

고려대학교  
빅데이터 연구회

# KU-BIG

---

Image Segmentation 2  
06.07 최종발표

이민수 정의석 김혜연 유승완 강유정 김혜빈



# Contents

I

**Review**

II

**Outcome analysis**

III

**Difficulties & Limitations**

IV

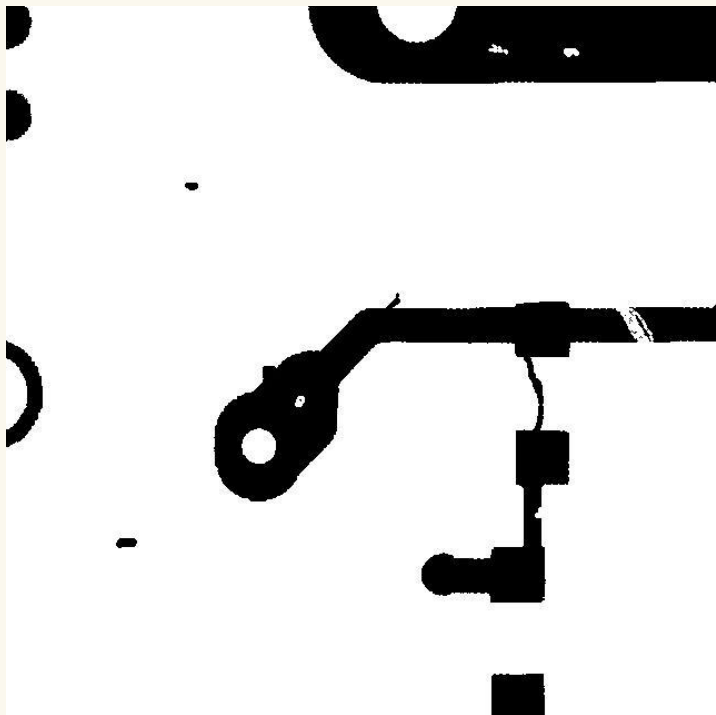
**Expected Value**

V

**Conclusion**

# Review

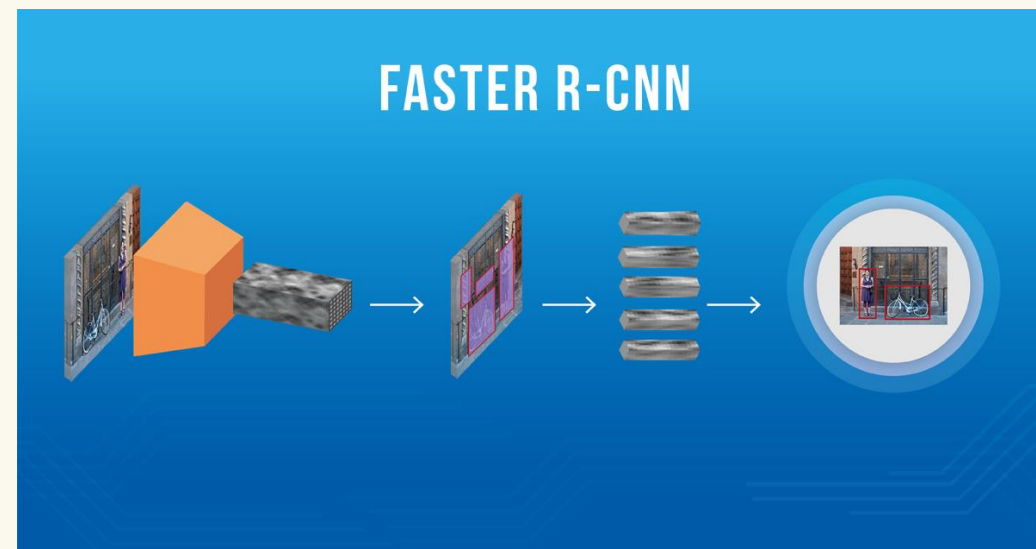
Input Data Set



[PCB Data]



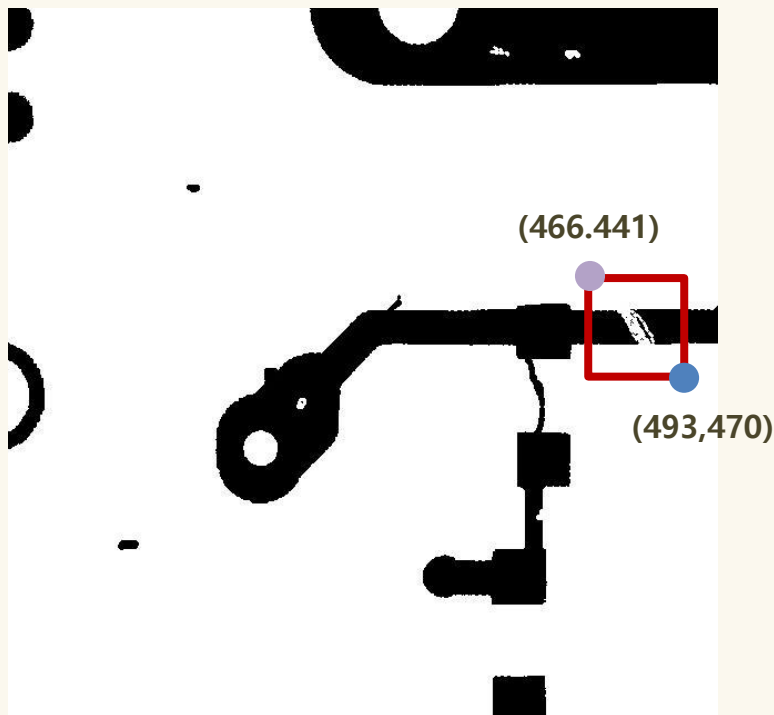
Adjusted Algorithm



[Faster RCNN]

# Review-Input Data

[PCB Defect]

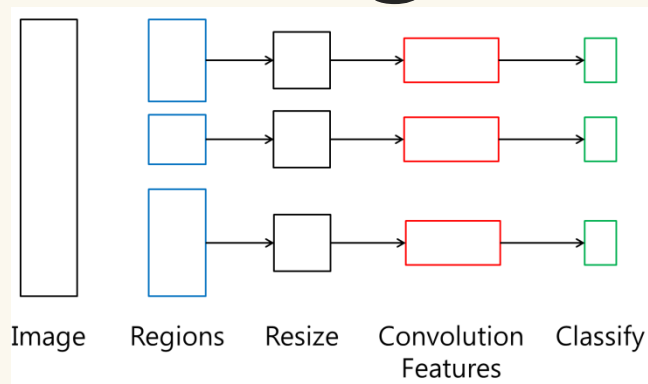


466	441	493	470	3
454	300	493	396	2
331	248	364	283	4
221	314	253	350	4
151	149	182	175	5

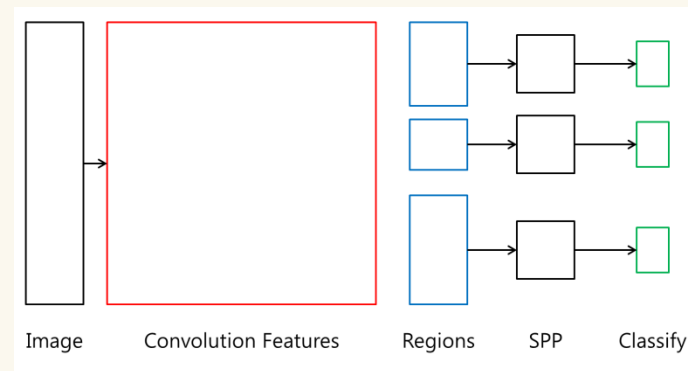
- 1 - open
- 2 - short
- 3 - mousebite
- 4 - spur
- 5 - copper
- 6 - pin hole

- 640 X 640 pixels의 이미지 데이터와 이에 대응하는 텍스트 데이터로 구성되어 있음
- 1500개의 이미지 데이터가 각각 3~12개의 defect를 가지고 있음
- 텍스트 데이터에 각 defect 위치의 모서리의 좌표 (x1,y1), (x2,y2) 와 defect 타입이 기록 되어 있음

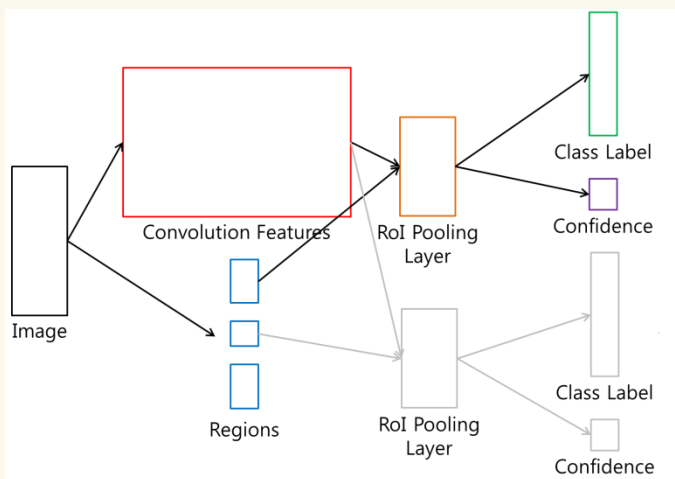
# Review-Algorithms



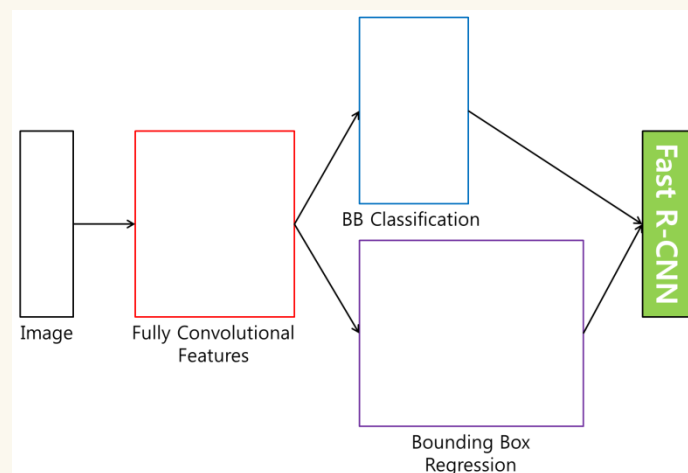
R-CNN



SPP net



Fast R-CNN



Faster R-CNN

# Review

KU-BIG, 빅데이터 연구회

## 분석 목표

Binary Classification

advance →

Multiclass Classification

Step 1 :

PCB(인쇄회로기판) 양산  
시 불량품 classification

Step 2 :

어떤 종류의 불량인지에  
대한 Classification

[4/11 목표 설정]

KU-BIG, 빅데이터 연구회

## 현 진행상황

**Binary Classification**

- 양품 or 불량품
- CNN 사용
- data resize (640\*640 -> 320\*320)를 통해 메모리 문제 해결
- 현재 Test Accuracy 99.3%까지 진행함

```

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.1, random_state=0)

model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
hist = model.fit(X_train, y_train,
                batch_size=60,
                epochs=25,
                verbose=1,
                validation_data=(X_test, y_test))
    
```

```

score = model.evaluate(X_test, y_test, verbose=0)
print('Test loss:', score[0])
print('Test accuracy:', score[1])
    
```

Test loss: 0.023883792920177487  
Test accuracy: 0.9931972789115646

[5/10 1차 목표 달성]

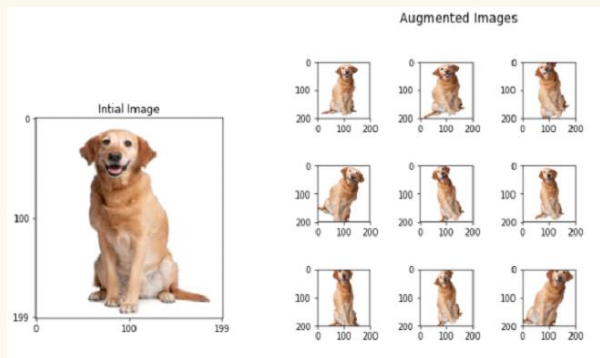
# Review - previous difficulties

컴퓨터 성능



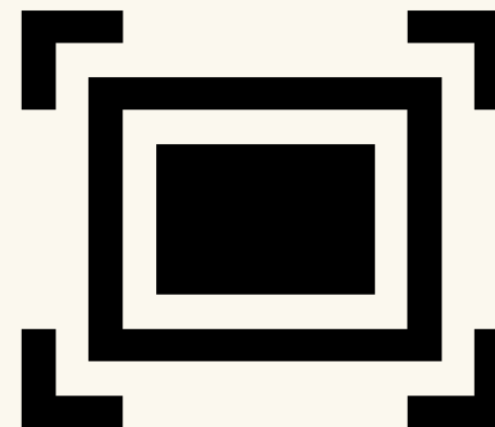
Google Co-lab

Data 수 부족



Data Augmentation

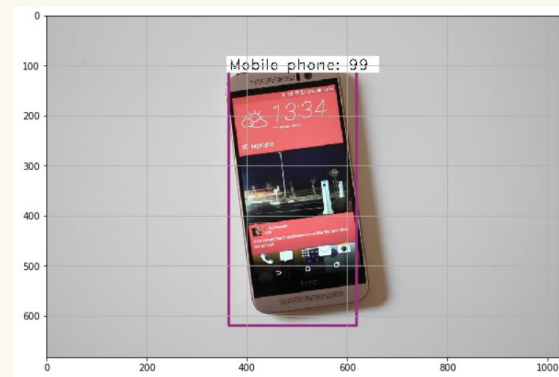
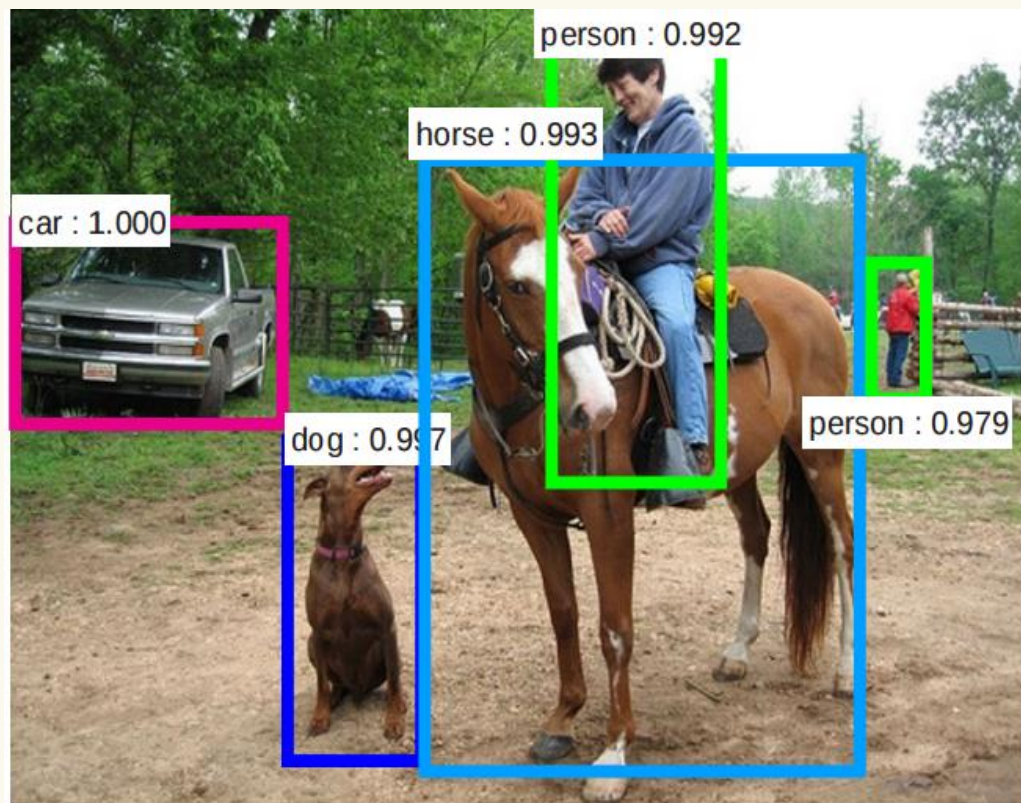
Data Size 문제



Data Resize



# Outcome analysis





# Outcome analysis

```

497/500
Elapsed time = 0.363126277923584
6 AP: 0.5173526201421325
4 AP: 0.4776095829573314
1 AP: 0.541059587112811
2 AP: 0.3171680468506309
3 AP: 0.5095988421927837
5 AP: 0.617158049582085
mAP = 0.496657788139629
498/500
Elapsed time = 0.3779468536376953
6 AP: 0.5173809734240256
4 AP: 0.4776095829573314
1 AP: 0.5402277977036402
2 AP: 0.3173382497197229
3 AP: 0.5085963824492382
5 AP: 0.6181075264313219
mAP = 0.4965434187808801
499/500
Elapsed time = 0.36393213272094727
6 AP: 0.5179402482658587
4 AP: 0.47755068928342637
1 AP: 0.5402277977036402
2 AP: 0.3173382497197229
3 AP: 0.5090202325995723
5 AP: 0.6177588022309037
mAP = 0.4966393366338541

mean average precision: 0.5048327569996517
    
```

각 image에서 1~6번까지 error에 대한 precision

Error들의 mean precision

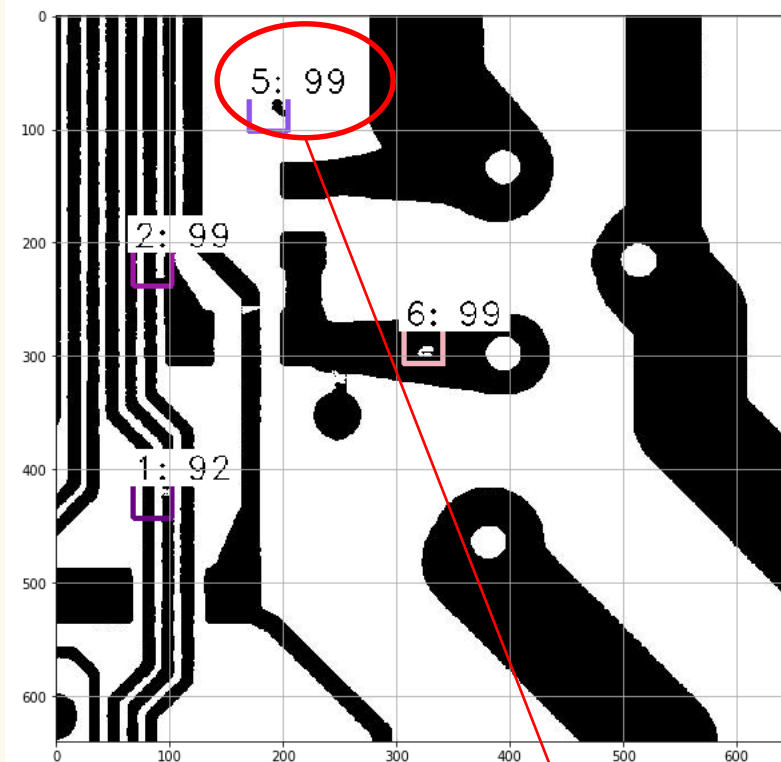
500개의 test image data의 mean average precision

# Outcome analysis

Measure	outcome
Precision	0.50432647
Recall	0.4222893
F-measure	0.1770682146

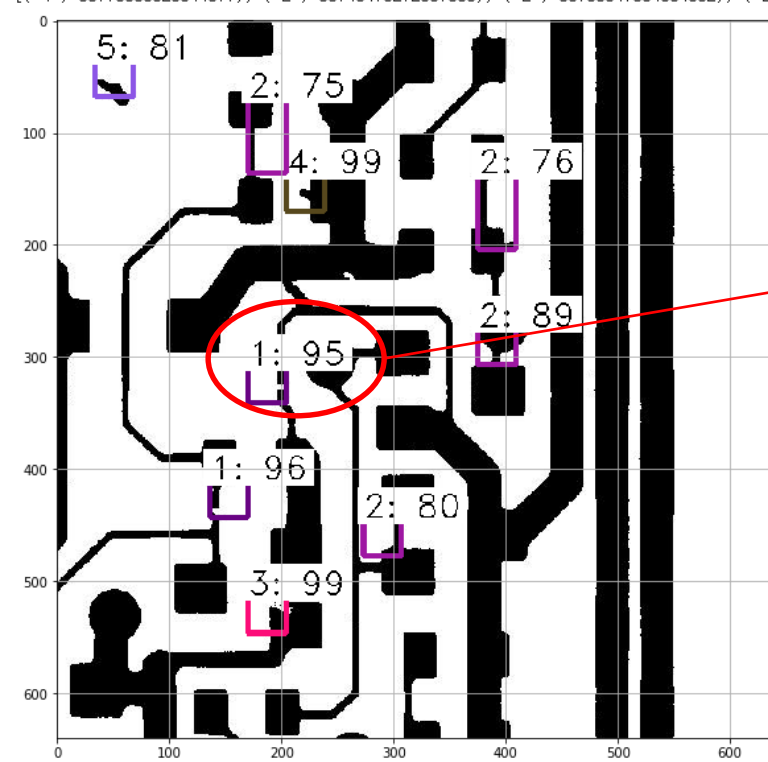
# Outcome analysis

Elapsed time = 0.5965328216552734  
 [('2', 99.29976463317871), ('6', 99.96336698532104), ('5', 99.95489716529846), ('1', 92.11323857307)



Error 5일 확률 99%

Elapsed time = 0.5961568355560303  
 [('4', 99.78858828544617), ('2', 89.45416212081909), ('2', 80.88847994804382), ('2', 76.6838014125824), ('2',



Error 1일 확률  
95%

# Difficulties & Limitations

## Colab RAM 부족 문제

GPU 메모리 사용량이 한도에 가까워짐

GPU가 메모리 한도에 거의 도달했습니다. 이 세션에서 추가 메모리를 사용할 수 없게 됩니다. 현재 13.14 GB/14.73 GB를 사용 중입니다. 일부 세션을 종료하여 GPU 메모리를 확보하시겠습니까? 이렇게 하면 종료하는 세션의 상태가 삭제됩니다.

[무시](#) [세션 관리](#)

## Data를 나눠서 학습

# Difficulties & Limitations

물리적인 시간 문제



- CoLab을 사용해도 학습을 하는데 시간이 오래 소요됨.
- Anchor box Size를 error size에 맞게 조절해서 한정하였으면 더 좋은 결과가 나올 것이라고 예측.
- Open image data set이 아닌 PCB image data set으로 feature map에 대해 pretraining 시켰으면 더 좋은 결과 예측.

# Expected Value

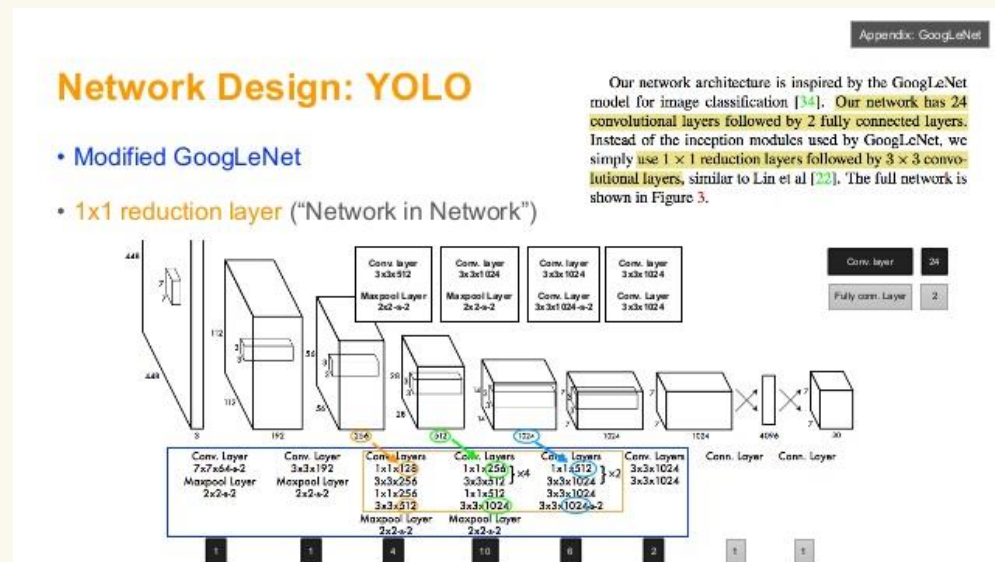
## Smart Factory에 적용



- LG CNS의 Factova
  - 공장의 효율성과 불량률 최소화. 데이터 기반의 실시간 자율 운영 공장 → AI 빅데이터 플랫폼 DAP의 딥러닝을 통해 품질 검사 정확도를 99.7% 까지 개선하는 등 생산 효율 극대화
  - 소터관리시스템(SMS): 자동으로 식자재 배송처 정보를 분석하고 분류, 소터 장비의 운영 상태를 실시간으로 확인해 작업 내용과 결과 통계를 산출
- 삼성SDS의 Nexplant, SK C&C의 Scala, 아디다스의 Speed Factory

# Conclusion - Another Algorithms

YOLO

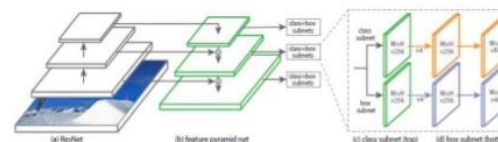


One-stage detection으로 기존의 R-CNN 알고리즘들보다 더 빠른 속도, but 낮은 정확도

# RetinaNet

## RetinaNet Detector

- Backbone : ResNet + **Feature Pyramid Network(FPN)**  
Constructs pyramid with levels P3 through P7, where  $i$  indicates pyramid level ( $P_i$  has resolution  $2^i$  lower than the input). All pyramid levels have  $C = 256$  channels.
- Two task-specific subnetworks :
  - i) The first subnet performs convolutional object classification on the backbone's output;
  - ii) The second subnet performs convolutional bounding box regression.



One-stage detector만큼 빠르면서도 기존의 모든 최고 성능의 detector들을 능가



# Conclusion





# Conclusion - Project Lila Complete



A stylized illustration of a person from the chest up, wearing a grey suit jacket, a white shirt, and a dark tie. The person's head is partially visible at the top, with a red circle representing the mouth. A large, thick black speech bubble originates from the mouth and points towards the left. Inside the speech bubble, the text "Do you have any question?" is written. The background is a solid light beige color.

Do you have any  
**question?**

Thank you  
for your attention.