1. YOLO

(1) 논문 : You Only Look Once : Unified, Real-Time Object Detection(https://www.cv-foundation.org/openaccess/content\_cvpr\_2016/papers/Redmon\_You\_Only\_Look\_CVPR\_2016\_paper.pdf)

- Compared to state-of-the-art detection systems, YOLO makes more localization errors but is less likely to predict false positives on background.

다른 최첨단 탐지 시스템들에 비교하여 YOLO는 보다 많은 localization errors를 만들어 냈지만, background에 대한 FP를 예측할 가능성은 낮다.

- Using our system, you only look once (YOLO) at an image to predict what objects are present and where they are. YOLO is refreshingly simple. A single convolutional network simultaneously predicts multiple bounding boxes and class probabilities for those boxes.

우리의 시스템은, 이미지를 한 번 보고 어떤 물체가 어디에 있는지 예측한다. Yolo는 매우 단순하다. 단일 컨볼루셔널 네트워크는 여러 바운딩 상자와 해당 상자의 확률을 동시에 예측한다.

- First, YOLO is extremely fast. Since we frame detection as a regression problem we don't need a complex pipeline. This means we can process streaming video in real-time with less than 25 milliseconds of latency. Furthermore, YOLO achieves more than twice the mean average precision of other real-time systems.

첫째, YOLO는 복잡한 pipeline이 필요하지 않기 때문에 매우 빠릅니다. 우리는 25 밀리 초 미만의 대기 시간으로 스트리밍 비디오를 실시간으로 처리할 수 있습니다. 또한 YOLO는 다른 실시간 시스템의 평균 정확도의 두 배 이상을 달성합니다.

- Second, Unlike sliding window and region proposal-based techniques, YOLO sees the entire image during training and test time so it implicitly encodes contextual information about classes as well as their appearance. Fast R-CNN, a top detection method, mistakes background patches in an image for objects because it can’t see the larger context. YOLO makes less than half the number of background errors compared to Fast R-CNN.

둘째, 슬라이딩 윈도우나 Region Proposal 기반 기술과는 달리, YOLO는 training 시간 동안 전체의 이미지를 본다. Fast R-CNN은 larger context를 볼 수 없기 때문에 객체의 배경 패치를 잘못 처리 한다. 하지만 YOLO는 Fast R-CNN과 비교하여 백그라운드오류의 수를 절반으로 줄인다.

- YOLO still lags behind state-of-the-art detection systems in accuracy. While it can quickly identify objects in images it struggles to precisely localize some objects, especially small ones.

YOLO는 최첨단 탐지 시스템들 보다 정확도가 떨어진다. 그것이 이미지 속의 대상을 신속하게 식별하긴 하지만, localize object나 작은 대상들에 대해 정확도가 떨어진다.

(2) 논문 : YOLOv3 An Incremental Improvement (https://pjreddie.com/media/files/papers/YOLOv3.pdf)

- YOLOv3 is extremely fast and accurate. In mAP measured at .5 IOU YOLOv3 is on par with Focal Loss but about 4x faster. Moreover, you can easily tradeoff between speed and accuracy simply by changing the size of the model, no retraining required!

YOLOv3는 매우 빠르고 정확하다. 0.5 IOU로 측정한 mAP에서 Focal과 동등한 Loss였지만 4배가 빨랐다. 또한 다시 training하지 않아도, 모델의 size를 변경하여 speed와 accuracy를 설정할 수 있다.

- It looks at the whole image at test time so its predictions are informed by global context in the image. It also makes predictions with a single network evaluation unlike systems like R-CNN which require thousands for a single image.

This makes it extremely fast, more than 1000x faster than R-CNN and 100x faster than Fast R-CNN.

Test time 동안 전체 이미지를 보고, 이미지 안의 Global context를 통해 예측한다. 또한 이미지에 수 천개의 네트워크가 필요한 R-CNN과 달리 단일 네트워크로 예측을 한다. 이로 인해 R-CNN보다 1000배 이상 빠르며, Fast R-CNN보다 100배 이상 빠르다.

2. YOLO와 Fast R-CNN Ensemble

(1) 논문 : You Only Look Once : Unified, Real-Time Object Detection(https://www.cv-foundation.org/openaccess/content\_cvpr\_2016/papers/Redmon\_You\_Only\_Look\_CVPR\_2016\_paper.pdf)

- YOLO makes far fewer background mistakes than Fast R-CNN. By using YOLO to eliminate background detections from Fast R-CNN we get a significant boost in performance.

YOLO는 Fast R-CNN보다 background 실수를 적게 합니다. 따라서 Fast R-CNN에서 배경 탐지를 제거할 때 YOLO를 사용하면 성능이 크게 향상(boost)됩니다.

- For every bounding box that R-CNN predicts we check to see if YOLO predicts a similar box. If it does, we give that prediction a boost based on the probability predicted by YOLO and the overlap between the two boxes.

R-CNN이 예측하는 경계 상자마다, YOLO가 비슷한 상자를 예측하는지 확인합니다. 만약 그렇다면, 우리는 Yolo에 의해 예측된 확률과 두 상자 사이의 중첩을 기반으로 prediction을 boost 시킵니다.

- The best Fast R-CNN model achieves a mAP of 71.8% on the VOC 2007 test set. When combined with YOLO, its mAP increases by 3.2% to 75.0%.

최고의 Fast R-CNN 모델은 VOC 테스트 셋에서 71.8%의 mAP를 달성합니다. YOLO와 결합하면 mAP가 3.2% 증가하여 75.0%가 됩니다.

- The boost from YOLO is not simply a byproduct of model ensembling since there is little benefit from combining different versions of Fast R-CNN. Rather, it is precisely because YOLO makes different kinds of mistakes at test time that it is so effective at boosting Fast R-CNN’s performance.

Fast R-CNN 끼리의 결합은 거의 향상이 없기 때문에, YOLO와 Fast R-CNN의 앙상블은 단순한 것이 아니다. YOLO가 테스트시간에 여러 종류의 실수를 저지르기 때문에 Fast R-CNN의 성능을 향상시키는 데 효과적이다.

- Unfortunately, this combination doesn’t benefit from the speed of YOLO since we run each model seperately and then combine the results. However, since YOLO is so fast it doesn’t add any significant computational time compared to Fast R-CNN.

이 조합은 YOLO의 '속도'라는 이익을 얻지 못한다. 각 모델을 개별적으로 실행한 다음 결과를 결합하기 때문이다. 그러나 YOLO는 매우 빠르기 때문에 Fast R-CNN만을 이용하는 것과 비교할 때 시간을 추가하지 않는다.

3. Fast R-CNN 끼리 Ensemble

(1) 논문 : You Only Look Once : Unified, Real-Time Object Detection(https://www.cv-foundation.org/openaccess/content\_cvpr\_2016/papers/Redmon\_You\_Only\_Look\_CVPR\_2016\_paper.pdf)

- We also tried combining the top Fast R-CNN model with several other versions of Fast R-CNN. Those ensembles produced

small increases in mAP between .3 and .6%, see Table 2 for details.

우리는 또한 최고의 Fast R-CNN 모델을 여러 다른 버전의 Fast R-CNN과 결합하려고 시도했습니다. 이러한 앙상블은 mAP .3에서.6% 사이의 작은 증가만을 가져왔다.

4. Understanding of Object detection based on CNN family and yolo.

1. CNN Family Overview(cnn, r cnn, fast r cnn, faster r cnn)[1]

CNN은 2012 년에 인기를 얻은 후 두 가지 방향으로 발전되었다. 첫번째는 최적화이고, 두번째는 이미지 처리의 프레임 워크를 개선하는 것이다. CNN의 강력한 기능 추출 및 분류 기능은 단일 개체로 이미지를 처리하는 데 효과적이지만 이미지에 다중 개체가 포함되어 있을 때 window selection 문제가 있다. 문제를 해결하기 위해 R-CNN에서는 region proposals, crop/warp, SVM classification and bounding-box regression 같은 새로운 기술을 도입했습니다. R-CNN은 Visual Object Classes (VOC) 2010에서 평균 평균 정밀도 (MAP)가 54 % 인 우수한 객체 탐지 ​​정확도를 얻었다. ((DPM)가 33%인 DPM보다 더 높은. 수치이다.) 그러나 R-CNN의 pre-extraction and multi-stage pipeline은 공간과 시간을 많이 요구한다. Spatial Pyramid Pooling Network (SPPNet)을 이용해 R-CNN을 개선해 테스트 속도를 10 배, 교육 속도를 3 배 가속화했으나, R-CNN과 비슷한 단점이 있다. Fast R-CNN은 ROI를 추가해 단일 스테이지 및 다중 작업 손실과 관련된 문제를 해결했다. 또한, 속도를 높이기 위해 SVD (Singular Value Decomposition) 알고리즘을 사용해 mAP 값은 70을 결과를 얻을 수 있었다. Fast R-CNN은 매우 깊은 네트워크를 사용하여 거의 실시간 속도를 얻었으나, 병목 현상이 발생하는 문제가 있었다. Faster R-CNN은 RPN (Region Proposal Networks)을 도입하여 share convolutional layers with object detection networks and computed proposals with deep net. ResNet [18]을 Fater R-CNN은 mAP 76.4, 프레임 당 초당 프레임 수 (FPS) 5를 얻었다. 이미지 인식의 정확성에 따라 Faster R-CNN의 속도를 처리하는 동안 Faster R- 최첨단 탐지 시스템에서 빠르게 움직일 공간이 있습니다.

1. CNN 소개[1]

CNN [19]은 1998 년 LeNet에서 시작하여 오류율 15.3 % AlexNet 때문에 2012 년에 널리 알려졌다. 그 후, ZF-net [20], GoogLeNet 및 VGGNet은 오류 발생률을 계속해서 낮추면서 발전하였고, 2015년 말까지, ResNet은 인간의 눈 (5.1 %)보다 훨씬 낮은 오류율 3.6 %를 기록했다.

2-1) CNN 장점과 단점

CNN은 입력 데이터의 특징을 표현하고 추출하는 효과적인 수학적 방법을 제공한다. The weight sharing 매커니즘으로 인해 연산량이 줄고 pooling layer는 쉽게 크기를 줄인다. CNN이 데이터 처리의 많은 분야에서 지능형 인식 분야에서 매우 경쟁력 있다. 그러나 더 나은 결과를 얻으려면 매개 변수 및 데이터 집합의 범위는 물론 하드웨어의 컴퓨팅 기능이 가장 중요하다.

2-2) R-CNN

R-CNN에서 처음으로 특징 추출 및 분류 능력이 뛰어난 CNN을 기반으로 neural network propounding the region proposal을 이용한 객체 탐지 방식을 제시했다. The region proposal gets the potential objects by sliding the proposals with different width and height, such as selective search. 그러나 지역 제안은 R-CNN이 거대한 데이터, 시간, 계산 및 에너지를 소비하도록 만든다.

2-3) Fast R-CNN

Fast R-CNN은 SPPNet을 기반으로 제안되었으며, FC 레이어 앞에있는 마지막 풀링 레이어를 SPP로 바꿉니다. 입력 이미지의 해상도에 상관없이 출력 이미지를 m \* n 부분으로 유지한다. 이러한 방법으로 테스트 속도를 24 ~ 102 배 가속시켰다. Fast R-CNN은 (ROI) 풀링 및 proposal reflection을 제공한다. 게다가 multi-task loss layer을 사용하고, SoftmaxLoss는 SVM을 대체하고 SmoothL1Loss는 Bouding-box를 대체했다. Fast R-CNN의 학습 속도와 테스트 속도는 각각 SPP보다 3 배, 10 배 빠릅니다. VOC07 데이터 세트는 Fast R-CNN의 mAP이 70이다

2-4) Faster R-CNN

Fast R-CNN을 기반으로 Faster R-CNN은 RPN 더해 region proposal 문제를 해결했다. ROI 풀링에 입력할 최종 이미지 이미지가 region proposal 가져온다. 피처 이미지의 해상도가 원래 이미지의 해상도보다 낮기 때문에 Faster R-CNN의 계산은 CNN의 모든 이전 모델보다 훨씬 작다. 각 Each sliding proposa는 서로 다른 크기, 너비 및 높이의 후보 앵커를 9 개 생성합니다. Faster R-CNN의 앵커 기능 추출 과정은 Fast R-CNN의 기능과 유사하지만 객체 분류는 기능이 전경인지 배경인지 만 인식하며, 단지 대상 객체의 더 정확한 위치를 찾기 위한 것이다. For each location of the proposal, RPN uses two full-connected layers (object classification and proposal regression) to judge and abandon the anchors. It never does explicit regional proposal. The main rules of selecting the anchors are: 1) 앵커 해제 2) 샘플과의 중복 영역이 0.7보다 크면 전경으로 분류되고, 중첩 영역이 0.3보다 작은 앵커는 배경으로 분류된다.

이렇게 각 sliding proposal는 약 300개의 앵커를 선택한다. 현재 네트워크에서 RPN의 웨이트를 초기화하고, train 데이터 세트에서 올바른 proposal을 추출하고 결과가 잘 수렴 될 때까지 반복적으로 proposal을 사용하여 빠른 R-CNN 모델을 교육합니다.

5. Fast R-CNN

(1) 논문 : Fast R-CNN(https://www.cv-foundation.org/openaccess/content\_iccv\_2015/papers/Girshick\_Fast\_R-CNN\_ICCV\_2015\_paper.pdf)

- Compared to previous work, Fast R-CNN employs several innovations to improve training and testing speed while also

increasing detection accuracy. Fast R-CNN trains the very deep VGG16 network 9× faster than R-CNN, is 213× faster at test-time, and achieves a higher mAP on PASCAL VOC 2012. Compared to SPPnet, Fast R-CNN trains VGG16 3× faster, tests 10× faster, and is more accurate.

Fast R-CNN은 이전 연구와 비교하여 교육 및 테스트 속도를 개선하고 감지 정확도를 높이기 위해 몇 가지 혁신 기술을 채택했습니다. Fast R-CNN은 매우 깊은 VGG16 네트워크를 R-CNN보다 9배 빠르게 전달하고 테스트 시 213배 빠르며 PASCAL VOC 2012에서 높은 mAP를 달성했습니다. SPPnet과 비교하여 Fast R-CNN은 VGG16을 3배 빠르고 테스트를 10배 빠르게 하며, 정확하게 합니다.

- We propose a new training algorithm that fixes the disadvantages of R-CNN and SPPnet, while improving on their speed nd accuracy.

1. Higher detection quality (mAP) than R-CNN, SPPnet

2. Training is single-stage, using a multi-task loss

3. Training can update all network layers

4. No disk storage is required for feature caching

우리는 R-CNN과 SPPnet의 단점을 수정하면서 속도와 정확도를 향상시키는 새로운 교육 알고리즘을 제안합니다.

1. R-CNN과 SPPnet보다 높은 탐지 품질(mAP)

2. 훈련은 단일 단계이며, multi-task loss를 사용합니다.

3. 교육을 통해 모든 네트워크 계층을 업데이트 할 수 있다.

4. feature caching에 디스크 저장 영역이 필요하지 않다.

(2) 논문 : You Only Look Once : Unified, Real-Time Object Detection(https://www.cv-foundation.org/openaccess/content\_cvpr\_2016/papers/Redmon\_You\_Only\_Look\_CVPR\_2016\_paper.pdf)

- Fast R-CNN speeds up the classification stage of R-CNN but it still relies on selective search which can take around 2 seconds per image to generate bounding box proposals. Thus it has high mAP but at 0.5 fps it is still far from realtime.

Fast R-CNN은 R-CNN의 분류 단계를 가속화 합니다. 하지만 여전히 selective search에 의존하기 때문에 이미지 당 약 2초가 걸립니다. 따라서 높은 mAP를 갖지만, 실시간과는 거리가 멉니다.

(3) 논문 : Faster R-CNN : Towards Real-Time Object Detection with Region Proposal Networks(https://arxiv.org/pdf/1506.01497.pdf)

- The latest incarnation, Fast R-CNN [2], achieves near real-time rates using very deep networks [3], when ignoring the time spent on region proposals. Selective Search, one of the most popular methods, greedily merges superpixels based on engineered low-level features. Yet when compared to efficient detection networks [2], Selective Search is an order of magnitude slower, at 2 seconds per image in a CPU implementation.

Fast R-CNN은 매우 깊은 네트워크를 사용하여 거의 실시간 속도를 달성합니다. Selective Search는 low-level features를ㅠ 기반으로 superpixel을 병합한다. 그러나 다른 효율적인 네트워크와 비교할 때, Selective Search는 CPU 구현시 이미지 당 2초의 속도가 느려집니다.

6. Faster R-CNN

(1) 논문 : Mask R-CNN(https://arxiv.org/pdf/1703.06870.pdf)

- Faster RCNN was not designed for pixel-to-pixel alignment between network inputs and outputs. This is most evident n how RoIPool, the de facto core operation for attending to instances, performs coarse spatial quantization for feature extraction. To fix the misalignment, we propose a simple, quantization-free layer, called RoIAlign, that faithfully preserves exact spatial locations.

Faster RCNN은 네트워크 입력과 출력에서 픽셀간의 alignment를 위한 설계가 되어 있지 않다. 이것은 RoI Pool의 방법에서, 피쳐를 추출할 때의 문제입니다. 이러한 오류를 해결하기 위해 정확한 공간 위치를 유지하는 RoI Align이라는 방법이 필요합니다.

(2) https://medium.com/@umerfarooq\_26378/from-r-cnn-to-mask-r-cnn-d6367b196cfd

- When run without modifications on the original Faster R-CNN architecture, the regions of the feature map selected by RoIPool were slightly misaligned from the regions of the original image. Since image segmentation requires pixel level specificity, unlike bounding boxes, this naturally led to inaccuracies.

기존의 Faster R-CNN 아키텍쳐를 수정하지 않고 실행하면 RoI Pool에서 선택한 피쳐 맵의 영역이 원본 이미지의 영역과 약간 어긋나게 됩니다. 이미지 분할은 경계 상자와 달리 픽셀 수준의 특수성을 필요로 하기 때문에 부정확한 결과를 가져 왔습니다.

- This problem by cleverly adjusting RoIPool to be more precisely aligned using RoIAlign. In RoIPool, we would round this down and select 2 pixels causing a slight misalignment. However, in RoIAlign, we avoid such rounding. Instead, we use bilinear interpolation to get a precise idea of what would be at pixel 2.93. This, at a high level, is what allows us to avoid the misalignments caused by RoIPool.

RoI Pool을 조정하여 Mask R-CNN에서 RoI Align을 사용하여 더 정확하게 정렬 되도록 하여 문제를 해결할 수 있다. RoI Pool에서는 반올림을 하기 때문에 어긋남을 일으켰다. 그러나 RoI Align에서는 이러한 반올림을 피합니다. 대신 쌍 선형 보간법을 사용 하여 정확함을 어습니다. 이것은 RoI Pool의 불균형함을 피할 수 있게 해줍니다.

(3) 논문 : Is Faster R-CNN Doing Well for Pedestrian Detection? (https://link.springer.com/chapter/10.1007/978-3-319-46475-6\_28)

- Faster R-CNN [11] is a particularly successful method for general object detection. It consists of two components: a fully convolutional Region Proposal Network (RPN) for proposing candidate regions, followed by a downstream Fast R-CNN [12] classifier.

Faster R-CNN은 특히 general한 object의 detection에 효과적인 방법이다. 그것은 2개의 구성 요소로 구성 된다. 후보 지역을 제안하기 위한 RPN과, Fast-RCNN의 Classifier이다.

- Its leading accuracy on several multi-category benchmarks

Faster R-CNN은 여러 카테고리의 벤치 마크에서 최고의 정확성을 가지고 있다.

7. Fast/Faster R-CNN의 문제점

(1) 논문 : Is Faster R-CNN Doing Well for Pedestrian Detection? (https://link.springer.com/chapter/10.1007/978-3-319-46475-6\_28)

- The convolutional feature maps of the Fast R-CNN classifier are of low solution for detecting small objects. the Region-of-Interest (RoI) pooling layer [12, 15] performed on a low-resolution feature map (usually with a stride of 16 pixels) can lead to "plain" features caused by collapsing bins. Fast R-CNN의 classifierdml feature map은 작은 object를 detecting 하기에 좋지 않다. 또한 RoI Pool이 저해상도의 피쳐 맵에서 수행되면 'plain' feature로 분류 되는, 붕괴가 일어날 수 있다.

8. retinanet 장점

- yolo와 ssd같은 기존의 one-stage detecter들은 빠르고 간단하지만 accuracy가 2-stage 보다 10~40% 정도 낮고 classifier가 dense, regular location에 적용되는 단점이 있다.

accuracy가 낮은 것은 easy negative(background)가 hard positive(object)보다 압도적으로 많이 검출되면서 발생하는 극단적인 class imbalance에서 기인하는 문제이다.

retinanet에서는 이 문제를 해결하기 위해 새로운 loss함수인 focal loss를 사용하고, 대부분의 negative 샘플들에 의해서 학습이 overwhelming 되는 것을 방지한다.

- corss entropy loss를 변형한 focal loss를 사용함으로써 hard negative example을 더 잘 분류할 수 있음. focal loss는 감마의 크기가 커짐에 따라 well-classified example에 상대적으로 낮은 loss를 부여하고 hard, misclassified examples에 큰 가중치를 부여한다. focal loss를 사용하면 one-stage network인 retinanet에서 2-stage net이상의 성능을 얻을 수 있다.

참고1) 실전에서는 weighting factor(알파)-balanced variant of the FL사용: FL(p)=-알파(1-p)^r\*log(p)

알파는 rare class에 할당되는 weight이며, 알파와 감마는 서로 영향을 주고받으므로 두 개를 같이 선택해야한다. 보통 알파와 감마는 반비례한다.

참고2) 최적 focal loss : 감마(r)=2, 알파=0.25 일 때 가장 ap가 높다.

-fpn backbone을 사용하기 때문에 이미지 해상도가 높고 multi-scale 예측을 효율적으로 할 수 있다. 각각의 피라미드 레벨은 서로 다른 크기의 객체를 검출한다. fpn없이 resnet의 최종 레이어만 사용해 예측하는 경우 성능이 좋지 않았다.

- yolov2, ssd, dssd, faster r-cnn보다 ap가 높다. (focal loss for dense object detection논문 8페이지에 retinanet과 비교한 다른 모델들의 backbone 및 ap 표(table 2) 참고 바람)

retinanet은 1-stage detector만큼 빠르면서도 기존 모든 최고 성능의 detector들을 능가하는 accuracy를 보인다.

- loss function만 변경하고 기존의 backbone network를 적절하게 선택하였기 때문에 구조가 단순할뿐만 아니라 효율적임.

- 마지막 layer를 제외하고 bias는 전부 0으로 설정한다.

마지막 layer의 bias=-log((1-파이)/파이)인데, 이 때 파이는 너무 많은 background anchor가 생기는 것을 방지하여 첫 iteration에서 학습을 안정화시킨다.

- 각 image에 대해 최대 100k 개의 anchor box에 FL이 적용된다. 이는 적은 anchors(e.g. 256)를 사용하는 rpn이나 ohem, ssd 와 대조적이며, retinanet의 정확성을 높인다.

<Ensemble ; Voting Ensemble>

9. 논문(ieee) : An Ensemble Method of CNN Models for Object Detection(https://ieeexplore.ieee.org/abstract/document/8539396/authors#authors)

(1) Abstract

- Traditional methods of object detection were based on pre-structured features and had limitation on accuracy and computational efficiency. As deep learning had been proved to be a breakthrough, researches about object detection method based on deep learning, especially CNN, started. CNN-based object detection methods can be divided into two types. One is two-stage detector that once region proposals are generated, then they are classified. The other is one-stage detector that detects and classifies the object without generating region proposals. In two-stage detector case, combining CNN models is one of the ways to improve the accuracy in detection, which is called ensemble.

기존의 객체 탐지 방법은 사전 구조화된 기능 기반이며, 정확도와 계산 효율성에 한계가 있었다. deep learning이 획기적인 것으로 판명되면서, 특히 CNN과 같은 deep learning에 대한 연구가 시작되었다. CNN 기반 객체 탐지 방법 유형은 두 가지 유형으로 나눌 수 있다. 하나는 region proposal이 있는 2 stage detector 이다. 다른 하나는 region proposal을 생성하지 않고 객체를 탐지하고 분류하는 1 stage detector이다. 2 stage detector의 경우 CNN 모델을 결합하는 것이 앙상블이라고 하는 검출 정확도를 향상시키는 좋은 방법이다.

- In ensemble method, the region proposals generated from each CNN models are combined, classified, and finally voted. When selecting CNN models to be used in ensemble method, various properties of them should be considered in order to enhance complementary strength.

앙상블 방법은, 각 CNN 모델에서 생성 된 Region Proposal이 결합되어 분류되고, vote 되는 것이다. 앙상블 방법으로 사용될 CNN 모델을 선택할 때 보완적인 강도를 높이기 위해서는 다양한 특성이 고려되어야 한다.

(2) Introduction

- This two-stage detector shows high accuracy in detection but requires a lot of time cost because of the algorithm that generates region proposals. One-stage detector such as YOLO [9] and SSD [10] is designed to relieve the time cost, but shows relatively lower accuracy in detection.

2 stage detector는 탐지에서 높은 정확도를 보여주지만, Region Proposal에서 많은 시간이 소요된다. YOLO나 SSD와 같은 1 stage detectopr는 시간 소요를 적게 하지만, 상대적으로 검출 정확도가 낮다.

- The accuracy of two-stage detector can be further increased by combining CNN models, which is called "ensemble". In ensemble method, the region proposals generated from the final feature map are shared between CNN models. Then, each CNN model classifies the shared region proposals. The results of classification from each model are unified and voted at last. The sharing and voting process in ensemble method requires additional time cost, so it is usually considered in time-free application.

2 stage detector의 정확도는 CNN 모델들을 결합시켜, ensemble이라고 하는 방법을 이용해서 증가시킬 수 있다. 앙상블의 방법으로 최종 feature map에서 추출 된 region proposal이 CNN 모델들 간에 공유된다. 그런 다음 각 CNN 모델들은 region proposals를 공유한다. 각 모델의 분류 결과가 통합되어 최종적으로 투표된다. 앙상블 방법 중의 share과 vote의 과정은 추가적인 시간이 필요한 과정이므로, time-free 응용 프로그램에서 주로 사용된다.

- In this paper, we propose advanced ensemble method in object detection, especially focusing on Faster R-CNN [11], with novel model selecting and box voting methods. With proposed model selecting method, the CNN models are selected based on not only overall mean accuracy precision (mAP) but also AP according to the class and the size of objects so that the complementary strength between each model can be leveraged. And with our proposed box voting method, the classes predicted by multiple classifiers are voted, based on their per-class AP, which makes the voting process more reasonable.

본 논문에서는 object detection, 특히 Faster R-CNN에 초점을 둔 방법을 제안한다. Model을 선택하는 방법에는 mAP 뿐만 아니라 AP이용할 수 있다. class와 객체의 크기에 따라 CNN 모델을 선택하여 각 모델 간의 보완적인 강도를 활용할 수 있다. Proposed box voting 방법을 사용하면 여러 classifiers가 클래스 별 AP에 따라 voted 되어기 때문에 합당하다고 볼 수 있다.

(3) Section 2

1) CNN-based object detection method

- The object detection methods mentioned above are regarded as two-stage detector in respect that the detection process is composed of region proposal generating part and classification part. These methods are high in accuracy but relatively low in speed to be applied to real-time application. This led the research in object detection method to one-stage detector. YOLO (You Only Look Once) [9] method is one of the one-stage detection methods, which considers the whole detection process as single regression problem. Instead of generating region proposals, the input image is divided into a number of grids, and a specific number of bounding boxes are assigned to each grid. With YOLO method, 45 images can be processed in a second, which implies that the object detection with YOLO can be applied to real-time system. However, the accuracy in detection was 10% lower than Faster R-CNN method. SSD (Single Shot multibox Detector) [10] method, another one-stage detector, exploits multiple feature maps with different sizes which were used in CNN computation, unlike other detection methods that exploit only the final feature map. The feature maps in early stage are large in size, which means they are used to detect small size of objects. In contrast, the feature maps in late stage are small in size, which means they are used to detect large size of objects. As getting rid of region proposal generating process, SSD method showed higher performance than Faster R-CNN in respect of speed and even in accuracy on some occasions.

앞에서 설명된 object detection 방법들(Fast, Faster)은 Region Proposal의 단계와 classification의 단계로 나누어지기 때문에 2 stage detector로 간주된다. 이 방법은 정확도가 높지만 실시간 응용 프로그램에 적용하기에는 상대적으로 속도가 느리다. 이것은 1단계 검출기에 대한 연구를 이끌었다. YOLO 방법은 전체 탐지 프로세스를 단일 회귀로 간주하는 1단계 탐지 방법의 하나이다. Region Proposal을 생성하는 대신에, 입력 이미지를 다수의 그리드로 분할하고 특정 개수의 bounding box가 각 그리드에 할당되게 한다. YOLO를 이용하면, 1초에 45개의 이미지를 처리할 수 있기 때문에 YOLO를 이용한 물체 감지를 실시간 시스템에 적용할 수 있다. 그러나 탐지의 정확도는 Faster R-CNN 방법보다 10% 더 낮다. SSD(Single Shot Multibox Detector) 방법 또한 1단계 탐지기로, 최종 feature map만 이용하는 다른 탐지 방법들과 달리, CNN 계산에 사용된 다른 여러개의 feature map들을 이용한다. 초기 단계의 feature map은 크기가 커서 작은 크기의 물체를 detect 하는데 이용된다. 이와 대조적으로 후기 단계의 feature map은 크기가 작으므로 크기가 큰 object를 감지하는 데 사용된다. Region Proposal을 제한하는 방법을 제거하면서, SSD는 속도 면에서나 정확도 측면에서 조차도 Faster R-CNN보다 높은 성능을 보였다.

2) CNN for object detection

- CNN architectures are developed in the way of increasing the depth of layers. However, deep neural network can lead to problems such as over-fitting, vanishing gradient, or degradation of accuracy with some reasons. K He et al. [6] developed a CNN architecture, named ResNet, adopting the concept of residual learning. Residual learning is a learning method that a set of layers learns not the output but the difference between the input and the output, which enables the model to learn subtle change sensitively. ResNet is a CNN architecture that has same structure with VGG Net but where residual learning is applied. With residual learning, the accuracy was consistently increased as the network gets deeper to 152 layers.

CNN은 레이어의 깊이를 늘리는 방식으로 개발되었다. 그러나 deep 신경 네트워크는 과도한 오버피팅, vanishing gradient 혹은 정확성 저하와 같은 문제를 유발할 수 있다. ResNet은 CNN 아키텍처를 개발하여 만들었으며, '잔여 학습'이라는 개념으로 만들어 졌다. 잔여 학습은 일련의 레이어가 출력이 아니라 입력과 출력의 차이를 학습하여 모델이 미묘한 변화를 민감하게 학습할 수 있게 하는 학습 방법이다. ResNet은 잔여 학습이 적용되는 VGG Net과 동일한 구조를 가진 CNN 아키텍처이다. 잔여 학습을 통해 정확도는 네트워크가 152 계층으로 깊어지면서 지속적으로 증가했다.

(4) Section 3

1) Ensemble of CNN Models

- One of the ways to enhance the accuracy in object detection is to combine CNN models, which is called "ensemble". J Huang et al. [16] proposed an ensemble method of Faster R-CNN, and this achieved the state of the art performance on the 2016 COCO object detection challenge. In order to construct ensemble detector, first of all, CNN models to be used must be selected. They proposed model selecting method based on category-wise AP (Average Precision) vectors in order to acquire complementary advantages of each model. The diversity between models is computed as cosine distance between the category-wise AP vectors. If cosine distance between two models is higher than some threshold, the model with higher mAP is selected, and the other is discarded. The region proposals generated from each selected model are combined into a set of region proposals, shown as in Fig. 1. The set is shared between each model so that each model classifies it and calibrates the box location. The results of box location from each model are voted in the way of assigning more weight to the result with high confidence. The coordinate of a voted box Loc(x,y,w,h) is calculated as:

object detection의 정확성을 높이는 방법 중 하나는 '앙상블'이라고 하는 CNN 모델을 결합하는 것이다. J Huang et al은 Faster R-CNN의 앙상블 방법을 제안하였고, 이것은 2016년 COCO object detection 대회에서 최고의 성능을 달성했다. 앙상블 detector를 구성하기 위해서는 우선 CNN 모델들을 선택해야 한다. 그들은 그 모델들의 각 보완적인 이점을 얻기 위해 category 별 AP 벡터를 기반으로 모델 선택 방법을 제안했다. 모델들 사이의 diversity는 category-wise AP 벡터들 사이의 코사인 거리로 계산된다. 두 모델 사이의 코사인 거리가 threshold 값보다 높으면 더 높은 mAP를 갖는 모델이 선택되고 다른 모델은 무시된다. 각 선택된 모델에서 생성된 region proposals는 set of region proposals로 결합되어 각 모델이 리를 분류하고 상자 위치를 교정 하도록 각 모델간에 공유된다. 각 모델의 상자 위치 결과는 신뢰도가 높은 결과에 더 많은 가중치를 할당하는 방식으로 투표된다.

- We propose novel model selecting and class voting methods in order to further increase the accuracy in detection of ensemble method. In model selecting, we consider AP according to not only the class but also the size of object. Some models may show high performance in detecting large size of objects, while others in detecting small size of objects. The complementary strength between them can be acquired, avoiding biased selection of models. The diversity between the models is computed as cosine distance between the object size-wise AP vectors. And in box voting, the confidence of class is voted by assigning the per-class AP as weight.

우리는 앙상블의 검출 정확도를 높이기 위해 모델 선택과 클래스 투표 방법을 정하는 것을 추천한다. 모델 선택시, AP를 class 뿐만 아니라 객체의 크기 또한 고려했다. 어떤 모델은 큰 객체에 높은 성능을 보이고, 어떤 모델은 작은 객체에 높은 성능을 보인다. 편향된 모델 선택을 피해서 이들간의 보완적인 강도를 얻을 수 있다. box voting에서, class별 confidence는 클래스 별 AP를 weight로 설정하여 vote한다.

(5) Conclusion

- The ensemble method in object detection is employed in the situation that accuracy is more important than time cost. However, it requires a lot of additional time cost for the reason that it is consisted of several feature extractors in parallel.

object detection에서의 앙상블은 정확도가 시간보다 더 중요한 상황에서 사용된다. 앙상블을 위한 여러 기능들이 있기 때문에 추가적인 시간 비용이 들기 때문이다.

10. Ensemble (<https://www.datasciencecentral.com/profiles/blogs/want-to-win-at-kaggle-pay-attention-to-your-ensembles>)

(1) Introduction

- Kaggle 대회에서 앙상블은 Kaggle에서 승리하기 위한 핵심 전략이었다. 앙상블에는 3가지 그룹이 있고, 각각 고유한 강점과 약점을 가지고 있다. 앙상블 알고리즘에는 3가지 클래스가 있다.

1) Bagging (Random Forest가 여기 포함됨)

2) Boosting

3) Stacking

앙상블의 배경은 간단하다. 여러 모델을 사용하고 결과를 결합하면 일반적으로 모델의 성능이 향상되거나 불량 모델을 선택할 확률이 줄어든다. 앙상블이 단일 모델보다 안 좋을 수 있지만, 전반적인 overall risk를 감소시킨다. 위의 세 가지 주요 그룹으로 대표되는 앙상블에 단일 predictor를 결합하거나, 전략적으로 그룹 내에서의 구현 간의 세부 변형이 많이 있을 수 있다. 우리가 경험한 대부분의 가장 정확한 모델은 앙상블이었다!

- 앙상블이 효과적이기 위해서는 다양한 모델들을 선택해야 한다. 개별 모델이 모두 동일한 오류를 생성하지 않기 위해서이다. 이것은 앙상블의 다양성에 대한 요구 사항이며, 다양한 방법으로 다양성을 이룰 수 있다. 일반적인 방법으로는, 각 classifer에 각각 다른 training data를 이용하는 것이다. bootstrap이나 jackknife resampling을 자주 이용한다. 또한 각 다른 유형의 모델링 알고리즘을 이용하는 방법도 있다.

- 앙상블은 ‘selected’나 ‘fused’로 카테고리 될 수 있다. ‘selected’ 전략은, 특정 데이터 영역에서 가장 정확하게 묘사하는 모델이 해당 영역에 대한 단일 예측자로 선택되는 것이다. 데이터의 다른 특정 영역을 가장 정확하게 묘사하는 다른 성공적인 모델과 결합하게 된다. ‘fused’ 전략은, 개별 모델의 결과를 평균, 합계, min, max를 이용한 majority voting 으로 합쳐진다.

(2) Bagging

- Popular Tool : Random Forest

- Concept : Bagging은 1996년 Leo Brieman에 의해 만들어진 최초의 앙상블 기술이다. 데이터의 bootstrapped 하위 집합에 대한 strong learners를 선택하고, 간단한 voting으로 그 결과를 산출한다.

- 장점 :

분산을 줄이고 일반적으로 오버 피팅을 방지한다.

실행 시간이 빠르다.

Default parameters에서 쉽게 사용할 수 있다. 약간의 추가 튜닝만 해야 잘 사용할 수 있다.

Tree 기반이므로 missing value, 변수 자동 선택 및 높은 non-linear의 상호 작용을 수용한다.

- 단점 :

복잡성이 증가하면 스코어가 떨어질 수 있다.

여러 tree의 복잡성으로, transparance가 떨어질 수 있다.

\* Bagging은 모델의 bias가 높을 때 추천되지 않고, Bossting은 모델의 bias가 높을 때 추천된다.

(3) Boosting

- Popular Tools : Adaboost, Gradient Boosted Models(GBM)

- Concept : Bagging과 비슷하지만, weak learners를 개선하는데 중점을 둔다는 점이 다르다. Boosting의 각 iteration마다 3개의 weak classifers가 생성된다. 첫 번째 classifier는 subset data 상에서 정상적으로 트레이닝 된다. 두 번째 classifier는 첫 번째 classifier가 50%만 식별해 낸 데이터에 대해 트레이닝 한다. 세 번째 classifier는 첫 번째 classifier와 두 번째 classifier의 분류가 일치하지 않는 데이터에 대해 트레이닝 한다. 세 가지 classifier는 majority vote에 의해 합산된다.

- Strengths :

종종 가장 좋은 model이다.

Cost function을 직접 optimize한다.

- Weaknesses :

Overfit 될 수 있다.

적절한 stopping point를 찾아야 한다.

하이퍼 매개 변수가 꽤 있다.

여러 tree의 복잡성으로 transparancy가 부족하다.

(4) 결론

- 우리는 지금까지 다양한 앙상블 기법을 배웠다. 하지만 Kaggle에서 우승 하는데 정말로 집중한다면 XGBoost, Extreme Gradient Boosted Trees 등과 같이 새롭게 등장한 알고리즘에 주의를 기울여야 할 것이다.

11. mAP

(1) 논문 : Temporal Convolutional Networks for Action Segmentation and Detection(<http://openaccess.thecvf.com/content_cvpr_2017/papers/Lea_Temporal_Convolutional_Networks_CVPR_2017_paper.pdf>)

- Detection metrics: Action detection papers tend to use segment-wise metrics such as mean Average Precision with midpoint hit criterion (mAP@mid) [26, 28] or mAP with a intersection over union (IoU) overlap criterion (mAP@k) [25]. mAP@k is computed my comparing the overlap score for each segment with respect to the ground truth action of the same class. If an IoU score is above a threshold of k percent it is considered a true positive otherwise it is a false positive. Average precision is computed for each class and the results are averaged.

탐지 측정 항목 : Action detection 논문들은 mAP@mid을 사용하거나 mAP@k를 사용한다. mAP@k는 maP과 IoU의 오버랩 기준이 있는 segment metric이다. mAP@k는 동일한 class의 ground truth와 관련하여 각 segment의 overlap score를 비교하여 계산된다. IoU 점수가 k 퍼센트의 임계값 이상이면 true positive로, 그렇지 않으면 false positive로 간주된다. 각 클래스에 대한 평균 정밀도가 계산되고 결과가 평균된다.

- mAP scores are computed using a ranked list of action predictions, which are ordered based on confidence scores assigned to each segment. These confidences are often simply the mean or maximum class score within the frames corresponding to a predicted segment. We find that by computing these confidences in subtly different ways you obtain wildly different results.

mAP 점수는 각 segment에 할당된 confidence score에 따라 예측 순위를 rank하여 계산된다. 이러한 confidence는 단순히 예측된 segment에 해당하는 프레임 내의 평균 혹은 최대 class 점수이다. 이러한 confidence를 미묘하게 다른 방식으로 계산하면 매우 다른 결과를 얻을 수 있다.

(2) 논문 : Temporal Action Detection with Structured Segment Networks(<http://openaccess.thecvf.com/content_ICCV_2017/papers/Zhao_Temporal_Action_Detection_ICCV_2017_paper.pdf>)

- Evaluation Metrics. As both datasets originate from contests, each dataset has its own convention of reporting performance metrics. We follow their conventions, reporting mean average precision (mAP) at different IoU thresholds.

평가 척도. 두 데이터 세트는 대회에서 비롯된 것이기 때문에 각 데이터 세트의 성능 메트릭 보고의 자체 규칙이 있다. 우리는 서로 다른 IoU 임계 값에서 mAP를 구하는 규칙을 따른다.

(3) <http://tarangshah.com/blog/2018-01-27/what-is-map-understanding-the-statistic-of-choice-for-comparing-object-detection-models/>

1) IoU

- We need to know to judge the correctness of each of these detections. The metric that tells us the correctness of a given bounding box is the - IoU - Intersection over Union.

우리는 이러한 각각의 탐지의 정확성을 판단하기 위해 알아야 한다. 지정된 경계 상자의 정확도를 나타내는 척도는 IoU이다.

* Intersection over Union is a ratio between the intersection and the union of the predicted boxes and the ground truth boxes. To get the intersection and union values, we first overlay the prediction boxes over the ground truth boxes. Now for each class, the area overlapping the prediction box and ground truth box is the intersection area and the total area spanned is the union.

IoU는 Predicted boxes와 Ground truth Boxes의 교차 영역과 합집합의 비율이다. 각 클래스에 대해 Predicted boxes와 Ground truth boxes가 겹치는 영역이 교차 영역이며, spanned 된 전체 영역은 합집합이다.

* For calculating Precision and Recall, we have to identify True Positives, False Positives, True Negatives and False Negatives. To get True Positives and False Positives, we use IoU. If the IoU > threshold, it is considered a True Positive, else it is considered a false positive. For calculating Recall, we need the count of Negatives. We only measure “False” Negatives ie. the objects that our model has missed out.

Precision = TP/(TP+FP)

Recall = TP/(TP+FN)

Precision과 Recall 계산에 있어서 TP, FP, TN, FN을 식별해야 한다. TP와 FP를 구하기 위해서 우리는 IoU를 사용한다. IoU>threshold 이면 True Positive로 간주되고, 그렇지 않으면 False Positive로 간주된다. Recall을 계산하기 위해서 우리는 False Negatives를 이용한다.

* We have atleast 2 other variables which determine the values of Precision and Recall, they are the IOU and the Confidence thresholds.

Precision과 Recall의 값을 결정하는 두 가지 변수가 있다. IoU와 Confidence threshold 값이다.

* The AP summarises the shape of the precision/recall curve, and is defined as the mean precision at a set equally spaced recall levels. The mAP hence is the Mean of all the Average Precision values across all your classes as measured above.

AP는 precision/recall 곡선의 모양을 요약하고, 일정한 간격으로 설정된 recall에서의 평균 precision으로 정의된다. mAP는 위에서 정의한 모든 클래스에서의 평균 정밀도의 평균이 된다.

12. Ensemble (http://www.scholarpedia.org/article/Ensemble\_learning#Voting\_based\_methods)

(1) Introduction

- An ensemble-based system is obtained by combining diverse models (henceforth classifiers).

앙상블 기반 시스템은 다양한 모델을 결합하여 얻을 수 있습니다.

(2) Model Selection

- The most commonly used procedure 'choosing the classifiers with the smallest error on training data' is unfortunately a flawed one. Performance on a training dataset can be misleading in terms of the classification performance on the previously unseen data.

가장 일반적으로 사용되는 ‘훈련 데이터에 대한 가장 작은 오류를 가지는 분류기 선택’ 절차도 불행히도 결함이 있습니다. 교육 데이터를 performance하는 데 있어서, 이전에 볼 수 없던 데이터의 분류 성능 측면에서 misleading이 있을 수 있기 때문이다.

* Of all classifiers that may all have the same training which one should be chosen? Everything else being equal, one may be tempted to choose at random, but with that decision comes the risk of choosing a particularly poor model. Using an ensemble of such models and combining their outputs by for example, simply averaging them can reduce the risk of an unfortunate selection of a particularly poorly performing classifier.

같은 training을 하는 전체의 classifiers 중에서 어떤 것을 선택해야 하는가? 다른 모든 것들 것 평등하다면, 무작위로 선택하고 싶은 유혹을 받을 수 있다. 하지만 그러한 결정을 내리면 특히 빈약한 모델을 선택할 위험이 있습니다. 이러한 모델의 앙상블을 사용하고 예를 들어 출력을 결합하면 간단히 평균화하면 특히 성능이 낮은 분류자를 불행하게 선택할 위험을 줄일 수 있다.

* It is important to emphasize that there is no guarantee that the combination of multiple classifiers will always perform better than the best individual classifier in the ensemble. Nor an improvement on the ensemble’s average performance can be guaranteed except for certain special cases (Fumera 2005). Hence combining classifiers may not necessarily beat the performance of the best classifier in the ensemble, but it certainly reduces the overall risk of making a particularly poor selection.

복수 분류 자의 결합이 앙상블에서 가장 우수한 개별 분류 자보다 항상 우수할 것이라는 보장은 없다! 특정 특수한 경우를 제외하고는 앙상블의 평균 성능에 대한 개선도 보장할 수 없다! 따라서 분류자를 결합하는 것이 앙상블에서 최상의 분류자의 성능을 반드시 넘을 수는 없지만, 특히 빈약한 선택을 하는 위험을 확실히 감소시킬 수는 있다.

* In order for this process to be effective, the individual experts must exhibit some level of diversity among themselves, as described later in this article in more detail. Within the classification context, then, the diversity in the classifiers (typically achieved by using different training parameters for each classifier) allows individual classifiers to generate different decision boundaries. If proper diversity is achieved, a different error is made by each classifier, strategic combination of which can then reduce the total error.

이 프로세스가 효과적이기 위해서는 개별 모델이 뒤에서 설명하는 것처럼 어느 정도 수준의 다양성을 보여야 한다. 분류자의 다양성은 개별 분류자가 상이한 decision boundaries를 생성하게 한다. 적절한 다양성이 달성 되면 각 분류 기준에 따라 다른 오류가 발생하여 전략적 조합을 통해 총 오류를 줄일 수 있다.

(3) Other Reasons for Using Ensemble System

- In his 2000 review article, Dietterich lists three primary reasons for using an ensemble based system:

i) statistical;

ii) computational; and

iii) representational (Dietterich 2000).

Note that these reasons are similar to those listed above. The statistical reason is related to lack of adequate data to properly represent the data distribution; the computational reason is the model selection problem, where among many models that can solve a given problem, which one we should choose. Finally, the representational reason is to address to cases when the chosen model cannot properly represent the sought decision boundary, which is discussed under divide and conquer section above.

2000년 리뷰 기사에서 Ditterich는 앙상블 기반 시스템을 이용하는 세 가지 주요 이유를 나열한다.

1. Statistical
2. Computational
3. Representational

통계적 이유는 적절한 데이터가 부족한 경우에, 데이터 분포를 적절히 나타내기 위한 것과 관련되어 있다. 계산상의 이유는 주어진 문제를 해결할 수 있는 모델 중에서 우리가 우리가 선택해야 하는 모델 선택의 문제이다. 마지막으로 Representational 이유는 선택된 모형이 decision boundary를 적절히 찾을 수 없는 경우이다.

(4) Diversity

- The success of an ensemble system (that is, its ability to correct the errors of some of its members) rests squarely on the diversity of the classifiers that make up the ensemble. After all, if all classifiers provided the same output, correcting a possible mistake would not be possible. Therefore, individual classifiers in an ensemble system need to make different errors on different instances. The intuition, then, is that if each classifier makes different errors, then a strategic combination of these classifiers can reduce the total error, a concept not too dissimilar to low pass filtering of the noise. Specifically, an ensemble system needs classifiers whose decision boundaries are adequately different from those of others. Such a set of classifiers is said to be diverse. Classifier diversity can be achieved in several ways. Preferably, the classifier outputs should be class-conditionally independent, or better yet negatively correlated.

앙상블 시스템의 성공(members 중 일부의 실수를 바로 잡을 수 있는 능력)은 앙상블을 구성하는 분류자의 다양성에 의존한다. 모든 분류기가 동일한 출력을 제공하면 결국 가능한 실수를 수정할 수 없게 된다. 따라서 앙상블 시스템의 개별 분류기는 서로 다른 인스턴스에서 다른 오류를 만들어야 한다. 따라서 각 분류기가 서로 다른 오류를 내면 이러한 분류기를 전략적으로 결합하면 전체 오류가 줄어들 수 있으며 이는 low pass filtering of the noise와 다르지 않은 개념이다. 특히 앙상블 시스템은 결정 경계가 다른 것들과 적절하게 다른 분류자를 필요로 한다. 이러한 분류기 집합은 다양하다. 분류자의 다양성은 여러 가지 방법으로 얻을 수 있다. 바람직하게는, 분류자의 아웃풋은 독립적이거나, ‘음의 상관관계’를 가져야 한다.

- Another approach to achieve diversity is to use different training parameters for different classifiers. For example, a series of multilayer perceptron (MLP) neural networks can be trained by using different weight initializations, number of layers / nodes, error goals, etc. Adjusting such parameters allows one to control the instability of the individual classifiers, and hence contribute to their diversity.

다양성을 달성하기 위한 다른 접근법은 상이한 classifiers에 대해 상이한 트레이닝 파라미터를 사용하는 것이다. 예를 들어, 일련의 MLP 뉴럴 네트워크가 있다면, 이 일련의 MLP들에 각각 다른 가중치, 노드수 등을 설정해서 이용할 수 있다. 이러한 파라미터를 조정하면 개별 분류기의 불안정성을 제어할 수 있고, 다양성을 개선시킬 수 있다.

13. One-stage vs Two-stage detection

(1) 논문 : <https://arxiv.org/pdf/1711.07264.pdf>

- Recent CNN-based object detectors can be categorized into single-stage detectors [26, 27, 22, 20, 4] and twostage [5, 28, 19, 7] detectors. The single-stage detector usually targets on a sweet-spot of very fast speed and reasonably good accuracy. The two-stage detector divides the task into two steps: the first step (body) generates many proposals, and the second step (head) focuses on the recognition of the proposals. Usually, in order to achieve the best accuracy, the design of the head is heavy. The two-stage detector often has a sweet-spot of (relatively) slow speed and very high accuracy.

(2) 논문 : <https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0203355>

There are two main object detector types: two-stage object detector and one-stage object detector. The more widely used type is the two-stage object detector. The Region-based Convolutional Neural Network (R-CNN) [26] is a representative two-stage object detector and it drastically improved detection performance. Modifications were made to its network structure to develop the subsequent models Fast R-CNN [27] and Faster R-CNN [28]. The Faster R-CNN model was employed in the study of Akselrod-Ballin et al. [23, 23] and Ribli et al. [24]. However, the main shortcoming of the two-stage object detector is that its complex network architecture makes training and inference less efficient.

The other detector type is the one-stage object detector. One-stage object detectors whose architecture is simpler than that of two-stage object detectors were introduced as an alternative. OverFeat [29], SSD [30], and YOLO [31], all of which are one-stage detectors, have attracted attention due to their fast processing, but they are limited in accuracy. However, RetinaNet [32], a recently proposed one-stage object detector, achieves high performance using Focal Loss function which addresses the drawback of the traditional cross-entropy loss function, while keeping the processing efficient, which is the main advantage of one-stage object detectors. For the mass detection task, we propose a model based on RetinaNet which is a robust region-based deep learning object detector