반도체디스플레이기술학회지 제19권 제2호(2020년 6월)
Journal of the Semiconductor & Display Technology, Vol. 19, No. 2. June 2020.

심층신경망을 이용한 PCB 부품의 검지 및 인식

조태훈[*†]
[*†]한국기술교육대학교 컴퓨터공학부

Detection of PCB Components Using Deep Neural Nets

Tai-Hoon Cho[*†]
[*†]School of Computer Science and Engineering, Korea University of Technology and Education

ABSTRACT

In a typical initial setup of a PCB component inspection system, operators should manually input various information such as category, position, and inspection area for each component to be inspected, thus causing much inconvenience and longer setup time. Although there are many deep learning based object detectors, RetinaNet is regarded as one of best object detectors currently available. In this paper, a method using an extended RetinaNet is proposed that automatically detects its component category and position for each component mounted on PCBs from a high-resolution color input image. We extended the basic RetinaNet feature pyramid network by adding a feature pyramid layer having higher spatial resolution to the basic feature pyramid. It was demonstrated by experiments that the extended RetinaNet can detect successfully very small components that could be missed by the basic RetinaNet. Using the proposed method could enable automatic generation of inspection areas, thus considerably reducing the setup time of PCB component inspection systems.

Key Words : Deep Neural Networks, Object Detection, PCB Component Inspection, RetinaNet

1. 서 론

최근 모바일폰 같은 전자기기의 고성능화, 소형화가 진전됨에 따라 기기 안의 인쇄회로기판 (PCB)위에 있는 부품의 고집적화, 초소형화가 급격히 진행되고 있다. 또한 점점 늘어나고 있는 차량의 지능적인 전자제어를 위한 PCB의 무결성이 요구되고 있다. 이러한 PCB 부품의 장착 및 납땜 상태를 자동 검사하는 시스템이 PCB 생산 품질 관리의 핵심 장비로 생산 현장에서 필수적으로 사용되고 있는 추세이다.

PCB 검사 장비에서 흔히 검사할 각 부품의 종류 및 위치, 검사 영역을 장비 셋업 과정에서 작업자가 수동으로 입력해야 하는 번거로움이 많이 발생하여 장비 셋업 시간의 상당 부분을 차지하고 있다.

†E-mail: thcho@koreatech.ac.kr

본 논문은 2D 카메라로 얻은 PCB 컬러 영상을 입력으로 각 부품의 종류와 위치, 부품영역을 심층신경망을 이용하여 자동으로 추출하는 방법을 제안한다.

범용적인 영상인식 딥러닝 네트워크는 크게 객체분류(object classification), 객체검출(object detection), 객체분할(object segmentation)의 세 부류로 나뉜다. 객체분류는 주어진 영상이 어떤 종류(class/category)인지 분류하는 것이고, 객체검출은 주어진 영상 내에 있는 물체들의 위치를 찾고 종류를 판별하는 것이며, 객체분할은 영상내의 물체들의 영역들을 분할하는 것을 뜻한다.

객체분류기로 흔히 사용되는 Convolutional Neural Networks(CNN)은 Lecun[1]이 이전의 신경망인 완전연결(fully connected) 신경망의 단점을 보완하여 네트워크 층수가 많은 신경망의 학습을 가능하게 하는 CNN을 제안하여 필기체 우편번호 인식에서 기존 신경망보다 우수한 인식 성능을 보여주었다. 이 후 보다 깊은 CNN인 AlexNet

[2]이 획기적인 영상분류 성능을 보여주면서 딥러닝 연구가 폭발적으로 일어나 ResNet[3], VGG16[4]같은 보다 발전된 CNN 구조가 발표되었다. 이러한 CNN을 기반으로 다양한 객체검출 방법이 제시되었다.

객체검출은 영상 내에 있는 각 객체의 위치를 객체를 감싸는 바운딩 박스(bounding box)로 찾고 객체 종류(class)를 분류한다. 객체검출 방법은 Region Proposal Network (RPN)[5]을 사용하여 객체후보영역을 추출한 후 이 영역들에 대해 분류넷 (classification net)와 회귀넷(regression net)을 통하여 객체의 종류를 분류하고, 위치를 찾아내는 2단계 방법(Two-stage method)과 RPN없이 정해진 위치와 크기를 기반으로 하여 객체의 위치를 찾고 분류하는 1단계 방법(One-stage method)으로 크게 나뉜다. 대표적인 2단계 방법으로는 Faster RCNN[5]이 있고, 1단계 방법으로는 Yolo[6], SSD[7], RetinaNet[8] 등이 있다. [9]는 Faster RCNN과 Yolo의 성능 비교를 보여준다. 일반적으로 2단계 방법이 1단계 방법보다 인식 성능이 더 우수하지만 속도가 느린 단점이 있다. 1단계 방법은 극심한 전경/배경(foreground/background) 클래스 불균형문제로 인해 성능이 2단계방법보다 뒤떨어지는 것으로 알려져 있다.

하지만 최근에 발표된 RetinaNet은 많은 배경샘플들의 loss는 낮춰서 개수가 훨씬 적은 전경/객체(foreground/object) 샘플들의 loss에 집중할 수 있게 하는 focal loss를 이용한 적절한 학습이 가능하게 하여 Faster RCNN 같은 최신의 2단계 방법보다 동등하거나 더 나은 성능을 내면서 속도가 빠른 장점이 있다.

RetinaNet은 ESRI Object Detection Challenge 2019, NATO Innovation Challenge 같은 최근의 경연대회에서 우승한 네트워크로서 현재 성능이 가장 우수한 객체검출기 중 하나로 손꼽힌다. 논문으로는 RetinaNet이 도로파손 검출[10], 도로면 마킹(marking) 검출[11], 선박 검출[12], CT lesion 검출[13] 등을 비롯한 여러 응용에서 성공적으로 적용되고 있지만, PCB 부품 검출에 적용된 적은 거의 없다.

본 논문에서는 컬러 입력 영상으로부터 PCB에 실장된 각 부품에 대해서 종류와 위치를 자동으로 검출하는 확장된 RetinaNet을 제안한다. 확장된 RetinaNet은 기본 RetinaNet의 특징피라미드(feature pyramid)에 한 단계 고해상도 특징맵(feature map)을 추가하여 만들어진다. 이를 이용하면 각 부품에 대한 검사 영역을 사용자의 입력 없이 자동 설정할 수 있게 되어 사용자의 편의성이 대폭 향상될 수 있다.

2. RetinaNet [8]

RetinaNet은 백본넷(backbone network)와 2개의 서브넷(subnet)으로 구성된다. 백본넷은 입력 영상 전체에 걸쳐서 convolutional feature map을 계산하는 역할을 하며, 백본넷 출력으로부터 첫 번째 서브넷은 객체분류를 하고, 두 번째 서브넷은 bounding box regression을 하는 넷이다.

RetinaNet의 백본넷으로 Feature Pyramid Network (FPN)[14]이 사용된다. FPN은 한 해상도의 입력영상으로부터 convolutional 다중스케일 특징 피라미드(feature pyramid)를 효율적으로 생성한다. 각 피라미드 레벨은 다른 스케일에서 객체를 검출하는데 사용된다. 기본 RetinaNet은 ResNet이나 VGG16같은 CNN 구조 위에 P3~P7 레벨을 갖는 FPN를 구성한다. (여기서 Pk는 입력영상보다 2^k 낮은 해상도를 가진다.) Fig. 1에 ResNet FPN 백본 구조의 RetinaNet을 보인다.

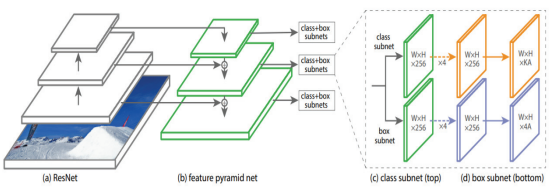

Fig. 1. RetinaNet network architecture with ResNet FPN backbone [7].

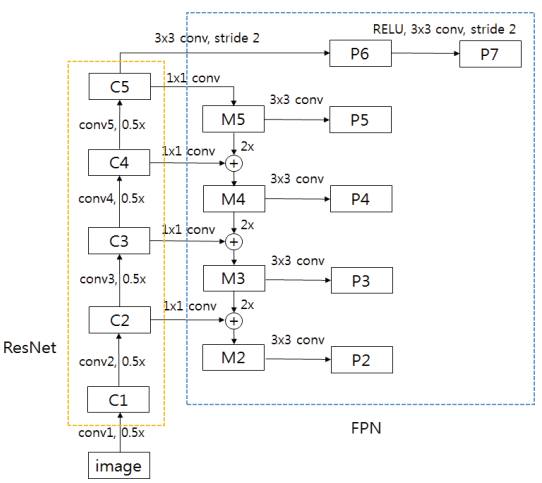

Fig. 2. FPN with P2 added to the basic RetinaNet.

앵커(Anchors)는 기본 RetinaNet 구조에서는 피라미드 레벨 P3~P7에서 32^2 ~ 512^2영역을 가진다. 각 피라미드 레벨에서 기본 aspect ratios {0.5, 1, 2}와 크기 $\{2^0, 2^{1/3}, 2^{2/3}\}$에서 앵커가 생성된다. 각 레벨마다 앵커의 개수 A는 9가 되고, 전체로는 입력영상에 대해서 32-813 픽셀까지의 스케일 범위를 갖게 된다. 한 단계 고해상도 피라미드레벨 P2을 추가하면 피라미드 레벨 P2~P7에서 16^2-512^2영역을 가지게 되어 기본 RetinaNet보다 검출 가능 최소 크기가 1/2로 작아져 보다 작은 부품들을 검출할 수 있다. P2를 추가한 FPN을 Fig. 2에 보인다.

Classification 서브넷은 A개의 앵커와 K개의 객체 클래스의 각각에 대해 각 공간위치에서 객체 존재 확률을 예측하는 Fully Convolutional Network(FCN)이다. 3x3 convolution (256 filters, ReLU)를 네 번 적용 후 3x3 convolution(KA filters, sigmoid)을 적용하여 KA개의 출력을 얻는다.

Box regression 서브넷은 각 피라미드 레벨에서 각 앵커 박스로부터 근처의 ground-truth 객체로 box regression하는 FCN이다. 구조는 classification 서브넷과 같지만, 4A개의 선형 출력을 낸다는 것이 다르다.

Classification 서브넷과 box regression 서브넷은 같은 구조를 가지지만 각각 독립적인 파라미터를 갖는다. 다만, 두 서브넷의 웨이트(weights) 파라미터는 모든 피라미드 레벨에 걸쳐서 공유된다.

3. 구현 및 실험

실험에 사용할 영상을 취득하기 위해서 같은 종류의 PCB를 24장 사용하였다. 하나의 PCB는 부품이 실장된 총 9개의 모듈(M0~M8)로 구성되어 있는데 M3~M5은 M0~M2 모듈이 180도 회전된 배열이고, M6~M8은 M0~M2 모듈과 같은 부품 배열을 갖고 있다. Fig. 3에 PCB 샘플을 보인다.

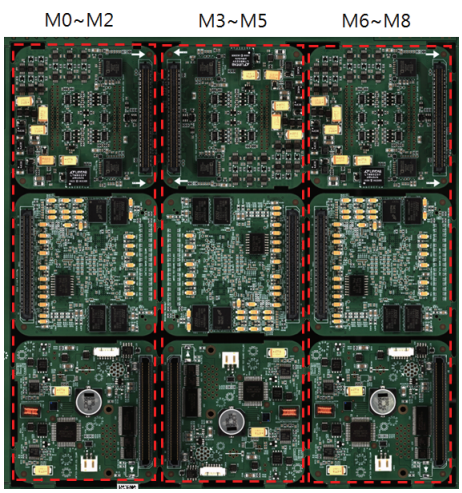

Fig. 3. A PCB sample.

M0~M2 모듈을 촬상한 영상은 학습세트로, 나머지 M3~M8 모듈 촬상 영상은 테스트세트로 사용하였다. M0~M2 모듈 전체 영역을 커버하기 위해 11개의 다른 위치에서 영상을 취득하였다. 비슷하게 M3~M8 모듈 촬상으로 22개의 영상을 얻었다. 따라서, 학습세트는 24x11=264, 테스트세트는 24x22=528개의 영상으로 구성된다. 각 영상은 3904x3904크기의 고해상도 컬러 영상이다. 검출할 부품의 종류는 모두 53개이다. 영상에서 부품의 크기는 픽셀 기준 40x20 ~ 880x920까지 다양하다. 하나의 영상 안에는 많게는 수백 개의 작은 부품이 있다. 크기가 작은 부품들은 주로 표면 실장 부품 저항, 커패시터 같은 종류이고 큰 부품은 IC, 용량이 큰 전해 커패시터 같은 부품이다. 학습세트는 부품 종류별로 작게는 24개, 많게는 약 15,700개의 부품을 가지고 있다. Fig. 4에 촬상된 영상 예를 보인다.

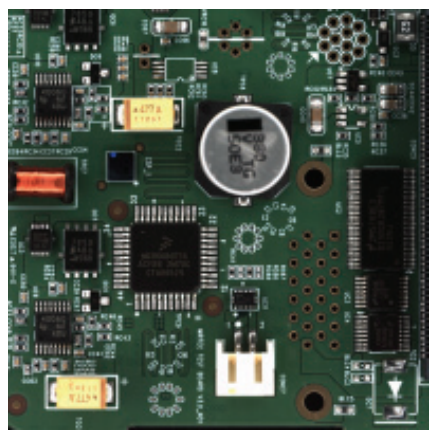

Fig. 4. An example of a captured image.

각 부품의 종류(class)와 부품 영역을 표시하는 bounding box 좌표를 기록하는 annotation 툴로 LabelImg[15]을 사용하였다. Fig. 5에 annotation 된 예를 보인다.

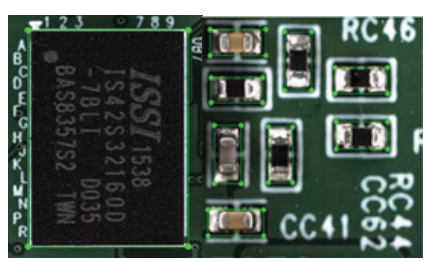

Fig. 5. Examples of annotation boxes.

실험에 사용된 PC는 Intel Core i7 CPU (4GHz, 16GB)와 GPU보드로 Nvidia GeForce GTX 1080ti (11GB)에 운영체제 Windows 10 Pro (64bit)가 설치된 것이다. 딥러닝 프레임워크로 Anaconda 3, Tensorflow, Keras를 사용하였다. RetinaNet은 Keras 구현 버전으로 오픈소스이나 유지, 보수가 활발히 이루어져 많은 응용에서 사용되는 keras-retinanet[16]을 기본으로 수정하여 사용하였다. 백본넷은 ResNet50 FPN을 채택하였다.

RetinaNet의 학습 및 테스트 시 GPU 메모리 부족과 긴 학습 및 처리 시간으로 입력 영상의 크기는 원 영상의 크

기 3904x3904에서 1600x1600 이하로 축소하였다.

다양한 크기의 부품들을 검출하기 위해 앵커 스케일과 aspect ratios 파라미터를 적절히 설정해야 한다. 가장 작은 크기 (40x20)의 부품을 검출하기 위해 디폴트 값인 $\{2^0, 2^{1/3}, 2^{2/3}\}$ 대신 {0.5, 1, 1.5}를 사용하고, {0.5, 1, 2}의 기본 aspect ratios를 사용했다.

학습은 10 epochs (5000 steps/epoch)정도로 수행하여 classification loss가 0.1이하, box regression loss가 0.2 근처에 이를 때까지 진행하였다. 학습 시 과최적화(overfitting)을 줄이기 위해 각 학습 입력 영상은 수평/수직 영상 flip, 이동(+-10%), 회전(+-10도), 스케일링(+-10%)등에 의한 랜덤 변환(random transform)을 수행하였다.

학습 완료 후 테스트 셋의 한 영상에 대한 부품 검출 예를 Fig. 6에 보인다. 녹색으로 그린 annotated box와 매우 근사하게 검출된 부품 경계가 표시됨을 볼 수 있다. (작은 부품의 경우 검출된 경계와 annotated box와 겹치는 픽셀들은 annotated box 컬러인 녹색으로 보인다.)

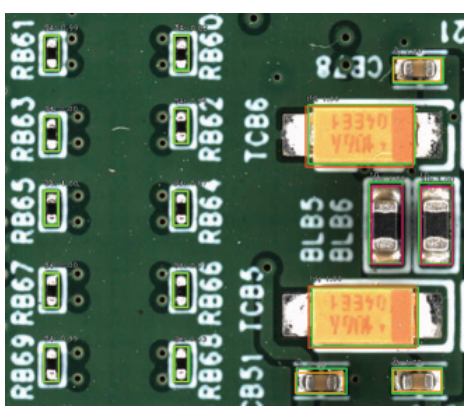

Fig. 6. Examples of component detections (a sub image). Annotated boxes are colored in green and detected boxes are colored in the color assigned to the class number.

객체검출기의 성능은 흔히 mAP(mean average precision)[17]으로 나타낸다. mAP는 각 클래스별 AP값들의 평균으로 구하는데, 각 클래스에 대한 AP는 해당 클래스에 대한 precision-recall 곡선의 아래 면적으로 구한다. precision과 recall은 아래와 같이 구해진다.

$$\text{precision} = \#\text{true positives} / \#\text{all positive detections}$$
$$\text{recall} = \#\text{true positives} / \#\text{all ground-truth boxes}$$

보통 ground-truth bounding box B_{gt}와 예측된 bounding box B_p의 IoU(Intersection over Union)>0.5이면 true positive로 판단하고, 그렇지 않으면 false positive로 판단한다. 여기서 IoU는 B_{gt}와 B_p의 겹치는 영역 넓이 / $B_{gt} \cup B_p$ 영역 넓이로 주어진다.

학습된 RetinaNet의 성능을 평가하기 위해서 테스트셋에 대해서 mAP와 추론시간(inference time)을 측정하였다. Table 1에 디폴트 스케일과 {0.5, 1, 1.5} 설정시의 mAP 비교를 보인다. 원 영상 크기 3904x3904를 1600x1600으로 축소하여 학습하였지만 앵커스케일을 {0.5, 1, 1.5}를 사용하는 것이 기본값 $\{1, 2^{1/3}=1.26, 2^{2/3}=1.59\}$으로 설정하는 것보다 더 우수한 검출 성능을 주는 것을 알 수 있다. 이는 가장 작은 크기(40x20)의 부품들을 정확하게 검출했기 때문이다.

Table 1. mAPs for the test set and inference time per image for different anchor scales with random transform

input size	scales	mAP	inference time [ms]
1600x1600	0.5, 1, 1.5	0.998	680
1600x1600	1, 1.26, 1.59	0.950	690

Random transform 없이 학습한 경우의 mAP는 Table 2에 보이는 바와 같이 random transfom에 의한 data augmentation을 이용해서 학습한 경우보다 좀떨어졌다. 과최적화로 인한 성능 하락으로 보인다.

Table 2. mAPs for the test set with/without random transform (scales {0.5, 1, 1.5})

input size	random transform	mAP
1600x1600	yes	0.998
1600x1600	no	0.975

Table 3. mAPs for the test set and inference time per image with/without P2 (random transform, scales {0.5, 1, 1.5})

input size	P2	mAP	inference time [ms]
800x800	no	0.957	560
800x800	yes	0.998	630

좀 더 작은 크기의 부품을 검출하기 위해서는 앵커스케일 설정만으로는 부족하다. 원영상(3904x3904)에서 가장 작은 부품은 40x20이므로 레티나넷 입력 크기1600x1600으로 축소되면 16x8 크기가 되어, 해상도가 1/8로 줄어드는 P3에서는 2x1 크기가 되어 기본 레티나넷으로 검출이 되지만 좀 더 작은 부품은 기본 레티나넷으로 검출되기 어렵다.

좀 더 작은 부품을 검출하기 위해서는 기본 RetinaNet FPN의 P3~P7에 고해상도 피라미드레벨 P2를 추가하여 P2~P7의 FPN을 사용하는 것이 필요하다. P2를 추가하면 검출 가능 부품의 최소크기가 1/2로 줄어들 것으로 예상할 수 있다. 좀 더 작은 크기의 부품을 검출할 수 있는지 보기 위해, 입력영상을 800x800으로 좀 더 줄여서 학습시

켰다. Table 3에서 알 수 있듯이, P2를 추가하지 않고 원 피라미드 P3~P7를 사용했을 때 mAP는 0.957이었고 P2를 추가해서 P2~P7을 사용했을 때 mAP는 0.998로 1600x1600, P3~P7의 경우와 비슷하게 우수하였다. 추론시간은 P2 추가로 인해 13% 정도 약간 길어졌다. 하지만 P2를 추가했음에도 불구하고 입력 영상크기 1600x1600, P2 없는 기본 RetinaNet보다는 입력영상의 크기가 반으로 줄기 때문에 추론속도가 약간 빠른 것을 알 수 있다. 따라서, P2가 추가된 레티나넷은 1600x1600으로 학습시, 3904x3904크기의 원영상에서 크기 20x10의 작은 부품도 성공적으로 검출할 수 있을 것이다.

4. 결 론

본 논문에서는 부품이 실장된 PCB의 컬러영상으로부터 각 부품의 종류 및 위치, 검사 영역을 최신 딥러닝 객체검출 넷인 RetinaNet을 이용하여 자동으로 추출하는 방법을 제안하였다. 기존의 feature pyramid P3~P7에 P3보다 한 단계 고해상도 feature map인 P2를 추가하여 1/2배 더 작은 크기의 부품 검출이 가능함을 보였다. 이를 이용하면 검사 장비의 사용자 편의성이 크게 향상되고, 초기 셋업 시간이 크게 단축될 것으로 보인다.

감사의 글

이 논문은 2019년도 한국기술교육대학교 교수 교육연구진흥과제 지원에 의하여 연구되었음.

참고문헌

1. Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-Based Learning Applied to Document Recognition,” Proceedings of the IEEE, vol. 86, no. 11, pp. 2278-2324, Nov. 1998.
2. A. Krizhevsky, I. Sutskever, G.E. Hinton, “ImageNet classification with deep convolutional neural networks,” Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1, pp. 1097-1105, Dec. 2012.
3. K. He, X. Zhang. S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” in Proceeding of 2016 IEEE Conference on Computer Vision and Pattern Recognition, pp. 770-778, 2016.
4. K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in Proceeding of the International Conference on Learning Representations (ICLR), 2015.
5. S. Ren, K. He, R. Girshick, and 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, June 2017.
6. J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in Proceedings of the 2016 IEEE International Conference on Computer Vision, pp. 779-788, 2016.
7. W. Liu, D. Anguelov, D. Erhan, C. Szegedy, and S. Reed, “SSD: Single shot multibox detector," in Proceedings of ECCV, 2016.
8. T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, “Focal loss for dense object detection,” in Proceeding of the 2017 IEEE International Conference on Computer Vision, Venice, Italy, pp. 2999-3007, 2017.
9. Y.-H. Lee and Y. Kim, “Comparison of CNN and YOLO for Object Detection,” Journal of the Semiconductor and Display Technology, vol.19, no.1, pp. 85-92, 2020.
10. L. Ale, N. Zhang, and L. Li, “Road damage detection using RetinaNet,” in Proceedings of 2018 IEEE International Conference on Big Data (Big Data), pp. 5197-5200, 2018.
11. T.M. Hoang, P.H. Nguyen, N.Q. Truong, Y.W. Lee and K.R. Park, “Deep RetinaNet-based detection and classification of road markings by visible light camera sensors,” Sensors, vol.19, no.2, 2019.
12. Y. Wang, C. Wang, H. Zhang, Y. Dong, and S. Wei, “Automatic ship detection based on RetinaNet using multi-resolution Gaofen-3 imagery,” Remote Sensing, vol.11, no.5, 2019.
13. M. Zlocha, Q. Dou, and B. Glocker, “Improving RetinaNet for CT lesion detection with dense masks from weak RECIST labels,” in Proceedings of International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI), Oct. 2019.
14. T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, “Feature pyramid networks for object detection,” in Proceeding of the 2017 IEEE International Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, pp. 936-944, 2017.
15. LabelImg : a graphical image annotation tool and label object bounding boxes in images [Internet]. Available: https://github.com/tzutalin/labelImg.
16. Keras implementation of RetinaNet object detection [Internet]. Available: https://github.com/fizyr/keras-retinanet.
17. E. Mark, L.V. Gool, C.K.I. Williams, J. Winn, and A. Zisserman, “The PASCAL Visual Object Classes (VOC) Challenge,” International Journal of Computer Vision, vol. 88, pp. 303-338, 2010.

접수일: 2020년 4월 20일, 심사일: 2020년 6월 8일,
게재확정일: 2020년 6월 11일