🤔

YOLO v2

정확도와 속도 사이의 trade off 간 균형을 잘 맞춘 모델

Better

- 정확도를 올리기 위한 방법
- **Batch Normalization**
  - 모든 convolution layer 뒤에 batch normalization을 추가함
  - overfitting 없이 다른 regularization 방법이나 dropout을 제거함
- **High Resolution Classifier**
  - YOLO v1과 달리 처음부터 448x448 사이즈로 pretrain시킴
  - 네트워크가 상대적으로 높은 해상도의 이미지에 적응할 시간을 제공
- **Convolutional with Anchor Box**
  - YOLO v1과 달리 anchor box를 이용하며 네트워크 수정함
    - pooling layer을 이용해 convolution layer의 output이 높은 resolution을 가지도록 함
    - input size는 416x416으로 줄여 최종 output feature map의 크기가 홀수가 되도록 함

      → feature map 내에 single center cell이 존재하도록 해, 이미지 크기가 클 경우 잘 detect할 수 있도록 함
    - 최종적인 feature map은 13x13 사이즈
  - anchor box를 이용해 보다 많은 수의 bounding box를 예측함

    → recall 값 상승
- **Dimension Clusters**
  - `k-means clustering` 을 통해 최적의 prior 선택 (ground truth box의 width, height 값 사용)

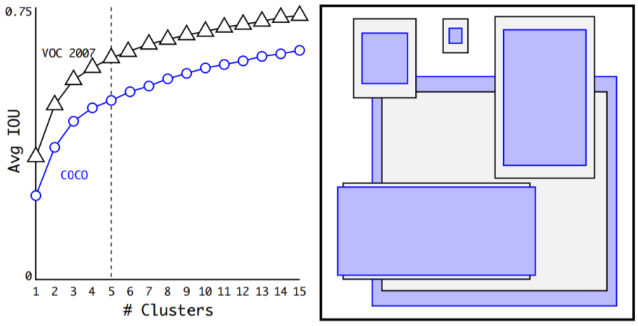
- distance metric을 사용하는 경우 큰 bounding box는 작은 box에 비해 큰 error을 발생시킴

  → 새로운 metric 사용

$$d(box, centroid) = 1 - IOU(box, centroid)$$

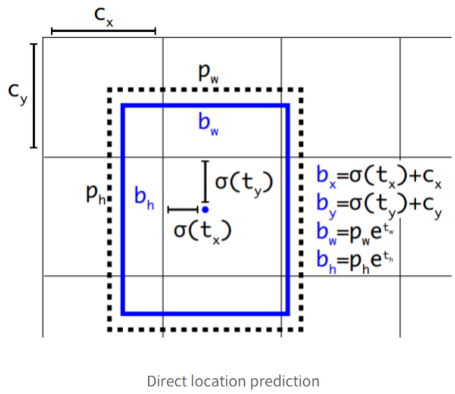
- box와 centroid의 IoU 값이 클수록 겹치는 영역이 크기 때문에 거리가 가까움
- k=5일 때 최적의 값을 가짐

  → 5개의 anchor box만으로도 최적의 prior을 선택하면 네트워크가 detection task를 보다 쉽게 학습함



Cluster 수와 평균 IoU 값

- **Direct Location Prediction**
  - 제한된 범위가 없어 anchor box를 임의의 지점에 위치할 수 있다는 문제점 → 초기 모델 불안
  - grid cell에 상대적인 위치 좌표를 예측하는 방법으로 이 문제를 해결함
    - bounding box regression을 통해 얻은 tx, ty 값에 logistic regression을 적용해 0~1 값을 갖도록 조정함



Direct location prediction

- 예측하는 위치의 범위가 정해짐으로써 네트워크는 안정적으로 학습할 수 있음

⇒ Dimension clustering을 통해 최적의 prior를 선택하고 anchor box의 중심부 좌표를 직접 예측해 recall 값을 5% 정도 향상시킴

- **Fine-Grained Features**
  - 13x13 feature map을 최종적으로 출력 → 작은 객체는 예측하기 어려움
  - 마지막 pooling을 수행하기 전에 feature map을 추출해 26x26 크기의 feature map을 얻음
  - 이후 channel은 유지하면서 4개로 분할 후 결합 → `13x13x2048` feature map
    - 보다 작은 객체에 대한 정보를 함축하고 있음
  - `13x13x1024` feature map에 추가해 `13x13x3072` feature map을 얻음
  - 최종적으로 3x3 conv와 1x1 conv를 적용해 `13x13x125` feature map 얻음
- **Multi-Scale Training**
  - 다양한 입력 이미지를 사용해 네트워크를 학습시킴
  - 10 batch마다 입력 이미지의 크기를 랜덤하게 선택해 학습하도록 설계
  - 이미지를 1/32배로 downsampling하기 때문에 이미지의 크기는 32배수 중 하나

Faster

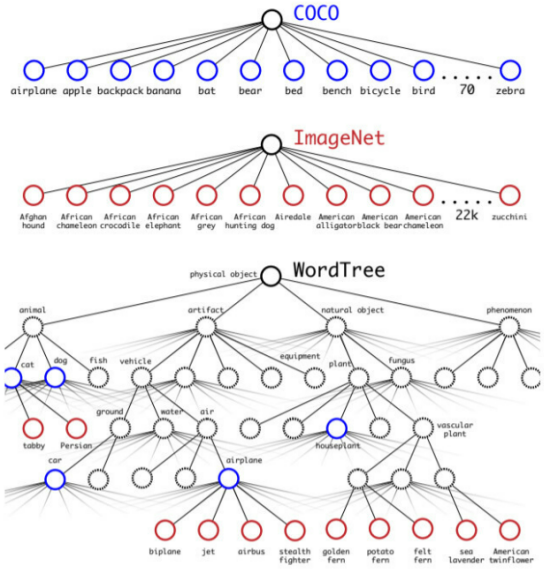
- detection 속도를 향상시키는 방법
- Darknet-19라는 독자적인 classification 모델을 backbone network로 사용함
  - 마지막 layer에 fc layer 대신 `global average pooling` 을 사용

    → 파라미터 수를 감소시키고 detection 속도 향상시킴

- **Training for classification**
  - class 수가 1000개인 imagenet 데이터셋을 통해 학습시킴
- **Training for detection**
  - 마지막 conv layer을 3x3x1024 conv layer로 대체하고, 이후 1x1 conv layer 추가

Stronger

- 더 많은 범위의 클래스를 예측하는 방법
- 학습 시 classification과 detection 데이터를 섞어 학습시킴 → '개'와 '요크셔테리어' 별개로 구분
- **Hierarchical Classification**
  - WordTree를 구성하는 방법
  - detection loss는 기존과 같이 loss를 backward pass
  - classification loss는 특정 범주와 상위 범주에 대해서만 loss 계산
  - `Joint training` 방식을 통해 이미지 내에서 객체를 찾는 detection task와 imagenet 데이터셋을 통해 보다 넓은 범주의 객체를 분류할 수 있도록 학습함



Combining ImageNet and COCO dataset

Training

- **Feature extraction by Darknet-19**
  - 13x13x1024 feature map 얻음
- **Reorganize feature map**
  - 앞서와 같은 방법을 통해 13x13x2048 feature map을 얻음
  - 보다 작은 객체에 대한 정보를 함축하고 있음
- **Concat feature maps**
  - 1번과 2번 과정에서 얻은 feature map을 채널에 따라 결합해 13x13x3072 feature map 얻음
- **Prediction by applying conv layers**
  - 3x3 conv와 1x1 conv 연산을 적용 → 13x13x125 feature map 얻음
  - 채널 수는 5개, bounding box별로 20개의 class score와 5개의 값을 예측

    → grid cell별로 125개의 값을 가지게 됨
- **Train YOLO v2 by loss function**
  - Localization loss, Confidence loss, Classification loss로 구성되어 있음 → SSE
  - 공식

Localization loss

$$\lambda_{obj}^{coord}\sum_{i}^{S^2}\sum_{j}^{B}1_{ij}^{responsible\_obj}(x_{ij}^{pred}-x_{ij}^{obj})^2+(y_{ij}^{pred}-y_{ij}^{obj})^2+(w_{ij}^{pred}-w_{ij}^{obj})^2+(h_{ij}^{pred}-h_{ij}^{obj})^2$$

$$+\lambda_{noobj}^{coord}\sum_{i}^{S^2}\sum_{j}^{B}1_{ij}^{no\_responsible\_obj}(x_{ij}^{pred}-x_{ij}^{anchor\_center})^2+(y_{ij}^{pred}-y_{ij}^{anchor\_center})^2+(w_{ij}^{pred}-w_{ij}^{anchor\_default})^2+(h_{ij}^{pred}-h_{ij}^{anchor\_default})^2$$

Confidence loss

$$+\lambda_{obj}^{conf}\sum_{i}^{S^2}\sum_{j}^{B}1_{ij}^{responsible\_obj}\{conf_{ij}^{pred}-iou(box_{ij}^{pred},box_{ij}^{truth})\}^2$$

$$+\lambda_{noobj}^{conf}\sum_{i}^{S^2}\sum_{j}^{B}1_{ij}^{no\_responsible\_obj}\{conf_{ij}^{pred}-0\}^2$$

Classification loss

$$+\sum_{i}^{S^2}\sum_{j}^{B}1_{ij}^{responsible\_obj}\{p_{ij}^{pred}(c)-p_{ij}^{truth}(c)\}^2$$

YOLO v2 loss function