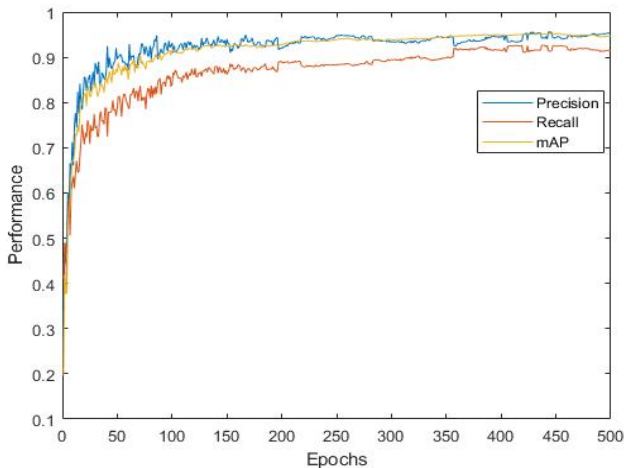


Fig. 4 Performance of YOLOv3

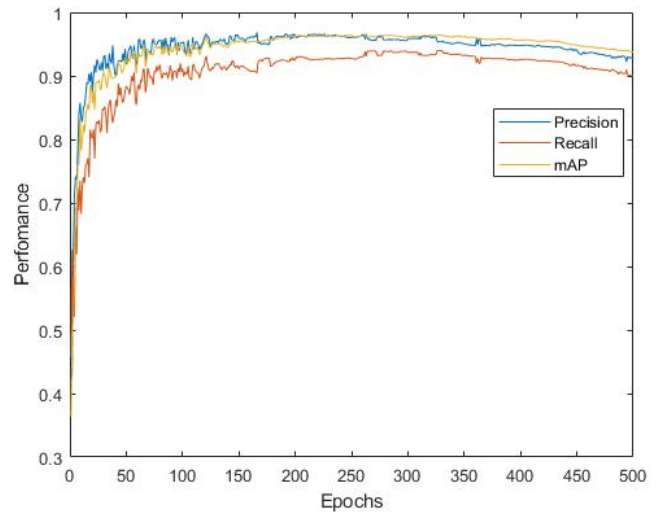


Fig. 5 Performance of YOLOv5s

었는데 본 연구에서 확인 결과 그 보다 약 4% 더 빠른 146 FPS로 측정되었다. 이는 class의 숫자 감소로 인해 기존 연구결과보다 더 빠르다는 것을 유추할 수 있다. 이러한 것은 YOLOv3와 YOLOv5s의 run-time을 비교했을 때 확연히 확인할 수 있는데, 이는 각각의 알고리즘의 parameter 값과 backbone이 달라 연산량 차이로 발생한 결과라 판단할 수 있으며 YOLOv5s의 YOLOv3의 precision 값이 0.57%의 차이 인데 반해 run-time 관점에서 500 Epochs를 수행하는데 YOLOv3는 8.582시간, YOLOv5s는 3.322시간으로 YOLOv5s를 이용하면 약 38%의 시간으로 유사한 결과를 도출함을 확인하였다. 그러나, YOLOv5의 경우 300 Epochs를 넘겼을 때부터 precision, recall이 감소하는 현상을 확인하였다. 이는 overfitting 현상이며 300번의 반복횟수가 YOLOv5 알고리즘의 알맞은 반복횟수임을 알 수 있다. 그러나, one-stage detection의 단점이라고 할 수 있는 작은 객체 탐지에 있어서 부진한 성능을 보여준다는 관점에서 본 연구 결과 큰 객체들의 검출 성능은 좋았지만 작은 객체들의 검출은 비교적 미흡했다. 이는 여전히 높은 정확도임에도 전체적인 mAP를 낮춘 결과라고 볼 수 있다. 그러나, 가중치가 높은 YOLOv5x6와 같은 모델을 사용했을 때 이 점이 다소 보완되는 것을 확인했고 추후 실험을 통해 보다 많은 양의 라벨링 이미지들과 적절한 Epochs를 사용하면 충분히 긍정적인 결과를 기대할 수 있을 것으로 판단된다. Table 3에는 앞서 설명한 YOLOv3와 YOLOv5s의 결과를 정리하였다.

Table 3 YOLOv3, YOLOv5 comparison

Model	Precision	Recall	mAP	FPS
YOLOv3	95.2	91.4	94.7	35
YOLOv5s	94.63	92.06	95.77	146

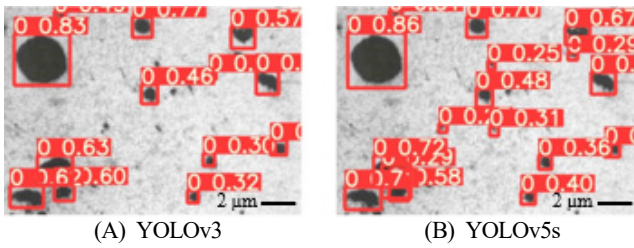


Fig. 6 Defect detection results comparison

4. 결론

본 논문에서는 공정 중 발견된 결함의 현미경 이미지를 이용하여 YOLO 알고리즘 별 비교분석을 진행하였다. 사용자의 확대와 기술의 발달에 따라 새로운 알고리즘이 지속적으로 제안되고 있으며, YOLO 시리즈 중 최신 모델로 제안된 YOLOv3와 YOLOv5s를 비교하였다. YOLOv3와 v5의 가장 큰 변화는 backbone의 구조 변화로 YOLOv5에서는 CSPNet를 채택하고 있기 때문에 parameter가 YOLOv3의 경우 61.9 M인데 반해 YOLOv5s의 경우 7.2 M로 작아지고 전체의 용량이 작아진 것을 확인할 수 있다. 가중치가 작아지면 정확도는 감소하게 되는 일반 논리에 반해 YOLOv5는 실험결과에서 보이는 것과 같이 precision, recall, mAP 등 YOLOv3에 비해 큰 차이가 없음을 확인하였다. 또한 YOLOv5의 경우 s, m, l, x, s6, m6, l6, x6과 같은 총 8가지의 모델 중에서 관심도에 따라 선택할 수 있기 때문에 정확도에 초점을 둔다면 정확도가 가장 좋지만 검출속도가 상대적으로 느린 x6를, 검출속도에 초점을 둔다면 정확도는 다소 좋지 않지만 검출속도가 빠른 s를 선택하여 객체감지를 할 수 있다는 장점도 있다.

YOLOv3와 YOLOv5s로 학습을 완료한 weights로 동영상 객체감지를 진행했을 때 YOLOv3에서 감지하지 못한 object들을 YOLOv5에서는 감지하는 것을 확인하였다. 학습 데이터 외의 새로운 데이터 이미지를 활용해 객체감지 테스트를 진행한 결과에서도 Fig. 6과 같이 YOLOv3에서는 감지하지 못하는 객체도 YOLOv5s는 감지함과 동시에 정확도 또한 더 높은 것을 볼 수 있다.

또한 weights의 용량도 YOLOv3의 경우 113 MB인데 반해, YOLOv5s의 경우 13 MB 수준으로 약 10% 밖에 되지 않는다. 그렇기 때문에 YOLOv3와 YOLOv5 중 YOLOv5의 모델이 더 개선된 모델이라고 볼 수 있다. 특히, 객체 탐지 속도를 우선으로 생각한다면 YOLOv5s를 선택하는 것이 좋다. Precision 측면으로는 93.53%로 YOLOv3에 비해 2% 정도의 차이가 나지만 146 FPS로 YOLOv3에 비해 약 5 배 가량 빠른 객체 탐지 속도를 볼 수 있기 때문에 효과적이다. 또한 precision, recall, mAP 등 정확도 측면에서 모델을 선택한다면 YOLOv5s를 제외한 다른 모델들

은 모두 YOLOv3를 능가하는 성능을 보여주기 때문에 필요에 따라 가중치에 부합하는 모델을 선택하여 사용할 수 있다. 따라서, 공정 중에 발생하는 기공, 미세 균열 등의 결함 검출과 검출의 실시간 관점에서 YOLOv5 알고리즘의 적용이 보다 유효할 것이라 보여지고, 아직 초기 개발 단계인 점을 감안하면, 향후 보다 좋은 성능과 안정성을 확보할 수 있을 것으로 기대된다.

후기

이 연구는 산업통상자원부(MOTIE)와 한국에너지기술평가원(KETEP)의 지원을 받아 수행한 연구 과제입니다(no. 20206410 100080).

References

- [1] Plessis, A., Yadroitsava, I., Yadroitsev, I., 2020, Effects of Defects on Mechanical Properties in Metal Additive Manufacturing: A Review Focusing on X-ray Tomography Insights, *Mater. Des.*, 187 108385, <https://doi.org/10.1016/j.matdes.2019.108385>.
- [2] Büchi, G., Cugno, M., Castagnoli, R., 2020, Smart Factory Performance and Industry 4.0, *Technol. Forecast. Soc. Chang.*, 150 119790, <https://doi.org/10.1016/j.techfore.2019.119790>.
- [3] Redmon, J., Divvala, S., Girshick, R., Farhadi, A., 2016, You Only Look Once: Unified, Real-Time Object Detection, *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 779-788, <https://doi.org/10.1109/CVPR.2016.91>.
- [4] Choi, B., Choi, Y., Lee, M. G., Kim, J. S., Lee, S. W., Jeon, Y., 2021, Defect Detection Using Deep Learning-Based YOLOv3 in Cross-Sectional Image of Additive Manufacturing, *Arch. Metall. Mater.*, 66:4 1037-1041.
- [5] Park, I. M., Kim B. S., 1995, Principle of Nondestructive Testing Technique and Its Recent Application, *J. Korea Foundry Society*, 15:4 299-316.
- [6] Bochkovskiy, A., Wang, C. Y., Liao, H. M., 2020, viewed 17 November 2021, YOLOv4: Optimal Speed and Accuracy of Object Detection, < <https://arxiv.org/pdf/2004.10934v1.pdf> >.
- [7] Krizhevsky, A., Sutskever, I., Hinton, G. E., 2017, ImageNet Classification with Deep Convolutional Neural Networks, *Commun. ACM*, 60:6 84-90, <https://doi.org/10.1145/3065386>.
- [8] Soviany, P., Ionescu, R. T., 2018, Optimizing the Trade-off between Single-Stage and Two-stage Deep Object Detectors using Image Difficulty Prediction, 20th International

- Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC), 209-214, <https://doi.org/10.1109/SYNASC.2018.00041>.
- [9] Girshick, R., 2015, Fast R-CNN, Proceedings of the IEEE International Conference on Computer Vision (ICCV), 1440-1448, <https://doi.org/10.1109/ICCV.2015.169>.
- [10] Redmon, J., Farhadi, A., 2018, viewed 17 November 2021, YOLOv3: An Incremental Improvement, <<https://arxiv.org/pdf/1804.02767.pdf>>.
- [11] Glenn-jocher, 2020, viewed 17 November 2021, YOLOv5, <<https://github.com/ultralytics/yolov5>>.
- [12] Wang, C. Y., Liao, H. M., Wu, Y. H., Chen, P., Hsieh, J., Yeh, I., 2020, CSPNet: A New Backbone That can Enhance Learning Capability of CNN, Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, 1571-1580, <https://doi.org/10.1109/CVPRW50498.2020.00203>.



Ik-Su Kim

Master Student in the Department of Mechanical Engineering, Ajou University. His research interest is deep learning on novel manufacturing processes.
E-mail: hiiksu@ajou.ac.kr



Moon Gu Lee

Professor in the Department of Mechanical Engineering, Ajou University. His research interest is design and control of precision system control.
E-mail: moongulee@ajou.ac.kr



Yongho Jeon

Professor in the Department of Mechanical Engineering, Ajou University. His research interest is novel manufacturing processes.
E-mail: princaps@ajou.ac.kr