Paper presentation

EfficientNetV2: Smaller Models and Faster Training

Authors :
Mingxing Tan, Quoc V. Le
2021

UNIVERSITÉ CÔTE D'AZUR

Presented by Paul Peyssard
Supervised by Diane Lingrand, Michel Riveill

MSc Data Science & Artificial Intelligence
03/23/2023

Content

Introduction

- The previous Version : EfficientNet (V1) :

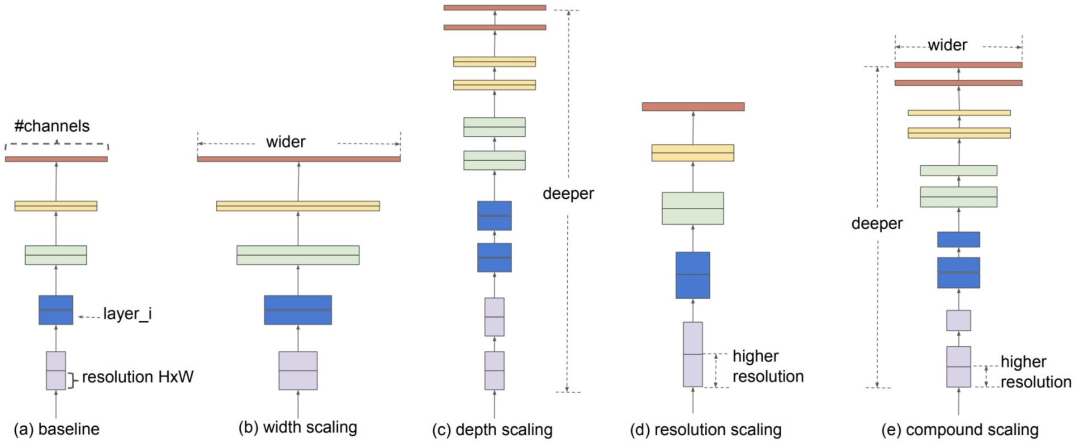


Figure 2. **Model Scaling.** (a) is a baseline network example; (b)-(d) are conventional scaling that only increases one dimension of network width, depth, or resolution. (e) is our proposed compound scaling method that uniformly scales all three dimensions with a fixed ratio.

Source : EfficientNet, https://arxiv.org/pdf/1905.11946v5.pdf

EfficientNet’s Upgrading

EfficientNetV1 Drawbacks :

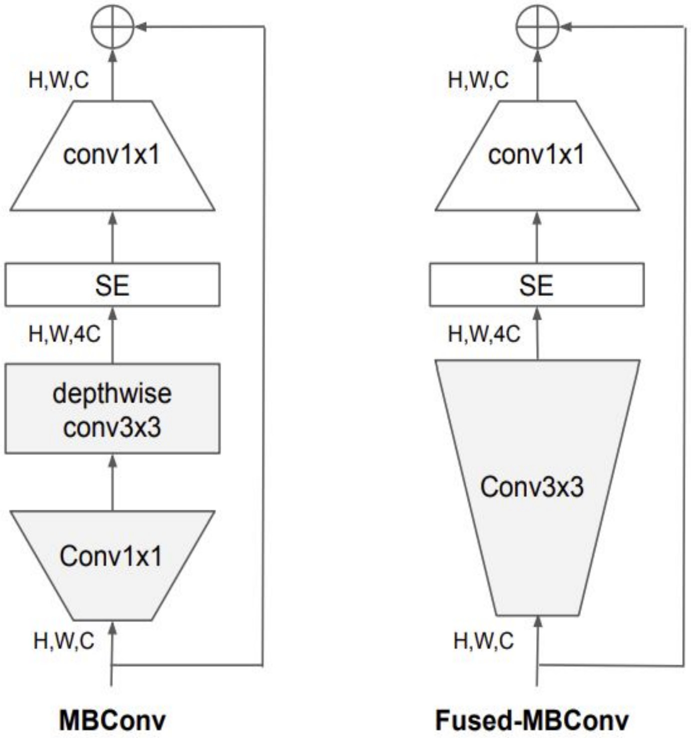
- Training with large images is slow
- Depthwise convolutions are expensive and slow in the early layers
- Compound scaling equally scales up everything

EfficientNet's Upgrading

EfficientNetV1 Drawbacks :

- Depthwise convolutions are slow

	Params (M)	FLOPs (B)	Top-1 Acc.	TPU imgs/sec/core	V100 imgs/sec/gpu
No fused	19.3	4.5	82.8%	262	155
Fused stage1-3	20.0	7.5	83.1%	362	216
Fused stage1-5	43.4	21.3	83.1%	327	223
Fused stage1-7	132.0	34.4	81.7%	254	206



Source : EfficientNetV2, https://arxiv.org/pdf/2104.00298.pdf

EfficientNet's Upgrading

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

Source : EfficientNet, https://arxiv.org/pdf/1905.11946v5.pdf

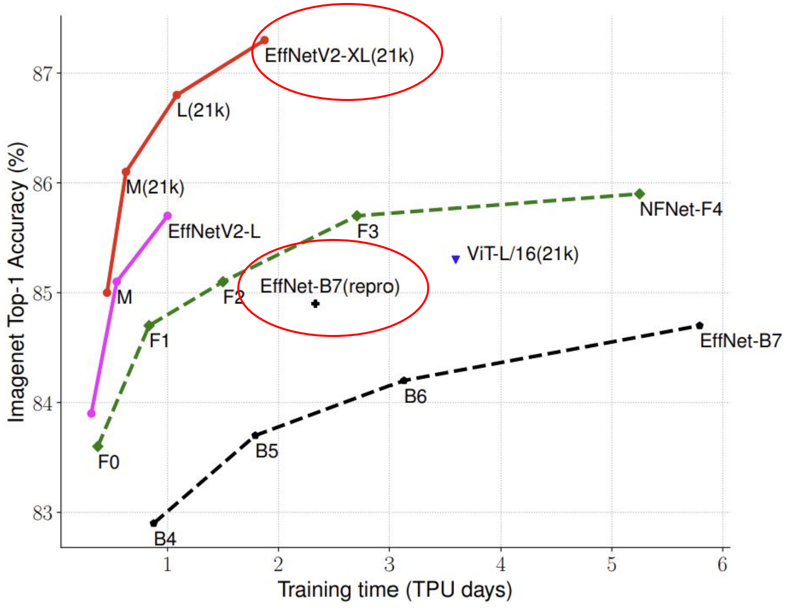
EfficientNet’s Upgrading

- New feature in EfficientNetV2 compared to previous version :
    - Progressively increase of the images size during training
    - Progressive learning : adjusting of the regularization
    - Smaller, faster model

EfficientNet’s Upgrading

Main goal of this model :

- Reduce training time
- Reduce number of parameter
- While not losing accuracy



Source : EfficientNetV2, https://arxiv.org/pdf/2104.00298.pdf

EfficientNet’s Upgrading

Less parameter and a better accuracy

	EfficientNet (2019)	ResNet-RS (2021)	DeiT/ViT (2021)	EfficientNetV2 (ours)
Top-1 Acc.	84.3%	84.0%	83.1%	83.9%
Parameters	43M	164M	86M	24M

(b) Parameter efficiency.

Source : EfficientNetV2, https://arxiv.org/pdf/2104.00298.pdf

EfficientNetV2’s Architechture Design

EfficientNetV2's Architechture Design

NAS : Training-aware neural architecture search

We need better accuracy **(A)**, less training step time **(S)** at the same time, a network with less parameters **(P)**.

$$A \cdot S^{w} \cdot P^{v}$$

w and v are experimentally determined.

The optimised ones were $w = -0.07$ and $v = -0.05$

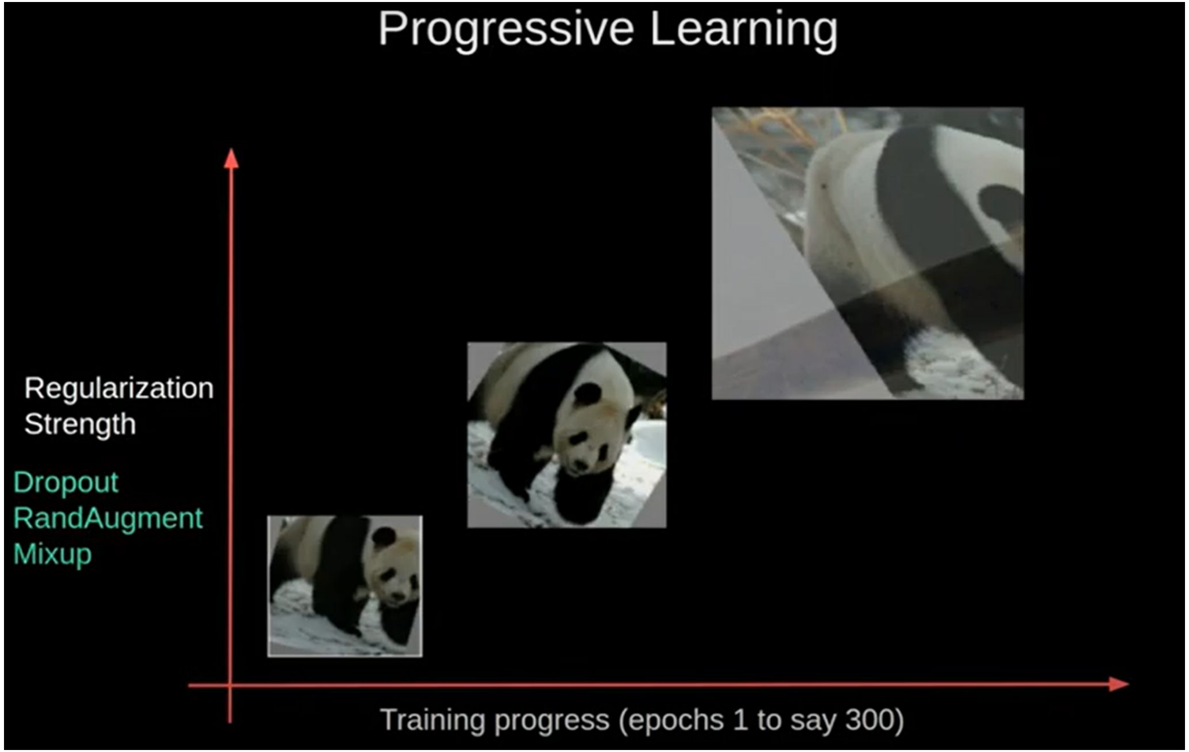
Source : EfficientNetV2, https://arxiv.org/pdf/2104.00298.pdf

EfficientNetV2's Architechture Design

Table 4. EfficientNetV2-S architecture – MBConv and Fused-MBConv blocks are described in Figure 2.

Stage	Operator	Stride	#Channels	#Layers
0	Conv3x3	2	24	1
1	Fused-MBConv1, k3x3	1	24	2
2	Fused-MBConv4, k3x3	2	48	4
3	Fused-MBConv4, k3x3	2	64	4
4	MBConv4, k3x3, SE0.25	2	128	6
5	MBConv6, k3x3, SE0.25	1	160	9
6	MBConv6, k3x3, SE0.25	2	256	15
7	Conv1x1 & Pooling & FC	-	1280	1

EfficientNetV2’s Architechture Design



Source : AI Bites, https://www.youtube.com/watch?v=qoSKbMbf1Pw&ab_channel=AIBites

EfficientNetV2's Architechture Design

Progressive Learning : Adapt image size step by step

- **Dropout** (Srivastava et al., 2014): a network-level regularization, which reduces co-adaptation by randomly dropping channels. We will adjust the dropout rate γ.
- **RandAugment** (Cubuk et al., 2020): a per-image data augmentation, with adjustable magnitude ϵ.
- **Mixup** (Zhang et al., 2018): a cross-image data augmentation. Given two images with labels $(x_i,\ y_i)$ and $(x_j,\ y_j)$, it combines them with mixup ratio λ: $\tilde{x}_i = \lambda x_j + (1-\lambda)x_i$ and $\tilde{y}_i = \lambda y_j + (1-\lambda)y_i$. We would adjust mixup ratio λ during training.

Source : EfficientNetV2, https://arxiv.org/pdf/2104.00298.pdf

EfficientNetV2's Architechture Design

Algorithm 1 Progressive learning with adaptive regularization.

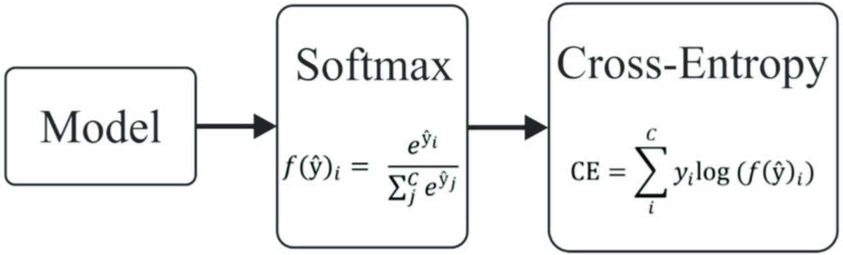
Input: Initial image size S_0 and regularization $\{\phi_0^k\}$.
Input: Final image size S_e and regularization $\{\phi_e^k\}$.
Input: Number of total training steps N and stages M.
for $i = 0$ **to** $M - 1$ **do**
  Image size: $S_i \leftarrow S_0 + (S_e - S_0) \cdot \frac{i}{M-1}$
  Regularization: $R_i \leftarrow \{\phi_i^k = \phi_0^k + (\phi_e^k - \phi_0^k) \cdot \frac{i}{M-1}\}$
  Train the model for $\frac{N}{M}$ steps with S_i and R_i.
end for

Source : EfficientNetV2, https://arxiv.org/pdf/2104.00298.pdf

EfficientNetV2’s Architechture Design

Loss function :

- Softmax Cross Entropy :



- L2 Loss :

$$L2LossFunction = \sum_{i=1}^{n} (y_{true} - y_{predicted})^2$$

Metric :

- Top-1 Accuracy

Source : https://www.researchgate.net/figure/Softmax-and-cross-entropy-loss-function_fig6_357633642

Main Results

Train dataset :

- ImageNet ILSVRC2012
- ImageNet21
- CIFAR-10
- CIFAR-100
- Flowers
- Cars

Main Results

Table 10. Comparison with the same training settings – Our new EfficientNetV2-M runs faster with less parameters.

	Acc. (%)	Params (M)	FLOPs (B)	TrainTime (h)	InferTime (ms)
V1-B7	85.0	66	38	54	170
V2-M (ours)	85.1	55 (-17%)	24 (-37%)	13 (-76%)	57 (-66%)

Source : EfficientNetV2, https://arxiv.org/pdf/2104.00298.pdf

Conclusion

Goal Achieved :

- Reduce Model Complexity
- Reduce Training Time
- Outperform previous model

Conclusion

Bibliography and related work

- Tan, M., & Le, Q. V. (2019). EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. arXiv preprint arXiv:1905.11946.

  https://arxiv.org/pdf/1905.11946.pdf
- Tan, M., & Le, Q. V. (2021). EfficientNetV2: Smaller Models and Faster Training. arXiv preprint arXiv:2104.00298.

  https://arxiv.org/pdf/2104.00298.pdf
- Gordić, A. (2021, May 26). EfficientNetV2 Explained (Smaller Models and Faster Training). YouTube.

  https://www.youtube.com/watch?v=CTsSrOKSPNo&ab_channel=AleksaGordi%C4%87-TheAIEpiphany
- Zhang, H., Cisse, M., Dauphin, Y. N., & Lopez-Paz, D. (2018). Mixup: Beyond Empirical Risk Minimization. arXiv preprint arXiv:1710.09412.

  https://arxiv.org/pdf/1710.09412.pdf
- Zoph, B., & Le, Q. V. (2016). Neural architecture search with reinforcement learning. arXiv preprint arXiv:1611.01578.

  https://arxiv.org/pdf/1611.01578.pdf

Thank you for listening !

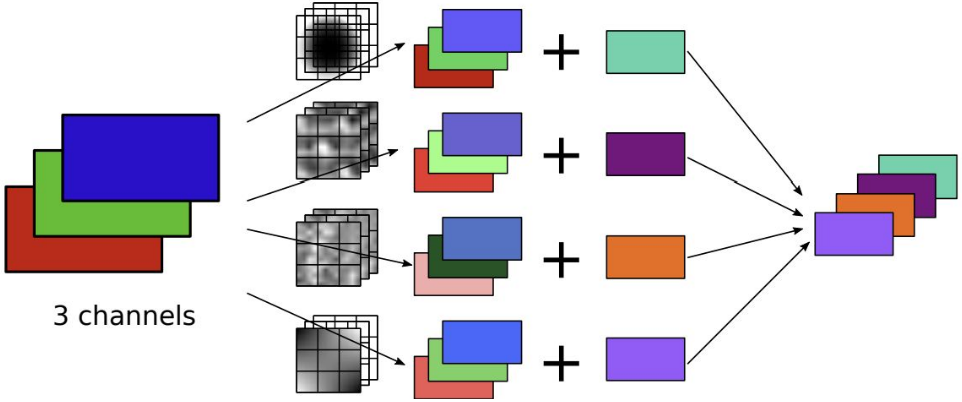
Presented by Paul Peyssard
Supervised by Diane Lingrand, Michel Riveill

Convolutional Layer :

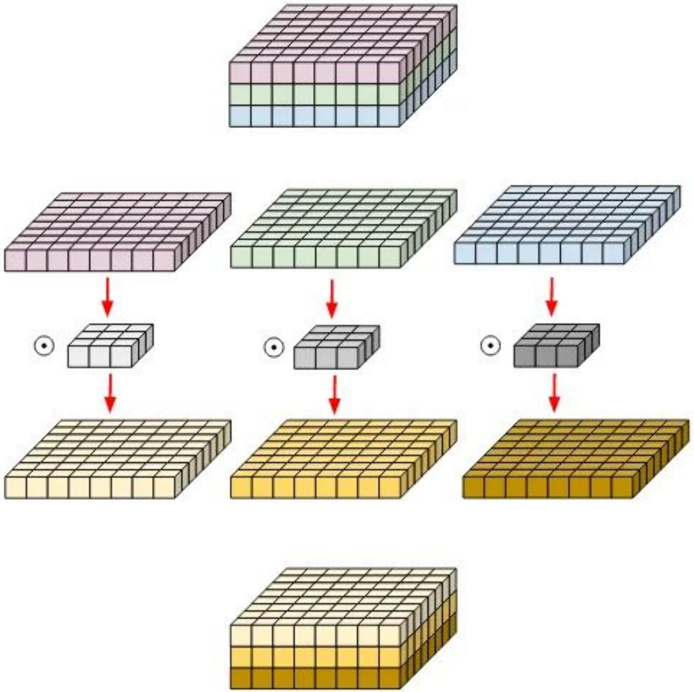
- Perform convolution operation on all the input channel with each filter

Depthwise Convolutional Layer :

- performs a convolution operation on each input channel separately using a different filter for each channel
- Does not mix information between channels
- Prevent overfitting



Diane Lingrand, CNN:Convolutionnal Neural Networks Course



https://medium.com/@zurister/depth-wise-convolution-and-depth-wise-separable-convolution-37346565d4ec

REMARQUE APRES PRESENTATION DANS LES COMMENTS :