

Ensemble-Object-Detection-using-Detectron2

Ensemble Object Detection Algorithm using Detectron2

Result

Models	Box AP @(IoU=0.50:0.95, area=all, maxDets=100)
faster_rcnn_R_50_C4_1x.yaml	0.357
faster_rcnn_R_50_DC5_1x.yaml	0.373
retinanet_R_50_FPN_1x.yaml	0.374
Ensemble Model	0.403 (+0.029)

Getting Started

1. Environments

- OS: Ubuntu20.04
- CUDA: 11.3
- Pytorch==1.10
- Detectron2==0.6 ([d1e0456](#))

2. Installation

```
sudo apt-get install -y python3-dev python3-venv
python3 -m venv env
source env/bin/activate
python -m pip install pip -U
python -m pip install -r requirements.txt
python -m ipykernel install --user --name env --display-name
ensemble_detectron2
python -m pip install
"git+https://github.com/facebookresearch/detectron2@v0.6"
```

3. Run (reproduce results)

- open "detectron2_ensemble.ipynb" and run all cells
- if you don't have GPU, you can utilize results json files without GPU running.
 - open "detectron2_ensemble_no_gpu.ipynb" and run all cells
- [Google Colab](#)

Explanation

Model Selection

- 모델 선택은 빠른 테스트 사이클을 구축하여 더 많은 테스트를 해볼 수 있도록, [speed benchmark](#) 기준으로 속도가 빠른 `_Detectron2_`를 선정하였다.

Implementation	Throughpu (img/s)
Detectron2	62
MMDetection	61
Detectron	19

- Ensemble을 통해 Box mAP를 높이는 것을 목적으로 하고, mAP 상승 gap 을 최대로 하는 것을 목적으로 했기 때문에, Pre-trained Model 중 Box mAP 가 낮은 모델을 선정하였으며, COCO Dataset Evaluation 이 정상적으로 수행되지 않은 *Fast R-CNN R50-FPN* 모델은 제외 하였다.
 - 선정 모델 (configuration name)
 - [faster_rcnn_R_50_C4_1x.yaml](#)
 - [faster_rcnn_R_50_DC5_1x.yaml](#)
 - [retinanet_R_50_FPN_1x.yaml](#)

Ensemble Method

Object detection 분야에서 많이 쓰이고 성능이 검증된 **Weighted Boxes Fusion** 방식을 사용하여 ensemble 을 적용하였고, 0.374에서 0.403으로 0.029 증가한 결과를 얻을 수 있었다.

시작은 ensemble을 위해서 기본적으로 voting 방식을 취하기 위해 hueristic 한 method를 설계해보려고 시도하였다. 하지만 아래 그림과 같이 score (confidence) 대비 over-detection 되는 object들이 너무 많았고, 이러한 여러개의 모델들에서 더 높은 정확도를 가지는 box를 찾아내기가 쉽지 않았다.

img_id: 522713, idx: 17



위 그림에서 흰색 박스가 ground truth 이고, 각 색상별로 다른 모델의 detection 결과를 시각화 하였다. 보이는 것처럼 over-detection 된 물체가 너무 많아 ensemble 전략을 만들기가 쉽지 않았다.

위와 같은 이유로 object detection 분야에서 많이 쓰이는 ensemble method를 찾아보았고, NMS (Non-Maximum Suppression), Soft NMS 방법을 개선한 Weighted Boxes Fusion 방법이 많이 사용되고 있으며, 좋은 성능을 보여주고 있음을 확인하였고, 이를 적용해보았다.

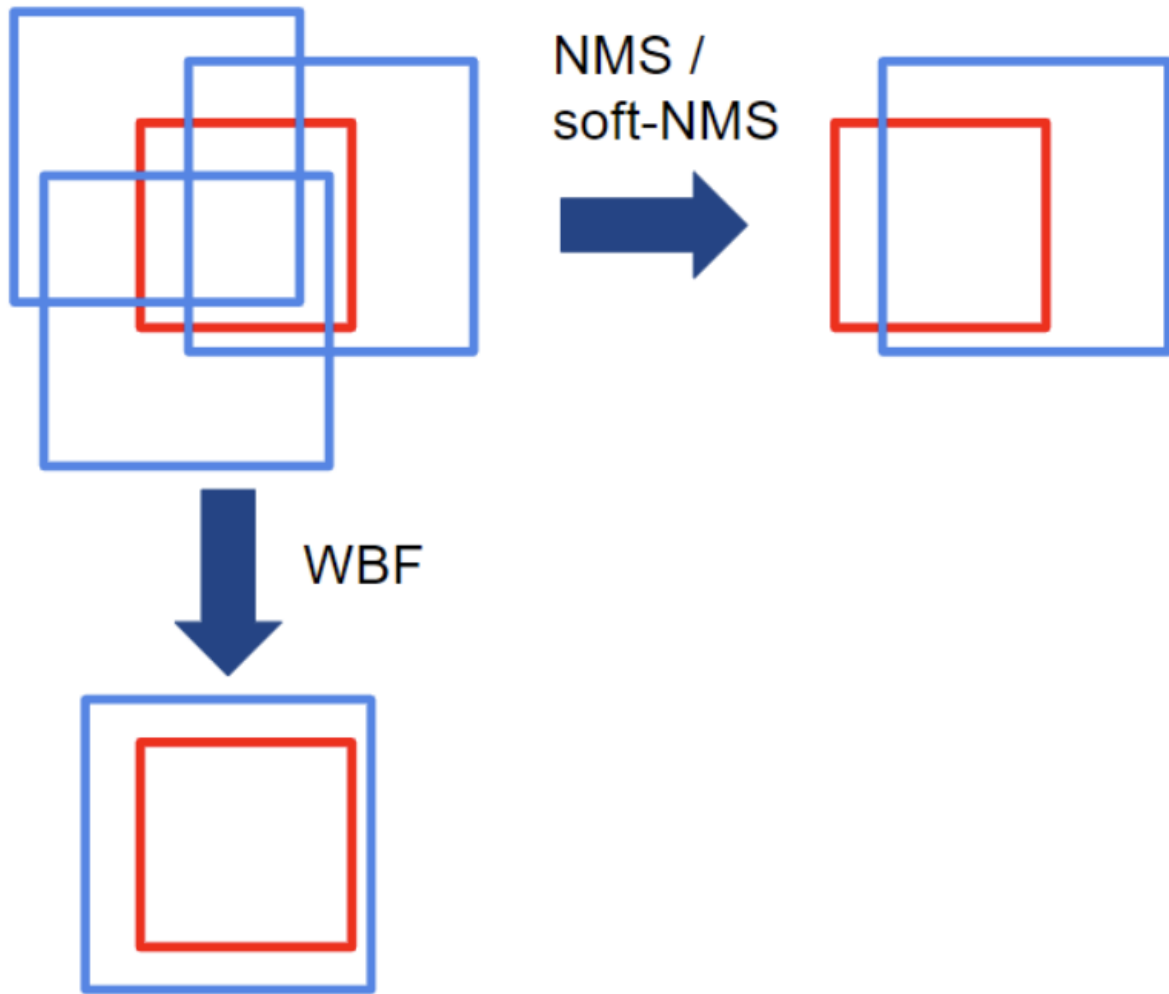


Figure 2: Schematic illustration of NMS/soft-NMS vs. WBF outcomes for an ensemble of inaccurate predictions. Blue – different models' predictions, red – ground truth.

Weighted Boxes Fusion 방법은 기존에 NMS/Soft NMS 방법이 IoU Threshold 값에 따라 필요없는 detection 정보를 지워버리는 것과는 달리, detection score를 이용하여 weighted box를 구하는 방법을 취한다. 이를 통해 하나의 물체를 여러 모델이 검출하였을때 score가 가장 높은 하나의 결과만 사용하게 되는 기존 방법들과 달리 모든 검출 결과를 score에 따라 반영하는 fusion된 결과를 얻을 수 있다. Weighted Boxes Fusion 방식은 여러모델에서 검출된 Box들을 clustering 하고 이를 fusion 한 Box list를 구성한다. 새로운 box는 fusion box list와 비교하고 IoU threshold를 넘기면 clustering된 box list에 추가하고 fusion 결과를 업데이트 하는 방식을 취한다.

Reference

- [Detectron2](#)
- [Detectron2 Install](#)
- [COCO Dataset](#)
- [Weighted Boxes Fusion](#)