[U-Net: Convolutional Networks for Biomedical Image Segmentation / Olaf Ronneberger, Philipp Fischer, and Thomas Brox]

U-Net은 Fully Convolution Network(FCN)를 기반으로 하여 구축하였으며, 적은 데이터를 가지고도 더욱 정확한 Segmentaion을 내기 위해 FCN 구조를 수정

(1) Introduction

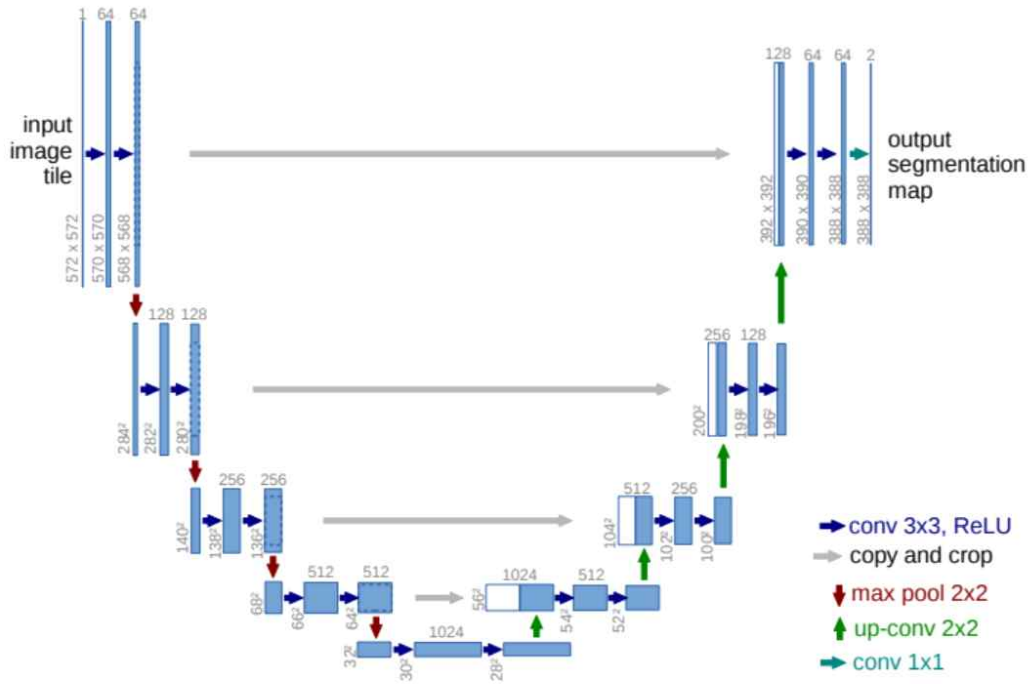
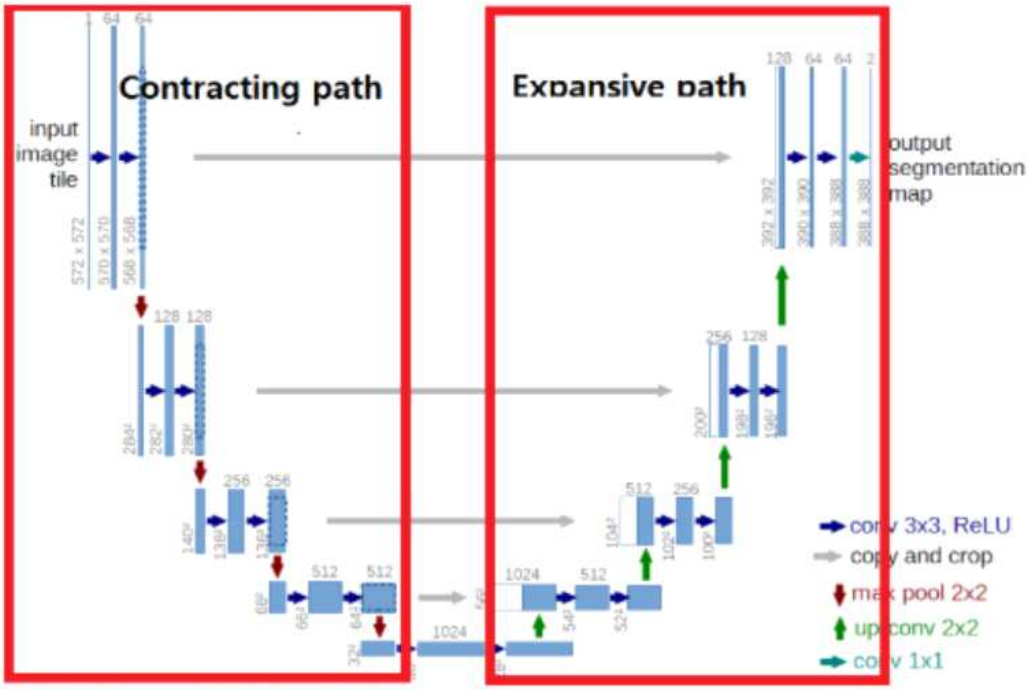


Fig. 1. U-net architecture (example for 32x32 pixels in the lowest resolution). Each blue box corresponds to a multi-channel feature map. The number of channels is denoted on top of the box. The x-y-size is provided at the lower left edge of the box. White boxes represent copied feature maps. The arrows denote the different operations.

- 왼쪽 Contracting Path (Encoding), 오른쪽 Expansive Path (Decoding)로 정의



1) Contracting Path : 이미지의 context 포착

2) Expansive Path :
feature map을 upsampling 하고 1)에서 포착한 feature map의 context와 결합
→ 더욱 정확한 localization을 하는 역할

- U-Net은 적은 데이터로 충분한 학습을 하기 위해 Data Augmentation을 사용

1) Elastic Deformation

적은 수의 이미지를 가지고 효과적으로 학습하기 위해 이 방식을 사용
(티슈 조직 등의 실질적인 변형과 유사)

2) Weighted Cross Entropy + Data Augmentation

많은 cell을 segmentation 해야 하는 문제에서의 도전 과제는 같은 클래스가 서로 인접해 있는 케이스

(2) Architecture

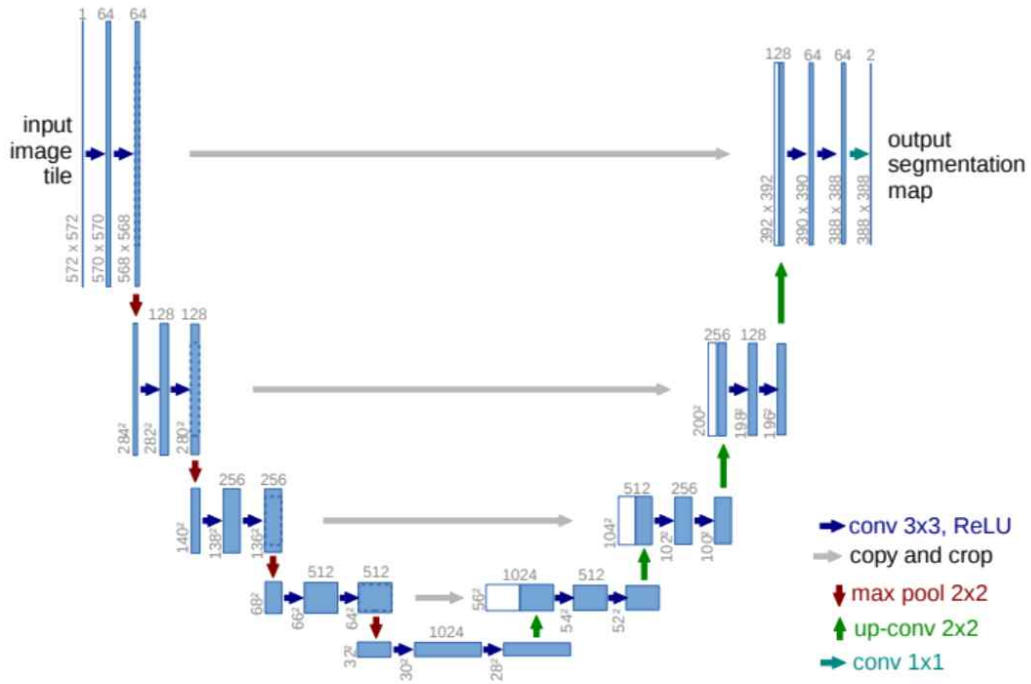


Fig. 1. U-net architecture (example for 32x32 pixels in the lowest resolution). Each blue box corresponds to a multi-channel feature map. The number of channels is denoted on top of the box. The x-y-size is provided at the lower left edge of the box. White boxes represent copied feature maps. The arrows denote the different operations.

1) Contracting Path

Contracting Path는 일반적인 CNN, Downsampling을 위한 Stride2, 2x2 max pooling 연산과 ReLU를 포함한 두 번의 반복된 3x3 unpadded convolutions 연산

3x3conv → ReLU → 2x2 max pooling → 3x3conv → ReLU → 2x2 max pooling

downsampling 과정에서 feature map channel의 수는 2배로 증가

2) Expansive Path

Expansive Path는 2x2 convolutions를 통해 upsampled된 feature map과 1)의 cropped feature map과의 concatenation 후에 ReLU 연산을 포함한 두 번의 3x3 convolutions 연산을 거침

이 때, 1)의 Crop된 featrue map을 보내는 이유는 Convolution을 하면서 border pixel에 대한 정보가 날아가는 것에 대한 양쪽의 보정을 위함이다.

마지막 layer는 1x1 convolution 연산 : 64개의 component feature vector를 desired number of classes에 mapping 하기 위해서

(3) Training

input image와 상응하는 segmentation map은 SGD와 함께 network를 학습하기 위해 사용
convolution 연산이 padding을 거치지 않기 때문에 input image보다 output image의 크기가 더 작아짐.

Overhead와 GPU memory의 사용을 극대화하기 위하여 큰 batch size보다는 큰 input tiles를 선호 -> 높은 momentum(0.99)를 사용해서 과거에 진행했던 training sample이 현재의 update에 더욱 관여하도록 함.

Energy function은 최종적인 feature map과 corss entropy loss function, pixel wise soft-max를 결합하여 계산.

각각의 Ground truth segmentation의 weight map을 서로 다른 frequency를 대처하고 network에게 작은 separation borders를 학습 시키기 위해 사전에 weight map을 계산.

separation border는 morphological operation을 통해 계산되어지며, weight map이 계산되어지는 식 :

$$w(x) = w_c(x) + w_0 \cdot e^{-\frac{(d_1(x)+d_2(x))^2}{2\sigma^2}}$$

where $w_c: \Omega \rightarrow \mathbb{R}$ is the weight map to balance the class frequencies

$d_1: \Omega \rightarrow \mathbb{R}$ denotes the distance to the border of the nearest cell

$d_2: \Omega \rightarrow \mathbb{R}$ denotes the distance to the border of the second nearest cell

w0를 10으로 표준편차(σ)를 약 5pixel로 설정.

deep network에서 초기 weight에 대한 good initialization의 중요성에 대해 언급

이상적으로는 초기 weights들은 각각의 feature map이 대략적인 unit variance를 가지도록 설정되어야 한다고 함

U-Net은 표준편차를 가지는 가우시안 분포로부터 초기 weights를 도출해낼 수 있다고 함

$$\sigma = \sqrt{\frac{2}{N}}$$

where N denotes the number of imcoming nodes of one neuron

만약 이전 layer에서 3x3 convolution 연산과 feature channel의 개수가 64개라면 N은 3x3x64 = 576으로 계산

[Focal Loss for Dense Object Detection / Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, Piotr Dollár]

모델이 예측하기 어려운 hard example에 집중하도록 하는 Focal Loss를 제안
ResNet과 FPN을 활용하여 구축된 one-stage 모델인 RetinaNet은 focal loss를 사용하여 two-stage 모델 Faster R-CNN의 정확도를 능가

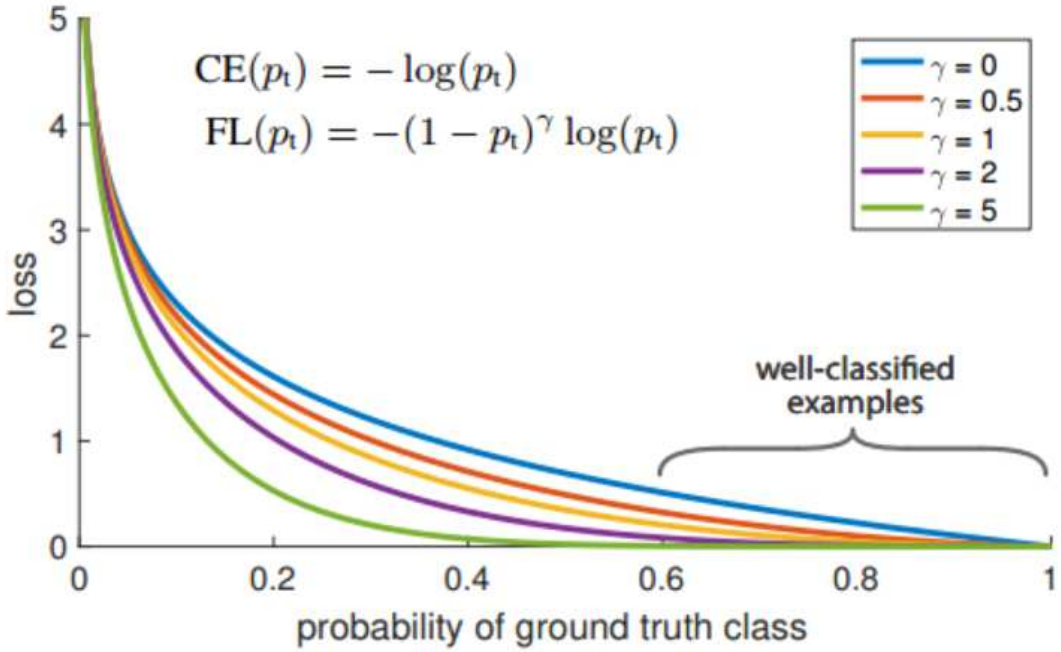


Figure 1. We propose a novel loss we term the *Focal Loss* that adds a factor $(1 - p_t)^\gamma$ to the standard cross entropy criterion. Setting $\gamma > 0$ reduces the relative loss for well-classified examples ($p_t > .5$), putting more focus on hard, misclassified examples. As our experiments will demonstrate, the proposed focal loss enables training highly accurate dense object detectors in the presence of vast numbers of easy background examples.

Foscal loss(초록색 선)와 기존 loss function(파랑색 선)과의 차이점입니다. 정답일 확률이 높은 예측에는 0에 가까운 loss가 부여

Cross entropy Loss

Focal Loss는 one-stage detector에서 클래스 불균형 문제를 해결하기 위해 제안된 loss function. classification에서 사용하는 cross-entropy loss에 인자를 하나 추가한 것

$$\mathrm{CE}(p, y) = \begin{cases} -\log(p) & \text{if } y = 1 \\ -\log(1-p) & \text{otherwise.} \end{cases}$$

이진 분류문제에서 사용하는 CE loss function입니다. y는 ground-truth class이고, p는 모델이 예측한 값

$$p_t = \begin{cases} p & \text{if } y = 1 \\ 1-p & \text{otherwise,} \end{cases}$$

$$\mathrm{CE}(p, y) = \mathrm{CE}(p_t) = -\log(p_t)$$

pt가 0.5이상이면 분류하기 쉬운 easy example, easy example은 loss값이 작지만, 엄청나게 많아지면 대부분의 loss를 차지하게 되어 hard example의 영향을 감소시킴. CE loss는 클래스 불균형이 존재할 때, 좋은 선택 x.

Focal Loss

cross entropy loss에 인자를 하나 추가
이 인자는 modulating factor
easy example의 영향을 감소시키고 hard example에 집중
앞에 인자가 추가함으로써 easy example이 loss에 미치는 영향을 감소

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

γ는 하이퍼파라미터. 감마값으로 easy/hard example의 가중치를 조절할 수 있습니다. γ=2 일때, 가장 성능이 좋음

example이 잘못 분류됐으면 pt는 낮은 값을 갖습니다. pt가 낮으면 modulating factor은 1에 가까운 값을 갖게 됩니다. 따라서 loss는 가중치에 영향을 받지않습니다.

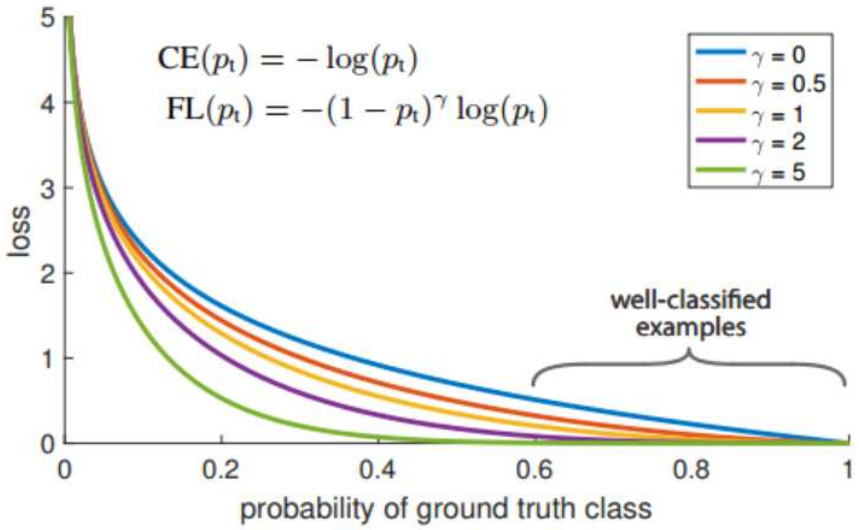


Figure 1. We propose a novel loss we term the *Focal Loss* that adds a factor $(1 - p_t)^{\gamma}$ to the standard cross entropy criterion. Setting $\gamma > 0$ reduces the relative loss for well-classified examples ($p_t > .5$), putting more focus on hard, misclassified examples. As our experiments will demonstrate, the proposed focal loss enables training highly accurate dense object detectors in the presence of vast numbers of easy background examples.

RetinaNet

RetinaNet은 ResNet-FPN을 backnone으로 하여 2개의 sub-network를 사용하는 신경망. 첫 번째 sub-network는 object classification을 수행하고, 두 번째 sub-network는 bounding box regression을 수행

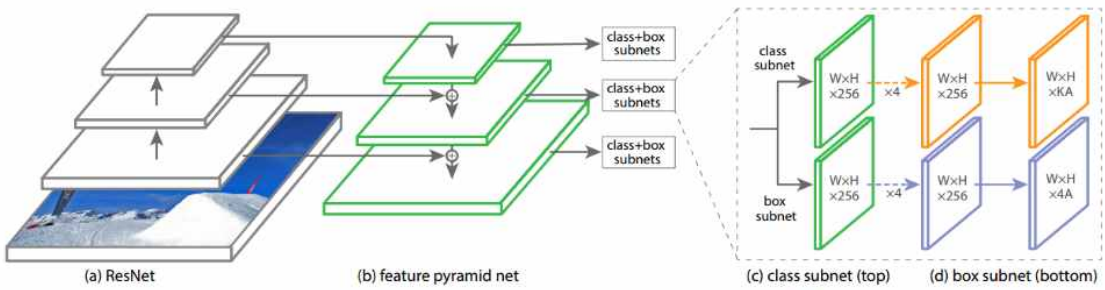


Figure 3. The one-stage **RetinaNet** network architecture uses a Feature Pyramid Network (FPN) [20] backbone on top of a feedforward ResNet architecture [16] (a) to generate a rich, multi-scale convolutional feature pyramid (b). To this backbone RetinaNet attaches two subnetworks, one for classifying anchor boxes (c) and one for regressing from anchor boxes to ground-truth object boxes (d). The network design is intentionally simple, which enables this work to focus on a novel focal loss function that eliminates the accuracy gap between our one-stage detector and state-of-the-art two-stage detectors like Faster R-CNN with FPN [20] while running at faster speeds.

1. Feature Pyramid Network

ResNet구조에 FPN을 backbone으로 사용. FPN은 top-down pathway와 lateral connection을 사용하여 multi-scale feature pyramid를 생성. pyramid의 각 level에서 다양한 크기의 객체를 검출. P3부터 P7의 pyramid를 사용하고, 각 pyramid level은 256 채널.

속도를 높이기 위해 각 pyramid level에서 1000개의 top-scoring perdiction을 가진 box를 사용. 모든 level에서 box가 병합되고 NMS를 수행하여 sub-network로 전달.

2. Anchors

Anchor를 사용합니다. 각 pyramid level에 aspect ratio={1:2, 1:1, 2:1}, size={ 2^0, 2^(1/3), 2^(2/3)} 을 사용하여 총 9개의 anchor를 할당
각 Anchor은 one-hot K vertor(K개의 class중 해당하는 class는 1, 나머지는 0)와 바운딩박스 offset 4개를 할당
따라서 하나의 Anchor에는 K*4의 vector가 할당

Anchor은 IOU가 0.5이상인 ground-thuth에 할당
IoU가 0~0.4이면 배경으로 할당합니다. 0.4~0.5 IOU를 가진 anchor은 무시

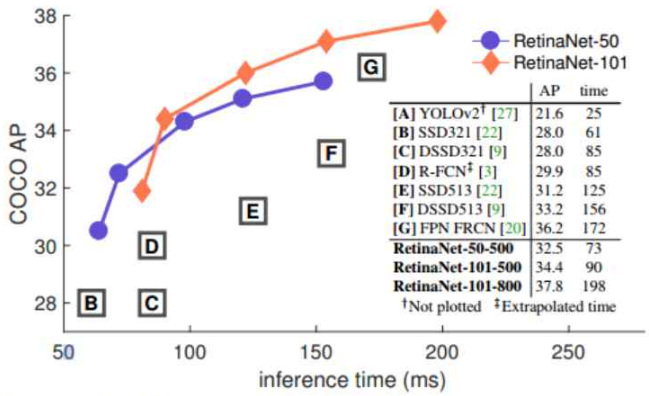
3. Classification Subnet

Anchor의 object class를 예측하는 network. 각 pyramid level에 KA개 filter를 지닌 3x3 conv layer가 4개로 구성된 Conv layer를 부착. K는 class 수, A는 anchor 수. 그리고 classification subnet의 출력값에 Focal Loss를 적용.

4. Box Regression Subnet

anchor와 ground-truth의 offset을 계산하는 network. Classification Subnet과 동일하지만, 마지막에 4A 길이를 출력. 각 anchor마다 offset 4개의 값을 출력

sota 모델과 비교한 성능



	AP	time
[A] YOLOv2† [27]	21.6	25
[B] SSD321 [22]	28.0	61
[C] DSSD321 [9]	28.0	85
[D] R-FCN‡ [3]	29.9	85
[E] SSD513 [22]	31.2	125
[F] DSSD513 [9]	33.2	156
[G] FPN FRCN [20]	36.2	172
RetinaNet-50-500	32.5	73
RetinaNet-101-500	34.4	90
RetinaNet-101-800	37.8	198

†Not plotted ‡Extrapolated time

Figure 2. Speed (ms) versus accuracy (AP) on COCO `test-dev`. Enabled by the focal loss, our simple one-stage *RetinaNet* detector outperforms all previous one-stage and two-stage detectors, including the best reported Faster R-CNN [28] system from [20]. We show variants of RetinaNet with ResNet-50-FPN (blue circles) and ResNet-101-FPN (orange diamonds) at five scales (400-800 pixels). Ignoring the low-accuracy regime (AP<25), RetinaNet forms an upper envelope of all current detectors, and an improved variant (not shown) achieves 40.8 AP. Details are given in §5.

	backbone	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Two-stage methods							
Faster R-CNN+++ [16]	ResNet-101-C4	34.9	55.7	37.4	15.6	38.7	50.9
Faster R-CNN w FPN [20]	ResNet-101-FPN	36.2	59.1	39.0	18.2	39.0	48.2
Faster R-CNN by G-RMI [17]	Inception-ResNet-v2 [34]	34.7	55.5	36.7	13.5	38.1	52.0
Faster R-CNN w TDM [32]	Inception-ResNet-v2-TDM	36.8	57.7	39.2	16.2	39.8	**52.1**
One-stage methods							
YOLOv2 [27]	DarkNet-19 [27]	21.6	44.0	19.2	5.0	22.4	35.5
SSD513 [22, 9]	ResNet-101-SSD	31.2	50.4	33.3	10.2	34.5	49.8
DSSD513 [9]	ResNet-101-DSSD	33.2	53.3	35.2	13.0	35.4	51.1
RetinaNet (ours)	ResNet-101-FPN	39.1	59.1	42.3	21.8	42.7	50.2
RetinaNet (ours)	ResNeXt-101-FPN	**40.8**	**61.1**	**44.1**	**24.1**	**44.2**	51.2

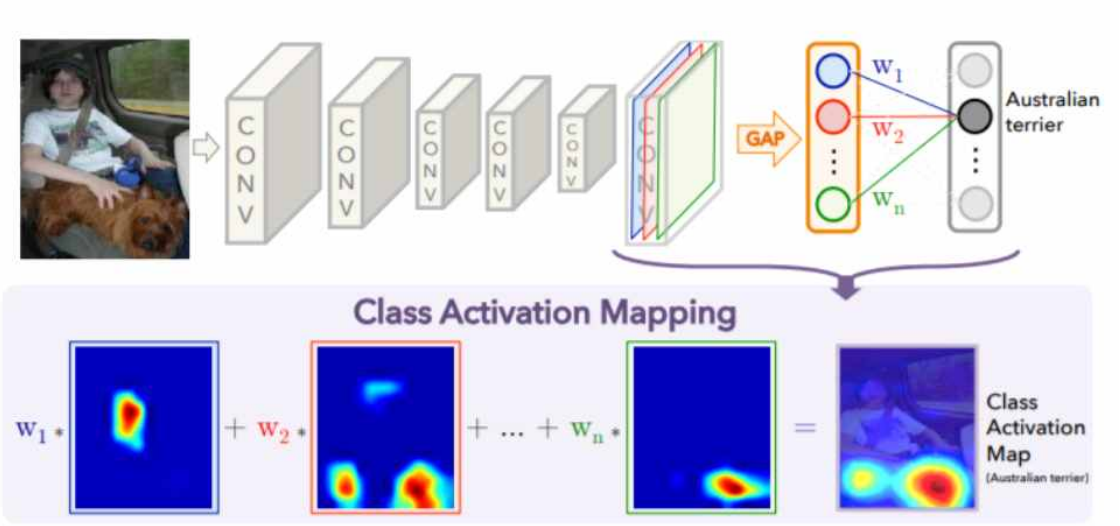
Table 2. **Object detection** *single-model* results (bounding box AP), *vs.* state-of-the-art on COCO `test-dev`. We show results for our RetinaNet-101-800 model, trained with scale jitter and for 1.5× longer than the same model from Table 1e. Our model achieves top results, outperforming both one-stage and two-stage models. For a detailed breakdown of speed versus accuracy see Table 1e and Figure 2.

[learning deep features for discriminative localization / Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, Antonio Torralba]

Class Activation Maps (CAM)

논문의 가장 중요한 핵심은 CAM
Convolution층 바로 다음 Global Average Pooling(GAP)을 붙이고 softmax를 붙이는 모델 구조를 만든다.

binary classification에서 Convolution층의 마지막 출력이 (width,height,channel) = (7,7,128)인 activation maps 가 있다고 하면 각 Channel 별로 모두 Global Average Pooling을 해주어 128개의 값이 나오면 softmax로 연결해주는 것이다. 위와 같이 [0,1]가 정답이라면 1 로가는 weights를 각 global average pooling을 한 마지막 activation maps에 곱해준다. 이렇게 하면 각 activation map과 weights를 곱하여 어디를 보고 판단을 했는지를 볼 수 있게 된다. 아래 그림은 이런 프로세스를 설명한 논문의 그림



Australian terrier를 분류하므로 사람의 얼굴에 집중한 첫 번째 activation map의 W_{1}은 낮은 값일 것이고 개의 특징에 주목한 두 번째, n 번째 activation map에 연결된 W_{2},W_{n}은 높은 값을 가질 것으로 유추

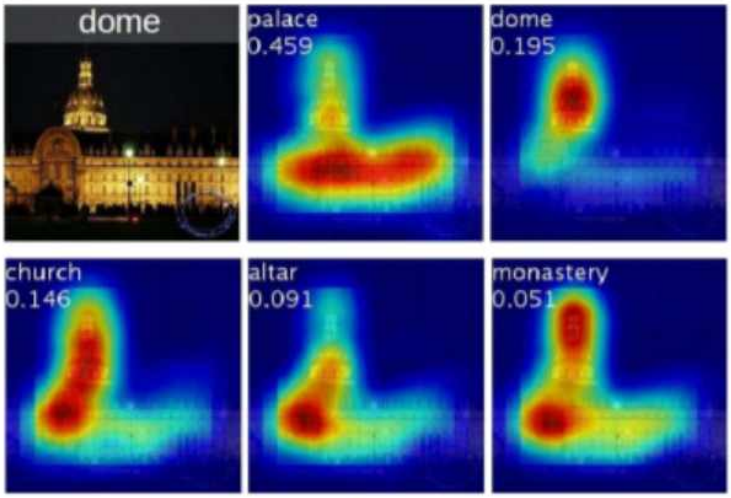


Figure 4. Examples of the CAMs generated from the top 5 predicted categories for the given image with ground-truth as dome. The predicted class and its score are shown above each class activation map. We observe that the highlighted regions vary across predicted classes e.g., *dome* activates the upper round part while *palace* activates the lower flat part of the compound.

각 class별로 같은 이미지를 인식할 때 다른 곳을 보았다는 것.
사람도 어떤 물체가 무엇인지 판단할 때 각 물체별 특성을 보고 판단하는 것과 같은 프로세스.
정답은 dome이지만 palace의 확률이 가장 높게 나왔는데 왜 그렇게 판단했는지를 알 수 있는 실험

Weakly-supervised Object Localization

Table 1. Classification error on the ILSVRC validation set.

Networks	top-1 val. error	top-5 val. error
VGGnet-GAP	33.4	12.2
GoogLeNet-GAP	35.0	13.2
AlexNet*-GAP	44.9	20.9
AlexNet-GAP	51.1	26.3
GoogLeNet	31.9	11.3
VGGnet	31.2	11.4
AlexNet	42.6	19.5
NIN	41.9	19.6
GoogLeNet-GMP	35.6	13.9

Table 2. Localization error on the ILSVRC validation set. *Back-prop* refers to using [23] for localization instead of CAM.

Method	top-1 val.error	top-5 val. error
GoogLeNet-GAP	**56.40**	**43.00**
VGGnet-GAP	57.20	45.14
GoogLeNet	60.09	49.34
AlexNet*-GAP	63.75	49.53
AlexNet-GAP	67.19	52.16
NIN	65.47	54.19
Backprop on GoogLeNet	61.31	50.55
Backprop on VGGnet	61.12	51.46
Backprop on AlexNet	65.17	52.64
GoogLeNet-GMP	57.78	45.26

CAM 방법으로 classification을 학습시킨 모델은 GAP를 사용하니 성능이 조금 저하되긴 했지만 경쟁력 있는 결과. classification을 학습시킨 모델로 thresholding 방법을 사용하여 bounding box를 만드는 실험 -> 성능 좋음
table 2의 결과를 보면 기존의 bounding box를 annotation 하여 backpropagation 시킨 방법보다 더 좋은 성능. 따로 학습을 하지 않았는데도 성능 좋음

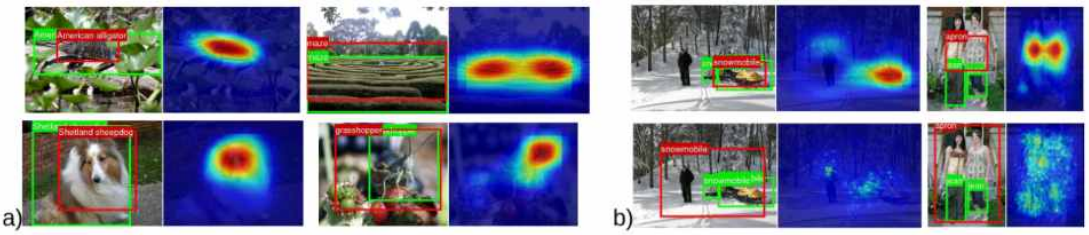


Figure 6. a) Examples of localization from GoogleNet-GAP. b) Comparison of the localization from GooleNet-GAP (upper two) and the backpropagation using AlexNet (lower two). The ground-truth boxes are in green and the predicted bounding boxes from the class activation map are in red.

Experimental results

이 외에도 논문에서는 많은 실험 결과들을 제시.
Fine-grained Recognition으로 200종의 새를 인식하는 문제에서 full image를 thresholding으로 crop하니 더 성능이 잘 나옴,
Pattern Discovery로 장면에서 informative한 물체들을 인식하는 것에서 어디를 보고 판단했는지 알 수 있다는 결과.
visual question answering에서 predictor가 문제에 대한 답을 무엇을 보고 맞혔는지 나타내는 것.

Figure 12. Examples of highlighted image regions for the predicted answer class in the visual question answering.

[EfficientNet: Rethinking Model Scaling for Convolutional Neural Network / Mingxing Tan, Auoc V.Le]

Abstract

논문의 저자는 Model Scaling 방법을 연구하고 Network의 Depth, Width, Resolution 사이의 균형이 더 좋은 성능을 이끈다는 것을 확인한다.
이러한 관점에서 입각하여, 새로운 Scaling Method 제안한다. 간단하지만 효과적인 Compound Coefficient 사용하여 Depth/Width/Resolution의 모든 Dimension 균등하게 확장한다.
Neural Network Search (NAS) 사용하고 새로운 Baseline Network 설계한다. 그리고 크기를 확장하여 같은 성질을 갖는 Networks 구축하고 EfficientNets 부른다. 해당 Networks는 이전의 ConvNets 보다 더 정확하고 효율적이다.

Introduction

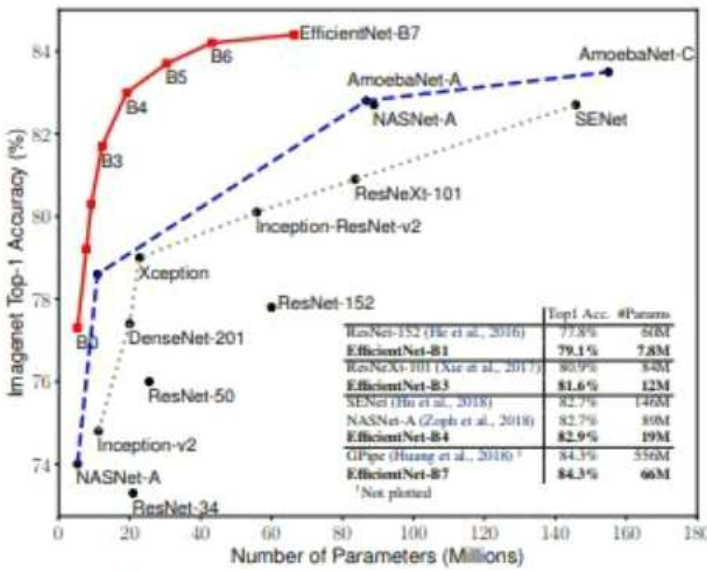


Figure 1. **Model Size vs. ImageNet Accuracy.** All numbers are for single-crop, single-model. Our EfficientNets significantly outperform other ConvNets. In particular, EfficientNet-B7 achieves new state-of-the-art 84.3% top-1 accuracy but being 8.4x smaller and 6.1x faster than GPipe. EfficientNet-B1 is 7.6x smaller and 5.7x faster than ResNet-152. Details are in Table 2 and 4.

- ConvNet의 크기를 확장하는 방법은 정확도를 달성하기 위해 사용. 그러나 이 과정은 잘 알려져 있지 않았고, 이를 위한 많은 방법들이 존재한다.

- 가장 흔한 방법은 ConvNet의 Depth 조절하는 방법. 또 다른 방법은 Iamge의 Resolution 늘리는 것.

- 마음대로 (Depth/Width/Resolution) 2개 또는 3개의 Dimension 늘리는 것은 가능하지만, 이런 방식은 반복 작업이 필요로 하고, 차선의 정확성과 효율성을 생산하는 경우가 많음.

- 저자들은 Scaling up 과정에 대해 다시 생각하였고, ConvNets 정확도와 효율성을 달성하기 위한 원칙 방법을 조사 -> 그 결과 (Width/Depth/Resolution)에 대한 Balance가 핵심이며, 간단한 Constant Ratio 통해 달성.

- Compound Scaling Method (고정된 Scaling Coefficients에 대해 균일하게 3dims 조절하는 방법)

- Efficient-B7 경우 존재하는 GPip 정확도를 능가하면서 8.4x 더 적은 Parameter 사용하며 Inference 속도가 빠름.

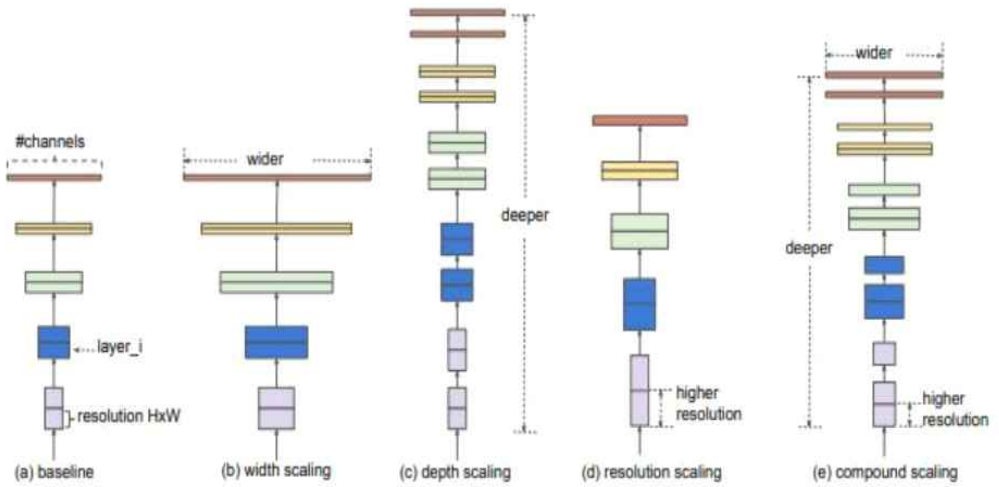


Figure 2. **Model Scaling.** (a) is a baseline network example; (b)-(d) are conventional scaling that only increases one dimension of network width, depth, or resolution. (e) is our proposed compound scaling method that uniformly scales all three dimensions with a fixed ratio.

Compound Model Scaling

다른 접근 방식으로 Scaling 문제를 공식화. 새로운 Scaling 방법을 제안.

Problem Formulation

ConvNet Layers 여러 단계로 분할되고 각 단계의 모든 Layer 동일한 Architecture 공유.

$$
\begin{aligned}
\max_{d,w,r} \quad & Accuracy(\mathcal{N}(d,w,r)) \\
s.t. \quad & \mathcal{N}(d,w,r) = \bigodot_{i=1\ldots s} \hat{\mathcal{F}}_i^{d\cdot\hat{L}_i}\left(X_{\langle r\cdot\hat{H}_i, r\cdot\hat{W}_i, w\cdot\hat{C}_i\rangle}\right) \\
& \text{Memory}(\mathcal{N}) \leq \text{target\_memory} \\
& \text{FLOPS}(\mathcal{N}) \leq \text{target\_flops}
\end{aligned}
\tag{2}
$$

가장 좋은 Layer Architecture F(x) 찾는 것을 집중하는 기존 ConvNet과 달리, Baseline Network의 F(x) 고정하고 length(L), width(C), resolution(H, W) 변경.

F(x) 고정하면서, 새로운 제약 조건에 대해 Model Scalings 방법은 간단하게 만들어 설계 문제를 해결 할 수 있지만 여전히 각 Layers 대해 많은 Design Space가 남아 있음

이러한 Design Space 줄이기 위해, 모든 Layer가 Constant Ratio에 균등하게 조정 될 수 있도록 제한.

Scaling Dimensions

위 수식에서 가장 큰 문제는 d, w, r에 대해 서로 의존적이며 다른 제한 조건에 의해 각 변수가 변경된다는 점.

Depth (d) :

Network의 Depth 조절 하는 방법은 많은 ConvNet에서 흔한 방법. 직관적으로 ConvNet이 깊어질 수록 더 많고 복잡한 Feature-map 확인 할 수 있고 Task에 대해 일반화가 잘 됨.

그러나 Network가 깊어질 수록 Vanishing Gradient 문제가 발생하고 이를 해결하기 위한 방법들이 따라옴. (Skip Connection etc)
넓은 Network는 더 많은 정제된 Feature-map 볼 수 있고 학습하기도 쉬움. 그러나 넓고 얇은 Network는 더 높은 단계의 Feature-map 확인 할 수 없음.

Resolution (r) :

높은 해상도인 Input Image에 대해, ConvNets 더 많은 정제된 Patterns
224 x 224 시작하여 현재 ConvNet은 299(331) x 299(331) 사용하는 경향.

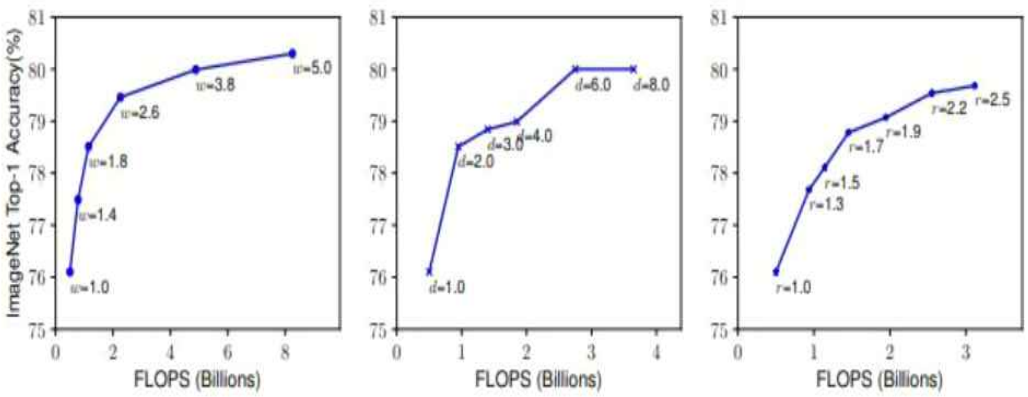


Figure 3. **Scaling Up a Baseline Model with Different Network Width (w), Depth (d), and Resolution (r) Coefficients.** Bigger networks with larger width, depth, or resolution tend to achieve higher accuracy, but the accuracy gain quickly saturate after reaching 80%, demonstrating the limitation of single dimension scaling. Baseline network is described in Table 1.

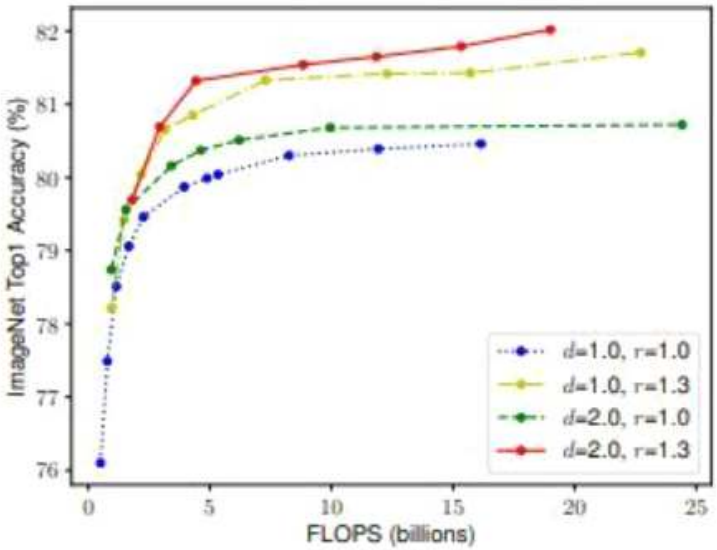


Figure 4. **Scaling Network Width for Different Baseline Networks.** Each dot in a line denotes a model with different width coefficient (w). All baseline networks are from Table 1. The first baseline network (d=1.0, r=1.0) has 18 convolutional layers with resolution 224x224, while the last baseline (d=2.0, r=1.3) has 36 layers with resolution 299x299.

Compound Scaling

직관적으로 높은 해상도의 이미지가 있다면, Network Depth 증가 시켜야 한다. 그래야 큰 Receptive Field는 큰 이미지 픽셀에 대해 비슷한 Feature 추출할 수 있다. 이와 상응하여 Network의 Width 증가 하였다. 이러한 관점들은 단일 Dimension Scaling 보다 여러 Diemension에 대한 Balance 제안한다.

Observation 2 더 좋은 정확도와 효율성을 달성하기 위해서 ConvNet Scaling 할 때, Network의 Depth, Width, Resolution의 `Balance` 중요하다.
따라서 Compound Scaling Method 제안한다. 이 방법은 Compound coefficient Φ 사용하여 Network (width/depth,resolution) 조절한다.

$$
\begin{aligned}
\text{depth: } & d = \alpha^\phi \\
\text{width: } & w = \beta^\phi \\
\text{resolution: } & r = \gamma^\phi \\
& \text{s.t. } \alpha \cdot \beta^2 \cdot \gamma^2 \approx 2 \\
& \alpha \geq 1, \beta \geq 1, \gamma \geq 1
\end{aligned}
\tag{3}
$$

α, β, γ 상수는 small grid search에 대해 결정된다. 직관적으로 Φ 는 Model Scaling에 사용할 수 있는 자원을 제어 하는 user-specified coefficient 이며, α, β, γ는 Network의 width, depth, resolution에 자원을 할당하는 방법을 말한다.

EfficientNet Architecture

Table 1. **EfficientNet-B0 baseline network** – Each row describes a stage i with $\hat{L}_i$ layers, with input resolution $\langle \hat{H}_i, \hat{W}_i \rangle$ and output channels $\hat{C}_i$. Notations are adopted from equation 2.

Stage i	Operator $\hat{\mathcal{F}}_i$	Resolution $\hat{H}_i \times \hat{W}_i$	#Channels $\hat{C}_i$	#Layers $\hat{L}_i$
1	Conv3x3	224 × 224	32	1
2	MBConv1, k3x3	112 × 112	16	1
3	MBConv6, k3x3	112 × 112	24	2
4	MBConv6, k5x5	56 × 56	40	2
5	MBConv6, k3x3	28 × 28	80	3
6	MBConv6, k5x5	14 × 14	112	3
7	MBConv6, k5x5	14 × 14	192	4
8	MBConv6, k3x3	7 × 7	320	1
9	Conv1x1 & Pooling & FC	7 × 7	1280	1

Baseline Network의 Layer Operators F(x)는 Model Scaling에서 고정되기 때문에 좋은 Baseline Network 획득하는 것이 중요.
해당 연구는 EfficientNet-B0 효율적인 Network 생산. 같은 Search Space를 탐색했기 때문에 해당 구조는 MnasNet과 비슷하지만, EfficientNet-B0가 FLOPs target 때문에 약간 더 큼.
Main Block은 Mobile Inverted Residual Bottleneck 사용하였으며, Squeeze-and-Excitation module 또한 사용.

Φ=1 고정한 상태에서 α, β, γ 찾는다. → 찾은 α, β, γ에 대해서 Φ값을 변경한다.

Conclusion

정확도와 효율성을 유지하면서 Network의 width, depth, resolution의 Balance 설정하는 방법을 연구.
Compound Scaling Method 제안, 쉽게 자원 제약 없이 Model Scaling 진행 할 수 있음. 제작한 EfficientNet Model은 효과적으로 Scaling 될 수 있음.