

# PCB 검사를 위한 YOLO 네트워크 기반의 PCB 부품 분류 알고리즘

윤형조<sup>†</sup>, 이준재<sup>††</sup>

## PCB Component Classification Algorithm Based on YOLO Network for PCB Inspection

HyungJo Yoon<sup>†</sup>, JoonJae Lee<sup>††</sup>

### ABSTRACT

AOI (Automatic Optical Inspection) of PCB (Printed Circuit Board) is a very important step to guarantee the product performance. The process of registering components called teaching mode is first perform, and AOI is then carried out in a testing mode that checks defects, such as recognizing and comparing the component mounted on the PCB to the stored components. Since most of registration of the components on the PCB is done manually, it takes a lot of time and there are many problems caused by mistakes or misjudgement. In this paper, A components classifier is proposed using YOLO (You Only Look Once) v2's object detection model that can automatically register components in teaching modes to reduce dramatically time and mistakes. The network of YOLO is modified to classify small objects, and the number of anchor boxes was increased from 9 to 15 to classify various types and sizes. Experimental results show that the proposed method has a good performance with 99.86% accuracy.

**Key words:** PCB Inspection, Defect Detection, AOI, Classification, Deep Learning, CNN

### 1. 서 론

표면실장기술(Surface Mount Technology)은 인쇄회로기판(Printed Circuit Board) 위에 납을 도포하고 전자부품을 장착하는 일련의 과정이다. PCB 위에 부착되는 표면 실장 기기의 총칭을 SMD(Surface Mount Device)라고 한다. PCB 검사는 크게 부품을 실장하기 전에 도포된 납의 위치나 크기 및 체적 등 형상을 검사하는 SPI(Solder Paste Inspection)의 사전검사와 부품을 장착한 후에 부품의 불량률 검사를 하는 AOI(Area Of Inspection)의 사후검사로 나눌 수

있다[1]. AOI 검사는 반도체 부품이 소형화, PCB 미세공정의 증가와 함께 제품의 성능을 보장하기 위한 매우 중요한 단계이다. AOI 검사항목 중 가장 기본적인 것은 특정 부품이 제 위치에 장착되었는지를 검사하는 것이다. 이를 위해서는 검사 전에 특정 위치에 실장되는 SMD의 사양을 미리 등록하는 과정인 학습모드(teaching mode)가 필요하고, 검사 시(testing mode)에는 SMD를 인식하여 저장된 부품과 비교하여 불량 여부를 판별한다[2]. 새로운 PCB가 도입될 경우, PCB 위에 실장된 부품을 등록하는 일은 현재 대부분의 검사기에서 수작업에 의존하고

※ Corresponding Author : JoonJae Lee, Address: (42601) 1095 Dalgubeol-daero, Dalseo-gu, Daegu, Republic of Korea, TEL : +82-53-580-6682, E-mail : joonlee@kmu.ac.kr

Receipt date : Jun. 10, 2021, Revision date : Jul. 29, 2021  
Approval date : Aug. 6, 2021

<sup>†</sup> Dept. of Computer Engineering, Graduate School, Keimyung University (E-mail : wlvkcms@naver.com)

<sup>††</sup> Faculty of Computer Engineering, Keimyung University (E-mail : joonlee@kmu.ac.kr)

※ This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2020R1I1A3074667)"

있어 시간소모가 매우 많다. 또한 해당 부품의 상세 명칭을 사람이 수작업으로 확인하므로 작업자의 오판이나 실수 등으로 문제가 생길 수 있다.

따라서 본 논문에서는 PCB 위에 실장된 부품의 종류를 자동으로 인식 및 분류하여, 이를 데이터베이스에 저장하고, 새로운 PCB가 들어올 경우, 일일이 각 부품 명칭의 등록을 수작업에 의존하지 않고, 자동으로 인식 분류하여 등록하는 방법을 딥러닝 알고리즘으로 제안한다. 이렇게 함으로써 PCB 등록 과정 시 작업시간을 대폭적으로 줄이고 사람의 오판이나 실수 같은 문제를 방지하여, 검사 효율을 높일 수 있다. 이와 같은 장점들 때문에 부품을 미리 등록하는 라이브러리화 작업이 PCB 제조 공정에서 필요하다. 본 논문에서는 딥러닝 알고리즘을 활용하여 학습모드에서 부품의 라이브러리화를 자동으로 등록하는 방법을 제안한다[3-4]. 이는 후에 검사 모드에서 특정 위치의 부품의 종류를 분류하고 해당 부품이 설계 과정에서 등록된 부품과 동일한지를 비교하는 방식으로 부품의 오장착을 검사하는 AOI 검사 알고리즘에 연동되는 매우 중요한 과정이다[5-8].

PCB에 장착되는 부품은 일반적으로 레지스터(resistor), 커패시터(capacitor), 인덕터(inductor), 가변저항(potentiometer), 변압기(transformer), 다이오드(diode), 트랜지스터(transistor), 사이리스터(thyristor), 집적 회로(IC), 발진기(oscillator), 스위치(switch), 센서(sensor) 등 크게 12가지의 부품이 사용되며 이들 중에서 가장 많고, 수집이 용이한 커패시터, 다이오드, 레지스터, 트랜지스터의 4가지 대분류 내에서 먼저 수행한다[5,9]. 수집에 많은 시간을 요하면서 많은 비중을 차지하지 않고, 또한 비교적 분류가 쉬운 다른 부품들은 점차적으로 수집하여 추후 분류할 예정이다. 4가지의 부품의 분류 방법은 중분류 소분류로 나눌 수 있다. 중분류로 나누는 기준은 부품의 종류 내에서 부품의 크기이다. 부품의 종류와 가로, 세로 길이를 기준으로 부품의 클래스를 나누어 분류한다. 소분류의 경우는 부품의 종류, 가로, 세로 길이뿐만 아니라 부품의 높이를 이용해서 더 세분화하여 분류한다. 부품의 중분류는 각 종류에 따라 가로(width) × 세로(height)이며, 소분류는 가로 × 세로 × 종류 × 높이(depth)로 분류한다. 소분류의 경우는 부품의 가로, 세로가 거의 비슷해서 높이 정보로만 구분이 가능하다. 즉 RGBD의 4채널 영상

정보가 필요하다. 그러나 제공받은 데이터가 RGB의 3채널 정보만 있어 본 논문에서는 중분류로 부품을 분류하는 방법을 먼저 진행하고[5-6], 추후 높이가 정보가 수집되는 대로 소분류의 부품을 분류하는 방법을 연구할 것이다.

중분류로 부품을 분류하기 전 부품의 클래스 명칭을 정한다. 클래스의 명칭은 부품의 가로, 세로, 부품의 종류를 이용해 명명한다. 예를 들어 부품이 커패시터이고 가로 길이가 10 um, 세로 길이가 5 um이면 1005-C 클래스가 된다. 클래스의 앞의 2개의 숫자는 가로의 길이이고, 뒤의 2개의 숫자는 세로 길이, 뒤의 알파벳은 부품의 종류의 첫 알파벳이다. 이런 방법으로 클래스의 명칭을 정하며 자세한 클래스 이름은 Table 1과 같다. Table 1의 마지막 칼럼은 PCB 제조사로부터 수집한 데이터의 개수이다. 여기서 총 23개의 클래스에서 수집된 데이터 수가 적은 아래 0402-D 등 4개의 클래스는 실험에서 제외하고 추후 충분한 양의 데이터가 수집되는 대로 분류에 추가할 예정이다.

본 논문에서 활용한 CNN 관련 알고리즘은 YOLO(You Only Look Once)이다. YOLO는 다양한 버전이 존재하며, 버전이 높을수록 이전 버전의 단점을 극복한 모델로 알려져 있으나 본 논문에서는 YOLO v2를 기반으로 하는 모델을 제안한다. CNN 기반의 물체 검출 알고리즘의 단점은 작은 물체 검출에 있어 좋은 성능을 가지지 못한다는 것이다. 이러한 단점을 네트워크를 수정해 작은 물체 검출의 성능을 높인다. YOLO는 앵커박스(anchor box)라는 물체 검출 경계 박스를 사용해 물체의 크기를 검출한다[6-8]. 본 논문에서는 색상 및 크기 변화 등 다양한 부품들에 대해 대응하기 위해 앵커박스의 개수를 조절하고 앵커박스의 크기 역시 수집한 데이터에 맞게 수정해 더 개선한 모델을 제안한다.

본 논문에서 제안한 주요 기여내용을 요약하면 다음과 같다. 첫째, 부품의 다양한 색상과, 약간의 회전 변화에 대해서 강건한 분류특성을 갖도록 영상을 수집 및 처리한다. 둘째, 분류성능을 높이기 위해 부품 간의 구별되는 특성을 고려하여 물체의 경계영역을 정한다. 세 번째, 물체의 검출 및 분류를 동시에 수행하는 모델이면서 크기 변화에 의존적인 YOLO v2를 선택하여, 부품 모양에 적합한 앵커박스를 만들고, 네트워크를 수정하여 개선함으로써 작은 부품도 검출할 수 있도록 시스템의 분류성능을 높인다. 이렇게

함으로 제안한 방법은 실험결과 17개의 클래스에 대해 99.86%의 높은 성능을 보여준다.

## 2. 부품 획득 및 분류

### 2.1 부품 획득 및 수집

부품의 등록과 검사에서 사용되는 부품 영상은 PCB 전체를 획득한 영상이 아닌 Fig. 1처럼 PCB 영상에서 검사할 부품의 영상을 따로 잘라내어 사용한다. 검사할 부품을 중심으로  $700 \times 700$ (pixel) 크기의 영상을 사용하고 영상 당 가운데 하나의 부품만을 검사한다.

PCB에 장착되는 부품들은 PCB 설계에 따라 가로나 세로로 장착된다. 학습모드나 검사모드에 사용되는 부품의 영상은 모두 부품의 가로 길이가 세로 길이보다 긴 상태가 되도록 회전시키고, 트랜지스터같이 리드(부품의 다리)가 위아래로 붙어있고 리드의 개수가 다양한 부품의 경우에는 부품의 리드가 적은 부분이 위로 향하게 회전시켜 부품의 형태를 동일하게 변경하여 사용한다.

$700 \times 700$  영상을 사용하는 이유는 사용되는 영상이  $700 \times 700$  크기일 때, 1pixel 당 실제 크기는 7  $\mu$ m 이고 이러한 분해능으로 영상을 사용하면 검사를 진

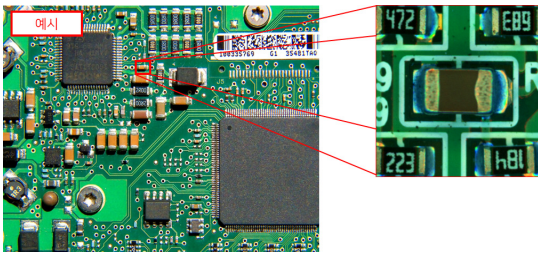


Fig. 1. Acquired image of PCB component.

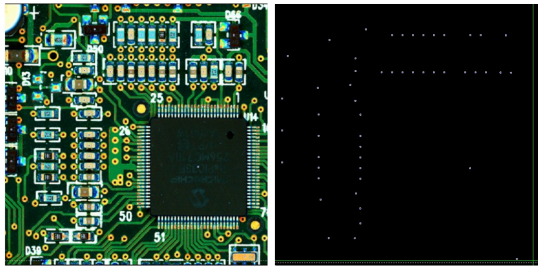


Fig. 2. PCB images and images showing PCB component locations.

행하는 부품 중  $700 \times 700$ 보다 큰 부품이 없기 때문에 이 크기의 영상을 사용한다. PCB에서 각각의 부품들을 기준으로 영상을 잘라서 사용할 수 있는 이유는 Fig. 2처럼 설계단계에서 완성된 설계도를 기준으로 PCB에 부품들의 장착될 위치를 알 수 있기 때문에 해당 정보를 활용해서 부품을 중심으로 영상을 획득할 수 있다[10].

### 2.2 부품 종류 분석 및 유형 분류

부품의 라이브러리화를 위해 데이터의 종류나 각 데이터들 간의 특징을 분석한다. 본 논문에서 사용되는 부품의 종류는 커패시터, 다이오드, 레지스터, 트랜지스터로 4가지로 구성되어 있으며, 각 종류에서 부품의 크기에 따라 부품의 종류가 세분화되어 있다. Fig. 3은 각 종류별 부품의 예시이며, 부품의 종류에 따라서 부품의 모양이나 크기 등이 다양하다. 모양이나 크기, 색상 등의 특징들은 딥러닝 알고리즘을 사용할 때 각 부품들의 종류를 구분할 수 있는 특징들이므로 각 종류별 리드의 개수나 납땜의 색상, 부품 색상 등의 특징들 위주로 분석을 하여 ROI 영역을 정한다.

커패시터의 경우는 부품의 색은 다양하지만 부품의 양쪽 끝에 납땜의 흔적이 있는 것이 특징이며, 부품 안에 문자는 존재하지 않는다. 레지스터의 경우는 Fig. 4처럼 커패시터와 약간 다르게 납땜의 흔적이 있다. 부품의 크기가 작은 부품들에서는 문자가 존재

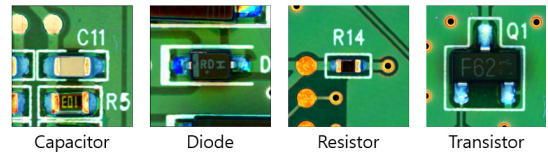


Fig. 3. Types of components.

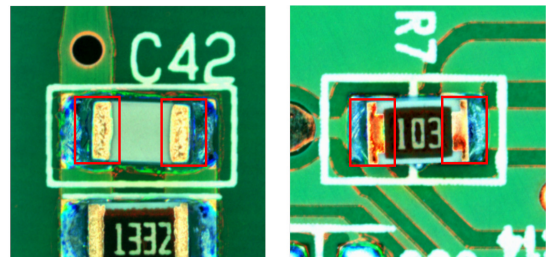


Fig. 4. Soldering region of capacitor (left) and resistor (right).

하지 않지만 부품의 크기가 일정 크기 이상부터는 부품 안에 문자가 존재하며 대부분 숫자가 존재한다.

다이오드의 경우는 Fig. 5처럼 크기가 작은 부품의 경우는 기울어진 채로 장착되어 색상이 실제 부품 색상과 다르게 영상이 찍히는 경우도 존재한다. 이러한 경우의 부품도 같은 다이오드도 인식할 수 있도록 분류해야 한다. 가장 큰 특징은 모든 부품마다 문자가 존재하며 그 문자 안에 세로로 된 긴 줄이 존재한다. 또한 크기가 작은 부품에는 없지만 크기 큰 부품에는 리드라는 것이 양쪽으로 존재한다. Fig. 6과 같이 다이오드에는 극성이 존재하지만 극성의 형태는 한 줄로 된 경우도 있고 여러 줄인 경우 등 다양하다.

트랜지스터의 경우, 문자들은 존재하지만 각 부품마다 제각각이다. 다른 부품들과 구분이 되는 가장 큰 특징은 납땜의 위치와 개수이다. 트랜지스터의 납땜은 위아래로 존재하며, 납땜의 개수 역시 다양하다. Fig. 7처럼 트랜지스터는 납땜이 3개나 6개 등 납땜의 개수가 다른 부품에 비해 많고 납땜의 위치가 위아래에 위치한다.

부품의 종류와 가로, 세로 길이를 기준으로 부품의 클래스를 나누어 분류하며 가로 × 세로 × 종류로 클래스를 명명한다. 중분류는 Table 1처럼 총 21개의 클래스로 분류되며, 각 종류별로 커패시터는 7가지, 다이오드는 4가지, 레지스터는 6가지, 트랜지스터는

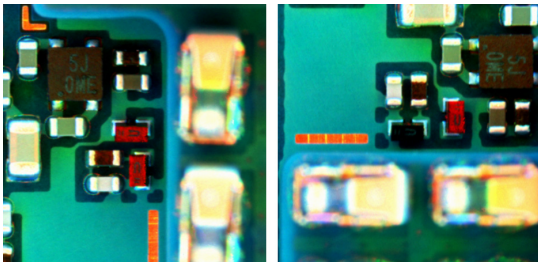


Fig. 5. Diode with different colors.

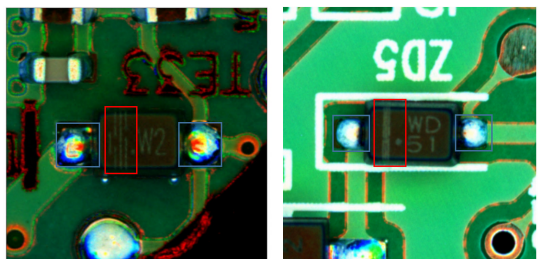


Fig. 6. String inside diode.

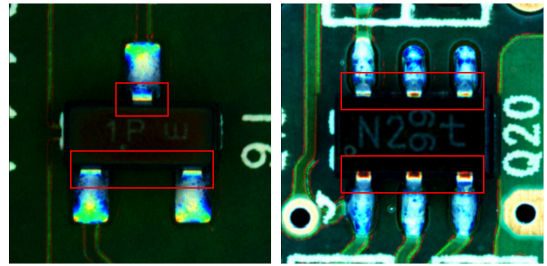


Fig. 7. Features of transistor.

4가지의 크기로 분류된다. 모양이나 색상이 비슷하더라도 부품의 크기가 다른 다른 클래스이며, 크기가 같은 부품이라도 부품의 종류가 다른 경우에는 다른 클래스로 분류된다. 클래스를 정의할 때 부품의 크기를 측정하는 단위는 mm이며, 종류, 가로 길이, 세로 길이를 이용하여 클래스를 분류한다. 이렇게 종류만을 분류하는 경우를 대분류, 종류와 크기별로 분류하는 경우는 중분류로 나뉘서 2가지로 분류를 할 수 있다. 대분류는 중분류에서 분류할 때 같은 크기로도 종류가 다른 다른 부품으로 분류함으로 중분류로도 대분류가 가능하므로 본 논문에서는 중분류를 목표로 연구를 수행한다. 부품을 크기별로 나뉘서 클래스를 분류하면 비슷한 크기의 부품들은 한 클래스로 분류될 수도 있고, 같은 클래스의 부품이라도 모양이나 크기, 색상 등이 다른 경우도 많이 존재한다. 이런 편차가 있더라도 부품을 잘 분류하는 것이 본 연구의 목표이다.

클래스의 종류는 Table 1과 같으며, 클래스 이름은 부품의 가로 길이와 세로 길이로 명명하며, 부품의 크기 뒤에 부품의 종류를 붙여서 클래스 이름을 명명한다. 만약 1.0 mm × 0.5 mm 크기의 커패시터 부품이면 해당 부품의 클래스 이름은 1005-C가 된다. 트랜지스터의 경우에는 크기가 같더라도 부품의 리드 개수가 다른 다른 부품이므로 2812-TR-6과 같이 리드의 개수가 3개가 아닌 트랜지스터의 경우에는 클래스 이름 뒤에 리드 개수도 붙여서 명명한다.

### 3. 제안한 방법

#### 3.1 데이터 전처리

학습을 진행하기 전에 데이터 전처리 과정으로 라벨링 작업이 필요하다. 부품의 종류별로 특징이 각각 다르기에 각 범주별로 각각 라벨링을 진행한다. 커패



Table 1. Class name by types.

Large	Medium	Class Name	Data Count	Large	Medium	Class Name	Data Count
Capacitor	C-CHIP_0402Size	0402-C	1000	Resistor	R-CHIP_0603Size	0603-R	1000
	C-CHIP_0603Size	0603-C	986		R-CHIP_1005Size	1005-R	979
	C-CHIP_1005Size	1005-C	1000		R-CHIP_1608Size	1608-R	983
	C-CHIP_1608Size	1608-C	1000		R-CHIP_2012Size	2012-R	1000
	C-CHIP_2012Size	2012-C	1000		R-CHIP_3216Size	3216-R	1000
	C-CHIP_3216Size	3216-C	1000		R-CHIP_3225Size	3225-R	100
	C-CHIP_3225Size	3225-C	1000	Transistor	TR_3PinTR_1816Size	1816-T	505
Diode	DIODE_0603Size	0402-D	110		TR_3PinTR_2012Size	2012-T	58
	DIODE_10065Size	10065-D	972		TR_3PinTR_2812Size	2812-T	1000
	DIODE_1511Size	1511-D	1000		TR_6PinTR_2812Size	2812-T-6	58
	DIODE_2515Size	2515-D	989	Large Class	4 Class	Medium Class	21 Class

시터와 레지스터의 경우는 크기나 모양의 특징이 비슷하지만 다이오드와 트랜지스터는 부품에 리드가 연결되어 있는 특징이 있다. 리드가 연결된 특성은 커패시터와 레지스터를 구분되는 특징이기에 다이오드와 트랜지스터의 경우는 라벨링시 리드 부분이 포함되게 작업을 한다. 커패시터와 레지스터는 다른 부품들과 달리 리드가 없어, Fig. 8처럼 부품 양 끝에 납땜부분까지 ROI 영역을 정해 라벨링을 진행한다.

다이오드의 경우엔 커패시터나 레지스터와는 달리 부품 양옆으로 리드가 연결되어 있다. 이는 다이오드를 커패시터나 레지스터와 구분할 수 있는 특징이기 때문에 리드부분까지 포함해 실제 부품보다 가로 길이가 더 길게 영역을 정해 라벨링한다. 다이오드의 경우는 Fig. 9처럼 실제 부품보다 가로가 더 긴 영역으로 라벨링을 진행한다. 트랜지스터의 경우는 다른 부품들과 달리 납땜이 위아래로 되어 있다. 트랜지스터의 경우는 리드 부분을 다이오드처럼 라벨링하며 Fig. 10처럼 리드 영역까지 라벨링에 포함시킨다.

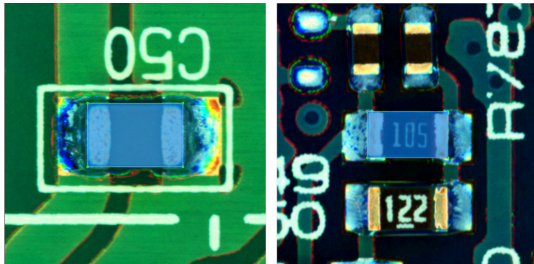


Fig. 8. Labeling image of capacitor (left) and resistor (right).

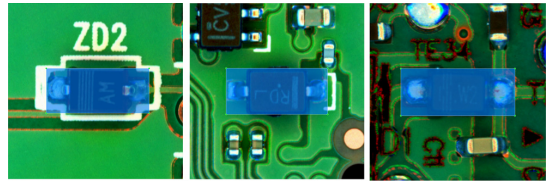


Fig. 9. Labeling images of diode.

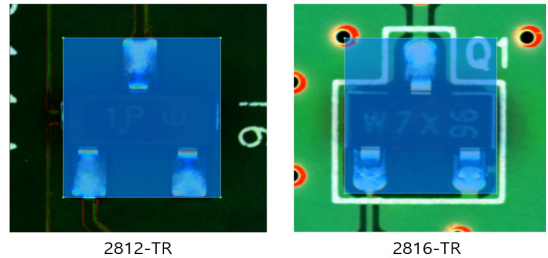


Fig. 10. Labeling image of 2812-TR and 2816-TR.

### 3.2 CNN 모델 선택

부품을 라이브러리화를 위해서는 PCB위에 실장된 부품의 종류를 인식 및 분류해야 한다. PCB에 장착된 하나의 부품은 획득된 영상에서 주위 배경이 다양하기 때문에 영상 분류 모델(image classification model)보다는 가운데 물체를 찾아서 분류하는 물체 검출 모델(object detection model)이 적합하다. 따라서 본 논문에서는 새로운 PCB가 들어올 경우, 일일이 각 부품을 수작업에 의존하지 않고, 자동으로 물체영역을 검출하고 인식 분류하는 방법을 딥러닝 알고리즘으로 제안한다[12]. 물체 검출 모델 중에서도 YOLO는 물체 검출의 성능도 높을 뿐더러 C++로

개발된 모델이며, 기존의 PCB 검사 프로그램 역시 대부분 C++로 개발되어 있어 YOLO의 가중치(weight) 파일을 이식하기 좋은 장점이 있어 해당 논문의 연구에 적합하다.

YOLO의 손실함수는 다음 식 1과 같으며, 식 (1)의 첫 번째 줄부터 경계박스의 중심값  $x, y$ 의 손실, 크기  $w, h$ 의 손실, 해당 경계박스의 confidence score 손실, 물체가 없는 셀의 confidence score 손실, 물체가 존재하는 셀의 클래스별 확률 손실이다.

$$\begin{aligned}
 & \lambda_{coord} \sum_{i=0}^S \sum_{j=0}^B 1_{ij}^{obj} [(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2] \\
 & + \lambda_{coord} \sum_{i=0}^S \sum_{j=0}^B 1_{ij}^{obj} [(\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2] \\
 & + \sum_{i=0}^S \sum_{j=0}^B 1_{ij}^{obj} (C_i - \hat{C}_i)^2 \\
 & + \lambda_{noobj} \sum_{i=0}^S \sum_{j=0}^B 1_{ij}^{noobj} (C_i - \hat{C}_i)^2 \\
 & + \sum_{i=0}^S 1_i^{obj} \sum_{c \in classes} (p_i(c) - \hat{p}_i(c))^2
 \end{aligned} \quad (1)$$

YOLO는  $7 \times 7(S \times S)$ 의 그리드(grid) 셀로 나누고, 나눈 그리드 셀 중 찾고자 하는 물체의 중심이 있는 셀을 찾고 해당 셀에서 앵커박스라는 경계 박스를 이용해서 물체의 크기를 찾고, 찾은 물체를 신뢰 지수(confidence score)를 통해 물체의 유무를 결정하고, 물체가 존재하면 소프트맥스(softmax) 알고리즘으로 클래스를 결정하는 방식으로 물체를 찾는다[6]. YOLO의 모델은 v1~v4까지 있으며 버전이 높아질수록 성능이 좋다고 알려져 있다. 하지만 해당 논문에서 찾고자 하는 부품들은 부품의 크기에 따라 해당 클래스도 달라지기 때문에 상대적으로 다른 모델들과 달리 크기에 어느 정도 의존적인 YOLO v2를 선택한다[11-12].

YOLO v3부터는 네트워크 구조에서 특징 피라미드(feature pyramid)라는 새로운 구조가 추가되었다. 이는 물체를 찾을 때 네트워크의 최종 출력에서 물체를 검출하는 것이 아니라 즉, 네트워크의 최종 특징맵에서만 물체를 검출하지 않고, 네트워크의 중간 특징맵을 활용하여 여러 크기의 특징맵에서 물체를 검출하는 구조의 네트워크이다. 특징 피라미드는 여러 크기의 특징맵에서 물체를 찾는 구조이다[8]. 특징맵의 특성상 특징맵이 클수록 작은 물체를 잘 찾고 특징맵이 작을수록 큰 물체를 잘 찾는다[12]. Fig. 11은

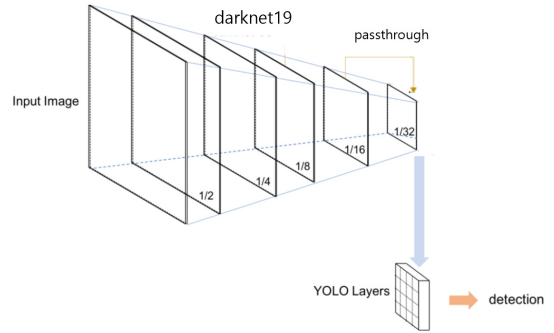


Fig. 11. Network structure of YOLO v2.

YOLO v2의 네트워크 구조로 물체를 검출은 컨볼루션(convolution)을 거친 가장 마지막 특징맵에서 진행한다[7]. 이와는 다르게 YOLO v3부터는 Fig. 12처럼 특징 피라미드 구조에서는 마지막 특징맵에서만 물체를 검출하지 않고 여러 크기의 특징맵에서 물체를 검출한다. 이와 같은 구조 때문에 같은 클래스의 물체의 크기가 다른 경우에도 같은 클래스로 잘 찾게 되는 것이다. 본 연구에서는 모양이 비슷하더라도 크기가 다르면 다른 클래스로 분류해야 하므로 이런 특징 피라미드가 있는 YOLO v3부터는 본 연구에 적절하지 않다. 본 논문에는 Fig. 15처럼 색상이나 모양이 비슷하더라도 부품의 크기가 다른 부품들이 다수 존재하며, 이런 경우는 모두 다른 클래스이다. YOLO의 성능은 버전이 높아질수록 좋다고 알려져 있지만 특징 피라미드가 있는 YOLO v3, v4보다는 특징 피라미드가 없는 YOLO v2가 본 연구에서는 더 적합하다[12-13].

Fig. 13을 보면 왼쪽이 0603-C 부품, 중간이 1005-C, 오른쪽이 2012-C 부품이다. 3개의 부품의 모두 모양이나 색상은 비슷하지만 3개의 부품 모두 다른

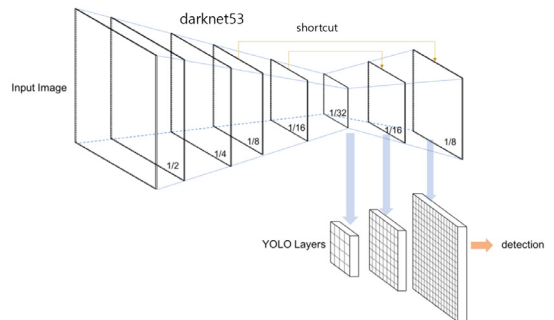


Fig. 12. Architecture of feature pyramid in YOLO v3.

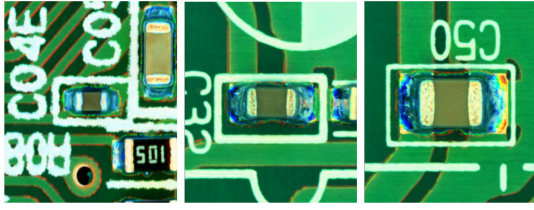


Fig. 13. Components of similar shape and color, 0603-C(left), 1005-C(middle), 2012-C(right).

클래스의 부품이다. 특징 피라미드 구조에서는 이와 같은 부품들은 다 같은 클래스로 분류할 확률이 높다. 그래서 YOLO v2를 통해 이처럼 모양이나 색은 다르지만 크기에 따라 다른 부품으로 분류하는 것이 본 연구의 목표이다.

### 3.3 YOLO 네트워크 수정

CNN 기반의 물체 검출 모델들은 모두 컨볼루션이 끝난 마지막 특징맵에서 물체를 검출하기 때문에 기본적으로 작은 물체 검출에 성능이 좋지 않다. CNN 기반 네트워크의 특징맵은 특징맵의 크기에 담고 있는 정보가 다른 특징이다. 특징맵의 크기가 클수록 작은 물체에 대한 정보를 많이 가지고 있고, 특

징맵의 크기가 클수록 큰 물체에 대한 정보를 많이 가지고 있다[14-16].

이런 작은 물체 검출에 대한 성능을 개선하기 위해 YOLO v1에서 v2로 개발되면서 상위 계층의 특징맵을 하위 계층의 특징맵에 붙여서 크기 작은 물체를 찾을 수 있게 되었다. 하지만 여전히 YOLO v2에서는 작은 물체를 잘 찾지 못한다는 단점이 존재한다. 본 논문에서는 0402 크기의 부품같이 크기가 매우 작은 부품들을 찾아야 하므로 이를 개선하기 위해 YOLO v2에서 사용한 방법을 응용해서 네트워크를 개선한다.

기존의 YOLO v2에서는 Fig. 14처럼  $26 \times 26$  특징맵을  $13 \times 13$  특징맵의 크기를 줄여  $13 \times 13$  특징맵과 크기를 동일하게 변경되 특징맵의 깊이를 4배로 늘린다. 깊이를 늘린 특징맵과 기존의  $13 \times 13$  특징맵을 합쳐서  $26 \times 26$  특징맵 정보와  $13 \times 13$  특징맵의 정보를 모두 포함하는 특징맵을 생성한다. 이렇게 작은 물체의 정보를 많이 가지고 있는 상위 계층과 큰 물체의 정보를 많이 가지고 있는 하위 계층을 합쳐 컨볼루션 단계에서 손실된 작은 물체에 대한 정보를 좀 더 많이 포함하여 작은 크기의 물체도 잘 검출할 수 있도록 네트워크가 설계되어있다[17-18].

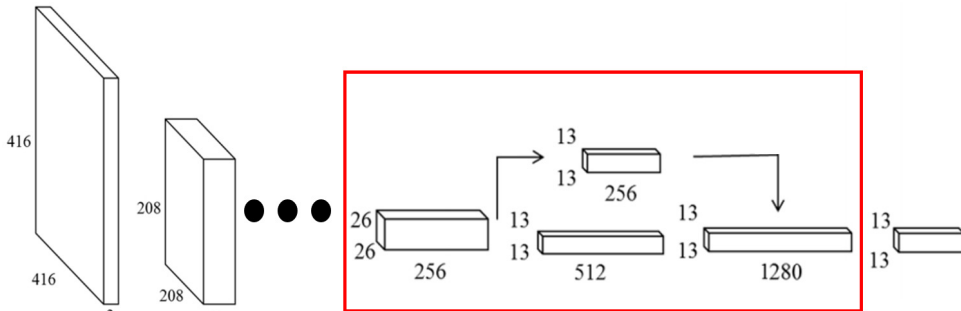


Fig. 14. Network structure of YOLO v2.

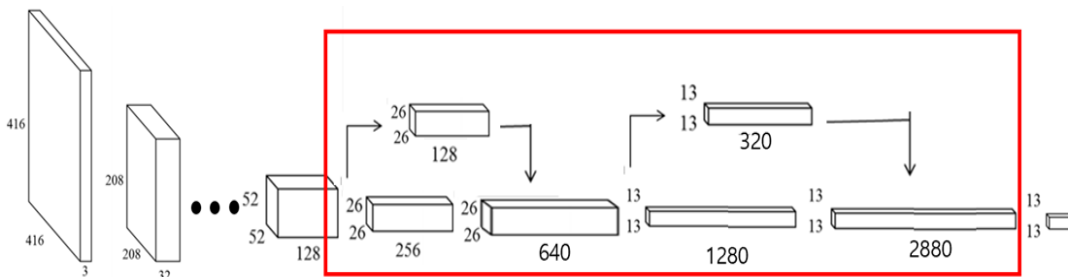


Fig. 15. Modified network structure.

작은 물체 검출의 성능을 높이기 위해 기존의 네트워크를 개선한다. Fig. 15과 같이 기존에 불러오던  $26 \times 26$  특징맵 불러오기 전에 더 상위 계층의  $52 \times 52$  특징맵을 기존과 같은 방법으로 특징맵을 합친다.  $52 \times 52$  특징맵을  $26 \times 26$  특징맵으로 줄이면서 깊이는 4배로 늘어난다. 늘린 특징맵을 다음 네트워크의  $26 \times 26$  특징맵과 합쳐  $13 \times 13$  특징맵으로 컨볼루션되기 전의  $26 \times 26$  크기의 특징맵이 작은 물체의 정보를 많이 가진 특징맵으로 만든다.  $26 \times 26$  특징맵에서  $13 \times 13$  특징맵으로의 컨볼루션도 같은 방법으로 진행한다. 이러한 방법으로 컨볼루션을 진행하면 기존의 YOLO 네트워크의 최종 특징맵보다 작은 크기의 물체의 정보를 더 많이 담고 있는 특징맵을 구할 수 있다. 기존의 네트워크보다  $13 \times 13$  특징맵이 더 작은 물체에 대한 정보를 많이 가지는 네트워크로 만들어 작은 물체 검출의 성능을 개선한다.

### 3.4 YOLO 앵커 박스 변경

YOLO v2에서는 영상을  $13 \times 13$ 의 그리드로 나누고 각 그리드에서 찾고자 하는 물체의 중심이 있는지를 결정한다. 네트워크의 최종 특징맵이 어떻게 구성되어 있는지를 분석하면 앵커박스의 역할을 알 수 있다[19]. Fig. 16과 같이 최종 특징맵의 깊이는  $(5 + \text{클래스 개수}) \times \text{앵커박스의 개수}$ 로, 학습하는 모델의 클래스 개수와 앵커박스의 개수에 따라 달라진다[6]. 최종 특징맵은 각 그리드 당 물체의 위치(x, y)  $\times$  물체의 크기(w, h)  $\times$  물체의 유무 확률  $\times$  클래스의 특징맵 덩어리들이 앵커박스의 개수만큼 존재한다[12]. 앵커박스가 5개 있다면 한 그리드당 5번씩 물체

의 유무를 판단한다. 만약 앵커박스를 5개를 사용하는 YOLO 모델에서 검출하고자 하는 물체의 크기가 클래스별로 차이가 크고 크기가 5개 이상으로 다양하다면 5개의 앵커박스로는 각 클래스의 크기에 맞게 설정할 수 없다. YOLO의 최종 특징맵에선 5개의 앵커박스에 대한 정보들만 있으므로 설정한 앵커박스의 크기와 차이나는 클래스의 물체가 있다면 그 물체는 잘 검출해내지 못한다. 앵커박스의 개수와 크기는 YOLO의 성능의 중요한 역할을 한다. 본 논문에서는 앵커박스의 개수와 크기를 부품 데이터에 맞게 변경하여 연구를 진행한다.

Fig. 16은 학습 데이터의 크기별 분포이며, 검정색 점은 앵커박스의 크기 분포이다. 앵커박스의 개수는 기존 5개의 앵커박스보다 9개 더 많은 15개로 늘려서 학습을 진행한다. 15개로 선택한 이유는 Fig. 16의 왼쪽처럼 COCO 데이터를 기반으로 앵커박스 크기와 5개의 앵커박스는 실험 데이터를 포함하지 못하지만 Fig. 16의 오른쪽 그림처럼 앵커박스의 개수를 15개로 늘리고 크기를 학습 데이터에 맞게 설정한 경우에는 학습 데이터 전체를 고루 포함하기 때문이다.

## 4. 실험 및 결과

### 4.1 실험 및 결과

학습데이터 14,548장, 실험 데이터는 5,511장으로 구성했다. 학습 데이터의 클래스별 세부 구성은 Table 3과 같다. 각 클래스 당 데이터 1000장을 기준으로 8:2로 학습데이터(800장)와 실험데이터(200장)로 나눴고 각 클래스마다 포함되는 부품들이 여러 개이고

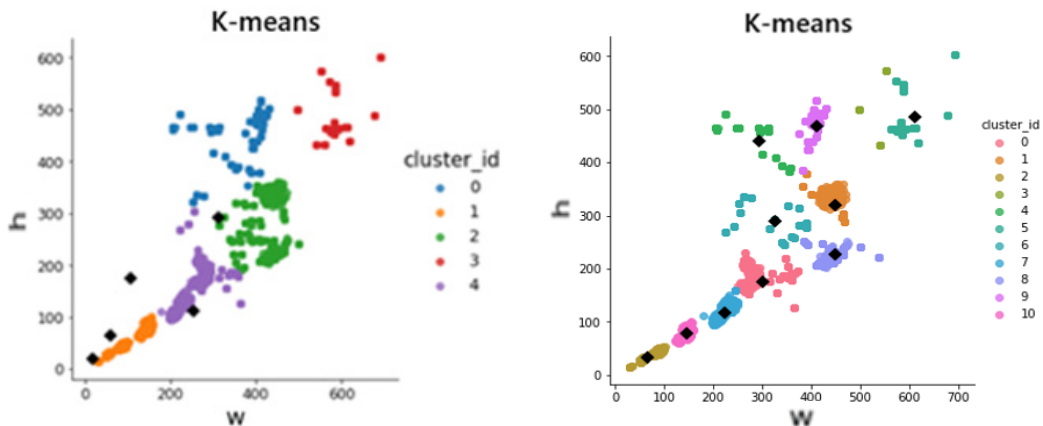


Fig. 16. Size distribution of Training data and anchor boxes, default anchor box (left), modified anchor box (right).



Table 2. Number of train data per class.

Num	Class	Count	Num	Class	Count
1	0402-C	800	10	1816-TR	800
2	0603-C	900	11	2012-C	900
3	0603-R	900	12	2012-R	900
4	1005-C	900	13	2515-D	900
5	1005-R	900	14	2812-TR	800
6	10065-D	800	15	3216-C	405
7	1511-D	800	16	3216-R	800
8	1608-C	900	17	3225-C	800
9	1608-R	564	TOTAL		14,548

배경도 다양하기 때문에 부품의 종류가 많고 배경이 다양한 경우는 학습 데이터를 900장으로 구성한다. 데이터가 1000장보다 적은 경우는 최대 영상의 개수를 8\*2로 나눠서 구성한다. 이 때 각 클래스에서 데이터 개수가 400개보다 작을 경우에는 학습 데이터의 균형을 생각해 학습 데이터에서 제외했다. Table 1을 참고하면 수집한 데이터의 편차가 크다. 데이터가 적은 클래스들을 학습에 포함시키더라도 각 클래스 당 학습 데이터의 편차가 심하면 데이터가 많은 클래스는 과적합 현상이 일어나고 데이터가 적은 클래스는 학습이 적절하게 되지 않을 것으로 판단되어 데이터가 적은 클래스들은 분류 클래스에서 제외한다. 따라서 중분류 기준으로 23개의 클래스의 부품을 분류하는 것이 본 연구의 목표이지만 수집 데이터가 부족한 클래스 0603-D, 3225-R, 2012-TR, 2812-TR-6, 4가지 클래스는 제외한다.

네트워크의 성능을 비교하기 위해 기본 YOLO v2의 모델과 본 논문에서 제안한 모델을 비교하여 실험하였다. 기본 YOLO v2의 실험 결과는 Table 3과 같이 전체적으로 정확도가 떨어진다. 특히 2515-D나 3225-C, 3225-R 같이 크기가 큰 클래스의 경우 정확도가 매우 낮다. 이는 기본 앵커박스가 크기가 큰 부품을 포함하지 못하기 때문이다. 또한 작은 물체에 대한 검출 정확도가 낮은 것은 CNN 기반의 물체 검출 모델들의 단점이다. 이를 개선하기 위해 네트워크와 앵커박스를 수정하였고, 실험 결과는 Table 4에 나타내었으며, 수정 전 모델에 비해 0402-C, 0603-C 같이 작은 클래스뿐만 아니라 크기가 큰 클래스에 대한 분류성능도 훨씬 더 높다. Table 4의 결과를 보면, 0603-C, 2012-C와 2012-R에서 불량이 다소 존재

하나 모두 10개 미만이고 전체적으로 불량률의 수는 매우 적은 것을 알 수 있다. Fig. 17과 18은 불량률의 예로, 부품의 종류는 제대로 검출하지만 부품의 크기를 오검출하는 경우가 대부분이다. 최종적으로 세부 부품별 분류 성능은 Table 8과 같이, 전체 분류기의 성능은 99.86%로 매우 좋은 결과를 나타낸다.

## 5. 결 론

본 논문에서는 PCB 부품 분류를 위해 부품의 특성

Table 3. Basic YOLO v2 experiment result.

Num	Class	Count	Pass	Fail	Accuracy
1	0402-C	200	195	5	97.5%
2	0603-C	352	346	9	98.29%
3	0603-R	393	391	2	99.49%
4	1005-C	319	319	0	100%
5	1005-R	249	242	7	97.18%
6	10065-D	285	285	0	100%
7	1511-D	260	260	0	100%
8	1608-C	355	321	34	90.42%
9	1608-R	355	305	50	85.91%
10	1816-TR	200	200	0	100%
11	2012-C	324	324	0	100%
12	2012-R	307	305	2	99.34%
13	2515-D	301	200	101	66.44%
14	2812-TR	264	176	88	66.66%
15	3216-C	347	325	22	93.37%
16	3216-R	200	149	51	74.5%
17	3225-C	200	30	170	15%
TOTAL		5,511	4,954	557	89.89%

Table 4. Experiment result of proposed network modification model (YOLO v2).

Num	Class	Count	Pass	Fail	Accuracy
1	0402-C	200	200	0	100%
2	0603-C	352	349	3	99.14%
3	0603-R	393	393	0	100%
4	1005-C	319	319	0	100%
5	1005-R	249	249	0	100%
6	10065-D	285	285	0	100%
7	1511-D	260	260	0	100%
8	1608-C	355	355	0	100%
9	1608-R	355	355	0	100%
10	1816-TR	200	200	0	100%
11	2012-C	324	320	4	97.83%
12	2012-R	307	306	1	99.67%
13	2515-D	301	301	0	100%
14	2812-TR	264	264	0	100%
15	3216-C	347	347	0	100%
16	3216-R	200	200	0	100%
17	3225-C	200	200	0	100%
TOTAL		5,511	5,503	8	99.86%

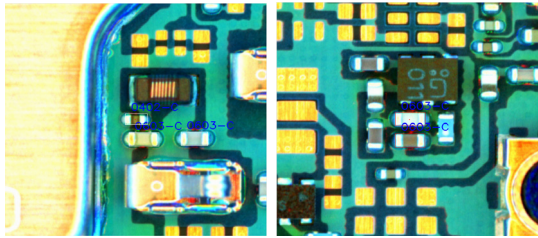


Fig. 17. Detection results of 0402-C (left) and 0603-C (right).

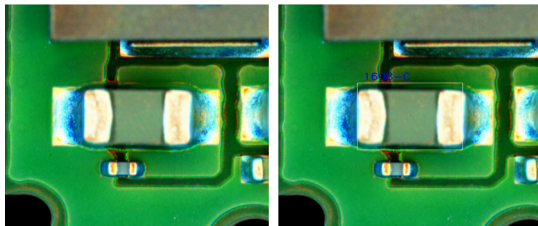


Fig. 18. Case of incorrectly finding 2012 size capacitor as 1608 size capacitor (original (left), detection result (right)).

을 조사하여 이들 간의 모양과 색상이 비슷하더라도 잘 구별해 낼 수 있도록 부품의 경계영역을 정하고

클래스를 정의하였다. 그리고 부품의 크기가 다른 다른 종류의 부품으로 분류할 수 있도록 YOLO의 다른 모델들보다 상대적으로 크기정보에 민감한 YOLO v2를 사용했다. 또한 앵커박스의 개수와 크기를 본 논문에 맞게 수정하였으며, 작은 부품도 인식할 수 있는 네트워크를 제안하였다. 본 논문에 사용된 부품의 클래스는 총 29개의 클래스이지만 부품의 크기가 같으나 높이 정보가 필요한 부품들은 제외하고, 모양도 비슷하고 크기 비슷한 클래스들은 같은 클래스로 묶었다. 데이터의 개수가 적은 부품은 학습의 신뢰성을 위해 제외해서 17개의 클래스를 분류결과를 보였으며, 추후 데이터가 확보되는 대로 확장할 예정이다. 실험은 학습 데이터 14,548장, 테스트 데이터는 5,511장으로 수행하였다. 기존의 앵커박스의 개수는 5개인데 비해, 본 논문에서는 크기가 다른 부품의 개수와 다이오드와 트랜지스터 등의 특성을 고려해서 앵커박스의 개수를 15개로 늘렸다. 앵커박스의 크기 역시 부품들의 실제 크기를 기준으로 변경하고, 실험을 통해 앵커박스를 수정한 경우가 큰 부품에 대해 정확도가 높고 ROI를 좀 더 정확하게 찾는 것을 확인하였다. 또한 앵커박스를 수정하면 작은 크기를 상대적으로 오검출을 많이 하는 단점이 존재하나, 네트워크를 수정하여 보완하고, 실험을 통해 성능을 확인하였다. 최종으로 개발한 YOLO 프로그램은 앵커박스과 네트워크를 수정한 모델이며, 성능 평가는 전체 5,511개의 데이터 중 5,503개를 정확히 찾고, 8개를 오검출하여 정확도는 99.86%로 매우 좋은 성능을 가진다.

소분류까지 정확한 분류를 위해서는 높이 정보를 이용해 RGB-D 4채널 영상을 활용하여야 한다. 하지만 높이 정보는 제공받지 못해 해당 논문에서는 RGB 영상만을 이용하는 중분류를 진행하였다. 추후 높이 정보를 수집하여 RGB-D 영상으로 소분류까지 부품을 분류하고자 한다. 즉, 1511-D와 1810-D의 분류와 2812-TR와 2816-TR, 2012-C와 3216-C의 부품들도 높이 값을 활용해 세부 분류하는 것이 추후 연구의 목표이다.

또한 본 논문에서는 주로 많이 사용되고 수집하기 쉬운 부품들인 커패시터, 다이오드, 레지스터, 트랜지스터, 4가지의 종류에 대해서만 수행하였으나 더 다양한 범주의 데이터가 확보되는 대로 나머지 8개의 부품들에 대해서 분류할 수 있는 분류기를 만들어

실제 산업 현장에서 사용되는 시스템에 이식할 계획이다.

## REFERENCES

- [1] J.S. Lee and T.H. Park, "Defect Classification of Components for SMT Inspection Machines," *Journal of Institute of Control, Robotics and Systems*, Vol. 21, No. 10, pp. 982-987, Oct. 2015.
- [2] J.H. Ryu, Y.H. Kim, and T.H. Park, "Assembly Defect Classification of SMD Components by Cascade Convolutional Neural Network," *KIEE*, Vol. 68, No. 10, 2019.
- [3] J.J. Lee, B.G. Lee and J.C. Yoo, "3-D Solder Paste Inspection Based on B-spline Surface Approximation," *Journal of the Korean Society for Industrial and Applied Mathematics*, Vol. 10, No. 1, 2006.
- [4] A. Fazakas, M. Purcar, and D. Turcu, "Polarity Determination of Electrolytic Capacitors in Power Supplies from External Terminals," *Design and Technology in Electronic Packaging (SIITME) 2019 IEEE 25th International Symposium*, pp. 395-398, 2019.
- [5] S.G. Youn, Y.A. Lee, and T.H. Park, "Automatic Classification of SMD Packages using Neural Network," *Journal of Institute of Control, Robotics and Systems*, Vol 21, No 3, pp. 276-282, 2015.
- [6] D.U. Lim, Y.G. Kim, and T.H. Park, "SMD Classification System on PCB Using Two Convolution Neural Networks," *Journal of Institute of Control, Robotics and Systems*, Vol. 25, No. 10, pp. 923-928, 2019.
- [7] Y.A. Lee, and T.H. Park, "Automatic Recognition of SMD Packages Using Neural Network," *The Korean Institute of Electrical Engineers*, pp. 38-39, 2013.
- [8] C.G. Cho and H.G. Park, "Main Parts Used in Electronic Ballast Diode, Transistor, PCB, Resistor," *The Korean Institute of Illuminating and Electrical Installation Engineers*, Vol. 13, No. 3, pp. 3-8, 1999.
- [9] S.M. Gang and J.J. Lee, "Coreset Construction for Character Recognition of PCB Components Based on Deep Learning," *Journal of Korea Multimedia Society*, Vol. 24, No 3, 382-395, 2021.03
- [10] U.H. Lee, Y.S. Choi, J.L. Kim, and G.S. Jung, "Generation of Reference Model for PCB Pattern Inspection from Gerber CAD Data," *The Institute of Electronics and Information Engineers*, pp. 317-323, 1994.
- [11] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 779-788, 2016.
- [12] J. Redmon and A. Farhadi, "YOLO9000: Better, Faster, Stronger," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6517-6525, 2017.
- [13] J. Redmon and A. Farhadi, "Yolov3: An Incremental Improvement," *IEEE Conference on Computer Vision and Pattern Recognition, (CVPR)*, 2018.
- [14] W. Huang and P. Wei, "A PCB Dataset for Defects Detection and Classification," *Journal of Latex Class Files*, Vol. 14, No. 8, pp. 1-9, 2018.
- [15] J. Sang, Z. Wu, P. Guo, H. Hu, Xiang H, Q. Zhang, and B. Cai, "An Improved YOLOv2 for Vehicle Detection," *Sensors*, Vol. 18, No. 12, 2018.
- [16] L. Zhang, Y. Jin, X. Yang, X. Li, X. Duan, and Y. Sun, "Convolutional Neural Network-Based Multi-Label Classification of PCB defects," *The Journal of Engineering*, pp. 1612-1616, 2018.
- [17] N. Kavitha and D.N. Chandrappa, "Optimized YOLOv2 Based Vehicle Classification and Tracking for Intelligent Transportation System," *Results in Control and Optimization*, Vol. 2, 2021.
- [18] L. Wang, W. Li, W. Zhang, and C. Wei,

“Pedestrian Detection Based on YOLOv2 with Skip Structure in Underground Coal Mine,” *2017 IEEE 3rd Information Technology and Mechatronics Engineering Conference (ITOEC)*, pp. 1216–1220, 2017.

- [19] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks,” *NIPS’15: Proceedings of the 28th International Conference on Neural Information Processing Systems*, Vol. 1, pp. 91–99, 2015.



#### 윤 형 조

2014년~현재 계명대 컴퓨터공학  
부 게임모바일공학과 입학  
2018년 계명대 컴퓨터공학과 대  
학원 입학~졸업예정  
관심분야: 영상처리, 패턴인식, 계  
임공학, 딥러닝



#### 이 준 재

1986년 경북대 전자공학과 졸업  
1990년 동 대학원 석사  
1994년 동 대학 박사  
1995년~2007년 동서대학교 컴퓨  
터 컴퓨터공학부 부교수  
1998년~1999년 Georgia Institute  
of Technology 방문 교수

2000년~2001년 (주)파미 연구소장

2007년~현재 계명대학교 컴퓨터공학부 교수.

관심분야: 영상처리, 3D 컴퓨터비전, 딥러닝, 게임.