Center Face : Joint Face Detection and Alignment Using Face as point

Abstract

Face detection

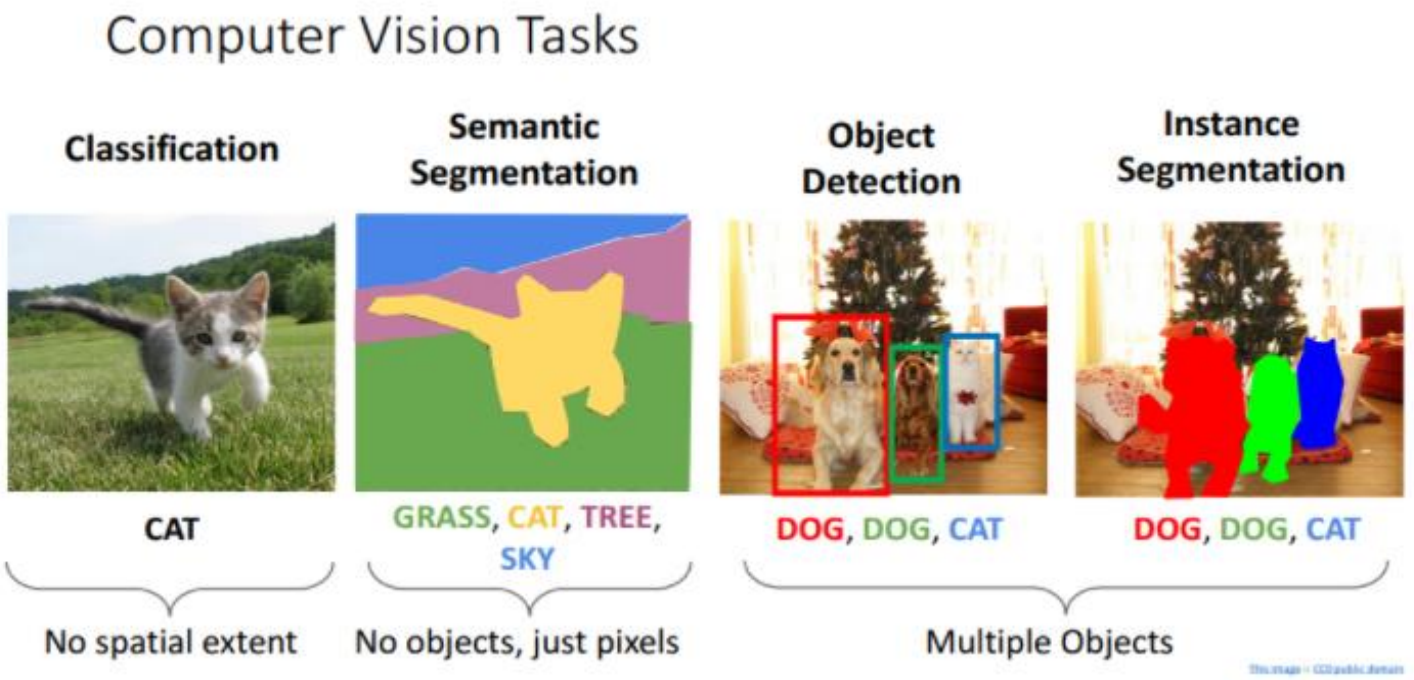
- object detection의 일환
- 낮은 메모리 용량과 컴퓨팅이 문제가 됨

Center Face – anchor free

→ semantic map에서 얼굴이 존재할 확률을 학습

→ 얼굴을 포함하고 있을 가능성이 높은 부분의 bounding box, offset, landmark 학습

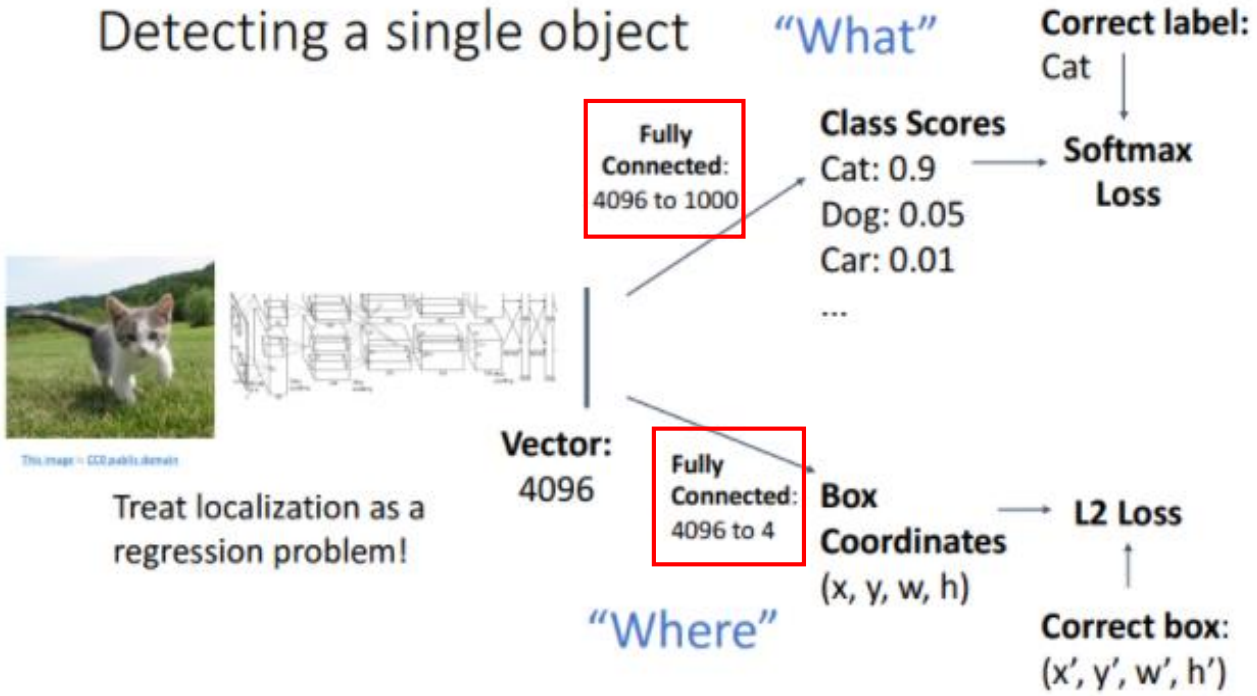
Abstract – Appendix 1 CV Tasks



Object Detection

- 입력 이미지를 주었을 때 출력으로 검출한 객체들을 가지는 작업
- 검출된 객체의 category, 객체가 이미지 상 공간의 위치 정보인 **bounding box : (x, y, w, h)** 출력

Abstract – Appendix 1 CV Tasks



Single Object Detection

① 이미지 분류

② - 4개의 출력을 갖는 완전 연결 계층

-실제 bounding box의 좌표계값, 예측한 bounding box의 좌표계값의 차이로 학습

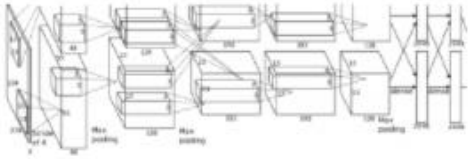
→ 두 개의 bounding box의 손실함수에 대한 가중합

Abstract – Appendix 1 CV Tasks

Multiple Object Detection – Sliding Window

Detecting Multiple Objects: **Sliding Window**

Apply a CNN to many different crops of the image, CNN classifies each crop as object or background



Dog? YES
Cat? NO
Background? NO

- CNN을 이미지의 수 많은 영역(sub-region)에 적용 → 분류 문제 (배경을 포함한 c+1개)
- 객체 검출기를 다양한 영역들에 slide → 각 영역을 CNN에 입력으로 넣어 판별

→ 비효율적 : bounding box의 경우의 수가 지나치게 많음

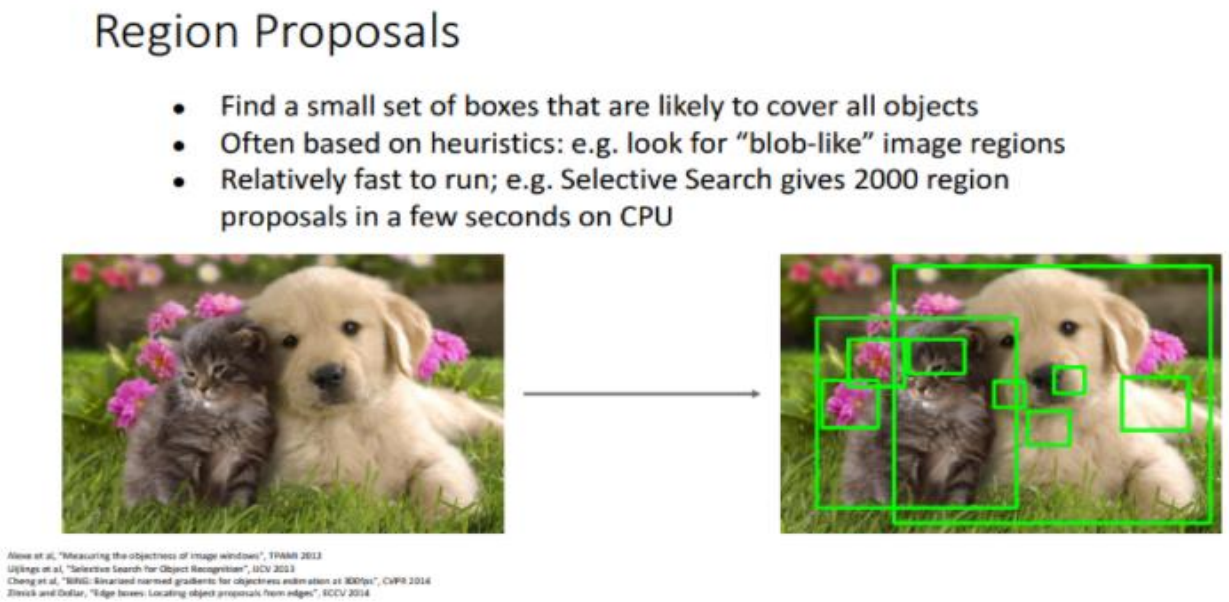
Consider a box of size h x w:
Possible x positions: W – w + 1
Possible y positions: H – h + 1
Possible positions:
(W – w + 1) * (H – h + 1)

Total possible boxes:

$$\sum_{h=1}^{H}\sum_{w=1}^{W}(W-w+1)(H-h+1)$$

$$=\frac{H(H+1)}{2}\frac{W(W+1)}{2}$$

Abstract – Appendix 1 CV Tasks

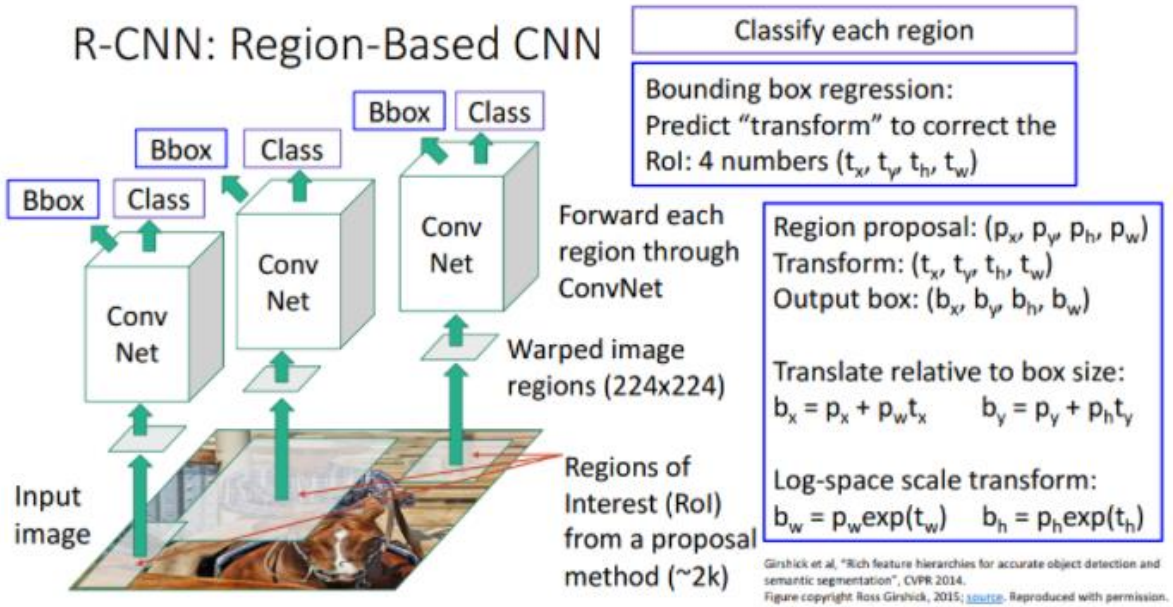


Multiple Object Detection – Regional Proposals

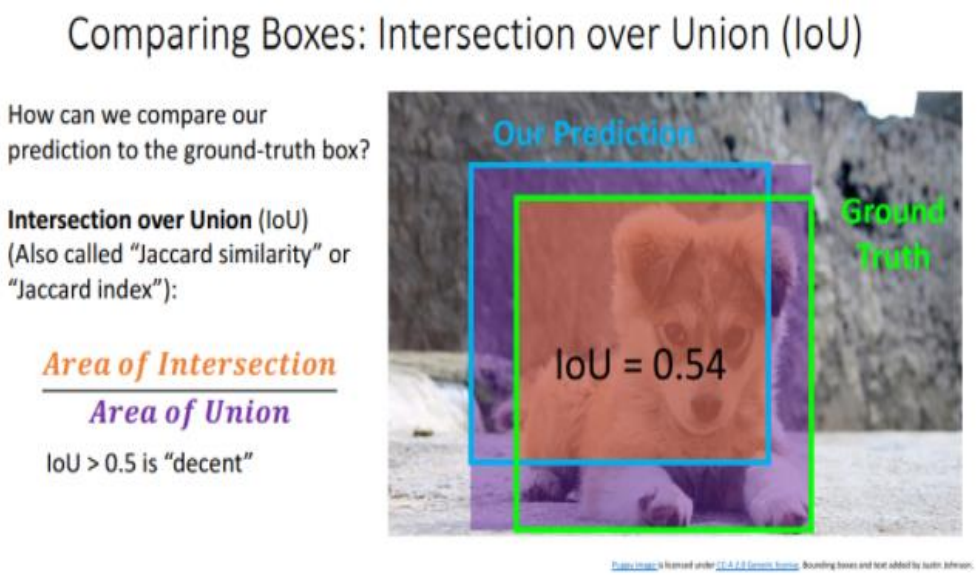
- 이미지의 후보 영역을 생성
- 이미지에서 객체에 대한 높은 확률을 가지고 있는 sub-region

해당 알고리즘 : selective search / R-CNN(Region Based CNN)

Abstract – Appendix 1 CV Tasks



Multiple Object Detection – Regional Proposals



1. 약 2000개의 **후보 제안 영역**을 제공 (selective search 등 사용)
2. 서로 다른 크기, 비를 갖는 제안 영역들을 고정된 크기로 변환
3. 각 후보 영역들에 독립적으로 CNN 적용

3-1. 분류 스코어 출력 (배경인지, 혹은 배경이 아니고 무엇인지)

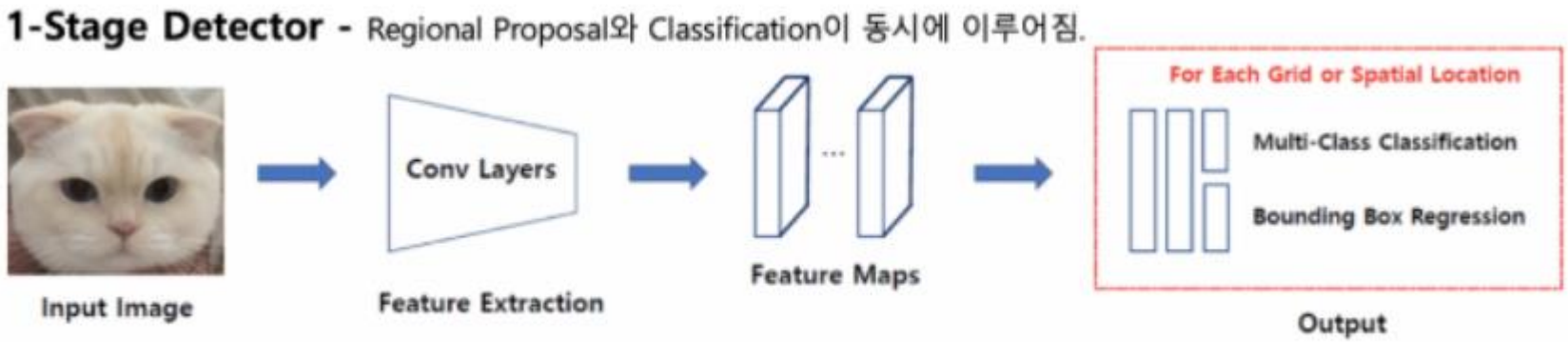
3-2. 입력으로 주어진 제안 영역을 **Bounding box**로 변환 (기존 제안 영역의 좌표가 변환됨)

→ **예측된 bounding box가 실제 box와 비슷한지 평가**

Abstract – Appendix 2 Object Detector

One-Stage Detector

- Classification, localization(regional proposal) 문제를 동시에 해결

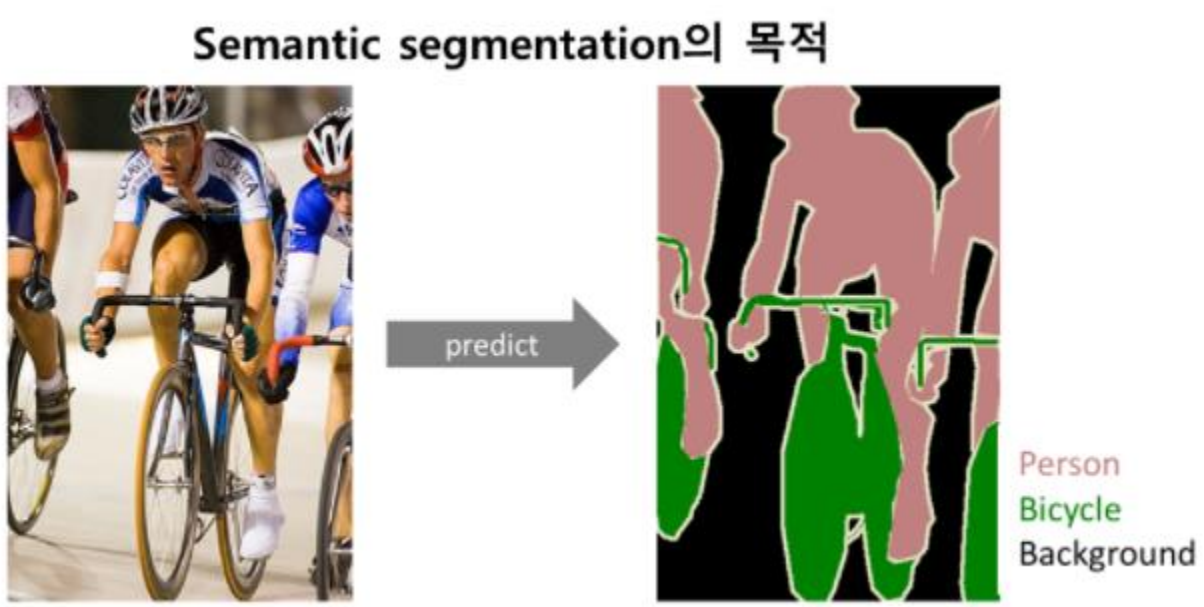


YOLO 계열, SSD 계열 (SSD, RetinaNet, RefineDet ...)

Abstract – Appendix 1 CV Tasks

Semantic Image Segmentation

- 이미지의 모든 물체들을 의미 잇는 단위로 분할 – 이미지의 각 픽셀이 어느 클래스에 속하는 지 예측
- 사진에 있는 모든 픽셀을 해당하는 category(미리 지정)로 분류 - 서로 다른 물체들을 분할
- Dense prediction – 이미지의 모든 픽셀에 대한 예측
- 검출된 객체의 category, 객체가 이미지 상 공간의 위치 정보인 **bounding box : (x, y, w, h)** 출력



자율주행

출처 : https://throwexception.tistory.com/1214?category=923813

Introduction – Precedent Methods

기존의 face detection 방식 – Anchor Based

① **one-stage** : SSD

② **two-stage** : Faster-RCNN / RPN

Drawbacks

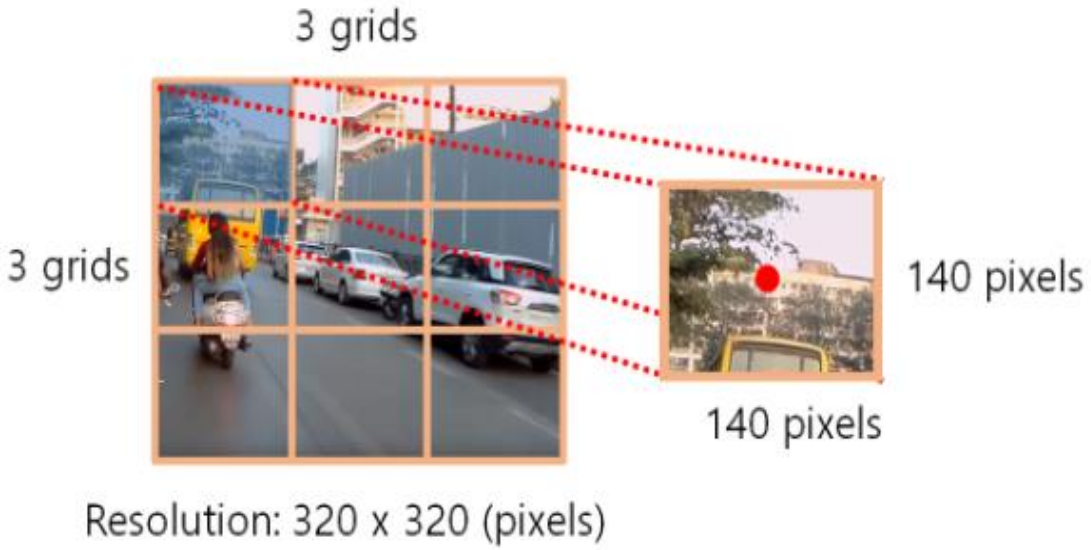
1. Anchor box와 ground truth의 높은 IOU를 위해, 많은 수의 dense anchor 필요
2. VGG16, Resnet – 높은 정확성을 보장하지만, 현업에서 사용되기 어려움 (heavy)
3. 얼굴의 facial landmark
4. 높은 정확성 못지 않게 **joint detection & alignment**도 현업에선 중요한 이슈

→ Center Face : Anchor Free

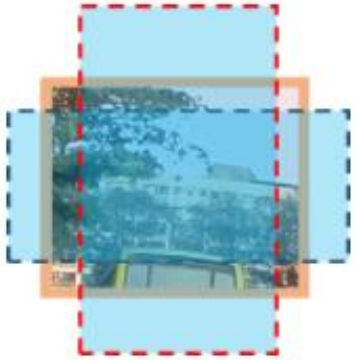
Appendix2 : Introduction – Precedent Methods

Anchor Boxes (Window, Boundary box)

1. Object detection에서 대상의 실측 bounding box를 정확하게 예측하기 위한 방법
2. 각 픽셀을 중앙에 두고 크기, 종횡비가 서로 다른 bounding box들을 생성



Anchor Boxes and Parameters



$$P_o, b_x, b_y, b_w, b_h$$

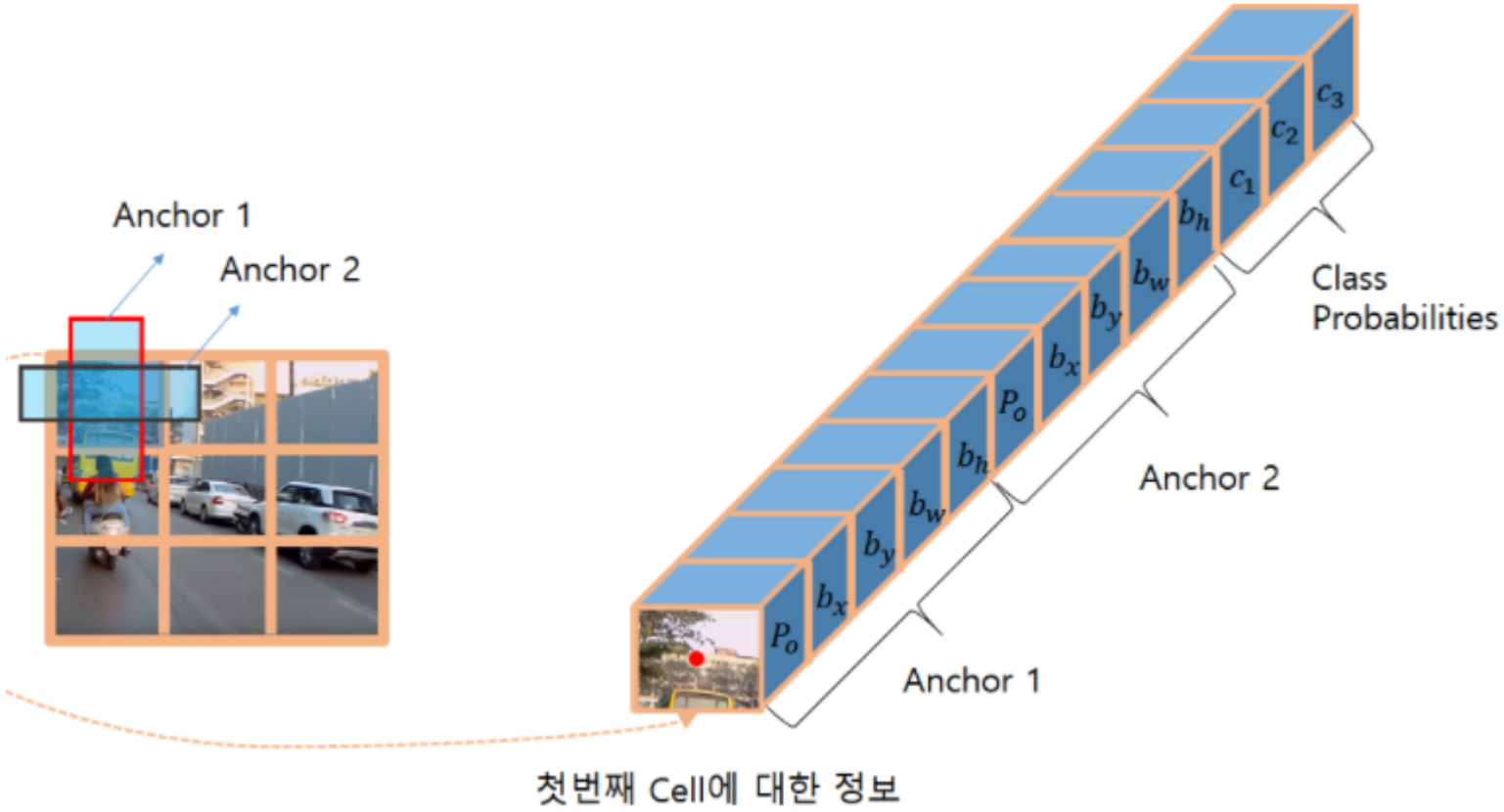
P_o: 해당 cell의 Objectness (물체가 있을 확률)

b_x, b_y: 해당 cell의 x, y 값

b_w, b_h: 해당 cell에 있는 Anchor Box의 w, h 값

$\boldsymbol{P_0}$: 클래스별 확률

Appendix2 : Introduction – Precedent Methods



Grid Cell의 정보

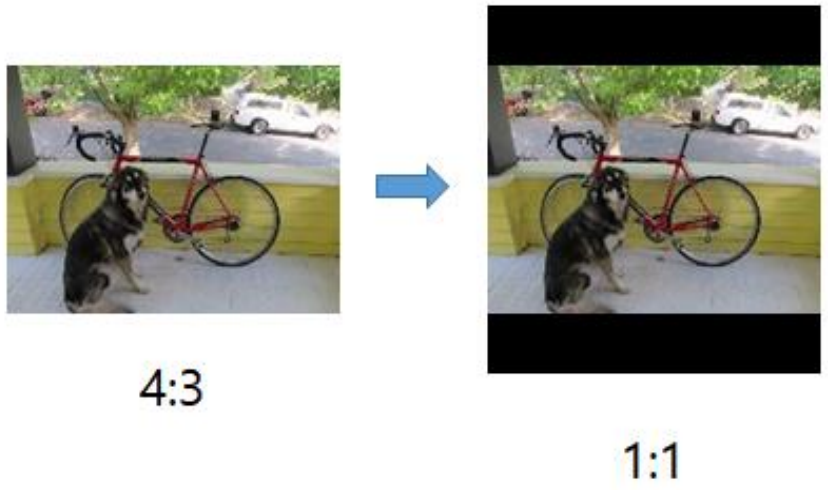
출처 : https://mickael-k.tistory.com/27

# of parameters = 9 x ((2 x 5) + 3) : **# of grid cell** x ((**# of anchors** x **# of parameters per anchor)** + **# of parameters of class probability for each anchor)**

Appendix2 : Introduction – Precedent Methods

YOLO with Anchor Boxes

① Letter Box Image 생성

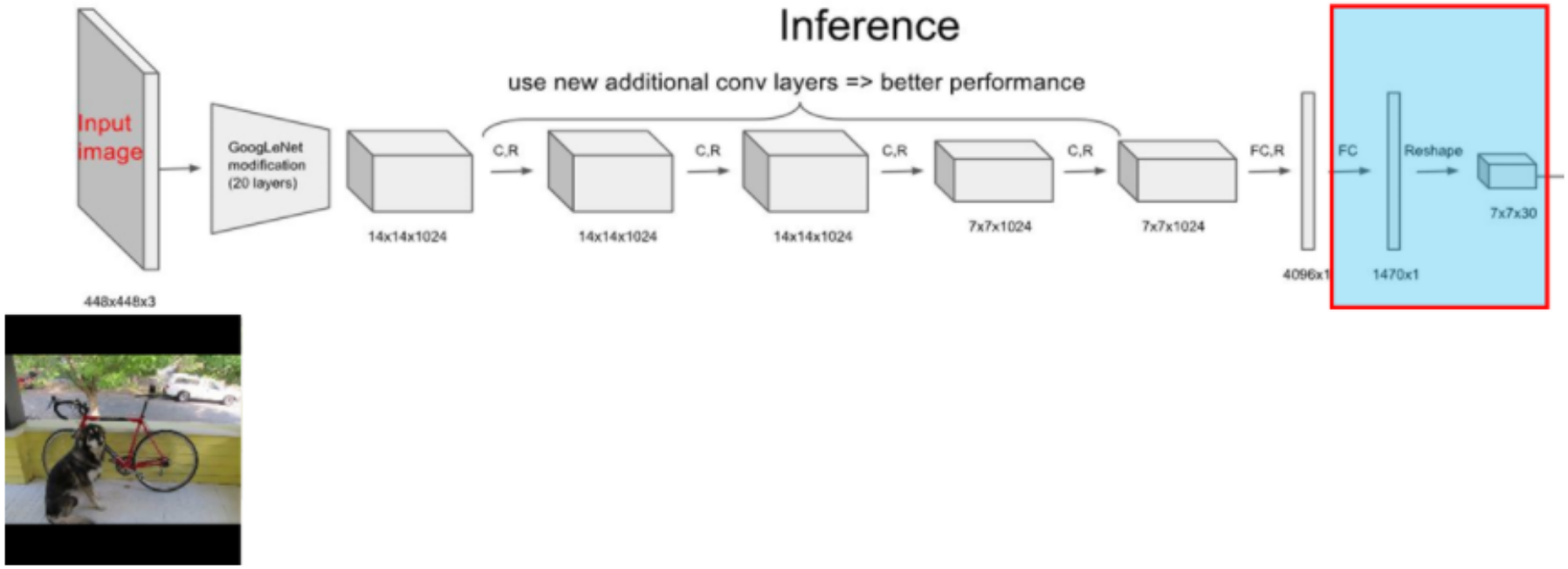


→ 입력으로 고정된 형태의 이미지가 주어져야 하므로 여분을 매꿔주는 이미지를 새로 생성

Appendix2 : Introduction – Precedent Methods

② GoogLeNet에 이미지 입력

③ 출력인 완전 연결층을(4096) 설정한 Grid Cell크기에(7x7) 맞게 변형(1470)

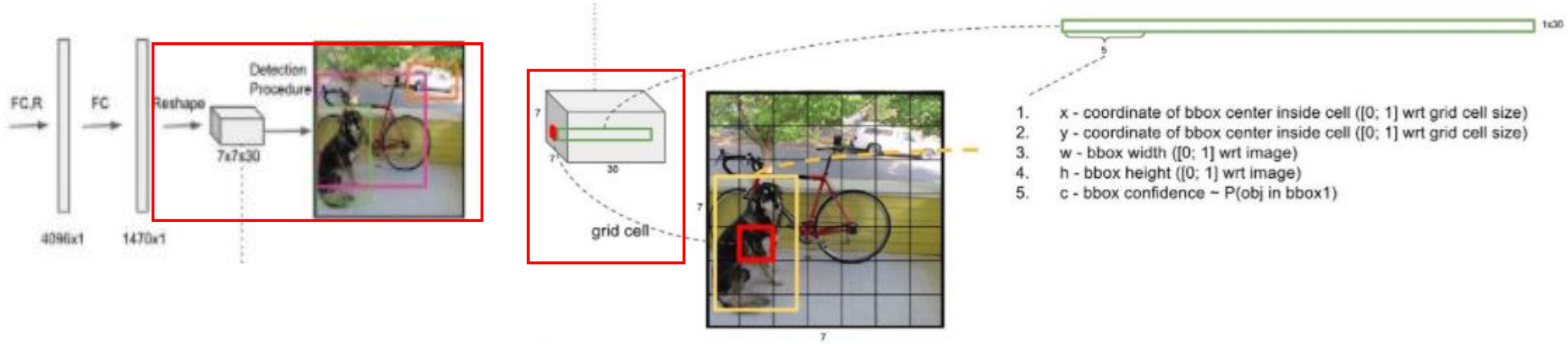


- 1470 = (7 x 7) x ((2 x 5) + 20)

1. 각 Grid Cell은 2개의 anchor box(bounding box)
2. 각 anchor box는 5개의 파라미터
3. Class는 20개

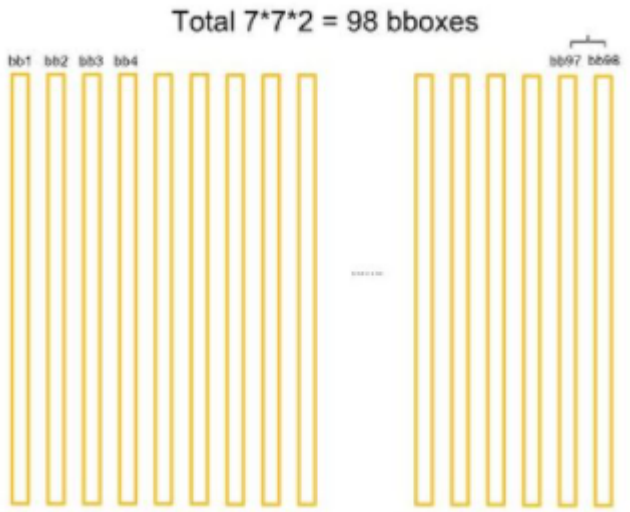
Appendix2 : Introduction – Precedent Methods

④ 각 anchor box에 대한 class confidence score계산 (P0)

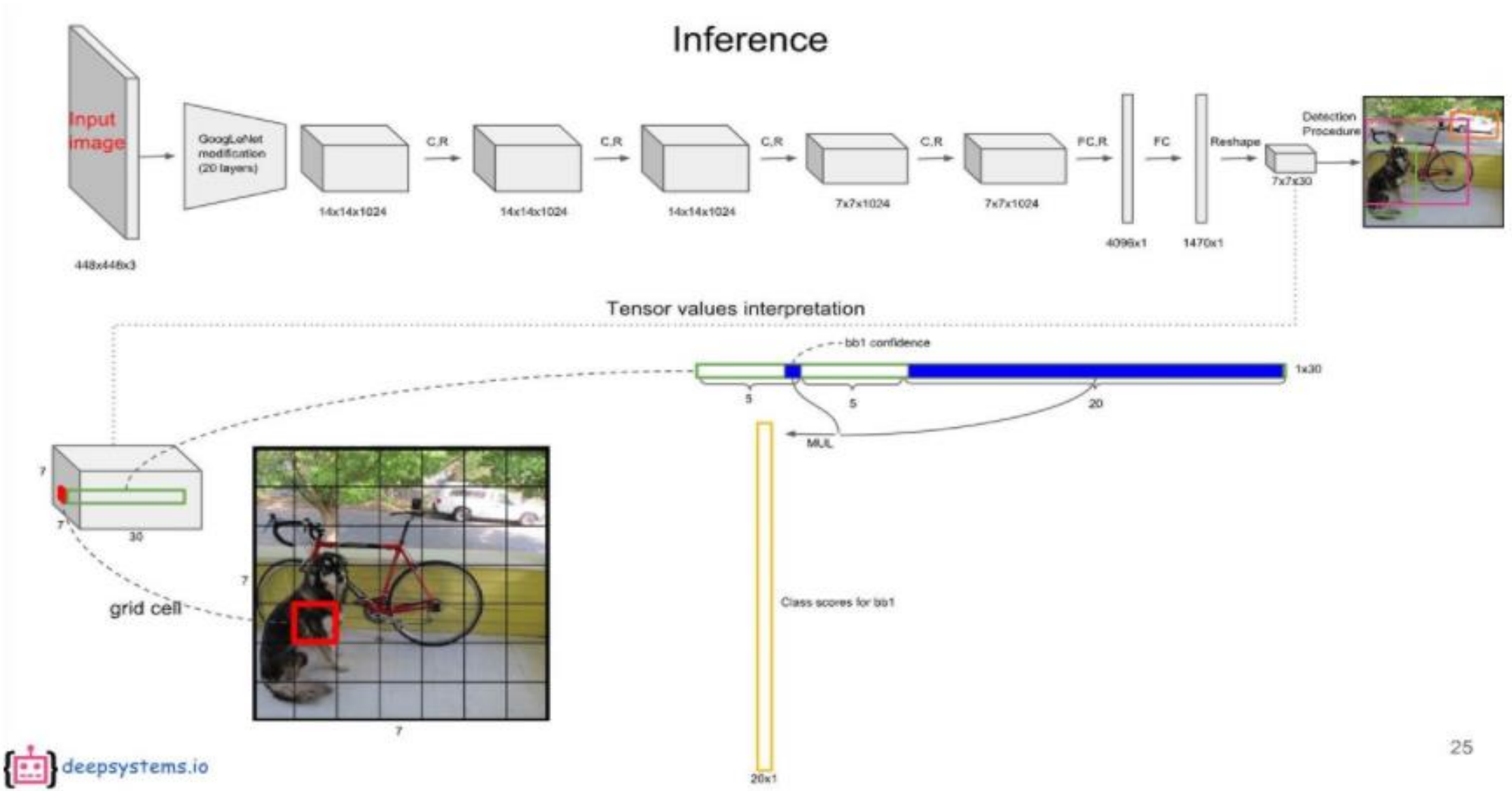


각 cell 마다 2개의 class confidence score 가짐

→ 7 x 7 x 2개의 cell은 모두 class-specific confidence score 가짐



Appendix2 : Introduction – Precedent Methods

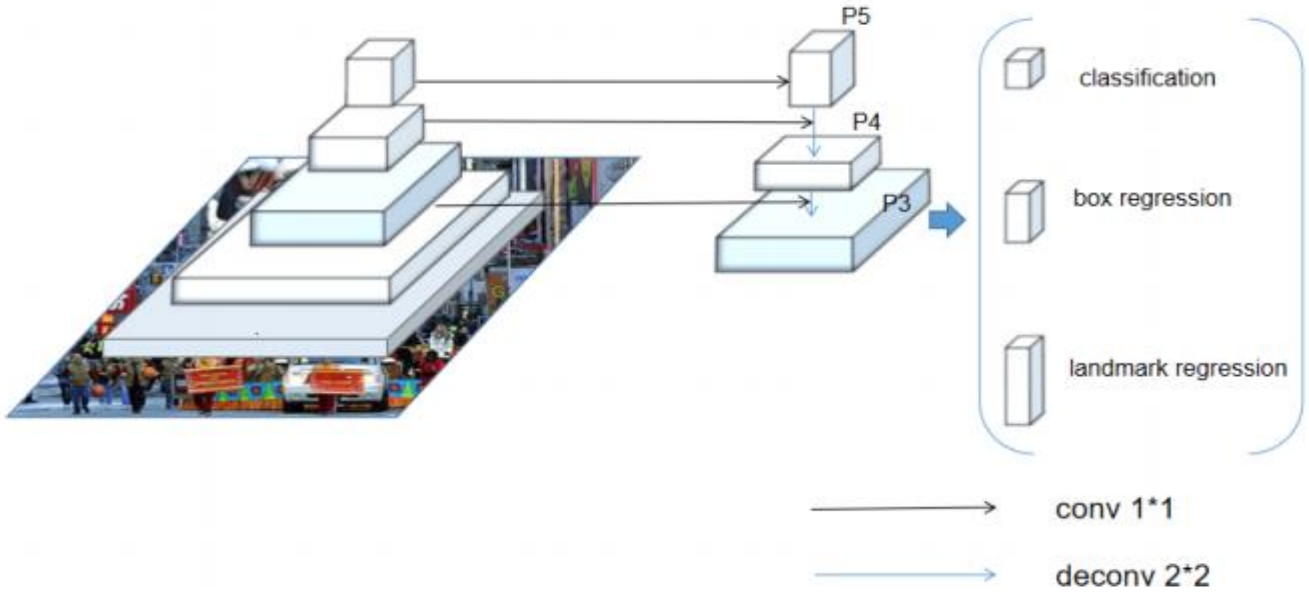


Introduction – Anchor Free Object Detection Framework

Center Face : The Face as Point Design

- Light and powerful, simpler and more effective
- 얼굴의 bounding box의 **center point** 이용 (peak in the heat map)

  → "facial box size and landmark are regressed directly to image features at the center location"
- 얼굴에 대한 object detection문제가 **standard key point**에 대한 추정 문제로 변환
- 각 **center point**에서의 image feature가 얼굴과 landmark를 예측



- Center point에 대한 추정을 통해 face box, key points 예측
- Feature pyramid network 제공
- 고성능 고효율

Related Works

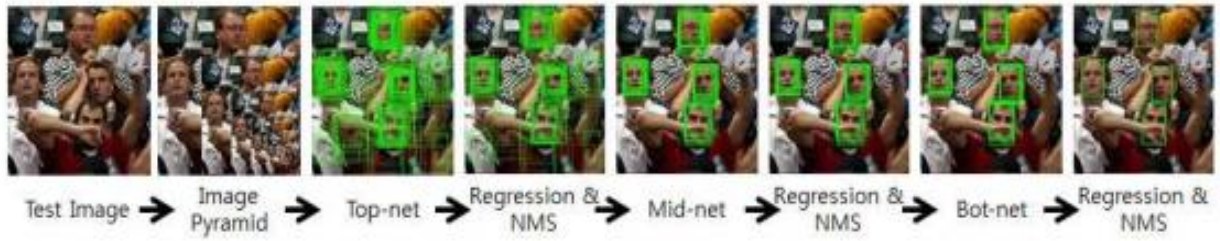
1. Cascaded CNN

- 입력 이미지를 다양한 크기로 변형하여 **Image Pyramid** 생성 (Cascaded CNN의 최초 입력)

① **Top Net** : 얼굴 후보군 필터링 (이후 단계의 연산 감소), Bounding Box와 해당 Box의 confidence, regression 정보 획득

② **Mid Net** : Top Net의 결과를 입력으로 받아 더욱 정교하게 얼굴을 검출

③ **Bot Net** : 최종 얼굴 검출



출처 :
https://www.dbpia.co.kr/pdf/pdfView.do?nodeId=NODE07295573&mark=0&useDate=&bookmarkCnt=0&ipRange=N&accessgl=Y&language=ko_KR

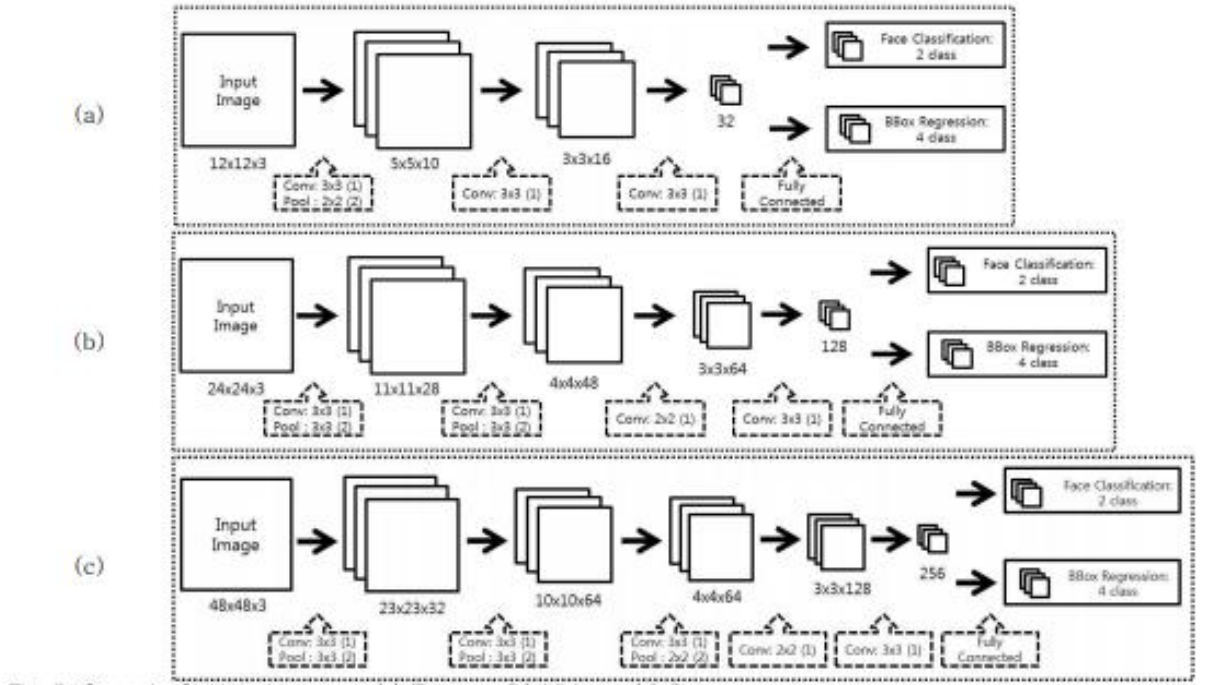


Fig. 5. Cascade CNN Architecture : (a) Top-net, (b) Mid-net, (c) Bot-net

Related Works

2. Anchor Methods

- 딥러닝의 성과들을 수용하며 많은 발전을 이룸
- Feature pyramid에서 얼굴의 위치와 크기를 dense sampling (single stage design)

3. Anchor Free Methods

- Cascaded CNN이 이 방식에 해당함 but 한계가 많은 window sliding방식 사용, image pyramid에 의존
- LFFD같이 anchor를 사용하는 방식은 시간이 오래 소요
- Center Face는 얼굴을 bounding box의 single point로 표현하고, 이에 대해 facial box size와 landmark가 회귀되기 때문에 상대적으로 적은 시간이 소요
- Center Face simply represents faces by a single point at their bounding box center, then facial box size and landmark are regressed directly from image features at the center location

Center Face

1. Mobile Feature Pyramid Network : Mobilenetv2 + FPN

- 이미지 내에 존재하는 다양한 크기의 객체를 인식 : Object Detection의 핵심적인 문제
- 다양한 크기의 물체를 탐지하기 위해 이미지 자체의 크기를 resize - 비효율적

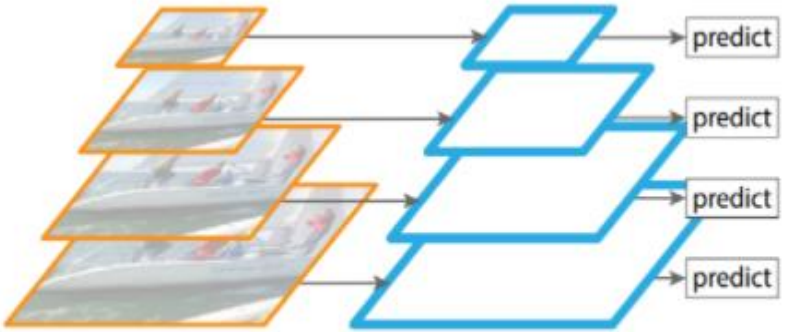
  → **Issue of Scale-Invariant**
- 컴퓨팅 자원을 적게 차지하면서 **다양한 크기의 객체를 인식**하는 방법

기존 방식 1 : Featurized Image Pyramid

- 각 레벨에서 독립적으로 특징을 추출 → 객체 탐지
- 입력 이미지의 크기를 **resize**하여 **다양한 scale**의 이미지를 네트워크에 입력

  (resize된 개별 이미지에서 물체를 탐지)
- 다양한 크기의 객체를 포착 but 추론 속도 느림

  (이미지 한 장을 독립적으로 모델에 입력하여 feature map 생성



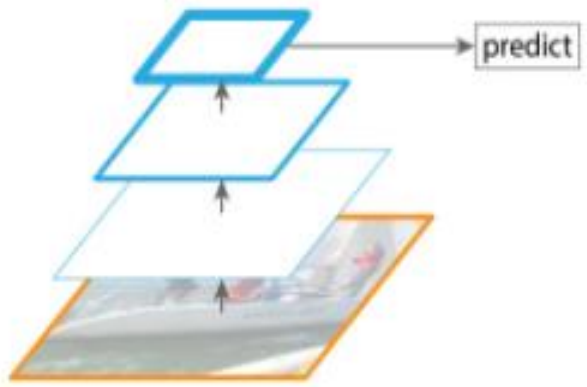
(a) Featurized image pyramid

- https://yeomko.tistory.com/44 / https://herbwood.tistory.com/18 / https://eehoeskrap.tistory.com/300

Center Face

기존 방식 2 : Single Feature Map

- **단일 scale**의 입력 이미지를 네트워크에 입력 → 단일 scale의 feature map을 통해 객체 탐지
- CNN을 통과하여 얻은 최종 단계의 feature map으로 객체 검출
- Convolution Layer를 통해 특징을 압축
- Multi scale을 사용하지 않고 한번에 특징을 압축하여, 마지막에 압축된 특징만을 사용 → 추론 속도가 빠르지만 성능이 떨어짐
- YOLO v1

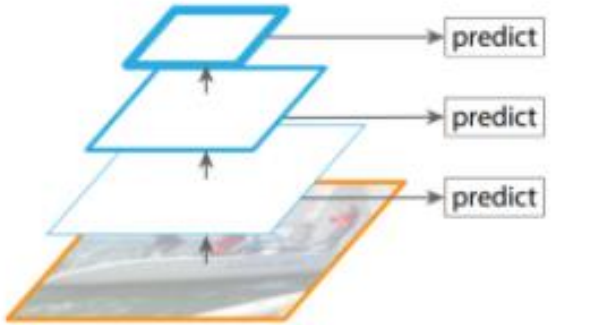


(b) Single feature map

Center Face

기존 방식 3 : Pyramidal Feature Hierarchy

- 네트워크에서 미리 지정한 Convolution Layer마다 feature map을 추출하여 객체 탐지 (SSD)
- CNN을 통과하는 중간 과정에서 생성되는 feature map들 각각에 객체 검출 시행
- **Multi Scale** Feature Map 사용 → 높은 성능
- Feature map 간 해상도 차이 → semantic gap
- **저수준 특징의 학습**이 때때로 객체 인식률을 낮춤



(c) Pyramidal feature hierarchy

* **SSD** : 전체 conv net 중간 지점 부터 feature map 추출

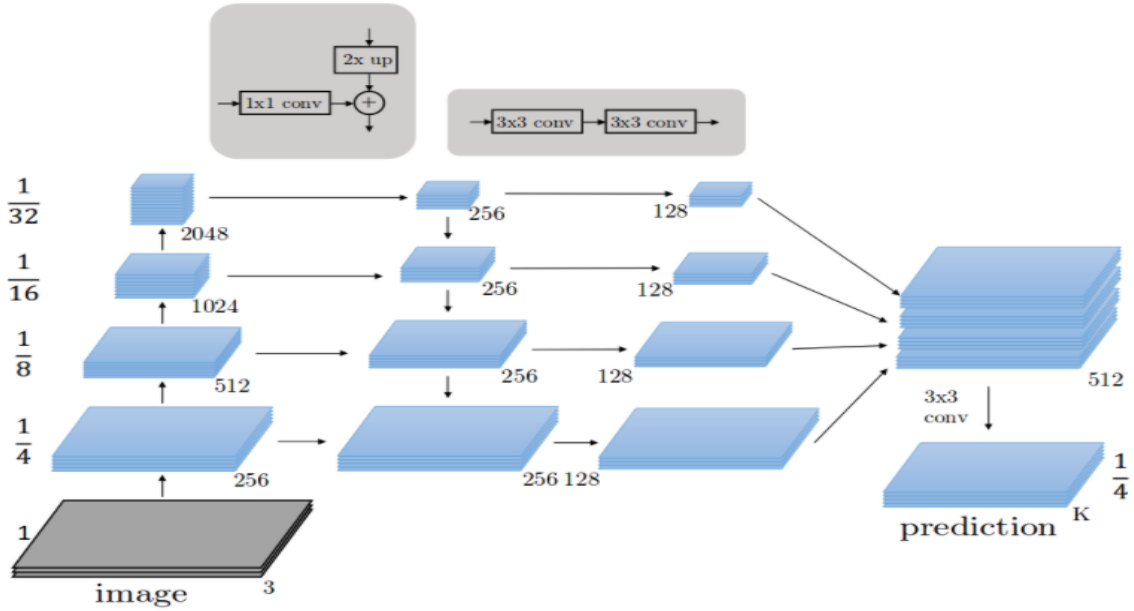
↔ **FPN** : 높은 해상도의 feature map은 작은 객체를 탐지할 때 유용하지 않음

Center Face

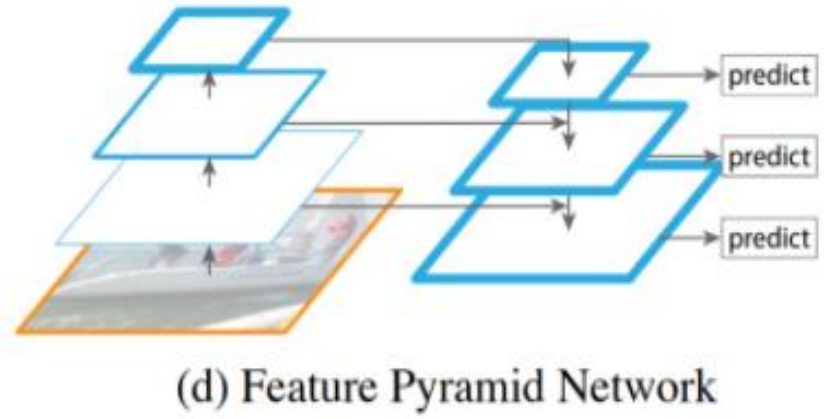
Feature Pyramid Network

- **임의의 크기의 single scale 이미지를 CNN에 입력하여 다양한 scale의 feature map출력**

  → CNN에서 지정한 layer별로 feature map을 추출하여 수정하는 네트워크
- Top-down 방식으로 feature 추출
- 각 레벨에서 독립적으로 특징 추출, **상위 레벨의 이미 계산된 특징을 재사용**



Feature Pyramid Network



(d) Feature Pyramid Network

Center Face

Feature Pyramid Network - summary

1. CNN을 통과하며 단계 별로 feature map 생성
2. 가장 상위 layer에서 거꾸로 내려오며 feature 통합

→ 상위 layer의 추상화된 정보와 하위 layer의 작은 물체들에 대한 정보 통합

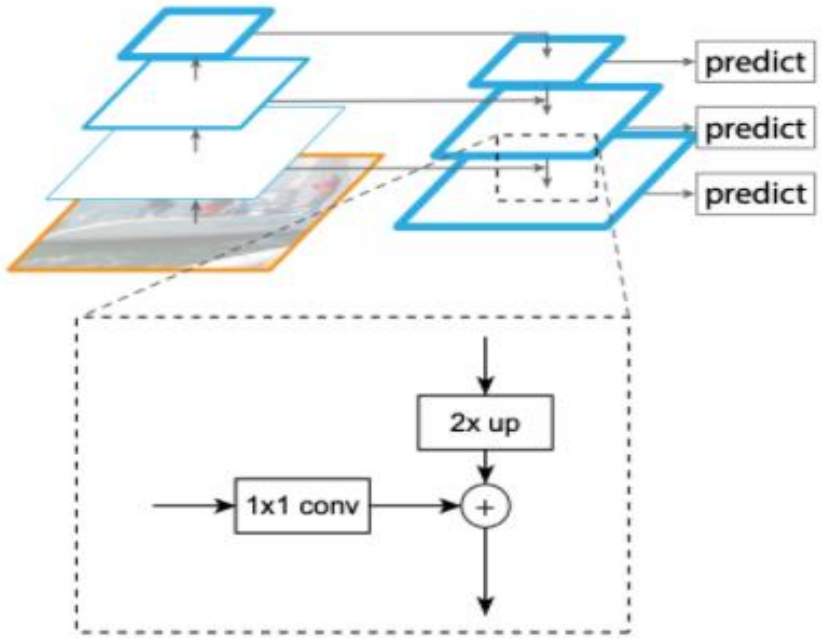
- CNN 자체가 layer를 거치면서 피라미드를 생성
- Forward를 거치면서 더 많은 의미(Semantic)를 가지게 됨
- 각 layer마다 예측 과정을 넣어서 scale 변화에 강건한 모델

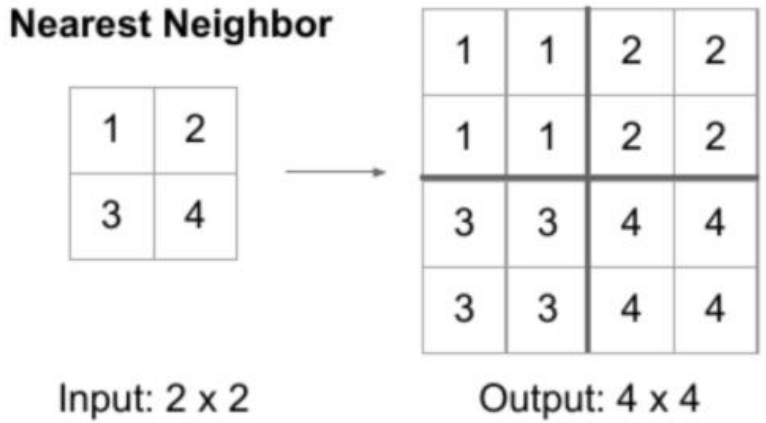
Center Face

Feature Pyramid Network – Feature Fusion

- 상위 feature map, 하위 feature map을 통합하는 과정
- Feature map이 layer를 통과하며 해상도가 2배 씩 작아진다고 가정

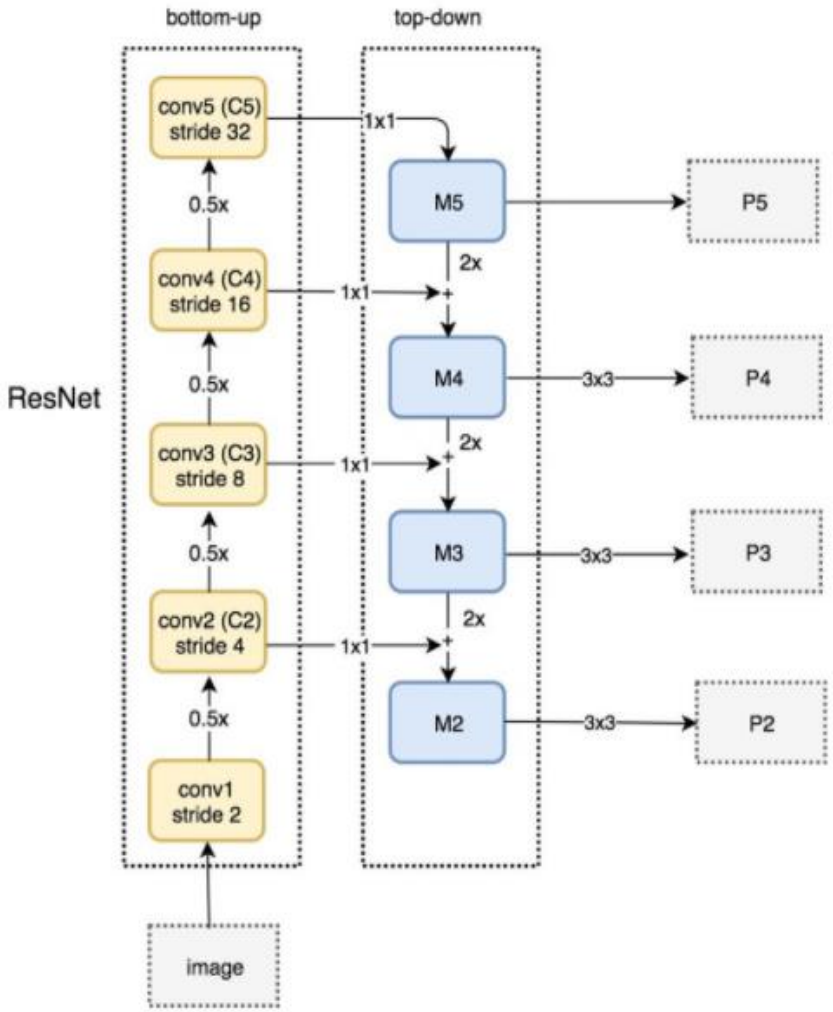
→ 상위 feature map의 해상도를 키워주는 과정 필요

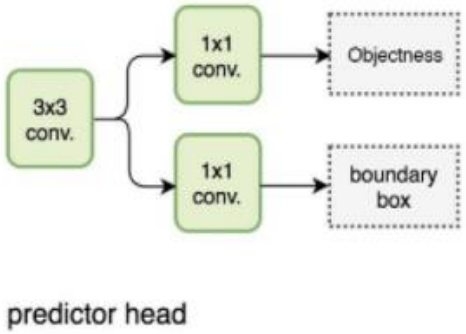




Center Face

Feature Pyramid Network – example





predictor head

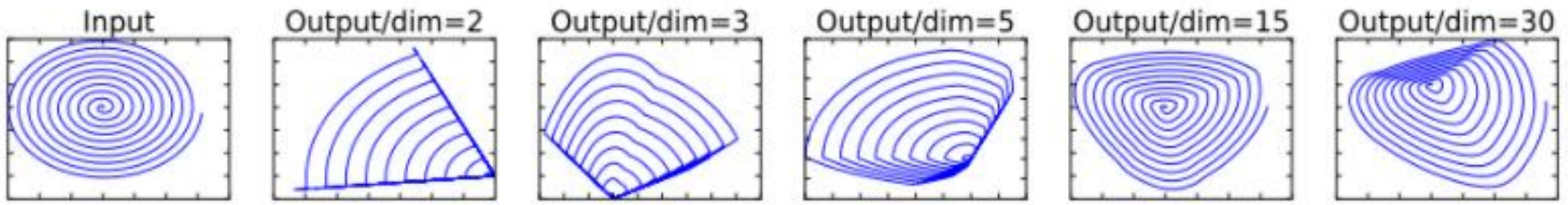
1. ResNet을 통과하며 중간 feature map들이 생성
2. 상위 layer부터 내려오면서 **feature들을 통합** (P5, P4, …)

- 상위 feature map들의 합으로 생성된 feature map (eg P5) : 크키가 큰 물체에 대한 정보 – 큰 anchor box

Center Face

MobileNetV2 : Inverted Residuals and Linear Bottlenecks

- ReLU 함수를 거치며 발생하는 정보 손실 최소화
- 출력 채널이 작을 수록 정보 손실 발생 – 채널 수가 적은 layer에 비선형 함수 적용



https://minimin2.tistory.com/43
https://deep-learning-study.tistory.com/541
https://gaussian37.github.io/dl-concept-mobilenet_v2/

Center Face

2. Face as Point

$[x_1, y_1, x_2, y_2]$ → 얼굴의 bounding box $c = [(x_1 + x_2)/2, (y_1 + y_2)/2]$ → 얼굴의 center point

$I \in R^{W \times H \times 3}$ → W x H 입력 이미지 (칼라) $Y \in [0, 1]^{W/R \times H/R}$ → output (**heatmap**) * 얼굴이 존재할 확률

$\hat{Y}_{x,y} = 1$ → face center $\hat{Y}_{x,y} = 0$ → back ground

$$L_c = \begin{cases} -(1-\hat{Y}_{xyc})^{\alpha} \log(\hat{Y}_{xyc}) & if\ Y_{xyc} = 1 \\ -(1-Y_{xyc})^{\beta} (\hat{Y}_{xyc})^{\alpha} \log(1-\hat{Y}_{xyc}) & otherwise \end{cases}$$

Training loss (variant of focal loss)

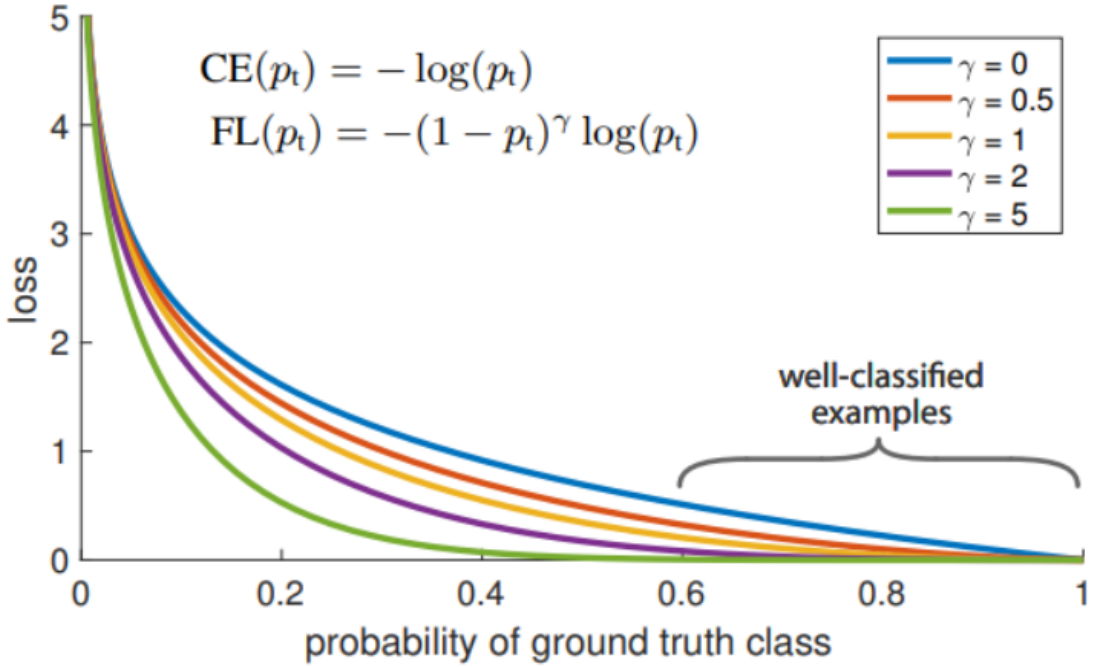
Center Face

Focal Loss

- RetinalNet에서 설계한 새로운 loss function
- 잘 찾은 class에 대해서는 loss를 적게, 잘 찾지 못한 class에 대해서는 큰 loss 부여

  → 극단적인 class imbalance 문제 해결

$$\mathrm{FL}(p_t) = -(1-p_t)^{\gamma}\log(p_t).$$



- **Cross Entropy의 단점** : box에 물체가 존재할 확률이 0.5를 넘어가도 loss값이 꽤 있다는 점 때문에, easy example이 많이 존재하는 경우 나쁜 방향으로 학습이 될 수 있음

Center Face

- **Mapping** - Image : (x, y) → heatmap : (x/n, y/n) n : down sampling factor
- Heatmap에서 다시 input image로 **remapping**시 어떤 픽셀들은 제대로 정렬되지 않을 수 있음 → 성능 저하

$$o_k = \left(\frac{x_k}{n} - \left\lfloor \frac{x_k}{n} \right\rfloor, \frac{y_k}{n} - \left\lfloor \frac{y_k}{n} \right\rfloor\right)$$

(x, y) : face center k / o : offset

→ 추정한 center position을 다시 input에 **remapping**하기 전에 좌표 조정

Center Face

2. Box and Landmark Prediction

$G = (x1, y1, x2, y2)$. → Ground-truth bounding box

$(\hat{h}, \hat{w})$ → (x, y) **Our goal** : map networks position outputs to center position in the feature map

$$\hat{h} = \log(\frac{x_2}{R} - \frac{x_1}{R})$$

$$\hat{w} = \log(\frac{y_2}{R} - \frac{y_1}{R})$$

Application of Deface

Hyper-parameter : ths

- How confident the detector needs to be for classifying region as face : **detection threshold**
- **trade off between F.P & F.N**

FP : 얼굴이 아닌 대상을 얼굴로 검출하여 blur 처리

FN : 얼굴인 대상을 얼굴이 아니라고 판단하여 blur 처리 x

1. **낮은 ths** : 얼굴에 대한 확신의 정도가 낮아도 얼굴로 검출 → 얼굴이라고 판단하는 근거가 적어도 됨
2. **높은 ths** : 얼굴에 대한 확신의 정도가 매우 높아야 얼굴로 검출

→ 낮은 ths 설정에서는 얼굴로 판별된 대상이, 해당 ths가 요구하는 확신의 정도를 충족시키지 못해 얼굴로 판별이 안됨

`--thresh 0.02` (notice the false positives, e.g. at hand regions)

`--thresh 0.7` (notice the false negatives, especially at partially occluded faces)

빨간색 : ths가 낮은 경우, 손을 얼굴로 검출하여(FP) blur처리

파란색 : ths가 높은 경우, 겹침 등의 요소로 인한 확신성의 부족으로 얼굴을 얼굴로 검출하지 못함(FN)