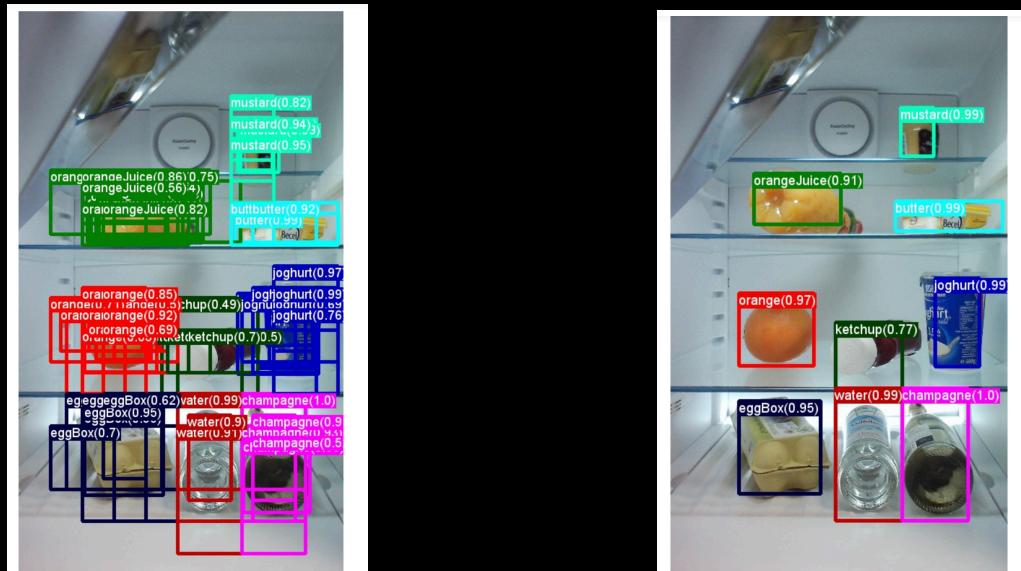
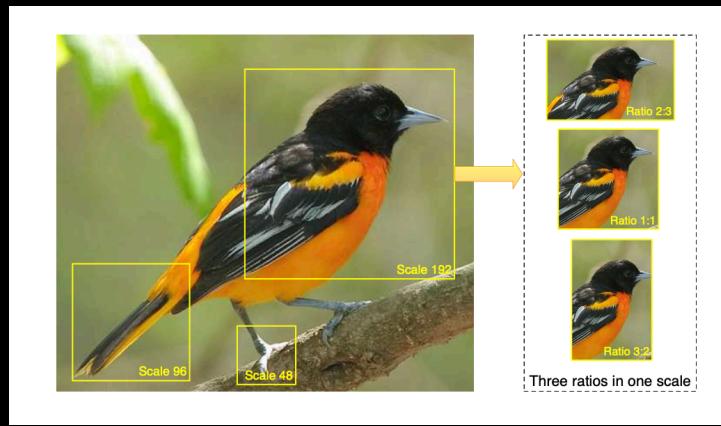


### < 3.2 Navigator and Teacher >

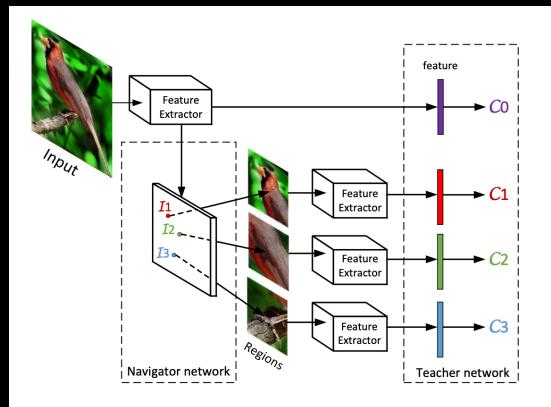
- Navigator는 가장 Informative 한 Region을 찾아내는 역할을 한다. 이때 RPN (Region Proposal Network과 내부 Anchor과 유사한 작동 방식을 따라한다.
  - RPN : Object detection에서 핵심적인 역할을 한다 (Faster R-CNN). 기존 Fast R-CNN 에서 가장 큰 계산 부하를 차지하는 region proposal 생성을 새로운 방식으로 작동시키고, 이를 내부(GPU)로 통합.
  - 만들어낸 region proposal이 객체인지 아닌지, 그리고 실제 ground truth에 맞춰 좌표를 regression할 수 있는지. 다양한 이미지를 입력 값으로 object score와 object proposal을 출력.
  - Regression과 classification을 수행해준다.
  - Feature map의 마지막 convolution 층을 작은 네트워크가 sliding 하여 저차원으로 mapping 해준다.
  - Anchor : 각 sliding window에서 bounding box의 후보로 사용되는 상자.
  - NMS : 동일한 클래스에 대해 높은-낮은 confidence 순서로 정렬. Confidence가 가장 높은 boundingbox와 IOU가 일정한 bounding box는 동일한 물체를 detect 했다고 판단하여 지워준다.
  - IOU : 두가지 물체가의 위치 (Bounding Box)가 얼마나 일치하는지 수학적으로 나타냄
    - >  $IOU = \text{Area of Overlap} / \text{Area of Union}$



- 위 사진에서 왼쪽은 NMS를 하기전, 오른쪽은 NMS를 진행해준 후. 가장 의미 있는 bounding box만 남겨주는 과정이라고 생각하면 되겠다.



- Anchor를 활용하여 multiple region proposal 을 진행한다. 각 anchor는 sliding window position, aspect ratio 그리고 box scale과 연관이 있다.
  - 448 size의 input 이미지는 anchor의 scale가 {48,96,192} 그리고 ratio는 {1:1,3:2,2:3}을 갖는다.
- Navigator는 이미지를 입력으로 받으며 여러 직사각형의 region {R1,R2,,RM}을 생성한다, 또한 각 Region은 informativeness score를 갖게된다.



- Region이 너무 많을 수도 있으니 이를 방지하기 위해 NMS(non-maximum suppression)을 활용하며, 이 중 상위 M개를 Teacher Network만을 보낸다. 또한 이에 대한 피드백을 Teacher network가 다시 Navigator에게 전송해준다.
  - NMS :
- Navigator가 informative function, Teacher는 Confidence function을 최적화 하는 방식으로 학습을 진행한다.

### < 3.3 Scrutinizer >

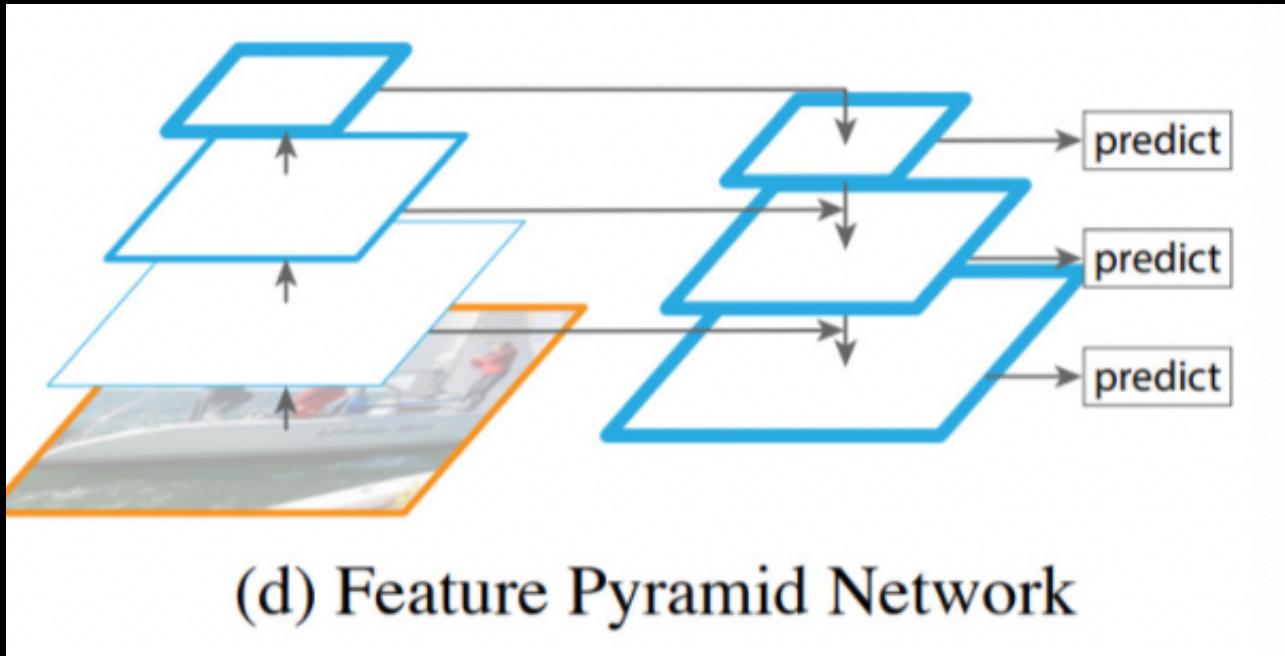
- Navigator가 학습이 될 수록, characteristic 이 도드라져 Scrutinizer가 의사결정을 내리기 용이하게 만든다.
- 상위 K 개의 Informative Region과 Full image를 활용하여 Scrutinizer를 학습시키며, 이때 informative region을 통해 intra class variance (같은 클래스의 이미지들 간의 분산)을 줄여주어 Scrutinizer가 fine grained classification을 하는데 도움을 준다

#### < 3.4 Network Architecture >

- 전반적으로 ResNet + NTS의 형태이고, 이때 ResNet가 Feature Extractor의 역할을, 나머지 NTS는 feature extractor의 parameter들을 공유한다.

##### - Navigator Network :

- Feature Pyramid Network를 따른다.



- 다양한 크기의 객체를 인식하는 것은 object detection의 핵심 문제. 그러나 기존 모델들은 추론 속도가 너무 느리고, 메모리를 지나치게 많이 사용하는 문제점이 있다.
- FPN은 컴퓨팅 메모리를 적게 차지하고, 다양한 크기의 객체를 인식하는 방법을 제시한다.
- 기본적으로 convolutional network에서 입력층에 가까울 수록 feature map은 높은 해상도를 가지며, 멀어질 수록 (가장자리, 곡선)은 낮은 특징과 낮은 해상도를 갖는다.
- 임의의 크기의 single scale 이미지를 convolutional network에 입력하여, 다양한 크기의 feature map을 추출 및 수정하는 네트워크.
- 계층적인 convolutional pyramid layer
- Pyramid : 이는 convolutional network에서 얻을 수 있는 서로 다른 해상도의 feature map을 쌓아 올린 형태이다 또한 level이란 피라미드 각층의 feature map을 의미한다. 여러 level이 쌓아올려진게 pyramid.
- Convolutional layer는 여러 convolution filter들로 쌓여있는 형태이다. 이때 각 convolution filter들을 low level부터 high level 이미지들의 feature를 추출한다.
- Pyramid 구축을 하는 과정에서 bottom up pathway, top-down pathway, lateral connection 과정을 거친다.
- Bottom-up pathway는 이미지를 convolutional network에 입력하여, forward pass하면서 2배씩 작아지는 feature map을 추출하는 과정이다. 이때 각 stage의 마지막 layer이 가장 깊음으로 feature map이 가장 좋은 정보를 가지게 되어, 마지막 Layer의 feature map을 추출하게 된다.
- Top-down pathway는 각 pyramid level에 있는 feature map을 2배로 upsampling 하며 channel 수를 맞춰주는 역할을 한다. 각 pyramid level의 feature map이 2배로 upsampling을 하며, 아래 pyramid level의 feature map과 동일한 크기를 가지게 됩니다. (Nearest neighbor up sampling 활용).

- Lateral Connection : upsampling된 feature map과 아래 level의 feature map의 element wise addition 연산을 수행.
- 해당 과정들을 통해 각기 다른 scale에서의 feature map들을 추출할 수 있다.

### **< Navigator Network >**

- Feature Extractor가 추출해준 Feature map을 받는다
- FPN과 같이 Top Down 구조를 갖는다. 활성화 함수로 ReLU를 사용하고 Max pooling을 활용한다.
- Anchor를 활용하여 informative Region을 찾고 이를 통해 informativeness를 환산
- 전체 이미지로부터 특정 Region을 Crop 및 Resize 후 Teacher Network에 전달

### **< Teacher Network >**

- Navigator에게 받은 정보로 confidence를 산출하고 이를 다시 Navigator에게 전달해준다.
- Navigator가 informativeness를 confidence와 동일한 순서를 갖도록 최적화시켜준다.

### **<Scrutinizer Network >**

- 상위 K개 informative regions을 Navigator로부터 전달받는다. ( 224\*224로 Resize 된 상태)
- Feature Extractor를 통해 K 개의 informative region은 2048의 feature vector로 변환되고, 이를 전달받는다.
- 본래 Full image의 Feature를 전달받는다
- 전달 받은 informative region과 full region의 feature들을 concatentate해주고 이를 fully connected layer에 넣어 예측을 해준다.