You Only Look Once(YOLO)

우리는 평가할 때 conditional class probabilties와 각 박스의 class-spectific confidence score를 주는 confidence prediction을 곱했다. 이 점수는 박스안에 존재하는지와 박스가 물체에 얼마나 적합한가를 모두 포함한다.

YOLO를 PASCAL VOC로 평가할 때 S = 7 B = 2이다. PASCAL VOC는 20개의 라벨 class가 있으므로 C = 20이다 따라서 우리의 최종 예측 결과를 7*7*30 tensor로 나타난다.

- GoogleNet을 참고하여 Classifier를 먼저 만들었다.

PASCAL VOC detection dataset으로 평가했다. inital convolution layer는 image로부터 feature를 추출하고, FC layer는 호가률과 좌표를 예측한다. 우리 신경망 구조는 이미지 분류를 위한 GoogleNet에 영향을 받았다. 우리 신경망은 24개의 합성곱층과 2개의 FC layer로 이루어져있다. GoogleNet에서 사용된 inception module을 대신하여 우리는 1x1 convolution layer와 3x3 convolution layer를 사용했다. 진체 신경망은 Figuer 3에 나타아 있다. 우리는 또한 빠른 버번의 YOLO를 만들었다. Fast YOLO는 9개의 합성곱층만을 이요하였으며 필터의 개수는 더 적다. 신경망의 크기를 제외한 다른 요소들은 동일하다.

우리는 ImageNet의 1000-class competition dataset으로 사전 훈련을 했다 사전 훈련을 위해 20개의 합성곱층을 사용했고 average-pooling alyer와 FClayer를 사용했다.

우리는 이를 detection을 위한 모델로 바꾸었다 .Ren etal은 사전 학습된 신경망에 합성곱층과 FC layer를 모두 추가하는 것은 성능을 높인다는 것을 보였다. 그들의 예제에 따라 우리는 임의의 초기 변수를 갖는 4개의 합성곱층과 2개의 FC layer를 더했다. detection은 때때로 매끄러운 시각 정보를 요구하기 때문에 입력 해상도를 224x224 에서 448x448로 올렸다.

- detector를 만들기 위해 class들에 대한 확률뿐만 아니라 Bounding box 위치 또한 예측하도록 했다.

마지막 layer는 classprobabtilites와 bounding box coordinates 모두 예측합니다. 우리는 bounding box의 너비와 높이를 image의 너비와 높이로 정규화하여 0~1 사이로 만들고 x,y 좌표를 특정 셀 격자 위치의 offset으로 설정하였고, 그들은 0과 1사이의 값을 가집니다. 마지막 layer에서는 선형 활성화 함수를 사용했고 그 외의 모든 layer에서 아래와 같은 leaky rectified linear activation을 사용했습니다.

$$\phi(x) = \begin{cases} x, & \text{if } x > 0 \\ 0.1x, & \text{otherwise} \end{cases}$$

- 알고리즘의 문제점

우리는 MSE를 바탕으로 신경망을 최적화했다. 제곱합 에러를 사용한 이유는 최적화가 쉬워서이지만, 우리의 목표인 평균 정호가도 최대화와는 조금 달랐다. 이는 localization error와 classification error의 비중을 같게 뒀지만 사실 적합하지 않을 수 있다. 또한 많은 격자셀은 물체를 포함하지 않는다. 이는 그들의 신뢰 점수를 0으로 만들었고, 종종 물체를 포함한 셀의 gradient를 압도했다.

- 개선 방법 1. 두가지 loss의 비중을 달리한다(localization, classification)
  2. 너비와 높이의 제곱근 값을 사용한다.

이를 개선하기 위해 물체를 포함하지 않은 box들에 대해 bounding box coordinate의 loss를 증가시키고 confidence predictions의 loss를 감소시켰다. 이를 위해 lamda_coord=5와 lamda_noobj=0.5라는 변수를 사용했다. MSE는 큰 상자오 ㅏ작은 상자의 에러에 대해 같은 비중을 뒀다. 우리의 에러 측정은 큰 상자와 작은 편차가 작은 상자의 것보다 영향이 작도록 반영했다. 이를 부분적으로 반영하기 위해 너비와 높이를 바로 사용하기보다 제곱근 값을 체측한다.

- '1 물체 = 1 bounding box'를 위해 가장 높은 IOU를 같은 grid cell에 'predictor'라는 이름을 붙인다.

셀격자 하나가 복수의 grid box를 예측한다. 그러나 우리는 하나의 물체에 대해 bounding box predictor가 하나만 존재하길 원한다. 가장 높은 current IOU를 갖게 예측하는 predictor가 "resopnsible"하다. 이로 인해 bounding box predictor 사이에서 specialization이 발생한다. 이로 인해 각 predictor는 특정 크기, 종횡비 또는 물체의 class를 더 잘 예측할 수 있꼬, 이는 전체적인 검출률을 향상시킨다.

만약 물체가 셀 격자 내에 있으면 loss function은 확률에 대한 error만 계산한다는 것을 알아야한다. 그것은 또한 predictor가 "respnsible"할 경우에만 coordinate error를 부과한다.

우리는 PASCAL VOC 2007과 2012의 학습과 점검 데이터셋을 epochs=134로 학습시켰다. 2012로 평가할 때도 batch size = 64, momentum = 0.9, decay = 0.0005으로 학습시켰다.

우리의 learning rate schedule은 다음과 같다. 첫 epoch에는 learning rate을 0.001에서 0.01로 천천히 증가시켰다. 만약 높은 learning rate으로 시작한다면 unstable gradient에 의해 diverge(발산)한다. learning rate를 0.01로 75 epochs를 돌린뒤, 0.001로 30 epoch, 0.0001로 30epoch를 더 돌렸다.

overfitting을 피하기 위해 우리는 dropout과 data argument를 사용했다. layer 사이의 co-adaptation을 막기위해 첫번째 FC layer이후에 dropout 비율을 0.5로 했다. data augmentation을 위해 random scaling고 ㅏ원래 이미지의 20%에 달하는 translation을 적용했다. HSV 색상맵에서 random한 expousre과 1,5의 saturation을 적용했다.

학습할 때와 같이, 평가 이미지에서 검출을 예상하는 것은 한 번의 network evaluation이면 된다. PASCAL VOC에서, 신경망은 이미지마다 98개의 bounding boxed를 예측하고, 각 box에 대해 class probabilities를 계산한다. YOLO는 classifier-based network를 사용한 다른 방법과 달리 한 번의 network evaluation만 요구하기 때문에 매우 빠르다.

격자 디자인은 bounding box 예측에서 공간적인 다양성을 야기한다. 물체가 어느 셀격자 안에 있는지 명확하고 물체 하나에 대해 box 하나만 예상될 경우가 종종 있다. 그러나, 몇몇 큰 물체나 여러 셀들의 의해 위치가 정해진다 이런 multiple detection을 수정하는데, non-maximal suppression(NMS)이 사용될 수 있다. 이는 mAP 값을 2~3% 정도 향상시키지만 R-CNN이나 DPM에서 보여준 성능 향상보다는 저조하다.

- non-maximal suppression(NMS)는 겹쳐진 bounding box를 제거하는데 사용된다. 한 문체에 여러 개의 bounding box가 겹쳐있으면 confidence가 가장 높은 bounding box를 기준으로 다른 bounding box와의 IOU를 계산한다. IOU가 높다는 것은 두 bounding box가 많이 겹쳐져 있다는 뜻이므로 IOU가 높은 bounding box는 제거한다.

YOLO에서 각 격자셀은 두 개의 box만을 예측하고 하나의 class만 가질 수 있기 때문에 bounding box를 예측할 때 강한 공간 제약을 야기한다. 이 공간 제약은 우리 모델이 예측 가능한 근처의 물체의 수를 제한한다. 우리 모델은 새 떼처럼 작은 물체들이 그룹지어 나타날 경우 혼란에 빠진다. → YOLO의 단점

우리 모델이 데이터로부터 경계상자를 예측하는 법을 배우기 때문에 새롭거나 특수한 종횡비 또는 구성을 갖는 물체를 일반화하기 어렵다. 우리 구조에서 입력 이미진느 다수의 downsampling layer를 거치기 때문에 우리 모델은 상대적으로 거친 feature를 사용한다.

마지막으로 detection 성능에 가까운 loss function으로 학습하는 동안 우리의 loss function은 작은 경계상자와 큰 경계상자의 에러를 같게 본다. 큰 박스의 작은 에러는 보통 무시할만 하지만, 작은 박스의 작은 에러는 IOU에 큰 영향을 준다. 에러의 가장 큰 원인은 정확하지 않은 위치이다.

Object detection은 computer vision에서 핵심 문제이다. 검출 pipline은 일반적으ㅗㄹ 입력 이미지에서 강력한 feature set을 추출하는 것으로 시작한다. 그러면 classifier나 localizer는 feature space에서 물체를 인식힌다. 이들은 전체 이미지를 sliding window 방

식이나 이미지의 부분을 보는 방식으로 작동한다. 우리는 YOLO와 top detection framework들을 비교하며 공통점고 ㅏ차별점을 찾아보았다.

Object detection은 computer vision에서 핵심 문제이다. 검출 pipeline은 일반적으로 입력 이미지에서 강력한 feature set을 추출하는 것으로 시작한다. 그러면, classifier나 localizer(?)는 feature space에서 물체를 인식한다. 이들은 전체 이미지를 sliding window 방식이나 이미지의 부분을 보는 방식으로 작동한다. 우리는 YOLO와 top detection framework들을 비교하며 공통점과 차별점을 찾아보았다.

Deformable parts models(DPM)은 sliding window 접근법을 사용하여 물체를 검출한다. DPM은 서로 다른 pipline을 사용하여 static feature를 추출하고, 높은 점수 영역의 bounding boxes를 예측하는 등의 일을 한다.