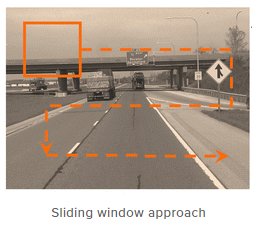
19/10/18 최초작성

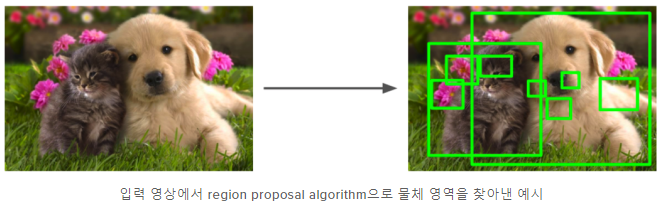
Object detecting

* Sliding window - 가장 간단한 Object detection 방법



image가 있다면 사각형 틀을 이동시키며 모든 영역에 대해 검출하는 방법

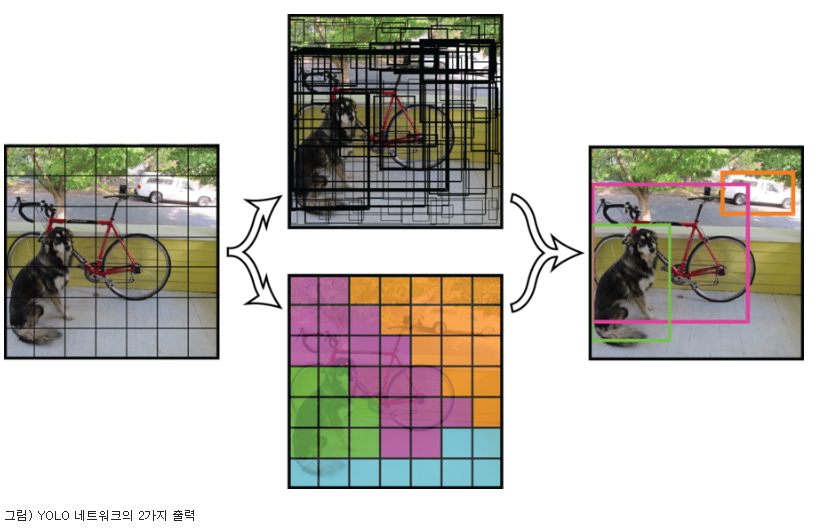
* 연산이 많아 비효율적.
* Region Proposal – 물체가 있을법한 영역을 찾아주는 방법



여러 알고리즘들이 있으며 selective search, Edge boxes 등 존재.

YOLO (You Only Look Once)

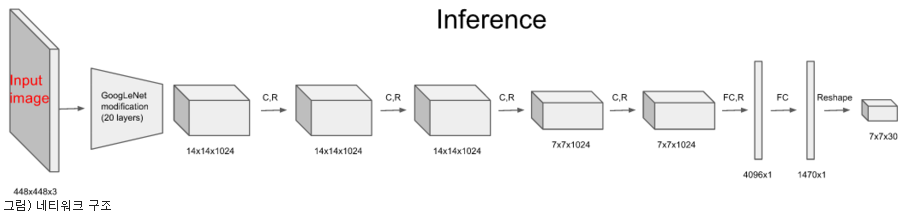
* 기존 R-CNN 계열은 속도가 느림 (Region Proposal 된 수가 많음)
* YOLO는 proposal 방식에 있어 grid 방식을 채택함.
* grid - image를 S x S 개의 cell로 나누어 각 cell이 곧 proposal 수가 됨.
* YOLO는 bounding box (네모 영역)와 class probability (어떤 물체일 확률)를 single regression 으로 간주하여 하나의 네트워크에서 진행하여 빠르다.



* 왼쪽 이미지가 input image 이고 7 x 7 grid(YOLOv1 case)로 나눈 것이다.
* 중간의 이미지 2개가 YOLO network 출력의 결과물이다. 위의 이미지가 각 cell 당 예측한 bounding box이며 object가 있을 확률(confidence score)이 높은 것은 굵게 표현되었다. 각 cell당 예측하는 bounding box의 개수(anchor)는 2개이다.(YOLOv1 case)

아래의 이미지는 해당 cell에 어떤 class(개, 자전거 등)가 있을 지에 대한 정보(class probability)이다. class는 20개이다. (YOLOv1 실험 논문 case)

* 우측 이미지는 network의 결과물을 통해 생성한 것이다. (NMS)



* YOLOv1 Network는 448x448x3 input image를 받아 24회의 convolution layer, 2회의 fully connected layer를 거쳐 reshape를 통해 7 x 7 x 30의 출력을 갖는다.

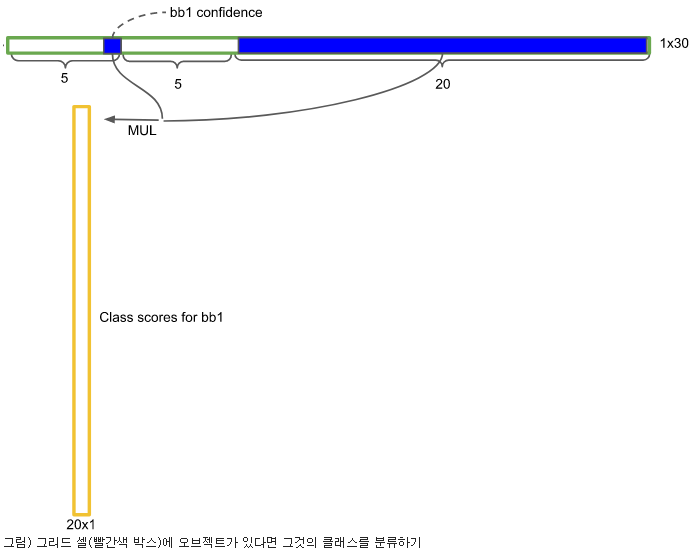
shape -> S x S x (5 x B + C), S x S : grid, B : Bounding Box 수, C : Class 개수

* 각 cell에 30개 채널에 정보는 2개의 Bounding Box에 대한 정보와 각 class에 대한 정보가 있다.
  + BoundBox 정보 – 1. Bbox에 object가 있을 확률(confidence)

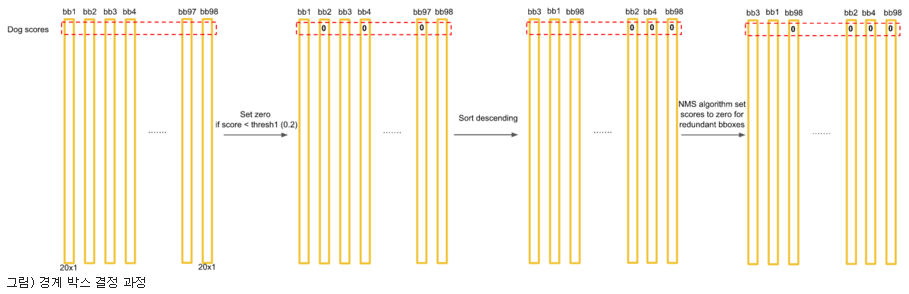
2~3. Bbox의 중심점 좌표(해당 cell의 중심에 대해 0~1값을 가짐)

4~5. Bbox의 가로 세로값(전체 image 크기에 대해 0~1값을 가짐)

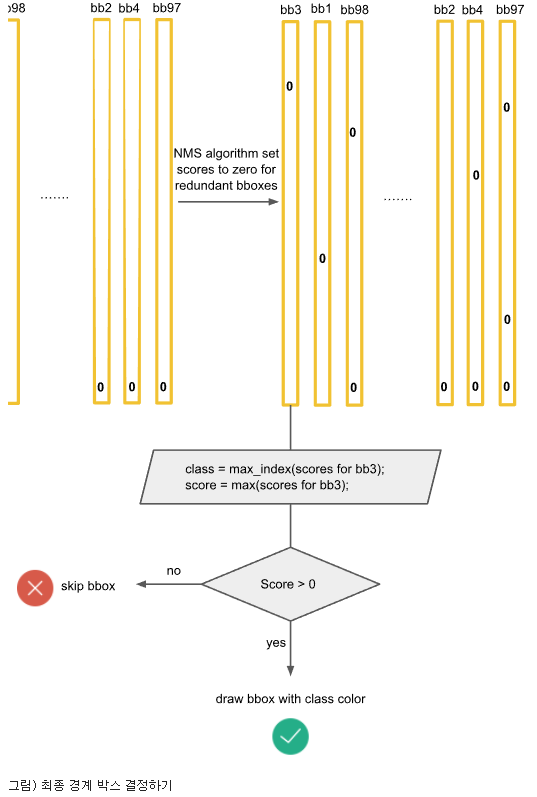
11~30. 각 class의 probability



* 각 grid cell의 채널에는 2개의 예측된 Bbox가 있고 그 Bbox에 object가 존재할 confidence와 각 class의 probability를 곱하면 2x7x7= 98개의 Bbox에 대한 class별 확률이 나오게 된다.



* 첫 번째 class 가 dog class 이고 그 경우를 보면 우선, score 가 threshold(0.2) 이하인 것은 해당 class는 절대 없다고 판단하고 0으로 만들어 준다.
* 그리고 높은 값 순으로 정렬(sort)한다.
* 위 과정까지 하면 한 object에 대해 여러 경계박스가 나타날 수 있다. 중복을 제거하기 위해 NMS(Non-Maximal Suppression, 비-최대값 억제) 알고리즘 사용하여 중복을 제거한다.
* NMS 알고리즘은 Bbox가 일정치(0.5) 이상 겹쳐있을 때 최대값을 갖는 하나만 남기는 것.
* NMS 에서는 0.5 이상 겹치지 않으면 또 다른 object가 있을 수 있다고 판단하고 삭제하지 않는다.
* 모든 class에 대해 위 과정을 수행한다.



* 각 Bbox에 대해 class 및 score를 정의한다. class는 가장 높은 score의 index이며 score는 가장 높은 score값이 된다. 그리고 score값이 0이라면 Bbox를 그리지 않는다.
* 그리고 결과값(score)값이 0.5 이하이면 지워진다고 하는 글도 있음.
* 위 과정을 모든 Bbox에 대해 수행한다.
* 최종적으로 object가 있을거다 라고 추측한 Bbox만이 그려진다.