



저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#)

석사학위논문

YOLO 개선을 통한
PCB 불량 검출 시스템 설계 및 구현
PCB Defect Detection System Based on improved YOLO

이 민 균

한양대학교 공학대학원

2023년 2월

석사학위논문

YOLO 개선을 통한
PCB 불량 검출 시스템 설계 및 구현
PCB Defect Detection System Based on improved YOLO

지도교수 조 인 휘

이 논문을 공학 석사학위논문으로 제출합니다.

2023년 2월

한양대학교 공학대학원
컴퓨터공학 전공
이 민 균

이 논문을 이 민 균의 석사학위논문으로 인준함

2023 년 2 월

심사위원장 이 병 호(인)

심 사 위 원 조 인 휘(인)

심 사 위 원 손 규 식(인)

한양대학교 공학대학원

차 례

그림 차례	iii
표 차례	iii
국문요지	iv
제 1장 서론.....	1
1.1. 연구의 배경 및 필요성	1
1.2. 연구 내용	3
제 2장 관련연구.....	4
2.1. 객체 인식 방법	4
2.2. Data augmentation	8
2.3. 모델 경량화 기법(가지치기 기법)	13
제 3장 설계 및 구현.....	15
3.1. 시스템 설계	15
3.2 객체인식 기술 구현	16
3.3. Image augmentation	23
3.4. Prune	24
제 4장 성능평가.....	30
4.1. 실험	30
4.2. 실험 결과	32

제 5장 결론.....	40
5.1. 결론	40
참고문헌.....	41
ABSTRACT.....	42
감사의 글.....	44



그림 차례

그림 1. R-CNN.....	4
그림 2. Fast R-CNN	5
그림 3. Faster R-CNN.....	6
그림 4. YOLO	7
그림 5. Dropout.....	9
그림 6. Cutout	11
그림 7. Cutout 실험 결과	12
그림 8. Pruning	13
그림 9. Sparsity	14
그림 10. Flowchart of the proposed system	15
그림 11. mouse bite	16
그림 12. Missing hole	16
그림 13. short	16
그림 14. YOLOv5 성능	17
그림 15. R-CNN.....	18
그림 16. Network 구조	21
그림 17. Cutout	23
그림 18. Pruning.....	24
그림 19. flowchart of pruning	28
그림 20. flowchart of PCB defect detection system.....	31

표 차례

표 1. image augmentation 결과.....	33
표 2. channel pruning	37
표 3. result of pruning.....	38
표 4. YOLOv5 VS Proposed system	39

국문 요지

전자제품의 지능화가 전 산업분야에서 가속화됨에 따라, 전자 하드웨어의 신뢰성 확보는 기업의 핵심 의제가 되고 있다.

지속적인 산업의 발전으로 점점 더 미세한 회로와 복잡성이 증가하는 전자 제품의 PCB 품질 요구 사항은 꾸준히 증가하고 있다.

하지만 아직도 PCB 품질 요구를 사람에 의한 육안 검사로 이루어지고 있는 상황이다.

그 때문에 작업자의 숙련도 기준에 따라 다르게 판단되는 경우가 많다. 즉, 불량 판정에 대한 정형화된 수치가 없는 상황이다.

게다가 실제 생산 현장에서는 불량 발생이 매우 적어, 불량 이미지를 수집하기 쉽지 않고, 핵심 기술 노출에 대한 문제로 불량 이미지 역시 얻기 힘든 상황이다.

결국 많은 기업에서 전통적인 방법을 사용할 수밖에 없는 상황이다.

이와 같은 방법은 계속해서 발전하는 PCB 시장에서 품질 요구를 충족하기 어렵다.

다시 말해, 불량 데이터의 부족과 수동 검사로 떨어지는 품질 일관성을 해결하기 위한 연구가 필요하다고 판단된다.

본 연구에서는 PCB의 불량을 검출하는 시스템을 개발한다.

시스템 개발을 통해 품질 일관성 저하에 대한 문제를 데이터를 통해 특징을 수동으로 추출할 필요가 없는 딥러닝 기법(YOLO)으로 해결하였다.

또한 적은 수의 이미지 데이터를 가지고 Neural Network를 훈련할 때 발생하는 문제인 과적합(overfitting)을 해결하기 위해 이미지 전처리 기법인 이미지 어그멘테이션(Image augmentation)기법으로 해결하였다.

또한 속도가 중요한 PCB 제조 환경을 위한 딥러닝 모델 경량화 기법인 가지치기(Pruning)기법을 추가하여 PCB 불량 검출 시스템을 설계 및 구현하였다.

PCB 불량 검출 시스템을 통해 85 %의 불량을 검출하였으며 모델 경량화를 통해 기존 대비 31 %의 모델 경량화를 구현하였다.

본 연구에서 제안한 PCB 불량 검출 시스템을 통해 해당 분야에서의 유용성을 입증하여 결과를 제시한다.

제 1장 서론

1.1. 연구의 배경 및 필요성

인쇄회로기판(Printed Circuit Board)은 현재 클라우드 컴퓨팅, 5G, 빅데이터, IoT, AI 등 디지털 기술의 발전과 최신 IT기술의 활용 수요 증가에 따라 PCB 중요성이 높아지고, 산업 발전이 가속화되고 있다. Prismark에 따르면, 글로벌 PCB 시장의 2021년 총 생산액은 전년대비 23.4% 증가한 804억 달러를 기록하였다.

2021년 기준 PCB 생산은 중국과 일본을 비롯한 아시아 국가들에 90% 이상이 집중되어 있다. 2021년 지역 국가별 PCB 총 생산액을 살펴보면 전세계 모든 지역에서 두 자릿수 성장률을 보였는데, 이는 코로나 팬데믹으로 인한 디지털 전환 가속화로 고속성장을 기록한 것으로 분석됐다.

이와 같이 인쇄회로기판 시장의 규모는 고속 성장을 하고 있으며, 전자제품의 지능화가 산업에서 가속화됨으로써, PCB의 신뢰성의 확보는 기업들의 핵심 의제가 되고 있다.

지속적인 산업의 발전으로 점점 더 미세한 회로와 복잡성이 증가하는 전자 제품의 PCB 품질 요구 사항은 꾸준히 증가하고 있다.

하지만 아직도 PCB 품질 요구를 사람에 의한 육안 검사로 이루어지고 있는 상황이다. 그 때문에 작업자의 숙련도 기준에 따라 다르게 판단되는 경우가

많다. 즉, 불량 판정에 대한 정형화된 수치가 없는 상황이다.

게다가 실제 생산 현장에서는 불량 발생이 매우 적어, 불량 이미지를 수집하기 쉽지 않고, 핵심 기술 노출에 대한 문제로 불량 이미지 역시 얻기 힘든 상황이다.

결국 많은 기업에서 전통적인 방법을 사용할 수밖에 없는 상황이다.

이와 같은 방법은 계속해서 발전하는 PCB 시장에서 품질 요구를 충족하기 어렵다. 다시 말해, 불량 테이터의 부족과 수동 검사로 떨어지는 품질 일관성을 해결하기 위한 연구가 필요하다고 판단된다.



1.2. 연구 내용

본 연구에서 제안한 PCB 불량 검출 시스템의 객체 검출 기술은 컴퓨터 비전 기술에서 중요한 연구 중 하나이다. 최근에는 다양한 객체 검출 알고리즘이 제안되고 있는데, 다양한 대상이 다양한 상태를 가지므로 여러 기준으로 대상을 탐지할 수 있어야 한다. 상태가 다른 대상에 대해 탐지 성능을 개선하기 위해 개선된 YOLO(You Only Look Once)를 제안하였다. 본 연구의 주요 내용은 아래와 같다.

- 딥러닝을 이용한 PCB 불량 검출 시스템 개발

객체 검출에 범용적으로 사용되는 딥러닝 모델인 YOLO를 이용하여 PCB 불량 검출 시스템을 개발하였다.

- 이미지 어그멘테이션 기법을 통한 성능 개선

적은 수의 이미지 데이터를 가지고 Neural Network를 훈련할 때 발생하는 문제인 과적합(overfitting)을 해결하기 위해 이미지 전처리 기법인 이미지 어그멘테이션(Image augmentation)기법으로 해결하였다.

- 모델 경량화를 통한 성능 개선

속도가 중요한 PCB 제조 환경을 위한 딥러닝 모델 경량화 기법인 가지치기(Pruning)기법을 추가하여 PCB 불량 검출 시스템을 설계 및 구현하였다.

제 2장 관련연구

2.1. 객체 인식 방법

2.1.1. R-CNN

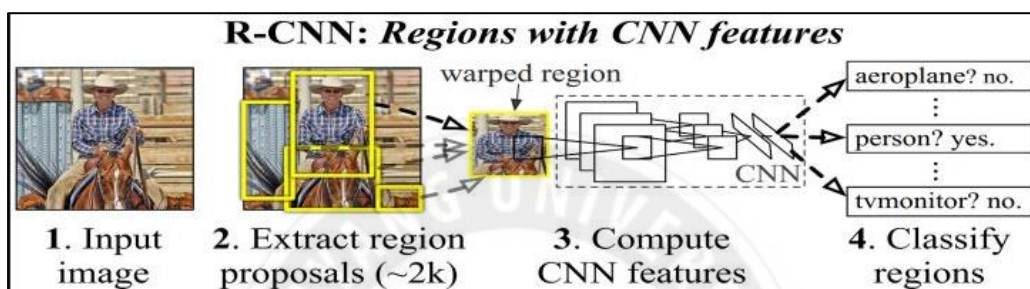


그림 1. R-CNN

R-CNN의 객체 인식 과정은 위의 그림과 같이 3 단계로 진행된다.

- 1) input image에서 selective search algorithm을 이용하여 region proposal을 생성한다.
- 2) 생성된 각 region proposal을 동일한 size로 변환 후 CNN을 사용하여 feature를 추출한다
- 3) 추출된 feature를 통해 region proposal 내의 객체를 support vector machine을 이용하여 분류한다.

최종적으로 회귀 학습을 이용해서 더 성능을 향상시킨다.

2.1.2. Fast R-CNN

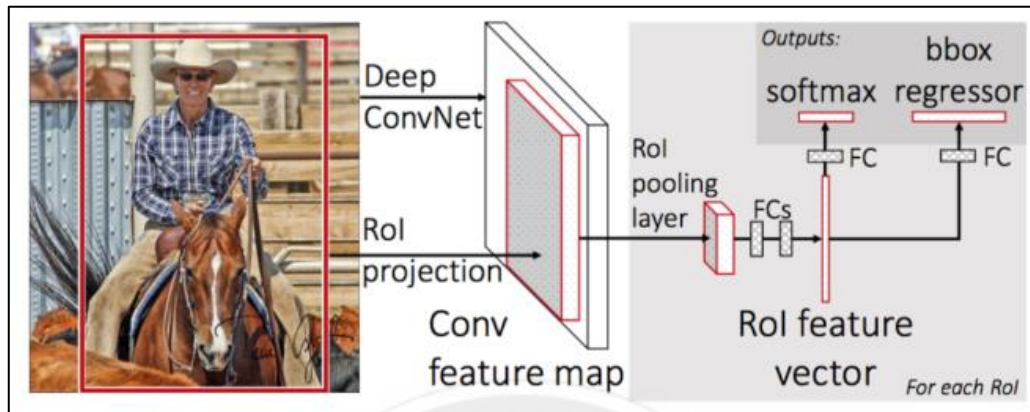


그림 2. Fast R-CNN

기존의 R-CNN은 CNN, support vector machine, 회귀의 학습이 분리되어 있고, 많은 region proposal에서 각각 CNN을 학습하기 때문에 많은 시간이 소요된다. Fast R-CNN은 이를 보완하기 위해 위의 사진과 같이 하나의 input image에 하나의 CNN을 학습한다. 이를 통해 Fast R-CNN은 기존의 R-CNN 보다 높은 정확도와 훈련 시간을 감소시켰다.

2.1.3. Faster R-CNN

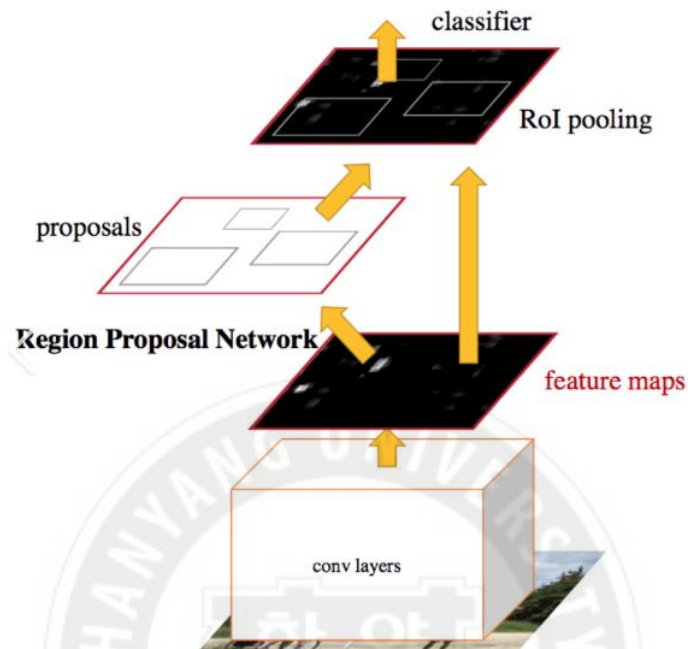


그림 3. Faster R-CNN

Fast R-CNN은 region proposal을 생성하는 알고리즘이 CNN 외부에서 수행된다. 이러한 구조는 학습 속도에서 비효율적이라는 단점이 있다.

Faster R-CNN은 region proposal을 생성하는데 기존의 selective search algorithm을 사용하지 않고 위의 그림과 같이 CNN의 마지막 layer에 region proposal network를 두어 proposal을 생성한다.

위와 같은 구조를 이용해 기존의 Faster R-CNN보다 훈련 시간을 감소시키면서, 정확도 또한 향상시켰다.

2.1.4. YOLO



그림 4, YOLO

YOLO는 network의 최종 출력단에서 경계박스의 위치 찾기와 class의 분류가 동시에 이루어진다. 이는 YOLO가 object detection 문제를 하나의 회귀 문제로 접근하여 전체적인 구조를 간소화함으로써 구현하였다.

2.2. Data augmentation

Data augmentation이 필요한 이유는 overfitting을 방지하기 위해서이다. CNN은 representational power가 증가하게 되면 overfitting의 가능성도 증가하게 된다.

Overfitting의 해결방법은 3가지가 대표적이다.

첫번째는 학습 데이터에 대한 절대적인 양을 늘리는 방법이다. 현재까지 수집된 학습 데이터가 테스트 데이터의 패턴을 잘 대변할 수 있도록 그 수를 늘리는 간단한 아이디어이다.

하지만 이는 시간과 비용적인 부담이 너무 크며, 본 연구 목적에 맞지 않는다. 두 번째는 생성 모델을 사용해서 현재 가지고 있는 학습 데이터와 유사한 샘플들을 더 많이 찍어낼 수 있는 방법이다. 이는 모델을 하나 더 학습해야 한다는 점에서 매력도가 떨어진다.

마지막으로 Data augmentation은 컴퓨터 비전 분야에서는 딥러닝 시대 이전부터 개발된 다양한 이미지 변형 방법들이 있기 때문에 상대적으로 쉽고 빠르게 적용이 가능하다는 장점이 있다. 본 연구에서는 Data augmentation 기법 중 cutout 기법을 사용하기로 정하였고 이의 타당성을 cutout의 선행 연구인 Dropout과 비교하여 보았다.

2.2.1. Dropout

컴퓨터 비전에서 overfitting을 방지하는 방법 중 data augmentation은 적용이 쉽고 효과적이기에 널리 사용되어 왔다.

Noise를 추가하는 방법 중 가장 널리 사용되는 방법은 Dropout이다.

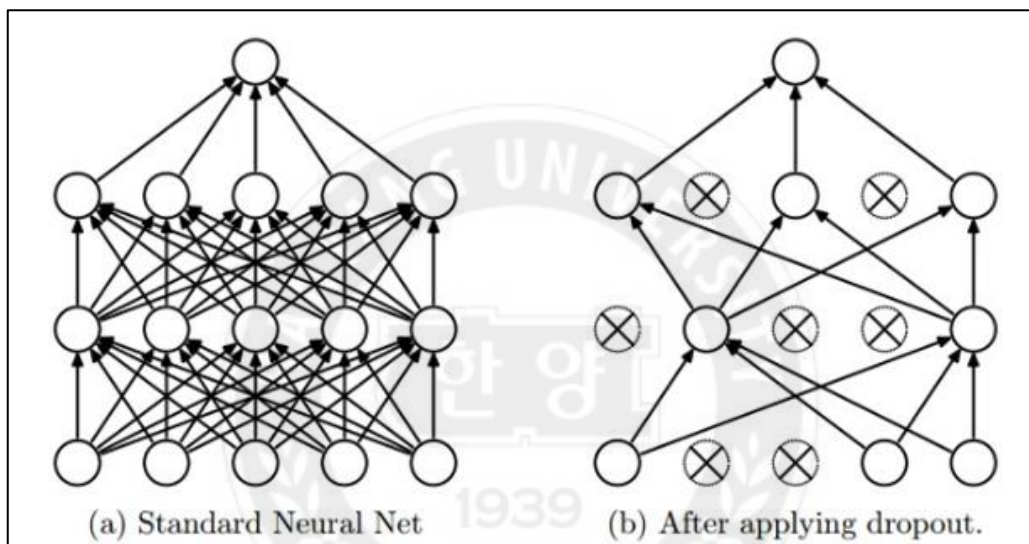


그림 5. Dropout

Dropout에서 Input image를 가리는 행위를 Occlusion이라 부르는데, 해당 방식은 스크래치, 점 등을 이미지에 합성해서 이미지를 가리는 방식을 사용하였다.

Dropout은 학습 단계에서 Network의 hidden unit activation을 일정 확률로 0으로 바꾸는 방식이다.

이 방식은 뛰어난 정규화 효과를 얻을 수 있었다.

하지만 Dropout은 Fully-connected layer에서는 좋은 효과를 보이지만

Convolutional layer에서는 그만큼 강력한 효과를 보지 못하였다.

그 이유로 첫번째는 Convolutional layer는 fully-connected layer에 비해 파라미터 개수가 훨씬 적다.

두번째로는 Convolutional layer에서는 input을 Fully-connected layer처럼 1-dimension으로 펼치지 않고 사용하기 때문에 인접한 픽셀은 비슷한 정보를 많이 공유하여 일부 픽셀이 0으로 바뀌어도 주변 픽셀들로 인해 해당 정보가 전달이 될 수 있으므로 Dropout 효과를 크게 얻기 힘들다는 점을 들 수 있다.

2.2.2. Dropout과 cutout의 차이점

Dropout에서는 feature map이 아닌 input에서 0으로 채우는 방식에서 차이가 있다. Dropout은 중간단계의 layer의 feature map에서만 random하게 일부 영역을 지우고 각 feature map의 channel간 독립성이 존재하여 어떤 channel에서는 지워진 영역이 다른 channel에서는 지워지지 않게 되어 불일치성이 일부 존재하게 된다. 하지만 Cutout은 input 단계에서 일부 영역을 지우고 그 결과 중간 단계의 layer에서도 같은 영역 주위를 지우는 것과 같은 효과를 볼 수 있으며 각 feature map의 channel간에도 상관 관계를 갖게 된다.

2.2.3. cutout의 구현방법

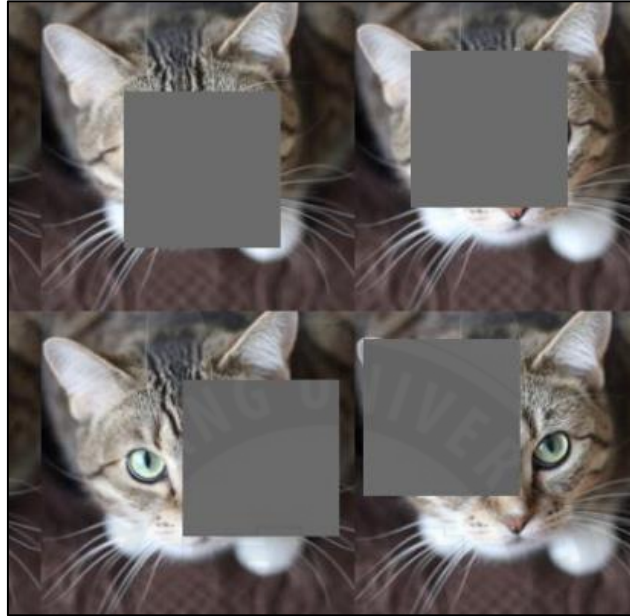


그림 6. Cutout

Cut의 영역을 정할 때 어떤 모양의 도형을 사용할 것인지와 도형의 크기를 어떻게 정할 지를 정해야 하는데, 실험적으로 분석한 결과 어떤 모양을 사용하느냐 보다는 어떤 크기를 사용할 지가 성능에 더 큰 영향을 주었다. 본 연구에서는 정사각형 모양을 default로 사용하였다.

여기서 짚고 넘어가야할 점은 Cutout으로 사용할 정사각형의 크기는 실험적으로 정해야 한다는 점이다.

Method	C10	C10+	C100	C100+	SVHN
ResNet18 [5]	10.63 ± 0.26	4.72 ± 0.21	36.68 ± 0.57	22.46 ± 0.31	-
ResNet18 + cutout	9.31 ± 0.18	3.99 ± 0.13	34.98 ± 0.29	21.96 ± 0.24	-
WideResNet [22]	6.97 ± 0.22	3.87 ± 0.08	26.06 ± 0.22	18.8 ± 0.08	1.60 ± 0.05
WideResNet + cutout	5.54 ± 0.08	3.08 ± 0.16	23.94 ± 0.15	18.41 ± 0.27	1.30 ± 0.03
Shake-shake regularization [4]	-	2.86	-	15.85	-
Shake-shake regularization + cutout	-	2.56 ± 0.07	-	15.20 ± 0.21	-

그림 7. Cutout 실험 결과

Cutout의 최적의 정사각형 크기를 적용하였을 때의 실험 결과는 위의 표에 나와있다. 많게는 2 % 이상의 성능 향상이 있었다. 간단한 방식이지만 성능 향상이 큰 것을 확인할 수 있다.



2.3. 모델 경량화 기법(가지치기 기법)

2.3.1. Pruning

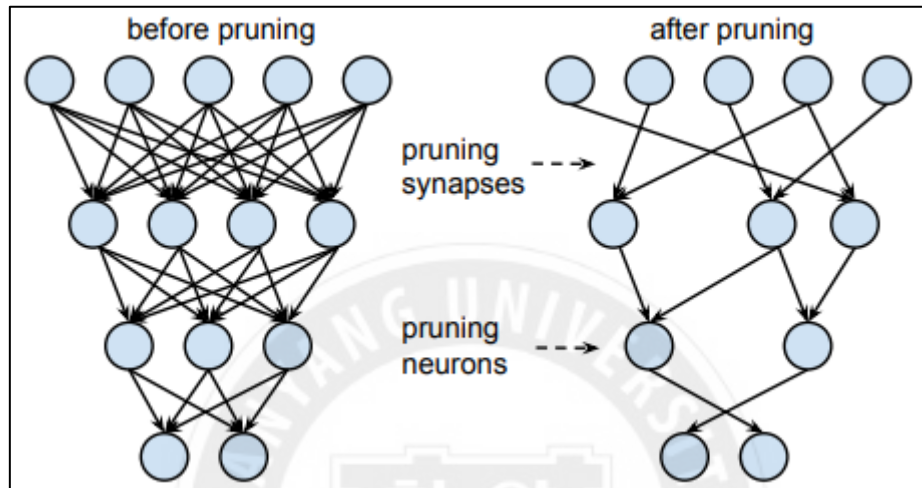


그림 8. Pruning

Pruning은 딥러닝 모델의 weight들 중 가중치가 낮은 weight의 연결을 제거하여 딥러닝 모델의 파라미터를 줄이는 방법이다.

만약 각 weight 값이 결과에 주는 영향이 다르다면, 상대적으로 영향력이 작은 weight는 삭제하여도 결과에 주는 영향이 작을 것이다. 따라서 최종 결과에 영향력이 적은 weight를 제거한다면 적은 파라미터를 가졌지만 유사한 성능을 보여주는 모델을 만들 수 있다.

즉, Pruning은 네트워크의 성능이 크게 저하되지 않는 선에서 weight를 최대한 sparse(희소하게) 하게 만드는 방법으로 정의할 수 있다.

2.3.2. Sparsity

$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 3 & 0 & 2 \\ 0 & 0 & 0 & 0 \\ 0 & 6 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 & 4 & 0 \\ 0 & 3 & 1 & 2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$
very sparse	little sparse	sparsity ?

그림 9. Sparsity

위 3 개의 행렬 중 왼쪽의 행렬은 한 개의 요소를 제외하고 0이기 때문에 희소(sparse)행렬로 판단된다.

이처럼 대부분의 weight가 0이면, sparse한 것으로 간주한다. Sparsity란 전체 네트워크에서 얼마나 많은 weight가 정확하게 0인지를 나타내는 척도이다. 이를 측정할 수 있는 가장 간단한 방법은 L_0 norm 을 활용하는 것이다.

이를 수식으로 나타내면 다음과 같다.

$$\|x\|_0 = |x_1|^0 + |x_2|^0 + \dots + |x_n|^0$$

위 수식에서 각 요소가 1 또는 0으로 되며, L_0 norm 요소의 값이 0이 아닌 값들의 개수가 된다. 이를 전체 weight 개수 대비 L_0 norm의 값을 확인하면 sparsity를 확인할 수 있다.

제 3장 설계 및 구현

3.1. 시스템 설계

본 시스템의 목적은 기존의 객체 인식 모델을 기반한 PCB 불량 검출 시스템보다 높은 정확도를 제공하고 딥 러닝 모델 경량화를 통해 성능이 크게 저하되지 않는 선에서 모델 성능을 개선하는 시스템이다.

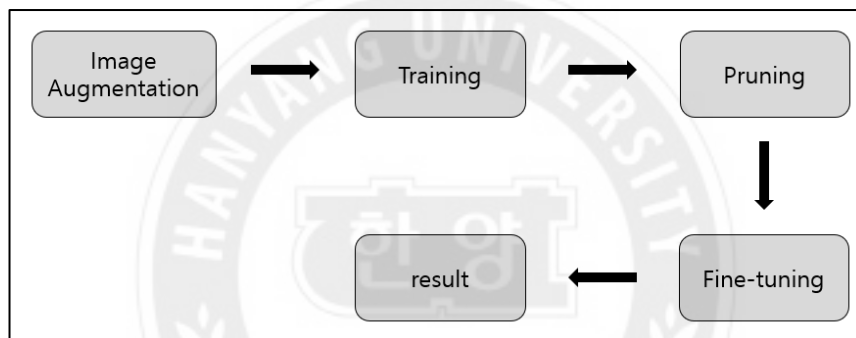


그림 10. Flowchart of the proposed system

- Image Augmentation, Training - Image augmentation을 통해 오버 피팅 문제를 해결하고자 하며, 다양한 cutout 크기로 Training하여 기존의 객체 검출 모델보다 향상된 성능을 확인하고자 한다.
- Pruning - 학습이 완료된 객체 검출 모델의 가중치들을 희소성이라는 지표를 도출한 뒤 해당 지표를 기준으로 일정 비율의 parameter들을 네트워크 그래프에서 pruning하여 객체 검출 모델의 경량화를 하고자 한다.
- Fine-tuning - 남은 가중치만을 가지고 fine tuning해 성능을 향상시키고자 한다.

3.2 객체인식 기술 구현

3.2.1 학습 데이터 구성

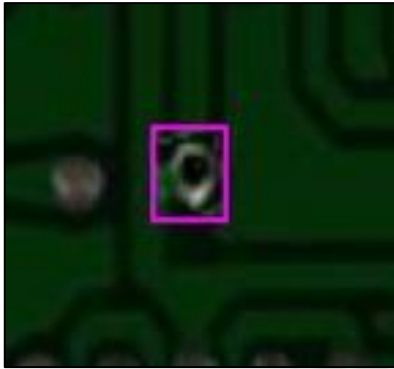


그림 11. mouse bite

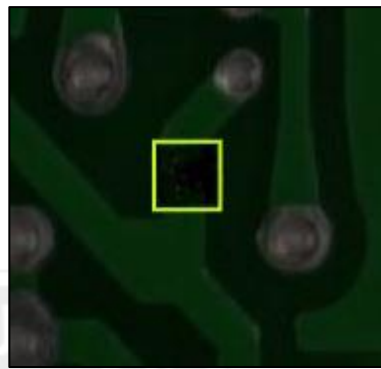


그림 12. Missing hole



그림 13. short

PCB 불량 인쇄 회로 기판 이미지를 Missing hole, mouse bite, Short의 3가지의 불량 케이스 Class로 나눠 총 1590 장의 이미지를 수집하였으며, Roboflow 사이트에서 Labelling을 진행하여 1590 장의 Dataset을 구축하였다.

3.2.2 PCB 불량 검출 딥러닝 모델 선정

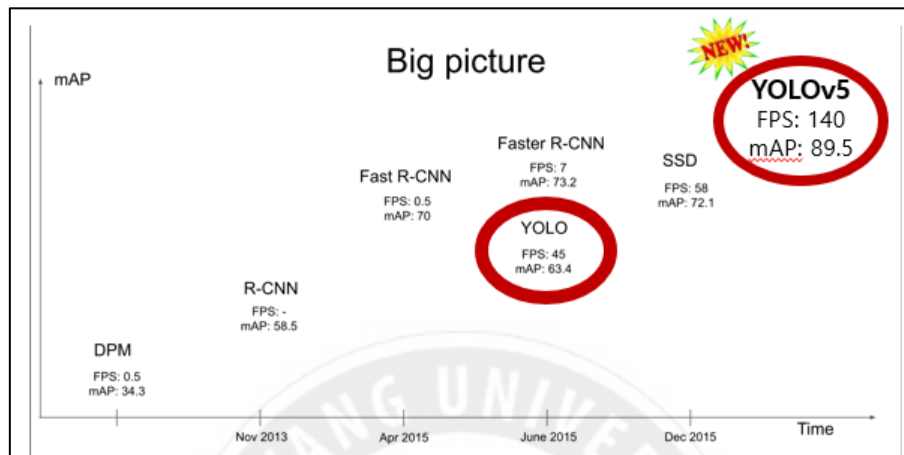


그림 14. YOLOv5 성능

PCB 불량 검출 기술을 구현하기 위해 딥러닝 모델인 YOLOv5를 사용하였다. YOLOv5는 기존의 YOLO 모델에 비해 FPS(Frame per seconds)와 Map(Mean Average Precision) 측면에서 뛰어난 성능을 발휘한다.

YOLOv5은 크게 Backbone과 Head 부분으로 구성된다.

여기서 Backbone은 이미지로부터 Feature map을 추출하는 부분으로, CSP-Darknet을 사용한다. YOLOv5의 Backbone은 s, m, l, x의 4가지 backbone을 가지고 있으며, 본 연구에서는 s를 사용하였다.

Head는 추출된 Feature map을 바탕으로 물체의 위치를 찾는 부분이다.

Anchor Box를 설정하고 최종적인 경계 박스를 생성한다. 3가지의 scale에서 경계 박스를 생성하며, 각 scale에서 3개의 Anchor Box를 사용한다. 그러므로 총 9개의 Anchor Box를 사용한다.

3.2.3 PCB 불량 검출 구현

YOLO란 You Only Look Once의 약자로, 다른 딥러닝 모델에 비해 빠른 처리속도를 가지고 있으며, 객체 탐지에 범용적으로 사용되는 딥러닝 모델이다. 해당 모델이 나오기 전 딥러닝 기반의 객체 검출 시스템들은 Classification 모델을 객체 검출 시스템에 적용한 모델들이 주를 이루었다. 제일 최근으로는 R-CNN으로, R-CNN 모델은 region proposal/ classification / box regression 3 가지 단계의 과정을 가지며, 3 단계를 개별적으로 학습해야 하므로 복잡한 파이프 라인을 가지게 된다. R-CNN은 이러한 복잡함으로 최적화가 어렵고 추론 시간이 길어진다는 단점을 가지고 있다.

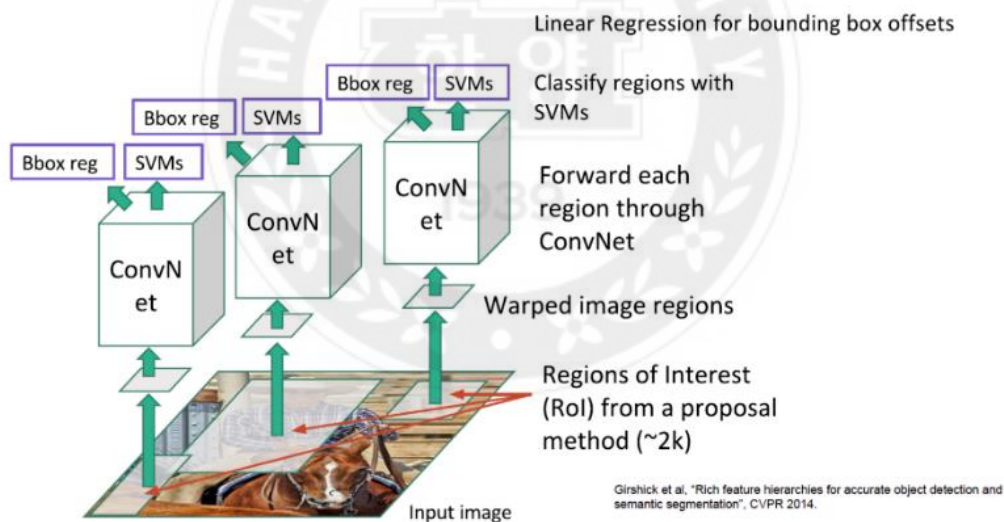


그림 15. R-CNN

YOLO는 Classification을 적용한 모델이 아닌 regression문제로 정의하는 것을 통해서 이미지에서 경계 박스 좌표와 각 클래스의 확률을 구한다.

YOLO는 End-to-End 방식의 통합된 구조로 이루어져 있어 이미지를 convolutional neural network에 한 번 inference하는 것을 통해서 동시에 경계 박스와 클래스 확률을 구하게 된다. 이처럼 모델을 통합했기에 YOLO는 몇 가지 장점을 가지게 된다.

첫 번째로는 빠른 성능과 함께 다른 실시간 객체 검출 시스템 대비 2배 이상의 mean average precision 성능을 보여준다.

두 번째로는 convolutional neural network를 사용하는 것으로 기존의 R-CNN의 경우 이미지의 백그라운드에 의해서 객체 검출이 실패하는 반면 yolo는 각 클래스에 대한 표현에 대해 학습이 유리하다.

3.2.3.1 Unified Detection

YOLO의 핵심은 Unified Detection인데, 이는 convolutional neural network가 가장 중요한 역할을 한다. YOLO는 단일 convolutional neural network 모델 하나로 특징을 추출하고, 경계 박스를 계산하며, 클래스 분류 모두를 수행한다. 즉 원본 이미지 입력으로 받은 모델이 객체 검출에 필요한 모든 연산을 수행할 수 있다는 의미이며, 이것이 바로 YOLO의 Unified Detection의 개념이다.

YOLO에는 기존의 모델에 없었던 다양한 강점이 있다. 이미지 전체를 통해 경계 박스를 예측할 수 있다. 이는 모델이 이미지의 전역적인 특징점을 잘 이용하여 추론하게 된다. 그리고 Unified Detection이라는 용어 그대로 모델이 경계 박스 분석과 multi-class classification을 동시에 수행할 수 있

다. 이러한 강점들로 인해 YOLO는 높은 정확도를 유지하면서, end-to-end 학습이 가능하며, 실시간 추론 속도가 높은 것이다.

YOLO 모델은 입력 이미지를 $S \times S$ 그리드로 나눈다. 만약 어떤 객체의 중심이 특정 그리드 셀에 존재한다면, 해당 그리드 셀이 객체를 검출해야 한다.

각 그리드 셀은 B개의 경계 박스를 예측한다. 그리고 각 경계 박스마다 confidence scores를 예측하는데, confidence scores란 해당 경계 박스 내에 객체가 존재할 확률을 의미하며 0에서 1 사이의 값을 가진다. Confidence scores를 수식적으로 나타내면 다음과 같다.

$$\text{Pr}(\text{Object}) \times \text{IOU_truth}^{\text{pred}(1)}$$

만약 셀 안에 객체가 없으면 confidence scores는 0이다. 또한 confidence scores는 모델이 예측한 경계 박스와 ground truth 간의 IOU(intersection over union)이 동일할수록 좋다.

각 경계 박스는 x, y, w, h , confidence 5개의 값을 예측한다. x, y 는 경계 박스의 중심점이며, 각 그리드 셀마다 상대적인 값으로 표현된다. w, y 는 경계 박스의 width와 height이며, 전체 이미지에 대해 상대적인 값으로 표현된다. Confidence는 앞서 다룬 confidence score와 동일하다.

각 그리드셀마다 확률값 C 를 예측한다. C 는 조건부 클래스 확률(confiditional class probalibilities)인 $\text{PR}(\text{Class}|\text{Object})$ 이다. YOLO에서

는 그리드 당 예측하는 경계 박스의 개수와 상관없이 그리드 당 오직 하나의 클래스 확률만을 예측한다.

이를 수식적으로 나타내면 다음과 같다.

$$\Pr(\text{Class}|\text{Object}) \times \Pr(\text{Object}) \times \text{IOU}_{\text{truth pred}} = \Pr(\text{Class}) \times \text{IOU}_{\text{truth pred}}$$

경계 박스마다 class-specific confidence scores를 얻을 수 있다. 이 score는 해당 경계 박스에서 특정 클래스 객체가 나타날 확률 과 객체에 맞게 경계 박스를 올바르게 예측했는지를 나타낸다.

3.2.3.2 YOLOv5 구조

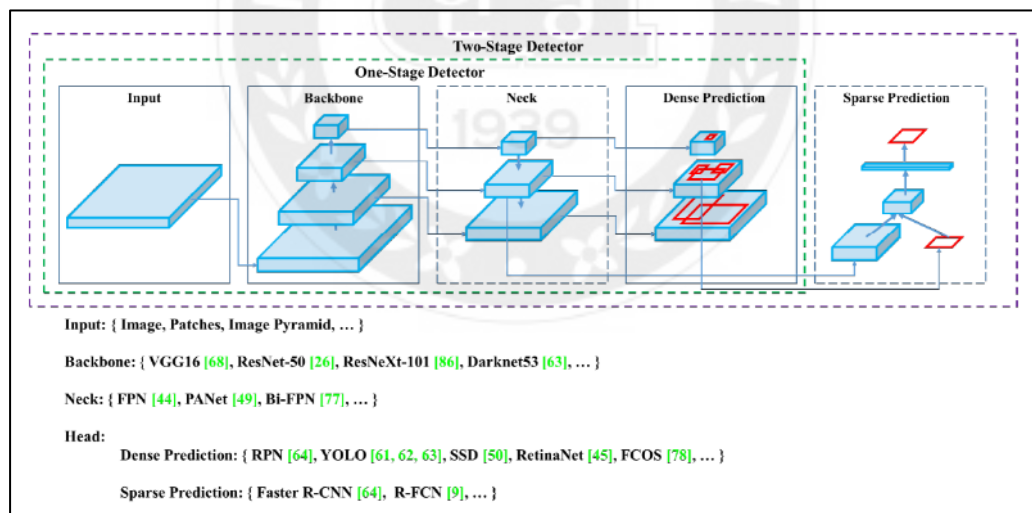


그림 16. Network 구조

구조는 크게 Backbone/Neck/Head 3 가지로 구분된다.

Backbone은 input image를 feature map으로 변형하는 부분이다.

Neck은 Backbone과 Head를 연결하는 부분으로써, Feature map을 Refinement, Reconfiguration 한다.

Head는 Backbone에서 추출한 feature map의 location을 하며, Predict classes와 경계 박스 작업이 이루어진다. Head는 크게 2가지로 구분되는데 Dense Prediction과 Sparse Prediction으로 구분된다



3.3. Image augmentation

Image augmentation이 필요한 이유는 overfitting을 방지하기 위해서이다. CNN은 representational power가 증가하게 되면 overfitting의 가능성도 증가하게 된다.

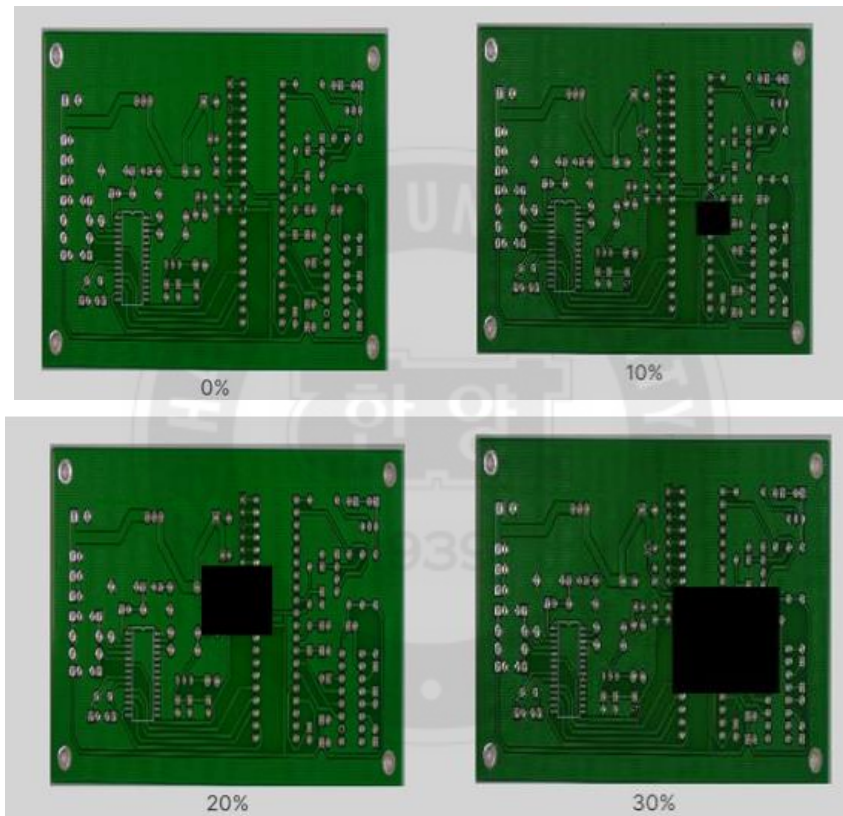


그림 17. Cutout

본 연구에서는 overfitting을 방지하기 위해 Image augmentation 기법 중 Cutout 기법을 사용하였으며, 다양한 cutout size를 사용하여 최적의 성능을 확인하였다.

Cutout size는 그림 14 와 같이 0%, 10%, 20%, 30% 로 적용하였다.

3.4. Prune

Convolutional Neural Networks (CNN) 모델이 커지면 임베디드 등 저사양 device에 deployment하기에 다음과 같은 제약 사항이 발생한다.

- 모델이 커지면서 parameter의 수가 매우 많아진다. 모델을 loading 하기 위해 큰 메모리 공간이 필요하게 된다.
- feature map을 저장하고 계산하는 run-time memory 도 많이 필요하게 된다. 특히 연산량 파워가 작은 임베디드 시스템에는 큰 문제가 될 수 있다.

위의 제약사항으로 본 연구에서는 network slimming이라는 Pruning 기법을 제안한다.

이 방법은 batch normalization의 scaling factor를 이용하여 가중치가 작은 채널을 제거하고 이를 여러 번 반복해 compact한 network를 구성하는 기법이다.

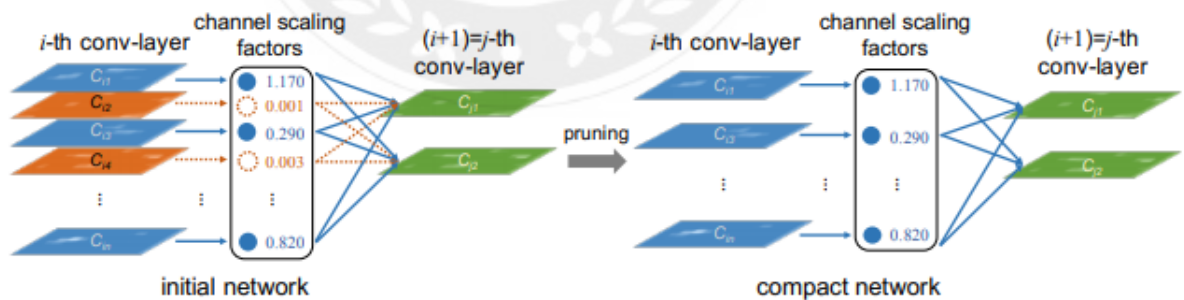


그림 18. Pruning

위의 그림과 같이 channel scaling factor를 통해 특정 가중치를 가지 못

하는 factor를 가지면 이를 제거하게 된다.



3.4.1 Network slimming

Pruning은 모델의 Sparsity를 야기한다. Sparsity는 모델의 개별 파라미터, filter, channel, layer 단에서 구현될 수 있는데, 개별 파라미터의 sparsity는 모델 압축률에 좋지만 sparse matrix 연산을 지원하는 별도의 라이브러리가 필요하며 layer 레벨의 sparsity는 특별한 연산 라이브러리가 필요하지 않지만 해당 layer를 없애기 때문에 효율적이지 못하다. 따라서 channel 레벨의 sparsity가 layer 레벨의 sparsity보다 더 유연하며, 구현이 용이하다는 특징으로 pruning에 일반적으로 많이 사용된다.

3.4.1.1 Scaling factors

Pruning할 channel을 결정하기 위해 channel 결과에 곱해지는 scaling factor γ 를 두고 일반적인 훈련 목적 함수에 더해 sparsity를 야기하는 L1 페널티 항을 추가하여 γ 가 작은 channel을 삭제한다. 이를 수식으로 나타내면 아래와 같다.

$$L = \sum_{(x,y)} l(f(x,W),Y) + \lambda \sum_{\gamma \in \Gamma} g(\gamma)$$

여기에서 Pruning할 channel을 선택하기 위해 γ 를 위한 메모리를 별도로 할당하지 않고 batch normalization의 scaling factor를 직접적으로 사용한다.

Batch normalization은 CNN 모델의 구성 요소가 된 기술로써 하나의

batch 의 통계를 이용하여 입력과 출력의 분포를 일정하게 해준다.

$$\check{z} = \frac{z_{in} - \mu_{\beta}}{\sqrt{\sigma_{\beta}^2 + \epsilon}}; z_{out} = \gamma \check{z} + \beta$$

위의 식에서 z_{in} , z_{out} 은 각각 batch normalization의 입력, 출력이며 γ , β 는 훈련가능한 parameter이다.

γ , β 의 차원은 layer의 feature map의 channel 수로 이 γ parameter를 pruning을 위한 scaling factor로 사용한다.

Batch normalization 없이 scaling layer를 따로 두었을 때 scaling layer와 convolutional layer 둘 다 선형 변환이므로 scaling layer의 값이 큰 의미가 없다. 또한, batch normalization 앞에 scaling layer를 따로 두었을 때는 scaling이 batch normalization에 의해 상쇄되고, 뒤에 scaling layer를 두었을 때는 scaling factor가 연속적으로 2개가 생기게 된다.

3.4.1.2 Channel pruning and fine-tuning

Scaling factor를 통해 pruning 할 channel을 선택하고 해당 channel에 연결된 incoming, outgoing parameter를 삭제하게 된다. Layer 마다 일정 비율이 아닌 네트워크 전체에 대해 scaling factor 분포를 통해 잘라낼 임계치를 설정하고 그 임계치보다 작은 scaling factor를 가지는 channel을 잘라낸다. 이후 fine-tuning을 통해 손실된 성능을 복구하게 된다.

3.4.1.3 Multi-pass scheme

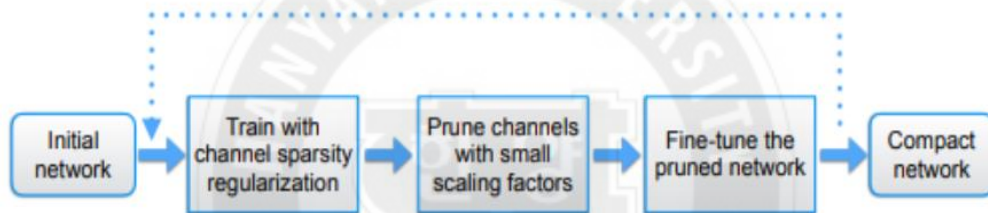


그림 19. flowchart of pruning

위의 방법을 한 번이 아닌 여러 번 반복한다. 이를 반복하여 네트워크가 점차적으로 slim해지며, 최종적으로 compact한 네트워크를 얻을 수 있다.

3.4.1.4 Handling cross layer connections and pre-activation structures

ResNet과 DenseNet 처럼 각 layer의 출력이 그 후 여러 layer의 입력으로 연결되는 경우 sparsity는 해당 layer의 들어오는 부분에 대해서만 sparsity를 적용하게 된다. 예를 들어 resnet의 한 layer를 구성하는 bottleneck block은 여러 단계의 batch normalization과 convolution이 있다. Bottleneck block 에서 마지막 convolution은 layer의 최종 출력이 되므로 이 convolution에 들어오는 방향으로만 pruning을 적용한다. 나머지 convolution 들은 최종 출력이 아니므로 들어오고 나가는 방향 모두 pruning을 적용한다.

마지막으로 위 작업을 효율적으로 하기 위해 channel selection layer를 만들어 중요하지 않은 채널의 인덱스를 기록해 마스킹 용도로 사용한다.

제 4장 성능평가

4.1. 실험

4.1.1 실험 환경 구성

운영체제	Ubuntu 18.04.6 LTS
CPU	Intel(R) Xeon(R) CPU @ 2.00GHz
메모리	13297228 kb
디스크	167 Gb
GPU	NVIDIA-SMI
Cuda	10.1
Python	3.7.15

환경 구성을 Google CoLaboratory를 사용하여 구현하였다.

Google CoLaboratory는 클라우드 기반 개발환경이다. 구글의 CPU, TPU, RAM을 이용하여 ‘주피터 노트북’과 같은 개발 환경을 클라우드 서비스로 제공하는 서비스이다.

4.1.2 성능 평가 방법

PCB 불량 검출 시스템의 성능(정확도)을 일반적인 객체 검출 모델 평가 지표인 Mean average precision(mAP)을 통해 확인한다.

4.1.3 실험 시나리오

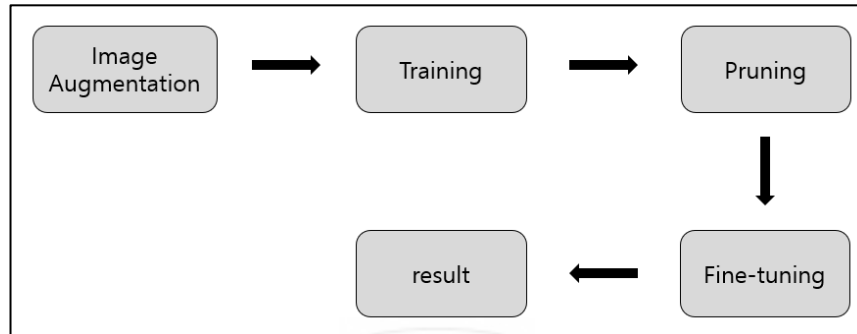


그림 20. flowchart of PCB defect detection system

- Image Augmentation, Training - 기존의 Training 결과와 다양한 크기로 cutout 된 이미지를 사용하여 Training 결과를 비교하여 최적의 cutout size 를 찾는다.
- Pruning - 학습이 완료된 객체 검출 모델의 가중치들을 희소성이라는 지표를 도출한 뒤 해당 지표를 기준으로 일정 비율의 parameter들을 네트워크 그래프에서 pruning하여 객체 검출 모델의 경량화를 확인한다.
- Fine-tuning - Fine-tuning을 통해 모델이 경량화 되었지만 정확도 성능이 유지됨을 확인한다.

4.2. 실험 결과

4.2.1. Image Augmentation, Training

YOLO 객체 검출 모델의 PCB 불량 검출 정확도와 다양한 Cutout size를 적용한 결과를 비교하였다. Image augmentation을 적용하기 이전의 검출 성능은 84.5% 이다. 학습 데이터 셋에 Cutout을 10% 적용하였을 때의 검출 성능은 87.3% 이다. Cutout을 20% 적용하였을 때의 검출 성능은 85.7% 이다. Cutout을 30% 적용하였을 때의 검출 성능은 80.4% 이다. 실험을 통해 최적의 Cutout size가 10% 임을 확인하였다.

Image augmentation을 적용하기 이전 보다 Cutout size를 10% 적용하였을 때 검출 성능이 약 3%가 향상됨을 확인하였다.

Defect system	Defect_Case	mAP50-95
YOLOv5	Missing_Hole	0.848
YOLOv5	Mouse_Bite	0.843
YOLOv5	Short	0.845
YOLOv5 + cutout 10%	Missing_Hole	0.867
YOLOv5 + cutout 10%	Mouse_Bite	0.879
YOLOv5 + cutout 10%	Short	0.873
YOLOv5 + cutout 20%	Missing_Hole	0.864
YOLOv5 + cutout 20%	Mouse_Bite	0.850

YOLOv5 + cutout 20%	Short	0.857
YOLOv5 + cutout 30%	Missing_Hole	0.813
YOLOv5 + cutout 30%	Mouse_Bite	0.789
YOLOv5 + cutout 30%	Short	0.810

표 1. image augmentation 결과



4.2.2. Pruning

희소성 훈련이 완료된 weight를 이용하여 Pruning 기법을 적용하였다.
아래의 표를 통해 원래 채널과 pruning 후 남은 채널을 확인할 수 있다.

Layer name	Origin channels	Remaining channels
Model.0.bn	32	32
Model.1.bn	64	64
Model.2.cv1.bn	32	32
Model.2.cv2.bn	32	32
Model.2.cv3.bn	64	64
Model.2.m.0.cv1.bn	32	32
Model.2.m.0.cv2.bn	32	32
Model.3.bn	128	128
Model.4.cv1.bn	64	64
Model.4.cv2.bn	64	61
Model.4.cv3.bn	128	128
Model.4.m.0.cv1.bn	64	64
Model.4.m.0.cv2.bn	64	64

Model.4.m.1.cv1.bn	64	64
Model.4.m.1.cv2.bn	64	64
Model.5.bn	256	256
Model.6.cv1.bn	128	128
Model.6.cv2.bn	128	37
Model.6.cv3.bn	256	256
Model.6.m.0.cv1.bn	128	128
Model.6.m.0.cv2.bn	128	128
Model.6.m.1.cv1.bn	128	128
Model.6.m.1.cv2.bn	128	128
Model.6.m.2.cv1.bn	128	128
Model.6.m.2.cv2.bn	128	128
Model.7.bn	512	486
Model.8.cv1.bn	256	256
Model.8.cv2.bn	256	107
Model.8.cv3.bn	512	504
Model.8.m.0.cv1.bn	256	256

Model.8.m.0.cv2.bn	256	256
Model.9.cv1.bn	256	147
Model.9.cv2.bn	512	238
Model.10.bn	256	196
Model.13.cv1.bn	128	100
Model.13.cv2.bn	128	111
Model.13.cv3.bn	256	204
Model.13.m.0.cv1.bn	128	31
Model.13.m.0.cv2.bn	128	44
Model.14.bn	128	100
Model.17.cv1.bn	64	64
Model.17.cv2.bn	64	63
Model.17.cv3.bn	128	127
Model.17.m.0.cv1.bn	64	64
Model.17.m.0.cv2.bn	64	64
Model.18.bn	128	128
Model.20.cv1.bn	128	125

Model.20.cv2.bn	128	127
Model.20.cv3.bn	256	256
Model.20.m.0.cv1.bn	128	125
Model.20.m.0.cv2.bn	128	127
Model.21.bn	256	256
Model.23.cv1.bn	256	167
Model.23.cv2.bn	256	236
Model.23.cv3.bn	512	511
Model.23.m.0.cv1.bn	256	190
Model.23.m.0.cv2.bn	256	251

표 2. channel pruning

Pruning 기법 적용 결과를 아래의 표에서 확인할 수 있다.

기존의 성능에 비해 pruning 기법을 적용한후 성능이 떨어졌음을 확인하였으며, parameter는 기존 대비 32% 의 parameter가 줄어듦을 확인하였다.

Defect system	Defect_Case	mAP50-95	parameter
YOLOv5 + cutout 10%	Missing_Hole	0.867	7027720
YOLOv5 + cutout 10%	Mouse_Bite	0.879	
YOLOv5 + cutout 10%	Short	0.873	
YOLOv5 + cutout 10% + pruning 30%	Missing_Hole	0.816	4841190
YOLOv5 + cutout 10% + pruning 30%	Mouse_Bite	0.823	
YOLOv5 + cutout 10% + pruning 30%	Short	0.819	

표 3. result of pruning

4.2.3. Fine-tuning

채널을 잘라내면서 손실된 성능을 fine-tuning을 통해 성능을 복구하였다.
결과적으로 정확도 성능을 유지하면서 모델의 경량화를 수행하였다.

Defect system	Defect_Case	mAP50-95	parameter
YOLOv5	Missing_Hole	0.848	7027720
YOLOv5	Mouse_Bite	0.843	
YOLOv5	Short	0.845	
Proposed system	Missing_Hole	0.84	4841190
Proposed system	Mouse_Bite	0.868	
Proposed system	Short	0.862	

표 4. YOLOv5 VS Proposed system

제 5장 결론

5.1. 결론

YOLO 개선을 통한 PCB 불량 검출 시스템 설계 및 구현을 목표로 본 연구를 진행하였다.

본 연구에서 주요 내용은 아래와 같다.

1) 딥러닝을 이용한 PCB 불량 검출 시스템 개발

객체 검출에 범용적으로 사용되는 딥러닝 모델인 YOLO를 이용하여 PCB 불량 검출 시스템을 개발하였다. Missing hole, short, mise bite의 3가지 불량 케이스를 대상으로 불량 검출을 시행하였을 때 평균 84%의 검출 성능을 확인하였다.

2) 이미지 어그멘테이션 기법을 통한 성능 개선

적은 수의 이미지 데이터를 가지고 Neural Network를 훈련할 때 발생하는 문제인 과적합(overfitting)을 해결하기 위해 이미지 전처리 기법인 이미지 어그멘테이션(Image augmentation)기법으로 해결하였다. 실험을 통해 최적의 이미지 어그멘테이션을 적용하였으며, 이미지 어그멘테이션을 적용하였을 때 기존 시스템 대비 3%의 성능 향상을 확인하였다.

3) 모델 경량화를 통한 성능 개선

속도가 중요한 PCB 제조 환경을 위한 딥러닝 모델 경량화 기법인 가지치기(Pruning)기법을 추가하여 PCB 불량 검출 시스템을 설계 및 구현하였다.

모델 경량화를 통해 PCB 불량 검출 시스템의 검출 성능을 유지하면서 기존 시스템 대비 parameter의 30% 감소를 확인하였다.

참고문헌

- [1] 권형안. “PCB 공급업체의 품질 인덱스 활용한 신뢰성 관리” , AEM.
- [2] 이윤식. “중국 인쇄회로기판(PCB) 시장동향, Kotra.
- [3] Lynn Shin. “YOLOv5 학습 튜토리얼1.” ,
<https://lynnshin.tistory.com/47>.
- [4] Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi. “You Only Look Once: Unified, Real-Time Object Detection.
- [5] Terrance DeVries, Graham W. Taylor. Improved Regularization of Convolutional Neural Networks with Cutout
- [6]Zhuang Liu, Jianguo Li, Zhiqiang Shen, Gao Huang, Shoumeng Yan, Changshui Zhang. “Learning Efficient Convolutional Networks through Network Slimming.”
- [7] 이진수, 이상광, 김대욱, 홍승진, 양성일. “딥러닝 기반 객체 인식 기술 동향”

ABSTRACT

PCB Defect Detection System Based on improved YOLO

Lee mingyun

Dept of Electrical Engineering and Computer Science

Graduate School of Engineering

Hanyang University

The object detection technology of the PCB defect detection system proposed in this study is one of the important studies in computer vision technology. Recently, various object detection algorithms have been proposed, and since various targets have various states, they should be able to detect targets based on various criteria. We propose an improved You Only Look Once (YOLO) to improve detection performance for targets with different states. The main contents of this study are as follows.

- Development of PCB Defect Detection System using Deep Learning

A PCB defect detection system was developed using YOLO, a deep learning model commonly used for object detection.

- Improved performance through image agglomeration techniques

To solve overfitting, a problem that occurs when training neural networks with a small number of image data, we solve it with Image Augmentation, an image preprocessing technique.

- Improved performance through lighter models

A PCB defect detection system was designed and implemented by adding Pruning, a deep learning model slimming technique for PCB manufacturing environments where speed is important.



감사의 글

석사 과정을 마무리하며 이렇게 감사의 글을 쓰니 감회가 새롭습니다. 이 자리까지 많은 사람들의 도움으로 지금까지 올 수 있었습니다. 그 감사함을 조금이나마 이 지면을 통해 표현하고자 합니다.

가장 먼저, 석사 과정을 무사히 수료할 수 있도록 세심하게 지도해주신 조인휘 교수님께 깊은 존경과 감사의 말씀을 올립니다.

팬데믹 상황으로 어려운 환경속에서도 늘 웃음으로 반겨 주시고, 석사 과정에서 길을 잃지 않도록 먼저 손 내밀어 주신 113기 동기분들에게도 감사를 전합니다.

학업을 이어가기 쉽지 않은 순간에서도 언제나 옆에서 격려와 응원으로 힘이 되어주고 쓰러지지 않도록 중심을 잡아주는 친구들에게도 감사를 전합니다.

항상 배움을 놓지 않고 본이 되어 주시는 어머니, 가족을 위해 늘 희생하는 동생에게 무한한 미안함과 감사를 드립니다.

모든 고마움을 잊지 않고 갚아 나갈 수 있도록 하겠습니다.
감사합니다.

2023년 2월

이 민 균

연구 윤리 서약서

본인은 한양대학교 대학원생으로서 이 학위논문 작성 과정에서 다음과 같이 연구 윤리의 기본 원칙을 준수하였음을 서약합니다.

첫째, 지도교수의 지도를 받아 정직하고 엄정한 연구를 수행하여 학위논문을 작성한다.

둘째, 논문 작성시 위조, 변조, 표절 등 학문적 진실성을 훼손하는 어떤 연구 부정행위도 하지 않는다.

셋째, 논문 작성시 논문유사도 검증시스템 "카피킬러"등을 거쳐야 한다.

2022년 12월 14일

학위명 : 석사

학과 : 전기·전자·컴퓨터공학과

지도교수 : 조인휘

성명 : 이민균



한 양 대 학 교 공 학 대 학 원 장 귀 하

Declaration of Ethical Conduct in Research

I, as a graduate student of Hanyang University, hereby declare that I have abided by the following Code of Research Ethics while writing this dissertation thesis, during my degree program.

"First, I have strived to be honest in my conduct, to produce valid and reliable research conforming with the guidance of my thesis supervisor, and I affirm that my thesis contains honest, fair and reasonable conclusions based on my own careful research under the guidance of my thesis supervisor.

Second, I have not committed any acts that may discredit or damage the credibility of my research. These include, but are not limited to : falsification, distortion of research findings or plagiarism.

Third, I need to go through with Copykiller Program(Internet-based Plagiarism-prevention service) before submitting a thesis."

DECEMBER 14, 2022

Degree :	Master
Department :	DEPARTMENT OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE
Thesis Supervisor :	INWHEE JOE
Name :	LEE MINGYUN

(Signature)