



EfficientNet

EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks

▼ 1. 어떤 문제를 풀고자 했는가? (Abstract)



for 모델의 정확도 ↑

1. network depth 2. width 3. resolution

→ 기존에는 3가지를 모두 수동으로 조절

→ EfficientNet은 **Compounding Scaling Method**를 이용해 균일하게 scaling 함!

- 입력이미지 크기와 깊이, 너비가 일정한 관계가 있다는 것을 찾고 수식으로 만들

▼ 2. 어떤 동기/상황/문제점에서 이 연구가 시작되었는가? (Introduction)



- 이전 연구

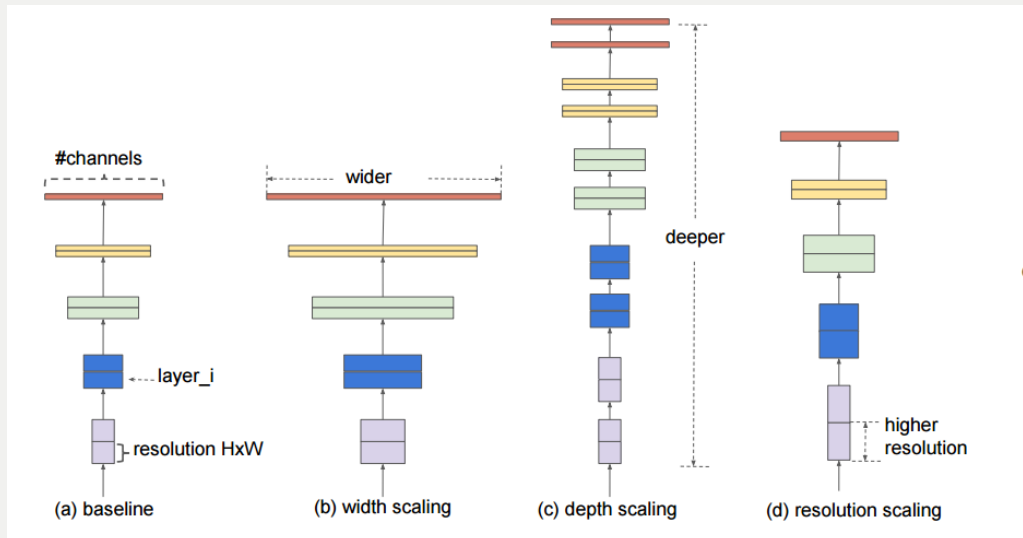
1. network depth 2. width 3. resolution 중 **하나만** scale하는 것을 다룸!

- 더 나은 정확도와 효율성으로 scale up 하는 방법?

: 3가지의 균형을 잘 맞추는게 중요하단 것을 발견!

fixed scaling coefficient로 균일하게 scale 하면 된다!

= **Compound Scaling Method**



- w,d,r이 일정값 이상이 되면 정확도가 빠르게 saturate
- 하나씩 조절하는게 아니라 함께 조절해서 최고의 효율을 찾아내는 것이 compound scaling

▼ 3. 이 연구의 접근 방법은 무엇인가?(Method)

1. Problem Formulation



H,W,C : 입력 tensor의 크기, F : Conv layer

$$\mathcal{N} = \bigodot_{i=1 \dots s} \mathcal{F}_i^{L_i}(X_{\langle H_i, W_i, C_i \rangle}) \quad (1)$$

- design space를 좁히기 위해 모든 layer를 균등하게 scaling하고 정확도를 optimize 하는 문제

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

2. Scaling Dimensions



최적의 d,w,r은 서로 연관되어 있으며, 다른 제약조건에서 값이 변함

- **Depth** : 네트워크 깊이가 증가할 수록 모델의 capacity 증가, 복잡한 feature 잡아 낼 수 있음! but gradient vanishing 문제
- **Width** : 각 레이어의 width를 키우면 정확도 높아짐 but, 계산량이 제공에 비례해 증가
- **Resolution** : 입력 이미지의 해상도를 키우면 세부적인 feature 학습할 수 있어서 정확도 높아지지만 계산량 증가

3. Compound Scaling



- 더 높은 해상도의 이미지에 대해서는 네트워크를 깊게 만들어서 더 넓은 영역에 걸쳐 있는 feature를 더 잘 잡아낼 수 있도록 하는 것이 유리!

- 더 큰 이미지일수록 세부적인 내용을 많이 담으므로, layer의 width를 증가시킬 필요 있음
- 계산량은 깊이에 비례, 너비 제공에 비례, 이미지 크기 제공에 비례!

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

- α, β, γ 은 small grid search로 결정되는 상수
- ϕ 는 주어진 연산량에 따라 사용자가 결정하는 상수

논문에서는, $\alpha \cdot \beta^2 \cdot \gamma^2 \approx 2$ $\alpha \cdot \beta^2 \cdot \gamma^2 \approx 2$ 로 맞춰서 전체 계산량은 2^ϕ 에 비례하게 잡음!

4. Architecture



[step1]

$\phi=1$ 로 고정한 뒤, resource가 두 배로 있다고 가정하고 eqn 2,3 을 이용해 α, β, γ 를 찾음

[step2]

이번엔 α, β, γ 를 고정하고 ϕ 값을 바꿔서 baseline network를 scale up 함

▼ 4. 실험은 어떻게 이루어졌는가? (Experiments)



1. Scaling up MobileNets and ResNets

- single-dimension에 대한 scaling method 보다 compound scaling method가 더 성능이 좋다!

2. ImageNet Results for EfficientNet

-더 적은 parameter와 FLOP으로 더 좋은 성능을 보임

3. Transfer Learning Results for EfficientNet

- 전이학습에서도 좋은 성능을 보임

▼ 5. 무엇을 알아냈으며, 한계점은 무엇인가? (Discussion)



Compound scaling을 적용한 모델이 object detail과 relevant region을 더 잘 캐치!

▼ 6. 결론 및 요약 (Conclusion)



- width, depth, resolution을 같이 scale up하는게 중요!
- 전이학습에서도 더 작은 parameter와 FLOP으로 잘 동작!