

Object Detector



공사현장 안전장비 검출 모델 및 엔진 개발

2021.08.28

김윤중 박사
ArtAI 인공지능연구소
Netvision Telecom Inc.
yjkim@netvision.com, yjkim@hanbat.ac.kr

Contents

- AI Hub 공사현장 안전장비 인식 이미지 데이터
- 안전장비 데이터세트, SGD_ds50000
- 공사현장 안전장비 검출 엔진
 - 공사현장 안전장비 검출엔진 구성도
 - 공사현장 안전장비 검출모델 SGD_50000
 - 공사현장 안전장비 검출엔진의 구성 모듈
 - 공사현장 안전장비 검출엔진 사용 방법
- 공사현장 안전장비 검출 엔진응용 사례 및 검토
- Issues 및 응용서비스개발시 유의사항

AI Hub 공사현장 안전장비 인식 이미지 데이터

- 개요
 - [AI Hub 공사현장 안전장비 인식 이미지 데이터](#)
 - image file number : 1,984,333
 - label file number : 2,361,008
- 근로자 착용 안전장비 8종 dataset

구분		Number	Description
Image file		851,063	
안전장구 Sample	1 벨트착용	1,205,131	0 벨트착용
	2 벨트미착용	878,174	1 벨트미착용
	3 안전고리착용	34,092	
	4 안전고리미착용	245,186	
	5 안전화착용	1,255,514	2 안전화착용
	6 안전화미착용	160,794	3 안전화미착용
	7 안전모착용	1,323,754	4 안전모착용
	8 안전모미착용	246,279	5 안전모미착용
	계	5,348,924	

안전장비 데이터세트, SGD_ds50000

- **SGD_ds50000**

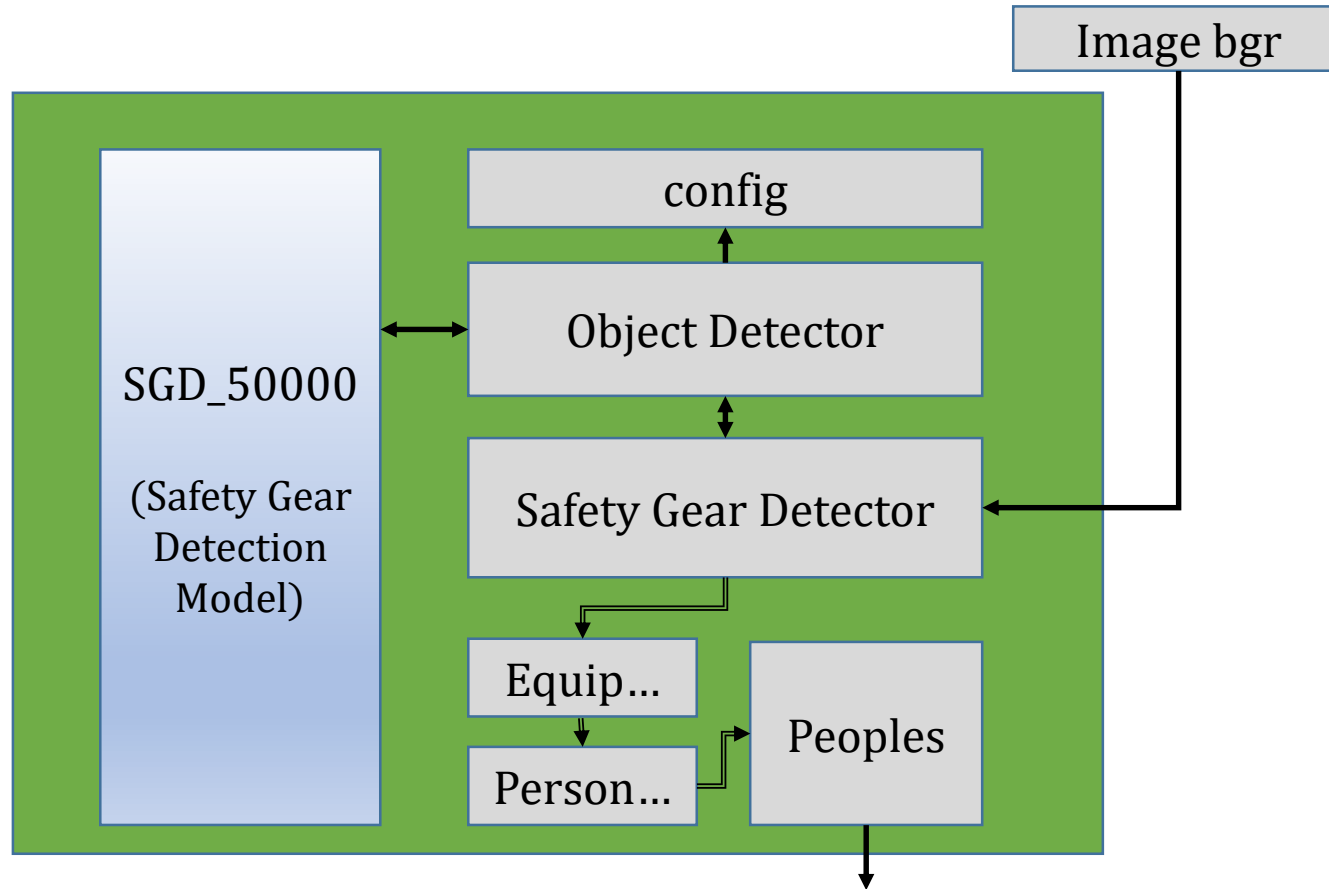
- description

안전장비당 50,000 샘플 이상 총 **674,928** samples, **112,886** 이미지

구분		Train	Test/Validation	Total
Image		84,664	28,222	112,886
안전장구 Sample	0 벨트착용	118,701	39,443	158,144
	1 벨트미착용	95,974	31,986	127,960
	2 안전화착용	110,558	37,207	147,765
	3 안전화미착용	37,748	12,252	50,000
	4 안전모착용	80,922	26,820	107,742
	5 안전모미착용	62,507	20,810	83,317
	계	506,410	168,518	674,928

공사현장 안전장비 검출 엔진

- 공사현장 안전장비 검출엔진 구성도



공사현장 안전장비 검출 엔진(cont.)

- 공사현장 안전장비 검출모델 SGD_50000

- 모델사양

YOLO_TYPE = "yolov4"
YOLO_FRAMEWORK = "tensorflow "
YOLO_INPUT_SIZE = 416
MODEL_NAME = "Yolov4-AI20210801_50000"
DATASET = "SGD_ds50000"

- 모델성능

- 학습처리시간

- 2021-08-09 14:13:36 ~ (13 days)

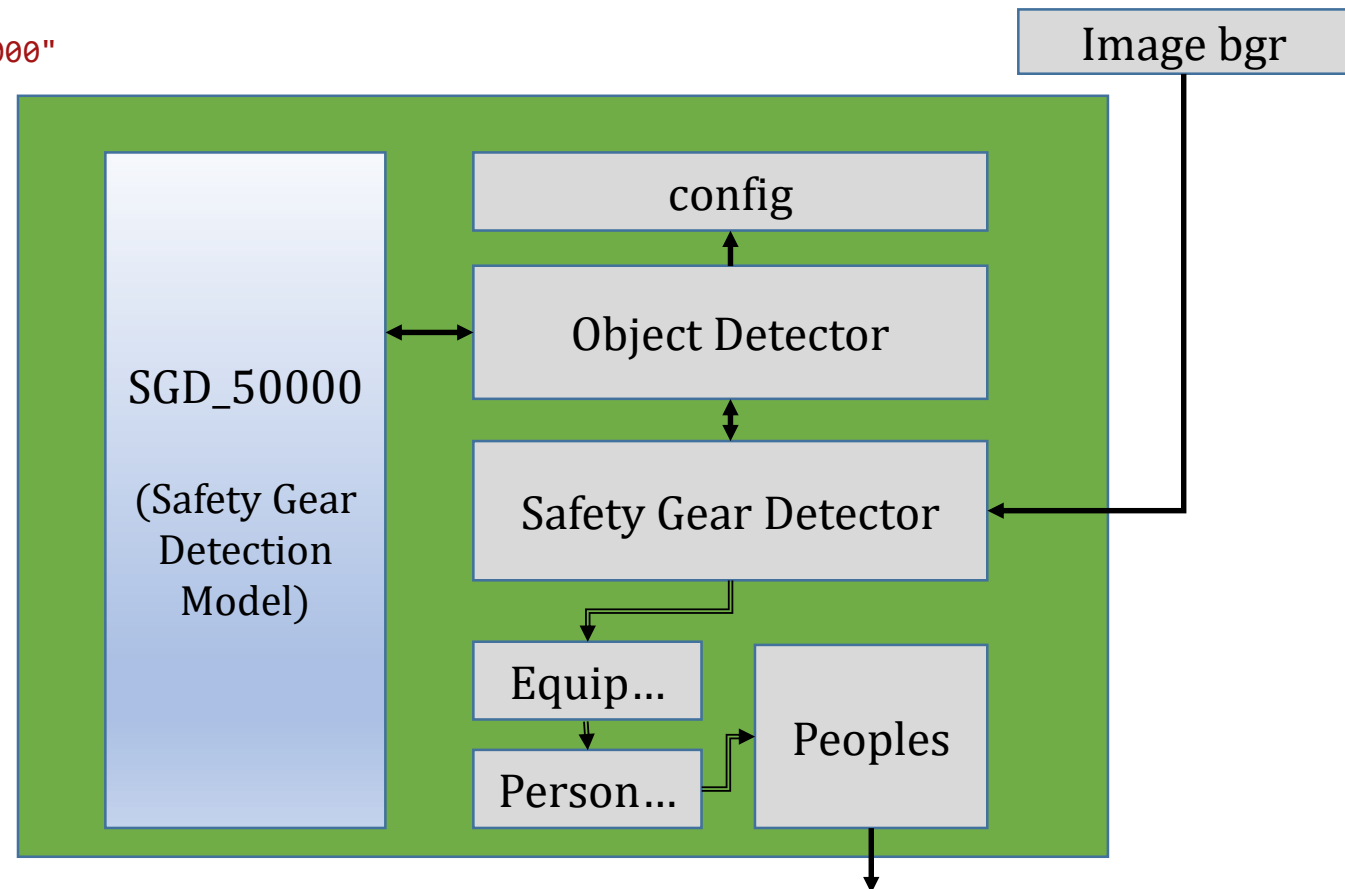
- Epoch : 78

- Validation loss

giou_val_loss : 5.444
conf_val_loss : 2.977
prob_val_loss : 3.263
total_val_loss/Best : 11.684

- Mean Average Precision

mAP : 88.60%



공사현장 안전장비 검출 엔진(cont.)

- 공사현장 안전장비 검출엔진의 구성 모듈

```
OPEN EDITORS 1 UNSAVED
test.py
SaftyGearDetector.py
_SAFETYGEARDETECTOS
  > __pycache__
  > models
    S2-N2201M02549.jpg
    yolov4-AI20210801_50000F.h5
  configs.py
  object_detector.py
  SaftyGearDetector.py
  test.py

configs.py > [object_model]
1 object_model = None

object_detector.py > ...
1 #
2 # 공사현장안전장비 검출
3 #
4 # 김윤중
5 # yjkim@netvison.com, mryjkim55@gmail.com
6 # 아트인공지능연구소
7 # 네트비전텔레콤(주)
8 # 2021-08-29
9
10 import os
11 from typing import List
12 os.environ['TF_CPP_MIN_LOG_LEVEL'] = '2'
13 import numpy as np
14 import tensorflow as tf
15 import cv2
16 import configs
17
18 class object_detector :
19 >     def __init__(self, model_path=r'models\yolov4-AI20210801_10000F.h5') : ...
34 >     def detect(self, img_rgb) : ...
51 >     def image_preprocess(self, image, target_size, gt_boxes=None): ...
71 >     def postprocess_boxes(self, pred_bbox, original_image, input_size, score_threshold): ...
110 >     def nms(self, bboxes, iou_threshold, sigma=0.3, method='nms'): ...
149 >     def bboxes_iou(self, boxes1, boxes2): ...
```

```
SaftyGearDetector.py > ...
1 #
2 # 공사현장안전장비 검출
3 #
4 # 김윤중
5 # yjkim@netvison.com mryjkim55@gmail.com
6 # 아트인공지능연구소
7 # 네트비전텔레콤(주)
8 # 2021-08-29
9
10 import os
11 os.environ['TF_CPP_MIN_LOG_LEVEL'] = '2'
12 from typing import List
13 import numpy as np
14 import cv2
15 import copy
16 from object_detector import object_detector
17
18 class Person :
19 >     def __init__(self, 모자=None, 안전벨트=None, 안전화=None): ...
27 >     def reset_frame(self): ...
36     @property
37 >     def SaftyEquips(self): ...
43 >     def set(self, name, value): ...
47 >     def info(self): ...
55
56 class People(List):
57 >     def __init__(Self): ...
59     @property
60 >     def 모자(self): ...
62     @property
63 >     def 안전벨트(self): ...
65     @property
66 >     def 안전화(self): ...
68 >     def info(self): ...
73
74 class SaftyGearDetector(object_detector) :
75 >     def __init__(self, model_path=r'models\yolov4-AI20210801_50000F.h5'): ...
80 >     def detect_rgb(self, img_rgb): ...
84 >     def detect_bgr(self, img_bgr): ...
87 >     def detect_path(self, path) : ...
90 >     def get_names(self, bbs) : ...
93 >     def iou(self, f, b): ...
128 >     def recognize_person(self, bs) : ...
272 >     def imread(self, path): ...
275 >     def detect_people_bgr(self, img_bgr): ...
279 >     def detect_people_path(self, path): ...
284
```

공사현장 안전장비 검출 엔진(cont.)

- 공사현장 안전장비 검출엔진 사용 방법

```
OPEN EDITORS 1 UNSAVED test.py > ...
x test.py
SAFTYGEARDETECTOS
  > __pycache__
  v models
    S2-N2201M02549.jpg
    yolov4-AI20210801_50000F.h5
  configs.py
  object_detector.py
  SaftyGearDetector.py
  test.py

1 from SaftyGearDetector import SaftyGearDetector
2
3 sgd=SaftyGearDetector()
4 path=r'models\S2-N2201M02549.jpg'
5
6 img_bgr = sgd.imread(path)
7 bbs      = sgd.detect_bgr(img_bgr)
8 peoples = sgd.recognize_person(bbs)
9
10 #or
11 #peoples = sgd.detect_people_path(path)
12 #or
13 #peoples = sgd.detect_people_bgr(img_bgr)
14
15
16 print(peoples)
17 """
18 [{'person': 0, 'frame': [257, 328, 449, 557], 'SaftyEquips': [[303, 328, 368, 4
19 | {'person': 1, 'frame': [1067, 345, 1164, 492], 'SaftyEquips': [[1089, 345, 1138,
20 | {'person': 2, 'frame': [1160, 360, 1471, 933], 'SaftyEquips': [[1160, 360, 1249,
21 | ]
22 | """]
```


공사현장 안전장비 검출 엔진응용 사례 및 검토

```
test.py
SaftyGearDetector.py
_SAFETYGEARDETECTOS
> __pycache__
models
  S2-N2201M02549.jpg
  yolov4-AI20210801_50000F.h5
configs.py
object_detector.py
SaftyGearDetector.py
test.py

1 from SaftyGearDetector import SaftyGearDetector
2
3 sgd=SaftyGearDetector()
4 path=r'models\S2-N2201M02549.jpg'
5
6 img_bgr = sgd.imread(path)
7 bbs      = sgd.detect_bgr(img_bgr)
8 peoples = sgd.recognize_person(bbs)
9
10 #or
11 #peoples = sgd.detect_people_path(path)
12
13 print(peoples)
14 """
15 [{"person": 0, 'frame': [257, 328, 449, 557], 'SaftyEquips': [[303, 328, 368, 405, 4, 0.7731397747993469], [257, 397, 449, 557, 1, 0.856677234172821]]},
16 {'person': 1, 'frame': [1067, 345, 1164, 492], 'SaftyEquips': [[1089, 345, 1138, 397, 4, 0.36461982131004333],[1067, 391, 1164, 492, 0, 0.6994885802268982]]},
17 {'person': 2, 'frame': [1160, 360, 1471, 933], 'SaftyEquips': [[1160, 360, 1249, 466, 4, 0.9398739337921143], [1231, 376, 1471, 588, 1, 0.8807631134986877], [1282, 698, 1398, 933, 2, 0.7132970690727234]]},
18 ]
19 """
```



```
path=r'models\S2-N2201M02549.jpg'
```

```
img_bgr = sgd.imread(path)  
bbs      = sgd.detect_bgr(img_bgr)  
peoples  = sgd.recognize_person(bbs)
```

```
#or  
#peoples = sgd.detect_people_path(path)
```

```
print(peoples)
```

```
"""  
[{'person': 0, 'frame': [257, 328, 449, 557], 'SaftyEquips': [[303, 328, 368, 405, 4, 0.7731397747993469], [257, 397, 449, 557, 1, 0.856677234172821]]},  
{'person': 1, 'frame': [1067, 345, 1164, 492], 'SaftyEquips': [[1089, 345, 1138, 397, 4, 0.36461982131004333], [1067, 391, 1164, 492, 0, 0.6994885802268982]]},  
{'person': 2, 'frame': [1160, 360, 1471, 933], 'SaftyEquips': [[1160, 360, 1249, 466, 4, 0.9398739337921143], [1231, 376, 1471, 588, 1, 0.8807631134986877], [1282, 698, 1398, 933, 2, 0.7132970690727234]]}  
]"""
```



응용사례 및 검토(cont.)



```
{'person': 0, 'frame': [592, 368, 703, 714], 'SaftyEquips' : [[627, 368, 678, 436, 4, 0.63], [592, 422, 703, 543, 1, 0.81], [593, 617, 682, 714, 2, 0.81]]}  
{'person': 1, 'frame': [685, 359, 767, 658], 'SaftyEquips' : [[703, 359, 744, 412, 4, 0.46], [685, 410, 767, 506, 0, 0.58], [704, 582, 762, 658, 2, 0.65]]}  
{'person': 2, 'frame': [1465, 354, 1579, 707], 'SaftyEquips':[[1516, 354, 1569, 413, 4, 0.63], [1465, 411, 1579, 545,1, 0.82], [1468, 606, 1568, 707, 3, 0.80]]}
```

응용사례 및 검토(cont.)



Person:0 frame:[750, 339, 891, 772] SaftyEquis:[[806, 339, 873, 408, 4, 0.51], [769, 400, 891, 555, 0, 0.56], [750, 638, 882, 772, 2, 0.49]]
Person:1 frame:[938, 403, 1169, 1059] SaftyEquis:[[972, 403, 1075, 513, 4, 0.93], [938, 518, 1140, 755, 0, 0.93], [999, 865, 1169, 1059, 2, 0.93]]

응용사례 및 검토(cont.)



Person:0 frame:[750, 339, 891, 772] SaftyEquis:[[806, 339, 873, 408, 4, 0.51], [769, 400, 891, 555, 0, 0.56], [750, 638, 882, 772, 2, 0.49]]
Person:1 frame:[938, 403, 1169, 1059] SaftyEquis:[[972, 403, 1075, 513, 4, 0.93], [938, 518, 1140, 755, 0, 0.93], [999, 865, 1169, 1059, 2, 0.93]]

응용사례 및 검토(cont.)



Person:0 frame:[4, 464, 115, 624]

SaftyEquis:[[60, 464, 113, 522, 4, 0.45], [4, 511, 115, 624, 1, 0.56]]

Person:1 frame:[390, 213, 711, 1034]

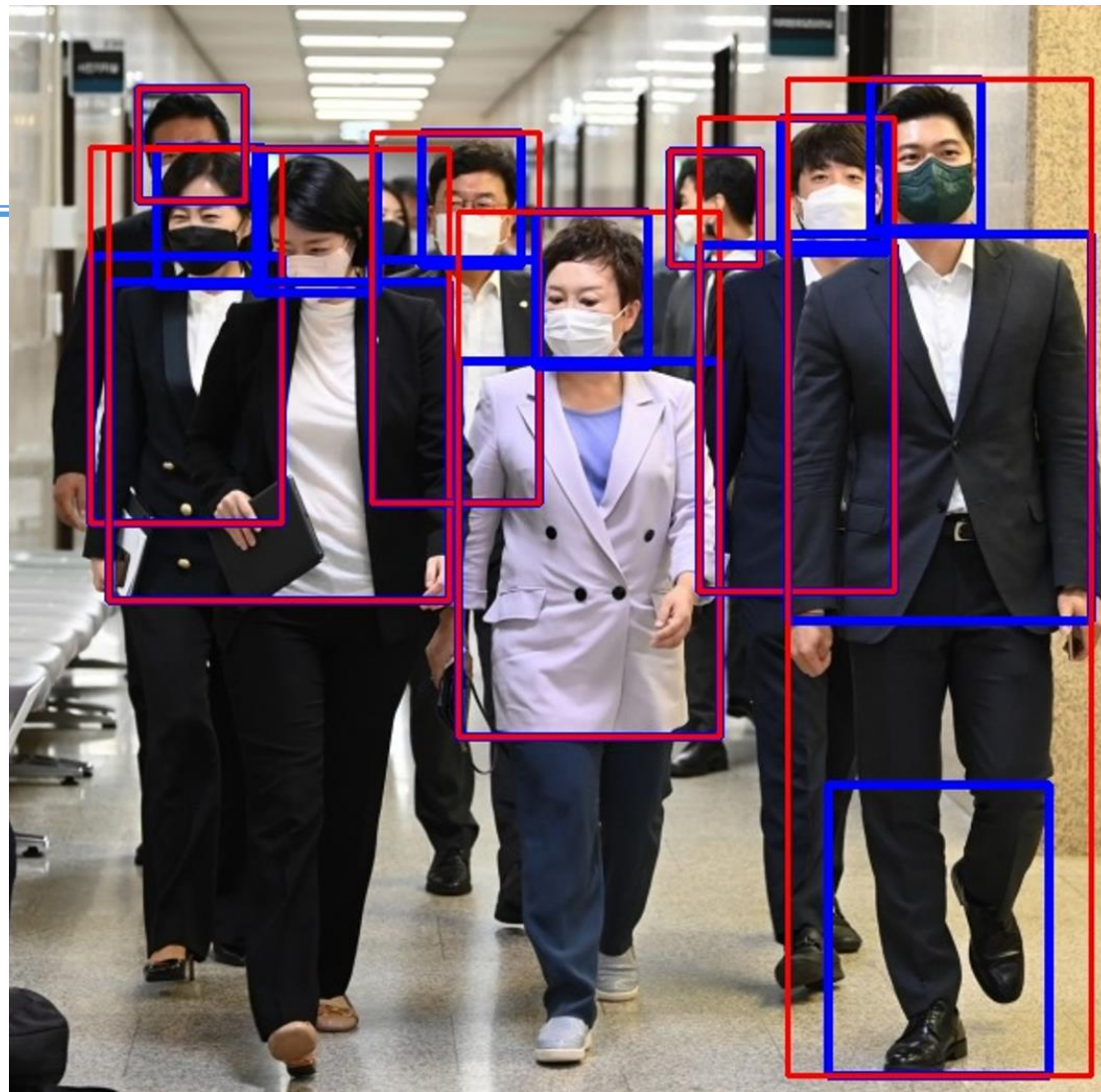
SaftyEquis:[[528, 213, 643, 317, 4, 0.93], [438, 310, 711, 590, 1, 0.95], [390, 685, 664, 1034, 2, 0.89]]

응용사례 및 검토(cont.)



Person:0 frame:[1660, 419, 1799, 769] SaftyEquis:[[1714, 419, 1768, 481, 4, 0.55], [1660, 477, 1799, 611, 1, 0.88], [1675, 663, 1754, 769, 2, 0.75]]

응용사례 및 검토(cont.)



Person:0 frame:[58, 95, 162, 299]

Person:1 frame:[68, 96, 253, 341]

Person:2 frame:[84, 63, 142, 124]

Person:3 frame:[211, 87, 302, 288]

Person:4 frame:[258, 130, 400, 415]

Person:5 frame:[373, 97, 422, 159]

Person:6 frame:[389, 79, 495, 336]

Person:7 frame:[437, 58, 602, 599]

SaftyEquis:[[94, 95, 152, 170, 4, 0.64], [58, 154, 162, 299, 1, 0.45]]

SaftyEquis:[[148, 96, 214, 174, 5, 0.88], [68, 168, 253, 341, 1, 0.84]]

SaftyEquis:[[84, 63, 142, 124, 4, 0.49]]

SaftyEquis:[[238, 87, 292, 159, 5, 0.86], [211, 156, 302, 288, 1, 0.62]]

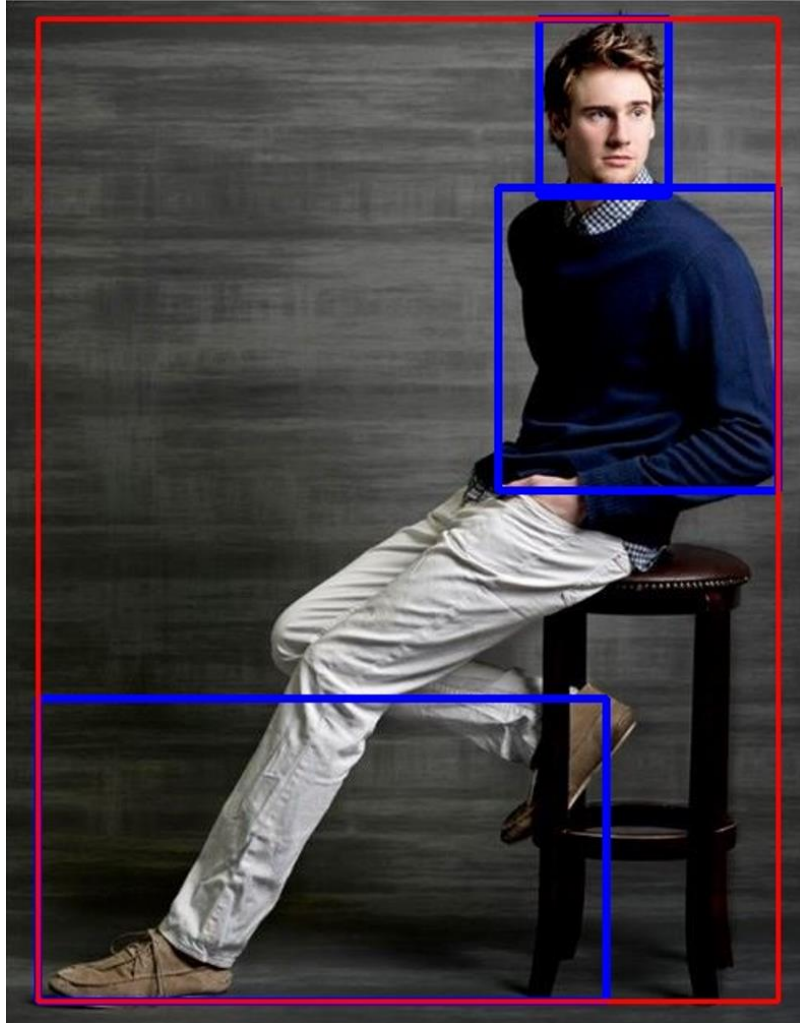
SaftyEquis:[[300, 130, 361, 214, 5, 0.93], [258, 211, 400, 415, 1, 0.87]]

SaftyEquis:[[373, 97, 422, 159, 5, 0.66]]

SaftyEquis:[[433, 79, 491, 152, 5, 0.63], [389, 148, 495, 336, 1, 0.57]]

SaftyEquis:[[482, 58, 542, 139, 5, 0.65], [437, 142, 602, 352, 1, 0.92], [459, 441, 579, 599, 2, 0.41]]

응용사례 및 검토(cont.)



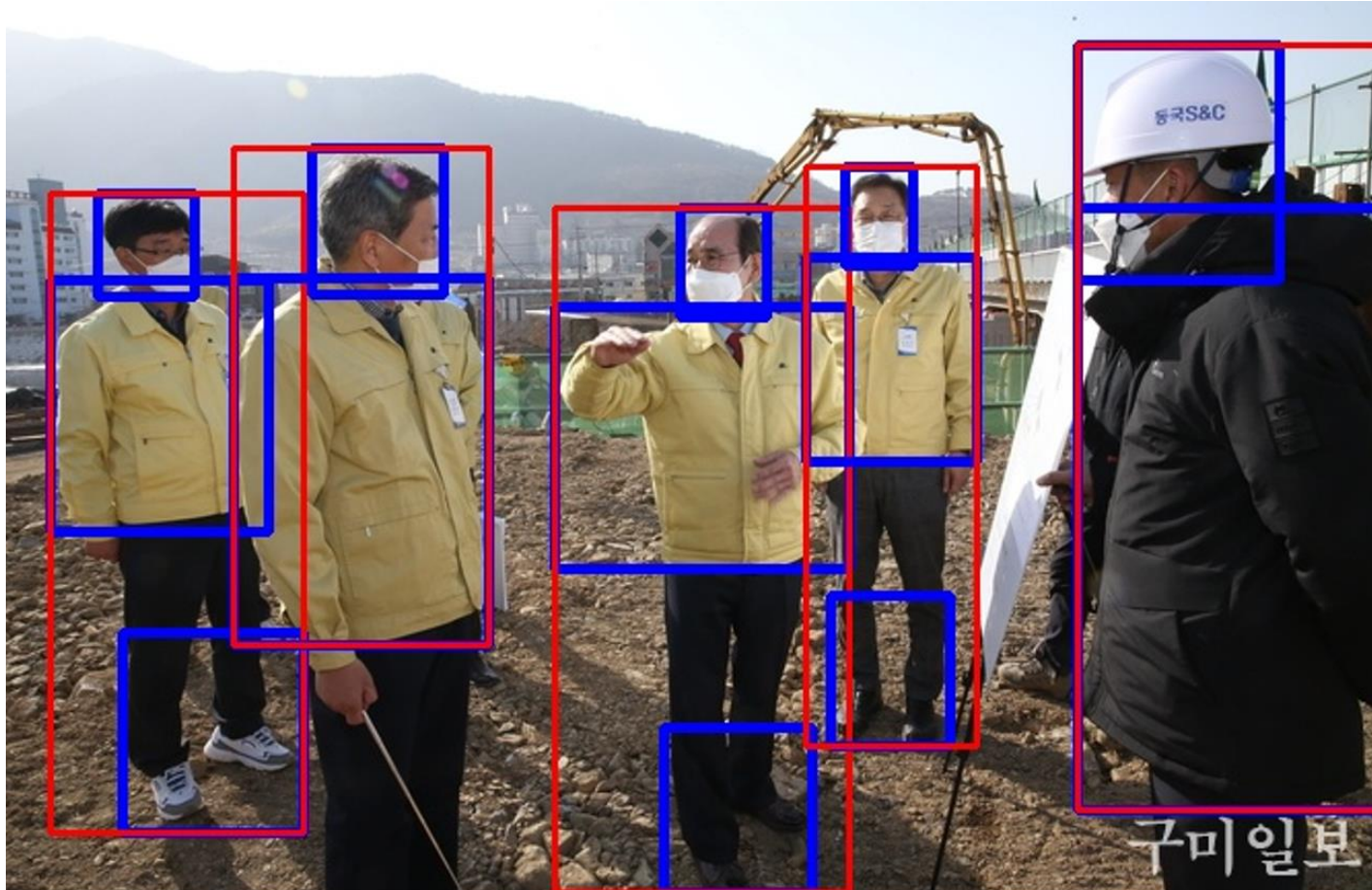
[{'person': 0, 'frame': [21, 13, 473, 613], 'SaftyEquips': [[327, 13, 406, 121, 4, 0.62], [302, 116, 473, 301, 1, 0.82], [21, 428, 368, 613, 2, 0.46]]}]

응용사례 및 검토(cont.)



```
[{'person': 0, 'frame': [238, 511, 387, 677], 'SaftyEquips': [[238, 511, 387, 677, 5, 0.73]]}
{'person': 1, 'frame': [498, 556, 618, 713], 'SaftyEquips': [[498, 556, 618, 713, 4, 0.54]]}
{'person': 2, 'frame': [576, 433, 1198, 1254], 'SaftyEquips': [[768, 433, 1009, 705, 5, 0.91], [576, 648, 1198, 1254, 1, 0.95]]}
{'person': 3, 'frame': [681, 469, 820, 660], 'SaftyEquips': [[681, 469, 820, 660, 5, 0.65]]}
{'person': 4, 'frame': [1239, 148, 1962, 1307], 'SaftyEquips': [[1239, 148, 1569, 561, 4, 0.96], [1383, 383, 1962, 1307, 1, 0.91]]}
{'person': 5, 'frame': [1150, 531, 1426, 1089], 'SaftyEquips': [[1197, 531, 1320, 706, 4, 0.31], [1150, 683, 1426, 1089, 1, 0.90]]}
```

응용사례 및 검토(cont.)



```
{'person': 0, 'frame': [30, 113, 153, 423], 'SaftyEquips': [[53, 113, 100, 163, 5, 0.40], [30, 155, 136, 277, 1, 0.93], [65, 326, 153, 423, 2, 0.71]]}  
{'person': 1, 'frame': [119, 91, 243, 332], 'SaftyEquips': [[157, 91, 221, 162, 4, 0.49], [119, 154, 243, 332, 1, 0.96]]}  
{'person': 2, 'frame': [275, 120, 418, 452], 'SaftyEquips': [[336, 120, 378, 173, 4, 0.76], [275, 168, 418, 295, 1, 0.87], [328, 372, 400, 452, 2, 0.61]]}  
{'person': 3, 'frame': [397, 100, 480, 381], 'SaftyEquips': [[416, 100, 449, 148, 4, 0.42], [397, 144, 480, 243, 1, 0.92], [409, 308, 467, 381, 2, 0.50]]}  
{'person': 4, 'frame': [529, 41, 679, 412], 'SaftyEquips': [[529, 41, 627, 155, 4, 0.88], [529, 120, 679, 412, 1, 0.85]]}
```

Issues 및 응용서비스시 개발유의 사항

- 학습데이터에 대한 문제
 - 얼굴의 비식별화처리 존재
 - 라벨링의 오류 다수 존재
- 응용서비스시 개발유의 사항
 - 사람과 장비 크기의 정도
 - 개체가 중첩되는 정도
- 장비 검출 비율 양호
- 장비의 착용/미착용 인식을
 - 안전모 미흡
 - 안전벨트 양호
 - 안전화 다소 양호

안전장비		학습용	검증용	total	검출정확도
안전모	착용	80922	26820	107,742	92.50%
	미착용	62507	20810	83,317	91.09%
안전벨트	착용	118701	39443	158,144	88.22%
	미착용	95974	31986	127,960	91.90%
안전화	착용	110558	37207	147,765	77.62%
	미착용	37748	12252	50,000	87.08%
total		506,410	168,518	674,928	88.07%