Investigate the Architecture of EfficientNet models

Van-Thanh Hoang
Quang Binh University, Vietnam
thanhhv@qbu.edu.vn

August 05, 2023

Architecture of EfficientNet

Introduction

- **EfficientNet can be considered as a family of network models**
  - State-of-the-art accuracy on the ImageNet challenge
  - Have much fewer parameters and computation cost
  - More efficient than most of their predecessors
  - Network architecture is designed by using neural architecture search
- **Family of EfficientNet models is produced by expanding the original EfficientNet-B0 model**

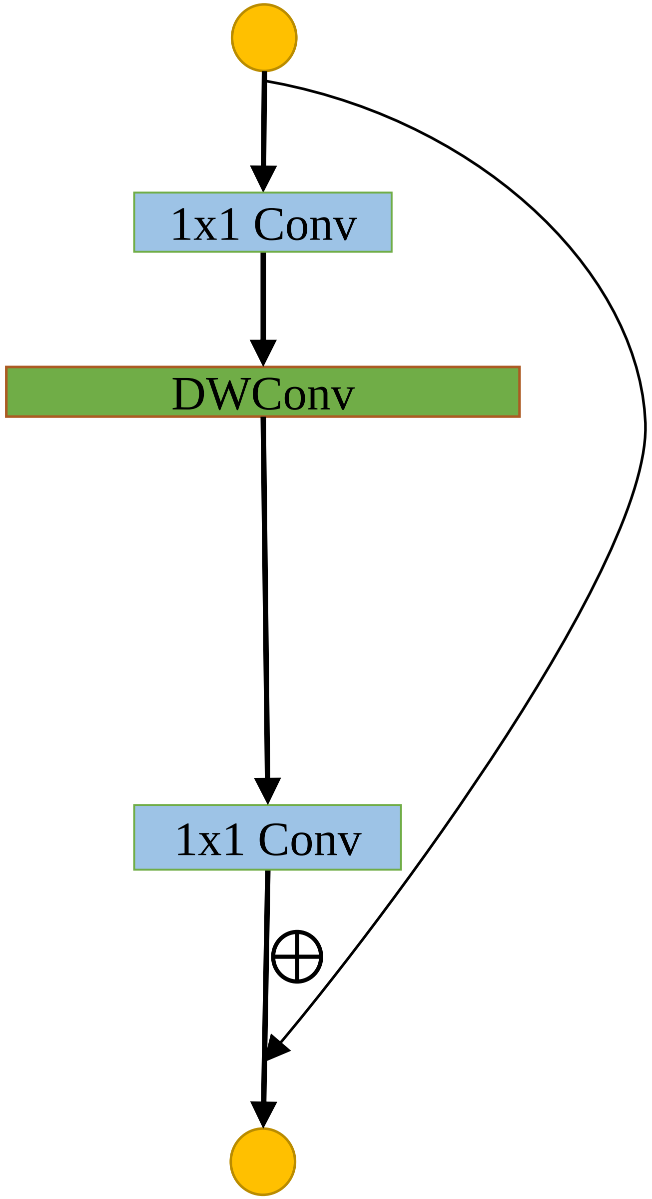
EfficientNet-B0 Architecture

- **Basic block: MBConv**

Stage i	Operator $\hat{\mathcal{F}}_i$	Resolution $\hat{H}_i \times \hat{W}_i$	#Channels $\hat{C}_i$	#Blocks $\hat{L}_i$
1	Conv3 $\times$ 3	224×224	32	1
2	MBConv1, $K = 3$	112×112	16	1
3	MBConv6, $K = 3$	112×112	24	2
4	MBConv6, $K = 5$	56×56	40	2
5	MBConv6, $K = 3$	28×28	80	3
6	MBConv6, $K = 5$	14×14	112	3
7	MBConv6, $K = 5$	14×14	192	4
8	MBConv6, $K = 3$	7×7	320	1
9	Conv1 $\times$ 1 & Pooling	7×7	1280	1
10	Fully Connected	1×1	1000	1

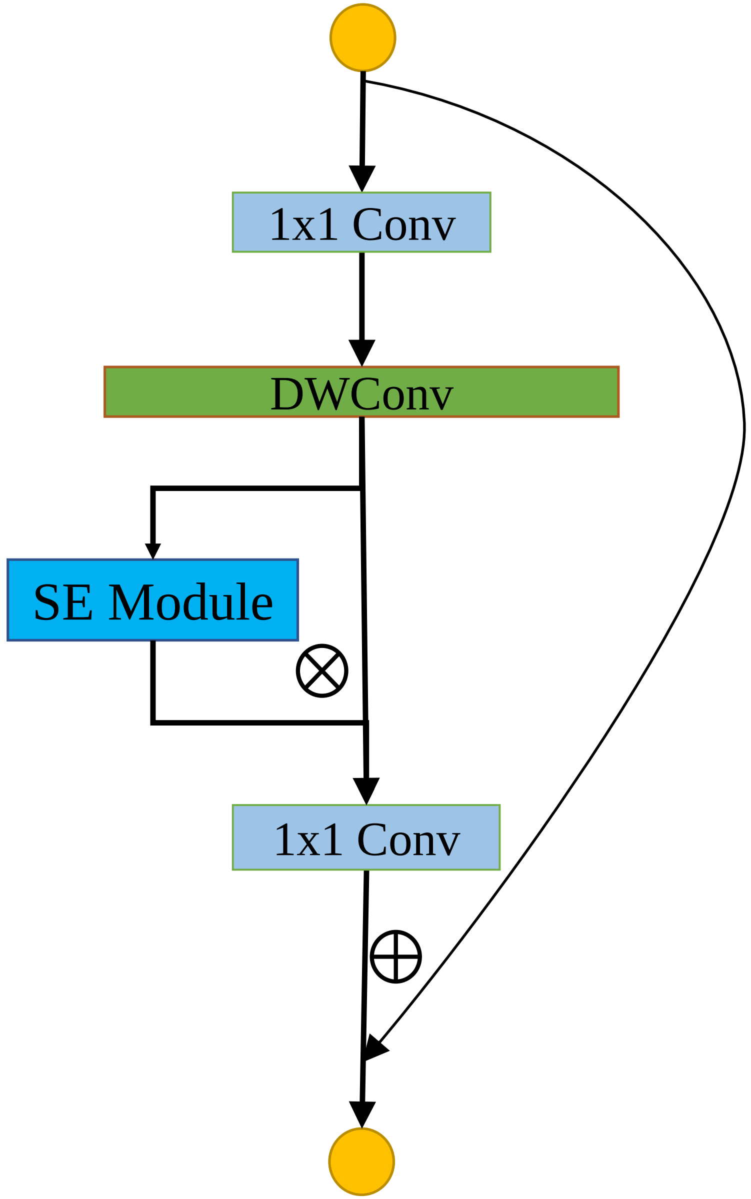
Network Architecture

- **Inverted bottleneck Residual Convolution (MBConv without SE)**
  - First layer is a 1 × 1 convolution to increase the channel dimension based on a expand ratio
  - Second layer is a depthwise Convolution of k × k
  - Last 1x1 point-by-point convolution to restore the channel dimension to the original
  - There is a skip connection if the sizes of input and output are same



Network Architecture

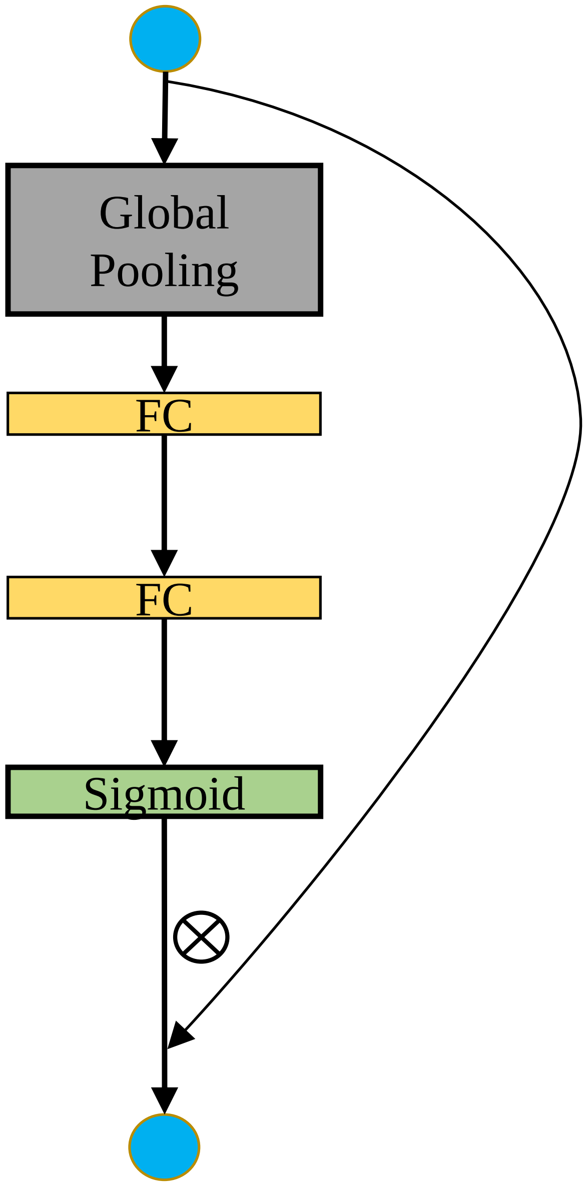
- **Mobile inverted bottleneck convolution (MBConv)**
  - First layer is a 1 × 1 convolution to increase the channel dimension based on a expand ratio
  - Second layer is a depthwise Convolution of k × k
  - Last 1x1 point-by-point convolution to restore the channel dimension to the original
  - There is a skip connection if the sizes of input and output are same
  - Add SE module right after Depthwise Convolution



- **Mobile inverted bottleneck convolution (MBConv)**
  - Has small number of parameters and FLOPS
  - Run slower than traditional convolution when performed on high resolution

Network Architecture

- **SE module**
  - Compress each channel to a single numeric value via global average pooling
  - A fully-connected layer to add the necessary nonlinearity and decrease the complexity of its output channel by a scale factor
  - Second fully-connected layer followed by Sigmoid activation for smooth gating each channel
  - Weigh each feature map in the input block based on those result via multiplication operation



Experiments

- **ImageNet dataset**
  - 1.2 million training images
  - 50,000 validating images
  - 1,000 classes

- **Implementation Details**
  - Implemented all models on TensorFlow
  - Trained on Google Colab service with TPU environment

Practical Analysis on Architecture of EfficientNet

Introduction

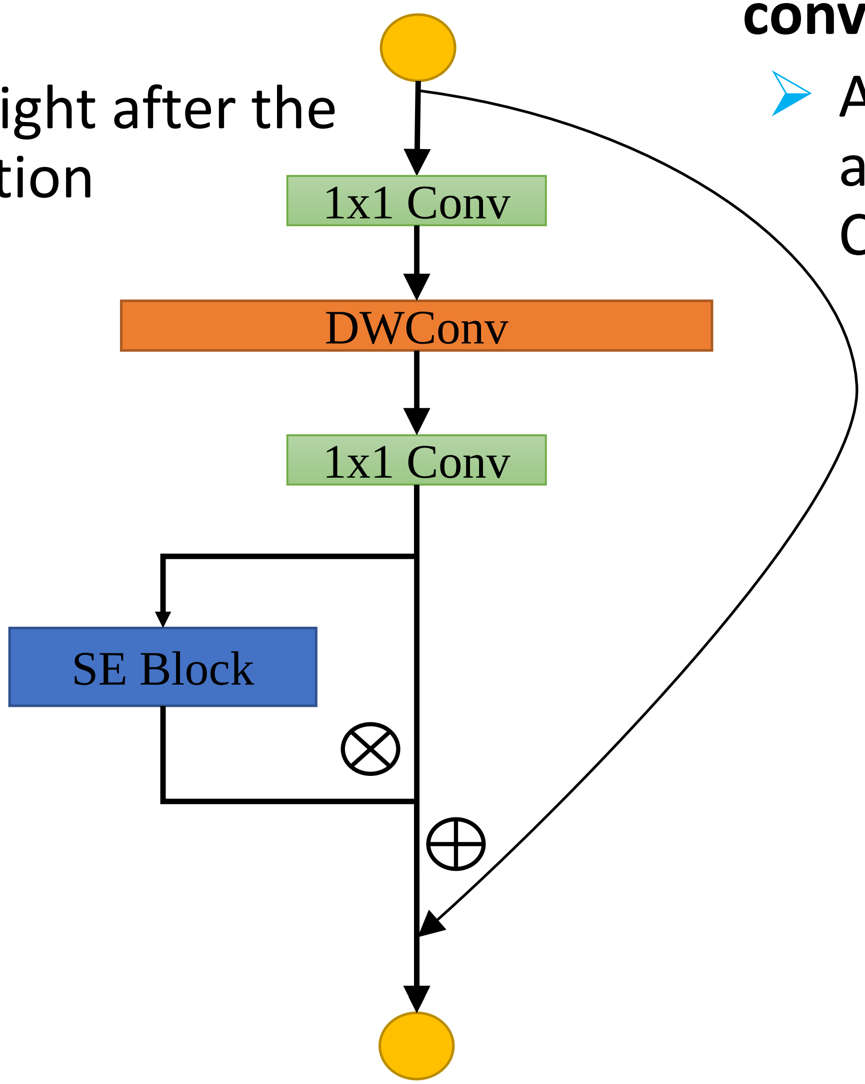
- **Contributions**

  - Create some variations of EfficientNet-B0 by repositioning/removing SE modules

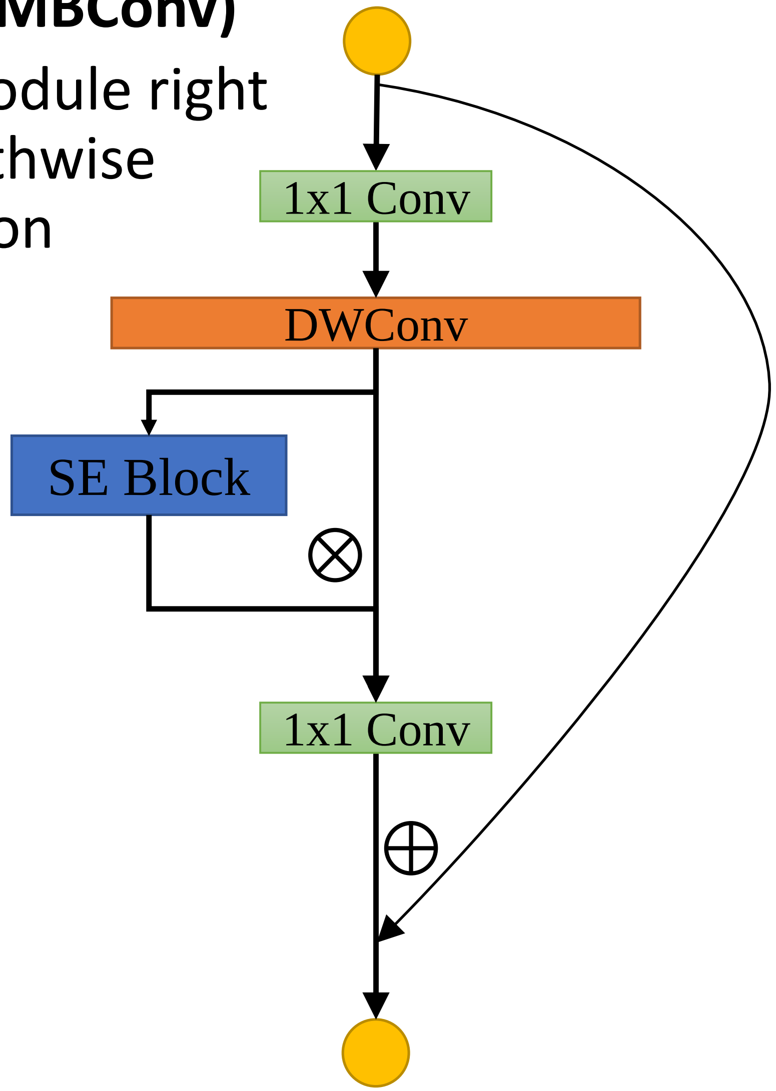
  - Evaluate on ImageNet dataset to know the effect of SE modules on the performance of EfficientNet-B0

Network Architecture

- **MBConv with SE module at the end**
  - Add SE module right after the last 1x1 Convolution



- **Mobile inverted bottleneck convolution (MBConv)**
  - Add SE module right after Depthwise Convolution



Experiments

➢ Results

Model	Block Arch.	#Params	Top-1	Top-5
EfficienNet-B0	Fig. 1b	5.29M	77.2	93.4
EfficienNet-B0-SEend@*noskip*	Fig. 1c	5.16M	77.1	93.3
EfficienNet-B0-SEend@*hasskip*		4.91M	76.5	93.2
EfficienNet-B0-SEend@*all*		4.78M	76.2	92.9
EfficienNet-B0-noSE@*noskip*	Fig. 1a	5.11M	77.1	93.4
EfficienNet-B0-noSE@*hasskip*		4.83M	76.6	93.0
EfficienNet-B0-noSE@*all*		4.65M	75.8	92.7

- *noskip: changes applied to MBConv has **no** skip-connection*
- *hasskip: changes applied to MBConv have skip-connection*
- *SEend: MBConv with SE module at the end*
- *noSE: MBConv with no SE module*

- If changes (re-positioning or removing) are on noskip-blocks, the accuracy slightly drops with just 0.1% while the number of parameters decreases at 3%
- If changes are on hasskip-blocks, the accuracy drops a little bit larger at 0.7% while the number of parameters decreases at 8%
- If all blocks are applied, the accuracy drops much at 1% and 1.4% compared to 12% drop for the number of parameters

=> repositioning SE modules to the end of the blocks is not a good idea

Experiments

Results

Model	Block Arch.	#Params	Top-1	Top-5
EfficienNet-B0	Fig. 1b	5.29M	77.2	93.4
EfficienNet-B0-SEend@*noskip*	Fig. 1c	5.16M	77.1	93.3
EfficienNet-B0-SEend@*hasskip*		4.91M	76.5	93.2
EfficienNet-B0-SEend@*all*		4.78M	76.2	92.9
EfficienNet-B0-noSE@*noskip*	Fig. 1a	5.11M	77.1	93.4
EfficienNet-B0-noSE@*hasskip*		4.83M	76.6	93.0
EfficienNet-B0-noSE@*all*		4.65M	75.8	92.7

- *noskip: changes applied to MBConv has **no** skip-connection*
- *hasskip: changes applied to MBConv have skip-connection*
- *SEend: MBConv with SE module at the end*
- *noSE: MBConv with no SE module*

- If we reposition the SE modules on only noskip-blocks or hasskip-blocks, the performance is similar to the removing case
- When the change is applied to all blocks, the variant with SE blocks has higher accuracy, 76.2% vs 75.8%

=> SE modules still can improve the accuracy for EfficientNet-B0, but we don’t need to use SE modules for all MBConv blocks.

Conclusion

- **Investigate the affect of SE module to EfficienNet-B0**

  - Repositioning SE module to the end of the block is not a good idea

  - Don’t need to use SE module for all MBConv blocks

=> there is a trade-off need to be further studied to have a more efficient network

Rethinking Mobile Inverted Bottleneck Convolution for EfficientNet

Introduction

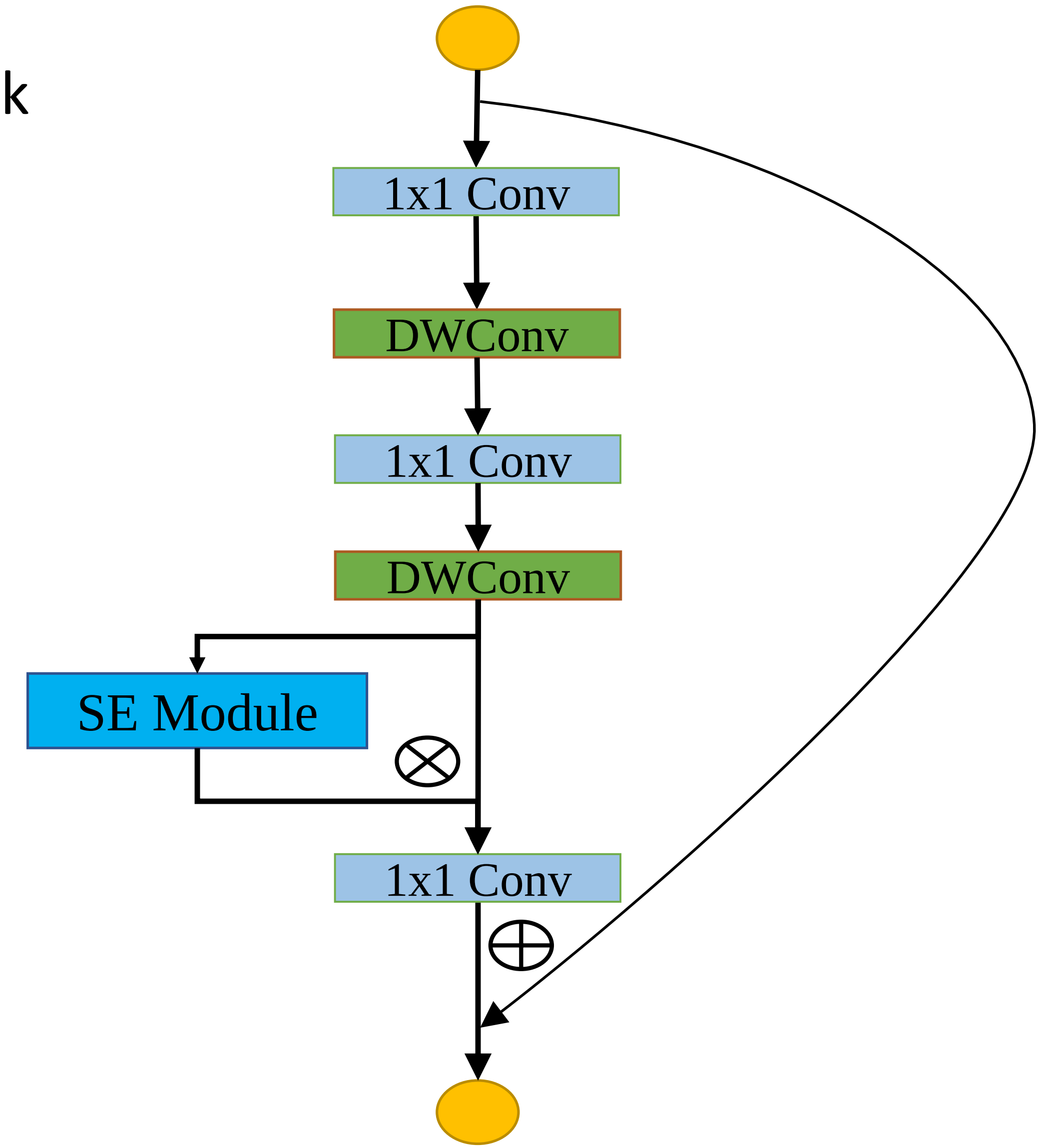
- **Contributions**

  - Introduce a new module called Mobile Equal channel Convolution

  - Proposed network has higher accuracy on ImageNet dataset while still has fewer number of parameters

Network Architecture

- **Mobile Equal channel Convolution**
  - It consists of three 1 × 1 convolution and two k × k depthwise convolution
  - All convolution layers have a same number of channels
  - The SE module after the second Depth-wise convolution layer



Experiments

- **Ablation Study on expanding in blocks with stride=2**

Model	#Params	Top-1 (%)	Top-5 (%)
Efficienet-B0 [19]	5.28M	77.21	93.36
Proposed network ($e = 1$)	4.62M	75.11	92.11
Proposed network ($e = 2$)	4.81M	75.90	92.86
Proposed network ($e = 3$)	5.00M	76.48	93.17
Proposed network ($e = 4$)	5.19M	77.03	93.54
Proposed network ($e = 5$)	5.37M	77.22	93.51
Proposed network ($e = 6$)	5.56M	77.49	93.67

Experiments

➢ **Ablation Study on expanding in blocks with changes in sizes**

Model	#Params	Top-1 (%)	Top-5 (%)
Efficienet-B0 [19]	5.28M	77.21	93.36
Proposed network ($e = 1$)	4.08M	75.04	92.34
Proposed network ($e = 2$)	4.35M	76.36	93.14
Proposed network ($e = 3$)	4.62M	77.06	93.49
Proposed network ($e = 4$)	4.89M	77.64	93.66
Proposed network ($e = 5$)	5.16M	77.79	93.79

Experiments

➢ **Ablation Study on expanding in blocks with change in sizes and expand-factor based on the number of output channels**

Model	#Params	Top-1 (%)	Top-5 (%)
Efficienet-B0 [19]	5.28M	77.21	93.36
Proposed network ($e = 1$)	4.24M	76.13	92.91
Proposed network ($e = 1.5$)	4.46M	76.65	93.25
Proposed network ($e = 2$)	4.67M	77.26	93.54
Proposed network ($e = 2.5$)	4.89M	77.59	93.67
Proposed network ($e = 3$)	5.10M	77.84	93.72
Proposed network ($e = 3.5$)	5.31M	78.10	93.99

Experiments

- **Performance Evaluation**
  - Proposed network can outperform EfficientNet-B0 and other models
  - It has higher accuracy than EfficientNet-B0 while having a similar number of parameters

Model	#Params	Top-1 (%)	Top-5 (%)
MobileNet-0.75[5]	2.59M	68.40	89.49
MobileNetV2-0.75[15]	2.63M	69.19	88.74
DenseNet 1.5× [8]	-	60.10	-
Xception 1.5× [1]	-	70.60	-
CondenseNet (G=C=8) [7]	-	71.00	-
ShuffleNetV1 1.5× (g=3) [22]	-	71.50	-
ShuffleNetV2 1.5×[11]	3.50M	72.60	90.32
MobileNet-1.0[5]	4.23M	70.60	91.29
MobileNetV2-1.0[15]	3.50M	72.00	90.10
DenseNet 2× [8]	-	65.40	-
Xception 2× [1]	-	72.40	-
CondenseNet (G=C=8) [7]	-	73.80	-
ShuffleNetV1 2× (g=3) [22]	-	73.70	-
PeleeNet [20]	2.8 M	72.60	90.60
ShuffleNetV2 2×[11]	7.39M	74.90	91.86
Efficienet-B0 [19]	5.28M	77.21	93.36
Proposed network ($e = 3$)	5.10M	77.84	93.72
Proposed network ($e = 3.5$)	5.31M	78.10	93.99

Conclusion

- **Proposed new kind of block for EfficientNet called Mobile Equal channel Convolution**

- **The new block has a same number of channels for all convolution layers inside to make the module more balance**

- **The experiments show that the new variant can have higher accuracy with lower number of parameters**

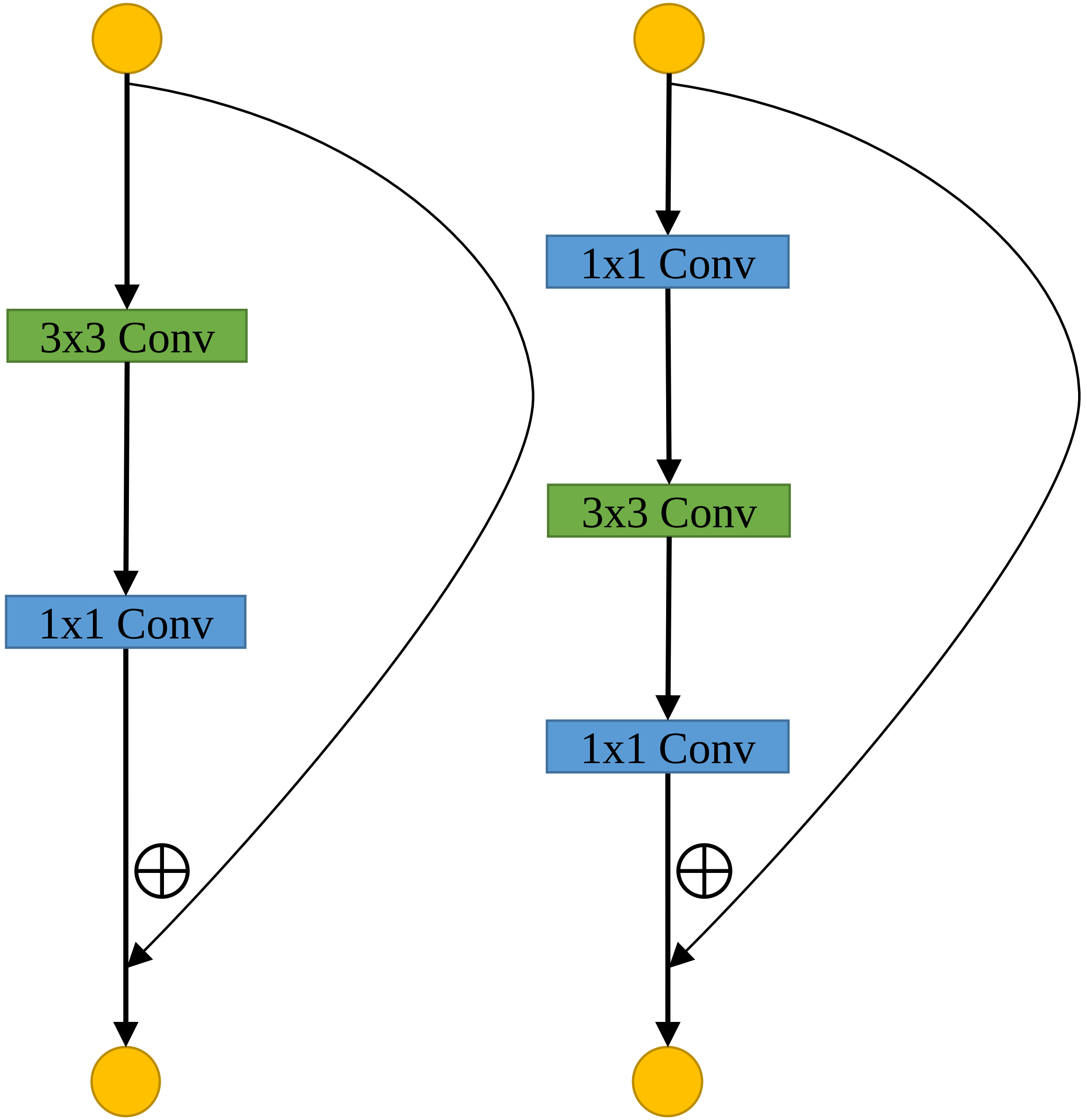
A Compact version of EfficientNet

Introduction

- **Contributions**
    - Proposed a compact version of EfficientNet that has similar accuracy on the ImageNet dataset but runs faster
    - This variation is more friendly for mobile devices

Convolution Blocks Architecture

- **Conv31 block**
  - Consist of 2 traditional convolution layer:
    - 3x3 convolution layer
    - 1x1 convolution layer
- **Conv131 block**
  - Consist of 3 traditional convolution layer:
    - 1x1 convolution layer
    - 3x3 convolution layer
    - 1x1 convolution layer



Network Architecture

- **Modifications**
  - Remove the first MBConv1 block which has resolution of 112 × 112
  - Change the second MBConv6 block that has a resolution of 112×112 to the Conv31 block

Stage i	Resolution $\hat{H}_i \times \hat{W}_i$	#Channels $\hat{C}_i$	Operator $\hat{\mathcal{F}}_i$	#Blocks $\hat{L}_i$
1	224×224	32	Conv3x3	1
2	112×112	16	MBConv1, k3x3	1
3	112×112	24	MBConv6, k3x3	2
4	56×56	40	MBConv6, k5x5	2
5	28×28	80	MBConv6, k3x3	3
6	14×14	112	MBConv6, k5x5	3
7	14×14	192	MBConv6, k5x5	4
8	7×7	320	MBConv6, k3x3	1
9	7×7	1280	Conv1x1 & Pooling	1
10	1×1	1000	FC	1

EfficientNet-B0

Stage i	Resolution $\hat{H}_i \times \hat{W}_i$	#Channels $\hat{C}_i$	Operator $\hat{\mathcal{F}}_i$	#Blocks $\hat{L}_i$
1	224×224	24	Conv3x3	1
2	112×112	24	MBConv3, k3x3	1
3	112×112	24	Conv31, k3x3	1
4	56×56	40	MBConv6, k5x5	2
5	28×28	80	MBConv6, k3x3	3
6	14×14	112	MBConv6, k5x5	3
7	14×14	192	MBConv6, k5x5	4
8	7×7	320	MBConv6, k3x3	1
9	7×7	1280	Conv1x1 & Pooling	1
10	1×1	1000	FC	1

Proposed Network

Experiments

- **Performance Evaluation**
  - The lite and compact versions have a slightly higher error compared to the original EfficientNet-B0 they can run 2 times faster.
  - The compact version can achieve performance similar to that of the lite version while running 10% faster

Model	#Params (10^6)	#FLOPS (10^6)	Top-1 (%)	Top-5 (%)	Speed (ms)
EfficientNet-B0	5.61	391.71	24.86	7.81	53.94
EfficientNet-B0-lite	4.97	385.69	26.75	8.72	37.69
EfficientNet-B0-compact	4.97	387.10	26.95	9.01	31.17

Experiments

- **Ablation study on stage 2**
  - When remove the stage from the original, it runs fastest while the error is similar to the original

Architecture of stage 2	#Params (10^6)	#FLOPS (10^6)	Top-1 (%)	Top-5 (%)	Speed (ms)
MBConv1	4.97	385.69	26.75	8.72	37.69
DWConv3×3	4.97	398.54	26.68	8.64	37.39
Conv3×3	4.98	433.32	26.68	8.71	36.20
Conv1×1	4.97	381.94	27.21	8.92	35.93
Remove	4.97	394.78	26.78	8.87	34.39

- **Ablation study on stage 3**
  - When we adopt the Conv31 architecture, the accuracy does not change much, but the speed of the model can be 10% faster

Architecture of stage 3	#Params (10^6)	#FLOPS (10^6)	Top-1 (%)	Top-5 (%)	Speed (ms)
MBConv1	4.97	394.78	26.78	8.87	34.39
Conv131	4.97	388.91	27.11	8.95	31.64
Conv31	4.97	387.10	26.95	9.01	31.17

Conclusion

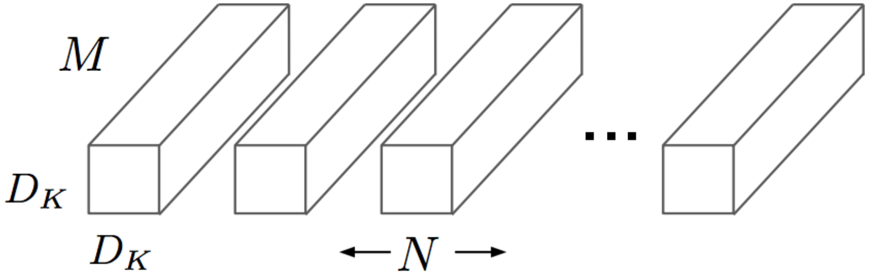
- **Proposed the EfficientNet-B0-compact model**
  - Remove the first MBCon1 block
  - Change the second MBConv block to a Conv31 block.

- **The proposed model has a similar accuracy but is faster than the original EfficientNet-B0 and EfficientNet-B0-lite model**

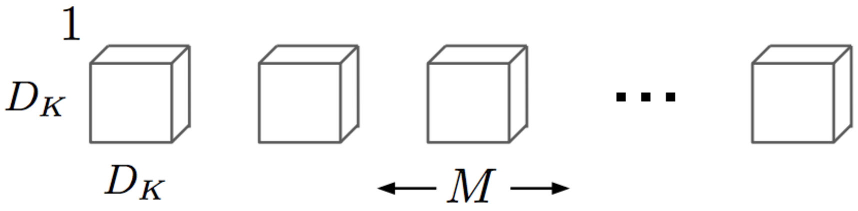
Thank you for attention!

Introduction

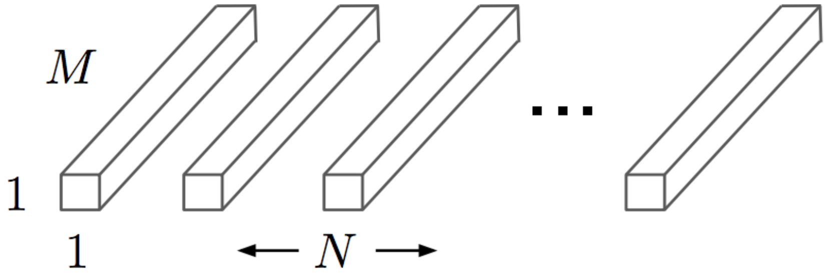
- **Standard Convolution can be replaced by Depthwise Separable Convolution consists of:**
  - Depthwise Convolution (DWConv): a spatial convolution performed independently over each channel to obtain spatial information
  - Pointwise Convolution: a 1x1 convolution, to obtain cross-channel information



a) Standard Convolution Kernels



b) Depthwise Convolution Kernels



c) Pointwise Convolution Kernels

Image source: Howard et al. “MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications”